

图书分类号: _____
密 级: _____

徐州工程学院

毕业设计(论文)

基于 SSH 网上购物商城系统的设计与实现

DESIGN AND IMPLEMENTATION OF ONLINE
SHOPPING MALL SYSTEM BASED ON SSH

学生姓名	汤满意
学院名称	信电工程学院
学号	20120511232
班级	12 软件 2 班
专业名称	计算机科学与技术
指导教师	鲍蓉

2016 年 6 月 3 日

徐州工程学院学位论文原创性声明

本人郑重声明： 所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用或参考的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标注。

本人完全意识到本声明的法律结果由本人承担。

论文作者签名：_____ 日期：_____年____月____日

徐州工程学院学位论文版权协议书

本人完全了解徐州工程学院关于收集、保存、使用学位论文的规定，即：本校学生在学习期间所完成的学位论文的知识产权归徐州工程学院所拥有。徐州工程学院有权保留并向国家有关部门或机构送交学位论文的纸本复印件和电子文档拷贝，允许论文被查阅和借阅。徐州工程学院可以公布学位论文的全部或部分内容，可以将本学位论文的全部或部分内容提交至各类数据库进行发布和检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

论文作者签名：_____ 导师签名：_____

日期：_____年____月____日 日期：_____年____月____日

摘要

现阶段,人们热衷于网上购物来满足日常生活。网上购物与线下购物有着明显的不同,其便利性、商品丰富度、安全性以及多付款渠道,非线下购物所能相比。本课题将积分系统与基本购物流程进行有机结合,设计并实现了一个网上购物商城系统,通过积分支付来取代普通购物网站的金钱支付手段,解决了支付过程中可能存在的安全性问题。

本系统采用 MySQL 作为系统开发的数据库,通过 Tomcat 进行网站发布,使用 JSP 技术实现系统界面的构建,通过 SSH(Struts+Spring+Hibernate)框架实现前端、后台以及数据库的交互。

本系统分为前台和后台,前台主要功能包括登录、商品浏览、个人信息管理、购物车、我的订单和积分支付,后台主要功能包括管理员登录、用户信息管理、商品管理、商品分类管理、订单管理。

关键词 SSH; 网上购物; 积分支付; MySQL

Abstract

At the present stage, most people are wild about shopping online to meet their daily life. Online shopping is clearly better than the offline shopping, such as convenience, goods types, safety and payment in many ways. This project design and achieve the shopping online system by combining integration system and basic process of shopping. The way paying by integration instead of ordinary cash solves the problem maybe exist in the payment progress.

The system uses MySQL as the system database, and it uses Tomcat to release website, and it also uses JSP technology to achieve the construction of the system interface, it implements front, back and database interaction through SSH (Struts + Spring + Hibernate) framework.

This system is split into two parts: foreground and background. The main function of foreground including: login, goods browsing, personal information management, shopping cart, my order and integration payment. The main function of background including: Administrator login, Users' information management, goods management, goods types management, orders management.

Keywords SSH Shopping online Integration payment MySQL

目 录

1. 绪论.....	1
1.1 系统的开发背景和意义.....	1
1.2 论文主要内容.....	1
2. 开发工具及技术简介.....	2
2.1 开发工具简介.....	2
2.1.1 ECLIPSE 简介.....	2
2.1.2 MYSQL 数据库简介.....	2
2.1.3 TOMCAT 简介.....	2
2.2 技术简介.....	2
2.2.1 JDBC 技术.....	2
2.2.2 MVC 模式.....	2
2.2.3 SSH 框架.....	3
3. 系统功能分析与设计.....	4
3.1 系统功能分析.....	4
3.2 系统功能设计.....	5
3.2.1 前台功能模块设计.....	5
3.2.2 后台功能模块设计.....	6
3.2.3 前后台操作交互设计.....	7
3.3 数据库设计.....	8
3.3.1 数据库概念模型设计.....	8
3.3.2 数据表设计.....	9
3.4 类文件设计.....	13
4. 系统的详细设计和实现.....	15
4.1 系统运行环境配置.....	15
4.2 前台功能模块实现.....	16
4.2.1 普通用户登录模块.....	16
4.2.2 商品查看模块.....	18
4.2.3 购物车模块.....	19
4.2.4 我的订单模块.....	20
4.2.5 下订单模块.....	21
4.3 后台管理模块设计.....	23
4.3.1 商品管理模块.....	23
4.3.2 订单管理.....	25
4.3.3 用户管理.....	26
5. 系统测试.....	28
5.1 登陆框非空值测试.....	28
5.2 关于订单状态可点击设置.....	28
5.3 关于订单使用积分付款的测试.....	28
结论.....	29
致谢.....	30
参考文献.....	31

1. 绪论

1.1 系统的开发背景和意义

随着互联网的普及,网上购物成为人们生活购物的首选,B2C 以及 C2C 等购物系统应运而生,其中如淘宝、京东、天猫...均已在网络购物中占据了无法撼动的地位,为促进网络消费,方便人们日常生活做出了极大的贡献。各购物网站的成功也从侧面证明了网上购物已经成为消费的主流,成为人们日常生活中不可分割的一部分。

再加上通过对中小型企业的调研发现,其内部或多或少的采用了员工积分系统来促进员工的工作积极性,但现阶段的员工积分也仅仅是与员工的薪资进行挂钩,严重制约了积分的灵活性,为开拓积分的使用渠道,以及为员工提供生活上的物质性的帮助,选择让积分与网上购物实现挂钩,方便中小型企业的内部积分购物系统的实现将成为必然。

本系统为企业积分提供了又一种使用渠道,必将促进员工的工作热情,自主自发的为企业的发展不遗余力。友好美观的网站界面,简单的用户操作,可以在员工休息之时达到放松心情的效果。多种商品的提供,通过低积分就能得到价值极高的商品,而且系统不会牵扯到现实货币,积分只要一经获得全部都可以自己做主选购自己喜欢的商品。本系统因为是基于中小型企业的内部,故省略了物流配送的流程,用户选购完成,企业的采购部或人事部将会在最短的时间内将商品送到企业员工的手里,彻底避免了物流配送过程中产生的丢包问题。本系统开发企业内部积分换购商城为企业员工提供切实可见的福利,具有一定的积极意义和应用价值。

1.2 论文主要内容

论文主要内容为以下五章:

第一章:绪论,主要介绍本课题研究的背景和意义。

第二章:简单介绍了开发过程中用到的工具及相关技术,包括 Eclipse、Tomcat、MySQL 以及 SSH 框架等。

第三章:介绍本系统的功能分析与总体设计。

第四章:介绍本系统的详细设计与实现方法。

第五章:介绍本系统的测试情况。

2. 开发工具及技术简介

本系统基于 SSH 框架等技术进行开发,使用 Eclipse 作为开发工具框架, MySQL 5.1 作为系统数据库, Tomcat v7.0 作为系统的 Web 服务器。

2.1 开发工具简介

2.1.1 ECLIPSE 简介

Eclipse 是一款开放源码的基于 java 的可拓展的开发平台,是一种框架,提供了丰富的插件集,不只是满足 Java 开发,也支持诸如 C/C++、PHP、Android 等编程语言,支持各种插件的使用,而且提供了丰富的 JSP 操作指令,良好的支持 Tomcat 等应用服务器的使用。本系统使用 Eclipse 实现系统关键代码的编写。

2.1.2 MYSQL 数据库简介

MySQL 是中小型企业普遍使用的数据库,为关系型数据库管理系统,MySQL 体积小、速度快、总体拥有成本低,而且操作简单。其开源特点成为中小型网站的不二选择。本系统使用 MySQL 实现系统数据库的建立。

2.1.3 TOMCAT 简介

Tomcat 也是一款开源的免费的 Web 应用服务器,属于轻量级应用服务器的范畴,是开发和 JSP 界面调试的首选。本系统使用的是 Tomcat v7.0,汲取了 Tomcat 6.0.x 优点并在其基础上进行优化,作为系统发布的服务器。

2.2 技术简介

2.2.1 JDBC 技术

JDBC 是一个用于执行 SQL 语句的 Java 接口,可以用于声明所使用的数据库,以及对数据库的用户名和密码,基本支持现阶段所有的数据库的使用。它提供的预处理 SQL,即 PreparedStatement,避免了 SQL 直接拼接可能存在的风险,JDBC 的执行流程基本上分为:加载 JDBC 驱动、连接数据库、执行 SQL 语句、返回结果集、关闭连接。

2.2.2 MVC 模式

MVC 模式即模型层(Model)+视图层(View)+控制层(Control),现阶段系统开发

均采用该种模式,使系统分层显得更加具有调理性,逻辑上也更为清晰,模型层里面声明系统运行所需要的各种实体类,视图层里面包括系统界面的设计,用户系统交互的触发,控制层用于数据库的各种操作,系统数据的增加删除查询修改。

MVC 模式如图 2-1 所示。

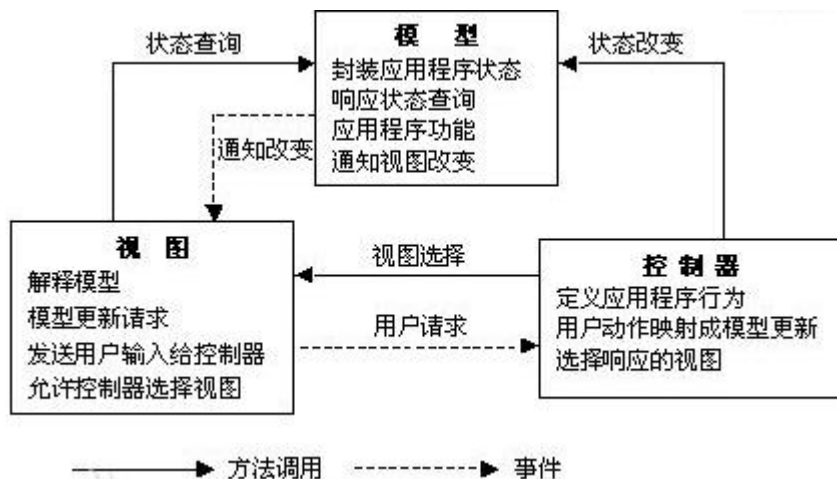


图 2-1 MVC 模式图

2.2.3 SSH 框架

SSH 框架即 Struts2+Spring+Hibernate, 是一款 Web 应用程序开源框架。

使用 SSH 框架进行开发的系统可以分为四层: 视图层, 业务逻辑层, 数据持久层和域模块层, 可以帮助开发人员在短期内搭建结构清晰、可复用性好、维护简单的 Web 程序。使用 Struts2 作为系统的基础架构, 负责 MVC 的严格分离, 使用 Struts2 进行界面实体的一一对应控制业务跳转, Hibernate 框架为数据持久层提供必要的支持, Spring 做管理, 管理 Struts2 和 Hibernate。

3. 系统功能分析与设计

3.1 系统功能分析

根据现存的网上购物系统进行分析,基本上购物系统都分为前台和后台两个部分,前台用于普通用户使用进行商品的浏览等一系列操作,后台用于管理员对用户商品订单的一部分操作。

前台是普通用户进行访问的界面,所以得按照用户的基本需求来设计前台。前台设计的一切功能都应该以方便用户为最终目的来设计。系统大体设计了以下功能:

(1) 用户登录:拥有账号和密码的用户可以直接输入用户名和密码来登录。登录之后就可以查看自己的购物车和订单情况以及自己持有的积分。也可以将商品加入购物车了并生成订单。

(2) 商品浏览:通过一级二级商品菜单分类,让用户直观的找到所需要的商品。

(3) 购物车:购物车是用来暂时存放用户喜欢但是还没有确定要生成订单的商品。或者是想要选购多种商品然后一起生成订单。购物车可以自动计算选购商品的总积分,便于客户了解。购物车里的商品也不一定要生成订单,购物车也应该提供删除不想要商品的功能。

(4) 下订单:用户根据个人积分的多少实现商品订单的生成,也就是下订单,当积分不足时,系统会提示用户积分不足,让用户不得不把赚取积分放在心上,这种方式也从侧面提高了用户工作的积极性。

(5) 我的订单:实时掌控用户订单的状态,包括是否付款,是否发货,是否交易完成等,而且可以分页显示所有的订单信息。

前台功能模块划分如图 3-1 所示。

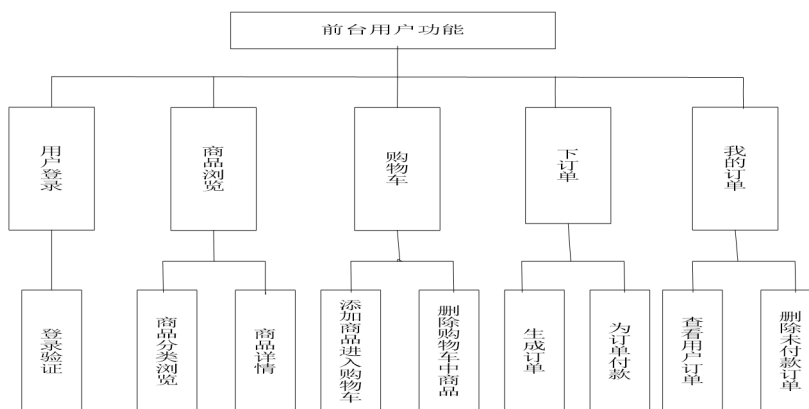


图 3-1 前台基本功能模块图

后台只有管理员才可以进入。管理员进入到后台后,首先是可以对系统所有普通用户进行操作,其次可以对订单进行详情查看或者对订单进行操作。还可以添加新的商品,更

新现有商品等操作。具体介绍如下：

(1) 管理员登录模块：严格限制只有管理员账号才能进入系统后台管理系统。

(2) 用户管理模块：管理员一进入到后台便可以看到普通用户信息，然后根据需求对用户进行一些操作。

(3) 商品管理模块：这个模块可以发布新的商品，编辑已存在的商品，删除下架的商品。网站的管理员可以发布新的商品，其次还可以对现有的商品进行编辑；

(4) 订单管理模块：网站前台进行的所有订单信息在后台都可以查询出来，管理员可以对相应的订单进行相应的处理，例如发货等操作。

(5) 前台商品分类管理模块：这个模块是将数据库中的所有商品按照相似属性进行细分类，目的是为了用户更便捷的查看商品。

管理员功能模块如图 3-2 所示。

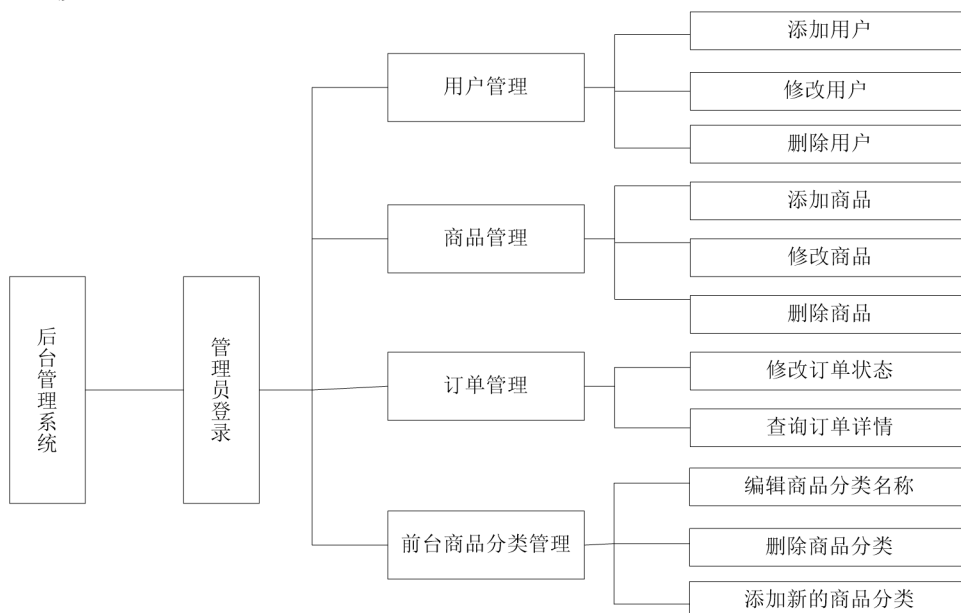


图 3-2 管理员功能模块图

3.2 系统功能设计

3.2.1 前台功能模块设计

前台功能模块主要包括：用户登录、商品浏览、添加购物车、下订单、查看订单等功能，总体操作流程介绍如下：

(1) 用户进入系统首页，进行商品浏览。用户可以根据首页的导航栏查看对应的相同分类的商品，用户点击商品对应图片可以进一步查看商品的详细信息，并且在该界面用户可以将商品添加进购物车。

(2) 用户将喜欢的商品加入购物车。购物车中已经存在该商品就使该商品的数量加一，并且计算该同类商品的总积分；若是购物车中不存在该商品，则将该商品添加进购物

车。购物车还提供删除商品以及清空购物车选项。

(3) 用户为购物车中的商品下订单。首先，系统先判断用户是否登录，用户只有已登录状态才能进行提交订单操作，用户为未登录状态时，系统应提醒用户只有登录才能进行该项操作。

(4) 登录成功的用户可以进行下订单操作，为订单支付相应的积分。

(5) 登录成功的用户可以进行我的订单查询操作，查看所有订单的状态，删除未付款或付款失败的订单。

前台基本操作流程如图 3-3 所示。

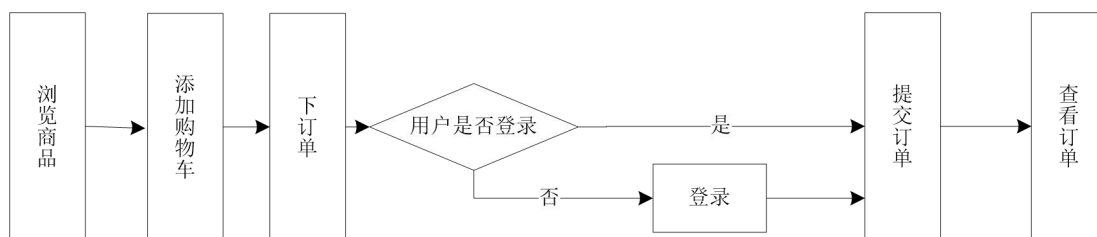


图 3-3 前台基本操作流程图

3.2.2 后台功能模块设计

3.2.2.1 用户管理模块设计

用户管理模块主要包括用户信息的查看，信息修改，添加新用户，删除用户等功能，操作流程介绍如下：

(1) 管理员可以对用户信息进行查询，一览用户信息。

(2) 管理员对用户进行修改操作，修改用户的部门，用户的真实姓名，用户的联系方式，用户的邮箱等。

(3) 管理员可以添加新的用户，按照员工的工牌号生成员工的登录账号，新用户对应部门信息。

(4) 管理员可以删除用户，例如员工离职，对用户进行逻辑删除操作。

3.2.2.2 商品管理模块设计

商品管理模块主要包括商品信息的查看、商品修改、添加新商品、删除商品等功能，操作流程介绍如下：

(1) 管理员查询商品信息，可以查看所有商品以及所有商品的详细信息。

(2) 管理员可以添加新的商品。

(3) 管理员修改商品信息，更新商品价格、图片、描述等内容。

(4) 管理员可以删除过时的商品，让商品下架，但用户已经下了单的商品不能被删除。

3.2.2.3 订单管理模块设计

订单管理模块主要包括订单查看, 订单状态修改等功能, 管理员不可删除用户订单, 订单信息归用户私人拥有。操作流程介绍如下:

- (1) 管理员可以查看所有用户的订单信息。
- (2) 管理员可以对已经支付成功的订单进行发货操作。
- (3) 管理员可以查看订单详细信息。

3.2.2.4 前台商品分类管理模块设计

商品分类管理模块主要包括商品分类名称查看, 分类名称修改, 添加新的商品分类名称, 删除分类名称等功能, 将商品分类分为一级商品分类和二级商品分类两项, 一级分类要清晰明了, 足够概括二级分类里面的所有内容。该项模块也将用于向前台导航栏提供导航信息, 导航信息对应的就是商品分类名称。操作流程介绍如下:

- (1) 管理员查看商品分类信息, 可以对商品分类名称进行修改。
- (2) 管理员可以添加新的商品分类名称, 细化商品分类。
- (3) 管理员可以删除过时的商品分类。

3.2.3 前后台操作交互设计

前后台交互主要包括订单信息的交互、商品信息的交互、导航栏信息的交互等模块, 具体介绍如下:

- (1) 前台用户提交订单给后台, 状态分为已付款和未付款, 后台管理员查看订单, 可以对已经付款的订单进行发货操作, 然后用户又可以在收到商品后进行完成订单操作。
- (2) 管理员修改商品信息, 或者添加新的商品, 又或是删除过时的商品时, 操作完成后, 前台的普通用户能够发现首页明显的变化。
- (3) 管理员对商品分类进行管理, 用户会从商品分类的变化获知导航信息更新。

前后台交互流程如图 3-4 所示。

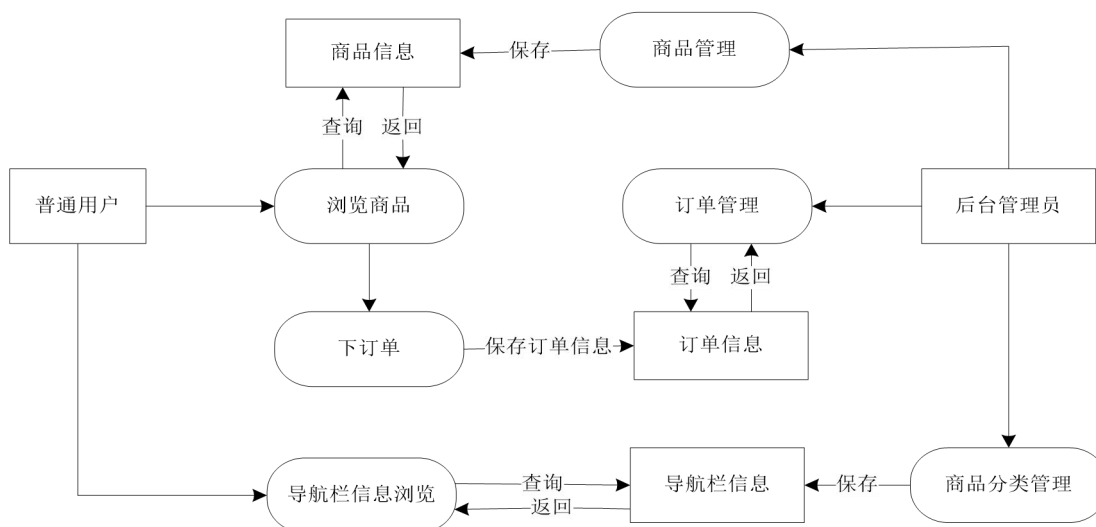


图 3-4 系统前后台交互流程图

3.3 数据库设计

3.3.1 数据库概念模型设计

网上商城涉及到很多数据的处理，数据库的设计尤为重要。

本系统涉及的实体对象包括：管理员、订单、订单明细、普通用户、商品、部门。

管理员实体包含的属性有：管理员编号、管理员姓名和管理员登录密码，如图 3-5 所示。



图 3-5 管理员实体图

订单实体包含的属性有：订单编号、订单总积分、订单日期、订单状态，如图 3-6 所示。

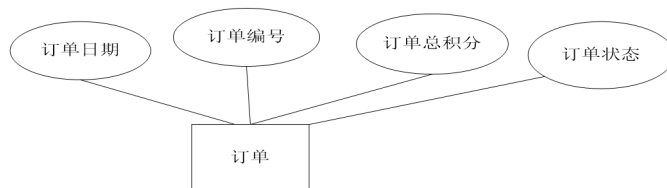


图 3-6 订单实体图

普通用户实体包含的属性有：用户编号、用户密码、用户账号、真实姓名、邮箱、用户状态、用户积分、联系方式，如图 3-7 所示。

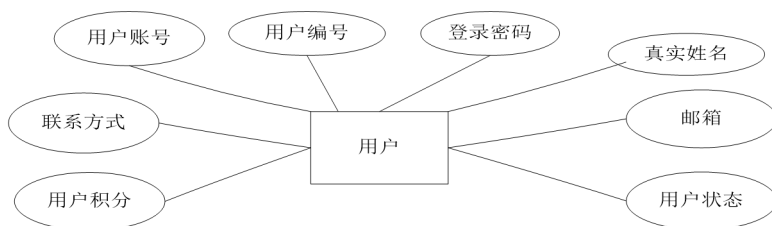


图 3-7 普通用户实体图

商品实体包含的属性有：商品编号、市场价格、商城积分、商品名称、图片路径、商品描述、是否热门、上架时间、商品分类、销量，如图 3-8 所示。

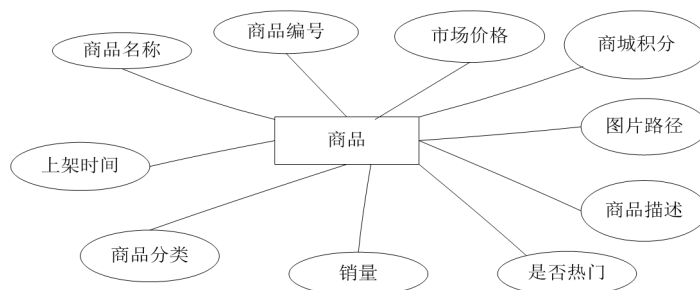


图 3-8 商品实体图

部门实体包含的属性有：部门编号、部门名称，如图 3-9 所示。

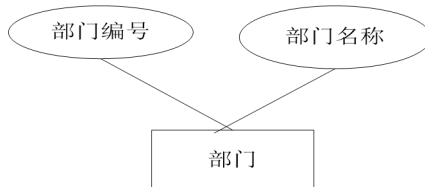


图 3-9 部门实体图

系统实体关系如图 3-10 所示。

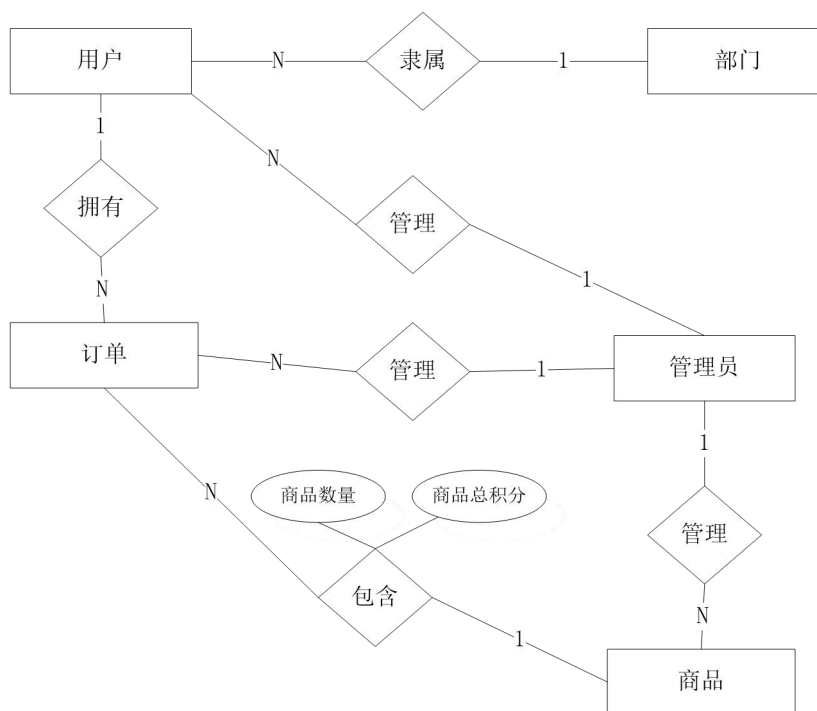


图 3-10 实体关系图

3.3.2 数据表设计

通过系统实体关系图分析得到表结构，其中包括前文提到的 5 个实体，又拓展出三张表，其中一张表用来存放订单明细信息，另外两张表用于进行商品细分类，这两张表分别

为一级分类表和二级分类表。各表结构详细信息如下：

(1) 管理员表用来存放管理员的登录名和密码，管理员表详细信息如表 3-1：

表 3-1 管理员表(adminuser)

字段名	含义	类型	是否主键
Uid	编号	Int	是
Username	登陆名	Varchar(50)	
Password	登陆密码	Varchar(50)	

(2) 订单表用来存放用户的订单以及购买日期，订单的信息应该还包括该笔订单的总价，该订单的状态，该订单对应的用户，用户的联系方式，用户所在的部门。订单表详细信息如表 3-2：

表 3-2 订单表(orders)

字段名	含义	类型	是否主键
oid	订单 id	Int	是
total	订单总价	Double	
ordertime	购买日期	Datetime	
state	订单状态	Int	
name	用户名	Varchar(50)	
phone	联系方式	Varchar(50)	
departmentId	部门编号	Int	部门外键
uid	用户编号	Int	用户外键

(3) 订单明细表用来存放相应的订单里包含的商品还有数量以及订单的编号和对应商品的总积分，订单明细表详细信息如表 3-3：

表 3-3 订单明细表(orderitem)

字段名	含义	类型	是否主键
itemid	编号	Int	是
count	数量	Int	
subtotal	商品总积分	Double	
pid	商品编号	Int	商品外键
oid	订单编号	Int	订单外键

(4) 普通用户表用来存放用户的真实姓名，部门 id，电话号码等基本信息，其中 state 字段用于后台用户的逻辑删除操作，points 字段即是用户持有积分用于商品换购，普通用户表详细信息如表 3-4：

表 3-4 普通用户表(user)

字段名	含义	类型	是否主键
uid	用户编号	Int	是
username	账号名称	Varchar(50)	
password	密码	Varchar(50)	

续表 3-4

字段名	含义	类型	是否主键
name	真实姓名	Varchar(50)	
email	邮箱	Varchar(50)	
phone	联系方式	Varchar(50)	
departmentId	部门编号	Int	部门外键
state	用户状态	Int	
points	用户积分	Int	

(5) 商品表用来存放商品的基本信息,其中包括商品编号、商品名称、商品描述、以及该商品图片对应的路径,其中商品图片均保存在用于系统发布的服务器上。商品表详细信息如表 3-5:

表 3-5 商品表(product)

字段名	含义	类型	是否主键
pid	商品编号	Int	是
pname	商品名称	Varchar(50)	
market_price	市场价格	Double	
shop_price	商城积分	Double	
image	图片路径	Varchar(50)	
pdesc	商品描述	Varchar(255)	
is_hot	是否热门	Int	
sales	销量	Int	

(6) 一级分类表用来存放商品的一级分类信息,一级分类表详细信息如表 3-6:

表 3-6 一级分类表(category)

字段名	含义	类型	是否主键
cid	一级分类编号	Int	是
cname	一级分类名称	Varchar(50)	

(7) 二级分类表用来存放商品的二级分类信息,使商品分类更加细致,二级分类表详细信息如表 3-7:

表 3-7 二级分类表(categorysecond)

字段名	含义	类型	是否主键
csid	二级分类编号	Int	是
csname	二级分类名称	Varchar(50)	
cid	所属一级分类编号	Int	一级分类表外键

(8) 部门表用来存放企业所有部门信息,其中包括部门的编号、部门的名称,部门表详细信息如表 3-8:

表 3-8 部门表(department)

字段名	含义	类型	是否主键
departmentId	部门编号	Int	是
departmentName	部门名称	Varchar(50)	

根据实体关系图可知,各表之间存在一定的联系,例如部门表的主键 departmentId 作为用户表和订单表的外键、用户表的主键 uid 作为订单表的外键、订单表的主键 oid 和商品表的主键 pid 作为订单明细表的外键、二级分类表的主键 csid 作为商品表的外键、一级分类表的主键 cid 作为二级分类表的外键。各表之间的关系如图 3-11 所示。

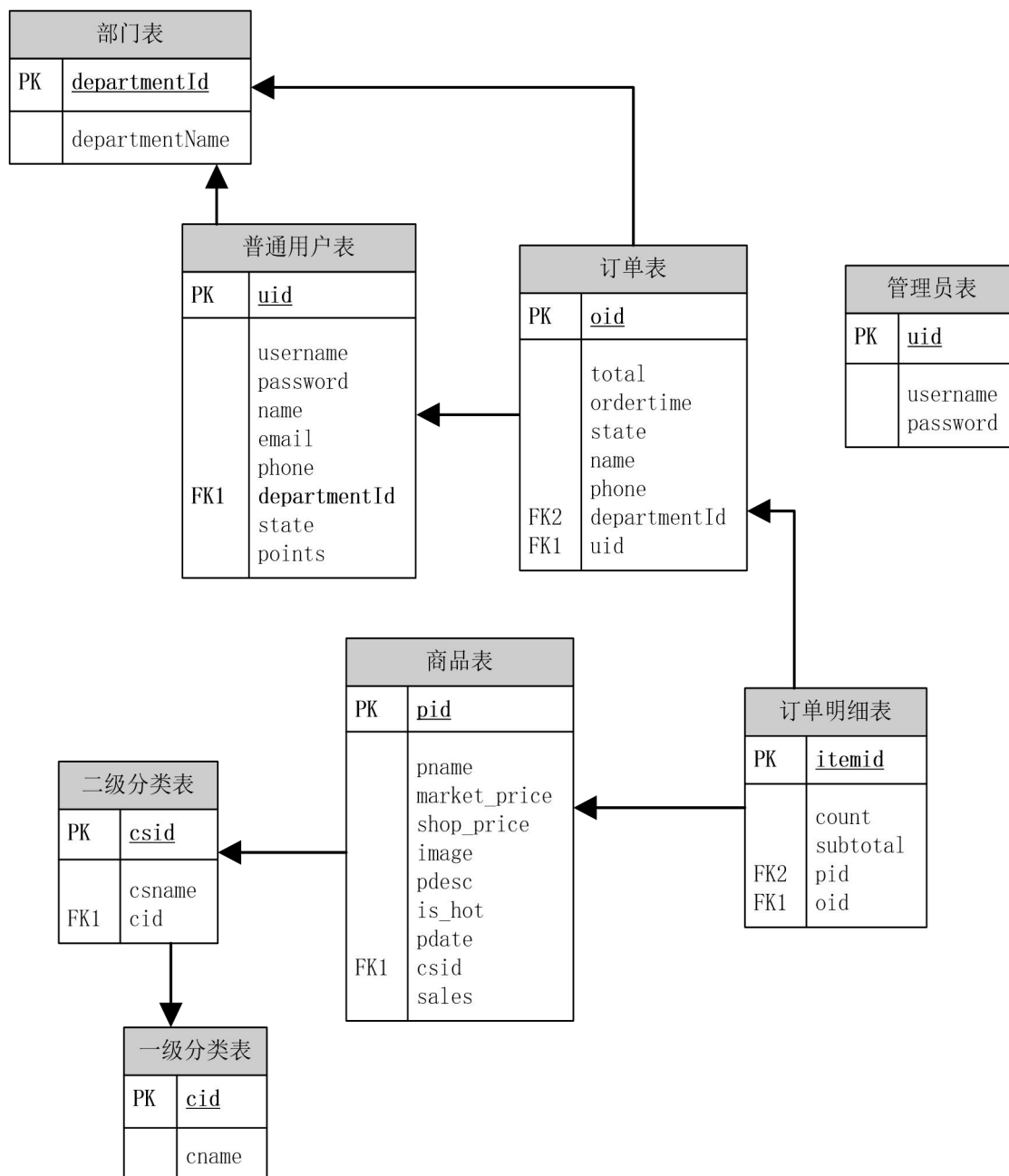


图 3-11 表关系图

3.4 类文件设计

该商城系统在 cn.xzit.com 包下共有 11 个包, 分别是 user 包、utils 包、product 包、order 包、interceptor 包、index 包、department 包、category 包、categorysecond 包、cart 包、adminuser 包。每个包里面都包含了很多的类, 现在挑几个较为重要的进行详细阐述。

(1) utils 包中含有 PageBean.java, PageHibernateCallBack.java, UUIDUtils.java 三个工具类。

UUIDUtils.java 类可以生成一串随机的字符串, 用于为验证码机制提供动态数据。

PageBean.java 类是用于分页的一个类, 里面含有分页的几个必要参数。

PageHibernateCallBack.java 类也是用于分页的类。

(2) interceptor 包中含有 PrivilegeInterceptor.java 类。

PrivilegeInterceptor.java 类是声明的一个后台校验拦截器, 主要是判断用户有无登录, 若是已登录可以继续登录, 若是没有登录则提示你没有该权限继续访问。

(3) index 包中含有 IndexAction.java 类。

IndexAction.java 类用于项目一开始启动时, 获取首页必须的信息, 例如导航栏信息, 商品推送等, 相当于帮助项目更好的实现初始化。

(4) user 包中含有 action 包、vo 包、service 包、dao 包。

action 包内含有 UserAction.java 类, 该类属于用户控制层用于接收前端数据, 数据库中获取的数据的判断也在该类中进行, 该类中有关于用户登录的方法还有用户退出的方法。

vo 包中含有 User.java 实体类还有与其相关的 User.hbm.xml 映射文件, 实现了与数据库中 User 表的个字段的一一对应。

service 包中含有 UserService.java 类, 该类属于用户模块业务层, 声明了好多查询用户, 更新用户等对用户进行操作的方法。

dao 包中包含 UserDao.java 类, 该类是把业务层中声明的方法进行具体的实现, 包括 SQL 语句的编写, hibernate 事物的开启和提交, 属于数据持久层的操作。

(5) order 包中也含有四个包, 分别是 action 包、vo 包、service 包、dao 包。

action 包中含有 OrderAction.java 类, 该类中声明了好多关于订单以及订单明细两个实体的操作, 包括订单添加, 订单删除, 订单的更新等操作。

vo 包中含有 Order.java, OrderItem.java, Order.hbm.xml, OrderItem.xml 两个实体类以及对应的映射文件。这两个类分别可以对订单实体和订单明细实体进行初始化。

service 包中只有 OrderService.java 类, 控制订单和订单明细模块的业务层。里面声明了对订单和订单明细的相关操作的方法, 交由数据持久层进行具体实现。

dao 包中包含 OrderDao.java 类, 里面要编写对订单和订单明细操的 SQL 语句, 借助 hibernate 的事务进行订单模块数据持久层的操作, 并将获取的结果返回给订单业务层。

(6) product 包中也是同样如此, 包含有 action 包、vo 包、service 包、dao 包。

action 包中包含 ProductAction.java 类, 该类实现对商品模块的控制, 包括商品删除、添加、修改等一系列操作。

vo 包中是商品的实体类以及其对的 hibernate 映射文件, 与数据库中的 product 的字段建立对应关系。

service 包中是 ProductService.java 类, 是商品模块的业务层, 业务层实现控制层的相应要求, 即创建有关商品的业务, 例如商品删除方法的声明, 商品添加, 商品更新等。

dao 包中的 ProductDao.java 类, 将业务层的方法进行具体实现, 编写相应的 SQL 语句, 达到数据持久层操作的目的。

(7) department 包中只有 Department.java 的实体类和其对应的映射文件 Department.hbm.xml, 该类就是用来表明用户的部门属性, 方便商品的派送, 实现企业内部购物的整体流程。

(8) category 包和 categorysecond 包中存放的是与商品分类实体有关的实体类以及操作的类, 实现了前台导航栏的实时更新, 方便用户的检索。CategoryDao.java 声明了一些 CategoryService.java 为实现控制层业务需求的各种方法, 实现了数据持久层有关于商品分类模块的各种数据操作。

(9) cart 包就是购物车相关的包, 其中实现了购物车的各种功能, Cart.java 和 CartItem.java 声明了购物车相关的实体, CartAction.java 类中实现了商品添加购物车, 清空购物车, 删除购物车中商品等各种操作。

(10) adminuser 包中包括了 AdminUser.java 实体和 AdminUser.hbm.xml 映射文件, 还有有关管理员登录, 管理员注销的 AdminUserAction.java 类, 以及该类相关的业务层类 AdminUserService.java 和数据持久层的 AdminUserDao.java 类。

4. 系统的详细设计和实现

4.1 系统运行环境配置

(1) 配置 JDBC 文件

配置内容如下:

```
jdbc.driver= com.mysql.jdbc.Driver
jdbc.url= jdbc:mysql:///shop
jdbc.user= root
jdbc.password=root
```

(2) 配置 C3P0 连接池以及 Hibernate 相关配置

配置内容如下:

```
<!-- 配置 C3P0 连接池: -->
<bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource">
    <property name="driverClass" value="{jdbc.driver}"/>
    <property name="jdbcUrl" value="{jdbc.url}"/>
    <property name="user" value="{jdbc.user}"/>
    <property name="password" value="{jdbc.password}"/>
</bean>
<!-- Hibernate 的相关信息 -->
<bean id="sessionFactory"
class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">
    <!-- 注入连接池 -->
    <property name="dataSource" ref="dataSource"/>
    <!-- 配置 Hibernate 的其他的属性 -->
    <property name="hibernateProperties">
        <props>
            <prop key="hibernate.dialect">org.hibernate.dialect.MySQLDialect</prop>
            <prop key="hibernate.show_sql">true</prop>
            <prop key="hibernate.format_sql">true</prop>
            <prop key="hibernate.connection.autocommit">false</prop>
            <prop key="hibernate.hbm2ddl.auto">update</prop>
        </props>
    </property>
```

(3) 为 web.xml 配置 Struts2 拦截器以及 Spring 监听器

配置内容如下:

```
<!-- 配置 Struts2 的核心过滤器 -->
<filter>
    <filter-name>struts2</filter-name>
    <filter-class>org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter</filter-class>
</filter>
<filter-mapping>
    <filter-name>struts2</filter-name>
```

```
<url-pattern>/*</url-pattern>
<!-- 请求转发都执行核心过滤器 -->
<dispatcher>REQUEST</dispatcher>
<dispatcher>FORWARD</dispatcher>
</filter-mapping>
<!-- 配置 Spring 的核心监听器 -->
<listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>
```

4.2 前台功能模块实现

4.2.1 普通用户登录模块

网站首页上提供了一个用于登录的超链接,用户一点击就会弹出一个登录窗口,拥有用户名和密码的用户可以直接登陆系统,点击登陆,用户输入的数据会提交到后台和数据库中的数据进行对比,如果确实有该条数据,则用户登录成功跳转回主界面,反之提示用户名或密码错误,返回登陆界面。

普通用户登录界面如图 4-1 所示。

图 4-1 普通用户登录界面

用户点击退出按钮之后应该在后台注销掉用户的 session,若是不注销,其他任意一个在 session 有效期内都可以使用该用户信息进行商品购买,由此会造成极大的风险,于用户的个人利益不符,而且系统不清除 session,这样也会占据系统资源,造成不必要的浪费。

普通用户登录流程如图 4-2 所示。

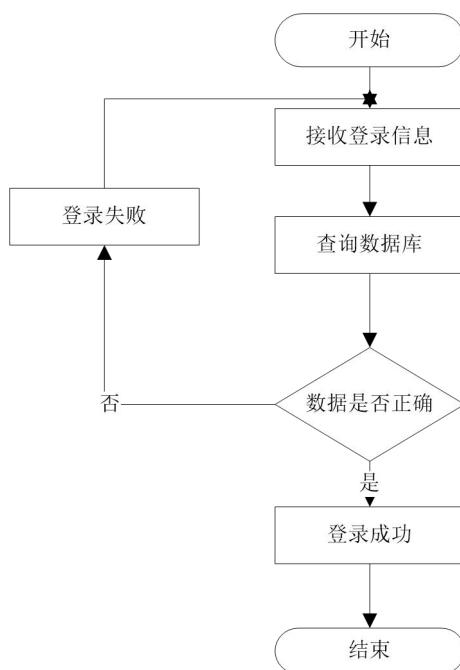


图 4-2 普通用户登录流程图

登录相关代码如下：

```

public String login() {
    User existUser = userService.login(user);
    // 判断
    if(existUser == null) {
        // 登录失败
        this.addActionError("用户名密码错误!");
        return LOGIN;
    } else {
        // 从 session 中获得验证码的随机值
        String checkcode1 = (String) ServletActionContext.getRequest()
            .getSession().getAttribute("checkcode");
        if(!checkcode.equalsIgnoreCase(checkcode1)) {
            this.addActionError("验证码输入错误!");
            return "checkcodeFail2";
        }
        // 登录成功获取部门信息
        Department
        department=userService.getDepartmentById(existUser.getDepartmentId());
        // 将用户的信息存入到 session 中
        existUser.setDepartmentName(department.getDepartmentName());
        ServletActionContext.getRequest().getSession()
            .setAttribute("existUser", existUser);
        // 页面跳转
        return "loginSuccess";
    }
}
  
```

4.2.2 商品查看模块

无论用户登录与否都能对该模块进行操作, 根据一级导航, 二级导航直接定位选择的商品, 点击商品可以返回商品的详细信息, 包括市场价格, 该种商品在本商城中所需要的积分, 以及商品的详细描述。

商品信息界面如图 4-3 所示。



图 4-3 商品信息界面

查看商品流程如图 4-4 所示。

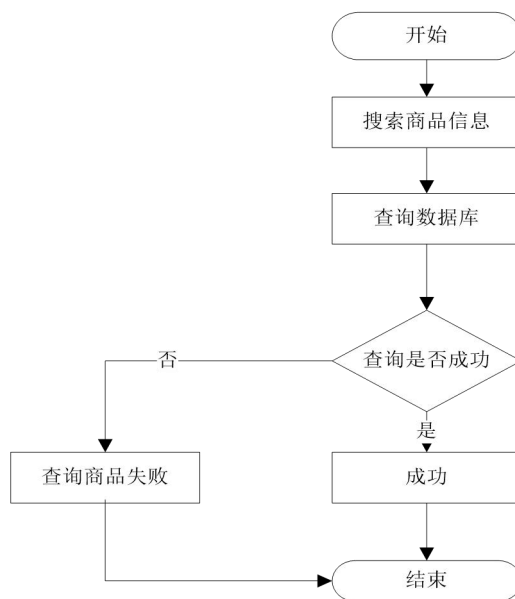


图 4-4 查看商品流程图

商品查看相关代码如下:

```

//根据商品 id 获取商品详细信息
public String findByPid() {
    // 调用 Service 的方法完成查询.
    product = productService.findByPid(product.getPid());
    return "findByPid";
}
  
```

4.2.3 购物车模块

购物车用来方便用户暂存喜欢的商品，顾名思义就是放东西的车子。用户可以把一时感兴趣的商品放进去，买不买无所谓，没有兴趣的时候也可以将商品从购物车中删除，已经添加的商品会在原有的基础上数量加 1，所有添加进入购物车的商品，购物也会帮助用户计算所挑选商品的总价，可以给用户提供极大的便利。

购物车界面如图 4-5 所示。



图 4-5 购物车界面

购物车的流程如图 4-6 所示。

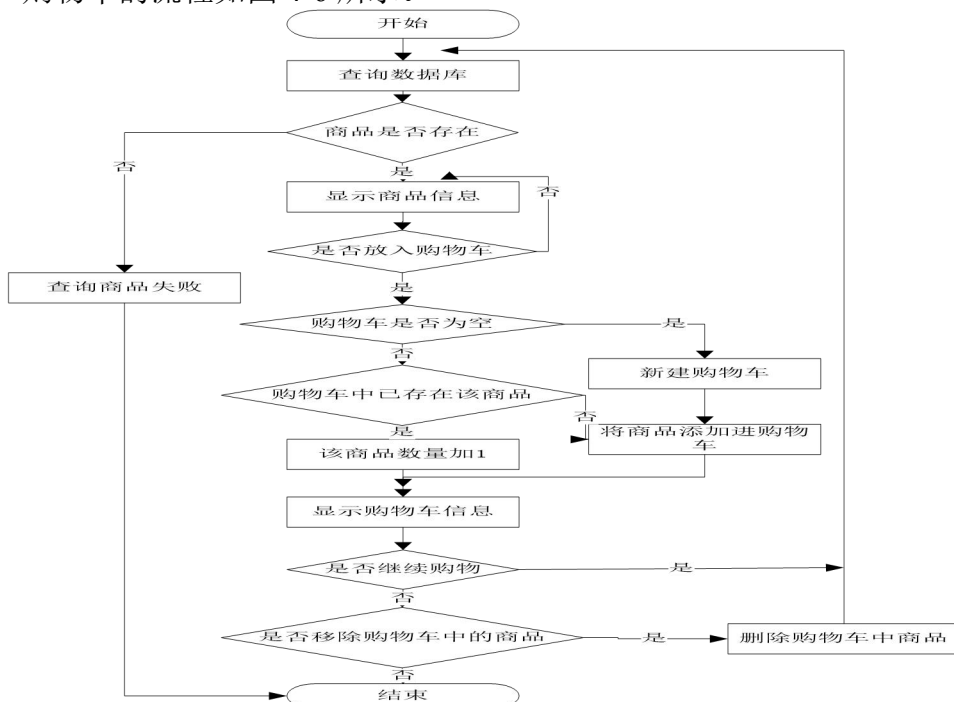


图 4-6 购物车模块流程图

购物车相关代码如下：


```

// 1.将购物项添加到购物车
public void addCart(CartItem cartItem) { // 判断购物车中是否已经存在该购物项
    /*如果存在: 数量增加 总计 = 总计 + 购物项小计
    如果不存在: 向 map 中添加购物项 总计 = 总计 + 购物项小计*/
    // 获得商品 id.
    Integer pid = cartItem.getProduct().getPid(); // 判断购物车中是否已经存在该购物项:
    if (map.containsKey(pid)) { // 存在
        CartItem _cartItem = map.get(pid); // 获得购物车中原来的购物项
        _cartItem.setCount(_cartItem.getCount() + cartItem.getCount());
    } else { // 不存在
        map.put(pid, cartItem);
    }
    total += cartItem.getSubtotal(); // 设置总计的值
}

// 2.从购物车移除购物项
public void removeCart(Integer pid) {
    CartItem cartItem = map.remove(pid); // 将购物项移除购物车
    total -= cartItem.getSubtotal(); // 总计 = 总计 - 移除的购物项小计
}



// 3.清空购物车
public void clearCart() {
    map.clear(); // 将所有购物项清空
    total = 0; // 将总计设置为 0
}

```

4.2.4 我的订单模块

用户提交的订单可以在我的订单模块查询出来。点击“我的订单”按钮，就会全部分页显示出来。订单显示界面会显示所有的订单编号，每个订单的订单总积分，订单的状态，是否付款，是否发货，是否交易成功等。

我的订单界面如图 4-7 所示。

我的订单				
订单编号:9037 订单金额:12.0 确认收货				
图片	商品	积分	数量	小计
	女款短靴3	12.0	1	12.0分
订单编号:9036 订单金额:12.0 交易成功				
图片	商品	积分	数量	小计
	女款短靴4	12.0	1	12.0分

第1/1页 1

图 4-7 我的订单界面

我的订单模块流程如图 4-8 所示。

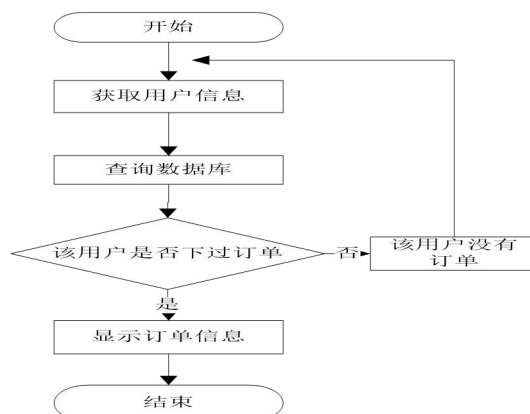


图 4-8 我的订单查询模块流程图

我的订单查询模块相关代码如下：

```

// 查询我的订单
public String findByUid() {
    User existUser = (User)
ServletActionContext.getRequest().getSession().getAttribute("existUser");// 获得用户
    Integer uid = existUser.getUid();// 获得用户的 id
    // 根据用户的 id 查询订单
    PageBean<Order> pageBean = orderService.findByUid(uid, page);
    // 将 PageBean 数据带到页面上
    ActionContext.getContext().getValueStack().set("pageBean", pageBean);
    return "findByUid";
}

```

4.2.5 下订单模块

下订单模块实现商品的积分支付功能，用户通过购物车内的商品向数据库提交订单，后台数据库会扣取用户拥有的积分，积分不足时无法完成支付功能，订单的状态应该显示为未付款状态。

下订单相关模块如图 4-9 到 4-11 所示。

生成订单成功

图片	商品	积分	数量	小计
	商务修身羊毛呢子风衣 中长款呢大衣外套	29.0	2	58.0

商品积分: 58.0分

部门:

收货人:

联系方式:

确认订单

图 4-9 订单待确认界面

用户积分不足时，系统提醒用户积分不足无法购买，如图 4-10 所示。

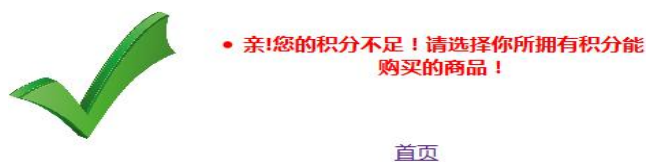


图 4-10 积分不足提示界面

用户积分支付成功时提示界面如下图 4-11 所示。

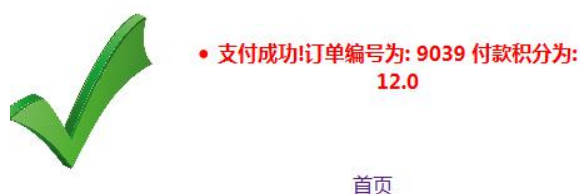


图 4-11 支付成功提示界面

下订单模块流程如图 4-12 所示。

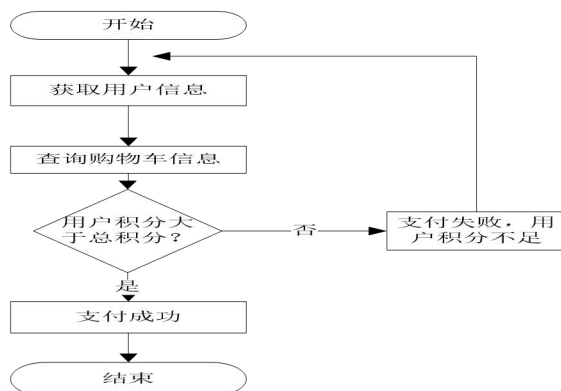


图 4-12 下订单模块流程图

下订单模块相关代码如下：

```

// 为订单付款
public String payOrder() throws IOException {
    // 1.修改数据
    Order curOrder = orderService.findByOid(order.getOid());
    curOrder.setDepartmentId(order.getDepartmentId());
    curOrder.setName(order.getName());
    curOrder.setPhone(order.getPhone());
    orderService.update(curOrder); // 修改订单
    // 2.完成付款
    User existUser = (User)
    ServletActionContext.getRequest().getSession().getAttribute("existUser");
    System.out.println(existUser.getPoints());
}
    
```

```
if(existUser.getPoints()<currOrder.getTotal()){//支付失败，用户积分不足
    this.addActionMessage("亲!您的积分不足!请选择你所拥有积分能购买的商品!");
}else{//支付成功
    int total=(int)(existUser.getPoints()-currOrder.getTotal());
    existUser.setPoints(total);
    userService.update(existUser);
    currOrder.setState(2);
    orderService.update(currOrder);
    this.addActionMessage("支付成功!订单编号为: " + currOrder.getOid() + " 付款积分为: " + currOrder.getTotal());
}
return "msg2";
}
```

4.3 后台管理模块设计

4.3.1 商品管理模块

该模块我们就商品发布模块进行详细分析，添加商品我们就要添加商品的各种信息，例如商品描述，商品市场价格，该商品在本商城的积分定位，商品图片，商品所属的二级分类，主要的属性缺一不可。

商品添加界面如图 4-13 所示。

添加商品			
商品名称：	<input type="text"/>	是否热门：	<input type="button" value="是"/>
市场价格：	<input type="text"/>	商城价格：	<input type="text"/>
商品图片：	<input type="button" value="浏览..."/> 未选择文件。		
所属的二级分类：	<input type="button" value="潮流女装"/>		
商品描述：	<div></div>		
<input type="button" value="确定"/> <input type="button" value="重置"/> <input type="button" value="返回"/>			

图 4-13 商品添加界面

商品添加流程如图 4-14 所示。

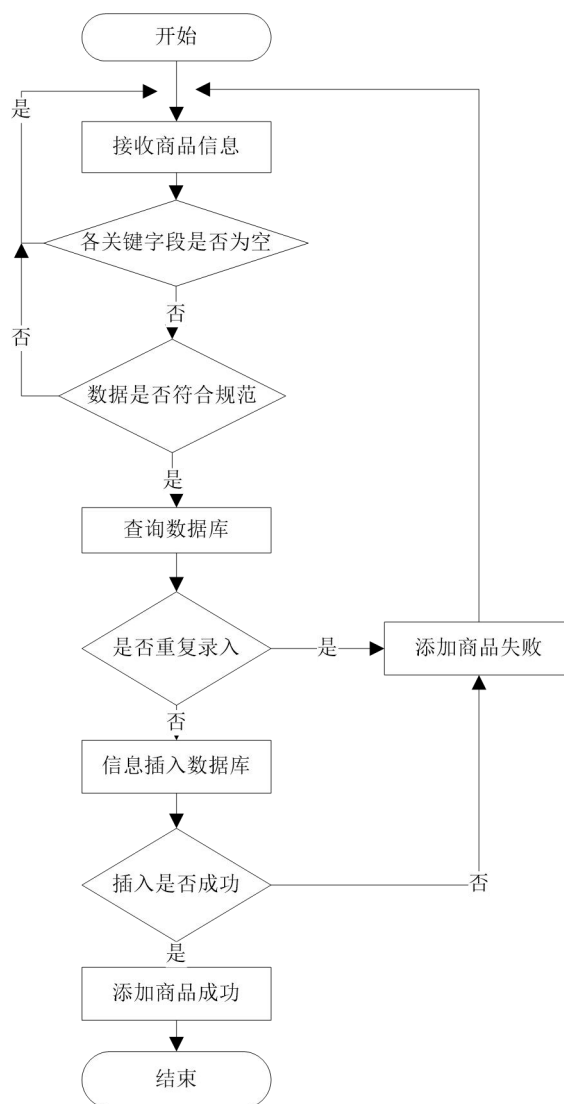


图 4-14 商品添加流程图

商品添加相关代码如下：

```

// 保存商品的方法
public String save() throws IOException {
    product.setPdate(new Date()); // 将提交的数据添加到数据库中。
    if(upload != null){
        // 将商品图片上传到服务器上。
        // 获得上传图片的服务器端路径。
        String path = ServletActionContext.getServletContext().getRealPath(
            "/" + product.getImage());
        String path2 = "E:\\workspace\\shop1\\WebContent\\products";
        // 创建文件类型对象
        File diskFile = new File(path + File.separatorChar + uploadFileName);
        File diskFile2 = new File(path2 + File.separatorChar + uploadFileName);
        // 文件上传，上传的时候文件的原始路径也要上传
        FileUtils.copyFile(upload, diskFile);
        FileUtils.copyFile(upload, diskFile2);
    }
}

```

```

        product.setImage("products/" + uploadFileName);
    }
    productService.save(product);
    return "saveSuccess";
}
    
```

4.3.2 订单管理

管理员登录后可以查询前台普通用户提交的订单信息，只有订单是已经付款成功的他才能进行商品的发货操作，可以查看订单的详细信息。

修改订单状态界面，如图 4-15 所示。

订单列表					
序号	订单编号	订单金额	收货人	订单状态	订单详情
1	9039	12.0	十五	发货	订单详情
2	9038	58.0	十五	未付款	订单详情
3	9037	12.0	十五	等待确认收货	订单详情
4	9036	12.0	十五	订单完成	订单详情
5	9035	12.0	十五	发货	订单详情
6	9034	12.0	十五	发货	订单详情
7	9004	119.0	??	未付款	订单详情
8	9003	5099.0	姜涛	订单完成	订单详情
9	9002	228.0	??	未付款	订单详情
10	9001	255.0		未付款	订单详情

第1/2页 下一页 尾页

图 4-15 修改订单状态界面

修改订单状态流程如图 4-16 所示。

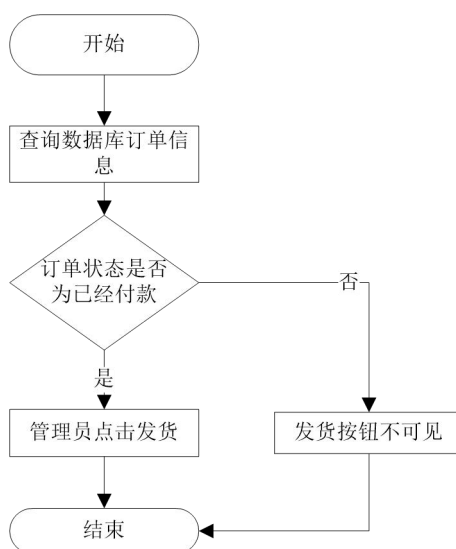


图 4-16 修改订单状态流程图

修改订单状态相关代码如下：

```

// 修改订单状态
public String updateState(){
    // 根据 id 查询订单
    Order curOrder = orderService.findByOid(order.getOid());
    
```

```
currOrder.setState(3);  
orderService.update(currOrder);  
// 页面跳转  
return "updateStateSuccess";  
}
```

4.3.3 用户管理

管理员登录后台管理系统后可以对用户进行管理,其中包括用户删除,用户修改,用户添加,其中用户修改功能中用户的账号和密码是不可修改的,只要用户一被添加进数据库,用户的账号和密码都不可变更,将账号与员工所在的企业工号挂钩,技能保证用户个人信息,又方便于统一管理。

用户添加界面如图 4-17 所示。

添加用户 USER ADD

* 用户ID:

* E-mail:

真实姓名:

手机:

部门: 开发事业部 ▼

确认

图 4-17 添加用户界面

用户管理模块就用户添加进行详细分析,用户添加流程如图 4-18 所示。

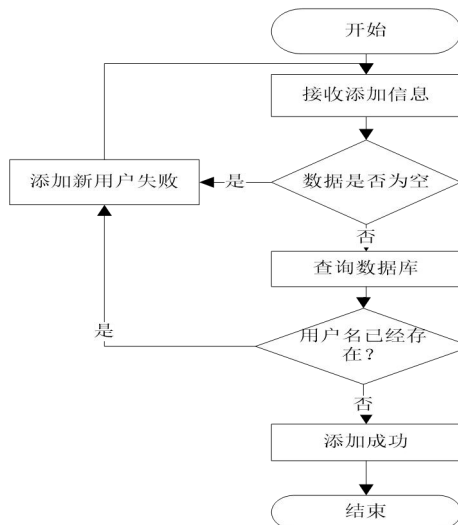


图 4-18 添加用户流程图

添加用户相关代码如下：

```
//用户添加的方法
public String regist(){
    //获取部门id
    String departId=ServletActionContext.getRequest().getParameter("departId");
    user.setDepartmentId(Integer.parseInt(departId));
    user.setPassword("888888");
    user.setPoints(0);
    user.setState(1);
    //保存用户
    userService.save(user);
    this.addActionMessage("添加成功!");
    return "addsuccess";
}
```


5. 系统测试

5.1 登陆框非空值测试

此功能针对系统登陆时，输入框用户名，密码不能为空进行测试。测试的前提条件为（用户名：aaa，密码：aaa）

测试情况如下：

（1）用户进入登录界面，若是不输入用户名，则当点击登录按钮时，提示用户名不能为空。

（2）用户进入登录界面，若是不输入密码，则当点击登录按钮时，提示密码不能为空。

（3）用户进入登录界面，若是输入错误账号或密码，则当点击登录按钮时，提示用户名密码错误。

（4）用户进入登录界面，若是输入正确账号或密码，则当点击登录按钮时，登录成功进入系统主页。

5.2 关于订单状态可点击设置

该功能主要针对订单的各个状态进行测试。

前置条件：用户只有支付成功的订单，管理员才可以点击发货按钮。

（1）管理员进入订单管理界面，点击订单管理，查看所有订单，未付款、已发货、交易完成三个状态均不可点击。

（2）管理员进入订单管理界面，点击订单管理，查看所有订单，已付款订单显示可点击的发货按钮。

5.3 关于订单使用积分付款的测试

此功能针对用户只能购买自身持有积分范围内的商品。

前置条件：购物车内商品已经生成订单，进入订单支付界面。

（1）用户进入订单支付界面，系统计算订单总积分，当订单积分超过用户持有积分时，系统提示用户，支付失败。

（2）用户进入订单支付界面，系统计算订单总积分，当订单积分不超过用户持有积分时，系统提示用户，支付成功。

结论

通过本次毕业设计，切实是收获到了很多东西，无论是技术上的还是心理上的，一点点的准备毕业设计各个阶段的不同文档，学习开题申报系统所需要的各种专业知识，前端界面的报错修改，指导老师的督促，让我一刻都不敢松懈。不过随着系统一步步的完成，成就感也是油然而生，这是我不断努力成果。

系统完成了任务书以及开题报告的要求，解决了图片难以存入数据库的问题，通过使用 Ajax 技术，实现了不刷新当前页面也能进行信息获取及验证的操作，使系统的业务逻辑更加完善。数据库的建立能够满足一整套业务流程的使用，切实实现了网上购物与中小企业的相互结合。

整个毕业设计，让我将无论是在校内还是在实习企业学到的专业知识进行了一次系统的整合，明白了整个系统的工作流程的各个阶段的执行方式，登录注册的验证，让我以后的工作都会更加严谨，避免各种让人啼笑皆非的错误的发生，系统开发过程中让我再一次深入理解了 hibernate 的数据库操作机制，数据持久层的便利性和安全性有了更进一步的领悟。spring 的事物托管，以及它对 struts2 和 hibernate 的整合，为系统开发提供了极大的便利。

后期根据实际的业务逻辑需要，去除了任务书中不必要的用户注册还有后台的对管理员进行管理这两部分的功能。

系统虽然已经完成，但一定还存在各种各样的不足之处。例如，前期开发的市场调研不够充分，系统并不能满足所有企业的需求；部门间的包裹配送过程中的丢包问题没有得到解决；积分发放机制考虑的不够完善等等问题。

致谢

经过两三个月的不断努力，终于完成了毕业设计的系统以及毕业论文，在这阶段遇到了各种各样的问题，感谢校内老师和校外老师的指导，感谢公司同事和学校同学的帮助。

无论在校外实习，还是在校内学习，都要时刻和指导老师保持联系，因为老师给你毕业设计的意见是不可忽略的，无论是前期的系统开发还是后期论文的完成，老师的帮助贯穿了整个毕业设计阶段。一开始的开题不够清晰，导致走了好多弯路，还好及时与老师进行交流，让我知道了系统的着眼点，突出体现在积分换购上。“没有亮点的毕设不是好毕设”，老师这句话铭记于心。

在这里要首先感谢我的指导老师鲍蓉老师，要不是老师不厌其烦的帮助我，我甚至都不知道毕业设计要从何着手，还记得一开始的时候，老师说跟我说说你的毕设开题，我直接没怎么想就直接报了个课题上去，当老师问我我的系统亮点在哪，和现有的同类系统有什么不同，当时我哑口无言，一开始的那种随便应付的心态再也不敢有了。是鲍蓉老师让我明白了毕业设计的重要性，让我知道毕业并不是随便对付下就行的，甚至给我指出了系统开发过程中的注意事项，帮我找到了系统的着眼点。在这里对我的指导老师表示由衷的感谢！

其次感谢校内的同学，他们帮我测试出了系统的好多 bug，并帮我思考解决掉了这些问题，测试本就是一个繁重的活计，我自己一个人不可能面面俱到，还好有他们，他们的细致，他们的不吝帮助，在这里谢谢你们了。

参考文献

- [1]钱乐秋 编著.软件工程[M].清华大学出版社,2012
- [2]王珊,萨师煊 编著.数据库系统概论[M].高等教育出版社,2014
- [3]李宁 编著.Java Web 编程实战宝典[M].清华大学出版社,2014
- [4]李刚 编著.Strut2.X 权威指南[M].电子工业出版社,2012
- [5]许勇 编著.Struts 2+Hibernate+Spring 整合开发 深入剖析与范例应用[M].清华大学出版社, 2013
- [6]曹化宇 编著.数据应用开发详解.网络资源:图灵社区 ,2015
- [7]林信良 编著.JSP & Servlet 学习笔记[M].北京:清华大学出版社,2012
- [8]Metsker S J 编著. Java 设计模式[M] .北京:电子工业出版社,2012
- [9]陈正举 编著.基于 HIRBERNATE 的数据库访问优化[J] 计算机应用与软件,2012
- [10]詹青龙,刘建卿 编著.物联网工程导论[M].北京:清华大学出版社,2012
- [11]淘宝网 淘宝公告 <http://bbs.taobao.com>
- [12]百度百科 <http://baike.baidu.com/view/5052997.html>
- [13]徐明华等编著. Java Web 整合开发与项目实战[M].人民邮电出版社, 2010
- [14] (美) Y. Daniel Liang 编著.Java 语言程序设计[M].机械工业出版社, 2011
- [15]计文柯 编著.深入解析 Spring 架构与设计原理[M].机械工业出版社, 2010
- [16]李兴华 等编著. Java Web 开发实战经典基础篇[M].清华大学出版社, 2010
- [17]张晓菲 等编著.锋利的 jQuery(第 2 版)[M].人民邮电出版社, 2012
- [18]张志锋 等编著.Java Web 技术整合应用与项目实战[M].清华大学出版社, 2013
- [19]道格拉斯·克罗克福德 编著. JavaScript 语言精粹(修订版)[M].人民邮电出版社,2013