

毕业设计说明书



|  |  |  |
| --- | --- | --- |
| 题目 | : | 手机商城app的设计与实现 |
| 学院 | : | 计算机与软件工程学院 |
| 年级专业 | : | 2018级软件工程 |
| 姓名 | : | 胡远林 |
| 学号 | : | 3120180905402 |
| 指导教师 | : | 陈晓军 |
| 完成时间 | : | 2022年12月18日 |

# 摘要

电子商务的理念是要简化传统购物的繁琐，让用户能够在休闲放松时就能轻松地买到自己心怡的物品；同时在市场竞争激烈的环境下，让每个商家都有机会暴露在大众的视野中，从而提升产品知名度。这个理念与移动互联网的特征(简捷、用户量大)相契合。所以现在各种商城app仍然层出不穷，其中不乏有取得不菲价值的。

本毕业设计使用Ionic、Spring Boot、Redis、Mybatis、MySQL技术实现了一个手机商城系统。本系统的各个功能模块如下，买家中心：注册、登录、密码找回、首页商品推荐、首页轮播图、商品分词搜索、当天热词统计、搜索历史管理、商品下单、订单付款、消息交互、购物车管理、个人信息维护、收货地址管理、商品收藏管理、店铺关注管理、浏览历史管理、订单定时取消、订单手动取消、订单确认收货、订单评论管理、查看商品详细信息、入驻商家；商家中心：店铺信息管理、店铺收款账户管理、上传店铺执照、商品发布、审核中商品管理、售卖中商品管理、订单发货、订单检索、消息交互、轮播图名额抢取；平台管理：商品付款方式管理、平台收付款账户管理、轮播图工厂配置、消息通知、商品审核、商品种类管理。

本系统通过测试，各个功能模块与预期基本一致，达到了毕业设计的目的和要求。

**关键词：**Ionic; Spring Boot; Mybatis; Redis; MySQL; 商城系统; App

# ABSTRACT

The idea of e-commerce is to simplify the complexity of traditional shopping, so that users can easily buy their own products when they relax, with the opportunity to expose them to the public, so as to enhance product awareness.This concept fits with the characteristics of the mobile Internet (simplicity, large number of users).So now a variety of mall app still emerge in an endless stream, some of them have achieved high value.

The main work of this manual: introduces the system demand analysis, system overall design, system detailed design, system implementation, system test.Completed the various functions of the system module: buyer registration/login, buyer personal information maintenance, commodity display and retrieval, shopping cart management, order payment, order logistics information query, order cancellation, confirm receipt, transaction evaluation, merchants in, store information management, commodity management, store order management, commodity management, commodity type attribute management, platform store management, payment management.

The system completed the client with angular, ionic,cordova. And service was completed with springboot, mysql, redis.And passed the test, to achieve the purpose and requirements of the graduation design.

**KEY WORDS:** Ionic; Angular; Cordova; SpringBoot; Redis; MySql; mall system; app

目录

[第1章 绪论 1](#_Toc104234389)

[1.1开发背景 1](#_Toc104234390)

[1.2国内外现状 1](#_Toc104234391)

[1.3实现技术介绍 2](#_Toc104234392)

[1.4说明书的组织结构 2](#_Toc104234393)

[第2章 需求分析 3](#_Toc104234394)

[2.1功能需求 3](#_Toc104234395)

[2.1.1买家中心功能需求分析 3](#_Toc104234396)

[2.1.2商家中心功能需求分析 12](#_Toc104234397)

[2.1.3平台后台功能需求分析 19](#_Toc104234398)

[2.2数据库分析 26](#_Toc104234399)

[2.2.1实体关系图 26](#_Toc104234400)

[2.3非功能需求分析 30](#_Toc104234401)

[第3章 系统概要设计 31](#_Toc104234402)

[3.1功能模块设计 31](#_Toc104234403)

[3.2系统架构设计 31](#_Toc104234404)

[3.3接口设计 32](#_Toc104234405)

[3.4数据库概要设计 46](#_Toc104234406)

[3.4.1实体转换为关系模式 46](#_Toc104234407)

[3.4.1实体关系转换为关系模式 48](#_Toc104234408)

[第4章 系统详细设计 50](#_Toc104234409)

[4.1公共模块详细设计 50](#_Toc104234410)

[4.1.1身份验证设计 50](#_Toc104234411)

[4.1.2文件上传设计 50](#_Toc104234412)

[4.2买家模块详细设计 51](#_Toc104234413)

[4.2.1买家注册设计 51](#_Toc104234414)

[4.2.2买家登录设计 52](#_Toc104234415)

[4.2.3买家个人基本信息修改设计 53](#_Toc104234416)

[4.3商家模块详细设计 53](#_Toc104234417)

[4.3.1商家入驻设计 53](#_Toc104234418)

[4.4订单模块详细设计 54](#_Toc104234419)

[4.4.1发起订单设计 54](#_Toc104234420)

[4.4.2订单支付宝付款设计 55](#_Toc104234421)

[4.4.3取消已付款未发货订单设计 56](#_Toc104234422)

[4.4.4未付款自动取消订单设计 56](#_Toc104234423)

[4.4.5确认收货设计 57](#_Toc104234424)

[4.4.6订单物流查询设计 58](#_Toc104234425)

[4.4数据库详细设计 58](#_Toc104234426)

[第5章 系统实现 69](#_Toc104234427)

[5.1公共模块实现 69](#_Toc104234428)

[5.1.1身份验证实现 69](#_Toc104234429)

[5.1.2文件上传实现 70](#_Toc104234430)

[5.2买家模块 70](#_Toc104234431)

[5.2.1注册实现 70](#_Toc104234432)

[5.2.2登录实现 71](#_Toc104234433)

[5.2.3个人基本信息修改实现 72](#_Toc104234434)

[5.2.4购物车管理实现 72](#_Toc104234435)

[5.3商家模块 73](#_Toc104234436)

[5.3.1商家入驻 73](#_Toc104234437)

[5.3.2商品管理 74](#_Toc104234438)

[5.3.2店铺管理 75](#_Toc104234439)

[5.4平台模块 75](#_Toc104234440)

[5.4.1商品管理 75](#_Toc104234441)

[5.4.2商家入驻 75](#_Toc104234442)

[5.4.3店铺管理 75](#_Toc104234443)

[5.4订单模块 75](#_Toc104234444)

[5.4.1发起订单实现 75](#_Toc104234445)

[5.4.2取消已付款订单实现 75](#_Toc104234446)

[5.4.3店铺管理 75](#_Toc104234447)

[第6章 系统测试 76](#_Toc104234448)

[6.1买家模块 76](#_Toc104234449)

[6.2商家模块 78](#_Toc104234450)

[6.3订单模块 83](#_Toc104234451)

[6.4平台模块 87](#_Toc104234452)

[总结 89](#_Toc104234453)

[致谢 90](#_Toc104234454)

[参考文献 91](#_Toc104234455)

# 第1章 绪论

## 1.1开发背景

随着智能手机的普及，5G网络的推进，全民的生活数字化已经成为常态。成熟的app开发技术使得移动端的app开发成本降低，开发效率大幅提升。安全便捷的第三方支付交易大幅代替现金交易。由此为商城app的搭建提供良好的技术支撑。现在人们的大多工作繁忙，难以有时间去购物，偶有闲暇，也难得休息。但是借助网络的便捷性，人们可以在休闲的时间动动手指就能购买自己心怡的物品，既省事又放松。另外线下的商家使用传统的交易方式，客流量很少，销售区域受到限制，即使商品品质很好却也难以发展，所以需要借助网络来提高自己的知名度与交易量。

## 1.2国内外现状

国内2020年全年电子商务交易额为37.21万亿元，同比增长6.9%，2021年上半年达9.61万亿元。

2021Adobe发布报告称，全球电商销售额预计将达到4.2万亿美元。

2021年11月12日零点，天猫双11总交易额定格在5403亿,相较去年的4982亿成交额。再度刷新记录，提高了421亿元。

2021年全年，京东净收入为9516亿元（约1493亿美元），同比增长27.6%；其中在第四季度，京东净收入为2759亿元人民币（约433亿美元），同比增长23.0%。

亚马逊发布2021年第四季度及全年财报。2021年第四季度，亚马逊净销售1374亿美元，同比增长9%，分析师预期1378.2亿美元；第四季度经营利润35亿美元，同比增长98%，预估24.3亿美元；净利润143亿美元，较去年同期几乎翻倍。

2021年国内电商平台市场份额：京东3.2万亿，占20%；阿里8.5万亿，占53%；拼多多2.4万亿，占15%；抖音电商8000亿，占5%；快手电商6800亿，占4%；得物800亿，占0.5%；唯品会1900亿，占1.2%；苏宁易购2500亿，占1.6%。

由此可见，尽管近年的疫情猖獗，导致人们消费欲望降低，但电商平台的销售额依旧可观。并且随着各类的新电商平台以及销售方式崭露头角，大量买家不再由各大电商巨头所占据，用户购物时会更加权衡各个平台商品的性价比，故每一款新的商城app都有机会赢得用户的青睐。

## 1.3开发目的及意义

买家能够通过系统便捷的进行商品选购、在线支付、商品评论大大减少买家为购买商品所花费的时间与精力。商家能够及时的对自己的商品进行维护、对买家的订单处理，并且商家不用为了管理店铺而下载不同app,减少了商家的操作的繁琐步骤，提高了商家对交易的处理效率。平台管理员能通过移动平台端完成对商品的审核、订单支付方式的控制，这将会扩大平台管理人员办公的场景，降低了平台管理人员的操作复杂度，提升平台管理人员的办公效率。

## 1.4说明书的组织结构

本说明书结构如下：

第1章：绪论

第2章：需求分析

第3章：系统总体设计

第4章：系统详细设计

第5章：系统实现

第6章：系统测试

总结

致谢

参考文献

# 第2章 需求分析

## 2.1系统涉众分析

### 2.1.1主要涉众

本系统的主要使用角色分为三类，分别是：买家、商家、平台管理人员。买家：该类用户是系统使用量占比最大的用户，主要使用系统进行商品检索、商品购买、商品评价。商家：该类用户是买家的一个子集，因为一部分买家会入驻为商家，是系统使用量占比第二的用户，主要使用系统对自身店铺的商品进行维护、对自身店铺的订单进行处理。平台管理人员：该类用户主要使用系统完成对商家发布的商品进行审核，以及对买家付款方式进行控制。



图2.1.1系统涉众图

## 2.2功能需求

先站在不同用户的角度对用户的需求进行简要的描述，然后归纳用户需求并使用UML[14]进行相应的用例模型创建。

### 2.2.1买家功能需求分析

买家希望能通过系统检索到所需要的商品，并能了解到商品比较全面的信息，然后能对物色好的商品进行购买，同时希望能够挑选的商品能够先存放起来，当选购完成后再统一付款购买，当然最后购买意愿能自由选择。买家希望能够把自己的喜欢的商品、店铺以及看过的商品记下来，方便以后寻找，同时希望当某天这些商品、店铺不存在了，亦或是不再受自己喜爱了，能够清除掉。买家希望自己能对自己的收货地址进行管理。买家希望系统能够按照喜好推荐一些商品给自己以及系统能给出一些当天比较受欢迎的商品。买家希望能够有段缓冲时间来处理因为某些突发情况在付款时导致交易中断的情况。买家希望能够有机会反悔自己已经付款的商品。买家希望能够对某件商品购买体验做出评价。买家希望能与店铺进行沟通。买家希望自己也能够开家小店。买家不希望自己每次打开系统都需要登录。买家不希望自己密码忘记了却无法找回。根据买家需求建立用例模型，如下图2.2.1，避免用例之间关系线过于混杂，图中只展示主要用例之间的关系线。



图2.2.1买家用例模型

### 2.2.2商家功能需求分析

商家希望能够维护自己的店铺信息。商家希望能自己能够管理相关证件。商家希望能动态地变化自己地收款账户。商家希望能够发布自己决定要销售地商品。商家希望能够对发布后审核失败的商品重新申请以及能取消掉发布申请。商家希望能够对发布成功后的商品调整价格、补充库存、添加更多的商品细节以及下架。商家希望能够对订单发货以及检索店铺订单。商家希望能够与买家进行沟通。商家希望能够参与平台提供的服务。根据商家需求建立用例模型，如下图2.2.2。



图2.2.2 商家用例模型

### 2.2.3平台管理员功能需求分析

平台管理员希望能够控制商品的付款方式。平台管理员希望能够管理收付款账户。平台管理员希望能够控制轮播图名额的数目和发放时间。平台管理员希望能接收到系统通知。平台管理员希望能够审核商品。平台管理员希望能够管理商品种类及其属性。根据平台管理员需求建立用例模型，如图2.2.3。



图2.2.3 平台用例图

## 2.3数据库分析

e-r图[7]，即实体关系图，是用于描述数据库系统的概念数据模型[7]，它抽象出了数据库系统中涉及的实体，描述了实体所拥有的属性，以及实体与实体之间的关系。实体与实体之间的关系[7]分为三种，分别是：一对一、一对多、多对多。

### 2.3.1实体

买家:



图2.3.1买家实体

收货地址:



图2.3.2收货地址实体

订单:



图2.3.3订单实体

商品:



图2.3.4商品实体

商家:



图2.3.5商家实体

店铺:



图2.3.6店铺实体

营业执照:



图2.3.7营业执照实体

商家收款账户:



图2.3.8商家收款账户实体

账户类型:



图2.3.9账户类型实体

平台管理人员:



图2.3.10平台管理人员实体

商品种类:



图2.3.11商品种类实体

商品种类属性:



图2.3.12商品种类属性实体

商品库存单位:



图2.3.13商品库存单位实体

平台账户:



图2.3.14平台账户实体

### 2.3.2实体之间的关系

一个买家能拥有多个收货地址:



图2.3.15买家与收货地址关系

一个买家能拥有多个订单:



图2.3.16买家与订单关系

一个订单中可包含多个商品库存单元，一个商品库存单元可以被多个订单所包含:



图2.3.17订单与商品库存单元关系

一个买家能浏览多个商品，一个商品能被多个买家浏览:



图2.3.18买家与商品关系

一个买家能选购多个商品库存单元，一个商品库存单元能被多个买家选购:



图2.3.19买家与商品库存单元关系

一个买家能收藏多个商品，一个商品能被多个买家收藏:



图2.3.20买家与商品关系

一个买家能关注多个店铺，一个店铺能被多个买家关注:



图2.3.21买家与店铺关系

平台管理员能审核店铺发布的商品:



图2.3.22平台管理员与商品关系

每种账户类型下可以多个平台收付款账户以及店铺收款账户:



图2.3.23账户与账户类型关系

店铺能下架商品:



图2.3.24店铺与商品关系

一个店铺能有多个商品:



图2.3.25店铺与商品关系

一个商品能有多个库存单元:



图2.3.26商品商品库存单元关系

一个商品种类下能有多个商品:

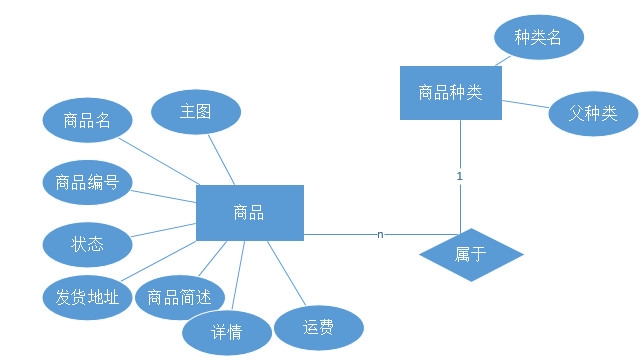


图2.3.27商品与商品种类关系

一个商品种类下能有多个种类属性:



图2.3.28商品种类与种类属性关系

一个商品能有多个种类属性，一个种类属性下能属于多个商品种类:



图2.3.29买家与商品关系

一个店铺能有多个收款账户:



图2.3.30店铺与商家收款账户关系

一个店铺能有多个营业执照:



图2.3.31店铺与营业执照关系

一个商家只能有一家店铺:

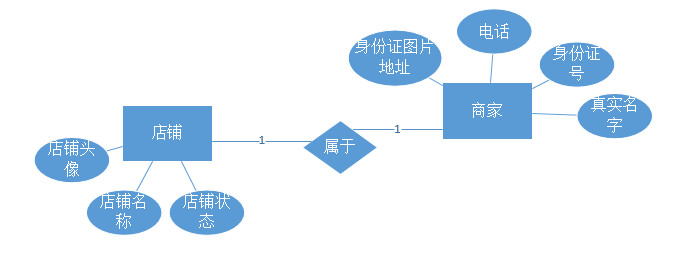


图2.3.32店铺与商家关系

一个买家能对一次订单细节进行一次评论:



图2.3.33买家与订单细节关系

### 2.3.3整体实体关系图



图2.3.34系统整体实体关系图

## 2.4非功能需求

1、安全性：要求系统对不同的用户的权限加以控制。

2、可靠性：要求系统在对用户的请求进行处理过程中可能出现的各种异常情况应该有对应的处理方式。不应该使得系统无响应或者导致系统崩溃。

3、响应时间：系统应该尽可能快的响应请求。

## 2.5其他需求

### 2.5.1开发环境需求

表2.5.1 开发环境需求表

|  |  |
| --- | --- |
| 开发环境 |  |
| 操作系统 | Windows10/64bit |
| 开发语言 | Java、TypeScript、HTML、CSS |
| 开发工具 | IntelliJ IDEA、Visual Studio Code、JDK11.0、MySQL5.5、Redis3.0、Node、Ionic脚手架、Spring Boot脚手架 |

Ionic是一款HTML5混合移动web开发框架[1]，它是以Angular[5]为基础的一款网页组件库，它提供一些常用的开发组件，提高开发效率。

Spring Boot: Spring Boot是一款用于快速开发服务端的轻量级一栈式框架。

Mybatis:是一款持久层解决方案，半自动ORM框架，使得我们的SQL编写更加灵活。

MySQL:是一款关系型数据库管理系统，通过它可以便捷的管理维护我们的数据。

Redis:是非关系型的键值型数据库，用于满足现在数据碎片化、临时化存储的需求。

### 2.5.2通信协议需求

表2.5.2 通信协议需求表

|  |  |
| --- | --- |
| 通信协议 |  |
| 移动端与服务端网络通讯协议 | HTTP协议、WebSocket协议 |
| MySQL数据库与应用之间通信协议 | TCP/IP协议、Unix Socket协议 |
| 邮件服务协议 | SMIP协议 |

# 第3章 系统概要设计

## 3.1功能模块设计

根据系统的需求分析，对系统的移动端与服务端分别进行模块划分。移动端的主要职责是与用户进行交互，以及与服务端进行通信。服务端的主要职责是对移动端发送的请求进行处理，并给予响应。两者共同协作完成用户任务。

对移动端模块的划分如下，分为平台移动端、非平台移动端。平台移动端包含的模块就是平台管理模块。而非平台移动端又分为：买家模块与商家模块。平台管理模块分为三个子模块：平台模块、消息模块、商品模块。买家模块包括四个子模块：首页模块、消息模块、购物车模块、个人中心模块。商家模块包含三个子模块：店铺模块、消息模块、平台服务模块。当然有些我们自定义的公共组件会在三个大模块中都会使用到，我们将它抽出来作为一个自定义公共模块，加上三个模块都需要使用的第三方依赖模块，我们将这部分整体作为公共模块。结构如下：



图3.1.1 移动端功能结构图

由于移动端主要是交互页面，就不再对其模块更细节划分进行展示，下面对服务端功能模块进行划分。

服务端主要分为三类模块：一类是业务模块，该类模块主要是完成系统的业务逻辑；一类是文件管理模块，主要是用于图片管理，业务模块常常依赖于它；最后一类模块是自定义公共模块，该模块用于定义业务模块都需要使用业务相关的实体类、工具类等等，业务模块也依赖于它。而业务模块又分为5个模块，分别是买家模块、商家模块、订单模块、消息模块、平台模块。



图3.1.2 服务端功能结构图

买家模块主要有登录、注册、密码找回、首页商品推荐、个人信息管理、收货地址管理、购物车管理、商品收藏管理、店铺关注管理、浏览历史管理。如下图3.1.3。



图3.1.3 买家模块功能结构图

商家模块主要有入驻、店铺信息管理、店铺商品管理、上传执照、商品分词搜索、当天商品搜索热词统计、店铺收款账户管理、商品种类管理、商品审核。如下图3.1.4。



图3.1.4 商家模块功能结构图

订单模块主要有商品下单、订单定时取消、订单手动取消、订单付款、订单收货地址修改、订单发货、订单确认收货、个人订单评论管理、店铺订单检索。如下图3.1.5。



图3.1.5 订单模块功能结构图

消息模块主要有系统消息通知、消息交互。如下图3.1.6。



图3.1.6 消息模块功能结构图

平台模块主要有支付方式管理、平台收付款账户管理、轮播图服务。如下图3.1.7。



图3.1.6 平台管理模块功能结构图

## 3.2系统整体结构

由功能模块划分，结合系统实现所需相关技术，得到系统的整体结构如下图。

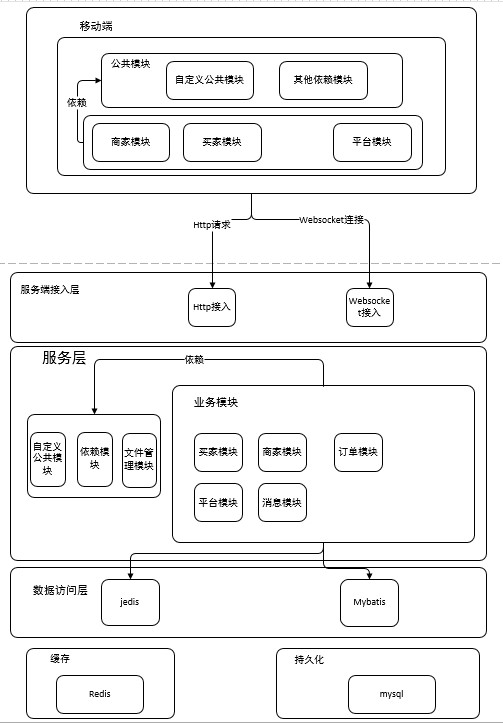


图3.2.1 系统整体结构图

## 3.3数据库概要设计

结合系统的需求分析，将数据库的概念数据模型-实体关系图转换为逻辑数据模型[7]-关系模式[7]。

### 3.3.1实体转换为关系模式

表3.3.1 买家关系模式

|  |  |
| --- | --- |
| 实体名 | 买家 |
| 关系模式 | (头像，密码，昵称，邮箱，性别，电话) |
| 主键 | 电话 |
| 外键 | 无 |

表3.3.2 收货地址关系模式

|  |  |
| --- | --- |
| 实体名 | 收货地址 |
| 关系模式 | (地址编号，买家手机号，地址，收货人，收货人电话) |
| 主键 | 地址编号 |
| 外键 | 买家手机号 |

表3.3.3 商品关系模式

|  |  |
| --- | --- |
| 实体名 | 商品 |
| 关系模式 | (商品编号，店铺编号，种类编号，商品名，商品简述，发货地址，主图，状态，运费，详情) |
| 主键 | 商品编号 |
| 外键 | 店铺编号，种类编号 |

表3.3.4 订单关系模式

|  |  |
| --- | --- |
| 实体名 | 订单 |
| 关系模式 | (订单编号，买家手机号，下单日期，订单总计金额) |
| 主键 | 订单编号 |
| 外键 | 买家手机号 |

表3.3.5 商品种类关系模式

|  |  |
| --- | --- |
| 实体名 | 商品种类 |
| 关系模式 | (种类编号，种类名，父种类编号) |
| 主键 | 种类编号 |
| 外键 | 父种类编号 |

表3.3.6 种类属性关系模式

|  |  |
| --- | --- |
| 实体名 | 种类属性 |
| 关系模式 | (属性编号，种类编号，属性名) |
| 主键 | 属性编号 |
| 外键 | 种类编号 |

表3.3.7 商品库存单位关系模式

|  |  |
| --- | --- |
| 实体名 | 商品库存单元 |
| 关系模式 | (库存单元编号，商品编号，规格，价格，数量，附图) |
| 主键 | 库存单元编号 |
| 外键 | 商品编号 |

表3.3.8 商家关系模式

|  |  |
| --- | --- |
| 实体名 | 商家 |
| 关系模式 | (身份证号，真实名字，电话，身份证图片地址) |
| 主键 | 身份证号 |
| 外键 | 无 |

表3.3.9 店铺关系模式

|  |  |
| --- | --- |
| 实体名 | 店铺 |
| 关系模式 | (店铺编号，商家身份证号，店铺头像，店铺名称，店铺状态) |
| 主键 | 店铺编号 |
| 外键 | 商家身份证号 |

表3.3.10 商家收款账户关系模式

|  |  |
| --- | --- |
| 实体名 | 商家收款账户 |
| 关系模式 | (账户编号，店铺编号，账户类别编号，账户号，是否用于收款) |
| 主键 | 账户编号 |
| 外键 | 店铺编号，账户类别编号 |

表3.3.11 账户关系模式

|  |  |
| --- | --- |
| 实体名 | 账户类型 |
| 关系模式 | (账户类型编号，类型名，是否可用) |
| 主键 | 账户类型编号 |
| 外键 | 无 |

表3.3.12 营业执照关系模式

|  |  |
| --- | --- |
| 实体名 | 营业执照 |
| 关系模式 | (执照编号，店铺编号，执照图片地址) |
| 主键 | 执照编号 |
| 外键 | 店铺编号 |

表3.3.13 平台管理人员关系模式

|  |  |
| --- | --- |
| 实体名 | 平台管理人员 |
| 关系模式 | (工号，姓名，电话，邮箱) |
| 主键 | 工号 |
| 外键 | 无 |

表3.3.14 平台收付款账户关系模式

|  |  |
| --- | --- |
| 实体名 | 平台账户 |
| 关系模式 | (账户编号，账户类别编号，账户号，密钥，是否可用) |
| 主键 | 账户编号 |
| 外键 | 账户类别编号 |

### 3.3.2实体关系转换为关系模式

表3.3.15 评论关系模式

|  |  |
| --- | --- |
| 关系名 | 评论 |
| 关系模式 | (评论编号，买家电话，订单编号，订单子编号，评论内容，评论时间附图) |
| 主键 | 评论编号 |
| 外键 | 买家电话，（订单编号，订单子编号） |

表3.3.16 购物车关系模式

|  |  |
| --- | --- |
| 关系名 | 购物车 |
| 关系模式 | (购物车记录编号，买家电话，商品库存单元编号，商品数目，总计费用) |
| 主键 | 购物车记录编号 |
| 外键 | 买家电话，商品库存单元编号 |

表3.3.17 订单细节关系模式

|  |  |
| --- | --- |
| 关系名 | 订单细节 |
| 关系模式 | (订单编号，订单子编号，商品库存单元编号，订单细节状态，收货地址信息，购买数目，总计费用，物流运单号，结束日期) |
| 主键 | (订单编号，订单子编号) |
| 外键 | 订单编号，商品库存单元编号 |

表3.3.18 商品属性值关系模式

|  |  |
| --- | --- |
| 关系名 | 商品属性值 |
| 关系模式 | (商品编号，种类属性编号，属性值) |
| 主键 | (商品编号，种类属性编号) |
| 外键 | 商品编号，种类属性编号 |

表3.3.19 商品下架信息关系模式

|  |  |
| --- | --- |
| 关系名 | 商品下架信息 |
| 关系模式 | (商品下架信息编号，商品编号，下架时间，下架备注) |
| 主键 | 商品下架信息编号 |
| 外键 | 商品编号 |

表3.3.20 商品审核信息关系模式

|  |  |
| --- | --- |
| 关系名 | 商品审核信息 |
| 关系模式 | (商品审核信息编号、商品编号，平台管理人员工号，审核状态，审核备注，申请日期，审核日期) |
| 主键 | 商品审核信息编号 |
| 外键 | 商品编号，平台管理人员工号 |

表3.3.21 关注店铺信息关系模式

|  |  |
| --- | --- |
| 关系名 | 关注店铺信息 |
| 关系模式 | (买家电话、店铺编号，关注日期) |
| 主键 | (买家电话、店铺编号) |
| 外键 | 买家电话、店铺编号 |

表3.3.22 收藏商品信息关系模式

|  |  |
| --- | --- |
| 关系名 | 收藏商品信息 |
| 关系模式 | (买家电话、商品编号，收藏日期) |
| 主键 | (买家电话、商品编号) |
| 外键 | 买家电话、商品编号 |

表3.3.23 浏览商品信息关系模式

|  |  |
| --- | --- |
| 关系名 | 浏览商品信息 |
| 关系模式 | (买家电话、商品编号，浏览日期，浏览时长) |
| 主键 | (买家电话、商品编号) |
| 外键 | 买家电话、商品编号 |

# 第4章 系统详细设计

由于移动端主要是页面设计，这部分就不对其进行详细设计阐述，只在系统实现中展示对应的设计成果。这部分主要对系统服务端的主要模块中相对复杂的功能进行详细设计，并使用程序流程图[15]进行描述。

## 4.1文件管理模块详细设计

### 4.1.1文件上传

设计思路：

提供单文件上传与批量文件上传接口，其他文件只需要调用接口即可。具体如下。

程序流程图：



图4.1.1 文件上传程序流程图

### 4.1.2文件删除

设计思路：

提供文件删除接口给其他模块调用，其他模块需要指定要删除的文件名即可。

程序流程图：



图4.1.2 文件删除程序流程图

## 4.2买家模块详细设计

### 4.2.1买家注册设计

设计思路：

买家填写注册信息，系统验证信息是否已存在，不存在就保存信息，然后返回注册成功信息，存在就返回注册失败。

程序流程图：



图4.2.1 买家注册程序流程图

### 4.2.2买家登录设计

设计思路：

买家填写登录信息，系统验证信息是否存在，存在就生成一个JWT（以后在JWT有效时间内不需要执行登录操作）并查询买家必要展示信息，然后返回到移动端缓存起来；如果不存在就返回验证失败。

程序流程图：



图4.2.2 买家登录程序流程图

### 4.2.3买家密码找回设计

设计思路：

买家填写已注册手机号，然后系统向对应注册时绑定的邮箱发送验证码，为保证安全性，服务端需要将验证码存起来。然后买家填写新密码，系统验证验证码，通过就修改，返回修改结果。

程序流程图：



图4.2.3 买家密码找回程序流程图

### 4.2.4买家首页商品推荐设计

设计思路：判断买家是否登录，没有就在当前在售商品的种类中随机选取5种，将在这5种类的商品按照好评率进行排序后返回。系统获取到买家未删除的浏览历史按浏览时间、浏览时长进行降序排列，然后按照浏览时间进行分组，每组取前1/3，已删除的历史记录同样如此。然后获取买家商品收藏信息。最后获取买家订单购买的商品信息。综合三者的商品种类，如果种类数量大于5，将在这些种类的商品按照好评率进行排序后返回。如果小于5，在当前在售商品的种类中随机选取补充到5种，将在这5种类的商品按照好评率进行排序后返回。

程序流程图：



图4.2.4 买家首页商品推荐程序流程图

## 4.3商家模块详细设计

### 4.3.1商家入驻设计

设计思路：

买家填写实名信息以及身份证人面照，系统验证实名信息是否存在，存在就返回入驻失败。不存在，系统调用阿里云实名认证接口，检测认证是否成功，成功就保存实名信息，并初始化店铺信息，然后返回入驻成功。

程序流程图：



图4.3.1商家入驻程序流程图

### 4.3.2商品分词搜索设计

设计思路：

买家填写要搜索的信息。系统将搜索信息进行分词，然后提炼分词中的种类、品牌。系统将商品名称与分词进行匹配。并将匹配结果进行权重计算，然后按照权重进行排序。最后如果分词中含有种类、品牌，就将所属种类所属品牌的商品追加到后面。

程序流程图：



图4.3.2商品分词搜索程序流程图

### 4.3.3当天商品搜索热词统计设计

设计思路：

在商品分词搜索的基础商，将权重最大的结果所匹配的分词组合起来，作为本次实际搜索词，检测该搜索词是否存在搜索词集合，在就搜索次数加一，不在就加入搜索词集合。

程序流程图：



图4.3.3当天商品搜索热词统计程序流程图

## 4.4订单模块详细设计

### 4.4.1商品下单设计

设计思路：

买家选取商品库存单元，系统进行商品库存单元数量减少。减少失败，则下单失败。如果减少成功，计算订单总金额并拆分订单，然后保存订单以及订单细节信息。

程序流程图：



图4.4.1 商品下单程序流程图

### 4.4.2订单定时自动取消设计

设计思路：

买家在下单成功后在支付界面取消付款后，所有订单细节进入10分钟倒计时，在10分钟内没有完成付款的订单细节直接删除，其对应商品库存单元增加对应的数量。

程序流程图：



图4.4.2 订单定时取消程序流程图

### 4.4.3订单手动取消设计

设计思路：

买家发起订单取消后，系统先发起订单退款，如果发起成功，就增加对应商品库存单元的数量，然后清除订单细节，返回取消成功。否则返回取消失败。



图4.4.3 订单手动取消程序流程图

### 4.4.4订单付款设计

设计思路：

买家发起订单支付，系统计算要支付的总金额，封装支付请求参数，调用支付接口拉取支付信息，移动端展示支付界面，买家确认支付，支付接口调用回调接口修改订单状态并保存支付信息，移动端循环检测订单细节状态直到改变。

程序流程图：



图4.4.4 订单付款程序流程图

### 4.4.5订单确认收货设计

设计思路：

买家发起订单确认收货，系统获取订单信息，系统发起转账，如果发起成功，修改订单状态，返回确认收货成功信息。否则返回失败。转账成功后保存转账信息，转账失败重新发起。

程序流程图：



图4.4.5 订单确认收货程序流程图

## 4.5消息模块详细设计

### 4.5.1消息交互设计

设计思路：

买家/商家填写消息，系统保存要发送的消息，然后检测当前websocket连接在线的商家/买家，如果接收方在线，就将消息推送过去。

程序流程图：



图4.5.1 消息交互程序流程图

## 4.6平台模块详细设计

### 4.6.1轮播图服务设计

设计思路：

平台管理员配置轮播图工厂信息，系统启动工厂，工厂定时生产，工厂发放，商家抢取，是否抢取完成，是在检测工厂是否完成所有生产任务，如果是就关闭工厂，如果没有生产完成继续生产。如果还没有完成抢取，就等待抢取完成。

程序流程图：



图4.6.1 轮播图服务程序流程图

## 4.7数据库详细设计

将概要设计阶段中数据库的逻辑数据模型转换为MySQL中的物理表。

表4.7.1 买家表(consumer)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段说明 | 默认值 | NULL | 主键 | 外键 | 唯一 | 长度 |
| consumer\_id | bigint | 编号 | 无 | 0 | 1 | 0 | 1 | 11 |
| consumer\_head | varchar | 头像 | Null | 1 | 0 | 0 | 0 | 255 |
| consumer\_nickname | varchar | 昵称 | 无 | 0 | 0 | 0 | 0 | 6 |
| consumer\_phone | bigint | 电话 | 无 | 0 | 0 | 0 | 1 | 20 |
| consumer\_email | varchar | 邮箱 | 无 | 0 | 0 | 0 | 0 | 255 |
| consumer\_sex | bit | 性别 | 0 | 0 | 0 | 0 | 0 | 1 |
| consumer\_password | varchar | 密码 | 无 | 0 | 0 | 0 | 0 | 255 |
| is\_delete | bit | 是否被删除 | 0 | 0 | 0 | 0 | 0 | 1 |

表4.7.2 购物车表(cart)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段说明 | 默认值 | NULL | 主键 | 外键 | 唯一 | 长度 |
| cart\_id | bigint | 编号 | 无 | 0 | 1 | 0 | 1 | 20 |
| commodity\_id | varchar | 商品编号 | 无 | 0 | 0 | 1 | 0 | 255 |
| consumer\_phone | bigint | 买家电话 | 无 | 0 | 0 | 0 | 0 | 20 |
| commodity\_name | varchar | 商品名 | 无 | 0 | 0 | 1 | 0 | 255 |
| commodity\_num | int | 商品数目 | 1 | 0 | 0 | 0 | 0 | 11 |
| sku\_map | varchar | 商品规格附图 | 无 | 0 | 0 | 0 | 0 | 255 |
| sku\_value | varchar | 商品规格 | 无 | 0 | 0 | 0 | 0 | 255 |
| total\_fee | decimal | 总计费用 | 无 | 0 | 0 | 0 | 0 | 10 |

表4.7.3 收货地址表(address)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段说明 | 默认值 | NULL | 主键 | 外键 | 唯一 | 长度 |
| address\_id | varchar | 地址编号 | 无 | 0 | 1 | 0 | 1 | 11 |
| consumer\_phone | bigint | 所属买家电话 | 无 | 0 | 0 | 0 | 1 | 11 |
| address | varchar | 具体地址 | 无 | 0 | 0 | 0 | 0 | 255 |
| take\_name | varchar | 收货人名 | 无 | 0 | 0 | 0 | 0 | 255 |
| take\_phone | bigint | 收货电话 | 无 | 0 | 0 | 0 | 0 | 20 |
| is\_default | bit | 是否默认收货地址 | 1 | 0 | 0 | 0 | 0 | 1 |

表4.7.4 订单表(order)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段说明 | 默认值 | NULL | 主键 | 外键 | 唯一 | 长度 |
| order\_number | varchar | 订单编号 | 无 | 0 | 1 | 0 | 1 | 255 |
| consumer\_phone | bigint | 买家电话 | 无 | 0 | 0 | 0 | 1 | 11 |
| order\_date | datetime | 下单日期 | 无 | 0 | 0 | 0 | 0 | 0 |
| total\_amount | decimal | 共计费用 | 无 | 0 | 0 | 0 | 0 | 10 |

表4.7.5 订单细节表(orderDetail)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段说明 | 默认值 | NULL | 主键 | 外  键 | 唯一 | 长度 |
| od\_id | int | 订单细节编号 | 无 | 0 | 1 | 0 | 0 | 11 |
| order\_number | varchar | 订单编号 | 无 | 0 | 1 | 1 | 0 | 255 |
| commodity\_id | bigint | 商品编号 | 无 | 0 | 0 | 1 | 1 | 11 |
| sku\_map | varchar | 商品规格附图 | 无 | 0 | 0 | 0 | 0 | 255 |
| sku\_value | varchar | 商品规格 | 无 | 0 | 0 | 0 | 0 | 255 |
| total\_amount | decimal | 总计费用 | 无 | 0 | 0 | 0 | 0 | 10 |
| address\_msg | varchar | 收货地址信息 | 无 | 0 | 0 | 0 | 0 | 255 |
| state | tinyint | 订单状态 | 无 | 0 | 0 | 0 | 0 | 4 |
| commodity\_num | int | 商品数目 | 无 | 0 | 0 | 0 | 0 | 11 |
| logistics\_no | varchar | 物流运单号 | Null | 1 | 0 | 0 | 1 | 255 |
| over\_date | datetime | 结束日期 | 无 | 0 | 0 | 0 | 0 | 11 |

表4.7.6 买家付款信息表(cpi)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段说明 | 默认值 | NULL | 主键 | 外  键 | 唯一 | 长度 |
| cpi\_id | varchar | 支付信息编号 | 无 | 0 | 1 | 0 | 1 | 255 |
| acca\_id | int | 支付类别编号 | 无 | 0 | 0 | 1 | 0 | 11 |
| plant\_account | varchar | 平台收款账户 | 无 | 0 | 0 | 0 | 0 | 255 |
| pay\_no | varchar | 第三方支付单号 | 无 | 0 | 0 | 0 | 0 | 255 |
| pay\_money | decimal | 付款金额 | 无 | 0 | 0 | 0 | 0 | 10 |
| pay\_date | datetime | 付款日期 | 无 | 0 | 0 | 0 | 0 | 0 |

表4.7.7 商家收款信息表(pts)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段说明 | 默认值 | NULL | 主键 | 外键 | 唯一 | 长度 |
| pts\_id | varchar | 收款信息编号 | 无 | 0 | 1 | 0 | 1 | 255 |
| acca\_name | varchar | 转账账户类型 | 无 | 0 | 0 | 0 | 0 | 255 |
| trade\_no | varchar | 第三方转账单号 | 无 | 0 | 0 | 0 | 0 | 255 |
| trade\_money | decimal | 转账金额 | 无 | 0 | 0 | 0 | 0 | 10 |
| trade\_date | datetime | 转账日期 | 无 | 0 | 0 | 0 | 0 | 0 |
| is\_trade | bit | 转账成功 | 无 | 0 | 0 | 0 | 0 | 1 |

表4.7.8 平台退款信息表(pbi)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段说明 | 默认值 | NULL | 主键 | 外键 | 唯一 | 长度 |
| pbi\_id | varchar | 退款信息编号 | 0 | 0 | 0 | 1 | 1 | 255 |
| acca\_name | varchar | 转账账户类型 | 0 | 0 | 0 | 0 | 0 | 255 |
| plant\_account | varchar | 平台退款账户 | 0 | 0 | 0 | 0 | 0 | 255 |
| back\_no | varchar | 第三方退款单号 | 0 | 0 | 0 | 0 | 0 | 255 |
| back\_money | decimal | 退款金额 | 0 | 0 | 0 | 0 | 0 | 10 |
| back\_date | datetime | 退款日期 | 0 | 0 | 0 | 0 | 0 | 0 |
| is\_back | bit | 退款成功 | 0 | 0 | 0 | 0 | 0 | 1 |

表4.7.9 账户类型表(account\_category)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段说明 | 默认值 | NULL | 主键 | 外键 | 唯一 | 长度 |
| acca\_id | int | 类型编号 | 无 | 0 | 1 | 0 | 0 | 11 |
| acca\_name | varchar | 类型名称 | 无 | 0 | 0 | 0 | 0 | 255 |
| is\_open | bit | 是否使用 | 无 | 0 | 0 | 0 | 0 | 1 |

表4.7.10 平台账户表(plant\_account)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段说明 | 默认值 | NULL | 主键 | 外键 | 唯一 | 长度 |
| pa\_id | int | 编号 | 无 | 0 | 1 | 0 | 0 | 11 |
| acca\_id | int | 类型编号 | 无 | 0 | 0 | 0 | 0 | 11 |
| account\_num | varchar | 账户号 | 无 | 0 | 0 | 0 | 0 | 255 |
| account\_key | varchar | 账户密钥 | 无 | 0 | 0 | 0 | 0 | 255 |
| is\_usable | bit | 是否启用 | 无 | 0 | 0 | 0 | 0 | 1 |

表4.7.11 平台管理人员表(staff)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段说明 | 默认值 | NULL | 主键 | 外键 | 唯一 | 长度 |
| satff\_id | bigint | 账号 | 无 | 0 | 1 | 0 | 1 | 20 |
| satff\_mail | varchar | 邮箱 | 无 | 0 | 0 | 0 | 0 | 255 |
| staff\_name | varchar | 姓名 | 无 | 0 | 0 | 0 | 0 | 255 |
| satff\_phone | bigint | 电话 | 无 | 0 | 0 | 0 | 0 | 20 |

表4.7.12 商家账户表(account\_seller)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段说明 | 默认值 | NULL | 主键 | 外键 | 唯一 | 长度 |
| account\_id | int | 编号 | 无 | 0 | 1 | 0 | 1 | 11 |
| store\_id | bigint | 店铺编号 | 无 | 0 | 0 | 1 | 0 | 20 |
| account\_num | varchar | 账户号 | 无 | 0 | 0 | 0 | 0 | 255 |
| acca\_id | int | 类别编号 | 无 | 0 | 0 | 1 | 0 | 255 |
| is\_collections | bit | 用于收款 | 无 | 0 | 0 | 0 | 0 | 1 |

表4.7.13 商品属性表(attribute)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段说明 | 默认值 | NULL | 主键 | 外键 | 唯一 | 长度 |
| attribute\_id | bigint | 编号 | 无 | 0 | 1 | 0 | 1 | 20 |
| category\_id | int | 种类编号 | 无 | 0 | 0 | 1 | 0 | 11 |
| attribute\_name | varchar | 属性名 | 无 | 0 | 0 | 0 | 0 | 255 |
| is\_mul | bit | 是否有多个值 | 无 | 0 | 0 | 0 | 0 | 1 |

表4.7.14 商品种类表(category)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段说明 | 默认值 | NULL | 主键 | 外  键 | 唯一 | 长度 |
| category\_id | int | 编号 | 0 | 0 | -1 | 0 | 0 | 11 |
| category\_name | varchar | 种类名 | 0 | 0 | 0 | 0 | 0 | 255 |
| father | int | 父种类编号 | 0 | -1 | 0 | 0 | 0 | 11 |

表4.7.15 商品属性值表(cav)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段说明 | 默认值 | NULL | 主键 | 外  键 | 唯一 | 长度 |
| cav\_id | bigint | 编号 | 无 | 0 | 1 | 0 | 1 | 20 |
| commodity\_id | varchar | 商品编号 | 无 | 0 | 0 | 1 | 0 | 255 |
| attribute\_name | varchar | 属性名 | 无 | 0 | 0 | 0 | 0 | 255 |
| attribute\_value | varchar | 属性值 | 无 | 0 | 0 | 0 | 0 | 255 |

表4.7.16 订单评论表(comment)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段说明 | 默认值 | NULL | 主键 | 外  键 | 唯一 | 长度 |
| comment\_no | bigint | 编号 | 无 | 0 | 1 | 0 | 1 | 20 |
| od\_id | int | 订单细节编号 | 无 | 0 | 1 | 0 | 0 | 11 |
| order\_number | varchar | 订单编号 | 无 | 0 | 1 | 1 | 0 | 255 |
| commodity\_id | varchar | 商品编号 | 无 | 0 | 0 | 1 | 0 | 255 |
| consumer\_phone | bigint | 买家电话 | 无 | 0 | 0 | 0 | 0 | 20 |
| comment\_content | varchar | 评论内容 | 无 | 0 | 0 | 0 | 0 | 255 |
| attched\_map | varchar | 附图 | NULL | 1 | 0 | 0 | 0 | 255 |
| star\_level | int | 评论星级 | 0 | 0 | 0 | 0 | 0 | 11 |
| is\_anonymous | bit | 是否匿名 | 0 | 0 | 0 | 0 | 0 | 1 |
| start\_date | date | 发布日期 | 0 | 0 | 0 | 0 | 0 | 0 |

表4.7.17商品表(commodity)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段说明 | 默认值 | NULL | 主键 | 外  键 | 唯一 | 长度 |
| commodity\_id | varchar | 编号 | 无 | 0 | 1 | 0 | 1 | 255 |
| store\_id | varchar | 店铺编号 | 无 | 0 | 0 | 1 | 0 | 255 |
| category\_id | int | 种类编号 | 无 | 0 | 0 | 1 | 0 | 11 |
| commodity\_name | varchar | 商品名 | 无 | 0 | 0 | 0 | 0 | 255 |
| master\_map | varchar | 主图 | 无 | 0 | 0 | 0 | 0 | 255 |
| commodity\_describe | varchar | 商品简述 | Null | 1 | 0 | 0 | 0 | 255 |
| address | varchar | 发货地址 | 无 | 0 | 0 | 0 | 0 | 255 |
| shipping | decimal | 运费 | 0 | 0 | 0 | 0 | 0 | 10 |
| detail | varchar | 详情 | 0 | 1 | 0 | 0 | 0 | 255 |
| status\_code | tinyint | 状态 | 0 | 0 | 0 | 0 | 0 | 4 |

表4.7.18 商品库存单位表(sku)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段说明 | 默认值 | NULL | 主键 | 外  键 | 唯一 | 长度 |
| sku\_id | bigint | 库存编号 | 无 | 0 | 1 | 0 | 1 | 20 |
| commodity\_id | varchar | 商品编号 | 无 | 0 | 0 | 1 | 0 | 255 |
| sku\_value | varchar | 规格 | 无 | 0 | 0 | 0 | 0 | 255 |
| sku\_map | varchar | 附图 | 无 | 0 | 0 | 0 | 0 | 255 |
| sku\_price | decimal | 价格 | 无 | 0 | 0 | 0 | 0 | 10 |
| num | bigint | 数量 | 无 | 0 | 0 | 0 | 0 | 20 |

表4.7.19店铺执照表(license)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段说明 | 默认值 | NULL | 主键 | 外  键 | 唯一 | 长度 |
| license\_id | bigint | 编号 | 无 | 0 | 1 | 0 | 1 | 20 |
| store\_id | varchar | 店铺编号 | 无 | 0 | 0 | 1 | 0 | 255 |
| license\_url | varchar | 执照图片 | 无 | 0 | 0 | 0 | 0 | 255 |

表4.7.20商家表(seller)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段说明 | 默认值 | NULL | 主键 | 外  键 | 唯一 | 长度 |
| seller\_id | varchar | 身份证号 | 无 | 0 | 1 | 0 | 1 | 255 |
| seller\_name | varchar | 姓名 | 无 | 0 | 0 | 0 | 0 | 10 |
| seller\_phone | bigint | 电话 | 无 | 0 | 0 | 0 | 0 | 20 |
| seller\_path | varchar | 身份证图片 | 无 | 0 | 0 | 0 | 0 | 255 |

表4.7.21店铺表(store)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段说明 | 默认值 | NULL | 主键 | 外  键 | 唯一 | 长度 |
| store\_id | varchar | 编号 | 无 | 0 | 1 | 0 | 1 | 255 |
| seller\_id | varchar | 商家身份证号 | 无 | 0 | 0 | 1 | 1 | 255 |
| store\_name | varchar | 店铺名 | 无 | 0 | 0 | 0 | 0 | 255 |
| store\_head | varchar | 店铺头像 | 无 | 0 | 0 | 0 | 0 | 255 |
| status | Bit | 状态 | 无 | 0 | 0 | 0 | 0 | 1 |

表4.7.22商品审核信息表(check\_info)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段说明 | 默认值 | NULL | 主键 | 外  键 | 唯一 | 长度 |
| commodity\_id | varchar | 商品编号 | 0 | 0 | 1 | 1 | 1 | 255 |
| apply\_date | datetime | 申请日期 | 0 | 0 | 0 | 0 | 0 | 0 |
| check\_date | datetime | 审核日期 | 0 | 1 | 0 | 0 | 0 | 0 |
| staff\_id | bigint | 审核员工工号 | 0 | 1 | 0 | 0 | 0 | 20 |
| check\_code | int | 审核状态 | 0 | 0 | 0 | 0 | 0 | 11 |
| remark | varchar | 审核备注 | 0 | 1 | 0 | 0 | 0 | 255 |

表4.7.23商品下架信息表(down\_info)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段说明 | 默认值 | NULL | 主键 | 外  键 | 唯一 | 长度 |
| commodity\_id | varchar | 商品编号 | 0 | 0 | 1 | 1 | 1 | 255 |
| down\_date | date | 下架日期 | 0 | 0 | 0 | 0 | 0 | 0 |
| down\_des | varchar | 下架原因 | 0 | 0 | 0 | 0 | 0 | 255 |

表4.7.24店铺关注表(attention)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段说明 | 默认值 | NULL | 主键 | 外  键 | 唯一 | 长度 |
| consumer\_phone | bigint | 买家电话 | 0 | 0 | 1 | 1 | 1 | 20 |
| store\_id | varchar | 店铺编号 | 0 | 0 | 1 | 1 | 1 | 255 |
| start\_date | datetime | 关注 日期 | 0 | 0 | 0 | 0 | 0 | 0 |

表4.7.25商品收藏表(collection)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段说明 | 默认值 | NULL | 主键 | 外  键 | 唯一 | 长度 |
| consumer\_phone | bigint | 买家电话 | 0 | 0 | 1 | 1 | 1 | 20 |
| commodity\_id | varchar | 商品编号 | 0 | 0 | 1 | 1 | 0 | 255 |
| start\_date | datetime | 收藏日期 | 0 | 0 | 0 | 0 | 0 | 0 |

表4.7.26浏览历史表(his)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段说明 | 默认值 | NULL | 主键 | 外  键 | 唯一 | 长度 |
| his\_id | varchar | 浏览历史编号 | 0 | 0 | 1 | 0 | 1 | 255 |
| consumer\_phone | bigint | 买家电话 | 0 | 0 | 0 | 1 | 0 | 20 |
| commodity\_id | varchar | 商品编号 | 0 | 0 | 0 | 1 | 0 | 255 |
| start\_date | datetime | 浏览日期 | 0 | 0 | 0 | 0 | 0 | 0 |
| look\_time | int | 浏览时长 | 0 | 0 | 0 | 0 | 0 | 11 |
| is\_delete | int | 11 | 0 | 0 | 0 | 0 | 0 | 11 |

表4.7.27消息交互信息表(msg)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 字段说明 | 默认值 | NULL | 主键 | 外  键 | 唯一 | 长度 |
| msg\_id | varchar | 信息编号 | 0 | 0 | 1 | 0 | 1 | 255 |
| sender | bigint | 发送者 | 0 | 0 | 0 | 1 | 0 | 20 |
| receiver | bigint | 接收者 | 0 | 0 | 0 | 1 | 0 | 20 |
| sender\_role | int | 发送角色 | 0 | 0 | 0 | 0 | 0 | 11 |
| receiver\_role | int | 接收角色 | 0 | 0 | 0 | 0 | 0 | 11 |
| msg | varchar | 信息内容 | 0 | 0 | 0 | 0 | 0 | 255 |
| msg\_type | varchar | 信息类型 | 0 | 0 | 0 | 0 | 0 | 255 |
| is\_read | int | 是否已读 | 0 | 0 | 0 | 0 | 0 | 11 |
| send\_date | datetime | 发送日期 | 0 | 0 | 0 | 0 | 0 | 0 |
| is\_delete | int | 是否删除 | 0 | 0 | 0 | 0 | 0 | 11 |

# 第5章 系统实现

本系统移动端使用的是Ionic技术实现，实现页面做到简洁、素雅。服务端使用Spring Boot、Mybatis、Redis、MySQL技术实现服务端。移动端请求服务端是使用angular的HttpModule下的HttpClient服务，服务端响应格式的是RpsMsg类（该类包含三个属性，statusCode表示请求执行结果状态码，msg表示请求执行结果携带的信息，data代表请求执行结果的数据）。为了更直观阐述系统实现，本章我们将移动端模块划分作为表述主体，在调用到具体服务端功能时，再进行其实现阐述。

## 5.1系统权限控制实现

### 5.1.1移动端权限控制实现

移动端不同的用户有不同的视图，如没有入驻的买家只能访问登录、注册、密码找回、首页商品、消息、购物车、个人中心、入驻页面等等，已经入驻的买家除此之外还能访问店铺信息、店铺商品管理、轮播图名额等页面。移动端控制视图权限方法是采用Angular路由守卫[5]的方法。Angular有4种路由守卫，canActivate，控制是否允许进入路由。canActivateChild，等同 canActivate，只不过针对是所有子路由。canDeactivat，控制是否允许离开路由。canLoad，控制是否允许延迟加载整个模块。我们使用的是canActivate在路由挂载[5]前检测用户的身份，检测通过则可以访问对应的路由。

我们定义了3个路由守卫，分别是consumer-login、seller-login、plant-login。分别用于验证买家、商家、平台的身份。

### 5.1.2服务端权限控制实现

虽然移动端可以通过路由守卫控制用户对页面的访问，但是用户也可以通过某些工具（如POSTMAN）绕开移动端，直接请求服务端提供的接口。所以服务端同样需要进行权限控制。我们使用Spring的Interceptor功能以及JWT令牌对用户所有请求进行拦截，然后检测请求头部中是否携带JWT[16]身份令牌，以及检测JWT令牌是否有效。如果有效就放行请求，反之就拦截。我们制定了三类JWT：CONSUMERJWT-买家身份,SELLERJWT-商家身份,PLANTJWT-平台管理员的身份。我们还制定了三类Intercepotor，分别是ConsumerInterceptor, StoreInterceptor,PlantInterceptor。三类JWT能标识用户的不同身份，三类Interceptor能对需要不同权限的接口进行请求拦截并进行对应的身份检测。

## 5.2买家中心功能实现

### 5.2.1登录实现

移动端登录界面如下：主要有手机号、密码、验证码三个字段。然后将三个字段封装后通过请求服务端登录接口“/consumer/login”。



服务端登录逻辑实现：其中图片验证码是通过hutool工具的CaptchaUtil类创建，创建成功后保存到session中，接着返回给移动端。登录验证逻辑：首先验证验证码是否有效，如果无效就返回验证失效消息，如果有效就通过ConsumerMapper的selectConsumerByConsumer执行通过手机号、密码进行买家信息查询，查询结果不为空说明买家存在，就使用手机号与头像地址作为负载，使用SHA256加密算法创建数字签名，经过BASE64编码后封装成一个CONSUMERJWT。将这个JWT缓存到移动端，以后买家权限操作都需要带上这个JWT。如果这个JWT失效或者丢失就需要重新登录。

### 5.2.2注册实现

移动端注册界面如下图：其中有手机号、密码、再次确认密码、邮箱、昵称、验证码、性别字段。其中再次确认密码用于移动端验证。将字段封装后通过请求服务端注册接口“/consumer/register”。



服务端注册逻辑实现：首先验证验证码是否有效，如果无效就返回验证失效消息，如果有效就通过ConsumerMapper的selectConsumerByConsumer执行通过手机号查询该手机号是否已经注册过了，如果已经注册过，就返回已注册。如果没有注册就调用ConsumerMapper的insertConsumer保存用户信息，并返回注册成功。移动端在得知已注册或是注册成功的情况下会跳转登录页面。

### 5.2.3密码找回实现

移动端密码找回界面如下图：其中有手机号、新密码、确认密码、邮箱验证码。其中确认密码用于移动端验证。移动端先请求邮箱验证码，填写验证码后，将手机号、新密码、邮箱验证码封装后通过请求服务端注册接口“/consumer/info/backpassword”。



服务端密码找回逻辑实现：首先验证邮箱验证码是否有效，如果无效就返回验证失效消息，如果有效就通过ConsumerMapper的updateConsumer修改密码，然后返回修改结果。

### 5.2.4首页商品推荐实现

移动端首页商品界面如下图：



移动端进入首页时，判断用户是否登录，如果已经登录就携带CONSUMERJWT请求服务端接口“/commodity/consumer/home”。服务端检测请求中是否有CONSUMERJWT，如果没有，就通过的CommodityMapper下的方法：selectCommodityCateIdByCommoditys()查询出所有在售中商品种类，判断种类数组长度，如果小于5，使用ConsumerCommodityMapper下的selectHomeCommoditySimpleOrderByfavorableRateByTypes()查询出指定种类范围的且按照好评率排序的商品信息。如果大于就通过 ThreadLocalRandom.current().nextInt()随机产生商品种类数组下标，直到取足5个不同的商品种类。再使用selectHomeCommoditySimpleOrderByfavorable- RateByTypes()查询出首页商品信息。

如果已经登录就通过HistoryControlle下的consumerHomeRecommendCommo dityId()获取到浏览历史商品id,其具体逻辑是先通过HistoryMapper 下的selectCountDateByRecentDay()分别获取到已删除浏览历史与未删除浏览历史按照浏览时间、浏览时长进行排序后按照浏览时间分组。

## 5.1系统权限控制实现

### 5.1.1移动端权限控制实现

# 第6章 系统测试

本章主要是要对系统的服务端的主要功能模块的接口进行测试。 测试工具：Jmeter、Fiddler。

## 6.1买家模块

表6.1.1登录接口测试用例

|  |  |
| --- | --- |
| 用例编号 | 1-1 |
| 测试目的 | 系统是否能正确验证用户并授予系统通行证 |
| 前置条件 | 1、服务端已启动  2、jmeter创建线程组并创建Http请求以及添加结果查看树 |
| 测试步骤 | 1、填写接口URL：http://localhost:8080/consumer/login  2、填写请求方式：GET  3、添加请求参数：  consumerPhone=15327348480，  consumerPassword=123456  4、发送请求 |
| 预期结果 | 验证通过并返回JWT |
| 实际结果 | {  "stausCode":200,  "msg":"登录成功",  "data":{"jwt":" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"}  } |
| 通过情况 | 符合预期 |

表6.1.2注册接口测试用例

|  |  |
| --- | --- |
| 用例编号 | 1-2 |
| 测试目的 | 系统是否能保存未注册用户信息，并授予系统通行证 |
| 前置条件 | 1、服务端已启动  2、jmeter创建线程组并创建Http请求以及添加结果查看树 |
| 测试步骤 | 1、填写接口URL：http://localhost:8080/consumer/register  2、填写请求方式：POST  3、添加请求体：  consumerHead=null,  consumerPhone=13327348435，  consumerNickName=李明  consumerPassword=123456  consumerSex=0  4、发送请求 |
| 预期结果 | 注册通过并返回JWT |
| 实际结果 | {  "stausCode":200,  "msg":"注册成功",  "data":{"jwt":" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"}  } |
| 通过情况 | 符合预期 |

表6.1.3 个人信息修改接口测试用例

|  |  |
| --- | --- |
| 用例编号 | 1-3 |
| 测试目的 | 系统是否根据用户提交的新信息覆盖就信息 |
| 前置条件 | 1、服务端已启动  2、jmeter创建线程组并创建Http请求以及添加结果查看树 |
| 测试步骤 | 1、填写接口URL：http://localhost:8080/consumer/inf  2、填写请求方式：PUT  3、添加请求体：  consumerNickName=李明  4、添加本地文件  5、添加头部信息：jwt=\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  6、添加cookie:jsessionid=\*\*\*\*\*\*\*\*\*\*\*\*  7、发送请求 |
| 预期结果 | 修改成功，返回修改成功信息 |
| 实际结果 | {  "stausCode":200,  "msg":"修改成功",  "data":null  } |
| 通过情况 | 符合预期 |

表6.1.4 购物车查询接口测试用例

|  |  |
| --- | --- |
| 用例编号 | 1-4 |
| 测试目的 | 系统能否返回当前登录用户的购物车商品信息 |
| 前置条件 | 1、服务端已启动  2、jmeter创建线程组并创建Http请求以及添加结果查看树 |
| 测试步骤 | 1、填写接口URL：http://localhost:8080/cart/inf  2、填写请求方式：GET  3、添加头部信息：jwt=\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  4、添加cookie:jsessionid=\*\*\*\*\*\*\*\*\*\*\*\*  5、发送请求 |
| 预期结果 | 返回购物车商品信息 |
| 实际结果 | {  "stausCode":200,  "msg":"查询成功",  "data":{"commodityId"："a2dfdf-23423-dfssanmcmx"，"num"：2，"commodityName"："鸿星尔克山寨鞋","cavs"=["黑色白纹","42码"]，"price"：55.5}  } |
| 通过情况 | 符合预期 |

## 6.2商家模块

表6.2.1 商家入驻接口测试用例

|  |  |
| --- | --- |
| 用例编号 | 1-5 |
| 测试目的 | 系统能否根据当前用户入驻情况进行入驻信息保存,并返回初始店铺信息 |
| 前置条件 | 1、服务端已启动  2、jmeter创建线程组并创建Http请求以及添加结果查看树 |
| 测试步骤 | 1、填写接口URL：http://localhost:8081/seller/enter  2、填写请求方式：POST  3、添加请求体：  sellerId=\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  sellerName=韩\*\*\*\*  4、添加头部信息：jwt=\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  5、添加cookie:jsessionid=\*\*\*\*\*\*\*\*\*\*\*\*  6、发送请求 |
| 预期结果 | 入驻成功并附带店铺信息 |
| 实际结果 | {  "stausCode":200,  "msg":"入驻成功",  "data":{"storeId"："343sfddf-23eqw3-sandsad"，"storeName"："13423342343","storeHead"："http://localhost:8081/static/default.jpeg"}  } |
| 通过情况 | 符合预期 |

表6.2.2 店铺信息修改接口测试用例

|  |  |
| --- | --- |
| 用例编号 | 1-6 |
| 测试目的 | 系统能否将用户提交的新信息覆盖店铺旧信息 |
| 前置条件 | 1、服务端已启动  2、jmeter创建线程组并创建Http请求以及添加结果查看树 |
| 测试步骤 | 1、填写接口URL：http://localhost:8081/store/inf  2、填写请求方式：PUT  3、添加请求体：  storeName=eeeeeeeee  4、添加上传文件  5、添加头部信息：jwt=\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  6、添加cookie:jsessionid=\*\*\*\*\*\*\*\*\*\*\*\*  7、发送请求 |
| 预期结果 | 修改成功并附带信息的店铺信息 |
| 实际结果 | {  "stausCode":200,  "msg":"修改成功",  "data":{"storeId"："343sfddf-23eqw3-sandsad"，"storeName"："eeeeeeeee","storeHead"："http://localhost:8081/static/head/store/weeqeq-12312-sfdf-2131.jpeg"}  } |
| 通过情况 | 符合预期 |

表6.2.3 上传店铺营业执照接口测试用例

|  |  |
| --- | --- |
| 用例编号 | 1-7 |
| 测试目的 | 系统能否将用户提交的文件进行保存 |
| 前置条件 | 1、服务端已启动  2、jmeter创建线程组并创建Http请求以及添加结果查看树 |
| 测试步骤 | 1、填写接口URL：http://localhost:8081/store/license  2、填写请求方式：POST  3、添加上传文件  4、添加头部信息：jwt=\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  5、添加cookie:jsessionid=\*\*\*\*\*\*\*\*\*\*\*\*  6、发送请求 |
| 预期结果 | 添加成功并附带文件路径 |
| 实际结果 | {  "stausCode":200,  "msg":"添加成功",  "data":["http://localhost:8081/static/head/store/w1231eq-343-sfdf-2sdf.jpeg","http://localhost:8081/static/head/store/wgfq-12fggf-sfdf-2fgf.jpeg"]  } |
| 通过情况 | 符合预期 |

表6.2.4 商品发布接口测试用例

|  |  |
| --- | --- |
| 用例编号 | 1-8 |
| 测试目的 | 系统能否将商家提交商品信息进行保存 |
| 前置条件 | 1、服务端已启动  2、jmeter创建线程组并创建Http请求以及添加结果查看树 |
| 测试步骤 | 1、填写接口URL：http://localhost:8081/commodity/publish  2、填写请求方式：POST  3、添加请求体：  commodityName=野生何首乌  categoryId=41  cavs={"颜色分类":"一盒5片；1盒10片；2盒20片送2片","功能":"滋养润补"，"产地"："云南"，"价格"："300；550；1100"}  4、添加上传文件主图以及附图  5、添加头部信息：jwt=\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  6、添加cookie:jsessionid=\*\*\*\*\*\*\*\*\*\*\*\*  7、发送请求 |
| 预期结果 | 添加成功并附带商品id |
| 实际结果 | {  "stausCode":200,  "msg":"添加成功",  "data":{"commodityId":"sdf23-sgfg-1231-dssfs"}  } |
| 通过情况 | 符合预期 |

表6.2.5 已上架商品简略信息接口测试用例

|  |  |
| --- | --- |
| 用例编号 | 1-9 |
| 测试目的 | 系统能否查询当前店铺已经上架的商品的简略信息 |
| 前置条件 | 1、服务端已启动  2、jmeter创建线程组并创建Http请求以及添加结果查看树 |
| 测试步骤 | 1、填写接口URL：http://localhost:8081/commodity/insell/simple  2、填写请求方式：GET  3、添加头部信息：jwt=\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  4、添加cookie:jsessionid=\*\*\*\*\*\*\*\*\*\*\*\*  5、发送请求 |
| 预期结果 | 查询成功并附带上架商品简略信息 |
| 实际结果 | {  "stausCode":200,  "msg":"查询成功",  "data":{"commodityId":"sdf23-sgfg-1231-dssfs"，"commodityName":"山海异经"，"masterMap":" http://localhost:8081/static/comodity/master/fdsc-12fggfwere-sfdfwe-2fdfgf.jpeg ","insellDay":67, "commodityDescribe":"山海异闻录，等你来看" }  } |
| 通过情况 | 符合预期 |

表6.2.6 商品审核中信息接口测试用例

|  |  |
| --- | --- |
| 用例编号 | 1-10 |
| 测试目的 | 系统能否查询当前店铺审核中的商品的信息 |
| 前置条件 | 1、服务端已启动  2、jmeter创建线程组并创建Http请求以及添加结果查看树 |
| 测试步骤 | 1、填写接口URL：http://localhost:8081/commodity/incheck  2、填写请求方式：GET  3、添加头部信息：jwt=\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  4、添加cookie:jsessionid=\*\*\*\*\*\*\*\*\*\*\*\*  5、发送请求 |
| 预期结果 | 查询成功并附带审核中商品简略信息 |
| 实际结果 | {  "stausCode":200,  "msg":"查询成功",  "data":[{"commodityId":"s12f23-fsfg-1241-dfsfs"，"commodityName":"唐诗三百首"，"masterMap":" http://localhost:8081/static/comodity/master/fdsc-12fggfwere-sfdfwe-2fdfgf.jpeg ", "price":20}......]  } |
| 通过情况 | 符合预期 |

表6.2.7 买家视图商品信息查询接口测试用例

|  |  |
| --- | --- |
| 用例编号 | 1-11 |
| 测试目的 | 系统能否根据查询条件查询买家视图的商品的信息 |
| 前置条件 | 1、服务端已启动  2、jmeter创建线程组并创建Http请求以及添加结果查看树 |
| 测试步骤 | 1、填写接口URL：http://localhost:8081/commodity/consumer  2、填写请求方式：GET  3、添加请求参数：  page=1  pageNum=8  keyword=书籍  sort=id  4、发送请求 |
| 预期结果 | 查询成功并附带买家视图商品信息 |
| 实际结果 | {  "stausCode":200,  "msg":"查询成功",  "data":[{"commodityId":"s12f23-fsfg-1241-dfsfs"，"commodityName":"唐诗三百首"，"masterMap":" http://localhost:8081/static/comodity/master/fdsc-12fggfwere-sfdfwe-2fdfgf.jpeg ","commodityDescribe":"唐诗三百首经典之经典，李白、杜甫…", "price":20}……]  } |
| 通过情况 | 符合预期 |

表6.2.8 商品已下架信息接口测试用例

|  |  |
| --- | --- |
| 用例编号 | 1-12 |
| 测试目的 | 系统能否查询当前店铺已下架的商品的信息 |
| 前置条件 | 1、服务端已启动  2、jmeter创建线程组并创建Http请求以及添加结果查看树 |
| 测试步骤 | 1、填写接口URL：http://localhost:8081/commodity/nosell  2、填写请求方式：GET  3、添加头部信息：jwt=\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  4、添加cookie:jsessionid=\*\*\*\*\*\*\*\*\*\*\*\*  5、发送请求 |
| 预期结果 | 查询成功并附带已下架商品简略信息 |
| 实际结果 | {  "stausCode":200,  "msg":"查询成功",  "data":[{"commodityId":"ddgf2f23-fafg-3441-fdfs"，"commodityName":"经典小辣条"，"masterMap":" http://localhost:8081/static/comodity/master/asdsc-12ffnkw73-sfsawe-2fddf.jpeg ", "nosellDay":5, "nosellCause":"无"}......]  } |
| 通过情况 | 符合预期 |

## 6.3订单模块

表6.3.1 订单发起接口测试用例

|  |  |
| --- | --- |
| 用例编号 | 1-13 |
| 测试目的 | 系统能否根据用于提交的信息生成订单 |
| 前置条件 | 1、服务端已启动  2、jmeter创建线程组并创建Http请求以及添加结果查看树 |
| 测试步骤 | 1、填写接口URL：http://localhost:8082/order/ newborn  2、填写请求方式：POST  3、添加请求体  commodityId= ddgf2f23-fafg-3441-fdfs  commodityNum=4  cav=1包4袋  price=10  address=幽州明山市中心瀚华街道56号，烙鸣谢，13234323231  4、添加头部信息：jwt=\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  5、添加cookie:jsessionid=\*\*\*\*\*\*\*\*\*\*\*\*  6、发送请求 |
| 预期结果 | 发起成功并附带订单编号 |
| 实际结果 | {  "stausCode":200,  "msg":"发起成功",  "data":{"orderId":"dasmhjdfnm-2323-dfsd-asda"}  } |
| 通过情况 | 符合预期 |

表6.3.2 取消订单接口测试用例

|  |  |
| --- | --- |
| 用例编号 | 1-14 |
| 测试目的 | 系统能否根据订单编号删除未付款订单 |
| 前置条件 | 1、服务端已启动  2、jmeter创建线程组并创建Http请求以及添加结果查看树 |
| 测试步骤 | 1、填写接口URL：http://localhost:8082/order/chanel  2、填写请求方式：DELETE  3、添加请求体  commodityId= ddgf2f23-fafg-3441-fdfs  4、添加头部信息：jwt=\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  5、添加cookie:jsessionid=\*\*\*\*\*\*\*\*\*\*\*\*  6、发送请求 |
| 预期结果 | 取消成功并附带订单编号 |
| 实际结果 | {  "stausCode":200,  "msg":"发起成功",  "data":{"orderId":" ddgf2f23-fafg-3441-fdfs "}  } |
| 通过情况 | 符合预期 |

表6.3.3 订单付款接口测试用例

|  |  |
| --- | --- |
| 用例编号 | 1-15 |
| 测试目的 | 系统能否根据订单编号拉起第三方支付 |
| 前置条件 | 1、服务端已启动  2、jmeter创建线程组并创建Http请求以及添加结果查看树 |
| 测试步骤 | 1、填写接口URL：http://localhost:8082/order/pay  2、填写请求方式：GET  3、添加参数  commodityId= dfdfs-ferg-122341-fddf  4、添加头部信息：jwt=\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  5、添加cookie:jsessionid=\*\*\*\*\*\*\*\*\*\*\*\*  6、发送请求 |
| 预期结果 | 拉取成功并附带支付信息 |
| 实际结果 | {  "stausCode":200,  "msg":"拉取成功",  "data":{"orderId":" ddgf2f23-fafg-3441-fdfs "，"code":int0, "inf":  "alipay\_root\_cert\_sn=687b593432782732462789dasdnjkasdj3  4bhcbsncbshldmaskdnsajbzxv9034u23jn234csn29234u123dsfns……"}  } |
| 通过情况 | 符合预期 |

表6.3.4 取消订单接口测试用例

|  |  |
| --- | --- |
| 用例编号 | 1-16 |
| 测试目的 | 系统能否根据订单编号删除未付款订单 |
| 前置条件 | 1、服务端已启动  2、jmeter创建线程组并创建Http请求以及添加结果查看树 |
| 测试步骤 | 1、填写接口URL：http://localhost:8082/order/chanel  2、填写请求方式：DELETE  3、添加请求体  commodityId= ddgf2f23-fafg-3441-fdfs  4、添加头部信息：jwt=\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  5、添加cookie:jsessionid=\*\*\*\*\*\*\*\*\*\*\*\*  6、发送请求 |
| 预期结果 | 取消成功并附带订单编号 |
| 实际结果 | {  "stausCode":200,  "msg":"取消成功",  "data":{"orderId":" ddgf2f23-fafg-3441-fdfs "}  } |
| 通过情况 | 符合预期 |

表6.3.5 确认收货接口测试用例

|  |  |
| --- | --- |
| 用例编号 | 1-17 |
| 测试目的 | 系统能否根据订单编号确认订单收货 |
| 前置条件 | 1、服务端已启动  2、jmeter创建线程组并创建Http请求以及添加结果查看树 |
| 测试步骤 | 1、填写接口URL：http://localhost:8082/order/over  2、填写请求方式：PUT  3、添加参数  commodityId= dfdfs-ferg-122341-fddf  4、添加头部信息：jwt=\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  5、添加cookie:jsessionid=\*\*\*\*\*\*\*\*\*\*\*\*  6、发送请求 |
| 预期结果 | 收货成功并附带订单编号 |
| 实际结果 | {  "stausCode":200,  "msg":"收货成功",  "data":{"orderId":" dfdfs-ferg-122341-fddf "}  } |
| 通过情况 | 符合预期 |

## 6.4平台模块

表6.4.1 商品种类添加接口测试用例

|  |  |
| --- | --- |
| 用例编号 | 1-18 |
| 测试目的 | 系统能否保存商品种类信息 |
| 前置条件 | 1、服务端已启动  2、jmeter创建线程组并创建Http请求以及添加结果查看树 |
| 测试步骤 | 1、填写接口URL：http://localhost:8081/cav/category  2、填写请求方式：POST  3、添加请求体  categoryName=橡皮擦  father=6  4、添加头部信息：jwt=\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  5、添加cookie:jsessionid=\*\*\*\*\*\*\*\*\*\*\*\*  6、发送请求 |
| 预期结果 | 添加成功并附带种类编号 |
| 实际结果 | {  "stausCode":200,  "msg":"添加成功",  "data":{"categoryId":40}  } |
| 通过情况 | 符合预期 |

表6.4.2 支付账户添加接口测试用例

|  |  |
| --- | --- |
| 用例编号 | 1-19 |
| 测试目的 | 系统能否保存平台收付款账户信息 |
| 前置条件 | 1、服务端已启动  2、jmeter创建线程组并创建Http请求以及添加结果查看树 |
| 测试步骤 | 1、填写接口URL：http://localhost:8081/plant/account  2、填写请求方式：POST  3、添加请求体  accaId=1  account=1231asdnajks23sdsasdaqwe  avariable=0  4、添加头部信息：jwt=\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  5、添加cookie:jsessionid=\*\*\*\*\*\*\*\*\*\*\*\*  6、发送请求 |
| 预期结果 | 添加成功并附带种类编号 |
| 实际结果 | {  "stausCode":200,  "msg":"添加成功",  "data":{"accountId":12}  } |
| 通过情况 | 符合预期 |

# 总结

本说明书的主要工作归纳如下：

1.对系统的背景以及相关技术进行简略说明。

2.对系统的功能性需求进行用例建模以及说明，对非功能性需求进行说明。

3.对系统功能模块进行划分以及对数据库进行分析创建。

4.对系统的主要功能模块使用查询流程图进行描述。

5.对系统进行代码实现，并对系统的主要功能的核心代码展示。

6.对系统的主要功能模块所对应的服务端接口进行测试，并使用测试用例进行表述。

本次毕业设计的收获：

本次系统的设计与实不足：

1、数据库设计不足：一是商品种类及其属性表分析与设计不彻底，使得其关系模式并没有完全满足数据库范式要求，从而使得系统对商品的属性操作及其不便，性能也大大折扣。二是某些表的主键采用bigint自增，可能导致在分布式场景下主键不唯一。

2、界面设计不足：界面不够美观。

3、模块设计上不足：一是系统的静态资源应该分离出来，比如用户上传的文件，应该分离到静态资源服务器上进行存储。二是系统订单模块中调用的付款转账功能应该分离出来进行封装到收付款模块中，不应该让收付款模块仅仅进行第三方支付的回调以及信息记录。

4、安全性不足：一是接口安全性，没有严格控制接口的访问规则。二是没有对客户端的数据进行严格检测。

# 致谢

毕业设计的顺利完成离不开陈老师的耐心指导。作为指导老师，陈老师有着深厚的专业知识、认真严谨的科研态度、丰富的实际开发设计经验，作为陈老师的学生我感到很幸运。在此表示对陈老师的最诚挚地谢意！

同时要感谢专业课的老师在专业知识方面认真负责，坚持理论与实践相结合，为我的专业知识打下了良好的基础、提高了我独立分析问题与解决问题的能力、树立了我强烈的团队合作意识。在生活处事方面，提醒我要真诚、冷静、感恩、宽容。

# 参考文献

|  |
| --- |
| [1]ionic官方文档: https://ionicframework.com/docs[Z].  [2]湖南移商动力网络技术有限公司. 一种基于移动APP商城系统技术:CN201610983855.X[P]. 2018-05-22.  [3] SS P, Kumar A. Web3. 0 E-Commerce Decentralized Application[J]. 2022.  [4]何俊建帆,孙仕云. 基于Android的地区特产在线销售管理App设计[J]. 百科论坛电子杂志,2021(10):806. DOI:10.12253/j.issn.2096-3661.2021.10.798.  [5]angular中文官方文档: https://angular.cn/[Z].  [6]张希,谢婧媛. 当前app技术开发存在问题与解决探讨[J]. 中国高新区,2019(12):248-249.  [7]胡孔法.数据库原理及应用[M].机械工业出版社.2020  [8]翁湦元,单杏花,朱建军,等. 基于Angular2的模块化Web应用架构设计[J]. 铁路计算机应用,2019,28(1):58-62. DOI:10.3969/j.issn.1005-8451.2019.01.012.  [9]常振业,刘佳伟,闫茜茜,刘鑫豪,周丽平,朱国良,鲁池梅,刘兴云.基于Ionic的App跨平台混合开发技术的研究[J].软件,2019,40(04):215-218.  [10]陈建强,周培勇,汪洋. 一种基于云计算和移动互联网技术的多商户电商云系统及其下单购物实现方法:CN201610798869.4[P]. 2017-02-01.  [11]Wei X, Wang Y. Research on the Application of Pixel Style in UI (User Interface) Design[C]//The 6th International Conference on Arts, Design and Contemporary Education (ICADCE 2020). Atlantis Press, 2021: 557-559.  [12] Cai Z, Liu Y, Gan Y, et al. Design and implementation of online mall system based on java web[J]. International Journal of Performability Engineering, 2019, 15(12): 3237.  [13]王斌.面向对象分析与设计[M].机械工业出版社.2020  [14]夏丽华,卢旭.UML建模与应用标准教程[M].清华大学出版社.2018  [15]李宗花,朱林.软件工程原理与实践[M].南京大学出版社.2020  [16]JWT官方文档：<https://jwt.io/>[Z] |