**目 录**

[摘 要 1](#_Toc512282620)

[1.绪论 3](#_Toc512282621)

[1.1开发背景 3](#_Toc512282622)

[1.2开发目的和意义 3](#_Toc512282623)

[1.2.1开发目的 3](#_Toc512282624)

[1.2.2开发意义 3](#_Toc512282625)

[1.3开发思路和主要内容 4](#_Toc512282626)

[1.3.1会员管理： 4](#_Toc512282627)

[1.3.2商品管理： 4](#_Toc512282628)

[1.3.3商品交易： 4](#_Toc512282629)

[1.3.4其他需求： 5](#_Toc512282630)

[1.3.5使用到的整体技术 5](#_Toc512282631)

[2 系统开发运行环境介绍 5](#_Toc512282632)

[2.1运行环境介绍 5](#_Toc512282633)

[2.2系统数据库介绍 6](#_Toc512282634)

[2.2.1Mysql简介 6](#_Toc512282635)

[2.2.2MySQL语言简介 6](#_Toc512282636)

[2.3开发语言与开发工具介绍 7](#_Toc512282637)

[3.系统分析 9](#_Toc512282638)

[3.1系统需求分析 9](#_Toc512282639)

[3.1.1系统需求概述 9](#_Toc512282640)

[3.1.2数据库需求分析 10](#_Toc512282641)

[3.2系统功能分析 11](#_Toc512282642)

[3.2.1用户功能分析 11](#_Toc512282643)

[3.2.2管理员功能分析 11](#_Toc512282644)

[3.3可行性分析 11](#_Toc512282645)

[3.4系统安全性分析 12](#_Toc512282646)

[3.5主要业务逻辑分析 12](#_Toc512282647)

[4系统设计 15](#_Toc512282648)

[4.1数据库设计 15](#_Toc512282649)

[4.1.1 概念结构设计 15](#_Toc512282650)

[4.1.2 逻辑结构设计 18](#_Toc512282651)

[4.1.3 物理结构设计 19](#_Toc512282652)

[4.2网站功能结构 22](#_Toc512282653)

[4.2.1前台功能结构 22](#_Toc512282654)

[4.2.2后台功能结构 23](#_Toc512282655)

[4.3网站页面结构 24](#_Toc512282656)

[4.3.1前台页面结构 24](#_Toc512282657)

[4.3.2后台页面结构 24](#_Toc512282658)

[5.系统实现 25](#_Toc512282659)

[5.1前台主要模块的实现 25](#_Toc512282660)

[5.1.1 产品展示模块 25](#_Toc512282661)

[5.1.2购物车模块 26](#_Toc512282662)

[5.1.3搜索模块 28](#_Toc512282663)

[5.2后台主要模块的实现 32](#_Toc512282664)

[5.2.1网站配置模块 32](#_Toc512282665)

[5.2.2产品管理模块 32](#_Toc512282666)

[5.2.3订单管理模块 33](#_Toc512282667)

[5.2.4用户管理模块 33](#_Toc512282668)

[6系统调试与测试 33](#_Toc512282669)

[6.1系统调试 33](#_Toc512282670)

[6.2单个页面调试 33](#_Toc512282671)

[6.3 多个页面调试 34](#_Toc512282672)

[6.4软件测试 34](#_Toc512282673)

[总结 35](#_Toc512282674)

[参考文献 36](#_Toc512282675)

[致谢 37](#_Toc512282676)

# 摘 要

网上数码商城系统是一个根据网上购物现状而开发的基于互联网，采用了B2C模式的中小型电子商务平台应用。它延长和扩大了商家的经营范围，更好的为用户提供快捷高效的购物，也能让商家做买卖的时候更加快效的达到经营的成本关系。在开发背景中介绍目前购物的现状，随后阐述了系统开发目标和意义。介绍了开发运行环境、系统分析，以及如何实现系统设计。最后，进行了相应的测试和调试不走并提出需要改进的方面。本系统为客户提供了会员注册三方登录注册，订单管理，用户信息管理，评论等管理功能。本系统采用B/S两层结构，使用Springboot搭建项目，mysql作为数据库，该数码商城具有前台商品显示，以及后台订单、用户信息等管理功能。

**关键词**：数码商城；购物车；SpringBoot；B/S结构

**Abstract**

The online digital mall system is a small and medium-sized e-commerce platform based on the Internet, which is based on the online shopping status and adopted the B2C mode. It extends and expands the business scope of the business to provide users with faster and efficient shopping, and can also allow businesses to do business more quickly to achieve the cost of business relations. The current situation of shopping is introduced in the development background, and then the goal and significance of system development are expounded. The development and operation environment, system analysis and how to design the system are introduced. Finally, the corresponding tests and debugging are not carried out, and some improvements are suggested. The system provides management functions such as membership registration, three party registration, order management, user information management, reviews and other management functions. This system uses the B/S two layer structure, uses Springboot to build the project, MySQL is the database, the website design is a front desk display, and the backstage order, user information and so on.

**Key words**: digital mall; shopping cart; SpringBoot; B/S structure.

# 1.绪论

## 1.1开发背景

电子商务就是把电子和电子技术当做一种工具[1]，把商务作为核心，打破原来传统的销售、购物渠道转移到了我们熟悉的互联网上，使生产的企业达到了全球化、网络化、一体化等特点。中国的电子商务发展迅猛，2007年全国电子商务的交易总额达到了2.17万亿元，同上一年同比增长90%。统计至2008年12月，电子商务类网站上的总体用户从9000万人提升至了9800万人。

随着电子商务的发展趋势，购物的发展实在必行，网上购物，就是通过互联网查找商品信息，并通过传输网络信息发起购物的请求，添上储蓄卡或者相应的支付宝账号，厂商通过邮购的方式来进行发货，或者通知快递公司上门取货送货等。

## 1.2开发目的和意义

### 1.2.1开发目的

创建一个中小型的网上购物系统，方便不同用户购买数码产品。用户可以通过Internet登录数码商城，浏览所有数码产品信息，使用搜索功能快捷方便找到感兴趣的产品。将自己感兴趣的数码产品加入购物车，如果决定购买可以直接下单，支付等信息。方便用户足不出户就能买到自己称心如意的数码产品。

管理员登录后台可以对商品信息和商品分类进行增加、删除和修改等；对于广告这类表示本网页不产生任何多余的广告，管理员注册用户信息；管理商品订单信息等功能。

### 1.2.2开发意义

过网上数码商城系统，顾客可以足不出户方便快捷地够到信用好质量高的数码产品，我们还可以添加热搜的功能让那些不知道从何下手选择什么样的数码产品还在迷茫的人给一个良好的建议。用户可以通过与商家沟通那些数码产品比较经济实惠，借助QQ微信联系进行对商家。通过会员注册信息以及购买记录可以很好了解顾客的需求，实现个性化服务。

## 1.3开发思路和主要内容

### 1.3.1会员管理：

用户直接点击网站上QQ第三方登录的按钮进行注册，如果QQ注册过就进行三方快捷的登录操作。如果没有注册过就填入对应的手机号密码验证码来进行注册操作做。注册里面包含了普通注册和三方登录注册，为了体现三方登录快捷性。

管理员能对用户进行删除和修改信息等操作。

### 1.3.2商品管理：

⑴在网站的页面上提供了大量的商品浏览查看目录，在每一个大类下又分了很多小类，在单个商品信息查看页面提供了商品基本描述和价格介绍。

⑵对于购买过的用户添加一个对商品的评价功能，这样真实有效的数据可以反应出一个商品的好坏。让用户选择起来更加的方便。

⑶为了方便用户购物，提供了购物车功能。在每个商品信息浏览中都有购物车图标，会员可以点击图片将该商品加入购物车，继而选择继续购物或是付款等操作，如果继续购物，那么可进行相同操作，购物车会自动对用户购买商品的数量和总额进行计算。

⑷网站还提供了关键字商品搜索功能；搜索的功能分为简单搜索和高级搜索分类搜索，分类搜索在主页面已经可以直接选择类型，头部有一个搜索使用模糊查询来搜索到自己心仪的产品。如果搜索出来的商品比较多，那么所有商品条目讲分页显示。

⑸用户购买商品时，网站对购物车里的商品会自动生成订单。生成订单后，根据付款方式的不同而进行不同的处理方法，采用了第三方支付的功能。

### 1.3.3商品交易：

⑴在顾客选择好商品后，就可以选择付款以及收获方式，提交商品订单，经过后台管理员得确认后就可以对订单进行处理了。

⑵已注册的顾客可以随时查看订单状态，通过首页的我的订单按钮，对已购买的订单进行查询和处理。其中，可以通过卖家已发货、买家已付款、卖家已收款和买家已收货的选项进行订单的查询。顾客可以在订单没有完成之前随时修改订单或者取消。另外，如果顾客没有注册成为我们的会员，那么该顾客是没有资格查看订单的。

### 1.3.4其他需求：

⑴需要一个信息中心。在任何一个过程中，都可以找到信息中心的链接，以方便顾客在使用网站时解决问题和困难。在商城首页，最好能在显眼的地方放上网站帮助之类的信息链接（这些信息包括了操作说明、付款说明、网站信息等）。

⑵不管是几层页面，都有一个导航可以链接到首页，用户可以直接点击按钮回到首页；在每个页面上都有商品分类页面，方便顾客进行浏览。

⑶在商城首页最好提供特价商品、商品排行、最新产品以及热销产品等商品浏览页面链接，这样可以方便顾客对商品进行了解和选择。

### 1.3.5使用到的整体技术

整个JavaEE采用了Springboot[2]来搭建整个项目，用Mysql[3]来作为数据库以及数据的收集，Maven来管理整个项目[4]，RestFul风格的编码来进行前后台数据之间的交互。权限控制采用2个表来简单的进行区分。简便的登录方式可以选择合适的三方进行登录/注册。用户在遇到忘记密码的时候可以采用短信验证的方式来找回忘记密码。采用Thyemeleaf来进行前端也数据渲染取值的问题，因为Thyemeleaf操作简便适用范围广。Spring Data Jpa[5]来进行数据库的CURD操作使用它来进行数据库[6]操作可以极大简化代码的工作量。前端页面我采用Sublime来进行相应的初步开发，使用到集合的框架来进行页面的美化，还会使用到H5结合Jquery[7]来进行输入框的有效性验证。

# 2 系统开发运行环境介绍

## 2.1运行环境介绍

## 2.2系统数据库介绍

### 2.2.1Mysql简介

（1）历史介绍

MySQL原本是一个开放源代码的关系型数据库关系系统）[8]，后因该公司在2008年被升阳系收购。2009年，后来甲骨文公司又受够了太阳微系统公司，MySQL就称为了Oracle旗下的产品了。

（2）发展情况

由于MySQL是开放源代码的数据库之一，任何人都可以下载后根据自己个性化的需求对其进行相应的修改。MySQL因为的性能高、成本低、可靠性良好，所以成为了最流行的开源数据库。MySQL在对事物处理的情况支持性上能够满足我们正常的需求，中小企业的人数远远大于大型企业的人数，所在我们在数据库选择的时候都会选择MySQL社区版来进行相应的开发，中小型企业可以选择MySQL商业版进行使用，从而大大提高了MySQL的使用量远远大于了其他数据库。

（3）选择

因此我们的这个数码商城采用了MySQL社区版作为开发的主要数据库。

### 2.2.2MySQL语言简介

（1）服务启动与停止命令

Net start mysql（启动服务）

Net stop mysql （终止服务）

Mysql –u用户名 –p密码 进行数据库连接

Exit、quit（数据库退出）

（2）常用命令介绍[8]

修改数据库密码：1：登录数据库 2：use mysql 3：update user set password=“NewPassword” where user=“用户名” 4：flush privieleges；更新权限完成密码修改

SHOW DATABASES： 找出服务器当前存在什么数据库

CREATE DATABASE Name：创建一个数据库Name

USE DATABASENAME 选择当前呢操作的数据库

SHOW TABLES：查看当前数据库里面存在什么表

CREATE TABLE TABLENAME(字段：限制类型) 创建一个数据库表

DESCRIBE TABLENAME 显示表的结构

（3）DML：

INSERT INTO TABLENAME VALUES (“字段”,”数值”) TABLE 中插入一条数据

UPDATE TABLENAME SET 字段=”数值” WHERE 条件=”数值”

DROP TABLE TABLENAME 删除表

SELECT \* FORM 表名 选中数据库里面所有数据

（4）DDL

ALTER: 修改已存在的数据表(例如增加/删除列)和索引

CREATE: 建立新的数据库或数据表

DELETE: 删除表的记录

DROP: 删除数据表或数据库

COMMENT 添加一个注释

GRANT –Gives USER ACCESS PRIVILEGES TO DATABASE 给某个用户授权

REVOKE 收回授予的权限

## 2.3开发语言与开发工具介绍

(1)SpringBoot简介 [9]

Spring Boot简化了基于Spring的应用开发，你只需要"run"就能创建一个独立的，产品级别的Spring应用。 我们为Spring平台及第三方库提供开箱即用的设置，这样你就可以有条不紊地开始。多数Spring Boot应用只需要很少的Spring配置。

你可以使用Spring Boot创建Java应用，并使用java -jar启动它或采用传统的war部署方式。我们也提供了一个运行"spring脚本"的命令行工具。

我们主要的目标是：

* 为所有Spring开发提供一个从根本上更快，且随处可得的入门体验。
* 开箱即用，但通过不采用默认设置可以快速摆脱这种方式。
* 提供一系列大型项目常用的非功能性特征，比如：内嵌服务器，安全，指标，健康检测，外部化配置。
* 绝对没有代码生成，也不需要XML配置。

环境配置：

默认情况下，Spring Boot 1.4.0.BUILD-SNAPSHOT 需要Java7环境，Spring框架4.3.2.BUILD-SNAPSHOT或以上版本。你可以在Java6下使用Spring Boot，不过需要添加额外配置。具体参考Section 82.11, “How to use Java 6” 。明确提供构建支持的有Maven（3.2+）和Gradle（1.12+）。

**注**：尽管你可以在Java6或Java7环境下使用Spring Boot，通常建议尽可能使用Java8。

Servlet容器开箱即用：如下表 2-1 版本表

| **名称** | **Servlet版本** | **Java版本** |
| --- | --- | --- |
| Tomcat 8 | 3.1 | Java 7+ |
| Tomcat 7 | 3.0 | Java 6+ |
| Jetty 9.3 | 3.1 | Java 8+ |
| Jetty 9.2 | 3.1 | Java 7+ |
| Jetty 8 | 3.0 | Java 6+ |
| Undertow 1.3 | 3.1 | Java 7+ |

表2-1 版本表

（2）LayUI框架介绍：[10]

是一款采用自身模块规范编写的前端 UI 框架，遵循原生 HTML/CSS/JS 的书写与组织形式，门槛极低，拿来即用。其外在极简，却又不失饱满的内在，体积轻盈，组件丰盈，从核心代码到 API 的每一处细节都经过精心雕琢，非常适合界面的快速开发。layui 首个版本发布于2016年金秋，她区别于那些基于 MVVM 底层的 UI 框架，却并非逆道而行，而是信奉返璞归真之道。准确地说，她更多是为服务端程序员量身定做，你无需涉足各种前端工具的复杂配置，只需面对浏览器本身，让一切你所需要的元素与交互，从这里信手拈来。

（3）IntelliJ IDEA 介绍：[11]

IDEA 全称 IntelliJ IDEA，是Java语言开发的一个集成换金，IntelliJ在业界被都公认是Java最好的开发工具之一，尤其是在代码重复率自动的提示，智能代码的助手、以及其他各种版本控制工具（git、SVN、github等）、Junit、代码分析这些方面可以说超常了。它的旗舰版本还支持HMTL、CSS、PHP、MySQL这些流行的语言及其框架的支持。

IDEA所提倡的是智能编码，是减少程序员的工作，IDEA的特色功能有以下几点。

智能选取某个方法、查找适合的丰富的导航模式、通过历史记录很容易恢复、Junit的完美支持、几乎完美支持的重构、编码辅助功能、项目不同采用不同的排版方式、XML文件安全提示、动态语法检测、代码自动分析、JSP完美支持、智能自动补充方法、完全支持EJB、列编辑模式、自动设置需要的模板、自动代码完成、git、svn等版本控制器的支持等等优势

（4）Java语言介绍:[12]

Java是一门面向对象编程语言，不仅吸收了C++语言的各种优点，还摒弃了C++里难以理解的多继承、指针等概念，因此Java语言具有功能强大和简单易用两个特征。Java语言作为静态面向对象编程语言的代表，极好地实现了面向对象理论，允许程序员以优雅的思维方式进行复杂的编程。

Java语言特点：

1. 简单性、2.面向对象、3.分布式、4.编译和解释性、5.稳健性、6.安全性、7.可移植性、8.高性能、9.多线索性、10.动态性。

应用范围广以及在手机端PC端都采用了不同的技术发展方向来开展。

# 3.系统分析

## 3.1系统需求分析

### 3.1.1系统需求概述

数码商城是基于Internet网上数码产品销售系统，用户可以通过Internet浏览网上数码产品的类型，在登录的情况下能进行数码产品加入自己的购物车中，如果用户决定购买该产品，则可以将购物车中的商品进行结算操作。随后管理员会定期对订单进行相关的处理。

本数码商城旨在于方便用户购买数码商品以及每一个用户都能购买到自己合适相应的正品数码产品。同时也具有商品的上架下架等功能这些都是需要相应的权限来进行相应的操作。

### 3.1.2数据库需求分析

设计一个性能良好数据库[13]，我们首要要明确应用环境对系统的次重点。因此，应该吧用户需求的收集和分析作为数据库设计的第一步。

需求分析的主要任务是通过详细调查要处理的对象，包括某个组织、某个部门、某个企业的业务管理等，充分了解原手工或原计算机系统的工作概况及工作流程，明确用户的各种需求，产生数据流图和数据字典，然后在此基础上确定新系统的功能，并产生需求说明书。值得注意的是，新系统必须充分考虑今后可能的扩充和改变，不能仅仅按当前应用需求来设计数据库。

如图3-1 需求分析 所示，需求分析具体可按以下几步进行：

(1) 用户需求的收集。

(2) 用户需求的分析。

(3) 撰写需求说明书。

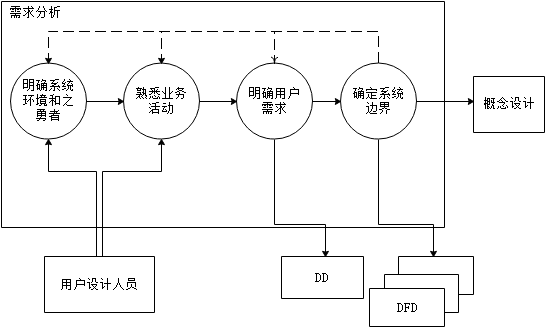


图3-1 需求分析的过程

需求分析的重点是调查、收集和分析用户数据管理中的信息需求、处理需求、安全性与完整性要求。信息需求是指用户需要从数据库中获得的信息的内容和性质。由用户的信息需求可以导出数据需求，即在数据库中应该存储哪些数据。处理需求是指用户要求完成什么处理功能，对某种处理要求的响应时间，处理方式指是联机处理还是批处理等。明确用户的处理需求，将有利于后期应用程序模块的设计。

## 3.2系统功能分析

### 3.2.1用户功能分析

（1）商品浏览功能：以列表方式显示数码商品信息，这样能展示大量的商品信息，同时可以为用户提供更多的商品浏览方式，分类显示浏览、最新商品、热销商品等方式展现给用户。

（2）购物车：当用户找到自己所喜欢的产品的时候，可以先将商品放入购物车中，然后继续寻找其他的商品，购物车中存放着当前用户想购买的所有数码产品。

（3）商品订单：当用户从购物车中点击购买以后，完成支付就是完成了下订单的操作。管理员会在定期处理用户下达的订单，并根据用户指定的收货地址进行发货处理。

（4）商品搜索：网上数码商城的搜索能按分类进行相关的分类搜索，还能对数码名称、品牌、生产商等进行条件中模糊条件查找，并能快速找到所搜索的相关商品。

（5）用户注册：只有称为数码商城的注册用户后方可在本数码商城中购物，统提供用户注册功能及个人信息的修改密码和维护等操作。

### 3.2.2管理员功能分析

数码商城的管理员功能是维护系统正常运行的重要角色，它需要提供如下的功能：

（1）信息管理：添加新管理员、删除管理员、修改密码等操作。

（2）账户管理：能对注册后的用户进行相应的添加、删除操作。

（3）商品管理：由于各类数码产品是数码商城的内容，管理员需要维护系统中的产品信息，也是就对产品分类进行相关的添加删除修改商品信息。

（4）订单管理：用户下达订单后，管理员需要对用户订单进行处理，为用户准备订购的商品，并组织送货以及后台的发货点击操作等。

## 3.3可行性分析

可行性分析目的就是用最小的代价在尽可能短的时间内确定问题是否能够解决、是否值得去解。下面从四个方面分析数码商城的可行性：

（1）技术可行性：本数码商城采用MySQL进行后台数据库管理、操作和维护，用SpringBoot搭建实现系统的后台功能。LayUi提供前端页面的美观信息可以方便地创建动态、交互性强的web网页。Jquery和JS提高用户的可操作性。这重复说明本数码商城在技术方面可行。

（2）经济可行性：由于目前我国互联网事业发展迅速，网上购物越演越烈之势，企业的商务网站想要以最低的投入成本获取更高的商业利益，必须开发个易于管理、后期维护费用低廉、界面友好安全性高的商务系统。因此本系统采用SpringBoot来搭建整个项目来减少用户的更少的配置信息。

（3）操作可行性：该系统设计思路简单清晰，体验更良好的用户界面，操作起来方便简洁，有完善的异常信息处理机制和如何提示信息的机制，因此操作方面可行的。

（4）法律可行性：本系统没有违反国家的相关法律规定，法律方面是可行的。

## 3.4系统安全性分析

在现代社会中，网上购物已经成为各个传统产业提高自己受益的有利武器。在各大企业纷纷建立自己的网上购物平台的时候，作为一个购物网站，安全性会直接影响到企业和客户的利益所在，因此网站的安全性应该放在首要的位置。

（1）用户权限安全性

系统设置了系统管理员和一般用户两种身份以满足安全性要求，他们分别来之与不同的数据库表信息，只有拥有管理权限的人才能对系统后台系统进行相关等权限操作。

（2）保证安全性的方法有：用户信息的数据的加密机制以及合法验证机制等。

（3）系统安全性：系统的处理过程对输入首先进行合法性验证，最大限度满足数据的安全性，保证系统的稳定性。

（4）本系统对安全性如下处理：

首先，在网站注册的用户用手唯一的账号，此账号唯一标识用户身份，用户凭此信息来进行相应的购物操作。其次，系统还提供了密码加密措施，利用DigestUtil来进行Sha加密的方式采用不可逆的字符串加密方式。

## 3.5主要业务逻辑分析

商品管理流程如图：3-2 商品管理

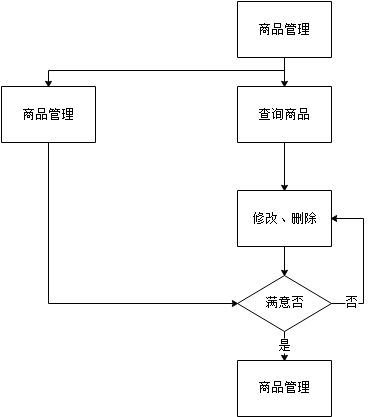


图3-2 商品管理

用户登录流程如图3-3

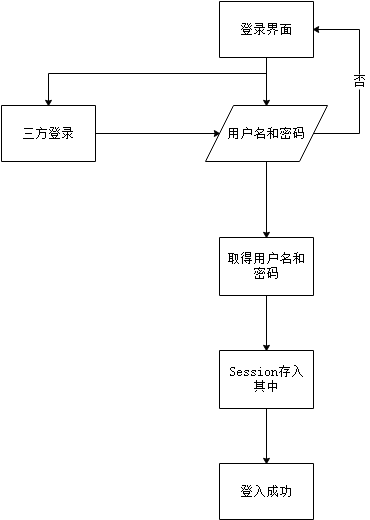


图3-3 用户登录

购物流程如图3-4

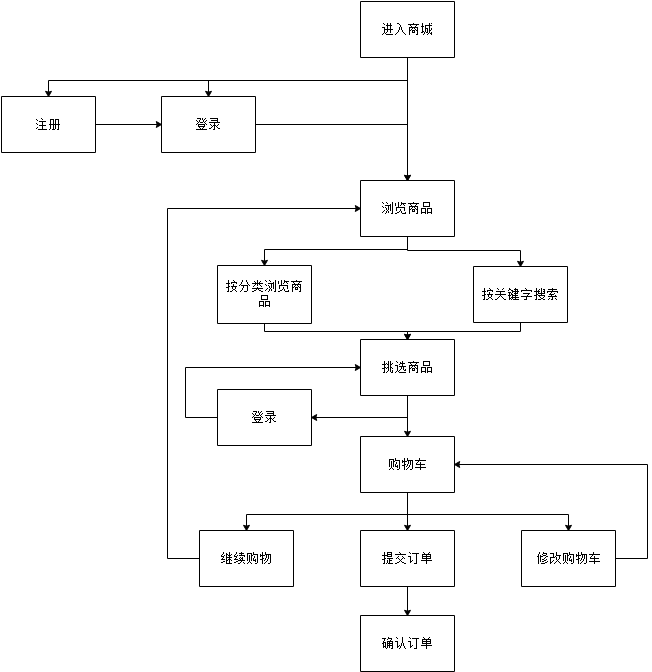


图3-4 购物流程

# 4系统设计

## 4.1数据库设计

### 4.1.1 概念结构设计

概念接受设计的任务是在需求分析阶段的时候产生的需求说明说的基础上[14],按照一些特定方法把他抽象成一个不依赖机器的一种数据模型，即成为概念模型。概念模型让我们设计者从复杂的实现过程中解脱了出来，只用去考虑组织之间的结构和处理模式。

（1）概念模型是对现实世界的一种抽象概括，它能充分反映现实世界中两种事物之间的关系，能满足用户对数据的处理请求。

（2）概念模型和不熟悉计算机的用户交换意见，使用户能积极参与数据的设计工作，让复杂的东西简单化，保证设计工作顺利进行。

（3）概念模型易于用户更新，当外界环境和应用要求发生改变的时候，概念的改变和扩充变得更加的容易。

（4）概念模型向网状、层次等其他数据模型转换便捷。

描述概念模型很有力的工具是E-R图[15]。E-R是也称实体-练习图（Entity Relationship Diagram）,提供了表示实体类型、属性和联系的方法，用来描述现实世界的概念模型。用“矩形框”表示实体型，矩形框内写明实体名称；用“椭圆图框”表示实体的属性，并用“实心线段”将其与相应关系的“实体型”连接起来。

系统使用实体-联系（E-R）方法建立概念模型，这样是数据库和程序结构更加清晰。产品实体E-R图如图4-1：

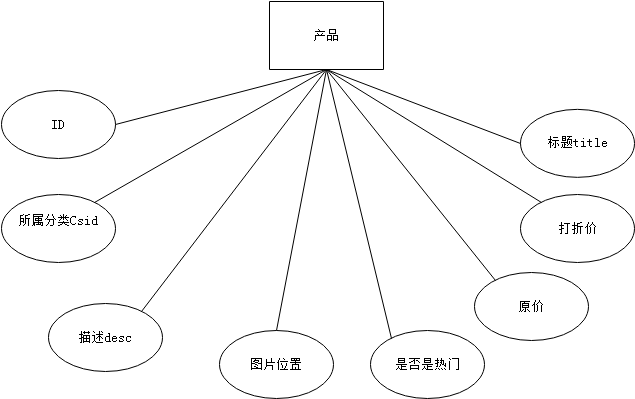


图 4-1 产品E-R图

用户实体E-R图如图 4-2：

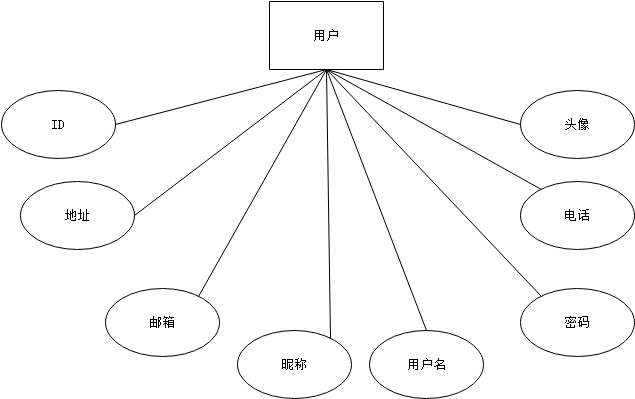


图 4-2 用户E-R图

订单实体E-R图如图 4-3：

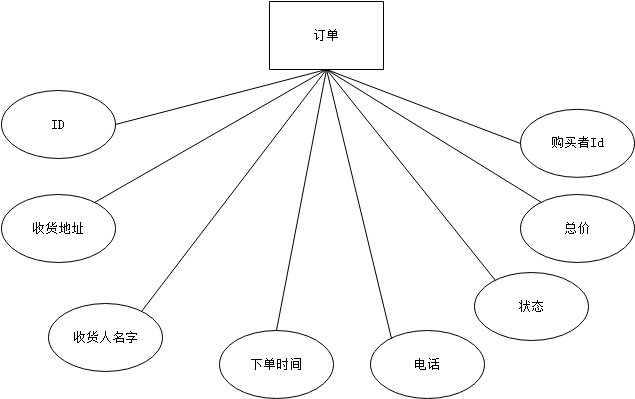


图4.3 订单E-R图

实体及其联系如图 4-4:

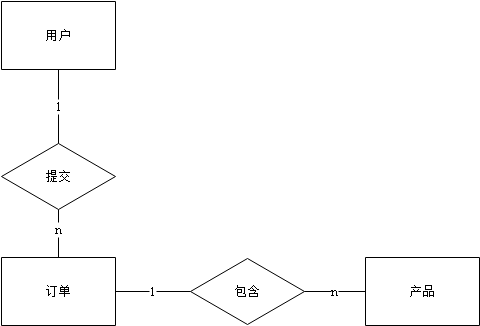


图4-4 实体-联系图

### 4.1.2 逻辑结构设计

逻辑结构设计是将概念结构设计阶段完成的概念模型[16]，转换成能被选定的数据库管理系统支持的数据模型。这里就是将E-R模型转换为相应关系模型。

逻辑结构设计一般分为三步进行：

（1）E-R图向关系模式转化

（2）数据模型的优化

（3）关系视图设计

如图 逻辑结构设计4-5所示：

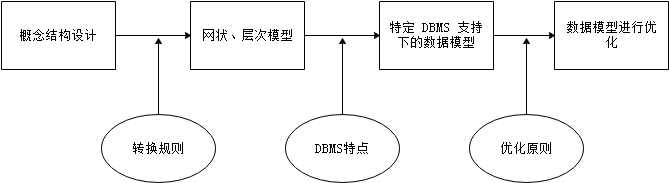


图4-5 逻辑结构设计的过程

下面我们将E-R模型转成为需要的关系模型，就是将实体属性和联系都要转化为相应的关系模型。下面我们是将用户订单实体关系模型如图4-6：

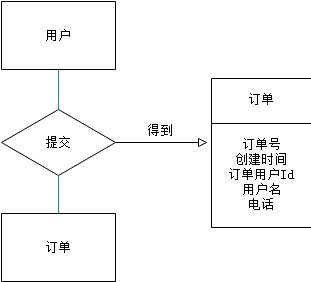


图4-6 用户-订单实体类关系模型图

下面是订单-产品实体关系模型4-7：

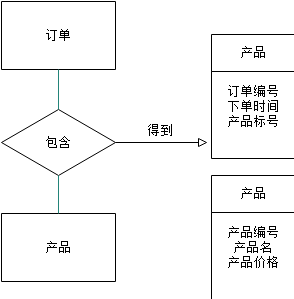


图4-7 订单-产品实体类关系模型图

### 4.1.3 物理结构设计

数据库物理设计师后半段[17]。将一个给定逻辑结构实施到具体的环境中时，逻辑数据模型要选取一个具体的工作环境，这个工作环境提供了数据存储结构与存取方法，这个过程就是数据库的物理设计。

确定数据库的物理结构包含下面四个方面的内容：

（1）、确定数据的存储结构

（2）、设计数据的存取路径

（3）、确定数据的存放位置

（4）、确定系统配置

从上面得出设计数据库表如图4-8 user表：

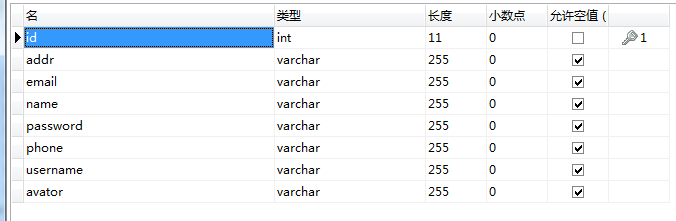


图4-8 user表

Admin表如 4-9图：

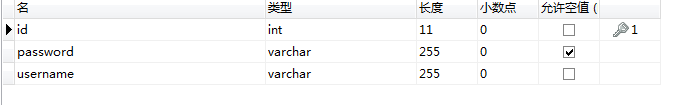


图 4-9 admin表

Order表如 4-10图：

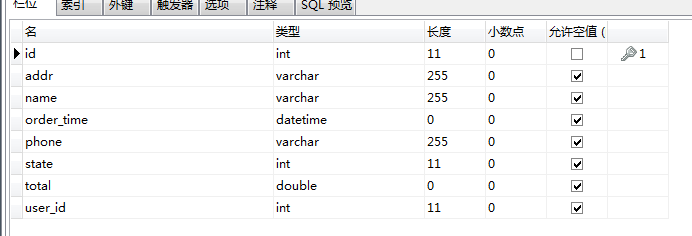


图4-10 order表

Product表 图4-11图：

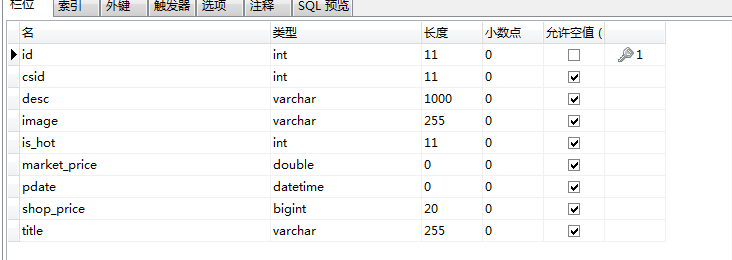


图 4-11 product表

t\_pub\_user\_oauth表 如图 4-12 图：



图4-12 t\_pub\_user\_oauth表

Tb\_comment表 如图 4-13 图：

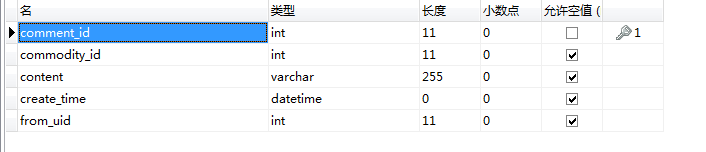


图 4-13 tb\_comment表

Tb\_reply表 如图 4-14 图：

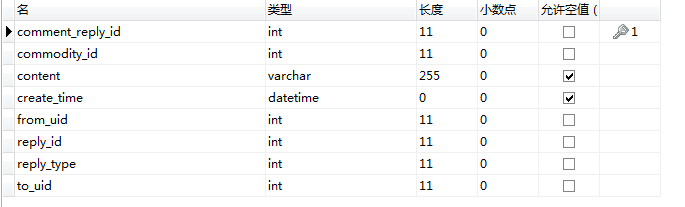


图 4-13 tb\_reply表

Classification表 如图4-14图：

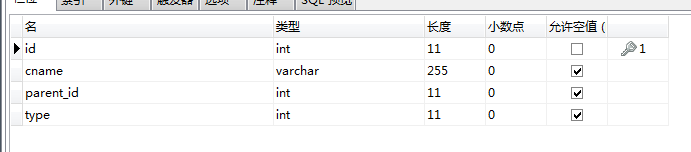


图4-14 classification 表

Order\_item 表 如图 4-15图：

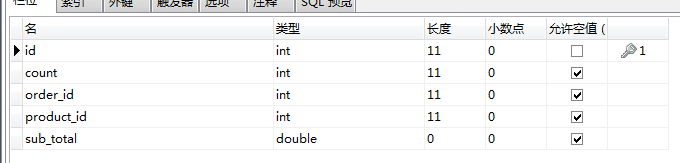


图4-15 order\_item表

## 4.2网站功能结构

### 4.2.1前台功能结构

数码商城前台功能的结构如图4-16

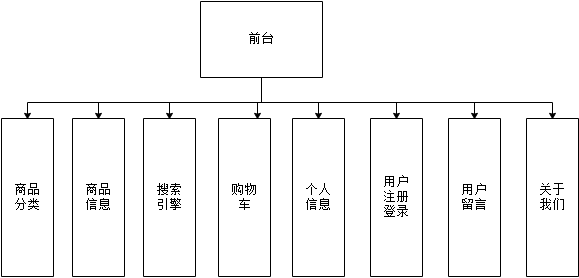


图 4-16前台功能结构图

### 4.2.2后台功能结构

数码商城后台结构如图 4-17

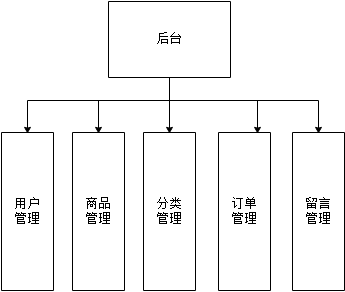


图4-17后台功能结构图

## 4.3网站页面结构

### 4.3.1前台页面结构

数码商城前台页面设计采用了layUi的框架搭建整个页面，搜索引擎，导航栏菜单，页面轮播，用户登录注册、商品详细分裂导航、销售品牌、热门搜索。使用thymeleaf模板引擎里面th：replace 方法来头部尾部公用。

|  |  |  |  |
| --- | --- | --- | --- |
| 网站Logo 搜索引擎 导航条 | | | |
| 前面分类导航 中间轮播效果 | | | |
| 热门商品 | | | |
|  |  |  |  |
|  |  |  |  |
| 最新商品 | | | |
|  |  |  |  |
|  |  |  |  |
| Footer | | | |

### 4.3.2后台页面结构

数码商城后台页面采用了头尾部公用的方法，以及iframe局部刷新主要的内容。

|  |  |
| --- | --- |
| 后台Logo 头部信息 | |
| 菜单栏 | 局部刷新的主要内容 |
| Footer | |

# 5.系统实现

## 5.1前台主要模块的实现

### 5.1.1 产品展示模块

对于购物者来说最关注的商品的就是商品型号商品价格，这样的设计吸引了用户的购买欲望。下面用户评论的摆放方式让能让用户快捷简便的看到评论况。

如图5-1：



图5-1 产品展示

点击购物车按钮后没有登录的话会提示你进行登录操作后在进行购买这个，如果用户进行登录了后就调用addCart.do?productId=?把相应商品Id保存在Session中。

### 5.1.2购物车模块

如图购物车5-2：



图 5-2 购物车

购物车在日常生活中扮演重要的角色，在看到自己喜欢的商品的还是点击购物车后，可以继续进行购物的操作。在添加购物车的时候就想好自己到底需要买几件，如果购买多件的时候就在图5-1中点击购物车按钮点击相应的次数就添加到自己购物车中了。在添加购物车后对比发现需要的不是这个时候可以点击图5-2中的删除按钮，去掉自己不想要的。

图5-2中点击提交订单前需要在里面填入收货人相应的信息。填完信息后跳到了我订单信息的情况页面。如图5-3 订单信息页面



图5-3 订单页面

图5-3点击查看详情按钮就会有相应的商品信息页面展现出来，点击付款按钮就会调到了如下图5-4支付页面：



图5-4 支付页面

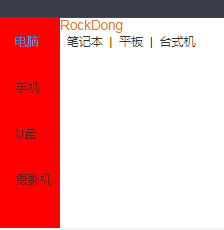
填入相应自己的信息就完成支付操作。本次支付的页面是支付宝沙箱以及支付宝提供的沙箱账号和沙箱用户来完成支付。模拟用户支付给商家的操作。完成支付后调到了如下图5-5 订单页面



图 5-5 订单页面

### 5.1.3搜索模块

分类搜索模块：如下图5-6所示：



这些信息采用了ajax方法调用了后台的menuli？parentId=？传入相应的父元素Id就能查找出下面对应的子元素的信息。由于每次都会采用ajax请求的方式，所以我后台采用了ehcache方式把请求同一个方法名的缓存起来，这样大大减少了服务器承受的压力。

代码如下：

Controll层代码：

@ResponseBody

@RequestMapping("/menuli")

public List<Classification> menulishow(int parentId){

List<Classification> classifications=classificationService.findByParentId(parentId);

Printer.info(CacheManager.class.getName());

return classifications;

}

Dao层代码：

@Cacheable(key = "#cid",value = "userCache")

public List<Classification> findByParentId(int cid) {

Printer.info("这里为cid做了缓存机制"+cid);

return classificationDao.findByParentId(cid);

}

以及ehcache.xml配置信息如下：

<?xml version="1.0" encoding="UTF-8"?>

<ehcache xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="http://ehcache.org/ehcache.xsd">

<diskStore path="user.dir/ehcache" />

<!-- 配置提供者 1、peerDiscovery，提供者方式，有两种方式：自动发现(automatic)、手动配置(manual) 2、rmiUrls，手动方式时提供者的地址，多个的话用|隔开 -->

<cacheManagerPeerProviderFactory

class="net.sf.ehcache.distribution.RMICacheManagerPeerProviderFactory"

properties="peerDiscovery=manual,rmiUrls=//localhost:40002/userCache" />

<!-- <cacheManagerPeerProviderFactory

class="net.sf.ehcache.distribution.RMICacheManagerPeerProviderFactory"

properties="peerDiscovery=automatic, multicastGroupAddress=230.0.0.1, multicastGroupPort=4446,timeToLive=255"/>

-->

<!-- 配置监听器 1、hostName 主机地址 2、port 端口 3、socketTimeoutMillis socket子模块的超时时间，默认是2000ms -->

<cacheManagerPeerListenerFactory

class="net.sf.ehcache.distribution.RMICacheManagerPeerListenerFactory"

properties="hostName=localhost, port=40001, socketTimeoutMillis=2000" />

<!-- <cacheManagerPeerListenerFactory

class="net.sf.ehcache.distribution.RMICacheManagerPeerListenerFactory"/> -->

<defaultCache eternal="false" maxElementsInMemory="1000"

overflowToDisk="false" diskPersistent="false" timeToIdleSeconds="0"

timeToLiveSeconds="600" memoryStoreEvictionPolicy="LRU" />

<cache

name="userCache"

maxElementsInMemory="1000"

eternal="false"

timeToIdleSeconds="300"

timeToLiveSeconds="300"

overflowToDisk="false"

memoryStoreEvictionPolicy="LRU">

<!-- 配置缓存事件监听器 replicateAsynchronously 操作是否异步,默认值为true. replicatePuts 添加操作是否同步到集群内的其他缓存,默认为true.

replicateUpdates 更新操作是否同步到集群内的其他缓存,默认为true. replicateUpdatesViaCopy 更新之后的对象是否复制到集群中的其他缓存(true);

replicateRemovals 删除操作是否同步到集群内的其他缓存,默认为true. -->

<cacheEventListenerFactory

class="net.sf.ehcache.distribution.RMICacheReplicatorFactory"

properties="

replicateAsynchronously=true,

replicatePuts=true,

replicateUpdates=true,

replicateUpdatesViaCopy=true,

replicateRemovals=true " />

<!-- 初始化缓存，以及自动设置 -->

<bootstrapCacheLoaderFactory

class="net.sf.ehcache.distribution.RMIBootstrapCacheLoaderFactory"

properties="bootstrapAsynchronously=true" />

</cache>

</ehcache>

使用SpringConfiguration来配置Ehcache bean

/\*\*

\*ehcache配置信息

\*/

@Configuration

@EnableCaching

public class CacheConfiguration {

/\*\*

\* ehcache主要的管理器

\*@return

\*/

@Bean

public EhCacheCacheManager ehCacheCacheManager(EhCacheManagerFactoryBean bean){

Printer.info("CacheConfiguration.ehCacheCacheManager()");

return new EhCacheCacheManager(bean.getObject());

}

@Bean

public EhCacheManagerFactoryBean ehCacheManagerFactoryBean(){

EhCacheManagerFactoryBean cacheManagerFactoryBean=new EhCacheManagerFactoryBean();

cacheManagerFactoryBean.setConfigLocation(new ClassPathResource("ehcache.xml"));

cacheManagerFactoryBean.setShared(true);

return cacheManagerFactoryBean;

}

}

## 5.2后台主要模块的实现

### 5.2.1网站配置模块

作为一个成熟的网络应用应该具备配置模块，使商家能够快捷方便的进行相应的配置。这使得商家更新企业信息时候能只需要更新对应的配置信息而不需要更新硬编码的页面。商家更换了话只需要更新对应的配置模块。

网站的配置信息主要依赖于AlipayConfiguration类中的配置信息修改对应的商家UUID，这样用户在支付成功之后才能收到来自用户的金额。详细实现代码参见 AlipayConfiguration.java中。

### 5.2.2产品管理模块

产品管理模块乃是一个数码商城的核心中的核心，管理员能够维护系统中所有产品信息。同时与产品相关的产品类型信息也需要相应的维护。具体包括：添加、删除商品类别；添加、修改、删除商品信息。

商品类添加的实现页面为：/mall/admin/category/add.html。用户在添加商品类的时候有一级分类和二分类之说。一级分类来区分每一件数码属于不同的类别，而二级分类就是来区分同一个类下不同型号的数码产品。我们在添加二级分类的时候需要选择对应的一级分类这样就合理设计出了分类的不同分类显示的要求。

在对商品分类处理了后就是相应的商品信息添加了，填入相应的数据进行商品上架，二级分类决定了商品属于哪个分类下。我们所有信息都是存入数据库了，除去上传图片的这个功能，利用BootStrap 框架来在图片上传后显示出来是什么样子。然后利用FileUtil.java这个工具类实现了上传到服务器，返回的时候得到一个对应的URL地址也就是保存在服务器的相对路径位置。在这里使用了MessageDigest.getInstance("MD5")这个方法让图片不能上传重复的图片到服务器上去。详细代码请参考FileUtil.java工具类。

### 5.2.3订单管理模块

客户下订单以后管理员就能对订单的状态进行修改或者删除订单，能对订单内容进行查看但不能对订单内容进行修改。订单管理页面实现页面为/mall/admin/order/list.html

这里list.html采用简单的方式来进行，直接点击list.html页面中的发货按钮就能进行相应的发货处理。这里点击发货按钮的时候发送ajax请求send.do?id=?传入对应订单号Id查询出订单这个对象，然后重新setState(state)状态信息修改为3已发货状态。然后在重新save一次这个order对象就能进行相应的修改了。

### 5.2.4用户管理模块

这里用户模块里面的信息就不再详细的阐述了，只是对数据库相应的修改删除操作。这里使用到数据操作利用了SpringData来方便的操作数据库信息。SpringData是结合SQL语句进行的一些封装只所以使用SpringData是因为相比SQL操作更加简洁和高效。但在链表查询的时候就要使用到SQL语句，我们就要使用到@Query这个注解来写自己定义的查询方法。

# 6系统调试与测试

## 6.1系统调试

为使系统在运行时能够达到安全、高效、稳定的技术指标我们对系统进行了整体的调试。在调试过程中遇到一些复杂的问题，这个时候需要我们耐心以待去处理好问题，包括每一个页面和多个关联页面组合起来的整体流程都要进行调试一次。

## 6.2单个页面调试

在开发过程中，每个页面的设置都会影响到整个系统的运行。因此，要对每个页面的主要组成部分进行调试，包括页面里面的背景颜色、字符风格、列表、文本等。在本次设计中，对以上内容都惊了反复调试，直到达到让人满意的效果。

在单个页面中一般用到的调试是页内变量的追踪，这时候就在调试的时候要注意变量的信息跟踪，后台如何能取到这个值的信息。

## 6.3 多个页面调试

多个页面的调试，主要有链接和表单调试。调试包括跳转的Url以及后面传值的对错，后台对应Controller里面对应的URL位置。以及表单测试包括input表单里面value取值和name传值是否能对应，注意使用的method方法选择以及Action是否在后台有对应的类。文本框和按钮制作，页面中大量Ajax传值和取值的调试程序。在完成页面调试的时候要注意是否修改的页面是否会影响到其他页面的改变，以及数值是否会有改变。

## 6.4软件测试

测试的目的在于发现已有的错误。由于代码在编写过程中是不可避免会有出现一些逻辑语法上的错误。因此，在程序投入运行值钱，必须纠正其中的大部分错误。否则，这些差错迟早会在程序运行过程中暴露出来，到时候会造成很恶劣的后果。测试的目的就是在运行以前，尽可能发现程序中的错误，这是保证软件质量的关键步骤。

在程序测试时对整个系统做了一个全面测试，测试的主要内容如下：

（1）界面可操作性行测试。

（2）商品分类、查询模块的测试。

（3）支付模块的测试。

（4）购物整体流程的测试。

（5）用户管理后台、订单管理、分类管理的分别测试。

（6）整体系统的测试。

在整个毕业设计完成后，就是按照以上内容对整个系统进行相应的测试。其中包含一些不足和至今没有实现的功能，经反复测试、修改，最后达到想要的效果。

# 总结

在经过几个月的开发设计，该网站基本实现了最初的设计思想，满足了用户在购买上基本需求，能够较为接近预定的功能指标，实现了用户注册登录，购买喜欢的数码产品，后台数据的管理；具有良好的用户信息安全保密性和一定容错性。拥有友好的用户界面，购物方便快捷，具有较强的可一致性、可扩充性和管理员维护起来简介性。但是，由于受到时间、技术条件、技术能力的约束，该网站仍然有许多不如意的地方：比如在用户并发量的这个点上，并没有使用SpringCloud微服务和Dubbo分布式以及负载均衡来减少服务器的压力。

通过对本程序的开发设计，我不仅学到了很多关于IDE的使用和开发技术知识，每次对自己出现的bug进行了分析和总结，增加了对JavaWeb开发的经验。尤其在对具体的问题的解决上，得到知道老师很大的帮助，改变了自己的思维方式，让一件复杂的事情变得简单起来，这对将来的发展奠定了一个号的开端。

随着对本程序的开发不断深入的时候，我对数据库理论知识有了更深入的理解，让我知道了数据设计是关乎到这个程序是否功能齐全的重要因素。

总体来说，在这次毕业论文的设计中，把以前学的知识和技能付诸于实践当中，达到了理论指导实践，实践巩固理论的目标。深化了自身的理论知识，提高了实践能力，为今后的学习和工作打下了坚实的基础。

# 参考文献

1. 凌启节 网络购物税收征管的探讨[J],中国证券期货,2011-04
2. [美]克雷格·沃斯 Spring Boot实战[M] 人民邮电出版社 2016-9
3. 李志伟 DB2基础教程[M] 清华大学出版社. 2006
4. 许晓斌 Maven实战 机械工业出版社[M] 2010-12
5. 纯洁的微笑 SpringBoot[EB/OL] http://www.cnblogs.com/ityouknow/p/5891443.html2018-04
6. (美)奥尼尔 著(美)数据库原理 [M]，性能与编程 2001年05月
7. 陶国荣 jQuery权威指南（第2版）[M] 北京机械工业出版社 2013
8. Internetchina MySQL数据库 [EB/OL] https://baike.baidu.com/item/MySQL%E6%95%B0%E6%8D%AE%E5%BA%93 2018-02
9. Hellojukay[译] Spring Boot 文档 [EB/OL] http://ifeve.com/spring-boot-documentation/ 2016-09
10. 贤心LayUi框架[EB/OL] http://www.layui.com/doc/ 2018
11. scvwn1631 IntelliJ IDEAj介绍 [EB/OL] https://baike.baidu.com/item/IntelliJ%20IDEA 2017-11
12. Yefeizi Java [EB/OL] https://baike.baidu.com/item/java/85979#1
13. book.csdn.net 数据库设计：需求分析[EB/OL] http://www.uml.org.cn/sjjm/201007224.asp 2010-07
14. fchope 概念结构设计 [EB/OL] https://baike.baidu.com/item/%E6%A6%82%E5%BF%B5%E7%BB%93%E6%9E%84%E8%AE%BE%E8%AE%A1 2016-01
15. Turelycafe E-R图 [EB/OL] https://baike.baidu.com/item/E-R%E5%9B%BE 2018-02
16. cyclonefly0000 数据库的逻辑结构设计 [EB/OL] https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E7%9A%84%E9%80%BB%E8%BE%91%E7%BB%93%E6%9E%84%E8%AE%BE%E8%AE%A1 2017-07
17. bk10103数据库物理设计 [EB/OL] https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E7%89%A9%E7%90%86%E8%AE%BE%E8%AE%A1 2016-11

# 致谢