



四川理工学院毕业设计（论文）

基于 B/C2C 模式的电商网站系统 的设计与实现

学 生：叶科忠

学 号：11101020323

专 业：软件工程

班 级：2011 级 3 班

指导教师：梁兴建

四川理工学院计算机学院

二〇一五年六月

基于 B/C2C 模式的电商网站系统的设计与实现

摘 要

本系统基于 OneThink 框架，根据目前大学校园综合性购物平台的市场特点和产品现状，采用 MVC 的三层架构模式开发了一套同时面向 B2C 和 C2C 的校园电商网站系统。系统根据现存系统的局限性，实现了在线购物、商品寄卖、二手交易等主要功能模块，作为实现这些功能模块的基础，系统同时实现了用户注册、用户管理、购物车管理、订单管理，商品和库存管理等基础模块。为了保证电商网站的灵活性，系统还实现了插件模块化。

本文分别从需求分析、设计方式、实现方案与细节等角度，详细地阐述了本系统的设计与实现过程和方法，最后经过了详细的功能和安全测试，找出并优化了系统存在的不足。

关键词：三层架构；在线购物；二手交易；电商平台；

The design and implementation of electronic business website system based on pattern B/C2C

ABSTRACT

According to the characteristics of the market and status of the product of current university campus comprehensive shopping platform, the system, on account of the framework OneThink, helps to develop a campus electronic business website system based on B2C and C2C by means of implementing the three layer MVC architecture model. The system realizes the running of the basic module: user registration and management, shopping cart, order management, goods and inventory management, and realizes several main function module: online shopping, secondary goods trading. Further more, it realizes modulization of plugin promising the flexibility of the electronic business website.

The process and the methods of the system's design and implementation are explained in the paper from three angles: requirements analysis, design methods and implementation methods. Under the detail testing towards the safety and the functions, the shortcomings of the system is picked out and solved . Thus, the system is optimized.

Keywords: Three-tier architecture; Online Shopping; Secondary trading; Electronic business platform

目 录

第 1 章 绪 论.....	1
1.1 基于 B/C2C 模式的校园电商网站系统的开发背景	1
1.2 国内外的发展现状和发展前景	1
1.3 本文的组织结构	1
第 2 章 校园电商系统需求分析.....	3
2.1 用户需求分析	3
2.1.1 市场优势	3
2.1.2 项目优势	3
2.2 系统基本功能模块分析	4
2.2.1 前台模块	5
2.2.2 后台模块	5
第 3 章 系统开发技术的介绍.....	7
3.1 PHP 编程语言介绍	7
3.2 MySQL 数据库介绍	7
3.3 NGINX 服务器介绍	8
3.4 PHP-FPM 介绍	8
3.5 RESTFul 标准介绍.....	8
3.6 MVC 架构模式介绍	8
第 4 章 数据库设计.....	9
4.1 数据流程图	9
4.2 数据库表的设计分析	9
4.3 数据库表的详细设计	10
第 5 章 系统功能模块设计分析.....	29
5.1 系统功能模块	29
5.2 功能流程图	29
5.2.1 系统设置/自定义页面模块管理.....	29
5.2.2 商品购买流程图	30
5.2.3 帐号、权限管理流程图	31
第 6 章 系统主要功能模块的具体实现.....	33
6.1 注册/登录模块接口及实现	33
6.1.1 接口	33
6.1.2 实现	37

6.2 分栏目分类商品展示	38
6.2.1 无限级别分类的数据结构	38
6.2.2 无限级别分类生成分类树接口	38
6.2.3 列表转换成分类树接口	39
6.3 购物车	41
6.3.1 购物车的友好化设计	41
6.3.2 购物车的 Ajax 实现	42
6.3.3 修改收货地址的实现	44
6.4 订单管理	44
6.5 积分系统	45
6.6 学校/团队管理	46
6.6.1 学校/管理员账户管理页面设计	46
6.6.2 管理员权限设置前端页面设计	47
6.6.3 将用户添加到用户组的实现	48
6.6.4 将用户从用户组移除的实现	49
6.6.5 将分类添加到用户组的实现	49
6.7 用户管理	50
6.8 商品/库存管理	50
6.9 订单管理	53
6.10 基于 RBAC 认证的全局权限管理	53
6.10.1 RABC 认证节点	53
6.10.2 RABC 认证权限	55
6.11 系统设置/自定义页面模块管理	57
第 7 章 功能模块测试	59
7.1 系统测试概述	59
7.2 基本模块测试	59
7.3 系统测试总结	64
第 8 章 总结	66
致 谢	67
参考文献	68
附 录	69
附录 A: 主要源代码	69
附录 B: 系统使用说明书	86

第1章 绪论

1.1 基于 B/C2C 模式的校园电商网站系统的开发背景

目前,随着互联网的普及和网络购物的兴起,网购已成为一种时尚和新型消费形式。网购已成为人们生活中不可或缺的一部分,其中在校大学生已然成为网购的主力军之一。然后随之而来的一些网络购物弊端也逐渐显现出来,如:送货周期长、商品质量难以保障、售后困难等。因此开发一个“一对一”的大学网络平台就能更好的为大学生提供在线购物等服务。系统可以为每一所大学创建购物页面,大学生可以通过大学管理员将自己的商品发布到所在学校的购物网上,也可以通过网站选购自己心仪的商品。大学管理员根据每天订单,将商品派送到消费者手中,用户可以先验货然后再付现金交易。这样就能很好地缩短购物周期,也能让用户购物更有保障,比传统网购更具有竞争力和吸引力。

1.2 国内外的发展现状和发展前景

随着市场需求的不断增大,各种大型购物网站不断出现在人们的视野中,从国内的“淘宝”到“拍拍”,从“京东商城”到“国美在线”,还有国外崛起的巨头亚马逊。但是对于在校学生来说,却缺少一个能够满足商品应需,配送及时,售后方便,和学生自主发布二手商品的需求的平台。淘宝,亚马逊等传统电商,都有配送慢,退换货等维权操作麻烦等问题。

大学生是一个拥有人口数量众多、高密度等特性的群体,购买力集中,对部分商品需求量大,容易受广告影响,宣传途径广且投入较小,同时网上购物的大学生也越来越多,这也让我们看到了巨大的市场市场潜力。

经过一段时间的市场调查,我们发现很多学校都还没有类似本系统的网站,或者还没有像本项目所提供的同校快速购物等服务。

所以本系统的主要目标是建立一个全新的大学校园综合性平台以及一个快速、安全购物的新模式。我们在向每一所开通的校园提供在线购物、商品寄卖、二手交易等主要服务的同时,我们希望建立一个新的校园购物模式,吸引更多的学生、创业者、商家加入平台。同时解决传统网购周期长、商品质量参差不齐、售后难等问题^[1]。

1.3 本文的组织结构

本文从实际出发,在分析了 B/C2C 领域国内外的市场现状及发展前景之后,

基于市场和项目做了详尽务实的需求分析，提取出用户需求和项目相对于该领域已有产品的优势。

基于市场分析得出的结论，对本系统前后台的基本功能模块，对应于 B2C 和 C2C 所需要实现的功能，做了基础划分，并对涉及到的相关技术做了介绍。然后，本文详细介绍了系统数据库的设计，并依模块划分对各个功能实现的方式和思路做了详细说明。

系统初步完成后，作者对系统从运行正确、权限控制等角度设计了测试用例，执行和记录了相关测试情况，并提供了详细的系统搭建和运行说明。

最后，本文对该系统做出了整体的总结，并向本文设计和实施的过程中提供了帮助的人和组织致谢。

第2章 校园电商系统需求分析

2.1 用户需求分析

2.1.1 市场优势

1、客户群体

大学生是一个数量多且相对集中的消费人群，对部分特定商品需求量巨大，同时随着电脑的普及，网购已成为众多大学生的一种购物习惯。据相关资料显示目前国内在校大学生人数已超过2000万，如此庞大的客户群体将是我们本项目的主要推广对象。

2.1.2 项目优势

1、价格：

本项目的主要目的就是为在校大学生提供优质廉价的商品，拒绝暴利。因此我们会在保证相同质量的前提下确保网站自营商品和大学生寄卖商品等价格低于淘宝、拍拍等一类的电商网站。

传统的网购价格组成：商品销售价+运费A。

本项目购物价格组成：商品销售价+运费B。

要求：运费B不大于运费A，我们将采用批量进货然后单独销售的原则，这样可以降低我们商品的单个运费，这样我们就有了一定的利润空间，同时消费者也可以花更少的钱买到自己心仪的商品。

2、服务：

由于我们主要针对校园市场，所以我们将每个学校建立一个独立的管理和营销团队，同时根据计划向每个学校派发一定的商品以满足库存需求，这样我们就可以保证商品24小时内送达。此外我们提供验货付款、现金交易的方式，这样更加安全放心。

3、附加：

商品寄卖：商家、代理商、在校学生可以直接向我们提供商品，我们可以根据情况向特定学校或者所有学校进行商品出售，一旦商品交易成功我们将收取一定的费用。

商品展示：在校大学生或者其他商家也可以在注册学校开设自己的校园店铺，他们可以在上面进行商品展示和广告宣传。

二手交易：我们将为每一所已开通的大学提供一个二手交易平台，该校注册会员可以发布和查询各种二手商品信息。

4、人力资源：

由于本项目主要面向大学校园，除开核心开发和管理人员外，一所大学仅需要一名管理和一些负责推广和派单的兼职人员，所以本项目所需人力资源不多，同时也可以为更多的在校大学生提供锻炼和兼职的机会。

2.2 系统基本功能模块分析

以前大多数商家出售商品都是使用实际店铺的销售方式，随着商店光顾的次数增多，经营业务也不断得增加，商店的管理工作就变得很繁重^[2]。当电子商务开始盛行的今天，越来越多的商家愿意把商店开到网上，这样做既可以减少房租等实际店铺带来的开销，可以增加商店的知名度以及增加人流量，同时还可以对商品进行网上管理，大量减少人力和物力的投入。因此，开发出一个功能强大，通用性强，适合我国消费水平的购物网站系统是很有必要的。

本系统的主要功能包括前台和后台两个部分。

前台主要功能包括分栏目分类商品展示、购物车、订单管理、注册、登录及用户积分系统，如图2.1所示：

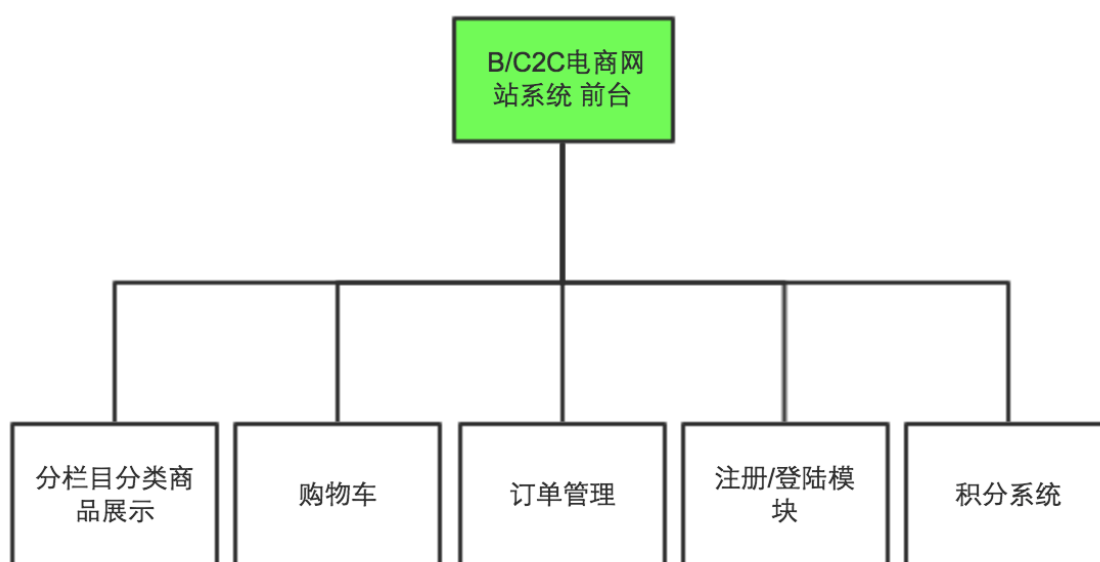


图 2.1 系统前台模块划分图

后台主要功能包括用户管理、自定义模块管理、学校团队管理、商品和库存管理、订单管理、权限管理，如图2.2所示：

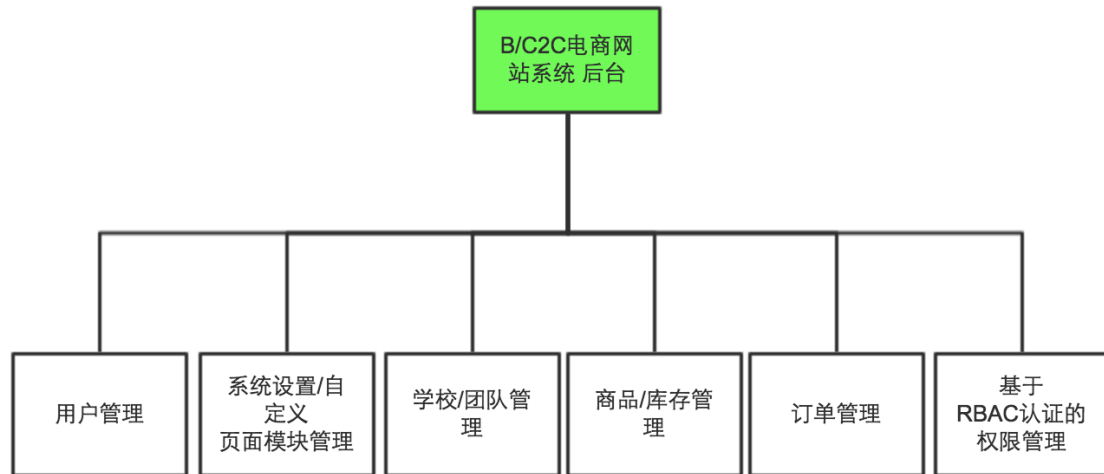


图 2.2 系统后台模块划分图

2.2.1 前台模块

1、注册登录模块

实现用户的注册登录是实现一个电商网站系统的第一步，注册登录流程应当做好数据正确性验证，使用验证码防止机器注册和暴力登录，用户也应当能够修改自己的密码。

2、分栏目分类商品展示

商家/用户可在后台配置多层次分类目录用于前台展示。电商系统的一大特点就是商品分类和层级都相当繁多，所以分类系统应该具备无限级别分类的能力。

3、购物车

购物车是电商系统的必备组件，在购物车的整个流程中，应当实现用户未登录也可以添加商品到购物车，登录后自动同步到登录账户，购物车页面应当能自动监测商品库存、商品区域等商品信息，当商品库存不足时，能够给予用户适当的反馈。购物车流程里也应当能够修改保存收货信息。

4、订单管理

订单管理页面需要实现查看订单状态的功能，管理员处理订单后，用户能够看到订单状态的变化。

5、积分系统

用户的各种行为可以积累积分，如购买/登陆等。例如，每次购买增加 10 分，每次订单金额 1 元可以获取 1 分。

2.2.2 后台模块

1、学校/团队管理

超级管理员可在后台添加区域/学校，并分配管理员账户，设置用户权限，管理所有商品、订单和系统设置。

2、用户管理

用户管理系统应当具备增删查改账户信息的能力，同时，也需要能够封禁账户。对于不同的账户，应当可以为他们各自配置权限。

3、商品/库存管理

管理商品详情和库存，可以方便地增删商品，修改商品信息，当商品库存为 0 时，前台将显示没有库存。

4、订单管理

订单管理系统应当具备增删查改订单信息的能力，能够更改订单状态，各个状态的订单可以分开管理。

5、基于 RBAC 认证的全局权限管理

系统采用耦合度低权限细分粒度高易使用的 RBAC 认证方式进行全局的权限控制，为权限管理提供节点和可判断的 url 规则，用户属于某个用户组。针对用户组授权后，在权限管理的用户授权里在当前的分组选项里输入用户 id 可加入当前分组。

6、系统设置/自定义页面模块管理

插件化的自定义模块，方便在页面中插入广告信息，新增页面模块等操作。如在页面头部或底部定义好了各种钩子，需要增加模块时，只需要定义好模块，并添加到某个钩子上，页面钩子就会自动加载该模块，使得页面耦合度低，扩展性强，维护方便。

第3章 系统开发技术的介绍

3.1 PHP 编程语言介绍

PHP 是一种开源的通用脚本编程语言，跨平台而且性能优越。相对于 JSP，性能消耗小很多，相对于 ASP，由于免费开源，不需要支付授权费用^[3]。是本系统的主要开发语言。PHP 可以和很多免费的开源项目相结合，比如 WAMP 和 LNMP，本项目采用 LNMP (Linux/Nginx/MySQL/PHP) 作为运行平台，皆是开源，稳定，应用广泛，性能优越的工具。PHP 可以和数十种数据库结合使用，当 MySQL 遇到应用或性能瓶颈时，可以方便地换用其它数据库，使得系统扩展容易，架构灵活。

相对于 C++/Java 等语言，PHP 语法非常简单，容易上手，有成熟的上下游开发工具。PHP 支持目前主流的 WEB 开发技术。如 RESTful, Ajax, XML 等，有成熟的面相对象体系，适合开发大型项目。国内外很多大型的系统和应用都是使用 PHP，如淘宝网，Wordpress，维基百科等，有强大的社区支持。

3.2 MySQL 数据库介绍

MySQL 是本系统使用的数据库系统。

它采用多线程架构，支持 1、2、3、4、和 8 字节长度自有符号 / 无符号整数、FLOAT、DOUBLE、CHAR、VARCHAR、TEXT、BLOB、DATE、TIME、DATETIME、TIMESTAMP、YEAR、和 ENUM 等常用的^[4]数据类型。并且全面支持 ANSI SQL 标准，对于 PHP，有 ODBC 和 mysql 数据源框架支持。另外，MySQL 可以运行在所有主流操作系统上，可以很方便地迁移，灵活选用开发和生产环境。

MySQL 提供了众多存储引擎：

- InnoDB: OLTP 在线事务处理，行锁设计，支持外键，非锁定读，通过多版本并发控制 (MVCC) 来获得高并发性，实现了 SQL 标准的 4 种隔离级别，插入缓冲，二次写，自适应哈希索引，预读，使用聚集方式存储表中数据，每张表的数据都按逐渐的顺序存放，如果没有显式地指定主键，会生成一个 6 字节的 ROWID 作主键。
- MyISAM: 不支持事务/表锁，对一些 OLAP 在线分析处理操作速度快，可以使用 myisampack 来进行赫夫曼压缩，压缩后是只读的。
- NDB Cluster: share nothing 的集群架构。数据全部放在内存中，主键查找速度极快，复杂的连接操作很慢。
- Memory: 使用哈希索引，所有数据存放在内存里，适用于临时表。只支持表锁，并发性能较差，不支持 TEXT 和 BLOB。

- Archive: 只支持 INSERT 和 SELECT, 使用 zlib 算法压缩存储数据, 适合存储归档数据, 如日志。使用行锁来实现高并发的插入操作。
- Federated: 指向一个远处 MySQL 数据库服务器上的表。
- Maria: 设计目标是取代 MyISAM。缓存数据和索引文件, 行锁设计, 提供 MVCC 功能, 支持事务, 以及更好的 BLOB 处理性能。

3.3 Nginx 服务器介绍

Nginx 是一个轻量级, 支持超高并发的 HTTP 及反向代理服务器, 比起 Apache, Nginx 占用更少的内存和资源, 相对于 Apache 阻塞性的请求处理方式, Nginx 采用异步非阻塞的方式处理请求, 在高并发下能保持较低的系统资源消耗。

另外, Nginx 采用高度模块化设计, 有众多的开源模块可以使用。

3.4 PHP-FPM 介绍

PHP-FPM 全称 php fastcgi process manager, 用于连接 Nginx 和 PHP, 能够根据访问的压力动态调用 PHP cgi 进程, 相比原生的 FastCgi, 可以有效降低内存压力和性能消耗。

3.5 RESTFul 标准介绍

RESTFul 全称 Representational State Transfer, 简称 REST, 是一组 WEB 应用的通用设计原则和规范定义, 主要用于 Client 和 Server 交互, 基于这个规范, 应用接口可以更简洁明了和易用, 使用 HTTP 标准的 GET, POST, PUT, DELETE, 也明确定义了每个请求的类型, 可以更方便有效地进行权限和安全控制。本系统部分模块采用 RESTFul 的标准定义接口结构。

3.6 MVC 架构模式介绍

本系统采用三层架构设计实现, 即数据持久化模型 Model, 视图 View 和控制器 Controller。通过 MVC, 实现了界面, 应用逻辑和数据持久层的分离。

通过替换 Model 的数据源, 可以方便地在 Controller 层进行单元测试。通过分层, 使得三个模块都是解耦合, 可定制的, 相互之间接口清晰, 扩展方便。MVC 的架构原理和 PC 是类似的, 构建在一系列相互独立的组件的基础之上, 这样就可以轻而易举地用自己或第三方的实现替换某些组件, 比如路由系统, 中间层, 视图引擎, ORM 等。

第 4 章 数据库设计

4.1 数据流程图

数据主要分为用户、管理员、UCENTER 三大结构模块，三个模块之间有不同的数据关联和数据操作流程，如图 4.1 所示。

从账户角度来说，分为用户和管理员两种登录注册流程，如果登录注册失败，系统会返回错误信息。然后，用户可以添加商品到购物车、提交订单和发布二手商品。管理员账户涉及的数据流程较多，会管理用户、订单、商品、设置、学校等模块。而所有和用户帐号有关的操作，都通过接口和 UCENTER 交换数据。

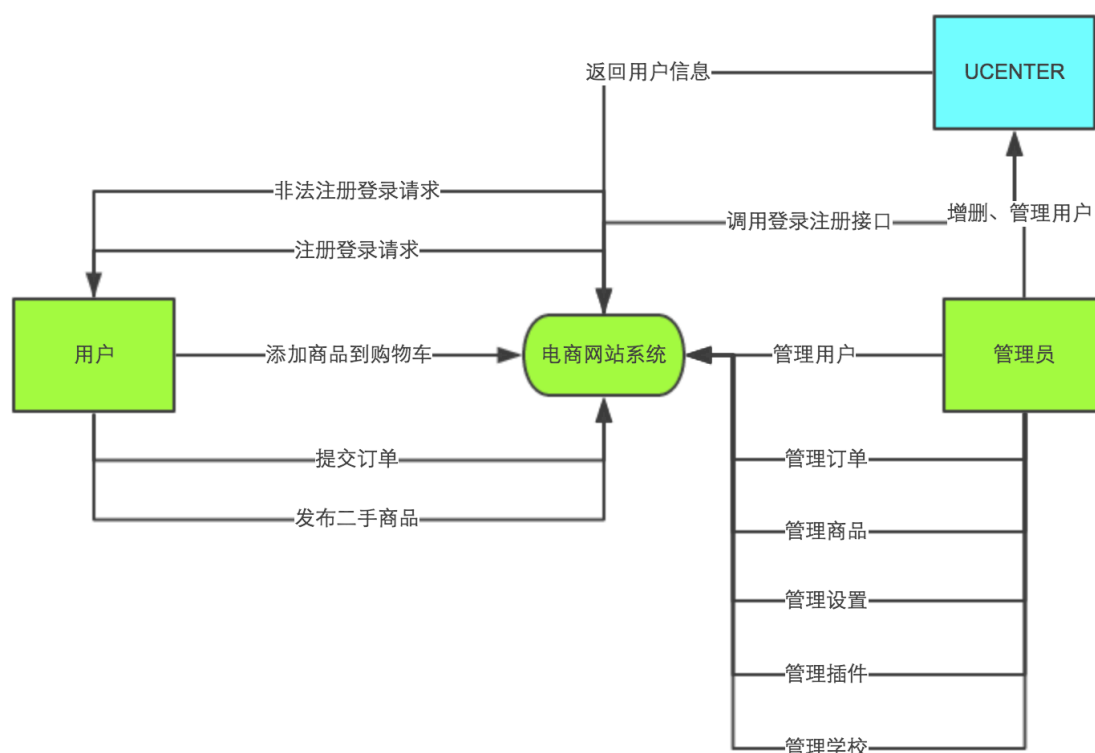


图 4.1 数据流程图

4.2 数据库表的设计分析

通过对该电商系统基本需求分析，市场需求分析，调研并确定了开发系统的编程语言和数据库及相应工具链，根据这些信息，进行了数据库逻辑设计，根据具体的业务需求，遵循数据库设计范式，将业务逻辑和数据抽象为相互关联且没有冗余的数据库表，并合理建立索引和联合索引，确保数据库性能到达业务要求。

另一方面，也要保证数据库系统具备良好的可扩展性，因此，数据库设计可明显看出模块性，如用户模块，商品模块，插件模块，用户行为模块等，最终确定，数据库有以下 26 张表：

1. shop_action 系统行为表
2. shop_action_log 行为日志表
3. shop_addons 插件表
4. shop_attachment 附件表
5. shop_auth_category_access 用户组与分类的对应关系表
6. shop_auth_group 用户组表
7. shop_auth_group_access 用户与用户组的对应关系表
8. shop_auth_rule 认证规则表
9. shop_cart 购物车表
10. shop_category 分类表
11. shop_channel 频道表
12. shop_config 配置表
13. shop_document 商品模型基础表
14. shop_document_model 商品模型表
15. shop_document_model_article 商品模型文章表
16. shop_document_model_download 商品模型下载表
17. shop_file 文件表
18. shop_hooks 系统钩子表
19. shop_member 会员表
20. shop_orderform 订单表
21. shop_picture 图片表
22. shop_ucenter_admin UCENTER 管理员表
23. shop_ucenter_app UCENTER 应用表
24. shop_ucenter_member UCENTER 用户表
25. shop_ucenter_setting UCENTER 设置表
26. shop_url 链接表

4.3 数据库表的详细设计

确定了表的分化之后,下一步就是对每张表做具体设计,在 MySQL 中建立表,并建立索引。

由于联合索引不便于直接在表结构表中表示出来,所以在下面的细节展示中,每张表分为结构表和索引表。

1、系统行为表结构与索引

用户注册、登录、购买、发布商品都是一种行为,通过行为定义,可以构建用户积分系统,在用户积分的基础上,就可以建立用户等级,开展积分抵换等促

销活动等，以增加用户粘性和活跃度。结构如表 4.1 所示，索引如表 4.1 所示。

表 4.1 系统行为表结构表

字段	类型	空	默认	注释
id (主键)	int(11)	否		主键
name	char(30)	否		行为唯一标识
title	char(80)	否		行为说明
remark	char(140)	否		行为描述
rule	Text	否		行为规则
status	tinyint(2)	否		状态 (-1: 已删除, 0: 禁用, 1: 正常)
update_time	int(11)	否	0	修改时间

表 4.2 系统行为表索引表

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
PRIMARY	BTREE	是	否	id	3	A	否	

2、行为日志表结构与索引

行为表定义了每一种行为，而用户的每一次行为都记录在行为日志表里，通过记录日志行为，可以进行用户行为分析，了解用户使用该系统的习惯，从而进行优化。结构如表 4.3 所示，索引如表 4.4 所示。

表 4.3 行为日志表结构表

字段	类型	空	默认	注释
id (主键)	int(10)	否		主键
action_id	int(10)	否		行为 id
user_id	int(10)	否		执行用户 id
action_ip	bigint(20)	否		执行行为者 ip
model	char(50)	否		触发行为的表
record_id	int(10)	否		触发行为的数据 id
create_time	int(10)	否		执行行为的时间

表 4.4 行为日志表索引表

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
----	----	----	----	----	----	------	---	----

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
PRIMARY	BTREE	是	否	id	48	A	否	
action_id_ix	BTREE	否	否	action_id		Ad	否	
user_id_ix	BTREE	否	否	user_id		A	否	
action_ip_ix	BTREE	否	否	action_ip		A	否	

3、插件表结构与索引

插件是本系统具备灵活性和扩展性的核心实现方式，插件通过钩子加载，每一种插件的信息在插件表做具体定义，包括插件信息、配置和是否启用等等。

结构如表 4.5 所示，索引如表 4.6 所示。

表 4.5 插件表结构表

字段	类型	空	默认	注释
id (主键)	int(10)	否		主键
name	varchar(40)	否		插件名或标识，区分大小写
title	varchar(20)	否		中文名
description	text	是	NULL	插件描述
status	tinyint(1)	否	1	状态 1-启用 0-禁用 -1-损坏
config	text	是	NULL	配置 序列化存放
author	varchar(40)	是	NULL	作者
version	varchar(20)	是	NULL	版本号
create_time	int(10)	否		安装时间
has_adminlist	tinyint(1)	否	0	1-有后台列表 0-无后台列表

表 4.6 插件表索引表

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
PRIMARY	BTREE	是	否	id	5	A	否	

4、附件表结构与索引

附件并非是电商系统的必须组件，但对于一些特殊的需求，如用户身份验证上传身份证等需求，也涉及到文件的上传和保存，附件表设计了用户与文件对应的映射结构。结构如表 4.7 所示，索引如表 4.8 所示。

表 4.7 附件表结构表

字段	类型	空	默认	注释
id (主键)	int(10)	否		
uid	int(10)	否	0	用户 ID
title	char(30)	否		附件显示名
type	tinyint(3)	否	0	附件类型 (0-目录, 1-外链, 2-文件)
source	int(10)	否	0	资源 ID (0-目录, 大于 0-当资源为文件时其值为 file_id, 当资源为外链时其值为 link_id)
record_id	int(10)	否	0	关联记录 ID
download	int(10)	否	0	下载次数
size	bigint(20)	否	0	附件大小 (当附件为目录或外链时, 该值为 0)
dir	int(12)	否	0	上级目录 ID (0-根目录)
sort	int(8)	否	0	排序
create_time	int(10)	否	0	创建时间
update_time	int(11)	否	0	更新时间
status	tinyint(1)	否	0	状态

表 4.8 附件表索引表

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
PRIMARY	BTREE	是	否	id	0	A	否	
idx_record_status	BTREE	否	否	record_id		A	否	
				status		A	否	

5、用户组与分类的对应关系表结构与索引

系统的权限管理系统由几部分元素组成：用户组，用户组可操作的商品分类，用户，该表为用户组与分类的映射关系。结构如表 4.9 所示，索引如表 4.10 所示。

表 4.9 用户组与分类的对应关系表结构表

字段	类型	空	默认	注释
group_id	mediumint(10)	否		用户 id
category_id	mediumint(8)	否		栏目 id

表 4.10 用户组与分类的对应关系表索引表

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
uid_group_id	BTREE	是	否	group_id		A	否	
				category_id	6	A	否	
uid	BTREE	否	否	group_id		A	否	
group_id	BTREE	否	否	category_id		A	否	

6、用户组表结构与索引

系统通过合理划分用户组来实现学校、不同层级管理员的划分，用户组表定义了用户组的基本信息。结构如表 4.11 所示，索引如表 4.12 所示。

表 4.11 用户组表结构表

字段	类型	空	默认	注释
id (主键)	mediumint(8)	否		用户组 id, 自增主键
module	varchar(20)	否		用户组所属模块
type	tinyint(4)	否		组类型
title	char(20)	否		用户组中文名称
description	varchar(80)	否		描述信息
status	tinyint(1)	否	1	用户组状态：为 1 正常，为 0 禁用，-1 为删除
rules	varchar(500)	否		用户组拥有的规则 id，多个规则，隔开

表 4.12 用户组表索引表

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
PRIMARY	BTREE	是	否	id	3	A	否	

7、用户与用户组的对应关系表结构与索引

每一个用户都被划分到一个用户组，该表定义了用户与用户组的对应关系。

该表主要的特点是同时具备各个字段的索引和用户 ID-用户组 ID 的联合索引，作为用户系统的核心关系表，必须保证最优的速度和性能。结构如表 4.13 所示，索引如表 4.14 所示。

表 4.13 用户与用户组的对应关系表结构表

字段	类型	空	默认	注释
uid	int(10)	否		用户 id
group_id	mediumint(8)	否		用户组 id

表 4.14 用户与用户组的对应关系表索引表

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
uid_group_id	BTREE	是	否	uid		A	否	
				group_id	14	A	否	
uid	BTREE	否	否	uid		A	否	
group_id	BTREE	否	否	group_id		A	否	

8、认证规则表表结构与索引

系统的权限认证系统采用 RBAC 实现，RBAC 主要基于认证节点和认证规则实现权限控制，该表定义了系统的认证规则。结构如表 4.15 所示，索引如表 4.16 所示。

表 4.15 认证规则表表结构表

字段	类型	空	默认	注释
id (主键)	mediumint(8)	否		规则 id, 自增主键
module	varchar(20)	否		规则所属 module
type	tinyint(2)	否	1	1-url;2-主菜单
name	char(80)	否		规则唯一英文标识
title	char(20)	否		规则中文描述
status	tinyint(1)	否	1	是否有效(0:无效,1:有效)
condition	varchar(300)	否		规则附加条件

表 4.16 认证规则表表索引表

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
PRIMARY	BTREE	是	否	id	84	A	否	
name	BTREE	是	否	module		A	否	
				name		A	否	
				type	84	A	否	

9、购物车表结构与索引

购物车是电商系统的必备组件，该表用于存储用户添加到购物车里的商品 id，通过该表在用户-商品之间建立关联。

通过商品数量字段，在结账时可以智能处理商品库存情况。

表 4.17 购物车表结构表

字段	类型	空	默认	注释
id (主键)	int(10)	否		
uid	int(10)	否		用户 ID
gid	int(10)	否		商品 ID
number	int(3)	否		商品数量
time	int(10)	否		添加时间
status	int(1)	否	1	

表 4.18 购物车表索引表

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
PRIMARY	BTREE	是	否	id	3	A	否	

10、分类表结构与索引

多层级的分类系统是电商网站的必备模块，并且，每个分类还可能具备不同的前端展示页面，因此，分类表的设计较为复杂但功能强大。

通过上级分类、二手商品等字段实现了多级分类和二手市场，通过各个页模板字段，可以实现不同分类拥有不同的前端展示效果。

结构如表 4.19 所示，索引如表 4.20 所示。

表 4.19 分类表结构表

字段	类型	空	默认	注释
id (主键)	int(10)	否		分类 ID
name	varchar(16)	否		标识
title	varchar(50)	否		标题
pid	int(10)	否	0	上级分类 ID
secondhand	int(1)	否	0	是否为二手商品
sort	int(10)	否	0	排序（同级有效）
list_row	tinyint(3)	否		列表每页行数
meta_title	varchar(50)	否		SEO 的网页标题
keywords	varchar(255)	否		关键字
description	varchar(255)	否		描述

字段	类型	空	默认	注释
template_index	varchar(100)	否		频道页模板
template_lists	varchar(100)	否		列表页模板
template_detail	varchar(100)	否		详情页模板
template_edit	varchar(100)	否		编辑页模板
model	varchar(100)	否		关联模型
type	varchar(100)	否		允许发布的内容类型
link_id	int(10)	否	0	外链（0-非外链，大于 0-外链 ID）
allow_publish	tinyint(3)	否	0	是否允许发布内容（0-不允许，1-允许）
display	tinyint(3)	否	0	可见性（0-所有人可见，1-管理员可见，2-不可见）
reply	tinyint(3)	否	0	是否允许回复
reply_model	varchar(100)	否		
reply_type	varchar(100)	否		
extend	text	否		扩展设置（JSON 数据）
create_time	int(10)	否	0	创建时间
update_time	int(10)	否	0	更新时间
status	tinyint(4)	否	0	数据状态（-1-删除，0-禁用，1-正常，2-待审核）

表 4.20 分类表索引表

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
PRIMARY	BTREE	是	否	id	5	A	否	
uk_name	BTREE	是	否	name	5	A	否	

11、频道表结构与索引

频道是高于分类的另一种分类，如数码产品频道，服装频道等，通过结合使用频道和分类，可以实现更为强大和灵活的分类系统。结构如表 4.21 所示，索引如表 4.22 所示。

表 4.21 频道表结构表

字段	类型	空	默认	注释
----	----	---	----	----

字段	类型	空	默认	注释
id (主键)	int(10)	否		频道 ID
pid	int(10)	否	0	上级频道 ID
title	char(30)	否		频道标题
url	char(100)	否		频道连接
sort	int(10)	否	0	导航排序
create_time	int(10)	否	0	创建时间
update_time	int(10)	否	0	更新时间
status	tinyint(4)	否	0	状态

表 4.22 频道表索引表

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
PRIMARY	BTREE	是	否	id	3	A	否	

12、配置表结构与索引

配置表是系统易于使用和维护的关键，通过将系统可能变动和经常变动的数据存储到配置表，可以很方便地控制、修改系统行为。结构如表 4.23 所示，索引如表 4.24 所示。

表 4.23 配置表结构表

字段	类型	空	默认	注释
id (主键)	int(10)	否		配置 ID
name	varchar(30)	否		配置名称
type	tinyint(3)	否	0	配置类型（0-数字，1-字符，2-文本，3-数组，4-枚举，5-多选）
title	varchar(50)	否		配置说明
group	tinyint(3)	否	0	配置分组（0-无分组，1-基本设置）
extra	varchar(255)	否		配置值
remark	varchar(100)	否		配置说明
create_time	int(10)	否	0	创建时间
update_time	int(10)	否	0	更新时间
status	tinyint(4)	否	0	状态

字段	类型	空	默认	注释
value	text	否		
sort	smallint(3)	否		

表 4.24 配置表索引表

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
PRIMARY	BTREE	是	否	id	16	A	否	
uk_name	BTREE	是	否	name	16	A	否	

13、商品模型基础表结构与索引

每一件商品，都具有大量的属性，如：

基础信息：名称、大小、颜色、售价等；

统计信息：库存量、浏览量、销售量等；

详细介绍：富文本的商品介绍、买家反馈等。

商品模型基础表用于保存这些信息。

结构如表 4.25 所示，索引如表 4.26 所示。

表 4.25 文档模型基础表结构表

字段	类型	空	默认	注释
id (主键)	int(10)	否		文档 ID
uid	int(10)	否	0	用户 ID
sid	int(11)	否		
name	char(40)	否		标识
title	char(80)	否		标题
price	varchar(10)	否	0	
stock	int(11)	否	0	库存
sold	int(10)	否	0	销售量
category_id	int(10)	否		所属分类
description	char(140)	否		描述
pid	int(10)	否	0	所属 ID
model_id	tinyint(3)	否	0	内容模型 ID

字段	类型	空	默认	注释
type	tinyint(3)	否	0	内容类型（0-专辑，1-目录，2-主题，3-段落）
position	smallint(5)	否	0	推荐位（1-列表推荐，2-频道页推荐，4-首页推荐，[同时推荐多个地方相加即可]）
link_id	int(10)	否	0	外链（0-非外链，大于 0-外链 ID）
cover_id	int(10)	否	0	封面（0-无封面，大于 0-封面图片 ID）
display	tinyint(3)	否	0	可见性（0-不可见，1-所有人可见）
dateline	int(10)	否	0	截至时间（0-永久有效）
attach	tinyint(3)	否	0	附件数量
view	int(10)	否	0	浏览量
comment	int(10)	否	0	评论数
extend	int(10)	否	0	扩展统计字段，根据需求自行使用
level	int(10)	否	0	优先级（越高排序越靠前）
create_time	int(10)	否	0	创建时间
update_time	int(10)	否	0	更新时间
status	tinyint(4)	否	0	数据状态（-1-删除，0-禁用，1-正常，2-待审核，3-草稿）

表 4.26 文档模型基础表索引表

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
PRIMARY	BTREE	是	否	id	8	A	否	
idx_name	BTREE	否	否	name		A	否	
idx_category_status	BTREE	否	否	category_id		A	否	
				status		A	否	
idx_status_type_pid	BTREE	否	否	status		A	否	
				type		A	否	
				pid		A	否	

14、商品模型表结构与索引

不同的商品，虽然描述细节各不相同，但同类商品通常具有相同的基础属性结构，因此将它们抽象为不同的模型，通过划分模型基础表和模型表，来实现重复信息的抽象。结构如表 4.27 所示，索引如表 4.28 所示。

表 4.27 文档模型表结构表

字段	类型	空	默认	注释
id (主键)	int(10)	否		模型 ID
name	char(16)	否		模型标识
title	char(16)	否		模型名称
create_time	int(10)	否	0	创建时间
update_time	int(10)	否	0	更新时间
status	tinyint(3)	否	0	状态

表 4.28 文档模型表索引表

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
PRIMARY	BTREE	是	否	id	3	A	否	

15、商品模型文章表结构与索引

由于商品介绍详情文章通常较大,所以单独存储在文章表,极大减小商品表的大小,以提高系统性能。结构如表 4.29 所示,索引如表 4.30 所示。

表 4.29 文档模型文章表结构表

字段	类型	空	默认	注释
id (主键)	int(10)	否	0	文档 ID
parse	tinyint(3)	否	0	内容解析类型 (0-html, 1-ubb, 2-markdown)
content	text	否		文章内容
template	varchar(100)	否		详情页显示模板

表 4.30 文档模型文章表索引表

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
PRIMARY	BTREE	是	否	id	15	A	否	

16、商品模型下载表结构与索引

商品可以具备一些附加文件,该表用以存储商品和文件的映射关系。结构如表 4.31 所示,索引如表 4.32 所示。

表 4.31 文档模型下载表结构表

字段	类型	空	默认	注释
id (主键)	int(10)	否	0	文档 ID
parse	tinyint(3)	否	0	内容解析类型 (0-html, 1-ubb, 2-markdown)
content	text	否		下载详细描述
template	varchar(100)	否		详情页显示模板
file_id	int(10)	否	0	文件 ID
download	int(10)	否	0	下载次数
size	bigint(20)	否	0	文件大小

表 4.32 文档模型下载表索引表

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
PRIMARY	BTREE	是	否	id	0	A	否	

17、文件表结构与索引

文件表用于存储附件的具体信息。结构如表 4.33 所示，索引如表 4.34 所示。

表 4.33 文件表结构表

字段	类型	空	默认	注释
id (主键)	int(10)	否		文件 ID
name	char(30)	否		原始文件名
savename	char(20)	否		保存名称
savepath	char(30)	否		文件保存路径
ext	char(5)	否		文件后缀
mime	char(40)	否		文件 mime 类型
size	int(10)	否	0	文件大小
md5	char(32)	否		文件 md5
sha1	char(40)	否		文件 sha1 编码
location	tinyint(3)	否	0	文件保存位置 (0-本地, 1-FTP)
create_time	int(10)	否		上传时间

表 4.34 文件表索引表

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
----	----	----	----	----	----	------	---	----

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
PRIMARY	BTREE	是	否	id	0	A	否	
uk_md5	BTREE	是	否	md5	0	A	否	

18、系统钩子表结构与索引

系统通过写在系统各处的钩子加载插件，如开始执行、登录、页面渲染等节点，钩子表记录了每个钩子要加载的插件。结构如表 4.35 所示，索引如表 4.36 所示。

表 4.35 系统钩子表结构表

字段	类型	空	默认	注释
id (主键)	int(10)	否		主键
name	varchar(40)	否		钩子名称
description	text	是	NULL	描述
type	tinyint(1)	否	1	1-Controller 2-Widget
update_time	int(10)	是	0	更新时间
addons	varchar(255)	是	NULL	钩子挂载的插件 ' , ' 分割

表 4.36 系统钩子表索引表

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
PRIMARY	BTREE	是	否	id	11	A	否	
搜索索引	BTREE	是	否	name	11	A	否	

19、会员表结构与索引

电商系统除了商品，最重要的就是用户了。

会员表记录了用户基础信息、积分、登录历史等信息。

这些信息可以用来分析该用户的使用习惯，通过性别等信息也可以进行智能化推荐。

结构如表 4.37 所示，索引如表 4.38 所示。

表 4.37 会员表结构表

字段	类型	空	默认	注释
uid (主键)	int(10)	否	0	用户 ID
nickname	char(16)	否		昵称

字段	类型	空	默认	注释
sex	tinyint(3)	否	0	性别（0-女，1-男）
birthday	date	否	0000-00-00	生日
qq	char(10)	否		
score	mediumint(8)	否	0	用户积分
login	int(10)	否	0	登录次数
reg_ip	bigint(20)	否	0	注册 IP
reg_time	int(10)	否	0	注册时间
last_login_ip	bigint(20)	否	0	最后登录 IP
last_login_time	int(10)	否	0	最后登录时间
status	tinyint(4)	否	0	会员状态

表 4.38 会员表索引表

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
PRIMARY	BTREE	是	否	uid	8	A	否	

20、订单表结构与索引

订单表存储每一个订单对应的用户、商品、金额等信息。

这样，当商品售价改变时，也不会影响到已经生成的订单。

结构如表 4.39 所示，索引如表 4.40 所示。

表 4.39 订单表结构表

字段	类型	空	默认	注释
id (主键)	int(10)	否		
uid	int(10)	否		
number	varchar(20)	否		
goods	varchar(1000)	否		
address	varchar(1000)	否		
money	varchar(10)	否		
status	int(1)	否	1	
time	int(10)	否		

表 4.40 订单表索引表

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
PRIMARY	BTREE	是	否	id	11	A	否	

21、图片表结构与索引

系统的 LOGO, 广告系统, 前端图片等多处需要可配置上传的图片, 图片表用以记录这些图片的信息, 以便方便地管理这些图片。结构如表 4.41 所示, 索引如表 4.42 所示。

表 4.41 图片表结构表

字段	类型	空	默认	注释
id (主键)	int(10)	否		主键 id 自增
path	varchar(255)	否		路径
url	varchar(255)	否		图片链接
md5	char(32)	否		文件 md5
sha1	char(40)	否		文件 sha1 编码
status	tinyint(1)	否	0	状态
create_time	int(10)	否	0	创建时间

表 4.42 图片表索引表

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
PRIMARY	BTREE	是	否	id	11	A	否	

22、UCENTER 管理员表结构与索引

系统实现了在 Discuz 等成熟系统常用的用户中心模式, 在本系统里, 电商网站只是接入 UCENTER 的一个应用, 未来其它的系统都可以通过统一的方式接入, 实现用户信息的中心化管理。

管理员表保存用户中心的管理员帐号信息。

结构如表 4.43 所示, 索引如表 4.44 所示。

表 4.43 UCENTER 管理员表结构表

字段	类型	空	默认	注释
id (主键)	int(10)	否		管理员 ID
member_id	int(10)	否	0	管理员用户 ID
status	tinyint(3)	否	0	管理员状态

表 4.44 UCENTER 管理员表索引表

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
PRIMARY	BTREE	是	否	id	0	A	否	

23、UCENTER 应用表结构与索引

如上所述，电商系统管理是接入用户中心的一个应用，每个应用有自己的加密 key，实现不同系统的安全隔离。结构如表 4.45 所示，索引如表 4.46 所示。

表 4.45 UCENTER 应用表结构表

字段	类型	空	默认	注释
id (主键)	int(10)	否		应用 ID
title	varchar(30)	否		应用名称
url	varchar(100)	否		应用 URL
ip	char(15)	否		应用 IP
auth_key	varchar(100)	否		加密 KEY
sys_login	tinyint(1)	否	0	同步登陆
allow_ip	varchar(255)	否		允许访问的 IP
create_time	int(10)	否	0	创建时间
update_time	int(10)	否	0	更新时间
status	tinyint(4)	否	0	应用状态

表 4.46 UCENTER 应用表索引表

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
PRIMARY	BTREE	是	否	id	0	A	否	
status	BTREE	否	否	status		A	否	

24、UCENTER 用户表结构与索引

用户表存储用户的基础信息，应用通过接口进行登录、注册等操作。每个应用特有的用户扩展信息存储在应用端的用户信息表，通过 id 与本表关联。结构如表 4.47 所示，索引如表 4.48 所示。

表 4.47 UCENTER 用户表结构表

字段	类型	空	默认	注释
----	----	---	----	----

字段	类型	空	默认	注释
uid (主键)	int(10)	否		用户 ID
username	char(16)	否		用户名
password	char(32)	否		密码
sid	int(3)	否		学校 id
qq	varchar(20)	否		
email	char(32)	否		用户邮箱
reg_phone	varchar(20)	否		
mobile	char(15)	否		用户手机
realname	char(20)	否		
address	varchar(200)	否		
reg_time	int(10)	否	0	注册时间
reg_ip	bigint(20)	否	0	注册 IP
last_login_time	int(10)	否	0	最后登录时间
last_login_ip	bigint(20)	否	0	最后登录 IP
update_time	int(10)	否	0	更新时间
status	tinyint(4)	是	0	用户状态

表 4.48 UCENTER 用户表索引表

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
PRIMARY	BTREE	是	否	uid	17	A	否	
username	BTREE	是	否	username	17	A	否	
email	BTREE	是	否	email	17	A	否	
status	BTREE	否	否	status		A	是	

25、UCENTER 设置表结构与索引

该表用于存储 UCENTER 的设置项。结构如表 4.49 所示，索引如表 4.50 所示。

表 4.49 UCENTER 设置表结构表

字段	类型	空	默认	注释
id (主键)	int(10)	否		设置 ID
type	tinyint(3)	否	0	配置类型 (1-用户配置)

字段	类型	空	默认	注释
value	text	否		配置数据

表 4.50 UCENTER 设置表索引表

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
PRIMARY	BTREE	是	否	id	0	A	否	

26、链接表结构与索引

链接表用以存储友情链接等链接信息，也可以用作商品的短链接分享实现。结构如表 4.51 所示，索引如表 4.52 所示。

表 4.51 链接表结构表

字段	类型	空	默认	注释
id (主键)	int(11)	否		链接唯一标识
url	char(255)	否		链接地址
short	char(100)	否		短网址
status	tinyint(2)	否		状态（-1：删除，0：禁用，1：正常，2：未审核）
create_time	int(10)	否		创建时间

表 4.52 链接表索引表

键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
PRIMARY	BTREE	是	否	id	0	A	否	
idx_url	BTREE	是	否	url	0	A	否	

第 5 章 系统功能模块设计分析

5.1 系统功能模块

注册/登录模块：用户可在所选的学校区域注册/登陆/购物/发布商品。

分栏目分类商品展示：商家/用户可在后台配置多层次分类目录用于前台展示。

购物车：全局的购物车功能，自动监测商品库存，商品区域，流畅的购物体验。

订单管理：查看订单状态。

后台：学校/团队管理：超级管理员可在后台添加区域/学校，并分配管理员账户。

后台：用户管理：管理注册用户的信息。

后台：商品/库存管理：管理商品详情和库存，可以方便地增删商品，修改商品信息，当商品库存为 0 时，前台将显示没有库存。

后台：订单管理：查看，更改订单状态。

基于 RBAC 认证的全局权限管理：系统采用耦合度低权限细分粒度高易使用的 RBAC 认证方式进行全局的权限控制，为权限管理提供节点和可判断的 url 规则，用户属于某个用户组。针对用户组授权后，在权限管理的用户授权里在当前的分组选项里输入用户 id 可加入当前分组。

系统设置/自定义页面模块管理：插件化的自定义模块，方便在页面中插入广告信息，新增页面模块等操作。如在页面头部或底部定义好了各种钩子，需要增加模块时，只需要定义好模块，并添加到某个钩子上，页面钩子就会自动加载该模块，使得页面耦合度低，扩展性强，维护方便。

积分系统：用户的各种行为可以积累积分，如购买/登陆等。

5.2 功能流程图

5.2.1 系统设置/自定义页面模块管理

系统分为前台和后台，也就对应了用户和管理员两类账户和功能流程，系统总流程如图 5.1 所示。

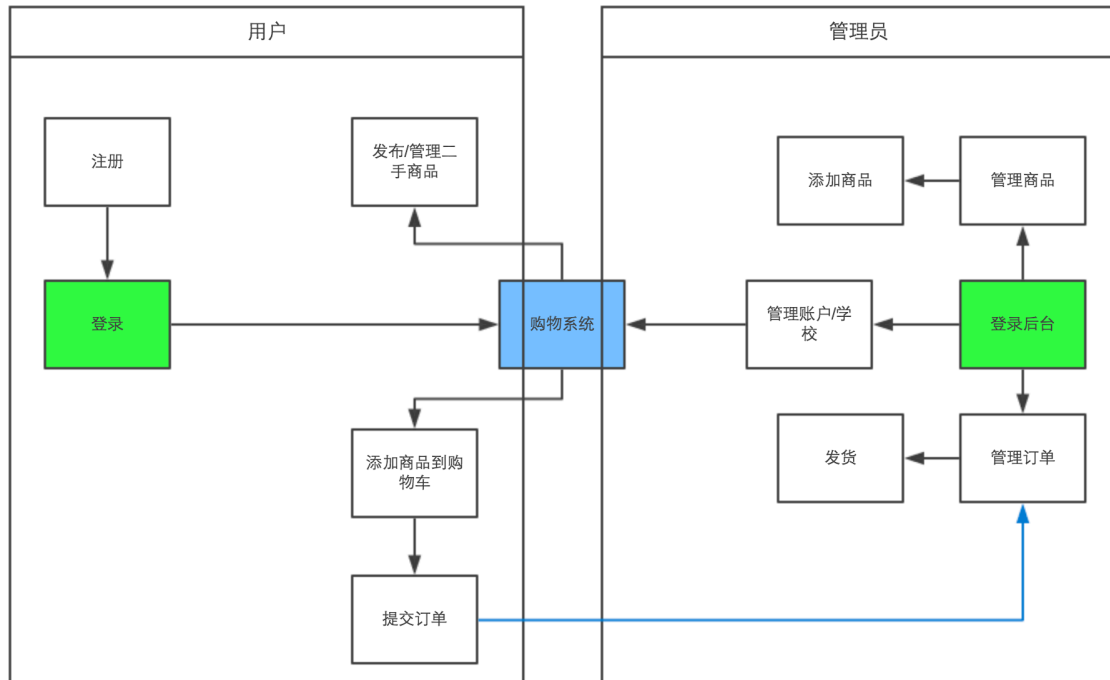


图 5.1 功能流程图

5.2.2 商品购买流程图

对于前台用户来说，电商网站最核心的操作就是购物，详细的购物流程如图 5.2 所示。

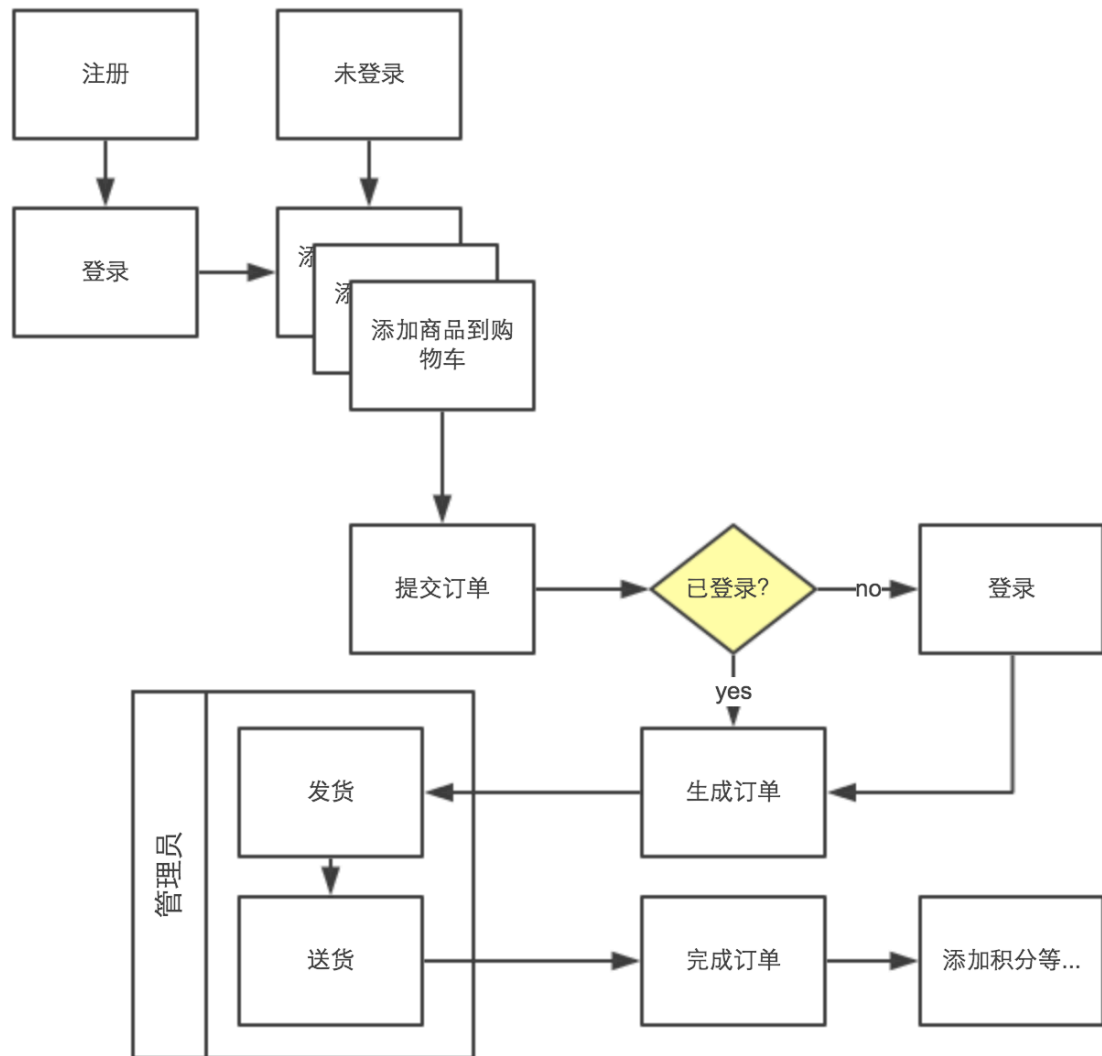


图 5.2 购物流程图

5.2.3 帐号、权限管理流程图

系统的学校、普通用户、管理员、二手市场均通过灵活的角色和权限管理实现，如图 5.3 所示，以实现普通用户发布二手商品为例，展示了管理员后台的操作流程。

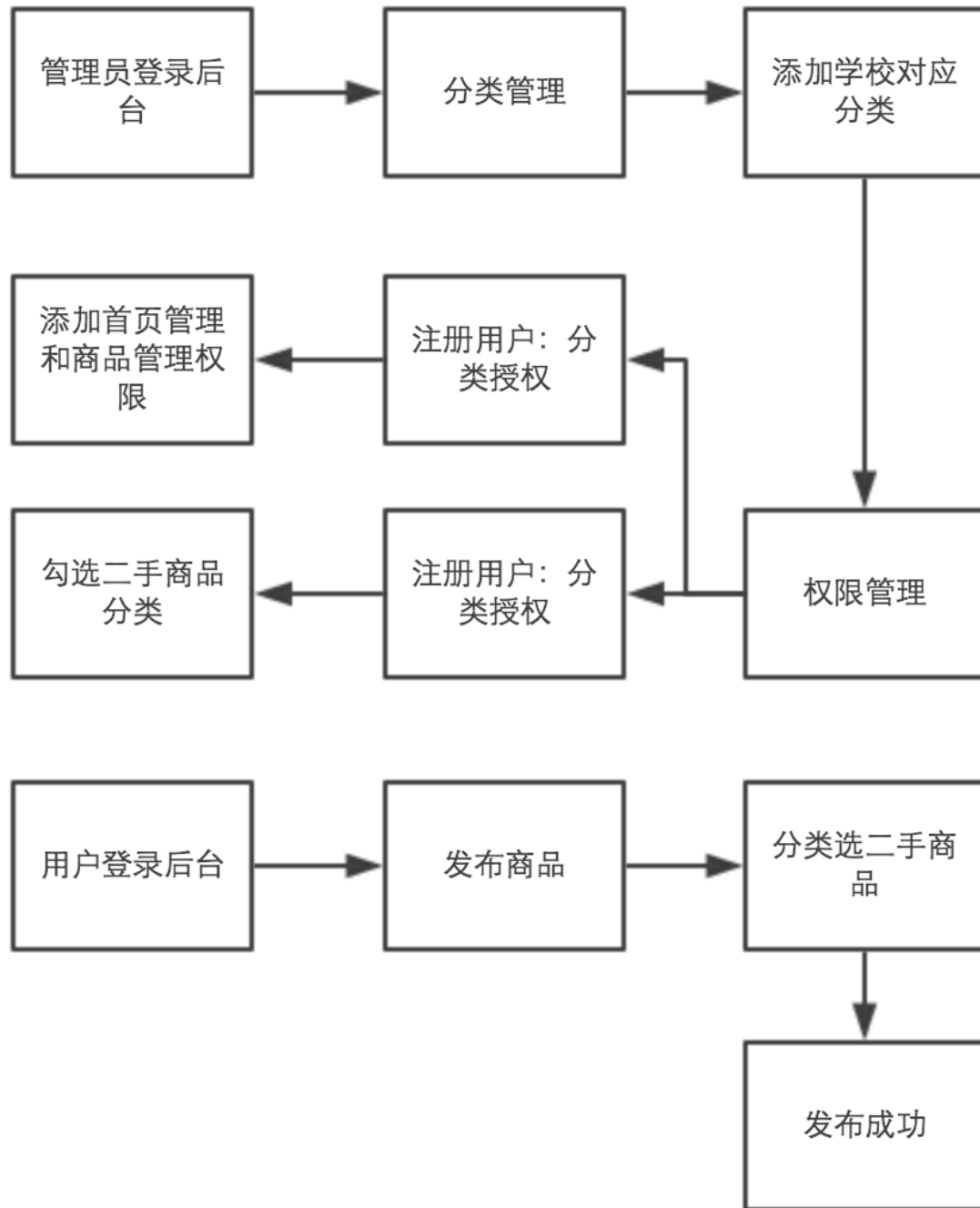


图 5.3 帐号、权限管理流程图

第 6 章 系统主要功能模块的具体实现

6.1 注册/登录模块接口及实现

系统设计了独立的用户中心 UCENTER，购物系统本身只是 UCENTER 中的一个应用，通过统一的用户接口对外部应用提供用户相关功能，实现了高度解耦合，方便系统拓展新的系统和模块。另外，独立的用户中心和统一的接口也便于实现安全验证，保证用户数据的安全性。

为了方便外层应用的使用，同时提交规范性和安全性，用户系统分装了最常用的登录注册验证及修改用户信息等接口。网站获取到用户提交的信息后，通过调用 UserApi 类的接口和用户系统交互。

UserApi 类 主要代码：

```
class UserApi extends Api{
    /**
     * 构造方法，实例化操作模型
     */
    protected function _init(){
        $this->model = new UcenterMemberModel();
    }
}
```

6.1.1 接口

1、注册接口

输入用户名，密码，邮箱，及手机号码等信息，进行注册。

注册成功返回用户信息，注册失败返回错误编号。

通过错误编号可以获取到对应的错误描述。

```
public function register(){
    return $this->model->register();
}
```

2、登录接口

输入用户名，用户密码和用户名类型登录。用户名类型默认为用户名。

支持用户名、邮箱和手机三种验证方式。

登录成功返回用户信息，登录失败返回错误编号。

通过错误编号可以获取到对应的错误描述。

```
public function login($username, $password, $type = 1){  
    return $this->model->login($username, $password, $type);  
}
```

3、获取用户信息接口

输入用户名或用户 ID 查询用户信息。默认使用用户 id。

```
public function info($uid, $is_username = false){  
    return $this->model->info($uid, $is_username);  
}
```

4、检测用户名接口

输入用户名查询用户名是否合法。

包括用户名格式，用户名是否存在等。

通过错误编号可以获取到对应的错误描述。

```
public function checkUsername($username){  
    return $this->model->checkField($username, 1);  
}
```

5、检测邮箱接口

输入邮箱查询邮箱是否合法。

通过错误编号可以获取到对应的错误描述。

```
public function checkEmail($email){  
    return $this->model->checkField($email, 2);  
}
```

6、检测手机接口

输入手机查询手机是否合法。

通过错误编号可以获取到对应的错误描述。

```
public function checkMobile($mobile){  
    return $this->model->checkField($mobile, 3);  
}
```

7、更新用户接口

输入新的用户信息及密码更新用户信息，需要密码进行验证。

返回 bool, true 修改成功, false 修改失败。


```

public function updateInfo($uid, $password, $data){
    if($this->model->updateUserFields($uid, $password, $data) !== false){
        $return['status'] = true;
        $return['info'] = "修改成功! ";
    }else{
        $return['status'] = false;
        $return['info'] = $this->model->getError();
    }
    return $return;
}

```

8、更新收货地址接口

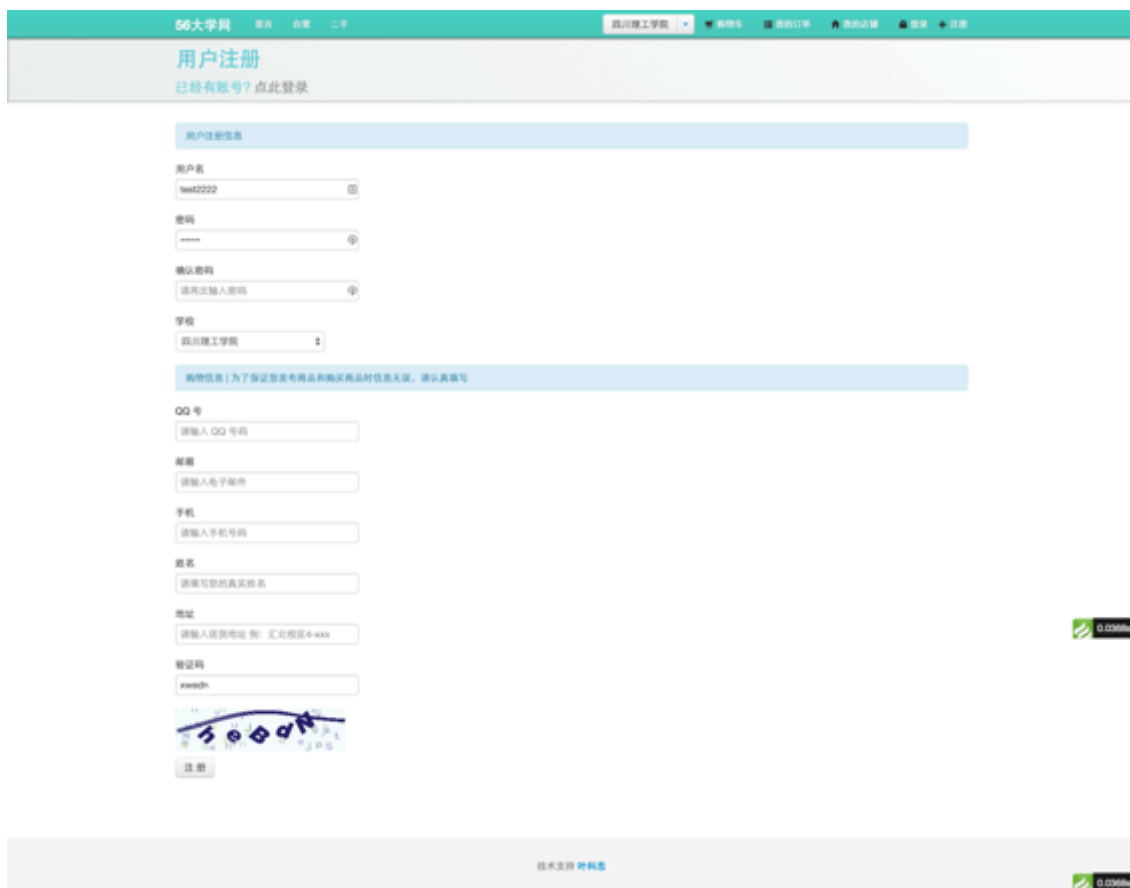
输入新的收货地址更新收货地址，返回 bool，true 修改成功，false 修改失败

```

public function updateAddress($uid, $data){
    if($this->model->updateUserAddress($uid, $data) !== false){
        $return['status'] = true;
        $return['info'] = "修改成功! ";
    }else{
        $return['status'] = false;
        $return['info'] = $this->model->getError();
    }
    return $return;
}

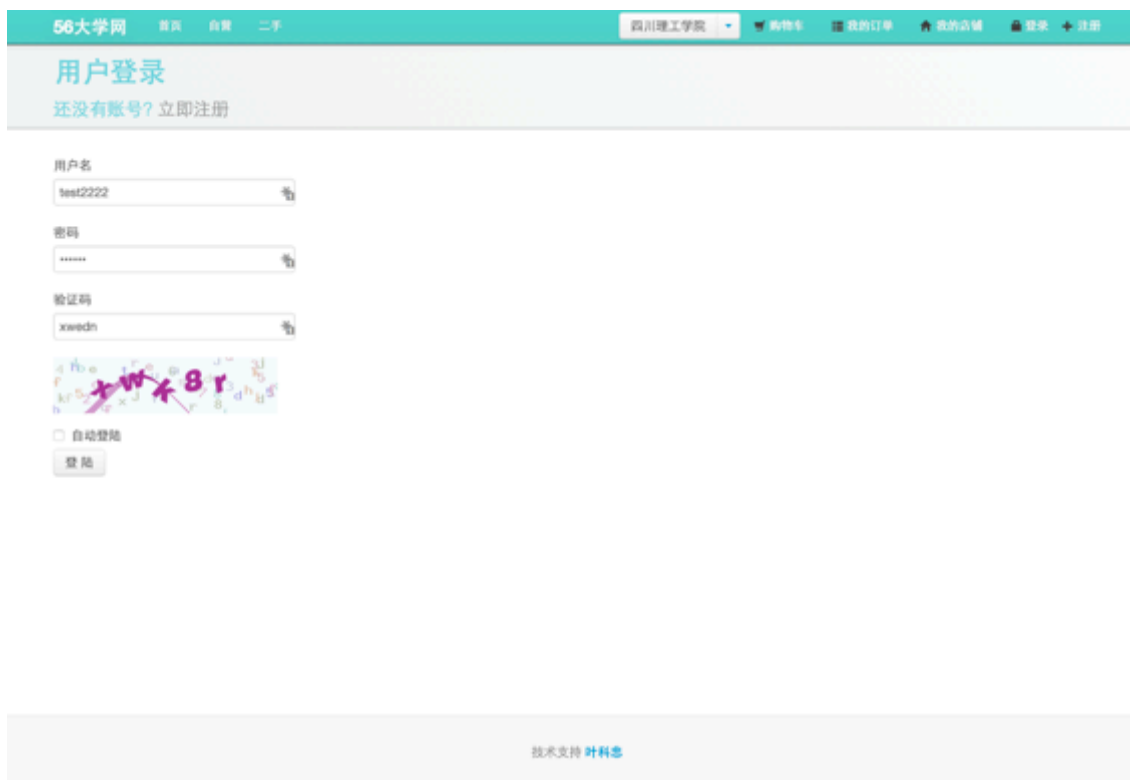
```

可见，UserApi 中实现了注册 register，登录 login，获取用户信息 info，数据检查：checkUsername, checkEmail, checkMobile, 更新信息 updateInfo, updateAddress 等接口，通过调用这些接口，实现了前端的登录和注册页面。注册界面如图 6.1 所示，登录界面如图 6.2 所示。



The registration interface features a teal header with the site name '56大学网' and navigation links. Below the header, a light blue banner displays '用户注册' and a link for existing users. The main form area is titled '用户注册信息' and includes fields for username, password, confirm password, school, QQ number, email, phone, real name, address, and verification code. A security tip about password safety is provided. A CAPTCHA image is shown before the '注册' button. The footer contains technical support information and a version number '0.03909a'.

图 6.1 注册界面



The login interface has a teal header with the site name and navigation links. A light blue banner shows '用户登录' and a link for new users. The form area contains fields for username, password, and verification code. Below these fields is a CAPTCHA image, an '自动登陆' checkbox, and a '登陆' button. The footer includes technical support information and a version number '0.03909a'.

图 6.2 登录界面

用户注册时，需要选择所在学校，用户只能在注册学校发布/购买商品。不同的学校即是一个独立运营的平台，有独立的管理员和商品、用户。

6.1.2 实现

1、用户注册的实现

网站获取到用户提交的注册信息后，通过调用 UserApi 类的 register 接口和用户系统交互。UserApi 会调用用户中心模型 UcenterMemberModel 的 register 方法注册。register 方法首先会检查用户信息是否已经注册，验证通过后，通过权限管理类 AuthGroupAccess 写入 RBAC 验证规则。

注册成功，该方法会返回 uid，注册失败会返回错误编号，错误详情在模型自动验证里有标注。

```
//验证手机
if(empty($data['mobile'])) unset($data['mobile']);
/* 添加用户 */
if($this->create($data)){
    $uid = $this->add();
    if ($uid != 0) {
        $M = M('AuthGroupAccess');
        $auth = array(
            'uid' => $uid,
            'group_id' => 3, // 注册用户组
        );
        $M->create($auth);
        $M->add();
    }
    return $uid ? $uid : 0; //0-未知错误，大于 0-注册成功
} else {
    return $this->getError(); //错误详情见自动验证注释
}
```

2、用户登录的实现

网站获取到用户提交的注册信息后，通过调用 UserApi 类的 login 接口和用户系统交互。UserApi 会调用用户中心模型 UcenterMemberModel 的 login 方法登录。

login 方法首先会检查用户数据是否存在，然后通过 md5 加密算法比对密码，成功，该方法会返回 uid，登录失败会返回错误编号，-1 代表用户不存在，-2 代表密码错误。

```

/* 获取用户数据 */
$user = $this->where($map)->find();
if(is_array($user) && $user['status']){
    /* 验证用户密码 */
    if(think_ucenter_md5($password) === $user['password']){
        $this->updateLogin($user['uid']); //更新用户登录信息
        return $user['uid']; //登录成功，返回用户 ID
    } else {
        return -2; //密码错误
    }
} else {
    return -1; //用户不存在或被禁用
}

```

6.2 分栏目分类商品展示

6.2.1 无限级别分类的数据结构

系统从底层实现了无限级别分类，但为了避免系统太过复杂，给用户造成困惑，前端仅展示至第三层。

为了实现无限层级分类，在数据库分类表设置了上级分类 ID 字段，并仅以此作为分类间的关系约束，见表 6.1 所示。

然后通过程序分析分类间的关系，生成前台的分类。

6.2.2 无限级别分类生成分类树接口

该接口用于根据指定分类 ID 获取分类树，指定分类则返回指定分类及其子分类，不指定则返回所有分类树。

```

public function getTree($id = 0, $field = true){
    /* 获取当前分类信息 */
    if($id){
        $info = $this->info($id);
        $id = $info['id'];
    }
}

```

```

    }
    /* 获取所有分类 */
    $map = array('status' => 1);
    $list = $this->field($field)->where($map)->order('sort')->select();
    $list = list_to_tree($list, $pk = 'id', $pid = 'pid', $schild = '_', $root = $id);
    /* 获取返回数据 */
    if(isset($info)){ //指定分类则返回当前分类及其子分类
        $info['_'] = $list;
    } else { //否则返回所有分类
        $info = $list;
    }
    return $info;
}

```

6.2.3 列表转换成分类树接口

PHP 中最常用的数据集是数组，这个接口用户将普通数组数据转换为分类树的形式。

```

function list_to_tree($list, $pk='id', $pid = 'pid', $schild = '_child', $root = 0) {
    // 创建 Tree
    $tree = array();
    if(is_array($list)) {
        // 创建基于主键的数组引用
        $refer = array();
        foreach ($list as $key => $data) {
            $refer[$data[$pk]] =& $list[$key];
        }
        foreach ($list as $key => $data) {
            // 判断是否存在 parent
            $parentId = $data[$pid];
            if ($root == $parentId) {
                $tree[] =& $list[$key];
            } else {

```

```
        if (isset($refer[$parentId])) {  
            $parent =& $refer[$parentId];  
            $parent[$child][] =& $list[$key];  
        }  
    }  
}  
}  
}  
return $tree;  
}
```

管理员在后台系统设置>分类管理里，可以方便地增加编辑和删除分类，如图 6.3 所示。



图 6.3 后台分类管理界面

管理员在后台配置好分类后，前台将根据不同的学校展示不同的分类目录，如图 6.4 所示，分类用红框标出。



图 6.4 后台分类管理界面

6.3 购物车

全局的购物车功能，自动监测商品库存，商品区域，流畅的购物体验。

如图 6.5 所示，在商品详情页，用户选择商品数量后，点击加入购物车，可以将商品添加到购物车。

6.3.1 购物车的友好化设计



图 6.5 添加到购物车

当商品库存不足时，购物车会自动提示库存不足。

购物车保存的商品数据通过 session 保存，所以，即使用户没有登录，也可以将商品添加到购物车，登录之后自动将购物车的商品数据同步到登录账户下，对于用户体验的提升有很大的帮助。

同样，在购物车列表页，系统会自动检测商品库存情况，并通过 Ajax 技术给予用户友好的提醒，避免页面刷新后用户才得到错误信息，造成用户体验的下降，如图 6.6 所示。



图 6.6 购物车列表页

当商品有问题时，提交订单按钮会自动设置为不可点击状态。

6.3.2 购物车的 Ajax 实现

购物车的前端友好化功能通过 Javascript Ajax 技术实现。主要实现代码：

```
<script type="text/javascript">
```

```
$('#submitOrderform').click(function() {  
    $('#submitOrderform').button('loading');  
});
```

点击修改收货地址后，页面不会刷新，而是弹出对话框，用户可以在当前页面直接修改保存收货信息。

```
$('#showAddressModal').click(function() {  
    $('#address-form-box-success').hide();  
    $('#address-form-box').show();  
    $('#saveChange').show();  
});
```

点击保存收货地址，通过 ajax 方法将新的收货地址提交到后台，并将新的收

货地址更新到前台，然后关闭对话框。

```
$("#saveChange").click(function(){
    $("#saveChange").button('loading');
    $.post("{:U('user/address')}", $("#address-form").serialize(), success, "json");
    return false;
    function success(data){
        $("#address-form-box-success").html(data.msg);
        if(data.success){
            $("#address-name").html($("#input#input-name").val());
            $("#address-phone").html($("#input#input-phone").val());
            $("#address-address").html($("#input#input-address").val());
        }
        $("#address-form-box-success").show();
        $("#address-form-box").hide();
        $("#saveChange").hide();
        $("#saveChange").button('reset');
    }
});
```

页面初始化时检查库存、购物车状态、商品学校等信息是否正确，如果有错误，给予用户提示并禁用提交订单的按钮。

```
<eq name='outOfStock' value="1">
    $("#submit-area").html("<button class='btn btn-success' disabled='disabled'>
提交订单</button>");
</eq>
<eq name='empty' value="1">
    $("#submit-area").html("<button class='btn btn-success' disabled='disabled'>
提交订单</button>");
</eq>
<eq name='schoolError' value="1">
    $("#submit-area").html("<button class='btn btn-success' disabled='disabled'>
提交订单</button>");
```

</eq>

</script>

6.3.3 修改收货地址的实现

为了进一步方便用户，在用户登录后，可以在当前页面直接修改收货信息，省去了传统电商系统需要跳转几个页面设置不同信息的问题，如图 6.7 所示。核心代码如 6.3.2 节所示。

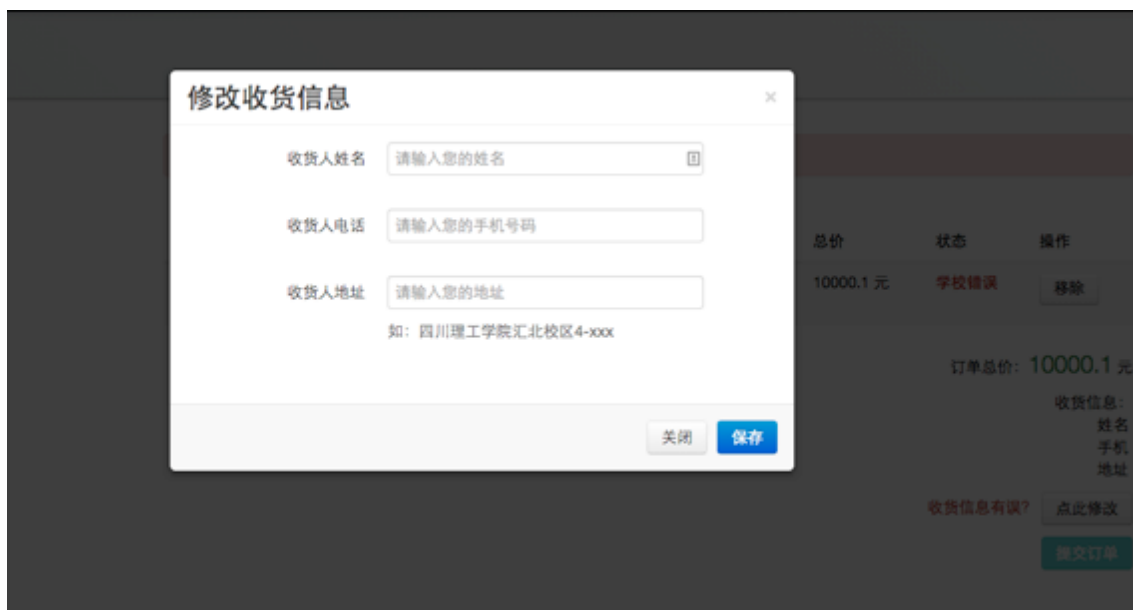


图 6.7 修改收货信息

6.4 订单管理

订单提交后，用户可以在订单管理页面查看订单状态。如图 6.8 所示。



图 6.8 订单管理

订单提交前会检查用户是否登录，如果未登录用户通过非正常手段强制提交，系统也会首先验证登录状态，核心代码如下：

```
if (is_login()) {
    $ret = D('Orderform')->addOrderform();
    if ($ret['status'])
        $this->success('订单提交成功！', U('Orderform/index'));
    else {
        $this->error($ret['msg'], U('Cart/index'));
    }
} else {
    $this->error('您还没有登录，请先登录！', U('User/login'));
}
```

订单提交后，管理员在后台可以对订单进行管理，发货等操作，用户在前台也能看到订单状态的相应改变。

6.5 积分系统

为了激励用户的积极性，系统实现了基于用户行为的积分系统，如图 6.9 所示。



图 6.9 用户积分系统管理面板

积分对应于行为规则：

规则定义：

```
table:$table|field:$field|condition:$condition|rule:$rule[|cycle:$cycle|max:$max][;...
...]
```

规则字段解释：

table->要操作的数据表，不需要加表前缀；

field->要操作的字段；

condition->操作的条件，目前支持字符串，默认变量为执行行为的用户；

rule->对字段进行的具体操作，目前支持加、减；

cycle->执行周期，单位（小时），表示\$cycle 小时内最多执行\$max 次；

max->单个周期内的最大执行次数（\$cycle 和\$max 必须同时定义，否则无效）；

单个行为后可加 ; 连接其他规则。

通过积分系统，后期可以很方便地实现会员等级系统，不同等级的用户可以享有不同的权益，同时，也可以极大地激励用户，增加用户粘性。

6.6 学校/团队管理

6.6.1 学校/管理员账户管理页面设计

超级管理员可在后台添加区域/学校，并分配管理员账户。

在基于分类设置好学校后，管理员在用户>权限管理页面可以方面灵活地给各个管理员分配权限，并且所有权限都细化到功能节点，如是否能管理订单，甚至是具体到某个状态的订单，如图 6.10 所示，设置了大学管理员有权限进行的操作。



图 6.10 设置管理员权限

通过灵活设置管理员权限，可以实现为不同的学校，团队分配管理员。

6.6.2 管理员权限设置前端页面设计

最重要的是，通过给普通用户设置二手市场分类下的商品发布，管理权限，系统实现了 C2C 的二手市场，如图 6.11 所示。



图 6.11 设置注册用户可以在二手市场下进行操作

对于恶意用户，也可以方便地在成员授权里方便地进行封禁处理，如图 6.12 所示。



图 6.12 设置用户授权

6.6.3 将用户添加到用户组的实现

```

public function addToGroup(){
    $uid = I('uid');
    $gid = I('group_id');
    if( empty($uid) ){
        $this->error('参数有误');
    }
    $AuthGroup = D('AuthGroup');
    if(is_numeric($uid)){
        if ( is_administrator($uid) ) {
            $this->error('该用户为超级管理员');
        }
        if( !M('UcenterMember')->where(array('uid'=>$uid))->find() ){
            $this->error('该管理员用户不存在');
        }
    }

    if( $gid && !$AuthGroup->checkGroupId($gid)){
        $this->error($AuthGroup->error);
    }
    if ( $AuthGroup->addToGroup($uid,$gid) ){
        $this->success('操作成功');
    }else{
        $this->error($AuthGroup->getError());
    }
}

```

```
}
```

6.6.4 将用户从用户组移除的实现

```
public function removeFromGroup(){
    $uid = I('uid');
    $gid = I('group_id');
    if( $uid==$this->getVal('uid') ){
        $this->error('不能解除自身授权');
    }
    if( empty($uid) || empty($gid) ){
        $this->error('参数有错误');
    }
    $AuthGroup = D('AuthGroup');
    if( !$AuthGroup->find($gid)){
        $this->error('该用户组不存在');
    }
    if ( $AuthGroup->removeFromGroup($uid,$gid) ){
        $this->success('操作成功');
    }else{
        $this->error('操作失败');
    }
}
```

6.6.5 将分类添加到用户组的实现

```
public function addToCategory(){
    $cid = I('cid');
    $gid = I('group_id');
    if( empty($gid) ){
        $this->error('参数有误');
    }
    $AuthGroup = D('AuthGroup');
    if( !$AuthGroup->find($gid)){
        $this->error('用户组不存在');
```

```

    }
    if( $cid && !$AuthGroup->checkCategoryId($cid)){
        $this->error($AuthGroup->error);
    }
    if ( $AuthGroup->addToCategory($gid,$cid) ){
        $this->success('操作成功');
    }else{
        $this->error('操作失败');
    }
}
}
```

6.7 用户管理

在后台用户管理页面。可以方便灵活地管理注册用户的信息。
同时，结合权限管理，可以为每个用户设置不同的权限，或者禁用某些账户，通过一套基础的系统，完成多个功能模块的实现，如图 6.13 所示。

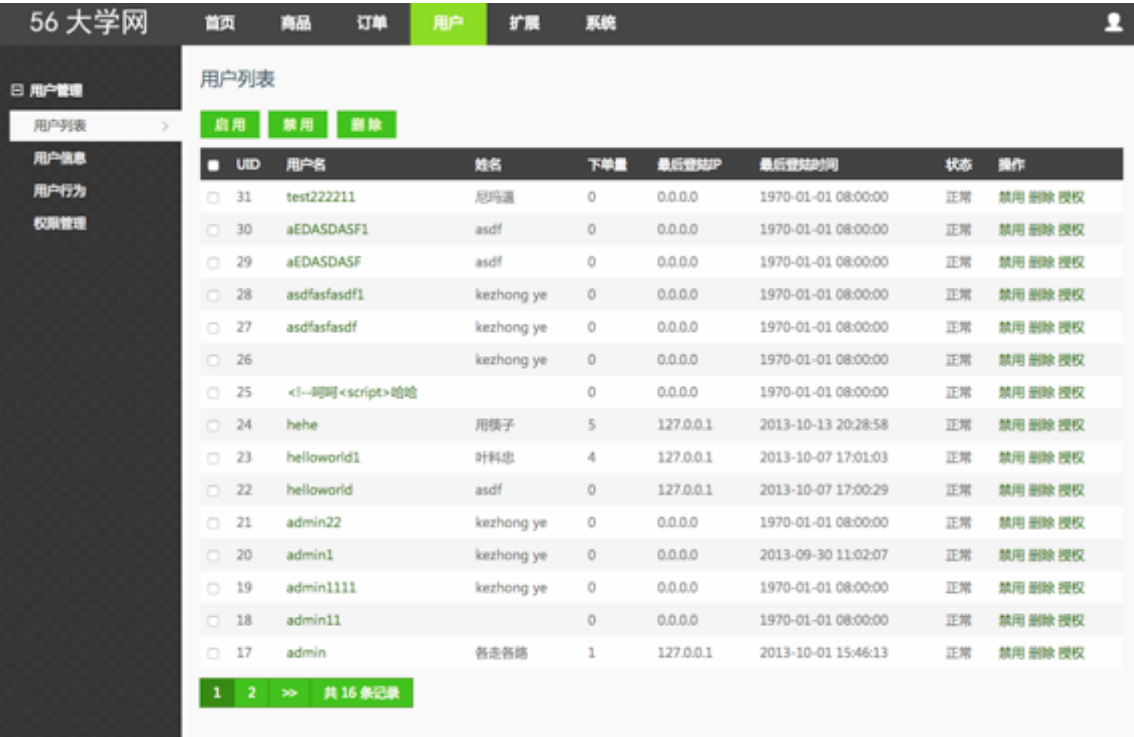


图 6.13 用户列表页面

6.8 商品/库存管理

商品库存管理页面提供了强大的管理功能，增加，启用，在不同分类间复制粘贴移动商品，都可以非常方便地实现操作，搜索功能可以通过标题，时间段，用

户名等条件灵活地检索商品。当商品库存为 0 时，前台将显示没有库存。如图 6.14 所示。



图 6.14 商品/库存管理页面

同时，系统还实现了草稿功能和回收站，进一步方便管理员管理商品，减少工作量，避免误删除带来的损失。

自动保存草稿功能通过前端 Ajax 实现：

保存草稿

var interval;

\$('#autoSave').click(function(){

var target_form = \$(this).attr('target-form');

var target = \$(this).attr('url')

var form = \$('.'+target_form);

var query = form.serialize();

var that = this;

\$(that).addClass('disabled').attr('autocomplete','off').prop('disabled',true);

\$.post(target,query).success(function(data){

if (data.status==1) {

updateAlert(data.info , 'alert-success');

\$('#input[name=id]').val(data.data.id);

} else {

updateAlert(data.info);

}

setTimeout(function(){

```
        $('#top-alert').find('button').click();
        $(that).removeClass('disabled').prop('disabled',false);
    },1500);
})
```

重新开始定时器

```
    clearInterval(interval);
    autoSaveDraft();
    return false;
});
```

Ctrl+S 保存草稿

```
$('#body').keydown(function(e){
    if(e.ctrlKey && e.which == 83){
        $('#autoSave').click();
        return false;
    }
});
```

每隔一段时间保存草稿

```
function autoSaveDraft(){
    interval = setInterval(function(){
        //只有基础信息填写了，才会触发
        var title = $('input[name=title]').val();
        var name = $('input[name=name]').val();
        var des = $('textarea[name=description]').val();
        if(title != "" || name != "" || des != ""){
            $('#autoSave').click();
        }
    }, 1000*parseInt({:C('AOTUSAVE_DRAFT')}));
}
```

```
autoSaveDraft();
```

6.9 订单管理

用户提交订单后,管理员在订单管理页面查看,更改订单状态,安排配送,如图 6.15 所示。

ID	编号	商品	金额	用户	收货信息	时间	操作
17	201505181150	捷安特自行车 x 1 x ¥500	500 元	admin	新主路 15609008651 四川理工学院江 北校区4-xxx	2015-05-18 11:50	待处理 派送 完成 关闭
16	201310092142	Edifier耳机线黑色 x 1 x ¥5	5 元	hehe	叶科忠 汇港4-111	2013-10-09 21:42	待处理 派送 完成 关闭
15	201310092141	Edifier耳机线黑色 x 1 x ¥5	5 元	hehe		2013-10-09 21:41	待处理 派送 完成 关闭
14	201310091427	华为U8800PRO手机华为U8800PRO手机华为U8800PRO手机 x 1 x ¥30 金士顿U盘8G x 2 x ¥32	94 元	hehe		2013-10-09 14:27	待处理 派送 完成 关闭
13	201310091425	Edifier耳机线黑色 x 1 x ¥5 华为U8800PRO手机华为U8800PRO手机华为U8800PRO手机 x 1 x ¥30	35 元	hehe		2013-10-09 14:25	待处理 派送 完成 关闭
12	201310091356	华为U8800PRO手机华为U8800PRO手机华为U8800PRO手机 x 2 x ¥30 笔记本电脑 x 1 x ¥10000.1 Edifier耳机线黑色 x 23 x ¥5 红玫瑰情香水50ml x 20 x ¥25	10675.1 元	hehe		2013-10-09 13:56	待处理 派送 完成 关闭
11	201310081504	捷安特自行车 x 1 x ¥500 Edifier耳机线黑色 x 2 x ¥5	510 元	helloworld1		2013-10-08 15:04	待处理 派送 完成 关闭
10	201310081134	捷安特自行车 x 1 x ¥500	500 元	helloworld1		2013-10-08 11:34	待处理 派送 完成 关闭
7	201310071744	金士顿U盘8G x 2 x ¥ Edifier耳机线黑色 x 1 x ¥	69 元	helloworld1		2013-10-07 17:44	待处理 派送 完成 关闭

图 6.15 订单管理

6.10 基于 RBAC 认证的全局权限管理

系统采用耦合度低权限细分粒度高易使用的 RBAC 认证方式进行全局的权限控制,为权限管理提供节点和可判断的 url 规则,用户属于某个用户组。针对用户组授权后,在权限管理的用户授权里在当前的分组选项里输入用户 id 可加入当前分组。

6.10.1 RBAC 认证节点

在 6.7 里,已经介绍了灵活的权限设置,该权限管理系统正是基于 RBAC 的权限认证体系,在该体系里,权限的粒度最小可以细化到任何一个页面操作,系统已经配置了以下基本权限控制节点:

```
static protected $nodes = array(
    //权限管理页
```

```

array('title'=>'权限管理','url'=>'AuthManager/index','group'=>'用户管理',
      'operator'=>array(
          //权限管理页面的五种按钮
          array('title'=>'删除',
            'url'=>'AuthManager/changeStatus?method=deleteGroup','tip'=>'删除用户组'),
          array('title'=>'禁用',
            'url'=>'AuthManager/changeStatus?method=forbidGroup','tip'=>'禁用用户组'),
          array('title'=>'恢复',
            'url'=>'AuthManager/changeStatus?method=resumeGroup','tip'=>'恢复已禁用的用户组'),
          array('title'=>'新增', 'url'=>'AuthManager/createGroup',
            'tip'=>'创建新的用户组'),
          array('title'=>'编辑', 'url'=>'AuthManager/editGroup',
            'tip'=>'编辑用户组名称和描述'),
          array('title'=>'保存用户组', 'url'=>'AuthManager/writeGroup',
            'tip'=>'新增和编辑用户组的"保存"按钮'),
          array('title'=>'授权', 'url'=>'AuthManager/group',
            'tip'=>'""后台 \ 用户 \ 用户信息"列表页的"授权"操作按钮,用于设置用户所属用户组'),
          array('title'=>'访问授权', 'url'=>'AuthManager/access',
            'tip'=>'""后台 \ 用户 \ 权限管理"列表页的"访问授权"操作按钮'),
          array('title'=>'成员授权', 'url'=>'AuthManager/user',
            'tip'=>'""后台 \ 用户 \ 权限管理"列表页的"成员授权"操作按钮'),
          array('title'=>'解除授权',
            'url'=>'AuthManager/removeFromGroup', 'tip'=>'""成员授权"列表页内的解除授权操作按钮'),
          array('title'=>'保存成员授权', 'url'=>'AuthManager/addToGroup',
            'tip'=>'""用户信息"列表页"授权"时的"保存"按钮和"成员授权"里右上角的"添加"按钮'),
          array('title'=>'分类授权', 'url'=>'AuthManager/category',
            'tip'=>'""后台 \ 用户 \ 权限管理"列表页的"分类授权"操作按钮'),

```

```

        array('title'=>'保存分类授权
', 'url'=>'AuthManager/addToCategory', 'tip'=>'分类授权"页面的"保存"按钮'),
    ),
);

```

6.10.2 RABC 认证权限

除了系统配置的基础节点, 管理员也可以添加自定义节点, 并给各管理员账户分配不同的权限, 从而实现灵活, 强大且安全的权限管理。

自定义权限认证节点实现:

后台节点配置的 url 作为规则存入 auth_rule。

执行新节点的插入, 已有节点的更新, 无效规则的删除三项任务。

需要新增的节点必然位于 \$nodes。

```

public function updateRules(){
    $nodes    = $this->returnNodes(false);
    $AuthRule = M('AuthRule');
    $map      = array('module'=>'admin','type'=>array('in','1,2'));
    status 全部取出,以进行更新, 需要更新和删除的节点必然位于$rules
    $rules    = $AuthRule->where($map)->order('name')->select();

```

构建 insert 数据, 保存需要插入和更新的新节点。

```

    $data      = array();
    foreach ($nodes as $value){
        $temp['name']    = $value['url'];
        $temp['title']   = $value['title'];
        $temp['module'] = 'admin';
        if(isset($value['controllers'])){
            $temp['type'] = AuthRuleModel::RULE_MAIN;
        }else{
            $temp['type'] = AuthRuleModel::RULE_URL;
        }
        $temp['status'] = 1;
    }

```

```

        $data[strtolower($temp['name'].$temp['module'].$temp['type'])] =
$temp;//去除重复项
    }

    $update = array();//保存需要更新的节点
    $sids    = array();//保存需要删除的节点的 id
    foreach ($rules as $index=>$rule){
        $key = strtolower($rule['name'].$rule['module'].$rule['type']);
        if ( isset($data[$key]) ) {
            如果数据库中的规则与配置的节点匹配，说明是需要更新的节点：
            $data[$key]['id'] = $rule['id'];//为需要更新的节点补充 id 值
            $update[] = $data[$key];
            unset($data[$key]);
            unset($rules[$index]);
            unset($rule['condition']);
            $diff[$rule['id']]=$rule;
        }elseif($rule['status']==1){
            $sids[] = $rule['id'];
        }
    }

    if ( count($update) ) {
        foreach ($update as $k=>$row){
            if ( $row!=$diff[$row['id']] ) {
                $AuthRule->where(array('id'=>$row['id']))->save($row);
            }
        }
    }

    if ( count($sids) ) {

        $AuthRule->where( array( 'id'=>array('IN',implode(',',$sids)) ) )->save(array('status'
=>-1));
    }

```

删除规则是否需要从每个用户组的访问授权表中移除该规则?

```

    }
    if( count($data) ){
        $AuthRule->addAll(array_values($data));
    }
    if ( $AuthRule->getDbError() ) {
        trace(['__METHOD__']: '$AuthRule->getDbError()');
        return false;
    }else{
        return true;
    }
}
}

```

6.11 系统设置/自定义页面模块管理

系统采用插件化的自定义模块,方便在页面中插入广告信息,新增页面模块等操作。如在页面头部或底部定义好了各种钩子,需要增加模块时,只需要定义好模块,并添加到某个钩子上,页面钩子就会自动加载该模块,使得页面耦合度低,扩展性强,维护方便。

开发时,预先在系统各处设置了大量的钩子,然后管理员在后台就可以设置每个钩子加载什么插件,如图 6.16 所示。

名称	描述	类型	插件
15 adminSecondHandArticleEdit	后台二手商品编辑	视图	编辑 删除
14 topicComment	评论提交方式扩展钩子。	视图	编辑 删除
13 AdminIndex	首页小格子个性化显示	视图	编辑 删除
8 adminArticleEdit	后台内容编辑页编辑器	视图	编辑 删除
7 documentEditFormContent	添加编辑表单的内容显示钩子	视图	编辑 删除
6 documentSaveComplete	保存文档数据后的扩展钩子	控制器	编辑 删除
5 documentDetailBefore	页面内容前显示用钩子	视图	编辑 删除
4 documentDetailAfter	文档末尾显示	视图	编辑 删除
3 documentEditForm	添加编辑表单的 扩展内容钩子	视图	编辑 删除
2 pageFooter	页面footer钩子,一般用于加载插件JS文件和JS代码	视图	编辑 删除
1 pageHeader	页面header钩子,一般用于加载插件CSS文件和代码	视图	编辑 删除

图 6.16 钩子列表

前台分类列表就是通过钩子系统实现:

```
<block name="side">
  <!-- 左侧 nav
  ===== -->

  <div class="span3">
    <ul class="nav nav-list bs-docs-sidenav">
      {W('Category/lists', array($category['id'], ACTION_NAME ==
'index'))}
    </ul>
  </div>
</block>
```

在导航之中加载了 Category/lists 插件，并传递了参数 array(\$category['id'], ACTION_NAME == 'index') 给插件。这样，在不同的页面，只需要设置不同的参数，就可以实现不同学校或其它情况下对应分类的加载。

第 7 章 功能模块测试

7.1 系统测试概述

一个合格的电商系统，首先要保证系统各个功能模块工作正常，其次，要确保用户输入的数据正确，在数据或操作不正确时，需要给用户反馈明确的提醒。

在以上要求之上，分别设计了基本模块功能测试，数据验证测试，系统权限及安全性测试等方面的测试用例，并进行了严格的测试，以下是详细的测试细节。

7.2 基本模块测试

1. 以一个用户为例，详细测试了从用户注册、登录、添加商品到购物车，下单，发布二手商品等操作，表 7.1 列出了详细的测试数据和结果。

表 7.1 用户使用流程测试

执行的测试操作	期望的结果	实际的结果	结论
注册一个名为 test123456 的用户	注册成功，系统跳转到登录页面	注册成功，系统跳转到登录页面	结果符合预期，测试通过
以用户名为 test123456 的账户登录（之前已注册成功）	系统跳转到首页，页面右上角显示用户名 test123456	系统跳转到首页，页面右上角显示用户名 test123456	结果符合预期，测试通过
点击一个商品图片，进入商品详情页，点击加入购物车	成功进入对应的商品详情页，点击添加购物车后系统提示添加成功	成功进入对应的商品详情页，点击添加购物车后系统提示添加成功。	结果符合预期，测试通过
打开购物车页面，点击提交订单	系统提示提交成功，并跳转到订单页	系统提示提交成功，并跳转到订单页	结果符合预期，测试通过
打开我的店铺页面，以账户 test123456 登录	登录成功，并跳转到二手商品发布页	登录成功，并跳转到二手商品发布页	结果符合预期，测试通过
管理员给用户 test123456 设置了发布二手商品的权限后，点击新增，添加	系统提示添加成功，并转到商品列表页，网站前台可以看到添加的商品	系统提示添加成功，并转到商品列表页，网站前台可以看到添加的商品	结果符合预期，测试通过

一个二手商品			
--------	--	--	--

2. 以一个用户为例，详细测试用户注册，登录，添加商品到购物车时，添加商品及账户被管理员移除权限后提交的数据不正确时系统能否提供正确的反馈和处理，表 7.2 列出了详细的测试数据和结果。

表 7.2 数据验证及权限测试

执行的测试操作	期望的结果	实际的结果	结论
在系统已存在用户名为 helloworld 的用户时，再注册一个名为 helloworld 的用户	系统提示用户名被占用	系统提示用户名被占用，如图 7.1 所示。	结果符合预期，测试通过
以用户名为 test 的用户登录（系统里没有这个用户）	系统提示用户名不存在或被禁用。	系统提示用户名不存在或被禁用，如图 7.2 所示。	结果符合预期，测试通过
在商品详情页，管理员在后台将商品库存修改为 0，然后用户点击添加到购物车	系统提示添加失败，商品库存不足	系统提示添加失败，商品库存不足，如图 7.3 所示。	结果符合预期，测试通过
管理员将用户 helloworld 从注册用户组移除，用户 helloworld 尝试编辑之前添加的商品	系统提示无权访问，您可能需要联系管理员为您授权	系统提示无权访问，您可能需要联系管理员为您授权，如图 7.4 所示。	结果符合预期，测试通过

15894586511

姓名

测试

地址

deyangshisonglingzhengdongguangcun1

验证码

u65wp



用户名被占用!

注册

图 7.1 注册失败提示页面

56大学网

首页

自营

二手

用户登录

还没有账号? 立即注册

用户名

test

密码

.....

验证码

6NDM2



用户不存在或被禁用!

☒ 自动登陆

登 陆

图 7.2 登录失败提示页面



图 7.3 商品添加失败提示页面

提示:无权访问,您可能需要联系管理员为您授权!

基础

高级

扩展

商品名称（请输入商品名称）

铅笔

商品价格

1

商品摘要（请输入简介）

一只铅笔

封面图片

上传图片



下一步

确定

返回

图 7.4 无权访问提示页面

7.3 系统测试总结

通过对功能、提示、权限等多个方面的测试，验证了系统的准确、稳定和安全，确保了用户能正常使用网站，并且，用户只能进行有权限的操作。在购物车里商品不足时，能够准确提示用户，确保不会出现数据不一致。

对于所有用户输入的数据，都做了完整的检测和约束，如用户名，邮箱格式，

手机号格式等，确保用户提供的信息正确，以便顺利完成订单提交之后的配送流程，在数据或操作不正确时，用户也能得到明确友善的提醒，引导用户正确使用系统。

通过这些测试，验证了系统流程的完整性，在测试过程中，对于系统存在的错误，系统流程的缺陷，系统是否有准确及时的提示等方面，发现了不少问题，并做了对应的修复和改动，确保系统达到了规划中的要求。

第 8 章 总结

电子商务领域经过多年的发展,早已成为一个巨头林立的成熟市场,尽管如此,在诸如大学这样的细分市场,仍有传统网购周期长、商品质量参差不齐、售后难等问题,用户的需求并未被良好地满足,在细微的用户需求及用户体验之上,有广阔的二级市场空间存在,因此才有了本系统所做的探索。

这个校园电商系统正是基于这样的困境,提出并尝试解决这些问题,尽可能地简化购物及送货流程,并且提供便捷的校内商品寄卖服务,一定程度上缩短送货时间,方便退换货。

本系统相对于传统电商,如淘宝、京东等,有多处明显的优势。首先,由于本项目主要面向大学校园,除开核心开发和管理人员外,一所大学仅需要一名管理和一些负责推广和派单的兼职人员,所以本项目所需人力资源不多,运营成本大大降低。同时也可以为更多的在校大学生提供锻炼和兼职的机会。其次,同样是由于校内平台,我们不仅可以做到当天送货,甚至可以实现一小时送货,这是传统电商没法做到的。另外,在系统已有功能的基础上,后期可以比较容易地实现第三方商家,如校园周边外卖,文具书本店家的接入,真正地在一套系统内,将 B2C/C2C,以及第三方商家接入了进来,形成一个灵活、低运营成本、多利润点的闭环系统。

在系统的开发过程中采用了众多当下流行的开发工具,技术理念,一方面,通过这些技术的使用,对于用户体验能够起到提升,另一方面,对于作者来说,也是一次锻炼运用并掌握专业知识的历程,验证并实践了如算法、设计模式等专业知识。

系统功能仍然处于早期阶段,有很大的提升空间,例如,系统还未能完整兼容移动端使用,对于寄卖商品的品质管理,也暂未有好的解决方案,这些都是本系统和传统电商领域需要继续探索并完善的地方。

致 谢

经过几个月的前期调研，校园电商市场需求分析，开发系统和撰写论文，终于完成了这项毕业设计课题，几个月以来，在选题、开题、系统开发及论文的写作过程中，遇到了各种问题和麻烦，得益于来自各个方面的无私帮助、指导，才能顺利完成本文和本文论述的校园电商系统。

首先，作者要对本次毕业设计的指导老师梁兴建老师致以强烈的感谢，论文的内容和格式细节的修正改进都是琐碎至极的事，感谢梁兴建老师提供的不厌其烦的帮助以及指导。

其次，本文及本校园电商系统的实现，参考了众多文献、技术框架、专业书籍，对于各位学者的研究和工作成果对本次毕业设计提供的帮助，未能一一列举，仅在此表达诚恳的谢意。

另外，在此预先向参与本次答辩的老师们的致以谢意。

感谢四川理工学校的老师和同学们，在学校所学所得的知识，是这篇论文及校园电商系统能够成功实现的基础。

作者水平和经验有限，在系统设计及论文撰写的细节上难免有不足及不合理之处，恳请批评！恳请指正！

参考文献

- [1] 王丹. 我国 C2C 电子商务网站的盈利模式探析[D]. 北京大学, 2007.
- [2] 姚青. 电子商务平台——购物网站的设计与实现[D]. 山东大学, 2012.
- [3] Matt Zandstra. 深入 PHP 面向对象、模式与实践（第二版）[M]. 北京：人民邮电出版社, 2010.
- [4] 姜承尧. MySQL 技术内幕 InnoDB 存储引擎[M]. 机械工业出版社, 2010.
- [5] 麦卡劳(Alex MacCaw). 基于 MVC 的 JavaScript Web 富应用开发[M]. 电子工业出版社, 2012.
- [6] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. 算法导论（原书第 2 版）[M]. 机械工业出版社, 2006.
- [7] 吴翰清. 白帽子讲 Web 安全[M]. 机械工业出版社, 2010.
- [8] 史蒂夫·克鲁克. 点石成金 访客至上的网页设计秘笈[M]. 电子工业出版社, 2006.
- [9] 列旭松, 陈文. PHP 核心技术与最佳实践[M]. 机械工业出版社, 2013.
- [10] Paul DuBois. MySQL 技术内幕（第 4 版）[M]. 人民邮电出版社, 2011.
- [11] 余平, 黄文琪. C2C 电子商务创业教程[M]. 清华大学出版社, 2008.
- [12] ShopNC 产品部. 高性能电子商务平台构建[M]. 机械工业出版社, 2014.
- [13] PHP China. PHP 5 项目开发实战详解[M]. 电子工业出版社, 2008.
- [14] 鸟哥. 鸟哥的 Linux 私房菜. 基础学习篇（第三版）[M]. 人民邮电出版社, 2010.
- [15] 鸟哥. 鸟哥的 Linux 私房菜. 服务器架设篇（第三版）[M]. 机械工业出版社, 2012.

附 录

附录 A:主要源代码

1、UserApi 类

```
<?php
```

```
namespace User\Api;
```

```
use User\Api\Api;
```

```
use User\Model\UcenterMemberModel;
```

```
//require_cache(dirname(__FILE__) . '/Api.class.php');
```

```
class UserApi extends Api{
```

```
    protected function _init(){
```

```
        $this->model = new UcenterMemberModel();
```

```
    }
```

```
    public function register(){
```

```
        return $this->model->register();
```

```
    }
```

```
    public function login($username, $password, $type = 1){
```

```
        return $this->model->login($username, $password, $type);
```

```
    }
```

```
    public function info($uid, $is_username = false){
```

```
        return $this->model->info($uid, $is_username);
```

```
    }
```

```
    public function checkUsername($username){
```

```
        return $this->model->checkField($username, 1);
```

```
    }
```

```
    public function checkEmail($email){
```

```

        return $this->model->checkField($email, 2);
    }

    public function checkMobile($mobile){
        return $this->model->checkField($mobile, 3);
    }

    public function updateInfo($uid, $password, $data){
        if($this->model->updateUserFields($uid, $password, $data) !== false){
            $return['status'] = true;
            $return['info'] = "修改成功! ";
        }else{
            $return['status'] = false;
            $return['info'] = $this->model->getError();
        }
        return $return;
    }

    public function updateAddress($uid, $data){
        if($this->model->updateUserAddress($uid, $data) !== false){
            $return['status'] = true;
            $return['info'] = "修改成功! ";
        }else{
            $return['status'] = false;
            $return['info'] = $this->model->getError();
        }
        return $return;
    }
}

```

2、UcenterMemberModel 类

```
<?php
```

```

namespace User\Model;
use Think\Model;
use Home\Model\CartModel as CartModel;
/**
 * 会员模型

```

```

*/
class UcenterMemberModel extends Model{
    /**
     * 数据表前缀
     * @var string
     */
    protected $tablePrefix = UC_TABLE_PREFIX;

    /**
     * 数据库连接
     * @var string
     */
    protected $connection = UC_DB_DSN;

    /* 用户模型自动验证 */
    protected $_validate = array(
        array('username', 'require', -16),
        array('password', 'require', -16),
        array('email', 'require', -16),
        array('reg_phone', 'require', -16),
        array('sid', 'require', -16),
        array('email', 'require', -16),
        array('qq', 'require', -16),
        array('address', 'require', -16),
        array('email', 'require', -16),

        /* 验证用户名 */
        array('username', '6,16', -1, self::EXISTS_VALIDATE, 'length'), //用户名
        长度不合法
        array('username', 'checkDenyMember', -2, self::EXISTS_VALIDATE,
        'callback'), //用户名禁止注册
        array('username', '', -3, self::EXISTS_VALIDATE, 'unique'), //用户名被占
        用
        array('username', '/^[0-9a-zA-Z]{6,15}$/', -16, self::EXISTS_VALIDATE,
        'regex'),

        /* 验证密码 */
        array('password', '6,30', -4, self::EXISTS_VALIDATE, 'length'), //密码长
        度不合法

```

```

/* 验证邮箱 */
array('email', 'email', -5, self::EXISTS_VALIDATE), //邮箱格式不正确
array('email', '1,32', -6, self::EXISTS_VALIDATE, 'length'), //邮箱长度不
合法
array('email', 'checkDenyEmail', -7, self::EXISTS_VALIDATE, 'callback'), //
邮箱禁止注册
array('email', '', -8, self::EXISTS_VALIDATE, 'unique'), //邮箱被占用

/* 验证手机号码 */
//array('mobile', '//', -9, self::EXISTS_VALIDATE), //手机格式不正确 TODO:
array('reg_phone', 'checkDenyMobile', -10, self::EXISTS_VALIDATE,
'callback'), //手机禁止注册
array('reg_phone', '11,11', -12, self::EXISTS_VALIDATE, 'length'), //手机
长度不合法
array('reg_phone', '', -11, self::EXISTS_VALIDATE, 'unique',
self::MODEL_INSERT), //手机号被占用

array('sid', '/^[1-9]{1,3}$/', -13, self::EXISTS_VALIDATE, 'regex'),
array('sid', 'checkSid', -13, self::EXISTS_VALIDATE, 'callback'), //学校
不存在
array('qq', '/^[0-9]{6,12}$/', -14, self::EXISTS_VALIDATE, 'regex'),

array('realname', '/^[x{4e00}-x{9fa5}]{2,4}$u', -15,
self::EXISTS_VALIDATE, 'regex'),
);

/* 用户模型自动完成 */
protected $_auto = array(
// array('username', 'filter', self::MODEL_INSERT, 'callback'),
array('password', 'think_ucenter_md5', self::MODEL_BOTH, 'function'),
array('address', 'filter', self::MODEL_BOTH, 'callback'),
array('reg_time', NOW_TIME, self::MODEL_INSERT),
array('reg_ip', 'get_client_ip', self::MODEL_INSERT, 'function', 1),
array('update_time', NOW_TIME, self::MODEL_INSERT),
array('status', 'getStatus', self::MODEL_BOTH, 'callback'),
);

```

```
protected function filter($string) {
    return htmlspecialchars(strip_tags($string));
}

/**
 * 检测用户名是不是被禁止注册
 * @param string $username 用户名
 * @return boolean         ture - 未禁用, false - 禁止注册
 */
protected function checkDenyMember($username) {
    return true; //TODO: 暂不限制, 下一个版本完善
}

protected function checkSid($sid) {
    if (M('Category')->where('pid = 0 and id='.$sid)->find()) {
        return true;
    }
    return false;
}

/**
 * 检测邮箱是不是被禁止注册
 * @param string $email 邮箱
 * @return boolean         ture - 未禁用, false - 禁止注册
 */
protected function checkDenyEmail($email) {
    return true; //TODO: 暂不限制, 下一个版本完善
}

/**
 * 检测手机是不是被禁止注册
 * @param string $mobile 手机
 * @return boolean         ture - 未禁用, false - 禁止注册
 */
protected function checkDenyMobile($mobile) {
    return true; //TODO: 暂不限制, 下一个版本完善
}
```

```

/**
 * 根据配置指定用户状态
 * @return integer 用户状态
 */
protected function getStatus() {
    return true; //TODO: 暂不限制, 下一个版本完善
}

/**
 * 注册一个新用户
 * @param string $username 用户名
 * @param string $password 用户密码
 * @param string $email 用户邮箱
 * @param string $mobile 用户手机号码
 * @return integer 注册成功-用户信息, 注册失败-错误编号
 */
public function register() {
    $data = array(
        'username' => $_POST['username'],
        'password' => $_POST['password'],
        'email' => $_POST['email'],
        'reg_phone' => $_POST['mobile'],
        'mobile' => $_POST['mobile'],
        'realname' => $_POST['realname'],
        'address' => $_POST['address'],
        'sid' => $_POST['sid'],
        'qq' => $_POST['qq'],
    );

    //验证手机
    //if(empty($data['mobile'])) unset($data['mobile']);

    /* 添加用户 */
    if($this->create($data)) {
        $uid = $this->add();
        if ($uid != 0) {
            $M = M('AuthGroupAccess');
            $auth = array(

```



```

        'uid' => $uid,
        'group_id' => 3,
    );
    $M->create($auth);
    $M->add();
}
return $uid ? $uid : 0; //0-未知错误, 大于 0-注册成功
} else {
    return $this->getError(); //错误详情见自动验证注释
}
}

/**
 * 用户登录认证
 * @param string $username 用户名
 * @param string $password 用户密码
 * @param integer $type 用户名类型 (1-用户名, 2-邮箱, 3-手机, 4-UID)
 * @return integer 登录成功-用户 ID, 登录失败-错误编号
 */
public function login($username, $password, $type = 1){
    $map = array();
    switch ($type) {
        case 1:
            $map['username'] = $username;
            break;
        case 2:
            $map['email'] = $username;
            break;
        case 3:
            $map['mobile'] = $username;
            break;
        case 4:
            $map['uid'] = $username;
            break;
        default:
            return 0; //参数错误
    }
}

```

```

/* 获取用户数据 */
$user = $this->where($map)->find();
if(is_array($user) && $user['status']){
    /* 验证用户密码 */
    if(think_ucenter_md5($password) === $user['password']){
        $this->updateLogin($user['uid']); //更新用户登录信息
        return $user['uid']; //登录成功，返回用户 ID
    } else {
        return -2; //密码错误
    }
} else {
    return -1; //用户不存在或被禁用
}
}

/**
 * 获取用户信息
 * @param string $uid      用户 ID 或用户名
 * @param boolean $is_username 是否使用用户名查询
 * @return array           用户信息
 */
public function info($uid, $is_username = false){
    $map = array();
    if($is_username){ //通过用户名获取
        $map['username'] = $uid;
    } else {
        $map['uid'] = $uid;
    }

    $user = $this->where($map)->field('uid,username,email,mobile,status')->find();
    if(is_array($user) && $user['status'] = 1){
        return array($user['uid'], $user['username'], $user['email'],
            $user['mobile']);
    } else {
        return -1; //用户不存在或被禁用
    }
}

```

```
/**
 * 检测用户信息
 * @param string $field 用户名
 * @param integer $type 用户名类型 1-用户名, 2-用户邮箱, 3-用户电话
 * @return integer 错误编号
 */
public function checkField($field, $type = 1){
    $data = array();
    switch ($type) {
        case 1:
            $data['username'] = $field;
            break;
        case 2:
            $data['email'] = $field;
            break;
        case 3:
            $data['mobile'] = $field;
            break;
        default:
            return 0; //参数错误
    }

    return $this->create($data) ? 1 : $this->getError();
}

/**
 * 更新用户登录信息
 * @param integer $uid 用户 ID
 */
protected function updateLogin($uid){
    $data = array(
        'id' => $uid,
        'last_login_time' => NOW_TIME,
        'last_login_ip' => get_client_ip(1),
    );
    $this->save($data);
}
```

```
/**
 * 更新用户信息
 * @param int $uid 用户 id
 * @param string $password 密码，用来验证
 * @param array $data 修改的字段数组
 * @return true 修改成功，false 修改失败
 */
public function updateUserFields($uid, $password, $data){
    if(empty($uid) || empty($password) || empty($data)){
        $this->error = '参数错误!';
        return false;
    }

    //更新前检查用户密码
    if(!$this->verifyUser($uid, $password)){
        $this->error = '验证出错：密码不正确!';
        return false;
    }

    //更新用户信息
    $data = $this->create($data);
    if($data){
        return $this->where(array('uid'=>$uid))->save($data);
    }
    return false;
}

/**
 * 更新用户收货地址
 * @param int $uid 用户 id
 * @param array $data 修改的字段数组
 * @return true 修改成功，false 修改失败
 */
public function updateUserAddress($uid, $data){
    if(empty($uid) || empty($data)){
        $this->error = '参数错误!';
        return false;
    }
}
```

```

    }
    //更新用户信息
    $data = $this->create($data);
    if(data){
        if ($this->where(array('uid'=>$uid))->save($data))
            return true;
        else if ($this->getError() == "非法数据对象!") {
            $this->error = "数据非法! <br/>姓名只能为 2-3 位中文。<br/>手机号只能为 11 位数字。";
        }
    }
    return false;
}

```

```

/**
 * 验证用户密码
 * @param int $uid 用户 id
 * @param string $password_in 密码
 * @return true 验证成功, false 验证失败
 */
protected function verifyUser($uid, $password_in){
    $password = $this->getFieldById($uid, 'password');
    if(think_ucenter_md5($password_in, UC_AUTH_KEY) === $password){
        return true;
    }
    return false;
}

```

```

}

```

3、购物车控制器类

```

<?php

```

```

// +-----
// | Author: 叶科忠 <yekz.qq.com> <http://www.yekezong.com>
// +-----

```

```

namespace Home\Controller;
use User\Api\UserApi as UserApi;

```

```

/**
 * 购物车控制器
 */
class CartController extends HomeController {
    // 购物车首页
    public function index() {
        $goods = null;
        if (is_login()) {
            $D = D('Cart');
            $goods = $D->getGoodsList(session('user_auth.uid'));
        } else {
            $goods = session("cart");
        }

        $M = M("Document");
        $Cover = M('Picture');
        $Category = M("Category");
        $outOfStock = 0; // 超出库存
        $empty = 0; // 购物车为空
        $schoolError = 0; // 非注册学校商品
        if (count($goods) == 0) {
            $empty = 1;
        }

        for ($i=0; $i < count($goods); $i++) {
            $id = $goods[$i]['gid'];
            $goodsInfo = $M->where("id = '{$id}' and status = 1")->find();
            $goods[$i]['goodsName'] = $goodsInfo["title"];
            $goods[$i]['price'] = $goodsInfo["price"];
            $goods[$i]['status'] = 1;
            if ($goods[$i]['number'] > $goodsInfo['stock']) {
                $goods[$i]['status'] = "2";
                $outOfStock = 1; // 超出库存
            }

            $sid = $M->where("id = '{$id}'")->getField("sid");
            if ($sid != session("sid")) {
                $schoolError = 1;
                $goods[$i]['status'] = 3;
            }

            $goods[$i]['secondhand'] =

```

```

$Category->where('id='.$goodsInfo["category_id"])->getField("secondhand");
    // $goods[$i]['pic'] = $Cover->where('id='.$M->where("id = '{$id}' and
status = 1")->getField("cover_id"))->getField("path");
    }
    if (is_login())
        $address =
M('UcenterMember')->where('uid='.$session('user_auth.uid'))->find();

    $schoolName =
M('Category')->where('id='.$session("sid"))->getField("name");
    $this->assign('schoolName', $schoolName);
    $this->assign('total', get_total_money_from_cart());
    $this->assign('list', $goods);
    $this->assign('outOfStock', $outOfStock);
    $this->assign('schoolError', $schoolError);
    $this->assign('empty', $empty);
    $this->assign('address', $address);
    $this->display();
}
// 添加商品到购物车
public function add($gid = 0, $number = 1) {
    if (IS_POST) {
        $number = I('post.number');
        $stock = M("Document")->where('id='.$gid)->getField("stock");
        if ($stock < $number) { // 检查库存
            $data = array("success" => 0, "msg" => "添加失败，商品库存不足！");
        } else if (($gid && $number && is_numeric($gid) && is_numeric($number))) {
            if (is_login()) { // 已登录，从数据库
                if (!D('Cart')->addGoodsToCart($gid, $number)) {
                    $data = array("success" => 0, "msg" => "添加失败，商品库存不足，去看看购物车里有几个吧！");
                }
                echo json_encode($data);
                return false;
            }
        } else { // 未登陆，从 session
            $okFlag = false;
            $cart = session("cart");

```

```

        for ($i=0; $i < count($cart); $i++) {
            if ($cart[$i]['gid'] == $gid) {
                if ($stock < ($cart[$i]['number'] + $number)) {
                    $data = array("success" => 0, "msg" => "添加失败，
商品库存不足，去看看购物车里有几个吧！");
                    echo json_encode($data);
                    return false;
                }
                $cart[$i]['number'] += $number; //购物车已存在，数量
相加

                $okFlag = true;
            }
        }
        if (!$okFlag)
            array_push($cart, array("gid" => $gid, "number" =>
$number));

        session("cart", $cart);
    }
    $data = array("success" => 1, "msg" => "添加成功！");
} else {
    $data = array("success" => 0, "msg" => "添加失败，非法数据！");
}
echo json_encode($data);
}
}

```

// 改变单件商品数量

```

public function changeNumber() {
    if (IS_POST) {
        $gid = I('post.gid');
        $number = I('post.number');
        if(($gid && $number && is_numeric($gid) && is_numeric($number))) {
            $stock = M("Document")->where('id='.$gid)->getField("stock");
            if (is_login()) {
                $goods = D('Cart')->getGoodsList(session('user_auth.uid'));
                foreach ($goods as $key => $value) {
                    if ($value['gid'] == $gid) {
                        if (!D('Cart')->setGoodsNumber($gid, $number))

```



```

        $this->error(' 修改失败, 商品库存不足! ');
        break;
    }
}
} else {
    $goods = session("cart");
    for ($i=0; $i < count($goods); $i++) {
        if ($goods[$i]['gid'] == $gid) {
            if ($stock < ($goods[$i]['number'] + $number)) {
                $this->error(' 修改失败, 商品库存不足! ');
            }
            $goods[$i]['number'] = $number;
            break;
        }
    }
    session("cart", $goods);
}
$this->success("修改成功!");
} else {
    $this->error(' 修改失败! ');
}
}

// 移除单件商品
public function remove($gid) {
    if (($gid && is_numeric($gid))) {
        if (D('Cart')->removeGoods($gid))
            $this->success("移除成功!");
        else
            $this->error(' 移除失败! ');
    } else {
        $this->error(' 移除失败! ');
    }
}
}
}

```

4、订单控制器类

```

<?php
// +-----
// | Author: 叶科忠 <yekz.qq.com> <http://www.yekezong.com>
// +-----

namespace Home\Controller;
use User\Api\UserApi as UserApi;

/**
 * 订单控制器
 */
class OrderformController extends HomeController {
    // 订单首页
    public function index() {
        if (is_login()) {
            $goodsList = null;
            $list = D('Orderform')->getOrderformList();
            $Document = M('Document');
            foreach ($list as $key => $value) {
                $goods = explode("|", substr($value['goods'], 0, -1));
                foreach ($goods as $good) {
                    $good = explode(",", $good);
                    $goodsList .= "<a target='_blank' href='". U('article/detail',
array("id"=>$good[0])) . " ' >"
$Document->where(' id='.$good[0])->getField("title") . "</a> x ". $good[1] . " x ¥" .
$good[2] . "<br/>";
                }
                $list[$key]['goodsList'] = $goodsList;
                $list[$key]['time'] = date("Y-m-d H:i", $value['time']);
                $goodsList = null;
            }
            $this->assign('list', $list);
            $this->display();
        } else {
            $this->error('您还没有登录，请先登录!', U('User/login'));
        }
    }
    // 提交订单

```

```
public function submit() {  
    if (is_login()) {  
        $ret = D('Orderform')->addOrderform();  
        if ($ret['status'])  
            $this->success(' 订单提交成功! ', U('Orderform/index'));  
        else {  
            $this->error($ret['msg'], U('Cart/index'));  
        }  
    } else {  
        $this->error(' 您还没有登录, 请先登录! ', U('User/login'));  
    }  
}  
  
protected function getGoodsList() {  
  
}  
  
}
```

附录 B: 系统使用说明书

基于 B/C2C 的电商网站系统使用说明书

1、运行环境与软件要求

- (1) Linux 2.6 及以上或 windows 2003 及以上
- (2) PHP 5.4 以上
- (3) MySQL 4.2.0 以上
- (4) Nginx 1.6.0 以上
- (5) PHP-FPM 5.4 以上

2、LNMP的安装与配置

(1) 编译安装 PHP

<http://php.net/archive>

增加用户

groupadd www

useradd -g www -s /sbin/nologin -M www

编译安装依赖包

yum install libxml2-devel.x86_64 openssl-devel.x86_64 bzip2-devel.x86_64

libcurl-devel.x86_64 mysql-devel.x86_64 libedit-devel.x86_64 readline-devel.x86_64

编译 php

./configure \

--prefix=/usr/local/php56 \

--with-config-file-path=/usr/local/php56/etc \

--with-fpm-user=www \

--with-fpm-group=www \

--with-mysql=mysqlnd \

--with-pdo-mysql=mysqlnd \

--with-mysqli=mysqlnd \

--with-curl \

--with-zlib \

--with-bz2 \

--with-readline \

--with-pear \

--with-gettext \

```
--with-iconv \
--with-curl \
--with-mcrypt \
--with-mhash \
--with-openssl \
--enable-opcache \
--enable-fpm \
--enable-pcntl \
--enable-soap \
--enable-mbstring \
--enable-xml \
--enable-bcmath \
--enable-shmop \
--enable-sysvsem \
--enable-inline-optimization \
--enable-shmop \
--enable-sysvmsg \
--enable-sysvsem \
--enable-sysvshm \
--enable-sockets \
--enable-zip \
--enable-mbregex \
--enable-ftp
make && make install
# 如果出错
# make ZEND_EXTRA_LIBS='-liconv'

# 配置文件
cp /root/tmp/php-5.6.8/php.ini-development /usr/local/php56/etc/php.ini
ln -s /usr/local/php56/etc/php.ini /etc/php.ini

# 添加到环境路径
PATH=$PATH:/usr/local/php56/bin

(2) 安装 PHP-FPM
cp /usr/local/php56/etc/php-fpm.conf.default /usr/local/php56/etc/php-fpm.conf
cp /root/tmp/php-5.6.8/sapi/fpm/init.d.php-fpm /etc/init.d/php-fpm56
chmod +x /etc/init.d/php-fpm56
```

(3) 安装系统所需 PHP 扩展

```
pecl install memcache
```

(4) 安装 nginx

```
yum install nginx
```

(5) 配置 Nginx

```
server
{
    listen 80;
    server_name shop.com;
    index index.html index.shtml index.php;
    root /home/wwwroot/shop;
    include /home/wwwroot/shop/nginx_rewrite.conf;
    location ~ .*\.php|php5)?$
    {
        try_files $uri =404;
        fastcgi_pass unix:/tmp/php-cgi.sock;
        fastcgi_index index.php;
        include fcgi.conf;
    }
    location ~ .*\.gif|jpg|jpeg|png|bmp|swf)$
    {
        expires 30d;
    }
    location ~ .*\.js|css)?$
    {
        expires 12h;
    }
    location ~ .*\.htaccess)?$
    {
        deny all;
    }
    access_log /home/wwwlogs/shop.access.log access;
}
```

(6) 其它配置

```
# 服务器
```

```
# 修改默认22端口到8123
```

```
vi /etc/ssh/sshd_config
```

```
Port 8123
```

```
# 重起ssh服务
```

```
service sshd restart
```

```
# 数据库仅允许本地访问
```

```
mysql: skip-networking
```

```
vi /etc/my.cnf
```

```
service mysql restart
```

```
# 打开php.ini httponly=1配置
```

```
vi /usr/local/php/etc/php.ini
```

```
service php-fpm restart
```

```
service nginx restart
```

(7) 启动服务

```
service php-fpm56 start
```

```
service nginx start
```

(8) 改写 hosts

```
127.0.0.1 shop.com
```

安装、配置并启动成功后，新建 PHP 文件 phpinfo.php

```
<?php
```

```
    phpinfo();
```

```
?>
```

在浏览器访问 [http://127.0.0.1 /phpinfo.php](http://127.0.0.1/phpinfo.php)，查看系统，如图 B. 1 所示。

<div>PHP Version 5.6.8</div> <div>php</div>	
System	Darwin YesMBP.lan 14.4.0 Darwin Kernel Version 14.4.0: Tue May 5 14:17:52 PDT 2015; root:xnu-2782.30.4~3/RELEASE_ARM64_T8020
Build Date	May 10 2015 16:21:18
Configure Command	./configure '--prefix=/usr/local/Cellar/php56/5.6.8' '--localstatedir=/usr/local/var' '--sysconfdir=/usr/local/etc/php/5.6' '--with-config-file-path=/usr/local/etc/php/5.6' '--with-config-file-scan-dir=/usr/local/etc/php/5.6/conf.d' '--mandir=/usr/local/Cellar/php56/5.6.8/share/man' '--enable-bcmath' '--enable-calendar' '--enable-dba' '--enable-xml' '--enable-ftp' '--enable-gd-native-ttf' '--enable-intl' '--enable-mbregex' '--enable-mbstring' '--enable-shmop' '--enable-soap' '--enable-sockets' '--enable-sysvmsg' '--enable-sysvsem' '--enable-sysvshm' '--enable-wddx' '--enable-zip' '--with-freetype-dir=/usr/local/opt/freetype' '--with-gd' '--with-gettext=/usr/local/opt/gettext' '--with-iconv-dir=/usr' '--with-icu-dir=/usr/local/opt/icu4c' '--with-jpeg-dir=/usr/local/opt/jpeg' '--with-kerberos=/usr' '--with-libedit' '--with-mhash' '--with-ndbm=/usr' '--with-openssl=/usr/local/opt/openssl' '--with-pdo-odbc=unixODBC,/usr/local/opt/unixodbc' '--with-png-dir=/usr/local/opt/libpng' '--with-unixODBC=/usr/local/opt/unixodbc' '--with-xmllib' '--with-zlib=/usr/local/opt/zlib' '--with-apxs2=/usr/bin/apxs' '--libexecdir=/usr/local/Cellar/php56/5.6.8/libexec' '--with-bz2=/usr' '--disable-debug' '--enable-fpm' '--with-fpm-user=_www' '--with-fpm-group=_www' '--with-curl' '--with-xsl=/usr' '--with-ldap' '--with-ldap-sasl=/usr' '--with-mysql-sock=/tmp/mysql.sock' '--with-mysql=mysqlnd' '--with-mysql=mysqlnd' '--with-pdo-mysql=mysqlnd' '--disable-openssl' '--enable-pcntl' '--enable-maintainer-zts' '--enable-phpdbg' '--enable-zend-signals' 'CC=clang' 'CXX=clang++'
Server API	FPM/FastCGI
Virtual Directory Support	enabled
Configuration File (php.ini) Path	/usr/local/etc/php/5.6
Loaded Configuration File	/usr/local/etc/php/5.6/php.ini
Scan this dir for additional .ini files	/usr/local/etc/php/5.6/conf.d
Additional .ini files parsed	/usr/local/etc/php/5.6/conf.d/ext-gearman.ini, /usr/local/etc/php/5.6/conf.d/ext-mcrypt.ini, /usr/local/etc/php/5.6/conf.d/ext-memcache.ini, /usr/local/etc/php/5.6/conf.d/ext-opcache.ini, /usr/local/etc/php/5.6/conf.d/ext-pthreads.ini
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,TS
PHP Extension Build	API20131226,TS
Debug Build	no
Thread Safety	enabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, compress.bzip2, php, file, glob, data, http, ftp, phar, zip
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, bzip2.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, mcrypt.*, mdecrypt.*
<div> This program makes use of the Zend Scripting Language Engine: Zend Engine v2.6.0, Copyright (c) 1998-2015 Zend Technologies with Zend OPcache v7.0.4-dev, Copyright (c) 1999-2015, by Zend Technologies </div> <div>zendengine</div>	

图 B.1 PHP 系统信息页面

3、运行说明

安装、配置并启动成功后，就可以使用链接：<http://shop.com/> 访问到系统首页，如图 B.2 所示。



图 B.2 网站首页