**北京理工大学现在远程教育学院**

**毕业实践报告**

**题目：网上商城系统设计**

**专 业：计算机信息管理**

**班 级： 20130111011**

**姓 名： 刘旋**

**学 号： 20130111011**

**导 师：姜增如**

**日 期：2016年04月**

**学生毕业实践（报告）评语表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 姓 名 | 刘旋 | 学 号 | 20130111011 | 班 级 | 20130111011 | |
| 第一阶段  （40分） | | | 第二阶段  （60分） | | 合 计  （100分） | |
|  | | |  | | |  |
| 毕业实践（报告）评语  刘旋同学以网上商城系统为毕业实践，对系统的功能模块进行了划分。设计了系统前台用户功能结构图和系统后台管理员功能结构图。设计了系统的数据库，画出了程序流程图，以JSP为开发平台开发了应用界面，使用Mysql环境建立了数据库，设计的应用程序模块包括：注册登录、商品购买、订单查询、修改个人密码、留言、商品分类会员等管理。该毕业实践符合计算机信息管理大专要求，同意毕业。  指导老师（签字）： 年 月 日 | | | | | | |
| 毕业实践（报告）成绩  指导老师组长（签字）： 年 月 日 | | | | | | |

**北京理工大学继续教育暨现代远程教育学院**

**专科学生毕业实践（报告一）**

学生姓名：刘旋 指导教师：姜增如 学号：20130111011号专 业：计算机信息管理

# 一、课题目的和意义

网上购物系统，投资少，回收快。一项针对中国中小企业的情况调查显示，个人在网下启动销售公司的平均费用至少5万元，而网上开店建店成本非常小。一般说，筹办一家网上的商店投入很小，不用去办营业执照，不用去租门面，不用囤积货品，所需资金不过1500元左右；网上商店比同等规模的地面商店“租金”要低得多，同时租金不会因为营业面积的增加而增加，投资者也不用为延长营业时间而增加额外的费用。基本不需要占压资金。传统商店的进货资金少则几千元，多则数万元，而网上商店则不需要压资金。24小时营业时间。网上商店延长了商店的营业时间，一天24小时、一年365天不停地运作，无须专人值班看店，都可照常营业。传统店铺的营业时间一般为8-12小时，遇上坏天气或者老板、店员有急事也不得不暂时休息。销售规模不受地盘限制。传统商店有多大就只能摆放多少商品，生意大小常常被小店面积限制。而在网上，即便在地面上只有一个小商店，或者干脆就没有门面，开店的生意却可以照样做得很大。不受店面空间的限制。哪怕只是街边小店，在网上却可以拥有百货大楼那么大的店面，只要投资者愿意，可以摆上成千上万种商品。目前国内最大的专业拍卖网站同时在线的商品要超过10万件——已超过一些大超市。不受地理位置影响。不管客户离店有多远，也不管顾客是国内还是国外，在网上，客户一样可以很方便地找到并购买商品。这令消费群体突破了地域的限制，变得无限广阔了。随着近年来网络媒体与网络购物平台的合作加深，消费者从获取资讯到发生购物行为的时空距离被进一步缩短，由此带来的巨大经济效益使得各大网购平台越来越重视网络产品销售。首先，网络销售成本优势。网上产品群有很高的利润空间，例如出版社的库存图书，网站也可以包销、定制、买断产品来做低价销售。同时，针对新产品线的百货、礼品、饰品等可以做贴牌销售，毛利空间很高，弥补了其他产品线低价的损失；其次 网上购物无店面成本，并且可以根据客户需求进行针对性的跟踪推广，市场广告成本比较低，整体的运营成本低。同时，网上购物给用户提供方便的购买途径，只要简单的网络操作，足不出户，即可送货上门，并具有完善的售后服务。货到付款，使网上购物的安全性得到了保障。这些都是顾客热衷网上购物和网络销售快速增长的原因。

# 二、本课题主要研究内容和预期目标

**2.1研究内容：**

经过对各购物网站的考察、分析，要求本系统满足以下要求：

（1）统一友好的操作界面，保证系统的易用性，方便用户的操作；

（2）具备产品信息的展示功能，方便浏览者对产品信息进行浏览与比较；

（3）规范的产品分类，方便浏览者分类查找商品；

（4）完善的购物车功能与用户结账功能；

（5）设置网站公告信息模块，使浏览者能够及时了解网站的各项动态变化，及销售的新品、特价活动等信息；

（6）商品销售排行功能；

（7）订单信息查询功能、商品信息查询功能；

（8）全面的后台管理功能，以方便管理员对网站信息进行更新与管理。

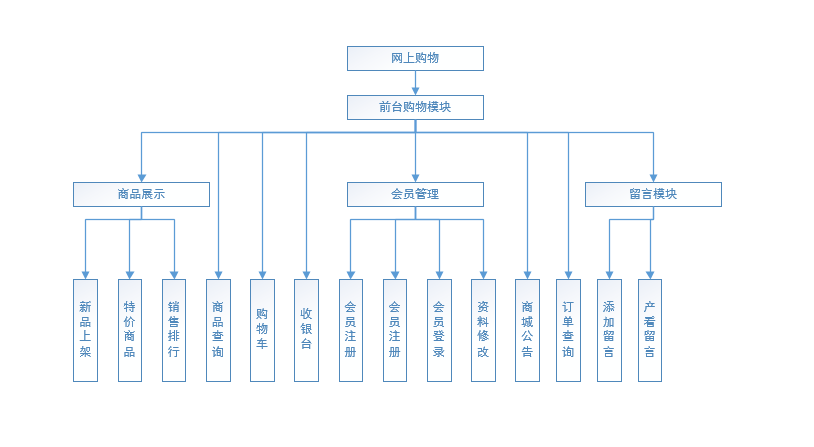
**2.2总体功能结构：**

图2.2.1前台系统功能图

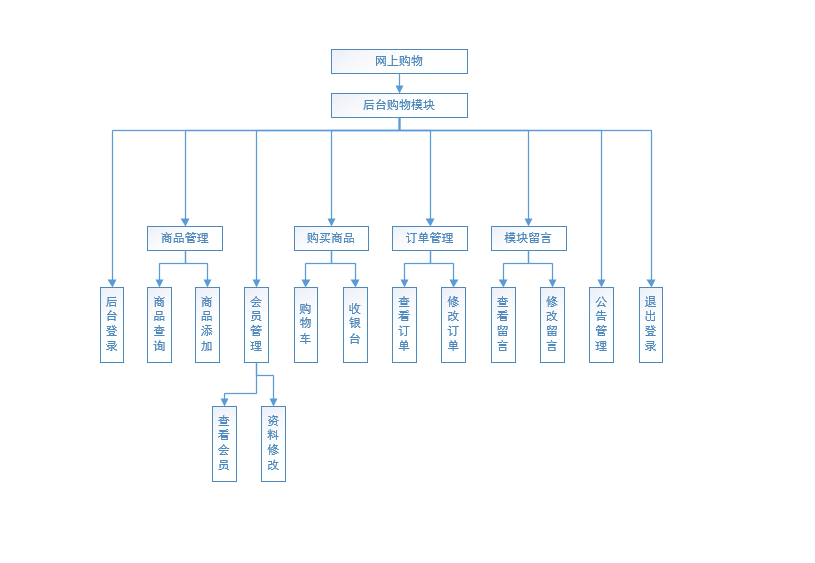


图2.2.2后台系统功能图

1、MyEclipse

MyEclipse免费、程序代码排版功能、有中文化包、可增设许多功能强大的外挂、支持多种操作系统，它是功能丰富的J2EE集成开发环境，包括了完备的编码、调试、测试和发布功能，完整支持HTML, Struts, JSF, CSS, Javascript, SQL。

2、网页排版工具—— Macromedia Dreamweaver 8

Macromedia Dreamweaver 8 是建立 Web 站点和应用程序的专业工具。它将可视布局工具、应用程序开发功能和代码编辑支持组合在一起，其功能强大，使得各个层次的开发人员和设计人员都能够快速创建界面吸引人的基于标准的网站和应用程序。从对基于 CSS的设计的领先支持到手工编码功能，Dreamweaver 提供了专业人员在一个集成、高效的环境中所需的工具。开发人员可以使用 Dreamweaver 及所选择的服务器技术来创建功能强大的 Internet 应用程序，从而使用户能连接到数据库、web服务。

3、Mysql数据库

MySql的核心程序采用完全的多线程编程，可运行在不同的操作系统下，有一个非常灵活而且安全的权限和口令系统，支持ODBC for Windows，支持大型的数据库，拥有一个非常快速而且稳定的基于线程的内存分配系统，强大的查询功能等优点在经济方面就给我们提供了很大的便利。

# 三、进度安排：

2015年 11月15日-2016年1月24日  收集资料，书写并提交开题报告

2016年1月24日-2016年1月30日  完成网站整体风格设计

2016年2月1日-2016年2月15日  完成网站功能设计

2016年2月16日-2016年2月28日  进行数据库链接设计

2016年3月1日-2016年3月15日  完成网站实现、调试

2016年3月16日-2016年3月31日 完成论文初稿、修改，终稿

**北京理工大学继续教育暨现代远程教育学院**

**专科学生毕业实践（报告二）**

学生姓名：刘旋 指导教师：姜增如 学号：20140851400006号专业：计算机信息管理

# 摘 要

随着近年来网络媒体与网络购物平台的合作加深，消费者从获取资讯到发生购物行为的时空距离被进一步缩短，由此带来的巨大经济效益使得各大网购平台越来越重视网络产品销售。

论文首先阐述了系统开发的背景和设计目标，并介绍了系统开发的技术。在系统分析中，对系统模型进行了简单的分析，明确了系统的运行过程和系统的可行性分析，确定了目前条件下网上购物系统的开发是可行的。在需求分析的基础上，对系统的功能模块进行了划分。设计了系统的数据库，包括数据流图、实体关系图和数据表结构。在系统的设计与实现中，确定系统的发展和环境的发展，并对系统中的Java Bean和Servlet代码进行设计。最后，详细设计了系统中的模块设计和实现，并对系统中的关键源代码进行了详细的分析。经过调试和运行表明，该系统能够满足用户的要求，建立网上商城。

**关键词：**网上商城系统,java,J2EE,SSH2,MySql

**目 录**

[一、课题目的和意义 3](#_Toc448755359)

[二、本课题主要研究内容和预期目标 5](#_Toc448755360)

[三、进度安排： 8](#_Toc448755361)

[摘 要 9](#_Toc448755362)

[**目 录** 10](#_Toc448755363)

[第一章 系统概述 13](#_Toc448755364)

[第二章 需求分析 14](#_Toc448755365)

[第三章 开发平台及系统功能架构 15](#_Toc448755366)

[前台系统功能模块 15](#_Toc448755367)

[后台系统功能模块 15](#_Toc448755368)

[数据库设计 16](#_Toc448755369)

[数据分析 16](#_Toc448755370)

[数据库的E-R图 17](#_Toc448755371)

[数据字典 18](#_Toc448755372)

[数据库的建立与设计 22](#_Toc448755373)

[第四章 系统实现 25](#_Toc448755374)

[前台模块 25](#_Toc448755375)

[会员注册模块： 25](#_Toc448755376)

[商品查询模块 26](#_Toc448755377)

[订单查询模块 26](#_Toc448755378)

[购物车模块 26](#_Toc448755379)

[新品上市模块 27](#_Toc448755380)

[销售排行模块 27](#_Toc448755381)

[公告浏览模块 27](#_Toc448755382)

[商家管理员后台管理模块 28](#_Toc448755383)

[商品管理模块 28](#_Toc448755384)

[会员管理模块 28](#_Toc448755385)

[商家管理员模块 28](#_Toc448755386)

[订单管理模块 28](#_Toc448755387)

[财务管理模块 28](#_Toc448755388)

[公告管理模块 28](#_Toc448755389)

[留言模块 29](#_Toc448755390)

[代码实现（详情：附录） 29](#_Toc448755391)

[主要操作界面效果图 29](#_Toc448755392)

[功能测试 34](#_Toc448755393)

[单元测试 34](#_Toc448755394)

[前台首页的测试 35](#_Toc448755395)

[购物车模块的测试 35](#_Toc448755396)

[用户注册模块的测试 35](#_Toc448755397)

[商品管理模块的测试 36](#_Toc448755398)

[添加功能的测试 36](#_Toc448755399)

[删除功能的测试 36](#_Toc448755400)

[查询功能的测试 36](#_Toc448755401)

[集成测试 36](#_Toc448755402)

[确认测试 36](#_Toc448755403)

[系统测试 37](#_Toc448755404)

[维护 39](#_Toc448755405)

[第五章 总结 41](#_Toc448755406)

[第六章 自我鉴定 42](#_Toc448755407)

[参考文献 43](#_Toc448755408)

[附录 44](#_Toc448755409)

[购物车核心代码 44](#_Toc448755410)

[商品信息核心代码 53](#_Toc448755411)

[会员注册代码 60](#_Toc448755412)

[商品订单代码 61](#_Toc448755413)

# 第一章 系统概述

经过对各购物网站的考察、分析，要求本系统满足以下要求：

（1）统一友好的操作界面，保证系统的易用性，方便用户的操作；

（2）具备产品信息的展示功能，方便浏览者对产品信息进行浏览与比较；

（3）规范的产品分类，方便浏览者分类查找商品；

（4）完善的购物车功能与用户结账功能；

（5）设置网站公告信息模块，使浏览者能够及时了解网站的各项动态变化，及销售的

新品、特价活动等信息；

（6）商品销售排行功能；

（7）订单信息查询功能、商品信息查询功能；

（8）全面的后台管理功能，以方便管理员对网站信息进行更新与管理。

# 第二章 需求分析

总体设计是对软件总体的规划，规划各部分功能如何联系起来。软件是一个有机的整体，只有各部分协调运做才能使系统正常运行。用户登录管理系统之后，根据用户名密码来区分管理员和会员的身份区别和界面的不同。

商家管理员：登录系统后管理员对系统内的信息处理，商品更新，添加及更改等操作，从而满足顾客及商家的需求。

会员：普通顾客通过注册成为商城的会员，登录帐号后可以查看商品的具体信息，对商品进行购买，下订单，查询订单信息，对商城进行留言。

系统业务流程描述如下：

（1）顾客浏览网站 顾客通过浏览器登录到商品在线销售系统网站后，可以通过新品上架、特价产品、销售排行的模块查看商品及详细信息，同时顾客也可以按类别查看商品。另外，顾客可以浏览公告，销售排行榜等模块；

（2）用户注册及登录 顾客如要购买商品，需要进行用户注册，注册成功并登录后，用户可以购买商品；

（3）购物车处理 用户通过浏览数码产品信息选择要购买的产品时，可以将产品放入购物车，并可以添加至购物车、查看购物车、移除购物车中的数码产品、修改购物车中商品数量、清空购物车；

（4）收银台处理 当用户决定购买购物车中的商品时，可以通过收银台结账填写相应的订单信息，确定所填写的订单信息无误后，系统将产生此次订单的编号，完成在线下达订单，并且用户可以查看到订单；

（5）商家管理员后台处理 系统管理员登录后可以查看到已受理和未受理的订单及详细信息。另外，管理员能够对系统的各项信息进行维护处理。

# 第三章 开发平台及系统功能架构

本系统一共分为前台和后台两大模块，两个模块之间虽然在表面上是相互独立的，但是在对数据库的访问上是紧密相连的，各个模块访问的是同一个数据库，只是所访问的表不同而已。每个模块的功能都是按照在调研中搜集的资料进行编排制作的。依据上述功能分析，系统在两个模块的基础上每一个模块又分为几个模块。

## 前台系统功能模块

（1）会员注册模块：商城需要顾客通过注册成为本商城的会员，注册后才可以购买商品。

（2）商品查询模块：按商品的类别、名称查询商品的相关信息；

（3）购物车模块：用户添加商品至购物车，查看购物车中的商品，从购物车中移除不

满意的商品，清空购物车中的产品，修改所要购买的商品的数量；

（4）订单查询模块：用户通过查看订单能够了解到自己的当前订单信息及历史订单记

录；

（5）公告浏览模块：会员通过浏览公告信息，能够及时了解到网站最新的各种信息。

（6）系统留言模块：客户可以进行留言给我们提意见，我们在不断地改进中前进。

（7）新品上市功能，会员在商城主页上可以看到最新商品的信息和预告。

（8）销售排行功能，商城主页展示近期销售最好的几种产品。

请参考报告一中前台功能图

## 后台系统功能模块

（1）商品管理模块：按类别查看商品，对商品的信息进行维护；

（2）会员管理模块：为了保护用户的信息，此模块与前台用户维护的区别是管理员只能查看用户信息和删除操作；

（3）商家管理员模块：这是对管理员的信息进行维护，可以修改商家管理员的信息。

（4）订单管理模块：管理员查询订单，查看订单详细信息，删除订单信息，进行订单受理；

（5）财务管理模块：商家管理员受理订单并发货时进行收款和对阶段性收益的统计。

（6）公告管理模块：管理员公告浏览，公告信息维护；

（7）留言模块：管理员可以查看客户的留言，对留言进行维护。

系统后台管理员功能结构图如图2-4所示。

请参考报告一中后台功能图

## 数据库设计

对于一个电子商务网站而言，为了支持较大的访问量带来的数据访问需求，使用桌面型的数据库管理系统是不能满足需要的，而且安全性也没有充分保障。因此，需要使用大型商业化企业级服务用途的数据库管理系统，如SQL Server，Oracle等。本系统采用 mySQL 数据库管理系统。

### 数据分析

对于本系统的数据库的需求而言，由于其主要是用于信息的提供、保存、更新和查询等。因此，需要分析该系统功能所隐含的对数据应用的需求，从而确定数据库的结构。

（1）用户注册、登录、验证等功能需要对用户信息建立数据表，其中的数据项可能包括用户E\_mail、昵称、密码、住址等；

（2）查看商品分类信息和商品详细信息等功能既需要对商品大小类别进行管理，也需要针对每一个商品进行管理，因此至少需要两张数据表；

（3）用户购物后产生的订单需要进行管理，这里也包括订单的基本信息和详细信息等；

（4）订单生成后，在订单处理的过程中，需要出货等，因此可能需要记录订单的发送情况；

（5）需要系统管理员对该系统进行管理，因而需要建立管理员信息数据表，其中的数据项包括管理员ID、密码等。

（6）客户要进行留言，所以必须建立留言表，其中包括留言标题、留言日期等。

这样，至少要创建如下的数据结构和数据项

（1）用户信息，包括用户ID，用户名等数据项；

（2）管理员信息，包括管理员ID，密码等数据项；

（3）商品信息，包括商品ID，产品名称、单价、特价等数据项；

（4）商品类别信息，包括电子ID，类别名称等数据项；

（5）订单信息，包括订单ID，用户编号，订货地址等数据项；

（6）订单明细信息，包括订单ID，数码产品ID，订货时间等数据项；

（7）公告信息，包括公告ID，公告名称及内容等数据项；

（8）留言信息，包括留言标题，留言日期，留言内容等。

### 数据库的E-R图

在设计数据库时，应考虑以下事项

（1）数据库用途及该用途将如何影响设计，创建符合用途的数据库计划；

（2）数据库规范化规则，防止数据库设计中出现错误；

（3）对数据完整性的保护；

（4）数据库和用户权限的安全要求；

（5）应用程序的性能需求，设计数据库时必须利用 Microsoft mySQL 中能够提高性能的功能。对于性能而言，在数据库大小和硬件配置之间权衡也是很重要的；

（6）数据库维护。

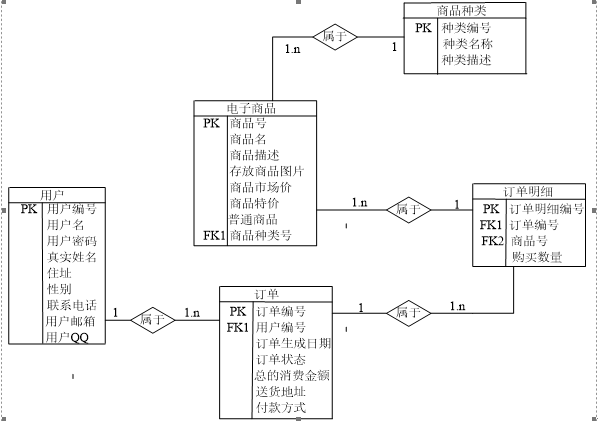


图2-5 数据库E-R图

### 数据字典

数据字典的作用是在软件分析和设计的过程中给人提供关于数据的描述信息。它主要是对数据流图中的数据流、处理逻辑、外部实体、数据存储和数据项等方面进行具体的定义。数据流程图配以数据字典，就可以从图形和文字两个方面对系统的逻辑模型进行完整的描述。

表2-1 系统数据项定义

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 别名 | 类型 | 长度（字符） |
| I01-01 | goods\_id | 商品号 | Int | 4 |
| I01-02 | goods\_name | 商品名 | varchar | 50 |
| I01-03 | goods\_miaoshu | 商品描述 | varchar | 5000 |
| I01-04 | goods\_pic | 存放商品图片 | Varchar | 50 |
| I01-05 | goods\_yanse | 商品颜色 | Varchar | 50 |
| I01-06 | goods\_shichangjia | 商品市场价 | int | 4 |
| I01-07 | goods\_tejia | 商品特价 | int | 4 |
| I01-08 | goods\_isnottejia | 普通商品 | varchar | 50 |
| I01-09 | goods\_catelog\_id | 商品种类号 | int | 4 |
| I01-10 | catelog\_id | 种类编号 | Int | 4 |
| I01-11 | catelog\_name | 种类名称 | Varchar | 50 |

表2-1 系统数据项定义（续表）

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | | 名称 | | 别名 | | 类型 | | 长度（字符） | |
| I01-12 | | catelog\_miaoshu | | 种类描述 | | Varchar | | 5000 | |
| I01-13 | | order\_id | | 订单ID | | Int | | 4 | |
| I01-14 | | order\_bianhao | | 订单编号 | | Varchar | | 50 | |
| I01-15 | | order\_date | | 订单生成日期 | | varchar | | 50 | |
| I01-16 | | order\_zhuangtai | | 订单状态 | | varchar | | 50 | |
| I01-17 | | order\_jine | | 总的消费金额 | | Int | | 4 | |
| I01-18 | | order\_songhuodizhi | | 送货地址 | | Varchar | | 50 | |
| I01-19 | | order\_fukuangfangshi | | 付款方式 | | Varchar | | 50 | |
| I01-20 | | order\_user\_id | | 用户ID | | Int | | 4 | |
| I01-21 | | orderItem\_id | | 订单明细编号 | | Int | | 4 | |
| I01-22 | | order\_id | | 订单编号 | | Int | | 4 | |
| I01-23 | | goods\_id | | 商品编号 | | Int | | 4 | |
| I01-24 | | goods\_quantity | | 购买数量 | | Int | | 4 | |
| I01-25 | | user\_id | | 用户ID | | Int | | 4 | |
| I01-26 | | user\_name | | 用户名 | | Varchar | | 50 | |
| I01-27 | | user\_pw | | 用户密码 | | Varchar | | 50 | |
| I01-28 | | user\_realname | | 真实姓名 | | Varchar | | 50 | |
| I01-29 | | user\_address | | 住址 | | Varchar | | 50 | |
| I01-30 | | user\_sex | | 性别 | | Varchar | | 50 | |
| I01-31 | | user\_tel | | 联系电话 | | Varchar | | 50 | |
| I01-33 | | user\_qq | | 用户QQ | | Varchar | | 50 | |
| I01-34 | | userId | | 管理员ID | | int | | 4 | |
| I01-35 | | userName | | 管理员名称 | | Varchar | | 50 | |
| I01-36 | | userPw | | 管理员密码 | | Varchar | | 50 | |
| I01-37 | | gonggao\_id | | 公告编号 | | Int | | 4 | |
| I01-38 | | gonggao\_title | | 公告标题 | | Varchar | | 50 | |
| I01-39 | | gonggao\_content | | 公告内容 | | Varchar | | 5000 | |
| I01-40 | | gonggao\_data | | 公告创建时间 | | Varchar | | 50 | |
| I01-41 | | gonggao\_fabuzhe | | 发布人 | | Varchar | | 50 | |
| I01-42 | | liuyan\_id | | 留言id | | Int | | 4 | |
| I01-43 | | liuyan\_title | | 留言标题 | | Varchar | | 50 | |
| I01-44 | | liuyan\_content | | 留言内容 | | Varchar | | 5000 | |
| I01-45 | | liuyan\_date | | 留言时间 | | Varchar | | 50 | |
| I01-46 | | liuyan\_user | | 留言客户 | | Varchar | | 50 | |

## 数据库的建立与设计

MySQL作为我们选用的数据库管理系统的后台数据库开发工具，其重要性显而易见。因此，作好后台数据库的设计工作，就为我的课题的实现打下了重要的基础。

下面我将介绍我所设计的表。

表3-1t\_goods（商品信息表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 别名 | 类型 | 长度（字符） |
| 1 | goods\_id | 编号 | Int | 4 |
| 2 | goods\_name | 商品名 | varchar | 50 |
| 3 | goods\_miaoshu | 商品描述 | varchar | 5000 |
| 4 | goods\_pic | 存放商品图片 | varchar | 50 |
| 5 | goods\_shangjia | 商品商家 | varchar | 50 |
| 6 | goods\_shichangjia | 商品市场价 | int | 4 |
| 7 | goods\_tejia | 商品特价 | int | 4 |
| 8 | goods\_isnottejia | 普通商品 | varchar | 50 |
| 9 | goods\_catelog\_id | 商品种类号 | int | 4 |

表3-2t\_order（订单明细表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 别名 | 类型 | 长度（字符） |
| 1 | order\_id | 订单ID | Int | 4 |
| 2 | order\_bianhao | 订单编号 | Varchar | 50 |
| 3 | order\_date | 订单生成日期 | varchar | 50 |
| 4 | order\_zhuangtai | 订单状态 | varchar | 50 |
| 5 | order\_jine | 总的消费金额 | Int | 4 |
| 6 | order\_songhuodizhi | 送货地址 | Varchar | 50 |
| 7 | order\_fukuangfangshi | 付款方式 | Varchar | 50 |
| 8 | order\_user\_id | 用户ID | Int | 4 |

表3-3t\_catelog（商品类别信息表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 别名 | 类型 | 长度（字符） |
| 1 | catelog\_id | 种类编号 | Int | 4 |
| 2 | catelog\_name | 种类名称 | Varchar | 50 |
| 3 | catelog\_miaoshu | 种类描述 | Varchar | 5000 |

表3-4t\_user（用户信息表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 别名 | 类型 | 长度（字符） |
| 1 | user\_id | 用户ID | Int | 4 |
| 2 | user\_name | 用户名 | Varchar | 50 |
| 3 | user\_pw | 用户密码 | Varchar | 50 |
| 4 | user\_realname | 真实姓名 | Varchar | 50 |
| 5 | user\_address | 住址 | Varchar | 50 |
| 6 | user\_sex | 性别 | Varchar | 50 |
| 7 | user\_tel | 联系电话 | Varchar | 50 |
| 8 | user\_email | 用户邮箱 | Varchar | 50 |
| 9 | user\_qq | 用户QQ | Varchar | 50 |

表3-5t\_gonggao（公告信息表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 别名 | 类型 | 长度（字符） |
| 1 | gonggao\_id | 公告编号 | Int | 4 |
| 2 | gonggao\_title | 公告标题 | Varchar | 50 |
| 3 | gonggao\_content | 公告内容 | Varchar | 5000 |
| 4 | gonggao\_data | 公告创建时间 | Varchar | 50 |
| 5 | gonggao\_fabuzhe | 发布人 | Varchar | 50 |

表3-6t\_liuyan（留言信息表）

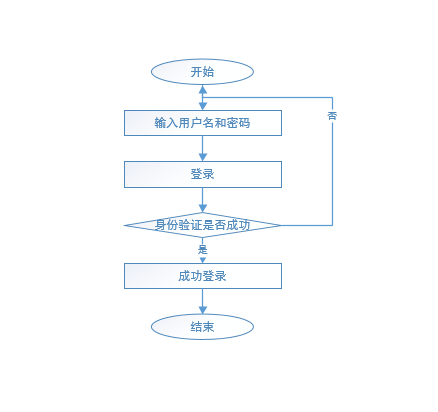
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 别名 | 类型 | 长度（字符） |
| 1 | liuyan\_id | 留言id | Int | 4 |
| 2 | liuyan\_title | 留言标题 | Varchar | 50 |
| 3 | liuyan\_content | 留言内容 | Varchar | 5000 |
| 4 | liuyan\_date | 留言时间 | Varchar | 50 |
| 5 | liuyan\_user | 留言客户 | Varchar | 50 |

表3-7t\_admin（商家管理员信息表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 别名 | 类型 | 长度（字符） |
| 1 | userId | 管理员ID | int | 4 |
| 2 | userName | 管理员名称 | Varchar | 50 |

# 第四章 系统实现

由于网上购物必须保证用户的正确性、购物的安全性，需要对用户资料进行有效的管理，因此用户管理是网站的重要内容。它主要包括用户注册、登录和验证、用户注册资料的修改更新和用户注销等功能。



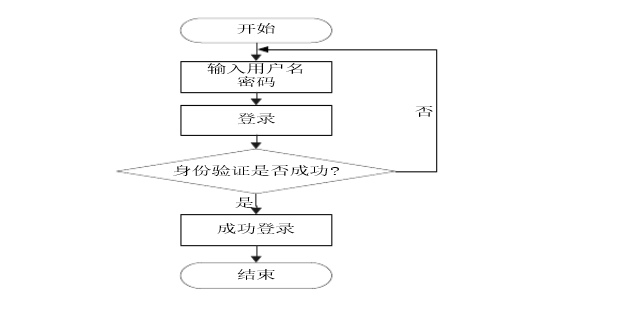


图4.1

## 前台模块

### 会员注册模块：

用户可以通过首页’会员注册‘按钮实现会员注册与登录。当用户输入用户名和密码后首先要进行身份验证，如果用户存在，则成功登录，反之，如果用户不存在，提示错误信息“输入的用户名或密码不存在，请注册”，返回登录界面重新输入用户名和密码。用户登录的程序流程图如图所示。

### 商品查询模块

会员登录商城后可以对自己想要购买的商品进行查询。在查询框中输入要查找的商品，点击查询按钮，会弹出所有相关的产品的界面，这样就可以更佳方便快捷的找到自己想要的商品。

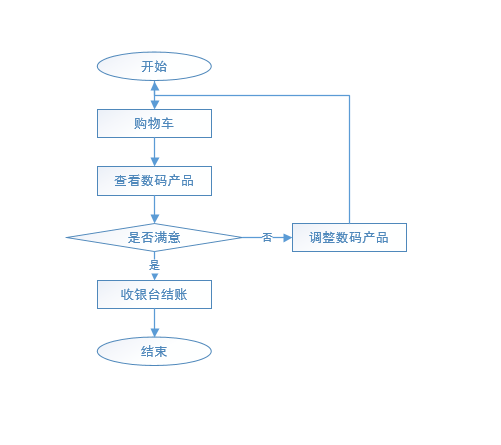
系统留言模块，会员可以对商城服务态度及商品质量的评论，可以向卖家提出自己的意见及看法。在留言板界面，点击留言，弹出文本框，在文本框中输入自己想要说的内容，点解确定，留言成功，留言时间为系统时间自动生成。同时还可以查看其他用户的留言内容。

### 订单查询模块

会员购买完商品并提交订单后可以对自己已下的订单进行查询。点击商城上边栏的‘我的订单’，弹出的界面会显示已经生成的订单，单击任意一条订单，可以查看每个订单的具体内容及订单处理情况。

### 购物车模块

会员选购商品后添加到购物车中，在购物车中进行查看，同时对不满意的产品可以进行移除，对要购买的产品可以调整数量，移除产品和清空产品。用户决定购买产品后可以去收银台进行结账。

图4.2

### 新品上市模块

会员在商城主页上可以看到最新商品的信息和预告。在新品区可以更好的了解商城新品动态，点击新品图片可以查看新品的价格、名称、型号等一切详细信息。

### 销售排行模块

商城主页展示近期销售最好的几种产品。排行区展示的是最近销售量最好的商品，点击商品图片可以看到商品的销售量。

### 公告浏览模块

商家在商城主页上对新品预告或商城活动供会员浏览。吸引顾客有很多种宣传措施，本系统采用公告的方式进行宣传。利用类似流动看板的方式显示公告信息。在顾客点击某一条公告信息时，系统可以自动弹出一个页面显示具体的公告信息。

## 商家管理员后台管理模块

### 商品管理模块

按类别查看商品，对商品的信息进行维护；在商品管理模块中除了基本的信息管理功能外，为了方便顾客浏览到商品的真实外貌，在添加商品模块中增加上传商品图片的功能。管理员在商品添加页面中输入添加的商品基本信息，点击提交后进入上传图片页面。在商品信息管理中，我将商品分为特价和非特价商品来管理。此页面中能够显示出已输入的特价商品基本信息，在上传图片位置上添加文件后点击提交，提交成功后有提示信息弹出。

### 会员管理模块

为了保护用户的信息，此模块与前台用户维护的区别是管理员只能查看用户信息和删除操作；

### 商家管理员模块

这是对管理员的信息进行维护，可以修改商家管理员的信息。商家管理员登录系统后，点击管理员信息修改可以对自己登录的用户名、密码进行修改，这样可以防止多次登录后密码泄露导致的安全问题。

### 订单管理模块

管理员查询订单，查看订单详细信息，删除订单信息，进行订单受理；商家管理员查看订单信息，根据顾客订单信息发货。如若没有对应的商品商家管理员可以将订单拒绝并删除该订单。

### 财务管理模块

商家管理员受理订单并发货时进行收款和对阶段性收益的统计。在该模块中在日历中选择几月几号到几月几号，在这阶段内收入的统计。

### 公告管理模块

吸引顾客有很多种宣传措施，本系统采用公告的方式进行宣传。利用类似流动看板的方式显示公告信息。在顾客点击某一条公告信息时，系统可以自动弹出一个页面显示具体的公告信息。

### 留言模块

管理员可以查看客户的留言，了解会员的需求及想法，管理员还可以对留言进行恢复和删除。

## 代码实现（详情：附录）

### 主要操作界面效果图

图4.1 系统首页



图4.2 注册界面



图4.3 登录界面



图4.4　用户修改模块



图4.5　销售排行模块



图4.6 购物车



图4.7 生成订单



图4.8　提交订单



图4.9　查看订单模块

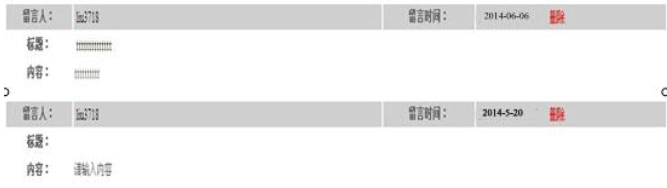


图4.10　公告浏览模块



图4.11 商品信息管理模块



图4.12 留言管理模块

## 功能测试

经过需求分析、设计和编码等阶段的开发后，得到了源程序，开始进入到软件测试阶段。然而，在测试之前的各阶段中都可能在软件产品中遗留下许多错误和缺陷，如果不及时找出这些错误和缺陷，并将其改正，这个软件产品就不能正常使用，甚至会导致巨大的损失。目前，程序的正确性证明尚未得到根本的解决，因此软件测试仍是发现软件中错误和缺陷的主要手段。

### 单元测试

单元测试是整体测试中的第一步，通常在编码阶段进行。在源程序代码经过编译、评审、确认没有语法错误之后，便可开始进行单元测试。

为了测试系统的正确性和完整性，本节将对系统中重点模块的测试进行介绍。

为了测试系统的正确性和完整性，本节将对系统中重点模块的测试进行介绍。

### 前台首页的测试

1．查询模块的测试

首先在主页中的查询产品中的下拉列表中单击某一类别，然后在文本框中输入要查询的商品关键字，在右边显示菜单中查询到商品，则说明查找成功。通过对系统的反复大量的测试，该模块测试成功。

2．商品分类浏览模块的测试

测试过程是单击页面左边的商品类别，选中某一类别后，就可以在页面右边显示该类别下的所有商品。通过对该功能的重复大量测试，该模块能够准确地实现此项功能。

### 购物车模块的测试

在本模块中首先对修改数量单元进行了测试，所修改的数量如果是有效数字，则修改所购数码产品数量成功，否则修改失败，系统自动将商品数量重置为“1”。通过随机测试了几组数据，其修改的数量和累计的金额均正确。如果客户购买了同一个商品，则就会进行该商品购买数量的累加，不会重复出现。然后针对购物车应该不限制顾客购买数码产品的数量这一原则进行测试，通过大量向购物车中添加数码产品来进行测试，测试结果为购物车中可以显示所添加的所有数码产品信息，测试结果取得了预期的效果。

### 用户注册模块的测试

在用户注册模块中涉及到了三项数据，项目分别为密码和确认密码、用户真实姓名、住址、性别、E\_mail、QQ。测试这个模块准备的数据有：

1．sang、123、123、—、哈尔滨道里区、女、bing@123.com、37472

2．sa、123、12、张三、哈尔滨道外区、男、bzhi@123.com、6434634

3．be、123、123、李四、哈尔滨香坊区、男、bli@123.com、53658403

首先进入注册页面，输入第一组数据，单击注册按钮系统会提示表单信息不能为空，因为表单信息要全部填写，而真实姓名没有写值。然后测试第二组数据，可以明显看出所输入的密码与确认密码是不一致的。然后测试第四组数据，其中九项均是系统正确的数据，因此测试通过。在填写用户名时，用户可以查看自己的用户名是否可以使用。

## 商品管理模块的测试

### 添加功能的测试

对数码产品添加模块准备的商品名、商品描述、商品图片、商品颜色、市场价、特价、商品种类，分别准备的测试数据为：摩托罗拉W220、翻盖手机，好用。1680、1500。其中类别的选择是通过下拉列表框来选择，颜色也是通过下拉列表来选择，图片单击上传图片后，找到图片的地址点击确定就可以上传，单击添加按钮，执行添加操作，数据提交后，就完成了商品的添加。之后，进行了几项测试，如漏填了几项数据、上传图片格式不正确，这些数据的操作结果都是失败。

### 删除功能的测试

以编号为4的这条记录为例，通过单击该条记录上的“删除”链接按钮，对数码产品信息进行删除的操作，数据提交后，系统提示“删除商品信息成功”，通过对该模块的浏览，该信息的确已被删除，查无此条记录，说明此模块运行无误，信息删除成功。如果是删除商品类别，假如该类别下还有商品，则会提示不能删除，如果类别下没有商品，则可以进行删除。

### 查询功能的测试

本系统中的查询功能都是通过点击链接实现的。将数码产品管理的编码进行了测试，输入管理员的密码登录后台后点击数码产品设置就可以直接看到查询出的数码产品信息。点击查看产品图片后查询出了数码产品是否上传了图片，系统没有出现错误，查询成功。

### 集成测试

集成测试在单元测试完成之后，将所有的模块概要设计要求组装成系统时进行的测试，主要目标是发现与接口有关的问题。

### 确认测试

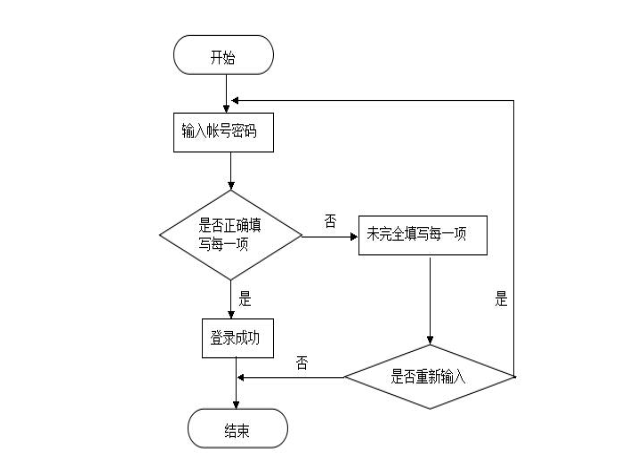
经过集成测试之后，接口错误已经发现被发现并改正了，接下便要进行确认测试。所谓确认测试就是验证所开发软件的功能性及其他特性是否符合软件需求规格说明书的要求。所以，确认测试又被称为有效性测试。

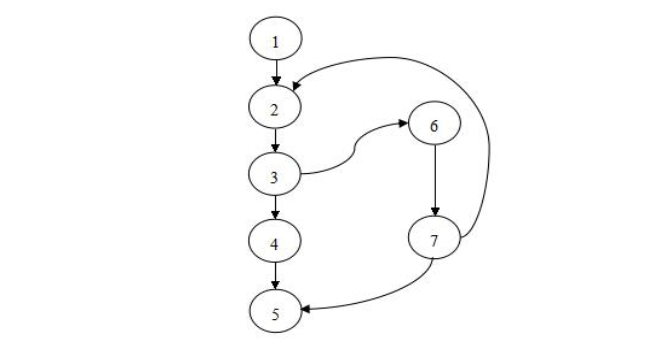
### 系统测试

系统测试是更大范围内进行测试，它将经过确认测试的软件作为整个基于计算机的系统的一个元素，在实际运行环境下，对系统进行的一系列集成和确认测试。

下面采用白盒测试中的基本路径测试方法，来测试此系统。

根据详细设计或者代码绘制流程图和程序流图如图4-1和图4-2所示。

图4-1 登录界面流程图



根据流图确定环形复杂性度量V(G)。

V(G)=E-N+2=3

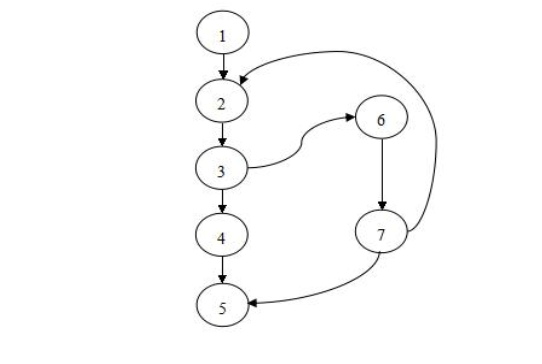


图4-2 登陆界面程序流程图

V(G)=3个区域

V(G)=P+1=2+1=3

确定基本路径集合 独立的路径是指，和其他的独立路径相比，至少引入一个新处理语句或一个新判断的程序通路。在（2）中求出的V(G)值正好等于该程序的独立路径的条数。于是可以确定几条独立的路径。

路径1：1-2-3-4-5

路径2：1-2-3-6-7-5

路径3：1-2-3-6-7-2

为每一条独立的路径各设计一组测试用例，以便强迫程序沿着该路径至少执行一次。

（1） 路径1的测试用例。

输入：输入帐号和密码及验证码。

期望结果：登录系统成功。

（2） 路径2的测试用例。

输入：未正确输入每一项内容且不重新输入。

期望结果：退出登录系统操作。

（3） 路径3的测试用例。

输入：未正确输入每一项内容但选择重新输入。

期望结果：系统返回到输入用户名和密码的登录界面

### 维护

网站建成后并不意味着网站建设的结束，网站内容需要不断更新，这样才会不断的吸引更多的浏览者访问，才能成为一个优秀的网站。通过FTP软件将更新过的网页上传到服务器或通过Dream weaver MX来远程管理站点，同时还要认真回复浏览者的电子邮件和留言，做到有问比答。平时多收集资料，多听听别人建议，定期更新版面，还有推广站点。

# 第五章 总结

经过不懈奋斗和努力，终于完成了这篇并不成熟的论文，自己对此并不满意，由于能力有限对论文的水平有些惶恐。虽然在论文编写的过程中查阅了大量的中英文资料，但由于计算机相关技术和理论的快速发展加之本人所学有限使得本论文的有些理论论述和技术仍显得滞后。

本系统采用当前流行的Myeclise平台结合J2EE框架进行开发，前台采用JSP动态网页，结合SSH框架技术来完成整个系统的设计，在数据库的设计上使用MySQL数据库系统。

本次设计完成了系统的大部分功能：包括会员的注册登录、购买商品、订单查询、修改个人密码、留言、商品分类会员管理、购物车、收银、商品展示、排行榜、特价商品等一系列功能，但由于个人实力有限，使系统没有实现第三方实现，主要表现在网上付款功能。

# 第六章 自我鉴定

由于本人在知识、经验方面都存在着不足，在整个开发的过程中时间也比较仓促，因此该系统必然会存在一些缺陷和不足。这些缺陷将在本系统的后续升级扩充的工作中逐步实现和完善。

# 参考文献

[1] 汪建文．我国零售企业的商务电子化应用商场现代化[M]. 信息技术与标准化, 2008:56-60.

[2] 胡燕．软件工程与标准化[M]．信息技术与标准化, 2007:17-21.

[3] 盛仲飙．JSP中数据库访问的性能优化研究[M]．微型电脑应用,2008:16-17.

[4] 董纪阳．软件设计中的需求分析[J]．电脑知识与技术,2008:56-57.

[5] 林华灵．基于Strut框架技术的Web应用[J]．中国新技术新产品,2008，（14）:9-11.

[6] 何玉洁．数据库原理与应用教程[M]．机械工业出版社,2003:23-78.

[7] 杨文红，赵德平，刘柯剑．基于MVC模式Struts框架研究与应用[M]．硅谷,2008:45-46.

[8] 赵蕾，程明定，刘琳．J2EE模型中Struts框架技术的研究和应用[M]．中国博览,2004:25-26.

[9] 桃子．网络产品市场前瞻[J]．微型计算机,2007:89-90.

[10] 周建儒．JSP中自定义标签的实现与部署[M]．科技信息（科学教研）,2008:51-53.

# 附录

### 购物车核心代码

在userAction中：

public String userLogin()

{

String sql="from TUser where userName=? and userPw=?";

Object[] con={userName,userPw};

List userList=userDAO.getHibernateTemplate().find(sql,con);

if(userList.size()==0)

{

this.setMessage("用户名或密码错误");

this.setPath("qiantai/index.html");

}

else

{

Map session= ServletActionContext.getContext().getSession();

TUser user=(TUser)userList.get(0);

session.put("user", user);

Cart cart=new Cart(); //用户登陆成功后生成购物车，并将其放入到session中。

session.put("cart", cart);

this.setMessage("成功登录");

this.setPath("qiantai/index.html");

}

return "succeed";

}

在buyAction中：

public String addToCart()

{

TGoods goods=goodsDAO.findById(goodsId);

TOrderItem orderItem=new TOrderItem();

orderItem.setGoods(goods);

orderItem.setGoodsQuantity(quantity);

Map session= ServletActionContext.getContext().getSession();

Cart cart = (Cart)session.get("cart"); //将session中的cart取出，也就是上边存储在session中的cart

cart.addGoods(goodsId, orderItem);

session.put("cart",cart); //再把cart放入session

//this.setMessage("");

this.setPath("myCart.action");

return "succeed";

}

public String myCart() //获取我的购物车

{

return ActionSupport.SUCCESS;

}

public String orderQueren() //确认提交订单

{

Map request=(Map)ServletActionContext.getContext().get("request");

return ActionSupport.SUCCESS;

}

public String orderSubmit() //订单提交

{

Map session= ServletActionContext.getContext().getSession();

Cart cart = (Cart)session.get("cart"); //将cart取出

TUser user=(TUser)session.get("user"); //将用户取出

TOrder order=new TOrder();

order.setOrderBianhao(new SimpleDateFormat("yyyyMMddhhmmss").format(new Date())); //设定订单编号

order.setOrderDate(new

SimpleDateFormat("yyyy-MM-dd hh:mm:ss").format(new Date())); //确认订单的日期

order.setOrderZhuangtai("no"); //订单状态默认为未受理

order.setOrderUserId(user.getUserId()); //设定订单的用户ID

order.setOrderJine(cart.getTotalPrice()); //商品的价格

orderDAO.save(order); //保存订单

for (Iterator it = cart.getItems().values().iterator(); it.hasNext();)

//把订单里的商品存入数据库

{

TOrderItem orderItem = (TOrderItem) it.next();

orderItem.setOrderId(order.getOrderId());

orderItem.setGoodsId(orderItem.getGoods().getGoodsId());

orderItemDAO.save(orderItem);

}

cart.getItems().clear(); //清空购物车

session.put("cart", cart); Maprequest=(Map)ServletActionContext.getContext().get("request");

request.put("order", order);

return ActionSupport.SUCCESS;

}

publicStringmyOrder() //根据用户查询订单

{

Map session= ServletActionContext.getContext().getSession();

TUser user=(TUser)session.get("user"); //获取用户

String sql="from TOrder where orderUserId="+user.getUserId();

List orderList=orderDAO.getHibernateTemplate().find(sql); //调用orderDAO中的find方法，执行sql语句

Map request=(Map)ServletActionContext.getContext().get("request");

request.put("orderList", orderList);

return ActionSupport.SUCCESS;

}

public String orderMana() //订单管理

{

String sql="from TOrder order by orderUserId";

List orderList=orderDAO.getHibernateTemplate().find(sql);

Map request=(Map)ServletActionContext.getContext().get("request");

request.put("orderList", orderList);

return ActionSupport.SUCCESS;

}

publicStringorderDel() //用户自己删除订单

{

TOrder order=orderDAO.findById(orderId);

orderDAO.delete(order);

Map session= ServletActionContext.getContext().getSession();

TUser user=(TUser)session.get("user");

this.setMessage("删除成功");

this.setPath("myOrder.action?userId="+user.getUserId()); //跳转到 myOrder.action

return "succeed";

}

publicStringorderDelByAd() //管理员删除订单

{

TOrder order=orderDAO.findById(orderId);

orderDAO.delete(order);

String sql="delete from TOrderItem where orderId="+orderId;

orderItemDAO.getHibernateTemplate().bulkUpdate(sql);

this.setMessage("删除成功");

this.setPath("orderMana.action");

return "succeed";

}

public String orderShouli() //对订单进行受理

{

TOrder order=orderDAO.findById(orderId);

order.setOrderZhuangtai("yes");//已经受理订单

orderDAO.attachDirty(order);

this.setMessage("受理订单成功");

this.setPath("orderMana.action");

return "succeed";

}

public String orderDetail() //订单明细

{

String sql="from TOrderItem where orderId="+orderId;

List orderItemList=orderItemDAO.getHibernateTemplate().find(sql);

for(int i=0;i<orderItemList.size();i++)

{

TOrderItem orderItem=(TOrderItem)orderItemList.get(i);

orderItem.setGoods(goodsDAO.findById(orderItem.getGoodsId()));

} //取出订单里的商品

Map request=(Map)ServletActionContext.getContext().get("request");

request.put("orderItemList", orderItemList);

return ActionSupport.SUCCESS;

}

在cart.java中

package com.util;

import java.util.HashMap;

import java.util.Iterator;

import java.util.Map;

import com.model.TGoods;

import com.model.TOrderItem;

public class Cart //定义购物车类

{

protected Map<Integer, TOrderItem> items;

public Cart() //定义构造方法

{

if (items == null)

{

items = new HashMap<Integer, TOrderItem>();

}

}

public void addGoods(Integer goodsId, TOrderItem orderItem) //添加购物车

{

if (items.containsKey(goodsId))

{

TOrderItem \_orderitem = items.get(goodsId);

\_orderitem.setGoodsQuantity(\_orderitem.getGoodsQuantity()+ orderItem.getGoodsQuantity()); //如果购买商品重复，则加上数量

items.put(goodsId, \_orderitem);

} else

{

items.put(goodsId, orderItem);

}

}

PublicvoiddelGoods(IntegergoodsId) //删除购物车

{

items.remove(goodsId);

}

public void updateCart(Integer goodsId, int quantity) //更新购物车的数量

{

TOrderItem orderItem = items.get(goodsId);

orderItem.setGoodsQuantity(quantity);

items.put(goodsId, orderItem);

}

public int getTotalPrice() //计算总的价格

{

int totalPrice = 0;

for (Iterator it = items.values().iterator(); it.hasNext();)

{

TOrderItem orderItem = (TOrderItem) it.next();

TGoods goods = orderItem.getGoods();

int quantity = orderItem.getGoodsQuantity();

totalPrice += goods.getGoodsTejia() \* quantity; //单价乘以数量

}

return totalPrice;

}

}

### 商品信息核心代码

<%@ page language="java" pageEncoding="UTF-8"%>

<%@ taglib prefix="s" uri="/struts-tags"%>

<%

String path = request.getContextPath();

%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="pragma" content="no-cache" />

<meta http-equiv="cache-control" content="no-cache" />

<meta http-equiv="expires" content="0" />

<metahttp-equiv="keywords" content="keyword1,keyword2,keyword3" />

<meta http-equiv="description" content="This is my page" />

<link rel="stylesheet" type="text/css" href="<%=path %>/css/base.css" />

<script language="JavaScript" src="<%=path %>/js/public.js" type="text/javascript"></script>

<script type="text/javascript" src="<%=path %>/js/popup.js"></script>

<script language="javascript">

function goodsDetailHou(goodsId)

{

var url="<%=path %>/goodsDetailHou.action?goodsId="+goodsId;

var n="";

var w="480px";

var h="500px";

var s="resizable:no;help:no;status:no;scroll:yes";

openWin(url,n,w,h,s);

}

function goodsNoTejiaDel(goodsId)

{

if(confirm('您确定删除吗？'))

{

window.location.href="<%=path %>/goodsNoTejiaDel.action?goodsId="+goodsId;

}

}

//修改商品

function goodsNoTejiaUpd(goodsId)

{

if(confirm('您确定修改吗？'))

{

window.location.href="<%=path %>/goodsDetail4Upd.action?goodsId="+goodsId;

}

}

function goodsNoTejiaAdd()

{

var url="<%=path %>/admin/goods/goodsNoTejiaAdd.jsp";

//var n="";

//var w="480px";

//var h="500px";

//var s="resizable:no;help:no;status:no;scroll:yes";

//openWin(url,n,w,h,s);

window.location.href=url;

}

function goodsShezhiTejia(goodsId)

{

varpop=new Popup({ contentType:1,isReloadOnClose:false,width:400,height:200});

pop.setContent("contentUrl","<%=path %>/admin/goods/goodsShezhiTejia.jsp?goodsId="+goodsId);

pop.setContent("title","文件上传");

pop.build();

pop.show();

}

function over(picPath)

{

if (picPath=="")picPath="/images/default.jpg";

x = event.clientX;

y = event.clientY;

document.all.tip.style.display = "block";

document.all.tip.style.top = y;

document.all.tip.style.left = x+10;

document.all.photo.src = ".."+picPath;

}

function out()

{

document.all.tip.style.display = "none";

}

</script>

</head>

<body leftmargin="2" topmargin="2" background='<%=path %>/images/allbg.gif'>

<table width="98%" border="0" cellpadding="2" cellspacing="1" bgcolor="#D1DDAA" align="center" style="margin-top:8px">

<tr bgcolor="#E7E7E7">

<td height="14" colspan="9" background="<%=path %>/images/tbg.gif">&nbsp;商品管理&nbsp;</td>

</tr>

<tr align="center" bgcolor="#FAFAF1" height="22">

<td width="10%">ID</td>

<td width="10%">商品名称</td>

<td width="10%">商品描述</td>

<td width="10%">分类</td>

<td width="10%">商品图片</td>

<td width="10%">市场价格</td>

<td width="10%">特价</td>

<td width="10%">库存</td>

<td width="10%">操作</td>

</tr>

<s:iterator value="#request.goodsList" id="goods">

<tr align='center' bgcolor="#FFFFFF" onMouseMove="javascript:this.bgColor='red';" onMouseOut="javascript:this.bgColor='#FFFFFF';" height="22">

<td bgcolor="#FFFFFF" align="center">

<s:property value="#goods.goodsId"/>

</td>

<td bgcolor="#FFFFFF" align="center">

<s:property value="#goods.goodsName"/>

</td>

<td bgcolor="#FFFFFF" align="center">

<a href="#" onclick="goodsDetailHou(<s:property value="#goods.goodsId"/>)" class="pn-loperator">商品描述</a>

</td>

<td bgcolor="#FFFFFF" align="center">

<s:property value="#goods.goodsCatelogName"/>

</td>

<td bgcolor="#FFFFFF" align="center">

<div onmouseover = "over('<%=path %>/<s:property value="#goods.goodsPic"/>')" onmouseout = "out()" style="cursor:hand;">

查看图片

</div>

</td>

<td bgcolor="#FFFFFF" align="center">

<s:property value="#goods.goodsShichangjia"/>

</td>

<td bgcolor="#FFFFFF" align="center">

<s:if test="#goods.goodsIsnottejia=='no'">无特价&nbsp;&nbsp;&nbsp;

<a href="#" style="color: red" onclick="goodsShezhiTejia(<s:property value="#goods.goodsId"/>)">设为特价</a>

</s:if>

<s:if test="#goods.goodsIsnottejia=='yes'">

<s:property value="#goods.goodsTejia"/>

</s:if>

</td>

<td bgcolor="#FFFFFF" align="center">

<s:property value="#goods.goodsKucun"/>

</td>

<td bgcolor="#FFFFFF" align="center">

<a href="#" onclick="goodsNoTejiaUpd(<s:property value="#goods.goodsId"/>)" class="pn-loperator">修改</a> &nbsp;

<a href="#" onclick="goodsNoTejiaDel(<s:property value="#goods.goodsId"/>)" class="pn-loperator">删除</a>

</td>

</tr>

</s:iterator>

</table>

<table width='98%' border='0'style="margin-top:8px;margin-left: 5px;">

<tr>

<td>

<input type="button" value="添加" style="width: 80px;" onclick="goodsNoTejiaAdd()" />

</td>

</tr>

</table>

<div id="tip" style="position:absolute;display:none;border:0px;width:80px; height:80px;">

<TABLE id="tipTable" border="0" bgcolor="#ffffee">

<TR align="center">

<TD><img id="photo" src="" height="80" width="80"></TD>

</TR>

</TABLE>

</div>

</body>

</html>

### 会员注册代码

顾客在商城首页通过注册成为会员。

public class TAdminDAO extends HibernateDaoSupport {

private static final Log log = LogFactory.getLog(TAdminDAO.class);

// property constants

public static final String USER\_NAME = "userName";

public static final String USER\_PW = "userPw";

protected void initDao() {

// do nothing

}

public void save(TAdmin transientInstance) {

log.debug("saving TAdmin instance");

try {

getHibernateTemplate().save(transientInstance);

log.debug("save successful");

} catch (RuntimeException re) {

log.error("save failed", re);

throw re;

}

}

### 商品订单代码

会员购买商品，进行下单处理代码。

public class TCatelogDAO extends HibernateDaoSupport {

private static final Log log = LogFactory.getLog(TCatelogDAO.class);

// property constants

public static final String CATELOG\_NAME = "catelogName";

public static final String CATELOG\_MIAOSHU = "catelogMiaoshu";

protected void initDao() {

// do nothing

}

public void save(TCatelog transientInstance) {

log.debug("saving TCatelog instance");

try {

getHibernateTemplate().save(transientInstance);

log.debug("save successful");

} catch (RuntimeException re) {

log.error("save failed", re);

throw re;

}

}

public void delete(TCatelog persistentInstance) {

log.debug("deleting TCatelog instance");

try {

getHibernateTemplate().delete(persistentInstance);

log.debug("delete successful");

} catch (RuntimeException re) {

log.error("delete failed", re);

throw re;

}

}