

高性能电子商务 平台构建

架构、设计与开发

ShopNC产品部 著

Building High Performance E-Commerce Platform
Architecture, Design and Development

- 中国最领先的电商系统解决方案提供商ShopNC官方核心研发团队撰写
- 宏观介绍电商平台技术架构，详细讲解各种套件的模块构成、设计思路和代码实现，以及电商平台的界面设计、用户体验，为构建安全、可靠、易维护、高性能电商平台提供完整技术实现和解决方案

Web开发技术丛书

高性能电子商务平台构建：架构、设计与开发

ShopNC产品部 著

ISBN: 978-7-111-48564-3

本书纸版由机械工业出版社于2015年出版，电子版由华章分社（北京华章图文信息技术有限公司，北京奥维博世图书发行有限公司）全球范围内制作与发行。

版权所有，侵权必究

客服热线：+ 86-10-68995265

客服信箱：service@bbbvip.com

官方网址：www.hzmedia.com.cn

新浪微博 @研发书局

腾讯微博 @yanfabook

目录

推荐序一

推荐序二

前言

第1章 电商平台标准化套件

1.1 商城系统

1.2 圈子

1.3 CMS

1.4 IM

1.5 移动端

1.6 本章小结

第2章 电商平台的技术架构

2.1 操作系统

2.2 应用服务器

2.3 负载均衡

2.3.1 F5

2.3.2 LVS

2.3.3 Nginx

2.3.4 HAProxy

2.4 缓存

2.4.1 客户端缓存

2.4.2 CDN加速

2.4.3 静态文件缓存

2.4.4 数据缓存

2.5 数据存储

2.5.1 关系型数据库

2.5.2 内存型数据库

2.5.3 分布式数据库

2.5.4 MySQL可扩展方案

2.6 文件存储

2.7 消息队列

2.8 搜索设计

2.9 开发框架

2.10 本章小结

第3章 商城套件的设计与实现

3.1 会员模块

3.1.1 模块构成

3.1.2 设计思路

3.1.3 代码实现

3.1.4 功能体验

3.1.5 开发和使用

3.2 商品模块

3.2.1 模块构成

3.2.2 设计思路

3.2.3 代码实现

3.2.4 功能体验

3.3 促销模块

3.3.1 模块构成

3.3.2 设计思路

3.3.3 代码实现

3.3.4 功能体验

3.3.5 开发和使用

3.4 购物车模块

3.4.1 模块构成

3.4.2 设计思路

3.4.3 代码实现

3.4.4 功能体验

3.5 配送模块

3.5.1 模块构成

3.5.2 设计思路

3.5.3 功能实现

3.5.4 物流跟踪

3.6 订单模块

3.6.1 设计思路

3.6.2 代码实现

3.6.3 功能体验

3.7 支付接口

3.7.1 接入原理

3.7.2 设计思路

3.7.3 代码实现

3.8 退单模块

3.8.1 模块构成

3.8.2 设计思路

3.8.3 代码实现

3.8.4 功能体验

3.8.5 开发技巧

3.9 结算模块

3.9.1 设计思路

3.9.2 代码实现

3.9.3 功能体验

3.10 统计模块

3.10.1 模块构成

3.10.2 设计思路

3.10.3 代码实现

3.10.4 功能体验

3.10.5 开发和使用

3.11 预存款

3.11.1 设计思路

3.11.2 代码实现

3.11.3 功能体验

3.12 本章小结

第4章 CMS套件的设计与实现

4.1 CMS文章

4.1.1 模块构成

4.1.2 设计思路

4.1.3 代码实现

4.1.4 功能体验

4.2 CMS画报

4.2.1 模块构成

4.2.2 设计思路

4.2.3 代码实现

4.2.4 功能体验

4.3 CMS专题

4.3.1 模块构成

4.3.2 设计思路

4.3.3 代码实现

4.3.4 功能体验

4.4 CMS首页编辑

4.4.1 模块构成

4.4.2 设计思路

4.4.3 代码实现

4.4.4 功能体验

4.5 其他功能

4.5.1 评论

4.5.2 心情

4.5.3 分享

4.5.4 导航管理

4.6 本章小结

第5章 社交平台的设计与实现

5.1 SNS模块

5.1.1 模块构成

5.1.2 设计思路

5.1.3 代码实现

5.1.4 功能体验

5.2 圈子（讨论组）套件

5.2.1 圈子特点

5.2.2 设计思路

5.2.3 代码实现

5.2.4 功能体验

5.3 本章小结

第6章 IM套件的设计与实现

6.1 模块构成

6.2 设计思路

6.3 代码实现

6.3.1 Node服务器端

6.3.2 PHP服务器端

6.3.3 页面客户端

6.4 功能体验

6.5 开发技巧

6.6 本章小结

第7章 移动应用套件的设计与实现

7.1 设计思路

7.1.1 产品设计

7.1.2 设计原则

7.2 代码实现

7.2.1 框架结构介绍

7.2.2 CrashHandler工具类

7.2.3 DbHelper工具类

7.2.4 HttpHelper工具类

7.2.5 ImageHelper工具类

7.2.6 SystemHelper工具类

7.3 相关工具及第三方框架介绍

7.3.1 原型设计工具介绍

7.3.2 AndroidAnnotations介绍

7.3.3 轻量级ORM框架ActiveAndroid介绍

7.4 本章小结

第8章 020套件的设计与实现

8.1 020模式概述

8.1.1 020模式的产生背景

8.1.2 020模式的特点

8.1.3 020模式的优势与核心

8.2 产品设计思路

8.3 商户模块的设计与实现

8.3.1 模块构成

8.3.2 设计思路

8.3.3 代码实现

8.3.4 功能体验

8.4 团购模块设计

8.4.1 模块构成

8.4.2 设计思路

8.4.3 代码实现

8.4.4 功能体验

8.5 优惠券模块设计

8.5.1 模块构成

8.5.2 设计思路

8.5.3 代码实现

8.5.4 功能体验

8.6 结算模块设计

8.6.1 模块构成

8.6.2 设计思路

8.6.3 代码实现

8.6.4 功能体验

8.7 本章小结

第9章 电商平台界面设计与用户体验

9.1 网页设计的那些事儿

9.1.1 视觉设计、交互设计、前端开发

9.1.2 为代码增加可读性注释

9.1.3 语义化的HTML5标签使用

9.1.4 base.css会减轻你的工作负担

9.1.5 如何给CSS命名

9.1.6 应该用id还是class

9.1.7 能少用就少用CSS Hack

9.1.8 一些CSS和浏览器之间的怪异显示及对应的解决办法

9.2 商城首页

9.2.1 B2B2C商城首页设计元素

9.2.2 色彩风格、导航、顶部的统一化

9.2.3 焦点广告区域的制作

9.2.4 内容块的制作与可视化编辑理念

9.3 商品详情页面

9.3.1 商品页面布局方式

9.3.2 针对搜索引擎的页面优化制作

9.4 购物车页面

9.4.1 购物车页面设计特征

9.4.2 根据页面需要选择HTML标签的技巧

9.5 订单页面

9.5.1 购物详单设计理念

9.5.2 订单中要有哪些内容

9.5.3 状态步骤的设计与制作

9.6 本章小结

第10章 电商平台的安装与部署

10.1 Web服务器部署

10.2 数据库部署

10.3 电商系统安装

10.3.1 基本安装

10.3.2 设置定时任务

10.3.3 开启伪静态

10.3.4 设置域名

10.4 图片存储部署

10.5 缓存服务器部署

10.6 搜索服务器部署

10.7 IM服务器部署

10.8 本章小结

附录A 移动端接口文档

附录B ShopNC B2B2C商城功能要点

ShopNC成员简介

推荐序一

互联网特别是电子商务的发展，让我们的生活有了太多的变化。每天看到快递员穿梭于大街小巷，各种包裹堆积在各个大厦的门口，公司的前台甚至都变成邮局的柜台了，我们感受到商品的流通方式发生了翻天覆地的变化。参加过太多的传统零售行业的会议，每次必有的一个话题就是如何做电商，甚至有人说“做电商找死，不做电商等死”。越来越多的企业将线下店转型为体验店，把电商当成一个新的渠道，以求建立新的商业模式；就连每个城市原本很强大的传统百货商场，都只能走向综合体，越来越少见到单纯卖货的商城了。这一切的变化都表明了电商已经改变了原有的商业模式。

电子商务带来的变化让很多企业不知所措，究其原因是不了解互联网，不知道这里的游戏规则，甚至

在某种程度上感觉电商平台还是一个神秘的科技，不知道怎么切入电子商务。实际上，电商平台早已经是大众化的产品，从广泛的层面来说，无非是找一个第三方聚合平台去开一个店，或者是做一个自己的电商平台，两种方式各有优劣。在聚合平台上开一个店，优势是不用去管商城产品，劣势是顾客永远是别人的；自己打造一个电商平台，虽然需要投入一定的资金和精力去维护，但是逐渐建立自己的用户群体，做出自己的平台口碑，更加是一种有理想和抱负的做法。这本由ShopNC撰写的书，正是帮助企业打造一个自己的电商平台。

ShopNC给企业提供商城产品已经有些年头了，一路走来，风风雨雨，伴随着中国电子商务的发展，也在不断壮大。这本书以ShopNC的B2B2C产品为例，进行了全面详细的讲解，对电商平台的前期规划、中期开发和后期的运维都有涉及，相信很多电商企业的决策

人员、运营人员及技术人员都能从中受益。ShopNC把自己积累的经验无偿地释放出来，对电商的发展的确是一种很大的贡献。我本人也和ShopNC的创始人“白菜”相识很久，看到他在这条道路上的坚持和执着，非常钦佩，他不是一个生意人，更像一个“程序猿”，只对给企业提供优质的产品感兴趣。因此这次出版这本书，我没有理由不力荐此书，期望更多的企业及人员能够从中受益。

耿志军

原腾讯微生活总经理，现主管微信支付推广与运营业务

推荐序二

看过这本书后，给我的第一个感觉就是非常接地气，实实在在的代码在那摆着，讲得都是干货。

现在电子商务发展得如火如荼，甚至有点疯狂，网购成为人们生活中不可或缺的一部分，刚刚过去的双11促销各电商平台破记录的销量就是例证，在这种情况下，越来越多的企业会加速进入电子商务，开始自建电商平台或入驻成熟的平台。目前大型的综合性电商平台基本上布局已定了，大众经常会去的淘宝、天猫、京东等都属于这一类的，大平台的品类相对齐全，但做得不够细致，产品质量参差不齐，退货率也越来越高，这就还需要大量的行业性的、专业性的，甚至是地方性的电商平台涌现出来细分这些行业，这些平台建起来需要什么？对于技术支持，这本书的推出恰逢其时。

我是技术出身，之前也看过一些WEB技术之类的书籍，但基本上讲的都是纯技术的内容，要么就是纯语言讲解，典型代表就是PHP了吧，要么就是偏向于大数据、集群、高性能等，这类书现在有很多了。电商方面的书基本上是偏实际运营操作的，跟技术无关。目前没有一本系统介绍电商平台搭建的书籍，但这本书做到了。它从一个企业的需求出发，从技术选型到代码开发再到部署都讲到了，特别是它的代码开发部分，电商平台的几块核心功能都有，并且不是流水账，从设计思路到最后实现都有，这一点是非常重要的，也是非常值得肯定的。据我所知目前还没有类似的书籍，讲得实在，有真实的代码，而且这些代码也是经过大量正在运营的网站证明的，这不是DEMO，这的确是本书一个亮点。

我了解ShopNC也有很多年了，一直在专注电商开发，很佩服“白菜”的坚持，专注才会专业，因此推

荐产品经理、技术人员都看一下这本书，不见得百分之百认同里面的思路、方法，但如果有一点的感触或收获，那就值得。

杨立东

原暴风影音CTO

前言

为什么写这本书

中国互联网络信息中心（CNNIC）2014年1月发布的《中国互联网络发展状况统计报告》中的数据表明：截至2013年12月，我国网民规模达6.18亿，互联网普及率为45.8%。网络购物用户规模持续增长，商务类应用继续保持较高的发展速度。2013年，中国网络购物用户规模达3.02亿人，使用率达到48.9%，相比2012年增长6.0个百分点。从消费者行为模式来看，搜索行为直接指向购买，电子商务平台正是购买行为的发生场所，并且由于营销推广成本有限，促使中小企业更倾向于选择投入可控、性价比较高的方式。

以上的统计数据向我们传递了一个强烈的信号，电子商务（简称“电商”）正在改变商业的经营模式

和人们的购物习惯。国内已经有越来越多的传统行业进入了电子商务领域，同时也有越来越多的网站转型为电子商务平台，随之而来的就是市场对电商的IT服务需求的迅速增长，而目前主要电商IT服务供应商服务水平、产品质量、设计理念参差不齐，有些产品更是一直处理停滞状态，显然它们无法满足迅速发展的电商行业。

ShopNC (www.shopnc.net) 一直专注于电商整体解决方案，拥有国内领先的技术人才、开发团队及资深的行业解决方案专家，自主研发出了成熟的B2B2C模式电商平台系统、C2C模式多店商城系统、O2O本地生活类电商系统、CMS文章资讯、社交圈子社区程序、微商城分享互动程序、在线IM聊天程序以及CRM客户管理系统、移动手机平台应用和微信商城等最流行、全面、完善的电商产品综合服务体系。ShopNC不断将电

商行业的新需求、新趋势融入到产品中，与时俱进，始终保持产品的先进性。

本书结合ShopNC商城系统，从构建大数据、高性能的电平台出发，详细介绍了电商平台核心模块的设计与实现，为大家提供安全、可靠、易维护、高性能的电商平台解决方案。

本书的主要内容和特色

全书分为四个部分。第一部分（第1章）是功能规划，依次介绍了电子商务平台标准的构成套件以及各个套件的功能组成部分。第二部分（第2章）是技术架构规划，规划实现第一部分功能需要用到的技术点及实现方法，搭建编写代码前需要的技术框架。第三部分（第3~9章）是功能实现，细致而全面地讲解了构成电子商务平台的各个功能模块的设计思路、技巧分享、重点代码分析、最终功能体验，完成了从理论到

实践的完美转换。第四部分（第10章）是安装部署，将已制作完成的商城部署至服务器，并介绍了集群部署中主要节点的部署过程。以上四部分是电子商务平台从规划，再到各个重点模块构建，再到部署及性能、用户体验的优化，进行了全面透彻地讲解，满足了从电子商务平台建立到运维的基本需求。按照以上内容逐一进行商城的开发工作，可以使本来烦琐杂乱的工作变得有条不紊、充满魅力。

本书的目标读者

本书主要适合如下人群阅读：

- 电子商务平台决策者、首席技术官（CTO）和架构师。
- 电子商务平台运维工程师。

- 具有3年以上工作经验并从事电子商务平台开发的软件工程师。

如何阅读本书

本书主要介绍商城的整体设计、架构搭建、功能实现及安装部署，所以在阅读本书之前需要你对电子商务、网上商城有所了解，有过一定的项目开发经验，学习过PHP、HTML、jQuery、数据库等技术知识。如果你对缓存、负载均衡、服务器集群部署等有一定的了解将更能帮助你理解本书中所讲解的知识点。所以笔者建议按以下阅读方式进行阅读。

如果你是平台功能决策者或架构师，想了解一套标准电商平台套件的具体构成和实现，可以详细阅读第1章、第3~9章。第1章对商城标准套件进行了概括性的介绍，而第3~9章对第1章所提到的套件功能的模块

构成、设计思路、具体实现、功能体验等有详细介绍。

如果你是从事技术管理工作的，想了解搭建商城的技术架构，可以首先详细阅读第2章和第13章。第2章介绍了商城开发前期需要的技术准备工作，对一套完善、高效的商城技术架构进行了细致的讲解。第10章则介绍了商城的安装部署，详细介绍了软件版本与部署参数。在阅读完第2章、第10章之后，再去了解一下第3~9章的各模块，这对以后商城的二次开发及运维工作是非常有帮助的。

如果你已经在运营电子商城，拥有了一定的运维经验，可以将该书作为一本参考资料，因为书中讲到的功能及技术点都是经过ShopNC商城系统的工作人员参考众多客户的实际使用经验及建议，进行优化总结后的结果，具有一定的参考价值。

最后，真诚地期望大家通过本书来研究和探索电商平台搭建和运维。相信你们在看过本书之后都会有所收获。期待大家能搭建出一个优秀、高效的电子商务平台。

致谢

本书由ShopNC B2B2C商城系统产品开发团队共同撰写，在撰写期间咨询和采纳了很多具有众多实战经验的ShopNC客户及其技术团队的意见与建议，力求使读者能从本书中获得很多实用的知识。此外，本书在编写过程中还参阅了大量国内外的文献和资料。

首先感谢ShopNC B2B2C商城系统开发组的同仁，大家各抒己见、通力合作才使本书的编写过程如此顺利。其次，感谢给予宝贵意见和建议的ShopNC商城系统客户。还要感谢我们所处的互联网时代，使文献和资料的查阅变得如此简单。本书着眼于商城模块设

计、技术实现，其中融入了很多作者的个人经验和理解，当然由于笔者水平有限，书中难免会出现一些疏漏或者与您理解上的不同，恳请大家批评指正。大家有任何问题意见或建议，可发邮件至
book_feedback@shopnc.net，我们会第一时间与大家沟通、交流。

第1章 电商平台标准化套件

随着电子商务近十年的高速发展，传统单一的商城交易系统已经不能满足市场需求，随之而来多元化的电商平台系统已经成为市场的主流。与商城配套的圈子/BBS论坛系统、新闻资讯CMS系统、商品推荐分享的微商城系统、买卖双方沟通的即时通讯IM系统以及手机等设备使用的移动端APP等都成为电商平台必不可少的组成部分，如图1-1所示。

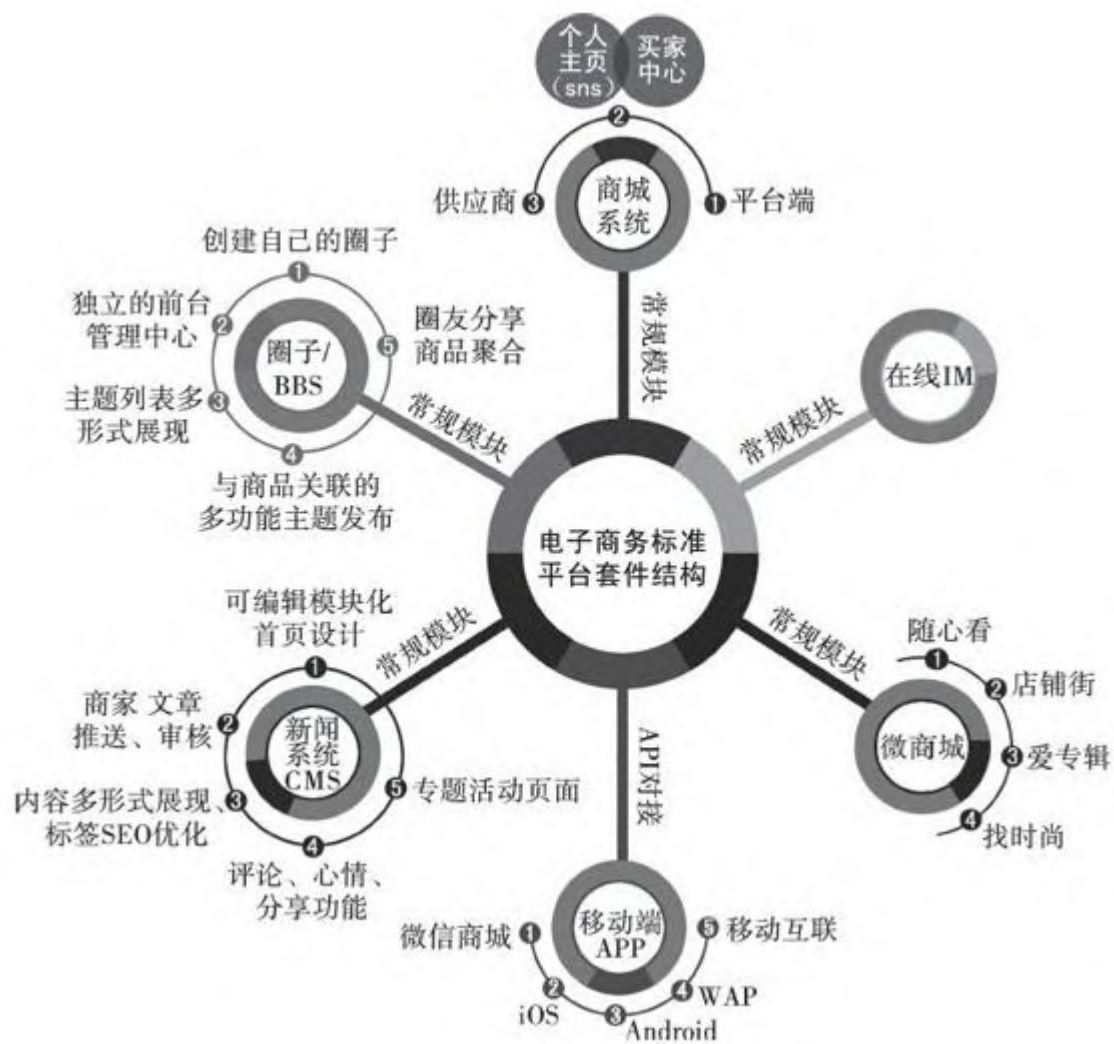


图1-1 电子商务标准平台套件结构示意图

1.1 商城系统

商城系统通过互联网为消费者提供一个新型的购物环境，除了传统的B2B、B2C、C2C等模式外，现今又延展出C2B、B2B2C、O2O等众多的商业模式，从单一的买卖功能扩展至极强的电商平台功能，如图1-2所示。

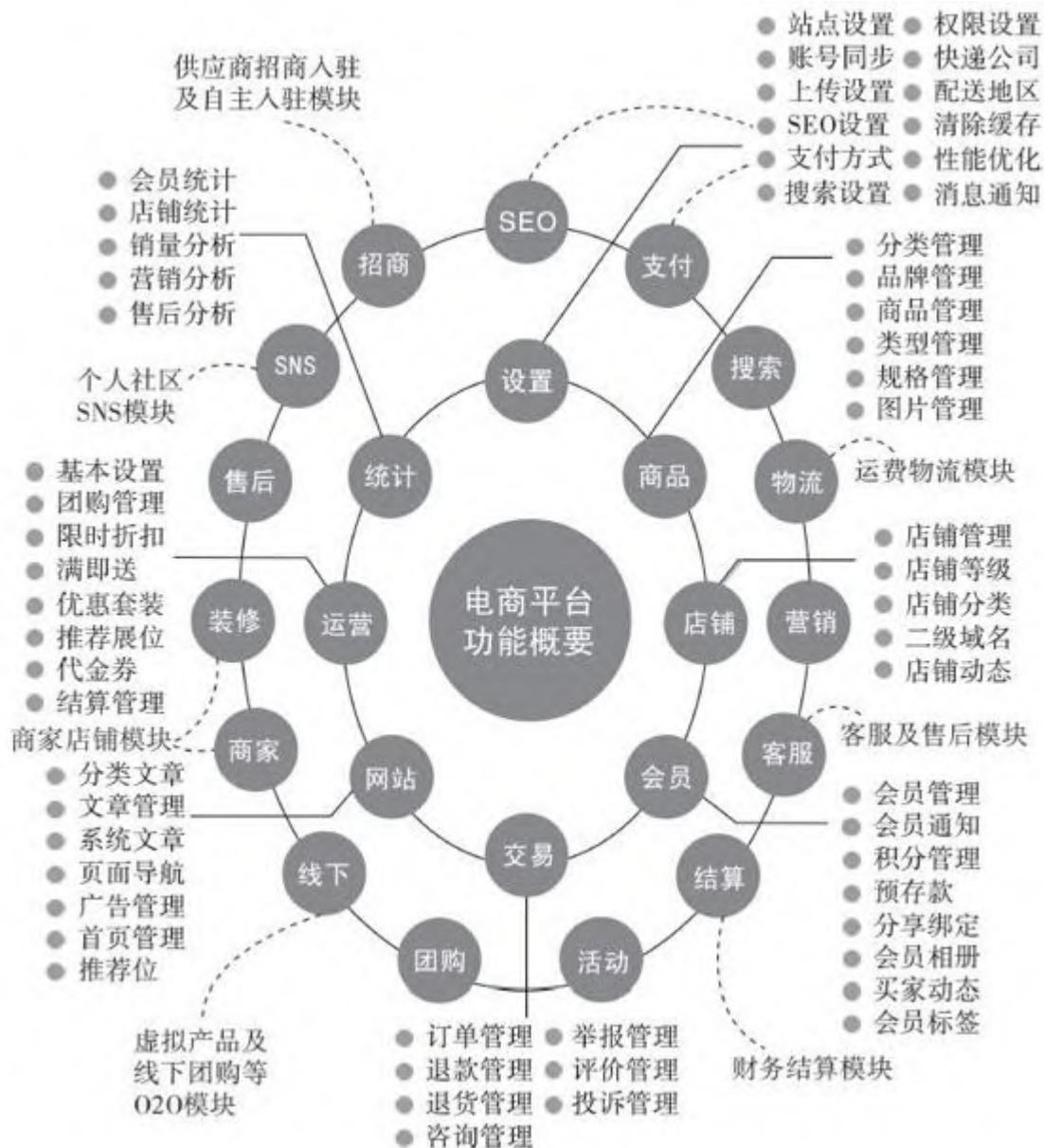


图1-2 电商平台功能概要

下面介绍一个标准的电商平台应该具备的功能。

(1) 设置

- 站点设置：设置网站的基本信息，包括网站名称、LOGO、联系电话、电子邮件、站点状态和防灌水设置。

- 账号同步：设定QQ互联和新浪微博登录等功能。

- 上传设置：设定上传参数、默认图片、登录主题图片、压缩工具和水印字体功能，上传参数能控制图片大小、尺寸、存放类型、图片扩展名等信息，登录主题图片可以设置4张主题图片，可以选择压缩工具、下载水印字体。

- SEO设置：设置首页、团购、品牌、优惠券、积分中心、文章、店铺、商品、商品分类和SNS等页面的标题、关键词和描述信息。

- 消息通知：包括邮件设置、邮件模板、商家消息模板和用户消息模板几部分。邮件设置可以进行邮件功能的开启与关闭操作，并在此进行SMTP账号的设置，设置完成后可以通过测试邮件功能进行邮件发送测试。

- 支付方式：进行平台、店铺支付方式的开启、关闭等操作的管理。支付方式包括：货到付款、预存款、支付宝、财付通、网银在线。

- 权限设置：账号信息的管理，可以增加或删除管理人员，可以修改系统管理员的管理权限。

- 配送地区：选择配送地区，按省、市、县三级分类。

(2) 商品

- 分类管理：对列表中的商品分类进行排序、是否首页显示等，并且可以对分类信息进行新增、编辑、删除、新增下一级等操作。系统支持对分类信息的导入、导出操作，并且支持TAG标签的管理，TAG值为分类搜索的关键字。

- 品牌管理：针对网站内的品牌信息进行编辑、删除、查询、新增和审核几项操作，也可以直接修改品牌排序和是否为推荐品牌。

- 商品管理：分为所有商品查看、违规下架商品管理、待审核商品及商品设置，可以根据商品信息如商品名、店铺名等进行查询，列表会显示商品价格和库存量，管理员可以选择该商品是否上架、是否违规下架，可以对商品进行查看、审核或删除操作。商品设置可对商家发布商品平台是否需要审核进行判定，

当此设定为需要审核时，商家所发布的所有商品均进入等待审核页面，由平台进行审核。

- 图片空间：商城运行一段时间后，在服务器上经常会产生一些垃圾图片，图片空间正是用于解决这一问题的。利用图片空间，管理员可以查看当前的所有店家的相册及其商品图片的数量，也可以删除无用的相册和图片。

（3）店铺

- 店铺管理：管理商城所有店铺信息，可进行查询、新增、编辑、删除等操作，支持新增或修改店铺经营类目，设置分佣比例，针对开店申请设置单独的审核入口，还可进行店铺续签申请处理及店铺新增经营类目审核。

- 店铺等级：显示店铺的等级列表，可以新增、编辑、删除、设置可选模板等操作，并且可设置不同等级店铺的年使用费用及相关权限等。

- 店铺分类：列出了店铺所有的分类，可以对店铺分类进行新增、编辑、删除等操作，还可以设定相关类目的保证金，以及按照排序值进行排序。

- 二级域名：用于进行平台二级域名的开启、关闭等相关设置，并且可以对店铺所使用的二级域名进行管理操作，可以根据店铺名称、二级域名进行查询，还可以进行二次域名及修改次数的编辑。

（4）会员

- 会员管理：显示所有会员的列表，在列表中显示登录次数、会员积分、预存款等信息，可以对会员信息进行增、删、改、查的操作。

- 积分管理：是对会员积分的操作入口，可以进行会员积分增加或减少的操作，并且可以记录操作描述。积分明细显示的是会员获得或消耗积分的情况，显示会员所有的积分变化，可以搜索会员积分操作记录。

- 预存款：记录了会员充值和提现金额的情况。在充值管理模块下，可以查看会员充值列表。在提现管理模块下，可以查看会员提现列表。预存款明细显示会员充值和提现金额的记录，可以通过会员名、管理员名、预存款类型等字段进行查询，查找所需要的信息。

- 分享绑定设置：可以设置QQ空间、腾讯微博、新浪微博的应用接口信息及是否开启相应站外分享功能。

- 买家动态：管理用户动态及评论信息，可以对用户动态及评论进行查看、显示、屏蔽和删除操作。

(5) 交易

- 订单管理：管理用户所有与订单相关的信息。根据订单信息、时间、金额和状态，我们可进行查询、查看、取消订单和确认收货等操作。

- 退款管理：审核退款订单，管理所有退货订单，可以对订单进行查看和删除操作。

- 退货管理：对退货订单进行审核，管理退货订单，可以对退货订单进行查看和删除操作。

- 咨询管理：是管理店铺咨询信息的操作入口，可以删除咨询信息。

· 举报管理：是管理举报产品的操作入口。“举报类型”和“举报主题”分别是举报类型的管理和举报主题的管理的操作入口。在管理人员处理完举报以后，举报信息被放入“已处理”模块下。

· 评价管理：买家可以在订单完成后对商品进行评价，也可以对店铺进行评价，管理员可对评价进行管理。

· 投诉管理：是管理投诉信息的操作入口。点击“详细”操作，可以查看投诉详细信息。通过“投诉设置”按钮可以进行投诉设置，通过“新投诉”、“待申诉”、“对话中”、“待仲裁”、“已关闭”按钮可以查看相应状态的投诉信息，通过“投诉主题”、“添加主题”按钮可以进行投诉主题的管理。

(6) 网站

- 文章分类：是对系统文章分类的管理，可以增加、编辑、删除、新增下级分类。

- 文章管理：是管理文章的操作入口，可以增加、编辑、删除文章，可以通过标题、文章分类等字段进行查询。

- 系统文章：对系统文章的管理，可以编辑，但不可新增或删除。

- 页面导航：是对页面导航的管理，可以增加、编辑、删除导航信息。对导航的添加主要在头部、中部和底部三个地方。导航类型包括自定义导航、商品分类、文章分类、活动四种。

- 广告管理：用于丰富界面和特色产品宣传，可查看全站所有广告位，可以新增、编辑、删除广告。

- 首页管理：对首页版块进行管理。基本设计可以修改版块名称、默认风格、排序；版块编辑可修改相对应的版块信息，如标题图片、推荐分类、活动图片、推荐商品、商品排行、广告图片、推荐品牌。

- 推荐位：是广告功能一个有效补充。推荐位不会生成JS代码，而是直接生成PHP代码，减少了http请求次数，可以复制推荐位代码放在商城前台展示。推荐位支持文字和图片形式。

（7）运营

- 基本设置：设置对预存款、金币、积分、积分中心、商品促销、团购等功能的开启与关闭操作，并可设置对会员注册、登录、订单商品评论、购物等赠送积分的数量。

· 团购管理：对平台团购信息的管理。管理员可以审核新的团购活动申请、取消进行中的团购活动或删除团购活动。可以查看参加团购活动的店铺和商品。对团购分类、团购地区和团购价格区间进行增、删、改、查等操作。设置购买团购活动所需的金币数量。

· 兑换礼品：是管理积分兑换礼品的操作入口，可以设置礼品的上架、推荐状态。“礼品列表”是礼品管理的操作入口，可以查看、编辑、删除礼品。

“新增礼品”可以进行礼品新增的操作。“兑换列表”是管理兑换礼品的操作入口，可以通过兑换单号、会员名称、支付方式、状态等字段进行查询。

· 活动管理：是管理平台活动的操作入口，可以新增、编辑、删除活动，可以设置活动的开始及结束

时间等。通过活动列表后的“处理申请”按钮可以查看商家提交的活动申请并作相关处理。

(8) 统计

- 会员统计：按照时间对新增会员、会员分析、会员规模分析、区域、购买等进行相关数据的分析。
- 店铺统计：按照店铺分类对新增店铺、热卖排行、销售统计、店铺等级、地区分布等进行相关的数据统计。
- 销量分析：按照时间对店铺的销售收入进行数据统计，以及按照相应的条件统计预存款及订单。
- 商品分析：统计商品销量明细、价格销量、热卖商品等。

· 营销分析：按照时间对促销商品及团购商品进行统计。

1.2 圈子

BBS，全称为Bulletin Board System，是Internet上的一种电子信息服务系统。它提供一块公共电子白板，每个用户都可以在上面通过书写来发布信息或提出看法。它是一种交互性强、内容丰富而及时的Internet电子信息服务系统，用户在BBS站点上可以获得各种信息服务、发布信息、进行讨论、聊天等。而圈子则是BBS的升级版本，与BBS相比圈子更加注重互动性与私密性。在整个电子商务平台中起到商品的推广、分享等重要作用。圈子的整体功能如图1-3所示。

下面介绍标准圈子应具备的功能。

- 圈子设置：设置圈子开关状态、圈子名称、LOGO、创建新圈子开关、发布/回复话题开关等。

- 成员头衔设置：设置头衔名称、所需经验值，也可以添加参考头衔。

- 圈子分类管理：是管理圈子分类的操作入口，可以设置圈子的名称、是否推荐、开关状态，还可以对圈子分类进行新增、编辑或删除操作。

- 圈子管理：是管理圈子的操作入口。可以对圈子进行新增、审核、编辑或删除操作，还可以把圈子推荐到首页。

- 圈子话题管理：可以对话题进行查看、回复或删除操作。

- 圈子成员管理：成员列表按照加入的圈子显示，可以对成员进行推荐、查看或删除操作。

- 圈子举报管理：管理圈子的举报信息，可以对举报信息进行删除操作。

1.3 CMS

CMS是Content Management System的缩写，即“文章管理系统”。在整个电子商务平台中起到重要的商品推广作用，如图1-4所示。在CMS中各种文章、画报、专题的发布及商品的软性推广，极大地增强了对买家的黏度，提高平台的浏览量与成交量。一篇好的图文推广往往能产生意想不到的结果。

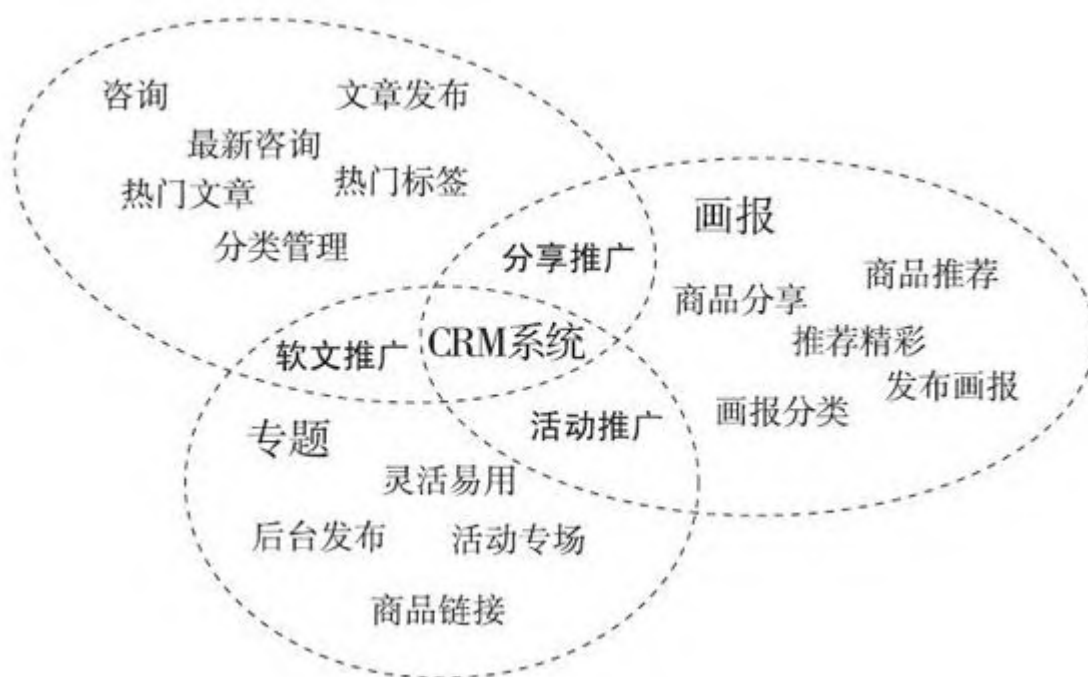


图1-4 CMS推广方式与途径

下面介绍成熟的CMS应具备的功能。

- CMS管理：对CMS开关、LOGO、评论开关、心情开关等进行设置。
- 首页管理：设置首页页面布局，可以对各模块进行编辑、删除或关闭操作。
- 文章管理：管理CMS中文章的操作入口。可以直接在列表中修改文章的浏览次数，开启或关闭心情和评论功能。可以对文章进行查看、审核、回收或删除操作。
- 文章分类：对文章进行编辑、新增或删除操作。

- 画报管理：管理CMS中画报的操作入口。可以直接在列表中修改画报的浏览次数，开启或关闭心情和评论功能。可以对画报进行查看、审核、回收或删除操作。

- 画报分类：对画报进行编辑、新增或删除操作。

- 专题管理：可以对专题进行新增、查看、编辑或删除操作。可对专题发布位置进行管理。

- 标签管理：管理CMS中标签的操作入口。可以直接在列表中修改标签名称、排序。可以对文章进行新增或删除操作。

- 评论管理：对CMS中的评论进行管理，删除违规评论。

1.4 IM

IM是即时通信系统Instant Messaging的缩写，是可以让买卖双方在线即时沟通的工具，其在电子商务平台中的作用是不言而喻的。每一个成功的电商平台都有自己的IM通信系统，如淘宝的旺旺、腾讯的QQ、京东的叮咚都是IM的表现形式。

1.5 移动端

如今手机等移动设备已经成为现代生活必不可少的一部分，随着智能手机的广泛普及，各种手机应用软件也应运而生。对于一个标准电子商务平台的手机客户端而言，Android、ISO、WAP等是必不可少的，并提供移动客户端下载设置，系统可控制并修改下载地址，以及在前台具有对应下载链接及手机二维码扫描功能，还可以自动判断机型并提供对应预设安装包下载。辅助手机客户端或其他类型扩展新增预设接口，如图1-5所示。



图1-5 移动端商城功能界面

以下是移动端的标准功能描述。

- 首页设置：用于设置手机首页版块，可对图片、标题、描述、关键词等进行编辑，可按照排序值

排序。

- 分类图片设置：用于设置手机端分类图标，可进行新增、编辑、删除等操作。

- 下载设置：输入APP下载地址及相应的安装版本信息，并且可在前台生成二维码。

1.6 本章小结

本章通过对电商平台标准化套件的介绍，帮助大家理解标准化这个概念，后面章节的内容均是围绕这几个标准化套件来阐述的。

第2章 电商平台的技术架构

电商平台的技术架构涉及很多方面，有软件的和硬件的，主要可以归纳为8个部分，它们分别是操作系统、应用服务器、负载均衡、缓存、数据存储、文件存储、消息队列、搜索设计和开发框架，本章将对这8个部分分别进行介绍。我们将电商平台技术架构整理成如图2-1所示，使大家有一个整体的认知。后面内容将对图中的主要点作介绍。

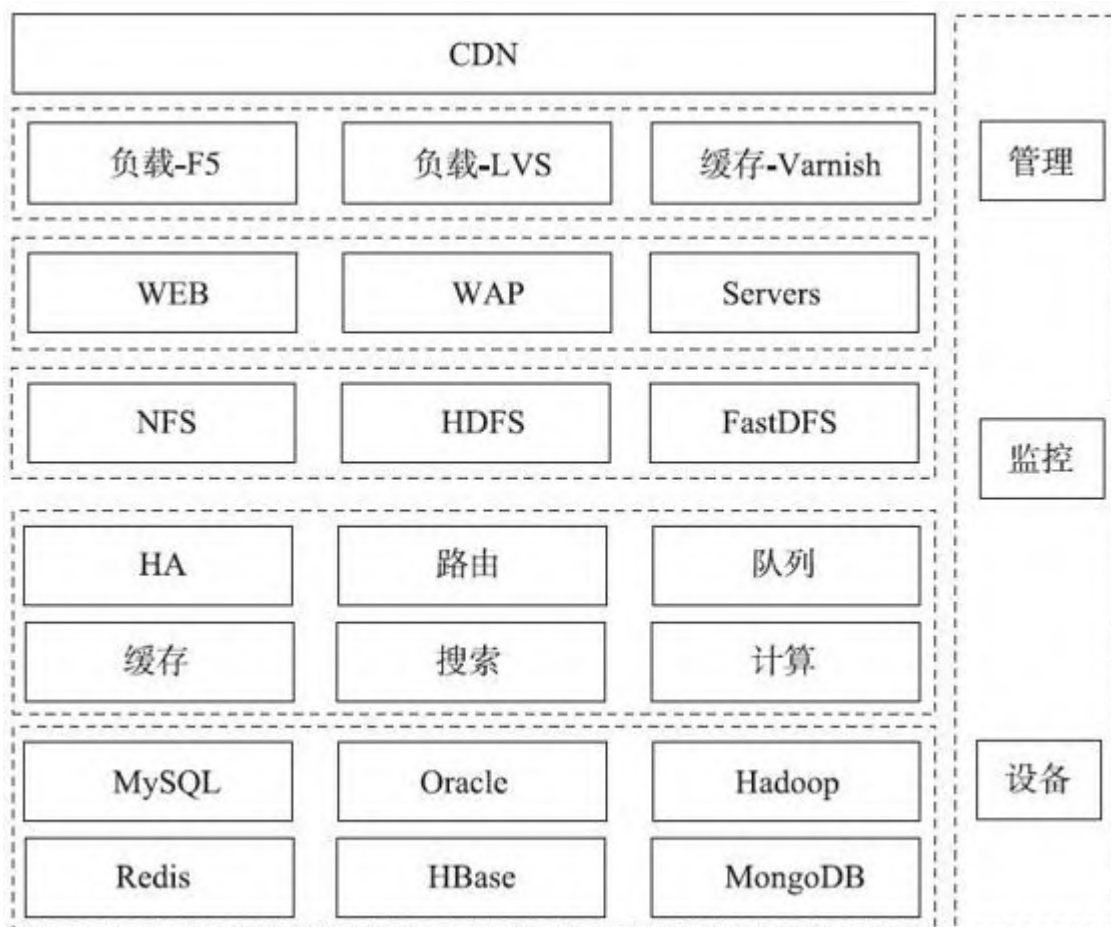


图2-1 大型电商平台架构

2.1 操作系统

服务器端的操作系统目前主要包括Windows、UNIX和Linux三大系列。

目前已知的Windows重要版本包括：Windows 2000 Server、Windows Server 2003、Windows Server 2003 R2、Windows Server 2008、Windows Server 2008 R2、Windows Server 2012。Windows服务器操作系统的典型应用是结合.NET构建服务器。

UNIX服务器操作系统由AT&T公司和SCO公司共同推出，主要支持大型的文件系统服务、数据服务等应用。目前市面上流传的主要有SCO SVR、BSD Unix、SUN Solaris、IBM-AIX、HP-U、FreeBSD等。

Linux操作系统与UNIX操作系统类似，不过它不是UNIX操作系统的变种，几乎所有UNIX的工具都可以运

行在Linux上。Linux的发行版可分为两类，一类是商业公司维护的发行版本，一类是社区组织维护的发行版本，前者以著名的Redhat（RHEL）为代表，后者以Debian为代表。Redhat系列包括RHEL（Redhat Enterprise Linux，收费版本）、FedoraCore（由原来的Redhat桌面版本发展而来，免费版本）、CentOS（RHEL的社区克隆版本，免费），稳定性方面RHEL和CentOS的稳定性都非常好，很适合于服务器使用，Debian系列（包括Debian和Ubuntu等）更适合桌面用户。

CentOS作为一种免费开源稳定的服务器操作系统，它支持众多软件平台的运行，目前被越来越多的网站所使用。电商平台的服务器系统优先推荐CentOS。

2.2 应用服务器

应用服务器在这里指的是完成一般业务的服务器，它可以分为两大类：前端服务器和后端服务器。前端服务器主要完成用户的响应，后端服务器主要完成数据处理，特别是大数据下的处理，中小规模的电商平台可能对前后端区分得并不是很严格，前后端可能会共用服务器。在运营规模达到一定量级后，众多模块（或子系统）将运行于后端，以支持整个平台的运行，这些模块有统计、行为分析、客户关系、结算等子系统。这也是云计算的小前端大后端的趋势，给用户带来最简单的用户体验。后端的技术实现模式可能综合利用多种技术手段共同实现，而不局限某一种语言或技术。

与后端服务器相比，前端服务器位置更靠前，其服务器性能的高低直接影响到用户的体验。前端应

用服务器根据实现业务的不同可能使用不同的规范（或架构）来实现，如J2EE、PHP、.NET、C、C++等都是可能使用的技术手段。关于这几种技术的比较已经是老生常谈的话题了，这里不做过多阐述，没有一种语言绝对比另一种语言强，开发者应当根据自己的实际情况，综合考虑成本支出、现有技术体系、实现业务、目前规模等要素，采用最适合自己的技术实现手段。

前端一般情况下需要频繁变化，使用PHP来实现工作更加高效快捷。或许有人认为PHP每次都需要时时解析会使它的响应速度变慢，我们先看一下PHP的编译过程。PHP执行代码会经过如下4个步骤。

1) Scanning (Lexing) ，将PHP代码转换为语言片段 (Tokens) 。

2) Parsing, 将Tokens转换成简单而有意义的表达式。

3) Compilation, 将表达式编译成Opocdes。

4) Execution, 顺次执行Opocdes, 每次一条, 从而实现PHP脚本的功能。

在源代码未发生改动的情况下, 如果把Opocdes缓存起来不就可以了吗? 是的, 当访问某一PHP文件时, 系统会直接从缓存中取得Opocdes来执行, 这样就省去了执行前3个步骤的时间。APC就是一个不错的缓存工具, 推荐使用。

就电商平台而言, 这里的前端服务器可以直接理解为Web服务器。提到Web服务器, 不得不说的就是Web服务软件, 目前主流的Web服务软件有Apache、Nginx、IIS、Lighttpd和Tomcat等。

(1) Apache

Apache是非常强大的老牌Web服务器，具有模块化结构，拥有众多非常成熟稳定的模块，目前仍是使用非常广泛的服务器，但它是基于多进程的HTTPServer，需要对每个用户请求创建一个子进程/线程来响应，当访问量较大时，会占用较多的系统资源（主要是CPU和内存），所以高并发的处理并不是Apache的强项。

(2) Nginx

Nginx是一款十分轻量级的HTTP服务器，同时也是一款非常不错的反向代理、负载均衡服务器。Nginx以事件驱动，专为性能优化而开发，支持内核Poll模型，能经受高负载的考验，并且在高负载下具有很高的稳定性。其他HTTP服务器在遇到访问的峰值或恶意攻击时，很可能会因服务器物理内存频繁交换而导致

内存耗尽，最终导致服务器失去响应，而Nginx采取了分阶段资源分配技术，使得它的CPU与内存占用率非常低。Nginx官方表示保持10000个没有活动的连接，只占2.5MB内存，所以在抵御类似DOS攻击时，Nginx有非常出色的表现。除此之外，就稳定性而言，Nginx比Lighttpd更胜一筹。

Nginx支持热部署，启动非常容易，并且几乎可以做到不间断运行，即使运行数月也不需要重新启动。Nginx还可以在不间断服务的情况下进行版本升级。

(3) Lighttpd

Lighttpd是一款轻量级Web服务器，占用内存小且CPU负荷低，是服务于静态内容的不错选择。

(4) IIS

IIS作为运行在Windows环境下的Web服务器软件，可以对.NET、PHP等进行解析。IIS只能运行于Windows下，而绝大多数开源架构或软件最佳搭配环境为Linux/UNIX，这是因为IIS不具有开源软件低成本、易扩展的特点。从开源、投入、扩展、性能等方面考虑，IIS均不是优先考虑对象。

(5) Tomcat

Tomcat服务器是一个免费的开源Web应用服务器，属于轻量级应用服务器，是开发和调试JSP程序的首选。Tomcat和IIS等Web服务器一样，具有处理HTML页面的功能，但处理静态页面的能力较差。

Lighttpd是一个单进程模型的Web服务器，内存使用量很小。Nginx在内存分配方面表现良好，它使用多线程来处理请求，这使得多个线程之间可以共享内存资源，从而使内存使用量大大减少。此外Nginx使用分

段内存分配策略，按需分配及时释放，总体占用内存很小，可支持较大的并发连接。Apache在运行时使用较大的内存，是多进程模型，使用基于内存池策略的内存管理方法，使得在运行开始时便一次性申请大片内存作为内存池，这样在随后需要的时候只在内存池中直接获取，不需要再分配，因此不适合大并发量的情况。Nginx作为一个后起之秀，发展速度非常迅猛，目前各大网站纷纷采用Nginx就说明它的优势是不言而喻的。笔者建议，如果开发框架采用了LNMP架构，可以优先考虑Nginx+php-fpm组合部署Web服务器，后期可视情形进行有针对性的调优。

2.3 负载均衡

随着用户量和业务量的提高，服务器的处理压力不断增大，一台服务器根本无法承担，在这种情况下，可行的办法就是增加一台服务器来分担它的压力，这时就需要一台调度器来完成调度工作。

负载均衡（Load Balance），就是将负载（工作任务）进行平衡、分摊到多个操作单元上执行，例如Web服务器组、FTP服务器组、数据从服务器组以及其他企业关键应用服务器等，从而共同完成工作任务。

负载均衡可以分为软件负载均衡和硬件负载均衡。常用的硬件负载均衡器有F5 BIG-IP、Citrix NetScaler、Cisco CSS、A10等，这些产品都价格不菲，费用高达几十万甚至更多，中小企业一般不会采用。软件负载均衡主要有LVS、HAProxy和Nginx等，这

些都是免费开源且成熟稳定的，所以很多客户更愿意优先考虑。下面将对主要的负载均衡产品进行介绍。

2.3.1 F5

F5是一种硬件负载均衡器，全称为F5 BIG-IP，官方名称为本地流量管理器，可做4~7层负载均衡，具有负载均衡、应用交换、会话交换、状态监控、智能网络地址转换、通用持续性、响应错误处理、IPv6网关、高级路由、智能端口镜像、SSL加速、智能HTTP压缩、TCP优化、内容缓冲、内容转换、连接加速、高速缓存、Cookie加密、选择性内容加密、应用攻击过滤、拒绝服务（DoS）攻击和SYN Flood保护、防火墙——包过滤、包消毒等功能。

下面介绍F5 BIG-IP作为HTTP负载均衡器的主要作用。

- F5 BIG-IP提供12种灵活的算法将所有流量均衡地分配到各个服务器，而对用户而言它就是一台虚拟

服务器。

- F5 BIG-IP可以确认应用程序能否对请求返回对应的数据。假如F5 BIG-IP后面的某一台服务器发生服务停止、死机等故障，F5会检查出来并将该服务器标识为死机状态，并且不再向该服务器分发任务。这样只要有一台服务器能正常工作，用户的访问就不会受到影响。死机的服务器一旦修复，F5 BIG-IP就会自动将该服务器重新加入分发的队列。

- F5 BIG-IP具有动态Session的会话保持功能。

- F5 BIG-IP的iRules功能可以实现HTTP内容过滤，根据不同的域名、URL，将访问请求传送到不同的服务器。

2.3.2 LVS

LVS全称为Linux Virtual Server，是由章文嵩博士（现为淘宝网基础核心软件研发负责人）主持的自由软件项目，它是一个负载均衡/高可用性群集，主要针对大业务量的网络应用（如新闻服务、网上银行、电子商务等）。LVS是由一个主控服务器及若干真实服务器所组成的集群。Real-server负责提供服务，主控服务器根据指定的调度算法对Real-server进行调度控制。而群集的结构对于用户来说是不透明的，客户端只与单个的IP（群集系统的虚拟IP）进行通信，从客户端的视角来看，这个集群就相当于一台服务器。

Real-server可以提供很多服务，如FTP、HTTP、DNS、TELNET、SMTP等。LVS集群技术可以使用Linux操作系统实现一个高性能、高可用的服务器，它具有很好的

可伸缩性、可靠性和可管理性。下面介绍LVS主要特点。

- 抗负载能力强，LVS是工作在网络4层之上仅作分发之用，没有流量的产生，所以它在负载均衡软件中性能最强。

- 配置简单，没有可太多配置的东西，减少了人为出错的几率。

- 工作稳定，自身有完整的双机热备方案，如LVS+Keepalived。

- 应用范围比较广，几乎可以对所有应用做负载均衡。

LVS有三种工作模式，它们分别是VS/NAT、VS/TUN和VS/DR。

1. VS/NAT技术

VS/NAT (Virtual Server Via NAT) 是指网络地址转换技术，由一台负载均衡服务器和后端几台真实服务器组成了一个服务器集群。此时负载均衡服务器又称为虚拟服务器，它是直接面向对外服务的。当用户从外网发起请求时，首先会请求到虚拟服务器（负载均衡器），虚拟服务器根据调度算法从一组真实的服务器中选出一台服务器，将请求报文的目标地址改成真实服务器的地址，同时记录转发信息到HASH表中；真实服务器接到请求后，做相应处理，将回复数据包发往虚拟服务器；虚拟服务器收到真实服务器的回复数据包，查找到HASH表中关联的连接记录，然后对数据包做SNAT操作，将数据包的目标地址改写为客户地址并发给客户机。VS/NAT的体系结构如图2-2所示。

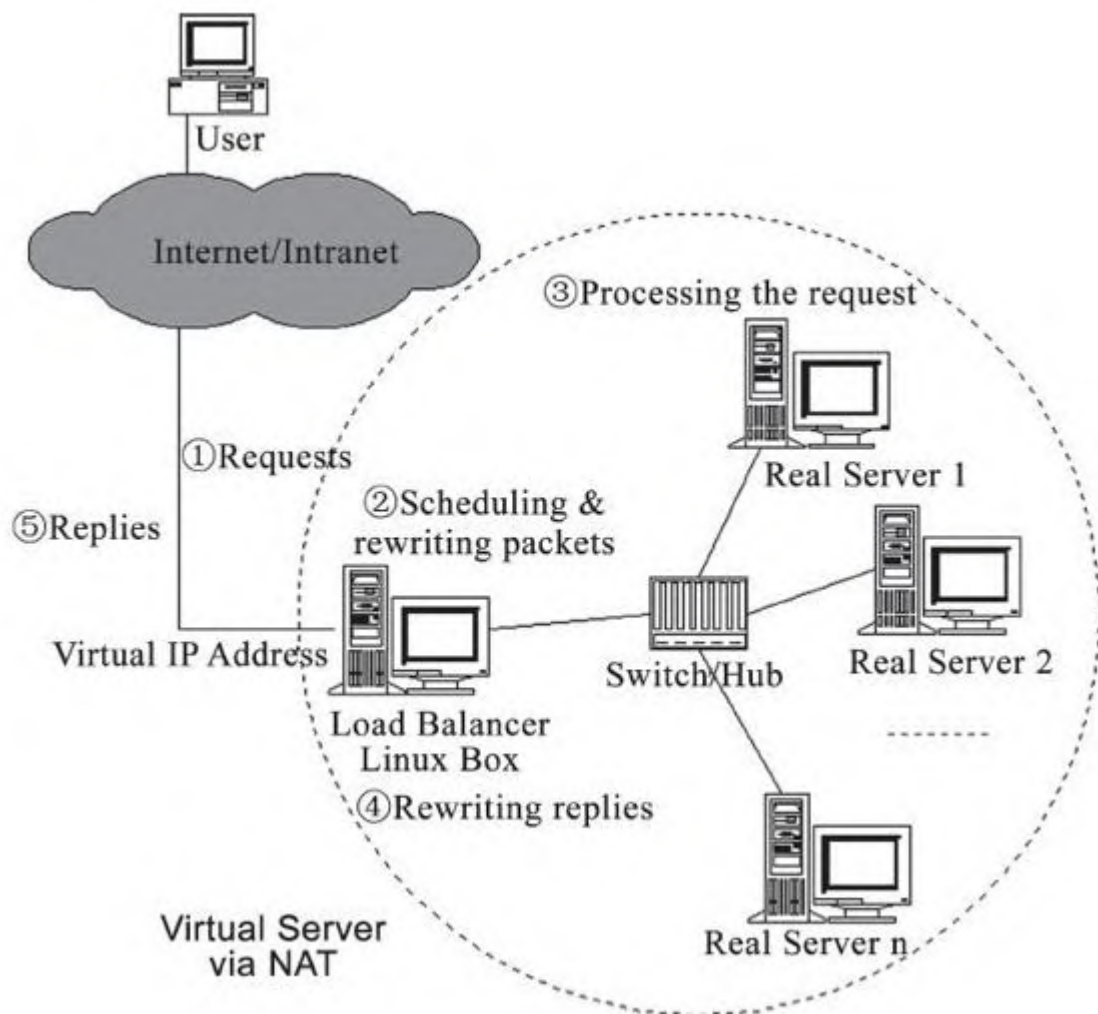


图2-2 VS/NAT技术

由图2-2可知，在一组服务器前有一个调度器（Load Balancer），它们之间是通过Switch/HUB相连接的。这些服务器提供相同的网络服务、相同的内容，即不管请求被发送到哪一台服务器，执行结果是

一样的。服务器Load Balancer被称为虚拟服务器，对于客户端来说它的IP地址被称为虚拟IP地址。

2. VS/TUN技术

VS/TUN (Virtual Server via IP Tunneling) 的工作原理是：连接调度和管理与VS/NAT中的一样，只是报文转发方法不同。调度器根据各个服务器的负载情况动态地选择一台服务器，将请求报文封装在另一个IP报文中，再将封装后的IP报文转发给选择的服务器；服务器在收到报文后，先将报文解析以获得原来目标地址为VIP的报文，服务器发现VIP地址被配置在本地的IP隧道设备上，所以会处理这个请求，然后根据路由表将响应报文直接返回给客户，如图2-3所示。

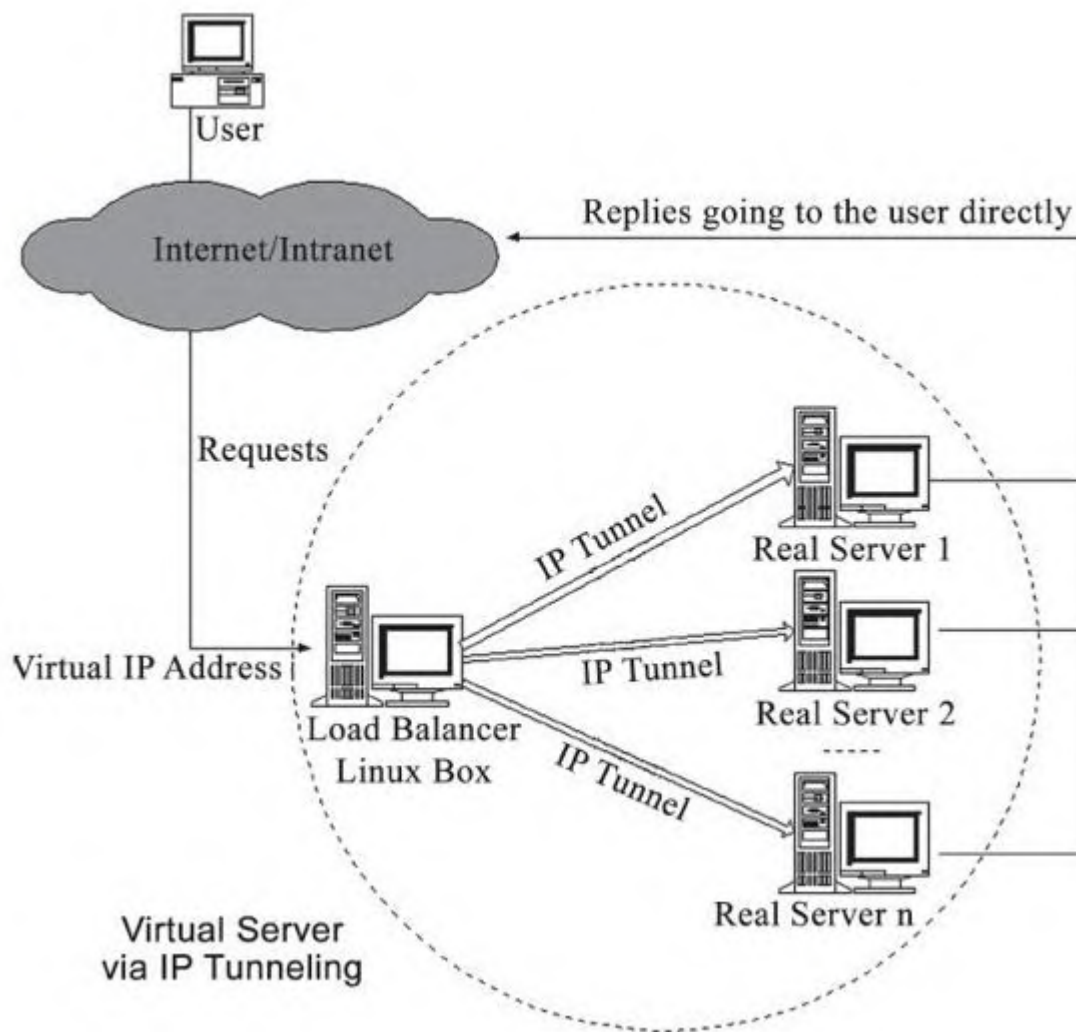


图2-3 VS/TUN技术

3. VS/DR技术

在VS/DR (Virtual Server via Direct Routing) 中，调度器根据各个服务器的负载情况，动态地选择一台服务器，不修改也不封装IP报文，而是

将数据帧的MAC地址改为选出服务器的MAC地址，再将修改后的数据帧向服务器组的局域网上发送。因为数据帧的MAC地址是选出的服务器，所以服务器肯定可以收到这个数据帧，从中可以获得该IP报文。当服务器发现报文的目标地址VIP是在本地的网络设备上，服务器会处理这个报文，然后根据路由表将响应报文直接返回给客户，如图2-4所示。

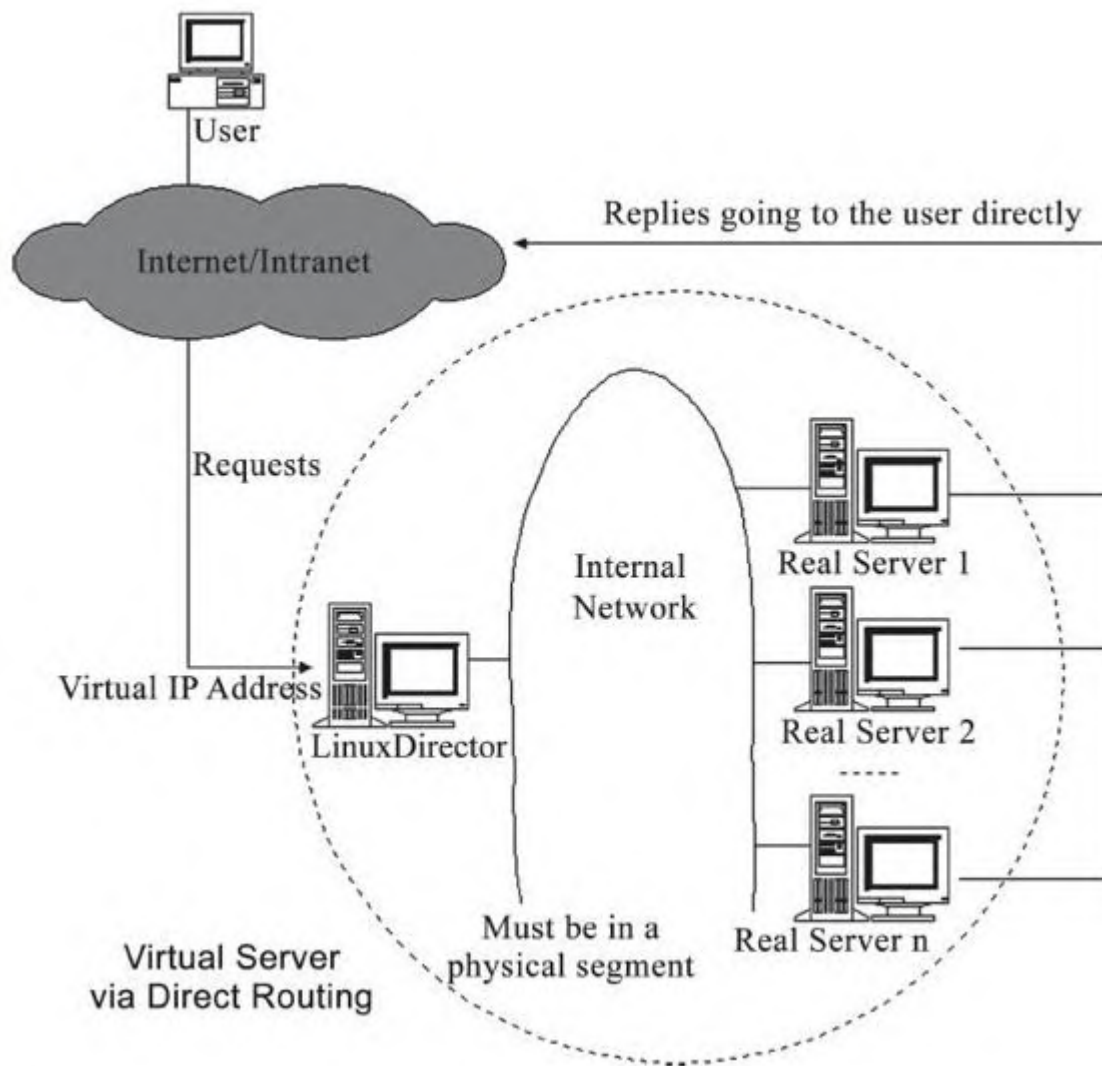


图2-4 VS/DR技术

4. 三种工作模式比较

(1) VS/NAT

VS/NAT的优点是服务器可以运行任何支持TCP/IP的操作系统，它只需要一个IP地址配置在调度器上，服务器组可以用私有的IP地址。缺点是它的伸缩能力有限，当服务器节点数目升到20时，调度器本身有可能成为系统的新瓶颈，因为在VS/NAT中请求和响应报文都需要通过负载调度器。

(2) VS/TUN

在VS/TUN的集群系统中，负载调度器只将请求调度到不同的后端服务器，后端服务器将应答的数据直接返回给用户。这样，负载调度器就可以处理大量的请求，甚至可以调度百台以上的服务器（同等规模的服务器）而不会成为系统的瓶颈。即使负载调度器只有100Mbps的全双工网卡，整个系统的最大吞吐量可超过1Gbps。所以，VS/TUN可以极大地增加负载调度器调度的服务器数量，可以用来构建高性能的超级服务

器。VS/TUN技术对服务器有要求，即所有的服务器必须支持“IP Tunneling”或者“IP Encapsulation”协议。目前，VS/TUN的后端服务器主要运行Linux操作系统。

(3) VS/DR

跟VS/TUN方法一样，VS/DR调度器只处理客户到服务器端的连接，响应数据可以直接从独立的网络路由返回给客户，可以极大地提高LVS集群系统的伸缩性。与VS/TUN相比，这种方法没有IP隧道的开销，但是要求负载调度器与实际服务器都有一块网卡连在同一物理网段上，服务器网络设备（或者设备别名）不作ARP响应，或者能将报文重定向（Redirect）到本地的Socket端口上。

通过IP隧道实现虚拟服务器的方法VS/TUN，和通过直接路由实现虚拟服务器的方法VS/DR都是实际部署

中不错的选择。

5. LVS的调度算法

LVS负载均衡将任务调度到不同的后端服务器是通过调度算法来实现的。LVS共内置了8种调度算法，它们分别是：

1) 轮询调度，主要指按顺序从Real Servers中选择一台Real Servers。

2) 加权轮询调度，将Real Servers设置一定权值，进行调度。

3) 最小连接调度，按照Real Servers的连接情况进行调度。

4) 加权最小连接，根据设置的权值和现有的连接数进行调度。

5) 基于局部性的最小连接，主要用于增大Cache命中。

6) 待复制的基于局部性的最小连接。

7) 目标地址散列调度。

8) 源地址散列调度。

轮询调度会均等地对待每一台服务器，分配到每台服务器的压力是均衡的，部署中各服务器的配置一般都是相同的。使用轮询是比较合适的，它也是目前最常用的调度算法。

2.3.3 Nginx

Nginx既可作为负载均衡器也可作为反向代理服务器，配置非常简单，可以按轮询、IP_HASH、URL_HASH、权重等多种方法对后端的服务器进行调度，同时还支持对后端服务器的健康检查。由于Nginx是基于第7层的负载均衡，根据头部的信息执行负载任务，可对http应用执行分流策略（如对域名、目录）。Nginx对网络的依赖性较小。

2.3.4 HAProxy

HAProxy是一个适用于高可用性环境的TCP/HTTP反向代理和负载均衡开源软件，它在7层负载均衡方面的功能很强大（支持cookie track, header rewrite等），支持双机热备，支持虚拟主机，支持健康检查（通过patch可以支持ECV），同时还提供直观的监控页面，可以清晰实时的监控服务集群的运行状况。同时支持Linux 2.6内核中System Epoll，大幅提高了网络I/O性能。

HAProxy的特点是：

1) 能够补充Nginx的一些缺点，比如Session的保持，Cookie的引导等工作。

2) 支持URL检测，如果后端的服务器出了问题，它可以通过URL检测，及时发现并进行相应处理。

3) 从效率上来讲，HAProxy会比Nginx有更出色的负载均衡速度。

4) HAProxy可以对MySQL读操作进行负载均衡，对后端的MySQL节点进行检测和负载均衡。

可以从访问量、资金预算、技术储备等方面综合考虑，从节约的角度来看，前期可以使用Nginx或HAProxy单点负载均衡，后期各方面条件成熟以后再使用LVS或F5。

2.4 缓存

缓存可以分为两部分：文件缓存和数据缓存。文件缓存主要指的是静态内容（如HTML网页、图片、音视频文件、脚本文件等），这些可以通过CDN分发处理，从而提高网站响应速度。当然还可以通过varnish或squid缓存服务器来处理，同样也可以提高响应速度。数据缓存是指对某些数据进行缓存，这些数据可以是经常使用的数据、经过复杂的运算才得出的数据等，如网站的基本配置信息、商品的访问量、评论数、商家旗下的商品数量等都是可以缓存的对象。缓存的工具同样也很多，比如redis、memcache都是不错的缓存工具，但对于商城而言，优先推荐持久化的redis。下面将缓存要素归纳为以下几个小节进行详细介绍。

2.4.1 客户端缓存

现在越来越多的图片、脚本、css、flash被嵌入到页面中，当我们访问它们的时候势必会做许多次http请求。其实，我们可以通过设置网页头部报文中时间属性来缓存这些文件。缓存时间（Expires）是通过头部（header）报文来指定特定类型的文件在浏览器中的缓存时间。大多数的图片、flash在发布后都是不需要经常修改的，做了缓存以后再次浏览就不需要再从服务器下载这些文件，直接从缓存中读取即可，这样再次访问页面的速度会大大加快。例如，在PHP中设置30天后过期：

```
Header("Cache-Control: must-revalidate");
$offset = 60 * 60 * 24 * 30;
$ExpStr = "Expires: " . gmdate("D, d M Y H:i:s", time() +
$offset) . " GMT";
Header($ExpStr);
```

2.4.2 CDN加速

CDN的全称是Content Delivery Network，即内容分发网络，其基本思路是尽可能避开互联网上有可能影响数据传输速度和稳定性的瓶颈和环节，使内容传输的更快、更稳定。通过在网络各处放置多个节点服务器从而构成一个在现有的互联网基础之上的一层智能虚拟网络，CDN系统能够实时地根据网络流量和各节点的连接、负载状况，以及到用户的距离和响应时间等综合信息将用户的请求重新导向离用户最近的服务节点上。其目的是使用户可就近取得所需内容，解决Internet网络拥挤的状况，提高用户访问网站的响应速度。

对于大型电商平台而言，一般需要建CDN做网络加速，如淘宝、京东都采用自建CDN，中小型企业可以

采用与第三方CDN厂商合作，如蓝汛、网宿、快网等。

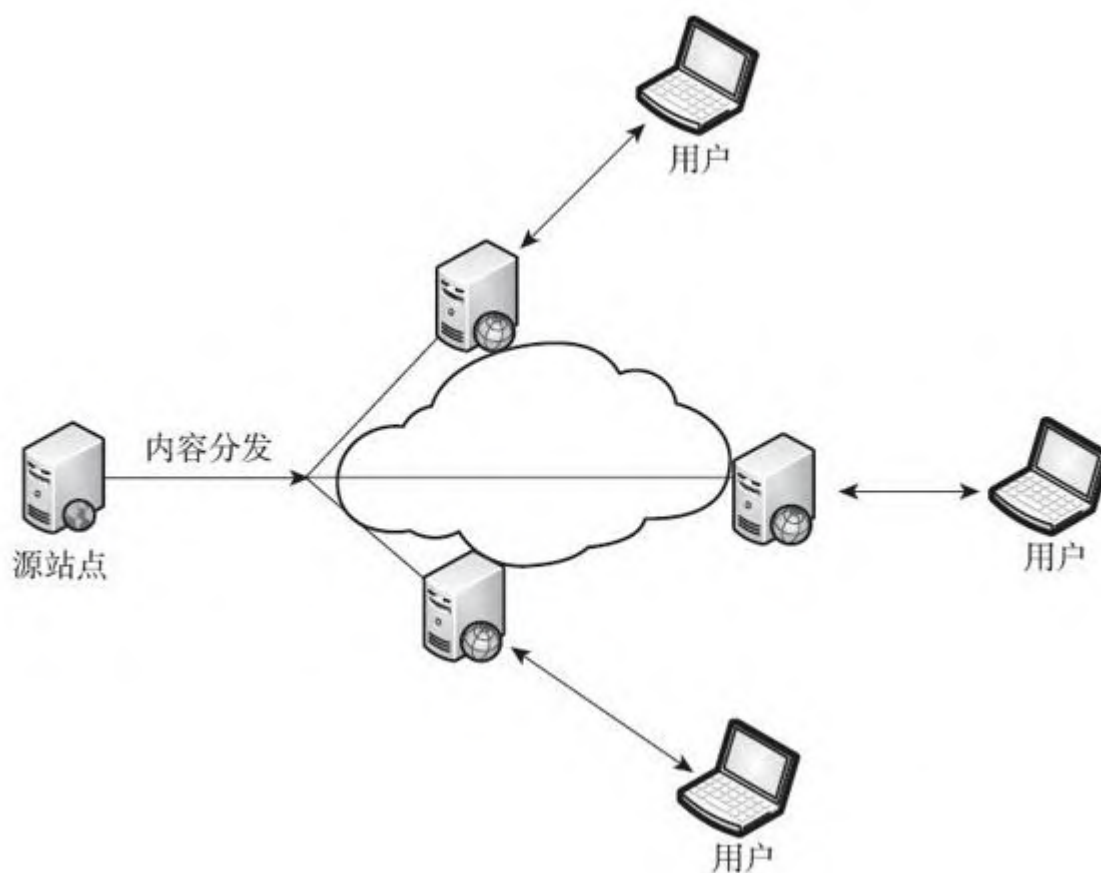


图2-5 CDN内容分发

2.4.3 静态文件缓存

静态文件缓存主要用到的是Varnish/Squid，它在Web服务器前面添加了一层缓存，可缓存图片、js等格式的文件，当访问这些文件时，会先从缓存服务器中查找，如果未命中，才会去请求Web服务器。关于Varnish和Squid的详细介绍大家可以去网上查询相关资料，这里不再赘述。

2.4.4 数据缓存

数据缓存方面使用较多的是Memcache和Redis。

Memcache是一款非常优秀的内存缓存工具，相信大家对它已经有所了解，这里不再赘述。Redis也是一个key-value存储系统，和Memcache类似，但它支持更多的存储类型，包括string（字符串）、list（链表）、set（集合）、zset（sorted set，有序集合）和hash（哈希类型）。Redis通常将全部的数据存储在内存中，但它也支持持久化。目前Redis通过两种方式实现持久化，一种是使用快照，它是一种半持久耐用模式，不时将数据集以异步方式从内存中以RDB格式写入硬盘；另一种是AOF，从1.1版本开始使用更安全的AOF格式替代，它是一种只能追加的日志类型，将数据集修改操作记录起来。Redis能够在后台对只可追加的记录作修改以避免无限增长的日志。Redis的出现很大

程度弥补了Memcache这类key-value存储的不足，在部分场合可以对关系数据库起到很好的补充作用。Redis提供了Python、Ruby、Erlang和PHP的API，使用很方便。

Redis支持主从同步。数据可以从主服务器向任意数量的从服务器同步，从服务器又可以是关联其他从服务器的主服务器，这使得Redis可执行单层树复制。由于完全实现了发布/订阅机制，使得从数据库在任何地方同步树时，可订阅一个频道并接收主服务器完整的消息发布记录。同步对读取操作的可扩展性和数据冗余很有帮助。

在电商平台中，很多数据都可以放入Redis缓存中，如网站的基本配置信息、商品的访问量、评论数、商家旗下的商品数量等都是可以缓存的对象。

2.5 数据存储

数据存储可以分为以下几类：

- 关系型数据库，以Oracle、MySQL为代表。
- key-value数据库，以Redis和Memcache为代表。
- 文档型数据库，以MongoDB为代表。
- 分布式面向列的数据库，以HBase为代表。

每种类型的数据库应用的业务领域是不一样的。

下面从关系型、内存型、分布式三个方面在电商平台的应用作一下分析。

2.5.1 关系型数据库

在当前电商平台中，使用较多的关系型数据库至少有三种，它们分别为：MySQL、Oracle、SQL Server。从使用率、技术成本、投入成本等综合考虑，MySQL肯定是首选。本节的内容我们只围绕MySQL展开。

MySQL最常用的存储引擎为MyISAM、InnoDB，其中InnoDB提供事务安全表，其他存储引擎都是非事务安全表。MyISAM是MySQL的默认存储引擎，支持事务，也不支持外键，但其访问速度快，对事务完整性没有要求。InnoDB存储引擎提供了具有提交、回滚和崩溃恢复能力的事务安全。但是比起MyISAM存储引擎，InnoDB写的处理效率要差一些，并且会占用更多的磁盘空间以保留数据和索引。

在电商平台中，使用最多的两个存储引擎也是MyISAM和InnoDB。由前面得知，InnoDB引擎支持事务，而电商平台的一大特性就是对事务的支持，电商的一些基础功能如下单、退单、充值等环节都需要作事务处理，这样才能最大程度保证数据的完整性、一致性，所以InnoDB存储引擎就成了电商平台唯一的选择。

2.5.2 内存型数据库

内存型数据库的代表是Redis、MongoDB等，它强调高并发，但在事务方面远不及关系型数据库。

关于Redis的介绍可查看2.4.4节，这里不再介绍。

MongoDB以多线程方式通信，主线程监听新的连接，连接后，启动新的线程做数据的操作（IO切换）。

MongoDB在数据存储上按命名空间来划分，一个collection是一个命名空间，一个索引也是一个命名空间，同一个命名空间的数据被分成很多个Extent，Extent之间使用双向链表连接。在每一个Extent中，保存了具体每一行的数据，这些数据也是通过双向链

接连接的，每一行数据存储空间不仅包括数据占用空间，还可能包含一部分附加空间，这使得在数据update变大后可以不移动位置。MongoDB索引以BTree结构来实现。

在MongoDB中使用了操作系统底层提供的内存映射机制，即mmap，MongoDB调用mmap把磁盘中的数据映射到内存中的，所以必须有一个机制时刻地刷数据到硬盘才能保证可靠性。

2.5.3 分布式数据库

在高并发的访问下，传统的关系型数据库（如MySQL）提供读写分离的方案，但在大并发下会产生数据不一致的问题。这时我们可能会考虑对MySQL数据进行切分、分库分表，但随着越来越多的海量数据，维护起来也越来越复杂。在大数据下，主备、主从、多主、分库、分表、数据切分等方案都不是很好的解决办法，它们无法做到高可用和高伸缩，这时就需要HBase了。分布式数据库HBase有一套完善的解决方案，适用于高并发海量数据存取的要求。

HBase基于列式存储，存储高效并且低IO。通常的查询不需要一行的全部字段，大多数只需要几个字段，而面向行的存储中，每次查询都会将全部数据取出，然后再从中选出需要的字段。面向列的存储系统

可以单独查询某一行，从而大大降低IO，同行数据具有很高的相似性，会增加压缩效率。

关于分布式数据库，更详细的信息建议大家去看相关专业资料，这里只是作为一种解决方案引出而已。分布式数据库适用于大数据，但对于中小平台而言，关系型MySQL数据库还是最佳首选。

2.5.4 MySQL可扩展方案

由于在目前电商平台中MySQL的使用率最高，所以我们将MySQL扩展方案专门用一节来介绍。在电商平台运营初期，由于用户量、访问量一般都比较低，不进行较大优化的MySQL是完全可以胜任的，但随着用户量、访问量的不断增大，就必须考虑MySQL的扩展方案了。目前有以下几个方案可选。

(1) MySQL Cluster

这是MySQL官方提供的集群方案，其特点为可用性及性能非常高，每份数据至少可以在不同主机上存一份副本，且冗余数据复制实时同步，但它的维护成本同样非常高，目前真正用于线上环境的极少，所以暂不推荐。

(2) DRBD硬盘网络镜像

DRBD (Distributed Replicated Block Device) 是一个用软件实现的、无共享的、服务器之间镜像块设备内容的存储复制解决方案，它的特点是功能强大，数据可在底层块设备级别跨物理主机镜像，且可以根据性能和可靠性要求配置不同级别的同步，I/O操作会保持顺序，可满足数据库对一致性的要求，但非分布式文件系统无法支持镜像数据同时可见，性能和可靠性相互矛盾，无法适用于性能和可靠性要求苛刻的环境，维护成本高于MySQL Replication方案。大家可以考虑根据自身环境来决定是否部署。

(3) MySQL Replication

MySQL Replication即MySQL的主从复制，是使用最多的一种架构，如图2-6所示，主服务器将SQL记录到日志文件，从服务器读取这个日志文件后将SQL应用

到自身。这是我们在实际应用中使用的最广泛的一种方案，也是优先推荐方案。

(4) MySQL数据切分

通过MySQL Replication方案可以从一定程度上缓解平台的压力，一旦数据库过于庞大，尤其是当写入过于频繁时，很难由一台主机支撑，还是会再次面临瓶颈。这时我们可以考虑数据库切分技术。

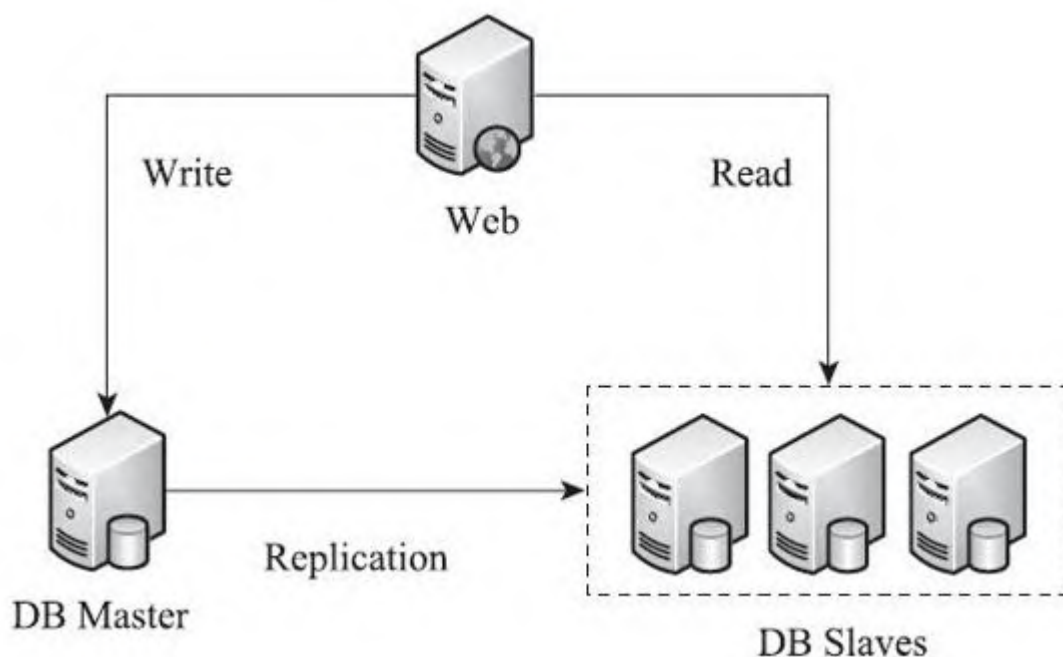


图2-6 MySQL Replication

关于数据切分，大家可能已经有所了解，它核心思想不是做成类似分布式数据库。简单来说，切分就是指通过某种特定的算法，将存放在同一个库（表）中的数据分散存放到多个库（表）中，以达到分散单台设备负载的效果。数据切分同时还可以提高系统的总体可用性，因为单台设备死机之后，只有这台设备上的数据不可用，其他设备的数据还可继续对外提供服务。

数据切分根据其切分规则可以分为两种：一种是按照不同的表来切分到不同的数据库（主机）之上，这种切分可以称为数据的垂直切分；另外一种是根据表中的数据的逻辑关系，将同一个表中的数据按照某种算法拆分到多个表中，这种切分称为数据的水平切分。

垂直切分的最大特点就是规则简单，实施方便，尤其适合各业务之间的耦合度非常低，相互影响很小，业务逻辑非常清晰的系统。在这种系统中，可以很容易做到将不同业务模块所使用的表分拆到不同的数据库中。根据不同的表来进行拆分，对应用程序的影响也更小，拆分规则也会比较简单清晰，但由于是垂直拆分，核心表在高并发下还是会出现瓶颈问题。

水平切分与垂直切分相比，相对来说稍复杂一些。因为要将同一个表中的不同数据拆分到不同的数据库中，对于应用程序来说，拆分规则本身就较根据表名来拆分更为复杂，后期的数据维护也会更复杂一些，但对于减轻系统压力来说，这种做法也更彻底，是在高并发大数据下的推荐处理方法。

2.6 文件存储

文件存储在电商平台中主要是指对图片的存储，因为在电商平台中图片文件是最多的。在网站初期，因为负载、数据都不是很大，程序文件和图片共同存放在一台服务器，但随着网站的发展，图片的存储很容易变成瓶颈，可能因为图片的I/O过高造成服务器崩溃，也可能图片过多而占满磁盘空间等，无论哪种原因，从长远看部署独立的图片存储是非常有必要的。

提到图片存储，我们很容易就会想到共享存储，目前最容易的解决方式就是使用NFS方式。NFS即网络文件系统，是使用软件方式实现的，对服务器端的要求很低，普通的PC甚至都可以，通过在每台Web服务器上挂载服务器端开放出来的目录，实现了文件的远程存储和共享。NFS是软件实现，当然目前也有许多专业的硬件存储设备可以选择，像IBM、DELL都有成熟的存

储设备，这些设备基本是通过SAN、NAS模式来实现的。

在大型平台上，一般是采用分布式文件存储系统，像HDFS、FastDFS等都是不错的选择。

2.7 消息队列

消息队列对电商平台特别是大型电商平台来说是不可或缺的部分，电商平台中的众多操作，如邮件发送、短信发送、商品属性更改后需要通知各类促销模块，以及执行时间较长的insert/update/delete等，这些操作如果在小型站点中使用同步方式处理姑且可以容忍，但在大型平台中，由于用户量、访问量特别大，使用同步处理会造成站点的堵塞，用户可能需要等待很长时间，有的甚至会直接超时。但如果只是简单将操作都改成js异步触发，同样不是很好的解决办法，用户需要发出js请求并等待返回，大量异步js请求同样会造成系统的压力猛增，直至堵塞，所以将这些费时但又不需要立刻完成的请求放入一个队列中，然后由队列来执行这些任务，在将任务放入队列后，

用户不需要等待任务的执行，而是继续进行下面的操作，这样极大提高了用户体验。

目前常见的消息系统有ActiveMQ、Gearman、MemcacheQ、RabbitMQ、HTTPSQS淘宝MetaQ NSQ等，除了以上专业的消息队列系统以外，还有许多基于第三方存储设备的消息系统，如基于Memcache/Redis的消息队列等。比如ShopNC B2B2C商城系统从兼容性、扩展性、易维护等角度考虑，采用PHP+Redis模式实现消息队列，系统对入列和出列操作进行了封装，易部署、易维护和易扩展。

2.8 搜索设计

搜索可以说是用户在购买商品前第一个要使用的功能。一般电商平台有成千上万甚至更多商品，用户如果想要快速精准地找到自己想要的商品，搜索是唯一的选择。

搜索的处理可能有多种方式，最初可能会停留在数据库阶段，通过模糊查询来处理，但在大数据下这会给数据库带来很大的压力。MySQL自身也提供了全文检索，但它有一些限制，只有MYISAM类型的表才支持FULLTEXT全文索引，只能在char、varchar和text建立索引，对中文支持欠佳，在大数据下的表现还有待验证，这时选择一个适合的搜索引擎就至关重要了。

目前开源的企业级搜索引擎主要有lucene、sphinx，以及国内的搜索引擎xunsearch。关于它们之

间的优点及弊端这里不再介绍，网上有许多介绍的文章，从易用性、对中文的支持、搜索效率等方面综合考虑优先推荐xunsearch。

2.9 开发框架

开发框架对大家来说并不陌生，Java、PHP、JS、Android等都有很多成熟的框架。使用成熟的框架开发可以避免一些重复性的基础工作，极大提高我们的开发效率，同时也易于扩展。但对于电商平台而言，使用一款目前成熟的通用型框架可以加快开发速度，同时也会带来部分性能损失，需要对框架进行有针对性的二次开发才能使性能达到最佳。

例如，ShopNC商城系统是ShopNC开发团队借鉴了国内外优秀的开源程序与开发模式，结合多年电商平台开发经验，采用PHP5+MySQL技术为基础，结合OOP&MVC模式，开发出LAMP下基于电商平台的专有框架，如图2-7所示。

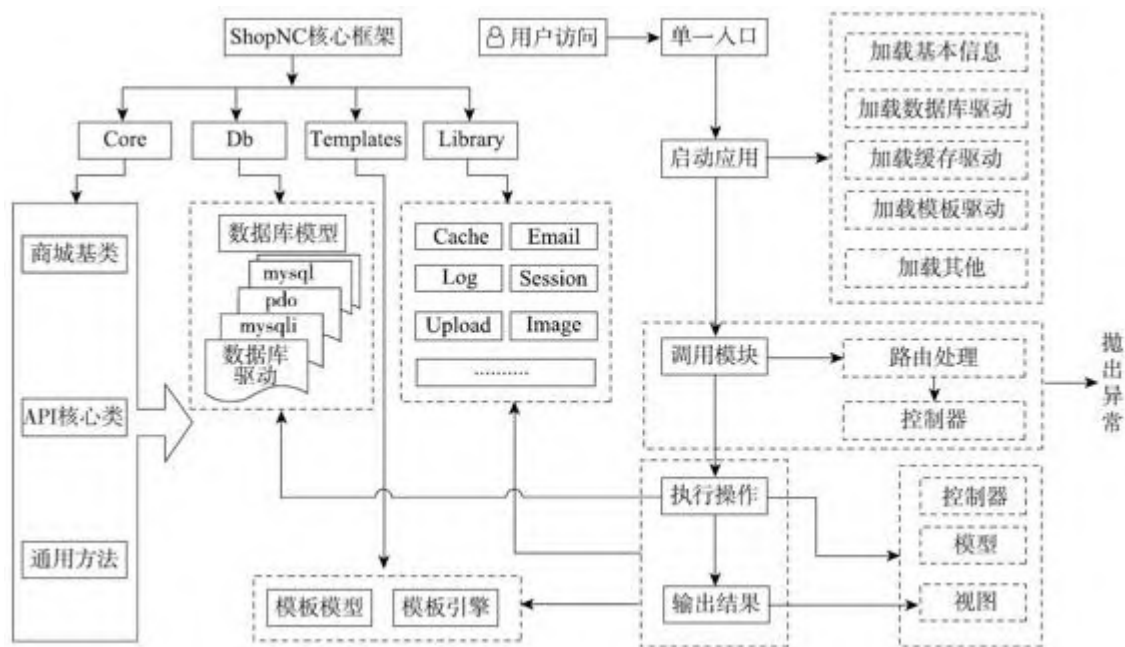


图2-7 ShopNC电商平台框架

2.10 本章小结

本章从操作系统、应用服务器、负载均衡、缓存、数据存储、文件存储、消息队列、搜索、开发框架8个方面对电商平台技术选型作了阐述，但限于篇幅未从实现角度做过多的介绍，本章仅作为抛砖引玉，对其中某些选型方案感兴趣的读者可以去查阅相关专业书籍。

第3章 商城套件的设计与实现

3.1 会员模块

电子商务网站推广的最终目的是获得更多的忠诚会员。会员是电子商务一切业务的基础。无论是C2C、B2C还是B2B2C模式的网站，发展会员并进行定期访问和维护的制度是其营销战略的一个重心。为了适应现代经济竞争的变化，电商平台开始了从“以产品为中心”的模式到“以客户为中心”的模式战略转变。

对于做电子商务的企业来说，不管在运营的哪个阶段，做好会员营销要比搞流量靠谱。这样不仅能直接促进销售，更有利于通过会员制度建立起忠实的客户群，从而期望形成较为稳定的销售支撑，并可能通过不断强化会员的忠诚度来展开口碑营销式的推广。可见会员对于企业是多么重要。

3.1.1 模块构成

B2B2C电商平台中分为买家和卖家两种会员。会员注册后默认为平台买家。如果想申请为卖家，则需要注册后提交商家入驻申请，经过平台审核后才能成为平台的卖家。同时，为了买家和卖家管理流程的清晰，买家和卖家登录口独立存在。会员作为网站的基础，基本涉及了网站各个模块。图3-1展示了会员构成模块和主要流程。

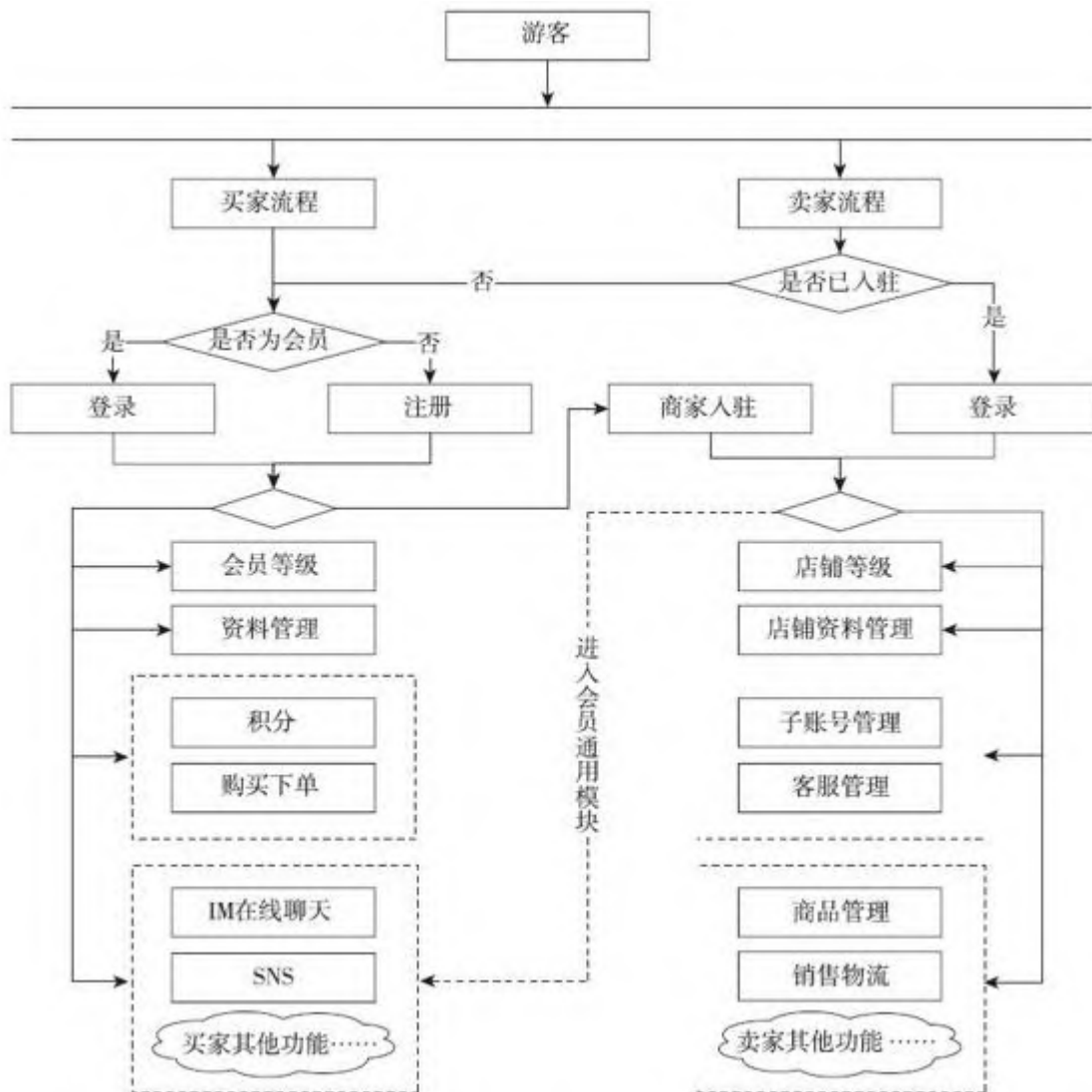


图3-1 会员构成模块和主要流程

3.1.2 设计思路

1. 设计要求

会员模块是网站最基本也是最重要的模块，它设计得好坏直接影响了会员对网站黏性的高低。下面我们来探讨一下会员模块设计的几点要求。

(1) 界面简洁方便

当今互联网高速发展，登录注册已经成了网民最司空见惯的事情了。几乎每个网站都有一个登录和注册页面。就是这么简单的操作流程也经历了很多发展的过程。刚开始单一个注册页面可能就要2~3个屏幕，需要输入性别、家庭住址、公司信息等很多信息，注册一个会员就要花好几分钟，相当烦琐，一看需要写这么多就让人产生了放弃的念头。后来慢慢变成了今

天的只需要输入一个邮箱（或者手机）和登录密码即可，页面简洁且让用户感觉轻松，让人们感觉注册一个会员原来如此轻松，不经意间就注册了很多的网站会员。所以会员的登录和注册页面也是很重要的，应该简洁方便，让人们感觉亲切、舒服，千万不要从一开始就吓退了客户。

（2）多收集会员信息

会员资料是一个网站很重要的信息。运营者可以通过会员资料挖掘很多的有效信息，为运营提供依据。这一点看似跟上面说的“界面简洁方便”是冲突的，其实不然。注册和登录简单、便捷，可以吸引更多的会员加入网站，然后在用户中心的会员资料或其他功能中以某种形式（例如送积分等）吸引会员完善资料，这样就可以两者兼得了。要多多收集并完好保管会员信息。

（3）差异化管理

网站不断运营发展，会员也在不断地增加，慢慢变成了一个很大的“家庭”，如果没有一定的约束，就会对家庭的每个成员失去凝聚力，使他们不追求上进，乱作一团，不知道会发生什么事。所以网站会员管理需要一定的管理纲要和激励机制，让家庭的每个成员都有上进的机会。会员等级制度、管理制度是很必要的。不同级别会员享受不同的服务，如此就能让大家朝着一个目标努力，网站有条不紊的运转，会员的雪球也就越滚越大。

（4）增加会员黏性

网站获得一个注册用户有时候需要付出很多营销成本。如果这个用户注册完会员之后就不再光临了，这对于网站来说就是劳而无功，而且还损失掉了营销费用。所以在用户注册成为会员后，制定合理的制度

和流程吸引会员再次浏览网站，显得如此重要。常见的方法有登录送积分等。久而久之，时常登录网站成为习惯，会员黏性自然提高。

（5）会员传帮带

每个人都是比较容易相信身边认识的人，这是人之常情。通过营销吸引一个新会员注册，要比通过老会员推荐新会员注册成本高得多，而且前者的稳定性也差。所以网站加入会员传帮带，老会员带新会员的制度和流程很有必要。比如推荐新会员奖励积分，或者代金券等。将营销的费用用到老会员推荐新会员上，不仅减少了营销费用、增加了新会员，而且也提高了老会员黏性，对网站好处多多。

（6）高内聚低耦合

会员分为买家会员和卖家会员，两者有着一定的联系，但很多功能又是独立存在的。如果合在一起去设计，则可能会让使用者摸不着头脑，因此在保持两者有一定联系的同时，应该尽量让其独立存在，也就是程序设计中的高内聚低耦合。例如买家会员登录和卖家会员登录是独立的两个页面，买家中心和卖家中心也是独立的两个模块，这样让使用者感觉很清晰，使用起来比较舒服。

（7）进行数据分析

会员资料对于网站来说是蕴藏着无限价值的宝藏。而如何挖掘它就决定了能够获得价值的多少。设计合理的数据分析功能，对会员数据进行有价值的数据挖掘，使前面的会员维护工作得到提升。挖掘的数据又为运营服务，形成良性的循环，有助于网站的不断发展。

2. 数据表设计

下面我们来看看会员相关数据表的设计需要注意些什么。

(1) 主从配合

会员需要存储的信息也是很繁杂的，会员包含买家和卖家两种身份，这两种身份都需要存储很多信息。例如会员基本信息有真实姓名、联系方式、兴趣、头像等细小信息。如果将这些信息都存储到一张表，显然这张表会相当大，这样做没有必要，还会影响查询的速度。所以一般会设计主从配合的数据库表结构。例如，会员基本信息表如表3-1所示，会员扩展信息如表3-2所示。

表3-1 会员基本信息表

字 段	类 型	Null	默 认	说 明
member_id	int(11)	否		会员 ID
member_name	varchar(50)	否		会员名称
member_truename	varchar(20)	是		真实姓名
member_avatar	varchar(50)	是		会员头像
member_pwd	varchar(32)	否		密码
member_paypwd	char(32)	是		支付密码
member_addtime	int(11)	否	0	注册时间
member_points	int(11)	否	0	会员积分
member_state	tinyint(1)	否	1	开启状态 0 为关闭, 1 为开启
member_loginnum	int(11)	否	0	登录次数
⋮	⋮	⋮	⋮	⋮

表3-2 会员扩展信息表

字 段	类 型	Null	默 认	说 明
member_id	int(11)	否		会员 ID
member_sex	tinyint(1)	是		会员性别
member_birthday	date	是		生日
member_email	varchar(100)	是		邮箱

(续)

字 段	类 型	Null	默 认	说 明
member_mobile	varchar(20)	是		移动电话
member_qq	varchar(50)	是		QQ 号码
member_ww	varchar(100)	是		旺旺号码
member_areaid	int(11)	是		所在地区 ID
member_cityid	int(11)	是		所在城市 ID
member_provinceid	int(11)	是		所在省份 ID
⋮	⋮	⋮	⋮	⋮

表3-1为主表，存储会员的重要且频繁使用的基本信息，表3-2为从表，存储会员不是十分重要且使用次

数较少的信息，在设计表时可以根据具体需要合理安排字段在主从表中的位置。

(2) 合理使用冗余

在数据库设计原则中有一条即减少数据冗余节省存储空间，可是有时候为了程序的设计和性能，合理的冗余是很必要的。这样可以减少多表联查和在程序中查询数据库的次数，提升运行性能。只要降低不必要的冗余，保留合理冗余数据，把握好一个度即可。店铺信息表如表3-3所示。

表3-3 店铺信息表

字 段	类 型	Null	默 认	说 明
store_id	int(11)	否		店铺 ID
store_name	varchar(50)	否		店铺名称
grade_id	int(11)	否		店铺等级
member_id	int(11)	否		会员 ID
member_name	varchar(11)	否		会员名称
seller_name	varchar(50)	否		店主卖家用户名
sc_id	int(11)	否		店铺分类
⋮	⋮	⋮	⋮	⋮

表3-3中member_id为店铺店主的会员ID，细心的读者会发现下面还有一个member_name，这个字段显然是冗余字段，但它是很有必要存在的。当查询店铺信息的时候，一般需要知道店主会员名称，如果该字段不存在，则需要联查会员表，在网站中会员表的数据量也是很大的，店铺表和会员表这两张数据量很大的表联查显然效率不会高。而如果该冗余字段存在，只需要查询一张店铺表即可，所以这样设计数据库是合理的。

（3）结构清晰

会员作为网站的基础数据与其他数据有着千丝万缕的联系。这就要缕清它们之间的关系，如果将看似可以设置到一张表中的字段都一股脑地添加到一张表中，可能导致程序编写时的麻烦和后续数据库扩展时的问题。例如：会员信息和卖家信息数据是一对一

的，一个会员对应一条卖家信息，当然也可以设置成一张表。但是为了结构的清晰和后续数据库的扩展，所以建立了member和seller两张表，这样后续对seller的信息扩展不会影响member表，也更容易理解。会员买家信息与卖家信息分表存储，如表3-4和表3-5所示。

表3-4 买家信息表

字 段	类 型	Null	默 认	说 明
member_id	int(11)	否		会员 ID
member_name	varchar(50)	否		会员名称
member_truename	varchar(20)	是		真实姓名
⋮	⋮	⋮	⋮	⋮

表3-5 卖家信息表

字 段	类 型	Null	默 认	说 明
seller_id	int(11)	否		卖家 ID
seller_name	varchar(50)	否		卖家名称
member_id	int(11)	否		会员 ID
seller_group_id	int(11)	否		卖家组编号
store_id	int(11)	否		店铺编号
seller_quicklink	varchar(1000)	是		卖家快捷操作
⋮	⋮	⋮	⋮	⋮

3. 模块设计

(1) 买家会员功能需求

1) 注册登录

当今网站几乎都需要会员注册和登录，电子商务网站更是如此，注册和登录是网站所有流程的基础。在B2B2C电商平台中买家和卖家只有注册或者登录后才能识别该会员是否具有权限执行某些操作或者记录其操作。登录和注册模块功能是比较单一的，具体操作即实现会员信息的有效性验证、必要安全设置和信息记录功能。除了简单的实现外，更多的是如何提高程序的可操作性，更好地方便用户使用。登录和注册模块由会员注册、用户登录和找回密码三部分组成。随着当今开发平台的兴起，第三方账号登录功能也是其很重要的构成部分，常用的第三方账号有腾讯QQ、新浪微博、支付宝等。

2) 买家会员等级

会员等级是网站对会员的最普遍管理制度，建立科学、完善有效的会员等级管理制度，可以协助公司明确业务重点对象及指标，达成预期的业绩成果，也是网站良好运作和会员经营管理的一个重要保障。一般情况下会员等级是与其他模块配合使用的。例如可以根据会员积分的分值，划分会员等级，并且根据各等级会员价值的不同，制定相应的优惠政策，吸引客户再次购买。

3) 资料管理

完善的会员资料可以帮助企业对会员进行分析，从中挖掘出有效信息，从而针对不同会员进行不同的营销和管理，对症下药方能立见奇效。但是完善会员资料这件事情，网站很被动，于是如何激励会员主动完善资料也是很重要的，当今大多数网站采用根据完

善资料的不同程度赠予不同数值的积分，吸引会员尽可能详细的完善资料。

4) 账号安全

账号安全问题是攻击侵入系统的主要手段之一，对于电子商务网站账号安全显得尤为重要。会员账号下可能存储了一定数额的预存款或积分，如果账号被盗，则意味着丢失了真金白银。所以电子商务网站中一定要加入账号安全模块，最大限度地保证账号安全。首先，应该分别设置登录密码和支付密码，其次绑定手机号或邮箱，当提交敏感信息的时候，进行手机和邮箱验证，通过多重验证保证账号的安全。在互联网发展的进程中，账号被盗一直伴随其中，我们应该引以为戒，建立完善的账号安全保障体系。

5) 其他相关功能

以上介绍了会员的基本构成模块，对于会员来说还有很多的相关扩展模块，例如消息通知、我的收藏、会员预存款、会员积分、浏览记录、我的好友等。通过这些扩展模块更好地服务于会员，提升会员黏性。例如会员登录赠送积分的营销方式，可以吸引会员经常登录网站。

（2）商家账户功能需求

1) 商家开店

商家开店前首先需要注册会员账号，通过会员账号申请开通店铺。在申请开通店铺时需要提交企业基本信息、各种证件的电子版、银行账号等信息。平台进行初步审核通过后，商家提交付款凭证，平台确认后为商家开通店铺。开通店铺的商家账号和会员账号进行绑定，该账号为店铺管理账号，拥有店铺的所有权限，可以建立子账户。

2) 权限管理

子账户的权限结构为账户组和账户的结构，首先建立账户组并为账户组分配权限。商家可以通过建立多个不同的权限组划分不同的权限，完成各自的职能，使运营团队可以高效地协作并完成销售。

3) 建立子账户

店铺可以建立多个子账户，子账户可以协助店铺负责人管理店铺，不同账户分配到不同权限组，以获得不同的工作权限，分清每个人的工作职责。所有子账户的操作都会记录到相应的操作日志中，方便负责人核对和管理，一旦出现问题可以通过查询日志来明确责任人。还可以通过日志，对每一位员工的工作情况，进行详细的了解，掌握每位员工的工作进度。

4) 店铺资料管理

店铺资料管理包含很多重要资料的管理，例如店铺Logo、联系方式等的设置。通过卖家中心灵活的设置店铺重要的资料，然后展示到前台店铺页面，可以简便地做到前台页面店铺信息的管理，使平台卖家开店变得简单起来。

5) 店铺装修

好的店铺形象能提升买家对店铺的整体印象，给买家留下深刻的印象。所以店铺需要通过店铺装修模块来设计和管理店铺形象，自定义店铺颜色、图片和显示版块等信息。

6) 店铺分类

在B2B2C电商平台中，一般会存在很多经营不同类目商品的店铺，不同商品种类或不同店铺需求，造成店铺商品的分类也是不同的，所以在店铺管理中加入

了店铺分类管理。店铺可以根据自己的意愿设置店铺分类，便于店铺的顾客快速找到自己需要的商品。

7) 店铺消费

将店铺促销等操作产生的费用，统一记录到店铺消费中，定期与平台进行结算。具体的结算周期以平台设置为准。

8) 其他相关功能

店铺模块与其他很多模块相关。例如商品管理、促销、订单管理、SNS店铺动态等。在制作店铺功能的时候，也要充分考虑到它对其他模块的影响。

3.1.3 代码实现

会员中最重要的就是登录注册模块了，高效和安全的登录注册模块对于网站来说很重要。下面我们就以登录和注册模块代码为例来看一下具体的实现代码。

1. 会员注册模块

会员注册模块实现的主要功能是验证用户填写的信息是否合法、验证用户是否已经存在、将注册信息存入数据库、建立用户session及注册需要增加的其他附属功能，例如注册赠送用户积分等。注册功能代码如代码清单3-1所示。

代码清单3-1 注册功能代码

/**

* 会员注册新增操作

```

*/
public function usersaveOp() {
//重复提交注册验证
if (process::islock('reg')){
showDialog('您的操作过于频繁，请稍后再试','index.php');
}
$model_member= Model('member');
//验证用户是否已登录
$model_member->checkloginMember();
//验证是否为合法提交，如果不是合法提交，或者超出提交次数，则终止操作
$result = chksubmit(true,C('captcha_status_login'),'num');
if ($result !== false){
showDialog('登录失败','','error');
}
//添加注册信息
$register_info = array();
$register_info['username'] = $_POST['user_name'];
$register_info['password'] = $_POST['password'];
$register_info['password_confirm'] =
$_POST['password_confirm'];
$register_info['email'] = $_POST['email'];
$member_info = $model_member->register($register_info);
if(!isset($member_info['error'])) {
$model_member->createSession($member_info);
process::addprocess('reg');
showDialog('注册成功','index.php?
act=member_information&op=member','succ',$synstr,3);
} else {
showDialog($member_info['error']);
}
}
}

```

2. 买家登录模块

登录模块是电子商务网站最基本功能之一，登录以后网站可以识别用户是否有权限进行某些操作。登

录的主要功能是检验用户信息是否存在、建立会员 session，以及登录需要进行的其他操作，例如登录送积分。登录功能代码如代码清单3-2所示。

代码清单3-2 登录功能代码

```
/**
 * 登录操作
 */
public function indexOp(){
//检查是否会员已经登录
$model_member->checkloginMember();
//验证是否为合法提交，如果不是合法提交，或者超出提交次数，则终止操作
$result = chksubmit(true,C('captcha_status_login'),'num');
if ($result !== false){
showDialog('登录失败','', 'error',$script,2);
}
//操作过于频繁，终止操作
if (process::islock('login')) {
showDialog('您的操作过于频繁，请稍后再
试',SHOP_SITE_URL, '', 'error',$script,2);
}
//查询会员记录
$array= array();
$array['member_name']= $_POST['user_name'];
$array['member_passwd']= md5($_POST['password']);
$member_info = $model_member->infoMember($array);
if(is_array($member_info) and !empty($member_info)) {
//会员已失效
if(!$member_info['member_state']){
showDialog('账号被停用');
}
}else{
process::addprocess('login');
showDialog('登录失败');
}
}
```

```
//建立会员session
$model_member->createSession($member_info);
process::clear('login');
//添加会员积分
$model_member->addPoint($member_info);
if ($_GET['inajax'] == 1){
showDialog('登录成功',$_POST['ref_url'],'succ');
} else {
redirect($_POST['ref_url']);
}
}
```

3. 退出登录

如代码清单3-3所示，在安全退出后，网站存储的用户cookie或session信息将自动清除，在某种程度上保证了用户信息的安全。

代码清单3-3 退出代码

```
/**
 * 退出操作
 */
public function logoutOp(){
    //释放session
    session_unset();
    session_destroy();
    setNcCookie('goodsnum','',-3600);
    redirect('index.php');
}
```

3.1.4 功能体验

会员由买家和卖家两部分构成。下面我们就来体验一下，买家注册及登录和卖家开店申请及登录功能。

1. 买家注册及登录

(1) 会员注册

会员注册页面除了需要填写必要的会员信息外，关于会员资料的信息填写要尽量简洁，节省注册时间。另外页面中还应该加入关于登录和找回密码的相关流程入口，如图3-2所示。



图3-2 会员注册页面

(2) 会员登录

在会员登录页面中，需要显示的内容包括用户名、密码及验证码，注册、忘记密码的流程入口，第三方合作伙伴的登录入口等。展示方式如图3-3所示。



图3-3 会员登录页面

2. 卖家申请开店及登录

(1) 开店申请操作说明

开店操作说明页面显示店铺入驻流程、入驻标准及指南，以及入驻申请页面入口，如图3-4所示。



图3-4 开店申请操作说明页面

(2) 店铺入驻协议

该页面功能很简单，只是显示卖家入驻的协议。互相同意协议是双方合作的基础条件，需要重点提示，故设置了单独的页面显示，如图3-5所示。



图3-5 店铺入驻协议页面

(3) 提交申请

进入填写资料提交申请流程，需要填写的信息有公司资质信息、财务资质信息、店铺经营信息。信息提交成功后等待后台管理员对申请信息进行审核。由于三个页面都为信息提交页面，展示类型一样，所以

下面仅以店铺经营信息页面为例进行展示，如图3-6所示。

商家入驻申请

- 入驻流程
- 签订入驻协议
- 提交申请
 - 公司资质信息
 - 财务资质信息
 - 店铺经营信息**
- 合同签订及缴费
- 店铺开通

平台联系方式

电话1: 23456789
电话2: 88997788
邮箱: abc@shopnc.net

进度条: 签订入驻协议 | 公司资质信息 | 财务资质信息 | **店铺经营信息** | 合同签订及缴费 | 店铺开通

注意事项:
店铺经营类目为商城商品分类, 请根据实际运营情况添加一个或多个经营类目。

店铺经营信息

- * 商家帐号:
此帐号为日后登录并管理商家中心时使用, 注册后不可修改, 请牢记。
- * 店铺名称:
店铺名称注册后不可修改, 请认真填写。
- * 店铺等级:
- * 开店时长:
- * 店铺分类:
请根据您所经营的内容认真选择店铺分类, 注册后商家不可自行修改。
- * 经营类目:

一级类目	二级类目	三级类目	操作

图3-6 店铺经营信息页面

(4) 合同签订及缴费

平台管理员审核通过申请信息后，就可以签订合同并提交缴费证明，然后再次等待管理员审核缴费信息，如图3-7所示。

(5) 卖家登录

经管理员审核通过缴费信息后，店铺则自动开启。卖家可以在登录页面进行店铺管理了，如图3-8所示。

商家入驻申请

签订入驻协议

提交申请

公司资质信息

财务资质信息

店铺经营信息

合同签订及缴费

店铺开通

平台联系方式

邮箱: abc@shopnc.net

签订入驻协议

公司资质信息

财务资质信息

店铺经营信息

合同签订及缴费

店铺开通

营业执照信息 (副本)

营业执照号:	328521122111
营业执照所在地:	天津 天津市 南开区
营业执照有效期:	2010-04-01 - 2015-04-30
法定经营范围:	服装、软件
营业执照号 电子版:	

店铺经营信息

卖家账号:	zyseller_seller																
店铺名称:	zyseller_seller小店																
店铺等级:	一星 (开店费用: 0.00 元/年)																
开店时长:	1 年																
店铺分类:	服装鞋帽 (开店保证金: 0 元)																
应付总金额:	0.00 元																
经营类目:	<table><thead><tr><th>分类1</th><th>分类2</th><th>分类3</th><th>比例</th></tr></thead><tbody><tr><td>服装鞋帽</td><td>女装</td><td>T恤</td><td>0%</td></tr><tr><td>服装鞋帽</td><td>男装</td><td>针织衫</td><td>0%</td></tr><tr><td>服装鞋帽</td><td>女装</td><td>卫衣</td><td>0%</td></tr></tbody></table>	分类1	分类2	分类3	比例	服装鞋帽	女装	T恤	0%	服装鞋帽	男装	针织衫	0%	服装鞋帽	女装	卫衣	0%
分类1	分类2	分类3	比例														
服装鞋帽	女装	T恤	0%														
服装鞋帽	男装	针织衫	0%														
服装鞋帽	女装	卫衣	0%														
审核意见:																	

上传付款凭证: 未选择文件

备注:

提交

图3-7 合同签订及缴费页面



图3-8 卖家登录页面

3.1.5 开发和使用

随着电子商务的发展，会员制营销正在成为企业的必然选择，建立并合理使用会员制可使企业在激烈的市场竞争中处于优势。在最佳时机，以最合适的价格和方式，为最合适的客户提供最适合的商品和服务，促使订单成交。那么如何合理使用会员制来进行营销，以及可能遇到什么问题呢，我们来探讨一下。

(1) 会员合理分层

合理使用网站系统中的会员等级模块，对会员进行分层管理。现在很多电子商务网站存在的问题是，会员等级模块已经在使用，可是在设置的等级之间不同会员享受的待遇区别不大，造成会员等级的作用没有发挥出来。网站运营者应该能从众多会员中识别出具有不同价值贡献的用户，并对应贡献值的不同设置

合理的会员等级，形成对会员差异明显的营销策略，从而吸引会员，使其在网站的生命周期延长，贡献更多的价值。

（2）运用口碑营销

如果企业为会员提供了良好的产品和服务体验，会员就会成为最好的宣传媒体，形成最初的口碑效应。扩大产品影响力，增强企业美誉度。运用会员晒单、评论和商品分享等方式，形成会员口碑营销，以活跃会员带动会员整体的活跃度。

（3）了解会员的生命周期

会员在网站存在一定的生命周期，分别为考察期、形成期、稳定期、退化期。考察期是会员对网站并没有产生信任感、没产生销售的时期。形成期是会员与网站关系快速发展的时期，会员频繁访问网站并

可能消费。稳定期是会员经过一段时间的使用后，新鲜感退去逐渐趋于平稳，但是也是信任感和黏性最高时期，关系已成熟。退化期是会员对网站失去兴趣，访问次数逐渐降低的时期。网站应时时关注会员所处的生命周期阶段，针对不同的生命周期对会员进行不同的营销策略。

（4）会员关怀

将促销变为优惠或者关怀，让会员感到归属感和亲切感，提升会员的体验。网站应收集足够的会员信息，为会员关怀提供依据。例如生日或者节日赠送礼品，发送E-mail或者短信提供促销信息或者关心祝福，让会员感到很温暖，体验度自然提升。

（5）合理全面的会员数据分析

对会员产生的数据进行分析，可以对其有更好的了解，提供有针对性的服务。在实际运营中存在数据分析不够合理、片面化的问题。数据分析一定要建立在数据量充足的情况下，从不同侧面进行分析，从而得到尽可能准确的数据结果。

当今电子商务环境下，会员生命周期价值对企业电子商务网站的建设有很大影响，例如对于B2B2C电商平台来说，买家和卖家是平台服务的两个点，作为连接两个点的纽带来说任何一个点缺失它都会受到严重影响。“以客户为中心”保持良好的客户关系并严格管控，是电子商务企业获利的必要条件。

3.2 商品模块

商品模块包括以下几个小模块：商品分类、品牌、类型、规格、属性以及商品。首先对各个小模块进行简单的介绍。

（1）商品分类

所谓的商品分类就是把商品分门别类地放置在不同的类目下面，比如：服饰鞋帽、礼品箱包、家居家装、数码办公、家用电器、个护化妆、珠宝手表、食品饮料等。商品分类是树状结构的，也就是说在每个分类下面可以定义自己的子分类，这个我们限制每个分类最多只能创建三级。品牌、类型、规格、属性与商品分类有密切的关系。

（2）品牌

每个商品都有一个自己的品牌，通过品牌用户可以查看该品牌下面的所有商品。在后台设置中，管理员需要把品牌绑定到类型上。品牌可以在平台后台和商家中心两个位置添加，商家添加需要平台审核。

（3）类型

这个功能只有在后台使用，方便管理员设置关联分类的规格、属性和品牌，可以说类型就是规格、属性、品牌与分类的桥梁，可以减少管理员的重复操作，方便快捷地修改具有同种规格和属性的分类。

（4）规格与规格值

我们这里定义的规格是直接参与商品购买的、影响商品价格的参数。比如在购买一款手机的时候，首先考虑手机的颜色、网络等，颜色和网络不同，手机的价格往往不同，所以我们就可以把价格和网络定义

成规格。规格需要后台管理员添加，具体的规格值可由商家自己定义。默认的安装程序会自带一个颜色规格，这个规格是不能删除的，这个规格影响到商品不同颜色图片的显示，具体使用方法我会在下面介绍。

（5）属性与属性值

我们这里定义的属性是不参与商品购买的参数。还以手机为例，选好了一款手机，这款手机的CPU、屏幕尺寸、分辨率等参数都是固定，我们就可以把这些参数定义为属性。

（6）商品

商品模块是整个电子商务平台的核心功能模块。任何数据，都是以商品为基础信息的。ShopNC商城系统前后三次对商品模块进行大修，最后确定以商品SKU

为一个销售单位，方便商家对不同SKU的同一种商品进行不同的优惠促销。

商品发布的具体关系如图3-9所示。

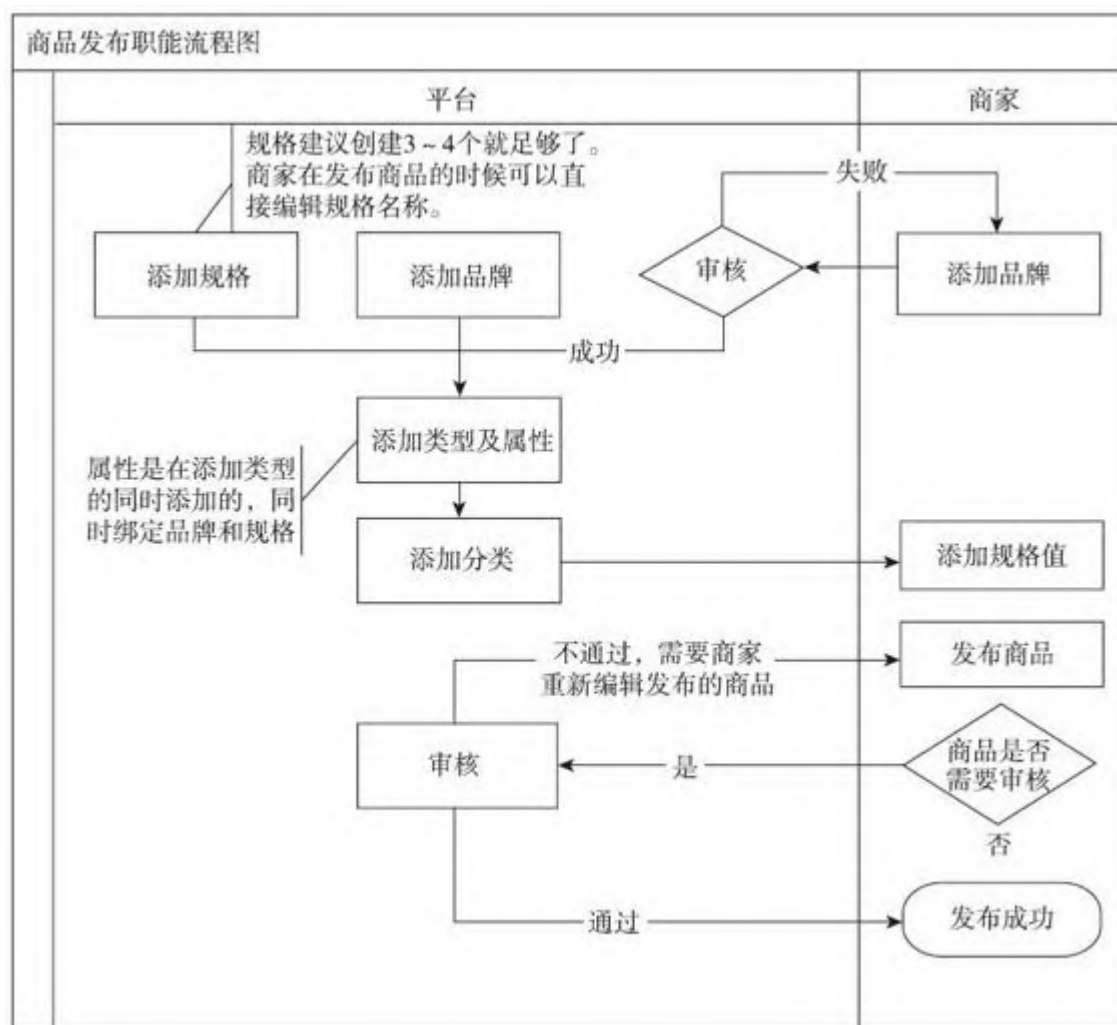


图3-9 商品发布职能流程图

3.2.1 模块构成

1. 商品分类

商品分类模块主要功能构成如图3-10所示。



图3-10 商品分类模块主要功能构成

商品分类的新增、编辑、删除操作全部由平台完成。商品分类与店铺、商品都有关联，平台可以为店铺绑定允许店铺经营的商品分类，所以删除的时候同时删除店铺绑定，同时商品需要下架。

在正常的商城运营中，商品分类是不会经常修改的，所有前台的分类显示都是读取缓存文件，减少查

库操作，提高代码执行效率。

2. 品牌

品牌模块主要功能构成如图3-11所示。



图3-11 品牌模块主要功能构成

品牌主要由现行平台创建。如果商家在发布商品的时候发现没有自己需要的品牌，可以自己添加申请，但是需要平台审核。

3. 规格与规格值

规格模块主要功能构成如图3-12所示。

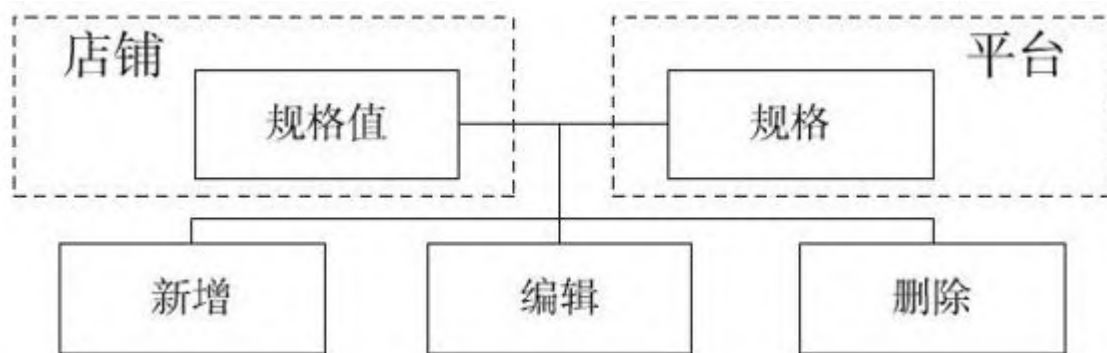


图3-12 规格模块主要功能构成

平台可以对规格进行增删改操作，店铺只能通过平台设置好的规格添加自己的规格值。在删除规格时需要同时删除规格值。

4. 类型与属性

类型与属性主要功能构成如图3-13所示。

类型和属性全是由平台进行操作的，在新建类型的时候同时新建属性。属性没有自己单独的列表，只能在操作类型的同时操作属性。

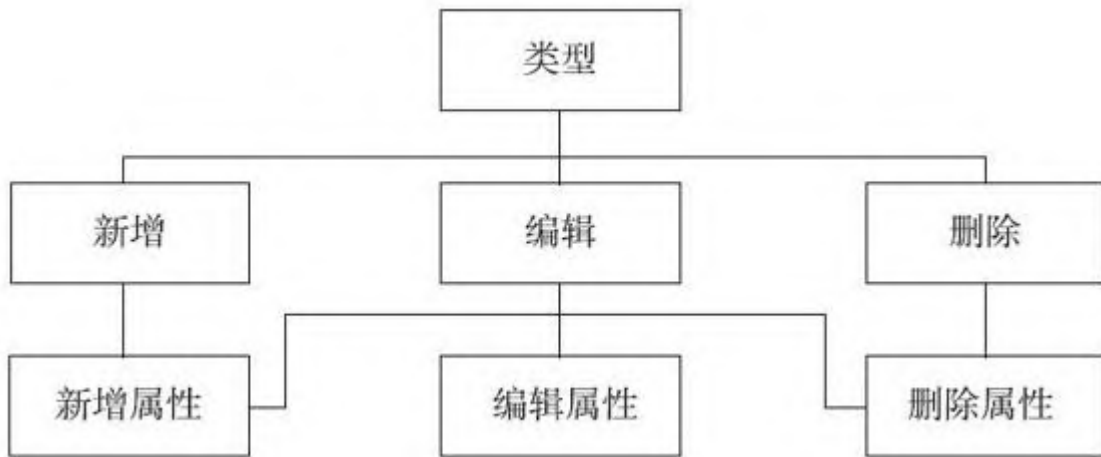


图3-13 类型与属性主要功能构成

5. 商品

商品主要功能构成如图3-14所示。

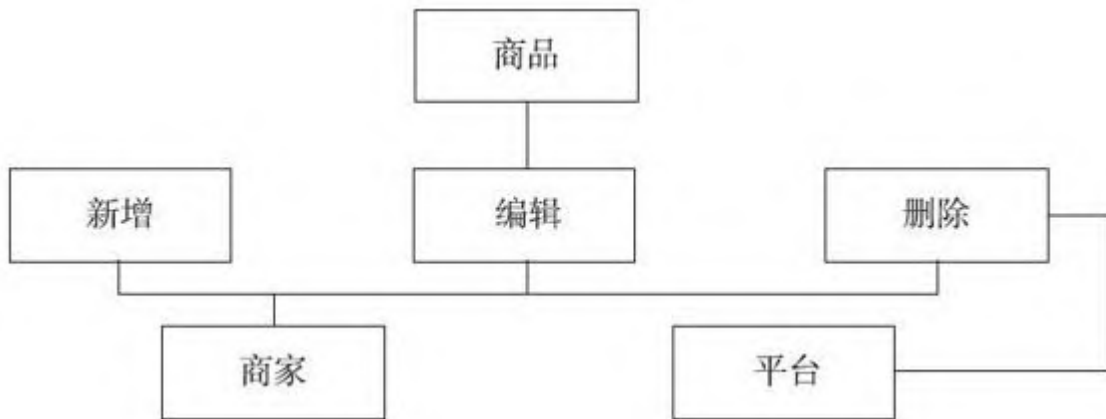


图3-14 商品主要功能构成

商品主要由店铺进行增删改。平台可以选择是否开启商品审核，在开启商品审核后商家新增和编辑商品都需要通过平台审核后才能正常销售。平台可以直接删除商品，但是不能对商品进行编辑操作。

3.2.2 设计思路

1. 商品相关数据表设计思路

商品各模块主要数据表关系如图3-15所示。

商品分类表与类型表是多对一的关系，商品分类通过类型与属性、规格、品牌产生关联。商品分类表与商品表是一对多的关系。

属性系列表包括属性表和属性值表，它们之间是一对多的关系。属性表与类型表的关系是多对一的关系，属性值表与商品表以属性与商品关系表为桥梁是多对多的关系。

规格系列表包括规格表和规格值表，它们之间是一对多的关系。规格表与类型表以类型与规格关系表

为桥梁是多对多的关系。规格表、规格值表与商品表的关系都是多对多的关系。

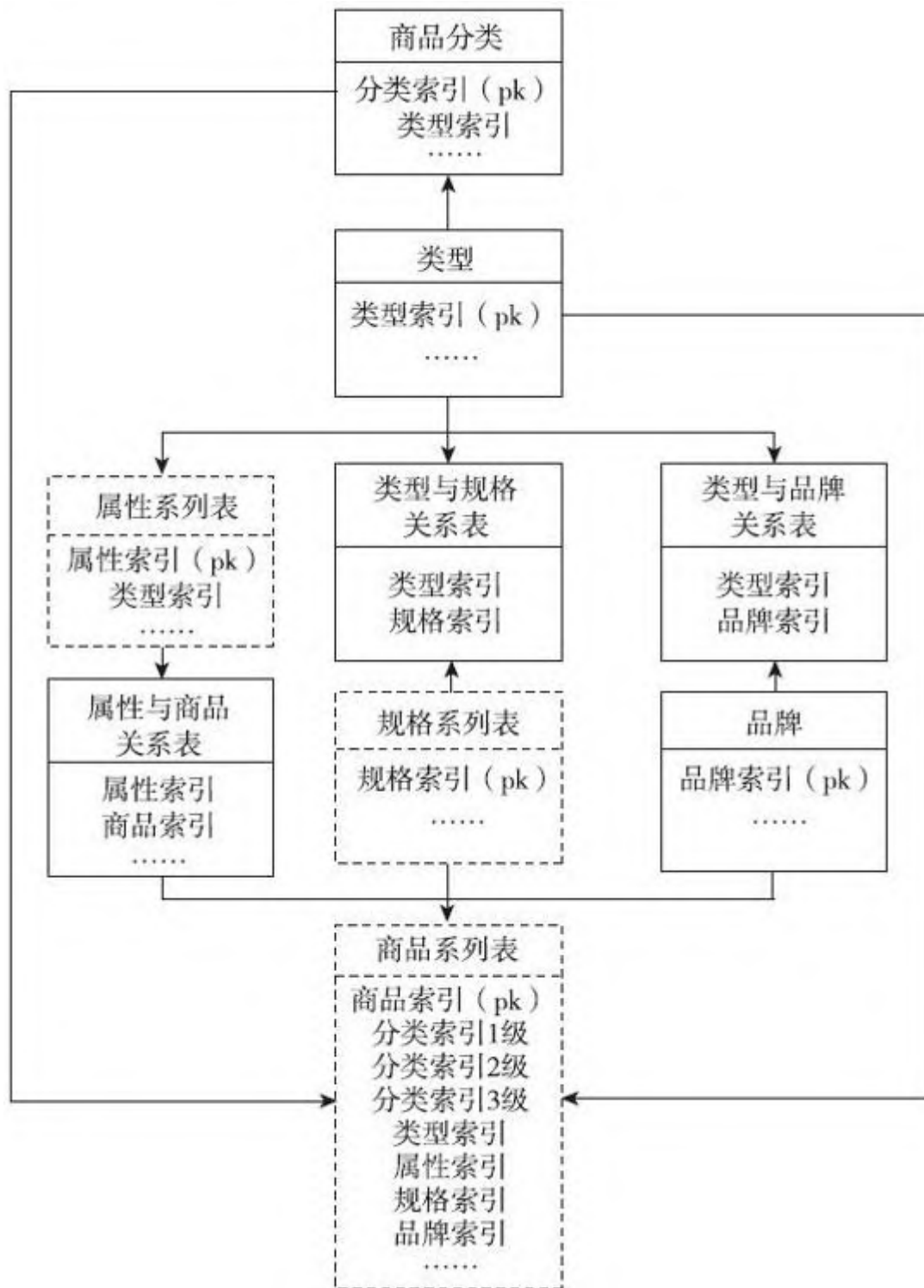


图3-15 商品各模块主要数据表关系图

品牌表以类型与品牌关系表为桥梁，与类型表是多对多的关系。品牌表与商品表是一对多的关系。

商品系列表包括商品表、商品公共表和商品图片表。商品表和商品公共表是多对一的关系，和商品图片表是多对多的关系。它们的关系如图3-16所示。

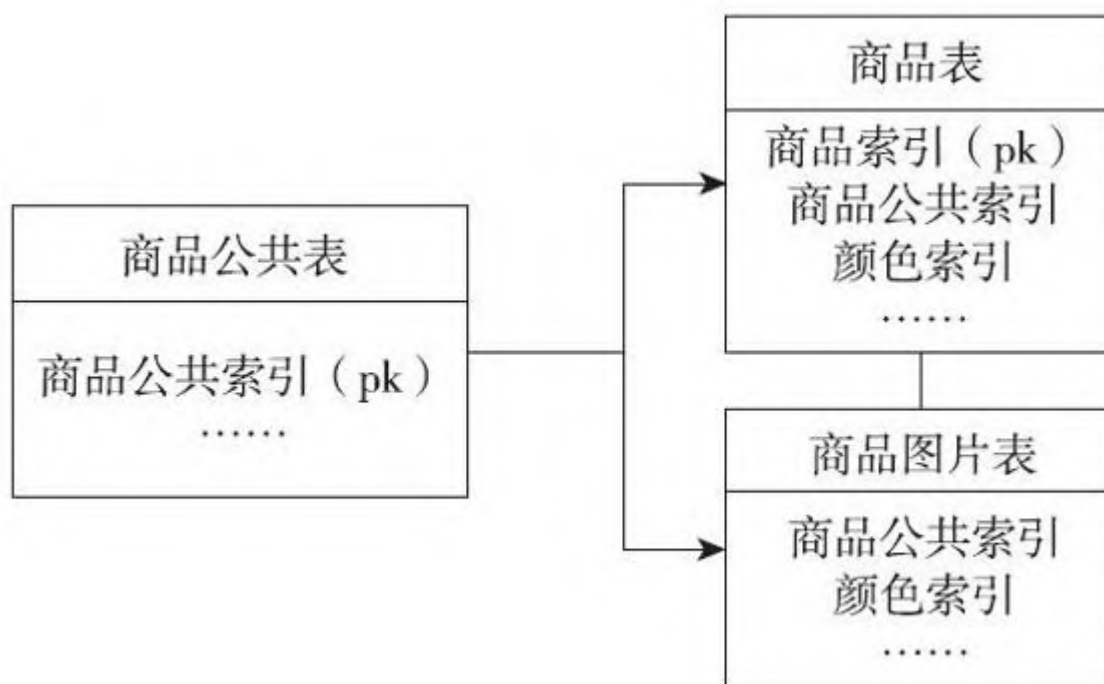


图3-16 商品系列表关系结构图

在图3-16中，商品表和商品图片表的颜色索引就是规格索引为1的规格值索引。在规格表中索引值为1

的规格为默认安装数据且平台不能删除。商品表和商品图片表是通过商品公共索引和颜色索引共同关联的。商品图片表这样设计的目的是使不同颜色的同一种商品可以单独上传不同颜色的图片。这时的设计是为了使向用户展示的商品图片更详细，给用户更直观的体验。

2. 平台管理商品的相关设计思路

在商家发布商品之前，平台需要先完成对商品分类、品牌、类型、规格、属性的设置，如图3-9所示的商品发布职能流程图。平台对分类、类型、规格、属性的修改都会影响到商品，所以在网站搭建以后需要平台管理员做一些前期的准备工作，商品分类、品牌、类型、规格、属性的添加。在正式运营以后，不建议删除商品分类、品牌、类型、规格、属性信息，但是可以新增和编辑。

删除商品分类后，同时删除店铺绑定的经营类目，该分类下面的商品违规下架，需要商家重新编辑以后才能正常销售。品牌、类型、规格、属性删除不会影响到商品的正常销售，但是在店铺重新编辑商品时，商品可能会丢失一些信息。比如，平台删除了某个规格，商家再编辑商品就需要重新设置一下商品的规格。

3. 商家发布商品的设计思路

商家是商品的直接发布者，在平台做好前期的工作后，商家就可以直接发布商品。ShopNC商城系统的商品是以SKU为销售单位，同种商品的不同SKU单独销售，方便商家根据不同SKU设置不同的促销活动。

ShopNC商城系统设计了多个小功能方便商户管理商品，下面介绍其中几个功能。

（1）设置规格值

平台设置好规格后，还需要商家自己设置规格值。为方便商家使用，有两个地方可以设置，直接在导航菜单的商品规格中设置或在商品发布第二步设置。

（2）商品图片

在设计商品图片的时候考虑到调用商品图片的地方很多，对图片尺寸的需要大小不一，为了节省流量提高访问速度，对商家上传的商品图片都进行了裁切处理，生成60px*60px、240px*240px、360px*360px、1280px*1280px四种不同大小的商品图片供不同的地方使用。需要注意对于1280px*1280px大小的图片，将宽度设置成了1280px，这样就可以使图片在宽度不超过1280px的情况下，长度能够达到1280px，方便商品描述调用图片。

（3）图片空间

把所有与商品有关的图片全部存入图片空间，方便商家同意管理图片。电子商务网站的商品展示主要是靠商品图片让买家了解商品的。随着商家的商品越来越多，商品图片的数量也会越来越庞大，商家管理商品图片也会越来越困难，因此制作图片空间来方便商家对商品图片的管理。

（4）库存报警

商家可以对不同SKU商品设置预警值。当库存不足时，通过多种方式提醒商家补货。

（5）关联板式

把相同的商品描述部分，单独拿出来统一管理。可以选择插入商品描述的头部或者尾部。

4. 用户检索商品的设计思路

为了方便用户对商品的检索，ShopNC商城系统设计制作了商品检索功能。在平台完成商品分类、类型、规格、属性设置后，用户就可以在前台通过商品分类、属性搜索商品。在商家添加商品时，商品的规格值会自动拼写到了商品名称后，在搜索时可以通过模糊搜索直接搜索具有某一规格值的商品。

在系统默认配置下，前台商品搜索采用时时查询数据库的方式处理，在大数据高并发下，这样的处理方式会对数据库造成较大的压力，所以ShopNC商城系统结合三方搜索引擎制作了全文检索功能。在开启全文检索后，系统会定时将商品的信息索引到搜索服务器（只索引前台搜索顺序相关的字段），当前台搜索某关键词、分类、品牌或属性时，系统会根据索引查

找搜索引擎，然后返回商品SKU，最后系统将这些商品SKU作为条件查询数据库，显示查询结果。

3.2.3 代码实现

1. 商品分类的代码实现

这部分代码主要是执行商品分类增加、编辑、删除操作。编辑时需要注意把分佣比例和类型关联到子集功能，如代码清单3-4所示。

代码清单3-4 后台分类编辑功能关联到子集部分

```
// 更新该分类下子分类的所有分佣比例
if ($_POST['t_commis_rate'] == '1'){
    $gc_id_list = $model_class-
>getChildClass($_POST['gc_id']);
    if (is_array($gc_id_list) && !empty($gc_id_list)) {
        $gc_ids = array();
        foreach ($gc_id_list as $val){
            $gc_ids[] = $val['gc_id'];
        }
        if (!empty($gc_ids)) {
            $model_class-
>editGoodsClass(array('commis_rate'=>$update_array['commis_r
ate']),array('gc_id'=>array('in',$gc_ids)));
        }
    }
}
// 更新该分类下子分类的所有类型
if (isset($_POST['t_associated']) && $_POST['t_associated']
== '1'){
```

```

        $gc_id_list = $model_class-
>getChildClass($_POST['gc_id']);
        if (is_array($gc_id_list) && !empty($gc_id_list)){
            $gc_id_str = '';
            foreach ($gc_id_list as $val){
                //只有更改了类型，才会下架子分类商品
                if ($val['gc_id'] != $_POST['gc_id'] &&
$_POST['t_id'] != $_POST['old_type_id']) {
                    $gc_id_str .= $val['gc_id'].',';
                }
            }
            $gc_id_str = rtrim($gc_id_str, ',');
            if ($gc_id_str != ''){
                $model_class-
>updateWhere(array('type_id'=>intval($_POST['t_id']),
'type_name'=>trim($_POST['t_name'])),
array('in_gc_id'=>$gc_id_str));
            }
        }
    }
}

```

其他的商品分类操作参考代码清单3-5。

代码清单3-5 商品分类核心操作

```

//商品分类列表
public function getGoodsClassList($condition, $field = '*')
{
    $result = $this->table('goods_class')->field($field)-
>where($condition)-> order('gc_parent_id asc,gc_sort
asc,gc_id asc')->limit(false)->select();
    return $result;
}
//商品分类详细信息
public function getGoodsClassInfo($condition, $field = '*')
{
    $result = $this->field($field)->where($condition)-
>find();
}

```

```

return $result;
}
//编辑商品分类
public function editGoodsClass($data = array(), $condition =
array()) {
return $this->where($condition)->update($data);
}
//取得店铺绑定的分类
public function getGoodsClass($store_id, $pid = 0, $deep =
1) {
    // 读取商品分类
    $gc_list = $this->getGoodsClassList(array('gc_parent_id'
=> $pid), 'gc_id, gc_name, type_id');
    // 如果店铺ID不为商城店铺的话, 读取绑定分类
    if (!checkPlatformStore()) {
        $gc_list = array_under_reset($gc_list, 'gc_id');
        $model_storebindclass = Model('store_bind_class');
        $gcid_array = $model_storebindclass-
>getStoreBindClassList(array('store_id' => $store_id,
'state' => 1), '', "class_{$deep} asc", "distinct
class_{$deep}");
        if (!empty($gcid_array)) {
            $tmp_gc_list = array();
            foreach ($gcid_array as $value) {
                if (isset($gc_list[$value["class_{$deep}"]])) {
                    $tmp_gc_list[] =
$gc_list[$value["class_{$deep}"]];
                }
            }
            $gc_list = $tmp_gc_list;
        } else {
            return array();
        }
    }
    return $gc_list;
}
//删除商品分类
public function delGoodsClass($condition) {
return $this->where($condition)->delete();
}
//删除商品分类及相关联数据
public function delGoodsClassByGcIdString($gcids) {
    $gcids = explode(',', $gcids);
    if (empty($gcids)) {
        return false;
    }
}

```

```

    }
    $goods_class = H('goods_class') ? H('goods_class') :
H('goods_class', true);
    $gcid_array = array();
    foreach ($gcids as $gc_id) {
        $schild = (!empty($goods_class[$gc_id]['child'])) ?
explode(',', $goods_class[$gc_id]['child']) : array();
        $schildchild = (!empty($goods_class[$gc_id]
['childchild'])) ? explode(',', $goods_class[$gc_id]
['childchild']) : array();
        $gcid_array = array_merge($gcid_array,
array($gc_id), $schild, $schildchild);
    }
    // 删除商品分类
    $this->delGoodsClass(array('gc_id' =>array('in',
$gcid_array)));
    // 删除常用商品分类
    Model('goods_class_staple')-
>delStaple(array('gc_id_1|gc_id_2|gc_id_3' =>array('in',
$gcid_array)));
    // 删除分类tag表
    Model('goods_class_tag')-
>delGoodsClassTag(array('gc_id_1|gc_id_2|gc_id_3'
=>array('in', $gcid_array)));
    // 删除店铺绑定分类
    Model('store_bind_class')-
>delStoreBindClass(array('class_1|class_2|class_3'
=>array('in', $gcid_array)));
    // 商品下架
    Model('goods')-
>editProducesLockUp(array('goods_stateremark' =>'商品分类被删
除，需要重新选择分类'), array('gc_id' =>array('in',
$gcid_array)));
    return true;
}

```

其中需要注意的就是在删除商品分类时需要清理与商品分类相互关联的数据，见代码清单3-1的delGoodsClassByGcIdString()方法。

在前台调用商品分类直接读取缓存文件，不进行查库操作，代码如下：

```
$goods_class = H('goods_class') ? H('goods_class') :  
H('goods_class', true);
```

2. 品牌的代码实现

品牌核心操作主要就是对品牌的增加、编辑、删除操作，如代码清单3-6所示。

代码清单3-6 品牌核心操作

```
//添加品牌  
public function addBrand($insert) {  
    return $this->insert($insert);  
}  
//编辑品牌  
public function editBrand($condition, $update) {  
    return $this->where($condition)->update($update);  
}  
//删除品牌  
public function delBrand($condition) {  
    $brand_array = $this->getBrandList($condition,  
    'brand_id,brand_pic');  
    $brandid_array = array();  
    foreach ($brand_array as $value) {  
        $brandid_array = $value['brand_id'];  
    }  
    @unlink(BASE_UPLOAD_PATH.DS.ATTACH_BRAND.DS.$value['brand_pi
```

```

c'']);
    }
    return $this->where(array('brand_id' =>array('in',
$brandid_array)))->delete();
}
//查询品牌数量
public function getBrandCount($condition) {
return $this->where($condition)->count();
}
//品牌列表
public function getBrandList($condition, $field = '*', $page
= 0, $order = 'brand_sort asc, brand_id desc', $limit = '')
{
return $this->where($condition)->field($field)-
>order($order)->page($page)->limit($limit)->select();
}
//通过的品牌列表
public function getBrandPassedList($condition, $field = '*',
$page = 0, $order = 'brand_sort asc, brand_id desc', $limit
= '') {
    $condition['brand_apply'] = 1;
    return $this->getBrandList($condition, $field, $page,
$order, $limit);
}
//未通过的品牌列表
public function getBrandNoPassedList($condition, $field =
'*', $page = 0) {
    $condition['brand_apply'] = 0;
    return $this->getBrandList($condition, $field, $page);
}
//取单个品牌内容
public function getBrandInfo($condition, $field = '*') {
return $this->field($field)->where($condition)->find();
}

```

在使用中发现，品牌过多导致用户选择品牌极其不便，所以制作了一个品牌快速检索模块，方便用户快速选自品牌，见代码清单3-7和代码清单3-8。

代码清单3-7 AJAX查询品牌PHP部分

```
//AJAX查询品牌
public function ajax_get_brandOp() {
    $initial = trim($_GET['letter']);
    $keyword = trim($_GET['keyword']);
    $type = trim($_GET['type']);
    if (!in_array($type, array('letter', 'keyword')) || ($type
    == 'letter' &&empty($initial)) || ($type == 'keyword'
    &&empty($keyword))) {
    echo json_encode(array());die();
    }
    // 实例化模型
    $model_type = Model('type');
    $where = array();
    // 验证类型是否关联品牌
    if ($type == 'letter') {
    switch ($initial) {
        case 'all':
            break;
        case '0-9':
            $where['brand_initial'] = array('in',
            array(0,1,2,3,4,5,6,7,8,9));
            break;
        default:
            $where['brand_initial'] = $initial;
            break;
        }
    } else {
        $where['brand_name|brand_initial'] = array('like',
        '%' . $keyword . '%');
    }
    $brand_array = Model('brand')->getBrandPassedList($where,
    'brand_id,brand_name,brand_initial', 0, 'brand_initial asc,
    brand_sort asc');
    echo json_encode($brand_array);die();
}
```

代码清单3-8 AJAX查询品牌JS部分

```
// AJAX选择品牌
(function($) {
$.fn.brandinit = function(options){
var brand_container = $(this);
    //根据首字母查询

$(this).find('.letter[nctype="letter"]').find('a[data-
letter]').click(function(){
var _url = $(this).parents('.brand-index:first').attr('data-
url');
var _letter = $(this).attr('data-letter');
var _search = $(this).html();
$.getJSON(_url, {type : 'letter', letter : _letter},
function(data){

$(brand_container).insertBrand({param:data,search:_search});
    });
    // 根据关键字查询

$(this).find('.search[nctype="search"]').find('a').click(fun
ction(){
var _url = $(this).parents('.brand-index:first').attr('data-
url');
var _keyword = $('#search_brand_keyword').val();
$.getJSON(_url, {type : 'keyword', keyword : _keyword},
function(data){

$(brand_container).insertBrand({param:data,search:_keyword})
;
    });
    });
    // 选择品牌
$(this).find('ul[nctype="brand_list"]').on('click',
'li', function(){
    $('#b_id').val($(this).attr('data-id'));
    $('#b_name').val($(this).attr('data-name'));
    });
    //搜索品牌列表滚条绑定
```

```
$(this).find('div[nctype="brandList"]').perfectScrollbar();
};
$.fn.insertBrand = function(options) {
    //品牌搜索容器
    var dataContainer = $(this);

    $(dataContainer).find('div[nctype="brandList"]').show();

    $(dataContainer).find('div[nctype="noBrandList"]').hide();
    var _ul = $(dataContainer).find('ul[nctype="brand_list"]');
    _ul.html('');
    if ($.isEmptyObject(options.param)) {

        $(dataContainer).find('div[nctype="brandList"]').hide();

        $(dataContainer).find('div[nctype="noBrandList"]').show().find('strong').html(options.search);
        return false;
    }
    $.each(options.param, function(i, n){
        $('<li data-id="' + n.brand_id + '" data-name="'
        + n.brand_name + '"><em>' + n.brand_initial + '</em>' +
        n.brand_name + '</li>').appendTo(_ul);
    });
    //搜索品牌列表滚条绑定

    $(dataContainer).find('div[nctype="brandList"]').perfectScrollbar('update');
};
})(jQuery);
```

3. 类型和属性代码实现

类型的操作比较复杂，需要同时考虑与其关联的品牌、规定和属性。类型与品牌是多对多的关系，类

型与规格是多对多的关系，类型与属性是一对多的关系。

在新增类型时需要新增属性并添加与品牌、规格和属性的对应关系数据，如代码清单3-9所示。

代码清单3-9 保存类型部分代码

```
if (chksubmit()){
.....
    $type_array = array();
    $type_array['type_name']      = trim($_POST['t_name']);
    $type_array['type_sort']      = trim($_POST['t_sort']);
    $type_array['class_id']       = $_POST['class_id'];
    $type_array['class_name']     = $_POST['class_name'];
    $type_id = $model_type->typeAdd('type',$type_array);

.....
    //添加类型与品牌对应
    if(!empty($_POST['brand_id'])){
        $brand_array = $_POST['brand_id'];
        $return = $model_type->typeRelatedAdd('type_brand',
        $brand_array, $type_id);
.....
    }
    //添加类型与规格对应
    if(!empty($_POST['spec_id'])){
        $spec_array = $_POST['spec_id'];
        $return = $model_type->typeRelatedAdd('type_spec',
        $spec_array, $type_id);
.....
    }
    //添加类型属性
    if(!empty($_POST['at_value'])){
        $attribute_array = $_POST['at_value'];
```

```

        foreach ($attribute_array as $v){
            if($v['value'] != ''){
.....
                //添加属性
                $attr_array = array();
.....
                $attr_id = $model_type-
>typeAdd('attribute',$attr_array);
.....
                //添加属性值
                $attr_value = explode(',', $v['value']);
                if (!empty($attr_value)) {
                    $attr_array = array();
                    foreach ($attr_value as $val) {
                        $tpl_array = array();
                        $tpl_array['attr_value_name'] =
$val;
.....
                        $attr_array[] = $tpl_array;
                    }
                    $return = Model('attribute')-
>addAttributeValueAll($attr_array);
.....
                }
            }
        }
    }
}
.....

```

在删除类型时也需要同时考虑删除相关联的数据，如代码清单3-10所示。

代码清单3-10 删除类型部分代码

```

// 删除属性值表
$attr_list = $model_type->typeRelatedList('attribute',

```

```

array('in_type_id'=>$id), 'attr_id');
if(is_array($attr_list) && !empty($attr_list)){
    $attrs_id = '';
    foreach ($attr_list as $val){
        $attrs_id .= ''. $val['attr_id'].',';
    }
    $attrs_id = trim($attrs_id, ',');
    $return1 = $model_type->delType('attribute_value',
array('in_attr_id'=>$attrs_id)); //删除属性值
    $return2 = $model_type->delType('attribute',
array('in_attr_id'=>$attrs_id)); //删除属性
.....
}
//删除对应品牌
$return = $model_type->delType('type_brand',
array('in_type_id'=>$id));
.....
//删除对应规格
$return = $model_type->delType('type_spec',
array('in_type_id'=>$id));
.....
//删除类型
$return = $model_type->delType('type',
array('in_type_id'=>$id));
.....

```

类型的新增和编辑操作比较复杂，具体流程可以参考图3-17。

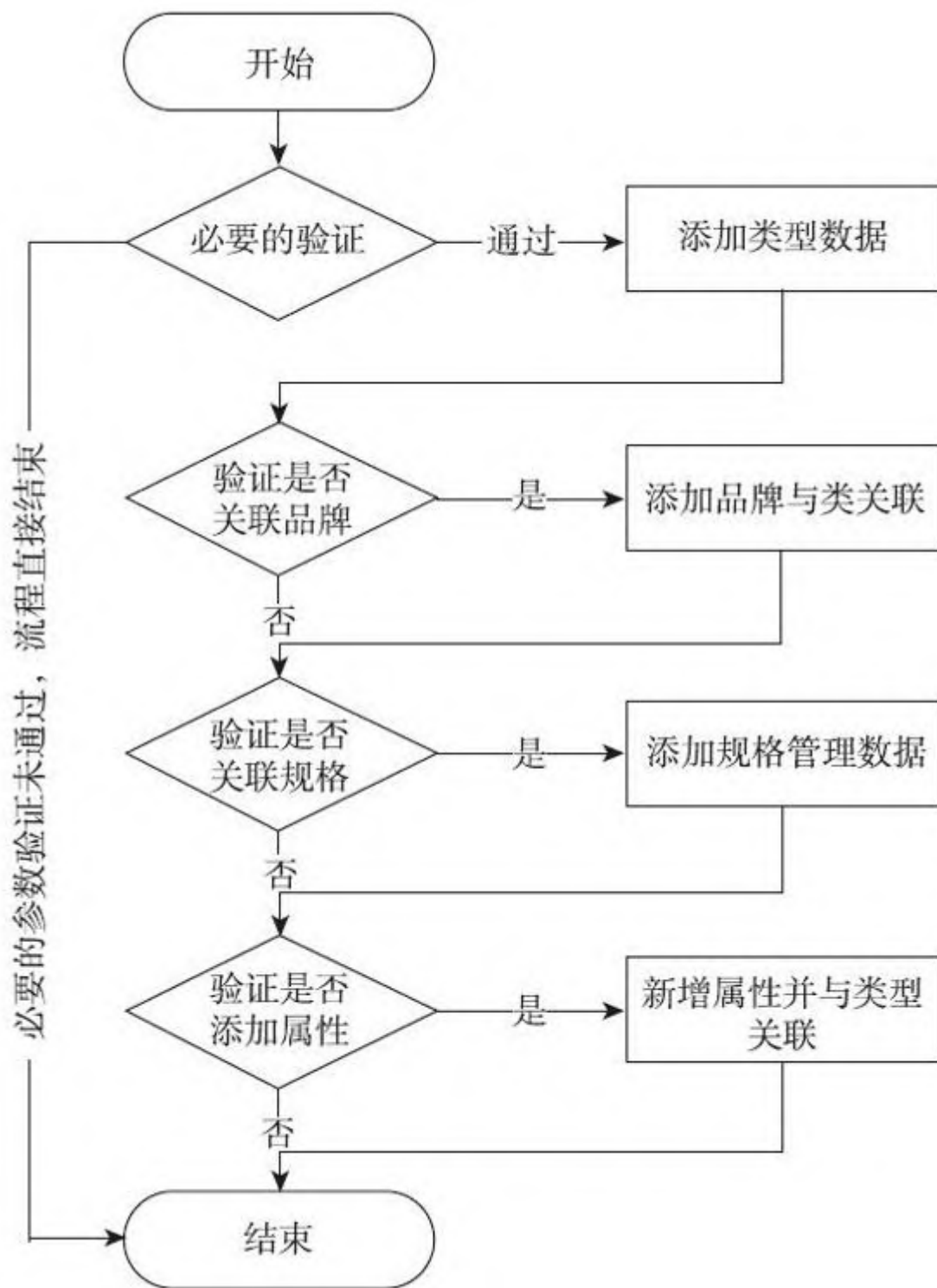


图3-17 类型保存操作流程图

4. 规格和规格值的代码实现

这部分实现的主要是后台对商品规格的增加、编辑、删除操作。在商家中心，商家可以对每个分类下的规格添加自己需要的规格值。基础方法如代码清单3-11所示。

代码清单3-11 规格基础代码

```
//单条规格信息
public function getSpecInfo($id, $field = '*') {
    return $this->field($field)->find($id);
}
//规格值列表
public function getSpecValueList($where, $field = '*',
$order = 'sp_value_sort asc,sp_value_id asc') {
    $result = $this->table('spec_value')->field($field)-
>where($where)->order($order)->select();
    return empty($result) ? array() : $result;
}
//更新规格值
public function editSpecValue($update, $where) {
    $result = $this->table('spec_value')->where($where)-
>update($update);
    return $result;
}
//添加数据
public function addSpecValue($insert) {
    $result = $this->table('spec_value')->insert($insert);
    return $result;
}
//添加数据多条
public function addSpecValueALL($insert) {
```

```
        $result = $this->table('spec_value')->insertAll($insert);
        return $result;
    }
    //删除规格值
    public function delSpecValue($where) {
        $result = $this->table('spec_value')->where($where)->delete();
        return $result;
    }
}
```

5. 商品代码实现

这部分实现的主要是由商家对商品进行新增、编辑、删除操作，在平台只能查看和删除，不能新增和编辑。

在商品表的数据结构中使用了两个字段来表示商品的状态，每个字段存在三个值，代码如下：

```
const STATE1 = 1;           // 出售中
const STATE0 = 0;           // 下架
const STATE10 = 10;         // 违规
const VERIFY1 = 1;          // 审核通过
const VERIFY0 = 0;          // 审核失败
const VERIFY10 = 10;        // 等待审核
```

为了方便二次开发，把所有的状态全部封装到模型（Model）中，想调用什么状态的代码，直接调用相应的方法就可以了，如代码清单3-12所示。

代码清单3-12 商品模型部分代码

```
//商品列表卖家中心使用
// 这个方式是用来查询全部商品列表的
public function getGoodsCommonList($condition, $field = '*',
$page = 10, $order = 'goods_commonid desc') {
    $condition = $this->_getRecursiveClass($condition);
    return $this->table('goods_common')->field($field)-
>where($condition)->order($order)->page($page)->select();
}
//出售中的商品列表卖家中心使用
// 这个方法是用来查询所有可以销售的商品
public function getGoodsCommonOnlineList($condition, $field
= '*', $page = 10, $order = "goods_commonid desc") {
    $condition['goods_state'] = self::STATE1;
    $condition['goods_verify'] = self::VERIFY1;
    return $this->getGoodsCommonList($condition, $field,
$page, $order);
}
//仓库中的商品列表（卖家中心使用）
// 这个方式是用来查询可以销售但是商家没有开始销售的商品列表
public function getGoodsCommonOfflineList($condition, $field
= '*', $page = 10, $order = "goods_commonid desc") {
    $condition['goods_state'] = self::STATE0;
    $condition['goods_verify'] = self::VERIFY1;
    return $this->getGoodsCommonList($condition, $field,
$page, $order);
}
//违规的商品列表（卖家中心使用）
// 正常出售中的商品被平台发现违规行为，禁止销售的商品列表
public function getGoodsCommonLockUpList($condition, $field
= '*', $page = 10, $order = "goods_commonid desc") {
```

```

        $condition['goods_state']    = self::STATE10;
        $condition['goods_verify']   = self::VERIFY1;
        return $this->getGoodsCommonList($condition, $field,
$page, $order);
    }
    //等待审核或审核失败的商品列表（卖家中心使用）
    // 平台开启审核后，等待平台审核或者审核是否的商品列表
    public function getGoodsCommonWaitVerifyList($condition,
$field = '*', $page = 10, $order = "goods_commonid desc") {
        if (!isset($condition['goods_verify'])) {
            $condition['goods_verify'] = array('neq',
self::VERIFY1);
        }
        return $this->getGoodsCommonList($condition, $field,
$page, $order);
    }
    // 出售中的商品SKU列表
    // 只显示不同颜色的商品（前台商品索引，店铺页商品列表等功能使用）
    public function getGoodsListByColorDistinct($condition,
$field = '*', $order = 'goods_id asc', $page = 0) {
        $condition['goods_state']    = self::STATE1;
        $condition['goods_verify']   = self::VERIFY1;
        $condition = $this->_getRecursiveClass($condition);
        $field = "CONCAT(goods_commonid,',',color_id) as
nc_distinct ," . $field;
        $count = $this->getGoodsOnlineCount($condition,"distinct
CONCAT(goods_commonid,',',color_id)");
        $goods_list = array();
        if ($count != 0) {
            $goods_list = $this->getGoodsOnlineList($condition, $field,
$page, $order, 0, 'nc_distinct', false, $count);
        }
        return $goods_list;
    }
}

```

模型中还封装了很多方法，这个就不再进行一一展示了。

3.2.4 功能体验

本小节就不再一一介绍各个功能的使用了，主要介绍一下比较复杂的类型设置、商品发布和商品检索。

1. 类型管理

在添加类型之前需要先创建好商品分类、品牌、规格。

登录ShopNC商品后台，选择商品→类型管理→新增，如图3-18所示。

点击“新增”按钮后页面跳转到类型新增页面，如图3-19所示。

在该页面上部录入类型基本信息，在中部靠左选择关联的规格，在中部靠右选择关联的品牌，在下面录入与类型相关的属性。点击“提交”按钮后保存相关信息。

类型的编辑和新增相似，这里就不再做介绍了。

2. 商品发布流程

登录商家中心，选择商品→商品发布，页面跳转到选择商品分类页，如图3-20所示。



图3-18 后台类型列表

类型管理

列表

新增

* 类型：

所属分类：

请选择...

选择分类，可关联大分类或更具体的下级分类。（只在后台快速定位中起作用）

* 排序：

0

请填写自然数，类型列表将会按照排序进行由小到大排列显示。

选择关联品牌：

请选择...

（快速定位）

选择关联品牌：

请选择...

（快速定位）

隐藏未选项

隐藏未选项

默认

☐ 颜色
☐ 尺码

默认

☐ 三光云彩
☐ 富安娜
☐ 空境大牌

☐ 韩洋家纺
☐ 安德星
☐ 好事达

☐ 美姿
☐ 罗莱
☐ 爱仕达

☐ 生活诚品
☐ 易地
☐ 慧乐家

☐ 大墅全
☐ 世家洁具
☐ 欧司朗

☐ 溢彩年华
☐ 美好家
☐ 玛克

添加属性：

排序	属性名称	属性可选项	显示	操作
0			显	删除

添加一个属性

图3-19 后台类型新增

STEP 1

选择商品分类

STEP 2

填写商品详情

STEP 3

上传商品图片

STEP 4

商品发布成功

您常用的商品分类：

请选择

服饰鞋帽

礼品箱包

家居家装

数码办公

家用电器

个护化妆

珠宝手表

食品饮料

运动健康

女装

男装

内衣

运动

女鞋

男鞋

配饰

童装

T恤

衬衫

针织衫

雪纺衫

卫衣

马甲

连衣裙

半身裙

牛仔裙

您当前选择的商品类别是：服饰鞋帽 > 女装 > 雪纺衫

下一步，填写商品信息

图3-20 商品发布第一步——选择商品分类

选择商品分类完成以后，点击下方绿色按钮，跳转到下一步，填写商品信息。在选择规格后，会自动生成一个表格，如图3-21所示。

颜色：

落月黄

蓝色

白色

绿色

粉红

黑色

橙色

+ 添加规格值

尺码：

s

m

l

xl

+ 添加规格值

库存配置：

颜色	尺码	*价格 ☒	*库存 ☒	预售值 ☒	商家货号
落月黄	s	11.00 元	1	2	
落月黄	m	11.00 元	1	2	
蓝色	s	11.00 元	1	2	
蓝色	m	11.00 元	1	2	

点击 ☒ 可批量修改所在列的列值。

图3-21 商品发布第二步——设计规格部分

录入完成商品信息后，点击下方绿色按钮，跳转到下一步，上传商品图片，如图3-22所示。

这里的图片上传是按照颜色区分的。在图3-21中，我们选择了两种颜色：落月黄、蓝色。所以在这

个步骤中需要上传两种颜色的图片。

STEP 1 选择商品分类 > STEP 2 填写商品详情 > STEP 3 上传商品图片 > STEP 4 商品发布成功

颜色: 落月黄

排序: 0 上传

排序: 0 上传

排序: 0 上传

排序: 0 上传

排序: 0 上传

从图片空间选择

颜色: 蓝色

排序: 0 上传

排序: 0 上传

排序: 0 上传

排序: 0 上传

排序: 0 上传

从图片空间选择

下一步, 确认商品发布

上传要求:

1. 请使用jpg/png/gif等格式, 单张图片大小不超过1M的正方形图片。
2. 上传图片最大尺寸分辨率建议为1280像素。
3. 每种颜色最多上传5张图片, 从图片空间中选择已有的图片, 上传后的图片也将被保存在店铺图片空间中以便复用。
4. 通过更改排序数字调整商品图片的展示顺序。
5. 图片请尽量清晰, 不能虚化, 要保证亮度充足。
6. 操作完成后请点下一步, 否则无法在网站生效。

建议:

1. 主图大白色背景正侧面。
2. 排序依次为正面图>侧面图>侧面图~细节图。

图3-22 商品发布第三步——图片上传

上传完成后, 点击“下一步, 确认商品发布”按钮, 会跳转商品发布成功页面, 如图3-23所示。



图3-23 商品发布第四步——发布完成

3.3 促销模块

随着互联网的快速发展，社会物质不断丰富，市场竞争越发激烈，消费需求也在不断增强，因而促销在市场中也占据了越来越重要的位置，并且已慢慢融入到人们消费购物的各个角落，成为人们日常生活中不可缺少的一部分。对于促销大家都不会陌生，因为你经常会在各种场合碰到各类形式不一的促销活动。你会感觉到各式各样的促销在接二连三地对你进行冲击和引诱，口袋里钱在促销的攻势下不知不觉地被掏走了，当你冲动过后再看账单的时候会惊讶道：天啊，怎么又支出了这么多钱，买了这么多无用的东西！于是心里暗暗决定一定不能为促销所动，结果再次看到账单还是会一样的惊讶，促销就是具有这种魔力，在不知不觉中让你自掏腰包。而当今电子商务飞速发展，网络促销也顺势火热起来，它已成为了一种

行之有效的促销方式，能为企业带来不小的利益。与传统促销相比，网络促销可以让顾客参加到营销活动中，因此网络促销具有交互性，它更能加强与客户的沟通和联系，更能促进对客户需求的了解，更容易得到客户的认同。任何企业都面临着如何实施网络促销这个新的课题，对产品信息在网络上传播的特点进行深入了解，分析和了解网络信息对应的接收对象，给网络促销设定合理的目标，科学的设置网络促销程序，打开网络促销的新局面，使你的事业有新的突破，不落后于时代的步伐。

淘宝的“双十一狂欢节”活动作为其首创的促销方式，它的全场商品半价的促销形式，对购物狂有非常大的吸引力，用盛宴来形容双十一也不为过。继191亿元的购物狂欢之后，2013年天猫双十一再次刷新自己的记录，最终以350.19亿元的成交额顺利收官。11日21时19分45秒，天猫双十一购物狂欢节支付宝成交

额突破300亿元，这相当于中国日均社会零售总额的4成。其中，手机淘宝的累计活跃用户达到1.13亿，支付宝交易金额达到45.8亿。本次双十一，天猫三大重要破亿时间段如下：55秒：1亿元；5小时49分：100亿元；24小时：350.19亿元。与去年的数据相比，支付宝全天成交金额比上一年的191亿元增长83%。同时，支付宝交易突破10亿元的时间今年只花了不到7分钟，而上一年则用了37分钟。据了解，10亿元已超过香港9月份日均社会零售总额。来自手机淘宝的数据显示，双十一购物狂欢节期间，用户通过手机淘宝购买的商品支付宝成交额高达53.5亿，是去年的5.6倍（去年全天成交9.6亿）；手机淘宝单日活跃用户突破1.27亿，创造了手机上单日活跃用户的全新峰值；同时，通过手机成交的商品笔数达到3590万笔，占整体交易额的21%。由此可见，促销对于电子商务的重要性，合理地使用促销将使销售额产生突破式的增长。

3.3.1 模块构成

营销的4P理论即为产品（Product）、价格（Price）、渠道（Place）、促销（Promotion），促销作为营销中的重要一环，承担着衔接沟通企业和客户之间重要桥梁的作用。促销是电子商务网站吸引客户目光的一把利器，勾起客户购买欲望，是给予客户心灵重重一击的一枚糖衣炮弹，能促使网站在残酷的市场中乘风破浪一路直上，成为市场防御的盾，是企业成长路程中前行和转变的必经之路。网站通过一系列的促销活动，将产品以最醒目的方式展现在客户面前，吸引客户，在最短的时间内增加注册会员和购买量，扩大新会员，同时巩固老会员，提升网站的黏性，以最直接和强烈的方式打击竞争对手，实现预期目标。图3-24展示了一些常用的促销方式及其具体的展现形式。

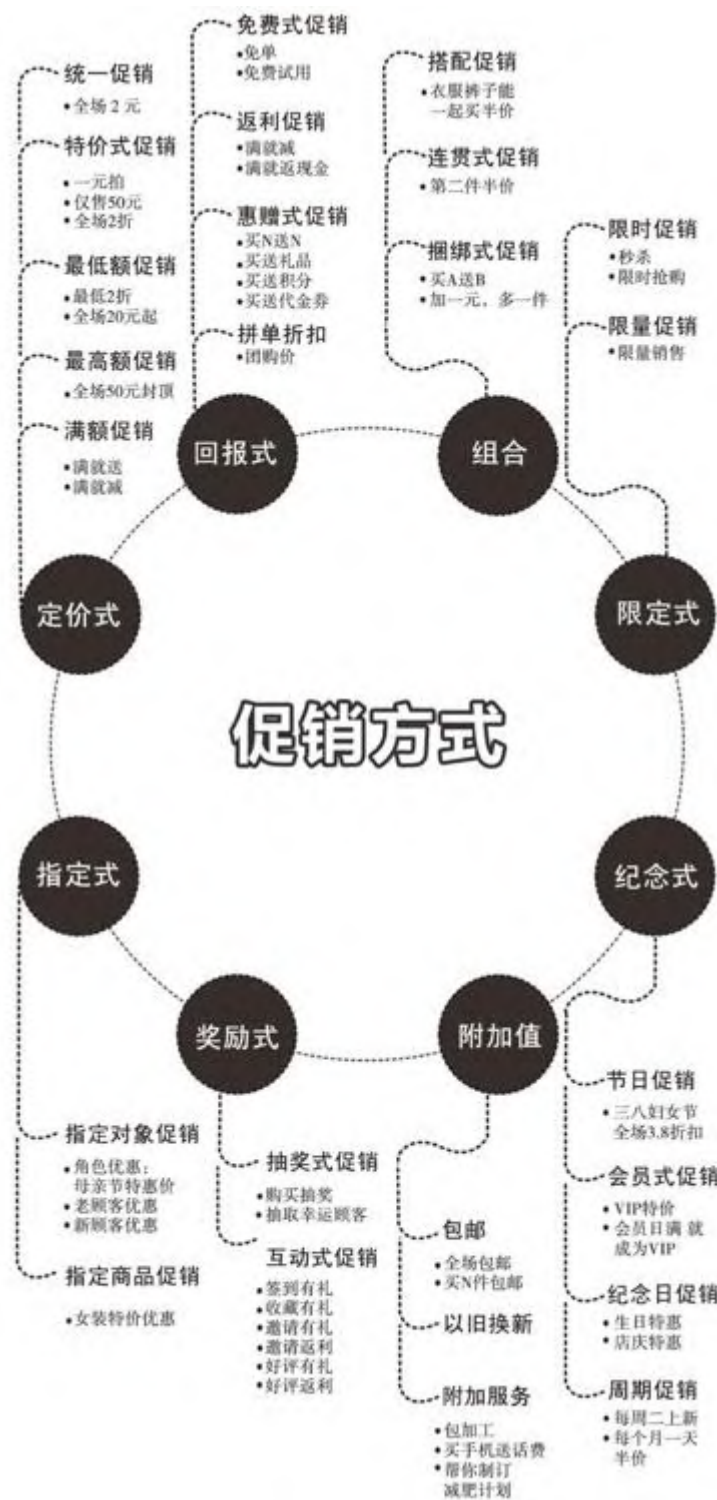


图3-24 电子商务常用的促销方式

下面介绍一下部分常用的促销方式。

（1）折价促销

价格一直都是电子商务运营中的重点，而折价促销也是电商最擅长且使用最多的促销方式。淘宝双十一、京东的8.15，如此大规模的促销都是以折价的方式体现的。折价促销的操作简单、效果显著，赢得了众多电商的青睐。而客户也很喜欢这种直接受益的促销方式，也对它最为敏感，幅度较大的折扣可以刺激客户快速消费。而且还慢慢衍生出了网上折扣店这一新兴的营销方式。折价促销有多种方式体现，例如全场2折，特定商品5折，全场2折起，等等。折价促销也有其劣势，它仅限于降低价格时吸引顾客，当价格恢复原价后可能就无人问津了。所以才出现了在淘宝双十一期间订单量激增，活动过后的一段时间内订单量

减少的情况。所以折价促销也不是万能的，要与其他促销形式搭配使用。

（2）团购——拼单式促销

团购是回报式促销的一种，它作为一种新兴的电子商务模式，通过客户自行组团、专业团购网站、商家组织团购等形式，提升客户与商家的议价能力，并极大程度地进行商品让利，引起客户及业内厂商，甚至是资本市场关注，而最常见的就是商家组织团购的形式。随着互联网的发展，团购网站在2010年如雨后春笋般的出现，走在电子商务网络购物前沿地带，已经成为了每个电子商务平台的标配促销形式。团购具备了网络购物的所有优点，如低价、方便、快捷、不受地域限制等。而价格优势又成为其最主要的核心优势，这种优势甚至要比普通网络购物更加明显。团购的好处主要表现在两方面：一是团购价格低于产品市

场最低零售价，二是产品的质量和服务能够得到有效的保证。它让每一个人都能找到更优惠的团购商品，让不相识的客户共同享受物美价廉的服务。

对于客户来说，团购的形式相当于线下的批发形式，团购价格相当于产品的批发价格。通过网络团购，可以将被动的分散购买变成主动的大宗购买，所以购买同样质量的产品，能够享受更低的价格和更优质的服务。

团购对电子商务网站来说也有很多好处，一是直接促进电子商务网站的销量；二是直接带动电子商务网站的注册会员数量增长；第三是通过促进客户的尝试购买，发现平台存在的问题，增强客户体验；第四是扩大电商网站的品牌曝光度和知名度；第五是增加老客户对网站的黏性。所以团购是一个电子商务网站必不可少的促销方式。

（3）惠赠式促销

惠赠式也是一种回报式促销形式，包含很多种实现方式，例如买一送一、买送礼品、买送积分、买送代金券等。下面来了解一下具体的促销方式。

· 买送礼品（或者买一送一、满就送礼品）：是指客户在购买某一商品或者达到一定条件的同时无偿赠送给他一件商品或者某些赠品。赠送礼品使客户在购买商品时得到意外的惊喜从而感动，促使其快速下单。赠送礼品相对于其他的降价促销来说有很多优点，第一可以提升网站和品牌的知名度；第二降价往往会给顾客以商品本身就廉价的感觉，而赠送礼品的促销形式则不会；第三赠品促销能够鼓励顾客重复消费或增加消费量；第四将新品作为赠品销售，简单地推广了新品，并可以进一步了解客户对于赠品的关注程度，评估新品是否受欢迎。一般在新产品试

用、产品更新、对抗竞争品牌、开辟新市场情况下利用赠品促销可以达到比较好的促销效果。

· 买送积分：是指客户在购买商品时根据消费金额的不同赠送不同的积分。根据不同的需要积分可以有不同的意义。比如有的网站将积分作为虚拟币使用，可以在消费的时候按照不同的比例抵现，而有的仅仅作作为一个分值，可以用分值兑换积分商品。可以根据不同的需要设置积分的不同意义。积分营销的网络应用比传统行业要方便很多，操作性和可信度都要比传统行业高，会员可以在自己的买家中心查看积分详细记录。虽然积分对客户下单的吸引力不如降价和赠送礼品，但是其作为一种电子商务很重要的促销活动有其独特的优势，积分营销是一种持续长期的促销形式，可以吸引会员多次光顾网站，从而提高网站会员黏性和忠诚度，而且其本身成本也低于其他促销活动，易于操作。

· 买送代金券：与以上促销活动一样，买送代金券就是指客户在购买商品的同时也获得了一张固定面额的代金券。代金券的本质其实就是优惠券的一种，是一个短期刺激客户的工具，它与积分（长期吸引顾客）刚好构成了日常营销的基本工具。代金券在购物中可以抵扣同等面额的现金。代金券是促进老客户再次消费和拉回即将流失的老客户的有效手段。还可以用积分兑换代金券，从而完成积分从产生到消亡的生命周期。赠送代金券不像其他的促销活动一样可以立即对客户产生回馈，而是在下次消费的时候进行优惠，这也正是其优点所在，对其他促销活动形成了有效的补充。代金券的最高境界是绑架消费，让客户买了本不打算买的东西，而且不去网上比价格，真是四两拨千斤。

（4）搭配销售

搭配销售是客户在浏览一件商品时再向他推荐其他商品，这件商品可以与其他商品搭配起来一起销售，同时总价格进行相应幅度的降低。配套的商品必须是客户需要的，且相互之间有一定的联系。搭配销售相对于其他的促销方式来说有很多的优点，一是更有利于扩大商品销量，提升客均购买件数，从而提高客单价；二是直接降价会损伤品牌价值，搭配销售降低价格既让客户得到便宜，同时不会有价格的横向比较，对原价购买的老顾客伤害更小；三是提升客户体验，让客户感到服务很周到，例如相同款式的衣服搭配销售免去了客户再去自己搭配，同时穿上的效果也更佳，更容易获得客户的满意。同时类似的销售还有连贯式促销（例如：第二件半价）、捆绑式促销（例如买A送B、加一元多一件）。

（5）限时限量促销

物以稀为贵，限量促销正是利用了人们的这个心理。“对失去某种东西的恐惧，似乎要比获得同一物品的渴望，更能激发人们的行动力”。限量销售会造成你即将失去获得某件心爱物品的机会，心情开始紧张起来，不愿意承受失去，于是不再顾虑很多，直接下单购买。由此延伸出了很多的促销方式比如“补货中”、“仅剩最后100件”、京东的“北京无货”等等。而其中以京东的地区性无货运用的最巧妙无痕。

限时促销则是对客户进行了步步紧催的心理策略，对客户获得某件商品的机会做出限制。告诉客户时间马上就到了，赶紧下单吧，再犹豫就没机会了，催促让客户一下子产生了紧张感，心里想着错过了就得不到自己喜欢的东西了，该买就买吧。将限时做到极限的促销方式就是秒杀了，秒杀时间一到客户蜂拥而上，瞬间抢购一空，没买到的客户还会觉得心里很是遗憾，而买到的会很有成就感，辛辛苦苦抢到的商

品即使不是百分之百满意也爱不释手。这就是人们心理上的通病。

限时和限量的促销都是在给客户制造紧张感，激发了他们的行动力，促使其快速下单。而且越难得到的东西，得到后越是珍惜，在心里的价值会高于其他东西，也容易对商品感到满意，为二次购买创造了机会。电子商务网站合理的运用限时限量也是很有必要的，限时限量促销容易造成很多客户同一时间涌入网站，对于网站负载、库存、客服、物流等都是不小的考验，所以在活动之前必须做好完善的准备工作，才能保证活动顺利完成。

（6）抽奖式促销

抽奖式促销是以某种奖励吸引客户积极参与调查、产品销售、注册、庆典、推广等活动。客户通过参加活动获得抽奖的机会。采用抽奖式促销能够覆盖

大范围的目标消费群体，对会员注册、销售额等有直接的拉动作用，可以吸引新顾客，也使老顾客多次光顾网站。中奖顾客在这其中不仅收货了物质，在心理上也会有获得中奖这份幸运带来的愉悦感。但是一般抽奖需要设置中奖率，中奖率设置得过低也会影响顾客的积极性。当网站为了解一些客户需求而举办调查活动时，抽奖不失为一种很有效的方式。

（7）互动式促销

随着网络的不断普及，上网人数也在不断激增。而电子商务也正朝着社区化的方向发展。会员与商家、会员与会员的互动成为了商家进行推广的一个特有的方式。随之也应运而生了一些特有的促销方式。下面来介绍一下这些新兴的促销方式中常用的几种。

- 好评有礼：在网上购物，评价是每个客户必看的，所以评价的好坏对于网站来说很重要。一条好的

评价也许就可以促成一单生意。所以商家不惜重金搞评价有礼的活动。很常见的方式就是评价送积分、评价返现金。淘宝上很多商家更是用现金买好评，当客户买了商品之后给予好评则直接返现，也是很常见的。这种促销对于提升销售是间接性的，但也是很有效的，俗话说金杯银杯不如好的口碑。而这种活动也是比较容易操作的，针对性也很强，所以在促销预算足够的情况下，使用该种促销也很有必要。

· 邀请有礼：邀请有礼这种在社区中常见的方式，慢慢地被移植到了电子商务中。老客户邀请新客户注册或者购买则得到相应的奖励。一般老客户和邀请的新客户之间在现实中是认识的，存在一定的信任感，通过老客户邀请进入网站，从而对网站也产生了一定的信任感，这种信任感一产生，则其购买的几率也就相应的增加。邀请这种方式也从另一方面降低了推广费用，且效果更佳，何乐而不为呢。

（8）附加值促销

在激烈的市场竞争中，价格战已经被广大客户所熟知，可是一味的价格战对商家来说是一种伤害，降低了利润也影响了商品的品牌价值，很容易造成一种恶性循环。随着社会物质的不断丰富，人们也不再一味的注重价格，良好的品质和优质的服务对客户的吸引力在不断地增强，在商家和客户理念的不断成熟中，附加值促销形式逐渐受到商家喜爱。

- 包邮：相对于传统行业来说，电子商务运营中的邮费是其一个弊端，客户还没收到商品就先付了邮费，从心理上来说总是有些不悦。所以当下电子商务网站都采用了包邮的形式来消除这一弊端。它也成为最常用的一种促销形式。

- 附加服务：附加服务作为价格之外的一种促销，与特价促销形成了有效的补充。常用的附加服务

有包加工、保修、7天包退、30天包换等服务。这种优质的服务更容易得到客户对品牌的认可。所以附加服务促销相对于降价的促销来说，不仅能吸引客户下单，有较低的促销费用，而且对于品牌价值的提升很有益处。这样就形成了一个良性的循环。所以建议电子商务网站应该加入不同形式的附加服务促销。

相对于降价促销来说，附加值促销更容易操作，且对品牌的提升很重要，品牌的建立需要优质的服务，正好将优惠促销和树立品牌这两个本身有点矛盾的事更好地融合起来，使电子商务变得更加轻松。

3.3.2 设计思路

电子商务网站促销通过维护老客户、吸引新客户、增加网站黏性、提升销售额和增加品牌价值，在促进网站的销售和提升活力的同时，形成了网站运营的良性循环。

1. 业务的设计原则

促销的作用主要体现在以下几个方面。

- 吸引注意力：促销通过比较吸引注意的方式，将产品、服务等信息传递给客户，引起他们的注意。

- 说服功能：通过价格、服务等方面的促销方式，说服客户消除疑虑，坚定购买。

- 反馈信息：通过促销的效果和调查抽奖等促销形式收集和汇总的信息，可以了解客户的需求和意见，而且其中很多的反馈信息是文字性的更具有可靠性，对网站运营有很高的参考价值。

- 刺激销售：好的促销可以挖掘客户的潜在需求，一条优质的促销信息，往往可以使客户产生购买的欲望，从而扩大客户需求，增加销售。

电子商务网站在建立促销时，其业务逻辑的设计原则应该遵从以上促销的几点作用来设计，使其充分发挥作用，收到良好的促销效果。

2. 模块设计实例

下面以B2B2C电商平台线上团购模块为例了解一下电商平台促销的基本模块设计。

· 套餐管理：在B2B2C电商平台中提供商品和服务的是平台上的各个店铺，而承办活动的却是平台本身。团购是作为平台的一项促销服务存在的，也是平台盈利的一个渠道。所以在团购模块中应该加入套餐管理版块，店铺在开始团购之前需要先向平台购买或者申请团购套餐，套餐规定了店铺可以参加团购活动的时间段。

· 团购管理：团购活动模块是平台统一管理的，对于参加平台上团购活动的店铺及其商品的管控显得尤为重要。平台应该可以完全控制团购中商品的上下架。平台对团购的管控体现在以下方面，第一提交参加团购的商品和相关信息，等待平台审核；第二当活动开始之后出现了问题，平台可以立即取消该团购活动。

- “即将开始”：团购一般是到某个时间才能进行的，为了在一定时间段内获得更多的客户购买，在团购页面展示“即将开始”的团购很重要，在团购活动经过平台审核之后就可以展示到“即将开始”版块中，延长活动的展示时间以获得更多客户的注意，让客户记住某某时间点将会开始某商品的团购，记得准时参加。

- 搜索设置：团购信息一般是将所有的团购活动不分类别的杂乱展示，这是本着公平公正和大多数客户的需要而展示的，而有些客户是有明确的购买意愿的，所以增加必要的搜索是很好的用户体验。所以团购中应该加入一些搜索项。比如团购商品分类、价格区间等。

- 详细信息：团购商品详细页面需要展示团购商品的图片、描述、状态、评价和已购买信息等。该页

面作为团购商品的展示页面一般会展示其比较吸引顾客的点，所以商品图片、价格、描述等会设计的比较醒目，易于引起客户的购买欲。一般团购商品的促销图和商品描述信息都是在店铺申请团购活动的时候单独设置的，是不同于商品详细页面的信息的。这点在制作团购模块的时候一定要注意。再者由于团购是一个限时限量的活动，不是长期存在的，其状态也有多种。包括即将开始、进行中、已完成（团购圆满完成）、已结束（团购由于人数不够或者其他原因中途结束）等。所以在团购活动详细页面展示清晰可见的状态对于客户来说也很有必要。

· 团购订单：线上团购与线下团购的订单流程是不同的，相反线上团购与一般的订单操作是一致的。所以在B2B2C电商平台中将线上团购订单与一般订单都统一到了订单模块中。只是在订单信息中标注其为

团购活动产生的订单，便于商家查看订单活动产生的相应团购订单，评测团购活动的效果。

图3-25即为线上团购模块构成及跨职能流程展示图，按照平台、店铺、买家三种身份在团购中不同的职能来描述线上团购整个过程。

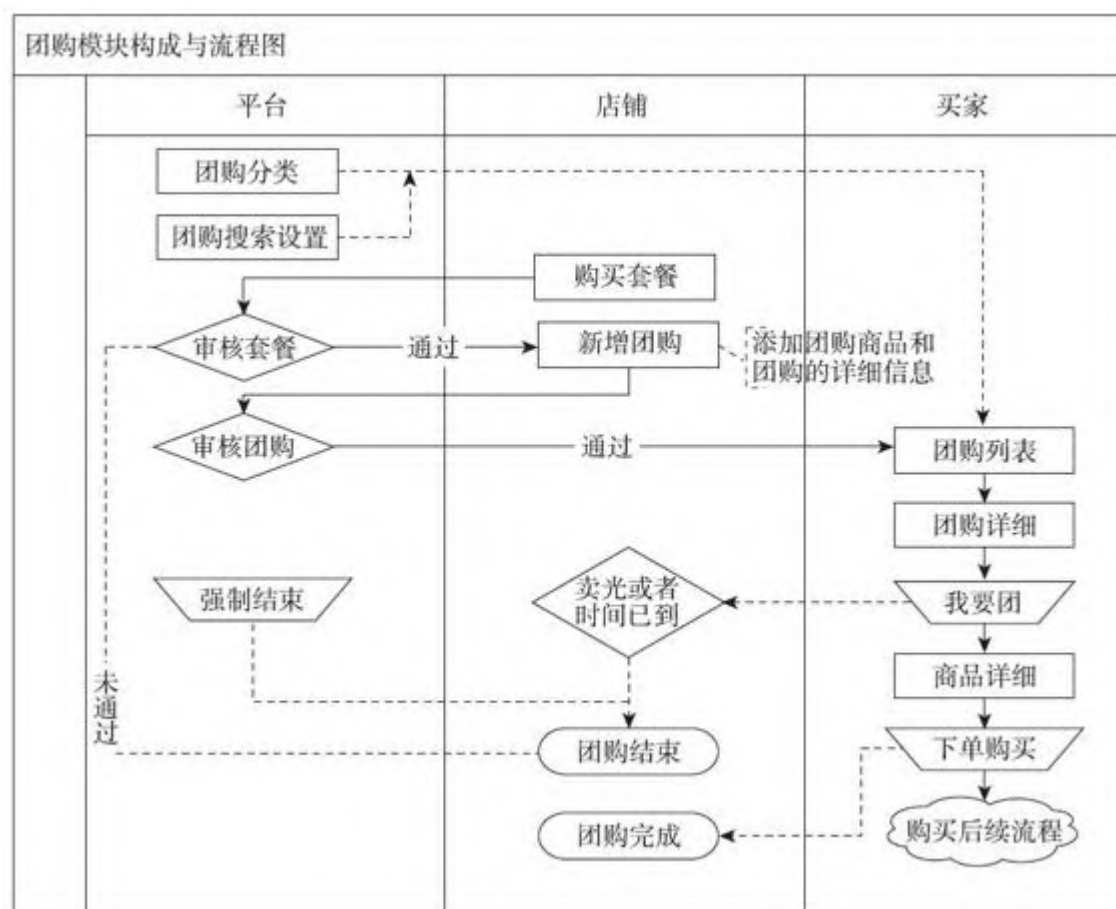


图3-25 B2B2C电商平台线上团购构成与跨职能流程图

3.3.3 代码实现

从以上模块可知，在电子商务网站中促销是多种多样的，除了促销特定的本身业务需要的部分代码不同外，其他的代码实现思路是一致的。下面还是以ShopNC B2B2C电商平台中的线上团购为例介绍一下促销模块的代码实现。

B2B2C电商平台中的线上团购功能中，团购申请和团购详情前台展示是比较重要的两块。其他例如搜索信息设置、分类管理、套餐管理的简单增删改功能就不再赘述了。

1. 团购申请实现

团购申请是店铺与平台之间申请团购信息上线展示的功能模块，申请过程的实现需要以下代码。

1) 购买套餐：按照需要的套餐数量购买套餐（一期套餐为30天），可以在购买的周期内发布团购活动。例如，购买2期团购活动（即为60天），则可以在该60天内发布团购活动。实现代码见代码清单3-13。

代码清单3-13 ShopNC B2B2C电商平台线上团购购买套餐

```
/**
 * 购买团购套餐
 **/
public function groupbuy_quota_add_saveOp() {
    //处理购买套餐数量
    $groupbuy_quota_quantity
    =intval($_POST['groupbuy_quota_quantity']);
    if($groupbuy_quota_quantity <= 0) {
        showDialog('购买数量不能为空');
    }
    $model_groupbuy_quota = Model('groupbuy_quota');
    //获取当前套餐价格
    $current_price = intval($GLOBALS['setting_config']
    ['groupbuy_price']);
    //获取该店铺当前可用套餐
    $current_groupbuy_quota= $model_groupbuy_quota-
    >getGroupbuyQuotaCurrent($_SESSION['store_id']);
    //计算可用套餐的时间段
    $add time = 86400 * 30 * $groupbuy_quota_quantity;
    //如果当前不存在可用套餐，则生成新套餐记录
    if(empty($current_groupbuy_quota)) {
        $param = array();
        $param['member_id'] = $_SESSION['member_id'];
        $param['member_name'] = $_SESSION['member_name'];
        $param['store_id'] = $_SESSION['store_id'];
```

```

        $param['store_name'] = $_SESSION['store_name'];
        $param['start_time'] = TIMESTAMP;
        $param['end_time'] = TIMESTAMP + $add_time;
        $model_groupbuy_quota->addGroupbuyQuota($param);
    } else {
        //如果当前已存在可用套餐则更新套餐结束时间
        $param = array();
        $param['end_time'] = array('exp', 'end_time + ' .
            $add_time);
        $model_groupbuy_quota->editGroupbuyQuota($param,
            array('quota_id' => $current_groupbuy_quota['quota_id']));
    }
    //记录店铺费用
    $this->recordStoreCost($current_price *
        $groupbuy_quota_quantity, '购买团购');
    showDialog(Language::get('groupbuy_quota_add_success'),
        urlShop('store_groupbuy', 'groupbuy_list'), 'succ');
}

```



注意 根据平台需要套餐购买成功后，需要支付相应费用，当支付成功并且平台审核通过后，即可计入下一步。

2) 申请团购活动：购买套餐成功后，店铺可以向平台提交团购活动申请，经过平台审核通过后，即可在团购活动列表中显示。实现代码见代码清单3-14。

代码清单3-14 ShopNC B2B2C电商平台线上团购活动申请提交

```

/**
 * 团购保存
 **/
public function groupbuy_saveOp() {
    //处理提交的商品编号
    $goods_id = intval($_POST['groupbuy_goods_id']);
    if(empty($goods_id)) {
        showDialog(Language::get('param_error'));
    }
    $model_groupbuy = Model('groupbuy');
    $model_goods = Model('goods');
    $model_groupbuy_quota = Model('groupbuy_quota');
    //检验当前是否有可用套餐
    $current_groupbuy_quota = $model_groupbuy_quota-
>getGroupbuyQuotaCurrent($_SESSION['store_id']);
    if(empty($current_groupbuy_quota)) {
        showDialog('当前没有可用套餐, 请先购买套
餐', urlShop('store_groupbuy', 'groupbuy_
quota_add'), 'error');
    }
    //获取参加团购的该店铺商品信息
    $goods_info = $model_goods->getGoodsInfo(array('goods_id'
=> $goods_id));
    if(empty($goods_info) || $goods_info['store_id'] !=
$_SESSION['store_id']) {
        showDialog(Language::get('param_error'));
    }
    //检查商品是否已经参加同时段团购活动, 不能重复参加团购活动
    $condition = array();
    $condition['end_time'] = array('gt',
strtotime($_POST['start_time']));
    $condition['goods_id'] = $goods_id;
    $groupbuy_list = $model_groupbuy-
>getGroupbuyAvailableList($condition);
    if(!empty($groupbuy_list)) {
        showDialog(Language::get('param_error'));
    }
    //整理团购信息数组
    $param = array();
    //.....
    //保存团购信息
    $result = $model_groupbuy->addGroupbuy($param);
    //.....
}

```




注意 提交团购活动申请后，需要平台审核该信息的真实性等，当平台审核通过后，活动便可在前台活动页面显示了。

2. 团购展示实现

团购申请成功之后，即可在前台显示了。团购详细页面作为团购信息全面展示的综合页面，应该将与团购相关的各方面信息全部展现，并且因为是促销活动的页面更应该吸引眼球。接下来我们看一下团购详细页面及团购信息在商品详细页面是怎么显示的。

团购详细页面代码实现见代码清单3-15。

代码清单3-15 ShopNC B2B2C电商平台线上团购活动详细页面代码

```
/**
 * 团购详细页面信息
 **/
```

```

public function groupbuy_detailOp() {
    $group_id = intval($_GET['group_id']);
    $model_groupbuy = Model('groupbuy');
    $model_store = Model('store');
    //获取团购详细信息
    $groupbuy_info = $model_groupbuy-
>getGroupbuyInfoByID($group_id);
    if(empty($groupbuy_info)){
        showMessage(Language::get('param_error'),'index.php?
act=show_groupbuy','','error');
    }
    Tpl::output('groupbuy_info',$groupbuy_info);
    // 输出店铺信息
    $store_info = $model_store-
>getStoreInfoByID($groupbuy_info['store_id']);
    Tpl::output('store_info', $store_info);
    // 浏览数次数加一
    $update_array = array();
    $update_array['views'] = array('exp', 'views+1');
    $model_groupbuy->editGroupbuy($update_array,
array('groupbuy_id'=>$group_id));
    // 商品好评率
    $model_evaluate = Model('evaluate_goods');
    $evaluate_info = $model_evaluate-
>getEvaluateGoodsInfoByCommonidID($groupbuy_info['goods_comm
onid']);
    Tpl::output('evaluate_info', $evaluate_info);
    Tpl::output('index_sign','groupbuy');
    Tpl::showpage('groupbuy_detail');
}

```

图3-26为团购活动详细页面代码实现效果图。



图3-26 团购活动详细页面代码实现效果图

团购活动在商品详细页面的实现见代码清单3-16。

代码清单3-16 ShopNC B2B2C电商平台线上团购活动在商品详细页展示代码

```
/**
 * 获取单条商品信息
 *
 * @param int $goods_id
 * @return array
 */
public function getGoodsDetail($goods_id) {
    if($goods_id <= 0) {
```

```

        return null;
    }
    //查询商品详细信息
    //.....
    //团购
    if (C('groupbuy_allow')) {
        $groupbuy_info = Model('groupbuy')-
>getGroupbuyInfoByGoodsCommonID($goods_info['goods_commonid'
]);
        if (!empty($groupbuy_info)) {
            $goods_info['promotion_type'] = 'groupbuy';
            $goods_info['remark'] =
$groupbuy_info['remark'];
            $goods_info['promotion_price'] =
$groupbuy_info['groupbuy_price'];
            $goods_info['down_price'] =
ncPriceFormat($goods_info['goods_price'] -
$groupbuy_info['groupbuy_price']);
            $goods_info['upper_limit'] =
$groupbuy_info['upper_limit'];
        }
    }
    //限时折扣:当商品团购活动存在时, 限时折扣让团购信息优先显示
    if (C('promotion_allow') && empty($groupbuy_info)) {
        //.....
    }
    // 商品受关注次数加1
    //.....
    //商品详细信息数组
    $result = array();
    $result['goods_info'] = $goods_info;
    $result['spec_list'] = $spec_list;
    $result['spec_list_mobile'] = $spec_list_mobile;
    $result['spec_image'] = $spec_image;
    $result['goods_image'] = $goods_image;
    $result['goods_image_mobile'] = $goods_image_mobile;
    $result['groupbuy_info'] = $groupbuy_info;
    $result['xianshi_info'] = $xianshi_info;
    $result['mansong_info'] = $mansong_info;
    return $result;
}

```



注意

以上代码中值得注意的一点是，商品可能同时参与了多种不同的促销活动，但是在客户下单时只能享受一种促销优惠，所以应该确定一种次序，有优先级的参与促销活动。例如，在以上代码中，当团购活动和限时折扣同时存在时，则优先参与团购活动，而不再重复参与限时折扣的促销优惠了。

图3-27为商品详细页面展示团购活动代码实现效果图。



图3-27 商品详细页面展示团购活动代码实现效果图

3.3.4 功能体验

介绍了这么多关于B2B2C电商平台中线上团购的相关内容，下面我们来体验一下它的使用。

1) 店铺向平台购买套餐，如图3-28所示。

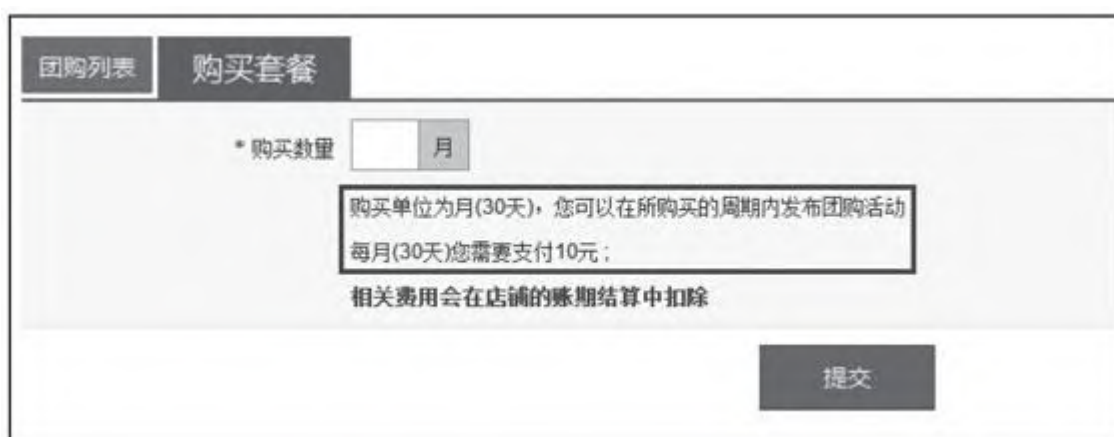


图3-28 店铺卖家中心购买套餐页面

2) 店铺提交团购活动申请，需要添加团购基本信息、活动商品信息、团购详细描述等。图3-29和图3-30展示了部分团购活动添加页面的内容。

* 团购名称:	
团购标题名称长度最多可输入30个字符	
* 开始时间:	
团购开始时间不能小于2014-04-21 16:05	
* 结束时间:	
团购开始时间不能大于2014-12-02 16:01	
* 团购价格:	元
团购价格为该商品参加活动加拍的促销价格 必须是0.01~1000000之间的数字(单位:元) 团购价格应包含邮费, 团购商品系统默认不收取邮费	
团购类别:	不限 
请选择团购商品的所属类别	
虚拟数量:	0
虚拟购买数量, 只用于前台显示, 不影响成交记录	
限购数量:	0
每个买家ID可团购的最大数量, 不限数量请填写"0"	

图3-29 店铺卖家中心团购申请添加页面基本信息展示



图3-30 店铺卖家中心团购申请添加页面选择参加活动的商品

3) 平台管理人员在管理员后台审核店铺提交的团购活动基本信息是否真实，如图3-31所示。

团购名称	开始时间	结束时间	邀请数	已购买	推荐	团购状态	操作
 六一团购活动 商品名称:春装 拼袖式 超短款 针织 衫开衫 女装 灰色 店铺名称:官方店铺	2014-05-25 00:00	2014-06-01 00:00	0	0	✓	审核中	通过 拒绝
 20140421团购活动 商品名称:新款 女款 拼袖 不规则摆 长袖针织衫开衫 杏色 店铺名称:官方店铺	2014-04-22 00:00	2014-04-24 00:00	0	0	✓	审核中	通过 拒绝
 SPALDING斯伯丁PU篮球馆内外投篮74-721 商品名称:SPALDING斯伯丁官方旗舰店PU球人气热卖篮球室内 外比赛篮球 74-721	2014-01-02 15:23	2014-12-25 00:00	111	1043	✓	正常	取消 删除

图3-31 平台审核店铺提交的团购活动信息

4) 团购活动列表展示可以分为3类显示，即是商品团购（进行中的团购）、即将开始（未开始的团购）、往期团购（已经结束的团购），如图3-32所示。



图3-32 团购活动列表

5) 团购活动详细页展示了团购的价格、数量、参与人数等基本信息，以及购买记录、商品评价、店铺信息等相关信息。点击“我要团”将进入下一步即团购商品购买页面，如图3-33所示。

6) 团购商品详细页面将展示团购活动信息，点击“立即购买”即可下单购买团购商品。具体购买流程与普通订单一样，不再赘述了。展示页面如图3-34所示。

东阿阿胶桃花姬阿胶糕300g，芝麻香，核桃脆，上品阿胶，养颜

闪电发货 48小时送达

¥95.00

我要团

DEEJ

桃花姬

DEEJ

桃花姬

原价

¥160.00

折扣

6.33折

节省

¥65.00

本商品已被团购 5 件

每人最多购买 3 件，数量有限，团购从速！

⌚

剩余时间：191天6小时16分24秒

商品信息

购买记录

商品评价(2)

桃花姬

DEEJ

产品参数

品牌：东阿阿胶 规格：300g

品名：桃花姬阿胶糕

配料表：黑芝麻、绍酒、核桃仁、阿胶、桃精明胶

批准文号：Q/OEB 0002S

保质期：常温下10个月

包装方式：盒装

适用人群：所有人

生产许可证

编号：QS3715 2401 0944

生产厂商名称：山东东阿阿胶股份有限公司

厂址：山东省东阿县阿胶街78号

食用方法：打开即食。每日推荐摄入量：1次2块，每日2次。

1 非物质文化遗产

流芳千载 上至皇宮貴族 下至平民百姓口口相传 其在与配方。

桃花姬

DEEJ

店铺信息

商家：官方店铺

综合评分：4.4 分

店铺动态评分 与行业相比

描述相符 4.7 分 持平

服务态度 4.4 分 持平

发货速度 4.1 分 持平

在线客服：

进入商家店铺

本期热门团购

228元任选一箱，国际大牌权威推荐“生”葡萄酒！

W228.00 13件已购买

去看看

东阿阿胶桃花姬阿胶糕300g，芝麻香，核桃脆，上品阿胶，养颜

¥95.00 5件已购买

去看看

春秋 披肩式 超细款 针织 衫开衫 女装 青鸟 绿色

¥120.00 100件已购买

去看看

图3-33 团购活动详细页面



中华老字号 东阿阿胶桃花姬阿胶糕300g

市 场 价: ¥168.00

商 城 价: **团购** ¥95.00 (原售价: ¥150.00)

促销信息: 最多限购3件 闪电发货 48小时送达

商品评价: ★★★★★ (2条评论)

至 全国: 卖家承担运费

累计售出: 3件

购买数量: (当前库存 40件)

立即购买

图3-34 商品详细页面团购信息

3.3.5 开发和使用

电子商务飞速发展的同时，相应的促销也在不断的发展变化，各种新的促销方式不断产生且变化多样，而促销在电子商务中发挥的作用越来越重要。网络经济有着不同于传统经济的规律与模式，电子商务也不可能是传统商务活动在网络上的翻版。电子商务自身所具有的特点决定了其大大优于传统商务，电子商务环境下促销手段运用的好坏就决定了网络销售取得的盈亏。怎么能够合理地开发和使用促销，使其更好地促进电子商务的发展，也是每个电子商务从业人员不断思考的问题。下面简单介绍一下自己的拙见。

电子商务平台促销开发应基于以下原则。

- 简单易懂：促销的受众是普通客户，商家通过促销向客户传递有关企业和商品的各種信息，说服或

者吸引客户使用或者购买其商品，其传达信息的方式越容易理解和操作也越吸引人，越容易着迷，谁也不愿意在享受购物的过程中去计算繁琐的数据、理解枯燥的字句，所以促销最好不要繁琐，让人一目了然才能触动人心。

- 吸引眼球：商家发出刺激消费的各种信息，把信息传递到目标对象，以影响其购买行为，达到促销的目的。那么发出的信息就要足够吸引人，使其容易被感动，所以促销开发中流程和界面设计要能够吸引住客户的目光，使其停留下来了解网站信息。

- 灵活组合：一个网站往往存在多种促销形式，而促销方式之间可以搭配使用达到更好的促销效果，例如限时折扣、满即送等可以同时存在于一个网站并同时使用。那么在设计促销流程时，各个形式的促销

应该不仅可以独立存在，并且在需要时也能组合起来。

- 数据统计：促销活动除了销售商品外还会产生很多的重要数据，例如：流量、转化率、客单价等。这些数据都值得我们去做分析。哪些方面做得好，哪些方面欠缺，为什么那么差，都可以通过对促销产生数据的有效分析之后得出，在今后的工作中加以改善。所以在开发促销模块时其对应的数据统计也是很必要的。

使用电子商务平台促销应注意以下几点。

- 利人利己：促销中商家认为已经让利很多，是绝无仅有的，客户从中得到了实惠，这是其利人的一面；同时他也从中获得了一小部分利润，提升了品牌价值 and 知名度，这是利己。让客户感到商家已经在“割肉”，自己捡了个便宜，赶紧下单购买吧。商家

的利人得到了利己的小小回报，客户也觉得遇到了天上掉馅饼的好事，客户和商家得到了双赢的局面，这样促销就成功了。本着这一点促销的路就不会走偏。

· 搭配使用：每种促销都各有利弊，在实际使用中需要灵活应对，很多情况下促销方式不会单一使用，运用促销组合优势互补往往会带来更好的效果。例如一件商品一折销售限时抢购，因为一折销售价格很低，商家基本没有利润所以不再包邮，再搭配一个满100即包邮活动对前一个活动进行补充，两个活动同时进行，这样即让客户感到获得了包邮的优惠，同时利用满100的限制提高了客单价。两个活动组合策划成功，起到了引人关注、吸引购买、增加利润等多个促销的目的。

· 不繁琐：促销要策划的易懂不烦琐，免得让客户心理因为不清楚而产生顾虑，反而弄巧成拙。商家

不要尝试过多方式，1~3种为佳。一是活动太多，计算太过复杂容易出错，降低客户对网站的评价甚至造成亏损；二是增加客服压力和强度，致使客户体验下降。

- 吸引力足够：促销有一个特点就是玩心跳，促销的商品和优惠力度要足够打动客户。要让客户感到错失机会，以后会后悔，给予其紧迫感，促使快速下单。

- 实事求是：商家为了吸引客户目光，避免不了使用一些客户关注的当下时髦的商品特点，但是不能只求轰动效应，而脱离实际情况。如果不符合实际，即伤害了客户的感情也损害了品牌价值，最后等于搬起石头砸自己的脚，反而弄巧成拙。

- 一诺千金：在促销中商家往往很注重价格优惠，但是服务的承诺却大打折扣。一般品牌都有一定

的服务理念，比如海尔承诺24小时服务到位，延迟保修期等，这些服务也是客户对品牌的基本认知。在参加促销活动后，不少商家就认为已经进行了优惠，服务只是一种附属品不是商品的组成部分，可以随意改变其服务项目和内容，从而实现节约开支的目的。但是从长远来看，这样的促销是没有意义的，为了一时的利益，损失了品牌的价值，得不偿失。所以促销应该是基于品牌原有的服务承诺而进行的，不可以随意变更。

电子商务的优越之处在于它可以突破地域和时间的限制，使处于不同地区的人们互通有无，开展贸易。它的快捷、迅速、自由和低成本为人们所乐道，因此电子商务越来越受欢迎，相应的促销也逐渐遍布到了电子商务网站的整个运营期间。但是促销无处不在，却不等于无所不能。一些企业恨不得把促销当做灵丹妙药，把价格当做唯一利器，只求形式，不求甚

解，模仿促销的表象，而不去思考和领会促销的理念和精髓，不求产品的创新和服务的提升。促销仅仅是一种刺激性的工具和手段，不能创造长期的品牌忠诚度，也不能挽救产品的致命缺陷，更不能替代产品的研发和创新。低层次的过度促销会使客户麻木，企业等于饮鸩止渴，慢性自杀。所以要科学使用促销，使其对品牌价值提升产生正向影响。

3.4 购物车模块

购物车是电商平台不可或缺模块之一，当用户喜欢某个商品而又不急于结账时，需要一个购物车来存储这个商品，这样方便用户统一结账，也方便用户作一下商品对比，选出最适合自己的一个。当然用户也可以将购物车中的商品随时移除或清空，当用户遇到网络中断或断电等异常情况时，购物车里还为用户保存着之前选中的商品。除此之外，购物车还为用户计算了商品总价、总数量及享受的促销优惠等信息，让用户一目了然。用户对购物车的操作，也恰恰反映了对哪些商品的关注程度，这就为商家运营决策提供了很好的参考依据。

3.4.1 模块构成

购物车主要包含添加、删除、编辑和收藏功能，我们可以将各种类型的商品加入购物车，可以随时移除这些商品，可以编辑购买数量，如果感兴趣还可以把它收藏起来。购物车作为购买流程中的一部分，与购买的其他环节也是密不可分的。图3-35展示了与购物车有关的主要操作及流程。

图3-35中实线部分为购物车核心内容，虚线部分为购物车关联内容。

3.4.2 设计思路

1. 设计要求

购物车的设计应该满足以下几个方面。

(1) 持久化保存

用户登录前和登录后系统均应支持加入到购物车：用户登录前，购物车使用cookie加密存储，由于cookie保存长度的限制，应适当限制加入购物车的上限，一般10个左右即可；用户登录后，购物车使用数据库存储，登录时系统应将登录前的购物车商品信息同步到数据库中，持久化保存，当用户下单成功后，从购物车永久清除已购买的商品。

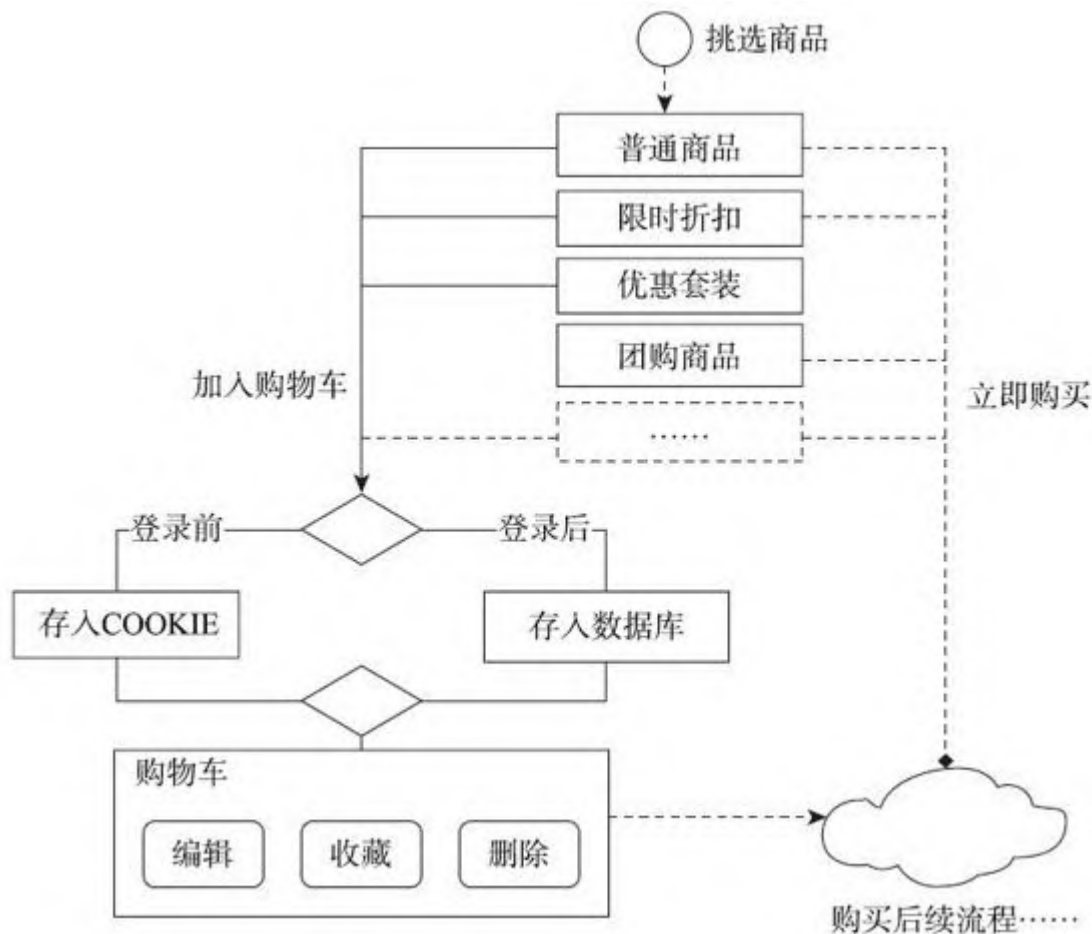


图3-35 购物车主要操作与流程

(2) 支持加入多种类型的商品

应尽可能支持多种商品加入购物车，除了普通商品外，商城很可能还有多种促销活动的商品，在不影响总体设计的情况下，购物车应尽可能支持这些促销商品的加入。

如果商城平台存在团购促销，建议采用立即购买形式处理，因为团购活动存在特殊性，团购商品无法以团购身份进入购物车，只能以普通商品身份进入购物车，但这样又会造成价格的不一致，所以建议设计购物车时，团购的商品只允许直接购买即可，目前ShopNC B2B2C商城就是这种处理方式。

（3）支持加入多个店铺商品

在B2B2C平台中，购物车应支持把多个店铺的商品加入购物车，查看购物车时以店铺分组显示。

（4）操作便捷

购物车应为用户提供便捷完善的操作，如增减商品数量、商品移除、加入收藏夹、筛选哪些商品参与最后的购买。

（5）数据完整性

购物车应显示当前商品的促销信息、当前店铺的促销信息、每个商品小计、每个店铺小计以及所有店铺小计。

（6）数据准确性

展示购物车内的商品信息时，对关键信息（如库存、价格、上下架状态、促销内容）应及时取得当前最新数据，对关键数据发生变更的商品在购物车应及时予以体现。

2. 数据表设计

购物车表设计应当主要考虑以下几点。

（1）表关系核心字段不可缺少

购物车会涉及其他相关表，如会员表、店铺表、商品表等，所以这些关联的外键是不可缺少的。

（2）必要的冗余字段

从数据表设计精简的角度出发，购物车只要包含了核心关联字段，我们就可以通过表的联查来取的我们想要的信息，但这样会大大增加数据库查询压力，所以必要的冗余字段是应该有的。建议商品价格、商品名称、商品图片、店铺名称都要包含。

（3）数据准确性

使用冗余字段，是为了减轻系统压力，但这不可避免会造成数据显示的滞后性，如商品的价格、名称、图片等可能无法显示最新数据，这就需要在关键节点处理时去查询数据库得到最新有效数据。

综合以上因素，购物车表关系图如图3-36所示。

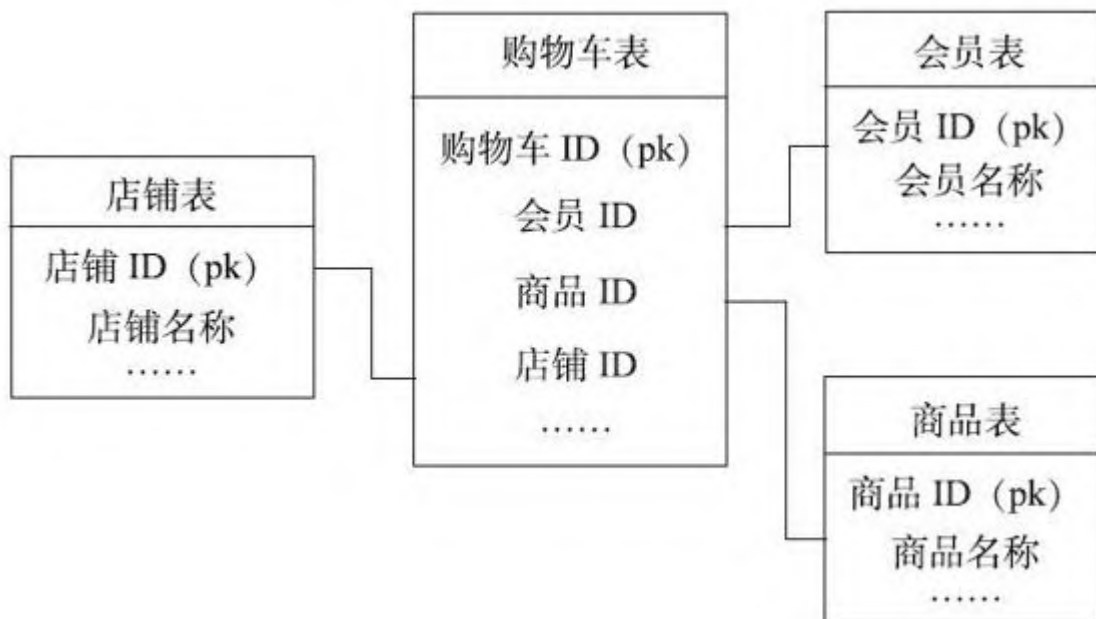


图3-36 购物车表关系图

购物车完整表结构如表3-6所示。

表3-6 购物车表结构

字 段	类 型	Null	默 认	说 明
cart_id	int(11)	否		购物车 ID
buyer_id	int(11)	否	0	买家 ID

(续)

字 段	类 型	Null	默 认	说 明
store_id	int(11)	否	0	店铺 ID
goods_id	int(11)	否	0	商品 ID
goods_name	varchar(100)	否		商品名称
store_name	varchar(50)	否		店铺名称
goods_price	decimal(10,2)	否	0.00	商品价格
goods_num	smallint(5)	否	0	购买数量
goods_image	varchar(100)	否		商品图
bl_id	mediumint(8)	否	0	优惠套装 ID

这里需要特别指出bl_id字段，此字段是优惠套装活动的ID。因为优惠套装是一组商品，目前对于以一组商品出现的促销形式，购物车表设计可以有两种方式：第一种方式是将该组内的所有子商品均作为一个独立商品加入购物车，然后为每个商品加一个促销活动的组标识；第二种方式是将该组商品作为一个虚拟的商品来处理，商品的名字、价格均使用整体促销的活动名字和促销价格，至于商品图片采用第一个子商品的图片，然后为每个商品加一个促销活动的组标识，每个子商品信息通过表查询来实现。第二种方式更符合购物车设计理念，所以这里采用了第二种方式。

3. 购物车模型设计

实现购物车功能，需要设计购物车自己模型，此模型需要完成以下功能。

1) 增删改查操作

该模型首先需要完成对购物车的增加、删除、编辑和删除操作，这是基础。

2) 入口封装

对购物车的添加、删除、编辑和查询操作都由该模型提供相应入口来完成，对cookie或数据库等存储的操作对外均表现为一个入口，以参数的形式来分流即可，方便开发与维护。

3) 数据统计

购物车模型应提供对商品数和总金额的统计入口，方便调用与使用。

4) 数据完整性与准确性

购物车模型应可以获取与购物车内商品相关的必要的信息，如商品状态、商品价格、商品促销信息（如限时折扣）、店铺促销信息（如满即送）等，并保证这些数据是最新、准确的。

3.4.3 代码实现

购物车代码实现主要分为购物车模型、控制器调用和模板输出三部分。模型设计和控制器调用最为关键，也是本节我们讲解的重点，模板输出只是涉及变量的抛出，没有复杂的逻辑处理，所以本节不再介绍。

1. 基本方法

购物车模型首先要实现对商品的基本处理，如增加、编辑和删除操作，实现这些操作的核心代码如代码清单3-17所示。

代码清单3-17 针对购物商品的基本操作

```
//取得单条购物车信息
public function getCartInfo($condition = array(), $field =
    '*') {
    return $this->field($field)->where($condition)->find();
}
```

```

//添加到购物车
public function addCart($data = array(), $save_type = '',
$quantity = null) {
    $method = 'addCart'.ucfirst($save_type);
    //这里根据参数类型(cookie/db)来决定调用哪个私有方法
    $insert = $this->$method($data,$quantity);
    $this->
>getCartNum($save_type,array('buyer_id'=>$data['buyer_id']))
;
    return $insert;
}
//编辑购物车
public function editCart($data,$condition) {
    $result = $this->where($condition)->
>update($data);
    if ($result) {
        $this->
>getCartNum('db',array('buyer_id'=>$condition['buyer_id']));
    }
    return $result;
}
//查询购物车列表
public function listCart($type, $condition = array(), $limit
= '') {
    if ($type == 'db') {
        //从数据中取数据
        $cart_list = $this->where($condition)->
>limit($limit)-> select(array('cache'=>false));
    } elseif ($type == 'cookie') {
        //从cookie中取数据，这里需要解密处理
        $cart_str = get_magic_quotes_gpc() ?
stripslashes(cookie('cart')) : cookie('cart');
        $cart_str = base64_decode(decrypt($cart_str));
        $cart_list = @unserialize($cart_str);
    }
    $cart_list = is_array($cart_list) ? $cart_list :
array();
    return !is_array($cart_list) ? array() : $cart_list;
}
//删除购物车商品
public function delCart($type, $condition = array()) {
    if ($type == 'db') {
        //直接从数据库中删除
        $result = $this->where($condition)->delete();
    } elseif ($type == 'cookie') {

```



```

        //解密cookie，然后删除相应的下标，然后再加密写到cookie中
        $cart_str = get_magic_quotes_gpc() ?
stripslashes(cookie('cart')) : cookie('cart');
        $cart_str = base64_decode(decrypt($cart_str));
        $cart_array = @unserialize($cart_str);
        if (key_exists($condition['goods_id'],
(array)$cart_array)) {
            unset($cart_array[$condition['goods_id']]);
        }
    }
    setNcCookie('cart',encrypt(base64_encode(serialize($cart_arr
ay))),24*3600);
    return $result;
}

```

2. 促销方法

购物车模型在实现对商品数据处理基础上，还需要对商品的促销信息进行调用和处理，如代码清单3-18所示。在实际运营中商品可能会参加各类型的促销活动，所以对促销信息及时展示和处理非常重要。

代码清单3-18 购物车对促销数据的处理

```

//取得限时折信息
public function getXianshiInfo($buy_goods_info, $quantity) {
    if (!C('promotion_allow') || empty($buy_goods_info) ||
!is_array($buy_goods_info)) return $buy_goods_info;
    //如果购买数量大于限时折扣下限，则按优惠价购买，否则按原价购买
    $xianshi_info = Model('p_xianshi_goods')->
getXianshiGoodsInfoByGoodsID($buy_goods_info['goods_id']);
}

```

```

        if (!empty($xianshi_info)) {
            if ($quantity >= $xianshi_info['lower_limit']) {
                $buy_goods_info['goods_price'] =
$xianshi_info['xianshi_price'];
                $buy_goods_info['promotions_id'] =
$xianshi_info['xianshi_id'];
                $buy_goods_info['ifxianshi'] = true;
            }
        }
        return $buy_goods_info;
    }
}
//取得商品最新的在售信息，如果商品下架或无库存，则在购物车内显示为无效
状态
public function getOnlineCartList($cart_list) {
    //先取出购物车内的商品ID
    $goods_id_array = array();
    foreach ($cart_list as $key => $cart_info) {
        if (!intval($cart_info['bl_id'])) {
            $goods_id_array[] = $cart_info['goods_id'];
        }
    }
    //根据以上ID，查询这些ID的最新在售信息
    $model_goods = Model('goods');
    $goods_online_list = $model_goods->
getGoodsOnlineList(array('goods_id'=>array(in,$goods_id_arr
ay)));
    $goods_online_array = array();
    foreach ($goods_online_list as $goods) {
        $goods_online_array[$goods['goods_id']] = $goods;
    }
    foreach ((array)$cart_list as $key => $cart_info) {
        if (intval($cart_info['bl_id'])) continue;
        $cart_list[$key]['state'] = true;
        $cart_list[$key]['storage_state'] = true;
        if
(in_array($cart_info['goods_id'],array_keys($goods_online_ar
ray))) {
//显示最新的商品属性，如图片、名称、价格，需要赋值的项很多，这里只显示
一部分
            $goods_online_info =
$goods_online_array[$cart_info['goods_id']];
            $cart_list[$key]['goods_commonid'] =
$goods_online_info['goods_commonid'];
            $cart_list[$key]['goods_name'] =
$goods_online_info['goods_name'];

```

```

//如果库存不足, 显示无效状态
        if ($cart_info['goods_num'] >
$goods_online_info['goods_storage']) {
            $cart_list[$key]['storage_state'] = false;
        }
    } else {
        //无效的商品作一下标识
        $cart_list[$key]['state'] = false;
        $cart_list[$key]['storage_state'] = false;
    }
}
return $cart_list;
}
//批量判断购物车内的商品是不是在限时折扣中, 如果购买数量>=规定的下限,
按折扣价格计算, 否则按原价计算
public function getXianshiCartList($cart_list) {
    $model_xianshi = Model('p_xianshi_goods');
    $model_goods = Model('goods');
    foreach ($cart_list as $key => $cart_info) {
        if (intval($cart_info['bl_id'])) continue;
        $xianshi_info = $model_xianshi->
getXianshiGoodsInfoByGoodsID($cart_info['goods_id']);
        if (!empty($xianshi_info)) {
            //如果是限时折扣, 按限时折扣属性显示
            if ($cart_info['goods_num'] >=
$xianshi_info['lower_limit']) {
                $cart_list[$key]['goods_price'] =
$xianshi_info['xianshi_price'];
                $cart_list[$key]['promotions_id'] =
$xianshi_info['xianshi_id'];
                $cart_list[$key]['ifxianshi'] = true;
            }
            $cart_list[$key]['xianshi_info']['lower_limit'] =
$xianshi_info['lower_limit'];
            $cart_list[$key]['xianshi_info']['xianshi_price'] =
$xianshi_info['xianshi_price'];
            $cart_list[$key]['xianshi_info']['down_price'] =
ncPriceFormat($cart_info['goods_price'] -
$xianshi_info['xianshi_price']);
        }
    }
    return $cart_list;
}
//取得购买车内优惠套装以及包含的商品数据, 一旦有商品下架或无库存, 则整个
套装显示无效状态

```

```

public function getBundlingCartList($cart_list) {
    $model_bl = Model('p_bundling');
    $model_goods = Model('goods');
    foreach ($cart_list as $key => $cart_info) {
        if (!intval($cart_info['bl_id'])) continue;
        $cart_list[$key]['state'] = true;
        $cart_list[$key]['storage_state'] = true;
        $bl_info = $model_bl->getBundlingInfo(array('bl_id'=>$cart_info['bl_id']));
        if (empty($bl_info) ||
        !intval($bl_info['bl_state'])) {
            $cart_list[$key]['state'] = false;
        }
        //取得优惠套装内的商品信息
        $cart_list[$key]['bl_goods_list'] = $model_bl->getBundlingGoodsList(array('bl_id'=>$cart_info['bl_id']));
        $goods_id_array = array();
        foreach ($cart_list[$key]['bl_goods_list'] as
        $goods_info) {
            $goods_id_array[] = $goods_info['goods_id'];
        }
        //取得优惠套装商品最新在售属性
        $goods_list = $model_goods->getGoodsOnlineList(array('goods_id'=>array(in,$goods_id_array)));
        $goods_online_list = array();
        foreach ($goods_list as $goods_info) {
            $goods_online_list[$goods_info['goods_id']] =
            $goods_info;
        }
        unset($goods_list);
        foreach ($cart_list[$key]['bl_goods_list'] as $k =>
        $goods_info) {
            if
            (array_key_exists($goods_info['goods_id'],$goods_online_list
            )) {
                $goods_online_info =
                $goods_online_list[$goods_info['goods_id']];
                if ($cart_info['goods_num'] >
                $goods_online_info['goods_storage']) {
                    //如查有一个商品库存不足，则整个套装都无效
                    $cart_list[$key]['storage_state'] =
                    false;
                }
            }
        }
        //按商品最新属性显示，这里只列出一部分赋值属性
    }
}

```

```

        $cart_list[$key]['bl_goods_list'][$k]
['goods_id'] = $goods_online_info['goods_id'];
        $cart_list[$key]['bl_goods_list'][$k]
['goods_commonid'] = $goods_online_info['goods_commonid'];
        $cart_list[$key]['bl_goods_list'][$k]
['store_id'] = $goods_online_info['store_id'];
        $cart_list[$key]['bl_goods_list'][$k]
['goods_name'] = $goods_online_info['goods_name'];
    } else {
        $cart_list[$key]['state'] = false;
        $cart_list[$key]['storage_state'] = false;
    }
}
}
return $cart_list;
}
//从购物车中取得商品列表形式的数据
public function getGoodsList($cart_list) {
    if (empty($cart_list) || !is_array($cart_list)) return
$cart_list;
    $goods_list = array();
    $i = 0;
    foreach ($cart_list as $key => $cart) {
        if (!$cart['state'] || !$cart['storage_state'])
continue;
        //购买数量
        $quantity = $cart['goods_num'];
        if (!intval($cart['bl_id'])) {
            //如果是普通商品
            $goods_list[$i]['goods_num'] = $quantity;
            $goods_list[$i]['goods_id'] = $cart['goods_id'];
            //.....
            $goods_list[$i]['bl_id'] = 0;
            $i++;
        } else {
            //如果是优惠套装商品
            foreach ($cart['bl_goods_list'] as $bl_goods) {
                $goods_list[$i]['goods_num'] = $quantity;
                $goods_list[$i]['goods_id'] =
$bl_goods['goods_id'];
                //.....
                //这里要标识优惠套装的编号
                $goods_list[$i]['bl_id'] = $cart['bl_id'];
                $i++;
            }
        }
    }
}

```

```

    }
}
return $goods_list;
}
//将下单商品列表转换为以店铺ID为下标的数组
public function getStoreCartList($cart_list) {
    $new_array = array();
    foreach ($cart_list as $cart) {
        $new_array[$cart['store_id']][] = $cart;
    }
    return $new_array;
}
//商品金额计算(分别对每个商品/优惠套装小计、每个店铺小计)
public function calcCartList($store_cart_list) {
    $store_goods_total = array();
    $order_goods_total = 0;
    foreach ($store_cart_list as $store_id => $store_cart) {
        $tmp_amount = 0;
        foreach ($store_cart as $key => $cart_info) {
            $store_cart[$key]['goods_total'] =
ncPriceFormat($cart_info['goods_price'] *
$cart_info['goods_num']);
            $store_cart[$key]['goods_image_url'] =
cthumb($store_cart[$key]['goods_image']);
            $tmp_amount += $store_cart[$key]['goods_total'];
        }
        $store_cart_list[$store_id] = $store_cart;
        $store_goods_total[$store_id] =
ncPriceFormat($tmp_amount);
    }
    return array($store_cart_list,$store_goods_total);
}
//取得店铺级活动-每个店铺可用的满即送活动规则列表
public function getMansongRuleList($store_id_array) {
    $model_mansong = Model('p_mansong');
    $mansong_rule_list = array();
    foreach ($store_id_array as $store_id) {
        $store_mansong_rule = $model_mansong-
>getMansongInfoByStoreID($store_id);
        if (!empty($store_mansong_rule['rules']) &&
is_array($store_mansong_rule['rules'])) {
            foreach ($store_mansong_rule['rules'] as
$rule_info) {
                if (!empty($rule_info['discount']) ||
(!empty($rule_info['mansong_goods_name']) &&

```

```

!empty($rule_info['goods_storage'])) {
    //返回数据以店铺ID为下标，方便使用
    $mansong_rule_list[$store_id][] = $this-
>_parseMansongRuleDesc($rule_info);
}
}
}
return $mansong_rule_list;
}
//取得店铺级满即送优惠，传入购买总金额，作为是否“满”的依据
public function
getMansongRuleCartListByTotal($store_goods_total) {
    $model_mansong = Model('p_mansong');
    $model_goods = Model('goods');
    $premiums_list = array();
    $mansong_rule_list = array();
    foreach ($store_goods_total as $store_id =>
$goods_total) {
        $rule_info = $model_mansong-
>getMansongRuleByStoreID($store_id,$goods_total);
        if (is_array($rule_info) && !empty($rule_info)) {
            if (empty($rule_info['discount']) &&
empty($rule_info['mansong_goods_name'])) {
                continue;
            }
            $rule_info['desc'] = $this-
>_parseMansongRuleDesc($rule_info);
            $rule_info['discount'] =
ncPriceFormat($rule_info['discount']);
            $mansong_rule_list[$store_id] = $rule_info;
            if (!empty($rule_info['mansong_goods_name']) &&
!empty($rule_info['goods_storage'])) {
                //赠品是一个购买数量为1，价格为0.00元的普通商品
                $data = array();
                $data['goods_id'] = $rule_info['goods_id'];
                $data['goods_name'] =
$rule_info['mansong_goods_name'];
                $data['goods_num'] = 1;
                $data['goods_price'] = 0.00;
                $data['goods_image'] =
$rule_info['goods_image'];
                $data['goods_image_url'] =
cthumb($rule_info['goods_image']);
                $data['goods_storage'] =

```

```

$rule_info['goods_storage'];
        $premiums_list[$store_id][] = $data;
    }
}
}
return array($premiums_list,$mansong_rule_list);
}
//拼装单条满即送规则页面描述信息
private function _parseMansongRuleDesc($rule_info) {
    if (empty($rule_info) || !is_array($rule_info)) return;
    $discount_desc = !empty($rule_info['discount']) ?
    '减'.$rule_info['discount'] : '';
    $goods_desc = (!empty($rule_info['mansong_goods_name'])
    && !empty($rule_info['goods_storage'])) ?
    " 送<a
href='\".urlShop('goods','index',array('goods_id'=>$rule_info
['goods_id']))\"' title='{ $rule_info['mansong_goods_name']}'
target='_blank'>[赠品]</a>" : '';
    return
sprintf('满%s%s%s',$rule_info['price'],$discount_desc,$goods
_desc);
}
//重新计算每个店铺最终订单总金额, 如果存在满即送、代金券、运费、在订单
过程中都可能需要重新计算订单总金额
public function reCalcGoodsTotal($store_goods_total,
$preferential_array, $preferential_type) {
    $deny = empty($store_goods_total) ||
    !is_array($store_goods_total) || empty($preferential_array)
    || !is_array($preferential_array);
    if ($deny) return $store_goods_total;
    switch ($preferential_type) {
        case 'mansong':
            foreach ($preferential_array as $store_id =>
$rule_info) {
                if (is_array($rule_info) &&
$rule_info['discount'] > 0) {
                    $store_goods_total[$store_id] -=
$rule_info['discount'];
                }
            }
            break;
        case 'voucher':
            foreach ($preferential_array as $store_id =>
$voucher_info) {
                $store_goods_total[$store_id] -=

```



```

$voucher_info['voucher_price'];
    }
    break;
    case 'freight':
        foreach ($preferential_array as $store_id =>
$freight_total) {
            $store_goods_total[$store_id] +=
$freight_total;
        }
        break;
    }
    return $store_goods_total;
}
//取得店铺级满x元免运费活动，返回信息以店铺ID为下标
public function getFreeFreightActiveList($store_id_array) {
    if (empty($store_id_array) ||
!is_array($store_id_array)) return array();
    $store_free_freight_active = array();
    $condition = array('store_id' =>
array('in',$store_id_array));
    $store_list = Model('store')->getStoreOnlineList($condition,null,'','store_id,store_free_
price');
    foreach ($store_list as $store_info) {
        $limit_price =
floatval($store_info['store_free_price']);
        if ($limit_price > 0) {

$store_free_freight_active[$store_info['store_id']] =
sprintf('满%s免运费',$limit_price);
        }
    }
    return $store_free_freight_active;
}
//验证代金券是否可用有效，若无效直接删除
public function reParseVoucherList($input_voucher_list =
array(), $store_goods_total = array(), $member_id) {
    if (empty($input_voucher_list) ||
!is_array($input_voucher_list)) return array();
    $store_voucher_list = $this->getStoreAvailableVoucherList($store_goods_total,
$member_id);
    foreach ($input_voucher_list as $store_id => $voucher) {
        $tmp = $store_voucher_list[$store_id];
        if (is_array($tmp) &&

```

```

isset($tmp[$voucher['voucher_t_id']])) {
    $input_voucher_list[$store_id]['voucher_id'] =
$tmp[$voucher ['voucher_t_id']]['voucher_id'];
    $input_voucher_list[$store_id]['voucher_code'] =
$tmp[$voucher ['voucher_t_id']]['voucher_code'];
    } else {
        unset($input_voucher_list[$store_id]);
    }
}
return $input_voucher_list;
}
//取得店铺可用的代金券
public function
getStoreAvailableVoucherList($store_goods_total, $member_id)
{
    $voucher_list = array();
    $model_voucher = Model('voucher');
    foreach ($store_goods_total as $store_id =>
$goods_total) {
        $condition = array();
        $condition['voucher_store_id'] = $store_id;
        $condition['voucher_owner_id'] = $member_id;
        $voucher_list[$store_id] = $model_voucher-
>getCurrentAvailableVoucher($condition,$goods_total);
    }
    return $voucher_list;
}

```

3. 方法调用

购物车模型的方法主要供前台购物车功能使用，主要分为对购物车的添加、编辑、删除和浏览操作。删除购物车商品调用很简单，直接使用购物车模型

delCart方法即可，所以本节只介绍购物车在完成添加、编辑和浏览功能时如何调用购物车模型方法。

添加购物车程序处理流程及所调用模型方法如图3-37所示。

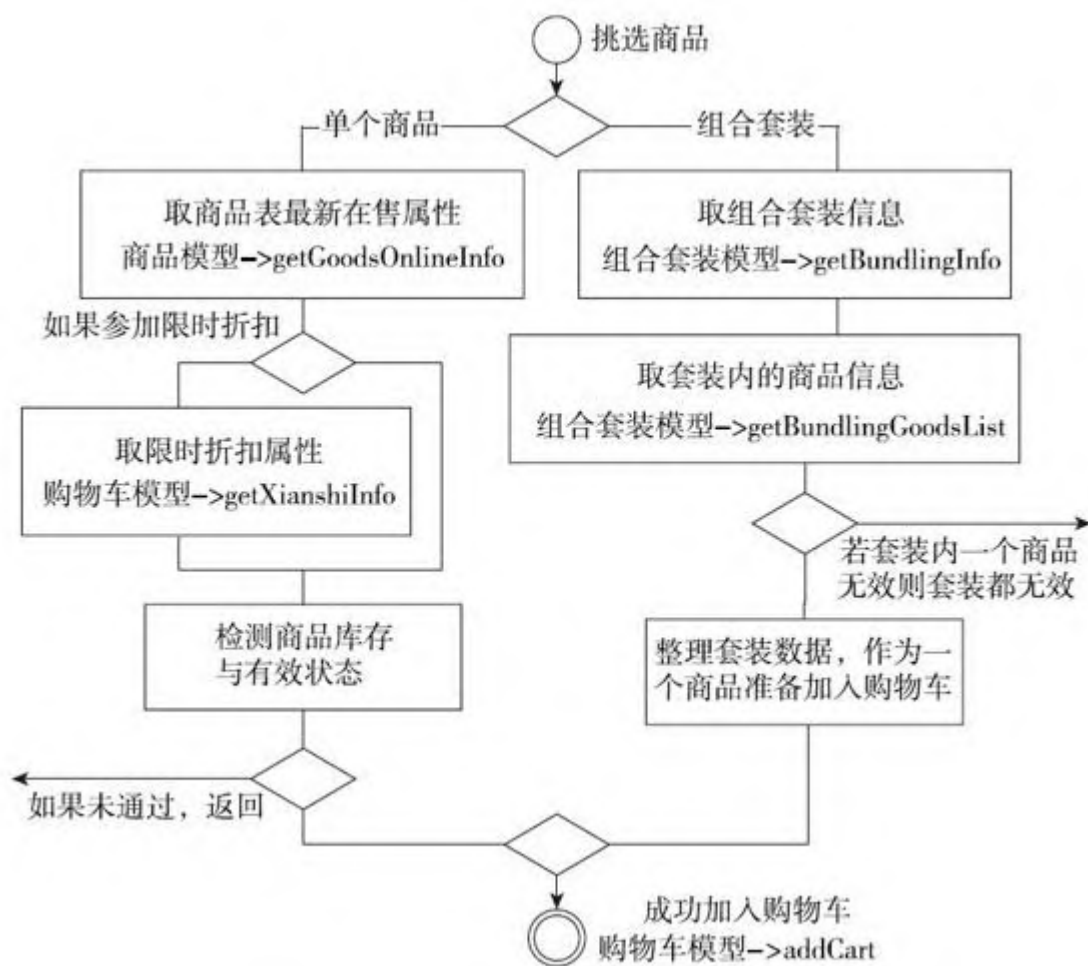


图3-37 添加购物车时模型方法的应用

编辑购物车程序处理流程及所调用模型方法如图3-38所示。

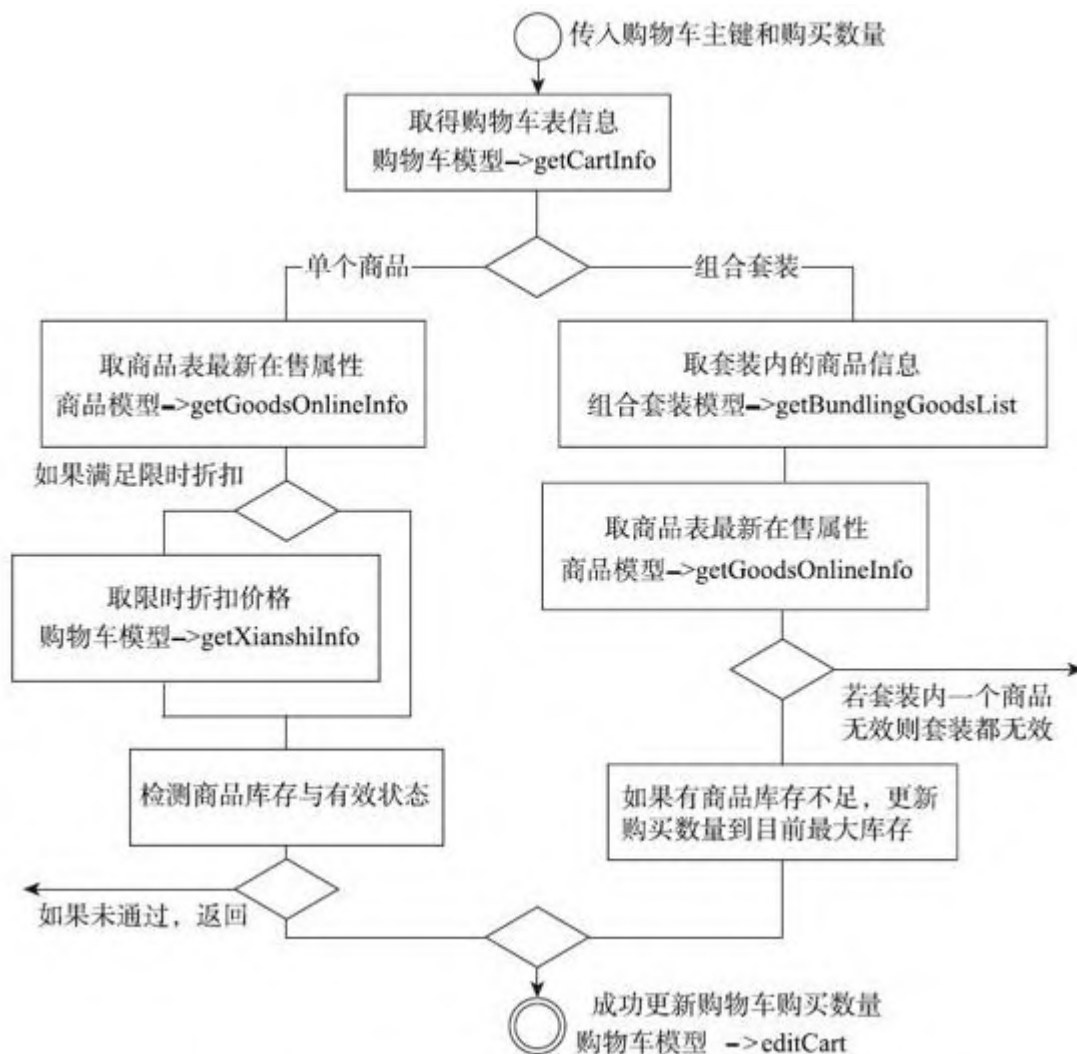


图3-38 编辑购物车时模型方法的应用

浏览购物车商品时程序处理流程及所调用模型方法如图3-39所示。



图3-39 浏览购物车时模型方法的应用

3.4.4 功能体验

本节将演示购物车的使用。首先将一个普通商品加入购物车，如图3-40所示。



图3-40 将商品加入购物车

挑选一个优惠套装加入购物车，如图3-41所示。



图3-41 将优惠套装加入购物车

从不同店铺随机加入一些商品到购物车，然后查看一下购物车快捷显示面板，如图3-42所示。

从图4-42可以发现，优惠套装作为一个商品来显示，这与我们前面的设计思路是一致的。点击“结算购物车中的商品”链接，进入“我的购物车”按钮页面，如图3-43所示。

从图3-43可以发现，购物车显示了商品基本信息、促销信息及统计信息等，这也印证了我们前面的设计思路。点击“下一步”按钮，如图3-44所示。

由于这一步购买数量已经确定，商品总金额就可以确定，本次购买所适用的满即送活动就可以确定，使用哪类代金券也可以确定，所以本次订单总金额可以确定，点击“提交订单”按钮，可以看到生成订单成功，如图3-45所示。

 我的商城 ▾
  购物车结算 ▴

最新加入的商品

	Truffles德菲丝松露巧克力 黑色传统型1000g 法国 进口原装	¥76.00×1 删除
	Truffles德菲丝松露巧克力 黑色传统型1000g 比利时 进口原装	¥76.00×2 删除
	KINGSMITH金史密斯动感单车 KS115 动感健身车 健身车 健身器材 红色 KS115	¥1999.00×2 删除
	跑步机加保护垫	¥2440.00×1 删除

共 4 种商品 总计金额: **¥6666.00**

结算购物车中的商品

图3-42 购物车快捷显示

我的购物车					
查看购物车商品清单，增加减少商品数量，并勾选需要购买商品进入下一步操作。					
☒ 全选	商品	单价(元)	数量	小计	操作
官方店铺					
☒	 Truffles傅罗丝松露巧克力 黑色传统型1000g 法国 进口原装	¥76.00	1	¥76.00	收藏 删除
☒	 Truffles傅罗丝松露巧克力 黑色传统型1000g 比利时 进口原装	¥76.00	2	¥152.00	收藏 删除
☐	 正品 2014春秋新款 女 韩版针织衫 开衫外套 毛领 黑色	¥189.00	无效		收藏 删除
店铺合计:				¥228.00	
健康健身器材					
免运费 满1000免运费					
☒	 KINGSMITH金史密斯动感单车 KS115 动感健身车 健身车 健身器材 红色 KS115 限时折扣 原价¥200.00	¥1999.00	2	¥3998.00	收藏 删除
☒	 跑步机加保护垫 优惠套餐	¥2440.00	1	¥2440.00	删除
	 金史密斯T221PLUS多功能跑步机	¥2400.00			收藏
	 专业跑步机垫 加厚减震垫 防滑抗震 2cm缓冲垫 隔曲垫 耐用	¥40.00			收藏
店铺合计:				¥6438.00	
商品总价(不含运费) ¥ 6666.00 元					
下一步，填写标购物信息					

图3-43 我的购物车

商品清单 返回购物车				
商品		单价(元)	数量	小计
官方店铺				满即送 满100减10
	Truffles傅罗丝松露巧克力 黑色传统型1000g 法国 进口原装	¥76.00	1	¥76.00
	Truffles傅罗丝松露巧克力 黑色传统型1000g 比利时 进口原装	¥76.00	2	¥152.00
买家留言:				
				运费: ¥0.00
				商品金额: ¥228.00
				满即送-满100减10: ¥-10.00
				满额10元 有效期至 2014-05-28 ☐ : ¥-10.00
				本店合计: ¥208.00
健康健身器材				免运费 满1000免运费
	KINGSMITH全史奈斯动感单车 KS115 动感健身车 健身车 健身器材 红色 KS115	¥1999.00	2	¥3998.00
限时折扣				

图3-44 设置订单信息

支付提交

订单详情内容可通过查看我的订单进行核对处理。

部分商品需要在线支付，请尽快付款！在线支付金额：¥6438.00

订单号	支付方式	金额	物流
由于您的商品由不同商家发出，此单将分为2个不同子订单直送！			
6000000000040902	在线支付	¥6438.00	快递
6000000000040901	货到付款	¥208.00	快递

支付选择

支付宝

财付通

网银在线
ChinaBank Payment

确认提交支付

图3-45 下单成功，待支付页面

至此一个相对完整的下单流程已经演示完毕。由于本节是以介绍购物车为主，下单过程中的物流配送、支付等环节我们会在其他章节做详细介绍。

3.5 配送模块

物流配送对于一个电商平台的重要性不言而喻，特别在当今主流电商平台中，送货速度已经成为服务和竞争的重要指标，这就决定了我们在电商平台规划筹备时期，物流配送模块一定要从技术和功能上做周密设计，通过平台的配送体系能为运营部门提供很好的技术支撑。

3.5.1 模块构成

电商平台的配送体系应该由多个角色下的多个功能点组成，买家、商家、平台都需要参与，平台需要初始化一些基本信息，如全国或地方性的地区行政区域、主要的快递公司等。商家需要设置快递公司（从大平台设置快递公司中选择），供发货时输入使用，这样买家可以随时跟踪物流，当然对于同城销售、自提货等情况，买家也可以不使用平台内置的物流公司。运费模板不但支持不同地区不同运费，还避免了商家对商品运费的重复设置，减轻了工作量。订单达到一定金额时包邮已经成为非常普遍的一种促销手段。买家下单时，必须要设置收货信息，系统据此来计算订单运费。配送体系涉及地方较多，为了便于更好理解，我们以购买流程为主线，为大家展示配送体系中各角色及功能点之间的关系，如图3-46所示。

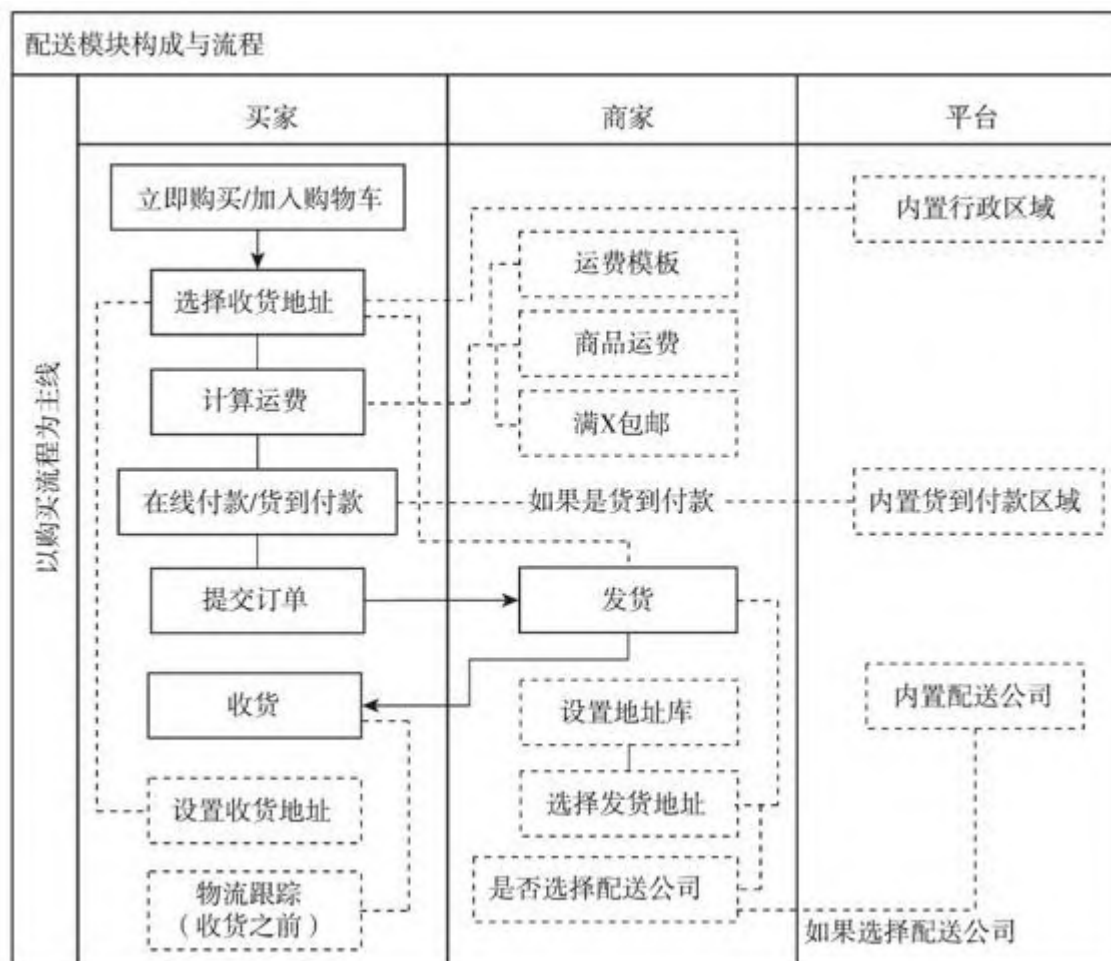


图3-46 配送模块构成与流程

图3-46中实线部分为购买流程与配送相关的主要节点，虚线部分为配送体系涉及的主要功能模块，以及这些模块与主要购买节点的关系。

3.5.2 设计思路

1. 设计要求

B2B2C电商平台的配送模块设计应至少满足以下需求。

(1) 内置行政区域

需要内置有足够的行政区域，系统中所有依赖地区选择的地方都以该行政区域为基础来完成，它的管理与维护由大平台来完成。

(2) 内置配送公司

需要内置有足够的配送公司，商家发货时可以从中选择，而不必自己手动输入，这样也方便系统提供物流跟踪功能。

（3）设置货到付款地区

这一功能由平台来完成，主要针对自营店设置的。

（4）运费模板

有配送就离不开配送费用，商家可以设置运费模板，从而减轻商家的工作量。

（5）收货地址

买家可以设置多个收货地址，下单时从中选择即可。

（6）物流跟踪

在商家发货后，买家应当时时查看自己的物流情况，对于B2B2C平台而言，这是必须要有的。

2. 数据表设计

配送体系由多个角色下的多个功能点组成，从大平台配送元素的初始化，到商家的配送公司、运费模板设计，再到买家收货地址等，都是配送体系的组成部分，这就决定了它由多个表组合而成，而且这些表多以平行关系为主，涉及相关表关系如图3-47所示。

图3-47中虚线内的订单系列表、促销系统表、商品系列并非三个表，而是代表某一类型的表，这里为了使表之间的关系容易理解把次要内容进行了合并。订单表一般会拆分成三个表，分别是订单主表、辅表和订单商品表。每种促销活动都至少会设计一个表，商品系列表一般包含商品主表、辅表、品牌、规格等。

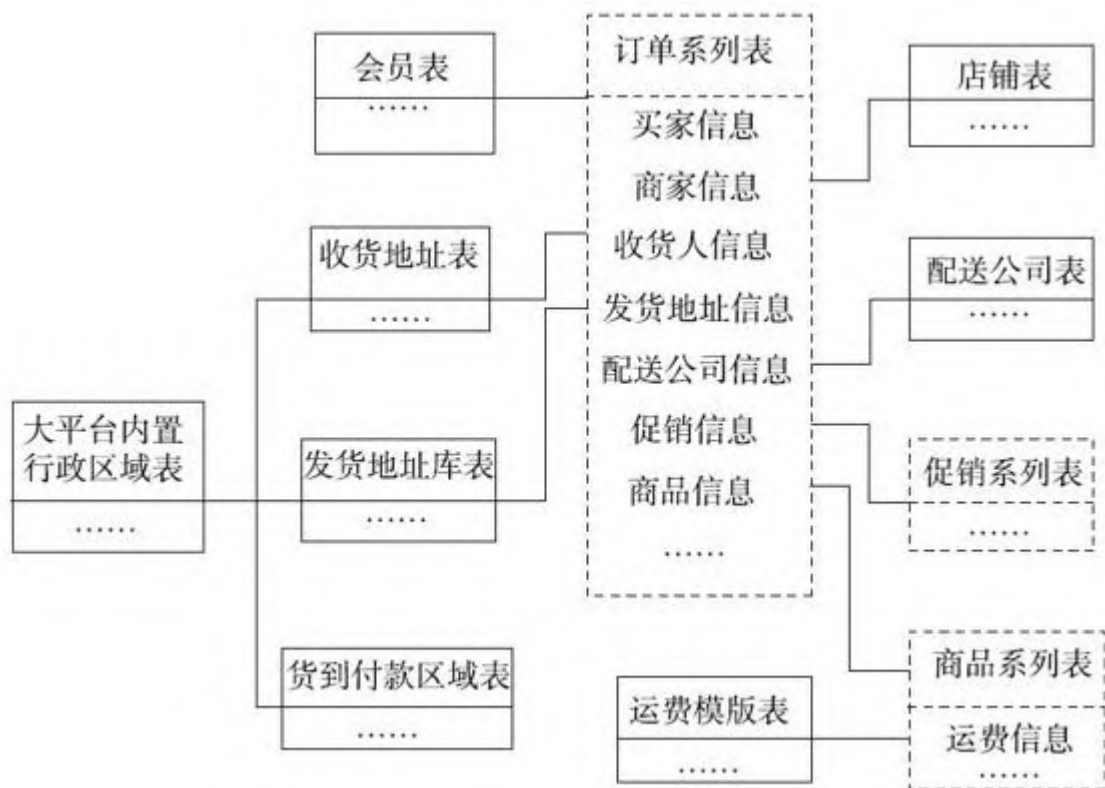


图3-47 配送体系表关系图

3.5.3 功能实现

配送模板由多个子模块组成，每个子模块的实现相对简单，没有复杂的流程参与，所以这里略过部分代码具体实现部分，以平台→商家→买家为主线，依次体验各主要子模块以及它们之间的关联关系。

1. 配送区域

配送区域包括两部分：一个是标准的行政区域设置，一般会设计到三级或四级地区，这些行政区域几乎很少变更，对它们的管理我们不再介绍；另一个是货到付款区域的设置，这是我们要介绍的，支持货到付款的商家一般多为平台自营店（也有非平台自营也支持货到付款的可能，但涉及具体的运营及配送签约等细节问题，不是我们讨论的重点），所以对这一项的管理应当归为大平台来设置。

目前电商平台对货到付款区域一般支持到三级（县级），管理这些区域时，由于区域是固定的，而且区域较多，经常会选中一级或二级地区（下面的子区域也会全部生效）即可，不可采用类似新闻逐条的形式。核心实现代码如代码清单3-19所示。

代码清单3-19 配送区域

```
//数据表保存的格式为县ID序列化后的内容
//取出支持货到付款县的ID，需要反序列化一下
$parea_info = $model_parea-
>getAreaInfo(array('store_id'=>$store_id));
if (!empty($parea_info['area_id'])) {
$parea_ids = @unserialize($parea_info['area_id']);
}
//取出支持货到付款县ID的上级市ID
$city_checked_child_array = array();
$county_array = $model_area-
>getAreaList(array('area_deep'=>3,'area_id'=>array('in',$parea_ids)), 'area_id,area_parent_id');
foreach ($county_array as $v) {
if (in_array($v['area_id'],$parea_ids)) {
$city_checked_child_array[$v['area_parent_id']][] =
$v['area_id'];
}
}
//市级下面的县是不是全部支持货到付款，如果全部支持，默认选中，如果其中
部分县支持货到付款，默认不选中但显示一个支持货到付款县的数量
$list = $model_area-
>getAreaList(array('area_deep'=>3),'area_parent_id,count(area
a_id) as child_count','area_parent_id');
foreach ($list as $k => $v) {
$city_count_array[$v['area_parent_id']] = $v['child_count'];
```

```

}
foreach ($city_checked_child_array as $city_id =>
$city_child) {
if (count($city_child) > 0) {
if (count($city_child) == $city_count_array[$city_id]) {
    $city_checked_array[$city_id] = true;
}
}
}
//取得省级地区及直属子地区(抛出到模板后, 会循环输出该数组)
require(BASE_DATA_PATH.'/area/area.php');
foreach ($area_array as $k => $v) {
if ($v['area_parent_id'] != '0') {
    $area_array[$v['area_parent_id']]['child'][$k] =
$v['area_name'];
    unset($area_array[$k]);
}
}
//计算哪些省需要默认选中(即该省下面的所有县都支持到付, 即所有市都是选中
状态)
$province_array = $area_array;
foreach ($province_array as $pid => $value) {
if (is_array($value['child'])) {
    foreach ($value['child'] as $k => $v) {
        if (!array_key_exists($k, $city_checked_array)) {
            unset($province_array[$pid]);
            break;
        }
    }
}
}
}
}

```

配送地区页面的加载时的全部地区数据都由服务器端来完成, 在加载页面时, 将已支持货到付款县ID放入Javascript数组中, 在编辑地区时, 上级地区是否选中以及数量的变化由客户端Javascript来完成。

图3-48展示了B2B2C平台对自营店铺货到付款地区的设置。

· 此处只能设置平台自营店铺支持货到付款的地区，注意：只有开启货到付款时以下设置的地区才会生效

· 选择完了地区确认后，系统并未保存，需要点击页面底部的保存按钮系统才会保存设置的地区

省	市
☒ 北京	☒ 北京 (17)
☒ 天津	☒ 天津 (16)
☒ 河北	☒ 石家庄 (23) ☒ 保定 (25) ☒ 沧州 (16) ☒ 唐山 (14) ☒ 张家口 (17) ☒
☒ 山西	☒ 太原 (10) ☒ 朔州 (6) ☒ 吕梁 (13) ☒ 大同 (11) ☒ 晋中 (11) ☒

图3-48 自营店货到付款地区设置

点击“编辑”按钮符号，可以编辑某区域下的子地区，编辑保存的子地区，如图3-49所示。



图3-49 编辑货到付款子地区

选择完地区后，确认并保存即可完成设置。

2. 配送公司

平台内置配送公司是为商家发货时服务的，配置公司内容至少要包含公司名称、网址、公司代码（第三方快递查询平台识别的公司标识），为方便平台日

常管理，还可以增加首字母（方便查询）、状态、是否常用等字段。

配送公司管理如图3-50所示。

首字母ABCDEFGHIJKLMNOPQRSTUVWXYZ				
操作提示				
系统内配送公司不能删除，只可调整状态，平台可禁用不需要的快递公司，默认按首字母进行排序，常用的快递公司将会排在靠前的位置				
快递公司	首字母	网址 (仅供参考)	常用	状态
顺丰快递	S	http://www.sf-express.com	✓	✓
圆通快递	Y	http://www.yto.net.cn	✓	✓
韵达快递	Y	http://www.yundaox.com	✓	✓
中通快递	Z	http://www.zto.cn	✓	✓
中国邮政	B	http://www.chinapost.com.cn	✓	✓
CCES	C	http://www.ccs.com.cn	✓	✓
德邦物流	C	http://www.cbwl.com	✓	✓
DHL快递	D	http://www.dhl.com	✓	✓

图3-50 配送公司管理

3. 收货地址

买家可以保存N（N建议为20个即可）个有效地址，可以对这些地区随时编辑这些地址，也可以设置一个默认地址。图3-51为收货地址列表。

地址列表				新增地址
操作提示： 最多可保存20个有效地址				
默认	收货人	所在地区	电话/手机	操作
	王某某	内蒙古自治区 呼和浩特市 和林格尔县 红山区红旗路XX号	☎ 15000000000	☐ 编辑 ☐ 删除
☒	张某某	河北省 保定市 南市区 红旗路XX号	☎ 11000000000	☒ 编辑 ☐ 删除
	孙某某	浙江省 宁波市 北仑区 红旗路XX号	☎ 12000000000	☐ 编辑 ☐ 删除
	李某某	四川省 成都市 双流县 红山区红旗路XX号	☎ 13000000000	☐ 编辑 ☐ 删除

图3-51 收货地址列表

编辑收货地址，如图3-52所示。

编辑地址

×

* 收货人：

张某某

请填写您的真实姓名

* 所在地区：

河北省 保定市 南市区

编辑

* 街道地址：

红旗路XX号

不必重复填写地区

* 电话号码：

区号 - 电话号码 - 分机

* 手机号码：

11000000000

设为默认地址：

☒

设置为默认收货地址

编辑地址

取消

图3-52 编辑收货地址

然后开始下单，此时发现默认地址已经生效，如图3-53所示。



填写核对购物信息
请仔细核对填写收货、发票等信息，以确保物流快递及时准确投递。

收货人信息 [修改]

张某某 河北省 保定市 南市区红旗路XX号 011000000000

图3-53 下单时显示默认地址

点击“修改”按钮，可以看到刚才保存的地址均出现在列表中，并且可以随时增加新的地址，如图3-54所示。

我们使用默认收货地址+货到付款（购买的是自营店铺商品，前面设置货到付款地区时包含保定市），会出现确认提示信息，如图3-55所示。

继续提交订单并付款，完成下单步骤。



收货人信息

☐ 王某某 内蒙古自治区呼和浩特市和林格尔县红山区红旗路XX号 15000000000 [删除]

☐ 张某某 河北省保定市莲池区红旗路XX号 11000000000 [删除]

☐ 孙某某 浙江省宁波市北仑区红旗路XX号 12000000000 [删除]

☐ 李某某 四川省成都市双流县红旗路XX号 13000000000 [删除]

☒ 使用新地址

* 收货人姓名:

* 所在地区:

* 详细地址:
请填写真实地址，不需重复填写所在地区

* 手机号码: 或 固定电话:

保存收货人信息

图3-54 收货地址列表



收货人信息

张某某 河北省保定市莲池区红旗路XX号 11000000000

支付方式

☐ 在线支付

☒ 货到付款

保存支付方式

请确认支付方式

以下商品支持货到付款

☒ 货到付款

确认支付方式

图3-55 货到付款确认提示

4. 商家发货

商家可以管理自己的发货地址库，如图3-56所示。

地址库		默认物流公司		新增地址	
发货地址	联系人	所在地区	电话	操作	
☐ 默认	王某某	吉林省 四平市 双辽市 红旗路XX号1号仓库	15000000000	☐ 编辑	☐ 删除

图3-56 商家地址库管理

商家可以添加自己需要的物流公司，如图3-57所示。

地址库		默认物流公司		物流公司	
☒ 顺丰快递	☒ 申通快递	☒ 圆通快递			
☒ 中通快递	☐ 包裹平邮	☐ 传喜物流			
☐ 大田物流	☐ DHL快递	☐ 德邦物流			

图3-57 商家设置物流公司

第三步 选择物流服务			
您可以通过“发货设置”->“默认物流公司”添加或修改常用发货物流。免运费自提商品可勾选下方无需物流运输服务并勾选卡并操作。			
自行联系物流公司		无需物流运输服务	
公司名称	物流单号	备注	操作
顺丰快递			确认
申通快递			确认
圆通快递	9253102752		确认
中通快递			确认

图3-58 商家发货设置配送信息

确认后，订单进入发货状态，这时如果快递发信息进入物流跟踪系统，买家可以跟踪物流了。

3.5.4 物流跟踪

物流跟踪是系统不可或缺的功能。建议平台运营者接入稳定、及时的第三方快递跟踪系统，使用Ajax异步调用数据，交互格式采用json格式。物流跟踪示例如图3-59所示。

物流动态
2014-04-01 16:41:25 卖家已发货
2014-03-29 19:48:35 广东省惠州市公司 已收入 操作员：谢关平
2014-03-29 21:38:58 广东省惠州市公司 已打包 操作员：梁碧带
2014-03-29 23:28:05 广东省惠州市公司 已发出 操作员：谢关平
2014-03-30 01:53:35 虎门转运中心公司 已收入 操作员：覃子文
2014-03-30 05:13:51 虎门转运中心公司 已发出 操作员：覃子文

图3-59 物流跟踪

3.6 订单模块

订单模块是交易流程的核心部分，购物车、物流、支付、退单、结算、统计等模块都与订单密不可分，订单的前期规划与设计显得至关重要。

3.6.1 设计思路

订单是一个大概念，我们可以将它拆分为几个主要的点，如订单状态、金额、编号、库存等，每个小细节的设计与规划都与整个订单实现密切相关，本节以订单实现中的几个主要环节作一下分析。

1. 订单状态

订单状态是订单流程的重要标志，订单处于哪个阶段，允许哪个角色来处理，主要判断依据就是订单状态。考虑到执行的效率一般都使用数字标识订单状态，具体有哪些状态通常因运营业务模式的不同而略有差别，一般情况下，订单至少包含默认、取消、支付、发货、收货这5种状态，除了以上状态，订单还可能有删除、审核、备货、出货、锁定、退货、退款、仲裁等状态。

订单下单成功，但还未支付，这时处于默认状态，支付完成以后变为支付状态，支付成功后，商家就准备发货了（这里可能还需要备货与出货），如果是货到付款，可能还需要审核一下，如果中间发生了纠纷，可能就会退款、退货甚至需要平台来仲裁了。图3-60展示了订单的主要处理状态。

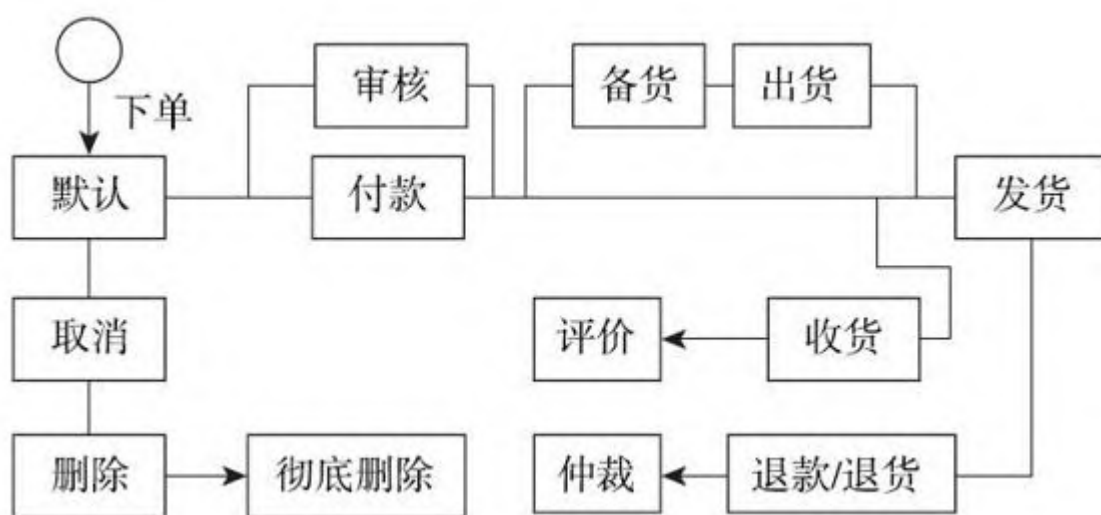


图3-60 订单主要处理状态举例

注意，图3-60仅作为一例参考，并不作为所有电商平台的标准。

2. 订单金额

这里的订单金额指的是订单中涉及金钱元素的统称，至少包括商品单价、商品总价、订单总金额、优惠金额、运费、代金券面额、退款金额等。

订单总金额是必须要有的，它是扣除各种优惠之后的实际支付金额（包含运费）。商品总金额是扣除商品优惠后的实际应支付金额，是对每种商品的小计。运费是在订单中所包含商品的运费总和基础上，应用店铺级运费优惠（如满100元包邮）规则后的计算结果。优惠金额是指店铺级促销活动所优惠的金额，如满500减50等。代金券是使用代金券的面额，通常一次下单只能使用一张代金券。退款金额是发生售后维权之类的操作后，所返还的金额。

如果支持使用预存款支付，还会涉及预存款的支付金额和冻结金额。

如果平台与商家采用佣金结算模式，还应当记录扣除所有优惠后分摊到每种商品的实际支付金额，因为平台与商家结算时会用到。

3. 订单编号

订单编号生成方式有很多种，有的是以时间+随机数来生成，有的是以自增ID来生成。结合B2B2C模式的特点，建议可以充分考虑时间、随机数、商家ID、会员ID、自增ID这些相关元素，结合自身的运营特点，制定生成方案。

利用时间作为订单号的组成元素是很多网站的做法，有的使用年月日时分秒开头，后面加微秒及随机数如“20140511121441****”，或者将年限制两位数，增加几位随机数，如“140511121441*****”，或者使用UNIX时间戳开头加随机数，如“1393812779*****”，以上做法时间都占据了较大的

长度，可以再缩短一些，比如可以使用当前时间减去某个时间点（如网站上线时间）所得的时间差值

（秒）作为订单一个参数。为了保持订单编号长度固定，可以对各组成元素补0或取余处理。订单编号设计的目的就是保证在高并发下，订单编号的重复几率降到最低。

4. 库存

库存是一个大概念，从字面理解意为仓库中的存货，对于生产型企业而言，库存可以分原料和成品两大类。对于零售型企业而言，它没有生产过程，从上游采购，然后存放在库房，最后销售给下游，库存就是库房中货品总和。库存不能一概而论，即便一个最小单位SKU也会有不同状态，例如，在运输过程中，可能会有磕碰，那么这个商品就不可以销售了，另外，还有可能会发生退货、换货、弃货等，在系统管理

时，就应当加以区分了。在实际操作中，下单、发货等在时间、空间上是异步进行的，因此有必要将库存结构区分开来。电商企业的库存结构一般分为以下几个部分。

（1）可销售库存

可销售库存也就是网站前台显示的库存，也是库存最主要的部分，大部分电商平台前台的库存与后端ERP类系统保持一致，当库存足够时，可以购买，当库存不足，可以提醒商家补货、接收补货通知。当顾客下单时，系统首先会请求后端，如果购买量与库存量匹配成功，则可以下单，否则会提醒顾客库存不足。

（2）订单占用库存

当顾客下单成功后，订单占用库存增加，可销售库存量减少，这样保证了用户下单后，有货可发。

(3) 不可销售库存

我们前面提到，货品可能因为破损而无法销售，因此在系统设计时也要做充分考虑，在实际操作中无法销售的原因很多，例如破损、功能故障、发错型号、弃货等。

(4) 锁定库存

在运营过程中，发起各种降价促销是常用的手段，发起降价后，可能会在很短的时间内将商品一售而空。但在有些情况下，商家并不希望过早就将全部商品卖完，可能因为全部降价的成本很高，还可能是防止竞争对手恶意采购，在更多情况下，只是将降价活动作为一个噱头，以此带动整个网站的流量和销量，为了达到这个目的，就需要将库存锁定，锁定后可以对外展示库存为零，从而无法直接销售，必须在

解除锁定后才可以继续销售，这也就人为实现了促销的分批次进行。

（5）虚拟库存

以上提到的都是实物，都是在库房切实存在的，但库房容量是有限的，不可能无限扩展。电子商务的优势之一就是几乎无限的商品展示与销售能力，这就需要虚拟库存了。有些商品虽然库房没有或库存不足，但供应渠道很通畅，新货品可以在短时间内到位销售。还有一些商品销售少但库存管理成本较高，只有产生订单后，才会向供应商采购。以上提到的这些都可以用虚拟库存来解决。虚拟库存可以使用网站前台展示的库存数大于实际库房中的库存量，避免了顾客下单时遇到库存不足的问题。

以上几点可以作为电商运营者前期库存设计的考虑因素，实际运作中库存处理可能会涉及WEB电商平

台、ERP、CRM等多个系统协作，而且在不同公司不同运营模式中，有着各自的特殊性，有不同的库存设计模式，以适合每个电商平台的实际情况，所以这里我们不作深入的探讨。

5. 合并支付

B2B2C平台区别于B2C平台最大的不同就是多商家同台销售，这样顾客在下单时可以把不同的商家订单进行合并统一支付，这里的合并也只是支付金额的合并，每个子订单还是相对独立的，每个商家都会有自己的一个订单，顾客还是可以对每个子订单单独操作。统一支付后，每个子订单会提交到不同的商家，后继处理及流程也会有所不同。

6. 角色与权限

下单后，买家可以对订单取消、删除（放入回收站）、退款、退货、收货、评价等。商家可以对订单审核、关闭、发货、售后处理等。大平台对订单可以取消、更改收款状态、删除等，如果买卖双方出现纠纷，大平台还可以进行仲裁。

7. 表设计

订单设计一般包含4个表，订单主表、辅表、订单商品表和日志表，在B2B2C平台中还会增加一个支付单表。

订单主表主要存放主要及常用的订单信息，如订单编号、订单金额、运费金额、订单状态等。订单辅表主要存放辅助信息，如发货信息、发票信息、收货人信息、促销信息等。订单商品表则存放的是订单中的商品列表信息。订单主表、辅表、订单商品表之间为一对一对多的关系。

支付单表主要是为合并付款设计的，该表中会保存一个支付单号，N条订单表记录使用一个支付单号，我们可以把支付单理解成为一个总订单号，订单表中的订单编号理解成为N个子订单号，商城提交给第三方支付平台是这个总订单号，而不是每个子订单号，如果第三方支付平台支付成功，每个子订单的支付状态也都会标志为支付成功状态。支付表与订单表之间为一对多的关系。

订单日志表负责在订单内容发生变化时记录操作日志，包括操作人、操作时间、操作内容等。

订单主要表关系如图3-61所示。

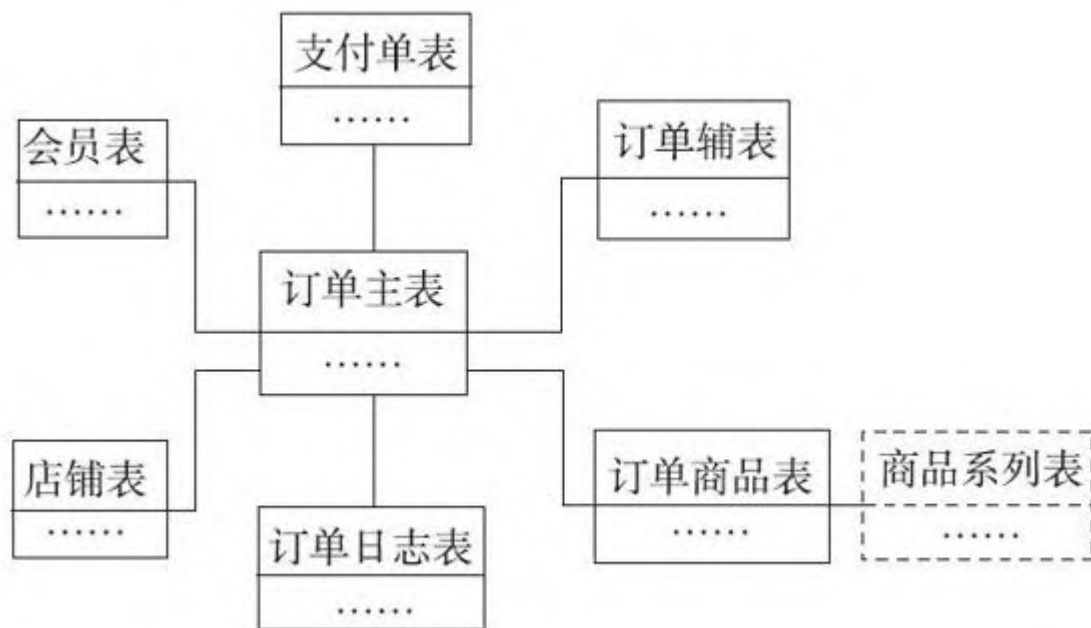


图3-61 订单主要表关系图

3.6.2 代码实现

1. 订单模型类

订单模型中封装了对订单操作所需要的绝大部分方法，下面介绍一下一些主要方法及其作用，如代码清单3-20所示，这里只保留部分核心流程代码，部分重复性内容或非主要代码予以省略。

代码清单3-20 订单模型类

```
class orderModel extends Model {  
    //取得单条订单信息，此方法还可以返回与订单相关的表信息，如订单辅表、订  
    单商品表、店铺表等  
    //在第二个参数$extend传入即可  
    $extend=array('order_common','order_goods','store')  
    public function getOrderInfo($condition = array(), $extend =  
    array(), $fields = '*', $order = '', $group = '') {  
        $order_info = $this->table('order')->field($fields)-  
        >where($condition)-> group($group)->order($order)->find();  
        //追加返回店铺信息.....  
        //追加返回买家信息.....  
        //追加返回商品信息.....  
        return $order_info;  
    }  
    //取得订单辅表单条信息  
    public function getOrderCommonInfo($condition = array(),  
    $field = '*') {
```

```

        return $this->table('order_common')->where($condition)->find();
    }
    //取得订单支付表信息
    public function getOrderPayInfo($condition = array()) {
        return $this->table('order_pay')->where($condition)->find();
    }
    //取得支付单列表
    public function getOrderPayList($condition, $pagesize = '', $field = '*', $order = '', $key = '') {
        return $this->table('order_pay')->field($field)->where($condition)->order($order)->page($pagesize)->key($key)->select();
    }
    //取得订单列表 (不包含回收站的订单)
    public function getNormalOrderList($condition, $pagesize = '', $field = '*', $order = 'order_id desc', $limit = '', $extend = array()){
        $condition['delete_state'] = 0;
        return $this->getOrderList($condition, $pagesize, $field, $order, $limit, $extend);
    }
    //取得订单列表 (所有订单, 包含回收站)
    public function getOrderList($condition, $pagesize = '', $field = '*', $order = 'order_id desc', $limit = '', $extend = array()){
        $list = $this->table('order')->field($field)->where($condition)->page($pagesize)->order($order)->limit($limit)->select();
        //追加返回订单扩展表信息.....
        //追加返回店铺信息.....
        //追加返回买家信息.....
        //追加返回商品信息.....
        return $list;
    }
    //待付款订单数量
    public function getOrderStateNewCount($condition = array())
    {
        $condition['order_state'] = ORDER_STATE_NEW;
        return $this->getOrderCount($condition);
    }
    //待发货订单数量
    public function getOrderStatePayCount($condition = array())
    {

```

```

        $condition['order_state'] = ORDER_STATE_PAY;
        return $this->getOrderCount($condition);
    }
    //待收货订单数量
    public function getOrderStateSendCount($condition = array())
    {
        $condition['order_state'] = ORDER_STATE_SEND;
        return $this->getOrderCount($condition);
    }
    //待评价订单数量
    public function getOrderStateEvalCount($condition = array())
    {
        $condition['order_state'] = ORDER_STATE_SUCCESS;
        $condition['evaluation_state'] = 0;
        $condition['finnshed_time'] = array('gt',TIMESTAMP -
ORDER_EVALUATE_TIME);
        return $this->getOrderCount($condition);
    }
    //取得订单数量
    public function getOrderCount($condition) {
        return $this->table('order')->where($condition)-
>count();
    }
    //取得订单商品表详细信息
    public function getOrderGoodsInfo($condition = array(),
$fields = '*', $order = '') {
        return $this->table('order_goods')->where($condition)-
>field($fields)->order($order)->find();
    }
    //取得订单商品表列表
    public function getOrderGoodsList($condition = array(),
$fields = '*', $limit = null, $page = null, $order = 'rec_id
desc', $group = null, $key = null) {
        return $this->table('order_goods')->field($fields)-
>where($condition)->limit($limit)->order($order)-
>group($group)->key($key)->page($page)->select();
    }
    //取得订单扩展表列表
    public function getOrderCommonList($condition = array(),
$fields = '*', $limit = null) {
        return $this->table('order_common')->field($fields)-
>where($condition)->limit($limit)->select();
    }
    //插入订单支付表信息
    public function addOrderPay($data) {

```

```

        return $this->table('order_pay')->insert($data);
    }
    //插入订单表信息
    public function addOrder($data) {
        return $this->table('order')->insert($data);
    }
    //插入订单扩展表信息
    public function addOrderCommon($data) {
        return $this->table('order_common')->insert($data);
    }
    //插入订单扩展表信息
    public function addOrderGoods($data) {
        return $this->table('order_goods')->insertAll($data);
    }
    //添加订单日志
    public function addOrderLog($data) {
        $data['log_role'] =
        str_replace(array('buyer','seller','system'),array('买家','商家','系统'), $data['log_role']);
        $data['log_time'] = TIMESTAMP;
        return $this->table('order_log')->insert($data);
    }
    //更改订单信息
    public function editOrder($data,$condition) {
        return $this->table('order')->where($condition)->update($data);
    }
    //更改订单扩展信息
    public function editOrderCommon($data,$condition) {
        return $this->table('order_common')->where($condition)->update($data);
    }
    //更改订单支付信息
    public function editOrderPay($data,$condition) {
        return $this->table('order_pay')->where($condition)->update($data);
    }
    //订单操作历史列表
    public function getOrderLogList($condition) {
        return $this->table('order_log')->where($condition)->select();
    }
    //取得单条订单操作记录
    public function getOrderLogInfo($condition = array(), $order = '') {

```

```

        return $this->table('order_log')->where($condition)->order($order)->find();
    }
    //订单操作权限判断方法，允许操作返回真，否则返回假
    public function getOrderOperateState($operate,$order_info){
        switch ($operate) {
            //买家取消订单
            case 'buyer_cancel':
                $state = ($order_info['order_state'] ==
ORDER_STATE_NEW) ||
                ($order_info['payment_code'] == 'offline' &&
$order_info['order_state'] == ORDER_STATE_PAY);
                break;
            //申请退款
            case 'refund_cancel':
                $state = $order_info['refund'] == 1 &&
!intval($order_info['lock_state']);
                break;
            //商家取消订单.....
            //平台取消订单.....
            //平台收款.....
            //买家投诉.....
            //调整运费.....
            //发货.....
            //收货.....
            //评价.....
            //锁定.....
            //快递跟踪.....
            //放入回收站.....
            //永久删除.....
            //从回收站还原.....
        }
        return $state;
    }
    //取消订单
    private function _memberChangeStateOrderCancel($order_info,
$member_id, $member_name, $extend_msg) {
        //解冻预存款.....
        //更新订单信息.....
        //添加订单日志.....
    }
    //收货操作
    private function _memberChangeStateOrderReceive($order_info,
$member_id, $member_name, $extend_msg) {
        //更新订单状态.....

```



```
        //添加订单日志.....
    }
    //回收站操作（放入回收站、还原、永久删除）
    private function _memberChangeStateOrderRecycle($order_info,
    $member_id, $member_name, $state_type) {
        //.....
    }
}
```

2. 下单业务模型

购买模型将下单过程中涉及各种业务处理的方法进行封装，使下单过程代码更加简洁高效。下面介绍一下一些主要方法及其作用，如代码清单3-21所示，这里只保留部分核心流程代码，部分重复性内容或非主要代码予以省略。

代码清单3-21 下单业务模型

```
class buyModel {
    //输出有货到付款时，在线支付和货到付款及每种支付下商品数量和详细列表
    //返回以支付方式为下标分组的商品列表
    public function getOfflineGoodsPay($buy_list) {
        //.....
    }
    //计算每个店铺(所有店铺级优惠活动)总共优惠多少金额
    //参数1为最初店铺商品总金额
    //参数2为去除各种店铺级促销后，最终店铺商品总金额(不含运费)
    public function getStorePromotionTotal($store_goods_total,
    $store_final_goods_total) {
```

```

        //.....
    }
    //取得每个店铺运费
    //参数1为购买商品列表
    //参数2为免运费的店铺ID数组
    public function getStoreFreightList($buy_list = array(),
    $free_freight_sid_list) {
        //先将免运费的店铺运费置0 (格式:店铺ID=>0).....
        //然后算出店铺里没使用运费模板 (优惠套装商品除外) 的商品运费之和 (格
        式:店铺ID=>运费) .....
        //定义数组, 存放店铺优惠套装商品运费总额 store_id=>运费.....
        //免运费店铺的商品不需要计算.....
        //优惠套装商品运费另算.....
        //计算优惠套装商品运费.....
        //最后再计算使用运费模板的信息 (店铺ID, 运费模板ID, 购买数量), 使用
        相同运费模板的商品数量累加.....
    }
    //根据地区选择计算出所有店铺最终运费
    //参数分别为运费信息和城市ID
    public function calcStoreFreight($freight_list, $city_id) {
        //首先处理免运费费和固定运费.....
        //然后计算使用运费模板的商品运费.....
    }
    //追加赠品到下单列表, 并更新购买数量
    //分别传入购买列表、赠品列表、满即送规则
    public function appendPremiumsToCartList($store_cart_list,
    $store_premiums_list, $store_mansong_rule_list) {
        //取得每种商品的库存.....
        //取得每种商品的购买量.....
        //本次购买后, 余库存为0的, 则后面不再送赠品.....
        //将赠品追加到购买列表.....
        //如果没有库存了, 则不再送赠品.....
    }
    //取得每种商品的库存, 传入购买列表和赠品列表
    private function
    _getEachGoodsStorageQuantity($store_cart_list,
    $store_premiums_list = array()) {
        //普通商品.....
        //优惠套装商品.....
        //赠品.....
    }
    //取得每种商品的购买量, 传入购买列表
    private function _getEachGoodsBuyQuantity($store_cart_list)
    {
        //普通商品.....

```

```

        //优惠套装.....
    }
    //生成订单
    public function createOrder($input, $member_id,
    $member_name, $member_email) {
        //拆分收集的购买信息
        extract($input);
        $model_order = Model('order');
        //存储生成的订单,函数会返回该数组
        $order_list = array();
        //每个店铺订单是货到付款还是线上支付,店铺ID=>付款方式[在线支付/货到付款]
        $store_pay_type_list = $this->getStorePayTypeList(array_keys($store_cart_list),
        $if_offpay, $pay_name);
        //产生支付单号
        $pay_sn = $this->makePaySn($member_id);
        //创建支付单
        $order_pay = array();
        $order_pay['pay_sn'] = $pay_sn;
        $order_pay['buyer_id'] = $member_id;
        $order_pay_id = $model_order->addOrderPay($order_pay);
        //处理收货人信息.....
        //循环生成子订单
        foreach ($store_cart_list as $store_id => $goods_list) {
            //取得本店优惠额度(后面用来计算每件商品实际支付金额, 结算需要)
            $promotion_total =
            !empty($store_promotion_total[$store_id]) ?
            $store_promotion_total[$store_id] : 0;
            //本店总的优惠比例,保留3位小数
            $should_goods_total =
            $store_final_order_total[$store_id]-$store_freight_total[$store_id]+$promotion_total;
            $promotion_rate =
            abs($promotion_total/$should_goods_total);
            if ($promotion_rate <= 1) {
                $promotion_rate =
                floatval(substr($promotion_rate,0,5));
            } else {
                $promotion_rate = 0;
            }
            //每种商品的优惠金额累加保存 $promotion_sum
            $promotion_sum = 0;
            //预定义订单主表、辅表、订单商品表存放数组
            $order = array();

```

```

$order_common = array();
$order_goods = array();
//生成订单主要数据
//以支付单为基础产生子订单号
$order['order_sn'] = $this->makeOrderSn($order_pay_id);
$order['pay_sn'] = $pay_sn;
$order['store_id'] = $store_id;
$order['store_name'] = $goods_list[0]['store_name'];
$order['buyer_id'] = $member_id;
// $order[.....] = .....
$order_id = $model_order->addOrder($order);
$order['order_id'] = $order_id;
$order_list[$order_id] = $order;
//生成订单辅助数据
$order_common['order_id'] = $order_id;
$order_common['store_id'] = $store_id;
$order_common['order_message'] =
$pay_message[$store_id];
//代金券信息
if (isset($voucher_list[$store_id])) {
    $order_common['voucher_price'] =
    $voucher_list[$store_id]['voucher_price'];
    $order_common['voucher_code'] =
    $voucher_list[$store_id]['voucher_code'];
}
//收货人信息
$order_common['reciver_info'] = $reciver_info;
$order_common['reciver_name'] = $reciver_name;
//发票信息
$order_common['invoice_info'] = $this-
>_createInvoiceData($invoice_info);
//促销信息
if (is_array($store_mansong_rule_list[$store_id])) {
    $order_common['promotion_info'] =
    addslashes($store_mansong_rule_list[$store_id]['desc']);
}
$order_id = $model_order->addOrderCommon($order_common);
//生成多条订单商品数据
$i = 0;
foreach ($goods_list as $goods_info) {
    if (!$goods_info['state'] ||
    !$goods_info['storage_state']) {
        throw new Exception('部分商品已经下架或库存不足，请重新选
择');
    }
}

```

```

        if (!intval($goods_info['bl_id'])) {
            //普通商品
            $order_goods[$i]['order_id'] = $order_id;
            $order_goods[$i]['goods_id'] =
$goods_info['goods_id'];
            $order_goods[$i]['store_id'] = $store_id;
            $order_goods[$i]['goods_name'] =
$goods_info['goods_name'];
            $order_goods[$i]['goods_price'] =
$goods_info['goods_price'];
            //$order_goods[$i][.....] = .....
            //计算商品金额
            $goods_total = $goods_info['goods_price'] *
$goods_info['goods_num'];
            //计算本件商品优惠金额
            $promotion_value = floor($goods_total*
($promotion_rate));
            $order_goods[$i]['goods_pay_price'] = $goods_total -
$promotion_value;
            $promotion_sum += $promotion_value;
            $i++;
        } elseif (!empty($goods_info['bl_goods_list']) &&
is_array($goods_info['bl_goods_list'])) {
            //优惠套装
            foreach ($goods_info['bl_goods_list'] as
$bl_goods_info) {
                $order_goods[$i]['order_id'] = $order_id;
                $order_goods[$i]['goods_id'] =
$bl_goods_info['goods_id'];
                $order_goods[$i]['store_id'] = $store_id;
                $order_goods[$i]['goods_name'] =
$bl_goods_info['goods_name'];
                //$order_goods[$i][.....] = .....
                //计算商品实际支付金额(goods_price减去分摊优惠金额后的
值)
                $goods_total = $bl_goods_info['bl_goods_price']
* $goods_info['goods_num'];
                //计算本件商品优惠金额
                $promotion_value = floor($goods_total*
($promotion_rate));
                $order_goods[$i]['goods_pay_price'] =
$goods_total - $promotion_value;
                $promotion_sum += $promotion_value;
                $i++;
            }
        }
    }
}

```

```

    }
    }
    $insert = $model_order->addOrderGoods($order_goods);
}
return array($pay_sn,$order_list);
}
//生成支付单编号(两位随机 + 从2000-01-01 00:00:00 到现在的秒数+微
秒+会员ID%1000)，该值会//传给第三方支付接口
//长度 =2位 + 10位 + 3位 + 3位 = 18位
public function makePaySn($member_id) {
    return mt_rand(10,99)
        . sprintf('%010d',time() - 946656000)
        . sprintf('%03d', (float) microtime() * 1000)
        . sprintf('%03d', (int) $member_id % 1000);
}
//以支付单号为基础生成订单号
//支付单号与订单号关系：一对多
public function makeOrderSn($pay_id) {
    //记录生成子订单的个数，如果生成多个子订单，该值会累加
    static $num;
    if (empty($num)) {
        $num = 1;
    } else {
        $num ++;
    }
    return (date('y',time()) % 9+1) . sprintf('%013d',
$pay_id) . sprintf('%02d', $num);
}
//收货地址变化时计算运费
public function changeAddr() {
    //选择不同地区时，异步处理并返回每个店铺总运费以及本地区是否能使用
货到付款
    //如果店铺统一设置了满免运费规则，则运费模板无效
    //如果店铺未设置满免规则，且使用运费模板，按运费模板计算
    //如果其中有商品使用相同的运费模板，则两种商品数量相加后再应用该运费
模板计算（即作为一种商品 //算运费）
    //如果未找到运费模板，按免运费处理
    //如果没有使用运费模板，商品运费按快递价格计算，运费不随购买数量增加
}
}

```

3. 下单

一般下单过程都要经过以下4步才能完成，如图3-62所示。

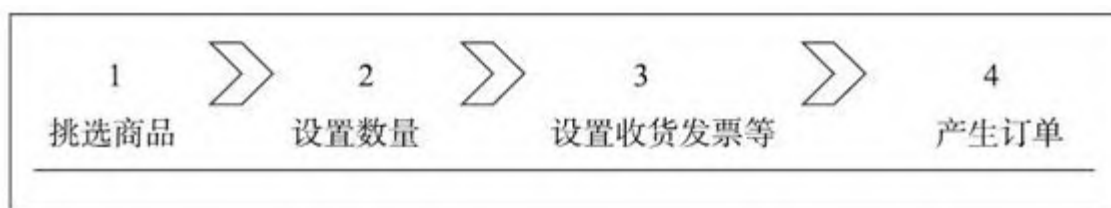


图3-62 下单步骤

前3步我们对购物车、配送模块已有所介绍，本节重点介绍第4步，也就是最后一步订单的生成过程。系统在产生订单时会做大量的处理工作，比如处理收货信息、发票信息、促销信息、运费、代金券、支付单、订单、日志等，如图3-63所示。



图3-63 生成订单系统内部详细处理流程图

系统保存订单时核心代码如代码清单3-22所示。

代码清单3-22 保存订单核心代码


```

//实例化购物车模型，因为有可能是从购物过来的
$model_cart = Model('cart');
//取得商品ID和购买数量
$input_buy_items = $this->_parseItems($post['cart_id']);
//处理收货地址.....
//处理发票信息.....
//是否支持货到付款.....
//付款方式:在线支付/货到付款(online/offline).....
//验证代金券.....
if ($post['ifcart']) {
    //来源于购物车
    //取购物车列表
    $condition =
array('cart_id'=>array('in',array_keys($input_buy_items)),'b
uyer_id'=>$member_id);
    $cart_list = $model_cart->listCart('db',$condition);
    //取商品最新的在售信息
    $cart_list = $model_cart->getOnlineCartList($cart_list);
    //得到限时折扣信息
    $cart_list = $model_cart-
>getXianshiCartList($cart_list);
    //得到优惠套装状态,并取得组合套装商品列表
    $cart_list = $model_cart-
>getBundlingCartList($cart_list);
    //得到商品列表
    $goods_list = $model_cart->getGoodsList($cart_list);
    //购物车列表以店铺ID分组显示
    $store_cart_list = $model_cart-
>getStoreCartList($cart_list);
} else {
    //来源于直接购买
    //取得购买的商品ID和购买数量
    foreach ($input_buy_items as $goods_id => $quantity)
{break;}
    //取得商品最新在售信息
    $goods_info = $model_cart-
>getGoodsOnlineInfo($goods_id,$quantity);
    //判断是不是正在团购中，如果是则按团购价格计算，购买数量若超过团购规
定的上限，则按团购上限计算
    $goods_info = $model_cart->getGroupbuyInfo($goods_info);
    //如果未进行团购，则再判断是否限时折扣中
    if (!$goods_info['ifgroupbuy']) {
        $goods_info = $model_cart-
>getXianshiInfo($goods_info,$quantity);
    } else {

```

```

        //记录团购数量，订单完成后需要更新一下团购表信息
        $groupbuy_info = array();
        $groupbuy_info['groupbuy_id'] =
$goods_info['groupbuy_id'];
        $groupbuy_info['quantity'] = $quantity;
    }
    //转成多维数组，与购物车统一使用购物车方法与模板
    $store_cart_list = array();
    $goods_list = array();
    $goods_list[0] =
$store_cart_list[$goods_info['store_id']][0] = $goods_info;
}
//商品金额计算(分别对每个商品/优惠套装小计、每个店铺小计)
list($store_cart_list,$store_goods_total) = $model_cart-
>calcCartList($store_cart_list);
//取得店铺优惠 - 满即送(赠品列表、店铺满送规则列表)
list($store_premiums_list,$store_mansong_rule_list) =
$model_cart-
>getMansongRuleCartListByTotal($store_goods_total);
//重新计算店铺扣除满即送后商品实际支付金额
$store_final_goods_total = $model_cart-
>reCalcGoodsTotal($store_goods_total,$store_mansong_rule_lis
t,'mansong');
//得到有效的代金券
$input_voucher_list = $model_cart-
>reParseVoucherList($input_voucher_list,$store_goods_total,$
member_id);
//重新计算店铺扣除优惠券送商品实际支付金额
$store_final_goods_total = $model_cart-
>reCalcGoodsTotal($store_final_goods_total,$input_voucher_li
st,'voucher');
//计算每个店铺(所有店铺级优惠活动)总共优惠多少
$store_promotion_total = $this-
>getStorePromotionTotal($store_goods_total,
$store_final_goods_total);
//计算每个店铺运费
list($need_calc_sid_list,$cancel_calc_sid_list) = $this-
>getStoreFreightDescList($store_final_goods_total);
$freight_list = $this-
>getStoreFreightList($goods_list,array_keys($cancel_calc_sid
_list));
$store_freight_total = $this-
>calcStoreFreight($freight_list,$input_city_id);
//计算店铺最终订单实际支付金额(加上运费)
$store_final_order_total = $model_cart-

```

```

>reCalcGoodsTotal($store_final_goods_total,$store_freight_to
tal,'freight');
//计算店铺分类佣金
$store_gc_id_commis_rate_list = $this-
>getStoreGcidCommisRateList($goods_list);
//将赠品追加到购买列表(如果库存不足, 则不送赠品)
$append_premiums_to_cart_list = $this-
>appendPremiumsToCartList($store_cart_list,$store_premiums_l
ist,$store_mansong_rule_list,$member_id);
if(!empty($append_premiums_to_cart_list['error'])) {
    return array('error' =>
$append_premiums_to_cart_list['error']);
} else {

list($store_cart_list,$goods_buy_quantity,$store_mansong_rul
e_list) = $append_premiums_to_cart_list;
}
//整理已经得出的固定数据, 准备下单
$input = array();
$input['pay_name'] = $input_pay_name;
$input['if_offpay'] = $input_if_offpay;
$input['if_vat'] = $input_if_vat;
$input['pay_message'] = $post['pay_message'];
$input['address_info'] = $input_address_info;
$input['invoice_info'] = $input_invoice_info;
$input['voucher_list'] = $input_voucher_list;
$input['store_goods_total'] = $store_goods_total;
$input['store_final_order_total'] =
$store_final_order_total;
$input['store_freight_total'] = $store_freight_total;
$input['store_promotion_total'] = $store_promotion_total;
$input['store_gc_id_commis_rate_list'] =
$store_gc_id_commis_rate_list;
$input['store_mansong_rule_list'] =
$store_mansong_rule_list;
$input['store_cart_list'] = $store_cart_list;
$input['input_city_id'] = $input_city_id;
try {
    //开始事务
    $model_cart->beginTransaction();
    //生成订单
    list($pay_sn,$order_list) = $this->createOrder($input,
$member_id, $member_name, $member_email);
    //记录订单日志
    $this->addOrderLog($order_list);

```

```
//变更库存和销量
$this->updateGoodsStorageNum($goods_buy_quantity);
//更新使用的代金券状态
$this->updateVoucher($input_voucher_list);
//更新团购购买人数和数量
$this->updateGroupbuy($groupbuy_info);
//使用预存款支付
$this->pdPay($order_list, $post, $member_id,
$member_name);
//提交事务
$model_cart->commit();
}catch (Exception $e){
    //回滚事务
    $model_cart->rollback();
    return array('error' => $e->getMessage());
}
//删除购物车中的商品
if ($post['ifcart']) {
    $model_cart-
>delCart('db',array('buyer_id'=>$member_id,'cart_id'=>array(
'in',array_keys($input_buy_items))));
}
//通知队列处理其他事项.....
```

3.6.3 功能体验

首先做好前面的准备工作，选择商品和购买数量，进入设置收货地址、发票信息页面，然后设置收货人地址，如图3-64所示。



图3-64 设置收货人地址

然后选择支付方式，这里选择货到付款，因为购买的是多个商家的商品，包含平台自营（它支持货到付款），所以本次下单会以货到付款+在线支付的组合形式支付，并且系统会弹出确定信息供顾客确认，如图3-65所示。

然后选择“发票信息”按钮，系统支持普通发票和增值税发票两种。只有自营店铺才支持增值税发票，这里我们选择个人即可，如图3-66所示。

然后设置商品信息，如留言、使用哪些优惠等，这里以设置平台店铺商品为例，如图3-67所示。



图3-65 确认支付方式



图3-66 设置发票

商品	单价(元)	数量	小计
官方店铺 满100减10			
 2014春装新品男士骷髅头个性图案长袖T恤韩版纯色修身打底衫 白色 S	¥112.00	1	¥112.00
 2014春装新品男士骷髅头个性图案长袖T恤韩版纯色修身打底衫 灰色 S	¥112.00	1	¥112.00
买家留言: 请给个赠品, 谢谢!			
			运费: ¥0.00
			商品总额: ¥224.00
			满100减10: ¥-10.00
			选择代金券: ¥-0.00
			选择代金券: ¥214.00
满10元 有效期至 2014-05-28			
背包包			
 欧美摩登通勤系列多用包 黑色	¥358.00	1	¥358.00
 欧美摩登通勤系列多用包 深蓝色	¥358.00	1	¥358.00
买家留言:			

图3-67 设置商品信息

由于有部分预存款余额，这里使用预存款支付10元，不足金额线上支付补齐，如图3-68和图3-69所示。

☒ 使用预存款支付 (当前可用余额: ¥10.00)

登录密码:

订单总金额: ¥ 11694.00 元

图3-68 使用预存款支付

部分商品需要在线支付，请尽快付款！在线支付金额：¥11470.00

订单号	支付方式	金额	物流
由于您的商品由不同商家发出，此单将分为2个不同子订单配送！			
6000000000045204	在线支付	¥264.00	快递
6000000000045203	在线支付	¥10500.00	快递
6000000000045202	在线支付（已使用预存款部分支付，支付金额¥10.00）	¥716.00	快递
6000000000045201	货到付款	¥214.00	快递

支付选择

确认提交支付

图3-69 订单待支付页面

可以发现自动将本次购买的商品拆分成了多个子订单，其中使用货到付款的是平台自营店铺，订单总金额中扣除预存款支付的金额和货到付款的金额后，就是需要在线支付的金额，这里是11470.00元。选择其中一个第三方支付平台完成支付即可。

如果我们未通过第三方平台进行支付，进入“我的订单”按钮列表，如图3-70所示。

我们可以对每个子订单进行取消和投诉操作（只有货到付款的订单才可以投诉）。可以分别查看每个子订单的详细信息。支付时这些子订单会作为一个整体统一支付。如果不想购买其中某些子订单可以取消它，如图3-71所示，我们只购买“妮妮服装店”的商品。

可以发现所需要支付金额也发生了变化，再去支付时，就只有一个子订单了，如图3-72所示。

下单时间: 2014-04-11 10:24:08		在线支付金额: ¥11480.00		订单支付	
订单号: 6000000000045204 下单时间: 2014-04-11 10:24:08 妮妮服装店  查看订单					
	阿狸国巾女冬天韩版韩国可爱萌萌双层加厚帽子围巾手套一体 萌米	¥96.00 x 1	在线付款 ¥264.00 (免运费)	待付款 取消订单	
	冬装新款女装韩版两件套娃娃领长袖厚毛衣针织打底 撞色 M	¥168.00 x 1			
订单号: 6000000000045203 下单时间: 2014-04-11 10:24:08 健康壹佰运动器材  查看订单					
	BH 必艾奇 tecnovita YV82 企鹅跑步机 台湾制造进口 YV82 企鹅跑步机	¥720.00 x 1	在线付款 ¥10500.00 (免运费)	待付款 取消订单	
	BH必艾奇 H919 动感单车 飞轮车 健身车 家用 原装进口 磁控铝盘	¥9780.00 x 1			
订单号: 6000000000045202 下单时间: 2014-04-11 10:24:08 麦包包  查看订单					
	浪美摩登通勤系列多用包 深蓝色	¥358.00 x 1	在线付款 ¥716.00 (免运费)	待付款 取消订单	
	浪美摩登通勤系列多用包 黑色	¥358.00 x 1			
订单号: 6000000000045201 下单时间: 2014-04-11 10:24:08 官方旗舰店  查看订单					
	2014春装新品男士骷髅头个性圆领长袖恤韩版纯色修身打底衫 灰色 S	¥112.00 x 1	货到付款 ¥214.00 (免运费)	待发货 取消订单 投诉	
	2014春装新品男士骷髅头个性圆领长袖恤韩版纯色修身打底衫 白色 S	¥112.00 x 1			

图3-70 我的订单列表

下单时间: 2014-04-11 10:24:08		在线支付金额: ¥264.00		订单支付	
订单号: 6000000000045204 下单时间: 2014-04-11 10:24:08 妮妮服装店  查看订单					
	阿狸国巾女冬天韩版韩国可爱萌萌双层加厚帽子围巾手套一体 萌米		¥96.00 x 1	在线付款 ¥264.00 (免运费)	待付款 取消订单
	冬装新款女装韩版两件套娃娃领长袖厚毛衣针织打底 撞色 M		¥168.00 x 1		
订单号: 6000000000045203 下单时间: 2014-04-11 10:24:08 健康壹佰运动器材  查看订单					
	BH必艾奇 tecnovita YV82 企鹅跑步机 台湾制造进口 YV82 企鹅跑步机		¥720.00 x 1	在线付款 ¥10500.00 (免运费)	已取消
	BH必艾奇 H919 动感单车 飞轮车 健身车 家用 原装进口 磁控铝盘		¥9780.00 x 1		

图3-71 我的订单列表

请您及时付款，以便订单尽快处理！在线支付金额：¥264.00

订单号	支付方式	金额	物流
6000000000045204	在线支付	¥264.00	快递

支付选择

支付宝

财付通

网银在线
ChinaBank Payment

确认提交支付

图3-72 订单待支付页面

3.7 支付接口

在线支付是电商平台需要提供的一项基础服务，买家购买商品需要在线支付，买家预存款充值也需要在线支付，商家购买各种服务也同样需要在线支付。

我们知道支付会涉及资金，而资金是存放在银行的，资金的变化必须经过银行，电商平台必须要与银行签约，由银行提供给商户技术接口、身份认证信息，然后由平台技术人员将接口程序接入，这样就实现了将买家账户的款项支付到自己的银行账户中，也就实现了在线支付。但这样又出现了新的问题，买家所持有的银行账户是各种各样的，所以要实现大多银行卡可以在平台上支付，平台运营者需要单独与各家银行进行签约，这样会造成接入的成本非常高，这时我们就考虑接入专业的第三方支付平台接口了，这些第三方支付平台直接与各商业银行进行签约，然后对

外提供统一的支付接口，接入效率大大提高，像大家熟知的支付宝、财付通、网银在线、快钱等都是拥有央行支付牌照的主流支付平台。

支付接口除了第三方支付平台之外，还有两个特殊的：分别是货到付款和预存款，本节主要介绍第三方支付接口的接入，货到付款不涉及在线支付程序的开发，预存款我们会有专门的章节作详细介绍。

3.7.1 接入原理

接入支付接口到电商平台，首先与支付公司签约，签约商户将获得相关的商户号、密钥、支付网关及开发文档等资料，技术人员通过阅读这些文档，获取支付时需要传递的数据参数，如商品名称、金额、运费等，并通过约定的加密方式结合密钥进行加密，最后以POST或GET的方式发送给支付网关。

支付网关收到商户提交的相关参数后进行数据处理及签名验证，如果数据和签名均验证成功，将引导用户进入银行选择页面，用户选择自己使用的银行后，进入相关银行的支付网关进行支付。

用户付款成功后，银行进行扣款，并将支付结果回传给第三方支付平台，支付平台收到银行的结果，同样进行签名验证，并将支付结果传递给商户网站。

支付公司将支付结果传递给商户网站也是通过POST或GET方式，只是该数据传递属于服务器后端处理的。商户网站向支付平台发起支付请求时，支付相关参数中就包含了用于接收支付结果的回传URL。

商户网站接入支付结果有两种方式：一种是同步的，即通过浏览器进行跳转通知；一种是异步的，即服务器后端执行。同步通知主要是基于用户访问的浏览器，如果用户在银行页面支付成功后，直接关闭了页面，并未等待银行跳转到支付结果页面，那么商户网站就收不到支付结果的通知，导致支付结果难以处理，这时就需要异步通知来处理。异步方式是支付平台服务器后台直接向用户指定的异步通知URL以POST/GET方式发送参数。商户网站对接收的数据及合法性进行验证，如果验证通过则进行后续逻辑操作，如更新支付状态、支付时间等。支付平台处理流程如图3-73所示。

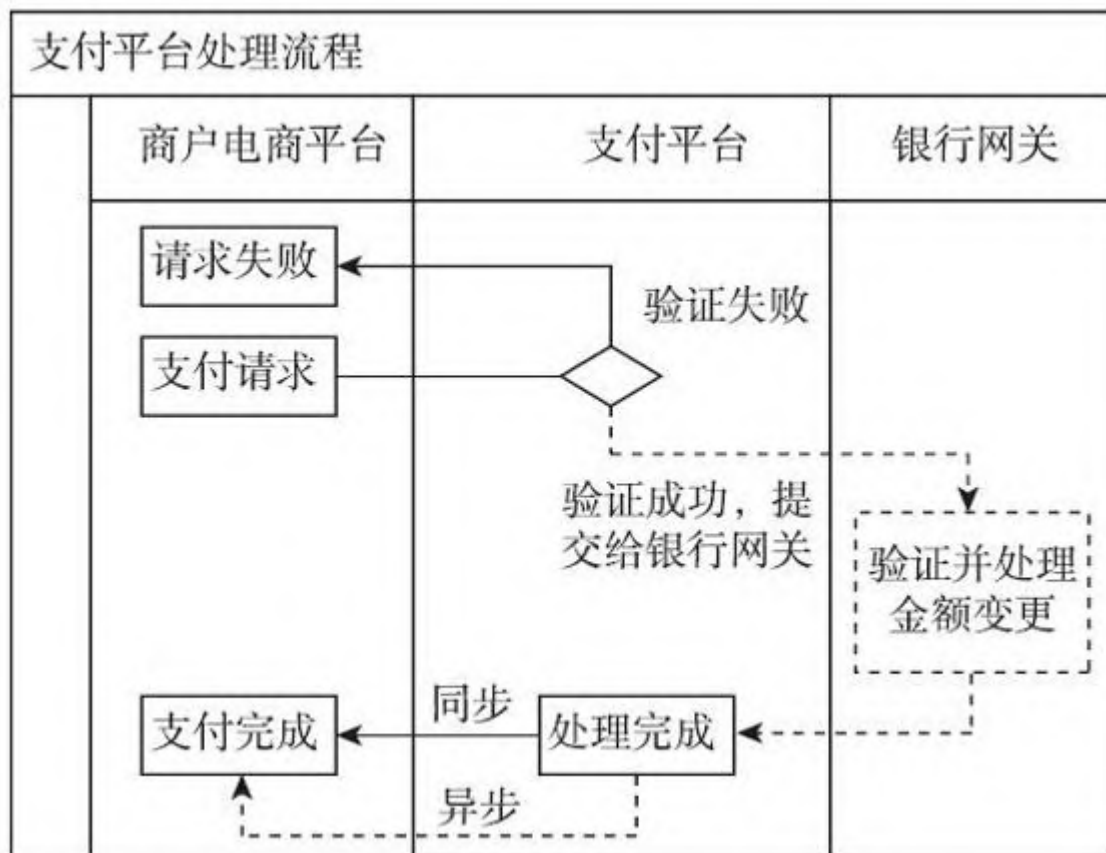


图3-73 支付平台处理流程

支付平台主要提供了两种支付方式：担保交易和即时到账，我们这里只介绍即时到账方式，这种方式更适合目前大多数电商平台。

3.7.2 设计思路

1. 设计要求

(1) 安全性

开发人员需要充分了解在线支付接口的工作原理，认真阅读支付平台提供的相关开发文档。接口集成时要严格按照开发文档的要求进行开发。

(2) 数据完整性

任何涉及金钱的操作都必须保证数据的完整性，对于重要的字段数据变更要全部使用事务进行处理，一个点发生错误，整个事务都需要回滚，以保证数据的完整性和准确性。

(3) 扩展性

对每种支付平台接入时目录存放、类库引入、安全校验等都要统一规划，方便代码维护与二次扩展。ShopNC B2B2C平台支付接口目录分布如图3-74所示。

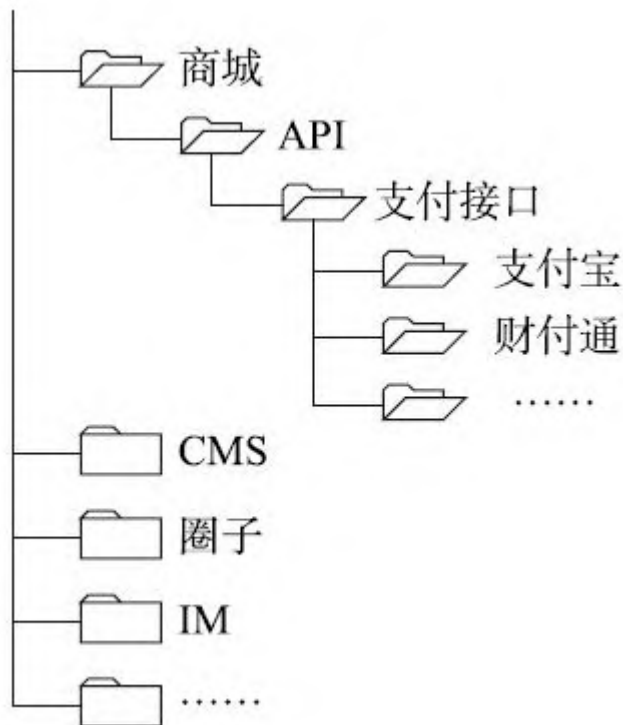


图3-74 支付接口目录分布

2. 数据表设计

支付接口表至少要包含支付方式的名称和标识码，这个标识码要与支付接口API程序存放的目录名一

致，这样会极大地方便后面的开发工作。由于每种支付接口对外提示的认证账号信息不太相同，我们不可能为每种支付方式都留有自己的字段，所以建议将每种支付方式的账号配置信息序列化后存入统一的接口配置字段，使用时再反序列化即可。另外还应该增加一个状态字段，供管理者随时停止或启用某种支付方式。支付接口表完整表结构如表3-7所示。

表3-7 支付接口表结构

字 段	类 型	Null	默 认	说 明
payment_id	tinyint(1)	否		主键
payment_code	char(10)	否	0	支付标识码
payment_name	char(10)	否	0	支付名称
payment_config	Text	否	0	支付接口配置信息
payment_state	enum('0','1')	否	0	支付接口状态 0 关闭 1 开启

3.7.3 代码实现

1. 支付模型类

支付模型类提供了对支付接口表的主要方法，供前台Control来调用，如代码清单3-23所示。

代码清单3-23 支付模型

```
//定义状态标识
const STATE_OPEN = 1;
//读取单行信息
public function getPaymentInfo($condition = array()) {
    return $this->where($condition)->find();
}
//读取开启中的单行信息
public function getPaymentOpenInfo($condition = array()) {
    $condition['payment_state'] = self::STATE_OPEN;
    return $this->where($condition)->find();
}
//读取多行
public function getPaymentList($condition = array()){
    return $this->where($condition)->select();
}
//读取开启中的支付方式列表
public function getPaymentOpenList($condition = array()){
    $condition['payment_state'] = self::STATE_OPEN;
    return $this->where($condition)->key('payment_code')->select();
}
//更新信息
```

```
public function editPayment($data, $condition){
    return $this->where($condition)->update($data);
}
```

2. 支付接口类

支付接口类是以第三方支付平台提供的API为基础，经过适当地二次开发而成，这里以使用较普遍的支付宝为例，如代码清单3-24所示。

代码清单3-24 支付接口类

```
//类名与支付表中的代码标识一致
class alipay{
//支付宝网关地址
private $alipay_gateway_new =
'https://mapi.alipay.com/gateway.do?';
//消息验证地址
private $alipay_verify_url =
'https://mapi.alipay.com/gateway.do?service=notify_verify&';
//支付接口标识
private $code = 'alipay';
//支付接口配置信息
private $payment;
//订单信息
private $order;
//发送至支付宝的参数
private $parameter;
//构造函数，完成基础数据初始化工作
public function
__construct($payment_info=array(),$order_info=array()){
    //建议安装openssl扩展，这样参数将进行加密处理，更加安全
    if (!extension_loaded('openssl')) $this-
```

```

>alipay_verify_url =
'http://notify.alipay.com/trade/notify_query.do?';
    if(!empty($payment_info) and !empty($order_info)){
        $this->payment      = $payment_info;
        $this->order        = $order_info;
    }
}
//获取支付接口的请求地址，这里会完成参数的整理与签名工作
public function get_payurl(){
    $this->parameter = array(
        //异步通知的地址
        'notify_url'      =>
SHOP_SITE_URL."/api/payment/alipay/notify_url.php",
        //同步通知地址
        'return_url'      =>
SHOP_SITE_URL."/api/payment/alipay/return_url.php",
        //商品名称，这里可以使用订单号 + 购买的其中一个商品名称
        //代替
        'subject'         => $this->
>order['subject'],
        //这里的订单总价是包含运费的
        'price'           => $this->order['pay_amount'],//
订单总价
        //数量都设置为1，具体数量会在商城平台的订单表中保存
        'quantity'        => 1, //商品数量
        //运费统一设置为0，在商城平台的订单表中会保存实际值
        'total_fee'       => 0, //物流配送费用
        //其他参数省略.....
    );
    $this->parameter['sign']      = $this->sign($this->
>parameter);
    //数据处理完成后，返回支付平台API+数据参数组成的完整URL
    return $this->create_url();
}
//异步通知验证函数
public function notify_verify() {
    $param      = $_POST;
    $param['key']      = $this->
>payment['payment_config']['alipay_key'];
    $verifyfy_url = $this->alipay_verify_url. "partner="
.$this->payment['payment_config']['alipay_partner'].
"&notify_id=".$param["notify_id"];
    $verifyfy_result = $this->
>getHttpResponse($verifyfy_url);
    $mysign = $this->sign($param);

```

```

        if (preg_match("/true$/i",$verify_result) && $mysign
== $param["sign"]) {
            return true;
        } else {
            return false;
        }
    }
    //同步通知返回验证
    public function return_verify() {
        $param = $_GET;
        //将系统的控制参数置空，防止因为加密验证出错
        $param['act'] = "";
        $param['op'] = "";
        $param['key'] = $this->
>payment['payment_config']['alipay_key'];
        $verify_url = $this->alipay_verify_url. "partner="
.$this->payment['payment_config']['alipay_partner'].
"&notify_id=".$param["notify_id"];
        $verify_result = $this->
>getHttpResponse($verify_url);
        $mysign = $this->sign($param);
        if (preg_match("/true$/i",$verify_result) && $mysign
== $param["sign"]) {
            $this->order_type = $param['extra_common_param'];
            return true;
        } else {
            return false;
        }
    }
    //取得订单支付状态，成功或失败
    public function getPayResult($param){
        return $param['trade_status'] == 'TRADE_SUCCESS';
    }
    //魔术方法，用于取得内部私有方法
    public function __get($name){
        return $this->$name;
    }
    //以下为接口提供标准的内容，此处省略
    //...
}

```

3. 返回通知

支付接口返回通知分为同步通知和异步通知两种，有的接口将同步和异步通知进行合并，如财付通，有的同步和异步是分离的，支持设置不同的通知接口，如支付宝、网银在线等。两种通知的核心代码是相同的，只是接收参数略有不同而已，我们这里以同步通知为例，如代码清单3-25所示。

代码清单3-25 返回通知代码

```
//取得交易号
$out_trade_no = $_GET['out_trade_no'];
//取得订单信息
$condition = array();
$model_order = Model('order');
$condition['pay_sn'] = $out_trade_no;
$order_pay_info = $model_order->getOrderPayInfo($condition);
//取得订单列表和API支付总金额
$order_list = $model_order->getNormalOrderList(array('pay_sn'=>$out_trade_no, 'order_state'=>ORDER_STATE_NEW));
$pay_amount = $api_pay_amount = 0;
foreach($order_list as $order_info) {
    $api_pay_amount +=
    ncPriceFormat(floatval($order_info['order_amount']) -
    floatval($order_info['pd_amount']));
    $pay_amount += floatval($order_info['order_amount']);
}
$order_pay_info['pay_amount'] = $api_pay_amount;
//取得支付接口信息
$payment_code = $_GET['payment_code'];
```



```

$model_payment = Model('payment');
$condition = array();
$condition['payment_code'] = $payment_code;
$payment_info = $model_payment-
>getPaymentOpenInfo($condition);
$payment_info['payment_config']          =
unserialize($payment_info['payment_config']);
$inc_file =
BASE_PATH.DS.'api'.DS.'payment'.DS.$payment_info['payment_co
de'].DS.$payment_info['payment_code'].'.php';
require_once($inc_file);
$payment_api = new $payment_info['payment_code']
($payment_info,$order_pay_info);
//返回参数判断
$verify = $payment_api->return_verify();
if(!$verify) {
    //如果未通过验证, 抛出异常
}
//取得支付结果
$pay_result          = $payment_api->getPayResult($_GET);
if (!$pay_result) {
    //如果未通过验证, 抛出异常
}
$this-
>_updateProduct_buy($payment_info['payment_code'],$order_lis
t,$pay_amount);
//购买商品在线支付成功后, 更新数据表
private function
_updateProduct_buy($payment_code,$order_list,$pay_amount) {
    $url = SHOP_SITE_URL."/index.php?act=member_order";
    $out_trade_no = $_GET['out_trade_no'];
    if ($_GET['trade_no'] != '') {
        $trade_no = $_GET['trade_no'];
    }
    //事务订单支付表, 订单表, 预存款表与订单日志表
    try {
        $model_order = Model('order');
        $model_pd = Model('predeposit');
        //开始事务
        $model_order->beginTransaction();
        //更新支付单表支付状态
        $data = array();
        $data['api_pay_state'] = 1;
        $update = $model_order-
>editOrderPay($data,array('pay_sn'=>$out_trade_no));

```

```

        if (!$update) {
            throw new Exception('更新订单状态失败');
        }
        //更新订单表信息
        $data = array();
        $data['order_state']          = ORDER_STATE_PAY;
        $data['payment_time']         = TIMESTAMP;
        $data['payment_code']         = $payment_code;
        $update = $model_order->
>editOrder($data,array('pay_sn'=>$out_trade_no,'order_state'
=>ORDER_STATE_NEW));
        if (!$update) {
            throw new Exception('更新订单状态失败');
        }
        //如果有预存款支付的，彻底扣除冻结的预存款，并记录订单日志
        foreach($order_list as $order_info) {
            //
            $pd_amount = floatval($order_info['pd_amount']);
            if ($pd_amount > 0) {
                $data_pd = array();
                $data_pd['member_id'] =
$order_info['buyer_id'];
                $data_pd['member_name'] =
$order_info['buyer_name'];
                $data_pd['amount'] =
$order_info['pd_amount'];
                $data_pd['order_sn'] =
$order_info['order_sn'];
                $model_pd->
>changePd('order_comb_pay',$data_pd);
            }
            //记录订单日志
            $data = array();
            $data['order_id'] = $order_info['order_id'];
            $data['log_role'] = 'buyer';
            $data['log_msg'] = L('order_log_pay').' ( 支付平台
交易号 : '.$trade_no.' )';
            $data['log_orderstate'] = ORDER_STATE_PAY;
            $insert = $model_order->addOrderLog($data);
            if (!$insert) {
                throw new Exception('记录订单日志出现错误');
            }
        }
        //提交事务
        $model_order->commit();

```

```
        return array('success' => true);
    } catch (Exception $e) {
        //回滚事务
        $model_order->rollback();
        return array('error' => $e->getMessage());
    }
}
```



注意 通知程序中不可使用\$_SESSION，因

为服务器间异步通知时，会话是无效的，所需数据从数据表中查询即可。

3.8 退单模块

3.8.1 模块构成

退单模块主要由退款功能和退货功能组成，支付完成的订单可申请退款，已发货的订单可进行退货操作。涉及会员提出申请、商家审核和系统平台处理三方面的流程。

当买家按要求支付成功后，如果遇到商家不能及时发货或其他情况，可申请订单全部退款或单个商品的退款，商家在商家中心可看到申请记录，不同意时买家可再次申请；同意后由管理员确认给买家退款。

退货是在商家完成发货后，买家因长时间没收到货、商品在配送中出现损坏、收到与网站介绍不符的商品等原因，向商家申请退货并退款，商家与买家沟

通确认是否应该退货。若不同意，则结束；若同意退货，则买家要发货，商家收货后由系统管理员进行退款；商家也可以弃货，即不用买家发货，直接由管理员确认给买家退款。

3.8.2 设计思路

各个网站的买卖规则都不一样，在实际运营中，关于退款和退货方面也不会相同，部分流程可能差别还比较大，所以本节不会列出具体逻辑实现代码。掌握程序语法的人很多，能设计并实现一套出色系统就有点难了，但难度最大的是制定出符合实际要求的业务流程，不能太复杂让商家和买家不会使用，太简单了又会给运营者增加管理负担。

在有新退款或退货申请但又没完成订单（确认收货）时，为防止产生纠纷，要锁定订单状态。在待发货和待收货时有退款申请后立即锁定，商家不能进行发货操作和买家不能确认收货，当退款流程完成后可解除锁定。同一个订单有多种商品，又不全部退款

时，可分别对其他商品提出申请，要锁定多次，当处理完最后一个申请时解锁订单。

退款流程如图3-75所示。

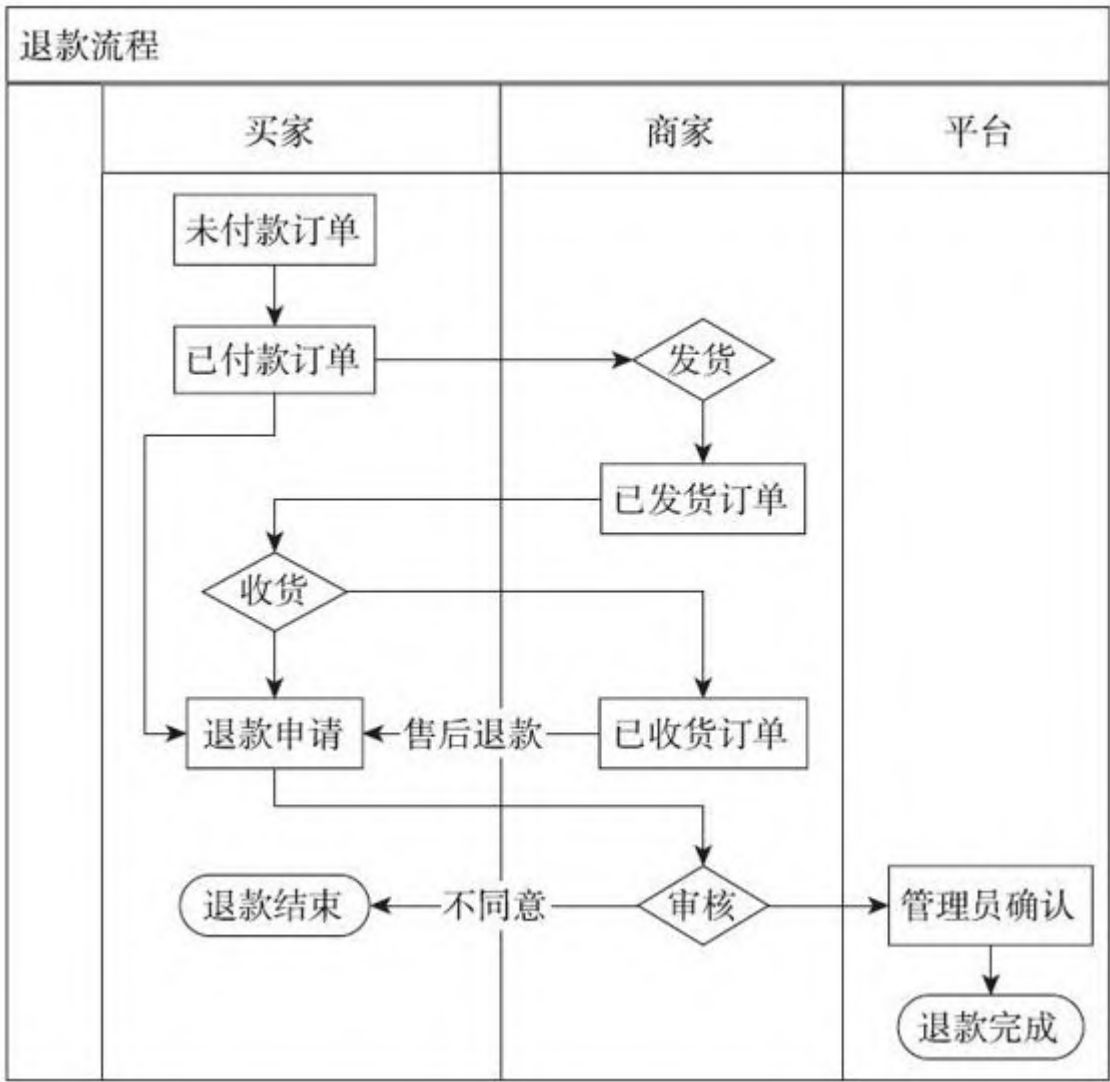


图3-75 退款流程

为什么一定要经过平台管理员处理？买家会员成功支付订单，钱是在平台设置的收款账号里，因为涉及订单结算周期，所以商家只是发出了商品并没有真正收到货款，退款以预存款的形式给买家，由管理员确认还能防止虚假交易的发生。每个成功的交易产生后平台都会收取一定比例的佣金，如果订单有退款产生在订单结算时要退还商家的佣金，平台管理员在处理退款时可以起到监督作用，及时发现交易异常的情况。

退货流程如图3-76所示，可以理解为：在退款流程基础上增加了买家发货和商家收货的步骤。退货只能是对订单内一种商品的操作，退款可以对商品也可以是整个订单。如果买家已付款而商家没有及时发货，此时的退款就是整个订单退款，这样更方便使用。当商家不同意退款或退货时，买家可再次申请，

也可以向平台投诉商家，提交相关证据，由系统管理员做仲裁。

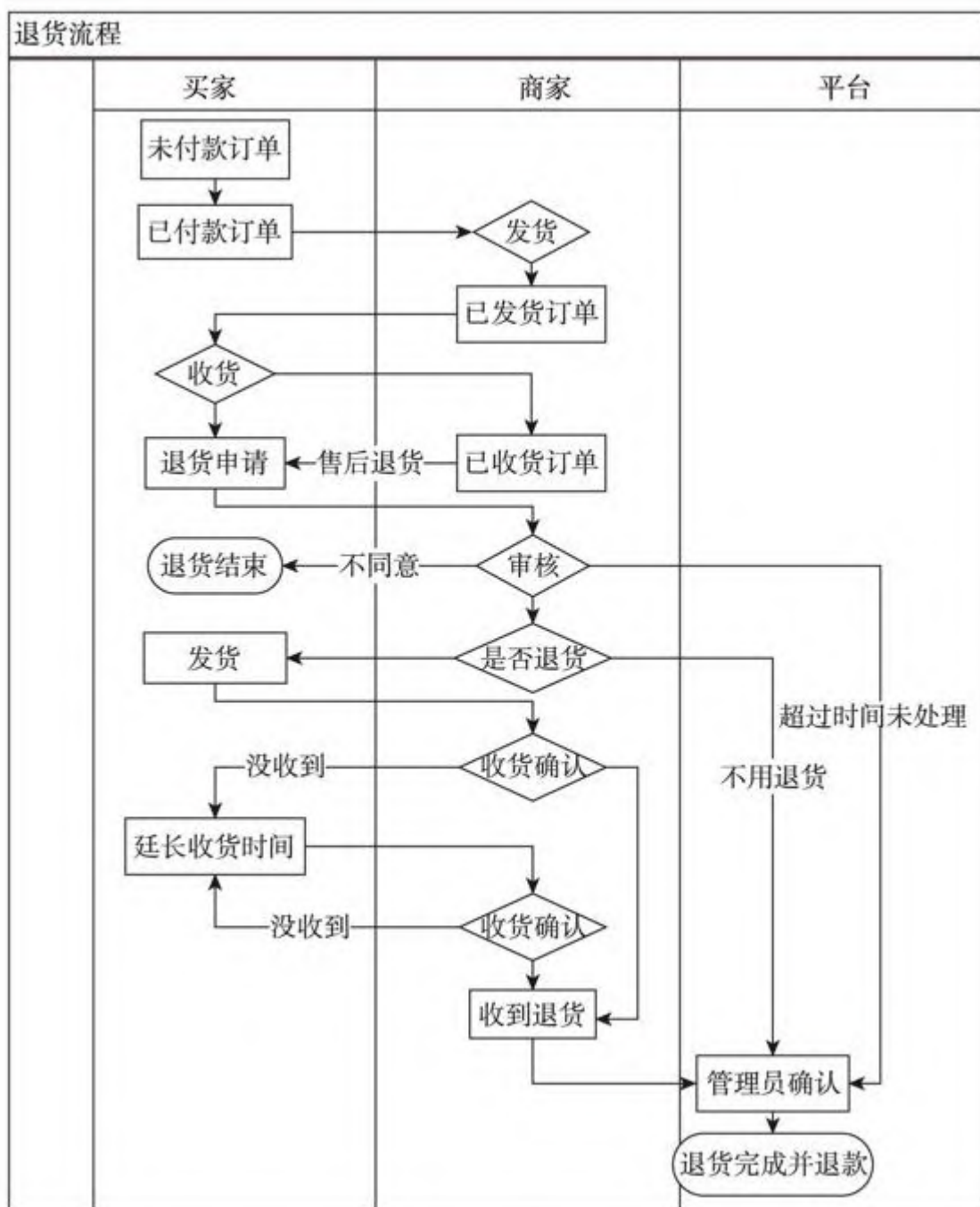


图3-76 退货流程

由于退货功能与退款功能类似，在设计表结构时，就可以使用同一个表，用一个字段标识是退款的还是退货的申请，这样既能减少程序的复杂度，还实现了部分代码的复用，便于系统维护和二次开发。

退款退货表数据关系如图3-77所示。

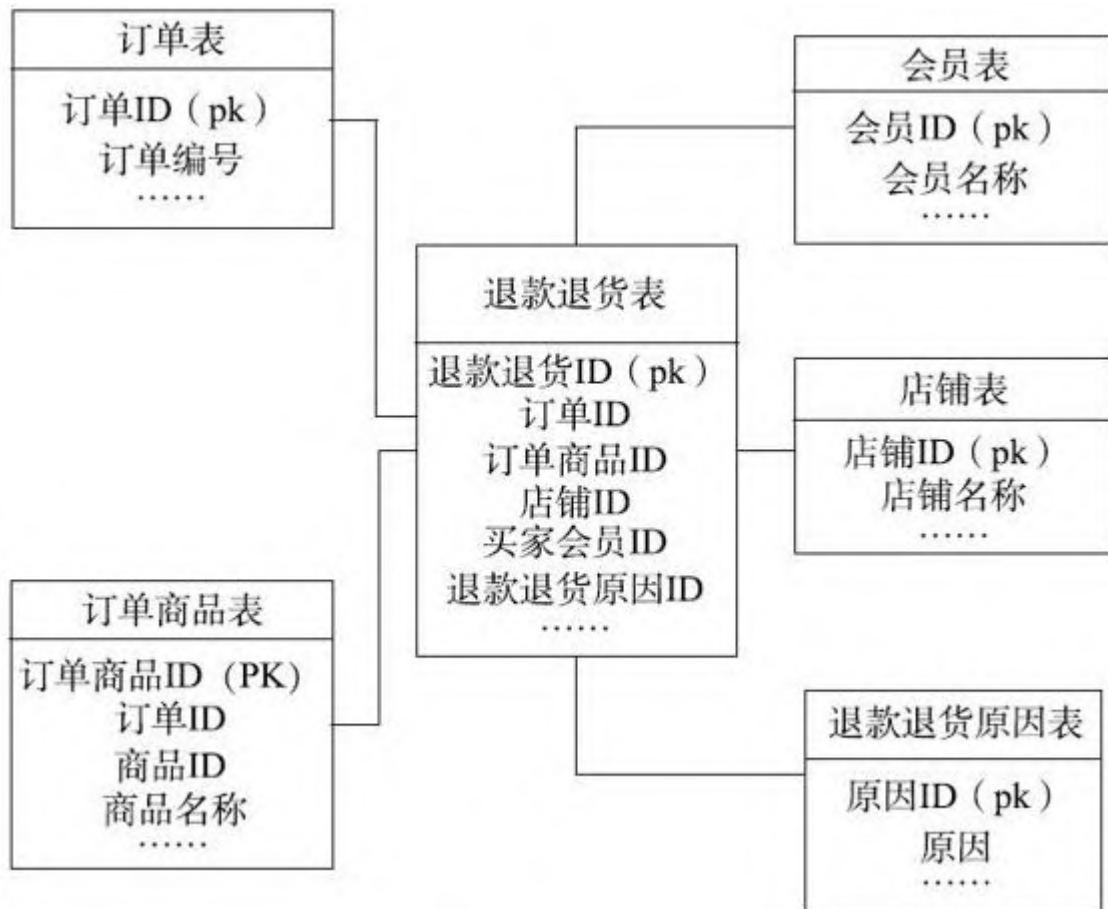


图3-77 退款退货表数据关系

退款退货原因是由系统管理员在后台录入，买家在提交申请时选择。一个订单里同一种商品只能有一个处理中的退款或退货申请，这样是为了不重复提交数据，减少商家的处理次数，防止出错。在商家同意退款或退货并完成后，将不能再申请了，比如某种商品同时买了多件，在申请退款时只是退了部分金额，就不能在系统中操作退款或退货了。在下节中会对此部分的验证进行详细说明。

本节的退款申请和退货申请类似，使用了一个数据表，在实际中这两个流程可能属于不同业务部门操作，逻辑就会改变很多，这时就要考虑到将数据分别保存到不同表中来实现，如果出现这种情况就尽可能地也将代码分离，做到相对独立互不干扰。由于系统平台定期对商家订单统一进行结算，就要监测交易相

关的数据，以免平台利益受损。当然想要程序来解决所有问题也不现实，有争议出现时，平台要根据实际情况人为线下调解处理，总结经验改善现有规则，在交易量大时隔段时间对程序升级一次很有必要，做到规则能优化，程序能调节。

3.8.3 代码实现

在本节以退款功能为例进行说明，分为买家申请、商家审核、平台处理三部分。退货和单商品退款使用一个入口，保存成功后在商家审核时分别列出。订单全部退款只有在待发货的订单状态下能操作，与单商品退款流程一样，只是申请页面的表单不同，因此本节的例子是单商品退款调用逻辑。

买家申请主要逻辑是根据订单的状态，调出符合规则的数据到页面，在页面中用JS做表单必填项的验证，提交到服务器端后用PHP对重要字段再进行数据合法性验证，都没问题时才能保存到数据库中。如果订单还没有完成要对订单锁定，即只有退款流程完成时才能继续后面的操作。

代码清单3-26是对商品和订单的验证：商品\$goods和订单\$order是从订单表中取出的基本数据，getRefundReturnList方法是根据订单编号和订单商品编号查询是否有已保存的有效退款退货记录；我们使用订单商品编号而不使用商品编号，是因为商品在同一单的时间段内可以参与不同的促销活动，在订单中为了进行区分是按不同的商品处理，价格也会不相同，如果用商品编号就可能会出现退款的金额以哪个成交价格为准的纠纷；getRefundState方法是用订单\$order的处理状态和时间来确认是否可申请退款或退货，防止提交表单前可以申请，提交后不可以而保存了无用数据，当然此种情况理论上很少发生，这样处理可以提高系统数据的安全性。

代码清单3-26 提交退款申请的商品验证

```
$condition = array();  
$condition['buyer_id'] = $_SESSION['member_id'];  
$condition['order_id'] = $order_id;  
$order = $model_refund->getRightOrderList($condition,
```

```

$goods_id);
$order_id = $order['order_id'];
//订单金额
$order_amount = $order['order_amount'];
//订单退款金额
$order_refund_amount = $order['refund_amount'];
$goods_list = $order['goods_list'];
$goods = $goods_list[0];
//商品实际成交价
$goods_pay_price = $goods['goods_pay_price'];
if ($order_amount < ($goods_pay_price +
$order_refund_amount)) {
    $goods_pay_price = $order_amount - $order_refund_amount;
    $goods['goods_pay_price'] = $goods_pay_price;
}
Tpl::output('goods',$goods);
$goods_id = $goods['rec_id'];
$condition = array();
$condition['buyer_id'] = $order['buyer_id'];
$condition['order_id'] = $order['order_id'];
$condition['order_goods_id'] = $goods_id;
$condition['seller_state'] = array('lt','3');
$refund_list = $model_refund-
>getRefundReturnList($condition);
$refund = array();
if (!empty($refund_list) && is_array($refund_list)) {
    $refund = $refund_list[0];
}
//根据订单状态判断是否可以退款退货
$refund_state = $model_refund->getRefundState($order);
//检查订单状态,防止页面刷新不及时造成数据错误
if ($refund['refund_id'] > 0 || $refund_state != 1) {

showDialog(Language::get('wrong_argument'),'index.php?
act=member_order&op=index','error');
}

```

上面的代码在填写申请表单前和提交后都会执行。还有要注意的验证就是，申请退款的金额不能超

过此商品的成交总价，也不能超过订单的可退金额，这些数字是程序根据订单和已有退款退货记录计算得出的。

对订单的退款验证是针对发货和收货状态。在已发货状态对非货到付款订单可以提出对订单商品的退款，而在收货后根据订单完成时间判断是否可进行退款，如代码清单3-27所示。

代码清单3-27 根据订单状态确定是否可退款

```
public function getRefundState($order) {
    $refund = '0';
    $order_state = $order['order_state'];
    $model_trade = Model('trade');
    $order_shipped = $model_trade->getOrderState('order_shipped');
    $order_completed = $model_trade->getOrderState('order_completed');
    switch ($order_state) {
        case $order_shipped:
            $payment_code = $order['payment_code'];
            //货到付款订单在没确认收货前不能退款退货
            if ($payment_code != 'offline') {
                $refund = '1';
            }
            break;
        case $order_completed:
            $order_refund = $model_trade->
```



```
>getMaxDay('order_refund');
    $delay_time =
$order['delay_time']+60*60*24*$order_refund;
    if ($delay_time > time()) {
        $refund = '1';
    }
    break;
default:
    $refund = '0';
    break;
}
return $refund;
}
```

商家在审核时对买家所提交的信息进行核对，决定是否退款。商家将数据提交到服务器后要进行状态确认，由程序检查商家是否已经审核过，只有商家处理状态为待审核时才能保存，防止多次更新数据。代码清单3-28中的\$refund['seller_state']的判断是商家是否已经审核过，未审核时是1，提交后这个状态就会改变，同意时更新为2，由平台管理员进行确认退款；不同意时更新为3并结束退款流程，如果订单还没有完成，要对订单进行解锁，以免影响其他数据。

代码清单3-28 商家审核退款申请

```

public function editOp() {
    $model_refund = Model('refund_return');
    $condition = array();
    $condition['store_id'] = $_SESSION['store_id'];
    $condition['refund_id'] =
intval($_GET['refund_id']);
    $refund_list = $model_refund-
>getRefundList($condition);
    $refund = $refund_list[0];
    if (chksubmit()) {
        //检查状态,防止页面刷新不及时造成数据错误
        if ($refund['seller_state'] != '1') {

showDialog(Language::get('wrong_argument'),'reload','error',
'CUR_DIALOG.close();');
        }
        $order_id = $refund['order_id'];
        $refund_array = array();
        $refund_array['seller_time'] = time();
        $refund_array['seller_state'] =
$_POST['seller_state'];
        $refund_array['seller_message'] =
$_POST['seller_message'];
        if ($refund_array['seller_state'] == '3') {
            $refund_array['refund_state'] = '3';
        } else {
            $refund_array['seller_state'] = '2';
            $refund_array['refund_state'] = '2';
        }
        $state = $model_refund-
>editRefundReturn($condition, $refund_array);
        if ($state) {
            if ($refund_array['seller_state'] ==
'3' && $refund['order_lock'] == '2') {
                //订单解锁
                $model_refund-
>editOrderUnlock($order_id);
            }
            $this->recordSellerLog('退款处理, 退款
编号: '.$refund['refund_sn']);

showDialog(Language::get('nc_common_save_succ'),'reload','su
cc', 'CUR_DIALOG.close();');
        } else {

```

```

showDialog(Language::get('nc_common_save_fail'),'reload',
'error','CUR_DIALOG.close();');
    }
}
Tpl::output('refund',$refund);
Tpl::showpage('store_refund_edit','null_layout');
}

```

平台处理的退款申请是商家已经同意的记录，为防止商家长时间不对买家的申请进行处理，同时也为了保护买家的利益，在超过期限后未审核的记录会自动按同意退款处理。如果是退货申请，则按同意并弃货处理，可以在买家和商家登录后触发程序执行一次。如代码清单3-29所示。

代码清单3-29 商家未及时处理退款和退货申请

```

public function editRefundConfirm($member_id=0, $store_id=0)
{
    Language::read('refund');
    //商家不处理退款退货申请时按同意并弃货处理
    $refund_confirm = $this->getMaxDay('refund_confirm');
    $day = time()-$refund_confirm*60*60*24;
    $condition = " seller_state=1 and add_time<".$day;
    $condition_sql = "";
    if ($member_id > 0) {
        $condition_sql = " buyer_id =
'".$member_id."' and ";
    }
}

```

```

        if ($store_id > 0) {
            $condition_sql = " store_id =
'".$store_id."' and ";
        }
        $condition_sql = $condition_sql.$condition;
        $refund_array = array();
        $refund_array['refund_state'] = '2';
        $refund_array['seller_state'] = '2';
        $refund_array['return_type'] = '1';
        $refund_array['seller_time'] = time();
        $refund_array['seller_message'] =
Language::get('order_max_day').$refund_confirm.Language::get
('order_day_refund');
        $this->table('refund_return')-
>where($condition_sql)->update($refund_array);
        //商家不进行收货时按弃货处理
        $return_confirm = $this-
>getMaxDay('return_confirm');
        $day = time()-$return_confirm*60*60*24;
        $condition = " seller_state=2 and goods_state=2 and
return_type=2 and delay_time<".$day;
        $condition_sql = "";
        if ($member_id > 0) {
            $condition_sql = " buyer_id =
'".$member_id."' and ";
        }
        if ($store_id > 0) {
            $condition_sql = " store_id =
'".$store_id."' and ";
        }
        $condition_sql = $condition_sql.$condition;
        $refund_array = array();
        $refund_array['refund_state'] = '2';
        $refund_array['return_type'] = '1';
        $refund_array['seller_message'] =
Language::get('order_max_day').$return_confirm.'天未处理收货,
按弃货处理';
        $this->table('refund_return')-
>where($condition_sql)->update($refund_array);
    }

```

因为管理员确认退款时要向买家的预存款账号中加钱，所以这个操作只能执行一次，防止多次更新造成平台损失。代码清单3-30中\$refund['refund_state']的判断类似商家审核的逻辑，待管理员确认的是2，提交保存后更新为3即已完成状态，只有待确认的记录才可调用执行editRefundReturn的增加预存款程序，如果订单还没有完成，要对订单进行解锁。

代码清单3-30 平台处理退款

```
public function editOp() {
    $model_refund = Model('refund_return');
    $condition = array();
    $condition['refund_id'] =
intval($_GET['refund_id']);
    $refund_list = $model_refund-
>getRefundList($condition);
    $refund = $refund_list[0];
    if (chksubmit()) {
        //检查状态,防止页面刷新不及时造成数据错误
        if ($refund['refund_state'] != '2') {

            showMessage(Language::get('nc_common_save_fail'));
        }
        $order_id = $refund['order_id'];
        $refund_array = array();
        $refund_array['admin_time'] = time();
    }
}
```

```
        $refund_array['refund_state'] = '3';
        $refund_array['admin_message'] =
$_POST['admin_message'];
        $state = $model_refund-
>editOrderRefund($refund);
        if ($state) {
            $model_refund-
>editRefundReturn($condition, $refund_array);
            $this->log('退款确认, 退款编
号'.$refund['refund_sn']);

showMessage(Language::get('nc_common_save_succ'),'index.php?
act=refund&op=refund_manage');
        } else {

showMessage(Language::get('nc_common_save_fail'));
        }
    }
    Tpl::output('refund',$refund);
    Tpl::showpage('refund.edit');
}
```

3.8.4 功能体验

买家从订单列表中可以进入图3-78的退款申请页面。默认是退款申请，可以在服务类型中选择退货退款，填写退货数量；中上部是一些提示文字，帮助买家更好地使用此功能；右侧是商品和订单的信息，从订单相关表中查询取出数据；中下部是需要填写的表单，可以传三张证据图片，以便商家和平台管理员进行核对。

申请退款服务类型：仅退款

操作提示：

1. 若您未收到货，或已收到货但与商家达成一致不退货仅退款时，请选择“仅退款”选项。

2. 若为商品问题，或者不想要了且与商家达成一致退货，请选择“退货退款”选项，退货后请保留物流底单。

3. 若提出申请后，商家拒绝退款或退货，可再次提交申请或选择“商品投诉”，请求商城客服人员介入。

4. 成功完成退款退货：经过商城审核后，会向退款金额以“原币种”的形式退还到您的余额账户中。

买家申请退款

商家处理退款申请

平台审核，退款完成

请填写退款申请

* 退款原因：

请选择退款原因

* 需要退款金额：

189.00元

(最多189.00元)

退款金额不能超过可退金额。

* 退款说明：

上传凭证：

选择文件

选择文件

选择文件

退款/退货申请

正昌 2014春季新款女 绣花针织衫 开

¥109.00 * 1 (数量)

运费：

《免运费》

订单总额：

¥567.00

订单编号：

500C003000C003601

更多

物流单号：

120090350E910

韵达快递

商家：

官方旗舰店

更多

图3-78 退款申请表单

买家提交表单保存后可以在退款申请列表页看到相关信息，可以按申请时间段、订单编号、退款编号、商品名称查询，如图3-79所示。

see more please visit: <https://homeofbook.com>

退款申请		退货申请			
申请时间		前	后	订单编号	搜索
商品		退款金额(元)		审核状态	平台确认
操作					
退款编号: 202101140403114542		申请时间: 2014-04-03 11:45:42		官方店铺	
 正品 2014春夏新款 女 绣花针织衫 开衫外套 修身 绿色 订单编号: 6000000000000001		199.00		待审核	无
				查看	
退款编号: 20110114031170030		申请时间: 2014-03-11 17:00:30		官方店铺	
订单商品全部退款		95.00		同意	已完成
订单编号: 6000000000000001				查看	
退款编号: 434101140310110704		申请时间: 2014-03-10 11:07:04		官方店铺	
 2014新款 打底衫 纯色 毛衣 长袖 圆领 针织衫 蓝色 订单编号: 6000000000000001		179.00		同意	已完成
				查看	
退款编号: 431101140303152003		申请时间: 2014-03-03 15:20:09		官方店铺	
 2014新款 打底衫 纯色 毛衣 长袖 圆领 针织衫 蓝色 订单编号: 6000000000000001		179.00		不同意	无
				查看	

图3-79 退款申请列表

商家可以在商家管理中心首页的交易提示看到退款及退货数量，点击相应链接可显示本店铺客户所提交的申请，如果是在确认收货前提交的会在“售前退款”中，已收货完成的订单商品退款在“售后退款”。在数据较多时，分开售前和售后会更方便管理，所用的程序和模板显示是同一套代码，只是在参数上有区别，如图3-80所示。

售前退款

售后退款

申请时间

处理状态

全部

订单编号

搜索

商品/订单号/退款号	退款金额	买家会员名	申请时间	处理状态	平台确认	操作
 正品 2014春秋新款女 棉花针织衫 开衫外套薄款初绿色 订单编号: 8000000000000001 退款编号: 202101140403114542	¥189.00	shopnc2	2014-04-03 11:45:42	待审核	无	处理
订单商品全部退款 订单编号: 8000000000000001 退款编号: 201101140311170830	¥95.00	shopnc2	2014-03-11 17:09:30	同意	已完成	查看

图3-80 商家的退款记录

商家处理时可以根据买家提供的信息或沟通情况决定是否退款，提交后会在买家的退款查看页面中显示处理结果和备注信息，图3-81中显示的就是商家已经同意的退款记录。

退款服务

提示:

1. 若售出申请后,商家拒绝退款或退货,可再次提交申请或选择“商品投诉”,请求商城客服人员介入。
2. 成功完成退款退货;经过商城审核后,会将退款金额以“储值款”的形式返还到您的余额账户中。

买家申请退款

商家处理退款申请

平台审核,退款完成

我的退款申请

退款编号: 202101140403114542

退款原因: 商品破损、有污渍

退款金额: ¥189.00

退款说明: 与网站介绍不符,差别很大。

凭证上传:

商家退款处理

审核状态: 同意

退款/退货申请

商品 2014春装新款女 棉花针织衫 开...
¥189.00 * 1 (数量)

运费: (免运费)

订单总额: ¥567.00

订单编号: 50000000000000601 更多 >

物流单号: 1200906506910 韵达快递

商家: 晋力店铺 更多 >

图3-81 会员查看退款记录

商家已经同意的退款记录会在系统后台退款管理的“待处理”中出现,管理员点“确认”进入审核页,在提交表单后这条申请就不会再在此列表中看到,填写的“备注信息”会在买家和商家的退款查看页中显示。

see more please visit: <https://homeofbook.com>

如图3-83所示，在系统后台退款管理的“所有记录”中可以查询买家提交的申请，包括商家未处理、不同意退款和退款完成的记录，可以方便管理员及时掌握平台退款信息，与商家共同提升服务水平。



订单编号	退款编号	店铺名	商品名称	买家会员名	申请时间	商家审核时间	退款金额	操作
6030000000000001	202101140403114542	官方旗舰店	正品 2014新款韩版女 韩版斜肩袖 韩版外套深蓝色 绿色	shopn2	2014-04-03 11:45:42	2014-04-03 13:57:14	109.00	确认

图3-82 系统后台的待处理退款



订单编号	退款编号	店铺名	商品名称	买家会员名	申请时间	退款金额	商家审核	平台确认	操作
0000000000000001	202101140403114542	官方旗舰店	正品 2014新款韩版女 韩版斜肩袖 韩版外套深蓝色 绿色	shopn2	2014-04-03 11:45:42	109.00	同意	待处理	查看
0000000000000001	201101140311170830	官方旗舰店	订单商品退货退款	shopn2	2014-03-11 17:08:30	95.00	同意	已处理	查看
0000000000000001	43410114031110704	官方旗舰店	2014新款韩版女 韩版斜肩袖 韩版外套深蓝色 绿色	shopn2	2014-03-11 10:7:04	179.00	同意	已处理	查看
0000000000000001	491101140309150909	官方旗舰店	2014新款韩版女 韩版斜肩袖 韩版外套深蓝色 绿色	shopn2	2014-03-09 15:29:09	179.00	不同意	无	查看

图3-83 系统后台的所有退款列表

如图3-84所示，系统后台的“退款退货原因”就是由基本的增、删、改、查程序组成的，所添加的数

据在买家申请退款时从下拉列表中选择，这没有什么技术难度，因此就没对此部分进行说明。

退款管理		
所有记录		
新增退款原因		
操作提示		
系统初始化的原因不能删除		
排序显示规则为排序小的在前，排序的在前		
操作	原因	操作
123	不能按时发货	编辑
123	认为是假货	编辑
123	质量问题不符	编辑
123	商品破损、有污渍	编辑
123	数量不符不喜欢	编辑
尾页 上一页 1 下一页 末页		

图3-84 系统后台的退款退货原因

3.8.5 开发技巧

在实际开发中要先定好规则、理清思路，对于逻辑中有不明白的地方，及时沟通解决——“磨刀不误砍柴工”，要是程序写到一半才发现问题，再调整就不容易了，有可能将前面的逻辑推翻重来，这样浪费时间和精力算是小事，影响到其他人员的开发进度就严重了。

在编写程序时，尽量做到代码的复用，这样有问题可以及时发现和修改；在可能影响到规则的逻辑中一定要进行服务器端的数据验证，一旦出现漏洞被利用的情况就可能造成大的损失，如敏感信息泄露、预存款账号异常等；在较复杂的流程中还要注意调用权限的问题，什么情况下什么人能进行什么操作很重要，不能跨时间跨步骤更新数据。

3.9 结算模块

结算是B2B2C平台的重要组成部分，这里所说的结算是大平台和商家之间的账单结算，普通买家并不参与。买家购买商品直接付款到大平台，商家负责发货，大平台和商家定期结算（一般为月结）。月结时，系统会在每月1号自动出账单，账单包括上个月每个商家的有效交易统计信息及详细信息；出账后系统会等待商家对账单进行确认，如果账单计算无误，商家确认后会进入系统审核环节，系统审核后账单会提交到财务部门进行付款操作；付款完成后，需在后台录入付款相关信息，账单结算完成。结算流程图如图3-85所示。

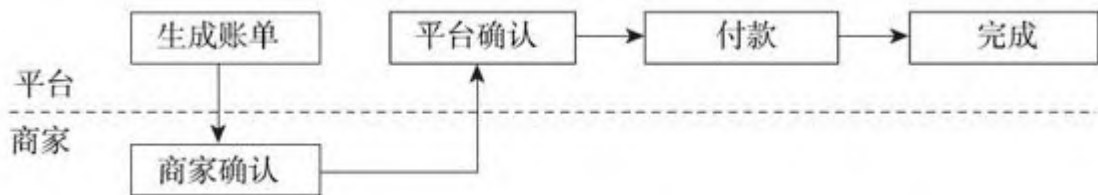


图3-85 结算流程图

3.9.1 设计思路

1. 数据表设计

结算数据表主要涉及两个：账单表和账单汇总表。

账单表记录的是每个月每个商家的结算信息，包括日期、订单总金额、总运费、退单总金额、佣金总金额、退还佣金金额、店铺费用、应结金额以及结算状态等字段，主键即为账单编号，由年份（4位）+月份（2位）+商家ID组成。

账单汇总表是对每个月所有商家结算信息的统计汇总，它反应的是每月平台各项结算参数的汇总数据，年份（4位）+月份（2位）组成主键。

2. 结算流程设计

结算模块主要流程包括出账、对账和付款三个部分。本节重点围绕这三部分介绍设计思路。

所谓出账就是系统自动计算出本月的结算账目。有以下几个方面需要注意。

（1）执行时机

一般情况下B2B2C平台结算周期为1个月，每月1号为出账日，服务器会在每月1号触发结算程序，计算上个月1号00：00：00至上个月最后一天23：59：59秒之间有效订单。系统的出账时间为结算单的生成时间。结算单除系统自动触发执行外，结算程序还应支持手动执行，如果服务器因故障未能出账，那么可以人工触发生成结算单，结算日期为人工触发执行的日期。

（2）结算对象

系统结算的对象是上月发生的交易完成的订单或退单。订单中买家签收货物后即可视为交易完成。退单中退款或退货成功后即可视为退单完成。

（3）计算公式

B2B2C商城默认是以即时到账的方式将支付款打到了平台账户中，平台需在扣除相关费用后，将剩余款项定时支付给商家。结算时主要涉及以下几个元素。

1) 订单金额。订单金额是上个月内买家实际支付金额（包含运费）。

2) 佣金金额。这里指的是上个月内平台应当收取的佣金总金额。佣金是平台收取商家交易费用的一种方式，结算时，平台会从订单总金额中扣除应该收取的佣金。每种商品都有自己的佣金抽取比例，佣金计算与运费无关，与订单总金额无关，由每种商品的单

价、数量、佣金比例来决定，如果购买时有优惠活动（如满减），则系统会将优惠的金额分摊到每种商品上面。

佣金 = 商品实际售价 × 购买数量 - 优惠分摊金额
(如果有优惠活动)

3) 退单金额。退单金额是指交易成功后，如果发生了退款（退货），并且经过商家（平台）审核成功退款，平台结算时需将这部分金额从本期结算金额中扣除，只有在上个月内成功退款的订单才会计入本身结算金额中。

4) 退还佣金。只有发生退款时才会退还佣金，平台结算时需将这部分金额追加到本期应结金额中。

5) 店铺促销费用。这里是指店铺运营中发起各种促销活动（团购、满即送、限时折扣等）时向平台支

付的费用，在平台结算时需将这些费用从本期应结金额中扣除。当然促销费用也可以不放入结算中处理，在商家发起促销时直接以即时到账方式向平台支付所需费用，这种也可以采用。

从上面所述我们可以得出结算的计算公式为：

$$\text{平台应付金额} = \text{订单金额} - \text{佣金金额} - \text{退单金额} + \text{退还佣金} - \text{店铺促销费用}$$

与结算还有一个相关的元素就是店铺开店保证金，当平台应结算金额为负数时（实际很少会出现这种情况），平台有可能会采用扣除保证金的方式来处理。

对账是结算中必不可少的环节，系统出账后，首先商家会进行核对，系统需要为商家提供详细的出账信息，并提供表格数据下载，核对无误后确认并提交

到平台审核。平台再次进行核对无误后，确认并提交，财务部门开始支付环节。

付款完成后，进入系统后台，输入相关付款信息并提交，完成结算流程。

3.9.2 代码实现

结算代码实现主要包括两大部分，第一部分是生成账单的代码，由系统任务自动触发执行，第二部分是生成结算后的平台与商家之间结算流程代码。第一部分涉及生成结算的思路，第二部分代码实现较容易，本节只介绍第一部分代码。

生成结算代码主要由MVC中的M（模型）和C（控制器）来完成，它是后台运行，不需要输出展示，所以没有V（视图）的参与。

1. 模型代码

模型主要完成的是对结算所需要的方法进行封装的工作，如代码清单3-31所示。

代码清单3-31 结算模型核心代码

```
//定义结算单状态
//默认初始状态
define('BILL_STATE_CREATE',1);
//店铺已确认
define('BILL_STATE_STORE_COFIRM',2);
//平台已审核
define('BILL_STATE_SYSTEM_CHECK',3);
//结算完成
define('BILL_STATE_SUCCESS',4);
class billModel extends Model {
    //取得平台月结算单
    public function getOrderStatisList($condition = array(),
    $fields = '*', $pagesize = null, $order = '', $limit = null)
    {
        return $this->table('order_statis')-
    >where($condition)->field($fields)->order($order)-
    >page($pagesize)->limit($limit)->select();
    }
    //取得平台月结算单条信息
    public function getOrderStatisInfo($condition = array(),
    $fields = '*', $order = null) {
        return $this->table('order_statis')-
    >where($condition)->field($fields)->order($order)->find();
    }
    //取得店铺月结算单列表
    public function getOrderBillList($condition = array(),
    $fields = '*', $pagesize = null, $order = '', $limit = null)
    {
        return $this->table('order_bill')-
    >where($condition)->field($fields)->order($order)-
    >limit($limit)->page($pagesize)->select();
    }
    //取得店铺月结算单条信息
    public function getOrderBillInfo($condition = array(),
    $fields = '*') {
        return $this->table('order_bill')-
    >where($condition)->field($fields)->find();
    }
    //取得订单数量
    public function getOrderBillCount($condition) {
        return $this->table('order_bill')-
    >where($condition)->count();
    }
    //增加结算汇总表记录
```



```
public function addOrderStatis($data) {  
    return $this->table('order_statis')->  
>insertAll($data);  
}  
//新增结算单表  
public function addOrderBill($data) {  
    return $this->table('order_bill')->insert($data);  
}  
//编辑结算单  
public function editOrderBill($data, $condition =  
array()) {  
    return $this->table('order_bill')->  
>where($condition)->update($data);  
}  
}
```

2. 控制器代码

控制器完成的任务就是生成结算单，系统会分批次生成账单信息和汇总信息。系统生成账单的处理过程如图3-86所示。

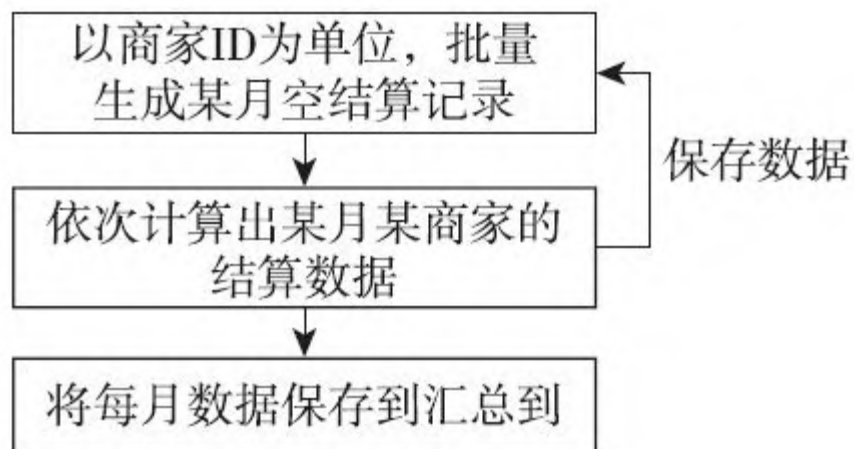


图3-86 购物车主要操作与流程

生成账单的核心代码如代码清单3-32所示。

代码清单3-32 生成账单核心代码

```
//生成上月账单
public function create_order_statistic() {
    $model_order = Model('order');
    $model_bill = Model('bill');
    //计算开始结算的时间起点，即上次最后一条结算末尾日期
    $order_statistic_max_info = $model_bill->getOrderStatisticInfo(array(), 'os_end_date', 'os_month desc');
    if (!$order_statistic_max_info) {
        //如果从未生成过结算单，则以系统第一个订单产品日期作为结算开始日期
        $order_min_info = $model_order->getOrderInfo(array(), array(), 'min(add_time) as add_time');
        $start_unixtime =
            is_numeric($order_min_info['add_time']) ?
            $order_min_info['add_time'] : TIMESTAMP;
    } else {
        $start_unixtime =
            $order_statistic_max_info['os_end_date'];
    }
    $data = array();
    $i = 1;
    $start_unixtime = strtotime(date('Y-m-01 00:00:00',
        $start_unixtime));
    while (($time = strtotime('-' . $i . ' month')) >=
        $start_unixtime) {
        if (date('Ym', $start_unixtime) == date('Ym', $time))
        {
            //检查库是否存在（如果两个月份相等，则不再生成结算单）
            $order_statistic = $model_bill->getOrderStatisticInfo(array('os_month' => date('Ym', $start_unixtime)));
            if ($order_statistic) {
```

```

        break;
    }
}
//账单结算的开始时间（上月该月第一天0时unix时间戳）
$first_day_unixtime = strtotime(date('Y-m-01
00:00:00', $time));
//账单结算的终止时间（该月最后一天最后一秒时unix时间戳）
$last_day_unixtime = strtotime(date('Y-m-01
23:59:59', $time)." +1 month -1 day");
$key = count($data);
$os_month = date('Ym',$first_day_unixtime);
$data[$key]['os_month'] = $os_month;
$data[$key]['os_year'] =
date('Y',$first_day_unixtime);
$data[$key]['os_start_date'] = $first_day_unixtime;
$data[$key]['os_end_date'] = $last_day_unixtime;
//生成所有商家某月的出账单
$this->create_order_bill($data[$key]);
//将某月所有商家出账单统计汇总，保存入结算汇总表
$fields = 'sum(ob_order_totals) as
ob_order_totals,sum(ob_shipping_totals) as
ob_shipping_totals,
sum(ob_order_return_totals) as
ob_order_return_totals,
sum(ob_commis_totals) as
ob_commis_totals,sum(ob_commis_return_totals) as
ob_commis_return_totals,
sum(ob_store_cost_totals) as
ob_store_cost_totals,sum(ob_result_totals) as
ob_result_totals';
$order_bill_info = $model_bill->
getOrderBillInfo(array('os_month'=>$os_month),$fields);
$data[$key]['os_order_totals'] =
floatval($order_bill_info['ob_order_totals']);
$data[$key]['os_shipping_totals'] =
floatval($order_bill_info['ob_shipping_totals']);
$data[$key]['os_order_return_totals'] =
floatval($order_bill_info['ob_order_return_totals']);
$data[$key]['os_commis_totals'] =
floatval($order_bill_info['ob_commis_totals']);
$data[$key]['os_commis_return_totals'] =
floatval($order_bill_info['ob_commis_return_totals']);
$data[$key]['os_store_cost_totals'] =
floatval($order_bill_info['ob_store_cost_totals']);
$data[$key]['os_result_totals'] =

```

```

floatval($order_bill_info['ob_result_totals']);
    $i++;
}
krsort($data);
$model_bill->addOrderStatis($data);
}
//生成所有商家某月的出账单
private function _create_order_bill($data){
    $model_order = Model('order');
    $model_bill = Model('bill');
    $model_store = Model('store');
    //取商家表数量
    $store_count = $model_store->getStoreCount(array());
    //分批生成该月份的商家空结算表，每批生成300个
    $insert = false;
    for ($i=0;$i<=$store_count;$i=$i+300){
        $store_list = $model_store-
>getStoreList(array(),null,'','store_id','{$i}',300");
        if ($store_list){
            $data_bill = array();
            foreach($store_list as $store_info){
                $data_bill['ob_no'] =
                $data['os_month'].$store_info['store_id'];
                $data_bill['ob_start_date'] =
                $data['os_start_date'];
                $data_bill['ob_end_date'] =
                $data['os_end_date'];
                $data_bill['os_month'] = $data['os_month'];
                $data_bill['ob_state'] = 0;
                $data_bill['ob_store_id'] =
                $store_info['store_id'];
                if (!$model_bill-
>getOrderBillInfo(array('ob_no'=>$data_bill['ob_no']))) {
                    //插入以月份为单位的空账单
                    $insert = $model_bill-
>addOrderBill($data_bill);
                    //对已生成空账单进行销量、退单、佣金统计
                    if ($insert){
                        $update = $this-
>_calc_order_bill($data_bill);
                        if (!$update){
                            showMessage('生成账单['.$data['os_month'].']失
                            败','','html','error');
                        }
                    }else{

```

```

        showMessage('生成账单['.$data['os_month'].']失败', '', 'html', 'error');
    }
}
}
}
}
}
//计算某月内，某商家的销量，退单量，佣金等数据
private function _calc_order_bill($data_bill){
    $model_order = Model('order');
    $model_bill = Model('bill');
    $model_store = Model('store');
    //定义查询条件，查询订单数据
    $order_condition = array();
    $order_condition['order_state'] = ORDER_STATE_SUCCESS;
    $order_condition['store_id'] =
    $data_bill['ob_store_id'];
    $order_condition['finnshed_time'] =
    array('time',array($data_bill['ob_start_date'],$data_bill['o
    b_end_date']));
    $update = array();
    //订单总金额
    $fields = 'sum(order_amount) as
    order_amount,sum(shipping_fee) as
    shipping_amount,store_name';
    $order_info = $model_order->
    getOrderInfo($order_condition,array(),$fields);
    $update['ob_order_totals'] =
    floatval($order_info['order_amount']);
    //总运费
    $update['ob_shipping_totals'] =
    floatval($order_info['shipping_amount']);
    //商家名
    $store_info = $model_store->
    getStoreInfoByID($data_bill['ob_store_id']);
    $update['ob_store_name'] = $store_info['store_name'];
    //佣金总金额
    $order_info = $model_order->
    getOrderInfo($order_condition,array(),'count(DISTINCT
    order_id) as count');
    $order_count = $order_info['count'];
    $commis_rate_totals_array = array();
    //分批计算佣金，最后取总和
    for ($i = 0; $i <= $order_count; $i = $i + 300){

```

```

        $order_list = $model_order-
>getOrderList($order_condition, '', 'order_id', '', "{$_i},300");
        $order_id_array = array();
        foreach ($order_list as $order_info) {
            $order_id_array[] = $order_info['order_id'];
        }
        if (!empty($order_id_array)){
            $order_goods_condition = array();
            $order_goods_condition['order_id'] =
array('in',$order_id_array);
            $field = 'SUM(goods_pay_price*commis_rate/100)
as commis_amount';
            $order_goods_info = $model_order-
>getOrderGoodsInfo($order_goods_condition,$field);
            $commis_rate_totals_array[] =
$order_goods_info['commis_amount'];
        }else{
            $commis_rate_totals_array[] = 0;
        }
    }
    $update['ob_commis_totals'] =
floatval(array_sum($commis_rate_totals_array));
    //退款总金额
    $model_refund = Model('refund_return');
    $refund_condition = array();
    $refund_condition['seller_state'] = 2;
    $refund_condition['store_id'] =
$data_bill['ob_store_id'];
    $refund_condition['goods_id'] = array('gt',0);
    $refund_condition['admin_time'] =
array(array('egt',$data_bill['ob_start_date']),array('elt',$
data_bill['ob_end_date']),'and');
    $refund_info = $model_refund-
>getRefundReturnInfo($refund_condition,'sum(refund_amount)
as amount');
    $update['ob_order_return_totals'] =
floatval($refund_info['amount']);
    //退还佣金总金额
    $refund = $model_refund-
>getRefundReturnInfo($refund_condition,'sum(refund_amount*co
mmis_rate/100) as amount');
    if ($refund) {
        $update['ob_commis_return_totals'] =
floatval($refund['amount']);
    } else {

```

```

        $update['ob_commis_return_totals'] = 0;
    }
    //店铺活动总费用
    $model_store_cost = Model('store_cost');
    $cost_condition = array();
    $cost_condition['cost_store_id'] =
$data_bill['ob_store_id'];
    $cost_condition['cost_state'] = 0;
    $cost_condition['cost_time'] =
array(array('egt',$data_bill['ob_start_date']),array('elt',$
data_bill['ob_end_date']),'and');
    $cost_info = $model_store_cost-
>getStoreCostInfo($cost_condition,'sum(cost_price) as
cost_amount');
    $update['ob_store_cost_totals'] =
floatval($cost_info['cost_amount']);
    //本期应结
    $update['ob_result_totals'] = $update['ob_order_totals']
- $update['ob_order_return_totals'] -
    $update['ob_commis_totals'] +
$update['ob_commis_return_totals']-
$update['ob_store_cost_totals'];
    //账单生成时间
    $update['ob_create_date'] = TIMESTAMP;
    //账单默认状态
    $update['ob_state'] = BILL_STATE_CREATE;
    //将以上数据更新到账单表
    $result = $model_bill-
>editOrderBill($update,array('ob_no'=>$data_bill['ob_no']));
    return $result;
}

```

3.9.3 功能体验

本节我们演示结算单的处理流程，演示之前先展示一下平台和商家结算界面。

1. 平台结算管理界面展示

进入后台后进入结算管理，可以看到每个月份平台的结算账目汇总信息，由于演示站点是在2013年12月搭建，目前为5月份，所以系统会生成2013-12至2014-04的账单，如图3-87所示。



账单(月)	开始日期	结束日期	订单金额	退款	收款金额	佣金金额	佣金金额	店铺费用	本期应结	操作
2014-04	2014-04-01	2014-04-30	0.00	0.00	0.00	0.00	0.00	125.00	-125.00	查看
2014-03	2014-03-01	2014-03-31	51508.00	0.00	1513.00	0.00	0.00	20.00	49995.00	查看
2014-02	2014-02-01	2014-02-28	109401.00	0.00	10940.10	0.00	0.00	200.00	98260.90	查看
2014-01	2014-01-01	2014-01-31	79182801.20	213.00	278821.78	40227181.00	1477.23	5255.00	28664010.65	查看
2013-12	2013-12-01	2013-12-31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	查看

图3-87 结算数据汇总信息

点击“商家账单列表”按钮，可以看到目前所有商家的账单列表信息，如图3-88所示。

账单管理 / 账单管理 商家账单列表												
账单处理流程为：系统出账 -> 商家确认 -> 平台审核 -> 微信支付(网银收款) 4个环节，其中平台审核和微信支付(网银收款)合并，请密切关注												
与微信支付												
账单编号	开始日期	结束日期	订单金额	运费	佣金金额	退款金额	退还佣金	店铺费用	本期应付	出账日期	账单状态	店铺
20140422	2014-04-01	2014-04-30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2014-05-15	已出账	my旗舰店 id:22
20140421	2014-04-01	2014-04-30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2014-05-15	已出账	双雄的门店 id:21
20140420	2014-04-01	2014-04-30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2014-05-15	已出账	双雄的门店 id:20
20140419	2014-04-01	2014-04-30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2014-05-15	已出账	双雄的门店 id:19
20140418	2014-04-01	2014-04-30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2014-05-15	已出账	双雄的门店 id:18

图3-88 账单列表

点击任意一个结算单，查看详细信息，查看账单2014012的基本信息，如图3-89所示。

结算管理 / 结算管理 201401期 商家账单列表 账单明细
结算单号：2014012
起止日期：2014-01-01 至 2014-01-31
出账日期：2014-02-10
平台应付金额：404.35 = 1192.00 (订单金额) - 59.60 (佣金金额) - 259.00 (退单金额) + 12.95 (退还佣金) - 482.00 (店铺促销费用)
结算状态：已出账

图3-89 账单详细信息

2014012表示编号为2的商家在2014年1月的结算单。

除账单基本信息外，本页还可以查询订单、退单、店铺费用的详细列表，并支持这些数据的导出操作。图3-90展示了账单2014012的订单列表。

订单编号	订单金额	运费	佣金	下单日期	成交日期	买家	操作
6060090000023001	219.00	0.00	10.95	2014-01-20	2014-01-20	tjshopnc	查看
6060090000008001	188.00	0.00	9.40	2014-01-10	2014-01-10	tjshopnc	查看
6060090000009001	259.00	0.00	12.95	2014-01-10	2014-01-10	tjshopnc	查看
6060090000007101	526.00	0.00	26.30	2014-01-07	2014-01-07	apple2	查看

图3-90 订单列表

可以导出订单数据如图3-91所示。

	A	B	C	D	E	F	G	H	I	J	K
1	订单编号	订单金额	运费	佣金	下单日期	成交日期	商家	商家编号	买家	买家编号	商品
2	NC60000000	219	0	10.95	2014/1/20	2014/1/20	妮妮服装店	2	tjshopnc	68	春装新款半
3	NC60000000	188	0	9.4	2014/1/10	2014/1/10	妮妮服装店	2	tjshopnc	68	冬季新款女
4	NC60000000	259	0	12.95	2014/1/10	2014/1/10	妮妮服装店	2	tjshopnc	68	乐町2014春
5	NC60000000	526	0	26.3	2014/1/7	2014/1/7	妮妮服装店	2	apple2	32	冬装新款女
6	合计	1192	0	59.6							
7											

图3-91 导出订单数据

账单2014012的退单列表如图3-92所示。

订单类型 退单列表 退款时间 - 导出数据明细							
退单编号	订单编号	原订单金额	退还金额	类型	退款日期	买家	操作
722132140110692064	6000000000007901	259.00	12.95	退货	2014-01-10	qzhopsc	查看

图3-92 退单列表

点击“查看”按钮可以查看订单、退单的详细情况。账单2014012的店铺促销费用列表如图3-93所示。

订单类型	店铺费用	退款时间			Q
店铺	促销名称	促销费用	申请日期		
妮妮服装店	购买团购	300.00	2014-01-07		
妮妮服装店	购买限时折扣	22.00	2014-01-07		
妮妮服装店	购买满减送	40.00	2014-01-07		
妮妮服装店	购买充值优惠	40.00	2014-01-07		
妮妮服装店	购买满赠展位	40.00	2014-01-07		
妮妮服装店	购买代金券套餐	40.00	2014-01-07		

图3-93 店铺促销费用列表

2. 商家结算界面展示

登录“妮妮服装店”的商家后台结算管理界面，可以看到该商家的结算列表，如图3-94所示。

结算管理					
		账单状态	请选择...	结算单号	搜索
结算单号	起止时间	本期应收	结算状态	付款日期	操作
2014012	2014-01-01 - 2014-01-31	404.35	已出账		查看
2014022	2014-02-01 - 2014-02-28	0.00	已出账		查看
2014032	2014-03-01 - 2014-03-31	0.00	已出账		查看
2014042	2014-04-01 - 2014-04-30	0.00	已出账		查看
首页 上一页 1 下一页 末页					

图3-94 结算列表

点击“查看”按钮可以看到详细的结算信息，如图3-95所示。

本期结算

本期结算无误, 我要确认

结算单号: 2014012 2014-01-01 至 2014-01-31

出账时间: 2014-02-10

本期应收: 404.35 = 1192.00 (订单金额) - 59.60 (佣金金额) - 259.00 (退单金额) + 12.95 (搬迁佣金) - 482.00 (促销费用)

结算状态: 已出账

订单列表

退款订单

促销费用

订单编号

成交时间

搜索

导出

订单编号	下单时间	成交时间	订单金额	运费	佣金金额	操作
60000000000020201	2014-01-20	2014-01-20	219.00	0.00	10.95	查看
60000000000008201	2014-01-10	2014-01-10	188.00	0.00	9.40	查看
60000000000007901	2014-01-10	2014-01-10	259.00	0.00	12.95	查看
60000000000002101	2014-01-07	2014-01-07	526.00	0.00	26.30	查看

图3-95 结算详细信息

这里显示的内容同大平台（这里是指商城系统的运营方）后台相似，同样也支持导出结算信息。

3. 结算流程

前两节对平台后台和商家后台的结算作了基本的展示和介绍，本节开始演示结算流程。系统出账后，默认进入等待商家审核阶段，商家确认系统结算数据无误后点击“确认”按钮，之后系统进入平台审核确认环节，进入系统后台可以发现2014012号结算单状态已经变更为“商家已确认”，如图3-96所示。

2014013	2014-01-01	2014-01-31	7596.00	9.00	75.96	2997.00	29.97	81.00	4472.21	2014-02-10	已结账	微信支付系统 提现	提现
2014012	2014-01-01	2014-01-31	1192.00	0.00	59.60	290.00	12.95	682.00	804.15	2014-02-10	商家已确认	微信支付系统 提现	提现

图3-96 结算状态已经变更

如果结算数据无误，点击“审核”按钮，系统进入付款环节，如图3-97所示。



图3-97 审核结算单

待平台向商家付款完成后，需要进入平台后台更新一下结算单的状态，点击“付款完成”输入相关付款信息，如图3-98所示。

结算管理 | 结算管理 账单付款

账单编号 :

2014012

付款日期 :

2014-05-03

付款备注 :

支付宝支付
支付宝交易号 : 12555550002154855

提交

图3-98 账单付款

至此，编号为2014012的结算单处理完成。

3.10 统计模块

我周围有很多从事电子商务运营的朋友，工作都非常辛苦。电子商务运营不应该是个体力活儿，但是2012年的运营工作看起来就是最拼体力的工作，因为大家比拼的是广告投放数量和种类，比拼在线时长，比拼商品数量，比拼活动数量。通过了解我很遗憾地发现，他们中有很多人忽略了数据在电子商务运营中的作用。未来的互联网世界一定是属于数据的，而电子商务运营也一定会转向以数据为基础的运营。当客户进入网站页面，他的一举一动都可以变成数据记载下来，从而会在网站中产生大量的真实数据，通过网站中这些大量的真实数据，可以有效地估计出访客的兴趣、对各种商品的不同反应、分享客户的各种行为等运营需要了解的信息，使运营有据可依，不再是凭经验盲目运作，而是有的放矢。如何能让我们的电商

人分析数据不再那么辛苦，是我写以下内容的初衷，希望能对从事电子商务相关工作的同行有些帮助。

做好数据工作，在电子商务中除了需要数据分析和数据挖掘的知识外，还需要对电子商务行业有深入的了解，两者缺一不可。如果只有数据挖掘知识，很可能做出的数据模型在理论上是正确的，但是不符合电子商务的实际情况；反之，如果只是对电子商务的运作有深入了解，而对数据挖掘不了解，那么可能会对数据模型提出不切实际的过分要求，或者做出一个在功能上有缺失的模型。

一家公司想要成功，必须让数据分析介入运营，应秉持以下几点。

- 以数据为基础，智能地制定运营决策。
- 以数据为目标，有效执行运营计划。

- 以数据为依据，优化商务过程。

下面我们看一个真实的数据应用案例，更加形象地领会数据分析在运营中的重要性。以下是阿迪达斯的“黄金罗盘”案例。

厦门育泰贸易有限公司与阿迪达斯合作已有13年，旗下拥有100多家阿迪达斯门店。公司负责人叶总说，“2008年之后，库存问题确实很严重，但我们通过合作解决问题，生意再次回到了正轨。”看着同行大多仍身陷库存泥潭，他庆幸自己选对了合作伙伴。

在最初降价、打折等清库存的“应急措施”结束后，外部环境、消费者调研和门店销售数据的收集、分析，成为了将阿迪达斯和其经销商们引向正轨的“黄金罗盘”。

现在，叶总每天都会收集门店的销售数据，并将它们上传至阿迪达斯。收到数据后，阿迪达斯对数据做整合、分析，再用于指导经销商卖货。研究这些数据，让阿迪达斯和经销商们可以更准确地了解当地消费者对商品颜色、款式、功能的偏好，同时知道什么价位的产品更容易被接受。

阿迪达斯产品丰富，过去面对展厅里各式各样的产品，经销商很容易按个人偏好下订单。现在，阿迪达斯会用数据说话，帮助经销商选择最适合的产品。首先，从宏观上看，一、二线城市的消费者对品牌和时尚更为敏感，可以重点投放采用前沿科技的产品、运动经典系列的服装以及设计师合作产品系列。在低线城市，消费者更关注产品的价值与功能，诸如纯棉制品这样高性价比的产品，在这些市场会更受欢迎。其次，阿迪达斯会参照经销商的终端数据，给予更具体的产品订购建议。比如，阿迪达斯可能会告诉某低

线市场的经销商，在其辖区普通跑步鞋比添加了减震设备的跑鞋更好卖；至于颜色，比起红色，当地消费者更偏爱蓝色。

通过这种订货方式，阿迪达斯得到了经销商们的认可。叶总说：“我们一起商定卖哪些产品、什么产品又会热卖。这样，我们将来就不会再遇到库存问题。”

挖掘大数据，让阿迪达斯有了许多有趣的发现。同在中国南部，那里部分城市受香港风尚影响非常大；而另一些地方，消费者更愿意追随韩国潮流。同为一线城市，北京和上海消费趋势不同，气候是主要的原因。还有，高线城市消费者的消费品位和习惯更为成熟，当地消费者需要不同的服装以应对不同场合的需要，上班、吃饭、喝咖啡、去夜店，需要不同风格的多套衣服，但在低线城市，一位女性往往只要有

应对上班、休闲、宴请的三种不同风格的服饰就可以。两相对比，高线城市，显然为阿迪达斯提供了更多细分市场的选择。

实际上，对大数据的运用，也顺应了阿迪达斯大中华区战略转型的需要。

库存危机后，阿迪达斯从“批发型”公司转为“零售驱动型”公司，它从过去只关注把产品卖给经销商，变成了将产品卖到终端消费者手中的有力推动者。而数据收集分析，恰恰能让其更好地帮助经销商提高售罄率。

“我们与经销商伙伴展开了更加紧密的合作，以统计到更为确切可靠的终端消费数据，有效帮助我们重新定义了产品供给组合，从而使我们在适当的时机，将符合消费者口味的产品投放到相应的区域市场。一方面降低了他们的库存，另一方面增加了单店

销售率。卖得更多，售罄率更高，也意味着利润更高。”阿迪达斯大中华区董事总经理对大数据的应用成果颇为满意。

3.10.1 模块构成

电子商务相关的数据分析模块包含九部分，如图3-99所示。

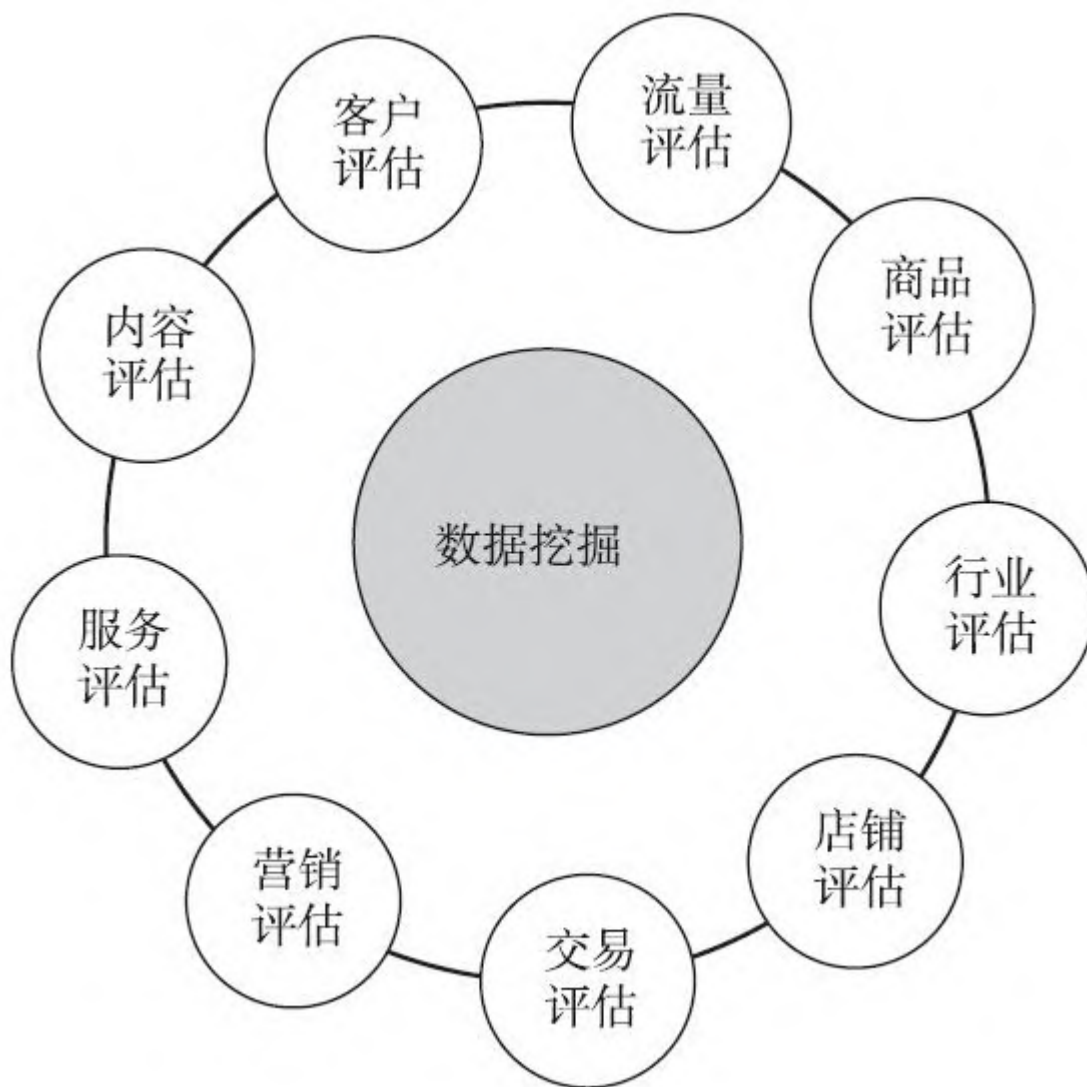


图3-99 电子商务数据分析主要的九部分

当然根据网站运营的需要，可能还需要对模块进行细分，在此就不再赘述了。

下面介绍一下图3-99九个部分中一些数据的含义及计算方法。

（1）浏览量（PV）

浏览量为流量评估统计中的数据。浏览量是网站所有页面被点击的总次数。一个IP点击N次，按N次统计。从技术层面讲，指浏览器加载网页的次数总和。

（2）访客数（UV: Unique Visitor）

访客数为流量评估统计中的数据。如果你统计访客数的时间限制为一天，那么一天内同一台计算机的访问只能算一次。访客数是统计访问你网站的人数，

而不是访问次数。从技术层面上讲，用Cookie来确定绝对唯一访问者，而不是IP。办公室内N台电脑访问，虽然同一个IP，统计时，按N次算。同一个电脑访问，即使IP换了，但计算机没换，按一次算。以上统计，时间可以选择。比如，统计近30天内本站的绝对唯一访问者，不管某用户在这30天内访问了多少次，只要他用同一个计算机（保存着Cookie）就只算一次。

（3）转化率

转化率为流量评估统计中的数据。它指产生实际消费的客户和来到网站的总客户数量的比值，是将流量转化为实际的销售额的一种衡量方式。成交转化率=成交客户数/总访客数。

（4）平均访问深度

平均访问深度为流量评估统计中的数据。它是指用户在一次浏览你的网站的过程中浏览了你网站的页数，就是PV和UV的比值，这个比值越大，用户体验度越好，网站的黏性也越高。

那么如何提高自己网站的访问深度呢？具体可以用以下几点。

- 第一点：网站合理的排版和布局。如果你的网站的结构简洁明了，用户看到后感觉很舒服，自然愿意多在你网站停留一段时间了。

- 第二点：网站的内容。总的来说，就算你的版面如何的让人喜欢，但是用户却无法在你的网站上获得有用的信息，那么用户也不会上面浪费时间。

- 第三点：合理的导航和适当的内部链接锚文本。导航是很重要的，它能让用户从任何页面都能够

很轻易地回到其他想去的页面。

（5）网站人均停留时间

网站人均停留时间为流量评估统计中的数据。它是指每位访问者平均在网站上停留了多少时间，平均网站停留时间=网站总停留时间/会话的数量（访问次）。

（6）页面跳失率

页面跳失率为流量评估统计中的数据。它是指访客到达该目标页面，到达后没有继续访问该网站其他页面既离开，我们就称之为一次Bounce！也就是跳失了。（跳失率=浏览了该页面就离开的访问次数/该页面的全部访问次数。）数值越小代表网站可能越受欢迎或者网站的产品更符合客户需求，客户更愿意访问更多的页面，反之说明越不受欢迎。

(7) 下单商品数

下单商品数为商品评估统计中的数据。它是指订单中商品总的售出数量。

(8) 商品下单量

商品下单量为商品评估统计中的数据。它是指包含该商品的所有订单数量。

(9) 客单价 (per customer transaction)

客单价为交易评估统计中的数据。它是指一定时期内网站每一个会员平均购买商品的金额，即平均交易金额。客单价=销售总额/顾客总数，或者是客单价=销售总金额/成交总笔数。采取后者算法的比较普遍。客单价的本质是：在一定时期内，每位顾客消费的平均价格，离开了“一定时期”这个范围，客单价这个指标是没有任何意义的。

（10）重复购买率

重复购买率为客户评估统计中的数据。它是指消费者对该产品或者服务的重复购买次数，重复购买率越高，则反映出消费者的忠诚度就越高，反之则越低。重复购买率有两种计算方法：一种算法是所有购买过产品的顾客，以每个人为独立单位重复购买产品的次数，比如有10个客户购买了产品，5个产生了重复购买，则重复购买率为50%；另一种算法是，单位时间内，重复购买的总次数占比，比如10个客户购买了产品，中间有3个人有了二次购买，这3人中的1个人又有了三次购买，则重复购买次数为4次，重复购买率为40%。推荐企业采取第一种算法。重复购买率又可以细化为二次购买率、三次购买率……。二次购买率即购买了二次商品的客户占总购买客户的比例。三次购买率即购买了三次商品的客户占总购买客户的比例。

3.10.2 设计思路

做数据分析，要先从商业角度出发，看到底需要完成什么样的商业目标，再回头制定数据分析过程。将数据加工处理成信息，最后转化为运营的依据。

1. 数据本身的设计原则

数据本身的设计原则应该遵循以下几点。

(1) 要有总体的概念

网站的分析数据有时候只需要展示一些总体的数据，直观明了地显示出来，对于只需要了解大致情况的人员，在很多时候只需要查看这些常规的总体数据就可以了。

例如ShopNC B2B2C电商平台中就设置了商城总体数据展示页面，如图3-100所示。

- 销售总体信息：包括下单金额、下单量、平均客单价。

- 商品总体信息：包括下单商品数、商品平均价格、新增商品数、商品总数量、7日内商品销售TOP30。

- 客户总体信息：包括会员总数、新增会员数、下单会员数。

- 店铺总体信息：包括店铺总数、新增店铺数、7日内店铺销售TOP30。



图3-100 ShopNC B2B2C商城总体数据展示实例

当然根据运营的具体需要还有其他的总体数据，比如网站总浏览量、访问量、点击率等，在此就不一一赘述了。

（2）统计要细致到一点

一种统计也许只是展现了某种现象的一点，有时得出某个运营结论需要由很多统计数据得到的各个点

组成，所以做统计要从全局考虑，然后落实到每个分点去实现。要细致到某个会员、某个时间点、某个地区等。

（3）定期数据分析

做好电子商务网站的运营，需要做各种分析和报表、定期展示、对比网站数据和运营数据。对于大部分CEO来说，周期的销量增长量可能是他们最为关心的数据分析。

· **同比：** 同比就是今年第n月与去年第n月比。同比发展速度主要是为了消除季节变动的影响，用以说明本期发展水平与去年同期发展水平对比而达到的相对发展速度。如本期2月比去年2月，本期6月比去年6月等。其计算公式为：同比发展速度=本期数/去年同期数 $\times 100\%$ ；同比增长速度=（本期数-去年同期数）/去年同期数 $\times 100\%$ 。

· **环比：** 环比就是报告期与前一统计时间段比较。环比的发展速度是报告期水平与前一时期水平之比，表明现象逐期的发展速度。例如今年7月份与今年6月份相比较。其计算公式为：环比发展速度 = $(\text{本期数} / \text{上期数}) \times 100\%$ ；环比增长率 = $(\text{本期数} - \text{上期数}) / \text{上期数} \times 100\%$ ，反映本期比上期增长了多少；环比发展速度，一般是指报告期水平与前一时期水平之比，表明现象逐期的发展速度。

(4) 重视日常运营数据

我们不需要等到所有的数据准备好了再做数据分析，可以从有限的数据开始分析。客户每次的点击、浏览等操作，都是他无形之中对于网站的反馈。如果能够结合网站上的多个日志分析，综合起来就能够得到更加有价值的数据。在电子商务网站上大量的数据分析都是需要以日志文件作为基础的。所以在进行数

据分析时，一定要重视日常运营中的数据，抽丝剥茧提取有价值的信息。例如客户平均访问页面数、平均停留时间、跳出率、商品被收藏次数等，都可以作为分析页面信息和商品信息是否受欢迎的数据依据。

（5）数据应具有实时性

传统的数据分析技术往往分析起来要进行批处理，要半天、一天甚至好几天的时间才能出结果。现在的互联网用户打开一个网站，往往只浏览几分钟时间，如果几分钟之内不能够吸引住用户则用户就跑掉了，所以半天、一天甚至几天的时间出结果，这样的速度对于互联网就是灾难了。因此数据分析应该体现实时性，达到需要的“快”，及时的反应网站的运营现状。

2. 业务层次的设计原则

业务层次的设计原则应该遵循以下几点。

（1）层层深入剖析业务

电子商务网站的运营需要多种岗位的配合才能完成。而每个岗位关心的数据又是不一样的。比如CEO关心的是总体的销售情况，采购关心的是哪一款产品好卖便于及时补货。所以数据分析应该采取层层深入的设计方法，以便于管理层从宏观的角度了解运营情况，而底层的员工具体到各个细分环节了解运营情况。从宏观到细节层层深入的剖析业务。如图3-101所示，电子商务网站数据分析按照行业分析→店铺分析→品牌分析→商品、会员的分析，从大到小层层深入进行数据剖析，再细致还可进行服务、售后、营销等模块的分析，在此就不赘述了。

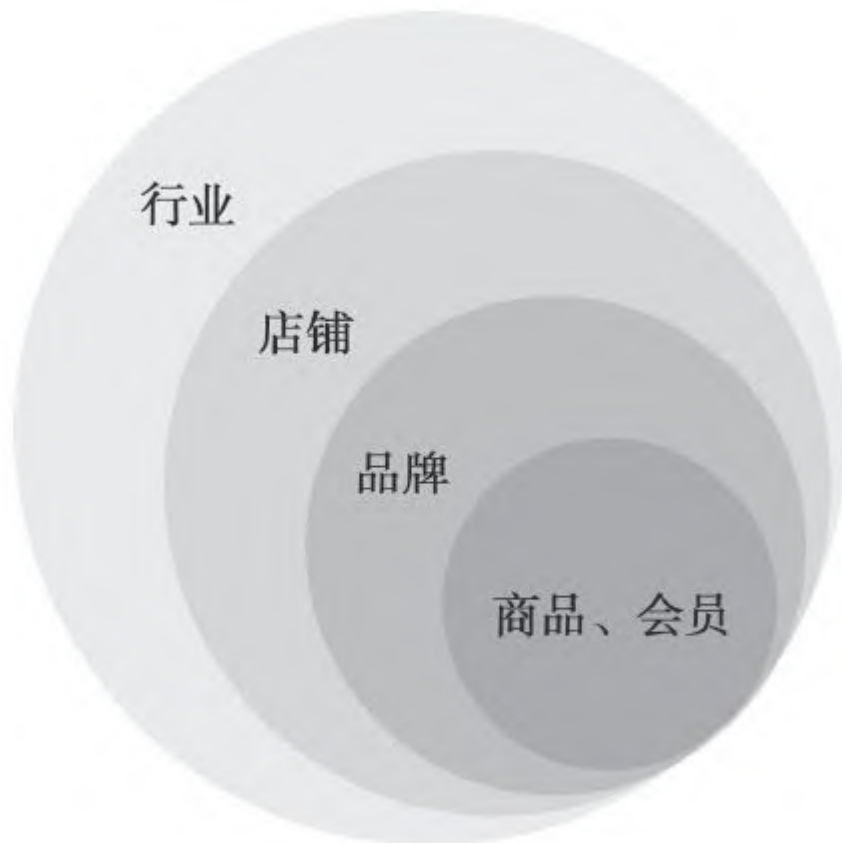


图3-101 层层深入进行数据分析

(2) 直观易懂

字不如表，表不如图，最好的数据分析就是看图说话。当我们从不同渠道获取数据的时候，数据可能是杂乱无章的，通常是不容易看出规律的。通过制作图形和报表，往往可以找出隐含在数据中的规律性。

· **表示变化趋势用折线图：** 以折线的上升或下降来表示统计数量的增减变化的统计图，叫作折线统计图。折线变化幅度越大，数量关系变化越大。折线统计图不仅可以表示数量的多少，而且可以反映同一事物在不同时间里的发展变化的情况，如图3-102所示。

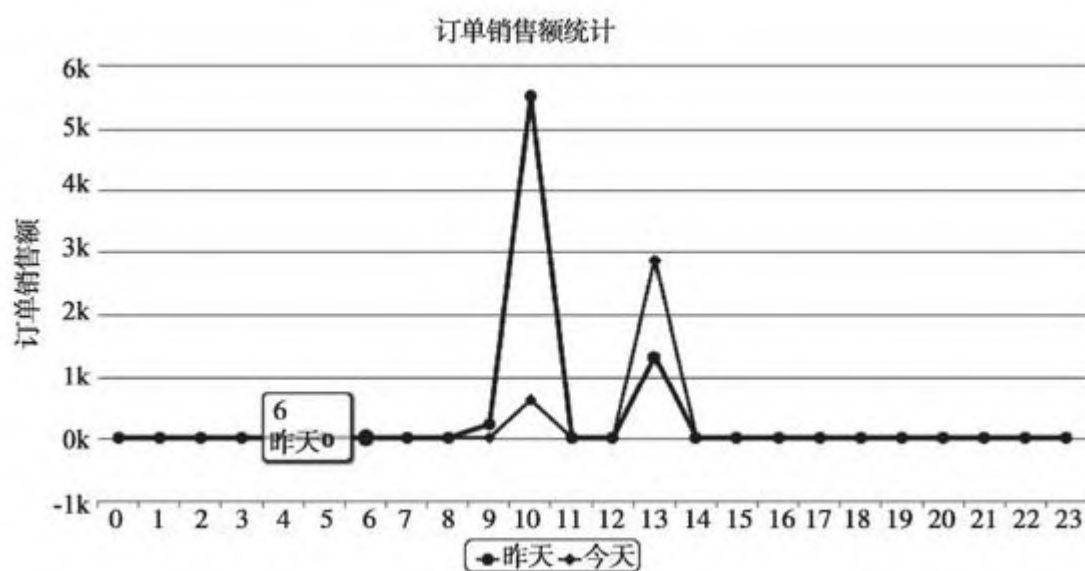


图3-102 折线图示意图

图3-102展示的是昨天和今天在不同时间段内的销售额变化情况，横坐标是小时，纵坐标是订单销售

额。可以非常直观地看出时间段内销售额的变化幅度和对比昨天和今天某个时间段的销售额情况。

· **简单比较用柱状图：**柱状图，也称条图、长条图、条状图，是一种以长方形的长度为变量的表达图形的统计报告图，由一系列高度不等的纵向条纹表示数据分布的情况，可以从直观上看出在同一维度上多个数据的变化和对比，如图3-103所示。

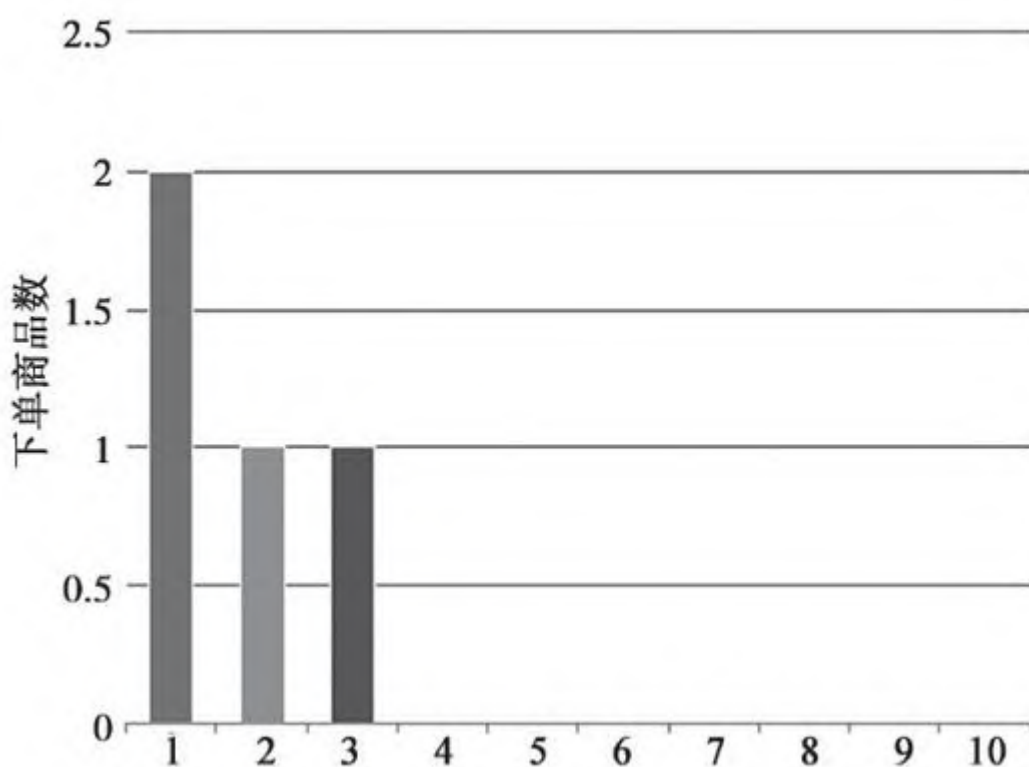


图3-103 柱状图示意图

图3-103为商品销量排名柱状图，横坐标是次序，纵坐标是商品售出数。可以很容易看出商品的销量及其差额。

- **用漏斗图展示流程转化率：** 漏斗图是对业务流程最直观的一种表现形式，适用于网站中某些关键流程的转化率分析，不仅显示了在进入流程到实现目标的最终转化率，同时还可以展示整个流程中每一步的转化率，从而说明问题的所在，客户从进入网站到转化为忠实客户所经历阶段的漏斗图如图3-104所示。

- **表示比例最好用饼图或者环形图：** 饼形图是用扇形的大小来表示数值大小的统计图。它主要用于表示总体中各组成部分所占的比例。环形图与饼图类似，但它们之间也有区别。环形图中间有一个“空洞”，总体或样本中的每一部分数据由环中的一段表

示。饼图只能显示一个总体和样本各部分所占的比例，而环形图则可以同时绘制多个总体或样本的数据系列，每一个总体或样本的数据系列为一个环。因此环形图可显示多个总体或样本各部分所占的相应比例，从而有利于进行比较研究，如图3-105所示。

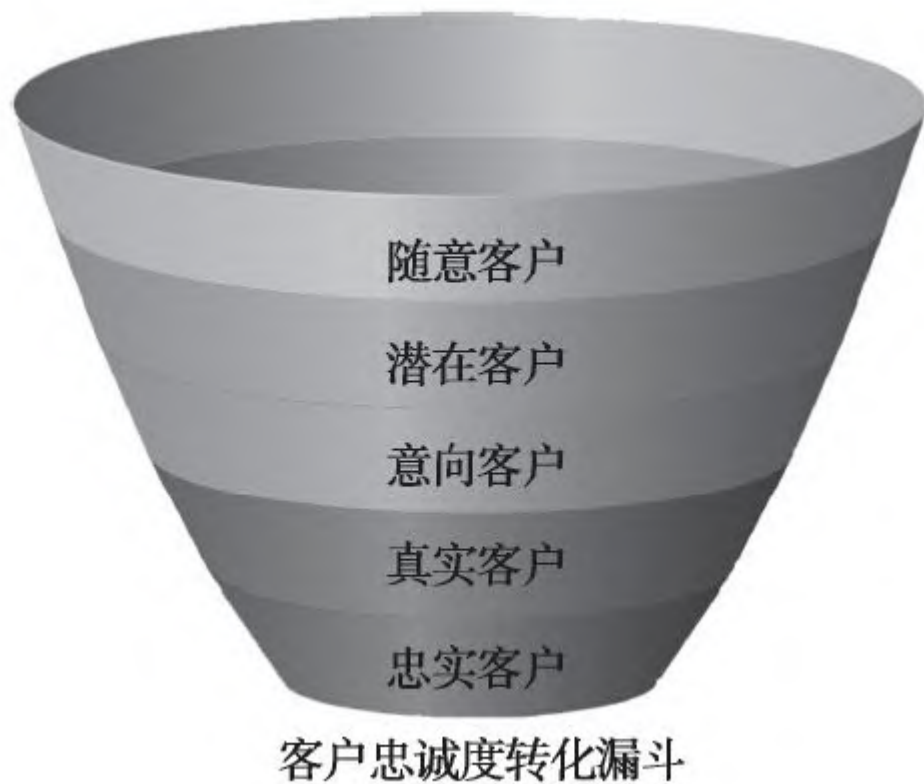


图3-104 漏斗图示意图

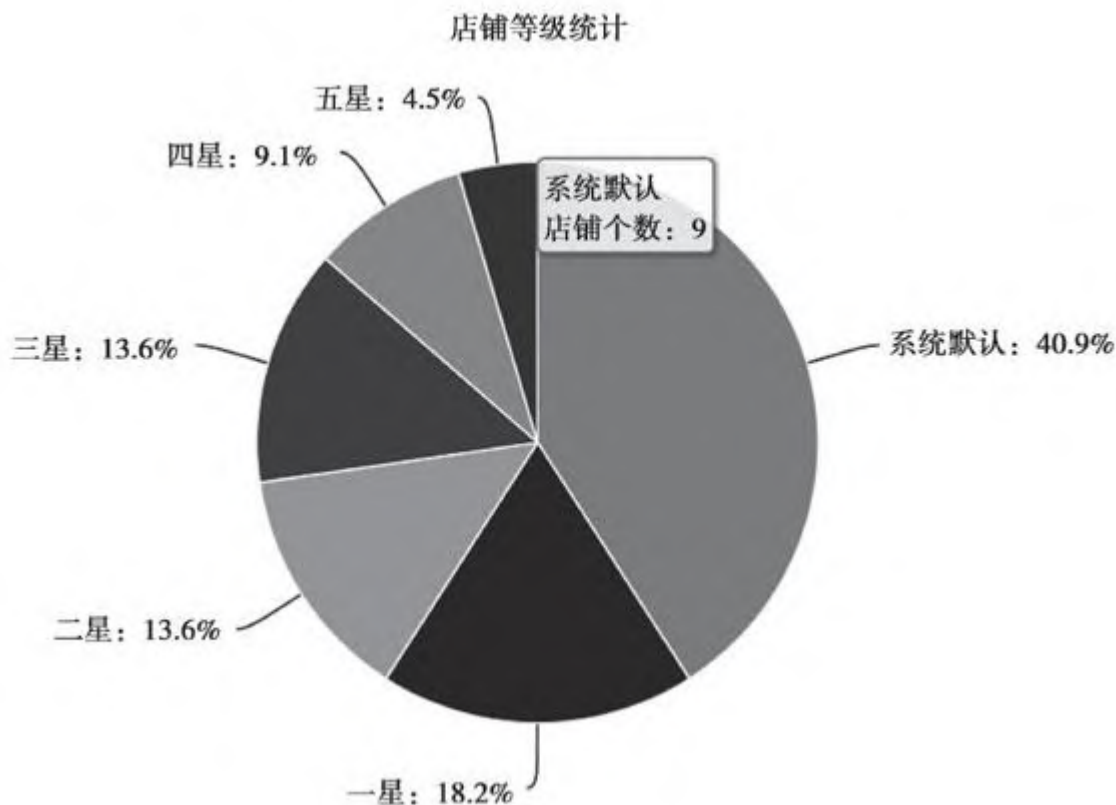


图3-105 饼图示意图

图3-105为各店铺等级下的店铺占有所有店铺比例的饼图。通过扇形的大小很容易了解哪个等级的店铺比较多。

· 表示在区域中的分布情况用统计地图：统计地图是用图表、图形表示整个制图区域统计数据的地图。分级统计地图按行政区划或经济区划，以不同的

颜色或疏密不等的晕线形象地反映现象相对指标的同一性和差异性，以分析它们在自然区域中的分布特征。主要表现各种现象的特征、规模、水平结构、地理分布、相互依存关系及其发展趋势，如图3-106所示。



图3-106 统计地图示意图

图3-106用不同的颜色表示电子商务网站的销售额在各区域的分布情况，比如红色代表销售额第一、粉色代表销售额第二、橘色代表销售额第三等。这样能更容易地分析销售额在哪个区域比较高，销售额高的区域将是重点服务的区域，使服务更有针对性。

3. 模块设计的设计原则

由于统计模块在电子商务平台中是一个相对独立的模块，跟其他模块的逻辑流程没有紧密的联系，并且统计模块伴随着系统地不断完善和数据量地逐渐增加，可以考虑独立成为一套数据分析系统或者应用到别的系统中去，也为了当其他模块的逻辑流程改变时不会牵连到统计模块，所以要求统计模块设计应遵循以下三条原则。

（1）可移植性

利用MVC的设计思想、设计方法及成熟的技术，既要保证统计模块运行的稳定性，又要具有良好的可移植性，因此要求统计模块功能脱离电子商务平台的其他逻辑流程独立存在。所以系统中增加了一个过滤器，当系统为需要移植时，只需要增加一个过滤器，并对模块代码稍加修改，就可以完成系统的移植，从而节省大量的时间，降低了移植过程中出错的可能。

（2）扩展性

随着业务、用户需求的变化，统计模块可以在不改变原有代码的基础上比较容易的添加新的功能。

（3）简便性和直观性

使用者无须计算机基础，只要点击相关按钮就能得到想要的相关图表信息。

（4）缓存的运用

统计模块除了一般的数据显示外，还有更多的统计运算等复杂操作，并且很多流程需要频繁操作数据库，为了提高系统运行效率，需要建立缓存，把查询结果缓存起来，下次查询时就可以不再查询数据库，直接从缓存中将以前查询的结果提取出来。

4. 数据表的设计原则

统计模块跟其他类型的模块相比对于数据上的操作不太一样，它除了一般检索外，还需要支持大量的统计运算和其他特有的操作。所以在设计统计数据表的时候有别于其他模块的数据表。要求统计模块数据表设计应遵循以下三条原则。

(1) 多建立缓存数据表

统计功能中显示的数据例如商品销量、订单量等，都是可以通过相关功能涉及的数据表得到这些数

据，但是为了统计模块的可移植性更好，也为了查询更快，建议应该多建立相关数据的缓存表。比如某会员相关的消费总金额、预存款总金额等数据可以经过处理后，存入缓存表，这样当统计模块移植的时候不会涉及其他功能模块，移植起来更容易。大大提高了它的可移植性。

（2）字段简洁，减少复杂判断

有时为了得到一个统计数据可能需要复杂的处理，使构造的SQL语句比较复杂，执行效率很低，当数据达到一定级别，则系统会相当的缓慢，所以在设计数据表字段的时候要尽量简洁，不要使以后构造的SQL语句过于复杂，比如在统计订单相关数据的时候，可能需要判断订单是否有效，这样就需要判断订单是否付款、订单是否取消等多个字段从而达到此订单是否有效的结论，使得SQL语句比较复杂，我们可以采取在

数据表中建立一列“order_isvalid”表示订单是否有效，然后用任务程序在后台处理得到该字段的值，那么当查看统计功能的时候SQL语句只需要判断该值即可，执行效率也就提高了。

（3）必要的冗余字段，减少联查

从数据表设计精简的角度出发，统计数据表只要包含了核心统计数据即可，我们就可以通过表的联查来取到我们想要的信息，但这样会大大增加数据库查询压力，所以必要的冗余字段是应该有的。建议会员名称、商品名称、商品图片、店铺名称等需要显示的字段都要包含。

综合以上因素，以会员相关统计数据表为例，表关系图如图3-107所示。

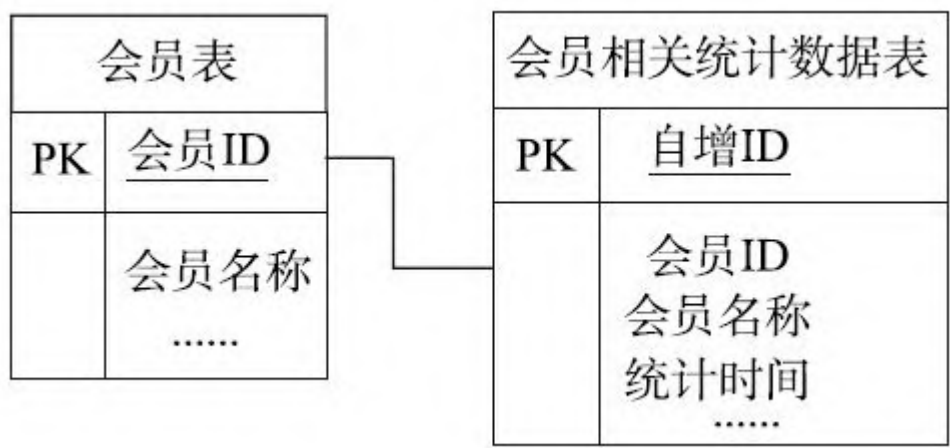


图3-107 会员相关统计数据表关系图

会员相关统计数据表结构如表3-8所示。

表3-8 会员相关统计数据表结构

字 段	类 型	Null	默 认	说 明
id	int(11)	否		自增 ID
member_id	int(11)	否	0	会员 ID
member_name	varchar(100)	否		会员名称
stat_time	int(11)	否	0	统计时间
updatetime	int (11)	否	0	记录更新时间
ordernum	int(11)	否	0	下单量
orderamount	decimal(10,2)	否	0.00	下单金额
goodsnum	int(11)	否	0	下单商品数
predincrease	decimal(10,2)	否	0.00	预存款增加金额
predreduce	decimal(10,2)	否	0.00	预存款减少金额
pointsincrease	int(11)	否	0	积分增加额
pointsreduce	int(11)	否	0	积分减少额

如表3-8所示，这样在显示会员统计信息的时候，“member_name”为冗余字段，而“ordernum”、“orderamount”等字段的值已经预先计算好缓存在表中，就只需简单的查询一张表即可实现复杂的统计数据显示。

3.10.3 代码实现

统计模块的代码主要实现了统计数据的图表展示功能，主要由统计模型类、控制器、模板输出、日期处理类、统计方法处理类和计划任务组成。下面介绍一下主要的代码片段。

1. 基本方法

(1) 统计图表及数据处理方法类

统计功能中图表的展示是很重要的一部分，简单灵活的构造统计图表是很关键的。这样可以使代码简洁且减少开发时间。代码清单3-33简单地介绍了几种构造统计图表的方法和一些其他的统计数据处理函数。

代码清单3-33 统计图表及数据处理方法类代码

```

/**
 *获得环比值
 * param float $updata上期数
 * param float $currentdata本期数
 * returnfloat 环比值
 */
function getHb($updata, $currentdata){
    if($updata != 0){
        $mtomrate = round(($currentdata -
$updata)/$updata*100, 2).'%';
    } else {
        $mtomrate = '-';
    }
    return $mtomrate;
}
/**
 *获得同比值
 * param float $updata上期数
 * param float $currentdata本期数
 * returnfloat 同比值
 */
function getTb($updata, $currentdata){
    if($updata != 0){
        $ytoyrate = round(($currentdata -
$updata)/$updata*100, 2).'%';
    } else {
        $ytoyrate = '-';
    }
    return $ytoyrate;
}
/**
 *获得地图统计图数据
 * param array $stat_arr 图表需要的设置项
 * returnstring 构造图表需要的Json数据，将该Json传递给特定JS处理类得
到统计地图
 * 返回值例如:
 *
[{"cha":1,"name":"\u5317\u4eac","des":"\uff0c\u65e0\u8ba2\u5
355\u6570\u636e","color":"#25aae2"},
{"cha":2,"name":"\u5929\u6d25","des":"\uff0c\u65e0\u8ba2\u53
55\u6570\u636e","color":"#25aae2"}]
 */
function getStatData_Map($stat_arr){
    $color_arr =

```

```
array('#fd0b07','#ff9191','#f7ba17','#fef406','#25aae2');
    $stat_arrnew = array();
    foreach ($stat_arr as $k=>$v){
        $stat_arrnew[] =
array('cha'=>$v['cha'],'name'=>$v['name'],'des'=>$v['des'],
'color'=>$color_arr[$v['level']]);
    }
    return json_encode($stat_arrnew);
}
```

(2) 日期处理方法类

对于统计时间的处理有时候也是有很多技巧的，时间处理的好，实际操作者使用很简便，并且代码写起来也比较优雅，所以统计一般是针对某个时间段进行的数据展示。代码清单3-34介绍了一些自定义的时间处理方法。

代码清单3-34 日期处理方法类代码

```
/**
 *获得系统年份数组
 * return array 系统统计可用年份数组
 */
function getSystemYearArr(){
    $year_arr =
array('2010'=>'2010','2011'=>'2011','2012'=>'2012','2013'=>'
2013','2014'=>'2014',
'2015'=>'2015','2016'=>'2016','2017'=>'2017','2018'=>'2018',
'2019'=>'2019','2020' =>'2020');
```

```

        return $year_arr;
    }
    /**
     * 获得系统月份数组
     * return array 系统统计月份数组
     */
    function getSystemMonthArr() {
        $month_arr =
array('1'=>'01','2'=>'02','3'=>'03','4'=>'04','5'=>'05',
'6'=>'06','7'=>'07','8'=>'08','9'=>'09','10'=>'10','11'=>'11',
'12'=>'12');
        return $month_arr;
    }
    /**
     * 获得系统周数组
     * return array 系统统计周数组
     */
    function getSystemWeekArr() {
        $week_arr = array('1'=>'周一','2'=>'周二','3'=>'周三',
'4'=>'周四','5'=>'周五','6'=>'周六','7'=>'周日');
        return $week_arr;
    }
    /**
     * 获得某月的最后一天
     * param int $year年份
     * param int $month月份
     * return int 某月的最后一天时间戳
     */
    function getMonthLastDay($year, $month) {
        $t = mktime(0, 0, 0, $month + 1, 1, $year);
        $t = $t - 60 * 60 * 24;
        return $t;
    }
    /**
     * 获得某月的周数组，第一周不足的需要补足
     * param int $current_year年份
     * param int $current_month月份
     * return array 某月的周数组
     */
    function getMonthWeekArr($current_year, $current_month) {
        //该月第一天
        $firstday = strtotime($current_year.'-'.
$current_month.'-01');
        //该月的第一周有几天
        $firstweekday = (7 - date('N',$firstday) + 1);
    }

```

```

        //计算该月第一个周年的时间
        $starttime = $firstday-3600*24*(7-$firstweekday);
        //该月的最后一天
        $lastday = strtotime($current_year.'-
        '.$current_month.'-01'." +1 month -1 day");
        //该月的最后一周有几天
        $lastweekday = date('N',$lastday);
        //该月的最后一个周末的时间
        $endtime = $lastday-3600*24*$lastweekday;
        $step = 3600*24*7;//步长值
        $week_arr = array();
        for ($i=$starttime; $i<$endtime; $i= $i+3600*24*7){
            $week_arr[] = array('key'=>date('Y-m-
            d',$i).'|'.date('Y-m-d',$i+3600*24*6), 'val'=>date('Y-m-
            d',$i).'~'.date('Y-m-d',$i+3600*24*6));
        }
        return $week_arr;
    }
    /**
     *获得本周的开始时间和结束时间
     * param int $current_time年份
     * return array 本周的开始时间和结束时间
     */
    function getWeek_SdateAndEdate($current_time){
        $current_time = strtotime(date('Y-m-d',$current_time));
        $return_arr['sdate'] = date('Y-m-d', $current_time-
        86400*(date('N',$current_time) - 1));
        $return_arr['edate'] = date('Y-m-d',
        $current_time+86400*(7- date('N',$current_time)));
        return $return_arr;
    }

```

2. 制作统计页面

上面已经介绍完了统计使用到的基本方法，接下来来简单介绍一下制作统计页面的基本流程。

步骤一：处理图表数据。

```
/**
 *新增会员统计
 */
public function newmemberOp(){
    if (!$this->search_arr['search_type']){
        $this->search_arr['search_type'] = 'day';
    }
    $model = Model('stat');
    $statlist = array();//统计数据列表
    $stat_arr= array();//统计图参数数组
    $where = array();
    $field = ' COUNT(*) as allnum ';
    //.....此处为查询数据和构造统计参数的过程
    //计算同比
    foreach ((array)$currlist_arr as $k=>$v){
        $tmp = array();
        $tmp['timetext'] = $v['timetext'];
        $tmp['seartime'] = $v['stime'].'|'.$v['etime'];
        $tmp['currentdata'] = $v['val'];
        $tmp['updata'] = $uplist_arr[$k]['val'];
        $tmp['tbrate'] = getTb($tmp['updata'],
    $tmp['currentdata']);
        $statlist['data'][] = $tmp;
    }
    //.....导出Excel
    if ($_GET['exporttype'] == 'excel'){
        //导出Excel
        exit();
    } else {
        //得到统计图数据
        $stat_arr['title'] = '新增会员统计';
        $stat_arr['yAxis'] = '新增会员数';
        $stat_json = getStatData_LineLabels($stat_arr);
        Tpl::output('stat_json',$stat_json);
        Tpl::output('statlist',$statlist);
        Tpl::showpage('stat.newmember');
    }
}
```

步骤二：页面输出图表。

```
<div id="container"></div>
<script
type="text/javascript"src="highcharts/highcharts.js">
</script>
<script>
$(function () {
$('#container').highcharts(<?php echo$output['stat_json'];?
>);
});
</script>
```

3.10.4 功能体验

本节将介绍部分统计功能并简单介绍一下它的使用方式。

1. 总体数据快捷展示

电子商务运营期间有一些数据是非常重要的且需要经常查看的，为此一般统计功能中会设置一个总体数据展示的页面，便于快速查看这些数据，如图3-108所示，展示了一些订单、客户、商品等统计数据。

2. 挖掘老客户价值

二八定律说的是企业80%的业务收入通常来自于20%的客户。带来一个新客户的成本是维护好一个老客户的3~5倍，当网站成功地把一个访客转化成一个客户之后，如何提高这个客户对于网站的忠诚度，继而增

加客户对于网站的整体贡献值变得非常重要。只有有效地提高每个客户的消费，才能快速提升电子商务网站的整体收入。通过Web数据挖掘，我们可以发现客户通常在什么样的时间段内下单，客户一般能承受的每单价格是多少，哪位是我们最重要的客户，大客户的主要分布区域，客户喜欢什么样的商品等，以便对其进行个性化营销和人性化关怀。下面我们看一些例子。

图3-109展现了订单量在各个时间段的分布情况，我们可以清晰的了解，客户喜好的下单时间，从而合理的安排客服工作时间，提高工作效率。在最佳的时间为客户提供最优的服务，使效益最大化。

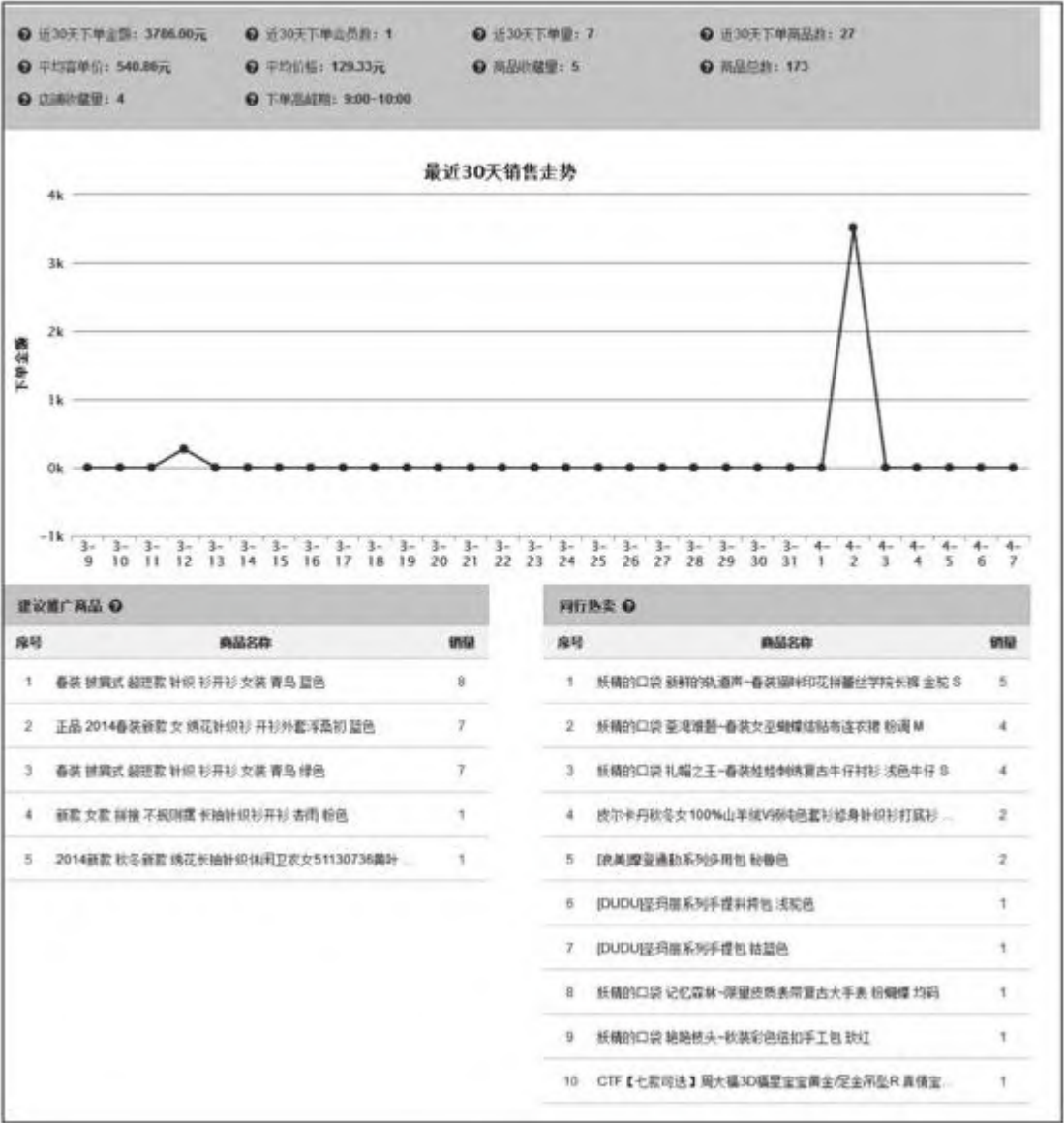


图3-108 总体数据快捷展示页面

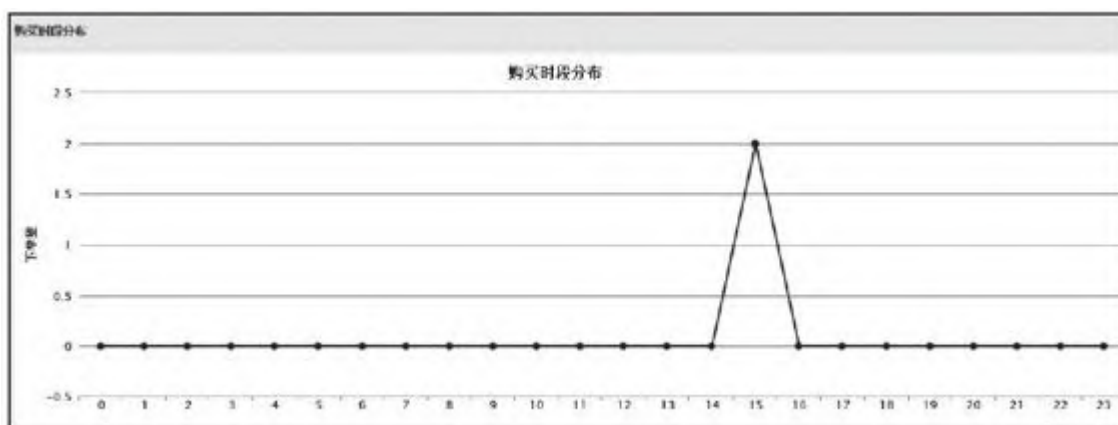


图3-109 购买时段分布情况展示页面

图3-110展现了各个下单次数的会员数和占整体下单数的比例，从而分析新会员和老会员在销售中的比重，比如购买1次的会员数占比为60%，说明网站的主要消费群体为新客户，客户的复购频率低，需要思考如何提高客户黏性，什么原因造成的客户黏性低，如何让客户不得不再次购买，是产品质量需要提高还是服务、售后需要提高，能够有针对性地改善，从而事半功倍的提升效益。

购买频次分析			
序号	购买频次	下单会员数	人数占比
1	购买频次1次	2000	33.72%
2	购买频次2次	1200	20.25%
3	购买频次3次	1000	16.85%
4	购买频次4次	500	8.44%
5	购买频次5次	450	7.6%
6	购买频次6次	312	5.27%
7	购买频次7次	206	3.43%
8	购买频次8次	160	2.7%
9	购买频次9次	83	1.4%
10	购买频次10次以上	12	0.2%

图3-110 购买频次分析展示页面

电子商务很重要的一点就是突破了区域的限制，电子商务网站的客户群更是遍布全国。中国幅员辽阔，每个区域的客户都有其特有的特点，例如由于气候和习惯的不同，冬季北方的客户可能需要羽绒服，而南方的客户可能就不需要，所以针对不同区域的客户进行不同的营销显得意义重大。而这点的基础是先了解自己的重点客户群在哪个区域。如图3-111所示，用不同的颜色显示了不同区域的客户数多少，这样可以一目了然地了解自己的重点客户群的区域分布情况，进行针对性营销。

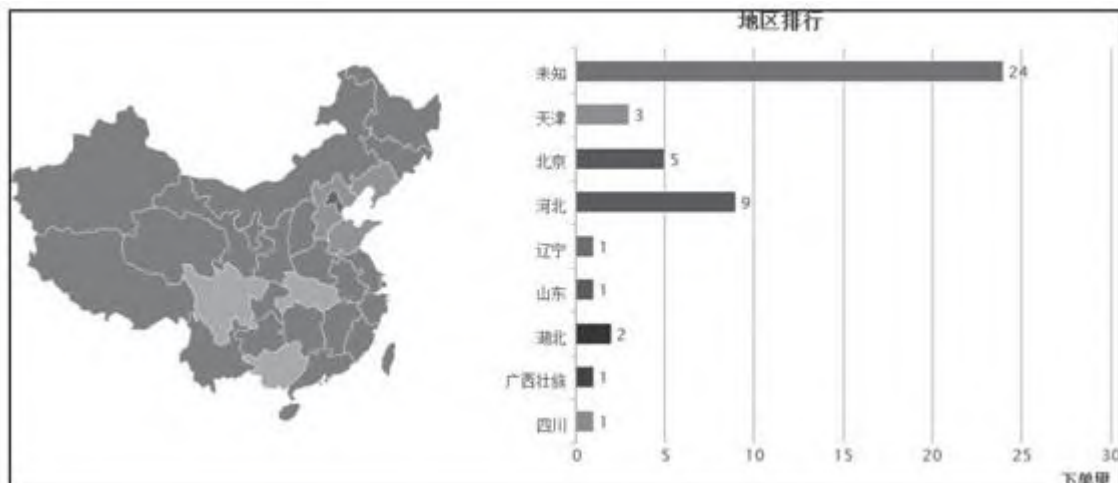


图3-111 客户下单区域分布页面

电子商务运营的时候常常出现的一个问题就是，谁是我们的重点客户，提高服务质量显然不能眉毛胡子一把抓，这样不仅服务提升不了还会降低客服人员的积极性。了解自己的大客户在哪，有针对性地提供服务，是一个事半功倍的办法。图3-112中列出了客户的消费数据，清晰地显示了谁才是重点客户，需要重点服务。

会员名称	下单金额	退货金额	退货金额	退货分	退货分
lzpnsako	4965.00	10000.00	5753.00	313	0
shopen	3459.00	0.00	580.00	301193	2000
lybyly	1343.50	125659.50	46902.50	90	0
hou	980.00	10.00	56800.00	1009	0
apple2	869.00	124.00	233.00	317	60
dabing	500.00	999999.00	1000.00	55	0
fishopenc	447.00	100259.00	1374.00	50281	2050
admirb	140.00	10130.00	280.00	67	0
shopenc	99.00	99.00	198.00	270	0

图3-112 客户消费数据列表

3. 客户需要什么商品

要留住客户，我们需要更懂客户，分析客户需要和喜好什么商品，针对客户的喜好上架商品。这样不仅增加了销售额，还可以防止商品滞销造成的库存积压，这对于商城来说很重要。下面介绍一些关于商品的数据分析。

电子商务运营的时候都要有自己的价格定位，什么价位的商品好卖受欢迎，什么价位利润高，都需要了解，才能更好地增加效益。图3-113展示了各价位商品的销量走势。

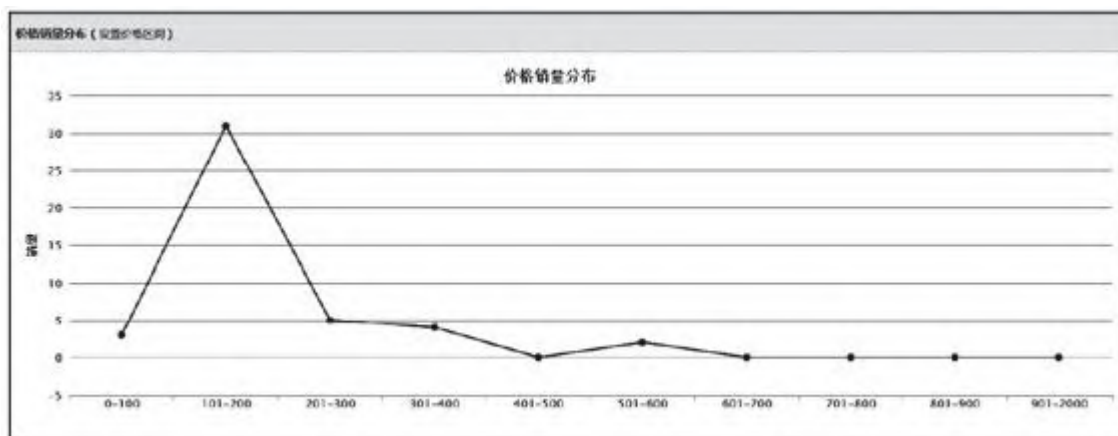


图3-113 各价位商品销量走势图

电子商务网站商品很多，因此运营时必须知道哪款商品热销。我们可以根据销售数据和访问数据找出热门商品，然后找到合适的关键字和广告创意与该热销商品形成最佳组合，进行广告投放，带来更多的流量和订单，让热销商品带动整个网站的销售。图3-114即是热销商品的展示图。

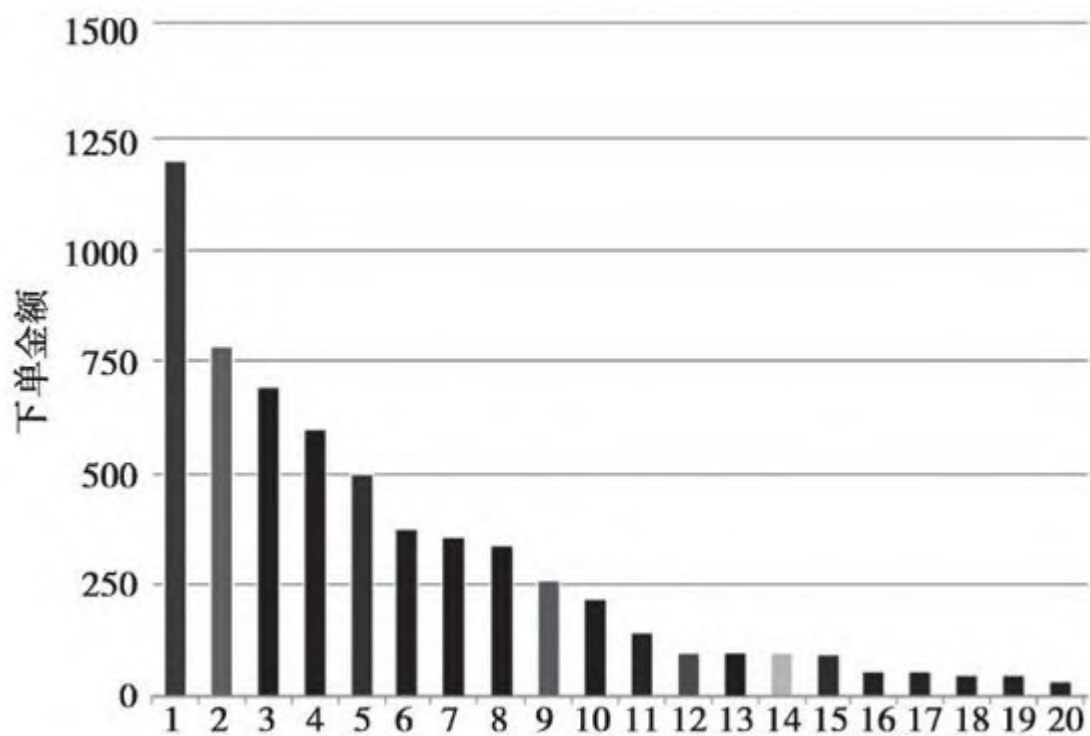


图3-114 热销商品排行示例图

还可以根据客户对商品的浏览次数，了解客户比较感兴趣的商品，并在此页面增加一些类似商品推荐，如图3-115所示。

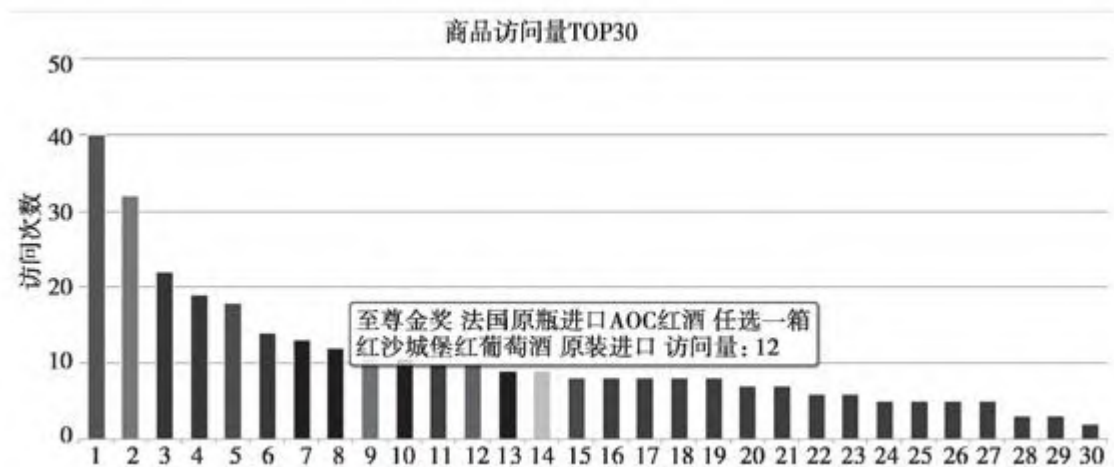


图3-115 商品访问量排行

4. 优化行业结构

对于经营范围较广的网站，比较行业的销售情况，哪个行业销售得好，需要增加库存和推广，哪个行业出现了滞销，需要减少库存，显得十分重要。电子商务网站有其擅长运营的行业，选对行业则会事半功倍，否则可能会造成库存积压，后果也是很严重的。从很多的真实运营实例中就可以发现这点。京东从创业之初聚焦3C类商品，到后来扩充品类至各行业，可以说是品类齐全，但是3C类商品始终是它的重点。因为这是它销售最好的商品、是平台的根基。下

面介绍几种针对行业的统计，从不同层面分析出热销行业。

图3-116展现了各个行业的商品平均价格、销售情况、库存量及滞销商品数，了解行业的基本情况，从而可以有的放矢地优化库存商品结构。

类目名称	平均价格 (元)	有销量商品数	销量	销售额 (元)	商品总数	无销量商品数
服饰鞋帽	267.15	14	45	6924.00	191	177
珠宝手表	2286.33	2	2	2262.00	77	25
礼品赠品	76887.15	4	5	1911.00	13	9
食品饮料	308.40	2	2	248.00	25	23
医疗保健	1506.00	0	0	0	1	1
母婴用品	0.00	0	0	0	0	0
宠物用品	999995.00	0	0	0	1	1
户外运动	255.00	0	0	0	2	2

图3-116 行业情况总览

商品的销量高只可以证明销售的商品多，而不能说明利润高，每个行业都有其销售的黄金价格，在黄金价格区间商品的利润是最高的。因此我们需要分析出各个行业的黄金价格，将商品价格定位在黄金价格区间中，易于销售。图3-117展示了女士服装在各个价

格区间的销售走势，从中可以简单地提取出黄金价格区间。

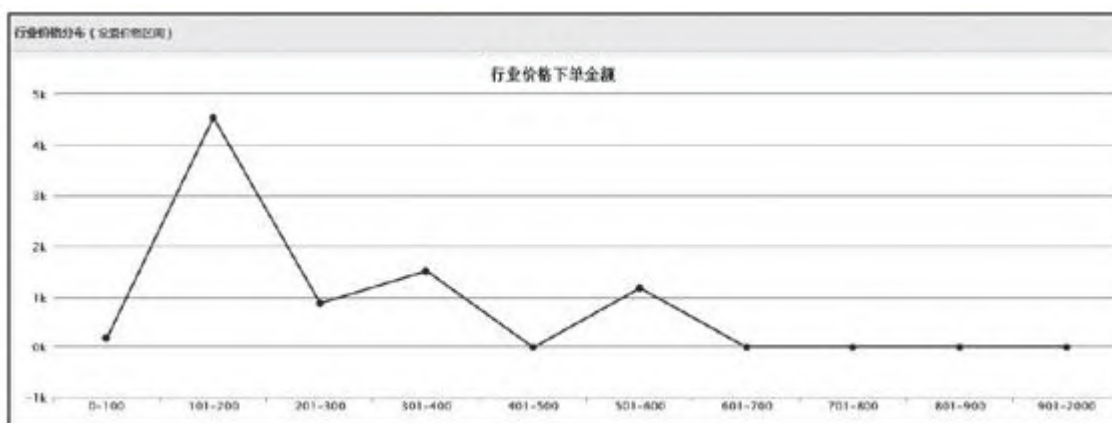


图3-117 行业价格区间销售走势图

由图3-117可知，101~200元是女士服装的销售黄金价格区间，因此应该将该品类商品的价格区间调整为101~200元，并主打该价格区间的商品。

5. 个性化推荐

个性化推荐系统是建立在数据挖掘基础上的一套系统，以帮助顾客购物提供完全个性化的决策支持和信息服务为目的。几乎所有的海外大型电子商务网

站，包括亚马逊和EBAY，都不同程度地使用了各种形式的推荐系统。国内的淘宝、京东等商城也在很多模块体现了个性化的推荐。举个例子，如图3-118所示，在京东加入购物车成功的提示页面中，经常会看到类似这样的一句话“购买了该商品的用户还购买了”，这句话的下面会展示一系列的商品，这就是最普通的关联商品推荐，通过对所有客户的消费记录进行分析，找出最常见的商品搭配，并以此来推荐。



图3-118 购买该商品的用户还购买了

这种有针对性的推荐功能，可以帮助客户找到他们感兴趣的并愿意买的某样商品，推荐客户购买一些相关的商品或者购买一些相关的但是对于商家来说利润更高的商品。这样不仅增加了网站的交叉销售和向上销售，而且因为客户往往更愿意到那些最能满足自己需求的网站去购物所以还有益于建立忠诚度，使得网站和用户得到双赢。

一个成功的推荐系统会有以下几个作用。

- 1) 把访客转化为购买者。
- 2) 提高网站交叉销售能力。
- 3) 提高客单价，增强向上销售能力。
- 4) 提高客户满意度和忠诚度。

再比如，ShopNC的B2B2C商城系统中的个性化推荐功能——“猜你喜欢”。它根据客户近期的浏览历史记录，判断客户可能喜欢的商品类型，并在商品列表页面和其他与商品购买关系密切的页面，展示“猜你喜欢”的商品推荐功能，如图3-119所示。



图3-119 ShopNC B2B2C商城系统猜你喜欢功能

3.10.5 开发和使用

挖掘不能产生商业价值的数据是没有意义的，只有把数据挖掘充分应用到商业上，将大量的数据经过加工处理转换成信息，这样我们之前做的所有数据准备和算法研究工作才有意义。因此应该从商业角度出发，看到底需要完成怎样的商业目标，再制定数据分析和挖掘过程。而以数据为基础的电子商务运营，不是一下子就能够成功的，它是一个永远不停地变化的过程，不是做了一次优化就能够一劳永逸的。我们根据当前的来源分析和访客分析得出客户兴趣点和热门商品对整个网站做了调整，在经过一段时间之后，客户的整体兴趣点一定会发生变化，而热门商品也会发生迁移，就需要再次进行优化。所以以数据为基础的运营应该是持续不断地进行优化。

数据运营应注意以下几点问题。

（1）人的问题

目前在中国的企业数据运营整体来说还不成熟。一些领导对数据的作用将信将疑，不愿意投入。另外，应用了一些数据分析产品的公司只追求最后的结果，而对数据挖掘的过程、数据的存储、数据挖掘结果的知识积累和呈现不重视。如果只有执行层的数据分析师在分析和审视数据，领导却不重视，那么数据运营是没有意义的。对数据的重视应该从领导做起。

（2）实际运用

很多时候许多企业只是将数据的重要性说在口头上，而实际操作的时候却仍然按照经验和习惯去做，即使用了数据也只是停留在看Excel报表的阶段。这样

是不够的，应该将数据实际使用起来，使它融入到运营的各个环节，“拿数据说话”。

（3）不要最好只求合适

对于一个需要解决的问题，可能有很多很好的数据挖掘算法可以使用，但是通常只有一个合适的算法。当我们选择一个数据挖掘算法时，要弄清楚它是否适合我们要解决的问题。如果本身方法选择不合适，那么再好地执行也没意义。

（4）求真实

电子商务网站的数据挖掘很有意义，但是难免会存在一部分虚假数据，例如客户评价、浏览量不同程度的存在一些虚假数据，在数据挖掘时，尽可能地提取有效信息。摒弃虚假数据，对于数据挖掘的意义很重要。

（5）重复性

数据挖掘每次导出的结果跟采用的数据集直接相关。如果数据集发生变化，就需要重新进行挖掘。如果没有考虑数据变化而盲目采用数据变化之前的策略，那么结果是不可预料的。数据挖掘需要隔一段时间重新执行，因为随着时间、商品和访客的变化，数据挖掘算法的产出结果也是不同的。

（6）数据积累

电子商务网站初期运营的时候，不管是商品数据、客户购买记录，还是客户浏览记录，数据量都是相对比较单薄的。当数据量没有积累到一定数据的时候，很多数据分析意义不大。比如一天中10点销售了6件，14点销售了10件，并不能足够说明14点是比较适合销售。数据分析需要积累一定量的数据，经过数据挖掘得出的结果才有说服力。

（7）快速反应

当一个客户访问了某件或者几件商品的详细信息时，这表明他在那一刻对于这个商品是很感兴趣的。而对于大部分客户来说，过了这个村就没有这个店了。如果过一段时间再向该客户进行个性化推荐，则意义不大。所以当数据产生之后如果不能及时处理或者分析，就可能没有价值或者价值极度下降了。应注意对于客户产生的数据要快速反应，得出数据挖掘的结果，使数据效果最大化。

实际运营数据分析和挖掘的时候，很可能疏忽了以上几点，造成结果偏差。一定要考虑这些因素，使积累的数据更好地为网站运营服务。

3.11 预存款

预存款支付是一种方便快捷的站内支付形式，它可以不依赖于第三方支付平台而独立完成支付。会员对预存款主要有三种操作，分别为充值、提现和购物（支出）。预存款主要操作如图3-120所示。

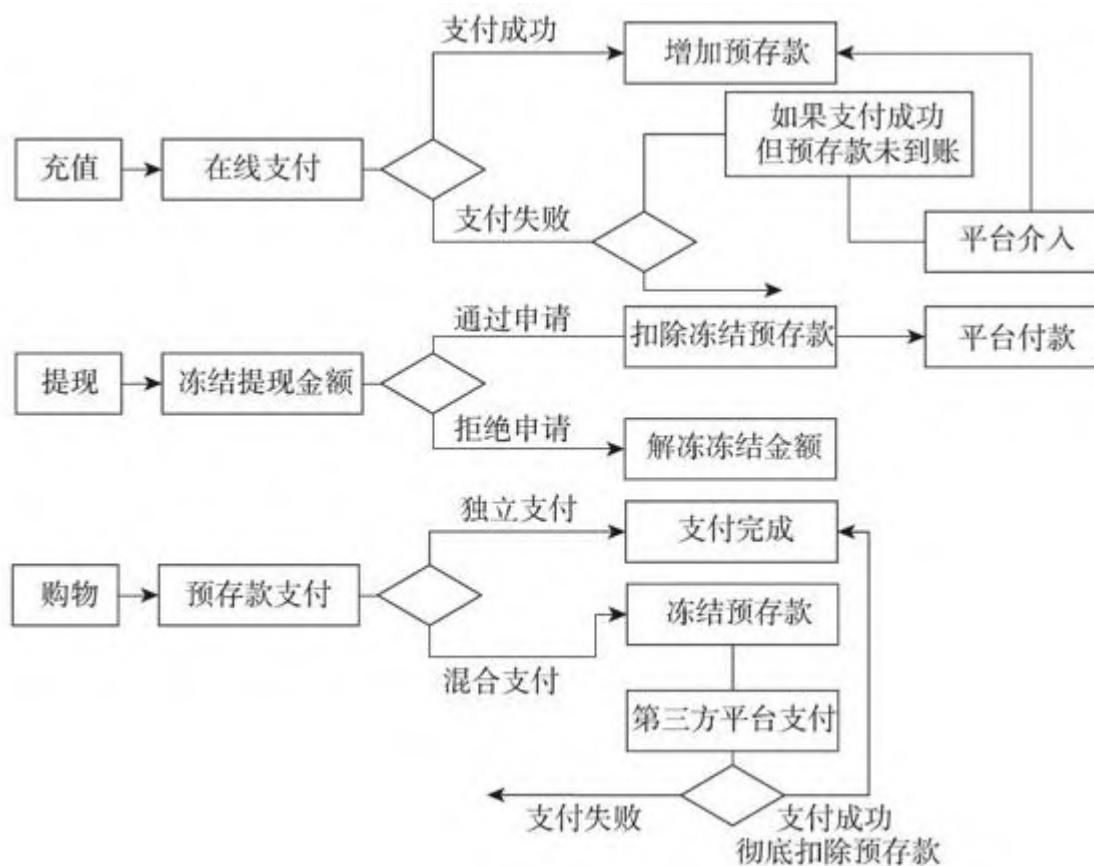


图3-120 支付平台处理流程

预存款使用前需要充值，充值方式以在线支付为主，通过第三方支付平台完成支付，支付成功后，相应的金额会充值到您的预存款账户中。

提现功能在预存款中不可缺少，提现时首先向大平台发起提现申请，申请成功后，系统会冻结您的提现金额，等待系统审核。如果审核通过，系统会彻底扣除之前冻结的预存款，平台向提款人支付相应的提现金额。如果未通过审核，则系统自动将之前冻结的预存款解冻。

购物是系统提供预存款功能的最终目的，预存款支付分为独立支付和混合支付两种情况：独立支付是指当预存款足够时，直接扣除预存款的情况；混合支付是指预存款不足时，差额部分需要通过第三方支付平台进行支付的情况。混合支付时，系统首先依次循环每个子订单，如果预存款可用金额大于该子订单，

则将预存款金额扣除，并将子订单状态置成已支付状态；如果预存款可用金额不足时，则将剩余的预存款暂时冻结，然后到第三方支付平台支付差额部分，支付成功后，系统再彻底扣除先前冻结的预存款。如果第三方支付平台支付失败，或未到第三方平台支付，预存款还是处于暂时冻结状态，买家可以到“我的订单”中进行再次支付。买家可以将未支付成功的订单取消，取消后，预存款冻结金额会重新变为可用金额，可继续购买其他商品。

3.11.1 设计思路

1. 设计要求

1) 安全性

预存款存放着用户资金，可以随时用来购物和申请提现，这就要求系统对预存款的操作要进行非常严格的校验，保障用户的资金安全。

2) 数据完整性

任何涉金钱的操作都必须保证数据的完整性，对于重要的字段数据变更要全部使用事务进行处理，一个点发生错误，整个事务都需要回滚，保证数据的完整性和准确性。对于预存款的任务变更都要有详细的日志记录，方便监控和日志查询。

2. 数据表设计

实现预存款功能，至少需要三个表，分别为充值表、提现表和日志表。

充值表用来记录会员的充值信息，主要字段包括充值单号（唯一值）、会员信息、充值金额、充值时间、充值状态等，如果管理员介入操作，还应记录管理员的身份。

提现表用来记录会员的提现信息，主要字段包括提现单号（唯一值）、会员信息、提现金额、收款银行信息、申请状态以及平台的付款信息（付款时间，操作人）等。

日志表用来记录所有变更预存款时的操作记录，包括对可用金额和冻结金额的变更都要作详细的记录，主要字段包括操作人信息、操作类型（如下单、

提现、充值、退款等）、可用预存款、冻结预存款、
操作时间、备注等。

3.11.2 代码实现

1. 预存款模型类

预存款模型类需要对充值、提现和日志表的所有操作进行封装，供业务层统一调用，如代码清单3-35所示。

代码清单3-35 预存款模型类

```
class predepositModel extends Model {
//生成充值编号
public function makeSn() {
    return mt_rand(10,99)
        . sprintf('%010d',time() - 946656000)
        . sprintf('%03d', (float) microtime() * 1000)
        . sprintf('%03d', (int) $_SESSION['member_id']
% 1000);
}
//取得充值列表
public function getPdRechargeList($condition = array(),
$page_size = '', $fields = '*', $order = '', $limit = '') {
    return $this->table('pd_recharge')->where($condition)-
>field($fields)->order($order)->limit($limit)-
>page($page_size)->select();
}
//添加充值记录
public function addPdRecharge($data) {
    return $this->table('pd_recharge')->insert($data);
}
```

```

}
//编辑充值记录
public function editPdRecharge($data,$condition = array()) {
    return $this->table('pd_recharge')->where($condition)->update($data);
}
//取得单条充值信息
public function getPdRechargeInfo($condition = array(),
$fields = '*') {
    return $this->table('pd_recharge')->where($condition)->field($fields)->find();
}
//取充值信息总数
public function getPdRechargeCount($condition = array()) {
    return $this->table('pd_recharge')->where($condition)->count();
}
//取提现单信息总数
public function getPdCashCount($condition = array()) {
    return $this->table('pd_cash')->where($condition)->count();
}
//取日志总数
public function getPdLogCount($condition = array()) {
    return $this->table('pd_log')->where($condition)->count();
}
//取得预存款变更日志列表
public function getPdLogList($condition = array(), $pagesize = '', $fields = '*', $order = '', $limit = '') {
    return $this->table('pd_log')->where($condition)->field($fields)->order($order)->limit($limit)->page($pagesize)->select();
}
//变更预存款时，日志记录统一方法入口
public function changePd($change_type,$data = array()) {
    $data_log = array();
    $data_pd = array();
    $data_msg = array();
    $data_log['lg_member_id'] = $data['member_id'];
    $data_log['lg_member_name'] = $data['member_name'];
    $data_log['lg_add_time'] = TIMESTAMP;
    $data_log['lg_type'] = $change_type;
    $data_msg['time'] = date('Y-m-d H:i:s');
    $data_msg['pd_url'] = urlShop('predeposit',

```

```

'pd_log_list');
    switch ($change_type){
        case 'order_pay':
            $data_log['lg_av_amount'] = -$data['amount'];
            $data_log['lg_desc'] = '下单, 支付预存款, 订单号:
'.'. $data['order_sn'];
            $data_pd['available_predeposit'] =
array('exp','available_predeposit-'.'. $data['amount']);
            $data_msg['av_amount'] = -$data['amount'];
            $data_msg['freeze_amount'] = 0;
            $data_msg['desc'] = $data_log['lg_desc'];
            break;
        case 'order_freeze':
            $data_log['lg_av_amount'] = -$data['amount'];
            $data_log['lg_freeze_amount'] = $data['amount'];
            $data_log['lg_desc'] = '下单, 冻结预存款, 订单号:
'.'. $data['order_sn'];
            $data_pd['freeze_predeposit'] =
array('exp','freeze_predeposit+'.'. $data['amount']);
            $data_pd['available_predeposit'] =
array('exp','available_predeposit-'.'. $data['amount']);
            $data_msg['av_amount'] = -$data['amount'];
            $data_msg['freeze_amount'] = $data['amount'];
            $data_msg['desc'] = $data_log['lg_desc'];
            break;
        case '.....':
            //取消订单
            //下单
            //充值
            //确认退款
            //申请提现
            //提现成功
            //取消提现申请
            //...
            break;
        default:
            throw new Exception('参数错误');
            break;
    }
    $update = $this->table('member')-
>where(array('member_id'=>$data['member_id']))-
>update($data_pd);
    if (!$update) {
        throw new Exception('操作失败');
    }
}

```

```

        $insert = $this->table('pd_log')->insert($data_log);
        if (!$insert) {
            throw new Exception('操作失败');
        }
        return $insert;
    }
    //删除充值记录
    public function delPdRecharge($condition) {
        return $this->table('pd_recharge')->where($condition)->delete();
    }
    //取得提现列表
    public function getPdCashList($condition = array(),
        $pagesize = '', $fields = '*', $order = '', $limit = '') {
        return $this->table('pd_cash')->where($condition)->field($fields)->order($order)->limit($limit)->page($pagesize)->select();
    }
    //添加提现记录
    public function addPdCash($data) {
        return $this->table('pd_cash')->insert($data);
    }
    //编辑提现记录
    public function editPdCash($data,$condition = array()) {
        return $this->table('pd_cash')->where($condition)->update($data);
    }
    //取得单条提现信息
    public function getPdCashInfo($condition = array(), $fields = '*') {
        return $this->table('pd_cash')->where($condition)->field($fields)->find();
    }
    //删除提现记录
    public function delPdCash($condition) {
        return $this->table('pd_cash')->where($condition)->delete();
    }
}

```

2. 在线充值

在线充值首先需要在商城平台输入充值金额，然后进入第三方平台完成支付。

代码清单3-36 在线充值代码

```
//格式化充值金额
$pd_r_amount = abs(floatval($_POST['pdr_amount']));
if ($pd_r_amount <= 0) {

    showMessage(Language::get('predeposit_recharge_add_pricemin_error'),'','html','error');
}
//添加一条充值记录
$model_pdr = Model('predeposit');
$data = array();
$data['pdr_sn'] = $pay_sn = $model_pdr->makeSn();
$data['pdr_member_id'] = $_SESSION['member_id'];
$data['pdr_member_name'] = $_SESSION['member_name'];
$data['pdr_amount'] = $pd_r_amount;
$data['pdr_add_time'] = TIMESTAMP;
$insert = $model_pdr->addPdRecharge($data);
if ($insert) {
    //转向到商城支付页面，这里统一使用第三方支付平台API
    redirect('index.php?act=buy&op=pd_pay&pay_sn='.$pay_sn);
}
```

3. 申请提现

要实现申请提现，首先要输入提现金额、收款银行账号、支付密码等信息，然后提交系统验证。系统

验证的主要代码如代码清单3-37所示。

代码清单3-37 申请提现代码

```
$model_pd = Model('predeposit');
$model_member = Model('member');
//取得会员信息
$member_info = $model_member-
>getMemberInfo(array('member_id'=>$_SESSION['member_id']));
//验证支付密码
if (md5($_POST['password']) !=
$model_member_info['member_paypwd']) {
    showDialog('支付密码错误','', 'error');
}
//验证金额是否足够
if (floatval($member_info['available_predeposit']) <
$pd_amount){

showDialog(Language::get('predeposit_cash_shortprice_error')
, 'index.php?act=predeposit&op=pd_cash_list', 'error');
}
//采用事务处理
try {
    //添加提现记录
    $model_pd->beginTransaction();
    $pd_sn = $model_pd->makeSn();
    $data = array();
    $data['pd_sn'] = $pd_sn;
    $data['pd_member_id'] = $_SESSION['member_id'];
    $data['pd_member_name'] = $_SESSION['member_name'];
    $data['pd_amount'] = $pd_amount;
    $data['pd_bank_name'] = $_POST['pd_bank_name'];
    $data['pd_bank_no'] = $_POST['pd_bank_no'];
    $data['pd_bank_user'] = $_POST['pd_bank_user'];
    $data['pd_add_time'] = TIMESTAMP;
    $data['pd_payment_state'] = 0;
    $insert = $model_pd->addPdCash($data);
    if (!$insert) {
        throw new
Exception(Language::get('predeposit_cash_add_fail'));
    }
}
```

```
}  
//冻结等额的预存款  
$data = array();  
$data['member_id'] = $member_info['member_id'];  
$data['member_name'] = $member_info['member_name'];  
$data['amount'] = $pdc_amount;  
$data['order_sn'] = $pdc_sn;  
$model_pd->changePd('cash_apply',$data);  
$model_pd->commit();  
//提现成功，事务提交  
} catch (Exception $e) {  
    $model_pd->rollback();  
    //提现失败，事务回滚  
}  
}
```

4. 预存款支付

代码清单3-38展示的是预存款支付时的核心代码，首先进行系统验证，验证通过后对每个子订单扣除预存款。

代码清单3-38 预存款支付代码

```
//验证系统是否开启预存款  
$model_payment = Model('payment');  
$pd_payment_info = $model_payment->  
getPaymentOpenInfo(array('payment_code'=>'predeposit'));  
if (empty($pd_payment_info)) return;  
//验证会员支付密码是否正确，是否有预存款  
$buyer_info = Model('member')->  
getMemberInfo(array('member_id' =>  
$member_id), 'available_predeposit,member_paypwd');  
if ($buyer_info['member_paypwd'] == '' ||
```

```

$buyer_info['member_paypwd'] != md5($input['password']))
return ;
$available_pd_amount =
floatval($buyer_info['available_predeposit']);
if ($available_pd_amount <= 0) return;
//依次循环每个子订单，如果预存款大于子订单金额，则完全用预存款支付
//如果不足，则先支付预存款（暂时冻结），使用其他API支付完成后，再完全扣除预存款
$model_order = Model('order');
$model_pd = Model('predeposit');
foreach ($order_list as $order_info) {
//跳过货到付款的订单
if ($order_info['payment_code'] == 'offline') continue;
$order_amount = floatval($order_info['order_amount']);
$data_pd = array();
$data_pd['member_id'] = $member_id;
$data_pd['member_name'] = $member_name;
$data_pd['amount'] = $order_info['order_amount'];
$data_pd['order_sn'] = $order_info['order_sn'];
if ($available_pd_amount >= $order_amount) {
//预存款立即支付，订单支付完成
$model_pd->changePd('order_pay',$data_pd);
$available_pd_amount -= $order_amount;
//记录订单日志(已付款)
$data = array();
$data['order_id'] = $order_info['order_id'];
$data['log_role'] = 'buyer';
$data['log_msg'] = L('order_log_pay');
$data['log_orderstate'] = ORDER_STATE_PAY;
$insert = $model_order->addOrderLog($data);
if (!$insert) {
throw new Exception('记录订单日志出现错误');
}
//订单状态置为已支付
$data_order = array();
$data_order['order_state'] = ORDER_STATE_PAY;
$data_order['payment_time'] = TIMESTAMP;
$data_order['payment_code'] = 'predeposit';
$data_order['pd_amount'] = $order_amount;
$result = $model_order-
>editOrder($data_order,array('order_id'=>$order_info['order_
id']));
if (!$result) {
throw new Exception('订单更新失败');
}
}
}

```

```
} else {  
    //暂冻结预存款,后面还需要 API彻底完成支付  
    if ($available_pd_amount > 0) {  
        $data_pd['amount'] = $available_pd_amount;  
        $model_pd->changePd('order_freeze',$data_pd);  
        //预存款支付金额保存到订单  
        $data_order = array();  
        $data_order['pd_amount'] = $available_pd_amount;  
        $result = $model_order->  
            >editOrder($data_order,array('order_id'=>$order_info['order_  
            id']));  
        $available_pd_amount = 0;  
        if (!$result) {  
            throw new Exception('订单更新失败');  
        }  
    }  
}  
}
```

3.11.3 功能体验

本节以充值→提现→支付为顺序进行功能演示。

1. 在线充值

首先买家进行在线充值，如图3-121所示。



图3-121 在线充值第一步

输入金额后提交，选择第三方支付平台，如图3-122所示。

支付提交

查看充值记录可以通过我的充值列表 进行查看。

充值单号：390450351816163007

您已申请账户余额充值，请立即在线支付！充值金额：¥10.00

支付选择







确认提交支付

图3-122 在线充值第二步

这里选择支付宝支付，如图3-123所示。

如果支付成功并同步到商城，则充值成功，如果第三方支付平台支付成功，但同步到商城失败，这时平台管理人员可以手工协助一下，在后台预存款充值管理中找到该充值单，输入付款时间、支付方式，以及第三方平台交易号，这里输入支付宝中的交易号，然后提交即可完成充值，如图3-124所示。

支付宝账户付款

有卡就能付

账户名：

忘记账户名？

手机号码/邮箱

支付密码：

忘记密码？

请输入账户的 支付密码，不是登录密码。

下一步

图3-123 支付宝平台付款

预存款	充值管理	提现管理
充值单号:		
390450351816163007		
充值金额(元):		
10.00		
会员名称:		
sina		
付款时间:		
2014-04-09		
付款方式:		
支付宝		
第三方支付平台交易号:		
301274462585		支付宝等第三方支付平台交易号
提交		

图3-124 系统后台协助更改充值支付信息

这时查看一下账户余额，发现充值成功，如图3-125所示。

账户余额	账户充值	余额提现	
可用金额: 10.00 元 冻结金额: 0.00 元			
创建时间	可用金额(元)	冻结金额(元)	变更说明
2014-04-09 09:50:17	+10.00		充值, 充值单号: 390450351816163007

图3-125 账户余额及预存款变更记录

2. 申请提现

首先需要进行安全验证码验证，用户可以选择使用邮箱或手机接收系统发送的验证码（这就要求用户首先要绑定手机或邮箱才可以进行提现操作），如图3-126所示。

* 选择身份认证方式:
邮箱 [kev****ou@foxmail.com]

获取安全验证码

* 请输入安全验证码:

* 图形验证码:

看不清? 换张图

确认, 进入下一步

图3-126 提现之前进行安全验证

进入下一步，需要用户输入相关提现信息，提交后需要系统审核，同时将提现金额冻结，如图3-127所示。



The image shows a web form for withdrawal. The form fields are as follows:

- * 提现金额: 5 元 (当前可用金额: 10 元)
- * 收款银行: 中国工商银行天津南开红旗路支行
强烈建议优先填写国有4大银行(中国银行、中国建设银行、中国工商银行和中国农业
“支付宝”、“财付通”即可。
- * 收款账号: 4665 银行帐
- 开户人姓名: 王某 收款账
- * 支付密码:

A modal box titled "提示信息" (Information) is overlaid on the form, displaying a success message: "您的提现申请已成功提交，请等待系统处理" (Your withdrawal application has been successfully submitted, please wait for system processing). The modal also includes a close button (X) and a timer indicating "2秒后窗口关闭" (Window will close in 2 seconds).

At the bottom of the form, there are two buttons: "确认提现" (Confirm Withdrawal) and "取消并返回" (Cancel and Return).

图3-127 输入提现信息并提交

管理员进入系统后台找到相关提现单，审核无误后付款给用户，并更改提现单状态，到此整个提现流程完成，如图3-128所示。

充值单号:
560450353146967007
会员名称:
sina
提现金额:
5.00 元
收款银行:
中国工商银行天津南开红旗路支行
收款账号:
4665 80002 4447 0071
收款人姓名:
王某某
更改支付状态

图3-128 管理员审核提现信息

最后再查看一下我的预存款，再次验证提现成功。如图3-129所示，系统以列表形式详细展示了预存款每次的变更信息，从日志中可以发现，系统首先冻结提现金额，审核通过后，再彻底扣除冻结金额，如果系统审核未通过，冻结的金额会增加到会员的可用金额中。

可用金额: 5.00 元 冻结金额: 0.00 元			
创建时间	可用金额(元)	冻结金额(元)	变更说明
2014-04-09 10:07:30		-5.00	提现成功, 提现单号: 560450353146967007
2014-04-09 10:05:46	-5.00	+5.00	申请提现, 冻结预存款, 提现单号: 560450353146967007
2014-04-09 09:50:17	+10.00		充值, 充值单号: 390450351816163007

图3-129 预存款日志列表

3. 预存款支付

随机选择两个商品加入到购物车，如图3-130所示。

☒ 全选	商品	单价(元)	数量	小计	操作
官方店铺					
☒	 新款女款 拼接 不规则露 长袖针织衫开衫 杏雨 粉色	¥2.00	- 1 +	¥2.00	收藏 删除
☒	 新款女款 拼接 不规则露 长袖针织衫开衫 杏雨 紫色	¥4.00	- 1 +	¥4.00	收藏 删除
				店铺合计:	¥6.00
				商品总价 (不含运费) ¥ 6.00 元	
下一步, 填写核对购物信息					

图3-130 购物车列表

点击“下一步”按钮，使用预存款支付，如图3-131所示。

官方店铺				
	新款女款 拼接 不规则露 长袖针织衫开衫 杏雨 粉色	¥2.00	1	¥2.00
	新款女款 拼接 不规则露 长袖针织衫开衫 杏雨 紫色	¥4.00	1	¥4.00
买家留言:				
				运费: ¥0.00
				商品金额: ¥6.00
				本店合计: ¥6.00
☒ 使用预存款支付 (当前可用余额: ¥5.00)				
支付密码:				支付
订单总金额: ¥ 6.00 元				

图3-131 使用预存款支付

进入“下一步”按钮，因为预存款金额不足以支付本次订单，所以还需要继续支付所差的金额，如图3-132所示。

请您及时付款，以便订单尽快处理！在线支付金额：¥1.00

订单号	支付方式	金额	物流
60000000000003601	在线支付（已使用预存款部分支付，支付金额 ¥ 5.00）	¥6.00	快递

支付选择

确认提交支付

图3-132 订单混合支付

支付成功后，查看一下预存款，如图3-133所示。

可用金额：0.00 元 冻结金额：0.00 元			
创建时间	可用金额(元)	冻结金额(元)	变更说明
2014-04-09 10:28:46		-5.00	下单，支付被冻结的预存款，订单号: 60000000000003601
2014-04-09 10:26:43	-5.00	+5.00	下单，冻结预存款，订单号: 60000000000003601
2014-04-09 10:07:30		-5.00	提现成功，提现单号: 560450353146967007
2014-04-09 10:05:46	-5.00	+5.00	申请提现，冻结预存款，提现单号: 560450353146967007
2014-04-09 09:50:17	+10.00		充值，充值单号: 390450351816163007

图3-133 预存款日志列表

可以发现，预存款已经被成功扣除，日志显示当下单遇到预存款不足时，系统先冻结预存款，待差额成功支付后，再彻底扣除预存款。如果用户未支付差额而是选择取消订单，系统会将冻结的金额取消并增加到可用金额中。

3.12 本章小结

本章对电商平台系统主要通用模块的设计、实现、效果展示作了详细介绍，虽然是以在B2B2C模式下PHP+MySQL实现为例，但这些模块的绝大多数设计思路和处理方法也适用于其他模式。

第4章 CMS套件的设计与实现

CMS作为大多数网站的标准套件，在电子商务平台中也逐步发展起来，现在的商城不再是单一的销售商品，更是在资讯、社交等领域全面布局，逐渐成为以购物为中心的综合性门户网站。尤其是在地方性商城和行业性商城中，CMS更是成为不可或缺的重要组成部分。CMS的成功运营，不仅可以使网站成为地方或行业的资讯平台，更可以为网站带来巨大的流量，使网站成为地方或者行业的权威网站，不仅本身可以为网站带来客观的广告收入，而且可以更好地促进商城的销售。

我们经过深入调研为商城系统开发了CMS套件，在选择要开发的模块时，我们结合CMS和商城的特点开发了文章、画报、专题和首页编辑四个模块。与传统CMS系统相比，我们的CMS套件能够更好地与商城系统相结

合，提供更丰富的内容，包括商品信息、会员信息、店铺信息等，把原本分散的内容整合到一个平台上。在与淘宝等第三方平台打通后还可以为网站带来多样的运营模式，使网站的内容更加丰富。

4.1 CMS文章

文章是每个CMS不可或缺的基础模块，是构成CMS的主体内容。在传统CMS的基础上，结合商城CMS组件的特点，增加了标签和绑定相关商品功能。用户在浏览文章时可以根据不同标签选择不同类的文章，还可以根据绑定的商品直接购买文章中提到的相关商品。文章成为商城重要的营销手段，可以更方便地让商家推荐商品。买家也可以通过这些文章的推荐购买到自己喜欢的商品。

4.1.1 模块构成

文章模块主要由发布、审核、文章管理、分类管理等几部分组成。用户首先撰写文章绑定好相关文章和相关商品，提交后平台对用户的文章进行审核，在审核通过后将其发布到前台。用户在文章未最终完成时，可以先将其保存到草稿箱，待最终完成后再提交平台审核。对于已经发布的文章，平台可以对文章进行收回，而用户不能做任何修改。

4.1.2 设计思路

1. 文章发布主体选择

在发布主体的选择上，主要有平台发布、平台发布结合用户发布、完全用户发布三种方式。平台发布方式完全由平台提供资讯内容，用户无法发布，这种方式比较像传统的CMS系统，但是对于一个电商平台来说，缺少了用户的参与可用性将大打折扣。平台发布结合用户发布这种方式，看似比较完美，但是前后台都需要有完整的发布功能，业务逻辑重复，日后维护成本较高，而且发布主体不同带来的权限管理也比较复杂。对于完全用户发布这种方式，虽然平台无法直接发布文章，但是可以通过在前台注册一个平台账号的方式间接实现该功能，而且业务逻辑比较简单，前台只负责发布后台只负责审核，权限设置也相对简单

很多，虽然需要平台单独注册一个前台账号，但是维护成本大大降低，是性价比较高的一种实现方式。我们也是从早期平台结合用户发布的方式经过一些积累，最终选择了完全用户发布这种方式，期间删除了大量的冗余代码和复杂的权限检查，提高了程序的稳定性和可维护性。

2. 文章状态

对于文章状态，我们设置了草稿箱、待审核、已发布三个状态。草稿箱是用户开始编写文章但还未提交到平台的中间状态，这个状态主要是用户在完善文章。待审核是用户已经完成了文章的编写，提交给平台进行审核，审核通过后文章将出现在前台页面，如果审核失败，则退给用户进行完善。对于审核通过后的已发布文章，用户将不能做任何修改，如果需要修改，必须在平台将文章退回用户的草稿箱后，用户才

能再次进行编辑。这主要是考虑到，在文章被推荐到首页或者某个活动页面后，如果用户不通过平台直接修改或删除了自己的文章，会对平台设置的页面造成影响，所以发布后的文章只有平台能够对其进行操作。

3. 商品绑定

在与商城的结合上，我们开发了发布文章时可以绑定相关商品的功能，考虑到商品的多样性问题，绑定的商品可以是自建商城的也可以是其他平台的，比如淘宝、京东等。这样可以做到和大的电商平台差异化经营，同时也能相辅相成。在添加商品的方式上，我们需要一个统一的添加方式，经过各种方式的对比，最终选择通过商品链接的方式进行添加。不管商品是站内还是第三方平台，这样的方式都能够有统一的操作方式，带给用户一致的操作体验，而且也方便

日后对支持的第三方平台进行扩展，极大地增强了功能的灵活性。其实这种方式的实现比较简单，通过用户输入的连接判断不同的商品来源，如果是本地商品，则调用本地接口读出商品信息，如果是第三方平台，则需要绑定相应的开发账号，然后调用第三方平台提供的接口读取相应数据。在这里不管是本地数据还是第三方平台数据，读取后都要对数据进行初步处理，保存为相同的格式，这样才能保证数据处理方式的一致性，我们只保留了一些共性数据，比如商品名称、商品图片、商品价格、购买链接等，对于不同平台的差异数据都没有使用。这样以后如果增加其他平台的支持，也都能够用一些共性的数据来展示，方便扩展和数据格式的统一。

4. 图片上传方式

图片上传方式有很多种，比如使用FileUpload、Flash、XMLHttpRequest2等。这些方式各有各的优点和不足：FileUpload一次只能上传一个文件而且需要刷新页面（Ajax依靠的XMLHttpRequest其实是不支持文件上传的，需要用一个iframe来模拟）；Flash依赖第三方插件而且现在Flash插件越来越不稳定；使用XMLHttpRequest2上传图片比较完美，但是需要比较新的浏览器才能支持（IE10以上版本才支持，Firefox、Chrome都没有问题，低版本IE是开发人员永远的痛）。

CMS需要大量的多图上传，而且需要采用Ajax方式不能刷新页面。我们在开发时经过测试，首先放弃了Flash方案，未来Flash的使用可能会越来越少，而且性能和安全问题越来越多。然后我们采用了一个折中方案，在低版本IE上使用隐藏IE的方式进行上传，不过每次只能上传一张图片，上传多张图片时需要选择

多次。目前在浏览器上（IE10以上、Firefox、Chrome）使用XMLHttpRequest2方案可以同时上传多张图片。

4.1.3 代码实现

先来实现文章图片上传功能，具体包括前台JavaScript调用接口异步完成图片上传，后台对上传的图片进行验证和裁切，上传成功后前台显示成功上传后的图片，代码如代码清单4-1所示。

代码清单4-1 CMS文章图片上传

```
// 文章图片上传javascript
$("#article_image_upload").fileupload({
    dataType: 'json',
    url: "index.php?
act=publish&op=article_image_upload",
    add: function(e,data) {
        // 遍历需要上传的图片，分别进行上传
        $.each(data.files, function (index, file) {
            // 开始上传时显示占位图片
            .....
        });
        data.submit();
    },
    done: function (e,data) {
        result = data.result;
        if(result.status == "success") {
            // 上传成功后显示图片
            .....
        } else {
            // 上传失败显示错误信息
            $image_box.remove();
        }
    }
});
```

```

        showError(result.error);
    }
}

});
// 文章图片上传PHP
public function article_image_uploadOp() {
    $data = array();
    $data['status'] = 'success';
    if(!empty($this->publisher_name)) {
        if(!empty($_FILES['article_image_upload']
['name'])) {
            $upload = new UploadFile();
            // 图片上传目录设置
            $upload->set('default_dir',
ATTACH_CMS_ARTICLE);
            // 自动裁剪图片设置
            $upload->set('thumb_width','1024,240');
            $upload->set('thumb_height','50000,5000');
            $upload->set('thumb_ext',
'_max,_list');
            // 上传图片
            $result = $upload-
>upfile('article_image_upload');
            // 上传失败处理
            if(!$result) {
                $data['status'] = 'fail';
                $data['error'] = '图片上传失败';
            }
            // 处理返回数据
            $data['file_name'] = $upload->file_name;
            $data['file_path'] = $this->attachment_path;
        }
    } else {
        $data['status'] = 'fail';
        $data['error'] = Language::get('no_login');
    }
    self::echo_json($data);
}

```

接下来实现文章绑定商品功能：通过用户输入的商品链接，在后台调用对应的接口读取商品信息，这些信息通过处理后显示到页面上。如果用户输入了非法链接，会提示错误信息。代码如代码清单4-2所示。

代码清单4-2 CMS文章绑定商品

```
// 绑定商品javascript
$("#btn_goods_search").click(function(){
    if($("#article_goods_list li").length < 3) {
        var search_type = $("
[name='goods_search_type']:checked").val();
        var search_keyword =
$("#goods_search_keyword").val();
        $("#goods_search_keyword").val("");
        if(search_keyword != "") {
            if(search_type == "goods_url") {
                var url =
encodeURIComponent(search_keyword);

$.getJSON("index.php?act=api&op=goods_info_by_url", { url:
url}, function(data){
    if(data.result == "true") {
        // 添加成功后将商品内容插入页面
    } else {
        // 如果出现错误弹出错误信息
        alert(data.message);
    }
    });
} else {
    $("#div_goods_select").load("index.php?
act=api&op=goods_list&search_type="+search_type+"&search_key
word="+search_keyword);
}
}
```

```

    }
    });
// 绑定商品php代码
    public function goods_info_by_urlOp() {
        $url = urldecode($_GET['url']);
        if(empty($url)) {

self::return_json(Language::get('goods_not_exist'),
'false');
        }
        $model_goods_info = Model('goods_info_by_url');
        $result = $model_goods_info-
>get_goods_info_by_url($url);
        if($result) {
            self::echo_json($result);
        } else {

self::return_json(Language::get('goods_not_exist'),
'false');
        }
    }
// 商品链接处理方法
    public function get_goods_info_by_url($url) {
        $url_host_name = self::get_url_domain($url);
        // 处理商城域名
        $store_host_name =
self::get_url_domain(SHOP_SITE_URL);
        switch ($url_host_name) {
            case 'tmall.com':
            case 'taobao.com':
                // 判断是否开启淘宝接口支持
                if(C('taobao_api_isuse')) {
                    // 调用淘宝接口
                    return
self::get_taobao_goods_info_by_url($url);
                } else {
                    return FALSE;
                }
                break;
            case $store_host_name:
                // 调用商城接口
                return self::get_store_goods_info_by_url($url);
            default:
                return FALSE;
                break;
        }
    }

```

}
}



4.1.4 功能体验

在文章发布页面完成填写文章内容、上传文章图片、绑定文章商品等操作，还可以在上传的图片中选择一张作为文章的封面图，这张图片将出现在文章列表中，如图4-2所示。

在文章列表页面展示文章列表，包括文章标题、文章简介、文章图片等内容，该页面右侧会显示文章标签和推荐文章，如图4-2所示。

在文章详细页面显示文章的正文和相关推荐商品，右侧显示热门文章标签和推荐文章，如图4-3所示。用户还可以点击“分享”按钮将自己喜欢的文章分享到微博等SNS平台。

分类	数码	 封面		图片上传  插入 封面 删除
标题	六问苹果iPhone 5c：改变在哪？该不该买？ 标题限定在24个字以内。			
短标题	 标题限定在12个字以内。			
摘要	苹果在一次次发布会上推出两款iPhone产品，由于发布之前曾一度传出苹果会推出廉价版iPhone的传闻，导致很多钱袋不是很充裕的人开始期待能花很低的价格用上新的iPhone，iPhone 5c 4488元的高昂价格再一次打消了很多人的积极性。 摘要限定在140个字以内。			
文章正文	121/140 			

图4-1 CMS文章发布表单



图4-2 CMS文章列表



图4-3 CMS文章详细页

4.2 CMS画报

画报是以图片为主的媒体形式，用户可以上传一组图片，图片的内容可以是相同的商品或相同的主题，在前台以幻灯片的形式展现给用户。相比以文字为主的文章，画报的优势是更加直观，俗话说一图胜千言，相比阅读大段的文字，直接看图更能让用户接受，用户也更愿意通过图片来获取信息。而且对于展示商品来说图片也比文字更加合适，商家可以把各种商品的组合通过图片更好地展示给用户，对于咨询发布以商品推介为主的商城来说，图片的展示相比文字更有说服力。

4.2.1 模块构成

从整体上来说，画报和文章类似，只不过内容从文字变成了图片。画报模块也主要由发布、审核、管理、分类管理等几部分组成。

4.2.2 设计思路

1. 功能设计思路

在开发早期做功能规划的时候，考虑到文章这种传统类型的一些不足，比如文字过多造成阅读困难，展示也不够直观，虽然也可以插入图片，但是图文混排以后还是没办法突出重点。我们考虑到是不是可以做一个更加直接的展示方式，经过对比我们决定做一个以幻灯片为展现形式，完全由图片构成的文章类型。没有大段的文字，主要以图片为主，每个图片可以绑定相关的商品，可以辅助不多的说明文字。用户可以对自己的商品进行组合，然后拍成照片发布上来进行展示，非常简单和直观。对于一个以购物为主的网站90%以上的流量都是图片，图片是最有说服力的表示形式。普通用户也更容易接收图片，可以很直观地

看到各种商品的组合和搭配效果，如果喜欢还可以点击对应商品的购买链接直接进行购买。

2. 图片延时加载

画报由图片组成，少则几张十几张多则可能有几十张。而幻灯片需要等图片加载完成后才能执行JavaScript代码，如果图片过多或者加载比较慢，页面可能会出现延迟。因此我们使用了图片延迟加载技术，也就是说只加载需要当前显示的图片，当前不显示的图片先用一个空白的占位图片进行代替，当需要加载的时候再修改img标签的属性去加载真正的图片。这样可以大幅提高页面加载速度，而且由于并不是每个用户都会从第一张一直看到最后一张，所以也可以减少很多无用的图片请求。

4.2.3 代码实现

这里实现的是图片延迟加载功能：图片开始并不显示，等图片出现到显示位置，用JavaScript替换图片地址，完成图片加载，如代码清单4-3所示。

代码清单4-3 CMS画报图片延迟加载

```
// 图片延迟加载jQuery扩展
(function($) {
    $.fn.nc_lazyload = function() {
        var lazy_items = [];
        this.each(function() {
            if($(this).attr("data-src") !== undefined){
                var lazy_item = {
                    object: $(this),
                    url: $(this).attr("data-src")
                };
                lazy_items.push(lazy_item);
            }
        });
        var load_img = function() {
            var window_height = $(window).height();
            var scroll_top = $(window).scrollTop();
            $.each(lazy_items, function(i, lazy_item) {
                if(lazy_item.object) {
                    item_top = lazy_item.object.offset().top
- scroll_top;
                    if(item_top >= 0 && item_top <
window_height) {
                        if(lazy_item.url) {
```



```
lazy_item.object.attr("src",lazy_item.url);
    }
    lazy_item.object = null;
    }
    }
    });
};
load_img();
$(window).bind("scroll", load_img);
};
})(jQuery);
```

4.2.4 功能体验

前台画报列表页面用于显示画报的封面和画报内的图片数量，右侧显示推荐的画报，如图4-4所示。



图4-4 CMS画报列表

前台画报详细页面以幻灯片的形式显示，用户可以通过箭头查看下一张或者上一张图片，也可以直接

查看自己喜欢的图片，其右侧显示画报标签等信息，如图4-5所示。



图4-5 CMS画报详细页

4.3 CMS专题

专题是一个方便制作专题页面的组件。现在商城都需要针对不同活动制作相应的专题页面，这些专题页面需要有丰富的展现形式，同时由于时效性很强需要快速上线。在开发专题组件时，针对这些特点选择使用几个基础组件组合的方式，这样可以快速制作出丰富的专题。针对商城网站的特点，我们并没有开发过分复杂的专题功能，功能过分复杂全面，学习成本肯定会非常高，而且大多数传统CMS专题功能对商城来说是没有用处的。因此，我们开发的目标是提供相对较少的表现力很强的组件，方便制作出内容丰富的专题。

4.3.1 模块构成

专题可以对页面背景、背景平铺方式等进行设置。在专题内可以使用图片热点、通栏图片、商品模块等。图片热点就是在上传的一张图片上用鼠标选取热点区域并添加链接。通栏图片是上传一张图片并设置链接。商品模块是通过商品链接添加多个商品的一种展现形式。

专题编辑时可以做到所见即所得，编辑完成后可以进行预览，确认无误后发布到前台页面，在发布时可以选择作为CMS专题或者商城专题来发布。作为CMS专题发布时专题页面整体风格与CMS一致，作为商城专题发布时，专题整体风格与商城保持一致。专题页面发布后直接生成静态html文件，前台显示时直接加

载，不需要读取数据库等操作，可以有效地提高专题页面的整体性能。

4.3.2 设计思路

1. 背景设计思路

一个内容丰富的专题页面，需要背景参数的辅助，但是背景一般都需要使用CSS进行设置，如果统一设置参数，那么所有的专题页面就会千篇一律。因此我们为专题页面增加了设置背景图片、对齐方式、边距等多个背景参数的功能，对这些背景参数进行合适的组合可以制作出丰富的页面背景效果，使每个专题页面都可以有自己不同的背景风格。

2. 图片热点设计思路

在开发的时候，我们分析了专题的使用场景，也咨询了一些客户实际使用的需求。发现专题一般使用的时间比较短，只使用几天，但是时效性比较强，需

要快速上线，可能刚刚定下来的促销方案马上就需要制作专题页面并上线。在对比了各大网站的专题页面后，也印证了这一点，其中大部分专题页面都是使用图片热点的方式制作的，这种方式的优势是制作非常快，只需要制作好图片并设定好热点区域就可以了。如果使用裁图再制作html页面的方式，本身就需要更大的工作量，而且还要考虑不同浏览器的兼容性，制作完成还需要大量的测试工作，这样的速度肯定不能满足专题的时效性。

但是怎么方便地制作图片热点呢？传统的流程是美工制作好图片，然后交给Web开发人员使用专业的工具制作相应的页面并生成页面代码，代码再将图片交给网站编辑更新网站。这样的流程环节多效率比较低，如果需要修改又可能出现很多沟通问题。为此我们开发了一种简单的制作图片热点的组件，网站编辑只需要上传美工制作好的图片，然后用鼠标在图片上

直接选择合适的区域并设置好链接地址，就可以完成专题的制作，不需要专业的Web开发人员，也不需要使用专用工具。而且这种图片热点组件使用比较简单，学习成本很低，普通的网站编辑就可以轻松完成，而且修改也十分方便。

3. 专题的加载方式

每一个专题页面可能是网站的主推的活动，可能在短时间内有大量的访问，如果每次访问都从数据库读取数据，可能会给数据库带来很大的压力。因此我们的专题采用直接生成静态页面的方式，网站编辑在完成专题制作后直接发布生成静态页面，每次访问直接访问静态页面，执行效率能有大幅度的提升。

4. 专题的整体风格

在最初设计的时候，专题是CMS套件的一个组成部分，只能和CMS使用统一的整体风格。经过用户反馈，我们发现很多用户也需要在商城中使用专题页面，而且商城内的专题页面需要与商城风格统一，因此在后续的开发中我们针对这个需求开发了专题发布到商城的功能。这样制作出来的专题页面在发布时可以发布到，CMS也可以发布到商城中，而且可以随时切换，非常灵活，既满足了CMS专题新闻的需要，也满足了商城制作活动专题的需要。

4.3.3 代码实现

生成专题静态文件部分的代码，将专题页面直接生成静态HTML文件，如代码清单4-4所示。

代码清单4-4 CMS专题生成静态文件

```
// 获取专题详细内容
private function get_cms_special_detail($special_id) {
    $model_special = Model('cms_special');
    $special_detail = $model_special-
>getOne(array('special_id'=>$special_id));
    Tpl::output('special_detail', $special_detail);
    Tpl::showpage('cms_special.detail', 'null_layout');
}
// 生成静态文件
private function generate_html($special_id) {
    $html_path =
BASE_UPLOAD_PATH.DS.ATTACH_CMS.DS.'special_html'.DS;
    if(!is_dir($html_path)){
        if (!@mkdir($html_path, 0755)){

showMessage(Language::get('cms_special_build_fail'),'','','e
rror');

        }
    }
    ob_start();
    $this->get_cms_special_detail($special_id);
    $result =
file_put_contents($html_path.md5('special'.$special_id).'ht
ml', ob_get_clean());
    if(!$result) {
```

```
showMessage(Language::get('cms_special_build_fail'),'','','e  
rror');  
    }  
}
```

4.3.4 功能体验

在专题发布页面中按照表单内容填写，完成后发布，如图4-6所示。

* 标题

标题不能超过24个字

* 专题类型

资讯

资讯类型将出现在资讯频道内，商城类型将出现在商城内

* 专题封面图：

浏览...

请上传990像素x240像素的图片作为专题页面的封面图

专题页面背景设置

背景色即专题页面CSS属性中body{ background-color}*值，作为专业页面整体

背景图选择

浏览...

背景图即专题页面CSS属性中body{ background-image}*值，选择本地图片上

背景图填充方式

☐ 不重复 ☐ 平铺 ☐ x轴平铺 ☐ y轴平铺

背景图填充方式即专题页面CSS属body{ background-repeat}*值，选择不重复

专题正文内容：

内容块距顶部

0

像素

内容块距顶部高度即专题主体部分距离页面顶部的边距(margin-top)值，如果您

预览

编辑

图4-6 CMS专题发布表单

专题范例显示效果，用户可以通过简单的操作就可以制作出专业的专题页面效果，并且不需要太多的专业知识，如图4-7所示。

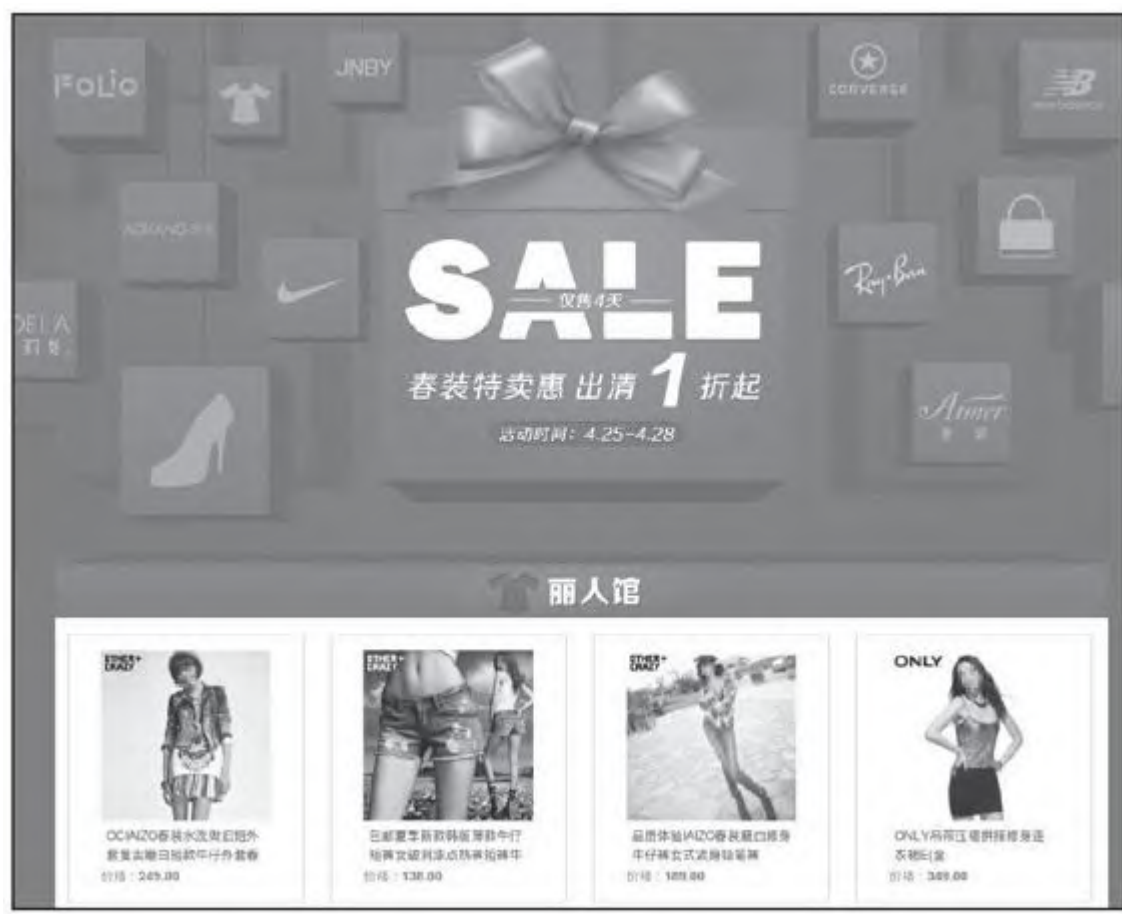


图4-7 CMS专题详细页

4.4 CMS首页编辑

首页是所有网站浏览量最大的地方，对于一个网站而言是最重要的地方，能否灵活地编辑首页内容，控制首页的展现形式是CMS首页编辑模块需要首要考虑的问题。我们在开发的时候为CMS首页编辑模块建立了丰富的布局、组件和模块，这样网站管理员可以利用这些组件快速制作出丰富的页面。针对首页高访问量的特点，采用后台编辑后直接生成静态html页面的方式对首页进行部署，访问首页时直接加载静态文件，不需要查询数据库和其他额外操作，极大地保证了首页的加载速度。

4.4.1 模块构成

首页由各个模块组成，可以自由添加多个模块。模块分为标准模块和自定义模块：标准模块是系统预先设置好的模块类型，布局 and 模块内容类型都是规定好的；自定义模块是用户可以自由设定的，用户可以自己选择布局 and 每个布局块所显示的内容类型。通过各个模块的组合可以制作出内容丰富的首页内容。

4.4.2 设计思路

1. 总体结构设计思路

在做功能分析的时候，对于首页的编辑功能进行了深入的分析，也对比了一些类似产品的实现方式。对于一些专业门户网站，首页布局和版块是固定的，不需要经常对布局进行修改，只需要更新内容即可，整体布局和风格可能几年才做一次比较大的改版，这样其实就不太需要强大的首页布局调整功能了，改版的时候专业的Web开发人员可以使用专业的工具进行重新开发，而且不受任何限制，任何想法都可以实现。

然而有一些专业的CMS程序又把首页编辑功能做得过于全面，想把所有专业开发工具的功能全部都实现，但是这几乎是不可能的，不仅功能无法实现，程序也十分复杂，维护成本非常高，而且出现问题的情

况也比较多，做一些复杂的操作对性能的影响也很大。更为重要的是操作起来十分复杂，普通用户很难学会，专业开发人员又更愿意用专业的Web开发工具，所以最终的结果是功能复杂，用户不会用，实现的功能基本没什么用处，用户想要的基本都没有，理想的效果做不出来，成为一个看着很美好实际用处不大的鸡肋功能，最后还需要专业开发人员去开发。在进行分析以后，我们决定采用一个折中的方式，在一定的规则下给用户一些自由选择的权利。最终目标是让一些非专业开发人员，可以方便快速地制作出内容丰富的首页来。我们采取的是搭积木的方式，每一个模块就相当于用户手中的积木，用户通过不同模块的组合和搭配，制作出各种风格的页面。早期版本用户只能使用系统预制的模块，后期我们又开发了自定义模块功能，使首页编辑功能更加强大灵活。其中，标准模块用户不能做任何修改，自定义模块用户可以自由进

行创建和删除，但是创建后不能做修改，而在删除后相对应已经使用的模块也会被同步删除。

2. 标准模块设计思路

在早期版本开发的时候，我们的第一个目标是能够让用户快速搭建出首页内容，因此我们先内置了几个固定的模块。这些模块的布局和内容形式都是固定好的，不过各种模块可以任意组合在一起。虽然灵活性受一些限制，但是对于大多数网站来说也是够用了，而且固定模块制作简单方便，很快就能学会。这些固定模块包括：综合模块、文章模块、通栏图片模块，其中综合模块内容比较丰富适合做首页的第一屏内容，两种不同风格的文章模块适合做主要新闻内容的填充，通栏图片模块比较适合做通栏广告。

3. 自定义模块设计思路

在完成了标准模块的开发之后，我们就开始规划自定义模块的开发工作。首先是自定义模块应该是什么样子的，哪些东西是是可以让用户自己选择的。经过权衡我们认为：首先模块内布局应该是可以用户自己选择的，比如是两栏、三栏还是通栏；其次每一栏具体放什么内容也应该可以用户决定，比如有的用户想左边放文章右边放商品，有的用户想左边放商品右边放店铺。

首先我们先实现布局方式的选择。如何让用户方便地选择布局方式，经过对比和测试我们还是决定内置几种常见的主流布局，让用户自己去选择，而不是让用户自己画或者输入宽度。由于每个块都可能放入不同的内容，而且不同的内容在不同的宽度下展示可能会有不同的效果，如果每个用户设定的宽度都不一样，都需要再单独调整样式，操作就会很复杂。经过选择我们为用户提供了通栏、两栏、三栏、多栏等8种

页面布局，能够满足用户的各种页面需要，即使用户有更多的要求，也可以方便地进行二次开发。

如果模块的布局只能使用固定的内容，那么自定义也就失去了实际的意义，所以我们让每个自定义模块的区域都可以在建立的时候选择不同的内容，用户可以建立多个结构相同的自定义模块，但是每个块的内容是不同的。这相当于自己建立多个自定义模块的模板，在建立好自己的模板库后，在使用时添加自己的自定义模块，然后填充内容，就完成了首页的编辑。为了方便区分，每个自定义模块的模板都可以设定自己的名称。

4. 操作方式设计思路

在操作方式上，我们在模块列表中使用了示意图的方式进行展示，这样比较方便排序和对模块进行设置。在模块内我们采用了所见即所得的可视化编辑方

式，这样更直观地方便用户进行操作。用户在设置完成以后可以先进行预览，然后进行修改，在内容最终确定以后，点击“生成”按钮直接生成首页的静态文件。

4.4.3 代码实现

添加和删除模块可以完成首页模块的添加和删除操作，同时对操作进行权限验证，如代码清单4-5所示。

代码清单4-5 CMS首页编辑模块添加、删除

```
/**
 * 模块添加
 */
public function add_page_moduleOp() {
    $data = array();
    $data['result'] = TRUE;
    if(empty($_POST['module_name'])) {
        $data['result'] = FALSE;
        $data['message'] = '参数错误';
    } else {
        $model_index_module = Model('cms_index_module');
        $new_module = array();
        $new_module['module_title'] =
$_POST['module_title'];
        $new_module['module_name'] =
$_POST['module_name'];
        $new_module['module_type'] =
$_POST['module_type'];
        $new_module['module_style'] = 'style1';
        $new_module['module_view'] = 1;
        $result = $model_index_module-
>save($new_module);
        if($result) {
```

```

        $data['module_id'] = $result;
        $data['module_style'] =
$new_module['module_style'];
        $data['module_view'] =
$new_module['module_view'];
    } else {
        $data['result'] = FALSE;
        $data['message'] = '添加失败';
    }
}
echo json_encode($data);
}
/**
 * 模块删除
 */
public function drop_page_moduleOp() {
    $data = array();
    $data['result'] = TRUE;
    $module_id = intval($_POST['module_id']);
    if(empty($module_id)) {
        $data['result'] = FALSE;
        $data['message'] = '参数错误';
    } else {
        $model_index_module = Model('cms_index_module');
        $result = $model_index_module->drop(array('module_id'=>$module_id));
        if(!$result) {
            $data['result'] = FALSE;
            $data['message'] = '删除失败';
        }
    }
    echo json_encode($data);
}

```

4.4.4 功能体验

在首页编辑操作页面，用户从右侧选择模块添加到左侧首页列表。还可以点击自定义模块标签，选择或者添加自定义模块，如图4-8所示。

在添加自定义模块时，首先选择模块布局，目前有8种布局形式，可以满足大多数的需要，如图4-9所示。

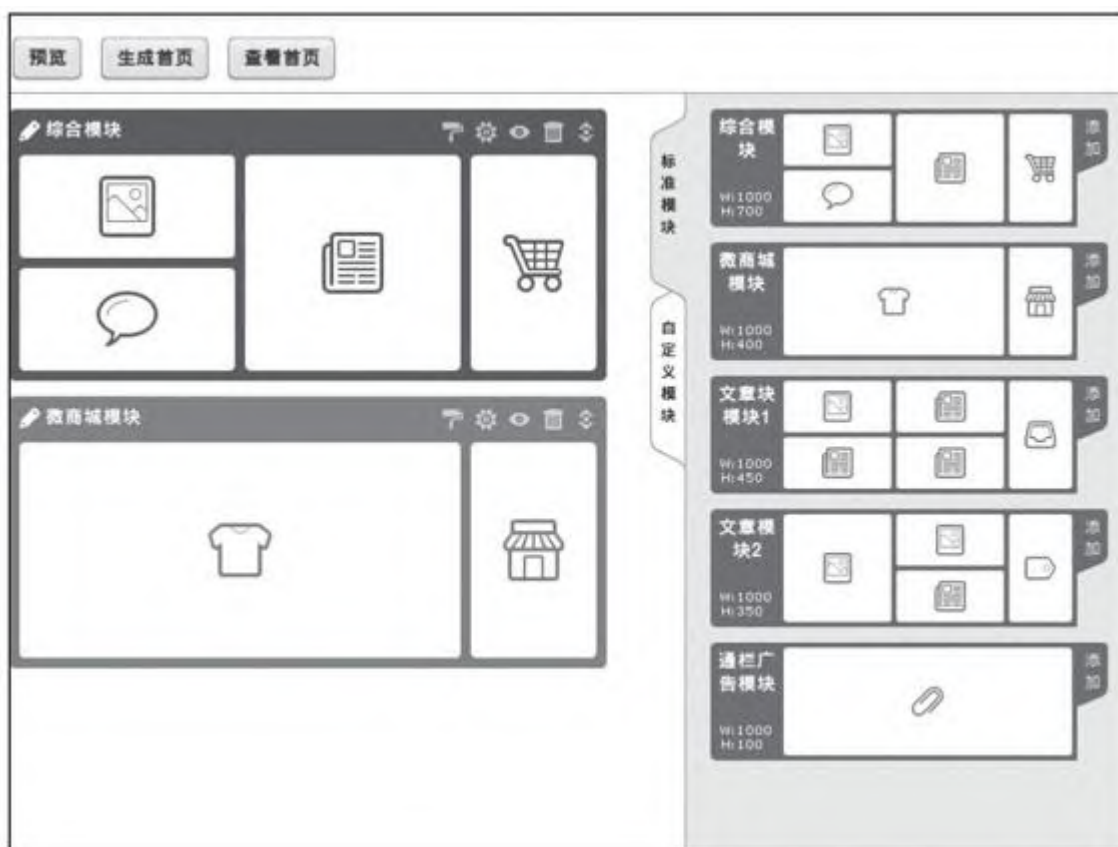


图4-8 CMS首页编辑操作页面

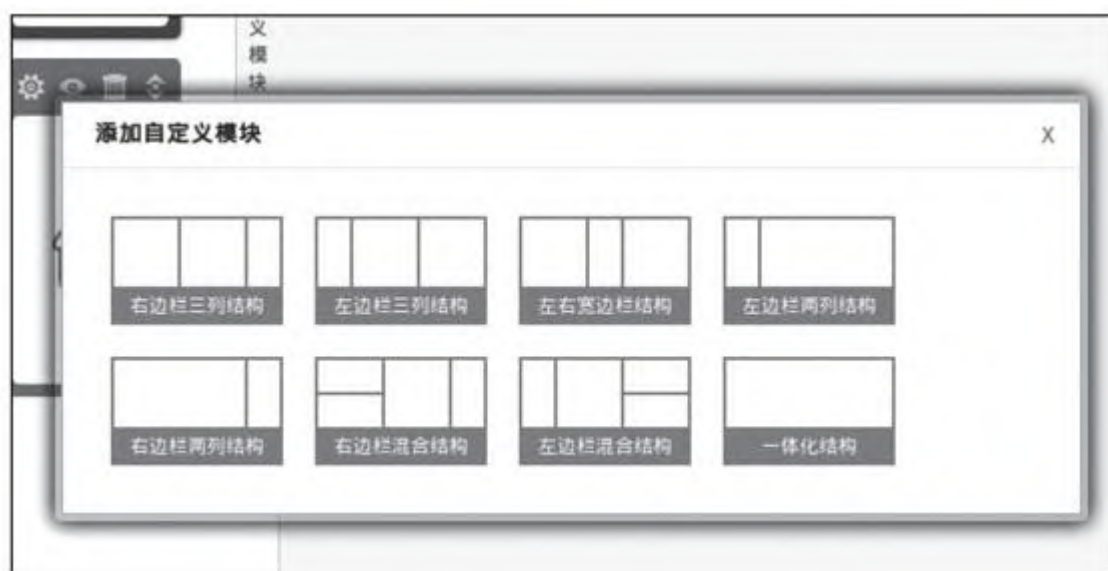


图4-9 CMS首页编辑自定义布局选择

在为自定义模块选择模块内容时，用户可以根据自己的需要任意组合，每一个块内都可以根据实际需要放置任何内容，如图4-10所示。



图4-10 CMS首页编辑自定义布局内容选择

接下来体验文章模块的可视化编辑，用户点击对应块的“编辑”按钮，可以完成对该块内容的编辑，在弹出的编辑窗口里对不同的内容进行添加修改，如图4-11所示。



图4-11 CMS首页块编辑

首页展示最终效果，用户只需要上传图片编辑文字就可以制作出专业的站点首页，而且展现形式和内容十分灵活，如图4-12所示。



图4-12 CMS首页最终效果

4.5 其他功能

4.5.1 评论

评论是CMS的重要功能之一，不仅可以增加用户的参与度，还可以为用户提供一个交流的途径，而且一个精彩的评论还可能为文章增色不少。在开发评论功能的时候，我们采用了现在比较流行的盖楼模式，这种模式类似早期的论坛，用户可以针对评论再进行评论，然后按照评论的顺序显示。这样的优点是可以在评论之间建立联系，相关的评论在一起，增加了互动性和用户的参与度。不过，普通的评论都是平级的，都是针对文章本身的，评论之间很难建立联系，互动性和讨论的效果大打折扣。

评论作为一个基础功能，文章和画报都可以使用，但必须是登录用户才可以发表评论，未登录的时候会提示需要登录，这样可以在一定程度上限制垃圾信息的产生。而且平台可以设置是否开启全站的评论功能，或者哪些文章可以评论哪些不可以评论，还可以在后台集中管理评论，针对用户和关键字进行搜索，对违规评论进行删除，极大地方便了平台的管理工作。

对于垃圾信息的处理，又是一个比较复杂的话题，涉及自动学习，关键字匹配和智能过滤等方面，这需要结合自己的实际需求有针对性地进行二次开发，这里不做深入讨论。在数据量不大的时候，可以暂时使用人工审核的方式。如图4-13所示。

最新评论

狐狸糊涂

发表于2014-01-10 10:17:39

狐狸糊涂1

必须土豪金

回复

狐狸糊涂2

自己顶自己呀

二楼定律怎么玩？

顶[0] 回复

狐狸糊涂

发表于2014-01-10 10:17:28

狐狸糊涂1

必须土豪金

自己顶自己呀

顶[0] 回复

图4-13 CMS评论

4.5.2 心情

心情是评论的一种补充，有时人们看了文章后有一些想法，但是又不想写一些文字或不知道有什么可以写的，就可以在各种心情图标中选择一个，或者支持或者反对或者无所谓，来表达自己的一个态度。这是一种比较简单方便的评论文章的方式，如图4-14所示。



图4-14 CMS心情

4.5.3 分享

分享是通过微博等接口，把自己喜欢的文章、画报分享出去，这样可以在更多的地方进行推广，扩大网站的影响力。用户在绑定相应的微博账号后，看到喜欢的文章可能想告诉自己的朋友，这时可以点击“分享”按钮，填写自己的推荐内容，选择要分享的第三方平台，点击“分享”按钮后程序会调用相应的接口把内容、图片和链接等信息自动发送出去，发送的同时还会同步到商城的SNS，出现在用户的SNS个人主页中，如图4-15所示。



图4-15 CMS分享

4.5.4 导航管理

平台可以自定义CMS的主导航菜单，在文章、画报、专题的基础上还可以添加其他站点的链接，比如链接到商城或者其他的自有站点，使各个功能相对独立的站点成为一个相互联系的整体，同时也方便用户在各个功能站点之间跳转。

4.6 本章小结

CMS模式是目前门户型电商运营模式之一，本章介绍了CMS的一些设计思路与处理方式，希望可以抛砖引玉，对大家有所启发。

第5章 社交平台的设计与实现

近年来，电子商务网站的社区化成为互联网行业热议的话题。从垂直媒体、社交网站、微博等利用流量优势为电子商务企业提供营销服务，到电子商务网站自行开设博客、微博、论坛等活跃新老客户，再到美丽说、蘑菇街等专门定位于购物分享社区的出现，社交化电子商务的形式多种多样。

对于商家和平台而言，电子商务社交化可以认为是通过分享、交流、讨论、互动等社交化的元素吸引用户购买的过程。从消费者的角度来看，社交化电子商务体现在消费者购物前的店铺选择、商品比较等，体现在购物过程中通过IM、论坛等与电子商务企业间的交流和互动，也体现在购买商品后消费评价及购物分享等。

ShopNC商城系统对电子商务网站的社交化进行了深入的探索，开发出SNS模块和圈子套件。

5.1 SNS模块

制作SNS模块的目的就是借用消费者群体的智慧，以用户购物的亲身体验和用户间的分享交流来提供更多对消费者有价值的商品信息，从而转化为消费行为，增加用户黏度，提高购买转化率。

5.1.1 模块构成

会员可以在SNS模块中分享心情、分享喜欢的商品和喜欢的店铺。关注他人并查看、转发、评论他人的动态，把喜欢的图片上传到相册，分享给大家看。

添加站外分享功能：在店铺页、商品页等与商品和店铺有关的列表页面添加“分享”按钮，可以把商品或店铺分享到站内SNS或站外。

独立的个人中心：每个会员都有自己用于展示分享过的动态，分享、喜欢过的商品，分享过的店铺、上传的相册。个人中心也是向他人展示的平台，其他用户可以浏览相关的信息。

多种关注他人方法：用户可以通过名称、所在地、年龄段、性别、兴趣标签等方式查找用户。

5.1.2 设计思路

在这里主要介绍一下分享和相册的设计思路。

1. 分享设计

分享是把140个字符的简短内容广播给关注自己的好友，主要分为四个部分：发表功能、评论功能、转发功能、删除动态。如图5-1所示为会员动态的操作流程。

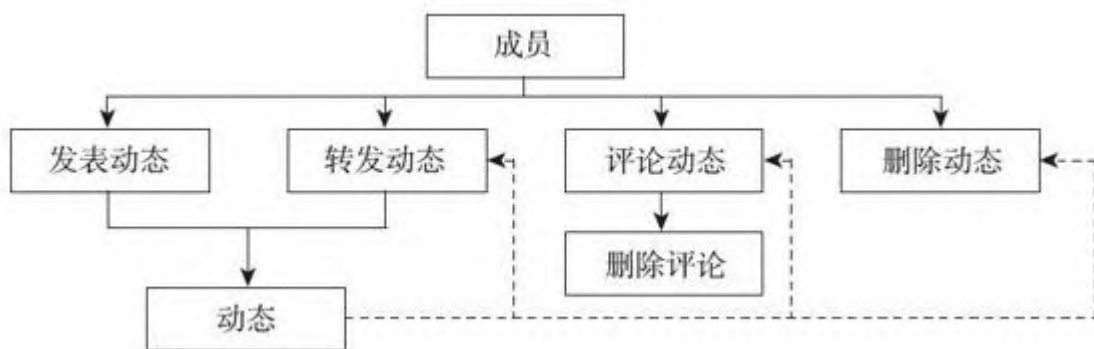


图5-1 会员动态的操作流程

由于SNS是基于商城开发的社交软件，需要考虑SNS怎样与商城相结合，所以我们把商城中最重要的商品和店铺与SNS结合起来，让用户可以把自己喜欢的商品和店铺分享给粉丝。其他用户在浏览动态的时候直接可以看到分享的商品和店铺，点击后直接跳转到相关页面，从而增加商品和店铺的访问量，产生交易。

2. 相册设计

相册是用户的一个收藏、展示图片的空间。利用相册用户可以实现创建相册、编辑相册、删除相册、上传图片、删除图片等功能。如图5-2所示为用户对相册的操作及流程。

在平台上可以设置允许用户上传图片的最大数量、删除图片功能。考虑到图片需要在不同的地方以不同的大小来展示，在上传图片的时候生成了几个图片的缩略图，用来供不同的地方使用。

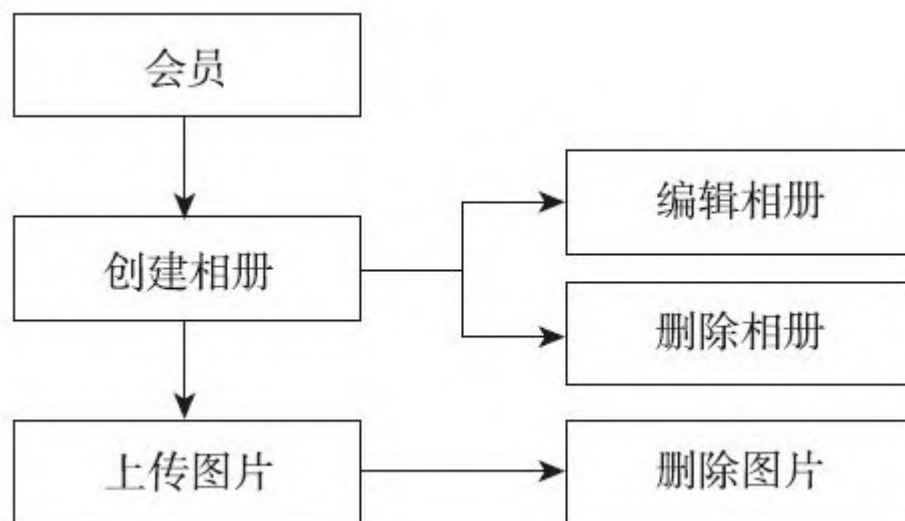


图5-2 用户对相册的操作及流程

5.1.3 代码实现

这里只介绍一下功能比较复杂的分享动态的代码实现。考虑系统的易用性，为了提高用户体验，在制作分享功能的时候多使用AJAX异步提交。

代码清单5-1 分享动态的JavaScript部分

```
//提交分享心情表单
$("#weibobtn").bind('click',function(){
    // 首先进行表单验证，验证分享内容是否超过140个字符
    if($("#weiboform").valid()){
        var cookienum = $.cookie(COOKIE_PRE+'weibonum');
        cookienum = parseInt(cookienum);
        if(cookienum >= max_recordnum &&
$("#weiboseccode").css('display') == 'none'){ // 验证是否需要
添加验证码
            //显示验证码
            $("#weiboseccode").show();

$("#weiboseccode").find("#codeimage").attr('src','index.php?
act=seccode &op=makecode&nchash=<?php echo
$output['nchash'];?>&t=' + Math.random());
        }else if(cookienum >= max_recordnum &&
$("#captcha").val() == ''){ // 如果没有填写验证码，弹出错误提示
            showDialog('<?php echo $lang['wrong_null'];?>');
        }else{ // 用AJAX方式提交表单
            ajaxpost('weiboform', '', '', 'onerror');
            //隐藏验证码并且重置相关信息（验证码图片和验证码输入框内
容）
            $("#weiboseccode").hide();
```

```
$("#weiboseccode").find("#codeimage").attr('src','');
    $("#captcha").val('');
    }
    }
    return false;
});
```

使用AJAX方式提交表单后，用PHP处理，如代码清单5-2所示。

代码清单5-2 分享动态的PHP部分

```
// 添加SNS分享心情
public function addtraceOp(){
    // 对提交的数据进行必要的验证
    $obj_validate = new Validate();
    $validate_arr[] = array("input"=>$_POST["content"],
    "require"=>"true", "message"
    =>Language::get('sns_sharemood_content_null'));
    $validate_arr[] = array("input"=>$_POST["content"],
    "validator"=>'Length',
    "min"=>0, "max"=>140, "message"=>Language::get('sns_content_beyond'));
    $obj_validate -> validateparam = $validate_arr;
    $error = $obj_validate->validate();
    if ($error != ''){
        showDialog($error, '', 'error');
    }
    //发帖数超过最大次数时出现验证码
    if(intval(cookie('weibonum'))>=self::MAX_RECORDNUM){
        if
        (!checkSeccode($_POST['nchash'], $_POST['captcha'])){
            showDialog(Language::get('wrong_checkcode'), '', 'error');
        }
    }
}
```

```

        //查询会员信息
        $member_model = Model('member');
        $member_info = $member_model->
>infoMember(array('member_id'=>"
{$_SESSION['member_id']}",'member_state'=>'1'));
        if (empty($member_info)){

showDialog(Language::get('sns_member_error'),'','error');
        }

        // 提交数据
        $tracelog_model = Model('sns_tracelog');
        $insert_arr = array();
.....
.....
        $result = $tracelog_model->tracelogAdd($insert_arr);
        if ($result){
            //建立cookie
            if (cookie('weibonum') != null &&
intval(cookie('weibonum')) >0){

setNcCookie('weibonum',intval(cookie('weibonum'))+1,2*3600);
//保存2小时
            }else{
                setNcCookie('weibonum',1,2*3600);//保存2小时
            }
            $js = "var obj = $('\"#weiboform\").find(\"
[nc_type='formprivacytab']\");$(obj).find('span').removeClas
s('selected');$(obj).find('ul li:nth-
child(1)').find('span').addClass('selected');"
            $js .=
"$(\"#content_weibo\").val('');$(\"#privacy\").val('0');$('#
friendtrace').lazyshow({url:\"index.php?act=
member_snsindex&op=tracelist&curpage=1\",'iIntervalId':true}
);"
            showDialog(Language::get('sns_share_succ'),'','succ',$js);
        }else {

showDialog(Language::get('sns_share_fail'),'','error');
        }
    }
}

```

5.1.4 功能体验

首先介绍一下分享商品的具体流程，进入商品详细页在“购买”按钮右侧找到“分享”按钮并点击，弹出的分享框如图5-3所示。



图5-3 商品页分享框

输入内容以后就可以把这个商品分享给自己的粉丝了。在图5-3的中部找到“同时分享到：”，单击其后的图片，这样在分享的同时就会把内容分享到相关的网站中。分享完成后进入个人中心就能看见刚刚发布的动态了，如图5-4所示。



图5-4 分享展示

在图5-4右上侧可以看见一个垃圾箱图标，点击它就可以删除动态了。在垃圾箱一侧的下方可以看到“转发”和“评论”按钮，点击“转发”按钮后在该条动态的下方就会出现一个输入框，输入内容后点击“转发”，效果如图5-5所示。



图5-5 转发展示

评论的功能和转发相似就不再介绍了。

5.2 圈子（讨论组）套件

俗话说“物以类聚，人以群分”，人们在沟通过程中基于相同的兴趣爱好自发地聚集到一起，广泛而深入地谈论某个话题，这样一个简单的圈子就建立起来了。人们可以在圈子中分享商品，晒自己购买过的商品的照片，探讨大家感兴趣的话题，发起一个小的投票调查等。圈子的功能有话题、有投票、有讨论、有分享、有成员关系、有人与人之间的沟通等。

圈子是论坛的轻量级产品，没有论坛臃肿的功能，却方便了站长的管理。

5.2.1 圈子特点

(1) 创建自己的圈子

每一名注册会员都有权利创建或加入一个圈子。创建圈子需要选择分类，自定义圈子名称、上传圈子LOGO、填写内容介绍和圈子公告等。

创建后的圈子由后台管理审核通过后即可使用。

圈子（讨论组）的形成是门户网站不可或缺的一部分，它可以大大地聚集人气，提高互动性与会员粘合度。

(2) 独立的前台管理中心

每个圈子都有独立的管理中心，圈主可对自己的圈子进行公告、LOGO更改、会员加入审批等基础操

作，也可将圈内好友提升为管理员，并可将表现积极的会员加为明星圈友，对违规者进行禁言、删除等操作。可设定圈内话题类型分类，便于圈内文章索引，还可添加友情圈子方便互访交流。

（3）主题列表多形式展示

圈子默认的展示形式为图文主题模式。主题列表让浏览者对帖子内容一目了然，还可在列表页面中展开主题内容，直接浏览并进行快速回复，具有便捷、直观，极具“微时代”特质。圈子也采用了标题列表形式供习惯使用传统论坛的浏览者选择。多形式列表展现极大程度地满足了各年龄层面使用者的浏览习惯。

5.2.2 设计思路

圈子主要由成员和话题构成。

首先创建圈子，用户在进入圈子首页后可以直接点击“创建圈子”按钮。圈子的创建流程如图5-6所示。

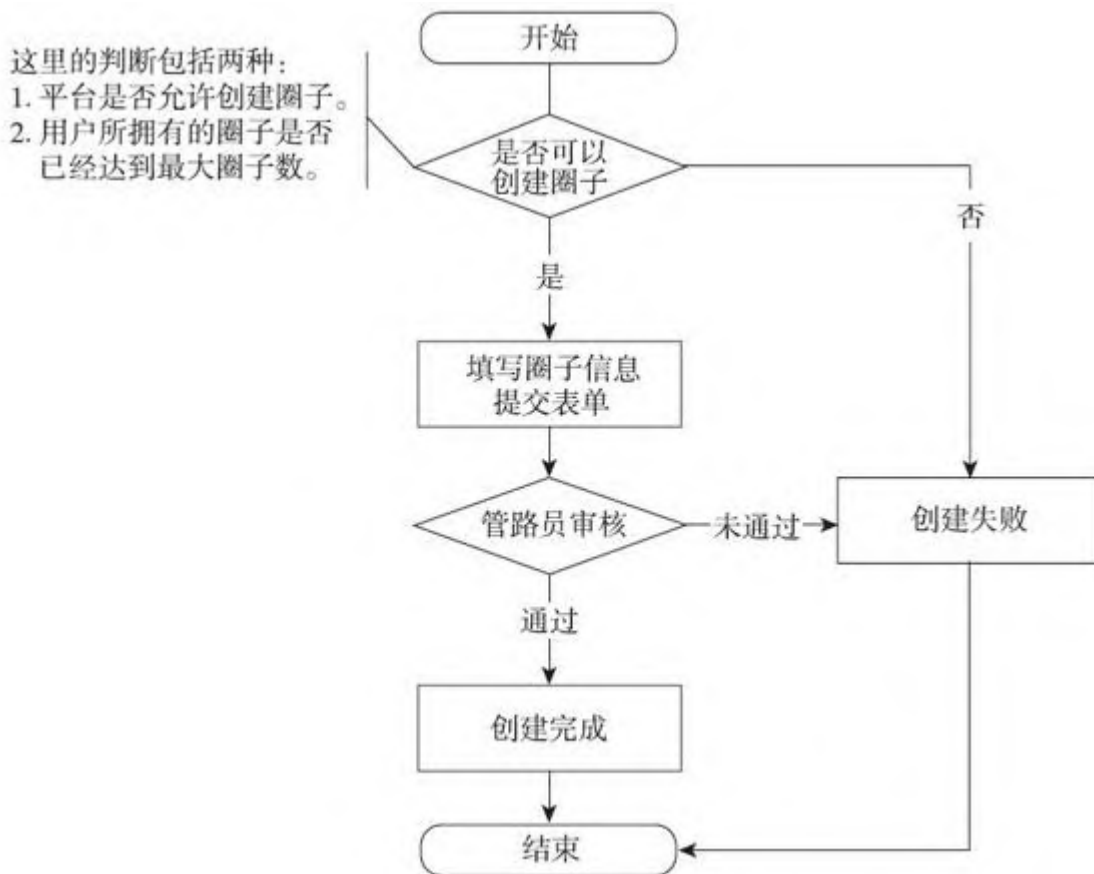


图5-6 创建圈子流程图

在圈子创建成功后，其他成员就可以加入圈子了。成员申请加入圈子的流程如图5-7所示。

加入成功后就可以发表或回复话题了。发表和回复流程都需要进行两个验证，如图5-8所示。

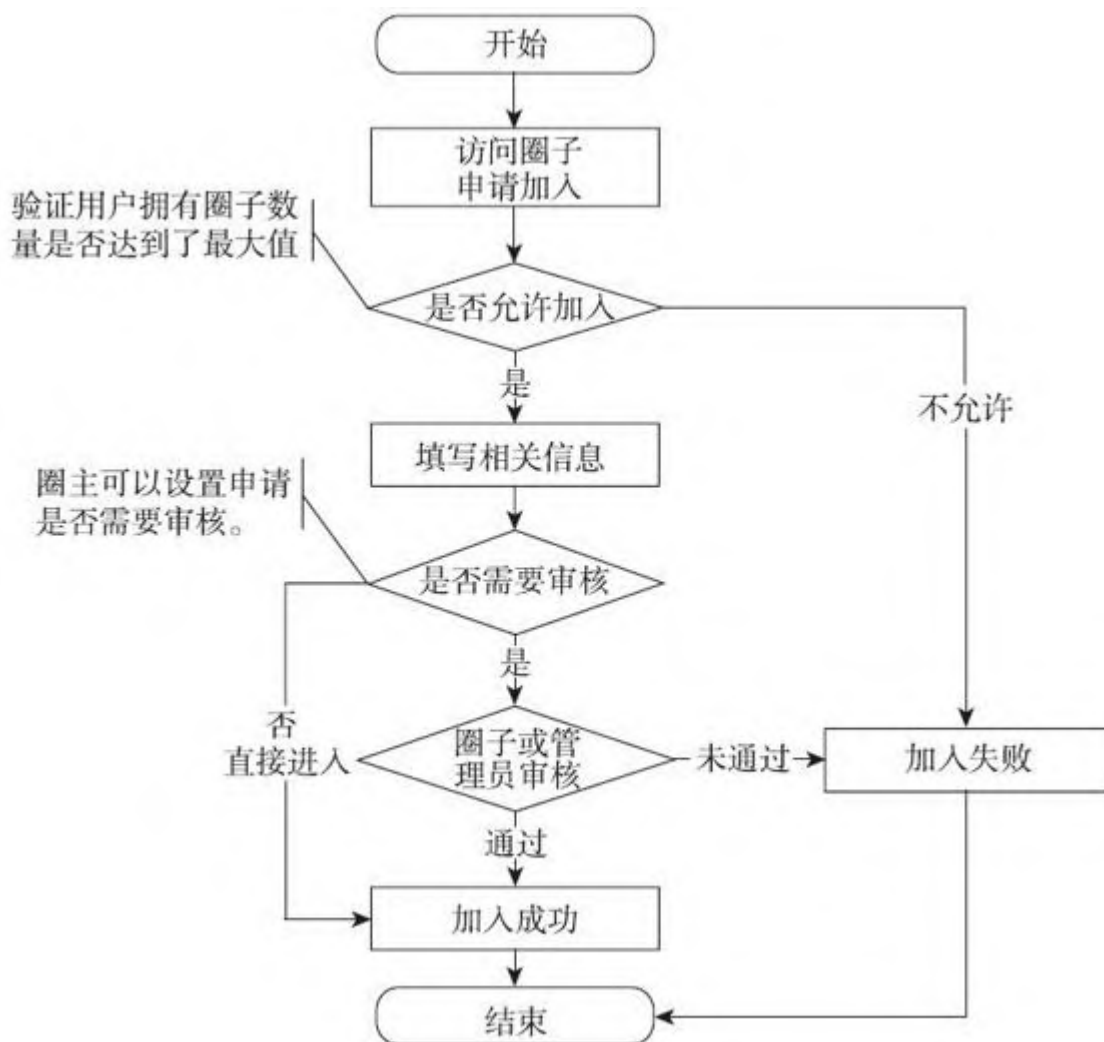


图5-7 申请加入圈子流程图

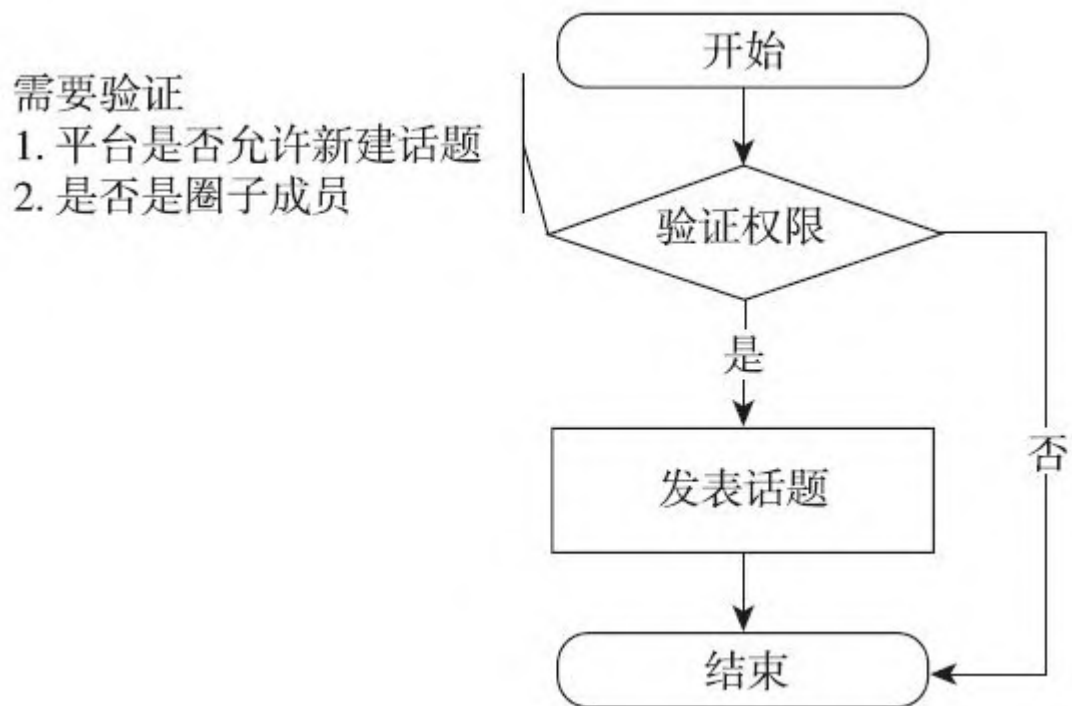


图5-8 发表话题流程图

1. 成员设计

(1) 超级管理员

超级管理员拥有对所有圈子的控制权限，可以直接对任意圈子的话题/回复进行管理。如，禁言/解禁

圈子成员，加精/取消、置顶/取消、关闭/开启，编辑、删除话题，编辑、删除回复，等等。

（2）圈主

圈主是圈子的创建者，拥有圈子的最高权限。可以在前台直接对话题/回复进行加精、置顶、编辑和删除等操作，也可以进入圈子管理中心对圈子进行操作，如圈子的基本设置、成员的管理（添加/删除管理员、禁言/解禁、设置/取消明星圈友、审核/删除成员）、头衔设置、分类管理、举报的管理、审核申请管理员、友情全局的管理。

（3）管理员

管理员拥有对圈子的部分权限，可以在前台直接对话题/回复进行编辑和删除操作，也可以进入圈子管理中心管理成员的审核申请和举报。

(4) 成员

成员可以在加入的圈子中发表话题以及回复，而且可以编辑话题和回复，以及删除回复。

(5) 游客

游客只能浏览话题及他的回复。

2. 话题设计

话题分为两种：普通话题和投票话题。

话题必要的功能有：字体、大小、粗体、斜体、下划线、颜色、图片附件、超链接、视频、图片链接、商品、表情。

由于话题全部是由用户输入，所以需要考虑XSS注入、防灌水、文字过滤等方面，以便对圈子进行防护。

（1）防止XSS注入

采用UBB代码形式。UBB与HTML一样，都是用来标记文本，并赋予文本一定的样式动作，但UBB作为HTML的一个变种，具备代码简单、功能实用、上手简单、安全性高、不存在XSS注入问题。

（2）防灌水

为防止成员恶意灌水，添加两次发帖时间间隔设置及帖子内容文字下限。恶意灌水会随着网站的知名度提升而与日俱增，严重影响论坛的整洁和用户对内容的正常浏览，为了改善这种情况，除了在技术方面进行限制还需要站长加强监管。

（3）关键字过滤

对色情、暴力、宗教或其他需要限制的言论进行控制。对用户的输入信息通过程序进行过滤、筛选关

关键词，进行智能识别，替换成指定字符。

（4）平台控制

平台可以控制圈子的各个细节，如圈子是否开启、是否可以创建新圈子、是否可以发表话题或回复、圈子的SEO设置等。

5.2.3 代码实现

这里主要介绍部分模块的代码实现。

1. 成员等级

当圈子的成员数量达到一定程度时，成员个体就会提现出互不相同的个性，如果使用同样的称号或者级别来标注所有用户，将会大大降低用户在一个圈子里面的归属感，也体会不到在圈子里面的成长乐趣，因此设计成员等级十分必要。

让圈子形成一个完善的成员成长体系，根据用户活跃度划分用户，最终达到提升用户活跃度的目的。总之就是，把用户的活跃度体现在经验上，然后按照经验决定用户等级。

圈子成员可以通过发表话题或者回复获得经验，当经验提升到一定程度后，其等级就会自动提升。为了奖励用户发表优质话题，设定话题被回复奖励发帖人经验。会员发帖获得经验流程如图5-9所示。

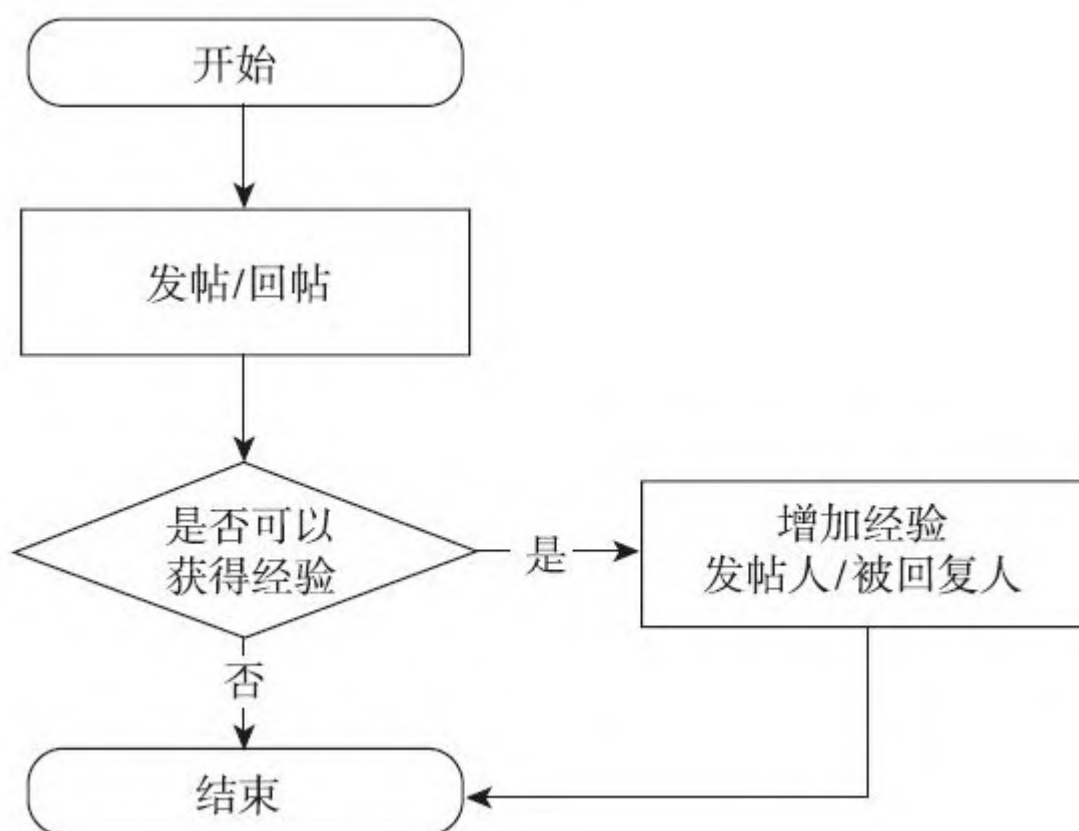


图5-9 发帖获得经验流程图

在回复帖子的时候需要同时验证回帖人和被回帖人当天是否可以获得经验。成员获得的经验数量需要

记录在相关的帖子中，如出现违规行为帖子被删除，应当同时删除成员所得的经验。

成员等级相关的数据库设计如图5-10所示。

在经验增长模型的设计中把发布话题、回复话题、话题被回复、删除话题、删除回复、管理员操作的经验升降全部封装到模型中，在对话题/回复进行操作的时候同时完成升降成员等级、记录等级日志操作。在控制层直接调用模型中的saveExp方法就可以了。具体代码见代码清单5-3。

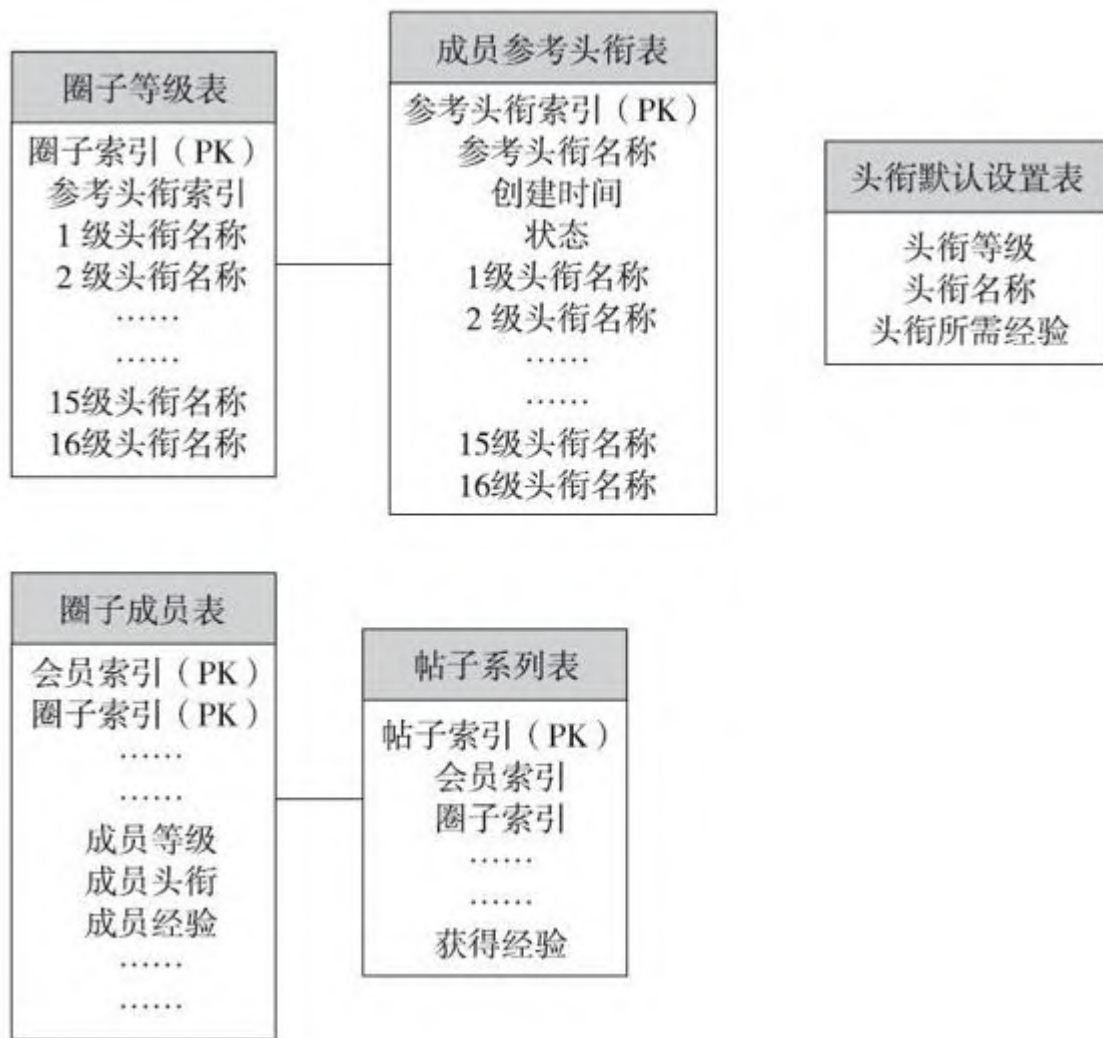


图5-10 圈子等级相关数据结构图

代码清单5-3 成员经验模型

```

// 保存经验
public function saveExp($param){
    switch ($param['type']){
        // 发布话题
        case 'release':
            $this->relOperation($param);
            break;
    }
}

```



```

// 回复话题
case 'reply':
    $this->relOperation($param);
    break;
// 话题被回复
case 'replied':
    $this->repldOperation($param);
    break;
// 删除话题
case 'delRelease':
    $this->delRelOperation($param);
    break;
// 删除回复
case 'delReplied':
    $this->delRepOperation($param);
    break;
// 管理员操作
case 'master':
    $this->masOperation($param);
    break;
}
}
// 发表话题/回复话题操作
private function relOperation($param){
    $exp = 0;
    $today = date('Y-m-d');
    $dexp = $param['type'] == 'release' ?
intval(C('circle_exprelease')) :
intval(C('circle_expreply'));
    // 验证是否限制每天获取经验最大数
    if(intval(C('circle_expreleasemax')) != 0){ //
限制
        // 计算经验
        .....
    }else{ // 没有记录
        // 计算经验
        .....
    }
    $insert = array();
    .....
    $this->table('circle_expmember')->insert($insert,
true);
    }else{ // 未限制
        $exp = $dexp;
    }
}

```

```

        if( $exp != 0 ){
            $param['exp'] = $exp;
            return $this->expOperation(1, $param);
        }
    }
    // 被回复操作
    private function repdOperation($param){
        $exp = 0; $today = date('Y-m-d');
        // 验证是否限制每天获取经验最大数
        if(intval(C('circle_exprepliedmax')) != 0){ // 限制
            // 计算经验
            .....
        }else{ // 没有记录
            // 计算经验
            .....
        }
        $insert = array();
        .....
        $this->table('circle_exptheme')->insert($insert,
true);
    }else{ // 不限制
        $exp = intval(C('circle_expelease'));
    }
    if($exp != 0){
        $param['exp'] = $exp;
        return $this->expOperation(1, $param);
    }
}
// 删除话题
private function delRelOperation($param){
    return $this->expOperation(0, $param);
}
// 删除回复
private function delRepOperation($param){
    return $this->expOperation(0, $param);
}
// 管理员操作
private function masOperation($param){
    return $this->expOperation(0, $param);
}
// 经验操作
private function expOperation($type, $param){
    // 查询成员信息
    $where = array();
    .....

```

```

        $cm_info = $this->table('circle_member')->where($where)-
>find();
        // 更新经验以及等级。
        $update = array();
.....
        $rs = $this->table('circle_member')->where($where)-
>update($update);
        if($rs){
            $this->recordsLog($type, $param);
        }else{
            return false;
        }
    }
    // 经验日志
    private function recordsLog($type, $param){
        $insert = array();
        .....
        switch ($param['type']){
            case 'release':
                // 记录增长多少经验
                $update = array();
                $update['theme_id']= $param['itemid'];
                $update['theme_exp']= array('exp',
'theme_exp+'. $param['exp']);
                $this->table('circle_theme')->update($update);
                .....
                break;
            case 'reply':
                .....
                break;
            case 'replied':
                .....
                break;
            case 'delRelease':
                .....
                break;
            case 'delReplied':
                .....
                break;
            case 'master':
                .....
                break;
        }
        $this->table('circle_explog')->insert($insert);
    }
}

```

2. UBB形式话题发布

首先要制定一个UBB标签的标准，然后根据制定的标准封装一个UBB形式的文本编辑器。

考虑到UBB文本编辑器不需要独立发布，所以没有把所有功能封装到一个JavaScript包中。把UBB文本编辑器分成了四部分：第一部分文本编辑器模板（使用PHP方法直接调用），第二部分JavaScript控件（封装成两个JavaScript文件，插入文本框和各个UBB标签的调用方法），第三部分样式代码（独立的css文件），第四部分UBB标签转化为HTML标签的正则代码（使用PHP方法直接调用）。

下面介绍一下模板调用方法，如代码清单5-4所示。

代码清单5-4 UBB文本编辑器调用方法

```

function showMiniEditor($cname, $content = '', $type =
'all', $affix = array(), $gname = '', $goods = array(),
$readperm = array(), $rpvalue = 0){
    // 工具栏显示类型，用来限制各个工具的显示
    switch ($type){
        case 'manage':
            $items = array('font', 'size', 'line', 'bold',
'italic', 'underline', 'color', 'line', 'url', 'flash',
'image', 'line', 'smilier');
            $return =
'$__content.$__maffix.$__goods.$__readperm';
            break;
        case 'quickReply':
            $items = array('font', 'size', 'line', 'bold',
'italic', 'underline', 'color', 'line', 'url', 'flash',
'line', 'smilier');
            $return = '$__content';
            break;
        case 'hQuickReply':
            $items = array('font', 'size', 'line', 'bold',
'italic', 'underline', 'color', 'line', 'url', 'flash',
'line', 'smilier', 'highReply');
            $return = '$__content';
            break;
        default:
            $items = array('font', 'size', 'line', 'bold',
'italic', 'underline', 'color', 'line', 'affix', 'line',
'url', 'flash', 'image', 'goods', 'line', 'smilier');
            $return =
'$__content.$__affix.$__goods.$__readperm';
            break;
    }
    // 工具栏各项条目标签
    $_line      = ".....";    // 分割线
    $_font      = ".....";    // 字体
    $_size      = ".....";    // 大小
    $_bold      = ".....";    // 粗体
    $_italic    = ".....";    // 斜体
    $_underline  = ".....";    // 下划线
    $_color     = ".....";    // 字体颜色
    $_affix     = ".....";    // 附件
    $_url       = ".....";    // 超链接
    $_flash     = ".....";    // false
    $_image     = ".....";    // 图片

```

```

$_goods      = ".....";    // 商品
$_smilier    = ".....";    // 表情
$_highReply= ".....";    // 高级回复
// 编辑器的内容
$__content = '';
$__content .= "<div class=\"content\">
               <div class=\"ubb-bar\">";
foreach ($items as $val){
    $val = ' '.$val;
    $__content .= $$val;
}
$__content .= "</div>
               <div class=\"textarea\">
                   <textarea id=\"".$cname.\"\"
name=\"".$cname.\"\">".$content."</textarea>
               </div>
               <div class=\"smilier\"></div>
               </div>";
// 附件部分
$__affix = '';
// 省略具体拼写成HTML的部分，下同。
.....
// 插入的商品显示部分
$__goods = '';
.....
// 阅读权限部分
$__readperm = '';
.....
eval('$return = '.$return.';');
return $return;
}

```

然后就可以在模板中直接调用了，代码如下：

```

<?php echo showMiniEditor('themecontent', '', 'all',
array(), 'goods', array(), $output['readperm']);?>

```

JavaScript插入控件的代码如代码清单5-5清单所示。

代码清单5-5 JavaScript UBB控件方法

```
(function($) {
    $.fn.ncUBB = function(options){
        var defaults = {
            c_id : 0,        // 圈子索引
            t_id : 0,        // 话题索引
            UBBContent: '',   // 内容对象
            UBBSubmit : '',   // 提交按钮对象
            UBBform : '',     // 表单对象
            UBBfileuploadurl : '', // 附件上传地址
            UBBcontentleast : 0, // 内容最小值
            run : ''          // 需要立即执行的方法，做个用“,”隔开
        };
        var options = $.extend(defaults, options);
        var c_id = options.c_id; var t_id = options.t_id;
        var UBBContent = options.UBBContent; var UBBSubmit =
options.UBBSubmit;
        var UBBform = options.UBBform; var UBBfileuploadurl
= options.UBBfileuploadurl; var UBBcontentleast =
options.UBBcontentleast;
        this.each(function() {
            // 需要立即执行的代码
            if(options.run != '') {eval(options.run);}
            var $this = $(this);
            // 加粗
            $this.find('a[nctype="b"]').click(function(){
                // 具体操作省略，下同
                .....
            });
            // 斜体
            $this.find('a[nctype="i"]').click(function(){
                .....
            });
            // 下划线
```

```

        $this.find('a[nctype="u"]').click(function() {
            .....
        });
        // 字体大小颜色
        $this.find('.ubb-
layer').children().click(function() {
            .....
        });
        // 表情
        $this.find('.smilier-
layer').children().click(function() {
            .....
        });
        // 超链接弹出层的显示与隐藏
        $this.find('.url-handle').toggle(function() {
            .....
        });

        // 超链接
        $this.find('.url-
layer').find('input[type="submit"]').click(function() {
            .....
        });
        // flash弹出层的显示与隐藏
        $this.find('.flash-handle').toggle(function() {
            .....
        });
        // flash插入编辑器
        $this.find('.flash-
layer').find('input[type="submit"]').click(function() {
            .....
        });
        // 图片上传
        $this.find("#test_file").fileupload({
            .....
        });
        // 调用选择商品的弹出层

        $this.find('a[nctype="chooseGoods"]').click(function() {
            .....
        });
        // 图片上传弹出层

        $this.find('a[nctype="uploadImage"]').click(function() {
            .....
        });

```



```

        // 插入附件

$this->find('a[nctype="affix_insert"]').click(function() {
    .....
});
// 删除附件

$this->find('a[nctype="affix_delete"]').click(function() {
    .....
});
// 管理员删除附件

$this->find('a[nctype="maffix_delete"]').click(function() {
    .....
});
// 删除商品
$this->find('.goods-del').click(function() {
    .....
});
// 相册图片插入话题

$('a[nctype="imagealbum"]').die().live('click',function() {
    .....
});
// 链接地址插入话题

$('a[nctype="imageurl"]').die().live('click',function() {
    .....
});
// 插入主题

$('a[nctype="insertGoods"]').die().live('click',function() {
    .....
});
// 提交表单
UBBSubmit.click(function() {
    UBBform.submit();
});
// 高级回复页面跳转

$this->find('a[nctype="highReply"]').click(function() {
    .....
});
/* group */
// 点击发话题

```

```

        $('.thread-layer').find('p').click(function() {
            .....
        });
    });
    // 插入图片
    function insertImgUBB(C) {
        .....
    }
    //插入商品
    function insertGoods(o) {
        .....
    }
    // 验证已插入商品数量, 决定$('div[class="insert-
goods"]')是否显示
    function checkInsertGoods() {
        .....
    }
    // 验证已插入附件数量, 决定$('div[nctype="affix"]')是否显
示
    function checkInsertAffix() {
        .....
    }
    // 附件插图到编辑器
    function affixInsert(o) {
        .....
    }
    // 删除附件
    function affixDelete(o) {
        .....
    }
    // 管理员删除附件
    function affixDeleteManage(o) {
        .....
    }
    // 附件上传插入html中
    function updateAffixInsert(param) {
        .....
    }
    // 未使用附件
    function getUnusedAffix() {
        .....
    }
}
}
})(jQuery);

```

调用ncUBB的方法如下：

```
$('.....').ncUBB({
  c_id : .....,
  UBBContent : .....,
  UBBSubmit : .....,
  UBBform : .....,
  UBBfileuploadurl : .....,
  UBBcontentleast : .....,
  run : .....
});
```

正则替换UBB标签为HTML标签，如代码清单5-6所示。

代码清单5-6 UBB标签正则替换为HTML标签方法

```
// 格式化UBB标签
function ubb($ubb){
    $ubb = str_replace(array(
        '[B]', '[/B]', '[I]', '[/I]', '[U]', '[/U]',
        '[IMG]', '[/IMG]', '[/FONT]', '[/FONT-SIZE]', '[/FONT-COLOR]'
    ), array(
        '<b>', '</b>', '<i>', '</i>', '<u>', '</u>',
        '<img class="pic" src=""', '</span>', '</span>',
        '</span>'
    ), preg_replace(array(
        "/\[URL=(.*)\](.*)\[\/URL\]/iU",
        "/\[FONT=([A-Za-z ]*)\]/iU",
        "/\[FONT-SIZE=([0-9]*)\]/iU",
        "/\[FONT-COLOR=([A-Za-z0-9]*)\]/iU",
        "/\[SMILIER=([A-Za-z_]*)\[\/\]/iU",
        "/\[FLASH\](.*)\[\/FLASH\]/iU",
    ))
```

```

        "\\\n/i"
    ), array(
        "<a href=\"$1\" target=\"_blank\">$2</a>",
        "<span style=\"font-family:$1\">",
        "<span style=\"font-size:$1px\">",
        "<span style=\"color:#$1\">",
        "<img
src=\"\".CIRCLE_SITE_URL.'/templates/'.TPL_CIRCLE_NAME.'/images/smilier/$1.png\">",
        "<embed src=\"$1\" type=\"application/x-shockwave-flash\" allowscriptaccess=\"always\" allowfullscreen=\"true\" wmode=\"opaque\" width=\"480\" height=\"400\"></embed>",
        "<br />"
    ), $ubb));
    return $ubb;
}

```

5.2.4 功能体验

进入圈子后，在首页幻灯片右侧点击“创建我的新圈子”按钮，跳转到创建圈子表单页面，如图5-11所示。

填写相关信息提交表单后，等待管理员审核，通过以后就可以在圈子中发表话题了，如图5-12所示。

填写完信息提交以后，就会在话题列表页中找到所发表的话题，如图5-13所示。

话题列表页设计了两种展现形式：列表形式和图文形式。列表形式就是类似论坛的展示形式，图文形式则把更新的话题信息展示到了列表页上，并可以在话题列表页直接浏览、回复话题。点击图5-13中的

“展开主题”按钮后，话题就会在列表页直接展开，极大地方便了用户的浏览，如图5-14所示。

创建一个圈子

欢迎你先在 发现精彩 发现喜欢的圈子。

欢迎在ShopNC圈子这个快乐和谐的地方，聚集和你爱好相同，品位相当的好朋友，畅谈交流，分享心情，享受生活！

最多可创建圈子数：6

已经创建圈子数：3

最多可加入圈子数：12

已经加入圈子数：4

所属分类：

☐ 服饰 ☐ 职场 ☐ 汽车 ☐ 音乐 ☐ 图书 ☐ 个性 ☐ 美容
☐ 宅男美女 ☐ 败家 ☐ 旅游 ☐ 美食 ☐ 动物 ☐ 亲子

根据您的圈子主题类型，选择适当的分类。

* 圈子名称：

圈子名称规定使用4~12个字符，确定后不可修改。

圈子简介：

还可以输入 255 字

对您建立的圈子进行简单的文字介绍，创建后圆主可做修改，字数不超过255字。

圈子标签：

建立圈子标签有利于全局搜索查找到您的圈子，多个标签请用“;”进行分隔。

申请理由：

还可以输入 255 字

认真填写申请圈子的理由提交至平台，以确保管理人员及时审核并通过，字数不要超过255字。

☒ 我已认真阅读并同意《圈子使用须知》中的所有条款

提交申请

图5-11 圈子申请表单页面

图5-12 发表话题表单

图5-13 话题列表

图5-14 话题列表页快速浏览及回复

5.3 本章小结

本章主要介绍了电子商务平台进行社交化的设计与开发：在具体的开发过程中我们要思考怎么样才能让社交平台与商城的结合更紧密，单单是平台的参与是远远不够的，店铺也需要参与进来。比如，创建或加入一个与自己店铺的产品相符合的圈子，根据圈子中的信息及时掌握消费者的需求，也可以了解消费者对商品的评价及时调整自己店铺的商品或营销策略。

第6章 IM套件的设计与实现

IM——即时通讯（Instant Messenger），核心功能是通过用户间相互收发消息，达到跨距离聊天的目的。大家用得最多的、最典型的的就是QQ，当然它现在集成了很多其他的功能，它的主要优势是在PC端，而我们要实现的是基于网页的向商城内商家与卖家双方提供服务的聊天系统（或称为简单的在线客服系统）。

6.1 模块构成

设计定位是服务于商城用户，没有单独的会员注册和登录程序，入口自然是商城会员系统，所以会员系统是必不可少的一部分。聊天程序最大的特点体现在时间上，如果达不到实时性，就和留言板的功能没区别了，在技术上通过消息推送实现，这构成了第二个模块，也是核心部分。有了前面两个相对没关联的模块，怎么在他们间搭起“桥梁”，由服务器端的PHP程序完成。会员（买家或商家）从页面客户端（浏览器）发出消息，服务器接收并处理，然后将消息返回给另一个页面客户端（浏览器），就是最基本的流程，在下节中会详细说明。

6.2 设计思路

为了了解此模块的核心功能，先来谈谈为什么要开发聊天系统。相信大家都收发过电子邮件，发出后什么时候能得到回复，不会有准确时间，如果不登录邮箱就不知道有新邮件，更别说回复了，所以后来就有了邮件推送这个业务，有新邮件时在手机上发个提醒。实现邮件推送要求接收的客户端处于在线状态，而在离线的情況下只能靠“第六感”了。因为商城与邮件属于不同系统，如果用邮件来进行买卖交流那效率太低了，后来很多网站开发了站内信系统，以减少沟通中等待时间。站内信比邮件用起来方便一些，但如果页面不刷新也不会看到新消息，为了达到更高的用户体验度，开发了在线客服系统。客服系统实质上是通过即时对话帮助用户解决问题的一个方式，最大

的进步就是实现了消息推送，不用频繁刷新查看新消息了。

本章介绍的是开发一套简单的在线客服系统，实现基本的即时聊天功能，整个系统从表面上分为发消息和收消息两部分，实际上从会员写消息到另一会员收到消息提醒，打开并回复的过程由很多小步骤组成。既然是在线聊天，当打开对话窗口时要能知道对方是否在线，在线时可以等他的回复，离线时发出后会变为留言，下次登录后就会看到提醒。以会员A给会员B发消息为例，发消息过程如图6-1所示。

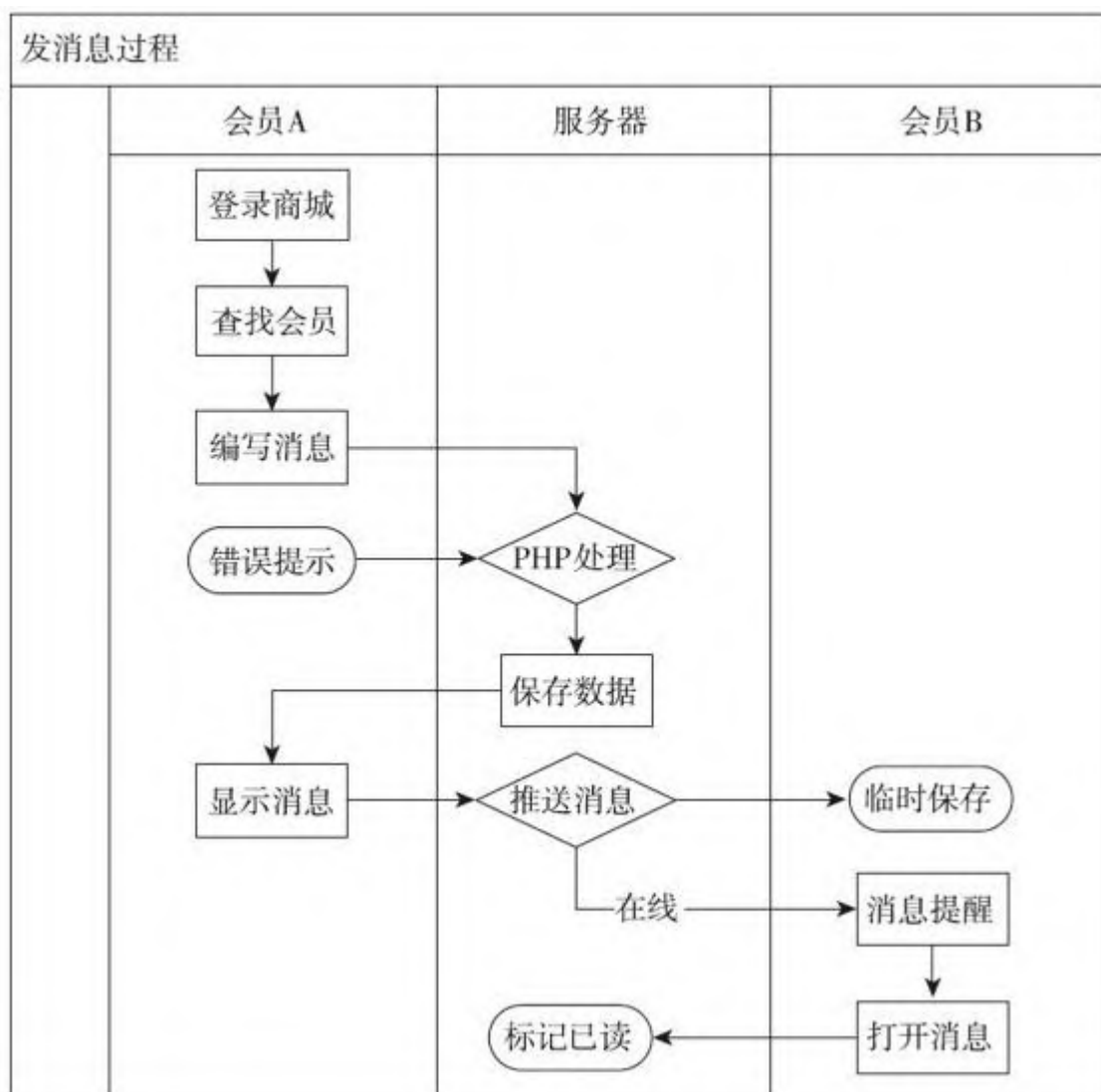


图6-1 发消息过程

如果认真看上面的步骤说明，可能会对“推送消息”这个环节提出疑问——为什么不在服务器上直接推送，而是返回给客户端后才进行操作？要解释这个问题，涉及服务器的资源分配和安全优化等方面。首

先说的是普通PHP加载的过程，浏览器向服务器发出请求，接收后进行各种逻辑处理，将数据返回给用户。当另一个用户请求同样数据时还是这样步骤，在大访问量下服务器很多都是在重复同样的操作，不管怎么优化单台服务器总有极限，解决办法是使用多台服务器进行集群部署。聊天程序也存在这种问题，为节约资源，将部分操作进行了拆分，在推送环节只处理最关键的逻辑，做到尽量少进行数据库的各种读写，将涉及较多的数据库操作的部分功能分解到集群部署的PHP服务器执行。

消息推送的好处是，当会员与服务器建立连接后不用定时请求查询数据，有新消息时服务器主动通知客户端（浏览器），从而节省了时间，也降低了服务器的负载压力。消息推送用的不是和PHP同一个平台软件（可以使用同一台硬件主机），其基础是

“Node.js”，这是以运行服务器端JavaScript为核心

目的建立的平台，可以在官方网站

“<http://nodejs.org/>”上看到更详细的介绍。

Node.js的功能很多，使用的具体技术工具是

“Socket.IO”，由它来实现对话消息的实时推送，该工具的官方网址是“<http://socket.io/>”，有兴趣的读者可以看看。这两方面的技术介绍远超出了本章的范围，有需要的读者可以利用网上的公开资料自行查询学习。

有一定开发经验的程序员可能会提出另一个问题，用Ajax也能实现不刷新页面来获取数据，那为什么又不采用呢？Ajax实现定时向服务器发请求，就有个时间长短的权衡问题，如果隔1秒就请求1次数据，只有几个人聊天还是能解决的，但会在没消息时浪费了大量查询。如果加大时间间隔，1分钟请求1次，会减少部分访问压力，在大访问量的情况下，服务器还是会忙不过来，间隔长了用户体验也会大打折扣。在

Node服务器进行消息推送时，客户端的JS程序已经对PHP返回的内容处理了，在Node服务器端不经过数据库读写直接将消息转发给另一会员，所有数据操作是在内存中执行，相比Ajax调用PHP查询数据库要快很多。

聊天系统涉及的核心数据就是对话双方的聊天记录，为能查到历史记录，要将消息保存到数据库中。由于商家可以添加多个账号进行日常管理，因此在商家中心要有添加客服子账号功能，在商品和店铺页面中显示，在业务较多时可减轻主账号的压力，所有的子账号的聊天记录都能在商家中心看到，便于对服务质量的监控。在商家中心能看到该店铺所有子账号与买家的对话记录，在普通会员中心只可以看到自己与他人的记录，在系统后台所有账号的数据都能查到。这三个子系统调用的是同一套程序逻辑，在访问权限方面要做好控制，防止出现数据泄露问题。

虽然这个系统主要功能是当做商家的在线客服工具使用，但不只是买家与商家之间对话，还可以是普通会员之间、商家账号之间都可以用，即商城会员内的任意两个人都可进行聊天。为方便经常联系，在会员登录后会异步调用“我的好友”和“最近联系人”的会员信息，点击对应的头像即可发起会话和查看聊天记录，类似QQ的功能。好友可在商城会员系统中添加，“最近联系人”的数据来源是根据当前账号的历史对话记录查询出来的。

会员在商城登录后页面加载完成时，由JS异步调用PHP获取会员联系人数据，成功返回后再与Node服务器建立连接，查询会员的在线状态并在页面中标识，如果有未读消息，则会出现提醒。会员登录后在商城浏览每个页面时都会与Node服务器建立连接，在有新消息提醒时所有的页面都会收到，会员在其中任何一个页面点开未读消息后都要将其他页面的提醒关闭，

防止再有新消息时看不到。完整的数据调用过程如图6-2所示。

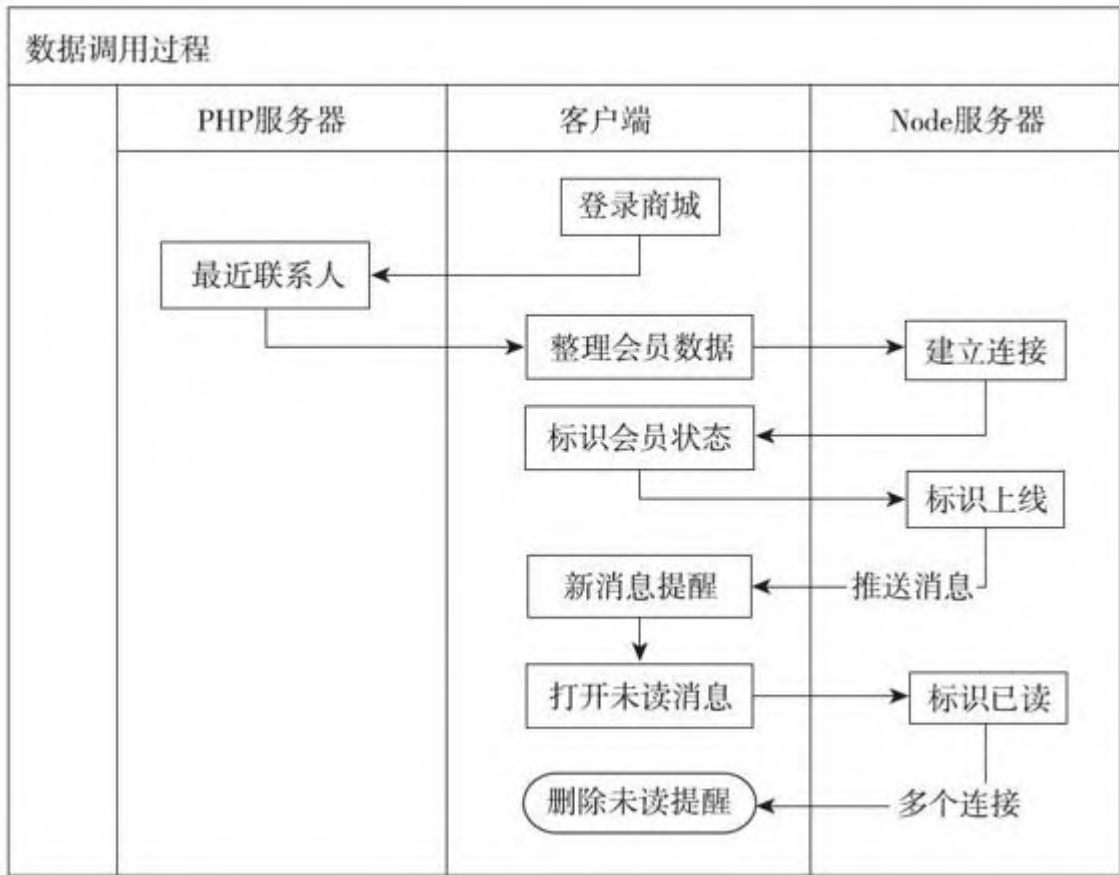


图6-2 数据调用过程

为防止Node服务器出现异常重启情况，在会员第一次连接时要在数据库查询是否有未读消息，以免看不到提醒。在聊天记录的保存方面，分为两个数据表来分担完成任务。“消息表”充当临时会话表，所有

的未读消息放在此表中，会员A收到会员B的已读消息只留一条，保持较少的数据量是为了更快查询速度。

“消息记录表”用来永久保存所有的对话记录，方便各方查询。聊天表数据关系如图6-3所示。

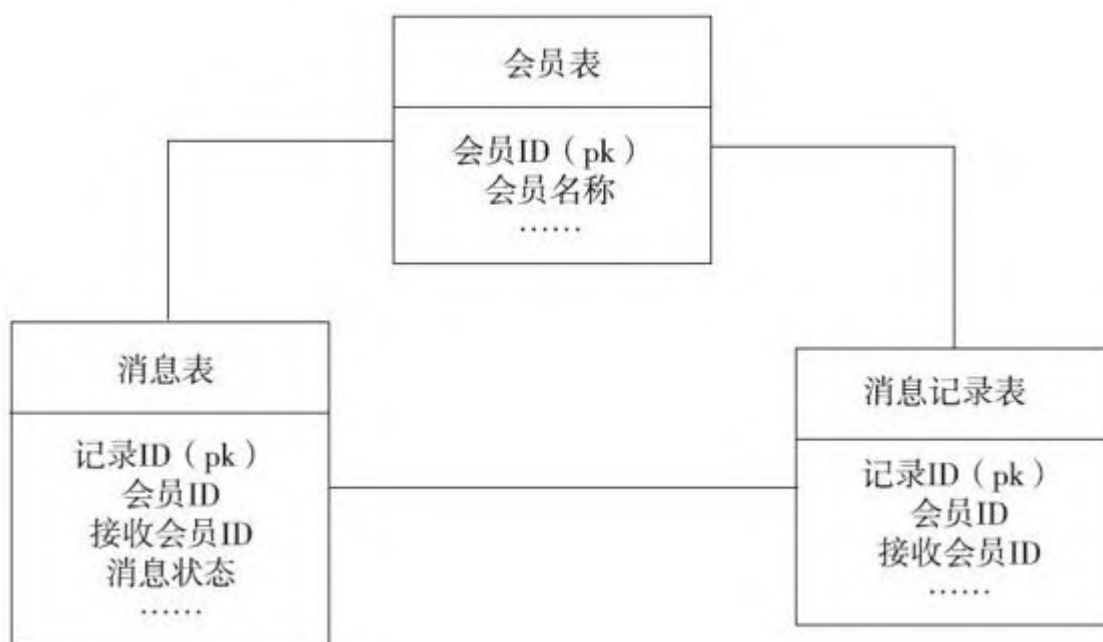


图6-3 聊天表数据关系

6.3 代码实现

从会员的角度看，聊天系统分为收消息和发消息两种功能，它只实现这两个基本操作，并不关心服务器怎么处理、数据是怎么流转的；从程序设计方面分为服务器运行端和客户端（浏览器）加载运行的两种；从技术方面来讲本章的程序以JavaScript为主，以PHP为辅，JavaScript的代码又分为页面客户端即浏览器加载部分和Node服务器端部分。理清了头绪写代码的效率会提高很多。

由于Node的代码与我们在页面中所用的JS代码规则还是有点区别的，建议先大概了解下原理，除了上节中所提的官方网站，还有个“强烈推荐”的资料——Node入门（The Node Beginner Book），在线版的地址是“<http://www.nodebeginner.org>”，有多种语言版本，英语水平不错的朋友直接看原版的为好，

中文的作为参考。初学者认真看完肯定会有很大收获，Node.js的开发者Ryan Dahl给的评论是 “This is an amazing introduction to Node.” 。

6.3.1 Node服务器端

Node服务器端代码主要分为三个功能，即数据库操作、消息处理、页面接口部分，通过商城页面的JS请求与服务器建立连接。本小节中所有代码都由JavaScript组成，要求程序员有一定的开发经验，能够理解基本的运行流程。Node服务器端的所有文件放在非Web服务器的目录，比如所用Apache的PHP环境，放到Apache访问不到的目录，防止关键文件泄露。为了方便二次开发，将涉及不同功能的代码放入四个文件中，根目录的config.js和chat.js，lib目录下的db.js和users.js，这节分别对主要代码进行相关逻辑调用说明。文件组织结构图如图6-4所示，node_modules文件夹是使用到的相关组件包。

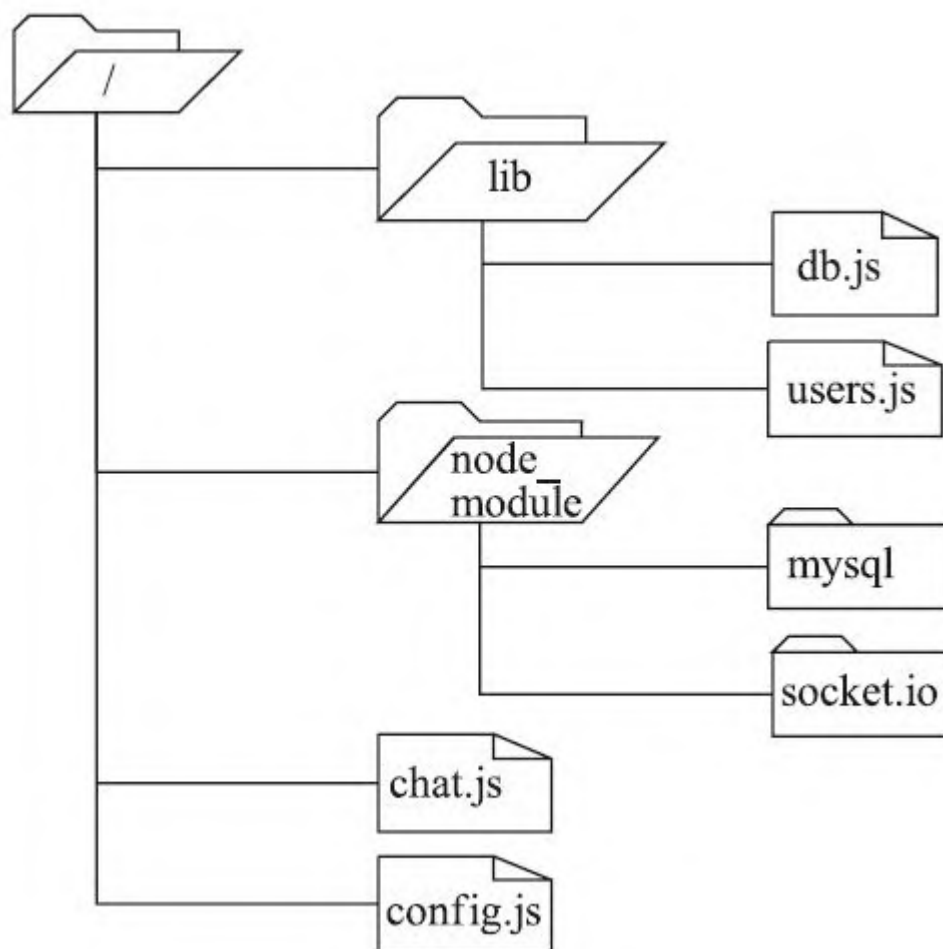


图6-4 文件组织结构图

“config.js”是Node服务器的相关配置项，主要内容是数据库的账号和对外访问端口的参数设置，内容如代码清单6-1所示。

代码清单6-1 Node服务器配置参数

```
//数据库账号设置
var config = {};
//数据库地址
config['host'] = 'localhost';
//数据库端口
config['port'] = '3306';
//数据库用户名
config['user'] = 'root';
//数据库密码
config['password'] = '';
//数据库名
config['database'] = '';
//表前缀
config['tablepre'] = 'shopnc_';
//兼容低版本
config['insecureAuth'] = true;
//开启调试MySQL, 默认false
config['debug'] = false;
//授权连接的域名或IP, 为空不限制
exports.hostname = '';
//服务器所用端口号, 默认8090
exports.port = 8090;
exports.config = config;
```

如代码清单6-2所示，数据库操作时使用 NPM（Node Packaged Modules）的MySQL包，类似PHP 中的MySQL扩展，调用说明可以在 “<https://www.npmjs.org/package/mysql>” 中看到。handleDisconnect方法是在数据库连接断开后自动重连，保证数据的及时更新。

代码清单6-2 Node连接数据库

```
//引用MySQL的驱动程序包
var mysql = require('mysql');
//从配置文件读取数据库的账号信息
var db_config = require('../config');
var db = '';
//创建数据库连接
db = mysql.createConnection(db_config.config);
//数据库异常处理
function handleDisconnect(db) {
  db.on('error', function(err) {
    if (!err.fatal) {
      return;
    }
    if (err.code !== 'PROTOCOL_CONNECTION_LOST') {
      throw err;
    }
    //输出错误提示
    console.log('Re-connecting lost mysql: ' + err.code);
    //重新连接
    db = mysql.createConnection(db.config);
    handleDisconnect(db);
    db.connect();
  });
}
handleDisconnect(db);
//连接成功后提示
db.connect(function(err) {
  if (err) throw console.log('mysql err: ' + err.code);
  console.log('mysql connected');
});
```

如代码清单6-3所示，其中三个方法是提供给“页面接口”部分的调用，针对“消息表”操作，类似PHP

服务器端的聊天代码模型类功能，调用get_query对参数进行过滤处理，只是基本的查询和更新，相对简单。与前面的代码组成了数据库操作文件——“db.js”。

代码清单6-3 消息表的数据库操作方法

```
//查询消息
exports.get_msg_list = function (sql, cb) {
var db_query = 'SELECT * FROM ';
    var table = tablepre+'chat_msg ';
    db_query += table+' WHERE ';
    db_query += get_query(sql);
    db.query(db_query, function(err, rows) {
    cb(rows);
    });
}
//删除消息
exports.del_msg = function (sql) {
    var db_query = 'DELETE FROM ';
    var table = tablepre+'chat_msg ';
    db_query += table+' WHERE ';
    db_query += get_query(sql);
    db.query(db_query, function(err, rows) {
    });
}
//更新消息
exports.update_msg = function (sql, values) {
    var db_query = 'UPDATE ';
    var table = tablepre+'chat_msg SET ';
    db_query += table;
    for (var k in values){
        db_query += k+"='"+db.escape(values[k])+"'";
    }
};
```

```
        db_query += ' WHERE ';\n        db_query += get_query(sql);\n        db.query(db_query, function(err, rows) {\n            });\n    }\n}
```

消息处理部分和数据库操作代码一样都属于数据库操作，没有逻辑调用关系，任何的消息都不能脱离于会员而存在，会员信息会在消息推送时发给客户端，以减少PHP对数据库的查询，因此会员数据在内存中长期保存，推送的消息数据在会员标识为已读后删除，如代码清单6-4所示。

代码清单6-4 会员数据处理

```
//与服务器连接过的会员信息\nvar user_list = new Array();\nexports.set_user = set_user;\nexports.set_user_info = set_user_info;\nexports.get_user = get_user;\n//更新会员的连接信息\nfunction set_user(user){\n    var user_info = new Object();\n    var u_id = user['u_id'];\n    user_info = get_user(u_id);\n    if ( typeof user_info['u_id'] === "undefined" ) {\n        user_info['u_id'] = u_id;\n        user_info['u_name'] = user['u_name'];\n        user_info['s_id'] = user['s_id'];\n        if ( typeof msg_list[u_id] === "undefined" )
```

```
msg_list[u_id] = new Object();
    }
    user_list[u_id] = user_info;
}
//设置会员信息
function set_user_info(u_id,k,v){
    if ( typeof user_list[u_id] === "object" )
user_list[u_id][k] = v;
}
//获取会员信息
function get_user(u_id){
    var user_info = new Object();
    if ( typeof user_list[u_id] === "object" ) {
        user_info = user_list[u_id];
    }
    return user_info;
}
```

Node服务器只处理未读消息，查询历史聊天记录由PHP服务器来完成，这样设计是为了减少Node服务器的压力，还降低了程序的复杂度，如代码清单6-5所示。处理消息和会员的代码共同放在“users.js”文件中，方便后期的功能扩展。

代码清单6-5 消息数据处理

```
//所有未读消息
var msg_list = new Array();
exports.set_msg = set_msg;
exports.get_msg = get_msg;
exports.del_msg = del_msg;
//添加未读消息
```

```
function set_msg(u_id,msg){
    var m_id = msg['m_id'];
    if ( typeof msg_list[u_id] === "undefined" )
msg_list[u_id] = new Object();
    msg_list[u_id][m_id] = msg;
}
//获取会员的未读消息
function get_msg(u_id){
    return msg_list[u_id];
}
//删除未读消息
function del_msg(u_id,m_id){
    delete msg_list[u_id][m_id];
}
}
```

页面接口功能是从商城页面JS中接收数据，保存到Node服务器内存中，然后推送给另一个客户端，此部分是本章功能的核心所在，涉及的代码放在“chat.js”中。在代码清单6-6中分别引用了“db.js”、“users.js”，先是建立一个类似Apache服务器——server，目的是在浏览器中直接访问时出现提示——“非正常访问”，对外的端口号在配置文件中可修改，再启用Socket.IO服务。其中“log level”是Socket.IO代码调试时使用，0为输出错误——error，1是输出警告——warn，2是输出提示信息

——info，3是排查错误——debug。“origins”是允许调用访问Node服务器的域名和端口，建议用默认的“*:*”，因为在商城中店铺可能会使用二级域名，如果自行设置会出现部分页面连接不上的情况，改用authorization接口通过正则表达式检测请求连接的网站域名与配置文件中设置的是否一致，不符合时在控制台出现一条提示信息，方便管理。“resource”是页面中JS调用的目录名，默认的是“/socket.io”，这个要求与浏览器端运行代码设置一致。在

“transports”中可设置数据传输方式优先使用顺序，websocket方式为最理想的，要求浏览器支持HTML5，从Firefox4、Chrome4、Safari5和IE10及以上版本可用，如果测试时将flashsocket放在前面，也能连接成功，不过这样很不稳定，所以不推荐。

代码清单6-6 建立Node服务器

```

var http = require('http');
var url = require('url');
var db = require('./lib/db');
var lib_user = require('./lib/users');
var _config = require('./config');
var hostname = _config.hostname;
var port = _config.port;
var server = http.createServer(function (req, res) {
res.writeHead(200, {'Content-Type': 'text/plain'});
res.end('Access Invalid!\n');
});
server.listen(port);
var io = require('socket.io').listen(server);
io.set('log level', 0);
io.set('origins', '*:*');
io.set('resource', '/resource');
io.set('transports', ['websocket', 'xhr-polling', 'jsonp-polling', 'htmlfile', 'flashsocket']);
io.set('authorization', function (handshakeData, cb) {
    var domain = hostname;
    var origin = handshakeData.headers.origin ||
handshakeData.headers.referer;
    var parts = url.parse(''+origin);
    var re = new RegExp(domain+"$", "g");
    var arr = re.exec(parts.hostname);
    if ( domain === "" ) {
cb(null, true);
    } else if ( arr !== null ) {
cb(null, true);
    } else {
var dt = new Date();
console.log(dateToString(dt)+' '+parts.hostname+' handshake
unauthorized');
cb(null, false);
    }
});

```

在客户端与Node服务器成功建立连接后，就可收发消息了，如代码清单6-7所示，socket分别监听对应

事件，除“disconnect”由程序自动触发外，其他都是在浏览器页面加载的JS中触发，执行完相应逻辑后再向客户端触发操作，交互进行这样的操作来完成会员双方的聊天对话。update_user事件功能是在会员与Node服务器建立连接后，调用update_user方法查询是否有未读消息（只是在第一次连接时执行一次），触发客户端的get_msg事件来推送离线时收到的消息，然后标识会员为在线状态；send_msg是接收从客户端发出的消息内容，查询另一会员在线时向他推送的消息，双方在线聊天时send_msg会被重复调用来传递对话内容；get_state根据接收的会员编号查询是否在线，并向客户端触发get_state事件后，在页面中标出对应会员的在线状态；del_msg是在会员查看消息后触发，删除内存中的未读消息防止重复提醒多次；disconnect是在会员的客户端与服务器连接断开时自

动触发，网络掉线、退出登录时执行，标识会员为离线状态。

代码清单6-7 Node服务器的接口

```
io.sockets.on('connection', function (socket) {  
    //连接成功后向客户端发未读消息  
    socket.on('update_user', function (user) {  
        var u_id = user['u_id'];  
        update_user(user, function () {  
            socket.set('u_id', u_id);  
            socket.join('user_'+u_id);  
            var msg_list = {};  
            msg_list = lib_user.get_msg(u_id);  
            socket.emit('get_msg', msg_list);  
            lib_user.set_user_info(u_id, 'connected', 1);  
  
            lib_user.set_user_info(u_id, 's_id', user['s_id']);  
  
            lib_user.set_user_info(u_id, 's_name', user['s_name']);  
  
            lib_user.set_user_info(u_id, 'avatar', user['avatar']);  
            connected_users[u_id] =  
            lib_user.get_user(u_id);  
        });  
        //接收消息并向客户端发通知  
        socket.on('send_msg', function (msg) {  
            var t_id = msg['t_id'];  
            var n = io.sockets.clients('user_'+t_id).length;  
            socket.get('u_id', function (err, u_id) {  
                msg['user'] = lib_user.get_user(u_id);  
                //会员在线时发通知  
                if ( n > 0) {  
                    var m_id = msg['m_id'];  
                    var msg_list = {};  
                    msg['online'] = 1;  
                    msg_list[m_id] = msg;
```

```

        io.sockets.in('user_'+t_id).emit('get_msg',
msg_list);
    }
    lib_user.set_msg(t_id,msg);
});
});
//查询在线状态并返回客户端
socket.on('get_state', function (u_state) {
    var list = {};
    var user_list = {};
    for (var k in u_state){
        var user_info = {};
        user_info = lib_user.get_user(k);
        user_info['online'] = 0;
    }
    var n = io.sockets.clients('user_'+k).length;
    //会员在线
    if ( n > 0) {
        u_state[k] = 1;
        user_info['online'] = 1;
    }
    if ( user_info['u_id'] > 0) user_list[k] =
user_info;
}
list['u_state'] = u_state;
list['user'] = user_list;
socket.emit('get_state', list);
});
//更新未读消息
socket.on('del_msg', function (msg) {
    .....
});
//断开连接
socket.on('disconnect', function () {
    .....
});
});
});

```

以上程序逻辑相对商城页面部分的PHP和JS并不复杂，所有的操作是在客户端触发执行，在开发时调试

会麻烦些，因此建议使用Firefox来辅助开发，监测输出Node服务器返回的数据是否正确（本小节代码在Socket.IO的0.9.16版本中测试运行无误，与最新的1.0版本会出现不兼容情况，稍做修改即可）。

6.3.2 PHP服务器端

与Node服务器端程序一样，PHP服务器端接口程序也是由商城页面中的JS来调用，为了在店铺页面跨域名正常使用，Ajax采用JSONP数据类型进行传输。

代码清单6-8实现的是会员登录后向服务器请求好友和最近联系人的程序，为了防止会员相关的数据被直接取出，加入了\$_SESSION数据与传入的会员编号是否一致的判断。调用getFriendList取出好友信息，而查询最近联系人有两个方法，第一个是根据发出的消息查询，第二个是根据收到的消息查询。这两个方法分别从数据库的记录中查找会员信息，会出现重复数据，经过相应方法内部处理，\$member_list中会员是唯一的，分别用状态标识好友和最近联系人。最后由

json方法将数组编码输出给Ajax，以供客户端JS程序使用。

代码清单6-8 查询好友和联系人接口

```
public function get_user_listOp(){
    $member_list = array();
    $model_chat= Model('web_chat');
    $member_id = $_SESSION['member_id'];
    $member_name = $_SESSION['member_name'];
    $f_id = intval($_GET['f_id']);
    if (($member_id < 1) || ($member_id != $f_id))
    $this->error(Language::get('nc_member_chat_login'));
    $n = intval($_GET['n']);
    if ($n < 1) $n = 50;
    $member_list = $model_chat-
>getFriendList(array('friend_frommid'=>$f_id),$n,$member_list);
    $add_time = date("Y-m-d");
    $add_time30 = strtotime($add_time)-60*60*24*30;
    $member_list = $model_chat-
>getRecentList(array('f_id'=>$f_id,'add_time'=>array('egt',$add_time30)),10,$member_list);
    $member_list = $model_chat-
>getRecentFromList(array('t_id'=>$f_id,'add_time'=>array('egt',$add_time30)),10,$member_list);
    $this->json($member_list);
}
```

客户端发消息要经过PHP的服务器，保证消息内容存入数据库后才让Node服务器进行推送，这样是利用PHP较成熟的数据过滤规则，防止被加入恶意代码，以

及避免影响数据安全和用户体验，如代码清单6-9所示。在发消息时要对当前登录的会员状态进行判断，以免在其他页面退出后或登录超时还能发消息。

代码清单6-9 发消息接口

```
public function send_msgOp(){
    $member = array();
    $model_chat = Model('web_chat');
    $member_id = $_SESSION['member_id'];
    $member_name = $_SESSION['member_name'];
    $f_id = intval($_POST['f_id']);
    $t_id = intval($_POST['t_id']);
    $t_name = trim($_POST['t_name']);
    if (($member_id < 1) || ($member_id != $f_id))
    $this->error(Language::get('nc_member_chat_login'));
    $member = $model_chat->getMember($t_id);
    if ($t_name != $member['member_name']) $this->error(Language::get('nc_member_chat_name_error'));
    $msg = array();
    $msg['f_id'] = $f_id;
    $msg['f_name'] = $member_name;
    $msg['t_id'] = $t_id;
    $msg['t_name'] = $t_name;
    $msg['t_msg'] = trim($_POST['t_msg']);
    if ($msg['t_msg'] != '') $chat_msg = $model_chat->addMsg($msg);
    if($chat_msg['m_id']) {
        $this->json($chat_msg);
    } else {
        $this->error(Language::get('nc_member_chat_add_error'));
    }
}
```

商家中心和系统后台的聊天记录查询都是用PHP从数据库调用数据，抛给模板页面。需要注意的是为防止大数据量下速度变慢，调用查询时加上时间段的限制，也可以考虑将一年前的数据导出备份到其他数据库表中，保持“消息记录表”只存放最近会调用到的数据。代码清单6-10是根据会员编号查询聊天记录的方法，根据发消息人的会员编号、收消息人的会员编号来准确匹配，用开始时间和结束时间来缩小数据范围。

代码清单6-10 查询聊天记录方法

```
public function getLogFromList($condition = array(), $page = 10) {
    $list = array();
    $f_id = intval($condition['f_id']);
    if ($f_id > 0) {
        $t_id = intval($condition['t_id']);
        if ($t_id > 0) {
            $condition_sql = " ((f_id = '". $f_id. "'
and t_id = '". $t_id. "') or (f_id = '". $t_id. "' and t_id =
'". $f_id. "'))";
        } else {
            $condition_sql = " (f_id = '". $f_id. "'
or t_id = '". $f_id. "'))";
        }
    }
```

```
$add_time_from = trim($condition['add_time_from']);
if (!empty($add_time_from)) {
    $add_time_from = strtotime($add_time_from);
    $condition_sql .= " and add_time >=
'".$add_time_from."'";
}
$add_time_to = trim($condition['add_time_to']);
if (!empty($add_time_to)) {
    $add_time_to = strtotime($add_time_to)+60*60*24;
    $condition_sql .= " and add_time <=
'".$add_time_to."'";
}
$list = $this->getLogList($condition_sql,$page);
}
return $list;
}
```

6.3.3 页面客户端

接下来的程序主要是JS向PHP、Node服务器分别发送和接收数据，根据不同内容进行处理后在页面中展示给会员，更多是为了提高用户体验而设计。由于篇幅限制此节中只介绍数据处理的逻辑部分程序，省略了页面布局、CSS样式、JS交互效果等相关代码。

会员登录成功后生成的\$_SESSION是保存在PHP服务器，JS要调用时从页面中获取。代码清单6-11就是由PHP根据系统的各项参数及会员的登录情况输出到页面的代码。APP_SITE_URL是商城会员登录页面中弹出框所使用的，CHAT_SITE_URL是聊天程序所调用的相关资源网址（PHP接口和JS文件），connect_url是Node服务器地址，user是会员相关信息，如会员编号、登录名、店铺编号、店铺名、头像等（因为不同会员在

不同的服务器页面会生成不同代码，因此没有列出具
体变量值）。当没有登录时会员user数组内各项值为
空，在后面的代码中会根据会员编号是否为空来确定
是否是登录状态，其他变量在不同服务器会出现不同
的值。

代码清单6-11 PHP输出的相关变量

```
<script type="text/javascript">
var APP_SITE_URL = '';
var CHAT_SITE_URL = '';
var SHOP_SITE_URL = '';
var connect_url = '';
var layout = '';
var act_op = '';
var user = {};
user['u_id'] = '';
user['u_name'] = '';
user['s_id'] = '';
user['s_name'] = '';
user['avatar'] = '';
$("#chat_login").nc_login({
    action: '/index.php?act=login',
    nchash: '',
    formhash: ''
});
</script>
```

以上代码会输出在PHP页面的底部，紧接着调用由本节后面内容所涉及的代码组成的文件user.js，利用PHP提供的各种变量分别与PHP、Node服务器进行数据交互，由客户端的JS处理后呈现给会员。由于相关的代码较多，我们只对重要的逻辑进行说明，在程序中会调用的变量如代码清单6-12所示。

代码清单6-12 客户端页面数据调用变量

```
//断开后计时
var interval = 1;
//消息提醒计时
var new_interval = 1;
//连接状态
var connect = 0;
//新消息数
var new_msg = 0;
var obj = {};
var socket = {};
var chat_log = {};
var connect_list = {};
var connect_n = 0;
//页面信息
var web_info = new Array();
//我的好友
var friend_list = new Array();
//最近联系人
var recent_list = new Array();
//所有会员信息
var user_list = new Array();
//收到消息
```

```
var msg_list = new Array();  
//对话框是否打开  
var dialog_show = 0;  
//当前选择的会员  
var user_show = 0;  
//消息数  
var msg_max = 20;  
//定时(分钟)刷新防止登录超时退出,为0时关闭  
var time_max = 10;
```

页面加载完成后会自动触发PHP服务器的
get_user_list0p接口（在上节中已经提到），查询好友和最近联系人，返回的u_list会员列表经过处理分别放入friend_list好友和recent_list最近联系人，设置1秒后用getconnect方法连接Node服务器，查询会员的在线状态并在页面中进行标识，如代码清单6-13所示。

代码清单6-13 异步调用好友和最近联系人

```
var ajaxurl = CHAT_SITE_URL+'/index.php?  
act=web_chat&op=get_user_list&n=99&f_id='+user['u_id'];  
$.ajax({  
    type: "GET",  
    url: ajaxurl,  
    dataType: "jsonp",  
    async: true,  
    success: function(u_list){
```

```

for (var u_id in u_list){
    var user_info = u_list[u_id];
    connect_list[u_id] = 0;
    connect_n++;
    set_user_info(u_id,"u_name",user_info['u_name']);
    set_user_info(u_id,"avatar",user_info['avatar']);
    if ( user_info['friend'] == 1 ) friend_list[u_id]
= user_info;
    if ( user_info['recent'] == 1 ) recent_list[u_id]
= user_info;
}
setTimeout("getconnect()",1000);
.....
}
});

```

为了与Node服务器断开后重连，将连接Node服务器的代码进行了封装，成功连接后会创建socket对象，并设置connect为1，表示与服务器已经连接，可以进行收发消息的操作了，内部方法功能类似服务器端，如代码清单6-14所示。调用send_state向Node服务器发送会员编号数组，服务器处理完后触发客户端的get_state事件来标识页面中会员的在线状态；update_user是更新当前会员的在线状态以备其他会员查询；get_msg是接收并处理服务器推送的消息；del_msg是删除从其他页面中已读消息在当前页面的提

醒；disconnect是与Node服务器断开后定时自动重连程序。后面会分别对主要部分进行说明。

代码清单6-14 客户端socket接口

```
function getconnect(){
$.getScript(connect_url+"/resource/socket.io.js", function()
{
    clearInterval(interval);
    if ( typeof io === "object" ) {
socket = io.connect(connect_url, { 'resource': 'resource',
'reconnect': false });
        socket.on('connect', function () {
            connect = 1;
send_state();
socket.on('get_state', function (u_list) {
get_state(u_list);
});
            //未登录时不取消息
if(user['u_id'] == '') return false;
$("#web_chat_dialog").show();
if($("#new_msg_dialog").size()==0)
$("#web_chat_dialog").after(web_info['msg_dialog']);
            obj = $("#new_msg_dialog");
socket.emit('update_user', user);
socket.on('get_msg', function (msg_list) {
get_msg(msg_list);
});
socket.on('del_msg', function (msg) {
del_msg(msg);
});
socket.on('disconnect', function () {
connect = 0;
$("#web_chat_dialog").hide();
//断开1分钟后重新连接服务器
interval = setInterval( getconnect, 60000);
});
    });
}
```

```
    }  
    });  
}
```

先来看下send_state方法。如代码清单6-15所示，u_list从connect_list获得当前登录会员的好友和最近联系人，connect_n是会员总数，根据不同的html标签规则在当前浏览的页面查找会员编号，已登录时向Node服务器触发查询会员是否在线的get_state事件，没登录又没有需要查询的会员时断开socket连接。

代码清单6-15 向Node服务器查询在线状态

```
function send_state(){  
    var u_list = connect_list;  
    var n = connect_n;  
    if(layout == 'layout/store_layout.php') {  
        .....  
    } else {  
        switch (act_op){  
            case "act_op":  
                break;  
            .....  
            default:  
                $("[member_id]").each(function() {  
                    n++;  
                    var u_id = $(this).attr("member_id");
```

```

        if ( u_id > 0 && u_id != user['u_id'] )
u_list[u_id] = 0;
        });
        break;
    }
}
if(connect === 1) {
    if(n > 0) {
        socket.emit('get_state', u_list);
    } else {
        if (user['u_id'] == '') {
            socket.disconnect();
        }
    }
} else {
    return n;
}
}

```

如代码清单6-16所示，get_state方法处理Node服务器返回给客户端的数据。list['u_state']中是会员的在线状态数据，list['user']则是发送的会员中已经登录过的会员信息，以免显示时再次向PHP服务器查询。set_user_list利用返回的信息对会员进行数据更新，调用get_chat方法对在线状态标识完后，当会员在页面中是登录状态时更新好友和最近联系人，未登录时主动关闭socket连接。

代码清单6-16 在页面标识会员在线状态

```
function get_state(list){
    var u_list = list['u_state'];
    set_user_list(list['user']);
    if(layout == 'layout/store_layout.php' || act_op ==
'goods_index') {
        .....
    } else {
        switch (act_op){
            .....
            default:
                $("[member_id]").each(function(){
                    var u_id = $(this).attr("member_id");
                    if($(this).find(".chat").size()==0) {

$(this).prepend(get_chat(u_id,u_list[u_id]));
                    }
                });
                break;
            }
        }
        if(user['u_id'] != '') {
            update_recent();
            update_friends();
        } else {
            socket.disconnect();
        }
    }
}
```

如代码清单6-17所示，发消息时要选择对话的会员，对消息的内容做必要的简单验证，向PHP服务器提交消息数据，返回成功时再向Node服务器触发

send_msg事件进行消息推送，最后调用show_t_msg将刚才发出的消息内容显示在页面对话框中。

代码清单6-17 向PHP服务器提交消息

```
function send_msg(){
    if(user_show < 1) {
        $("#send_alert").html('未选择聊天会员');
        return false;
    }
    var msg = {};
    msg['f_id'] = user['u_id'];
    msg['f_name'] = user['u_name'];
    msg['t_id'] = user_show;
    msg['t_name'] = user_list[user_show]['u_name'];
    msg['t_msg'] = $.trim($("#send_message").val());
    if(msg['t_msg'].length < 1) {
        $("#send_alert").html('发送内容不能为空');
        return false;
    }
    if(msg['t_msg'].length > 255) {
        $("#send_alert").html('一次最多只能发送255字');
        return false;
    }
    if(connect < 1) {
        $("#send_alert").html('处于离线状态,稍后再试');
        return false;
    }
    $.ajax({
        type: "POST",
        url: CHAT_SITE_URL+'/index.php?
act=web_chat&op=send_msg',
        dataType:"jsonp",
        data: msg,
        async: false,
        success: function(t_msg){
            if(t_msg['error']) {
                $("#send_alert").html(''+t_msg['error']);
            }
        }
    });
}
```

```

return false;
} else {
if ( connect === 1 ) {
socket.emit('send_msg', t_msg);
$("#send_message").val('');
$("#send_message").focus();
$("#send_alert").html('');
show_t_msg(t_msg);
return true;
} else {
$("#send_alert").html('由于网络原因未发送成功,稍后再试');
return false;
}
}
});
}

```

处理服务器推送消息的代码如代码清单6-18，其中get_msg是接收Node服务器推送来的消息，list是未读消息数组列表，因为list消息内容可能由不同会员发出，但在页面中同时只能有一个会员是处在对话中的状态，所以一对多聊天时在客户端要对未读消息以会员编号u_id、消息编号m_id为键值进行保存。调用set_user方法对发送消息的会员信息初始化，如果返回消息中有发消息的会员相关信息，用返回的数据对其他数据进行更新，当没有获得头像时请求PHP服务器

的get_info接口查询一次数据，以供页面显示时使用。dialog_show为0表示页面中没打开对话框，保存收到的消息到msg_list[u_id]，alert_user_msg方法的功能是在页面中对单个会员消息做未读标记，最后调用alert_msg对所有的未读消息进行提醒。打开对话框但不是和发消息人对话时处理方式同上，如果正在与发消息人对话，用show_msg来显示消息内容并删除客户端msg_list[u_id]及Node服务器中对应的未读消息。

代码清单6-18 处理服务器推送的消息

```
function get_msg(list){
    var msg = {};
    for (var k in list){
        msg = list[k];
        var m_id = msg['m_id'];
        var u_id = msg['f_id'];
        set_user(u_id,msg['f_name']);
        if (typeof msg['user'] === "object" && typeof
msg['user']['avatar'] !== "undefined" ) {
            var user_info = msg['user'];
            var u_name = user_info['u_name'];
            set_user_info(u_id,"u_name",u_name);
            set_user_info(u_id,"s_id",user_info['s_id']);
            set_user_info(u_id,"s_name",user_info['s_name']);
        }
    }
}
```

```

        set_user_info(u_id,"avatar",user_info['avatar']);
        if ( user_info['online'] > 0 )
set_user_info(u_id,"online",1);
    }
    //当没获得会员信息时调用一次
    if ( typeof user_list[u_id]['avatar'] === "undefined" )
    {
        var ajaxurl = CHAT_SITE_URL+'/index.php?
act=web_chat&op=get_info&t=member&u_id='+u_id;
        .....
    }
    msg_list[u_id][m_id] = msg;
    //没有打开对话窗口时计数
    if(dialog_show == 0 ||
obj.find("li[select_u_id='"+u_id+"'").size()==0) {
        user_list[u_id]['new_msg']++;
        new_msg++;
    } else {
        if ( user_show == u_id) {
            show_msg(u_id);
        } else {
            user_list[u_id]['new_msg']++;
            new_msg++;
        }
    }
    alert_user_msg(u_id);
}
alert_msg();
}

```

如果一个会员同时打开了多个页面，那么与Node服务器会有多个处在连接中的socket，此时一旦有新消息所有页面都可能会出现提醒，在一个页面中标记为已读时，其他页面中的提醒应该关闭。因为浏览器安全机制问题，客户端的JS只能在所加载的页面中执

行，所以当读取消息后要向服务器触发del_msg事件来删除已读消息，再由Node服务器向客户端触发socket中的del_msg接口。

```
function del_msg(msg) {
    //最大的消息编号
    var max_id = msg['max_id'];
    var u_id = msg['f_id'];//消息发送人
    for (var m_id in msg_list[u_id]){
        if ( max_id >= m_id) {
            delete msg_list[u_id][m_id];
            if ( user_list[u_id]['new_msg'] > 0 ) user_list[u_id]
            ['new_msg']--;
            if ( new_msg > 0 ) new_msg--;
            alert_user_msg(u_id);
        }
    }
    alert_msg();
}
```

6.4 功能体验

这节的重点是借助图片对页面中聊天模块的布局进行说明。

会员登录后，会在页面底部加载聊天系统的工具栏，如图6-5所示。在商品列表中，鼠标放到商品图时会出现“在线客服”的图标，查询到在线状态时显示蓝色图标，离线时用灰色的显示。在有新消息提醒时，工具栏上的“小鸟”图标会闪动，点击可打开对话框查看消息内容。单击工具栏右侧的箭头可打开联系人列表，分为好友和最近联系人两栏，具体见图6-6。



图6-5 聊天工具栏

聊天对话框的布局如图6-6所示。点开联系人列表，显示头像和会员名，也能看到每个会员的在线状态。单击“我的好友”和“最近联系人”可对相应的

列表进行收起和展开操作。单击会员头像打开聊天对话框，当同时与多个会员聊天时，对话框左侧会出现对话中的会员列表，有新消息时也会在左侧进行标识。对话框中间部分显示双方的消息内容，为方便区分，当前会员发出的消息靠右显示，收到的他人消息靠左，在内容较多时会出现滚动条。单击对话框右上角的“关闭”按钮可结束对话，单击“联系人”右上角的按钮能收起会员列表到默认状态。



图6-6 聊天对话框

在对话框中单击“聊天记录”可打开聊天记录查询窗口，默认显示7天内的消息内容，也可以查看15天、30天内的数据，单击底部的“左右箭头”按钮可以进行翻页。



图6-7 查看聊天记录

6.5 开发技巧

本章的代码与其他章节的有所不同，主要集中在JS部分，而在实际工作中编写和调试这部分程序比PHP的要麻烦些，在开发前要对相关技术有深入理解，这样出现问题才能及时找到。

我们介绍的核心功能，只是整个聊天系统中一个组成部分，依靠这些还达不到能应用到项目中的标准，就像生产汽车，一台好的发动机和合适的车架是不能少的，但一辆真正能开动的汽车还需要轮胎、车座等其他部件。在系统开发时，数据在页面中展示方式与用户的操作习惯也很重要，要让商城会员使用起来感到简单方便，还要能实现交流对话的目的，平衡各方面因素，吸收利用大家的意见对开发会有不少帮助。

6.6 本章小结

本章以商城会员系统基础开发了实时聊天系统，涉及的内容较多，其中开发思路最为重要，它决定了整个系统的成败。代码实现过程和按图纸盖大楼一样，都是从想象到实体的步骤，如果设计有缺陷，那大楼就很难盖成，在系统开发过程中团队内要经常进行沟通协调，相互帮助解决问题，这样才能开发出一个优秀的系统。

第7章 移动应用套件的设计与实现

随着时代与技术的进步，人们对移动性和信息的需求急速上升，移动互联网已经渗透到人们生活、工作的各个领域。随着3G时代的到来，移动电子商务成为各个产业链竞相争抢的“大蛋糕”。移动电子商务可以为用户随时随地提供所需的服务、应用、信息和娱乐，同时满足用户及商家从众、安全、社交及自我实现的需求，因而深受用户的欢迎。移动电子商务成为了电子商务发展的未来趋势。移动电子商务作为一种新型的电子商务方式，利用了移动无线网络的优点，是对传统电子商务的有益补充。尽管目前移动电子商务的开展还存在安全与带宽等很多问题，但是相比与传统的电子商务方式，移动电子商务具有诸多优势，得到了世界各国的普遍重视，发展和普及速度很快。

当然，由于基于固定网的电子商务与移动电子商务拥有不同特征，移动电子商务不可能完全替代传统电子商务，两者是相互补充、相辅相成的。移动通信所具有的灵活、便捷的特点，决定了移动电子商务应当定位于大众化的个人消费领域，应当提供大众化的商务应用，因此B2B可能成为移动电子商务发展的主要模式。

7.1 设计思路

一个好的产品，不光要有技术强大的团队，还需要完美的产品设计。从古至今人类有过无数场战争，而每场战争胜利的关键不单单只是靠强大的军队和雄厚的物质资源，更需要的是一个思维敏捷、方向正确的领导者，由这样的人设计出完美的方案才能战胜敌人。

7.1.1 产品设计

其实我们都知道，自2003年淘宝问世以来，这几年中国互联网领域比较火的就是电子商务了，网购也像逛超市一样平常。这种新型的网络交易平台已经完全被广大的用户群体所接受，并且正在慢慢改变我们的生活，成为习惯。最近几年很多实体店也都纷纷转向电子商务领域，那些跟不上时代步伐的传统店铺在今后的几年也必将被电商取代。而2013年以后传统电商发展并不乐观，很多电商的崛起、用户需求不断提高、发展速度过快把传统电商推到了瓶颈，随着移动设备的普及，传统电子商务企业把矛头转向移动电商。为什么移动电商有这么大的影响力？我们来简单分析一下移动电商应用的几个特点。

1) 便捷性：与传统电商相比移动电商最大的特点就是随时随地的特性。传统电商局限于电脑上，携带不方便，而移动电商则可以让人们随时随地浏览、购物。

2) 及时性：移动终端几乎每个人每天都会带在身上，当有新的折扣消息通知时，能即时推送到移动终端，方便人们即时抢购所需的商品。

3) 用户量：随着移动设备的普及，移动终端会逐渐超过pc的用户量，移动的时代才刚刚开始。

7.1.2 设计原则

1) 设计观念：从传统方向转向移动领域，设计者们首先要做的是丢弃旧的思维模式和设计理念，重新认识移动产品的特性。

- 专注：移动应用的本质是给人以最高的效率、简单、便捷的完成特定的任务，所以移动功能设计要比传统产品功能少很多。

- 独特：在产品设计之前就要清楚你的应用和同类产品相比有什么独特之处，将其体现到产品的整体用户体验策划当中，并在交互、视觉、设计流程当中突出卖点。

- 迷人：移动设备是相当个人化的工具，通常人们会由于长久使用而习惯、依赖这个软硬件。应用程序

序同样要与用户在情感上产生互动，通过友好、有趣、可信赖的设计和功能让用户爱不释手。

- 体贴：不要全部注意力都投入到开发上，一个好的产品不光是闷头苦干就能写出来的，在开发中要设身处地、身临其境，站在用户的角度去体验和开发这款产品，这样开发出来的产品更人性化，也能够被更多用户所接受。

2) 使用环境：提到设备的使用环境，我们通常会先想到商业巨头在机场提着皮箱、摆弄手机的画面。不过这只是移动应用使用环境的其中之一，我们需要考虑的使用情景可以归纳为3类：打发无聊、忙碌工作、身处异地。

打发无聊：很多用户闲暇之时无事可做就会窝在家里的沙发上摆弄手机，在这种情况下人机会话时间

长，需求则是偏娱乐休闲，所以令人愉悦的体验感是必需的。

- 忙碌工作：现在手机几乎人人具备，因此手机应用必须能帮助用户快速完成一些小的任务目标，界面当中的文字及交互对象要够大、够清晰、够简洁，这样用户即使在不稳定的情况下依然可以操作。

- 身处异地：针对用户在出差、旅途或是陌生的环境的情况，不仅要考虑可读性和易用性的问题，还要考虑网络连接、电量等方面。针对不同类别的产品做出相对合理的设计和优化，尽量精简那些可能造成电量消耗的特性功能。

3) 通用的设计规范：移动应用类型大致分为三种。实用型、效率型、沉浸型，不同类型的应用有不同的设计与开发规则。在为小尺寸触屏移动设备的应

用进行设计的过程中，有一些全局性的规范需要我们注意：

- 响应性：如果用户执行了操作，应用应该立刻做出响应给用户，让用户知道应用还在运行，不然谁都不知道应用是挂掉了还是在运行中。

- 细节：在设计和开发过程中可能没有在意的细节，用户在使用过程中轻而易举地注意到了。所以游戏的细节设计对产品整体的体验感很有帮助。想象一个界面外观很华丽的应用，在用户使用过程中一直崩溃，那么用户的体验感显然会被大打折扣的。

- 拇指：关于触屏操作，其实在多数情况下，我们只是在为拇指进行设计，除非你确定用户会同时使用两只手来操作设备。即使用户是在双手持机，通常也是使用两只拇指进行操作。所以，设计方案通常需要围绕着拇指进行考虑。

- 点击目标：看看自己拇指用来触控界面的地方。无论怎么看上去形状都只是像个瓶盖儿。即使长的再好看，对于移动设备界面当中那些微小的操作对象来说也显得大了许多。

- 相邻元素之间的布局关系：这一点也需要留意的，例如，在创建内容类的应用中，将“返回”按钮直接搁在“保存”按钮旁边就不是个好主意。

- 内容：“直接操纵”是触屏设备交互模式的根基。用户可以直接与界面进行交互，而不需要鼠标一类的辅助设备。

这种模式的优点就不用多说了，现在几岁的小孩子都可以轻松使用pad和phone，但他们可能还搞不定计算机。对于运行在触屏设备当中的应用来说，更是要精简界面元素，使内容呈现方式与功能的操作方法尽可能符合“直接操纵”框架下的直觉模式。

- 控制元素的布局：在通常情况下，具有控制功能的界面元素需要放置在内容的下方，这样我们在执行操作的时候才不会将内容遮挡住。想想看手机的home键都是在底部好像还没有放到上面的。虽然传统的桌面软件和Web页面都是将相关的导航和操作放在顶部的，但那只适用于鼠标操纵的情况，毕竟指针的尺寸相对于整个显示设备来说是微不足道的。

- 滚屏：在多数时候，“折线以下”的忌讳同样存在于移动应用的界面设计当中。另外，对于某些应用来说，单屏模式可以带来更加坚实可靠的感觉，因为所有的内容都是可预知的。

当然，在多数应用中，避免滚屏是不现实的，不过可以思考一下是否有什么方式可以减少滚屏带来的不确定感，让用户了解到那些还没有进入可视范围的内容的存在。

4) 第一印象

- 应用图标：对于一个应用来说，第一印象很重要。从应用的图标开始就要给用户留下深刻的印象，这样能让你的应用图标在各种同类产品的图标海洋中脱颖而出。

- 首次启动：首次启动的表现虽然只显示一次，但是它往往可以决定一款应用的成功或失败。如果新的用户在这里就感到困惑和受挫，那么他们很快会放弃这款产品的使用。如果你的应用软件很烦琐、复杂，那么可以考虑增加一些引导来帮助用户在短时间内对产品能了解和操作。

7.2 代码实现

7.2.1 框架结构介绍

因为手机的限制，Android开发其实不会像Web开发一样有很庞大的框架。从开发者的角度，看到整齐的代码和优美、有层次的框架是一种享受，从艺术角度我们是在追求一种美。下面是我们产品的架构，如图7-1和图7-2所示。

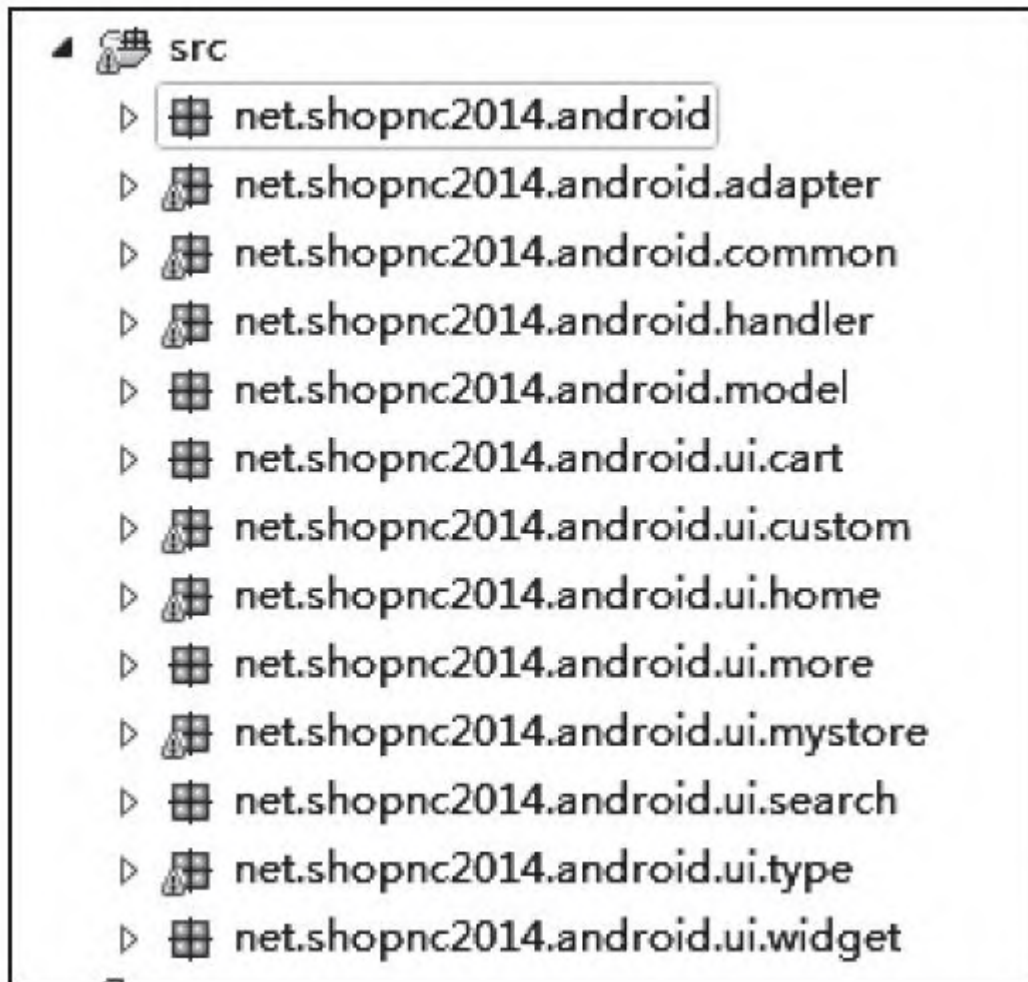


图7-1 产品框架目录

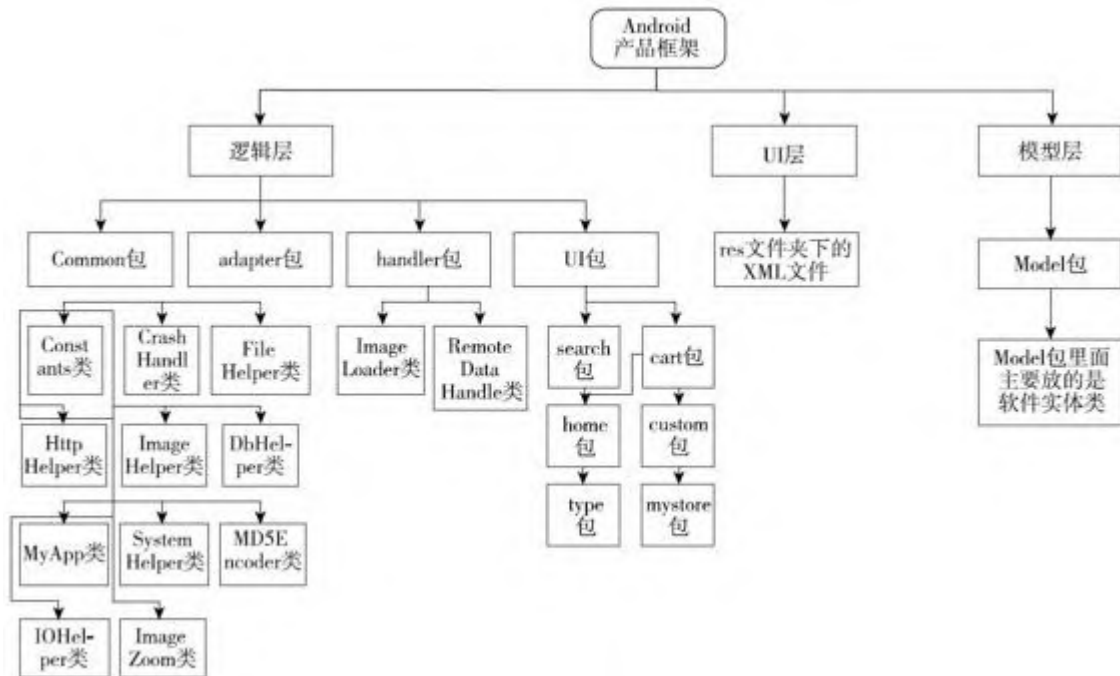


图7-2 产品框架目录流程图

7.2.2 CrashHandler工具类

大家知道，无论你的Android应用写得多么完美，发生crash都是不可避免的，可能是因为Android系统底层的Bug或者是不同机型适配导致的。当crash发生的时候，系统就会自动终止应用，使应用闪退或停止运行，这对用户体验感有很大影响，这也是开发者、Boss不想看到的。更糟糕的是，在用户使用过程中发生了crash，而开发者却无法知道Bug的日志，不能找到发生crash的原因并解决。其实Android提供了处理这样问题的方法，如代码清单7-1所示。

代码清单7-1 CrashHandler工具类

```
public class CrashHandler implements
UncaughtExceptionHandler {
    // 调试日志标记
    public static final String TAG = "CrashHandler";
    //是否开启日志输出,在Debug状态下开启, 在Release状态下关闭以
提示程序性能
    public static final boolean DEBUG = false;
```

```

        //系统默认的UncaughtException处理类
        private Thread.UncaughtExceptionHandler
mDefaultHandler;
        // CrashHandler实例
        private static CrashHandler INSTANCE;
        //程序的Context对象
        private Context mContext;
        //使用Properties来保存设备的信息和错误堆栈信息
        private Properties mDeviceCrashInfo = new
Properties();
        private static final String VERSION_NAME =
"versionName";
        private static final String VERSION_CODE =
"versionCode";
        private static final String STACK_TRACE =
"STACK_TRACE";
        //错误报告文件的扩展名
        private static final String CRASH_REPORTER_EXTENSION
= ".cr";
        //保证只有一个CrashHandler实例
        private CrashHandler() {}
        //获取CrashHandler实例 ,单例模式 */
        public static CrashHandler getInstance() {
            if (INSTANCE == null) {
                INSTANCE = new CrashHandler();
            }
            return INSTANCE;
        }
        /**
         * 初始化,注册Context对象, 获取系统默认的
UncaughtException处理器, 设置该CrashHandler
         为程序的默认处理器
         * @param ctx
         */
        public void init(Context ctx) {
            mContext = ctx;
            mDefaultHandler =
Thread.getDefaultUncaughtExceptionHandler();

Thread.setDefaultUncaughtExceptionHandler(this);
        }
        /**
         * 当UncaughtException发生时转入该函数来处理
         */
        @Override

```

```

        public void uncaughtException(Thread thread,
        Throwable ex) {
            if (!handleException(ex) && mDefaultHandler
            != null) {
                // 如果用户没有处理则让系统默认异常处理
                器来处理
                mDefaultHandler.uncaughtException(thread, ex);
            } else {
                // Sleep一会后结束程序
                try {
                    Thread.sleep(5000);
                } catch (InterruptedException e) {
                    Log.e(TAG, "Error : ", e);
                }

                android.os.Process.killProcess(android.os.Process.myPid());
                System.exit(10);
            }
        }
    /**
     * 自定义错误处理,收集错误信息发送错误报告等操作均在此完成.开
     发者可以根据自己情况来自定义
     异常处理逻辑
     * @param ex
     * @return true:如果处理了该异常信息;否则返回false
     */
    private boolean handleException(Throwable ex) {
        if (ex == null) {
            Log.w(TAG, "handleException ---
ex==null");

            return true;
        }
        final String msg = ex.getLocalizedMessage();
        if (msg == null) {
            return false;
        }
        // 使用Toast来显示异常信息
        new Thread() {
            @Override
            public void run() {
                Looper.prepare();
                Toast toast =
                Toast.makeText(mContext, "程序出错,即将退出:\r\n" + msg,

```

```

Toast.LENGTH_LONG);

toast.setGravity(Gravity.CENTER, 0, 0);
                                toast.show();
                                //
MsgPrompt.showMsg(mContext, "程序出错啦", msg+"\n点`确认`退出");

                                Looper.loop();
                                }
                                }.start();
                                // promptMessageDialog();
                                // 收集设备信息
                                collectCrashDeviceInfo(mContext);
                                // 保存错误报告文件
                                saveCrashInfoToFile(ex);
                                // 发送错误报告到服务器
                                // sendCrashReportsToServer(mContext);
                                return true;
                                }
                                /**
                                * 在程序启动时候, 可以调用该函数来发送以前没有发送的报告
                                */
                                public void sendPreviousReportsToServer() {
                                    sendCrashReportsToServer(mContext);
                                }
                                /**
                                * 把错误报告发送给服务器, 包含新产生的和以前没发送的
                                *
                                * @param ctx
                                */
                                private void sendCrashReportsToServer(Context ctx) {
                                    String[] crFiles = getCrashReportFiles(ctx);
                                    if (crFiles != null && crFiles.length > 0) {
                                        TreeSet<String> sortedFiles = new
TreeSet<String>();

sortedFiles.addAll(Arrays.asList(crFiles));
                                        for (String fileName : sortedFiles)
{
                                            File cr = new
File(ctx.getFilesDir(), fileName);
                                            postReport(cr);
                                            cr.delete(); // 删除已发送的报
告
                                        }

```

```

    }
}
private void postReport(File file) {
    //这里可以写方法发送错误报告到服务器
}
/**
 * 获取错误报告文件名
 *
 * @param ctx
 * @return
 */
private String[] getCrashReportFiles(Context ctx) {
    File filesDir = ctx.getFilesDir();
    FilenameFilter filter = new FilenameFilter()
{
    public boolean accept(File dir,
String name) {
        return
name.endsWith(CRASH_REPORTER_EXTENSION);
    }
};
    return filesDir.list(filter);
}
//应用carsh之后，跳出Dialog，点击之后才会退出系统
private void promptMessageDialog() {
    new Thread() {
        @Override
        public void run() {
            new
AlertDialog.Builder(mContext)

.setTitle("Error")

.setCancelable(false)

.setMessage(
"The
system was crashed. Please restart it later.")

.setNeutralButton("OK", new OnClickListener() {

@Override

public void onClick(Dialog- Interface dialog,

int which) {

```



```

System.exit(-1);
    }

}).create().show();
    }
    }.start();
}
/**
 * 保存错误信息到文件中
 * @param ex
 * @return
 */
private String saveCrashInfoToFile(Throwable ex) {
    Writer info = new StringWriter();
    PrintWriter printWriter = new
PrintWriter(info);
    ex.printStackTrace(printWriter);
    Throwable cause = ex.getCause();
    while (cause != null) {
        cause.printStackTrace(printWriter);
        cause = cause.getCause();
    }
    String result = info.toString();
    printWriter.close();
    mDeviceCrashInfo.put("EXEPTION",
ex.getLocalizedMessage());
    mDeviceCrashInfo.put(STACK_TRACE, result);
    try {
        // long timestamp =
System.currentTimeMillis();
        Time t = new Time("GMT+8");
        t.setToNow(); // 取得系统时间
        int date = t.year * 10000 + t.month
* 100 + t.monthDay;
        int time = t.hour * 10000 + t.minute
* 100 + t.second;
        String fileName = "crash-" + date +
        "-" + time
        +
CRASH_REPORTER_EXTENSION;
        FileOutputStream trace =
mContext.openFileOutput(fileName,
Context.MODE_PRIVATE);

```

```

        mDeviceCrashInfo.store(trace, "");
        trace.flush();
        trace.close();
        return fileName;
    } catch (Exception e) {
        Log.e(TAG, "an error occurred while
writing report file...", e);
    }
    return null;
}
/**
 * 收集程序崩溃的设备信息
 * @param ctx
 */
public void collectCrashDeviceInfo(Context ctx) {
    try {
        PackageManager pm =
ctx.getPackageManager();
        PackageInfo pi =
pm.getPackageInfo(ctx.getPackageName(),
PackageManager.GET_ACTIVITIES);
        if (pi != null) {

mDeviceCrashInfo.put(VERSION_NAME,

pi.versionName == null ? "not set" : pi.versionName);

mDeviceCrashInfo.put(VERSION_CODE, "" + pi.versionCode);
        }
    } catch (NameNotFoundException e) {
        Log.e(TAG, "Error while collect
package info", e);
    }
    // 使用反射来收集设备信息.在Build类中包含各种设备信
    息,
    // 例如: 系统版本号,设备生产商等帮助调试程序的有用
    信息
    // 具体信息请参考后面的截图
    Field[] fields =
Build.class.getDeclaredFields();
    for (Field field : fields) {
        try {
            field.setAccessible(true);

```

```

mDeviceCrashInfo.put(field.getName(), "" + field.get(null));
                    if (DEBUG) {
                        Log.d(TAG,
field.getName() + " : " + field.get(null));
                    }
                } catch (Exception e) {
                    Log.e(TAG, "Error while
collect crash info", e);
                }
            }
        }
    }
}

```

7.2.3 DbHelper工具类

SQLite是一个小型的数据库，它是Google为Android处理大数据量的数据而提供的。想要用Android操作SQLite必须使用SQLiteOpenHelper类。SQLiteOpenHelper是一个抽象类，负责数据库的创建和版本的管理，必须要继承才能使用，所以DbHelper继承了SQLiteOpenHelper类来帮SQLiteOpenHelper完成管理数据的工作，如代码清单7-2所示。

代码清单7-2 操作DbHelper类

```
public class DBHelper extends SQLiteOpenHelper {
    private static String db_name;
    static{
        db_name = Constants.DB_NAME;
    }
    public DBHelper(Context context){
        super(context, db_name, null,
Constants.DB_VERSION);
    }
    public DBHelper(Context context, String name,
CursorFactory factory,
        int version) {
        super(context, name, factory, version);
    }
}
```

```

    }
    //在第一个安装本应用程序时，会回调的方法，主要用于执行数据库表的
    创建和数据初始化
    @Override
    public void onCreate(SQLiteDatabase db) {
        System.out.println("执行oncreate-----");
        //db.execSQL(Constants.SQL_USER_CREATE);
        //注意这个SqliteDatabase实例不需要关闭
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int
oldVersion, int newVersion) {
        System.out.println("执行onUpgrade-----");
        //清理旧版本中的表
        //db.execSQL(Constants.SQL_USER_DROP);
        //重新创建新的表
        onCreate(db);
    }
    public SQLiteDatabase getSQLiteDatabase(){
        SQLiteDatabase db = null;
        try{
            db = this.getWritableDatabase();
        }catch (Exception e) {
            db = this.getReadableDatabase();
        }
        return db;
    }
}

```

这个DbHelper类只是完成对数据库和数据库表的管理，如果想对数据库做增删改查的工作，我们还需要针对实体的Dao类来对数据完成增删改查的操作。下面我以查询为例，如代码清单7-3所示。

代码清单7-3 操作DbHelper的Dao类

```
public class HistoryDao {
    private DbHelper dbHelper;
    public HistoryDao(Context context) {
        dbHelper = new DbHelper(context);
    }
    /**
     * 查询所有数据
     * @param null
     * @return ArrayList<NotePad>
     * */
    public ArrayList<ScanningHistory> findall() {
        ArrayList<ScanningHistory> shList = new
ArrayList<ScanningHistory>();
        SQLiteDatabase db = null;
        try {
            db = dbHelper.getSQLiteDatabase();
            db.beginTransaction();
            Cursor c =
db.rawQuery(Constants.SQL_SCANNING_SELECT_ALL, null);
            while (c.moveToNext()) {
                ScanningHistory NotePad =
new ScanningHistory();

                NotePad.setScanningMsg(c.getString(c.getColumnIndex("scannin
gMsg")));

                NotePad.setScanningTime(c.getString(c.getColumnIndex
("scanningTime")));

                shList.add(NotePad);
            }
            db.setTransactionSuccessful();
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            if (db != null) {
                db.endTransaction();
                db.close();
            }
        }
        return shList;
    }
}
```

7.2.4 HttpHelper工具类

HTTP协议应该是Internet上使用最多、最重要的协议了，大多数Java需要直接通过HTTP协议来访问网络资源。JDK的java.net包中已经提供了访问HTTP协议的基本功能URLConnection，但是对于Android大部分程序来讲，JDK提供的这个功能还不够灵活，所以Android同样集成了Apache提供的HttpClient，用来提供高效的、最新的、功能丰富的支持HTTP协议工具包。简单来说HttpClient就是URLConnection的加强版，URLConnection可以做的事HttpClient同样可以做，URLConnection做不到的HttpClient也能做到，但是HttpClient侧重于如何发送、接收响应，以及管理HTTP连接。我们的HttpHelper就是对HttpClient的封装，部分代码如代码清单7-4所示。

代码清单7-4 HTTP协议工具类

```
public class HttpHelper {
    /** 本身就是线程安全的 */
    private static HttpClient httpClient;
    static {
        if(null == httpClient){
            //httpClient = new
DefaultHttpClient();

//httpClient.getParams().setParameter(CoreProtocolPNames.PROTO
COL_VERSION, HttpVersion.HTTP_1_1);

//httpClient.getParams().setIntParameter(CoreConnectionPNames.
CONNECTION_TIMEOUT, 20000);
//以下代码处理了同一个HttpClient同时发出多
个请求时可能发生的多线程问题
        HttpParams httpParams = new
BasicHttpParams();

        HttpProtocolParams.setVersion(httpParams,
HttpVersion.HTTP_1_1);

        HttpProtocolParams.setContentCharset(httpParams, HTTP.UTF_8);

        HttpProtocolParams.setUseExpectContinue(httpParams, true);
        // 设置最大连接数

        ConnManagerParams.setMaxTotalConnections(httpParams, 10000);
        // 设置获取连接的最大等待时间
        ConnManagerParams.setTimeout(httpParams,
60000);
        // 设置每个路由最大连接数
        ConnPerRouteBean connPerRoute = new
ConnPerRouteBean(10000);

        ConnManagerParams.setMaxConnectionsPerRoute(httpParams,connPer
Route);
        // 设置连接超时时间

        HttpConnectionParams.setConnectionTimeout(httpParams, 20000);
        // 设置读取超时时间
        HttpConnectionParams.setSoTimeout(httpParams,
30000);

        SchemeRegistry schreg = new
SchemeRegistry();
        schreg.register(new Scheme("http",
```



```

PlainSocketFactory.getSocketFactory(), 80));
        schreg.register(new Scheme("https",
SSLSocketFactory.getSocketFactory(), 443));
        ClientConnectionManager connManager =
new ThreadSafeClientConn-Manager(httpParams, schreg);
        httpClient = new
DefaultHttpClient(connManager, httpParams);
    }
}
public static HttpClient getHttpClient(){
    return httpClient;
}
/**
 * 发送GET请求, 并返回响应消息体的字符串内容
 * @param url 请求URL
 * @return 响应消息体的字符串内容
 * @throws IOException
 */
public static String get(String url) throws
IOException{
    String result = null;
    HttpGet get = new HttpGet(url);
    HttpResponse response = httpClient.execute(get);
    if(HttpStatus.SC_OK ==
response.getStatusLine().getStatusCode()){
        result =
EntityUtils.toString(response.getEntity());
    }
    return result;
}
/**
 * 发送POST请求, 并返回响应消息体的字符串内容
 * @param url 请求URL
 * @return 响应消息体的字符串内容
 * @throws IOException
 */
public static String post(String url, HashMap<String,
String> params)throws IOException{
    String result = null;
    HttpPost post = new HttpPost(url);
    if(null != params){
        List<NameValuePair> pairList = new
ArrayList<NameValuePair>();
        for (Entry<String, String> paramPair :
params.entrySet()) {
            NameValuePair pair = new

```

```

BasicNameValuePair(paramPair.getKey()
paramPair.getValue());
pairList.add(pair);
}
HttpEntity entity = new
UrlEncodedFormEntity(pairList, HTTP.UTF_8);
post.setEntity(entity);
}
HttpResponse response =
httpClient.execute(post);
if(HttpStatus.SC_OK ==
response.getStatusLine().getStatusCode()){
result =
EntityUtils.toString(response.getEntity());
}
return result;
}
}

```

7.2.5 ImageHelper工具类

ImageHelper工具类封装的是图片处理的一些方法，在开发中我们可能会用到很多次对图片操作的方法，比如下载图片、加载本地图片、加载assets等操作，下面的代码是这个类中部分方法介绍。如代码清单7-5所示。

代码清单7-5 图片处理的类

```
public class ImageHelper {
    public static Bitmap getHttpBitmap(String url){
        URL myurl = null;
        InputStream in = null;
        Bitmap bitmap=null;
        try{
            myurl = new URL(url);
            HttpURLConnection conn =
                (HttpURLConnection)myurl.openConnection();
            conn.setConnectTimeout(0);
            conn.setDoInput(true);
            conn.connect();
            in = conn.getInputStream();
            bitmap = BitmapFactory.decodeStream(in);
        }catch (IOException e){
            e.printStackTrace();
        }
        finally{
```

```

        try {
            if (null != in){ in.close(); }
        }catch(IOException e) {
            e.printStackTrace();
        }
    }
    return bitmap;
}
/**
 * 加载本地图片
 *
 * @param url
 * @return
 */
public static Bitmap getLoacalBitmap(String url) {
    try {
        FileInputStream fis = new
FileInputStream(url);
        return
BitmapFactory.decodeStream(fis); // 把流转化为Bitmap图片
    } catch (FileNotFoundException e) {
        e.printStackTrace();
        return null;
    }
}
/**
 * 根据资源ID获取对应的Bitmap实例
 * @param ctx
 * @param id
 * @return
 */
public static Bitmap getBitmap(Context ctx, int id){
    return
BitmapFactory.decodeResource(ctx.getResources(), id);
}
/**
 * 把png或jpg(jpeg) 格式图片按指定名称写入指定目录下
 * @param bmp
 * @param path
 * @param fileName
 */
public static void write(Bitmap bmp, File file){
    FileOutputStream fos = null;
    try {
        if(!file.exists()){

```

```

        file.createNewFile();
    }
    fos = new FileOutputStream(file);

    bmp.compress(Bitmap.CompressFormat.PNG, 75, fos);
    fos.flush();
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } finally{
        if(null != fos){
            try {
                fos.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
/**
 * 把png或jpg(jpeg) 格式图片按指定名称写入指定目录下
 * @param bmp
 * @param path
 * @param fileName
 */
public static void write(Bitmap bmp, String
fileName){
    String extension =
IOHelper.getExtension(fileName);
    FileOutputStream fos = null;
    try {
        File file = new File(fileName);
        if(!file.exists()){
            file.createNewFile();
        }
        fos = new FileOutputStream(file);

        if("png".equalsIgnoreCase(extension)){

            bmp.compress(Bitmap.CompressFormat.PNG, 75, fos);
        }else
            if("jpg".equalsIgnoreCase(extension) ||
            "jpeg".equalsIgnoreCase(extension)){

```

```

        bmp.compress(Bitmap.CompressFormat.JPEG, 75, fos);
    }
    fos.flush();
} catch (FileNotFoundException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
} finally{
    if(null != fos){
        try {
            fos.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

/**
 * 把png或jpg(jpeg) 格式图片按指定名称写入指定目录下
 * @param bmp
 * @param path
 * @param fileName
 */
public static void write(Bitmap bmp, String path,
String fileName){
    String extension =
IOHelper.getExtension(fileName);
    FileOutputStream fos = null;
    try {
        fos = new FileOutputStream(new
File(path, fileName));

        if("png".equalsIgnoreCase(extension)){
            bmp.compress(Bitmap.CompressFormat.PNG, 75, fos);
        }else
        if("jpg".equalsIgnoreCase(extension) ||
"jpeg".equalsIgnoreCase(extension)){
            bmp.compress(Bitmap.CompressFormat.JPEG, 75, fos);
        }
        fos.flush();
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {

```

```

        e.printStackTrace();
    } finally{
        if(null != fos){
            try {
                fos.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}

public static Bitmap loadFromFile(File name) throws
IOException{
    Bitmap bmp = null;
    InputStream is = null;
    try {
        is = new FileInputStream(name);
        bmp =
BitmapFactory.decodeStream(is);
    } catch (IOException e) {
        throw e;
    }finally{
        if(null != is){
            try {
                is.close();
            } catch (IOException e) {
                throw e;
            }
        }
    }
    return bmp;
}

public static Bitmap loadFromFile(String name)
throws IOException{
    Bitmap bmp = null;
    InputStream is = null;
    try {
        is = new FileInputStream(name);
        bmp =
BitmapFactory.decodeStream(is);
    } catch (IOException e) {
        throw e;
    }finally{
        if(null != is){
            try {

```

```

        is.close();
    } catch (IOException e) {
        throw e;
    }
}

return bmp;
}

/**
 * 根据图片名称, 从assets中加载对应的Bitmap对象
 * @param ctx
 * @param fileName
 * @return
 */
public static Bitmap getImageFromAssetsFile(Context ctx,
String fileName){
    Bitmap image = null;
    AssetManager am = ctx.getResources().getAssets();
    InputStream is = null;
    try{
        is = am.open(fileName);
        image = BitmapFactory.decodeStream(is);
    }catch (IOException e){
        e.printStackTrace();
    }finally{
        if(null != is){
            try {
                is.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
    return image;
}
}

```

7.2.6 SystemHelper工具类

SystemHelper工具类是对一些获取系统信息的方法的封装，如代码清单7-6所示。

代码清单7-6 通用常用的设置方法

```
public class SystemHelper {
    private SystemHelper() {
    }
    /**
     * 创建本应用的桌面快捷方式<br/>
     * 注意：需要添加权限<uses-permission
android:name="com.android.launcher.permission.INSTALL_SHORTC
UT"/>
     * @param paramContext
     */
    public static void createShortcut(Context context,
Class<?> clazz) {
        Intent shortcut = new
Intent("com.android.launcher.action.INSTALL_SHORTCUT");
        //快捷方式的名称

        shortcut.putExtra(Intent.EXTRA_SHORTCUT_NAME,
context.getString(R.string.app_name));
        shortcut.putExtra("duplicate", false); //不
允许重复创建

        Intent localIntent2 = new Intent(context,
clazz);
        localIntent2.setAction(Intent.ACTION_MAIN);

        localIntent2.addCategory(Intent.CATEGORY_LAUNCHER);
        shortcut.putExtra(Intent.EXTRA_SHORTCUT_INTENT,
```

```

localIntent2); //指定快捷方式要启动的Activity类型
                //快捷方式的图标
                ShortcutIconResource iconResource =
Intent.ShortcutIconResource.fromContext(context,
R.drawable.ic_launcher);

shortcut.putExtra(Intent.EXTRA_SHORTCUT_ICON_RESOURCE,
iconResource);
                context.sendBroadcast(shortcut);
        }
        /**
        * 检查是否已经创建了桌面快捷方式<br/>
        * 注意：需要添加权限<uses-permission
android:name="com.android.launcher.permission.READ_SETTINGS"
/>
        * @param ctx
        * @return
        */
        public static boolean hasShortCut(Context context) {
                String url = "";
                if (android.os.Build.VERSION.SDK_INT < 8) {
                        url =
"content://com.android.launcher.settings/favorites?notify=
true";
                } else {
                        url =
"content://com.android.launcher2.settings/favorites?notify=
true";
                }
                ContentResolver resolver =
context.getContentResolver();
                Cursor cursor =
resolver.query(Uri.parse(url), null, "title=?",
                        new String[] {
context.getString(R.string.app_name) }, null);
                if (cursor != null && cursor.moveToFirst())
        {
                        cursor.close();
                        return true;
                }
                return false;
        }
        /**
        * 获取当前机器的屏幕信息对象<br/>
        * 另外：通过android.os.Build类可以获取当前系统的相关信息

```

```

        * @param context
        * @return
        */
        public static DisplayMetrics getScreenInfo(Context
context) {
            WindowManager windowManager =
(WindowManager) context

.getService(Context.WINDOW_SERVICE);
            DisplayMetrics dm = new DisplayMetrics();

windowManager.getDefaultDisplay().getMetrics(dm);
            // dm.widthPixels; //宽度
            // dm.heightPixels; //高度
            // dm.density; //密度
            return dm;
        }
        /**
        * 获取手机号<br/>
        * 注意: 需要添加权限<uses-permission
        *
        android:name="android.permission.READ_PHONE_STATE"/>;。另外
        很多手机不能获取到当前手机号
        *
        * @param context
        * @return
        */
        public static String getMobileNumber(Context
context) {
            TelephonyManager tm = (TelephonyManager)
context

.getService(Context.TELEPHONY_SERVICE);
            return tm.getLine1Number();
        }
        /**
        * 检测当前的网络连接是否可用<br/>
        * 注意: 需要添加权限<uses-permission
        *
        android:name="android.permission.ACCESS_NETWORK_STATE"/>;
        *
        * @param context
        * @return
        */
        public static boolean isConnected(Context context) {

```

```

        boolean flag = false;
        try {
            ConnectivityManager connManager =
(ContextivityManager) context

.getService(Context.CONNECTIVITY_SERVICE);
            if (null != connManager) {
                NetworkInfo info =
connManager.getActiveNetworkInfo();
                if (null != info &&
info.isAvailable()) {
                    flag = true;
                }
            }
        } catch (Exception e) {
            Log.e("NetworkInfo", "Exception",
e);
        }
        return flag;
    }
    /**
     * 检测当前网络连接的类型<br/>
     * 注意：需要添加权限<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE"/>;
     * @param context
     * @return 返回0代表GPRS网络;返回1,代表WIFI网络;返回-1代
表网络不可用
     */
    public static int getNetworkType(Context context) {
        int code = -1;
        try {
            ConnectivityManager connManager =
(ContextivityManager)
context.getService(Context.CONNECTIVITY_SERVICE);
            if (null != connManager) {
                State state =
connManager.getNetworkInfo(ConnectivityManager.
TYPE_WIFI).getState();
                if (State.CONNECTED ==
state) {
                    code =
ConnectivityManager.TYPE_WIFI;
                } else {
                    state =
connManager.getNetworkInfo(Connectivity-

```

```

Manager.TYPE_MOBILE).getState();
                                if (State.CONNECTED
== state) {
                                code =
ConnectivityManager.TYPE_MOBILE;
                                }
                                }
                                } catch (Exception e) {
                                Log.e("NetworkInfo", "Exception",
e);
                                }
                                return code;
}
/**
 * 返回当前程序版本代码,如:1
 * @param context
 * @return 当前程序版本代码
 */
public static int getAppVersionCode(Context context)
{
    int versionCode = -1;
    try {
        PackageManager pm =
context.getPackageManager();
        PackageInfo pi =
pm.getPackageInfo(context.getPackageName(), 0);
        versionCode = pi.versionCode;
    } catch (Exception e) {
        Log.e("VersionInfo", "Exception",
e);
    }
    return versionCode;
}
/**
 * 返回当前程序版本名,如:1.0.1
 *
 * @param context
 * @return 当前程序版本名
 */
public static String getAppVersionName(Context
context) {
    String versionName = "";
    try {
        PackageManager pm =

```

```

context.getPackageManager();
        PackageInfo pi =
pm.getPackageInfo(context.getPackageName(), 0);
        versionName = pi.versionName;
    } catch (Exception e) {
        Log.e("VersionInfo", "Exception",
e);
    }
    return versionName;
}
/**
 * 安装指定的APK文件，主要用于本应用程序的更新
 *
 * @param context
 * @param apk(apk文件的全路径名)
 */
public static void installAPK(Context context,
String apk) {
    Intent intent = new
Intent(Intent.ACTION_VIEW);
        intent.setDataAndType(Uri.fromFile(new
File(apk)),
"application/vnd.android.package-archive");
        context.startActivity(intent);
}

```

7.3 相关工具及第三方框架介绍

7.3.1 原型设计工具介绍

产品的原型可以概括整个产品，为整个产品外观和交互的框架设计，简单来说是将产品各个模块中的界面元素和交互的形式利用图形描述的方法更加具体、生动地表达出来。原型设计是交互设计师与产品设计师、产品经理以及开发工程师沟通的最好工具。在策划方案制订后，交互设计师根据策划方案的思路在原型设计工具上把产品设计形式设计出来，这就是原型图。常见的原型设计工具有PPT、Visio、Axure等。Axure是一款非常强大的原型设计工具（官方是<http://www.axure.org/>）。

现在让我们来了解一下Axure的简单功能和使用，如图7-3所示是Axure的主页。



图7-3 Axure主界面操作

如图7-4所示，圈1和圈2是Axure提供的元件库，直接拖到右面这个工作区就可以了，很方便、简洁；圈3是预览功能，点击之后会自动打开浏览器生成的设计图。而且Axure库的元素有些是动态的。



图7-4 Axure元件库

Axure软件还有很多强大地方，比如网站架构图、示意图、流程图、交互设计、规格文档等，它可以直接生成图片，也可以生成流程图，点击图7-5中的“生成流程图”链接，会生成如图7-6的效果图。



图7-5 生成流程图步骤



图7-6 生成流程图效果图

关于Axure其他功能这里就不再赘述了，如有兴趣可以去Axure的社区深入了解一下。

7.3.2 AndroidAnnotations介绍

AndroidAnnotations是一个第三方框架，通过注释来开发应用。使用AndroidAnnotations能大大减少代码量，从而简化代码，便于维护，如代码清单7-7所示。（使用的时候需要在项目中导入AndroidAnnotations的jar包，详细介绍可以去官网看<http://androidannotations.org/>）。

代码清单7-7 AndroidAnnotations使用实例

```
//不需要标题
@NoTitle
// 全屏显示
@Fullscreen
//Eactivity注释可以设置Layout,相当于setContent方法
@EActivity(R.layout.activity_main)
public class MainActivity extends BasicActivity {
    @Bean
    FoodComponent foodComponent;
    @Bean
    OrderComponent orderComponent;
    @Bean
    CalculatorComponent calculatorComponent;
    //AfterViews注释定义的方法会在onCreate方法的
    setContentView后执行
```

```
@AfterViews
public void init() {
    // 手动依赖注入，解决循环问题
    foodComponent.setOrderComponent(orderComponent);

    calculatorComponent.setOrderComponent(orderComponent);
    orderComponent.setFoodComponent(foodComponent);

    derComponent.setCalculatorComponent(calculatorComponent);
}
}
```

AndroidAnnotations的一些常用注释的使用方法：

- `@AfterInject`，定义的方法在类的构造方法执行后执行
- `@AfterTextChange`，定义的方法在TextView及其子类的Text属性改变后执行
- `@AfterViews`，定义的方法在setContentview后执行
- `@Background`，定义的方法在后台线程执行

- `@BeforeTextChanged`，定义的方法在TextView及其子类的Text属性改变前执行

- `@Click`，定义点击监听器

- `@EActivity`，在Activity中启用Annotations

- `@EProvider`，在ContentProvider中启用

Annotations

- `@EReceive`，在BroadcastReceiver中启用

Annotations

- `@Eservice`，在Service中启用Annotations

- `@EView`，在自定义的View的子类中启用

Annotations

- `@Fullscreen`，全屏

· @NoTitle, 无标题栏

7.3.3 轻量级ORM框架ActiveAndroid介绍

ActiveAndroid算是一个Android轻量级的ORM框架。在Android上访问数据库是一件麻烦事，ActiveAndroid把所有混乱的东西和所有的配置简化到只用几个简单的步骤就可以完成，让你无须写一条SQL语句就能完成对sqlite数据库的增删改查操作。（使用的时候需要在项目中导入ActiveAndroid的jar包，详细介绍可以去官网。

<http://www.activeandroid.com/>查看。）

下面开始配置的介绍，首先需要在AndroidManifest.xml中添加这两个name：
AA_DB_NAME（这个name不能改，但是可选，默认的值

是"Application.db"这个值)、AA_DB_VERSION, 如代码清单7-8所示。

代码清单7-8 ActiveAndroid配置

```
<meta-data android:name="AA_DB_NAME"
android:value="ShopNC_2014.db" />
<meta-data android:name="AA_DB_VERSION" android:value="1" />
```

配置完成之后我们创建表, 如代码清单7-9所示。

代码清单7-9 ActiveAndroid使用示例

```
/**
 * 搜索历史Bean
 * */
@Table(name = "shopnc_history_browse")
public class HistoryBrowse extends Model {
    @Column(name = "goodsID")
    public String goodsID;
    @Column(name = "imageUrl")
    public String imageUrl;
    @Column(name = "goodsName")
    public String goodsName;
    @Column(name = "goodsPrice")
    public String goodsPrice;
```

声明表名使用@Table (name="表名")，声明列名使用@Columnn (name="字段名")，这样就可以了。

下面我们来保存数据，如代码清单7-10所示。

代码清单7-10 ActiveAndroid使用示例

```
/**
 * 保存历史浏览
 * */
public static void historyBrowseSava(GoodsList bean)
{
    HistoryBrowse hb = new HistoryBrowse();
    hb.goodsID = bean.getGoods_id();
    hb.goodsName = bean.getGoods_name();
    hb.goodsPrice = bean.getGoods_price();
    hb.imageURL = bean.getGoods_image_url();
    hb.save();
}
```

让我们的Class继承Model，这样只需要用Class实体直接调用Save()方法就可以把数据保存到我们的数据库中。查找、删除等就不一一介绍了，详细介绍可以去官网 (<http://www.activeandroid.com>) 了解。Activeandroid几乎涵盖了所有的常用指令Sava、

Delete、From、Join、Select、Set、Update、
Limit、As、Group by、Desc/Asc等。

7.4 本章小结

本章是对移动产品的简单介绍，内容涵盖了Android开发的主要知识点，从程序的设计理念到框架搭建等。从中我们不仅可以获得Android开发的设计理念和开发框架的思路，还可以了解到比较强大的第三方框架和工具的介绍和使用，从而加深对Android应用开发的认识。

第8章 020套件的设计与实现

8.1 020模式概述

020营销模式又称离线商务模式，是指线上营销线上购买带动线下经营和线下消费。020通过打折、提供信息、服务预订等方式，把线下商店的消息推送给互联网用户，从而将他们转换为自己的线下客户，这就特别适合必须到店消费的商品和服务，比如餐饮、健身、看电影和演出、美容美发等。2013年02P营销模式出现，即本地化的020营销模式，正式将020营销模式带入了本地化进程当中。

虽然020模式与B2C、C2C一样，均是在线支付，但不同的是，通过B2C、C2C购买的商品被装箱后快递至消费者手中，而020则是消费者在线上购买商品与服务后，需去线下享受服务。这是支付模式与为店主创造

客流量的一种结合，对消费者来说，也是一种新的“发现”机制。

8.1.1 O2O模式的产生背景

随着互联网的快速发展，电子商务模式除了原有的B2B、B2C、C2C商业模式之外，近来一种新型的消费模式O2O已快速在市场上发展起来。为什么这种模式能够悄然产生？在B2B、B2C商业模式下，买家在线拍下商品，卖家打包商品，找物流企业把订单发出，由物流快递人员把商品派送到买家手上，完成整个交易过程。这种消费模式已经发展得很成熟，也被人们普遍接受，但是在美国这种电子商务非常发达的国家，在线消费交易比例只占8%，线下消费比例达到92%。正是由于消费者大部分的消费仍然是在实体店中实现，把线上的消费者吸引到线下实体店进行消费，有很大的发展空间，所以有商家开始了这种消费模式。

8.1.2 O2O模式的特点

O2O对用户而言：

- 1) 获取更丰富、全面的商家及其服务的内容信息。
- 2) 更加便捷地向商家在线咨询并进行预售。
- 3) 获得相比线下直接消费较为便宜的价格。

O2O对商家而言：

- 1) 能够获得更多的宣传、展示机会，吸引更多新客户到店消费。
- 2) 推广效果可查、每笔交易可跟踪。

3) 掌握用户数据，大大提升对老客户的维护与营销效果。

4) 通过与用户的沟通、释疑更好地了解用户心理。

5) 通过在线有效预订等方式，合理安排经营节约成本。

6) 对拉动新品、新店的消费更加快捷。

7) 降低线下实体对黄金地段旺铺的依赖，大大减少租金支出。

对O2O平台本身而言：

1) 与用户日常生活息息相关，并且能给用户带来便捷、优惠、消费保障等，能吸引大量高黏性用户。

2) 对商家有强大的推广作用及可衡量的推广效果，可吸引大量线下生活服务商家加入。

3) 数倍于C2C、B2C的现金流。

4) 巨大的广告收入空间及形成规模后更多的盈利模式。

8.1.3 O2O模式的优势与核心

O2O模式的益处在于订单在线上产生，每笔交易可追踪，展开推广效果透明度高。让消费者在线上选择心仪的服务然后到线下享受服务。

另外，O2O电子商务模式的优势主要还体现在以下几个方面。

1) 对于实体供应商而言：以互联网为媒介，利用其传输速度快，用户众多的特性，通过在线营销，增加了实体商家宣传的形式与机会，为线下实体店面降低了营销成本，大大提高营销的效率，而且减少它对店面地理位置的依赖性；同时，实体店面增加了争取客源的渠道，有利于实体店面经营优化，提高自身的竞争。在线预付的方式，方便实体商家直接统计在线推广效果及销售额，有利于实体商家合理规划经营。

2) 对于用户而言：不用出门，可以在线了解商家的信息及所提供服务的全面介绍，还有已消费客户的评价可以借鉴；能够通过网络直接在线咨询交流，减少客户的销售成本；还有在线购买服务，客户能获得比线下消费更便宜的价格。

3) 对于O2O电子商务网站经营者而言：一方面利用网络快速、便捷的特性，而且能为用户带来日常生活实际所需的优惠信息，因此可以快速聚集大量的线上用户；能为商家提供有效的宣传效应，以及可以定量统计的营销效果，因而可以吸引大量线下实体商家，巨大的广告收入及规模经济为网站运营商带来更多盈利模式。

O2O营销模式的核心是在线预付。

数据显示，即使在电子商务最发达的美国，线下消费的比例依旧高达92%。TrialPay创始人兼CEO Alex

Rampell在描述庞大的线下消费规模时举例说，“普通的网络购物者每年花费约1000美元，假使普通美国人每年收入为4万美元，那么剩下的39000美元到哪里了？答案是，大部分都在本地消费了，人们会把钱花在咖啡店、酒吧、健身房、餐厅、加油站、水电工、干洗店和发廊。”

这不仅仅是因为线下的服务不能装箱运送，更重要的是快递本身无法传递社交体验所带来的快乐。但如果能通过O2O模式，将线下商品及服务进行展示，并提供在线支付“预约消费”，这对于消费者来说，不仅拓宽了选择的余地，还可以通过线上对比选择最令人期待的服务，以及依照消费者的区域性享受商家提供的更适合的服务。但如果没有线上展示，也许消费者会很难知晓商家信息，更不用提消费二字了。另外，目前正在运用O2O摸索前行的商家们，也经常会使

用比线下支付要更为优惠的手段吸引客户进行在线支付，这也为消费者节约了不少的支出。

对于本地商家而言，原本线上广告的成效可以直接被转换成实际的购买行为，由于每笔完成的订单在确认页面都有“追踪码”，商家在更为轻松地获知在线营销的投资回报率的同时，还能一并持续深入进行“客情维护”。另外，O2O是一个增量的市场，由于服务行业的企业数量庞大，而且地域性特别强，很难在互联网平台做广告，就如同百度上很少出现酒吧、KTV、餐馆的关键词，但O2O模式的出现，会让这些服务行业的商家们可以到线上展开推广。

从表面上看，O2O的关键似乎是网络上的信息发布，因为只有互联网才能把商家信息传播得更快、更远、更广，可以瞬间聚集强大的消费能力。但实际上，O2O的核心在于在线支付，一旦没有在线支付功

能，O2O中的online不过是替他人做嫁衣罢了。就拿团购而言，如果没有能力提供在线支付，仅凭网购后的自家统计结果去和商家要钱，结果双方无法就实际购买的人数达成精确的统一而陷入纠纷。

在线支付不仅是支付本身的完成，是某次消费得以最终形成的唯一标志，更是消费数据唯一可靠的考核标准。尤其是对提供online服务的互联网专业公司而言，只有用户在线上完成支付，自身才可能从中获得效益，从而把准确的消费需求信息传递给offline的商业伙伴。无论B2C，还是C2C，均是在实现消费者能够在线支付后，才形成了完整的商业形态。而在以提供服务性消费为主，且不以广告收入为盈利模式的O2O中，在线支付更是举足轻重。

8.2 产品设计思路

O2O本地生活平台网站将线下商铺和互联网结合，让互联网成为线下交易的前台，专注于本地生活服务，集生活和购物为一体的专业网站，体现本地生活特色，满足消费者多样性的需求，也为商户提供了很好的展示和销售渠道。商户在线上揽客增加客流，消费者在线下消费，享受更多的优惠，商户可以通过参加团购来提高利润和知名度，还可以设立优惠券和会员卡，吸引更多客户，用户可以对商家进行点评，参加团购，并且用优惠券购物，享受折扣，站点提供更丰富的商户内容、商品信息，把原本分散的内容整合到一个平台上，同时加入BBS类社交圈子功能更能提高网站对用户的黏性与参与度。

系统主要分为8个主要模块：

1) 商户模块：每一个使用O2O本地生活平台的商家都有一个自己的商户管理后台，可以进行商铺信息设置、团购及订单的管理、优惠券的管理、商户首页商品的管理、商户评论的管理及店铺活动信息发布等。

2) 团购模块：该部分是O2O本地生活平台网站的一个重点模块，是网站唯一产生线上交易支付的功能，是平台的主要盈利来源。

3) 优惠券模块：商户可以发布自己店铺的优惠券，是商户的一种传统而有效的促销手段，往往能起到事半功倍的效果。网站用户可以从网页上点击打印优惠券或直接将优惠券信息发送到自己的手机上，之后就可以到店享受相应的优惠服务非常方便。

4) 结算模块：O2O本地生活平台应具备基本的与商家结算团购订单的功能，应可以对每个商家分类分

别设置分佣比例，并有完整的结算流程。

5) 预约模块：网站用户可以在线进行店铺预约。

6) 会员卡模块：网站用户可以免费成为商户的会员，并得到一个会员卡号，拥有会员卡的用户到店消费可以享受一定的折扣优惠。

7) 社区模块：类似BBS或百度贴吧的社区版块，主要为了增加网站的用户黏性和参与度。

8) 会员中心模块：网站的每个用户都拥有自己的会员中心，可以查看自己所购买的团购券、预约状态、收藏的店铺或团购及所拥有的店铺会员卡。如图8-1所示。

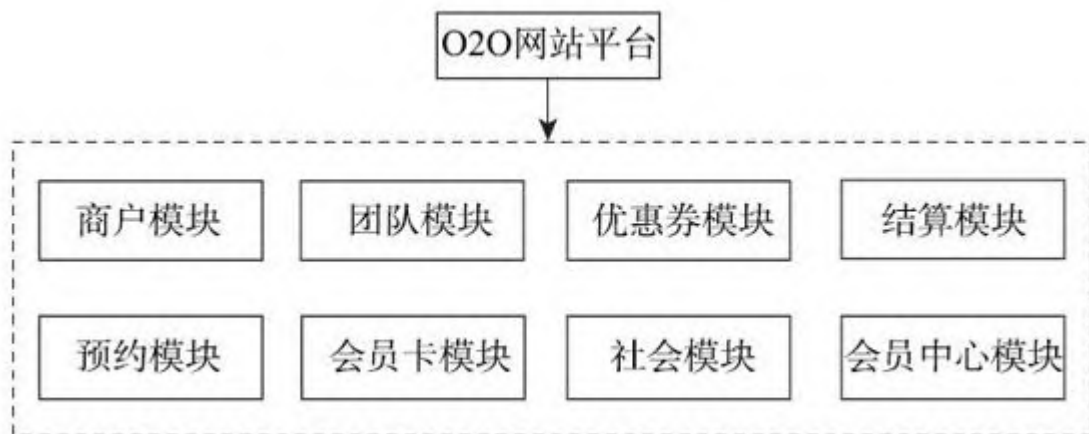


图8-1 O2O网站平台的模块组成

以上是一个O2O本地生活类平台网站应具备的基本功能模块，本章接下来将对前四个重点模块进行详细分析讲解。

8.3 商户模块的设计与实现

8.3.1 模块构成

商户模块由店铺设置、订单管理、团购管理、商品管理、评论管理、优惠券管理、活动管理和结算管理、会员卡管理和预约管理几个部分组成，如图8-2所示。

(1) 店铺设置

商家可以在这个子模块中设置店铺的基本信息，包括店铺名称、店铺描述、店铺分类、店铺地址、店铺LOGO等，此外还可以添加微信二维码图片。

(2) 订单管理

商家可以在这个子模块中查看所有团购购买记录和订单状态并可以查看详情，最重要的是可以使用团购验证功能来验证已卖出的团购券。已验证过的团购券会更改订单状态为已消费并计入结算清单中，同时还会发短信给用户方便其使用团购券。

（3）团购管理

商家可以在这个子模块中发布团购商品并进行编辑等操作，新发布或重新编辑过的团购商品需要平台后台的审核才可以在前台展示。

（4）商品管理

商家可以在这个子模块中发布店铺商品信息，可以填写商品名称、价格、描述，并可以上传一张图片，商品会展示在前台店铺主页上。

（5）评论管理

商家可以在这个子模块中查看用户对商户的评价信息、评价时间等并可以解释说明。

（6）优惠券管理

商家可以在这个子模块中查看已添加的优惠券信息、包括下载次数和浏览次数等，还可以发布新的优惠券，新发布的优惠券需要平台后台审核才可以生效并在前台展示。

（7）活动管理

商家可以在这个子模块中查看已添加的活动信息并添加新活动。添加店铺活动有利于增加用户对商户的关注度和参与度，是一种很好的营销手段。

（8）结算管理

商家可以在这个子模块中查看已生成的结算单，结算单由平台后台生成，商家可以看到起止日期、结算金额、结算日期和结算状态等信息，可以进行确认结算和确认收款两种操作。

（9）会员卡管理

商家可以在这个子模块中选择是否开启会员卡、设置会员卡折扣、添加描述以及上传会员卡图片等操作。此外还可以查看已申请会员卡的用户列表，查看他们的消费记录以及开启或关闭某用户的会员卡资格。

（10）预约管理

商家可以在这个子模块中选择是否开启前台店铺的预约功能，以及上传一张预约图片。此外还可以查

看到用户预约列表，里面详细展示了用户的预约时间、预约人数、联系人姓名及电话等信息。

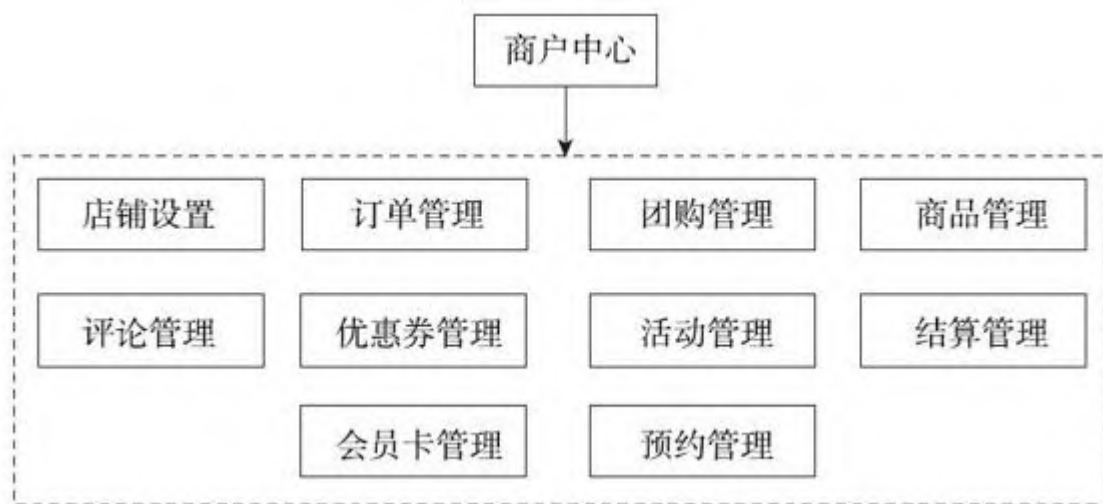


图8-2 商户中心的模块构成

这10个子模块加在一起就构成了一个完整的O2O商户管理后台模块，商户模块是商家与用户、平台之间直接进行互动的地方，承载了商家在O2O平台的所有日常管理工作，是一个非常重要的组成部分。此外O2O平台还为每个商户建立了一个精美的店铺主页，上面会展示商户形象图片、基本信息、商品信息、团购信息、用户评价等内容。

8.3.2 设计思路

1. 店铺相关数据表设计思路

店铺表结构设计主要考虑店铺检索的便捷性。由于店铺分类只能设置到二级，所以店铺表设计了两个字段用来储存各个级别的店铺分类来索引，这样在网站前台，用户就可以根据不同级别的店铺分类索引直接搜索店铺了。

店铺分类表中除常规分类名称、父类ID等常规字段外，还应设计分类图片和分佣比例这两个字段，分别用来推荐到首页的分类展示和平台款项结算。

在O2O的网站中店铺往往是有店铺标签的，这个设计可以更好地为网站用户展示店铺的特点。店铺标签表需要单独在数据库中设计一个表，除去标签名称等

常规字段外，还应设计一个标签数量的字段，当用户对商户进行评价时就可以选择店铺的标签，这样通过标签数量就可以直观反映某个店铺标签被用户选择过的次数了。

2. 店铺地区属性设计思路

ShopNC 020产品设计店铺所属地区分为三个等级：城市、区县、商区，商家在后台进行店铺设置的时候必须选择店铺所属的这三个地区属性，以方便020网站的用户使用地区来检索商家信息。根据大多数团购网站的设计来看，三个地区等级足以应对用户的检索了。

3. 店铺主页元素设计思路

020网站中每个店铺均拥有一个独立的商户主页，在这个主页中展示了商家的基本信息、地理位置、店

铺标签、团购优惠信息、店铺商品、用户点评等，如图8-3所示。



图8-3 店铺主页

商户基本信息中需要展示的内容有商户的地址、电话、营业时间、店铺标签以及预约和收藏等功能按钮。优惠信息区会以文字形式显示店铺最新的团购和

优惠券信息并会展示已参与人数量。店铺介绍部分是纯文字的。商品展示区会显示店铺上传的商品图片，鼠标滑过会放大并显示出价格和简介。网友点评部分会包含较多的信息，是提供给访问商户主页的用户最主要参考的信息区域，它主要包括网友头像名称、评分、人均消费金额、评价内容、图片、停车信息、商家的解释说明，此外还包括送鲜花和收藏点评这两个小功能按钮。

整个商户主页的最下方是用户发表点评的地方，登录后这里会显示添加点评的表格，否则显示登录提示的按钮链接。

8.3.3 代码实现

1. 团购券验证的代码实现

团购验证的代码部分需要完成几个重要的事情：更新订单表状态、更新团购密码表状态、验证成功后需要向用户发送短信并给出页面提示信息，如代码清单8-1所示。

代码清单8-1 团购券验证

```
if(isset($_POST) && !empty($_POST)){
    $model = Model();
    $order_pwd = $model->table('order_pwd')->where(array(
        'order_pwd'=>$_POST['order_passwd']))->find();
    //密码错误
    if(empty($order_pwd)){
        $this->showTip(L('nc_member_store_verify_password_is_wrong'),'','html',
        'error');
    }
    //状态无效
    if($order_pwd['state'] != 1){
        $this->showTip(L('nc_member_store_verify_password_is_use'),'','html',
        'error');
    }
    $use_time = time();
    //更新订单表状态
    $result_a = $model->table('order_pwd')->where(array(
        'order_group_id'=>$order_pwd['order_group_id']))->
    >update(array('state'=>2,'use_time'=>$use_time));
    //更新团购密码表状态
    $result_b = $model->table('order')->where(array(
        'order_id'=>$order_pwd['order_id']))->update(array('use_time'=>$use_time));
    if($result_a && $result_b){
        $orderinfo = $model->table('order')->where(array(
        'order_id'=>$order_pwd['order_id']))->find();
        //消费成功，发送短信
        $post_data = array(
            'phone' => $orderinfo['mobile'],
            'text' => '您的团购密码：'. $order_pwd['order_pwd'].'已经使用'
        );
        //发送短消息通知用户
        $ch = curl_init();
        curl_setopt($ch,CURLOPT_URL,BASE_SITE_URL."/api/message/demo.php");
        curl_setopt($ch,CURLOPT_HEADER, 0);
        curl_setopt($ch,CURLOPT_RETURNTRANSFER, 1);
        curl_setopt($ch,CURLOPT_POST, 1);
        curl_setopt($ch,CURLOPT_POSTFIELDS, $post_data);
```

```

        $output = curl_exec($ch);
        curl_close($ch);
        //修改状态
        $count = $model->table('order_pwd')->where(array(
            'order_id'=>$orderinfo['order_id'], 'state'=>2))->count();
        if($count == $orderinfo['number']){
            $model->table('order')->where(array('order_id'=>$orderinfo['order_id']))->
            update(array('state'=>3));
        }
        //验证成功，此处需要单独跳转到一个非跳转的提示页面
        $r_url = '<a href="index.php?act=storeorder&op=verify"
            >点击返回</a>';
        Tpl::output('msg', L('nc_member_store_verify_succ')).'<br>团购名称: '
        . $orderinfo['item_name']. ' <br>价格: '.intval($orderinfo['price']).'元<br>'. $r_url);
        Tpl::output('msg_type', 'succ');
        Tpl::showpage('msg', 'null_layout');die;
    }else{
        //验证失败
        $this->showTip(L('nc_member_store_verify_fail'), 'index.php?
        act=storeorder', 'html', 'error');
    }
}

```

此段代码中涉及发送短信的部分使用了cURL函数库，cURL是一个利用URL语法规则来传输文件和数据的工具，支持很多协议，如HTTP、FTP、TELNET等。该库常用函数如下：

- curl_init()：初始化。
- curl_setopt()；设置选项。
- curl_exec()；执行并获取HTML文档内容。
- curl_close()；释放curl句柄。

2. 店铺基本设置的代码实现

店铺的基本设置会涉及很多店铺信息的修改，比如店铺名称、店铺分类、地址分类、店铺标签，以及各种图片如店铺LOGO和店铺展示图片等等。

当商家选择店铺分类时应该采用Ajax技术即时显示出该分类的分佣比例供商家参考。店铺设置页面检测商家分类的选择并请求服务器PHP脚本程序返回分类佣金比例数据，最后将该数据即时展示在该页面中，如代码清单8-2所示。

代码清单8-2 后端PHP脚本响应程序

```
public function ajax_get_sc_settleOp(){
    $s_class = intval($_GET['s_class']);
    if($s_class > 0){
        $model = Model();
        $class_info = $model->table('store_class')->where(
            array('class_id'=>$s_class))->find();
        if($class_info['class_settle'] > 0){
            echo json_encode(array('done'=>true,'msg'=>'
分佣比例: '.$class_info['class_settle'].'%'));die;
        }else{
            $pclass_info = $model->table('store_class')->where(
                array('class_id'=>$class_info['parent_class_id']))->find();
            if($pclass_info['class_settle'] > 0){
                echo json_encode(array('done'=>
true,'msg'=>'分佣比例: '.$pclass_info['class_settle'].'%'));die;
            }else{
                echo json_encode(array('done'=>false));die;
            }
        }
    }else{
        echo json_encode(array('done'=>false));die;
    }
}
```

通过数据库查找分类的佣金比例并放在一个数组中通过json的形式返回。

前台页面的ajax js调用代码使用jquery的ajax调用方法，如代码清单8-3所示。

代码清单8-3 前台ajax方法

```

$("select[name=s_class]").change(function(){
    var s_class = $(this).val();
    $.getJSON('index.php?
act=storesetting&op=ajax_get_sc_settle&s_class='+s_class,function(result){
    if(result.done){
        $('#settle_show').html(result.msg);
    }
});
});

```

当每次更新店铺信息时，会更新所选地区商家数量，如代码清单8-4所示。

代码清单8-4 选择商家数量

```

//更新地区，分类商铺数
$info = $this->getStoreInfo($_POST['store_id']);
$model = Model();
$area_model = Model('area');
$city = $area_model->getList(array(
'parent_area_id'=>'0'));
if(!empty($city)){
    foreach($city as $city){
        $area = $area_model->getList(array('
parent_area_id'=>$city['area_id']));
        if(!empty($area)){
            foreach($area as $area){
                $small = $area_model->getList(array('
parent_area_id'=>$area['area_id']));
                if(!empty($small)){
                    foreach($small as $small){
                        $small_num = $model->table('store')->where(array('
mall_id'=>$small['area_id'],'is_audit'=>2))->count();
                        //1.待审核 2.审核通过 3.审核未通过
                        $area_model->modify(array('number'=>$small_num),array('area_id'=>$small['area_id']));
                    }
                }
                $area_num = $model->table('store')->
>where(array('area_id'=>$area['area_id'],'is_audit'=>2))->count();
                //1.待审核 2.审核通过 3.审核未通过
                $area_model->
>modify(array('number'=>$area_num),array('area_id'=>$area['area_id']));
            }
        }
        $city_num = $model->table('store')->where(array('
city_id'=>$city['city_id']))->count();
        $area_model->modify(array('number'=>$city_num),array('area_id'=>$city['area_id']));
    }
}
$city_path = BASE_DATA_PATH.'/cache/city/area_'.$info['city_id'].'.php';
if(is_file($city_path)){
    unlink($city_path);
}
$class_path = BASE_DATA_PATH.'/cache/class/class_'.$info['city_id'].'.php';

```



```
if(is_file($class_path)){
    unlink($class_path);
}
$this->showTip(L('nc_member_store_op_succ'),'','succ');
```

这段代码除了更新数据库中地区店铺数量外，还在最后更新了地区缓存文件，方便调用。

3. 店铺分类的代码实现

如代码清单8-5所示，主要实现了店铺分类的增加、修改和删除等操作。

代码清单8-5 店铺增加、修改、删除

```
//添加一个店铺分类
public function addOp(){
    $store_model = Model('store_class');
    $condition = array();
    $condition['parent_class_id'] = 0;
    $class_list = $store_model->getList($condition);
    //新增分类
    if(isset($_POST) && !empty($_POST)){
        //数据验证
        $obj_validate = new Validate();
        $validate_array = array(
            array('input'=>$_POST['class_name'], 'require'=>'true', 'validator'=>'Length', 'min'=>
                "1", 'max'=>"10", 'message'=>Language::get('nc_store_class_name_error')),
            array('input'=>$_POST['class_sort'], 'require'=>'true', 'validator'=>'Range',
                'min'=>0, 'max'=>255, 'message'=>Language::get('nc_store_class_sort_error'))
        );
        $obj_validate->validateparam = $validate_array;
        $error = $obj_validate->validate();
        if ($error != ''){
            $this->showTip(Language::get('error').$error, '', '', 'error');
        }
        $param = array();
        $param['class_name'] = trim($_POST['class_name']);
        if(isset($_POST['parent_class_id']) && intval($_POST['parent_class_id']) >
0){
            $param['parent_class_id'] = $_POST['parent_class_id'];
        }else{
            $param['parent_class_id'] = 0;
        }
        $param['class_sort'] = intval($_POST['class_sort']);
        $param['class_recommend'] = intval($_POST['is_recommend']);
        if(!empty($_FILES['class_image']['name'])) {
```

```

        $upload          = new UploadFile();
        $uploadaddir = ATTACH_CLASS_PATH;
        $upload->set('default_dir',$uploadaddir);
        $result = $upload->upfile('class_image');
        if(!$result) {
            $this->showTip($upload->error);
        }
        $param['class_image'] = $upload->file_name;
    }

    $result          = $store_model->save($param);
    if($result){
        $this->showTip(Language::get('nc_store_class_add_succ'),'index.php?
act=store&op= storeclass');
    }else{
        $this->showTip(Language::get('nc_store_class_add_fail'));
    }
    }
    $parent_class_id = intval($_GET['parent_class_id']);
    if(!empty($parent_class_id)) {
        Tpl::output('parent_class_id',$parent_class_id);
    }
    Tpl::output('op','add');          Tpl::output('list',$class_list);
    Tpl::showpage('storeclass.add');
}
//店铺分类删除
public function dropOp() {
    $class_id = trim($_POST['class_id']);
    $store_model = Model('store_class');
    $condition = array();
    $condition['parent_class_id'] = array('in',$class_id);
    $class_list = $store_model->getList($condition,
    '', '', 'class_id');
    if(!empty($class_list) && is_array($class_list)) {
        foreach($class_list as $val) {
            $class_id .= ','. $val['class_id'];
        }
    }
    $class_id = rtrim($class_id, ',');
    $condition = array();
    $condition['class_id'] = array('in',$class_id);
    //删除分类
    $result = $store_model->drop($condition);
    if($result) {
        $this->showTip(Language::get('nc_store_class_detele_succ'),'');
    }
}
else {
    $this->showTip(Language::get('
nc_store_class_detele_fail'),'','error');
}
}

```

店铺分类的添加和删除操作代码类似，故只列出了添加操作的代码，分类名称和分类的排序字段是必须要添加的，此外还应上传一个店铺分类图标，这个在前台分类展示时会用到。在删除分类的代码部分，注意到这

一段程序既可以完成单独一个店铺分类的删除也可以实现多个店铺分类的同时删除操作。

4. 店铺首页的代码实现

店铺首页是商户信息在前台的集中展现，因此会调用多方面的店铺相关信息，如店铺基本信息、店铺标签信息、团购优惠券信息、店铺商品信息、店铺点评信息、附近店铺列表等。

代码清单8-6 店铺展示代码

```
//店铺首页展示
public function detailOp(){
    Language::read('appointment');
    $store_id = intval($_GET['id']);
    $store_model = Model('store');
    $storeinfo = $store_model->getOne(array('
store_id'=>$store_id));
    Tpl::output('storeinfo',$storeinfo);
    $class = F('class_'. $storeinfo['city_id'],
null,'cache/class');
    if(!empty($class['class_root'])){
        foreach ($class['class_root'] as $val){
            if($val['class_id'] == $storeinfo['class_id']){
                Tpl::output('class_name',$val['class_name']);
            }
        }
        if(!empty($class['class_menu'][$storeinfo['class_id']])){
            foreach ($class['class_menu'][$storeinfo['class_id']] as $val){
                if($val['class_id'] == $storeinfo['s_class_id']){
                    Tpl::output('sub_class_name',$val['class_name']);
                }
            }
        }
        $city = F('area_'. $storeinfo['city_id'],null,'cache/city');
        if(!empty($city)){
            foreach ($city as $val){
                if($val['area_id'] == $storeinfo['area_id']){
                    Tpl::output('area_name',$val['area_name']);
                }
            }
        }
    }
    //点评列表
    $model_comment = Model('comment');
    $condition = array(
        'comment.store_id' => $store_id
    );
    $comment_list = $model_comment->getMemberList(
    $condition,10,'comment_id desc');
```

```

    Tpl::output('comment_list',$comment_list);
    Tpl::output('show_page',$model_comment->showpage(2));
    $model = Model();
    $member_info = $model->table('member')->where(array('
        member_id'=>$_SESSION['member_id']))->find();
    Tpl::output('member_info',$member_info);
    //增加访问量
    $model->table('store')->where(array('store_id'=>$store_id))-
>setInc('store_click',1);
    //调用店铺团购信息
    $groupbuy_list = $model->table('groupbuy')-
>where(array('store_id'=>$store_id))->select();
    Tpl::output('groupbuy_list',$groupbuy_list);
    //调用店铺优惠券信息
    $coupon_list = $model->table('coupon')->where(array(
        'store_id'=>$store_id))->order('download_count desc')->select();
    Tpl::output('coupon_list',$coupon_list);
    //调用标签
    $label_list = $model->table('store_label')->where(array(
        'store_id'=>$store_id))->order('label_num desc')->select();
    Tpl::output('label_list',$label_list);
    //计算评价分数
    $all_comment = $model->field('amount_score')->table('
comment')->where(array('store_id'=>$store_id))->select();
    $final_score = 0;
    if(count($all_comment) > 0){
    if(!empty($all_comment)){
        foreach ($all_comment as $val){
            $final_score += $val['amount_score'];
        }
        $final_score = round($final_score/count($all_comment));
    }

    Tpl::output('final_score',$final_score);
    Tpl::output('comment_num',count($all_comment));
    //调用店铺商品
    $goods_list = $model->table('goods')->where(array(
        'store_id'=>$store_id))->order('add_time desc')->select();
    Tpl::output('goods_list',$goods_list);
    //附近的商铺
    $sidestore = $store_model->getList(array('mall_id'
=>$storeinfo['mall_id'],'store_state'=>2,'store_id'=>array('neq',$store_id)));
    Tpl::output('sidestore',$sidestore);
    //收藏总数
    $count = $model->field('count(*) as sum')->table('favorites')-
>where(array ('fav_id'=>$store_id,'fav_type'=>'store'))->find();
    Tpl::output('fav_num', $count['sum']);
    Tpl::showpage('store.detail');
}

```

8.3.4 功能体验

本节将对商户模块中的活动管理、会员卡管理、预约管理的实际操作做简要说明。

1. 活动管理

店铺活动是商户吸引人气的主要促销手段之一，做好活动营销经常可以起到事半功倍的效果，如图8-4所示。

在商户活动的后台管理页面上会有“新增活动”、“编辑”、“删除”、“查看申请”等几个主要功能按钮，此外还应有一个活动搜索框，如图8-5所示。



水游城[鱼酷活鱼现烤优惠券]吃一条送一斤鱼券

鱼酷是专门针对情侣闺蜜等时尚年轻人的现代化的餐饮品牌，创立有8年，打造了以上海为中心辐射全国天津、济南、西安、沈阳、大连、郑州、武汉等核心城市的受年轻人喜欢的时尚餐饮品牌，鱼酷，目前全国近100家门店，成为深受时尚年轻人喜欢的高端洋气上档次的餐饮连锁品牌。

吃一条送一斤鱼券

荔枝香辣

更熟悉 更经典

小酸甜

新香辣2代 全面升级



鲜香扑鼻 辣过瘾 香过瘾 更多鲜鱼现烤 满足好基友 闺蜜聚会 吃货聚会 吃一条送一斤鱼券 活动有效期 2015-01-01至2015-12-31 水游城鱼酷全国100家门店 仅限鱼酷门店消费 每人限领一张 鱼酷优惠券 不可叠加 鱼酷优惠券 仅限鱼酷门店消费 详情请咨询门店

已经0人申请参加活动

有效期: 2015-12-31

报名截止时间: 2015-01-01

参加活动

最新评论

★★★★★ 2014-07-18

hosa dahdsadhaolad

★★★★★ 2014-07-13

拒收还不错，我要去感受下

★★★★★ 2014-03-19

送的荔枝香辣 不到3斤的黑鱼~味道不错 脆脆口 辣度可以接受 完全不辣~点了新出的奥利奥沙拉和一个脆黄瓜和海藻的凉菜 味道一般 还是老醋花生有味道~很泪点的花枝担担+泡饼~煮了两次水冻猪汤汁，三个人吃的好饱~

★★★★★ 2014-03-19

很不错的店 开了很多家 总体都差不多吧 泡饼强推!!! 鱼豆腐和金针菇煮过后也不错 我吃的是完全不辣的清江鱼 也很好吃啊 同去的小伙伴吃的是中等辣度的 他们说比不辣的好吃。。。就是越吃越热 店也不算大 服务还不错

★★★★★ 2014-03-19

很不错的店 开了很多家 总体都差不多吧 泡饼强推!!! 鱼豆腐和金针菇煮过后也不错 我吃的是完全不辣的清江鱼 也很好吃啊 同去的小伙伴吃的是中等辣度的 他们说比不辣的好吃。。。就是越吃越热 店也不算大 服务还不错

图8-4 活动前台页面

商户中心 > 活动管理				
活动管理				
活动名称:		查询		
活动名称	活动时间	申请数量	活动描述	操作
3店通用晚餐43元! 原价88元的阿鲁米双人套餐。米饭鲜滑爽	2014-03-05-2015-12-30	3	阿鲁米是一种源自云南的特色美食。高汤由大骨、老母鸡、火腿经长时间熬煮而成，具有浓郁鲜香味，一层鸡油盖住汤中的热气。米饭鲜滑爽口，配上各色佐料，红白黄绿相交辉映，别具风味。店家从2014年3月5日开始接待本地生活用户。有效期至2015年12月30日，无预约，请您合理安排用餐时间。每份骨肉相连含2串哦~	编辑 删除 查看申请
新增活动				

图8-5 活动后台管理页面

2. 会员卡管理

商户模块的会员卡管理由2个部分组成，第一个部分是会员卡的设置界面，如图8-6所示，包含了是否开启会员卡功能、设置会员卡的折扣、会员卡描述以及上传一个会员卡的图片；第二个部分是店铺会员的管理，如图8-7所示，包含了已申请会员的列表，在这个列表中可以开启和关闭某一个会员的会员资格以及查看会员的消费记录。

消费记录通过商家后台进行录入，包括了会员卡号、消费金额、消费人数等。

会员卡 (*必填填写项)

请填写会员卡信息

* 开启会员卡:

☒ 开启 ☐ 停止

会员卡折扣:

折

会员卡描述:

阿香米线会员卡全店通用, 不限时间段随时使用。口口美味, 浓香回味! 尊享会员优惠

会员卡图片:

选择文件

未选择文件

已上传:



保存

图8-6 会员卡设置页面

商户中心 > 会员卡					
会员卡					
会员名称	开自	卡号	消费总额	消费记录	操作
rain8boy	否		0.00	查看	激活
小黄屯货	否		0.00	查看	激活
shopncyan	否		0.00	查看	激活
测试账户	否		0.00	查看	激活
tzp	是	112585	0.00	查看	关闭
xcz0106	是	1121245	0.00	查看	关闭
meilin116	是	1126700	0.00	查看	关闭
qgm9qr	否		0.00	查看	激活
lizimu123	是	1130134	0.00	查看	关闭
alen	是	1131761	0.00	查看	关闭
kkz085	是	1133550	0.00	查看	关闭
Jayden	是	1135431	0.00	查看	关闭
ib66	是	1141904	0.00	查看	关闭
qqqqqq	是	1140928	0.00	查看	关闭
longjun	是	1147868	0.00	查看	关闭

图8-7 会员列表界面

商户中心 > 商户设置	
输入消费记录	
卡号:	
价格:	
消费人数:	
保存	

图8-8 消费记录录入界面

3. 预约管理

商户的预约管理包括2个部分，第一个部分是预约设置管理，如图8-9所示，包含设置是否开启预约功能以及上传一张预约图片；第二个部分是查看预约列表，如图8-10所示，包含了预约时间、预约人数、联系人以及联系电话等信息，并支持按时间的检索功能。



商户中心 > 商户设置

预约 (*必填填写项)

* 预约开启: ☒ 开启 ☐ 停止

预约图片: 未选择文件

已上传:

美团

保存

图8-9 预约设置页面

商户中心 > 商户设置				
预约列表				
		从		到 查询
店铺名称	预约时间	预约人数	联系人	预约电话
阿香米线(水游城店)	2014-04-24	1	55888	18896666666
阿香米线(水游城店)	2014-04-23	1	吴经理	15044772010
阿香米线(水游城店)	2014-04-22	1	fang	15000000070
阿香米线(水游城店)	2014-04-20	1	123123	15841833667
阿香米线(水游城店)	2014-04-20	1	88811122	13559220213
阿香米线(水游城店)	2014-04-19	9	张	13897898541
阿香米线(水游城店)	2014-04-17	2	小张	15800506660
阿香米线(水游城店)	2014-04-14	1	135	13305431759
阿香米线(水游城店)	2014-04-13	1	1	18601059811
阿香米线(水游城店)	2014-04-13	1	qwtsedf	13884561234
				首页 上一页 1 2 下一页 末页

图8-10 预约列表页面

8.4 团购模块设计

8.4.1 模块构成

团购是O2O平台商家进行促销的最主要的手段，商家可以通过后台发布团购活动，平台进行审核，审核成功之后，会员可以进行团购，团购成功，平台会将团购券以短信的形式发送给会员，会员根据线上商家地址，携带团购券到实体店铺去消费，消费成功之后，该团购券状态变成已消费状态，此时会员可以对店铺进行评价。在O2O开发中，这部分主要分为4个模块：商户团购管理、平台团购管理、会员订单管理、团购前台展示。

商家可以发布和编辑团购，团购开始之前可以编辑和删除，团购开始之后不可以编辑和删除，可以按

照团购名称、团购状态进行索引。

（1）平台团购管理

平台团购管理分为两个模块，分别是：团购管理和团购订单。团购管理可以查看团购列表，可以对商家发布的团购进行审核，可以查看该团购下生成的团购券，可以选择该团购停用，可以按照团购状态和团购审核情况进行搜索。可以查看订单情况，可以按照订单编号、会员名称、商铺名称、团购名称进行查询。

（2）会员订单管理

会员订单管理，会员可以查看团购订单，可以查看订单详情，也可以再次发送短信，将团购券发送给会员手机号。

（3）团购前台展示

团购前台展示有团购列表模块、团购详情模块、选择团购数量模块、团购支付模块。团购列表模块，可以按照分类和商圈进行筛选，可以按照销售量、有效期、价格进行排序。团购详情模块，显示团购价格、团购次数、折扣、团购介绍等信息。选择团购数量模块，可以按照团购项目选择数量。团购支付模块，显示订单号，有支付宝和预存款两种支付方式，点击“提交”按钮进行支付。

8.4.2 设计思路

1. 团购数据表设计思路

团购数据表一共包括3张表，一张是团购信息数据表，一张是团购订单表，还有一张团购券表。

团购信息数据表，包括：团购名称、团购介绍、团购开始时间、团购结束时间、所属店铺ID、店铺名称、原始价格、团购价格、团购数量、购买上线、购买人数、团购介绍、团购图片、发布时间等字段信息。

团购订单数据表，包括：订单编号、会员ID、会员名称、手机号、商铺ID、商铺名称、下单时间、团购索引ID、团购名称、团购数量、团购价格、团购状态等字段信息。

团购券数据表，包括：索引订单ID、团购券状态、团购码、使用时间。

2. 团购模块流程设计思路

团购模块流程设计思路，商家进入团购管理，选择发布团购，填写团购名称、开始时间、结束时间、原价、团购价格、团购数量、购买上线、团购图片、团购说明、团购介绍，发布成功之后，提交到平台进行审核，审核成功，会员可以参加团购活动，会员浏览团购项目，选择自己喜欢的团购项目，可以进行购买，购买单个团购项目之后，可以选择数量，点击“提交”按钮以后，进入生成订单，选择支付方式，点击“支付”按钮，即可完成整个团购活动，消费成功之后可以对店铺进行评价。如图8-11所示。

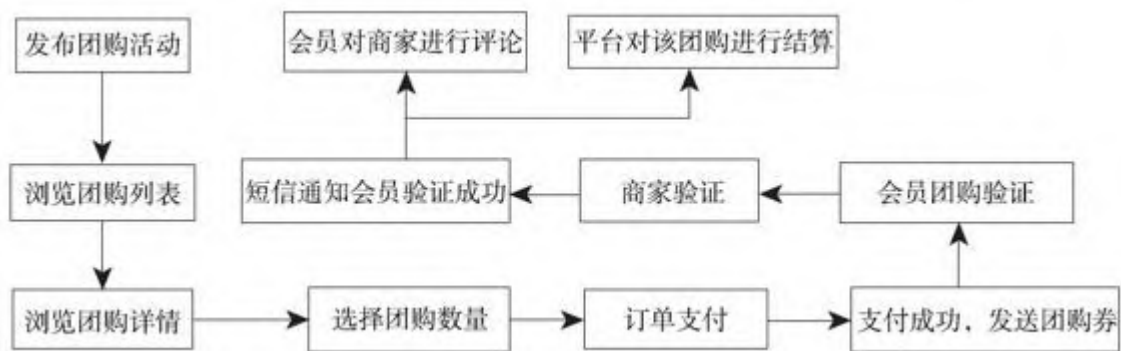


图8-11 团购模块流程设计思路

8.4.3 代码实现

1. 发布团购代码展示

发布团购部分需要实现的有：显示团购模板、上传图片、发布团购信息，如代码清单8-7所示。

代码清单8-7 发布团购

```
/*
 * 添加团购活动
 */
public function addOp(){
    if(isset($_POST) && !empty($_POST)){
        $uploadArr = $this->upload_pic('groupbuy_pic');//上传团购图片
        if($uploadArr['state'] == false){
            $this->showTip($uploadArr['name']);
        }
        $storeinfo = $this->getStoreInfo($_SESSION['store_id']); //获取店铺信息
        $params = array();
        $params['group_name'] = trim($_POST['group_name']);
        $params['group_help'] = trim($_POST['group_help']);
        $params['start_time'] = strtotime(trim($_POST['start_time']));
        $params['end_time'] = strtotime(trim($_POST['end_time']));
        $params['store_id'] = $storeinfo['store_id'];
        $params['store_name'] = $storeinfo['store_name'];
    }
}
```

```

        $params['original_price']          =
trim($_POST['original_price']);
        $params['group_price']              =
trim($_POST['group_price']);
        $params['buyer_count']              =
trim($_POST['buyer_count']);
        $params['buyer_limit']              =
trim($_POST['buyer_limit']);
        $params['group_intro']              =
trim($_POST['group_intro']);
        $params['group_pic']                =
$uploadArr['name'];
        $params['publish_time']              = time();
        $params['city_id']                  = $this-
>store['city_id'];
        $params['area_id']                  = $this-
>store['area_id'];
        $params['mall_id']                  = $this-
>store['mall_id'];
        $params['class_id']                  = $this-
>store['class_id'];
        $params['s_class_id']                = $this-
>store['s_class_id'];
        $group_model = Model('groupbuy');
        $result = $group_model->save($params);//保存团购
        if($result){//新增团购
            $this-
>showTip(L('nc_member_store_add_groupbuy_succ'),'index.php?
act=storegro-upbuy&op=list','succ');
        }else{
            $this-
>showTip(L('nc_member_store_add_groupbuy_fail'),'','error');
        }
        Tpl::showpage('groupbuy.add');
    }

```

2. 团购列表代码展示

团购列表部分需要实现的有：显示团购列表、团购排序、团购分类、团购区域、热门团购等信息，如代码清单8-8所示。

代码清单8-8 团购列表

```
/*
 * 团购列表
 */
public function listOp(){
    $groupbuy_model = Model('groupbuy');
    $condition = array();
    $condition['start_time'] =
array('elt',time());
    $condition['end_time'] = array('egt',time());
    $condition['is_open'] = 1;//1.开启 2.关闭
    $condition['city_id'] = $this->
city_info['area_id'];
    $condition['is_audit'] = 2;//1.待审核 2.审核通过 3.
审核未通过
    //城市分类
    $condition['city_id'] = $this->
city_info['area_id'];
    if(isset($_GET['area_id']) && !empty($_GET['area_id']))
    {
        //区域
        $condition['area_id'] =
intval($_GET['area_id']);
        Tpl::output('area_id',intval($_GET['area_id']));
        if(isset($_GET['mall_id']) &&
!empty($_GET['mall_id'])) {
            //商区
            $condition['mall_id'] =
intval($_GET['mall_id']);
            Tpl::output('mall_id',intval($_GET['mall_id']));
        }
    }
    //商铺分类
```

```

        if(isset($_GET['class_id']) &&
!empty($_GET['class_id'])) {
            $condition['class_id'] =
intval($_GET['class_id']);

Tpl::output('class_id',intval($_GET['class_id']));
            if(isset($_GET['class_id_1']) &&
!empty($_GET['class_id_1'])) {
                $condition['s_class_id'] =
intval($_GET['class_id_1']);

Tpl::output('class_id_1',intval($_GET['class_id_1']));
            }
        }
        $list = $groupbuy_model-
>getList($condition,'20','buyer_count desc');
        Tpl::output('list',$list);
        Tpl::output('show_page',$groupbuy_model-
>showpage());
        //分类
        $this->classlist();
        //区域
        $this->arealist();
        //热门团购
        $this->hotgroupbuy();
        Tpl::showpage('groupbuy');
    }

```

3. 团购详情代码展示

团购详情部分需要实现的有：显示团购信息、团购验证、热门团购，如代码清单8-9所示。

代码清单8-9 团购详情

```

/*
 * 团购详情

```

```

*/
public function detailOp(){
    //团购详细
    $group_id      = intval($_GET['group_id']);
    $group_model= Model('groupbuy');
    $group = $group_model-
>getOne(array('group_id'=>$group_id,'is_open'=>1,'is_audit'=>2)
);//is_audit=1.待审核 2.审核通过 3.审核未通过,is_open=1.开启 2.关闭
    //验证团购是否存在
    if(empty($group)){
        $this-
>showTip(L('nc_groupbuy_is_not_exists'),'','error');
    }
    //验证团购是否开始
    if($group['start_time']>time()){
        $this-
>showTip(L('nc_groupbuy_is_not_start'),'','error');
    }
    //验证团购是否结束
    if($group['end_time']<time()){
        $this-
>showTip(L('nc_groupbuy_is_already_end'),'','error');
    }
    Tpl::output('group',$group);
    $storeinfo = $this->getStoreInfo($group['store_id']);
    Tpl::output('storeinfo',$storeinfo);
    //热门团购
    $this->hotgroupbuy();
    Tpl::showpage('groupbuy.detail');
}

```

4. 团购订单代码展示

团购订单部分需要实现的有：显示团购信息、验证数据、生成订单，如代码清单8-10所示。

代码清单8-10 团购订单

```

/*
 * 团购订单
 */
public function groupbuyorderOp(){
    if(isset($_POST) && !empty($_POST)){
        //表单验证
        $obj_validate = new Validate();
        $obj_validate->validateparam          =          array(

array("input"=>intval($_POST['q_number']),"require"=>"true","va
lidator"=>"number","message"=>L('nc_groupbuy_is_not_null_and_nu
mber'))

    );
        $error = $obj_validate->validate();
        if ($error != ''){
            $this->
>showTip(Language::get('error').$error, '', 'error');
        }
        $group_id          = intval($_POST['group_id']);
        $group_model        = Model('groupbuy');
        $group = $group_model->
>getOne(array('group_id'=>$group_id));
        //检测是否登录
        if(!isset($_SESSION['member_id'])){
            $this->
>showTip(L('nc_groupbuy_is_not_login'),'index.php?
act=login','error');
        }
        //验证团购是否存在
        if(empty($group)){
            $this->
>showTip(L('nc_groupbuy_is_not_exists'),'','error');
        }
        //验证团购是否开始
        if($group['start_time']>time()){
            $this->
>showTip(L('nc_groupbuy_is_not_start'),'','error');
        }
        //验证团购是否结束
        if($group['end_time']<time()){
            $this->
>showTip(L('nc_groupbuy_is_already_end'),'','error');
        }
        //验证团购数量
        if(intval($_POST['q_number'])>$group['buyer_limit'){
            $this->
>showTip(L('nc_groupbuy_quantity_is_not_limit'),'','error');
        }
    }
}

```

```

    }
    $store_id      =      $group['store_id'];
    $storeinfo     =      $this->
>getStoreInfo($store_id);
    $member_model = Model('member');
    $member = $member_model->
>getOne(array('member_id'=>$_SESSION['member_id']));
    //生成团购订单
    $params
    array();
    $params['order_sn']      = $this->snOrder();
    $params['member_id']     = $member['member_id'];
    $params['member_name']   = $member['member_name'];
    $params['mobile']        = $member['mobile'];
    $params['store_id']      =
    $storeinfo['store_id'];
    $params['store_name']    =
    $storeinfo['store_name'];
    $params['add_time']      = time();
    $params['order_type']    = 1;//未支付状态
    $params['item_id']       =
    intval($_POST['group_id']);
    $params['item_name']     = $group['group_name'];
    $params['number']        =
    intval($_POST['q_number']);
    $params['price']         =
    intval($_POST['q_number'])*$group['group_price'];
    $params['state']         = 1;
    $params['order_out']     = $this->order_sn;
    $order_model = Model('order');
    $result = $order_model->save($params);
    if($result){
        //修改团购购买数量
        $model = Model();
        $model->table('groupbuy')->
>where(array('group_id'=>$group_id))->
setInc('buyer_num',intval($_POST['q_number']));
        header("Location:index.php?
act=groupbuy&op=grouppayment&order_id=".$result);
        exit;
    }else{
        $this->
>showTip(L('nc_groupbuy_operation_is_wrong'),'','error');
    }
    }
    //团购信息
    $group_id      = intval($_GET['group_id']);

```



```

        $group_model= Model('groupbuy');
        $group = $group_model-
>getOne(array('group_id'=>$group_id));
        if(empty($group)){
            $this-
>showTip(L('nc_home_groupbuy_is_not_exists'),'index.php','error
');
        }

        Tpl::output('group',$group);
        Tpl::showpage('groupbuy.order');
    }

```

5. 团购支付代码展示

团购支付部分需要实现的有：显示支付信息、订单信息、会员信息，如代码清单8-11所示。

代码清单8-11 团购支付

```

/*
 * 支付
 */
public function grouppaymentOp(){
    $order_id      =      intval($_GET['order_id']);
    $order_model = Model('order');
    $order = $order_model-
>getOne(array('order_id'=>$order_id));
    if(empty($order)){
        $this-
>showTip(L('nc_group_order_is_not_exists'),'','error');
    }
    //会员登录
    if($_SESSION['is_login'] != 1){
        $this-
>showTip(L('nc_group_member_is_not_login'),'','error');
    }
    if($order['member_id']!= $_SESSION['member_id']){

```

```
        $this->showTip('该记录不存在','', 'error');
    }
    //会员信息
    $member_id = $_SESSION['member_id'];
    $member_model = Model('member');
    $member = $member_model-
>getOne(array('member_id'=>$member_id));
    Tpl::output('member', $member);
    //支付方式
    $model = Model();
    $payment = $model->table('payment')-
>where(array('payment_state'=>1))-> select();
    Tpl::output('payment', $payment);
    Tpl::output('order', $order);
    Tpl::showpage('groupbuy.payment');
}
```

8.4.4 功能体验

1. 团购列表

团购列表页面从首页点击“团购”按钮进入，浏览团购列表，如图8-12所示。



图8-12 团购列表

2. 团购详情

团购列表选择团购进入团购详情，如图所8-13示。



图8-13 团购详情

3. 团购选择数量

团购详情点击“购买”按钮进入选择团购数量，如图所示。



图8-14 团购选择数量

4. 团购支付

选择团购数量点击“提交订单”按钮，会生成订单，会员支付购买，如图8-15所示。



图8-15 团购支付

8.5 优惠券模块设计

8.5.1 模块构成

优惠券是O2O平台商家进行促销的一种主要且有效的手段，网站用户可以将优惠券打印或下载到自己的手机上，之后到商户那里使用享受一定的优惠折扣。在O2O平台的开发中对于这个部分主要分为3个子模块：商户后台的优惠券管理、平台后台的优惠券管理和优惠券前台展示。

（1）商户后台的优惠券管理

商户可以在此子模块中添加和编辑优惠券，查看已添加优惠券的信息，可以删除优惠券，此外列表界面还具备一个检索功能。

（2）平台后台的优惠券管理

O2O平台管理人员可以在后台查看所有商品的优惠券信息，进行审核、删除和首页推荐等操作，此外还可以查看优惠券的下载记录。

（3）优惠券前台展示

该子模块主要用来在网站前台向用户展示优惠券的详情信息并提供打印和下载到用户手机的功能，同时页面上还展现了所属店铺的一些基本信息。如图8-16所示。

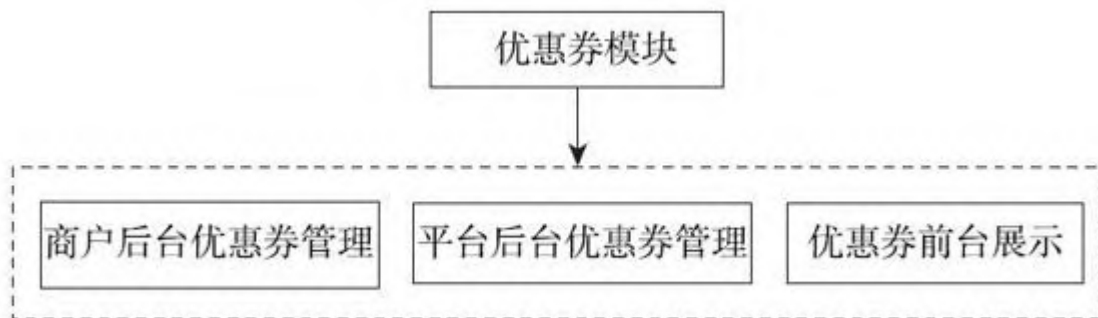


图8-16 优惠券模块的组成

8.5.2 设计思路

1. 优惠券数据表设计思路

优惠券模块一共需要建立2张数据表，一张是优惠券信息主表，另一张是优惠券下载记录表。优惠券信息主表中主要记录优惠券名称、图片、起止时间、短信内容等，下载记录表中主要记录下载优惠券名称、下载时间、下载方式等信息。其中优惠券ID要设为主键，方便连表时提高数据库的执行效率。

为了方便连表查询一些诸如会员、店铺等相关信息，这两个表中还要存储一下会员或店铺的ID，这也是为了提高数据库的执行效率。

2. 优惠券模块流程设计思路

020平台的优惠券模块的整体设计思路是商家先在后台发布新的优惠券信息，包括填写优惠券的名称、使用有效期、上传一张优惠券的图片以及编辑优惠券的详情和短信内容，当商家保存后则该新添优惠券进入了待审核状态，此时需要020网站管理员在平台后台进行优惠券内容审核，只有审核通过的优惠券才能在网站前台显示从而被用户看到，此时商家还可以对所添加的优惠券进行编辑和删除操作。

平台管理员在后台查看到该商家所新添加的这条优惠券信息后进行审核，此外还可以将某个优惠券推荐到网站首页上进行展示。如果审核不通过，则该优惠券无法在前台进行展示。

用户在网站前台选择自己喜欢的优惠券后，可以选择直接打印或者是下载短信到手机上这两种方式。直接打印优惠券会通过计算机相连的打印机一次性

将4张优惠券打印到一张A4纸上，用户自己裁剪后到店消费时出示即可。选择获取短信的用户则可以填写自己的手机号，然后点击“发送”按钮得到一个优惠券短信，之后可以凭此短信到店消费享受优惠。无论用户使用哪种方式进行优惠券下载，系统都会进行记录，O2O的平台管理员可以在平台后台看到这些记录。如图8-17所示。

3. 优惠券页面设计思路

网站前台优惠券的展示页面需要包括以下几个部分。

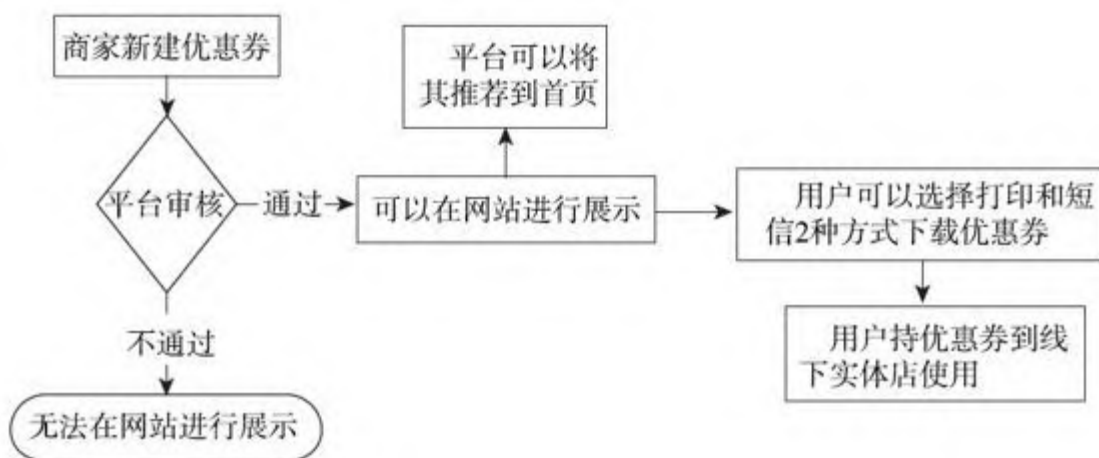


图8-17 优惠券模块流程图

(1) 商家基本信息

此部分的展示主要包括一个商家LOGO图片、商家名称、店铺评分、人均消费、店铺地址和联系电话等，放在了详情页面顶部的位置上。

(2) 优惠券基本信息

此部分的展示主要包括优惠券标题、优惠券图片、浏览和下载次数、截止日期的倒计时显示和打印、下载优惠券的按钮。

(3) 优惠券详情

此部分主要展示出了商家在后台多媒体编辑器中编辑的优惠详细图文。此外该页面还可以加上推荐优惠券展示区，可以放在右边侧栏上。

优惠券详情页面如图8-18所示。



图8-18 优惠券详情页面

8.5.3 代码实现

1. 优惠券增改的代码实现

如代码清单8-12所示，主要是实现了优惠券的增加、修改和删除等操作。

代码清单8-12 团购详情

```
//新增一个优惠券
public function add_couponOp(){
    if(isset($_POST) && !empty($_POST)){
        $uploadArr = $this->upload_pic('coupon_pic');
        if($uploadArr['state'] == false){
            $this->showTip($uploadArr['name']);
        }
        //表单验证
        $obj_validate = new Validate();
        $obj_validate->validateparam      = array(
array("input"=>trim($_POST['coupon_name']),"require"=>"true","message"=>Language
::get('nc_member_store_coupon_name_is_not_null')),

array("input"=>trim($_POST['coupon_start_time']),"require"=>"true","message"=>
Language::get('nc_member_store_coupon_start_is_not_null')),

array("input"=>trim($_POST['coupon_end_time']),"require"=>"true","message"=>
Language::get('nc_member_store_coupon_end_is_not_null'))
);
        $error = $obj_validate->validate();
        if ($error != ''){
            $this->showTip(Language::get('error').$error,'','html','error');
        }
        $storeArr = $this->getStoreInfo($_SESSION['store_id']);
        $params    = array();
        $params['coupon_name'] = trim($_POST['coupon_name']);
        $params['coupon_pic']  = $uploadArr['name'];
        $params['coupon_start_time'] =
strtotime(trim($_POST['coupon_start_time']));
        $params['coupon_end_time']    = strtotime(trim($_POST['coupon_end_time']));
        $params['coupon_des'] = trim($_POST['coupon_des']);
        $params['short_message'] = trim($_POST['short_message']);
        $params['store_id'] = $storeArr['store_id'];
        $params['store_name'] = $storeArr['store_name'];
        $params['city_id'] = $this->store['city_id'];
        $coupon_model      = Model('coupon');
        $result            = $coupon_model->save($params);
    }
}
```

```

        if($result){
            $this->showTip(Language::get('nc_member_store_coupon_name_save_succ'),'index.php?
act=storecoupon','succ');
        }else{
            $this->showTip(Language::get('nc_member_store_coupon_name_save_fail'),'','
html','error');
        }
    }
    Tpl::showpage('storecoupon.add');}
//删除优惠券
public function del_couponOp(){
    $model = Model();
    $coupon = $model->table('coupon')->where(array('
coupon_id'=>intval($_GET['coupon_id']))->find();
    //优惠券不存在
    if(empty($coupon)){
        $this->showTip(L('nc_member_store_coupon_is_not_exists'),'','html','error');
    }
    if($coupon['store_id'] != $_SESSION['store_id']){
        $this->showTip(L('nc_member_store_coupon_not_consistent'),'','html','error');
    }
    $result = $model->table('coupon')->where(array('
coupon_id'=>intval($_GET['coupon_id']))->delete();
    if($result){
        $this->showTip(L('nc_member_store_coupon_name_del_succ'),'index.php?
act=storecoupon','succ');
    }else{
        $this->showTip(L('nc_member_store_coupon_name_del_fail'),'','html','error');
    }
}
}

```

优惠券的添加和删除操作代码类似，故只列出了添加操作的代码，优惠券名称和起止时间字段是必须要添加的。删除优惠券的时候需要注意检查当前操作账号是否为该优惠券所属商户账号，如不是则需要给出一个警告。

2. 优惠券展示的代码实现

网站前台优惠券的展示需要后台服务器调用多个与该优惠券相关的信息，如优惠券基本信息、相关优惠券列表信息、优惠券所属店铺信息等。

代码清单8-13 优惠券

```
//优惠券展示
public function detailOp(){
    $coupon_id = intval($_GET['coupon_id']);
    //优惠券详情
    $coupon_info = $this->couponinfo($coupon_id); //优
    惠券有效期    if($coupon_info['coupon_end_time']<time()){
        $this->showTip('该优惠券已经过期','', 'html', 'error');
    }
    Tpl::output('coupon_info', $coupon_info);
    $model = Model();
    $model->table('coupon')->where(array('coupon_id'=>$coupon_id))-
>setInc('view_count',1);
    //相关优惠券
    $couponlist = $model->table('coupon')->where(array('
    city_id'=>$this->city_info['area_id']))->order('coupon_id desc')->limit(3)-
>select(); Tpl::output('couponlist', $couponlist);
    //优惠券所属店铺信息
    $field
    ="store.store_id,store.store_name,store.pic,store.address,store.telephone,store.b
    us,store.logo,store.person_consume,store.description";
    $on = "coupon.store_id=store.store_id";
    $model->table('coupon,store')->field($field);$store_info = $model-
>join('left') ->on($on)->where(array('coupon_id'=>$coupon_id))->find();
    Tpl::output('store_info', $store_info);
    //计算评价分数
    $all_comment = $model->field('amount_score')->table(
    'comment')->where(array('store_id'=>$coupon_info['store_id']))->select();
    $final_score = 0;
    if(count($all_comment) > 0){
    if(!empty($all_comment)){
        foreach ($all_comment as $val){
            $final_score += $val['amount_score'];
        }
    }
    $final_score = round($final_score/count($all_comment));
    Tpl::output('final_score', $final_score);
    Tpl::showpage('coupon.detail');
}
```

需要注意的是程序开始时要进行优惠券是否过期的判断，如果过期，则需要给出提示警告，另外由于优惠券详情页面上要显示一个平均评价分数，因此需要从数据库中调取评价记录并进行平均值计算（最终分数进行四舍五入计算）。

3. 优惠券下载的代码实现

优惠券的下载部分有打印和手机短信两种下载方式。

代码清单8-14 优惠券下载

```
//打印优惠券
public function printOp(){
    $coupon_id = intval($_GET['coupon_id']);
    //优惠券详情
    $coupon_info = $this->couponinfo($coupon_id);
    if(empty($coupon_info)){
        $this->showTip(L('nc_coupon_is_not_exists'),'','html','error');
    }
    Tpl::output('coupon_info',$coupon_info);
    $model = Model();
    $params = array(
        'member_id' => isset($_SESSION['member_id'])?
$_SESSION['member_id']:'',
        'member_name' => isset($_SESSION['member_name'])?
$_SESSION['member_name']:'',
        'coupon_id' => $coupon_id,
        'coupon_name' => $coupon_info['coupon_name'],
        'download_time' => time(),
        'download_type' => 1
    );
    $model->table('coupon_download')->insert($params,true);
    Tpl::showpage('coupon.print','null_layout');
}
//短信下载优惠券
public function sendmsgOp(){
    //验证手机号
    if(!$this->checkphone($_GET['phone'])){
        $arr = array(
            'result' => 'fail',
            'msg' =>Language::get('offline_phone_format_error')
        );
        echo json_encode($arr);
        exit;
    }
    //验证码
    if (!$this->checkSeccode($_GET['nchash'],$_GET['captcha'])){
        $arr = array(
            'result' => 'fail',
            'msg' =>Language::get('offline_captcha_format_error')
        );
        echo json_encode($arr);
        exit;
    }
    //优惠券详情
    $coupon_id = intval($_GET['coupon_id']);
    $coupon_info = $this->couponinfo($coupon_id);
    $post_data = array(
        'phone' => trim($_GET['phone']),
        'text' => $coupon_info['short_message']
    );
}
```


8.5.4 功能体验

1. 优惠券的商家管理

商家可以在商户中心查看到已添加优惠券的列表（支持搜索功能），可以做新增编辑和删除优惠券的操作。点击“优惠券名称”按钮可以直接跳转到优惠券详情页面。如图8-19和图8-20所示。

商户中心 > 优惠券管理					
优惠券管理					
优惠券名称:		搜索			
优惠券名称	商户名称	有效期	下载次数	浏览次数	操作
阿香米线20元代金券, 全场通用	阿香米线(水游城店)	2014-03-14 ~ 2020-03-05	32	368	编辑 删除
阿香米线10元代金券, 全场通用	阿香米线(水游城店)	2014-03-14 ~ 2020-03-11	0	58	编辑 删除
阿香米线双人套餐一份	阿香米线(水游城店)	2014-03-14 ~ 2020-03-12	3	100	编辑 删除
新增优惠券			首页 上一页 1 下一页 末页		

图8-19 商户中心的优惠券管理页面

效期、商铺名称、下载次数等信息。此外平台管理员还可以看到优惠券的下载记录表（支持按优惠券名称的搜索操作）。如图8-21和图8-22所示。

优惠券管理

优惠券名称

审核状态

操作提示

可以对优惠券进行编辑和审核操作

优惠券列表显示优惠券名称、有效期、商铺名称、下载次数等信息

可以设置是否将优惠券投放广告下方进行展示（最多展示3个）

列表

	优惠券名称	有效期	商铺名称	下载次数	浏览次数	数量	审核	置顶推荐	操作
☐	阿香米线20元代金券，全场通用	2014-03-14~2020-03-05	阿香米线(永源城店)	32	368	0	审核通过	是 (取消推荐)	审核 删除
☐	阿香米线10元代金券，全场通用	2014-03-14~2020-03-11	阿香米线(永源城店)	0	58	0	审核通过	是 (取消推荐)	审核 删除
☐	阿香米线双人套餐一份	2014-03-14~2020-03-12	阿香米线(永源城店)	3	100	0	审核通过	是 (取消推荐)	审核 删除
☐	大渔铁板烧30元代金券一张	2014-02-05~2020-12-30	大渔铁板烧	0	255	0	审核通过	否 (设为推荐)	审核 删除
☐	芭风拉面	2014-01-01~2014-05-31	芭风拉面	1	84	0	审核通过	否 (设为推荐)	审核 删除
☐	[V] house量贩式KTV	2014-03-01~2014-05-31	[V] house量贩式KTV	2	59	0	审核通过	否 (设为推荐)	审核 删除
☐	荔枝香（鱼胆烤鱼）两吃两送 30元代金券	2014-02-05~2020-12-30	廖希庭(鱼胆烤鱼火锅)	0	42	0	审核通过	否 (设为推荐)	审核 删除
☐	哈根达斯冰淇淋，任意一款 30元代金券	2014-02-05~2020-12-30	[X]哈根达斯 哈根达斯	2	55	0	审核通过	否 (设为推荐)	审核 删除
☐	Costa coffee 10元代金券	2014-02-15~2020-12-30	COSTA COFFEE咖啡	0	207	0	审核通过	否 (设为推荐)	审核 删除
☐	Vista鲜城多喜区30元代金券一张	2014-02-05~2020-12-30	Vista鲜城	2	223	0	审核通过	否 (设为推荐)	审核 删除
☐	全选	删除	审核通过	审核不通过	首页 上一页 1 下一页 末页				

图8-21 平台后台的优惠券管理页面

优惠券管理 新增			
优惠券名称 🔍			
🔍 操作提示 +			
+ 快速筛选会员名称、优惠券名称、下载时间等选项			
列表			
会员名称	优惠券名称	下载时间	下载方式
游客	阿香米线10元代金券，全场通用	2014-04-23	打印
游客	[V] house量贩式KTV	2014-04-23	短信
游客	[V] house量贩式KTV	2014-04-23	短信
游客	大润发储值20元代金券一张	2014-04-23	打印
游客	Costa coffee 10元代金券	2014-04-23	打印
123	阿香米线20元代金券，全场通用	2014-04-23	短信
游客	妈妈咪呀冰淇，任意一款 30元代金券	2014-04-23	打印
游客	[V] house量贩式KTV	2014-04-23	打印
游客	阿香米线20元代金券，全场通用	2014-04-23	打印
游客	阿香米线20元代金券，全场通用	2014-04-22	打印
小野马	Vesta咖啡东南区30元代金券一张	2014-04-22	打印

图8-22 平台后台的优惠券下载记录页面

3. 优惠券的下载操作

用户可以在优惠券的详情页面上进行优惠券的下载操作，优惠券的详情页面上会出现“立即打印”和“免费下载短信”这两个按钮，用户可以根据自己的需要自行选择下载方式。

1) 选择打印的方式，系统会显示出优惠券打印预览页面（4张优惠券打印在1张A4纸上），如图8-23所

示。



图8-23 优惠券打印预览页面

2) 选择短信下载的方式，系统会显示短信内容并要求用户填写自己的手机号码和验证码。如图8-24所示。

短信优惠券下载

×

您将收到如下短信:

阿香米线10元代金券, 本地生活优惠券#215816:凭此短信[阿香过桥米线]20元优惠券仅售13.9元, 限当次到店就餐客人获赠, 持券就餐客人不再赠券, 赠券限下次使用, 3-1至3-31有效。详询商家

请准确填写手机号码。

请输入手机号码:

请输入验证码:

42V6

看不清? 换一个

发送

图8-24 优惠券短信下载页面

8.6 结算模块设计

8.6.1 模块构成

结算管理在O2O网站平台的运营过程中具有极其重要的地位，它直接影响平台商家的收入，平台高效的结算管理能增强商家对O2O网站的信任感，使其可以与平台建立长期良好的合作关系。

如图8-25所示，O2O网站的结算管理包括两个主要的子模块。

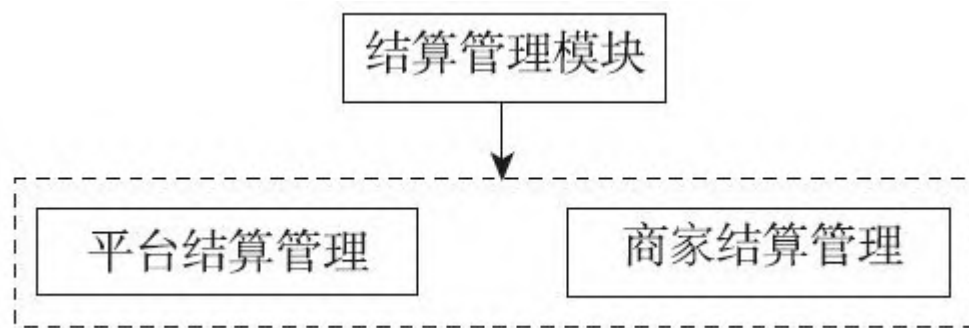


图8-25 结算管理模块的组成

1) 平台后台的结算管理模块：平台管理员可以在这里生成新的结算单并统一管理全平台的结算单，包括审核和付款等操作，此外还可以查看某个结算单的明细。

2) 商家后台的结算管理模块：商家可以在这里查看自己店铺已出账的结算单并进行确认或收款等相关操作。

如图所示。

8.6.2 设计思路

1. 结算数据表设计思路

结算数据表中需要存储结算单号、店铺ID、店铺名称、起始结算时间、结束结算时间、总金额、最终支付金额、结算状态和结算日期等字段。其中结算单的起止时间还用来调取某个结算单对应的订单明细列表。存储店铺名称的意义在于当列表中只要显示结算单所属店铺名称时可以避免连表查询带来的额外数据库性能开销。

2. 结算流程设计思路

结算单共有5个状态：已出账、已审核、已确认、已支付、已完成。

每个结算单都是一个商户在一段时间内的订单结算，是由平台管理员生成的。具体流程是管理员生成结算单，此时结算单状态为已出账，管理员进行结算单审核，如果审核通过则结算单状态修改为已审核，商家查看平台已审核的结算单并进行确认，此时结算单状态为已确认，平台对已经过商家确认的结算单进行款项支付，此时结算单状态变为已支付，商家最终确认收到该结算单款项后，结算单状态修改为已完成。如图8-26所示。

由于结算流程涉及资金流转，因此设计流程步骤时需要格外严谨。上面这个总体流程是由平台管理员和商家交叉完成的，很多步骤需要审核确认等操作，基本上避免了流程漏洞，较为安全可靠。

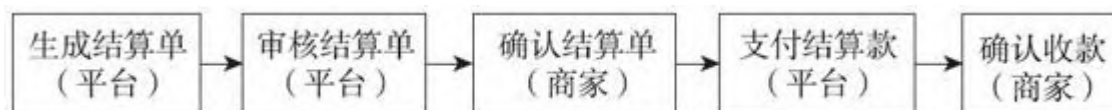


图8-26 结算流程图

3. 平台结算管理设计思路

O2O网站平台后台的结算管理涉及多个结算方面的功能，包括结算单的生成、审核、款项支付、查看结算单明细表等。

平台后台的结算管理页面为已生成的结算单的列表页面，显示结算单号、商铺名称、起始日期、结束日期、订单金额、结算金额、结算时间、结算状态等信息，管理员可以进行审核或支付款项的操作，该结算单列表具有根据商铺名称或结算状态进行搜索的功能。生成结算单的操作为点击一个“开始结算”按钮，之后后台会将消费时间截至昨天的未结算订单进行批量结算并生成结算单。每个生成的结算单可以点击“明细”查看该结算单的订单列表，结算单明细页面上会显示订单编号、用户名称、团购名称、下单时间、消费时间、数量、金额等。

8.6.3 代码实现

1. 生成结算单的代码实现

管理员在平台后台的结算管理点击“开始结算”按钮后，页面通过Ajax的方式请求服务器脚本统计并生成结算单，一切都是异步完成的。

代码清单8-15 订单结算单

```
//生成结算单
public function ajax_settleOp(){
    $model = Model();
    $data_array = array();
    $setting_info = $model->table('setting')->where(array(
        'name'=>'last_settle'))->find();
    $today_date = strtotime(date('Y-m-d',time()));
    if(intval($setting_info['value']) > 0){
        $condition = 'order.state=3 and
order.use_time>= '.$setting_info['value'].' and order.use_time <
 '.$today_date;
    }else{
        $condition = 'order.state=3 and order.use_time <
 '.$today_date;
    }
    $order = $model->
>field('order.store_id,order.store_name,order.price,order.use_time,
groupbuy.settle')->table('order,groupbuy')->join('left join')->on
    ('order.item_id=groupbuy.group_id')->where($condition)->
>limit('')->select();
    if(!empty($order)){
        foreach ($order as $val){
            $data_array[$val['store_id']] = array(
                'store_id'=>$val['store_id'],
```

```

        'store_name'=>$val['store_name'],
        'all_price'=>$data_array[$val['store_id']]
['all_price']+$val['price'],
'date_start'=>$data_array[$val['store_id']]['date_start']==''?
$val['use_time']:($val['use_time']<$data_array[$val['store_id']]
['date_start'])?
        $val['use_time']:$data_array[$val['store_id']]
['date_start']),
        'date_end'=>$data_array[$val['store_id']]
['date_end']==''? $val['use_time']:
($val['use_time']>$data_array[$val['store_id']]['date_start'])?
        $val['use_time']:$data_array[$val['store_id']]
['date_start']),
        'pay'=>$data_array[$val['store_id']]
['pay']+$val['settle']>0?($val['all_price']*
($val['settle']/100)):$val['all_price']
    );
    }
    $insert_array = array();
    foreach ($data_array as $da){
        $insert_array[] = array(
            'settle_sn'=>date('YmdHis'.$da['store_id'],time()),
            'store_id'=>$da['store_id'],
            'store_name'=>$da['store_name'],
            'date_start'=>$da['date_start'],
            'date_end'=>$da['date_end'],
            'amount'=>$da['all_price'],
            'final_pay'=>$da['pay'],
            'state'=>1,
            'settle_time'=>time()
        );
        $rs = $model->table('settle')->insertAll($insert_array);
        if($rs){
            //更新最后结算日期
            $model->table('setting')->
>where(array('name'=>'last_settle'))-> update(array
('value'=>$today_date));
            echo json_encode(array('done'=>true));die;
        }else{
            echo json_encode(array('done'=>false,'msg'=>'订单结算失
败'));die;
        }
    }
    }else{
        echo json_encode(array('done'=>false,'msg'=>'当前没有要结算
的订单'));die;
    }
}

```

程序流程为先从设置表中调取上次结算结束日期，之后调取从这个日期开始到昨天未结算订单的信息，调取到的订单信息按店铺进行分离分别进行统计并生成每个店铺的结算单，最后还要更新系统设置表中的结算结束日期。

2. 结算单明细的代码实现

结算单的明细页面调取的是该结算单统计的订单信息列表，具体调用时的数据库SQL条件为已经结算的并且属于这个店铺的，消费时间在结算单的结算起止时间之内的订单数据。

代码清单8-16 结算单明细

```
//结算单明细
public function settle_detailOp(){
    $settle_id = intval($_GET['settle_id']);
    if($settle_id <= 0){
        $this->showTip('参数错误','', 'error');
    }
    $model = Model();
    $settle_info = $model->table('settle')->where(array('
settle_id'=>$settle_id))->find();
    if(empty($settle_info)){
        $this->showTip('未找到该结算单信息','', 'error');
    }
    $condition = 'state = 3 and store_id =
' . $settle_info['store_id'] . ' and use_time >=
' . $settle_info['date_start'] . ' and use_time <=
' . $settle_info['date_end'];
```



```
        $order_list = $model->table('order')->where($condition)->
>page(15)-> order('use_time desc')->select();
        Tpl::output('settle_info',$settle_info);
        Tpl::output('order_list',$order_list);
        Tpl::output('show_page',$model->showpage(2));
        Tpl::showpage('settle.detail');
    }
```

8.6.4 功能体验

1. 平台结算单管理

平台结算管理包含结算单管理和查看结算单明细两个部分，其中结算单管理界面可以进行结算单的生成、审核等操作，结算单列表支持按商铺名称和结算状态的查询。结算单明细页面会显示出该结算单统计的所有订单信息，方便对账。如图8-27所示。



结算单号	商铺名称	起始日期	结束日期	订单金额	结算金额	结算日期	结算状态	操作
2014040916471213	三井拉面	2014-04-04	2014-04-04	58.00元	0.00元	2014-04-09 16:47:52	已逾期	审核 编辑
2014040415201413	三井拉面	2014-04-03	2014-04-03	174.00元	8.70元	2014-04-04 15:20:14	已逾期	审核 编辑

图8-27 平台结算单管理页面

结算管理 管理 结算单明细						
商户名称: 益州拉面 结算日期: 2014-04-03至2014-04-03 订单总额: 174.00元 结算金额: 8.70元						
列表						
订单编号	用户名称	团购名称	下单时间	消费时间	数量	金额(元)
2014040399564898	测试用户	奇石彩雕鱼	2014-04-03 16:38:39	2014-04-03 16:41:13	1	58
20140403152989757	测试用户	奇石彩雕鱼	2014-04-03 15:25:17	2014-04-03 16:49:03	1	58
						首页 上一页 1 下一页 末页

图8-28 结算单明细页面

2. 商户结算单管理

商户结算单管理功能可以查看所有本店铺已出账的结算单信息，支持按状态进行的过滤查看操作，可以进行结算单的确认和收款这两个操作，如图8-29所示。

商户中心 > 结算管理							
结算管理							
状态: 全部结算单 查询							
结算单号	起始日期	结束日期	订单金额	结算金额	结算日期	结算状态	操作
201404041408142	2014-04-03	2014-04-03	360.00元	268.00元	2014-04-04 14:08:14	已出账	已出账, 等待平台审核
						首页 上一页 1 下一页 末页	

图8-29 商户结算单管理页面

8.7 本章小结

本章对O2O这一电商模式进行了深入的探讨，对于O2O平台网站的重要模块线下商户、团购、优惠券、结算等进行了详细的实战开发讲解，特别是结算模块因为是和商家收入款项有直接的关系，因此对于结算流程开发时要格外地严谨，避免给网站带来不必要的麻烦。希望读者能在看完本章内容后可以对O2O这一商业模式有一个系统的了解，能对今后的开发有所帮助。

第9章 电商平台界面设计与用户体验

9.1 网页设计的那些事儿

当我们准备定义网页设计在开发过程中有着怎样的作用时，我们首先要确定一个问题：什么是设计？

“所谓设计，就是通过创造与交流来认识我们生活在其中的世界。”

——原研哉

创造、交流、认识，设计就是为了认识而存在的，而创造和交流只是其手段。网页设计就是通过使用更合理的颜色、字体、图片、样式以及特效类元素进行页面美化修饰，在功能限定的情况下，尽可能给予用户完美的视觉体验。

那么如何在一个产品开发团队中做好自己的角色，与程序员协同合作，将一款互联网产品实现用户体验度高，代码合理、易于维护、浏览器兼容性强等特性呢，这是所有网页设计师们最关心、也最现实的问题。本章将谈论一些网页设计工作中总结的常见问题和技巧，并通过对B2B2C电商网站几个重点功能页面分析与实例分析为大家更加直观地讲解。

9.1.1 视觉设计、交互设计、前端开发

术业有专攻，虽然都是从事互联网开发工作，但如果严格来说，“网页设计”这个名词似乎过于笼统了。网页设计又分为视觉设计（GUI）、交互设计（UCD）以及前端开发（Web Front-End Development），如图9-1所示。对于大型的互联网公司，这种细分是有必要的，分工明确，责任清晰，专注于自己的领域；而小型企业（一般20人以下），程序员、工程师往往占据了半壁江山，而视觉设计、交互设计和前端开发往往就是“网页设计”人员干的，因为那里不需要能够手绘、出矢量图的艺术家的，也不需要搞什么眼动仪研究把用户研究个底儿朝天的，只要能够画出网站草图，做一些简单的ICON图标，熟悉用户体验的基本原则，设计出实用的功能能满足客户的需求，能把网页写好，在主流浏览器下不变形……

换句话说，就是你把网页按指定需求写出来，那么你就是合格的网页设计师。

这种企业的做法错了吗？老板不尊重我的职位或压榨我的劳动价值了吗？如果永远纠结于自己的工作内容，细分工作范畴，或仅仅那个名称是什么，就会出现严重的协作脱节问题。

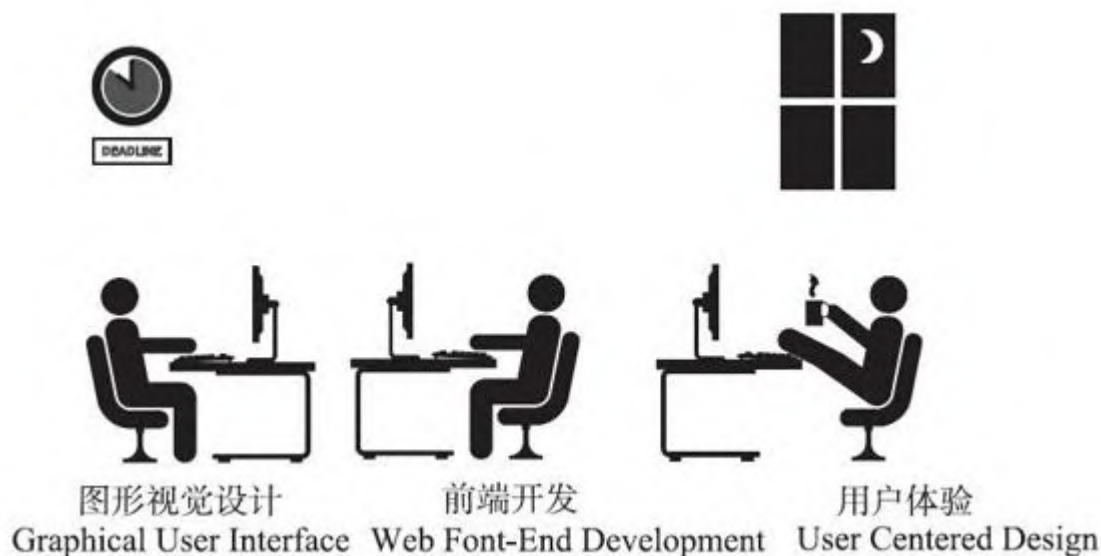


图9-1 网页设计工作具体内容划分

如果你把网页设计作为自己终身的职业，实际上，应具备各方面的素质，从作图到切片到书写

HTML、CSS，乃至把自己作为一位普通网站浏览者或客户，从他们的角度出发去分析需求。否则将永远无法突破自己的瓶颈，永远仅仅擅长绘制几个ICON、一些PSD原稿，或只能等着别人的设计图来切成网页。如果你现在正“身兼数职”，何不把自己就作为一个小的团队来沟通，自己设计的图形界面自己最清楚应该用什么合适的HTML标签来布局，靠什么CSS来绘制；不要等着别人给的“半成品”再去机械性地制作，主动去创造并将其实现。求人不如求己，当对图形设计、用户体验及前端代码都有所研究并专长其一二时，你将发现自己的职业生涯会变得豁然开朗，更多的机会将等待着你。

9.1.2 为代码增加可读性注释

一款互联网产品的开发过程中，可能需要很多人合作来完成，比如产品经理、构架工程师、程序员、图形设计、前端开发等。总的来说有可能会出现两种情况：一种是有明确的开发计划、内容及分工，可以有组织地合作，这可叫做“直接团队合作”；另一种是事先没有考虑到的，由于种种原因要求你去维护他人开发过的代码系统，可称其“间接团队合作”，例如我们的ShopNC商城系统客户定制及二次开发工作。

开发时首先要考虑到，现在编写的代码，将来很有可能需要别人进行后期的维护或更新，“间接团队合作”中常常出现之前没有考虑到的问题。即使你的工作是持续性开发，后期依旧由自己来维护，但随着时间的推移，在进行了其他内容的代码编写之后反过

来重新维护3个月前自己写的内容时，依旧会觉得陌生。因为这时候的网页模板已经布满了程序部分的循环，连文字都被语言包所代替。

所以不管是为了让团队中的其他人读懂你的模板代码，还是自己能搞清当初书写时的意图，一个保证代码良好可读性的最佳方法就是“注释”。

在写注释代码的时候。我们可以用“区域注释”、“单行注释”、“注释层级”和“协助注释”四个部分。大部分朋友使用的区域注释的方法会以“注释内容开始”或“注释内容结束”、“start”或“end”等，相比之下，以“S”或“E”开始更简单、易读。毕竟注释代码的时间越少越好。

(1) 单行注释

单行注释的信息应写在需注释的内容区域里，如代码清单9-1所示。

代码清单9-1 单行注释书写形式

```
<div>  
<!--注释内容-->  
...  
</div>
```

(2) 区域注释

在实际工作中，有时会出现分不清注释应该在标签之上还是标签之下的情况，为了避免这种情况，注释信息统一写在区域标签开始之前和结束之后，并以“S”或“E”开始，表示区域注释的开始或结束，如代码清单9-2所示。

代码清单9-2 单行注释书写形式

```
<!--S 注释内容-->  
<div> ... </div>  
<!--E 注释内容-->
```

(3) 注释层级

在模块制作中，可能会出现区域中还有区域的情况，为了更好地区分区域之间的层级，引入了注释层级的概念。区域注释前面的等号表示了当前注释的层级，通过这样的办法，也可以将注释的层级进行区分，如代码清单9-3所示。

代码清单9-3 注释层级书写形式

```
<!--1S 注释内容-->
...
<!--2S 注释内容-->
...
<!--3S 注释内容-->
...
<!--3E 注释内容-->
...
<!--2E 注释内容-->
...
<!--1E 注释内容--> ?
```

(4) 协助注释

协助注释是非作者维护时所加入的表示修改时间、修改人等标识信息。在区域注释或单行注释的基础上加上修改人和修改时间等信息，如代码清单9-4所示。

代码清单9-4 协助注释书写形式

单行协作注释：

```
<style type="text/css">
.class{
/*注释内容[修改人和修改时间]*/
...
}
</style>
<div>
<!-- 注释内容[修改人和修改时间]-->
... </div>
```

区域协作注释：

```
<style type="text/css">
/*S 注释内容[修改人和修改时间]*/
.class{
...
}
.class{
...
}
/*E 注释内容[修改人和修改时间]*/
</style>
<!--S 注释内容[修改人和修改时间]-->
<div> ... </div>
<!--E 注释内容[修改人和修改时间]-->
```

9.1.3 语义化的HTML5标签使用

HTML5是用于取代1999年所制定的HTML 4.01和XHTML 1.0标准的HTML标准版本；现在仍处于发展阶段，但大部分浏览器已经支持某些HTML 5技术。HTML 5有两大特点：首先，强化了Web网页的表现性能；其次，追加了本地数据库等Web应用的功能。广义论及HTML 5时，实际指的是包括HTML、CSS和JavaScript在内的一套技术组合。有时为了实现丰富的网络应用，浏览器不得依赖众多插件来实现，而HTML 5恰恰是希望将众多功能标准化并集合起来，从而使用浏览器减少对众多插件的依赖。

未来，HTML 5能做的事情非常多，涉及的方面也比较广，例如可以使用2D/3D canvas技术来完成过去需要Flash完成的一些效果，更好地嵌入多媒体API数

据，甚至于请求IP数据定位，等等。而我们这里要谈谈一些新的HTML标签的加入以及合理的使用。

最早接触网页时，也就是Web 1.0时代，我们都是使用table、tr、td这些标签去做网页，甚至于可以不使用CSS文件也能完成页面制作，只要你将图片切得合理，能利用表格将网页图切片聚在一起就好了，做网页就是如此简单。随着Web 2.0时代的到来，网页需要更大的信息承载量，更快的载入速度，更符合搜索引擎收录规则，这时DIV+CSS的组合来了。以CSS来控制网页中元素的样式，包括位置、大小、颜色等是其核心思想，相比table，无须再为布局定位而插入大量的重复标签，使HTML部分代码从样式、结构混淆的局面中摆脱。但因为CSS的强大，我们就应该忽视HTML标签的重要性了么？答案当然不是，如果只考虑最终的页面视觉效果实现，通过无数并无实际含义的标签+类命名制作网页，等于又重新回到了table的老路上。正确

的做法是，先分析页面布局结构，确定HTML语义化的标签，再选择合适的CSS实现最终浏览效果。

HTML 5引入了很多新的标记元素，基于我们讨论的课题，仅以片段类元素标签为例，如表9-1所示。

表9-1 HTML 5中新的片段类元素

元素 标 签	描 述
header	标记头部区域的内容（用于整个页面或页面中的一块区域）
footer	标记尾部区域的内容（用于整个页面或页面中的一块区域）
section	Web 页面中的一块区域
article	独立的文章内容
aside	相关内容或者引文
nav	导航类辅助内容

上述的这些标签都可通过原有的形式靠CSS设定样式。而HTML 5推崇表现与内容分离，更注重效率优先，更易于搜索引擎在抓取和索引网页的时候利用的页面元素。代码清单9-5是一个HTML 5页面的概貌，其中使用了新的DOCTYPE、字符集和语义化元素。

代码清单9-5 HTML 5语义化标签示例页面

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>HTML5</title>
<link href="style.css" rel="stylesheet" type="text/css">
</head>
<body>
<header>
  <h1>电子商务平台构建与开发实战</h1>
  <h2>电商平台的构成与规划</h2>
  <h4>标准化套件</h4>
</header>
<div id="container">
  <nav>
    <h3>目录</h3>
    <ul>
      <li><a href="javascript:void(0)">1.1.1  商城</a></li>
      <li><a href="javascript:void(0)">1.1.2  圈子</a></li>
      <li><a href="javascript:void(0)">1.1.3  CMS</a></li>
      <li><a href="javascript:void(0)">1.1.4  IM</a></li>
      <li><a href="javascript:void(0)">1.1.4  移动端</a></li>
    </ul>
  </nav>
  <section>
    <article>
      <header>
        <h1>买卖双方分离模式</h1>
      </header>
      <p>买卖双方各有独立的登录后台，商家可自行设置客服账号，分工明确，可更快速高效的工作，使平台对商家的结算及管理更便捷。</p>
      <footer><a href="javascript:void(0)">更多内容...</a>
    </footer>
  </article>
  <article>
    <header>
      <h1> 佣金结算模式 </h1>
    </header>
    <p>平台可按照商家入驻所选类目分别设置各个供应商佣金。交易成功后，平台可按照交易金额和佣金比例进行抽佣。实现平台快速盈利，平台与商家结算按照月结账期进行结算，保护平台与消费者的利益。</p>
    <footer><a href="javascript:void(0)">更多内容...</a>
  </footer>
</article>
```

```
</section>
<aside>
  <h3>环境要求 </h3>
  <p>ShopNC B2B2C系统具备跨平台特性，可运行于
Linux/FreeBSD/Unix及微软Windows 2000/2003/2008/XP/NT等各种操作
系统环境下。推荐使用Linux/FreeBSD操作系统。</p>
</aside>
<footer>
  <h2>未完待续...</h2>
</footer>
</div>
</body>
</html>
```

在上述代码中使用了语义化的标签，并无过多的class命名，保持了良好有序的网页组织结构，代码的可读性高，尤其是有利于搜索引擎的优化识别。图9-2所示为未关联CSS时的页面。



图9-2 HTML 5示例以页面去除样式后显示效果

没有样式的页面看起来枯燥、乏味、不美观，如代码清单9-6所示，与该页面对应的CSS样式代码里面融入了一些CSS3新特性，比如圆角（border-radius）、旋转变换（transform: rotate();）和阴影效果（box-shadow();）。CSS3与HTML 5一样也在开

发完善中，并且为了便于浏览器逐步支持，也采用了模块化的方式发布了规范。

代码清单9-6 HTML 5页面对应的CSS文件

```
@charset "utf-8";
/* CSS Document */
body {
    font: 14px/22px arial, "microsoft yahei";
    background-color: #CCC;
    max-width: 980px;
    margin: 0 auto;
    border: solid 4px #FFF;
}
ul, li {
    margin: 0;
    list-style-type: none;
}
h1 {
    font-size: 32px;
    line-height: 32px;
    margin-top: 30px;
}
h2 {
    font-size: 18px;
    line-height: 20px;
    color: #72F0E5;
    text-align: center;
    margin: 0;
}
h3 {
    font-size: 14px;
    color: #F47D31;
    text-align: center;
    margin: 0;
}
h4 {
    color: #FFF;
```

```

        line-height: 28px;
        background-color: #FAA933;
        text-align: center;
        padding: 0 80px;
        position: absolute;
        top: 30px;
        left: -60px;
        -webkit-box-shadow: 2px 2px 20px #1C9B8E; /*WebKit核心浏览器识别阴影效果*/
        -moz-box-shadow: 2px 2px 10px #1C9B8E; /*If Firefox*/
        box-shadow: 2px 2px 10px #1C9B8E;
        -webkit-transform: rotate(-45deg); /*WebKit核心浏览器识别旋转-45度效果*/
        -moz-transform: rotate(-45deg); /*If Firefox*/
        transform: rotate(-45deg);
    }
    header {
        color: #FFF;
        background-color: #1FBBA6;
        text-align: center;
        height: 150px;
        display: block;
        position: relative;
        overflow: hidden;
    }
    nav {
        display: block;
        width: 25%;
        float: left;
    }
    nav a:link, nav a:visited {
        font-weight: bold;
        text-decoration: none;
        display: block;
        padding: 10px;
        border-bottom: solid 3px #FFF;
        margin: 5px;
    }
    nav a:hover {
        color: #FFF;
    }
    nav h3 {
        margin: 15px;
        color: #1FBBA6;
    }

```

```

#container {
    background: #EAEDF4;
}
section {
    display: block;
    width: 50%;
    float: left;
}
article {
    background-color: #FFF;
    display: block;
    margin: 10px;
    padding: 10px;
    -webkit-border-radius: 5px; /*WebKit核心浏览器识别圆形边
角效果*/
    -moz-border-radius: 5px; /*IF Firefox*/
    border-radius: 5px;
    -webkit-box-shadow: 2px 2px 10px #DEE3EB;
    -moz-box-shadow: 2px 2px 10px #DEE3EB;
    box-shadow: 2px 2px 10px #DEE3EB;
}
article header {
    height: auto;
    padding: 5px;
    -webkit-border-radius: 5px;
    -moz-border-radius: 5px;
    border-radius: 5px;
}
article footer {
    background-color: transparent;
    text-align: right;
    padding: 5px;
    -webkit-border-radius: 5px;
    -moz-border-radius: 5px;
    border-radius: 5px;
}
article footer a {
    padding: 4px 8px;
    border-radius: 5px;
}
article h1 {
    font-size: 18px;
    line-height: auto;
    margin: 0;
}

```

```
aside {
    display: block;
    width: 25%;
    float: left;
}
aside h3 {
    margin: 15px;
    color: #1FBBA6;
}
aside p {
    margin: 15px;
    color: #555;
    font-weight: bold;
    font-style: italic;
}
footer {
    clear: both;
    display: block;
    background-color: #25C8B3;
    color: #FFF;
    text-align: center;
    padding: 15px;
}
footer h2 {
    font-size: 14px;
    color: #FFFFFF;
}
/* 链接色 */
a {
    color: #1FBBA6;
}
a:hover {
    color: #FFF;
    background-color: #FAA933;
    text-decoration: underline;
}
```

示例中的标签和样式使用仅仅为了突出HTML 5新增标签以及CSS3部分新特效的表现，结构比较简单，

语义化的标签也远远不止上述内容，但无实体含义的布局块，DIV命名class依旧是我们日常工作中的常用，不过笔者认为合适的地方使用合适的标签是提高代码质量的手段之一。图9-3为附加样式后的页面内容及标签说明。

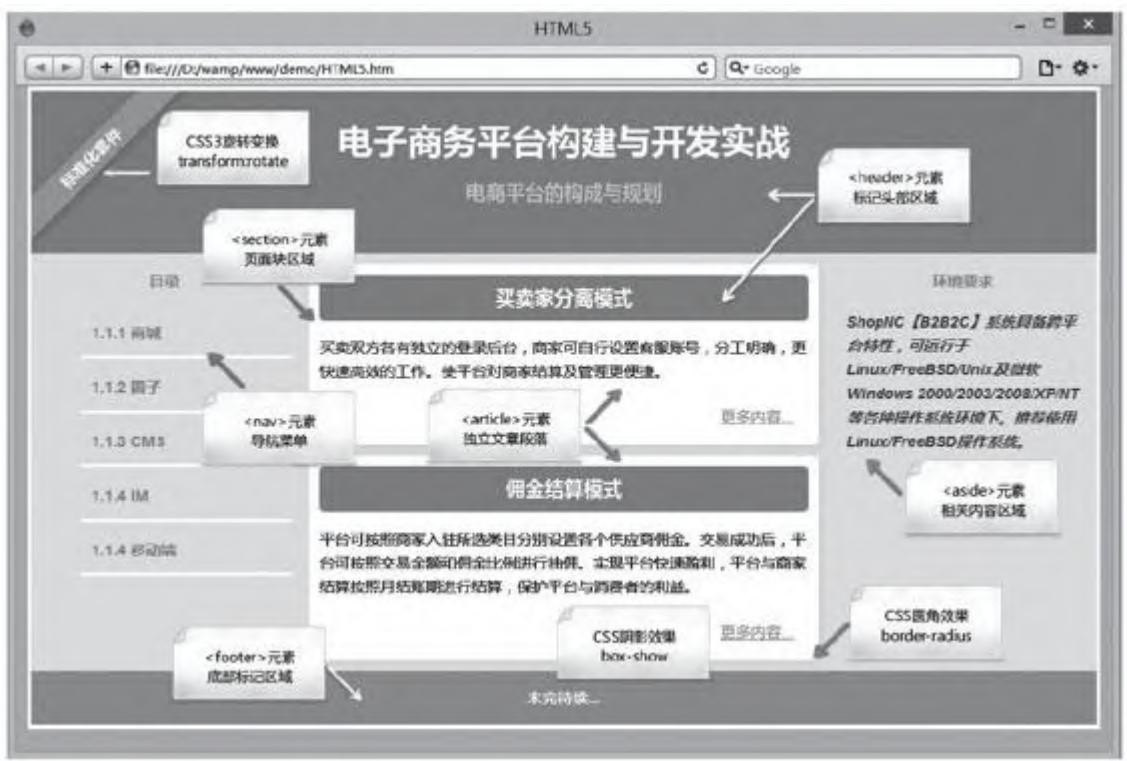


图9-3 HTML 5示例页面预览及标签说明

9.1.4 base.css会减轻你的工作负担

我们提出要尽量使用语义化的HTML标签来书写代码。顾名思义，语义化自然有其特定的含义，例如：<h1>~<h6>表示内容标题，显示的字体大小也根据不同的浏览器有所区别；表示强调文本，而在浏览器中表现的形式却和标签<i>一样都是倾斜显示；和会给列表项前面加入行符号……这些也许我们都不希望出现。所以也就需要先对一些HTML标签进行格式化，把默认的样式去掉，只将其作为语义标签应用。代码清单9-7为电子商城系统所使用的base.css，文件中重新定义标签部分。

代码清单9-7 商城系统base.css文件重新定义标签部分

```
/* =====  
* 重新定义HTML元素
```

```

* ===== */
html, body, div, ul, ol, li, dl, dt, dd, h1, h2, h3, h4, h5,
h6, pre, form, p, blockquote, fieldset, input {
    padding: 0;
    margin: 0;
}
h1, h2, h3, h4, h5, h6, pre, code, address, caption, cite,
code, em, strong, table, th, td {
    font-size: 1em;
    font-style: normal;
    font-weight: normal;
}
strong {
    font-weight: bold;
}
ul, ol {
    list-style: none outside none;
}
fieldset, img {
    border: medium none;
}
caption, th {
    text-align: left;
}
table {
    border-collapse: collapse;
    border-spacing: 0;
}
body {
    font: 12px/20px Arial, Verdana, "宋体", "Lucida
Grande", "Lucida Sans Unicode", Helvetica, sans-serif;
    color: #666;
    background: #FFF none repeat scroll 0 0;
    min-width: 1200px;
}
input, select, textarea {
    font: 12px/20px Tahoma, Helvetica, Arial, sans-
serif;
}
i, cite, em {
    font-style: normal;
}
input, button, select, textarea {
    outline: none
}

```

```
html {
    min-height: 101%;
}
/* 链接 */
a {
    color: #333;
    text-decoration: none;
    outline: medium none;
}
a:link, a:visited, a:active {
    text-decoration: none;
}
a:hover {
    color: #D93600;
    text-decoration: underline;
}
/* =====
 * 为旧版本浏览器格式化HTML5元素
 * ===== */
article, aside, dialog, footer, header, section, footer,
nav, figure, menu {
    display: block;
}
```

此外在实际工作中，经常会遇到一些CSS属性的重复使用情况，最常见的就是字体、颜色、边距、间距、边框、浮动，等等。这些也可以写入base.css文件内，在标签中添加对应的class名称，减少代码量，提高效率，如代码清单9-8所示。

代码清单9-8 商城系统base.css文件字体、宽高、间距、边距等截取内容

```
/*文字排版
*****/
.fl2 { font-size:12px;}
.fl4 { font-size:14px;}
.fl6 { font-size:16px;}
.fb { font-weight:bold;}
.fn { font-weight:normal;}
.unl { text-decoration:underline;}
.no_unl { text-decoration:none;}
.noborder { border:none 0;}
/*定位
*****/
.tl { text-align:left !important;}
.tc { text-align:center !important; }
.tr { text-align:right !important;}
.bc { margin-right:auto ;margin-left:auto;}
.fl { float:left; display:inline;}
.fr { float:right !important; display:inline;}
.cb { clear:both;}
.cl { clear:left;}
.cr { clear:right;}
.vm { vertical-align: middle;}
.vt { vertical-align: top;}
.pr { position:relative;}
.pa { position:absolute;}
.zoom { zoom:1}
/*.hidden { visibility:hidden;*/}
.none { display:none;}
.clear { clear: both; height: 0; font-size: 0; line-height:0; }
/*长度高度
*****/
.w10 { width:10px;}
.w20 { width:20px;}
.w30 { width:30px;}
.w { width:100%;}
.h20 { height: 20px;}
```

```

.h { height:100%}
/*边距
*****/
.m0 { margin: 0!important; }
.m10 { margin:10px;}
.mt5 { margin-top:5px;}
.mt10 { margin-top:10px;}
.mb5 { margin-bottom:5px;}
.mb10 { margin-bottom:10px;}
.ml5 { margin-left:5px;}
.ml10 { margin-left:10px;}
.mr5 { margin-right:5px;}
.mr10 { margin-right:10px;}
/*边距
*****/
.p10 { padding:10px;}
.pt5 { padding-top:5px;}
.pt10 { padding-top:10px;}
.pb5 { padding-bottom:5px;}
.pb10 { padding-bottom:10px;}
.pr5 { padding-right:5px;}
.pr10 { padding-right:10px;}
.hr { font-size: 1px; display: block; height: 0; margin:
12px 0; border-top: 1px
    solid #E3E3E3; overflow: hidden;}
.hr-double { height: 1px; border-top: 1px solid #E3E3E3;
border-bottom: 1px solid #E3E3E3;}
.hr8 { margin: 8px 0;}
.hr16 { margin: 16px 0;}
.hr32 { margin: 32px 0;}
.hr.dotted { border-top-style: dotted;}
.hr-double.dotted { border-bottom-style: dotted;}

```

最后，作为一个大型电商网站，会有很多公共组件在模板页面中多次出现，例如一些商品图的尺寸及控制，网站的顶部、足部内容等，这些部分的样式可以放在common.css文件里，但为了减少模板页关联过

多的CSS文件，让后期维护更明朗，我们也可以选择把它放在base.css中。根据网站的风格需要，模板对于部分样式的使用频率，灵活掌握基础（base.css）与公共（common.css）的完善补充，这能有效地减少代码量，提高效率，也易于后期维护。一个模板页面中，可能会引入以下三种内容的CSS文件来控制样式，如图9-4所示。

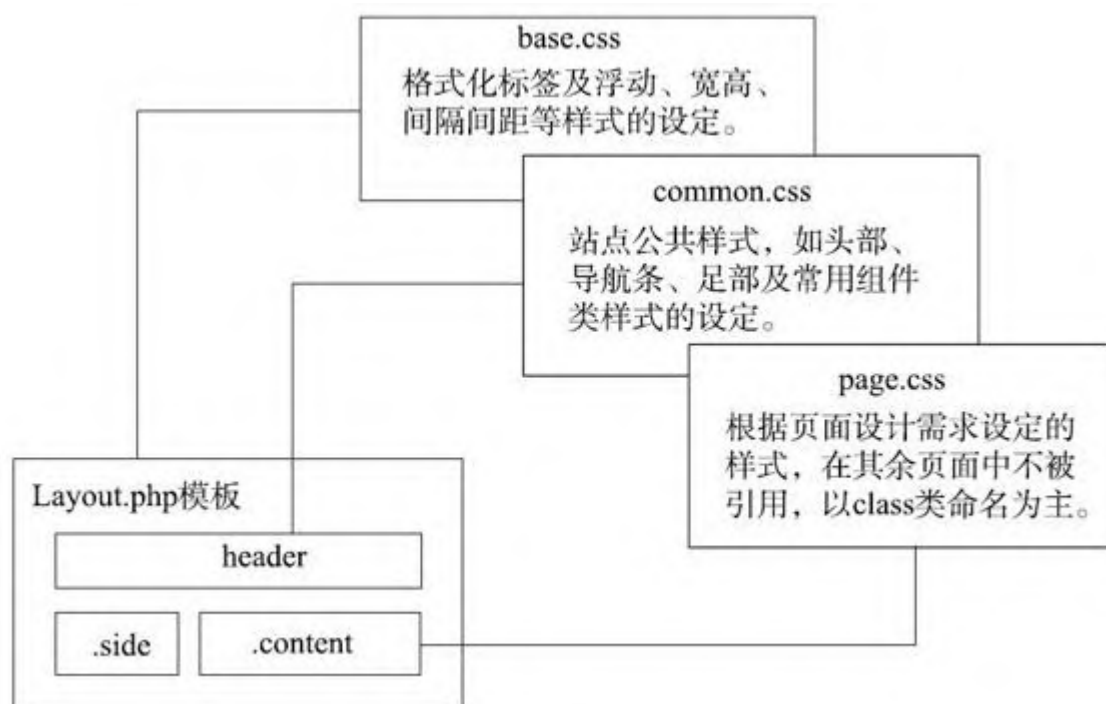


图9-4 模板页面中用到的CSS文件

9.1.5 如何给CSS命名

有了语义化的标签，又有了base.css和common.css的基础公共样式定义，接下来的工作就是页面中具体块的样式设定，这就离不开class和id的命名问题。养成清楚、明确的命名习惯，对于前期开发与后期维护都有着很大的影响。命名应尽量使用具有含义的小写英文单词，如果一个单词无法表明含义class，可以用“-”（中划线）进行连接，id可以用“驼峰式”写法，即“#AaaBbbCcc”形式；而像商城系统这种具有商城前台、用户中心、卖家中心等多栏目的大型网站，还可以加入前缀代表其含义，这个规则可以根据自己的开发习惯和团队要求来决定。例如“.nch-goods-liast”或“.ncm-order-list”从字义上来看，这是一个商品列表的样式命名，而前缀nch（nc homepage缩写）使用在首页页面，ncm（nc

members缩写) 使用在用户中心页面, 这样就能很清楚地知道其使用位置。笔者不建议使用过短的缩写形式来为class命名, 比如“.r-l-t”这样的命名, 这样的命名虽然简单, 但时间长了估计自己也不知道当初起名时的目的和含义, 尤其是大型电商网站, 涉及的模板页面很多, 样式也是数以千计, 看似复杂的命名加上段落注释却可以在后期帮你的团队提升效率。

我们以图9-5为例。这是一个简单的商品列表形式, 只为ul标签上的class进行了命名, 而区域块内部的列表、图片、名称、价格则直接使用特定标签, 代码结构更清晰准确, 同时减少了过多的class命名工作, 命名实例及对应样式如代码清单9-9所示。



图9-5 class命名实例

代码清单9-9 class命名实例对应样式

```
<style type="text/css">
.nch-goods-list { background-color: #F5F5F5; padding: 10px;
overflow: hidden;}
.nch-goods-list li { display: block; padding: 5px 10px;
border-bottom: solid 1px #EEE; clear: both;}
.nch-goods-list .goods-thumb { background-color: #FFF;
width: 38px; height: 38px; vertical-align: top; display:
inline-block; border: solid 1px #999;}
.nch-goods-list .goods-thumb img { max-width: 36px; max-
height: 36px; padding: 1px;}
.nch-goods-list dl { vertical-align: top; display: inline-
block; margin-left: 10px; }
.nch-goods-list dt { font-weight: 600; }
.nch-goods-list dd { color: #999; }
.nch-goods-list em { font: 600 14px/20px Verdana; font-
```

```
style: normal; color: #F30; margin: 0 4px;}
</style>
<ul class="nch-goods-list">
  <li>
    <div class="goods-thumb"></div>
    <dl>
      <dt>商品a</dt>
      <dd>价格: <em>100.00</em>元</dd>
    </dl>
  </li>
  <li>
    <div class="goods-thumb">
  </div>
  <dl>
    <dt>商品b</dt>
    <dd>价格: <em>200.00</em>元</dd>
  </dl>
</li>
  <li>
    <div class="goods-thumb">
  </div>
  <dl>
    <dt>商品C</dt>
    <dd>价格: <em>500.00</em>元</dd>
  </dl>
</li>
</ul>
```

笔者总结了一些电子商务网站在实际开发及维护中常用的样式命名规则，如表9-2所示。

表9-2 电子商务网站常用样式命名

CSS样式命名	描 述	使 用 范 围
.wrapper	页面外围，控制整体布局宽度	布局及公共常用
.container	容器，用于最外层	布局及公共常用
.layout	布局	布局及公共常用
.topbar	顶部条	布局及公共常用
.header	头部区域	布局及公共常用
.footer	足部区域	布局及公共常用
.nav	主导航	布局及公共常用
.subnav	二级导航	布局及公共常用
.sidebar	侧边栏	布局及公共常用
.breadcrumb	面包屑导航（所在位置）	布局及公共常用
.member	会员、买家	电商网站常用
.seller	商家、卖家	电商网站常用
.store	店铺	电商网站常用
.goods	商品	电商网站常用
.brand	品牌	电商网站常用
.order	订单	电商网站常用
.price	价格	电商网站常用
.cart	购物车	电商网站常用
.complain	投诉	电商网站常用
.comment	评论	电商网站常用
.evaluation	评价、评分	电商网站常用
.stat	统计	电商网站常用
.groundbuy	团购	电商网站常用
.coupon	优惠券	电商网站常用

(续)

CSS样式命名	描 述	使 用 范 围
.promotion	促销	电商网站常用
.discount	折扣	电商网站常用
.refund	退款	电商网站常用
.return	退货	电商网站常用
.transport	运输物流	电商网站常用
.address	地址	电商网站常用
.pay	支付	电商网站常用
.recharge	充值	电商网站常用
.accounts	结算	电商网站常用

9.1.6 应该用id还是class

作为CSS的命名方式，我们有多种选择，常用的形式为类class或id，如图9-6所示。例如使用Dreamweaver添加一个新的样式时，会首先让开发者进行类型选择及命名。从软件的介绍中可以看出，class可应用于任何HTML元素，而id仅应用于一个HTML元素，换句话说，一个页面中，class可以反复出现多次而id仅能出现一次。

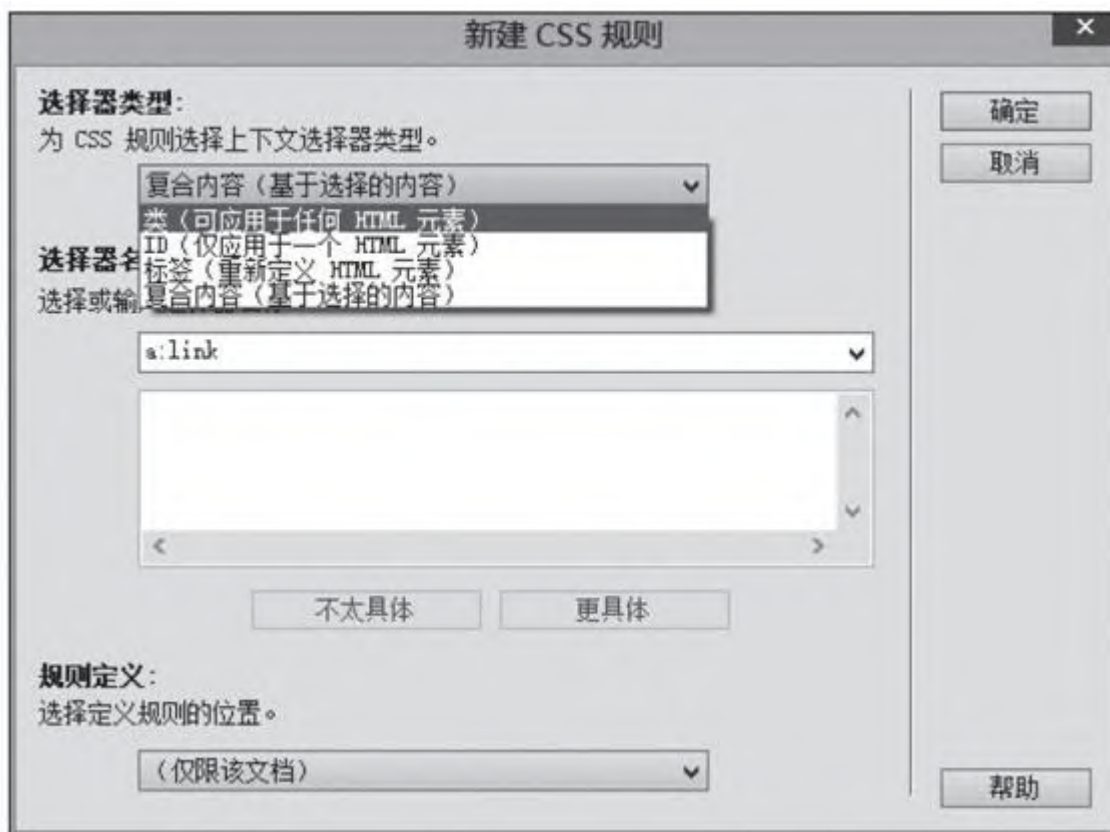


图9-6 Dreamweaver中，对新建css的选择

具有id的CSS选择符权重远远大于class，当页面中出现class命名冲突或者父级或子级样式干扰等问题，我们可以通过特定位置上加入id定义样式来解决，如代码清单9-10所示。

代码清单9-10 标签、class及id的权重关系代码演示

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>HTML/id/class</title>
<style type="text/css">
li { font-size: 12px; line-height: 20px; background-color:
#FFF; border: 1px dashed #CCC; padding: 5px; margin-bottom:
5px; list-style-type: none;}
li.repeat { font-weight: bold; background-color: #CCC;
border: 1px dashed #FFF;}
li#key { font-size: 18px; font-style: italic; font-weight:
normal; color: #FFF; background-color: #666; border: none;}
</style>
</head>
<body>
<ul>
  <li>正常的列表项，靠标签也能控制</li>
  <li>正常的列表项，靠标签也能控制</li>
  <li id="key" class="repeat">唯一性的样式可以用id来区别，权重大
于class</li>
  <li class="repeat">用class可以让列表项不一样，并出现多次</li>
  <li class="repeat">用class可以让列表项不一样，并出现多次</li>
  <li>正常的列表项，靠标签也能控制</li>
  <li>正常的列表项，靠标签也能控制</li>
</ul>
</body>
</html>
```

标签li本身被定义了样式，具有class命名的li会区别于其他，而同时具有id的li则获得了最高权重，样式也是唯一的。如图9-7所示。



图9-7 标签、class及id的权重关系代码生成效果

如果是纯静态的HTML页面，使用id还是class其实都可以，只要遵循重复性和唯一性的原则，但作为商城类型的页面模板，内容会关联PHP、JS等相关文件及操作，一般情况下，建议尽量使用class，把id留给程序和JS部分定义选择符吧。

9.1.7 能少用就少用CSS Hack

自从CSS布局诞生的那天开始，CSS Hack和滤镜曾一度成为使用中不可避免的恶魔。CSS Hack是为页面在不同内核浏览器获得较为一致的表现效果时，采用的一种辅助手段。当页面HTML结构和CSS样式书写都很符合Web标准时，页面在良好支持Web标准的浏览器下一般是不需要CSS Hack的，正是由于对Web标准支持不够良好的浏览器的存在，才有CSS Hack的产生。

知道了什么是CSS Hack，还得知道它的原理是什么，这样也有利于以后的编写过程。其实就是上面所提到的，由于不同的浏览器对CSS的支持及解析结果不一样，还根据CSS中的优先级的关系，来书写CSS Hack，这对于认识不同浏览器的解析特性、解析优先级，从而来写CSS Hack代码就至关重要了。

(1) IE Hack

IE系列浏览器的Hack大致如下：

```
_nowamagic:1px;-----ie6  
*nowamagic:1px;-----ie7  
nowamagic:1px\0;-----ie8  
nowamagic:1px\9\0;-----ie9  
:root nowamagic:1px;-----ie9 (实际情况可能IE 9还是有问题，再用这种方式)
```

这样就基本上可以兼容所有IE。

(2) Firefox与Chrome的Hack

· Firefox：

```
@-moz-document url-prefix()
```

· Chrome：

```
@media screen and (-webkit-min-device-pixel-ratio:0)
```

如图9-8所示，为部分浏览器对CSS Hack兼容性使用列表。

随着各种浏览器版本的不断更新，之前为了兼容性而存在的CSS Hack却会为未来新浏览器的兼容性带来隐患，良好的书写习惯，符合W3C的页面标准，随着老版IE浏览器的逐步淘汰，相信CSS Hack的使用将越来越少。

标准模式	IE6	IE7	IE8	IE9	FF	Chrome	使用示例
-	√	×	×	×	×	×	#tmp(-color:red;)
_	√	×	×	×	×	×	#tmp(_color:red;)
+	√	√	×	×	×	×	#tmp(+color:red;)
=	√	√	×	×	×	×	#tmp(=color:red;)
~	√	√	×	×	×	×	#tmp(~color:red;)
!	√	√	×	×	×	×	#tmp(!color:red;)
@	√	√	×	×	×	×	#tmp(@color:red;)
#	√	√	×	×	×	×	#tmp(#color:red;)
\$	√	√	×	×	×	×	#tmp(\$color:red;)
%	√	√	×	×	×	×	#tmp(%color:red;)
^	√	√	×	×	×	×	#tmp(^color:red;)
&	√	√	×	×	×	×	#tmp(&color:red;)
*	√	√	×	×	×	×	#tmp(*color:red;)
'	√	√	×	×	×	×	#tmp('color:red;)
\9	√	√	√	√	×	×	#tmp(color:red\9;)
html>/**/body	×	×	√	√	√	√	html>/**/body #tmp(color:red;)
:root	×	×	×	√	√	√	:root #tmp(color:red)

图9-8 部分浏览器对CSS Hack兼容性使用列表

9.1.8 一些CSS和浏览器之间的怪异显示及对应的解决办法

有的时候编写的CSS样式没有什么错误，但在浏览器中却会出现一些很奇怪的问题，明明没有间距的地方出现了间距，没有空隙的地方多了一条白线，或者IE、Firefox下显示的效果截然不同，这些依旧与前面关于CSS Hack使用中所提到的原因相同。下面列举几个在工作中经常会用到的样式属性所存在的问题及其解决方案。

(1) inline-block的兼容与4px间隙问题。

在实际工作中，我们会经常用到“inline-block”（内联块）这个属性来代替“float”来使内容横向排列，而不产生浮动。但IE 6、IE 7下并不能很好地支持该属性，即使支持这个属性的Firefox中依

旧会产生4px的空白间隙。例如，我们需要制作一组横向导航菜单，如代码清单9-11所示。

代码清单9-11 使用“inline-block”属性制作的横向菜单

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title></title>
<style type="text/css">
ul li {
    list-style-type: none;
    margin: 0;
    padding: 0;
    border: 0;
}
li {
    background-color: #CCC;
    display: inline-block;
    width: 100px;
    height: 30px;
}
</style>
</head>
<body>
<ul>
    <li>1</li>
    <li>2</li>
    <li>3</li>
    <li>4</li>
</ul>
</body>
</html>
```

li标签使用了“inline-block”触发Layout，形成行布局，但IE 7下的显示依旧是纵向排列，Firefox下则出现了间距，即使li上并没有margin和border属性的设置，如图9-9所示。

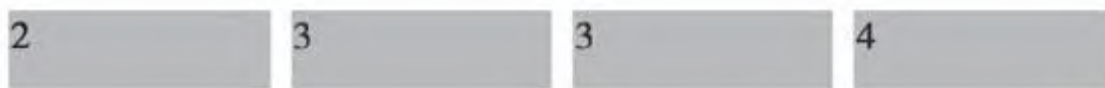


图9-9 Firefox浏览器下inline-block属性会产生间隔

想解决上面的问题，代码如下：

```
ul { font-size: 0;}
li { font-size: 12px; background-color: #CCC; vertical-align: top; display: inline-block; *display: inline/*IE7*/; width: 100px; height: 30px; *zoom: 1/*IE7*/;}
```

父级ul添加了“font-size: 0”、子级li添加“font-size: 12px”来设置字体大小；“vertical-align: top”垂直居上对齐，解决内联块之间的空隙问题；“*display: inline/*IE7*/; *zoom: 1/*IE7*/; ”触发IE 7下Layout并定义块为内联，这

时，在所有浏览器下都能很好地兼容“display: inline-block”了，如图9-10所示。

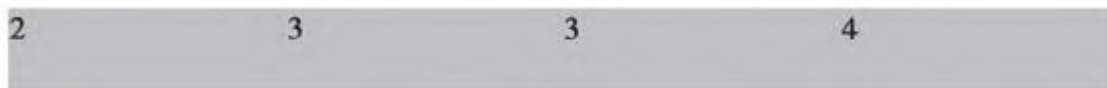


图9-10 Firefox浏览器兼容正常显示效果

(2) margin-top、margin-bottom的重叠。

当相邻的两个盒模型具有margin-top和margin-bottom属性时，浏览器会取其中大的值作为间距，如代码清单9-12所示。

代码清单9-12 相邻的布局块具有margin-top和margin-bottom属性

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title></title>
</head>
<body>
<div>1</div>
<div>2</div>
```

```
</body>
</html>
```

这时块1与块2之间并不是15px和20px相加的间隔值，而是只取了块2的“margin-top: 20px”，因此可以改用padding来做间隔或只设定两个块其中之一的margin属性，还可以给它们之间加一个具有“clear: both”属性的DIV做分割。

(3) 图片同比例缩放，DIV内水平垂直居中。

如果块内只有单行的文字，那么实现水平和垂直方向上都居中很简单，如代码清单9-13所示。

代码清单9-13 单纯文字内容在DIV中水平垂直居中

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title></title>
</head>
<body>
```



```
<div>文字居中</div>
</body>
</html>
```

但如果内容是一组图片且大小未知，宽和高都不一样该如何实现水平和垂直方向上都居中呢？如代码清单9-14所示。

代码清单9-14 未知大小图片同比例缩放后水平和垂直方向上居中

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title></title>
<style type="text/css">
* { padding: 0; margin: 0}
html { min-height: 101%}
body { font-family: verdana, arial, tahoma; font-size: 12px;
color: #333}
div { margin: 0 auto}
ul, ol, li { list-style: none}
a { text-decoration: none; word-wrap: break-word}
a:hover { text-decoration: underline}
img { border: 0}
p { line-height: 26px}
/* 容器 */
.middle { font-size: 0; width: 880px; margin: 10px auto;
border: solid 4px #999;}
.middle li { font-size: 12px; background-color: #F5F5F5;
display: inline-block; *display: inline/*IE7*/; margin: 1%;
width: 23%; *zoom: 1/*IE7*/;}
```

```

/* 未知大小的图片在已知容器中的垂直居中和水平居中 */
.middle li img { margin-top:expression(100-this.height/2);
max-width: 200px; max-height: 200px;}/* max-width与max-
height可限制图片的最大宽高度 */
.middle li a { text-align: center; vertical-align: middle;
display: table-cell; width: 200px; height: 200px; overflow:
hidden;}
</style>
</head>
<body>
<!--代码开始-->
<ul class="middle in-box">
  <li><a href="javascript:void(0)"></a>
</li>
  <li><a href="javascript:void(0)"></a>
</li>
  <li><a href="javascript:void(0)"></a>
</li>
  <li><a href="javascript:void(0)"></a>
</li>
</ul>
<!--代码结束-->
</body>
</html>

```

实现效果如图9-11所示，在商城商品图片列表中经常会遇到这样的问题，应特别注意。



图9-11 未知大小图片同比例缩放水平与垂直方向上
居中的浏览效果

9.2 商城首页

9.2.1 B2B2C商城首页设计元素

在前面的章节里，介绍了一些网页设计的技法及常见问题。下面，我们以ShopNC B2B2C商城系统为例，具体说明一下在电商平台开发中所需的设计思路、注意事项以及制作技巧。

商城的首页，几乎是每个访问者都会看到的页面，而且是有些访问者会看到的唯一页面，这仿佛就是Web上的海景地产，页面上的各处位置都让人渴望却又“供应不足”，与网站利益相关的信息都希望在首页上有一席之地，但仅仅几屏的空间又会造成内容“大塞车”。那么如何分配以利用好各个空间呢？

首页是一个网站整体风格定位的标志性页面，站内所有的次级页面都是以其为风格基准进行延续设计，如何合理把握这个风格定向标式的页面呢？

商城运营方作为还希望经常进行首页换装，内容更新，让浏览者几乎每天都能在首页看到不一样的广告与商品，但维护时间和成本越低越好，那么如何做到最快、最简单呢？

带着这一连串的问题，来看看ShopNC B2B2C商城系统的首页设计（见图9-12），让我们逐一对首页布局区块的设计思路与作用进行解读。

- **顶部条**（Topbar）。可用来放置商城欢迎语、用户登录以及快捷菜单。全站公用这个顶部条，用户可以在站点的任何页面位置都清楚网站的标题名称、如何注册、登录操作，了解还可以做哪些常用操作，

等等。对于栏目页面众多的综合性网站，这个顶部条是必不可少的元素之一。

- **头部**（Header）。顾名思义，就像人体的头部一样，商城系统的这部分也是最重要的部位。商城的Logo、搜索条以及用户菜单、购物车被布局其间。对于购买目的明确的用户，搜索条无疑是最快速、最有效找到商品的途径；而用户菜单和购物车则是进行购物的必要操作；因此这部分也是作为全局型设计而出现。

- **导航及分类菜单**（Nav&Category）。B2B2C商城区别于传统的B2C模式，不仅仅因为它是多店铺系统，更重要的是，它有一套完善的门户型电子商务整站解决方案，栏目丰富，包含了团购、活动专题、资讯、社交圈等频道，因此主导航菜单也就承担了这个导向的作用。而商城本身最重要的职能还是卖东西，

如何能让用户从成千上万的商品中找到自己想要的，分类索引菜单可以胜任该职。导航条及分类菜单也被作为全局型设计。

· **焦点广告**（Focus Layout）。除去了上面说到的三块全局型设计模块之外，在访问者打开站点时，第一屏还需要看到什么？对了，广告，而且要尽可能多的广告。广告上醒目的口号、吸引人的价格以及精美的图片往往是商家最有效的促销手段之一。这就好比是一个商场里面的橱窗广告，甚至于比那些来得更直接，因为只要点击就可以把用户引向商品页面，进而形成一次潜在的消费。

· **信息区域**（Information）。站点公告、新闻、商家入驻、消费保障承诺等可以放在这部分次重点区域。

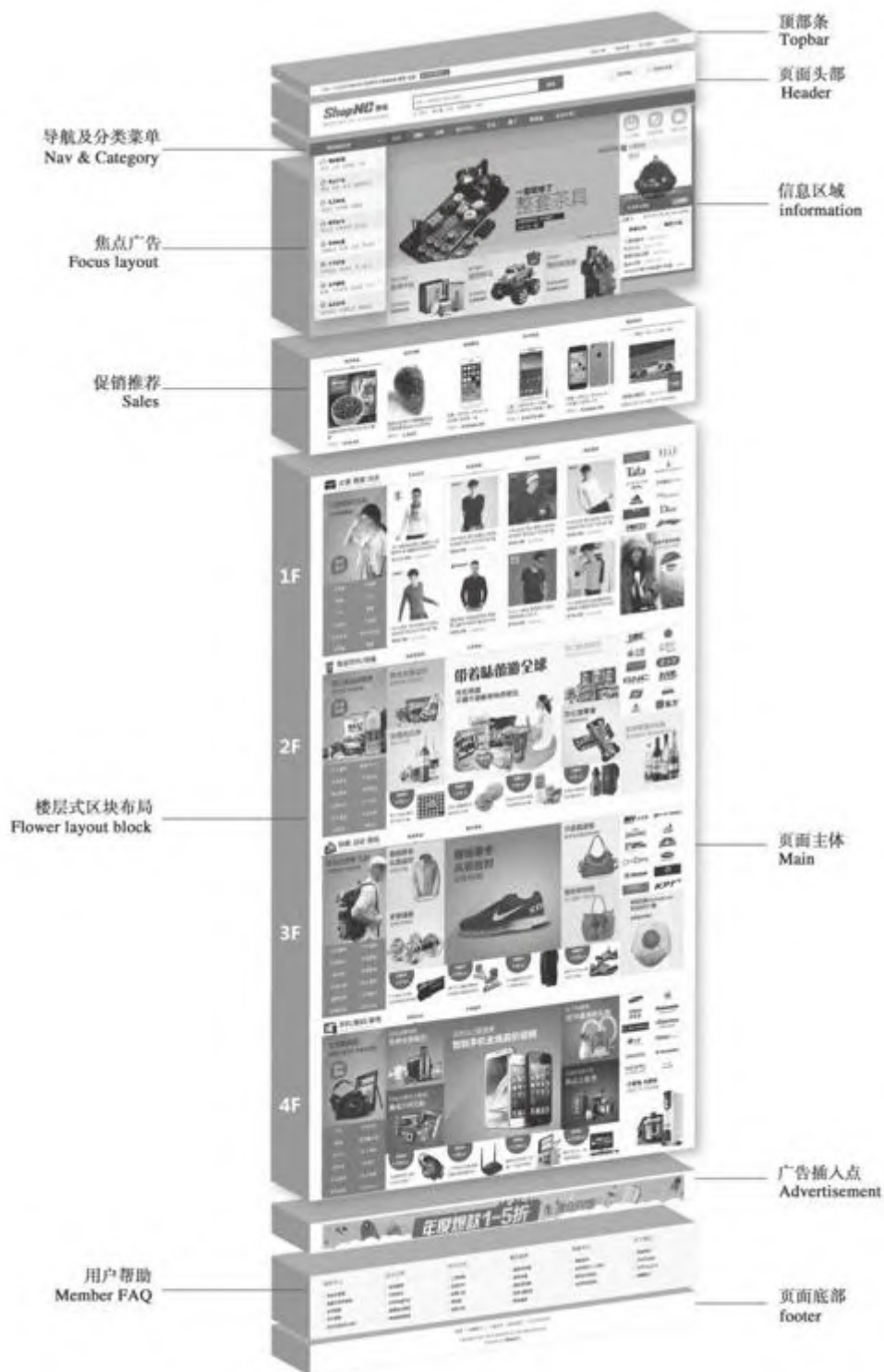


图9-12 B2B2C商城首页布局块分析

· **促销推荐** (Sales)。依旧是推广型商品展示的橱窗位置，可以直接把访问者引向商品页面。但由于访问者所使用的浏览器分辨率不一，台式机显示器横宽比例比较大，这部分可以出现在第一屏，又比如用笔记本浏览，那么一屏的高度很有限，这部分可能需要滚动鼠标后才会出现在第二屏的顶部了。考虑到这些特性，促销区域应算为次级推广区域。

· **页面主体** (Main)。楼层式区块布局让商城首页的主体既统一又丰富。每一层的布局形式保持一致，不会让人觉得杂乱无章；但各区块所出现的广告、商品、品牌等又完全不一样，尽可能多地将推广型商品在首页进行展示；各层之间又用不同的色系进行类型划分，整体上风格统一，楼层区块却不拘一格。

- **广告插入点**（AD）。在适当的位置可以进行广告banner的插入。

- **用户帮助**（Member FAQ）。公共型模块内容，帮助用户熟悉站点的各项操作及购买流程等。放在下部位置，不是很突出，却又方便用户找到。

- **页面足部**（Footer）。站点信息、版权及声明等，全局型设计。

在首页这片“寸土寸金”的地方，要尽可能多地将促销、广告、商品等进行安置。什么？这样的首页太乱了，没有重点呀！嗯，确实，那么多的内容塞满一个页面，似乎重点并不突出，但这些却都是“卖点”。卖点实际就是商业销售中的重点，对于购买目标并不确定的用户，首页上精美的广告、实惠的促销价格才可能引发他们的购买欲或者说愿意点进去看

看；如果访客明确自己需要什么，也许这时已经在搜索关键字或点击分类菜单索引了。

再对这个首页页面做一个数据统计，如图9-13所示，各类广告数量可设置67个，推广商品数量可增加到153个，推荐品牌可设置40个，团购等活动4个以上。如此多的商品展示推广分布在区区几屏幕的范围内，注意这些可不是新闻网站上的文字链接，而是需要图文并茂地呈现给用户。而从首页到商品详情页，步骤应尽可能简化，不要大于三步为佳。

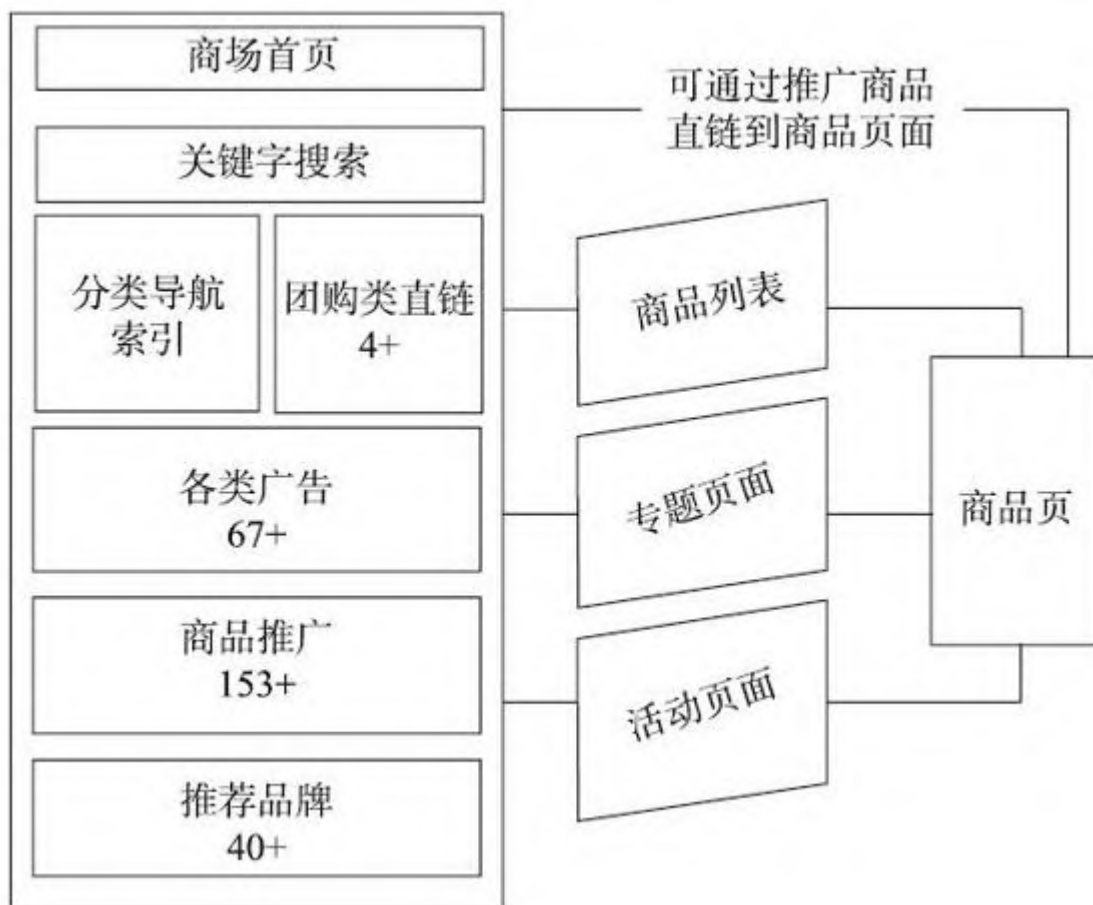


图9-13 首页商品广告数量统计对商品购买的影响

9.2.2 色彩风格、导航、顶部的统一化

所谓网站风格是指网站页面设计上的视觉元素组合在一起的整体形象，展现给人的直观感受。这个整体形象包括网站的配色、字体、页面布局、页面内容、交互性、广告、宣传语等因素。

我们所有网页上的颜色，在HTML下是以颜色英文单词或者十六进制的方法（如#000000表示为黑色）表示的。不同的颜色有着不同的含义，给人各种丰富的感觉和联想。

- 红色：代表热情、活泼、热闹、温暖、幸福、吉祥、奔放、喜悦、庄严；

- 橙色：代表光明、华丽、兴奋、甜蜜、快乐；

· 黄色：代表明朗、愉快、高贵、希望、富有、灿烂、活泼；

· 黑色：代表崇高、坚实、严肃、刚健、粗犷、夜晚、沉着；

· 白色：代表纯洁、纯真、朴素、神圣、明快、简单、洁净；

· 蓝色：代表深远、永恒、沉静、理智、诚实、寒冷、天空、清爽、科技；

· 绿色：代表新鲜、平静、和平、柔和、安逸、青春、植物、生命、生机；

· 灰色：代表忧郁、消极、谦虚、平凡、沉默、中庸、寂寞、庄重、沉稳；

- 紫色：代表优雅、高贵、魅力、自傲、浪漫、富贵；

- 棕色：代表大地、厚朴。

在首页的头部区域是最能体现网站风格的重点位置。商城的Logo、导航菜单和搜索条的色彩可以很明确让访客知道这个站点的主题风格色是什么，并且这个区域作为全局型设计，在其余的页面都会出现，因此页面中其他部分也应以此为基准进行色彩搭配使用。

由于商城系统首页、商品列表页、商品详情页等重要页面都会出现大量的商品图片，这些商品图片的风格色彩是不可控的并且有可能占据页面的主体内容。大多数情况下，商城的页面风格应该服务于所承载的商品信息，换句话说，我们要做的就是商场中的柜台、货架、银台、购物车等。明确了这一点，页面

的背景、字体、框线、按钮等，应尽量选择简洁、干净的形式。试想如果一个商城的页面元素：边框、按钮做得风格迥异，字体各式各样，色彩缤纷，那么到底在一个页面中谁才是重点？其实，有了页面头部的logo和导航等部分相关色彩的界定已经足够让用户了解这个商城的风格特征了。

在页面宽度的选择上，随着客户端浏览器分辨率的不断增加，网站的流行宽度也在慢慢地发生着改变。记得多年前，可能还存在760px这样的宽度标准，但现今980px、1000px这样的宽度已相当普遍，更有网站选择使用1200或98%、100%显示的形式。过宽的文字页面也许不利于阅读。据分析，页面文字阅读区域单行超过850px或70个汉字就会让读者产生厌倦情绪，但这里要设计的是一款商城程序界面，更多的是使用商品图、标题、价格等混排的小模块列表布局形式，因此，更高一些的分辨率能让设计师有空间做大图片

的尺寸，并能留白而不显得局促，因此，B2B2C商城的首页选择使用1200px分辨率。

在这个宽度下，页面主体部分按功能被分成了几个大的通栏区域（见图9-14），但仔细看不难发现，实际上每个区域中，都是保持着同样的左中右布局形式与块内宽度。这个格局也会被沿用到站点中的其他页面中，比如商品列表页、商品详情页、用户中心页面等，取得整站的统一性，而不会产生访客点击链接跳转后，页面主题及边栏宽窄不一的尴尬局面。

如图9-14所示，B2B2C商城首页及其他页面的比例关系非常清晰、明确，将页面竖列五等分，根据功能决定页面布局，布局块的宽度永远遵循1/5为基准的原则。实际上除了整体页面布局按照这个方式，块内列表、模块、图片也应在这个原则下展开，并适当留白保持空间感。

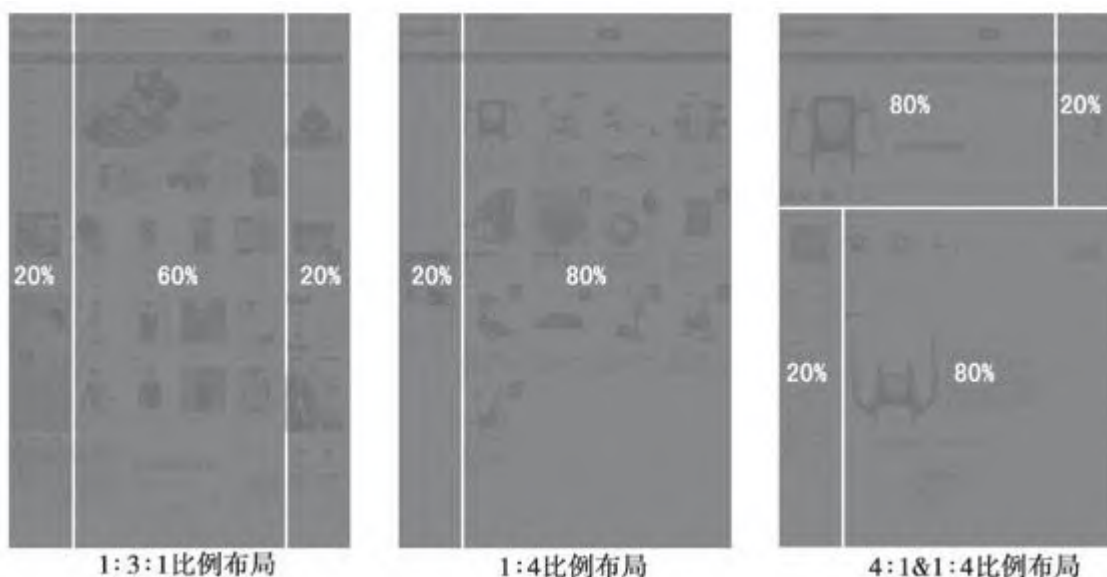


图9-14 首页与相关页面的比例与宽度的一致性

了解了颜色、布局对全站统一的作用，再来看看头部及导航对商城用户操作上的帮助。如9.2.1节所述，顶部条、头部、导航以及面包屑导航（所在位置）都是站点全局性的设计，将在商城各个页面中的最上方出现。这些内容让访客清楚，在什么网站，当前页面位置，在当前页面能进行哪些操作及有哪些重点，还有什么相关的操作需要进行等，这就好比是计算机系统任务栏、快捷方式和导航条所起的作用一样。

每一个链接文字实际都是一个下拉型菜单，将这类操作涉及的最重要的信息反馈给访问者，如图9-15所示，例如你在购物车中加入了几件商品，那里就会出现提示性的数字，点击后展开下拉菜单，将购物车内的商品、价格、数量以及该去完成的操作按钮显示出来；“我的商城”则会在用户登录后提示一些待处理的订单或消息数量，点击后可直达相关页面；搜索栏和分类导航则提供给访问者如何找到商品的途径。

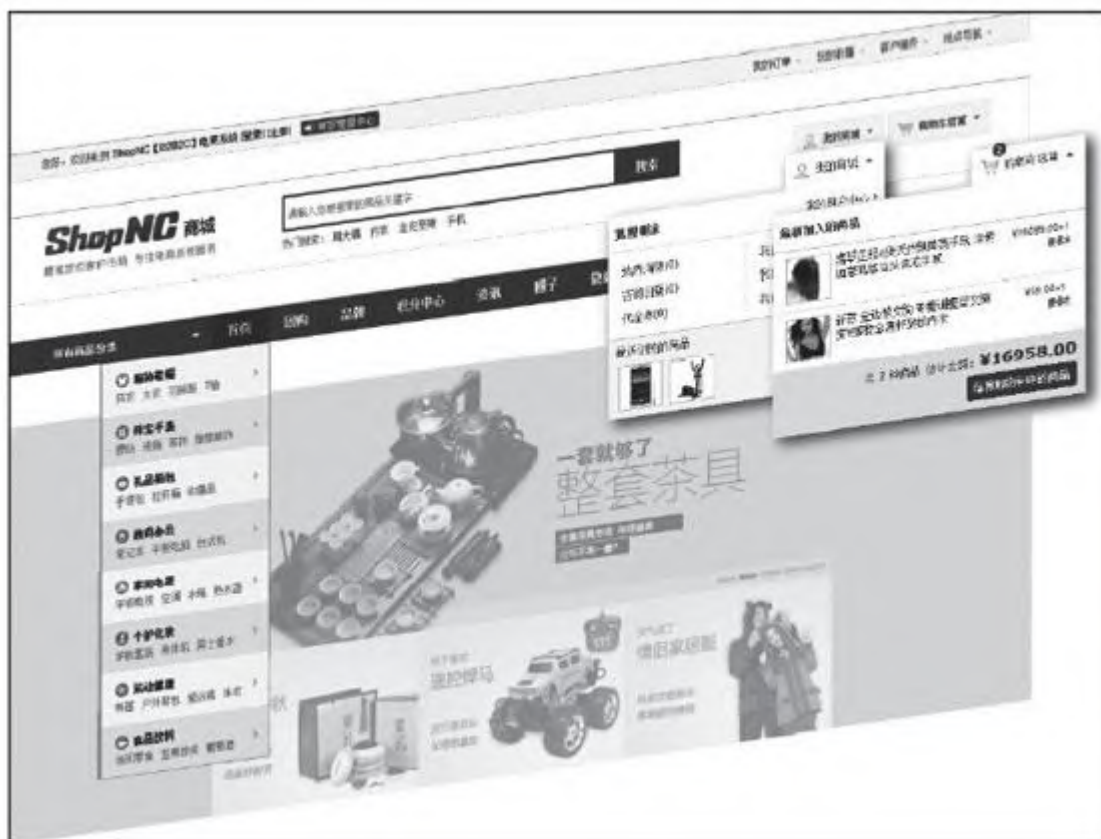


图9-15 商城头部结构分析

作为购物流程中必不可少的各类操作，应尽可能地放在最容易看到的地方，让用户可以连贯地完成一次购物体验。试想如果在商品详情页面中点击按钮将商品放入了购物车，却因为找不到进入购物车的相关按钮而让用户产生疑惑^[1]或者说用户虽然找到了购物车按钮，却忘记了里面曾经加过什么商品，买了多

少件，已经花了多少钱？而是还要跳转进入结算页面后才能看到，这样也就丧失了让用户继续购买的时机。我们日常生活中，在超市购物，把商品放入购物车，自己可以清晰地看到，购物完毕后才去银台结算，现实生活中的购物车不是一个“黑色的纸箱子”，商品放进去了就不知道是什么了，应该透明、清晰、直观地体现已加入购物车的商品。这些能提高用户体验的细节控制也是在设计页面时必须考虑到的问题。

[1] 购物车操作的形式不同于直接购买，不会立即跳转进入结算界面。

9.2.3 焦点广告区域的制作

焦点广告占据着首页第一屏重要的位置，将最重点的活动、商品进行切换或滚动展示，也是使整体风格搭配协调，个性鲜明的有效手段。这里，暂不对如何做好广告图的作图方法进行描述，着重说一下首页可切换通栏banner的页面代码实现部分，如代码清单9-15所示。

代码清单9-15 首页焦点区域满屏广告切换效果 代码示例

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>首页焦点区域满屏广告切换效果</title>
<style type="text/css">
/* 重新定义Html元素
----- */
html, body, div, ul, ol, li { padding: 0; margin: 0;}
ul, ol { list-style: none outside none;}
img { border: medium none;}
/* 首页焦点区域
----- */
```

```

.home-focus-layout { width: 100%; height: 481px; position:
relative; z-index: 1;}
/* 满屏背景切换焦点图 */
.full-screen-slides { width: 100%; height: 481px; position:
relative; z-index: 1;}
.full-screen-slides li { width: 100%; height: 100%;
position: absolute; z-index: 1; top: 0; left: 0;}
.full-screen-slides li a { text-indent: -9999px; display:
block; width: 776px; height: 300px; margin-left: -388px;
position: absolute; z-index: 2; left: 50%;}
/* 切换控制按钮 */
.full-screen-slides-pagination { font-size: 0; filter:
progid:DXImageTransform.Microsoft.gradient(enabled='true',
startColorstr='#33FFFFFF', endColorstr='#33FFFFFF');
background: rgba(255,255,255,0.2); *word-
spacing:-1px/*IE7*/; text-align: right; list-style: none;
display: block; width: 760px; height: 6px; padding: 7px 8px;
margin-left: -388px; position: absolute; z-index: 3; top:
280px; left: 50%; *zoom: 1/*IE7*/;}
.full-screen-slides-pagination li {
filter:progid:DXImageTransform.Microsoft.gradient(enabled='t
rue', startColorstr='#3F000000', endColorstr='#3F000000');
background: rgba(0,0,0,0.25); vertical-align: top; letter-
spacing: normal; word-spacing: normal; display: inline-
block; *display:inline/*IE6、7*/; width: 24px; height: 6px;
margin-left: 4px; list-style: none; overflow: hidden; *zoom:
1/*IE7*/; cursor: pointer;}
.full-screen-slides-pagination a { width: 100%; height:
100%; padding: 0; margin: 0; text-indent: -9999px;}
.full-screen-slides-pagination .current { background: #F30;}
</style>
</head>
<body>
<!-- 首页焦点区域 Begin-->
<div class="home-focus-layout">
  <!-- 自适应横向宽屏背景广告切换 B -->
  <ul id="fullScreenSlides" class="full-screen-slides">
    <li><a href="javascript:void(0);">&nbsp;</a></li>
    <li><a href="javascript:void(0);">&nbsp;</a></li>
    <li><a href="javascript:void(0);">&nbsp;</a></li>
    <li><a href="javascript:void(0);">&nbsp;</a></li>
    <li><a href="javascript:void(0);">&nbsp;</a></li>
  </ul>
</div>
<!-- 首页焦点区域 End-->

```

```

</body>
<script src="js/jquery.js"></script>
<script>
$(function(){
    var numpic = $('#fullScreenSlides li').size()-1;
    var nownow = 0;
    var inout = 0;
    var TT = 0;
    var SPEED = 5000;
    $('#fullScreenSlides
li').eq(0).siblings('li').css({'display':'none'});
    var ulstart = '<ul id="pagination" class="full-
screen-slides-pagination">',
        ulcontent = '',
        ulend = '</ul>';
    ADDLI();
    var pagination = $('#pagination li');
    var paginationwidth = $('#pagination').width();
    $('#pagination').css('margin-left',(372-
paginationwidth))
    pagination.eq(0).addClass('current')
    function ADDLI(){
        //var lilicount = numpic + 1;
        for(var i = 0; i <= numpic; i++){
            ulcontent += '<li>' + '<a href="#">'
+ (i+1) + '</a>' + '</li>';
        }
        $('#fullScreenSlides').after(ulstart +
ulcontent + ulend);
    }
    pagination.on('click',DOTCHANGE)
    function DOTCHANGE(){
        var changenow = $(this).index();
        $('#fullScreenSlides li').eq(nownow).css('z-
index','900');
        $('#fullScreenSlides
li').eq(changenow).css({'z-index':'800'}).show();

        pagination.eq(changenow).addClass('current').siblings('li').
removeClass('current');
        $('#fullScreenSlides
li').eq(nownow).fadeOut(400,function(){$('#fullScreenSlides
li').eq(changenow).fadeIn(500);});
        nownow = changenow;
    }
}

```



```

        pagination.mouseenter(function(){
            inout = 1;
        })
        pagination.mouseleave(function(){
            inout = 0;
        })
        function GOGO(){
            var NN = nownow+1;
            if( inout == 1 ){
                } else {
                    if(nownow < numpic){
                        $('#fullScreenSlides
li').eq(nownow).css('z-index','900');
                        $('#fullScreenSlides
li').eq(NN).css({'z-index':'800'}).show();

                    pagination.eq(NN).addClass('current').siblings('li').removeClass('current');
                        $('#fullScreenSlides
li').eq(nownow).fadeOut(400,function(){$('#fullScreenSlides
li').eq(NN).fadeIn(500);});
                        nownow += 1;
                    }else{
                        NN = 0;
                        $('#fullScreenSlides
li').eq(nownow).css('z-index','900');
                        $('#fullScreenSlides
li').eq(NN).stop(true,true).css({'z-index':'800'}).show();
                        $('#fullScreenSlides
li').eq(nownow).fadeOut(400,function(){$('#fullScreenSlides
li').eq(0).fadeIn(500);});

                    pagination.eq(NN).addClass('current').siblings('li').removeClass('current');
                        nownow=0;
                    }
                }
                TT = setTimeout(GOGO, SPEED);
            }
            TT = setTimeout(GOGO, SPEED);
        })
    </script>
</html>

```

分析上述代码内容，主体部分很简单，即对“`.home-focus-layout`”块中的`ul li`内容进行切换，注意，这里是以`li`背景填充形式出现的图片，通过JS实现淡出淡入的轮换播放，并且产生`id=“pagination”`的控制条，可手动指定切换到想要的图片。CSS部分设定这个全屏广告区域宽度为100%，高度为481像素，而实际链接区域<A>则是固定了宽和高的一个小区域，之所以这样做是为了配合首页特殊的布局形式，给焦点广告区域的左侧商品分类导航与右侧商城公告等内容让出位置来，而整体上又作为大区块背景出现。

这部分焦点广告也可以通过平台后台进行设定，只需上传符合需求的广告图片和设置好配合这张图的背景色保存即可。程序会控制将模板中“`.full-screen-slides li`”的`bakcground`的相应属性来插入设定值。

9.2.4 内容块的制作与可视化编辑理念

除去本节提到的首页头部及焦点部分，首页大篇幅的内容为楼层形式的版块区域，它包含了标题、广告、商品、热门分类、推荐品牌等商城需要在首页推广的相关信息。考虑到一个商城在实际运营中，首页的内容经常要进行调整、更新、版块维护，这就需要程序员及网页美工等协作完成，这势必影响效率增加维护成本。那么能不能让首页变得清新、便捷、维护简单，稍加专业培训的文员都可以进行设置并推荐商品和更新广告呢？

作为一款成熟的商城产品，用户体验度不应简单地停留在访问层面上，而是要渗透到操作层、管理层等更深的范围。平台可视化编辑版块区域内容的设计理念也就应运而生。

内容块可视化编辑应满足以下几个设计重点。

- 可开启关闭楼层模块，并可调节楼层排序；
- 层版块可设定不同的颜色风格；
- 版块标题可设定为图片标题或文字化标题；
- 推荐分类、推荐商品、推荐品牌等内容均可手动选择；
- 推荐商品与广告区域可增加标签切换；
- 具有轮播形式的广告。

带着这些思路与设计重点，让我们具体来看看如何实现前后台对应可视化完成首页模块内容编辑。如图9-16所示，首页版块设定部分满足了设计重点中的前两点需要，通过选择不同的预设值色彩，给版块附加对应的class值。例如，版块被设定为红色系的，则

模板页面增加版块，对应语句是

* 版块名称:

红色

* 色彩风格:

☒

☐

☐

☐

☐

☐

☐

☐

红色 粉色 橘色 绿色 蓝色 紫色 褐色 灰色

* 排序:

1

显示:

是

否

提交

see more please visit: <https://homeofbook.com>

图9-16 首页模块设定

有了初期的设定，还需要对版块内容进行编辑管理。如何能让使用者在最短的时间内上手操作呢？最有效的方法就是“所见即所得”形式。把管理的界面做得和生成后的界面基本统一，只是多了“编辑”等按钮以进行框体参数设定，只要点几下鼠标，就能把一个版块的内容更新好，而且不需要生成首页预览，直接在管理界面就能看到比例缩放后的效果。可视化编辑的原理是将一个楼层版块又划分成6~7个小的模块，每个模块都可以进行独立的编辑，然后存入数据库，版块从数据库读取后调入布局中，如果满意后再点击生成版块，则这一楼层的内容就保存起来了。反反复复几次这样的操作，色彩不同、内容不同、多广告、多商品的首页楼层版块即可形成。图9-17为首页模块可视化编辑示意图。



图9-17 首页模块可视化编辑示意图

通过上述几小节的描述，大家应该清楚了一个B2B2C类型的商城首页的结构特征，功能类型以及对整站起到什么样的作用，如何提升其实用性与易操作性等。

9.3 商品详情页面

9.3.1 商品页面布局方式

一个商城的前台，主要以三种功能类型的页面构成：第一是聚合类，即商城的首页、频道页面、主题页面等；第二是索引类，即商品列表、店铺列表、品牌列表等，而它们最终都将访问者引向第三种展示类，即商品详情页面，也可以说它是最终商品销售的重要环节。

商品详情页面是商城及店铺直接面向访问者的界面，是其最终决定是否购买该商品，形成一笔交易的唯一途径。那么，让我们从消费者的角度出发，分析一下有哪些因素会促成一次成功的商品销售，如图9-18所示。



图9-18 商品详情页中各种元素对访客的影响

从入口页面进入商品详情页面后，首先要有明确的标题、价格、图片、运费、规格等与商品直接相关的内容，这就和身份证件一样，是每个商品所必需的；其次，一段详尽、精彩的商品描述也是必不可少的内容之一。当上面的条件都具备了，这时候，作为消费群体，他们会看看这款商品的销量如何、口碑如何、店铺的服务如何等侧向辅助信息，来增强对该商品的认知度，网络购物不同于实体销售，因为并不是

实物购买，所以很多情况下购买者缺少信心，始终在寻找别人的帮助，销售记录列表、已购者评价与晒图等无疑是最好的“旁敲侧击”手段。突出显示的“立即购买”和“加入购物车”按钮可以让访客轻松完成接下来要做的事儿，即使这时并不想购买也能加入收藏或分享商品给别人，这又为该商品提供了更多的销售机会。另外，店铺商品推荐、店铺客服等也是必不可少的元素，可以主动推荐店内其他商品给消费者，获得更大的销量。一个页面中要体现如此之多的内容，似乎每一个环节都是那么的重要，如何在一个页面中排版呢？

如图9-19所示，商品详情页面中的头部、足部与商城其他页面同样使用全局化设计，为保持B2B2C商城的统一性，包含顶部、导航与所在位置、帮助、版权等内容；第一屏上半部分为商品信息主体，以及前面所提到的商品名称、价格、运费、规格等，还有具有

放大镜效果的多图展示特效，无论是什么类型的商品，这些内容都是必不可少的，在商品发布时也是必选项；该页右侧为店铺相关信息、动态评分、联系方式以及进入、收藏店铺等操作提示或链接；其下方为“立即购买”、“加入购物车”两个最突出的按钮以及收藏、分享商品的操作链接。说到这里不得不提一下，“立即购买”与“加入购物车”有必要同时存在吗？笔者认为这是两种截然不同的操作体验，点击“立即购买”按钮后，直接进入购物车进行结算，选择“加入购物车”则并不离开当前页面，访客还可以浏览其他商品并再次添加到购物车，最后一起进行结算，这样可以满足A、B两种购买意愿：

- A类：购买意愿非常强烈，可以马上结账付款，给人消费的冲动感，适合购物目的明确的人群需要；

· B类：购买意愿比较强烈，等待付款，但喜欢一次性购买多宗商品，寻求更多搭配或优惠的人群需要。

其实，收藏和分享也可以被划分到具有潜在购买意愿的群体中，满足有兴趣但暂时不需要购买，想做个标记以后再说或者分享给别人看看的群体。

商品主体信息部分的下方被分为左右两个区域，左侧的侧边栏用来显示该商品的二维码（方便手机用户扫描网址并分享）、店铺客服列表（结合在线IM功能使用）、店内搜索、店铺销售排行、店铺分类等这些店铺级的辅助信息，让访问者能更好地了解该店铺还有什么商品，什么是最畅销的商品，从而增加同店多款商品购买的可能性。

右侧依次为商品详情、商品评价、购买记录和购买咨询。正常情况下平铺展开，随向下浏览次序排

列，同时也具有Tab切换功能，点击某内容的Tab标签按钮，则可将这部分的内容直接显示到顶端位置，方便访客快速了解想知道的信息。商品详情为店铺自行编辑的内容，这里不做过多阐述，但好的商品介绍如同一个精彩的故事一般，能很好地吸引消费者，越是详尽的商品介绍对产品的销售越有帮助。商品销量、评价、咨询这些都是很有效的推动销售的方式，增加购买者对于该商品另一个侧面的了解，是不可或缺的内容。

最下方为店铺推荐商品，根据店铺自行推荐的商品随机显示。

对应图9-15的内容来看，该页面将所有必须的内容都合理安排进去，丰富而不杂乱，并且商城所有页面保持统一，使访客迅速形成固定的浏览习惯，想看什么内容到什么地方去找，变得很容易。

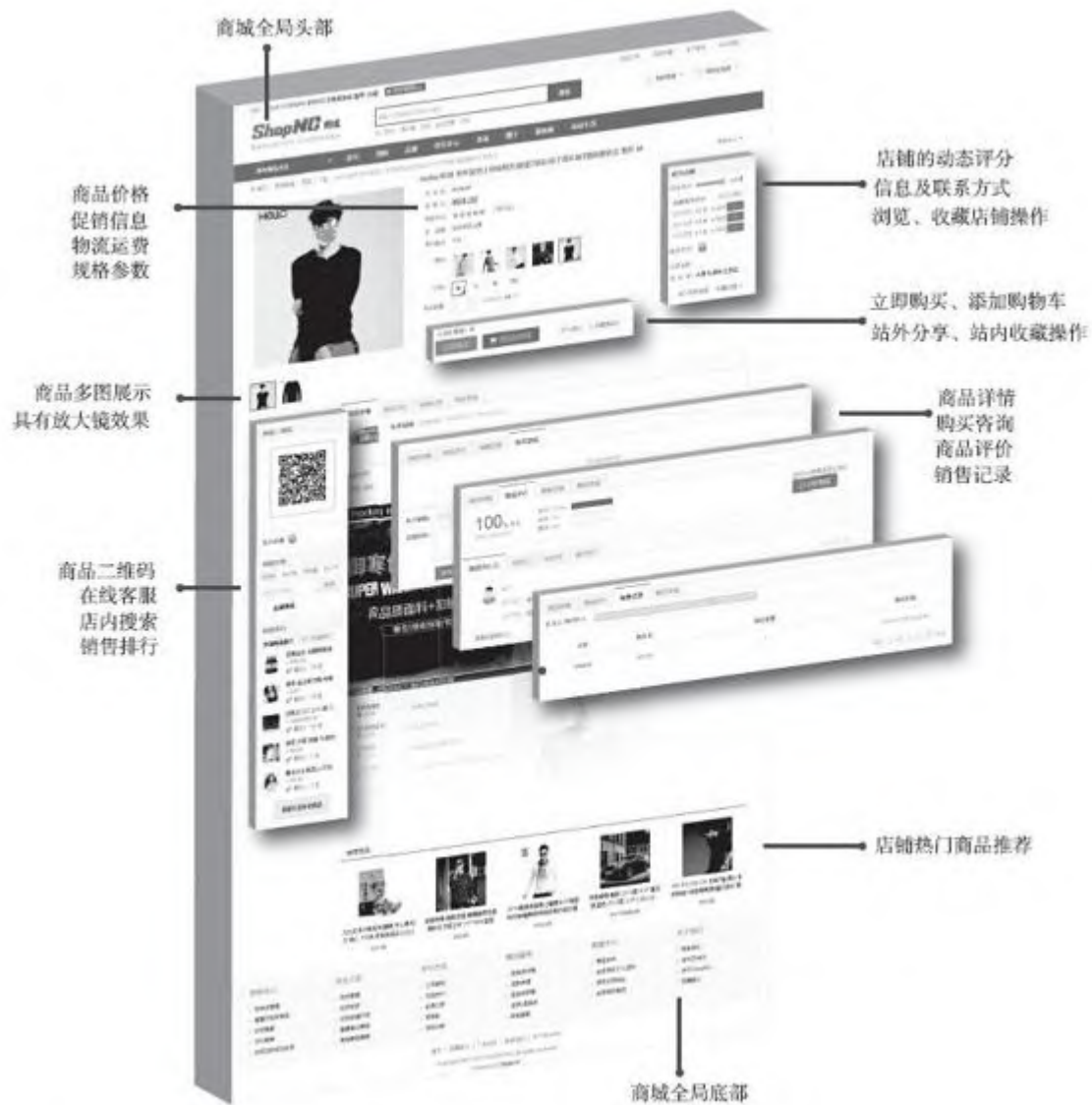


图9-19 商品详情页内容元素布局

9.3.2 针对搜索引擎的页面优化制作

SEO（搜索引擎优化）始终是商城运营中被重视的部分，商品详情页面是商城前台页面中最常出现的，那么在它的优化制作上有什么技巧，应该注意哪些问题呢？

HTML标签头title（标题）内容，是最重要的页面关键元素，被设定为商品的名称，被搜索引擎收录后，如果搜索内容与商品标题相符，可直接对该页面访问。Keywords（关键字）及Description（页面描述）由平台管理设置，为商城级的SEO优化，如图9-20所示。

生成后的页面源代码可参考代码清单9-16。



图9-20 平台对商品页面HTML标签头的设定

代码清单9-16 商品详情页面HTML标签头部分

```
<!doctype html>
<html lang="zh">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
<title>Hello/哈奴 男冬装男士休闲纯色加绒打底衫扣子领长袖T恤保暖秋衣
黑色 M - ShopNC B2B2C电商系统</title>
<meta name="keywords" content="ShopNC,ShopNC B2B2C电商系统,"
/>
<meta name="description" content="ShopNC,ShopNC B2B2C电商系
统," />
<meta name="author" content="ShopNC">
<meta name="copyright" content="ShopNC Inc. All Rights
Reserved">
.....
```

商品详情中的商品标题和广告语也应使用<h1>这类语义化标签定义，作为该页面中最重要的内容标题来。在突出显示的价格等位置尽量使用（加粗）或者（重）标签来制作页面，如代码清单9-17所示。

代码清单9-17 商品详情页面的商品信息代码截取

```
<!-- S 商品基本信息 -->
<div class="ncs-goods-summary">
  <div class="name">
    <!-- S商品标题 -->
    <h1>Hello/哈奴 男冬装男士休闲纯色加绒打底衫扣子领长袖T恤保暖秋衣
    黑色 M</h1>
    <!-- S商品广告短语 -->
    <strong>季节性促销优惠</strong> </div>
  <div class="ncs-meta">
    <!-- S 商品参考价格 -->
    <dl>
      <dt>市&nbsp;场&nbsp;价: </dt>
      <dd class="cost-price"><strong>&yen;179.00</strong>
    </dd>
    </dl>
    <!-- S 商品发布价格 -->
    <dl>
      <dt>商&nbsp;城&nbsp;价: </dt>
      <dd class="price"><strong>&yen;69.00</strong> </dd>
    </dl>
    .....
  </div>
</div>
```

合理运用Alt标签为图片添加对应名称，为链接<a>标记添加title的内容也对优化有一定的好处，如代码清单9-18所示。

代码清单9-18 商品详情页面规格值代码截取： Alt及title的使用

```
<!-- S 商品规格值-->
<dl nctype="nc-spec">
  <dt>颜色: </dt>
  <dd>
    <ul nctyle="ul_sign">
      <!-- 图片类型规格-->
      <li class="sp-img"><a href="javascript:void(0);"
class="" data-
                        param="{valid:413}"
title="灰色"><i></i></a></li>
      <!-- 图片类型规格-->
      <li class="sp-img"><a href="javascript:void(0);"
class="" data-param="{valid:414}"          title="深灰色"><i></i></a></li>
      .....
    </ul>
  </dd>
</dl>
<dl nctype="nc-spec">
  <dt>尺码: </dt>
  <dd>
    <ul nctyle="ul_sign">
```

```
<!-- 文字类型规格-->
<li class="sp-txt"><a href="javascript:void(0)"
class="hovered" data-param="{valid:408}">M<i></i></a></li>
.....
</ul>
</dd>
</dl>
<!-- E 商品规格值-->
```

此外代码内部注释、站内链接跳转、永久性导航、语义化标签的合理使用都能起到一定意义上的作用。

9.4 购物车页面

9.4.1 购物车页面设计特征

B2B2C商城系统中的购物车就像平时生活中在超市购物时使用的购物车一样，它用来随意选购添加或删除商品，最后根据车中的商品统一结算下单。前面的章节介绍过购物车功能在程序设计时的结构、设计思路与代码分析，本节主要介绍其前台页面展示部分的特征与细节内容分析。

首先，让我们来看看购物车实际操作中的流程都有哪些，如图9-21所示。

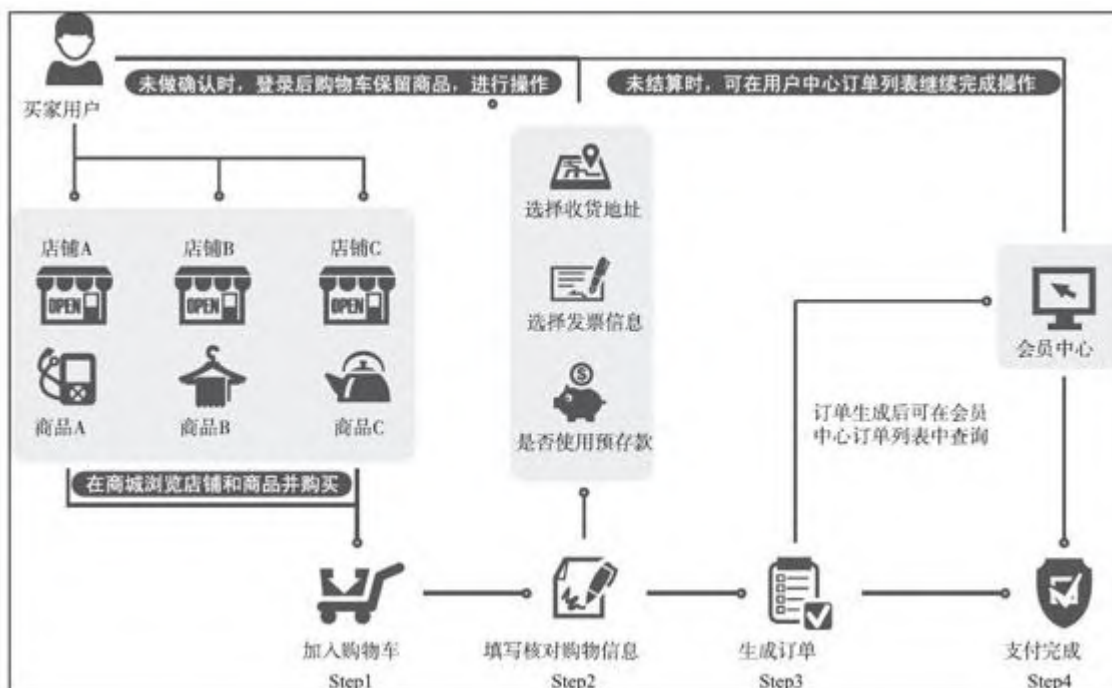


图9-21 购物车流程分析图

从图9-21中我们可以了解到，B2B2C商城的设计思路是将多店铺、多商品加入购物车，合并进行结算，步骤大体上可以分为四步完成。

步骤一，查看我的购物车；

步骤二，填写核对购物信息；

步骤三，提交生成订单；

步骤四，支付订单完成。

在这几个步骤中，有可能会会员没有进行连贯性的操作，比如说：购物后没有立即进行购物信息的填写，生成订单后没有立即进行支付等等情况，系统都应考虑到，并使其再次完成该笔结算单时能准确的切入操作。

在购物车的设计中还应考虑到会员可以在订单未提交前，进行删减取消商品购买的情况；收货地址与货到付款商品之间的关联，各种价格、促销、店铺优惠、运费的明确表述等，让购买者对购物车的操作可以一目了然并放心使用。

图9-22展示的是ShopNC商城购物车——我的购物车实际页面情况，头部与其他页面有所区别，以突出的步骤样式指引会员完成购物及结算流程。第二部分则是明显的文字标题显示会员正在操作什么内容，都

能做哪些操作。第三部分是已经加入购物车的商品列表情况，这里可能存在多家店铺的多件商品，单格、数量与店铺合计、商品总价都有所差别，另外还可以对商品的数量进行增减操作，以及对店铺级别的促销活动进行表述，等等。最下方的右侧则是一个很突出的按钮，目的是引导客户进行下一步操作。



图9-22 购物车——我的购物车

在填写核对购物信息，也就是第二步中，应出现收货人信息、支付方式、发票信息及商品清单四块内

容，每一个功能块都可以在本页内进行编辑操作，即使初次购物的用户也不必再跳转到会员中心等其他页面填写诸多内容而打断一次购物支付步骤，如图9-23所示。

填写核对购物信息
请仔细核对填写收货、发货等信息，以确保物流快速及时准确配送。

收货人信息

☐ 同城天创 天津 天津市 红桥区 大丰源(水游城西侧)大源顺 013188888888 [删除]

☒ 使用新地址

* 收货人姓名:

* 所在地区:

* 详细地址:
请填写真实地址，不需要重复填写所在地区

* 手机号码: (或)固定电话:

图9-23 购物车——收货人信息

会员可以选择已有的地址作为收货地址或者添加新的地址到收货人信息列表中，保存后生效，这样就大大提高了购物整体的流畅性，并且收货地址的选择也与支付方式中货到付款类商品订单有直接的联系，

后面的操作会根据地址进行支付方式判断，避免了误操作等。

图9-24展示的是商品清单的内容，通过与步骤一这部分内容比对可以看出区别，没有删除或增减的操作性按钮链接了，因为步骤一是对订单的确认、步骤二则是信息的填写，在正式提交前，还可以通过返回购物车进行商品数量上的修改。我们可以这样来看步骤一、二之间的关系，前者还在你自己的购物车里，而后者则已经要放到结账柜台上了，所以在设计时应注意他们之间的特征与区别。商品清单是明示所购商品数量、价格、运费等等相关信息的部分，所以价格要突出并在色彩上有所区分，如使用预存款进行支付（会员预存于商城的金额），应设置手机验证或密码验证等安全保护程序。步骤二似乎看起来需要会员完成很多项内容的填写，实际如果使用默认的已有信

息，则可直接点击“提交订单”按钮进行下一步操作。



图9-24 购物车——商品清单

图9-25为订单提交待支付步骤，订单一旦提交，之前的商品数量以及购物信息都不能再做修改了，于是在多店铺中进行了多商品的购买，系统会自动拆

成两个订单号，方便会员在已购商品列表中进行后续步骤的操作。

支付提交

订单详情内容可通过查看我的订单进行核对处理。

请您及时付款，以便订单尽快处理！在线支付金额：¥21887.00

订单号	支付方式	金额	状态
由于您的商品由不同商家发出，此单将分为2个不同子订单配送！			
6309000000046802	在线支付	¥4983.00	快通
0309000000046801	在线支付	¥15899.90	快通

支付选择

支付宝

货到付款

网银在线
ChinaBank Payment

确认提交支付

图9-25 购物车——订单支付

即便这时候会员没有对订单进行及时的支付，还是可以从会员中心已购商品列表中找到这笔待支付的订单回到购物车支付页面继续完成订单，如图9-26所示。

订单完成

订单已支付完成，祝您购物愉快。

✓

订单支付成功！您已成功支付订单金额¥21887.00。

可通过用户中心已买到的商品查看订单状态。

继续购物

查看订单

图9-26 购物车——订单支付完成

这些购物车的步骤内容页面都是根据图9-18的流程图思路设计完成的，我们在页面的布局、内容结构部分都应在制作前提前进行策划并与程序部分进行沟通，进行团队小组会议，将流程步骤图分析并探讨，哪些问题会在什么时候出现而影响正常的购物车订单提交与结算的操作合理性，最终的目的只有一个，就是为了让会员购物能尽可能的简单并流畅，获得更高的用户体验度。

9.4.2 根据页面需要选择HTML标签的技巧

上一节对B2B2C商城系统的购物车步骤流程设计思路进行了详细的分析，以其中最为复杂的步骤二——“填写核对购物信息”页面为例，具体谈一谈在一个页面中根据不同的内容使用不同的HTML标签的技巧。

当一个页面中既存在公用型样式又存在独立型样式时，我们应该尽量的合理使用HTML配合CSS文件使页面整体效果统一又符合代码规范。

填写核对购物信息

请仔细核对填写收货、发票等信息，以确保物流快速及准确投递。

收货人信息

修改

网城天创 天津天津市红桥区大丰路(水游城)锦大厦8层 01318886888

支付方式

修改

在线支付

发票信息

修改

普通发票 个人物品

商品清单

返回购物车

商品	单价(元)	数量	小计
官方店清			
 HelloKitty 秋冬装男士休闲时尚加绒卫衣衫扣子带长袖T恤保暖秋衣 黑色 XL	¥69.00	1	¥69.00
买家留言：			
			运费：¥0.00
			商品金额：¥69.00
			本店合计：¥69.00
☒ 使用支付宝支付（当前可用余额：¥98106243.99）			
			登录密码： 支付
订单总金额：¥ 69.00 元			

图9-27 购物车——填写核对信息步骤整体效果

如图9-27所示，不难发现，收货人信息、支付方式、发票信息、商品清单部分的标题内容及描述类文字都很统一，使用了同样的样式，如代码清单9-19所示。

代码清单9-19 购物车页面HTML内容截取

```

<!-- S 收件人信息 -->
<div class="ncc-receipt-info">
  <div class="ncc-receipt-info-title">
    <h3>收货人信息</h3>
    <a id="edit_reciver" nc_type="buy_edit"
href="javascript:void(0)">[修改]</a>      </div>
    <div id="addr_list" class="ncc-candidate-items">
      <ul>
        <li> <span class="true-name">网城天创</span> <span
class="address">天津 天津市 红桥区大丰路(水游城)冠錡大厦8层</span>
<span class="phone"> <i class="icon-
mobile-phone"></i> 13188888888 </span> </li>
      </ul>
    </div>
    .....
  </div>
<!-- E收件人信息-->
<!-- S支付方式-->
<div id="paymentCon" class="ncc-receipt-info">
  <div class="ncc-receipt-info-title">
    <h3>支付方式</h3>
    <a id="edit_payment" nc_type="buy_edit"
href="javascript:void(0)">[修改]</a>
  </div>
  <div class="ncc-candidate-items">
    <ul>
      <li>在线支付</li>
    </ul>
  </div>
  .....
</div>
<!-- E支付方式-->
<!--S发票信息-->
<div class="ncc-receipt-info">
  <div class="ncc-receipt-info-title">
    <h3>发票信息</h3>
    <a id="edit_invoice" nc_type="buy_edit"
href="javascript:void(0)">[修改]</a> </div>
    <div id="invoice_list" class="ncc-candidate-items">
      <ul>
        <li>普通发票 个人 饰品</li>
      </ul>
    </div>
    .....
  </div>

```



```
<!-- E发票信息-->
<!-- S商品清单-->
.....
<!-- S商品清单-->
```

每一个信息块是以class=“ncc-receipt-info”的DIV来布局，其内部又都以class=“ncc-receipt-info-title”和class=“ncc-candidate-items”两个DIV分成标题与内容两个区域，内容区域又都使用了ul、li的列表形式，而css部分只需要定义一次对应的class内容就可以了，因此这些内容在表现形式上取得了统一性。

那么独立性又体现在什么地方呢？每个内容块的标题后面都有“编辑”的链接文字，点击后内容区域会变成相关的表单提交形式，而这些提交表单的内容与正常信息显示的列表内容截然不同。例如图9-23所示的提交收件人信息部分的表单就使用了dl、dt、dd的标签来控制，如代码清单9-20所示。

代码清单9-20 购物车页面-收件人信息新增表单

页面代码截取

```
<div id="addr_list" class="ncc-candidate-items">
  <ul>
    <li class="receive_add address_item">.....</li>
    <li class="receive_add addr_item">
      <div id="add_addr_box">
        <div class="ncc-form-default">
          <form id="addr_form" action="index.php"
method="POST">
            <input type="hidden" name="act" value="buy">
            <input type="hidden" name="op" value="add_addr">
            <input type="hidden" value="ok"
name="form_submit">
            <dl>
              <dt> <i class="required">*</i> 收货人姓名: </dt>
              <dd>
                <input id="true_name" class="text w100"
type="text" value="" maxlength="20" name="true_name">
              </dd>
            </dl>
            <dl>
              <dt> <i class="required">*</i> 所在地区: </dt>
              <dd>
                <div id="region">
                  <select class="w110">
                    <option>-请选择-</option>
                    <option value="1">北京</option>
                    <option value="2">天津</option>
                    <option value="3">.....</option>
                  </select>
                  <input id="city_id" type="hidden"
name="city_id" value="">
                  <input id="area_id" class="area_ids"
type="hidden" name="area_id">
                  <input id="area_info" class="area_names"
type="hidden" name="area_info">
                </div>
              </dd>
            </dl>
          </form>
        </div>
      </li>
    </ul>
  </div>
```

```
        </dl>
        .....
    </form>
</div>
</div>
</ul>
</div>
```

在shopnc商城系统中的表单类型HTML标签选择上，都是使用的这类格式，dt用来定位输入项名称，dd用来定位input内容、验证、提示信息等，所以这里依旧符合其余页面的规则，但又区别于本页原有块内容形式。

又如图9-24所示，商品清单这里结构规则，更适合使用table来制作，因此这里的html标签我们选择使用table、tbody、td、tr等，如代码清单9-21所示。

代码清单9-21 购物车页面-商品清单部分表格布局代码截取

```
<table class="ncc-table-style">
  <thead>
    <tr>
      <th class="w20"></th>
```

```

        <th></th>
        <th>商品</th>
        <th class="w120">单价(元)</th>
        <th class="w120">数量</th>
        <th class="w120">小计</th>
    </tr>
</thead>
<tbody>
    <tr>
        <th colspan="20"> <i class="icon-home"></i> <a
href="http://www.shopnctest.com/b2b2c/2014/demo/shop/shop-
1.html" target="_blank">官方店铺</a>
        <div class="store-sale"> </div>
        </th>
    </tr>
    <tr id="cart_item_612" class="shop-list ">
        <td><input type="hidden" name="cart_id[]"
value="612|1"></td>
        <td class="w60"><a class="ncc-goods-thumb"
target="_blank"
href="http://www.shopnctest.com/b2b2c/2014/demo/shop/item-
320.html">  </a></td>
        <td class="t1"><dl class="ncc-goods-info">
            <dt> <a target="_blank"
href="http://www.shopnctest.com/b2b2c/2014/demo/shop/item-
320.html">Hello/哈奴 男冬装男士休闲纯色加绒打底衫扣子领长袖T恤保暖秋
衣 黑色 XL</a> </dt>
            <dd> </dd>
        </dl></td>
        <td class="w120"> ¥ <em>69.00</em></td>
        <td class="w60">1</td>
        <td class="w120"> ¥ <em id="item612_subtotal"
nc_type="eachGoodsTotal">69.00</em></td>
    </tr>
    <tr>
        <td class="w10"></td>
        <td class="t1" colspan="2"> 买家留言:
            <input class="text w340" type="text" maxlength="150"
name="pay_message[1]" value=""></td>
        <td class="t1" colspan="10"><div class="ncc-form-
default"> </div></td>
    </tr>

```

```
|  |
| --- |
| 运费:  ¥  商品金额:  ¥  本店合计:  ¥ |
| ☐ 使用预存款支付（当前可用余额:  ¥98106243.99  登录密码:    >使用 |
| 订单总金额:  ¥  元 | |

```

```
</tfoot>  
</table>
```

table类标签更适合制作规整的多项列表、表格内容，但由于表内内容各异，根据具体情况使用其余HTML标签，并在价格等需要突出显示的位置进行合理的控制，如使用标签并定义特殊的class控制色彩等方法。

通过B2B2C商城购物车页面的实例分析，说明一个页面的制作需要考虑页面的内容都有哪些，元素之间有什么共同点，又有什么不同点，然后再根据所需选择合适的标签，能重复使用的就不再独立创建，如需新建样式也应选择符合站点其他页面的HTML标签使用规则。因为我们分析的代码内容都是页面生成后的效果，而真实模板并非如此，夹杂了语言包与PHP标签、循环等内容的页面，合理地选择标签能让代码条例更

清晰，在嵌套循环的时候更简单、日后的维护工作也更容易一些。

9.5 订单页面

9.5.1 购物详单设计理念

B2B2C商城系统中的购物详单也称为“订单详情”，是在买家中心的订单列表页面中用来查询具体某一笔订单时显示的页面。会员可以通过对订单详情的查看，了解这笔订单的一切相关信息，包括成交时间、金额、数量、收货信息以及商家的发货状态、物流单号等，可以说它是一张电子消费清单。

随着电子商城不断发展，日趋成熟，功能越做越细，信息量越来越大，一些页面模板就需要更新以适应需求。如图9-28所示，这是商城系统原有使用中的订单详情模板，一些信息内容都有所体现，但只是平铺内容，按照标题将所有要展示的信息摆在那里，整

体看上去虽然没有什么过多的问题，但也似乎缺少重点，如在页面中找到想要的信息，不熟悉的用户则需要逐一内容去看。另外页面中缺少步骤流程性的表现，也没有操作提示类的帮助，仅仅算是一个标准的订单信息展示页。

ShopNC 商城

买家首页个人主页好友站内信1设置

订单详情

订单编号: 6000000000046702 下单时间: 2014-04-14 17:20:11

卖家信息

店铺名: 我们只卖手机 电话号码: 所在地区: QQ: 12345678 旺旺: 112313123

订单信息

	商品名称	单价	数量	商品总价
	苹果 (APPLE) iPhone 5s 16G 3G手机 (白色) WCDMA/GSM	4988.00	1	4988.00

订单总价: ¥4988.00 (含运费)

支付方式: 支付宝

下单时间: 2014-04-14 17:20:11

付款时间: 2014-04-14 17:20:11

付款单号: 750450811211171025

物流信息

收货人: 同成天创 13188888888,022-88888888 天津 天津市红桥区 大港路体育城诚通大厦808

发票信息

操作历史

买家 于 2014-04-14 17:20:11 提交了订单

卖家 于 2014-04-14 17:20:11 完成了付款

首页 | 招贤纳士 | 广告合作 | 联系我们 | 关于ShopNC

Copyright 2007-2014 ShopNC Inc. All rights reserved.

Powered by ShopNC

图9-28 原订单详情页面

为了解决以上提及的不足之处，使订单详情页更加条理清晰、步骤明确、信息丰富且又不拖泥带水，我们重新设计了这个页面，如图9-29所示。



图9-29 新订单详情页面

重新设计过的订单详情页面看上去要比要有的内容少很多，其实是将更多的附属信息以鼠标触及显示的形式做成了隐藏块，增加了中间部分的交易流程步骤以及右侧上方的订单状态与操作提示内容，如果把全部内容有所体现，信息量非常丰富，如图9-30所示。

通过立体结构图我们看到，这个订单详情页面其实隐含了大量的相关信息，仅仅通过这个页面用户就能对一笔订单的所有内容进行查询并根据提示完成操作，步骤及逻辑性加以优化。将重要的信息直接显示、更多内容隐藏显示，待处理的事项进行文字提示帮助并突出可操作按钮，即使初次使用商城系统的用户，通过直观的流程步骤图也能知道当前商品交易正处于什么状态。这些改进比先前那种平铺式的布局要清晰直观很多，而页面的篇幅却要远远小于过去。美

观、简洁、指向性强，作为订单详情页面改造的目标在此有所体现。

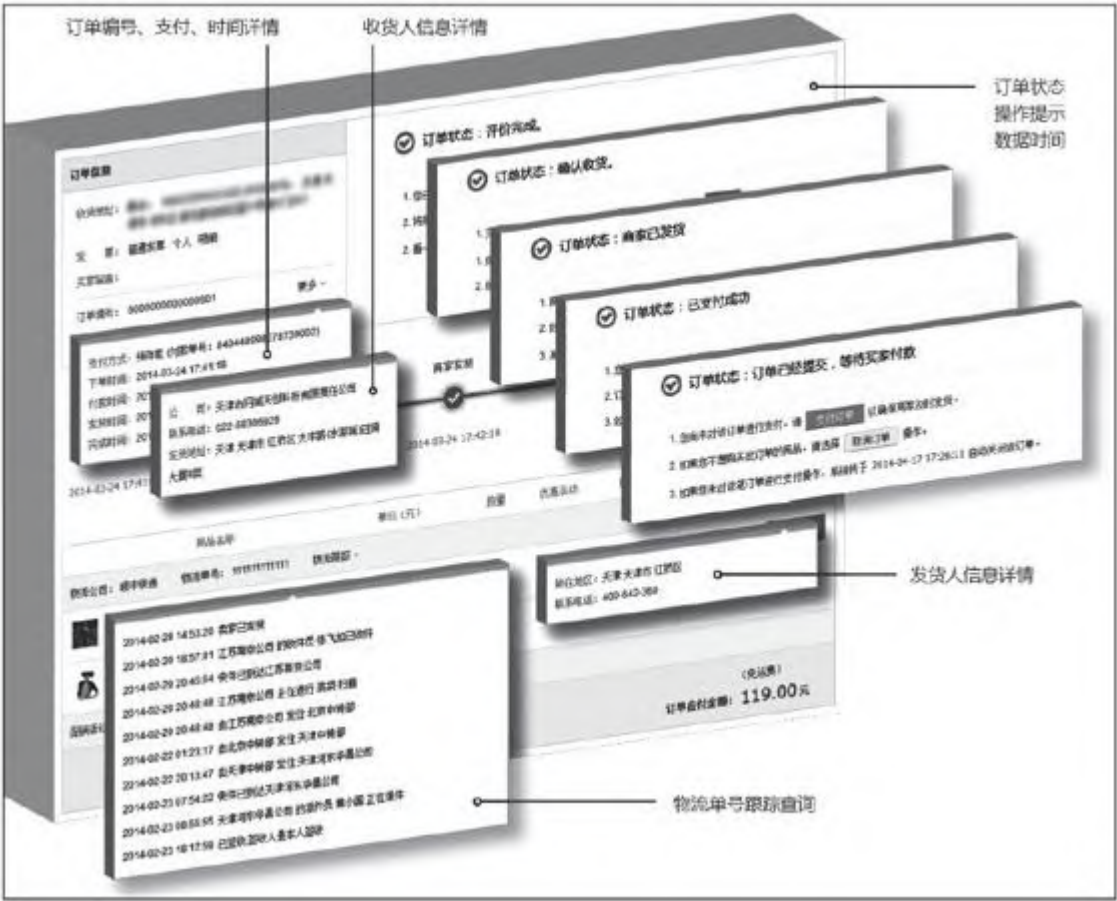


图9-30 订单详情页面结构分析

9.5.2 订单中要有哪些内容

一个订单中需要有哪些内容，首先要了解交易的业务流程关系。前面的章节我们介绍过用户购买商品加入购物车以及结算订单的步骤流程，这仅仅是整体交易流程的前半部分，当订单提交并付款后，才真正进入了交易流程。



图9-31 订单处理整体流程图

图9-31中，对订单整体流程做了形象化的图示，其中从“生成订单”到“交易评价”之间，五步实线相连的内容为一个正常的成功订单主线，当然，也可能会有取消订单、退款退货等非成功订单问题的存在，图中虚线位置为这些可能性的简单描述，具体流程请参考第三章相关章节内容。

因此我们在订单详情页面应该明确地将这五个步骤告知买家，并在相应位置提示如遇非正常流程应如何处理及相关链接按钮，这样买家就很清楚目前交易的进度以及自己能做什么、该做什么。这些对于一次成功的交易至关重要，也可大大提升用户体验度。例如图9-32所示，这是订单交易的第二步订单提交待支付状态下的提示，买家能够通过这些文字知道自己应该及时地对订单进行支付，最后期限是什么时候，如果不想购买了应该如何操作，等等。



图9-32 订单详情待支付订单操作提示信息

除了明确的交易状态提示，买家的收货信息、卖家的发货信息与联系方式、发货后的物流状态查询、所购商品列表等都必须有正确的显示，这些内容都有助于买家更好的了解一次交易的全部信息。如图9-30 订单详情页面结构图中所示。

在产品设计开发过程中，永远要将自己设身处地的当成一位用户去看待自己制作的页面，如果我是一名对计算机操作或网络购物并不熟悉的用户，会产生什么疑虑？又需要哪些帮助才能很好地解决问题？不要让用户在你制作的页面中四处寻找他们想要的内

容，而应该把这些放在他们的眼前，这才是一款成熟产品理应具备的素质。

9.5.3 状态步骤的设计与制作

进度式的流程步骤及操作提示是一种很直观的表现形式，在用户中心除了商品详情页面中使用了这样的表现形式，在退款\退换货、投诉、举报等有可能涉及买卖双方操作的流程类详情内容页，都采用了此类设计。见图9-33和图9-34。



图9-33 订单详情步骤样式设计——开始时



图9-34 订单详情步骤样式设计——全部完成时

如代码清单9-22所示，我们截取此段页面的部分代码，分析一下它是如何制作的。

代码清单9-22 订单详情页面步骤样式制作的HTML部分

```
<div id="order-step" class="ncm-order-step">
  <dl class="step-first current">
    <dt>提交订单</dt>
    <dd class="bg"></dd>
    <dd class="date" title="下单时间">2014-02-20
17:20:17</dd>
  </dl>
  <dl class="">
    <dt>支付订单</dt>
    <dd class="bg"> </dd>
    <dd class="date" title="付款时间">2014-02-20
17:20:17</dd>
  </dl>
  <dl class="">
    <dt>商家发货</dt>
    <dd class="bg"> </dd>
    <dd class="date" titile="发货时间">2014-02-21
10:27:37</dd>
  </dl>
  <dl class="">
    <dt>确认收货</dt>
    <dd class="bg"> </dd>
    <dd class="date" title="完成时间">2014-02-27
11:34:37</dd>
  </dl>
  <dl class="">
    <dt>评价</dt>
    <dd class="bg"></dd>
    <dd class="date" title="完成时间">2014-03-11
15:49:07</dd>
  </dl>
</div>
```

代码清单9-23 订单详情页面步骤样式制作的CSS

部分

```
.ncm-order-step {
    font-size: 0;
*word-spacing: -1px/*IE6、7*/;
    margin-top: 30px;
    position: relative;
    z-index: 1;
}
.ncm-order-step dl {
    font-size: 12px;
    line-height: 20px;
    background: url(../images/pics.png) no-repeat -45px
0;
    vertical-align: top;
    letter-spacing: normal;
    word-spacing: normal;
    display: inline-block;
*display: inline/*IE7*/;
    width: 215px;
    height: 36px;
    margin: 50px 0 60px -1px;
    position: relative;
    z-index: auto;
*zoom: 1/*IE7*/;
}
.ncm-order-step dl.step-first {
    background-position: 0 0;
    width: 36px;
    margin-left: 50px;
}
.ncm-order-step dl dt {
    font-weight: 600;
    text-align: center;
    width: 60px;
    position: absolute;
    z-index: 1;
    top: -30px;
    right: -12px;
```

```

}
.ncm-order-step dl.current dt {
    color: #FD6760;
}
.ncm-order-step dl dd.bg {
    background: url(../images/pics.png) no-repeat -40px
-40px;
    display: none;
    width: 220px;
    height: 36px;
    position: absolute;
    z-index: 1;
    top: 0;
    right: 0;
}
.ncm-order-step dl.step-first dd.bg {
    background-position: 0 -40px;
    width: 36px;
}
.ncm-order-step dl dd.date {
    font: 12px/20px Tahoma, Arial;
    color: #999;
    text-align: center;
    display: none;
    width: 120px;
    position: absolute;
    z-index: 2;
    bottom: -40px;
    right: -42px;
}
.ncm-order-step dl.current dd {
    display: block;
}

```

代码片段9-16为订单详情页面流程步骤块的HTML部分代码内容，在一个div布局中使用了5个并列的dl、dt、dd形式，没一个dl代表一个节点段，dt为当

前订单节点文字，两个dd则用来设定图片和时间内
容，分别以不同的class控制，结构简单清晰。

再来看看对应控制这部分的CSS代码内容，见代码
清单9-17。

“ncm-order-step”是外围整体div的class，其
中“font-size: 0; *word-spacing: -1px/*IE6、
7*/;”是为了去除下级dl使用“display: inline-
block;”属性时，浏览器产生间距的问题。下级

“ncm-order-step dl”使用CSS sprite技术，靠背景
图片表现了每个步骤节点段的样式；并且具有

“position: relative;”相对层属性，为了控制后
面的dt及dd位置，而其下级的dt、dd都具有

“position: absolute;”绝对层属性，靠top和left
值来定位对于该段内容正确的显示位置。这种靠外围
进行内联块的横向排列，子级内容绝对定位的写法能

很准确地让几个节点段都显示定位很准确、浏览器兼容性高。“`.ncm-order-step dl dd.bg`”这个class用来控制每个节点处圆的图形表现，“`ncm-order-step dl dd.date`”则是下方的日期小字。而步骤进行到什么阶段是靠程序判断“`ncm-order-step dl`”上的class来控制的，如当前的class为空则下属的dt dd内容都会相应“`display: none;`”只有当class=“`current`”时才是已完成的步骤样式，如图9-34所示，则是五个“`ncm-order-step dl`”的class都是current。最后，仔细观察这五个dl会发现，第一个起始状态点要窄于其余的阶段，因为没有左侧的横向连接线，只需要给这个特殊的dl再加上“`step-first`”的class，设定它独立的宽度及背景图background-position值即可。

这种形式的步骤制作也可以转化成菜单、Tab切换卡等，根据页面需求灵活使用。

9.6 本章小结

本章概述了在网页设计、制作过程中一些常见问题及工作技巧，并以ShopNC B2B2C商城系统为例，介绍了设计制作电子购物类网站所应遵循的原则，设计的理念，以及页面的结构特征与购买流程之间的关系和部分代码分析。由于原商城模板中含有部分PHP语句，因此示例中的代码片段都是节选页面生成后的内容，仅供学习交流。通过这个章节，笔者也希望对网页前端开发从业者、爱好者能更好地了解电子商城的交易流程、操作原理，能对大家的学习与工作起到帮助，共同进步。

第10章 电商平台的安装与部署

本章我们以使用LNMP（Linux+Nginx+MySQL+PHP）架构的ShopNC B2B2C商城系统在CentOS 6.4 x64安装为例，讲解安装与部署过程。核心软件信息如表10-1所示。

表10-1 核心软件列表

名 称	版 本	安 装 路 径	用 途
Nginx	1.4.7	/usr/local/nginx	Web 服务器
PHP	5.4.26	/usr/local/php	Web 服务器
MySQL	5.6	/usr/local/mysql	数据库服务器
Redis	2.8.8	/usr/local/redis	缓存服务器
Node.js	0.10.25	/usr/local/node	即时通信服务器
xunsearch	1.4.8	/usr/local/xs	搜索服务器

10.1 Web服务器部署

Web服务器主要安装Nginx和PHP。

1. Nginx的安装与部署

1) 下载依赖库：openssl、zlib、pcre。

```
wget http://www.openssl.org/source/openssl-1.0.0l.tar.gz
tar xzf openssl-1.0.0l.tar.gz
mv openssl-1.0.0l openssl
wget http://zlib.net/zlib-1.2.8.tar.gz
tar xzf zlib-1.2.8.tar.gz
mv zlib-1.2.8 zlib
wget
ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/pcre-
8.33.tar.gz
tar xzf pcre-8.33.tar.gz
mv pcre-8.33 pcre
```

2) 开始安装Nginx。

```
# 创建Nginx运行所需账户及所在组
groupadd www
useradd -r -g www www -s /sbin/nologin
# 下载源码安装包并解压
cd /usr/local/src/
wget http://nginx.org/download/nginx-1.4.7.tar.gz
tar xzf nginx-1.4.7.tar.gz
```

```
cd nginx-1.4.7
# 配置Nginx编译参数
./configure \
"--prefix=/usr/local/nginx" \
"--sbin-path=/usr/local/nginx/sbin/nginx" \
"--conf-path=/usr/local/nginx/conf/nginx.conf" \
"--error-log-path=/usr/local/nginx/log/error.log" \
"--http-log-path=/usr/local/nginx/log/access.log" \
"--pid-path=/usr/local/nginx/var/nginx.pid" \
"--lock-path=/usr/local/nginx/var/nginx.lock" \
"--http-client-body-temp-path=/tmp/clientbody" \
"--http-proxy-temp-path=/tmp/proxy" \
"--http-fastcgi-temp-path=/tmp/fastcgi" \
"--http-uwsgi-temp-path=/tmp/uwsgi" \
"--http-scgi-temp-path=/tmp/scgi" \
"--user=www" \
"--group=www" \
"--with-file-aio" \
"--with-http_realip_module" \
"--with-http_ssl_module" \
"--with-openssl=/usr/local/src/openssl" \
"--with-http_gzip_static_module" \
"--with-zlib=/usr/local/src/zlib" \
"--with-http_stub_status_module" \
"--with-pcre=/usr/local/src/pcre" \
"--without-select_module" \
"--without-poll_module" \
"--without-http_ssi_module" \
"--without-http_userid_module" \
"--without-http_geo_module" \
"--without-http_empty_gif_module" \
"--without-http_map_module" \
"--without-mail_pop3_module" \
"--without-mail_imap_module" \
"--without-mail_smtp_module"
# 编译安装
make
make install
```

3) 防火墙开放80端口:

```
vi /etc/sysconfig/iptables
-A INPUT -m state --state NEW -m multiport -p tcp --
dport 22,80 -j ACCEPT
# 重启防火墙服务
service iptables restart
# 创建Nginx相关目录
mkdir /usr/local/nginx/conf/vhosts
mkdir -p /var/log/nginx/shop
mkdir /var/run/nginx
chown -R www:www /var/log/nginx
chown -R www:www /var/run/nginx
# 创建商城主程序存放目录
mkdir /wwwroot
```

编辑Nginx配置文件，设置基本参数：

```
vi /usr/local/nginx/conf/nginx.conf
user www;
worker_processes 2;
error_log /var/log/nginx/error.log error;
pid /var/run/nginx/nginx.pid;
events {
    worker_connections 10240;
    use epoll;
    multi_accept on;
}
http {
    include mime.types;
    default_type application/octet-stream;
    client_body_buffer_size 8k;
    client_body_temp_path /tmp/client_body_temp;
    client_body_timeout 30;
    client_header_buffer_size 32k;
    large_client_header_buffers 4 32k;
    client_header_timeout 30;
    client_max_body_size 32m;
    #keepalive_disable msie6 safari;
    keepalive_timeout 3;
    tcp_nodelay on;
    send_timeout 30;
```

```

    sendfile      on;
    tcp_nopush    on;
    server_names_hash_max_size      512;
    server_names_hash_bucket_size   128;
    server_tokens off;
    open_file_cache off;
#index
    index    index.php index.html index.htm;
#fastcgi
    fastcgi_connect_timeout        60;
    fastcgi_read_timeout           60;
    fastcgi_send_timeout           60;
    fastcgi_temp_path              /tmp/fastcgi_temp 1 2;
    fastcgi_buffer_size            64k;
    fastcgi_buffers                 256 4k;
    fastcgi_max_temp_file_size     256k;
    fastcgi_intercept_errors       on;
    fastcgi_index                  index.php;
#proxy
    proxy_temp_path                /tmp/proxy_temp;
    proxy_buffer_size              4k;
    proxy_buffering                 on;
    proxy_buffers                   256 4k;
    proxy_busy_buffers_size         8k;
#gzip
    gzip                            on;
    gzip_buffers                    16 4k;
    gzip_comp_level                 1;
    gzip_http_version              1.1;
    gzip_min_length                1024;
    gzip_types                     text/css text/xml text/plain
text/vnd.wap.wml application/x-javascript
application/rss+xml application/xhtml+xml;
#realip module
    set_real_ip_from               127.0.0.1;
    set_real_ip_from               10.0.0.0/8;
    set_real_ip_from               172.16.0.0/20;
    set_real_ip_from               192.168.0.0/16;
    set_real_ip_from               192.168.1.0/24;
    real_ip_header                  X-Real-IP;
    #real_ip_header                 X-Forwarded-For;
#log
    log_format    main    '$remote_addr - $remote_user
[$time_local] "$request" '
                        '$status $body_bytes_sent

```

```
"$http_referer" '
                                "$http_user_agent"
"$http_x_forwarded_for";
    #virtualhost
        include vhosts/*;
    }
```

4) 创建商城配置文件:

```
vi /usr/local/nginx/conf/vhosts/shop.conf
server {
    listen                80;
    server_name           localhost;
    access_log            /var/log/nginx/shop/access.log
main;
    root                 /wwwroot;
    location ~ .*\.
(gif|jpg|jpeg|png|bmp|zip|exe|txt|ico|rar|htm|html) $
    {
        expires 30d;
    }
    location ~ .*\.
(swf|mp3|wmv|wma|mp4|mpg|flv) $
    {
        expires 30d;
    }
    location ~ .*\. (js|css)? $
    {
        expires 30h;
    }
    location ~ /\.php$ {
        fastcgi_pass 127.0.0.1:9000;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME
/wwwroot$fastcgi_script_name;
        include fastcgi_params;
    }
}
```

5) 创建服务脚本文件:

```
vi /etc/init.d/nginx
#!/bin/sh
#
# nginx - this script starts and stops the nginx daemon
#
# chkconfig:   - 85 15
# description:  Nginx is an HTTP(S) server, HTTP(S)
reverse \
#                  proxy and IMAP/POP3 proxy server
# processname: nginx
# config:         /usr/local/nginx/conf/nginx.conf
# config:         /usr/local/nginx/sysconfig/nginx
# pidfile:        /usr/local/nginx/var/nginx.pid
# Source function library.
. /etc/rc.d/init.d/functions
# Source networking configuration.
. /etc/sysconfig/network
# Check that networking is up.
[ "$NETWORKING" = "no" ] && exit 0
nginx="/usr/local/nginx/sbin/nginx"
prog=$(basename $nginx)
sysconfig="/usr/local/nginx/sysconfig/$prog"
lockfile="/usr/local/nginx/lock"
pidfile="/usr/local/nginx/var/${prog}.pid"
NGINX_CONF_FILE="/usr/local/nginx/conf/nginx.conf"
[ -f $sysconfig ] && . $sysconfig
start() {
    [ -x $nginx ] || exit 5
    [ -f $NGINX_CONF_FILE ] || exit 6
    echo -n $"Starting $prog: "
    daemon $nginx -c $NGINX_CONF_FILE
    retval=$?
    echo
    [ $retval -eq 0 ] && touch $lockfile
    return $retval
}
stop() {
    echo -n $"Stopping $prog: "
    killproc -p $pidfile $prog
    retval=$?
```

```

        echo
        [ $retval -eq 0 ] && rm -f $lockfile
        return $retval
    }
    restart() {
        configtest_q || return 6
        stop
        start
    }
    reload() {
        configtest_q || return 6
        echo -n $"Reloading $prog: "
        killproc -p $pidfile $prog -HUP
        echo
    }
    configtest() {
        $nginx -t -c $NGINX_CONF_FILE
    }
    configtest_q() {
        $nginx -t -q -c $NGINX_CONF_FILE
    }
    rh_status() {
        status $prog
    }
    rh_status_q() {
        rh_status >/dev/null 2>&1
    }
    # Upgrade the binary with no downtime.
    upgrade() {
        local oldbin_pidfile="${pidfile}.oldbin"
        configtest_q || return 6
        echo -n $"Upgrading $prog: "
        killproc -p $pidfile $prog -USR2
        retval=$?
        sleep 1
        if [[ -f ${oldbin_pidfile} && -f ${pidfile} ]];
then
            killproc -p $oldbin_pidfile $prog -QUIT
            success $"$prog online upgrade"
            echo
            return 0
        else
            failure $"$prog online upgrade"
            echo
            return 1
        fi
    }

```

```

        fi
    }
    # Tell nginx to reopen logs
    reopen_logs() {
        configtest_q || return 6
        echo -n $"Reopening $prog logs: "
        killproc -p $pidfile $prog -USR1
        retval=$?
        echo
        return $retval
    }
    case "$1" in
        start)
            rh_status_q && exit 0
            $1
            ;;
        stop)
            rh_status_q || exit 0
            $1
            ;;
        restart|configtest|reopen_logs)
            $1
            ;;
        force-reload|upgrade)
            rh_status_q || exit 7
            upgrade
            ;;
        reload)
            rh_status_q || exit 7
            $1
            ;;
        status|status_q)
            rh_$1
            ;;
        condrestart|try-restart)
            rh_status_q || exit 7
            restart
            ;;
        *)
            echo $"Usage: $0
{start|stop|reload|configtest|status|force-
reload|upgrade|restart|reopen_logs}"
            exit 2
    esac

```

6) 添加到系统服务并开机启动:

```
chmod u+x /etc/init.d/nginx
chkconfig --add nginx
chkconfig nginx on
service nginx
{start|stop|reload|configtest|status|restart}
```

2. PHP的安装与部署

1) 安装依赖库libmccrypt、mysql-devel。

```
wget
ftp://mccrypt.hellug.gr/pub/crypto/mccrypt/libmccrypt/libmccrypt
-2.5.7.tar.gz
tar zxvf libmccrypt-2.5.7.tar.gz
cd libmccrypt-2.5.7/
./configure
make
make install
/sbin/ldconfig
cd libltdl/
./configure --enable-ltdl-install
make
make install
# 安装依赖库mysql-devel
yum install mysql-devel
# 设置依赖库
ln -s /usr/lib64/libXpm.so* /usr/lib/
mkdir /usr/lib/mysql
ln -s /usr/lib64/mysql/libmysqlclient* /usr/lib/mysql
```

2) 安装PHP:

```
wget http://tw1.php.net/get/php-5.4.26.tar.gz/from/this/mirror
tar xzf php-5.4.26.tar.gz
cd php-5.4.26
./configure \
"--prefix=/usr/local/php" \
"--with-config-file-path=/usr/local/php/etc" \
"--with-mysql=/usr" \
"--with-mysqli=/usr" \
"--with-libxml-dir" \
"--with-pcre-dir" \
"--with-mcrypt=/usr/local/lib/libmcrypt" \
"--with-gd" \
"--with-jpeg-dir" \
"--with-png-dir" \
"--with-xpm-dir" \
"--with-gettext" \
"--with-freetype-dir" \
"--with-bz2" \
"--with-iconv-dir" \
"--with-zlib" \
"--with-zlib-dir" \
"--with-curl" \
"--enable-fpm" \
"--with-fpm-user=www" \
"--with-fpm-group=www" \
"--with-icu-dir=/usr" \
"--enable-intl" \
"--enable-sockets" \
"--enable-gd-native-ttf" \
"--enable-gd-jis-conv" \
"--enable-ftp" \
"--enable-zip" \
"--enable-calendar" \
"--enable-mbstring" \
"--enable-exif" \
"--enable-fd-setsize=4096" \
"--disable-short-tags" \
"--disable-ipv6"
make
```

```
make install
# 创建配置文件
cp /usr/local/src/php-5.4.26/php.ini-production
/usr/local/php/etc/php.ini
cp /usr/local/php/etc/php-fpm.conf.default
/usr/local/php/etc/php-fpm.conf
```

3) 安装mysqli扩展:

```
cd php-5.4.26/ext/mysqli/
./configure --with-php-config=/usr/local/php/bin/php-
config --with-mysqli=/usr/local/mysql/bin/mysql_config
make
make install
echo "extension=/usr/local/php/lib/php/extensions/no-
debug-non-zts-20100525/mysqli.so" >>
/usr/local/php/etc/php.ini
# 检测是否安装成功
/usr/local/php/bin/php -m
```

4) 安装openssl扩展:

```
cd php-5.4.26/ext/openssl/
/usr/local/php/bin/phpize
./configure --with-php-config=/usr/local/php/bin/php-
config
make
make install
echo "extension=/usr/local/php/lib/php/extensions/no-
debug-non-zts-20100525/openssl.so" >>
/usr/local/php/etc/php.ini
# 检测是否安装成功
/usr/local/php/bin/php -m
```

5) 安装redis扩展。

```
wget
https://github.com/nicolasff/phpredis/archive/master.zip
unzip master.zip
cd phpredis-master/
/usr/local/php/bin/phpize
./configure --with-php-config=/usr/local/php/bin/php-
config
make
make install
echo "extension=/usr/local/php/lib/php/extensions/no-
debug-non-zts-20100525/redis.so" >>
/usr/local/php/etc/php.ini
# 检测是否安装成功
/usr/local/php/bin/php -m
```

6) 安装zendloader扩展:

```
wget
http://downloads.zend.com/guard/6.0.0/ZendGuardLoader-70429-
PHP-5.4-linux-glibc23-x86_64.tar.gz
tar xzf ZendGuardLoader-70429-PHP-5.4-linux-glibc23-
x86_64.tar.gz
cd ZendGuardLoader-70429-PHP-5.4-linux-glibc23-x86_64
cd php-5.4.x/
mv ZendGuardLoader.so
/usr/local/php/lib/php/extensions/no-debug-non-zts-
20100525/ZendGuardLoader.so
vi /usr/local/php/etc/php.ini
# 追加以下内容
[zendloader]
zend_loader.enable=1
zend_extension=/shop_app/php/lib/php/extensions/no-
debug-non-zts-20100525/ZendGuardLoader.so
zend_loader.disable_licensing=0
zend_loader.obfuscation_level_support=3
```

```
zend_loader.license_path=  
# 保存退出  
# 检测是否安装成功  
/usr/local/php/bin/php -m
```

7) 创建PHP运行所需要的目录并设置权限:

```
mkdir /var/run/php  
mkdir /var/log/php  
chown -R www:www /var/run/php/  
chown -R www:www /var/log/php/
```

8) 设置php-fpm配置, 修改以下参数项:

```
vi /usr/local/php/etc/php-fpm.conf  
pid = /var/run/php/php-fpm.pid  
error_log = /var/log/php/error.log  
user = www  
group = www  
listen = 127.0.0.1:9000  
pm = static  
# 保存退出  
# 测试配置是否正确  
service nginx configtest
```

9) php-fpm加入系统服务并开机启动:

```
cp /usr/local/src/php-5.4.26/sapi/fpm/init.d.php-fpm  
/etc/init.d/php-fpm  
# 编辑PHP相关路径  
vi /etc/init.d/php-fpm
```

```
chmod u+x /etc/init.d/php-fpm
chkconfig --add php-fpm
service php-fpm start
chkconfig php-fpm on
```

10.2 数据库部署

这里以使用较多的主从模式进行部署。

1) 主数据库服务器部署，下载mysql-server二进制安装包：

```
wget http://cdn.mysql.com/Downloads/MySQL-5.6/mysql-5.6.17-linux-glibc2.5-x86_64.tar.gz
```

2) 创建MySQL运行所需账户及所在组：

```
groupadd mysql
useradd -r -g mysql mysql -s /sbin/nologin
./scripts/mysql_install_db --user=mysql --
datadir=/var/lib/mysql/
```

3) 编辑配置文件：

```
vi /usr/local/mysql/my.cnf
[mysqld]
basedir = /usr/local/mysql
datadir = /var/lib/mysql
port = 3306
server_id = 1
```

```
socket = /var/lib/mysql/mysql.sock
log_error = /var/log/mysql/error.log
log-bin = /var/log/mysql/log_bin/log_bin
binlog-do-db=shop
sql_mode=NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES
[mysqld_safe]
pid-file = /var/run/mysql/mysqld.pid
```

4) 创建数据库存储信息所需目录:

```
mkdir /var/lib/mysql
mkdir /var/log/mysql
mkdir /var/log/mysql/log_bin
mkdir /var/run/mysql
# 设置目录所有者
chown -R mysql:mysql /var/lib/mysql/
chown -R mysql:mysql /var/log/mysql/
chown -R mysql:mysql /var/run/mysql/
# 测试启动是否成功
bin/mysqld_safe --user=mysql &
```

5) 加入系统服务:

```
# 编辑 mysql.server 内容
vi support-files/mysql.server
basedir=/usr/local/mysql
datadir=/var/lib/mysql
mysqld_pid_file_path=/var/run/mysql/mysqld.pid
# 保存退出
# 将mysql加入系统服务
cp support-files/mysql.server /etc/init.d/mysqld
chkconfig --add mysqld
chkconfig mysqld on
# 启动mysql服务
service mysqld start
```

6) 设置数据库账号密码等信息:

```
# 设置数据库root密码
./bin/mysqladmin -u root password 'xxxxxx'
# 以root身份进入数据库
mysql -uroot -p
# 创建商城使用的账号和数据库并授权
mysql> CREATE DATABASE `shop` CHARACTER SET utf8 COLLATE
utf8_general_ci;
mysql> CREATE USER 'mall_shop'@'localhost';
mysql> use mysql;
mysql> select user from user;
mysql> update user set password=PASSWORD('xxxxxx') where
user="mall_shop";
mysql> GRANT ALL PRIVILEGES ON `shop` . * TO
'mall_shop'@'localhost' WITH GRANT OPTION ;
```

7) 创建数据同步账号并授权:

```
mysql> CREATE USER 'repl_user'@'%' IDENTIFIED BY
'xxxxxx';
mysql> GRANT REPLICATION SLAVE , REPLICATION CLIENT ON *
. * TO 'repl_user'@'%' IDENTIFIED BY '111111';
mysql> flush privileges;
# 查看数据库主服务器同步状态
mysql> show master status;
+-----+-----+-----+-----+
-----+
| File           | Position | Binlog_Do_DB |
Binlog_Ignore_DB |
+-----+-----+-----+-----+
-----+
| log_bin.000001 |      107 | shop         |
|
+-----+-----+-----+-----+
-----+
1 row in set (0.00 sec)
```

8) 从数据库服务器安装与主服务器安装基本相同，主要在配置文件方面有所区别，从数据库服务器需要指明数据同步的数据库名称，并启动同步进程。

```
# 编辑从服务器配置文件
vi /usr/local/mysql/my.cnf
[mysqld]
server-id = 2
replicate-do-db=shop
```

9) 从服务器以同步账户身份连接到主服务器开始数据同步：

```
mysql> CHANGE MASTER TO MASTER_HOST='10.0.0.1',
MASTER_PORT=3306,MASTER_
USER='repl_user',
MASTER_PASSWORD='111111',Master_Log_Pos=0,MASTER_LOG_FILE='l
og_bin.000001';
```

10) 检测从服务器是否配置成功：

```
mysql> SHOW SLAVE STATUS\G;
Slave_IO_State = Waiting for master to send event
Seconds_Behind_Master = 0
# 以上信息表明从服务器进入同步等待状态成功
```

配置完成以后，需要测试一下，可以在主数据库服务器中创建一个任意表，然后观察从数据库服务器是否同步创建了这个表，如果从数据库成功创建，表明主从同步设置成功，如果从数据库创建表失败，则主从配置存在问题，需要进一步调试。

10.3 电商系统安装

10.3.1 基本安装

在Nginx、PHP、MySQL安装并配置完成以后，可以安装ShopNC B2B2C商城了。

首先需要仔细阅读安装协议，如图10-1所示。



图10-1 阅读安装协议

阅读并同意后开始检查安装环境，如图10-2所示。如果检测有未通过项，将显示红色，提示无法进行下一步。

Step.1 开始安装

检测服务器环境及文件目录权限

检查安装环境

选择安装方式

创建数据库

安装

环境检查

项目	ShopNC所需	最佳配置推荐	当前服务器
操作系统	无限制	linux	✔ Linux
PHP版本	5.2	5.3	✔ 5.4.24
GD库	2.0	2.0	✔ 2.1.0
附件上传	未限制	2M	✔ 2M
磁盘空间	100M	>100M	✔ 44467M

目录、文件权限检查

目录文件	所需状态	当前状态
data/cache	可写	✘ 不可写
data/cache/adv	可写	✘ 不可写
data/cache/rec_position	可写	✘ 不可写
data/config	可写	✘ 不可写
data/log	可写	✘ 不可写

图10-2 权限检测

待所有检查项均通过后，进入下一步来选择安装方式（目前只有一个选项），如图10-3所示。



图10-3 选择安装方式

接下来输入数据库账号等信息，如图10-4所示。

如果选择安装演示数据，请从官方下载图片压缩包，解压到商城根目录。下一步开始安装数据库，如图10-5所示。

检查安装环境

选择安装方式

创建数据库

安装

数据库信息

数据库服务器

localhost

数据库服务器地址，一般为localhost

数据库名

shopnc

数据库用户名

root

数据库密码

数据库表前缀

shopnc_

同一数据库运行多个ShopNC程序时，请修改前缀

数据库端口

3306

数据库默认端口一般为3306

☒ 演示数据，如果选择安装演示数据，请下载图片文件压缩包

网站信息

站点名称

ShopNC B2B2C商城 - 演示站

输入站点名称，安装后可在平台设置中进行修改

管理员帐号

admin

管理员密码

管理员密码不少于6个字符

重复密码

*****|

确保两次输入的密码一致

图10-4 安装信息设置



图10-5 安装数据库

安装数据库成功，下一步提示安装成功并显示安装相关信息，如图10-6所示。



图10-6 安装信息展示

安装完成后, 进入后台, 选择设置→清理缓存→命令清理全部缓存, 如图10-7所示。



图10-7 系统后台——清理缓存

点击后台右上角的商城首页链接，查看一下商城首页，如图10-8所示，内容正确展示，至此安装过程顺利完成。

10.3.2 设置定时任务

系统中的一些操作（如生成统计数据、生成搜索索引数据等）需要使用系统任务定时触发来完成。待触发的程序文件位于商城目录下的data/crontab目录中，index.php是统一入口文件，通过include目录中的文件来完成各个子功能。定时任务触发目录及触发频率参考如下：



图10-8 首页展示

```
# /usr/local/php/bin/php为PHP可执行文件所在路径
# 更新商品促销到期状态，建议执行频率：1天
/usr/local/php/bin/php /wwwroot/data/crontab/index.php
goods promotion
# 执行通用任务，建议执行频率：5分钟
# 主要是商品上架、商品促销价格、店铺促销活动开始/过期等，执行频率越高，商品列表页价格显示越准确
/usr/local/php/bin/php /wwwroot/data/crontab/index.php
goods common
# 更新首页商品价格，建议执行频率：10分钟
/usr/local/php/bin/php /wwwroot/data/crontab/index.php
goods web_update
# 将缓存中的浏览记录存入数据库，建议执行频率：1天
/usr/local/php/bin/php /wwwroot/data/crontab/index.php
goods browse
# 邮件发送，建议执行频率：15分钟
# 主要是店铺、会员的消息发送
/usr/local/php/bin/php /wwwroot/data/crontab/index.php
mail index
# 会员相关数据统计，建议执行频率：1天
/usr/local/php/bin/php /wwwroot/data/crontab/index.php
stat index
# 发送到货通知，建议执行频率：每天执行一次
/usr/local/php/bin/php /wwwroot/data/crontab/index.php
arrival_notice index
# 生成结算单，建议执行频率：每月1号(1:00 ~ 4:00)执行
/usr/local/php/bin/php /wwwroot/data/crontab/index.php
order create_bill
# 全量创建索引，初始安装时，执行一次即可
/usr/local/php/bin/php /wwwroot/data/crontab/index.php
xs create
# 更新增量索引，建议执行频率：5小时
/usr/local/php/bin/php /wwwroot/data/crontab/index.php
xs update
```

下面给出一个配置示例：

```
# 每天1:00 执行
0 1 * * * /usr/local/php/bin/php
/wwwroot/data/crontab/index.php goods promotion
# 每天1:30 执行
30 1 * * * /usr/local/php/bin/php
/wwwroot/data/crontab/index.php goods common
# 每10分钟执行一次
*/10 * * * * /usr/local/php/bin/php
/wwwroot/data/crontab/index.php goods web_update
# 每天1:30 执行
30 1 * * * /usr/local/php/bin/php
/wwwroot/data/crontab/index.php goods browse
# 每天2:00 执行
0 2 * * * /usr/local/php/bin/php
/wwwroot/data/crontab/index.php stat index
# 每天2:30 执行一次
30 2 * * * /usr/local/php/bin/php
/wwwroot/data/crontab/index.php arrival_notice index
# 每15分钟执行一次
*/15 * * * * /usr/local/php/bin/php
/wwwroot/data/crontab/index.php mail index
# 每月1号3:00执行
0 3 1 * * /usr/local/php/bin/php
/wwwroot/data/crontab/index.php order create_bill
# 每5小时执行一次
* */5 * * * /usr/local/php/bin/php
/wwwroot/data/crontab/index.php xs update
```

10.3.3 开启伪静态

系统支持动态网址和伪静态两种URL形式，在CONFIG（data/config/config.ini.php）中：
url_model为true，表示开启伪静态；url_mode为false表示使用默认动态网址形式。系统伪静态是基于PATHINFO模式实现的，所以若开启伪静态，除更改CONFIG外，还需要配置APACHE/NGINX，以支持PATHINFO模式。

注意，当开启伪静态时，CONFIG中的各子模块URL网址必须全部配置并启用，下面以NGINX为例。

```
# nginx配置
location / {
    if (!-e $request_filename)
    {
        rewrite ^/shop/(.*)$ /shop/index.php?$1;
    }
}
location ~ \.php$ {
    fastcgi_param PATH_INFO $request_uri;
}
```



10.3.4 设置域名

ShopNC B2B2C商城默认可以设置众多子模块的域名，归纳如表10-2所示。

表10-2 域名列表

说 明	CONFIG 配置项	网 址	指 向 目 录
商城	shop_site_url	http://www.shopnc.net	shop
CMS	cms_site_url'	http://cms.shopnc.net	cms
圈子	circle_site_url	http://circle.shopnc.net	circle
微商城	microshop_site_url	http://microshop.shopnc.net	microshop
后台	admin_site_url	http://system.shopnc.net	admin
手机端 API	mobile_site_url	http://mobile.shopnc.net	mobile
WAP	wap_site_url	http://wap.shopnc.net	wap
IM Client	chat_site_url	http://chat.shopnc.net	chat
IM Server	node_site_url	http://node.shopnc.net:8090	IM Server
图片	upload_site_url	http://img.shopnc.net	data/upload
CSS、JS、IMG 等	resource_site_url	http://resource.shopnc.net	data/resource

以上二级域名也可自行命名，建议主域名指向商城作为主站。除以上提到的指向目录以外，其他目录强烈建议均不开放域名指向。

二级域名配置完成后，需要修改配置文件的
subdomain_suffix项，将其设置为根域名，例如：

```
$config['subdomain_suffix'] = 'shopnc.net';
```

10.4 图片存储部署

图片部署采用NFS来实现。安装NFS服务器端与客户端：

```
# NFS服务器端部署
yum install nfs nfs-utils portmap
# 创建图片存储目录
mkdir /nfs_server
mkdir /nfs_server/upload
# 设置客户端访问权限
vi /etc/exports
/nfs_server/upload          10.0.0.*
(rw,async,anonuid=500,anongid=500)
# 保存退出
chkconfig nfs on
service nfs start
# 将upload内容同步到NFS服务器
rsync -avz --progress --rsh=ssh
root@ip:/wwwroot/data/upload/ /nfs_server/upload/
# 客户端(即Web服务器)同样需要安装nfs nfs-util portmap, 安装完后, 需要挂载到服务器上
mount -t nfs -o soft,intr,bg,timeo=10
10.0.0.9:/nfs_server/upload /wwwroot/data/upload
# 查看是否挂载成功
mount
# 测试成功后, 把挂载写到/etc/fstab中, 开机加载
vi /etc/fstab
10.0.0.9:/nfs_server/upload /data/wwwroot/data/upload
nfs soft,intr,bg,timeo= 10 0 0
# 保存退出
```

10.5 缓存服务器部署

缓存服务器采用Redis主从架构来实现。安装主Redis服务器如下所示：

```
# 安装主Redis服务器
wget http://download.redis.io/releases/redis-
2.8.8.tar.gz
tar xzf redis-2.8.8.tar.gz
cd redis-2.8.8
make
# 测试一下可否启动
./src/redis-server ./redis.conf
# 创建Redis安装目录
mkdir /usr/local/redis
cp -rf ./src/redis-cli ./src/redis-server ./redis.conf
/usr/local/redis/
cd /usr/local/redis/
# 编辑配置文件
vi redis.conf
# 修改以下参数
daemonize yes
pidfile /var/run/redis.pid
port 6379
# 保存退出
# 复制启动文件到init.d目录
cp /usr/local/src/redis-2.8.8/utils/redis_init_script
/etc/init.d/redis
# 编辑相关参数，并在顶部插入以下两行内容
vi /etc/init.d/redis
#chkconfig: 2345 80 90
#description:auto_run
# 保存退出
# 加入系统服务
chkconfig --add redis
```

```
chkconfig redis on  
service redis start
```

从Redis服务器安装过程与主服务器安装完全相同，只需在从Redis配置文件中追加一行内容，来指定主服务器的IP和端口：

```
vi redis.conf  
# 追加以下内容  
slaveof 主IP 6379
```

10.6 搜索服务器部署

搜索服务器使用xunsearch来实现，系统将搜索关键词交给xunsearch，xunsearch通过分词搜索，返回商品主键ID的集合，然后系统通过主键查询数据库，返回需要显示的字段信息。

```
# 运行下面指令以下载、解压安装包
wget http://www.xunsearch.com/download/xunsearch-full-latest.tar.bz2
tar -xjf xunsearch-full-latest.tar.bz2
# 执行安装脚本，根据提示进行操作，主要是输入 xunsearch 软件包的
安装目录，强烈建议单独规划
# 一个目录，而不是混到别的软件目录中，这里安装到/usr/local/xs
目录下
cd xunsearch-full-latest/xunsearch-full-1.4.8/
sh setup.sh
# 待命令运行结束后，如果没有出错中断，则表示顺利完成安装，然后就可
以启动/重新启动
# xunsearch 的后台服务
cd /usr/local/xs; bin/xs-ctl.sh restart
# 建议将此命令添加到开机启动脚本中，以便每次服务器重启后能自动启动
搜索服务程序
# 关于搜索项目的数据目录规划。搜索系统将所有数据保存在
$prefix/data 目录中。 如果您希望数
# 据目录另行安排或转移至其他分区，请将 $prefix/data 作为软链
接指向真实目录
# 启动xs
bin/xs-ctl.sh -b local start      // 监听在本地回环地址
127.0.0.1 上
bin/xs-ctl.sh -b inet start       // 监听在所有本地 IP 地址上
bin/xs-ctl.sh -b a.b.c.d start    // 监听在指定 IP 上
```

```
bin/xs-ctl.sh -b unix start // 分别监听在  
tmp/indexd.sock 和 tmp/searchd.sock
```

安装完成后，更改

CONFIG (data/config/config.ini.php) 文件全文搜索部分，配置如下：

```
# 全文检索配置，支持true（开启）或false（关闭），全文检索的详细配置，请参考本帮助的全文检索章节  
# 需编辑 data\api\xs\app\shopnc.ini，将server.index 和  
server.search的值修改成自己的IP和端口  
$config['fullindexer']['open'] = true;  
# 全文检索配置文件名(默认为shopnc，不需要更改)  
$config['fullindexer']['appname'] = 'shopnc';
```

然后更改全文搜索配置文件

(data/api/xs/app/shopnc.ini)，将server.index和server.search的值修改成自己的IP和端口，shopnc.ini文件完整解释如下：

```
# 项目名称（不需要更改）  
project.name = shopnc  
# 默认字符集（该项要与当前商城字符集一致）  
project.default_charset = UTF-8  
# 索引服务端配置，默认端口 8383  
server.index = 127.0.0.1:8383  
# 搜索服务端配置，默认端口 8384  
server.search = 127.0.0.1:8384
```

```
# 后面内容为搜索字段配置，不需要更改
[pk]
type = id
# 商品ID
[goods_id]
type = numeric
.....
```

配置完成后，启动xunsearch进程、将商品内容索引到搜索服务器：

```
# 启动 xunsearch 进程
/usr/local/xunsearch/bin/xs-ctl.sh -b inet start
# 清空索引
/usr/bin/php /wwwroot/html/crontab/index.php xs clear
# 全量更新索引
/usr/bin/php /wwwroot/html/crontab/index.php xs create
# 增量更新索引，此命令可以加入系统任务计划中，定时执行，建议执行周期为1小时
/usr/bin/php /wwwroot/html/crontab/index.php xs update
```

至此配置工作已经完成，前台可以体验全文搜索了。那么如何验证是否使用全文搜索呢？打开系统的调试模式，底部会显示全部的SQL执行追踪记录，然后输入一个有搜索结果的关键词，比如“女装”，观察底部的SQL执行记录，如果未出现典型的SQL模糊查询，则说明配置成功。

10.7 IM服务器部署

IM意为“即时通讯”，在商城系统，它是买家与商家的沟通工具，通过安装Node.js来实现，从官网（<http://nodejs.org/download/>）下载二进制包解压、下载后直接使用，也可以下载源码包编译安装，这里以安装二进制包为例：

```
wget http://nodejs.org/dist/v0.10.28/node-v0.10.28-  
linux-x64.tar.gz  
tar xzf node-v0.10.28-linux-x64.tar.gz  
mv node-v0.10.28-linux-x64 /usr/local/node
```

将安装包内的im/lib/db.js和im/lib/users.js复制到/usr/local/node/lib下，将安装包内的im/chat.js和im/config.js复制到/usr/local/node下，然后编辑/usr/local/node/config.js文件，配置示例如下所示：


```
var config = {}; //数据库账号设置
config['host']      = '10.10.10.50'; //数据库地址
config['port']      = '3306'; //数据库端口
config['user']      = 'shopnc'; //数据库用户名
config['password']  = 'xxxxxx'; //数据库密码
config['database']  = 'shopnc'; //MySQL数据库名
config['tablepre']  = 'shopnc_'; //表前缀
config['insecureAuth'] = true; //兼容低版本
config['debug']     = false; //默认false
exports.hostname = ''; //授权连接的域名或IP, 该值为空表示不限制
                        域名或IP
exports.port = 8090; //服务器所用端口号, 默认是8090, 也可自定义
exports.config = config;
```

然后使用npm安装最新的MySQL与socket.io模块:

```
cd /usr/local/node
/usr/local/node/bin/npm install mysql
/usr/local/node/bin/npm install socket.io
```

启动node进程:

```
/usr/local/node/bin/node /usr/local/node/chat.js
```

如果出现以下内容说明配置成功:

```
info - socket.io started
mysql connected
```

然后停止当前node，将node以后台方式启动：

```
nohup /usr/local/node/bin/node  
/usr/local/node/chat.js >>  
/usr/local/node/output.log &
```

使用会员账号登录到商城，如果发现右下角出现IM图标，说明IM配置成功，如图10-9所示。



图10-9 IM安装

启动成功后，将node启动命令加入系统启动中。
如果右下角未出现IM框，请检查防火墙中是否禁止了node监听端口。

10.8 本章小结

本章以目前普遍采用的LNMP架构为例进行部署，使用一台Web服务器（Ngnix+PHP）、两台数据库服务器（MySQL主从部署）、一台图片服务器（NFS）、一台缓存服务器（Redis）和一台即时通信服务器

（Node.js）来实现。系统采用何种部署架构主要依据系统规模而定，如果系统规模较大，可以对以上的配置进行扩展，根据实际需要可以加入负载均衡等设备。如果规模较小，还可以对以上部署进行适当减配。

前面第3~9章对B2B2C商城系统主要模块从设计到实现进行了详细的讲解，本章作为最后一章，演示了ShopNC B2B2C商城在CentOS 6.4 x64下的安装与配置过程，安装完成后，可以亲身体会第3~9章的理论，从而增强对商城的整体认知。

附录A 移动端接口文档

A.1 接口约定

1. 数据格式

```
{code: 200, datas: {}}
```

返回数据为json格式，code为200表示数据正常，datas为返回的数据。

2. 获取数据

```
//有内容直接返回数组，无内容返回空数组[]  
//hasmore为是否有下一页  
{"code":200,"hasmore":true,"datas":{"goods_list":[]}}
```

3. 操作成功

```
//操作成功返回"1"  
{"code":200,"datas":{"1"}}
```

4. 错误

```
//出现错误返回error, error内为错误信息
{"code":200,"datas":{"error":"参数错误"}}
//登录失败, login:0
{"code":200,"login":"0","datas":
{"error":"\u8bf7\u767b\u9646"}}
```

A.2 商城API

1. 首页

(1) 首页数据接口 (get)

```
index.php?act=index
```

(2) 请求参数

```
type : json/html json格式或者html页面
```

(3) 返回数据

- adv_list: 首页广告列表

- home1: 首页类型1
- home2: 首页类型2
- home3: 首页类型3
- home4: 首页类型4
- goods: 首页商品

2. 专题

(1) 专题接口 (get)

```
index.php?act=index&op=special
```

(2) 请求参数

- special_id: 专题编号
- Type: json/html json格式或者html页面

(3) 返回数据

- adv_list: 专题广告列表
- home1: 专题类型1
- home2: 专题类型2
- home3: 专题类型3
- home4: 专题类型4
- goods: 专题商品

3. 商品分类

(1) 一级分类接口 (get)

```
index.php?act=goods_class
```

(2) 指定分类接口 (get)

```
//有下级分类返回分类列表，无分类返回'0'  
index.php?act=goods_class&gc_id=2
```

(3) 返回数据

Class_list : 分类列表

4. 商品列表

(1) 商品列表接口 (get)

```
index.php?act=goods&op=goods_list
```

(2) 请求参数

· Key: 排序方式: 1-销量, 2-浏览量, 3-价格,
空-按最新发布排序

· order: 排序方式: 1-升序, 2-降序

· page: 每页数量

- curpage: 当前页码
- gc_id: 分类编号
- keyword: 搜索关键字

(3) 返回数据

- goods_id: 商品编号
- goods_name: 商品名称
- goods_price: 商品价格
- goods_marketprice: 商品市场价
- goods_salenum: 销量
- evaluation_good_star: 评价星级
- evaluation_count: 评价数

- group_flag: 是否团购
- xianshi_flag: 是否限时折扣
- goods_image: 图片名称
- goods_image_url: 图片地址

5. 商品详细信息

(1) 商品详细信息接口 (get)

```
index.php?act=goods&op=goods_detail
```

(2) 请求参数

```
goods_id : 商品编号
```

(3) 返回数据

- goods_name: 商品名称

- goods_jingle: 商品说明
- spec_name: 规格名称
- spec_value: 规格名
- goods_price: 商品价格
- goods_marketprice: 商品市场价
- goods_id: 商品编号
- goods_click: 商品点击数
- goods_commentnum: 商品评论数
- goods_salenum: 商品销量
- goods_spec: 商品规格
- goods_storage: 商品库存

- evaluation_good_star: 评价等级
- evaluation_count: 评价数
- promotion_type促销类型: groupbuy-团购, xianshi-限时折扣
- promotion_price: 促销价格
- upper_limit: 最多购买数

6. 商品介绍信息

(1) 调用接口 (get)

```
index.php?act=goods&op=goods_body
```

(2) 请求参数

```
goods_id : 商品编号
```

(3) 返回数据

html : 商品介绍html页面

附录B ShopNC B2B2C商城功能要点

本附录将罗列ShopNC B2B2C功能要点。

功 能	说 明
买卖家分离模式	买卖双方各有独立的登录入口，买家登陆后由买家中心、SNS 等模块组成，可满足买家的购物、物流跟踪、分享、晒单、圈子等多方面的需求。商家后台由商品、订单、促销、店铺、物流等组成
平台自营模式	系统采用市场主流的（类京东、亚马逊等）平台店铺与第三方供应商结合的模式，使平台前期更快速的充实商品
定期佣金结算模式	平台可按照商家入驻所选类目分别设置各个供应商佣金，按月进行账单结算并收取佣金
在线 IM 通信系统	自主开发独立在线 IM 系统，使用买卖双方沟通不在的障碍
商品独立的 SKU	stock keeping unit, 简称 SKU，定义为保存库存控制的最小可用单位，例如纺织品中一个 SKU 通常表示：规格、颜色、款式等，详细展示商品特性
物流自提点功能	买家可选择最近的物流自提点进行代收货，自提点收到货物后，由系统发送自提码至买家手机中。买家根据自提码至相应的自提点进行取货
F 码商品销售功能	商品发布时可在特殊商品项中选择“F 码商品”，买家只有凭 F 码才能购买
线下兑换码	发布以线下兑换码形式进行交易的虚拟类型商品，完善商城系统在 O2O 部分的功能扩展
强大促销支持	内置满即送、限时折扣、团购、优惠套装、绑定赠品等多种促销手段
整合 CMS 系统	商城与 CMS 实现无缝对接，CMS 为商城引流最终促成交易完成
SNS 用户中心	以用户组为中心的 SNS，用户晒单、分享更加便捷
店铺装修	网店的美化如同实体店的装修一样重要。网店的页面其实是附着了店主灵魂的销售员，只有独具心裁的网店装修才能打动顾客
数据分析与统计	帮助平台与商家迅速掌握运营趋势

ShopNC成员简介

(按汉语拼音顺序排列)

邓启文

资深PHP开发工程师、.NET工程师，拥有多年LAMP架构开发经验，长期关注CMS与商城的融合，目前致力于ShopNC桌面系列产品的开发。主要参与本书第3章（部分内容）、第4章的撰写。

韩冰

资深产品经理、前端设计师，拥有多年前端设计与开发经验，负责把握ShopNC核心产品的开发方向。主要参与本书第9章的撰写。

何金刚

资深移动应用开发工程师（Android方向），专注于电商领域移动通信软件开发，实战经验丰富，ShopNC移动端产品的核心设计与开发者。主要参与本书第7章的撰写。

侯宝剑

具有多年LAMP架构开发经验，在服务器集群/分布式部署、性能优化等方面有独到见解，目前致力于大数据下电商系统的处理。主要参与本书第2章、第3章（部分内容）、第10章的撰写。

李梓平

资深移动应用开发工程师（IOS方向），负责ShopNC移动端开发与设计，资深PHP开发工程师，曾参与过多个电商产品项目的开发，对电子商务商城、ERP

领域有较多的研究和丰富的实战开发经验。主要参与本书第8章（部分内容）的撰写。

刘海峰

资深PHP开发工程师，ShopNC商城最核心的商品模块设计与开发者，具有多年大型互联网应用架构经验，一直关注社交网络，对社交化电子商务网站有深入的研究。主要参与本书第3章（部分内容）、第5章的撰写。

王景辰

资深PHP开发工程师，一直关注电商O2O趋势，也是ShopNC O2O系列产品的核心开发者。主要参与本书第8章（部分内容）的撰写。

王伟

资深即时通讯开发工程师，拥有多年开发经验，也是ShopNC在线IM的核心开发者。主要参与本书第3章（部分内容）、第6章的撰写。

尹震海

资深C开发工程师，拥有多年C开发经验、致力于将PHP向C扩展迁移，提高系统运行性能，目前主要致力电子商城安全防护的研究。主要参与本书第3章（部分内容）的撰写。

杨华成

资深网络营销师、电子商务运营师、电子商务职业经理人。从事电商运营及平台策划工作10多年，现任ShopNC大客户经理。主要参与本书第1章的撰写。

周喜颖

资深CRM、ERP开发工程师，拥有多年软件（特别是电商库管软件、CRM软件等）开发经验，致力于桌面软件和WEB端数据互通，提高运营效率。主要参与本书第3章（部分内容）的撰写。