



湖南信息学院

本科生毕业论文(设计)

论文题目: 基于 JavaEE 的社区团购系统的

设计与实现

学生姓名: 胡辉家

学 号: 201701420729

二级学院: 计算机科学与工程学院

专业名称: 软件工程

班 级: 2017 级软件工程 7 班

指导老师: 杨 顺

2021 年 6 月 12 日

教务处制



湖南信息学院

毕业论文（设计）诚信声明

本人声明：所呈交的本科毕业论文（设计），是本人在指导老师的指导下，独立开展工作所取得的成果，成果不存在知识产权争议，除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或创作过的作品成果。对本文工作做出重要贡献的个人和集体均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

本科毕业论文（设计）作者签名：

二〇二一年六月十二日

摘 要

本文通过分析国内社区团购行业的市场需求和国内外社区团购系统的现状,总结出适合社区和村镇的社区团购系统的功能需求,给出了详细的系统设计方案,分析得到用户端商品展示、注册登录、购物车管理、自提点管理、订单管理和佣金提现等 6 个模块,管理端商品管理、商品类型管理、会员管理、自提点管理、佣金管理、订单管理和综合管理等 7 个模块,然后对系统功能进行实现和测试,测试结果表明系统符合预期的需求。同时考虑到成本和易用性等因素,系统采用 B/S 结构,后端使用 SpringBoot 框架开发,前端采用主流的 Bootstrap、Layui 和 ECharts 技术开发,数据库使用 MySQL 和 Redis,应用服务器使用 Tomcat 服务器。系统借助 Web 手机网页方便社区和村镇用户随时随地下单,畅享社区团购服务,方便社区和村镇门店赚取佣金。

针对社区和村镇用户的特点,系统给出了一套相对全面的、能够快速下沉到社区和村镇的团购电商方案,解决了网络团购的最后一公里问题,系统的最终实现对于社区及村镇电商发展有着重要的意义。

关键词: 社区团购; 电子商务; SpringBoot 框架; B/S 结构

ABSTRACT

This paper analyzes the market demand of the domestic community group purchase industry and the status quo of the community group purchase system at home and abroad, summarizes the functional requirements of the community group purchase system suitable for communities and villages, gives a detailed system design scheme. Through the analysis of the system, we can get six user-side modules of commodity display, register and login, shopping cart management, self-pick-up point management, order management, commission withdrawal, and management side seven modules of commodity management, commodity type management, member management, self-pick-up point management, commission management, order management, and comprehensive management. Then the system functions is implemented and tested, and the test results show that the system meets the expected requirements. At the same time, taking into account factors such as cost and ease of use, the system adopts B/S structure, the back-end uses SpringBoot framework, the front-end uses mainstream Bootstrap, Layui and ECharts technology, the database uses MySQL and Redis, and the application server uses Tomcat server. The system makes it easy for users in communities and villages and towns to place orders anytime and anywhere with the help of mobile web pages, enjoy community group buying services, and facilitate communities and village stores to earn commissions.

According to the characteristics of communities and villages, the system provides a relatively comprehensive group purchase e-commerce scheme that can quickly sink into communities and villages, and solves the problem of the last kilometer of online group purchase. The final implementation of the system is of great significance to the development of communities and villages e-commerce.

KEYWORDS: Community Group Buying; Electronic Commerce; SpringBoot Framework; B / S Structure

目 录

1 绪 论	1
1.1 项目研究的背景与意义	1
1.1.1 项目研究背景	1
1.1.2 项目研究意义	2
1.2 项目研究现状	2
1.2.1 国外研究现状	2
1.2.2 国内研究现状	3
1.2.3 研究现状综述	4
1.3 研究内容与研究方法	4
1.3.1 主要研究内容	4
1.3.2 主要研究方法	4
1.4 系统技术简介	5
1.4.1 前端技术	6
1.4.2 后端技术	6
1.4.3 数据库技术	6
1.5 论文主要结构	6
1.6 本章小结	7
2 系统需求分析	8
2.1 可行性分析	8
2.2 功能需求分析	8
2.2.1 用户端需求分析	8
2.2.2 管理端需求分析	11
2.3 系统数据流分析	13
2.4 非功能需求分析	14
2.5 软硬件需求	15
2.6 本章小结	15
3 系统设计	16
3.1 系统功能模块	16

3.2 系统主要业务流程	17
3.2.1 用户端业务流程	17
3.2.2 提现申请业务流程	18
3.2.3 购物业务流程	18
3.2.4 服务端业务流程	19
3.2.5 商品管理业务流程	20
3.3 系统时序设计	20
3.4 数据库设计	21
3.4.1 数据库概念设计	21
3.4.2 数据库逻辑设计	26
3.4.3 数据库物理实现	26
3.5 本章小结	30
4 系统实现	31
4.1 用户端会员功能	31
4.1.1 注册登录功能	31
4.1.2 首页功能	34
4.1.3 分类展示功能	35
4.1.4 查看商品详情功能	36
4.1.5 搜索功能	37
4.1.6 购物车功能	38
4.1.7 订单结算功能	39
4.1.8 选择自提点功能	40
4.1.9 订单管理功能	41
4.1.10 会员个人中心功能	42
4.2 用户端管理员功能	44
4.2.1 登录注册功能	44
4.2.2 管理员个人中心功能	45
4.2.3 查看佣金记录功能	45
4.2.4 提现申请功能	46
4.2.5 查看提现明细功能	46
4.3 管理端功能	47

4.3.1 综合管理功能	47
4.3.2 管理员信息管理功能	48
4.3.3 自提点信息管理功能	49
4.3.4 会员管理功能	50
4.3.5 商品类型管理功能	51
4.3.6 商品管理功能	52
4.3.7 订单管理功能	53
4.3.8 佣金管理功能	54
4.3.9 提现管理功能	55
4.3.10 系统统计功能	56
4.4 本章小结	58
5 系统测试	59
5.1 系统测试目的与方法	59
5.1.1 测试目的与意义	59
5.1.2 测试的方法和工具	59
5.2 用户端功能测试	59
5.2.1 注册登录找回密码功能测试	59
5.2.2 购物车功能测试	61
5.2.3 订单结算及管理功能测试	62
5.2.4 选择自提点功能测试	62
5.2.5 佣金记录功能测试	63
5.2.6 提现申请功能测试	63
5.3 管理端功能测试	64
5.3.1 管理员管理功能测试	64
5.3.2 自提点管理功能测试	65
5.3.3 会员管理功能测试	65
5.3.4 商品类型管理功能测试	65
5.3.5 商品管理功能测试	66
5.3.6 佣金管理功能测试	67
5.3.7 订单管理功能测试	67
5.3.8 系统统计功能测试	68

5.3.9 提现管理功能测试	68
5.4 功能测试结果分析	69
5.5 系统性能测试	70
5.6 本章小结	71
6 结 论	72
参考文献	73
致 谢	75

1 绪 论

1.1 项目研究的背景与意义

1.1.1 项目研究背景

随着计算机技术的飞速发展，电子商务在我国蓬勃发展，各种电子商务模式应运而生，例如 B2B(Business to Business)、B2C(Business to Customer)、C2C(Customer to Customer)、O2O (Online to Offline)、O2P (Online to Partner) 等，涌现出一批非常优质的电商平台，如淘宝商城、京东商城等^[1]。截止到 2019 年底，我国重点网络零售平台（含服务类平台）店铺数量有 1946.9 万家，全国电子商务交易额已达 34.81 万亿元^[2]。与此同时，在我国的农村地区仍有 74.9% 的村没有农村电商配送站点^[3]。农村传统电商“最后一公里”仍存在薄弱环节，很多农村地区快递存在“长运距+低需求密度”的特点，农村快递资费较贵、二次加价问题仍较突出。

从 2009 年第一次“双十一”到现在，我国电商行业已经走过了一个高速发展的黄金时期。当人们以为阿里和京东之后，再也不会出现更好的电商平台时，洋码头、1 号店 1 号海购和海沃全球购等平台凭借跨境电商分红的优势逐渐站稳了脚跟，随后专注于消费升级的蜜芽、聚美极速免税店、网易严选、小米有品等垂直自营电商平台诞生了。当人们又认为电商不再有新的机会时，拼多多、洋葱等社交电商平台迅速崛起。阿里、京东、小红书也纷纷效仿加入社交电商战场。无人不分销，无站不拼团，成了当下的电商趋势^[4]。

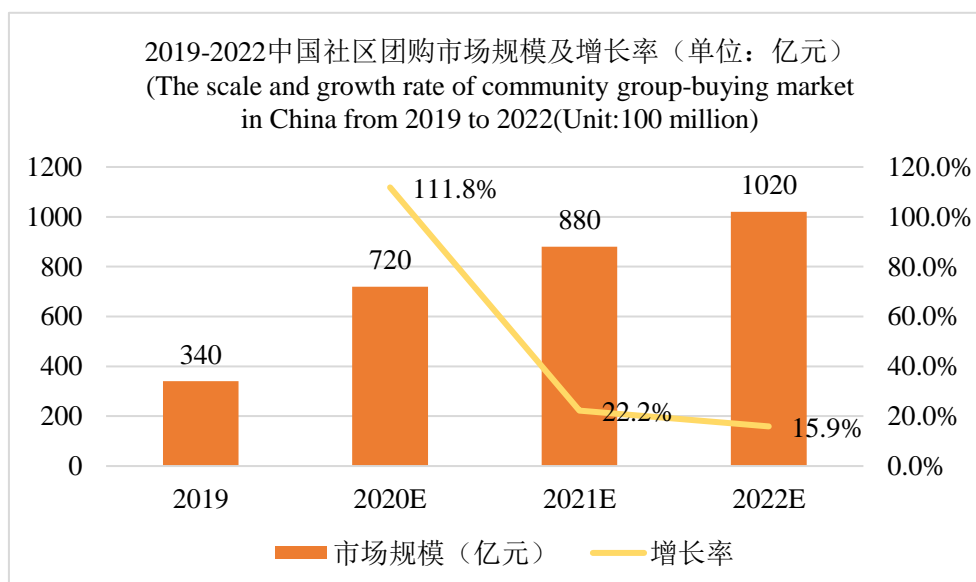


图 1.1 2019-2022 中国社区团购市场规模及增长率^[5]

社区团购作为一种新电商、新零售模式，通过平台组织强大的供应链体系和快速

下沉到社区及村镇门店的自提服务，为用户提供平价或低价商品进行团购的主要形式，使消费者在最大程度上获取商品让利。由此，社区团购电商模式成功引起了消费者、供应链企业乃至资本市场的关注，据艾媒咨询决策机构的调研和预测，到 2022 年我国社区团购市场规模将超过 1220 亿元，增长率将达到 28.4%^[5]。2019-2022 中国社区团购市场规模及增长率，如图 1.1 所示。

1.1.2 项目研究意义

疫情的特殊形势和越来越碎片化的时间里，人们的消费习惯发生重要改变，“宅家网购”的时效性和便利性成为大多数人选择电商平台的首要条件。社区团购电商行业依靠社区和村镇实体门店，为用户提供“预售+自提”的优质、便利、快捷服务，能够解脱人们在越来越碎片化的时间里对购物时间、地点的限制。由此，社区团购作为一种新电商模式，越来越深入人心。

开展社区团购服务具体的社会意义总结如下：

（1）开辟供应链企业产品流通新渠道，使企业的产品能快速被消费者接受，从而扩大企业的市场份额。

（2）减少产品流通的中间环节，降低企业产品的固定成本，实现用户、企业、平台和团长（社区实体门店）四方的互惠、共赢。

（3）用户单独购买商品也能获得团购价格，节省了用户的采购成本和时间，促进用户消费，扩大国家消费内需，优化消费结构，促进消费升级。

（4）一旦平台销售的产品出现质量问题，甚至发生法律纠纷，有利于发挥平台用户群体维权优势，确保用户权益。

基于上述选题背景和研究意义，特别是一系列相对成熟的技术和最新研究成果为解决网络团购在社区和村镇的最后一公里的应用问题带来了新的、较好的方案，即设计并实现基于 JavaEE 的社区团购系统。

1.2 项目研究现状

1.2.1 国外研究现状

电子商务在互联网服务中出现较早，在国外有 eBay、亚马逊这样重推荐、轻广告，走精品化路线、品牌化路线的电商公司^[6]。亚马逊在移动客户端首页提供自营模块入口，各种商品排行榜供用户选择，还通过热门促销等活动刺激消费者消费。而谈论到团购，人们就会想到团购网站创始人 Andrew Mason 打造的 Groupon，Groupon 即团体加优惠券，这种模式的运营方式为在每天的特定时间内，团购网站会推出低价格的服务类商品，该服务类商品会限制买家地域和团购时间，且每人每天只能下单一次。Groupon 模式的核心利益是收取商户 30%到 50%不等的佣金^[7-9]。

在阅读了一些国外学者的文章后，对本人的研究有着很大的帮助，如下：

在日本具有决策支持的团购系统得到实现，在该系统中买家通过自行成团，可以以较低的价格购买商品。该系统还支持买方利用层次分析法决策，可以提供一个基于多属性的偏好(公用事业)买家集成服务^[10]。

在消费者感知价值、信任和人际关系对消费者参与网络团购意图影响的研究中，研究者使用结构方程模型检验了样本中 553 名受访者，表明人际关系间的信任在消费者参与网络团购中起着核心作用。消费者感知价值也表现出强烈的感知信任关系^[11]。

社会的电子商务和卫星网络模型：改善合作建设创新的概念一文中指出，电子商务平台需要朝着新兴的模式发展，电子商务还必须整合协作工具，支持社交网络机制，以改善协作，发展信任，以及实施网络关系的战略方法。社会化电子商务概念定义了这种创新的电子商务愿景，它整合了基于 Web 的协作工具，强调了社会资本和社交网络的重要性，增强了供应链管理的作用。该文提出的卫星网络模型符合这种协作工作逻辑，定义了一种功能性方法，以增强电商行业的社交网络行为^[12]。

1.2.2 国内研究现状

国内著名电商公司有阿里巴巴、京东等，这些大型电商公司都是面向不特定人群即全社会人群，依靠高流量支撑公司，并没有考虑社区用户的需求^[13]。社区用户搜索到的卖家基本分布在全国各地，即使选择的发货地在本地，也有可能因为物流中间环节，导致送货时间较长，从而使用户购物的时效性和便利性无法得到完全的保障。

国内社区社群电商的起步相对较晚，但发展迅猛，且呈现出跟风、聚集效应。大部分社区社群电商平台都于 2017 年左右上线，如十荟团、你我您、食享会、兴盛优选等。在资本市场相对萎靡的大环境下，社区社群电商还能先后赢得投资机构的青睐，获得红杉资本、淡马锡、春华资本等知名风投机构数千万乃至数亿元的融资，这足以说明资本市场看重社区社群电商未来的发展潜力和社区社群电商尚处于野蛮生长中，机会众多^[14,15]。

在阅读了一些国内学者的文章后，对本人的研究有着很大的帮助，如下：

智慧社区电商系统的核心概念是“集成”和“服务”，集成是将小区安防、家居等等集成在一起打造智慧社区，服务是为社区用户提供优质服务，例如医疗、教育和日常生活消费品等^[16]。

基于 JavaEE 的社区电商平台系统面向社区及村镇等小范围内的应用场合，通过该平台用户可以进行完整的网络购物流程。该系统可用于社区和城镇的电商管理^[17]。

云智慧社区平台将电商与社区巧妙结合在一起，通过接入周边商家的方式，为社区居民提供区域化的社区购买及配送服务，提升了居民在社区内生活的体验。同时有效为周边商家和社区带来人口信息流动与盈利，完成了多方共赢^[18]。

基于 ThinkPHP 框架的团购网站分为用户平台、主后台和商家平台三个模块，编写了会员管理、商家入驻管理、商品列表、商家管理、团购商品分类等功能。该平台的成功搭建，对于顾客、商户以及中介平台来说三方受益，形成了共赢的局面^[19]。

在基于仲裁计算机制的公平、安全的团购系统中，研究者利用安全计算理论来满足交易双方的身份认证、交易数据的完整性和买方的匿名性等安全要求。该系统的实现成功的避免了交易纠纷的出现^[20]。

安全的移动团购系统分析了每个阶段和参与方的时间成本，并计算了每个阶段的通信成本。该系统使用的 Schnorr 数字签名技术，能够满足许多安全要求，能够在自发团购网站上应用^[21]。

1.2.3 研究现状综述

针对现阶段传统电商无法解决的最后一公里配送、在大型电商平台上购物时效性受到限制以及很多农村地区快递存在的“长运距+低密度需求”导致的快递二次收费加价等问题，社区电商给出了新方案。社区电商平台经营的商品都是人们所需的日用品、生鲜品等，配送人员都是周边的商户，且无需二次收费，就可以达到快速配送商品的要求。同时，稳定的社区有稳定的用户流量，平台和商家不必投入大量的营销费用来维持流量，就能有较好的收益。

上述国内外学者研究并实现的系统基本是 PC 网页端，没有进行手机网页端的适配，在移动互联网大规模普及的时代无法提供随时随地、无需安装卸载的便捷服务。基于该研究现状，本课题选择构建基于 B/S 模式的社区团购系统，结合基于 SpringBoot 框架的开发技术对系统进行设计和实现。本系统通过深入挖掘社区团购服务，借助网页用户端、微信群等等资源，实现用户、企业、平台和团长（社区实体便利店）四方的互惠、共赢。因此结合当前社区用户的特点，设计满足社区团购需求的系统是本文的研究重点。

1.3 研究内容与研究方法

1.3.1 主要研究内容

系统的实现秉持着用户至上的原则，通过用户调查和自悟等研究方法，吸收和采纳了许多的用户意见，明确了社区团购系统的实际功能需求，为了满足课题实现的可行性、降低系统的开发维护成本以及系统的可扩展性等需求，本文决定采用 SpringBoot 框架结合 JavaEE 技术搭建服务后端，采用 H5 网页实现移动网页用户端，采用 Layui 等框架实现系统管理端。在实现系统核心业务的同时，系统还研究了用户密码加密存储（MD5 加密算法）、ECharts 基础数据报表、阿里云短信接口、支付宝沙箱支付接口、百度地图接口和 Token 登录鉴权等。

1.3.2 主要研究方法

本文采用的研究方法如下：

（1）文献阅读法：通过查找和阅读国内外有关社区团购行业、社区团购系统和基于 JavaEE 技术的期刊、硕士博士论文，总结归纳社区团购行业发展现状以及国内外学者的最新研究成果，为设计和实现系统做好充分的准备。

（2）软件工程法：运用软件工程瀑布模型的设计流程，从可行性分析、需求分析、系统设计、编码、测试等方面设计并实现基于 JavaEE 的社区团购系统。

（3）自悟法：通过自悟法，将系统中的对象、类、属性、操作和方法等模型构建出来，用于指导系统的设计与实现。

（4）用户调查法：通过面向对象的用户调查，完善自悟过程中构建的相关模型，完善系统的设计与实现。

（5）经验总结法：在系统的设计与实现过程中，要善于总结经验，发现规律，并将这些经验和规律用于指导实践。

1.4 系统技术简介

系统的表现层将借助 HTML5、Bootstrap 和 Layui 实现，借助 ECharts 实现每日销售额、订单数、提现金额的折线图和自提点在全国的分布数量地图，数据交换基于 HTTP 协议，使用 Json 数据格式，服务支持层使用 Tomcat 服务器，服务实现层使用 SpringBoot、Spring、SpringMVC、Mybatis 等框架，数据存储使用 MySQL 数据库，用户登录 Token 信息缓存使用 Redis 数据库。系统技术架构，如图 1.2 所示。

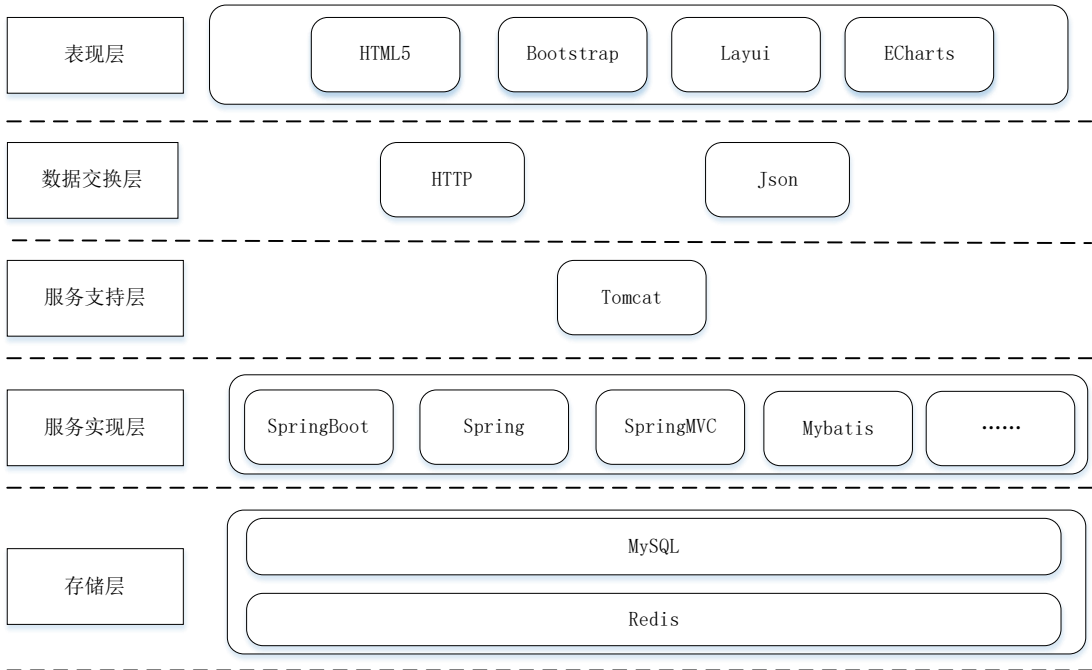


图 1.2 系统技术架构图

1.4.1 前端技术

Bootstrap 是一款前端业内应用十分广泛的响应式布局、移动端优先框架。

Layui 采用模块化的组件，通过极其简单的数据渲染方式，被广泛应用于管理端系统的实现。

ECharts 是一款由百度团队开发，Apache 开源基金会孵化的基于 JavaScript 图表库，该库能够很好的为广大开发者提供数据可视化图表。

1.4.2 后端技术

SpringBoot 的设计目的是用来简化新 Spring 应用的初始搭建以及开发过程^[22]。SpringBoot 通过减少不必要配置文件的编写和使用注解等方式方法，优化代码结构和代码编写量，深受广大开发者的喜爱。

MyBatis 是一个支持普通 SQL 查询，存储过程和半自动化映射的持久层框架^[23]。MyBatis 使用简单的 XML 或注解，采用半自动化的映射方式，方便开发者编写和维护复杂的 SQL 语句。此外 MyBatis 采用 ONGL 表达式完成动态 SQL，通过动态 SQL 语句能够减少手动拼装 SQL 语句的烦恼，能够提高代码的复用率，能够减少批量插入数据时，数据库连接数，提高数据库运行性能。

1.4.3 数据库技术

MySQL 是一种关系型数据库，该类型的数据库能够使用分库分表技术，解决数据库吞吐量的瓶颈问题，从而提高数据查询的速度和灵活性。

Redis 是一个开源的使用 ANSI C 语言编写、支持网络、可基于内存亦可持久化的日志型、Key-Value 数据库^[24]。

1.5 论文主要结构

按照本文的工作内容，本文将分为 6 章，每章的主要论述安排如下：

第 1 章为绪论。首先介绍了本课题的研究背景和意义，并从电商行业的发展趋势及国内外学者对社区团购的研究现状入手，分析得到传统电商模式和现有社区团购系统的优点、缺陷和不足。同时还阐述了论文的主要研究内容、方法和组织结构，最后介绍系统实现使用的相关技术；

第 2 章为系统需求分析。结合国内外研究现状和社区团购行业的发展需求，对社区团购系统进行可行性分析，通过用例图分析业务功能需求，最后阐述了系统的非功能性需求和开发、测试、部署的软硬件需求，确保开发出来的软件系统能够得到实际的应用；

第 3 章为系统设计。结合第二章的系统需求分析，对系统的功能、结构、业务流程、数据库表的概念和逻辑进行设计，对数据库表进行物理实现，为系统稳定、高效、

安全的运行提供理论基础；

第 4 章为系统实现。按照系统实际需求进行 UI 原型设计，采用 Bootstrap、Layui、ECharts 等框架，实现前端页面的构建。采用 SpringBoot、Mybatis 等框架，实现基于 Knife4j 风格的后端接口。随后在前端利用 Ajax 对系统接口进行请求，获取接口返回的数据并渲染到前端页面，实现基于 JavaEE 的社区团购系统；

第 5 章为系统测试。根据系统需求分析相关文档，设计并执行功能和性能测试用例，得到实际测试结果，从中分析出系统功能缺陷及各项性能指数，修复 Bug、完善系统，进行回归测试，确认系统的功能和性能都符合实际需求，最后对测试结果进行总结；

第 6 章为总结与展望。总结现阶段本课题研究所取得的主要成果，并针对本系统不足，为下一步的研究提出建议。

1.6 本章小结

本章阐述了社区团购系统的研究背景与意义，详细分析了现有国内外学者所实现的社区团购系统的现状，得到了现有的系统以及传统电商模式的优点、缺陷和不足，并给出了相应的解决方案——设计并实现一个基于 JavaEE 的社区团购系统。最后本章还分析了论文的主要研究内容和方法，阐述了部分关键技术和论文的结构。

2 系统需求分析

2.1 可行性分析

(1) 经济可行性

在系统的设计和实现过程中，使用的软件和技术都是开源免费的。从课题实现的角度看，现阶段系统的部署对软硬件的配置要求并不高。在系统部署的 1Mbps 公网带宽、2GB 运行内存和 50GB 云硬盘的腾讯云学生服务器上也能够流畅的运行。社区团购系统的用户端（指浏览器方面）对设备软硬件配置的要求较低，只要是能够正常运行的手机或电脑都能流畅运行系统。

(2) 技术可行性

系统所使用的前端和后端技术都已经十分的成熟，行业和市场的应用也十分广泛，本人在大学期间也曾利用相关技术开发过其他的系统，具有一定的熟练度。系统所用数据库是 MySQL 和 Redis，这两种数据库安全性较高、数据处理能力较强。当然受限于实际开发经验的不足，在开发中可能会遇到一些技术盲区，如 Layui 等前端技术方面，但是我相信在老师的指导、自身的努力和同学的帮助下，这些困难都会一一克服，最终将社区团购系统设计并实现出来。

(3) 操作可行性

在设计与实现的系统过程中，始终秉持以用户为中心的一致性、简洁明确、体现特色、明确的导航设计等原则。系统操作简单，用户使用一般的手机或电脑浏览器都能进行并完成系统的购物等操作，管理员使用一般的电脑浏览器都能进行并完成系统的管理操作。同时系统的设计遵循软件设计的总体原则，在着眼于当前实用的基础上，系统采用前后端分离技术，且后端接口完全遵循 Knife4j 风格，为后期数据接口的开放保留了可能性，也给未来系统的扩展和升级保留了一定的空间。

2.2 功能需求分析

系统主要有用户端和管理端两个子系统，用户端为会员及自提点管理员所使用的手机网页端，使其能够进行商品购买、个人资料管理、订单管理、佣金管理等操作，管理端为超级管理员所使用的电脑网页端，使其能够利用管理端对平台所有信息进行管理操作。

2.2.1 用户端需求分析

用户端分为游客、会员、自提点管理员三类用户和一个由会员、自提点管理员泛化得到的普通用户，用户端四类用户的用例分析如下。

(1) 未注册的用户称为游客，游客可以访问系统用户端进行浏览商品、搜索商

品、查看商品详情、注册成为会员和门店入驻报名等操作。游客用例图，如图 2.1 所示。

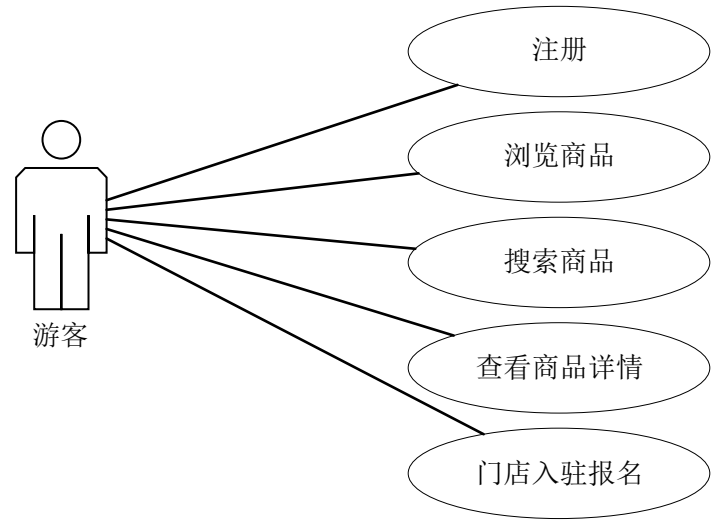


图 2.1 游客用例图

(2) 注册成功或门店入驻报名审核成功的用户泛化后称为普通用户，普通用户除了具备游客的所有权限外，还能进行登录、退出登录、查看资料、修改资料、修改密码、查看所属订单、查看订单详情等操作。普通用户用例图，如图 2.2 所示。

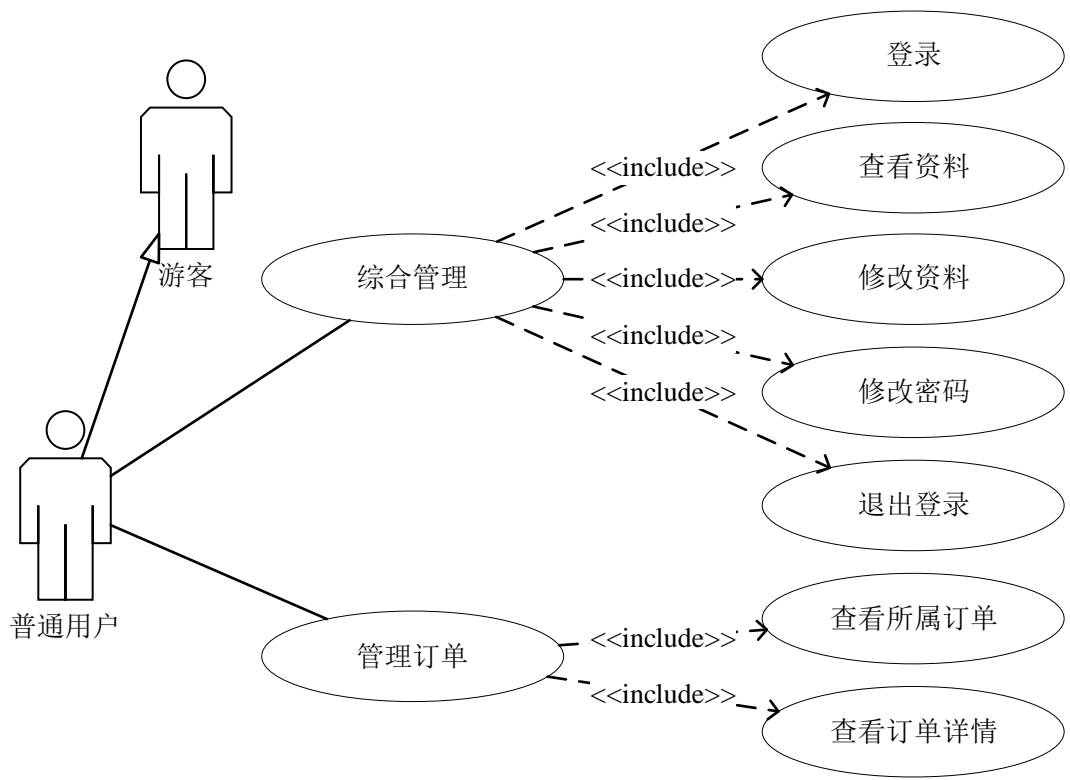


图 2.2 普通用户用例图

(3) 注册成功的用户称为会员，会员除了具备普通用户的所有权限外，还能进

行选择自提点、查看历史自提点、将商品加入购物车、增减购物车商品数量、删除购物车商品、结算购物车商品、生成订单以及和第三方系统支付宝进行交互实现支付等操作。会员用例图，如图 2.3 所示。

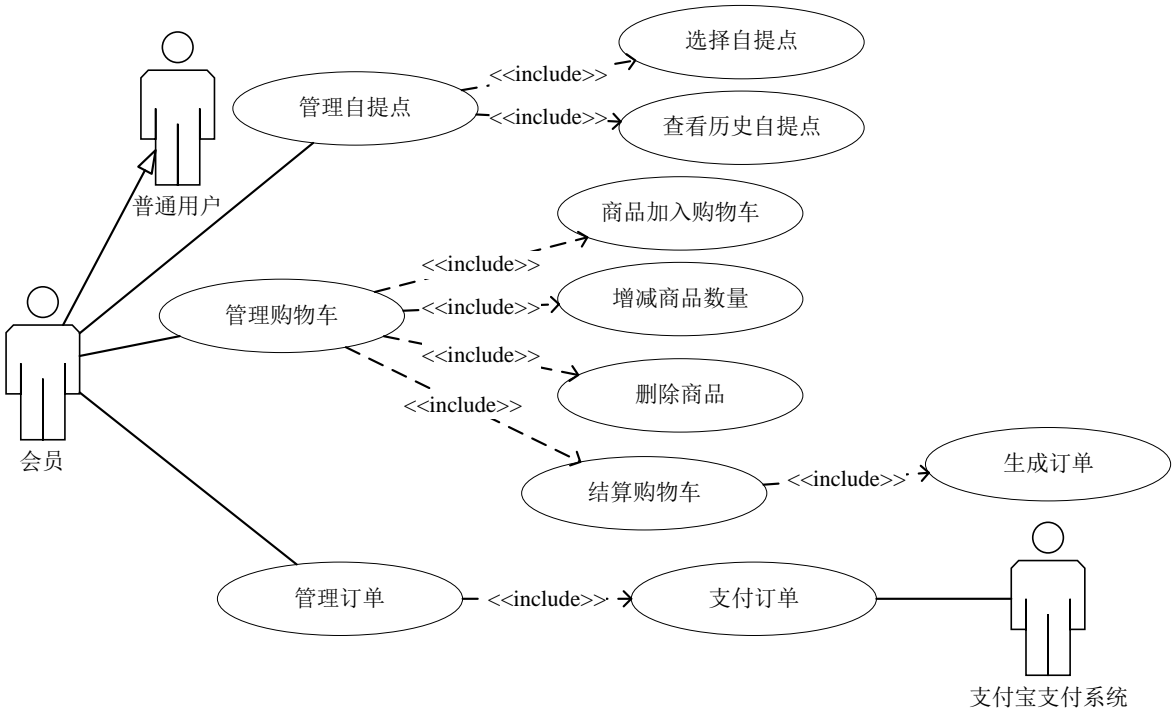


图 2.3 会员用例图

(4) 门店入驻报名审核成功的用户称为自提点管理员，自提点管理员除了具备普通用户的所有权限外，还能进行查看佣金记录、查看提现记录、申请提现、申请退货等操作。自提点管理员用例图，如图 2.4 所示。

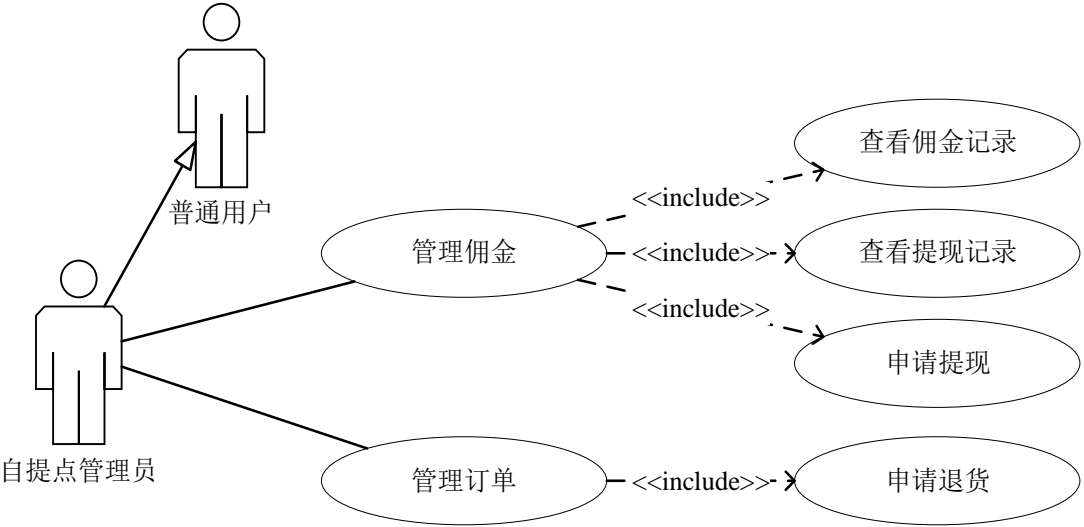


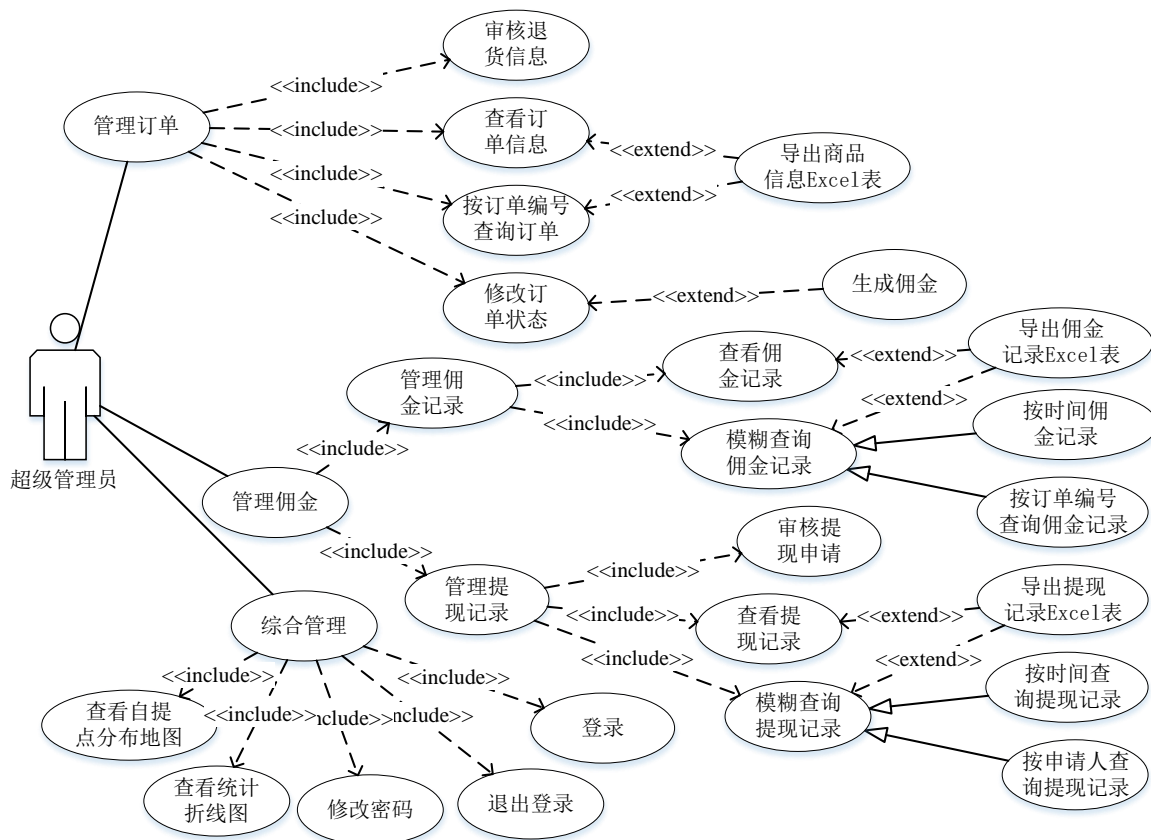
图 2.4 自提点管理员用例图

根据上述客户端用例分析，得到用户端功能划分如下。

- (1) 商品展示：所有人都可以对商品进行浏览和搜索，查看商品详细信息。
- (2) 注册登录：游客通过手机号注册、登录、找回密码。
- (3) 购物车管理：会员可以将商品加入购物车，在购物车中管理商品（增减、删除商品），也可以将购物车中的商品生成订单、使用支付宝支付。
- (4) 订单管理：会员和自提点管理员可以按订单状态分类查看所属订单，查看订单详情，会员还可以通过自提点管理员对订单商品进行退货申请。
- (5) 自提点管理：会员在进行购物时必须选择自提地址，加入自提点团购，享受就近自提服务，此外符合条件的会员和门店还可以申请成为自提点。
- (6) 佣金管理：自提点管理员可以查看所属订单佣金、进行提现申请、查看提现记录。

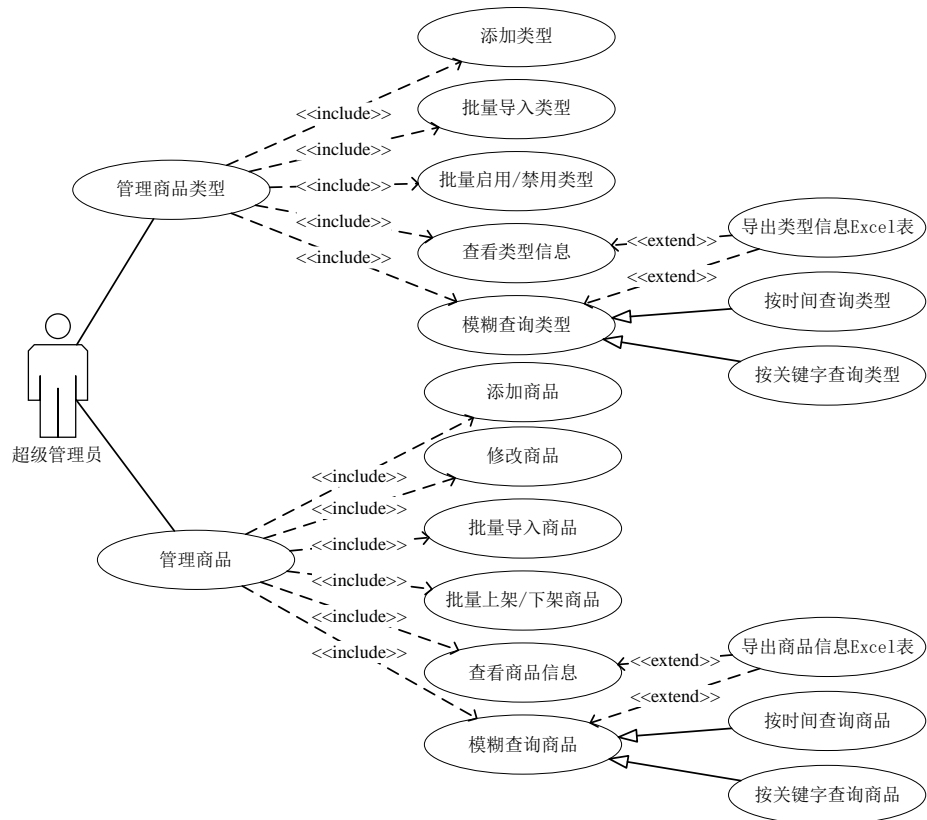
2.2.2 管理端需求分析

管理端只有超级管理员一类用户，在管理端超级管理员能够进行商品类型、商品、自提点、会员、订单、佣金以及综合信息的管理等操作。超级管理员用例图，如图 2.5 所示。



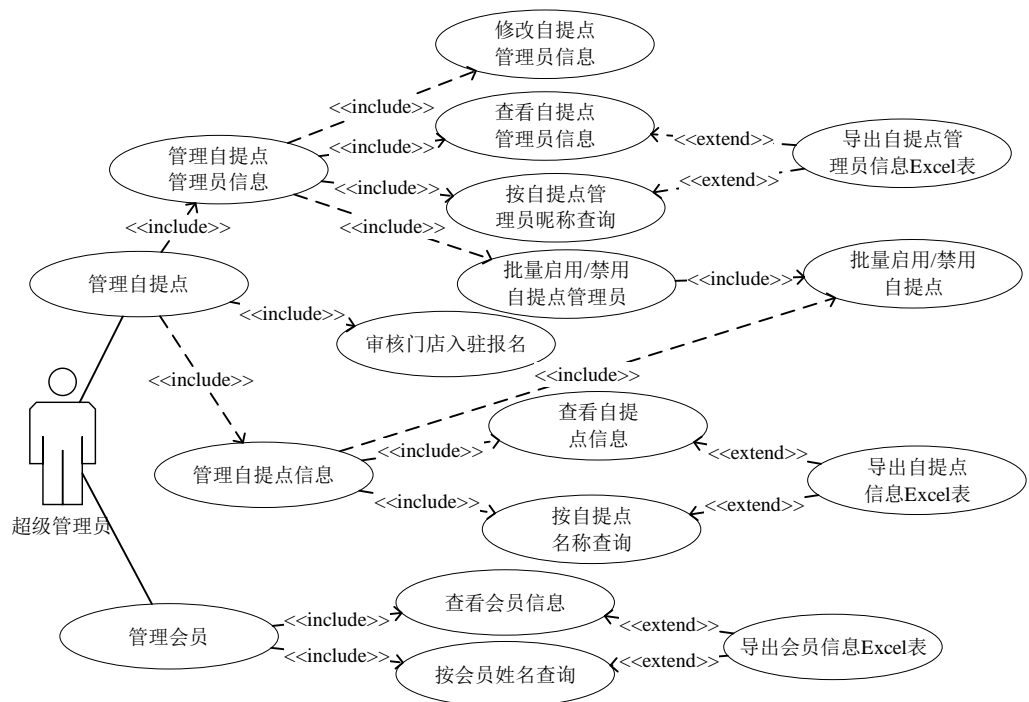
(a)

图 2.5 超级管理员用例图



(b)

图 2.5 超级管理员用例图



(c)

图 2.5 超级管理员用例图

根据上述管理端用例分析，得到管理端功能划分如下。

(1) 商品管理：包括商品添加、商品修改、分页查看商品信息、批量上架/下架、Zip 压缩文件批量导入、Excel 文件导出、按创建时间查找、按商品名称模糊查询等操作。

(2) 商品类型管理：包括商品类型添加、分页查看类型信息、批量启用/禁用、Excel 批量导入/导出、按创建时间查找、按商品类型名称模糊查询等操作。

(3) 自提点管理：包括审核门店入驻报名、查看自提点管理员信息、修改管理员信息、批量启用/禁用自提点管理员和自提点、按自提点管理员昵称模糊查询、批量导出、按自提点名称模糊查询等操作。

(4) 佣金管理：包括查看佣金记录、提现申请审核、查看提现记录、批量导出、按时间和订单编号查询佣金记录、按时间和申请人查询提现记录等操作。

(5) 会员管理：包括查看会员信息、按会员姓名模糊查询、批量导出等操作。

(6) 订单管理：包括查看订单、订单详情、审核退货、Excel 文件导出、按订单编号查询、按订单结算时间查询、修改订单状态、生成订单佣金等操作。

(7) 综合管理：包括查看统计折线图、自提点分布地图、登录、退出登录、修改密码等操作。

2.3 系统数据流分析

系统为会员、自提点管理员提供从商品查看、加入购物车、购物车结算、生成订单、支付宝支付、商品自提、订单分佣、佣金提现全过程的社区团购服务，系统为管理端超级管理员提供自提点管理、商品类型管理、商品管理、订单管理、会员管理、佣金管理等功能。系统顶层数据流图，如图 2.6 所示。系统 0 层数据流图，如图 2.7 所示。

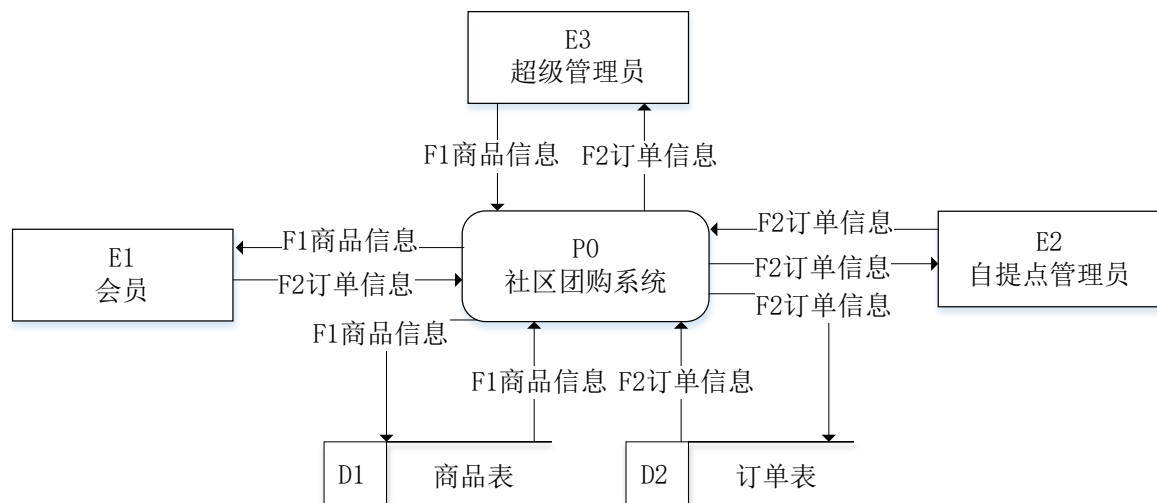


图 2.6 系统顶层数据流图

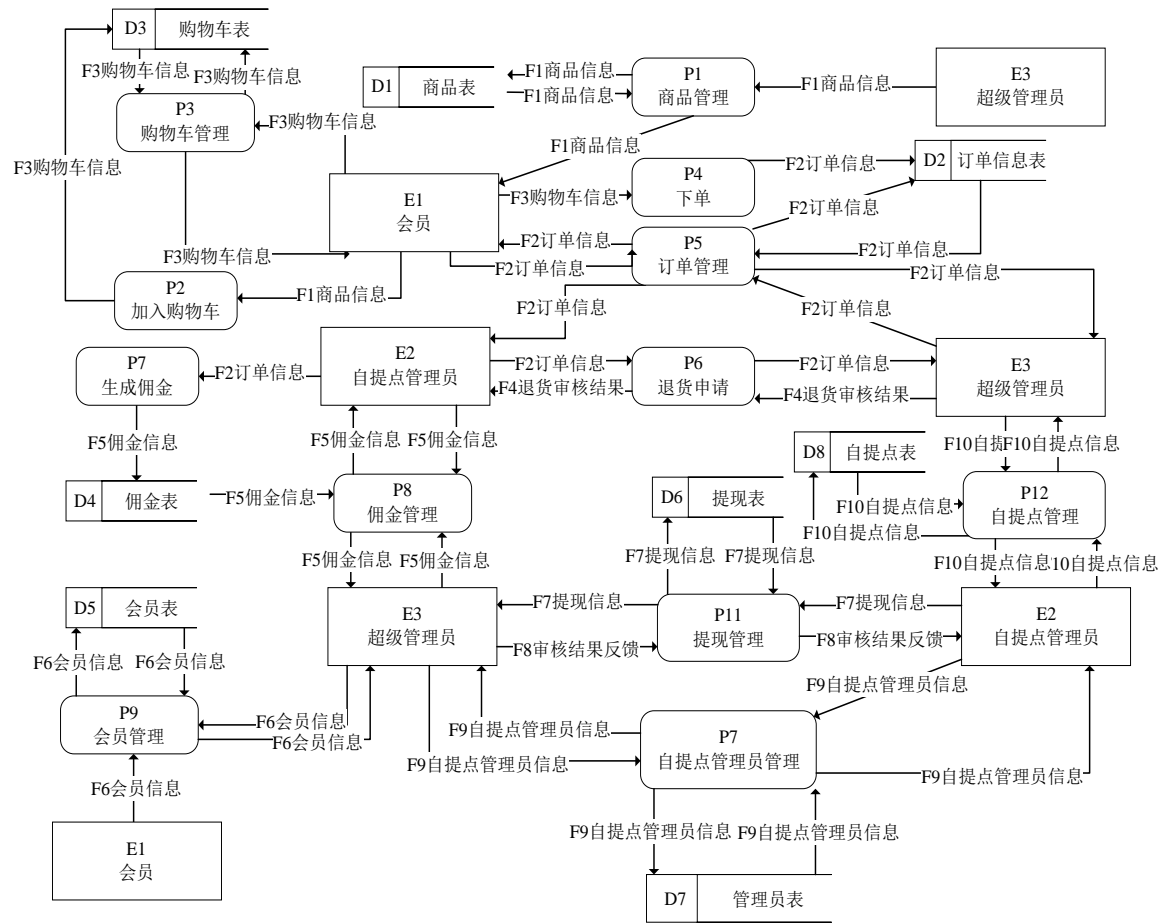


图 2.7 系统 0 层数据流图

2.4 非功能需求分析

(1) 安全性分析

系统应对用户的密码进行 MD5 加密运算，采用 Post 传值来保障用户的账号密码安全，系统应拥有防 SQL 注入、防 XSS（Cross Site Scripting，跨站脚本攻击）和防 CSRF（Cross Site Request Forgery，跨站点请求伪造）攻击等安全保护措施^[22]。

(2) 易用性分析

系统的页面应简洁美观，应秉持以用户为中心的一致性、简洁明确、体现特色、明确的导航设计等 Web 界面设计原则^[25]。同时为了保持系统的操作方便，要给用户在操作的过程中提供与业务流程相关的、简洁的引导性提示，而不是使用专业词汇。

(3) 稳定性分析

根据电商系统的普遍特性，系统往往需要足够的并发性来保证系统能够正常运行。系统接口的平均响应时间应遵循 2-5-8 原则^[26]。因此，综合 2-5-8 原则和课题实

现的需求，系统接口请求的异常百分比不能超过 0.3%，平均响应时间应该在 3 秒内。

2.5 软硬件需求

开发和测试系统的软件需求，如表 2.1 所示。

表 2.1 软件需求表

软件平台	版本
MySQL	5.5
JDK	1.8
Tomcat	8.0
Maven	3.6
Redis	5.0
CentOS	7.6
Windows	10
jQuery	3.2.1
SpringBoot	2.4.5
Layui	2.6.5
ECharts	4.1.0
Jmeter	5.4
Postman	4.1.2

系统开发完成后部署上线的硬件需求，如表 2.2 所示。

表 2.2 硬件需求表

硬件名称	大小
CPU 内核	1 核
运行内存	2GB
SSD 云硬盘	50GB
公网带宽	1Mbps

2.6 本章小结

本章首先对系统进行可行性研究与分析，得出设计与实现社区团购系统的可行性。然后，根据自悟和用户调查划分出四类主要用户，分别为游客、会员、自提点管理员、超级管理员，对四类用户进行用例分析和功能需求分析，明确系统要实现的功能，分析系统的顶层及 0 层数据流，用于指明系统各个模块间的逻辑关系。最后本章还针对系统提出非功能性需求，得出系统要达到的性能标准。

3 系统设计

3.1 系统功能模块

根据前一章的需求分析，可将系统分为用户端和管理端。管理端主要是对商品类型信息、商品信息、订单信息、自提点信息、佣金信息、提现信息等进行集中统一管理。用户端的核心业务是进行购物和订单结算等。

管理端主要有 7 大功能模块，分别为商品管理、商品类型管理、自提点管理、佣金管理、会员管理、订单管理、综合管理。超级管理员通过系统管理端实现对商品信息、购物流程中的订单退货信息、自提点管理员提现申请进行审核，对门店入驻报名信息申请进行审核，以及对系统其他各类信息进行日常管理。管理端功能模块，如图 3.1 所示。

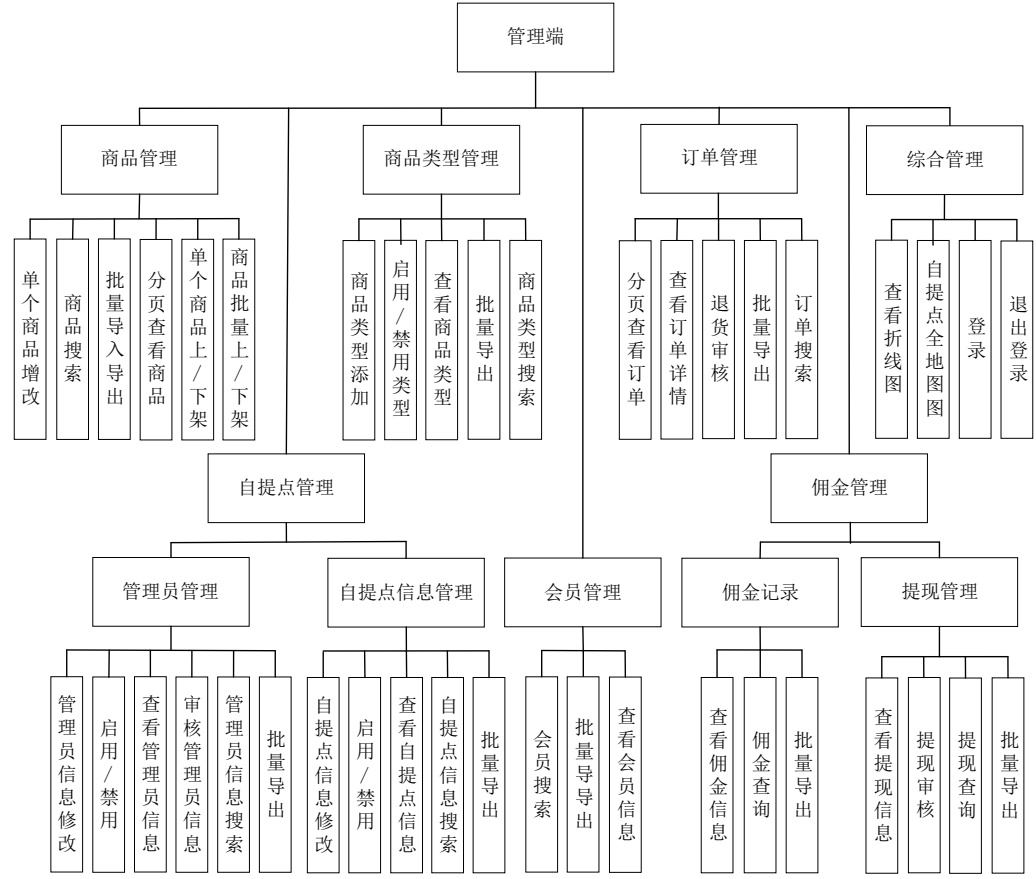


图 3.1 管理端功能模块图

用户端主要有 6 大功能模块，分别为商品展示、注册登录、购物车、自提点管理、订单管理以及佣金管理。游客通过用户端浏览、搜索商品，了解商品的详细信息；会员通过本系统用户端实现购物、自提管理、订单管理；自提点管理员通过用户端实现订单管理、分佣、提现，达到互利共赢的效果。用户端功能模块，如图 3.2 所示。

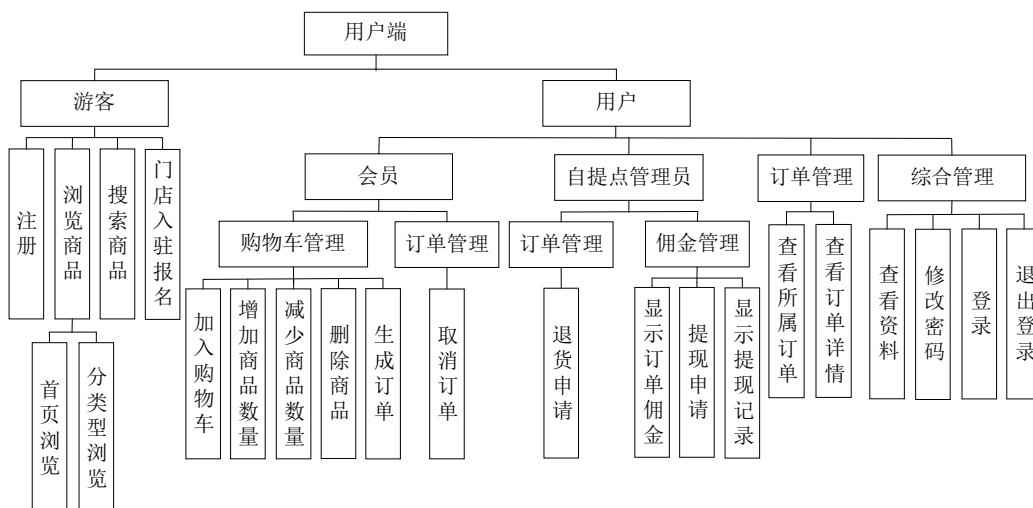


图 3.2 用户端功能模块图

3.2 系统主要业务流程

3.2.1 用户端业务流程

用户端业务流程，游客或会员通过用户端手机网页访问系统，可以进行查看商品基本信息、搜索商品等操作，会员注册登录后，可以进行购物车管理，订单管理，门店入驻报名等操作。用户端业务流程，如图 3.3 所示。

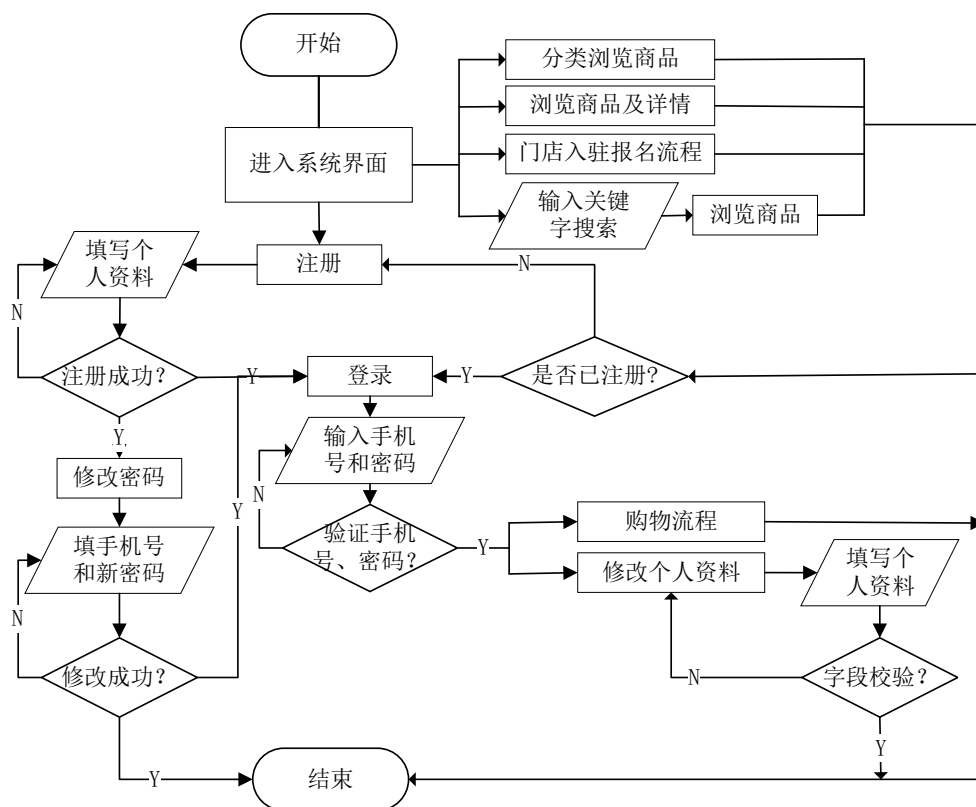


图 3.3 用户端业务流程图

3.2.2 提现申请业务流程

自提点管理员通过用户端成功登录系统，进行佣金提现申请，超级管理员在管理端查看提现申请，并审核通过后人工放款给自提点管理员。提现申请业务流程，如图 3.4 所示。

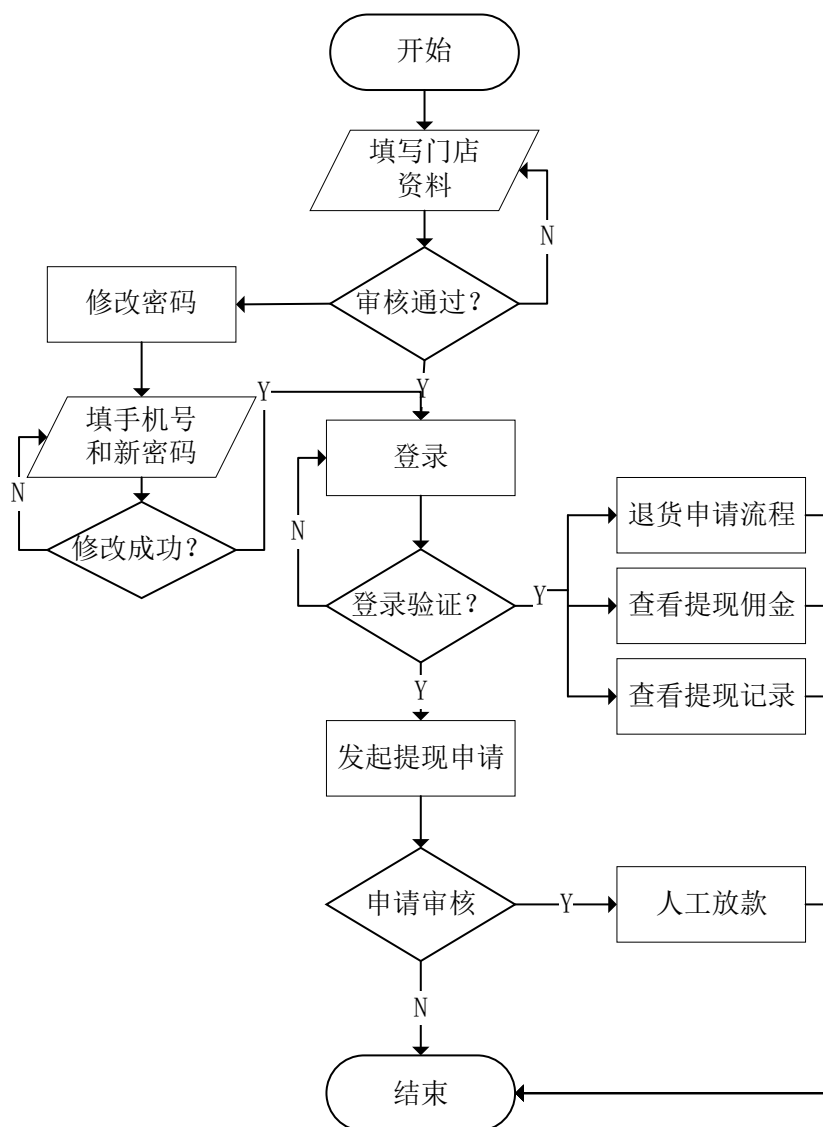


图 3.4 提现申请流程图

3.2.3 购物业务流程

会员通过用户端成功登录系统，进行商品加入购物车、下单、选择自提点，支付成功后，进行商品出库，通过第三方或自建物流的配送人员驾驶配送车辆将订单商品送达自提点，交付到自提点管理员手中，会员可到自提点进行自提，订单完成后会生成一笔佣金给自提点管理员，会员可向自提点管理员反馈，并发起退货申请，退货申请经过管理端超级管理员审核后，可进行退货和人工退款。购物业务流程，如图 3.5 所示。

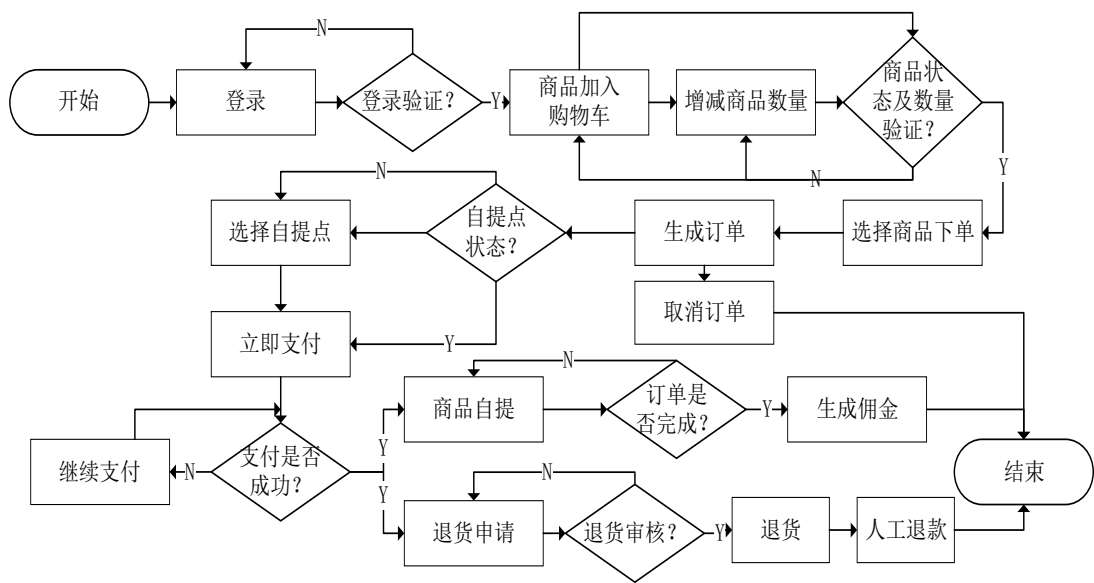


图 3.5 购物业务流程图

3.2.4 管理端业务流程

超级管理员访问管理端，对商品、佣金等信息进行管理。服务端业务流程，如图 3.6 所示。

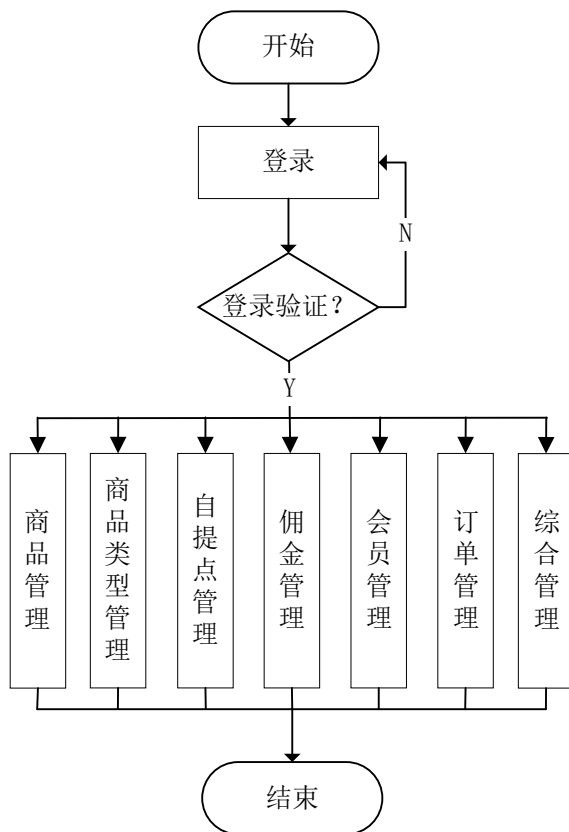


图 3.6 服务端业务流程图

3.2.5 商品管理业务流程

超级管理员进入商品列表页面进行商品添加、商品修改、批量上架/下架，批量导入/导出，搜索等操作。商品类型管理、自提点管理、会员管理、订单管理等业务流程与商品管理业务流程类似，本文不再赘述。商品管理业务流程，如图 3.7 所示。

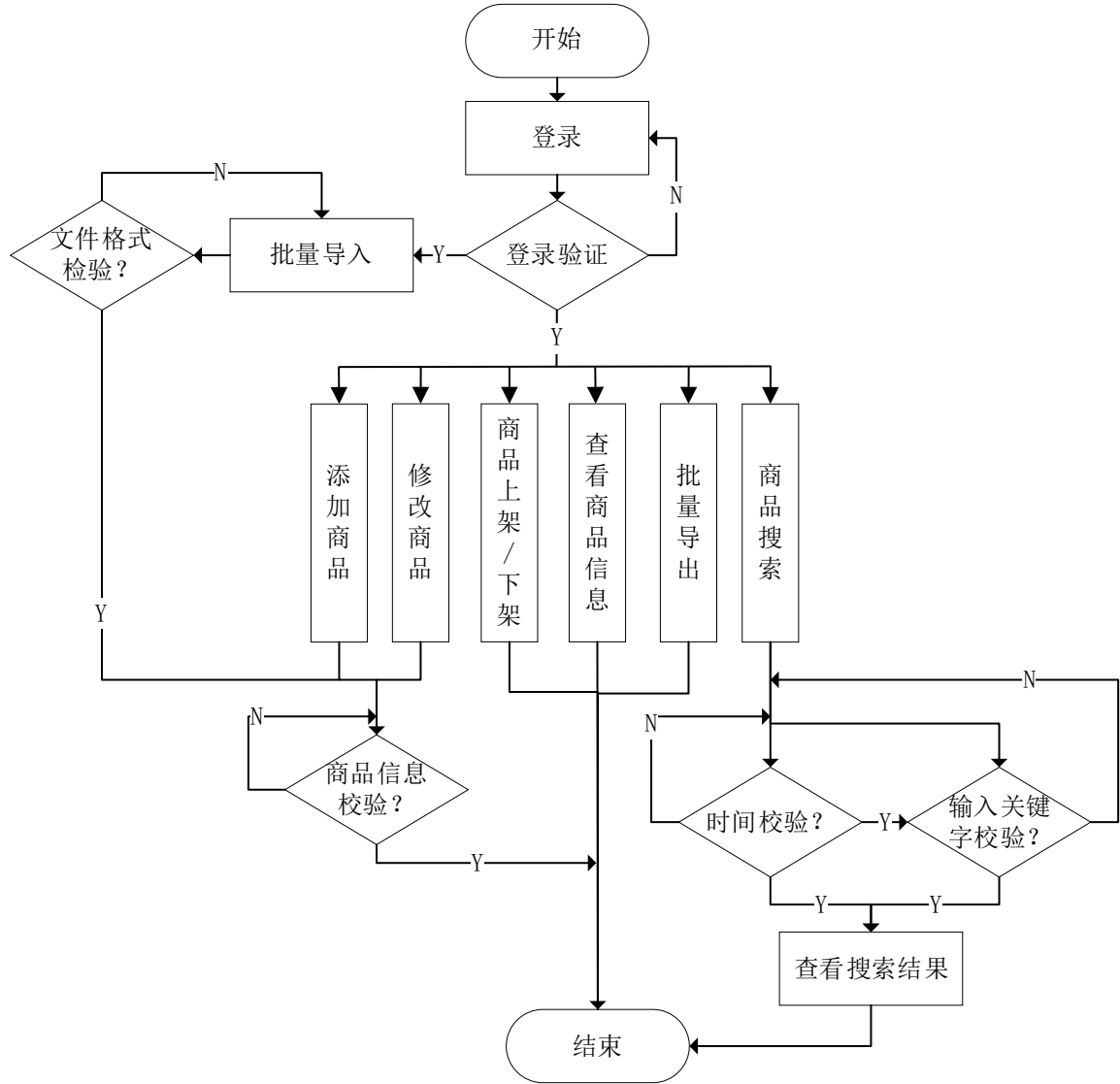


图 3.7 商品管理业务流程图

3.3 系统时序设计

用户填写表单发起登录请求，前端浏览器利用 HTTP、Post 提交表单参数至后端应用服务。后端应用服务对提交的参数进行校验，如果登录成功，后端应用服务会根据 Token 生成规则自动创建一个唯一的 Token，随后后端应用服务会将 Token 保存至 Redis 数据库并设置失效时间。Token 保存成功后，后端应用服务会将该用户登录产生的 Token 信息返回给前端浏览器，前端浏览器拿到 Token 后，将使用 sessionStorage

属性把 Token 信息缓存在 Session 对象中。

用户发起请求访问某个受保护的资源、链接，前端浏览器取出缓存的 Token，并在请求头中写入 Token 字段及对应的属性值，随后通过 Ajax 请求后端应用服务的接口。后端应用服务获取、校验 Token 合法性，如果 Token 校验不通过，则给前端浏览器返回不通过提示，如果 Token 校验通过，则给前端浏览器返回请求处理响应结果，前端浏览器给用户展示响应结果。Token 认证、授权时序图，如图 3.8 所示。

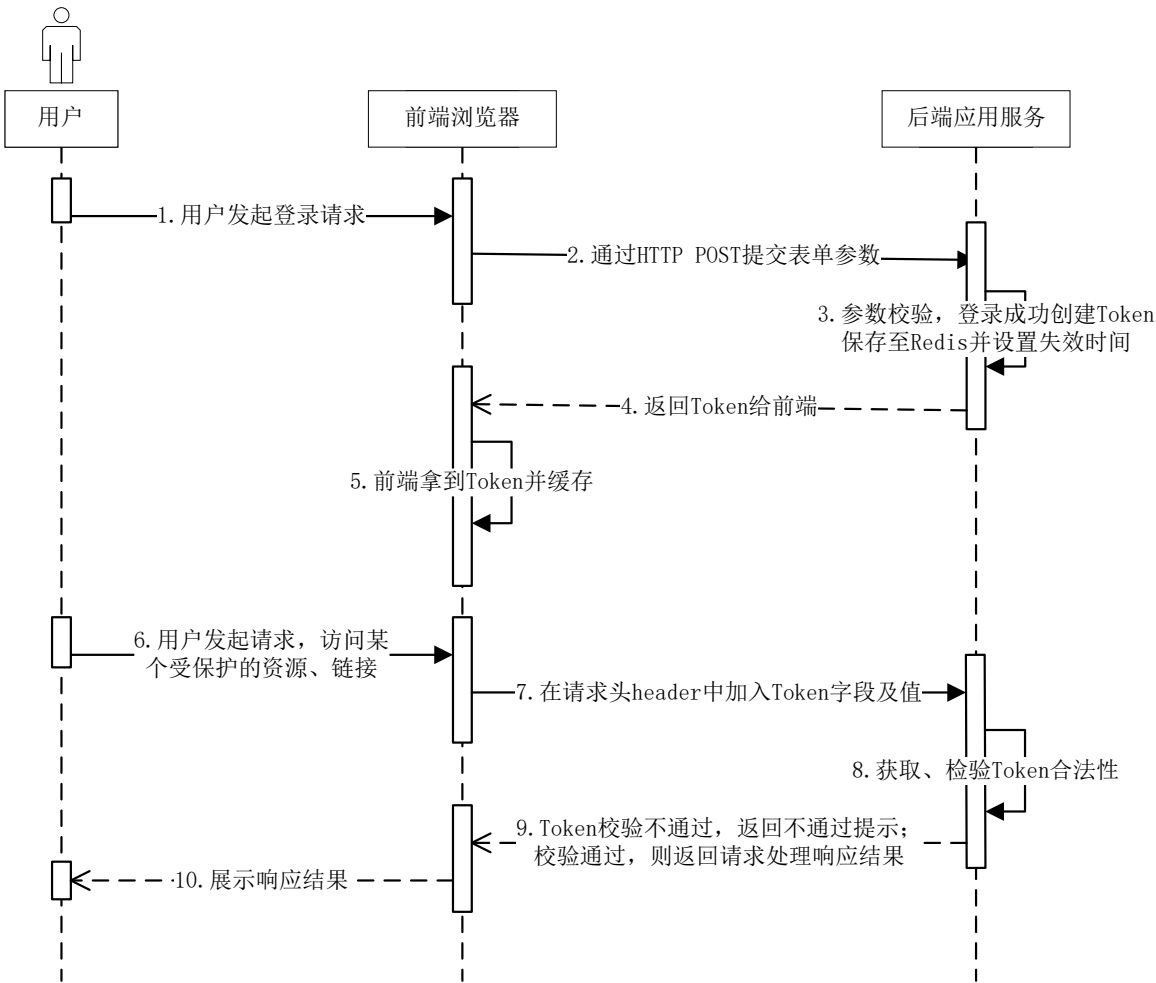


图 3.8 Token 认证、鉴权时序图

3.4 数据库设计

3.4.1 数据库概念设计

一般来说，在设计数据库时为了使数据库冗余较小、结构合理，每张数据表字段之间的关系应尽量满足第三范式，但是不考虑数据库实际需求，完全满足第三范式要求的数据表设计，通常不是最佳的设计。如本系统中订单表中的订单总价字段就是一个较为常见的冗余字段，该字段的值等于商品单价乘以商品数量的结果，但是为了提

高数据库的查询速度，不得不适当降低范式标准，在该表中增加冗余字段订单总价，从而达到用数据库的储存空间换取查询时间的效果。数据库的实体设计如下。

（1）商品类型实体集：包括的属性有类型 ID、类型名称、简介、状态、创建时间、修改时间。商品类型实体图，如图 3.9 所示。

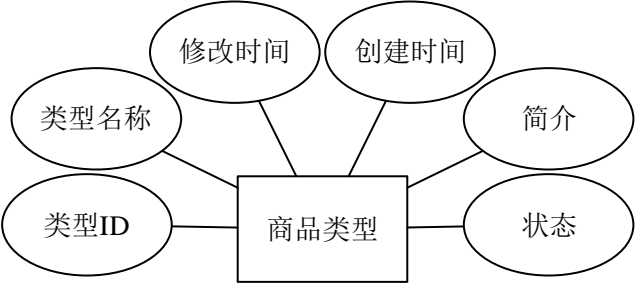


图 3.9 商品类型实体图

（2）地区实体集：包括的属性有地区 ID、省级行政区名。地区实体图，如图 3.10 所示。

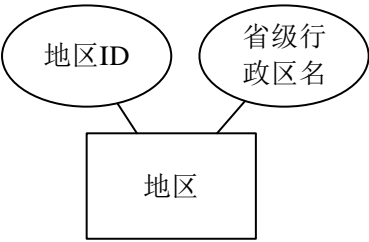


图 3.10 地区实体图

（3）管理员实体集：包括的属性有管理员 ID、昵称、登录密码、手机号、账户头像地址、账户余额、管理员类型、管理员状态。管理员实体图，如图 3.11 所示。

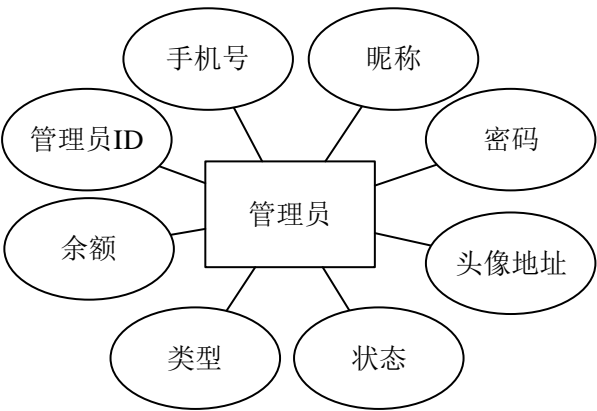


图 3.11 管理员实体图

（4）会员实体集：包括的属性有会员 ID、昵称、登录密码、手机号、账户、账户头像地址。会员实体图，如图 3.12 所示。

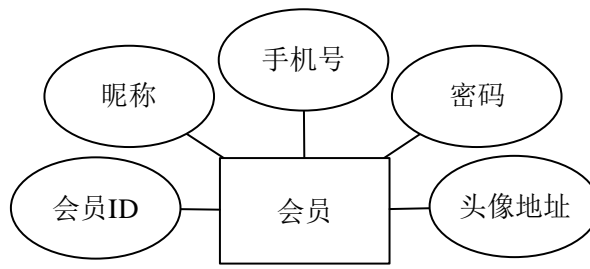


图 3.12 会员实体图

（5）自提点实体集：包括的属性有自提点 ID、自提点名称、所属地区 ID、管理员 ID、自提点状态、自提点详细地址。自提点实体图，如图 3.13 所示。

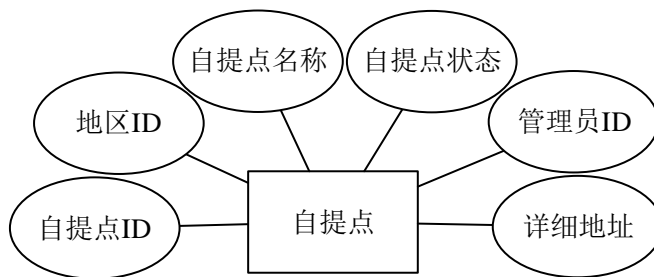


图 3.13 自提点实体图

（6）商品实体集：包括的属性有商品 ID、商品名称、价格、库存、已售数量、图片地址、详情、是否轮播、类型 ID、规格、状态、创建时间。商品实体图，如图 3.14 所示。

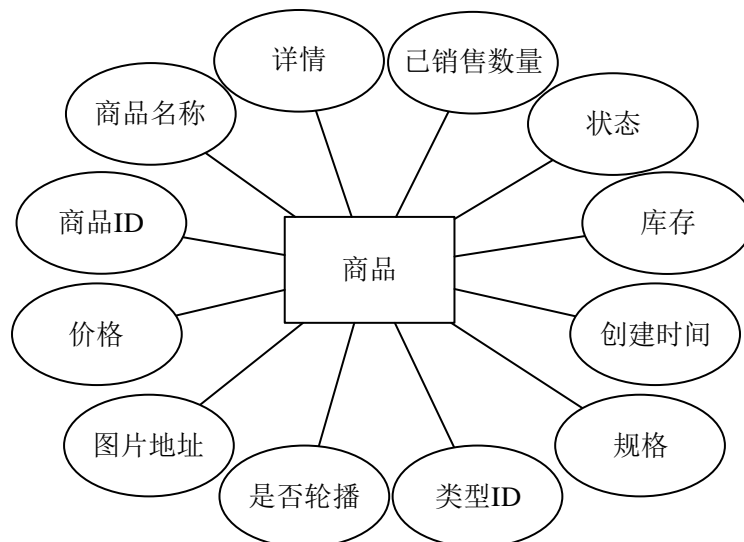


图 3.14 商品实体图

（7）订单实体集：包括的属性有订单 ID、会员 ID、管理员 ID、订单状态、创建时间、自提点 ID、订单结算时间、订单总价。订单实体图，如图 3.15 所示。

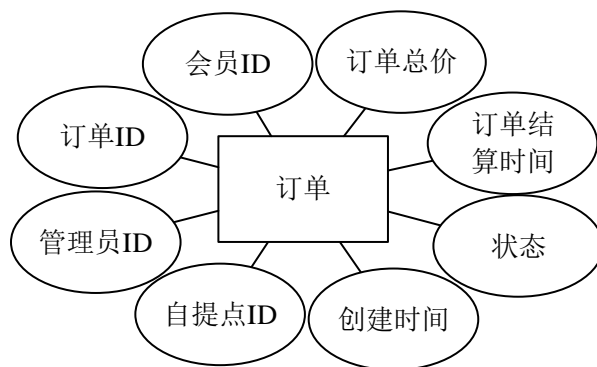


图 3.15 订单实体图

(8) 佣金实体集：包括的属性有佣金 ID、订单 ID、金额、佣金状态。佣金实体图，如图 3.16 所示。

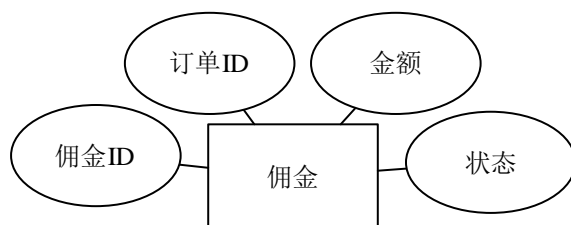


图 3.16 佣金实体图

(9) 提现记录实体集：提现记录 ID、管理员 ID、金额、提现状态。提现记录实体图，如图 3.17 所示。

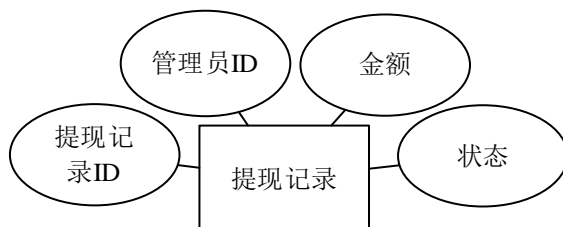


图 3.17 提现记录实体图

系统局部 E-R 关系分析如下：

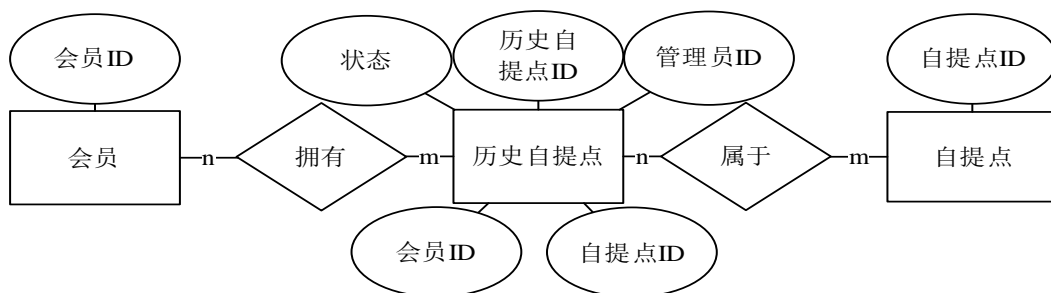


图 3.18 会员、历史自提点与自提点 E-R 图

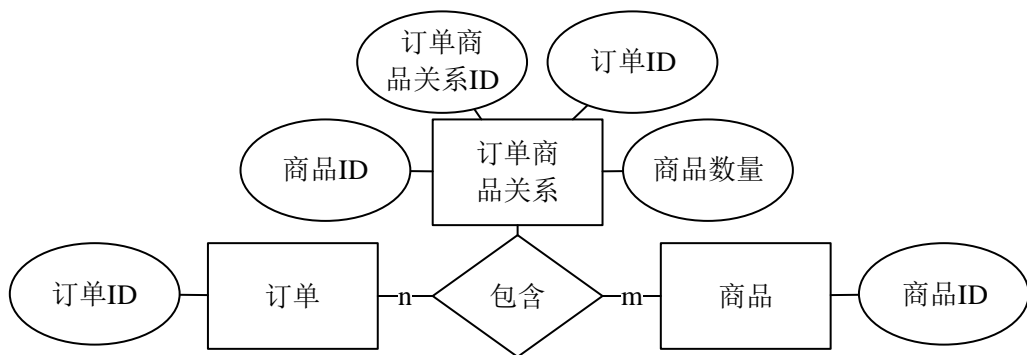


图 3.19 订单与商品 E-R 图

一个会员可以有一个或多个历史自提点，其两者之间的关系是多对多的，一个自提点可以成为多个会员的历史自提点，其两者之间的关系是多对多的。由此，可以得到会员、历史自提点与自提点的 E-R 图，如图 3.18 所示。一笔订单可以包含一个或多个商品，其两者之间的关系是一对多的。由此，可以得到订单与商品的 E-R 图，如图 3.19 所示。合并局部 E-R 图，得出社区团购系统总体 E-R 图，如图 3.20 所示。

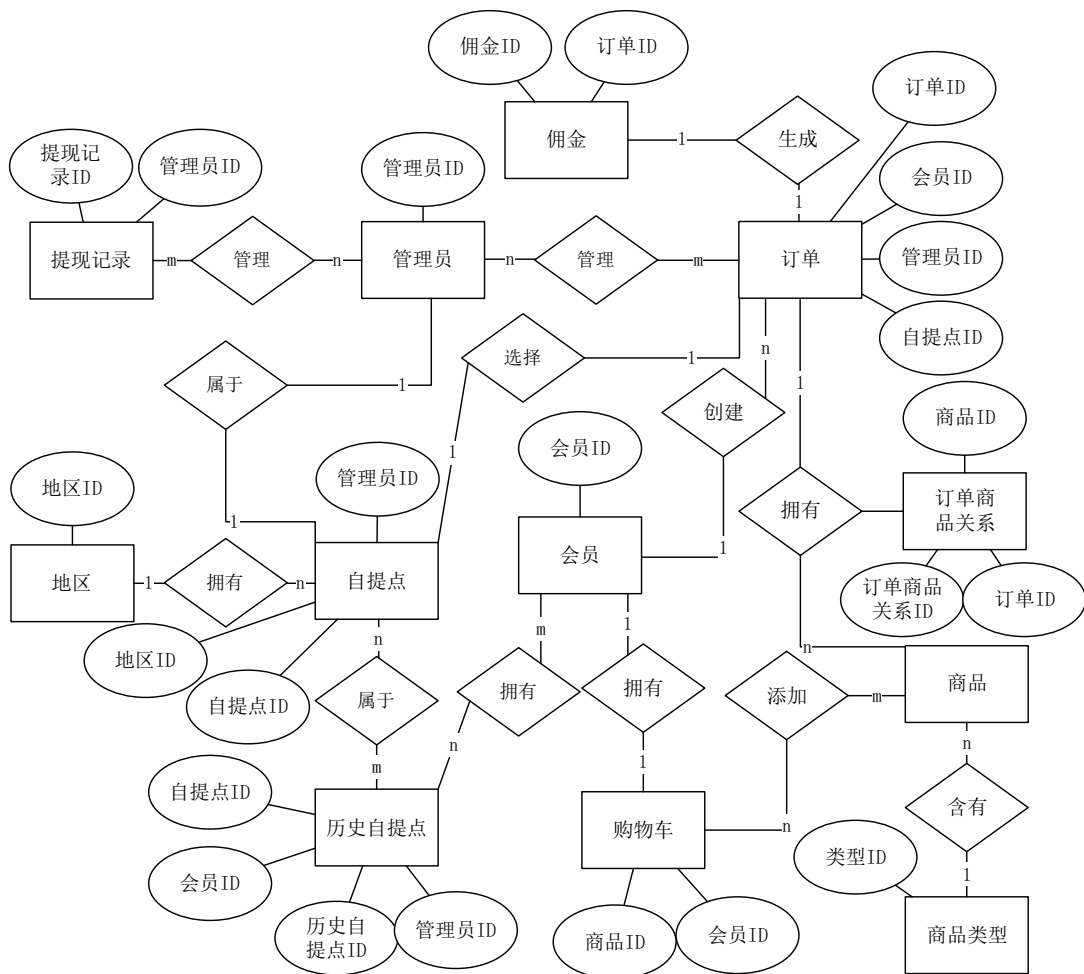


图 3.20 系统总体 E-R 图

3.4.2 数据库逻辑设计

根据以上实体关系图，设计的系统数据库关系模型如下：

会员(会员 ID、昵称、登录密码、手机号、账户头像地址)。会员的 PK 为会员 ID。

自提点(自提点 ID、自提点名称、所属地区 ID、管理员 ID、自提点状态、自提点详细地址)。自提点的 PK 为自提点 ID，FK 为所属地区 ID、管理员 ID。

商品类型(商品类型 ID、类型名称、简介、商品类型状态、创建时间、修改时间)。商品类型的 PK 为类型 ID。

管理员(管理员 ID、昵称、手机号、登录密码、余额、账户头像地址、管理员类型、管理员状态)。管理员的 PK 为管理员 ID。

地区(地区 ID、省级行政区名)。地区的 PK 为地区 ID。

商品(商品 ID、商品名称、价格、库存、已售数量、图片地址、详情、是否轮播、类型 ID、规格、状态、创建时间)。商品的 PK 为商品 ID，FK 为类型 ID。

订单(订单 ID、会员 ID、自提点 ID、管理员 ID、订单状态、创建时间、订单结算时间、订单总价)。订单的 PK 为订单 ID，FK 为会员 ID、自提点 ID、管理员 ID。

订单商品关系(订单商品关系 ID、订单 ID、商品 ID、商品数量)。订单商品关系的 PK 为订单商品关系 ID，FK 为订单 ID、商品 ID。

购物车(购物车 ID、订单 ID、商品 ID、商品数量)。购物车的 PK 为购物车 ID，FK 为订单 ID、商品 ID。

历史自提点(历史自提点 ID、管理员 ID、会员 ID、历史自提点状态、自提点 ID)。历史自提点的 PK 为历史自提点 ID，FK 为管理员 ID、会员 ID、自提点 ID。

佣金(佣金 ID、订单 ID、金额、佣金状态)。佣金的 PK 为佣金 ID，FK 为订单 ID。

提现记录(提现记录 ID、管理员 ID、金额、申请时间、提现状态)。提现记录的 PK 为提现记录 ID，FK 为管理员 ID。

3.4.3 数据库物理实现

根据数据库概念和逻辑分析，系统共设计 12 张数据表，系统数据表设计如下。

(1) 管理员表命名为 user，主要包括管理员 ID、昵称、手机号、登录密码、余额等信息，详细设计如表 3.1 所示。

表 3.1 管理员表(user)

字段名	字段类型	字段说明	是否为空	约束条件
user_id	int	管理员 ID	Not null	主键

(续表)

字段名	字段类型	字段说明	是否为空	约束条件
user_phone	varchar(11)	手机号	Not null	唯一键
user_password	varchar(32)	登陆密码	Not null	
user_portrait	varchar(255)	账户头像地址		
user_type	thinint	管理员类型	Not null	默认值: 0
user_name	varchar(50)	昵称	Not null	
user_leave_amount	decimal(10,2)	余额		
user_status	thinint	管理员状态	Not null	默认值: 0

(2) 会员表命名为 **member**，主要包括会员 ID、手机号、登录密码等信息，详细设计如表 3.2 所示。

表 3.2 会员表(member)

字段名	字段类型	字段说明	是否为空	约束条件
m_id	int	会员 ID	Not null	主键
m_phone	varchar(11)	手机号	Not null	唯一键
m_name	varchar(30)	昵称	Not null	
m_password	varchar(32)	登录密码	Not null	
m_portrait	varchar(255)	账户头像地址		

(3) 商品类型表命名为 **classify**，主要包括商品类型 ID、类型名称、简介、商品类型状态等信息，详细设计如表 3.3 所示。

表 3.3 商品类型表(classify)

字段名	字段类型	字段说明	是否为空	约束条件
cls_id	int	商品类型 ID	Not null	主键
cls_name	varchar(50)	类型名称	Not null	
cls_synopsis	varchar(255)	简介	Not null	
cls_status	thinint	商品类型状态	Not null	默认值: 0
cls_create_time	datetime	创建时间	Not null	
cls_update_time	datetime	修改时间		

(4) 商品表命名为 **goods**，主要包括商品 ID、商品名称、价格、库存、已售数量、图片地址、详情、是否轮播、规格、状态等信息，详细设计如表 3.4 所示。

表 3.4 商品表(goods)

字段名	字段类型	字段说明	是否为空	约束条件
g_id	int	商品 ID	Not null	主键
g_name	varchar(30)	商品名称	Not null	
g_price	decimal(10,2)	价格	Not null	
g_photos	varchar(255)	图片地址	Not null	
g_particulars	varchar(255)	详情	Not null	
g_inventory	int	库存	Not null	
g_sales	varchar(30)	已售数量	Not null	
g_carousel	tinyint	是否轮播	Not null	默认值：0
cls_id	int	类型 ID	Not null	外键
g_spec	varchar(25)	规格	Not null	
g_status	tinyint	状态	Not null	默认值：0
g_create_time	datetime	创建时间	Not null	

(5) 自提点表命名为 addeess，主要包括自提点 ID、所属地区 ID、管理员 ID、自提点状态、自提点详细地址等信息，详细设计如表 3.5 所示。

表 3.5 自提点表(address)

字段名	字段类型	字段说明	是否为空	约束条件
ad_id	int	自提点 ID	Not null	主键
ad_name	varchar(25)	自提点名称	Not null	
user_id	int	管理员 ID	Not null	外键
ad_detailed	varchar(25)	自提点详细地址	Not null	
reg_id	int	所属地区 ID	Not null	外键
ad_status	tinyint	自提点状态	Not null	默认值：0

(6) 历史自提点表命名为 history，主要包括历史自提点 ID、管理员 ID、会员 ID、历史自提点状态等信息，详细设计如表 3.6 所示。

表 3.6 历史自提点表(history)

字段名	字段类型	字段说明	是否为空	约束条件
his_id	int	历史自提点 ID	Not null	主键
user_id	int	管理员 ID	Not null	外键
m_id	int	会员 ID	Not null	外键
his_status	tinyint	状态	Not null	默认值：0
ad_id	int	自提点 ID	Not null	外键

(7) 订单表命名为 **order**，主要包括订单 ID、会员 ID、管理员 ID、订单结算时间、订单总价等信息，详细设计如表 3.7 所示。

表 3.7 订单表(order)

字段名	字段类型	字段说明	是否为空	约束条件
o_id	varchar(20)	订单 ID	Not null	主键
user_id	int	管理员 ID	Not null	外键
m_id	int	会员 ID	Not null	外键
o_status	tinyint	订单状态	Not null	默认值：0
o_create_time	datetime	创建时间	Not null	
o_settlement_time	datetime	订单结算时间		
o_amount_total	decimal(10,2)	订单总价	Not null	
ad_id	int	自提点 ID	Not null	外键

(8) 订单商品关系表命名为 **order_goods**，包括订单商品关系 ID、订单 ID、商品 ID、商品数量等信息，详细设计如表 3.8 所示。

表 3.8 订单商品关系表(order_goods)

字段名	字段类型	字段说明	是否为空	约束条件
o_g_id	int	订单商品关系 ID	Not null	主键
o_id	varchar(20)	订单 ID	Not null	外键
g_id	int	商品 ID	Not null	外键
o_g_number	int	商品数量	Not null	

(9) 购物车表命名为 **shop_cart**，包括购物车 ID、商品 ID、商品数量等信息，详细设计如表 3.9 所示。

表 3.9 购物车表(shop_cart)

字段名	字段类型	字段说明	是否为空	约束条件
cart_id	int	购物车 ID	Not null	主键
o_id	varchar(20)	订单 ID	Not null	外键
m_id	int	会员 ID	Not null	外键
cart_number	int	商品数量	Not null	

(10) 佣金表命名为 **commission**，包括佣金 ID、订单 ID、金额、佣金状态等信息，详细设计如表 3.10 所示。

表 3.10 佣金表(commission)

字段名	字段类型	字段说明	是否为空	约束条件
comm_id	int	佣金 ID	Not null	主键
o_id	varchar(20)	订单 ID	Not null	外键
m_id	int	会员 ID	Not null	外键
comm_amount	decimal(10,2)	金额	Not null	
comm_status	tinyint	佣金状态	Not null	默认值：0

(11) 地区表命名为 region，包括地区 ID、省级行政区名等信息，详细设计如表 3.11 所示。

表 3.11 地区表(region)

字段名	字段类型	字段说明	是否为空	约束条件
reg_id	int	地区 ID	Not null	主键
reg_name	varchar(10)	省级行政区名	Not null	

(12) 提现记录表命名为 withdrawal，包括提现记录 ID、管理员 ID、金额、提现状态等信息，详细设计如表 3.12 所示。

表 3.12 提现记录表(withdrawal)

字段名	字段类型	字段说明	是否为空	约束条件
w_id	int	提现记录 ID	Not null	主键
user_id	int	管理员 ID	Not null	外键
w_amount	decimal(10,2)	金额	Not null	
w_create_time	datetime)	申请时间	Not null	
w_status	tinyint	提现状态	Not null	默认值：0

3.5 本章小结

本章从系统实际需求出发，对系统进行总体和详细设计，根据系统的主要功能模块划分画出了系统功能模块图，以流程图和时序图的形式详细分析了系统的主要业务流程以及 Token 登录鉴权时序，最后根据功能模块分析出数据库实体及每个实体的属性，画出 E-R 实体联系图，并将数据库表进行物理实现，为接下来系统的具体实现奠定了坚实的基础。

4 系统实现

4.1 用户端会员功能

4.1.1 注册登录功能

游客在注册页面输入手机号注册，后端会判断手机号是否已注册，如果手机号已注册，系统会给出相应提示：手机号已注册，请直接登录或找回密码！通过点击直接登录和忘记密码按钮，系统会跳转至相应页面。注册页面 UI 效果，如图 4.1 所示。注册提示页面 UI 效果，如图 4.2 所示。



图 4.1 注册页面

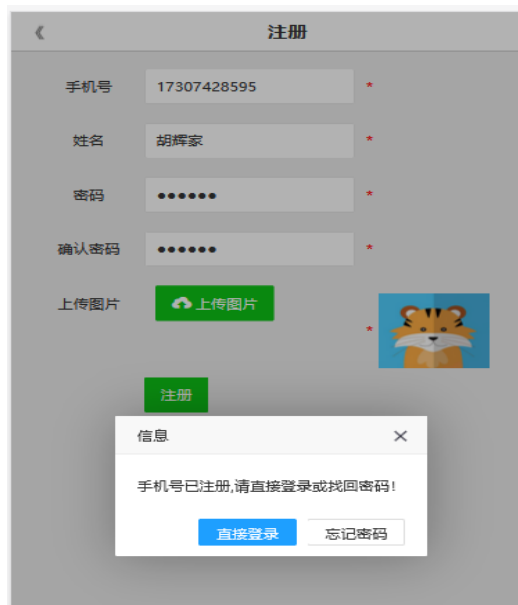


图 4.2 注册提示页面

前端通过 Ajax 的 Post 请求方式，请求地址为/member/add 的注册接口，传递 Json 格式的数据，进入控制层 add()方法中，在该方法中调用了业务实现层的 insert()方法，在 insert()方法中通过判断数据库返回值，返回相应的响应结果。注册功能实现的关键代码如下。

```
@PostMapping("/add")
public ResultDTO<String> add(@RequestBody UserDTO userDTO) {
    return memberService.insert(userDTO);
}

@Override
public ResultDTO<String> insert(UserDTO userDTO) {
    int result = memberRepository.insert(member);
    if (result > 0) {
```

```

        return new ResultDTO<>(ResultStatus.SUCCESS, Null);
    } else
        return new ResultDTO<>(ResultStatus.ERROR, Null);
    }

```

游客注册成为平台会员，系统会自动跳转至登录页面。在登录页面，如果生成的验证码图片过于杂乱，导致用户无法正确读取，可以点击图片切换一个验证码。会员输入手机号、密码和验证码进行登录操作，如果用户未进行注册，可免费注册。如果会员忘记密码，可通过手机号、阿里云短信验证码找回密码。登录页面 UI 效果，如图 4.3 所示。

图 4.3 登录页面

前端通过 Ajax 的 Post 请求方式，请求地址为/member/login 的登录接口，传递 Json 格式的数据，进入控制层 login()方法中，在该方法中调用了业务实现层的 login()方法。在控制层 login()方法中通过判断业务实现层 login()方法的返回值，返回相应的响应结果。登录功能实现的关键代码如下。

```

@PostMapping("/login")
public ResultDTO<LoginMDTO> login(@RequestBody Member member,
HttpServletRequest request, HttpSession session) {
    LoginMDTO dto = memberService.login(member, request, session);
    if (dto.getIsLogin().equals("1004")) {
        return new ResultDTO<>(ResultStatus.AUTHENTICATION, Null);
    }
}

```

```

    } else if (dto.getIsLogin().equals("1003")) {
        return new ResultDTO<>(ResultStatus.AUTH_FAIL, Null);
    } else if (dto.getIsLogin().equals("1006")) {
        return new ResultDTO<>(ResultStatus.CODE_FAIL, Null);
    } else
        return new ResultDTO<>(dto);
}

```

会员输入注册的手机号、新密码、确认密码、短信验证码（两次获取验证码的时间间隔为 60s），最后点击提交。找回密码成功后，页面会自动跳转至登录页面。获取验证码页面 UI 效果，如图 4.4 所示。重新获取验证码页面 UI 效果，如图 4.5 所示。

图 4.4 获取验证码页面

图 4.5 重新获取验证码页面

前端通过 Ajax 的 Post 请求方式，请求地址为/member/SmsCode 的短信发送接口，传递 Json 格式的数据。在该接口中首先生成一个六位数的随机验证码，通过阿里云短信 API 将短信验证码发送到相应的手机号中，随后通过判断短信是否发送成功，返回相应的响应结果。调用阿里云 API 发送短信验证码的核心代码如下。

```

@PostMapping("/SmsCode")
public ResultDTO<String> SmsCode(@RequestBody SmsParams smsParams) {
    String code = RandomCode.RandomCode();
    boolean sendFlag=sendSmsService.sendSms(smsParams.getPhone(), code);
    if (sendFlag) {
        this.tokenService.saveCode(smsParams.getPhone(),code);
        return new ResultDTO<>(ResultStatus.SUCCESS, code);
    } else {

```

```

        return new ResultDTO<>(ResultStatus.ERROR, Null);
    }
}

```

用户收到短信验证码后，填写至前端页面，前端通过 Ajax 的 Post 请求方式，请求地址为/member/update 的修改密码接口，传递 Json 格式的数据。在该接口中调用业务层的 update()方法，将密码使用 MD5 加密，进行数据库数据修改操作，随后通过判断修改是否成功，返回相应的响应结果。找回密码功能实现的关键代码如下。

```

@PostMapping("/update")
public ResultDTO<String> update(@Valid @RequestBody Member member) {
    return memberService.update(member);
}

@Override
public ResultDTO<String> update(Member member) {
    member.setPwd(MD5Util.string2MD5(member.getPwd()));
    int result = memberRepository.updateDynamic(member);
    if (result > 0) {
        return new ResultDTO<>(ResultStatus.SUCCESS, Null);
    }
    return new ResultDTO<>(ResultStatus.ERROR, Null);
}

```

4.1.2 首页功能

管理端添加商品设置了首页轮播的商品可以在首页轮播（按时间倒序轮播 4 个商品），其余商品按时间倒序列显示。系统会根据销售量选择三个最畅销的商品进行推荐，点击轮播图、热卖推荐和商品展示的图文可以进入商品详情页面，了解商品详情，进行商品购买。会员登录成功后，还可在首页进行加入购物车、购物车商品数量增减操作。首页页面 UI 效果，如图 4.6 所示。

前端通过 Ajax 的 Get 请求方式，请求地址为/goods/findGoods 的上架状态全部商品接口。在该接口中调用业务层的 findWithResult()方法，进行数据库数据查询，返回相应的响应结果。首页功能实现的关键代码如下。

```

@GetMapping("/findGoods")
public ResultDTO<List<Goods>> goodsFindAll(ShopcartDTO shopcartDTO) {
    return goodsService.findWithResult(shopcartDTO);
}

```

@Override

```
public ResultDTO<List<Goods>> findWithResult(ShopcartDTO shopcartDTO){  
    List<Goods> goodsList=goodsRepository.findPageWithResult(pageDTO);  
    return new ResultDTO<>(ResultStatus.SUCCESS,goodsList);  
}
```



图 4.6 首页页面

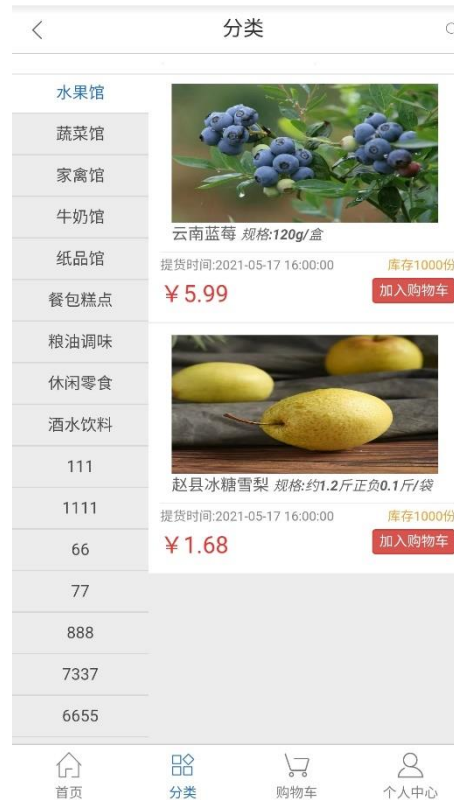


图 4.7 商品分类展示页面

4.1.3 分类展示功能

每个商品均有一个所属分类，点击相应的分类可以查看相应的商品。商品分类展示页面 UI 效果，如图 4.7 所示。

前端通过 Ajax 的 Get 请求方式，请求地址为/goods/findGoodsWithClassify 的某一分类上架状态商品接口。在该接口中调用业务层的 findGoodsWithClassify()方法，进行数据库商品和购物车数据查询，通过对比购物车商品是否包含某一商品，返回相应的响应结果。商品分类展示功能实现的关键代码如下。

```
@GetMapping("/findGoodsWithClassify")  
public ResultDTO<List<Goods>> findGoodsWithClassify(ShopcartDTO shopcartD  
TO) {  
    return goodsService.findGoodsWithClassify(shopcartDTO);  
}
```

```

public ResultDTO<List<Goods>> findGoodsWithClassify(ShopcartDTO shopcartD
TO) {
    PageDTO pageDTO=new PageDTO();
    int totalCount =goodsRepository.findPageWithCount(pageDTO);
    pageDTO.setPage(0);
    pageDTO.setLimit(totalCount);
    Goods goods=new Goods();
    goods.setClsId(shopcartDTO.getClsid());
    List<Goods> goodsList1=goodsRepository.findWithResult(shopcartDTO);
    if (shopcartDTO.getMid()!=Null){
        List<Goods> goodsList=goodsRepository.findGoodsWithCls(goods);
        if (goodsList.size()!=0){
            return new ResultDTO<>(ResultStatus.SUCCESS,goodsList);
        }
    }
}

```

4.1.4 查看商品详情功能

通过点击首页、分类展示商品中的商品图片和商品名称等，可以跳转到商品详情页面，查看相应的商品详情。在商品详情页面可以进行加入购物车，去购物车结算，回到用户端首页等操作。商品详情页面 UI 效果，如图 4.8 所示。

前端通过 Ajax 的 Post 请求方式，请求地址为/goods/findById 的某一上架状态商品接口。在该接口中调用业务层的 findById()方法，进行数据库商品和购物车数据查询，通过对比购物车商品是否包含某一商品，返回相应的响应结果。查看商品详情功能实现的关键代码如下。

```

@PostMapping("/findById")
public ResultDTO<Goods> findById(@RequestBody Goods goods) {
    return goodsService.findById(goods.getId());
}

public ResultDTO<Goods> findById(Integer id) {
    Goods goods=goodsRepository.findById(id);
    if (goods!=Null) {
        return new ResultDTO<>(ResultStatus.SUCCESS, goods);
    }
}

```




图 4.8 商品详情页面

4.1.5 搜索功能功能

用户可输入关键字，点击搜索按钮，系统会检索出包含该关键字的商品信息。

商品搜索页面实现的关键代码与首页商品展示的请求接口一致，在此不在累述，只需在请求参数中加入关键字即可，请求参数示例：{"mid":"1001","keywords":"云南"}，mid 和 keywords 皆可以省略。商品搜索页面 UI 效果，如图 4.9 所示。



图 4.9 商品搜索页面



图 4.10 购物车页面

4.1.6 购物车功能

会员可以在相关页面（如首页）中将想要购买的商品加入购物车，在这些页面中也可进行加减数量和删除商品操作。商品添加到购物车后，根据加减图标可以更改商品数量，选择部分或全选商品可以删除购物车商品、生成订单结算购物车商品。购物车页面 UI 效果，如图 4.10 所示。

前端通过 Ajax 的 Get 请求方式，请求地址为/shopcart/findGoodsWithCart 的上架状态购物车商品接口。在该接口中调用业务层的 findWithResult()方法，进行数据库购物车数据查询，返回相应的响应结果。购物车功能实现的关键代码如下。

```
@GetMapping("/findGoodsWithCart")
public ResultDTO<List<Goods>> findGWithCart(ShopcartDTO shopcartDTO){
    return shopcartService.findWithResult(shopcartDTO)
}

public ResultDTO<List<Goods>> findWithResult(ShopcartDTO cartDTO){
    List<Goods> goodsList = goodsRepository.findWithResult(cartDTO);
    if(goodsList!=Null){
        return new ResultDTO<>(ResultStatus.SUCCESS, goodsList);
    }else    return new ResultDTO<>(ResultStatus.ERROR, Null);
}
```



图 4.11 订单详情页面



图 4.12 扫码付款页面

4.1.7 订单结算功能

购物车选中商品去结算,进入订单结算详情。自提点管理员在订单结算详情页面,自动获取会员的姓名,手机号,自提点名,详细地址,手机号等信息并显示。点击立即支付(使用支付宝沙箱支付,支付成功后回调跳转至订单详情页,订单状态变为待提货)。如果点击取消支付订单状态变为已取消。订单详情页面 UI 效果,如图 4.11。扫码付款页面 UI 效果,如图 4.12 所示。

pay 接口需要传入一个订单号,通过该订单号从数据库中查出订单相关信息,设置支付宝请求的相关参数,如 App_id、支付宝网关、应用私钥、支付宝公钥、编码格式、以及回调地址(回调地址需外网可访问)等,将查询出的订单信息写入支付宝接口请求的参数中,请求支付宝接口,根据请求返回的结果绘制出二维码,会员通过二维码扫码付款,付款成功后,支付宝会自动请求回调地址。支付功能实现的关键代码如下。

```
@RequestMapping("/pay")
private Map pay (HttpServletRequest responses,Order order) throws AlipayApi
Exception {
    AlipayClient alipayClient = new DefaultAlipayClient(AlipayConfig.ServiceUrl,
AlipayConfig.app_id, AlipayConfig.private_key,"json", AlipayConfig.input_charset, Ali
payConfig.zfb_public_key, "RSA2");

    AlipayTradePrecreateRequest request = new AlipayTradePrecreateRequest();
    request.setNotifyUrl("http://3tubp.natapp.cc/shop/alipay/aliPayCallBack");
    Map<String, Object> maps = new HashMap<>();
    maps.put("out_trade_no", order.getId());
    maps.put("total_amount", order.getGoodAmountTotal());
    request.setBizContent(postdata);
    AlipayTradePrecreateResponse response = alipayClient.execute(request);
    String body = response.getBody();
    JSONObject jsonObject = JSONObject.parseObject(body);
    String qr_code = jsonObject.getJSONObject("alipay_trade_precreate_response").
getString("qr_code");
    ServletOutputStream sos = responses.getOutputStream();
    QR qr = new QR();
    qr.makeQRCode2(qr_code, sos);
    return map;
}
```

支付宝回调调用接口，在该接口中需要接收支付宝支付传过来的参数，并将这些参数循环遍历用键值对的形式保存在 Map 集合中，然后判断是否支付成功，如果支付成功就调用 `orderService.batchUpdate()` 方法修改订单状态，最后跳转至订单详情页。支付回调功能实现的关键代码如下。

```
@RequestMapping("/aliPayCallBack")
public String aliPayCallBack(HttpServletRequest response, HttpServletRequest request) {
    boolean flag = AlipaySignature.rsaCheckV1(params, AlipayConfig.zfb_public_key, AlipayConfig.input_charset, "RSA2");
    String order_nio = request.getParameter("out_trade_no");//获取订单号
    if (flag == true) {
        BatchDTO batchDTO = new BatchDTO();
        batchDTO.setOids(order_nio);
        batchDTO.setStatus(1);
        this.orderService.batchUpdate(batchDTO);
    }
    return "redirect:/index/order_info";
}
```

4.1.8 选择自提点功能

如果是新会员，当前自提点和历史自提点为空，点击添加自提点，显示所有的门店，点击进店，将选中的门店设置为当前的自提点，非首次设置自提点时，历史自提点有相关数据，点击进店，将选中的门店设置为当前自提点。历史自提点页面 UI 效果，如图 4.13 所示。自提点列表页面 UI 效果，如图 4.14 所示。

前端通过 Ajax 的 Get 请求方式，请求地址为 `/history/ByPage` 的历史自提点列表接口。在该接口中调用业务层的 `findPageWithResult()` 方法，进行数据库历史自提点数据查询，返回相应的响应结果。选择自提点功能实现的关键代码如下。

```
@GetMapping("/ByPage")
public ResultDTO historyByPage(PageDTO pageDTO) {
    return historyService.findPageWithResult(pageDTO);
}

public ResultDTO findPageWithResult(PageDTO pageDTO) {
    int totalCount = historyRepository.findPageWithCount(pageDTO);
    List<History> userList = historyRepository.findPageWithResult(pageDTO);
}
```

```

if(userList!=Null){
    ResultDTO resultDTO=new ResultDTO();
    resultDTO.setCode(0);
    resultDTO.set Count(totalCount);
    resultDTO.setMsg("请求成功");
    resultDTO.setData(userList);
}
return resultDTO;
}

```



图 4.13 历史自提点页面



图 4.14 自提点列表页面

4.1.9 订单管理功能

点击个人中心的全部订单（待支付、待提货、已提货）进入订单管理页面。显示不同状态的订单（待支付、待提货，待付款，订单已取消等）。点击订单状态跳转至订单详情页，点击商品跳转至商品详情页。每个订单列表显示商品信息（商品图片、名称、规格单位、价格、提货时间、订单状态、价格等）。全部订单页面 UI 效果，如图 4.15 所示。待提货订单页面 UI 效果，如图 4.16 所示。

前端通过 Ajax 的 Get 请求方式，请求地址为/order/findByIdResult 的订单详情接口。在该接口中调用业务层的 findByIdResult()方法，进行数据库订单数据查询，返回

相应的响应结果。订单管理功能实现的关键代码如下。

```
@GetMapping("/findByIdResult")
public ResultDTO<List<Order>> findByIdResult(PageDTO pageDTO,) {
    return orderService.findByIdResult(pageDTO);
}

public ResultDTO<List<Order>> findByIdResult(PageDTO pageDTO) {
    int totalCount = orderRepository.findPageWithCount(pageDTO);
    List<Order> orderList=orderRepository.findPageWithResult(pageDTO);
    for (Order order : orderList) {
        OrderG orderG = new OrderG();
        orderG.setOid(order.getOid());
        order.setGoodsDTOList(orderGRepository.findGWithOrder(orderG));
    }
    return new ResultDTO<>(ResultStatus.SUCCESS,orderList);
}
```



图 4.15 全部订单页面

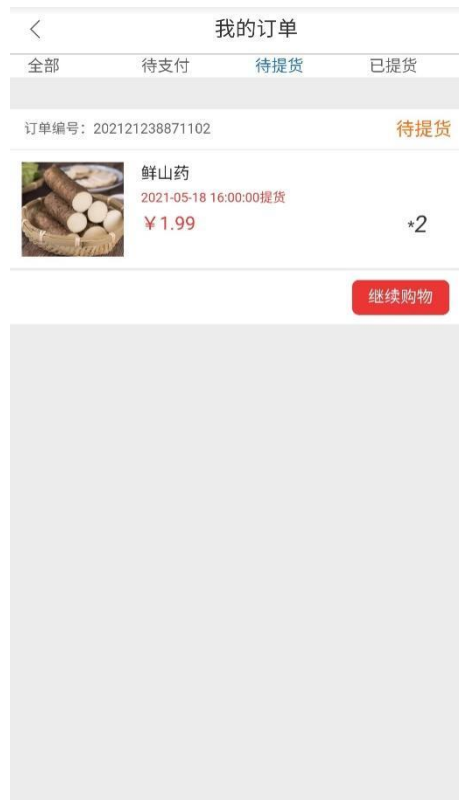


图 4.16 全部订单页面

4.1.10 会员个人中心功能

会员个人中心页面显示个人信息（个人头像、昵称等）、我的订单（全部订单、

待支付、待提货、已提货)、自提地址、商家中心、门店入驻和设置。如果没登录则提示请登录, 点击进入登录页面。会员个人中心页面 UI 效果, 如图 4.17 所示。

前端通过 Ajax 的 Get 请求方式, 请求地址为/member/findById 的会员个人信息详情接口。在该接口中调用业务层的 findById()方法, 进行数据库会员数据查询, 返回相应的响应结果。会员个人中心功能实现的关键代码如下。

```
@GetMapping("/findById")
public ResultDTO<Member> findById(Member member) {
    return memberService.findById(member.getId());
}

public ResultDTO<Member> findById(Integer id) {
    Member member = memberRepository.selectById(id);
    return new ResultDTO<>(ResultStatus.SUCCESS, member);
}
```



图 4.17 会员个人中心页面

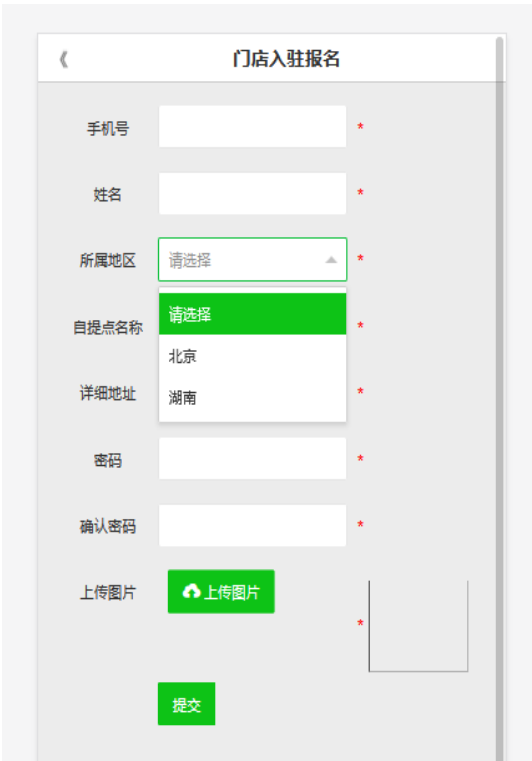


图 4.18 门店入驻报名页面

(1) 自提地址

显示自提地址, 进入切换自提点页面, 点击进店后, 成功切换自提点, 详细功能同 4.1.8 小节中的选择自提点。

(2) 门店入驻

点击门店入驻，进入门店入驻报名页面，用户输入相关信息，点击提交耐心等待审核，审核成功后，可以使用手机号和密码进入商家中心登录。申请成为自提点页面 UI 效果，如图 4.18 所示。

前端通过 Ajax 的 Post 请求方式，请求地址为/user/add 的门店入驻报名接口，传递 Json 格式的数据，在该接口中调用业务层的 add()方法，新增数据库自提点管理员，返回相应的响应结果。门店入驻报名功能实现的关键代码如下。

```
@PostMapping("/add")
public ResultDTO<String> add(@RequestBody UserDTO userDTO) {
    return userService.insert(userDTO);
}
```

（3）设置

点击设置进入设置页面，点击修改密码进入忘记密码页面，点击退出当前账户，清除系统的缓存信息，跳转至系统首页。设置页面 UI 效果，如图 4.19 所示。



图 4.19 设置页面

（4）商家中心

点击商家中心，进入自提点登录页面，管理员输入手机号和密码，进行登录操作，如果管理员将设置的登录密码忘记，可点击忘记密码进入找回密码页面，功能同 4.1.1 小节中的登录功能的实现。

4.2 用户端管理员功能

4.2.1 登录注册功能

管理员账号的注册需要等待后台审核通过后才能进行登录、找回密码等操作。登录、找回密码等操作具体实现，同 4.1.1 小节中的注册登录功能的实现。

4.2.2 管理员个人中心功能

个人中心页面显示个人信息（个人头像、昵称、余额等）、我的订单（全部订单、待支付、待提货、已提货）、佣金记录、提现记录、设置和提现等操作，其中订单管理功能的实现同 4.1.9 小节中的订单管理功能，设置功能同 4.1.10 小节中会员个人中心的设置功能。管理员个人中心页面 UI 效果，如图 4.20 所示。

前端通过 Ajax 的 Get 请求方式，请求地址为/user/findById 的管理员个人信息详情接口。在该接口中调用业务层的 findById()方法，进行数据库自提点管理员数据查询，返回相应的响应结果。管理员个人中心功能实现的关键代码如下。

```
@GetMapping("/findById")
public ResultDTO<User> findById(@Valid User user){
    User user1=userService.findById(user.getId());
    return new ResultDTO<>(ResultStatus.SUCCESS,user1);
}

public User findById(Integer id) {
    return userRepository.selectByLeaveAmount(id);
}
```



图 4.20 管理员个人中心页面

佣金记录	
订单编号: 202120969167101 2021-05-15 20:58:14	+0.5
订单编号: 202122270829101 2021-05-29 14:49:55	+0.5
订单编号: 202122426965101 2021-05-31 14:18:18	+0.5
订单编号: 202122270829101 2021-05-29 14:49:55	+0.5

图 4.21 佣金记录页面

4.2.3 查看佣金记录功能

点击管理员个人中心页面中的佣金记录，进入佣金记录页面，该页面显示当前登录管理员的所属订单佣金。佣金记录页面 UI 效果，如图 4.21 所示。

前端通过 Ajax 的 Get 请求方式，请求地址为/commission/findCommissionByUid 的管理员佣金列表接口。在该接口中调用业务层的 selectByuId()方法，进行数据库佣

金数据查询，返回相应的响应结果。查看佣金记录功能实现的关键代码如下。

```
@GetMapping("/findCommissionByUid")
public ResultDTO<List<CommissionDTO>> findByCommission(User user) {
    List<CommissionDTO> commList=commissionService.findById(user.getId());
    return new ResultDTO<>(ResultStatus.SUCCESS,commDTOList);
}
```

4.2.4 提现申请功能

点击管理员个人中心页面中的提现按钮，进入提现申请页面，该页面显示当前登录管理员的手机号，账户余额，可以输入不大于账户余额的提现金额，点击提交按钮，等待后台管理端审核。提现申请页面 UI 效果，如图 4.22 所示。

前端通过 Ajax 的 Post 请求方式，请求地址为/withdrawal/add 的管理员佣金列表接口。在该接口中调用业务层的 insert()方法，新增提现申请，返回相应的响应结果。提现功能实现的关键代码如下。

```
@PostMapping("/add")
public ResultDTO<String> insert(@RequestBody Withdrawal withdrawal,) {
    return withdrawalService.insert(withdrawal);
}
```

提现	明细
支付宝账号: 17307428595	
您的当前账户余额: ¥2.4	
金额	0.4
<button>提交</button>	

图 4.22 提现申请页面

提现明细		
提现	2021-04-30 22:09:15	-0.4 审核失败
提现	2021-04-30 22:09:15	-0.7 待审核
提现	2021-04-30 22:09:15	-2 待审核
提现	2021-04-30 22:02:13	-2 成功
提现	2021-04-30 22:09:15	-0.5 成功

图 4.23 提现记录页面

4.2.5 查看提现明细功能

点击管理员个人中心页面中的提现明细，进入提现明细页面，该页面显示当前登录管理员所有的提现记录明细。提现明细页面 UI 效果，如图 4.23 所示。

前端通过 Ajax 的 Get 请求方式,请求地址为/withdrawal/findById 的管理员佣金列表接口。在该接口中调用业务层的 selectById()方法,进行数据库提现数据查询,返回相应的响应结果。查看提现明细功能实现的关键代码如下。

```
@GetMapping("/findById")
public ResultDTO<List<Withdrawal>> findById(Withdrawal withdrawal) {
    return withdrawalService.selectById(withdrawal);
}
```

4.3 管理端功能

4.3.1 综合管理功能

(1) 后台管理端登录功能

管理端登录功能的具体实现同 4.1.1 小节中的用户端会员登录功能。

(2) 修改个人信息功能

超级管理员在页面中输入想要修改的信息,点击修改,修改成功后系统会将弹出框关闭。管理端修改个人信息页面 UI 效果,如图 4.24 所示。

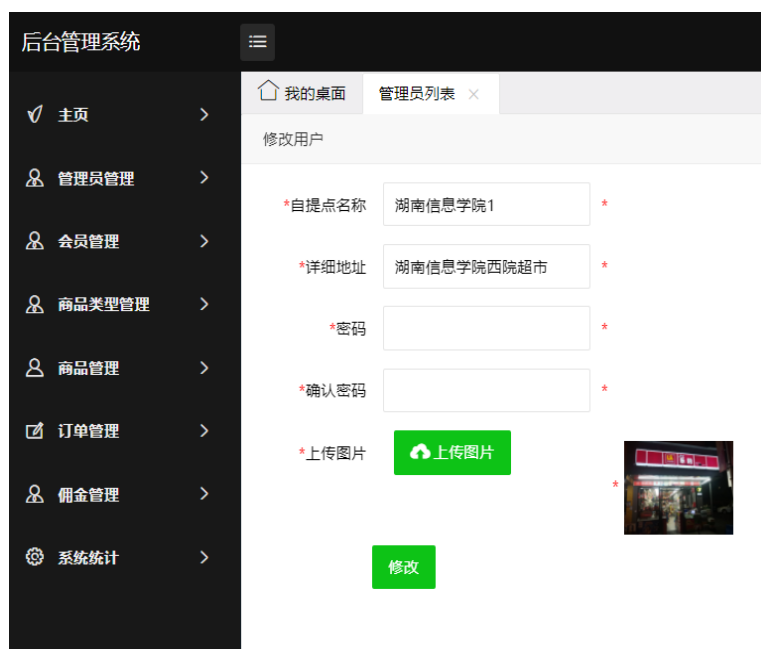


图 4.24 管理端修改个人信息页面

(3) 管理端控制台功能

超级管理员登录成功后,跳转至管理端首页,即控制台页面,在该页面显示部分系统统计的数据,如会员数、自提点数、在售商品数、总销售额、今日订单数和总订单数,下方显示近一个月内的每日销售额、订单数、提现金额折线图。管理端控制台页面 UI 效果,如图 4.25 所示。



图 4.25 管理端控制台页面

dataStatistics 接口为系统统计接口,在该接口中通过相应的方法查询了会员人数、自提点数量、在售商品数、总销售额、当日销售额等数据。管理端控制台功能实现的关键代码如下。

```
@GetMapping("/dataStatistics")
public ResultDTO<List<String>> dataStatistics() {
    List<String> dataStatistics = statisticsService.dataStatistics();
    return new ResultDTO<>(ResultStatus.SUCCESS, dataStatistics);
}
```

4.3.2 管理员信息管理功能

管理员信息管理主要是对自提点管理员信息进行管理,在管理员列表页面超级管理员能够查看每页 10 条的管理员信息分页展示,也可点击选择每页 20、30、40、50 等条切换每页显示条数,能够按照管理员名称对管理员信息进行模糊查询,能够进行添加管理员、修改管理员信息、批量禁用/启用、导出管理员信息 Excel 文件等操作。管理员列表页面 UI 效果,如图 4.26 所示。

在管理员管理的 ByPage 的接口中,通过 SQL 语句中 limit 关键字实现数据分页。管理员信息管理功能实现的关键代码如下。

```
@GetMapping("/ByPage")
public ResultDTO UserByPage(PageDTO pageDTO) {
    return userService.findPageWithResult(pageDTO);
}
```

```

public ResultDTO findPageWithResult(PageDTO pageDTO) {
    List<User> userList = userRepository.findPageWithResult(pageDTO);
    ResultDTO resultDTO = new ResultDTO();
    resultDTO.setCode(0);
    resultDTO.setCount(totalCount);
    resultDTO.setMsg("请求成功");
    resultDTO.setData(userList);
    return resultDTO;
}

```

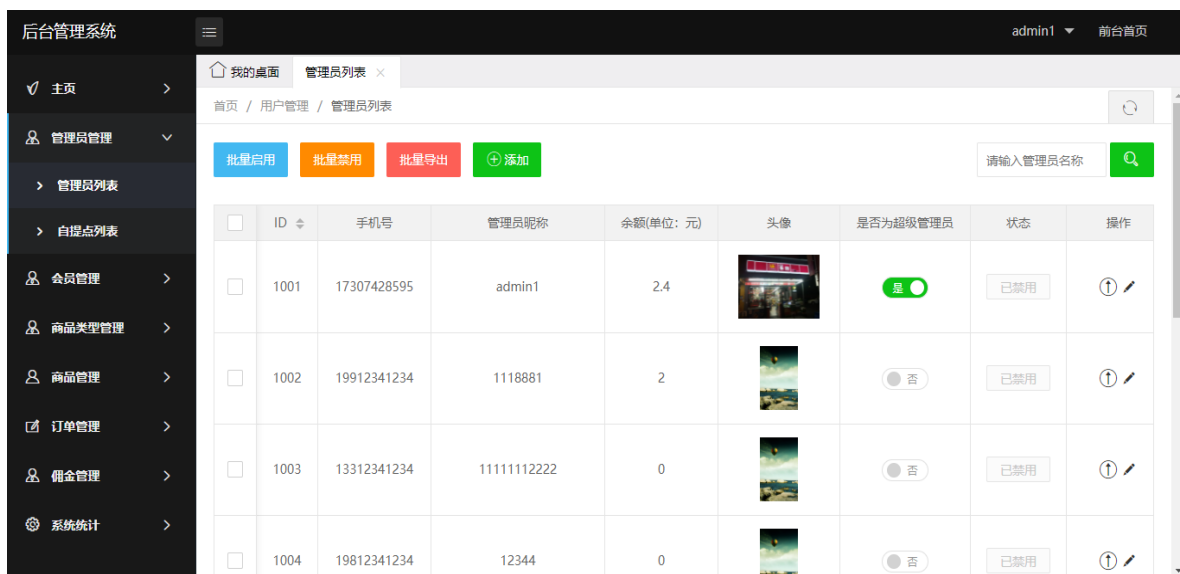


图 4.26 管理员列表页面

4.3.3 自提点信息管理功能

在自提点列表页面超级管理员能够查看每页 10 条的自提点信息分页展示，也可点击选择每页 20、30、40、50 等条切换每页显示条数，能够按照自提点名称对自提点信息进行模糊查询，能够对自提点进行批量禁用/启用、导出自提点信息 Excel 文件操作等。自提点列表页面 UI 效果，如图 4.27 所示。

在自提点管理的 batchUpdate 接口中，通过将 ids 数组转换成 address 对象的 List 集合，传递给 addressRepository.batchUpdate 方法，通过 Mybatis 的动态 SQL 语句实现批量更新。自提点管理功能实现的关键代码如下。

```

@PostMapping("/batchUpdate")
public ResultDTO<String> batchUpdate(@RequestBody BatchDTO batch) {
    return addressService.batchUpdate(batch);
}

```



图 4.27 自提点列表页面

4.3.4 会员管理功能

在会员列表页面超级管理员能够查看每页 10 条会员信息的展示，也可点击选择每页 20、30、40、50 等条切换每页显示条数，能够按照会员姓名对会员信息进行模糊查询，能够导出会员信息 Excel 文件。会员列表页面 UI 效果，如图 4.28 所示。

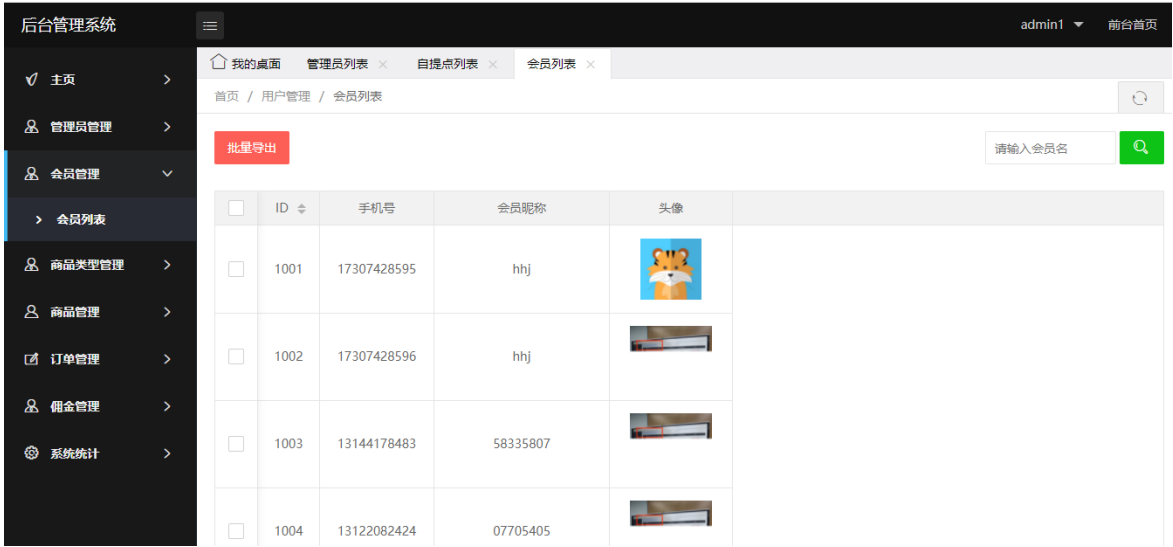


图 4.28 会员列表页面

在会员管理的 ByPage 的接口中，通过 SQL 语句中 limit 关键字实现数据分页。会员管理功能实现的关键代码如下。

```
@GetMapping("/ByPage")
public ResultDTO UserByPage(PageDTO pageDTO) {
    return memberService.findPageWithResult(pageDTO);
}
```

```

public ResultDTO findPageWithResult(PageDTO pageDTO) {
    List<Member> memberList=memberRepository.findPageWithResult(pageDTO);
    ResultDTO resultDTO=new ResultDTO();
    if (memberList.size()!=0) {
        resultDTO.setCode(0);
        resultDTO.setCount(totalCount);
        resultDTO.setMsg("请求成功");
        resultDTO.setData(memberList);
        return resultDTO;
    }else {
        resultDTO.setCode(5000);
        resultDTO.setMsg("未知错误");
        return resultDTO;
    }
}

```

4.3.5 商品类型管理功能

在商品类型列表页面超级管理员能够查看每页 10 条商品类型信息的展示，也可点击选择每页 20、30、40、50 等条切换每页显示条数，能够按照商品类型名称对商品类型信息进行模糊查询，能够进行单个/批量添加商品类型、批量启用/禁用和导出商品类型信息 Excel 文件操作等。商品类型列表页面 UI 效果，如图 4.29 所示。

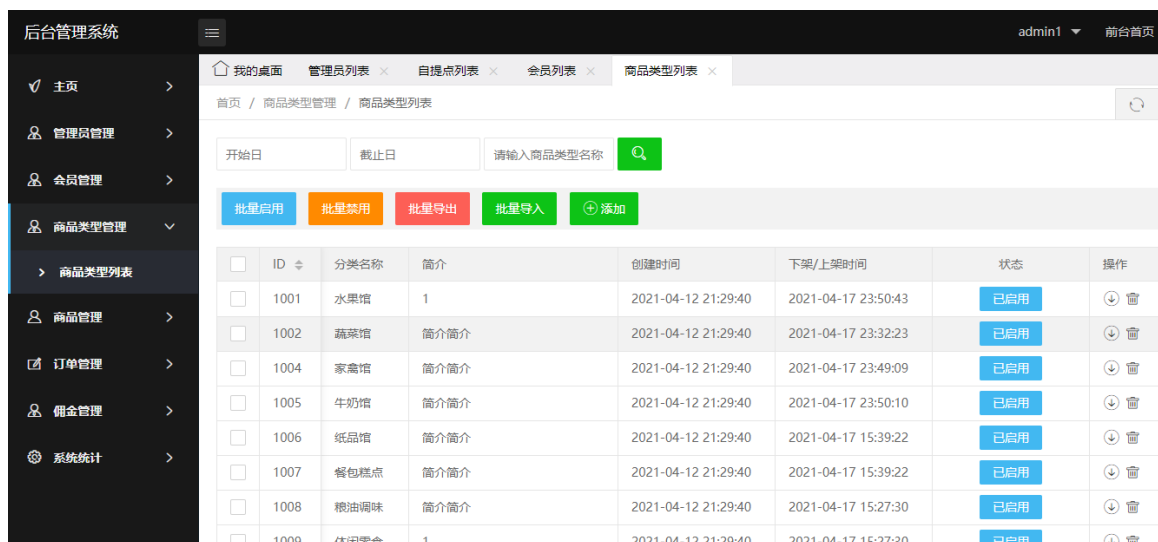


图 4.29 商品类型列表页面

在商品类型管理的 add 接口中，通过接收前端传入的 classify 对象，强转

System.currentTimeMillis()方法获取的当前时间戳，并 set 到 classify 对象中，进行数据库新增商品类型操作。商品类型管理功能实现的关键代码如下。

```
@PostMapping("/add")
public ResultDTO<String> add(@RequestBody Classify classify) {
    return classifyService.insert(classify);
}

public ResultDTO<String> insert(Classify classify) {
    int count = classifyRepository.findByName(classify.getClsName());
    if (count == 0) {
        Long startTs = System.currentTimeMillis()/1000; // 当前时间戳
        classify.setCreatetime(String.valueOf(startTs));
        int result = classifyRepository.insert(classify);
        if (result > 0)
            return new ResultDTO<>(ResultStatus.SUCCESS, Null);
        else
            return new ResultDTO<>(ResultStatus.ERROR, Null);
    }
}
```

4.3.6 商品管理功能

在商品列表页面超级管理员能够查看每页 10 条商品信息的展示，也可点击选择每页 20、30、40、50 等条切换每页显示条数，能够按照商品名称和时间对商品信息进行模糊查询，能够进行单个/批量添加商品、批量上架/下架和导出商品信息 Excel 文件操作等。商品列表页面 UI 效果，如图 4.30 所示。

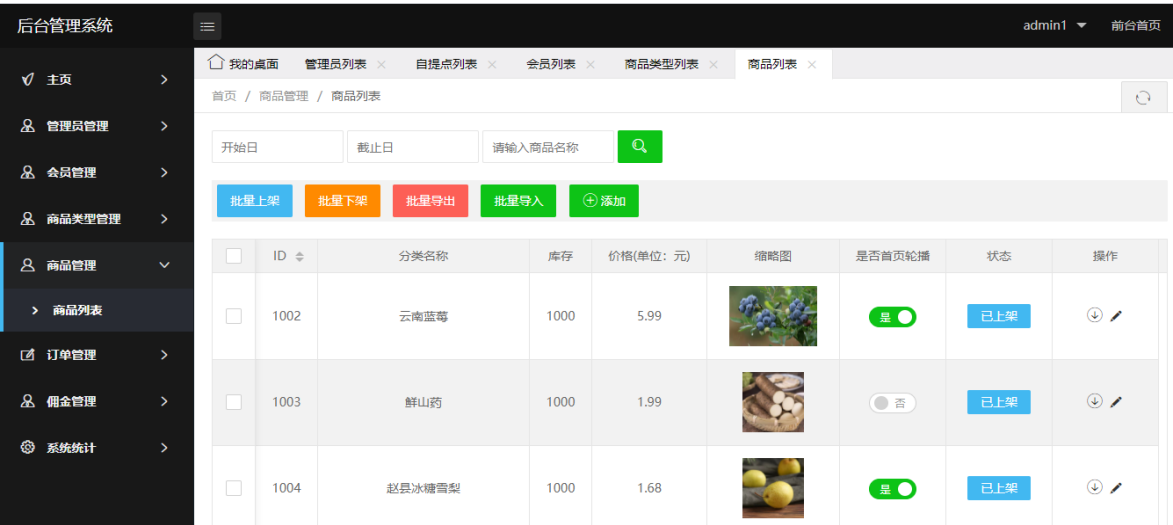


图 4.30 商品列表页面

zip 接口是批量上传导入压缩包文件接口。在该接口功能的具体实现分为五步：
(1) 进行参数校验，判断上传的文件是否为 Zip 格式；(2) 如果是 Zip 格式文件，就将压缩包保存至服务器指定位置；(3) 解压压缩包；(4) 找出 Excel 文件，获取 Excel 的行列数量，遍历循环取出数据，存入 goods 对象的 List 集合中，使用 MyBatis 的批量插入语句新增商品；(5) 删除压缩文件。商品管理功能实现的关键代码如下。

```
@PostMapping("/zip")
public ResultDTO<String> uploadImgZip(MultipartFile zipFile) {
    if (Objects.isNull(zipFile)) {
        return new ResultDTO<>(ResultStatus.ERROR, Null);
    }
    String fileContentType = zipFile.getContentType();
    String filePath = imgStorePath + File.separator + zipFile.getName();
    boolean saveFile = FileUtils.saveFile(zipFile, filePath);
    if (!saveFile) {
        return new ResultDTO<>(ResultStatus.ERROR, Null);
    }
    String destPath = imgStorePath + File.separator;
    boolean unzip = FileUtils.unpackZip(new File(filePath), "", destPath, true);
    String Name = FileUtils.getExtensionName(zipFile.getOriginalFilename());
    goodsRepository.batchInsert(goodsList);
    return new ResultDTO<>(ResultStatus.SUCCESS, Null);
    return new ResultDTO<>(ResultStatus.ERROR, Null);
}
```

4.3.7 订单管理功能

在订单列表页面超级管理员能够查看每页 10 条订单信息的展示，也可点击选择每页 20、30、40、50 等条切换每页显示条数，能够按照订单编号和时间对订单信息进行模糊查询，能够审核退货信息和导出订单信息 Excel 文件操作等。商品类型列表页面 UI 效果，如图 4.31 所示。

在订单管理的 batchUpdate 接口中，通过将 ids 数组转换成 order 对象的 List 集合，传递给 orderService.batchUpdate 方法，通过 Mybatis 的动态 SQL 语句实现批量更新。订单管理功能实现的关键代码如下。

```
@PostMapping("/batchUpdate")
public ResultDTO<String> batchUpdate(@RequestBody BatchDTO batch) {
```

```

        return orderService.batchUpdate(batch);
    }

    public ResultDTO<String> batchUpdate(BatchDTO batchDTO) {
        List<Order> orderList = new ArrayList<>();
        for (String i : batchDTO.getOids()) {
            Order order = new Order();
            order.setOid(i);
            order.setOrderStatus(batchDTO.getStatus());
            orderList.add(order);
        }
        orderRepository.batchUpdate(orderList);
        return new ResultDTO<>(ResultStatus.SUCCESS, Null);
    }
}

```

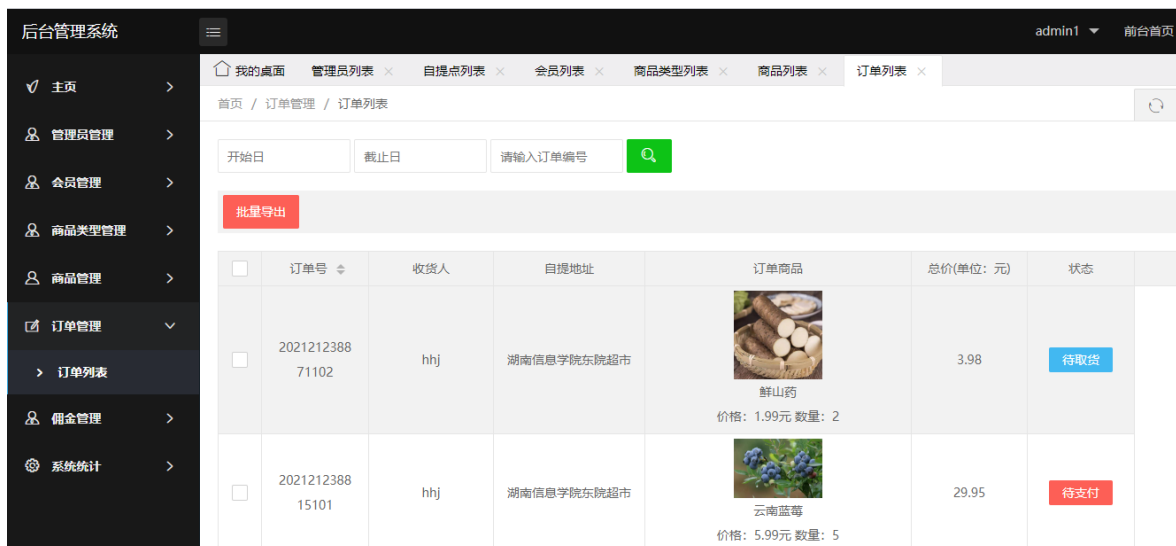


图 4.31 订单列表页面

4.3.8 佣金管理功能

在佣金列表页面超级管理员能够查看每页 10 条佣金信息的展示，也可点击选择每页 20、30、40、50 等条切换每页显示条数，能够按照订单编号和时间对佣金信息进行模糊查询，能够导出佣金信息 Excel 文件操作等。商品类型列表页面 UI 效果，如图 4.32 所示。

在佣金管理的 ByPage 的接口中，通过 SQL 语句中 limit 关键字实现数据分页。佣金管理功能实现的关键代码如下。

```

@GetMapping("/ByPage")
public ResultDTO commissionByPage(PageDTO pageDTO) {

```

```

return commissionService.findPageWithResult(pageDTO);
}

```



图 4.32 佣金列表页面

4.3.9 提现管理功能

在提现记录页面超级管理员能够查看每页 10 条提现信息的分页展示，也可点击选择每页 20、30、40、50 等条切换每页显示条数，能够进行模糊搜索提现记录信息、按时间显示提现记录信息、单个/批量审核提现申请信息和导出提现记录 Excel 文件操作等。提现记录页面 UI 效果，如图 4.33 所示。

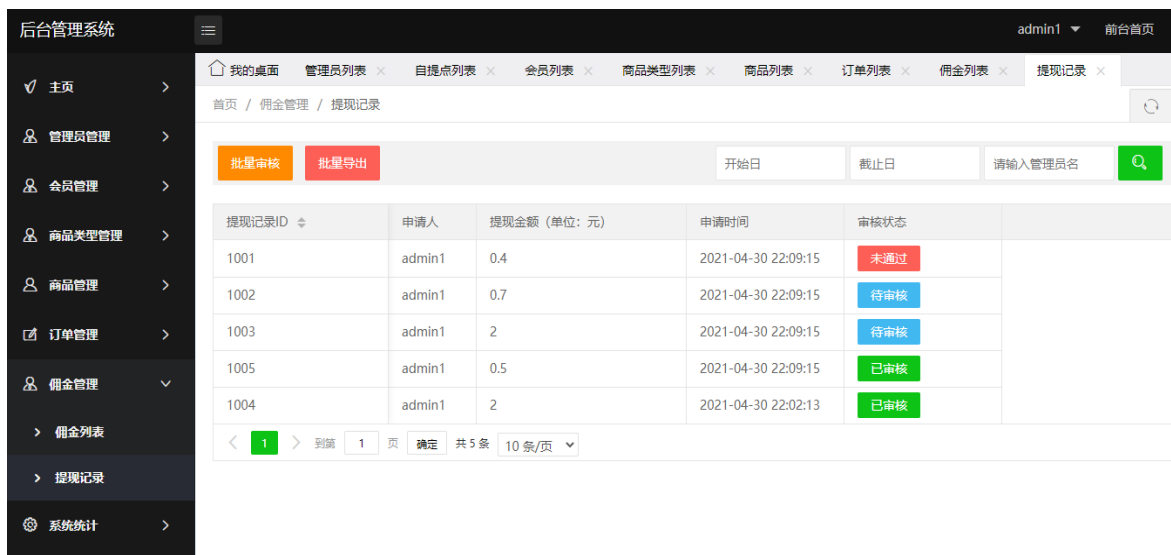


图 4.33 提现记录页面

在提现管理的 ByPage 的接口中，通过 SQL 语句中 limit 关键字实现数据分页。提现管理功能实现的关键代码如下。

```
@GetMapping("/ByPage")
```

```

public ResultDTO withdrawalByPage(PageDTO pageDTO) {
    return withdrawalService.findPageWithResult(pageDTO);
}

public ResultDTO findPageWithResult(PageDTO pageDTO) {
    List<WithdrawalDTO> withdrawalDTOList=withdrawalRepository.findPageWithResult(pageDTO);
    if (withdrawalDTOList.size()!=0) {
        resultDTO.setCode(0);
        resultDTO.setData(withdrawalDTOList);
    }
    return resultDTO;
}

```

4.3.10 系统统计功能

自提点数分布地图页面 UI 效果，如图 4.34 所示。折线图主要功能同 4.3.1 小节综合管理中折线图的功能一致，此处只是把图放大，方便超级管理员查看。折线图页面 UI 效果，如图 4.35 所示。



图 4.34 自提点数分布地图页面

特别声明：此处展示的为系统近一个月的每日订单总数、销售总额、用户提现金额。

折线图堆叠



图 4.35 折线图页面

`findAllRegion` 接口为查询自提点分布数据接口，在该接口中查询了自提点在各个省级行政区的分布数量，并且利用接口响应的数据在前端渲染出 ECharts 地图。自提点数分布地图功能实现的关键代码如下。

```
@PostMapping("/findAllRegion")
public ResultDTO<List<AllDTO>> findAllRegion() {
    return regionService.findAllRegion();
}

public ResultDTO<List<RegDTO>> findRegWithCount(String[] p) {
    List<RegDTO> regDTOList=addressRepository.findRegWithCount();
    return new ResultDTO<>(ResultStatus.SUCCESS,regDTOList);
}
```

`LineChart` 接口为查询折线图数据接口，在该接口中查询了每日的总销售额、订单数、提现金额，并且利用接口响应的数据在前端渲染出 ECharts 折线图。折线图功能实现的关键代码如下。

```
@GetMapping("/LineChart")
public ResultDTO<List<List<String>>> LineChart() {
    List<List<String>> LineChart = statisticsService.findByOrderSales();
    return new ResultDTO<>(ResultStatus.SUCCESS, LineChart);
}
```

4.4 本章小结

本章首先阐述用户端会员功能的实现，主要实现了查看商品及详情、加入购物车、购物车管理、订单管理等功能，使用阿里云短信接口实现注册登录验证码功能，使用支付宝沙箱支付接口实现订单结算功能。其次是用户端自提点管理员功能的实现，主要实现了提现申请、订单管理等功能。然后是管理端超级管理员功能的实现，主要实现了商品管理、提现审核、系统统计等功能。最后通过用户端、管理端的业务交互组成了社区团购系统。

5 系统测试

5.1 系统测试目的与方法

5.1.1 测试目的与意义

软件系统功能测试目的在于发现系统中的各种错误和缺陷，然后解决相关问题，提高系统的质量，在某种程度上大幅度提高产品的易用性，提高用户的使用满意度。软件系统性能测试的主要目的在于保障系统的稳定性、可靠性和运行的流畅性^[27]。

5.1.2 测试的方法和工具

软件的测试方法主要分为两种：白盒测试和黑盒测试。黑盒测试也称为功能测试，将系统看做一个密不透光的黑盒子，不考虑系统的内部结构，通过输入、输出的数据验证软件功能。

软件性能测试工具主要有 LoadRunner 和 Jemter^[28]。Jemter 是一款免费开源的工具，它通过模拟出多个虚拟用户向服务器发送普通和并发请求，检测响应返回情况。本系统主要使用的测试方法为黑盒测试，并利用 Jmeter 工具进行系统性能测试。

5.2 用户端功能测试

5.2.1 注册登录找回密码功能测试

注册功能测试主要针对手机号码、普通验证码和密码三个测试点设计测试用例。测试用例及实际测试的结果，如表 5.1 所示。

表 5.1 注册功能测试用例

编号	用例	测试数据	预期结果	实际结果
1	输入正确的手机号码	手机号码：17307428595	注册成功	与预期 结果一致
2	输入错误的手机号码	手机号码：138749405	注册失败	与预期 结果一致
3	输入已经注册过的号码	手机号码：17307428595	提示已注册，直接 登录或找回密码	与预期 结果一致
4	输入正确的手机号码、 普通验证码和密码	手机：13874940542 密码：123456 验证码：khg3(假设)	注册成功，跳入 登录页面	与预期 结果一致
5	不输入手机号码	手机号码：Null	手机号码不能为空	与预期 结果一致

(续表)

编号	用例	测试数据	预期结果	实际结果
6	不输入密码	密码: Null	密码不能为空	与预期 结果一致
7	必填项为空	头像: Null 姓名: Null	必填项不能为空	与预期 结果一致

登录功能测试主要针对手机号码和密码两个测试点设计测试用例。测试用例及实际测试的结果, 如表 5.2 所示。

表 5.2 登录功能测试用例

编号	用例	测试数据	预期结果	实际结果
1	输入正确的手机号码、普通验证码和密码	号码: 17307428595 密码: 123456 验证码: khg3(假设)	登录成功, 进入主页面	与预期 结果一致
2	输入空的号码	手机号码: Null	手机号码不能为空	与预期 结果一致
3	输入正确的手机号码、空密码	号码: 17307428595 密码: Null	密码不能为空	与预期 结果一致
4	输入正确的手机号码、错误密码	号码: 17307428595 密码: 123	用户名或密码错误	与预期 结果一致
5	输入没有注册的手机号	号码: 17307428596 密码: 123456	手机号未注册	与预期 结果一致
6	输入错误的验证码	验证码: khg3(假设) 输入: 55jb	验证码有误	与预期 结果一致
7	输入正确的手机号码、错误密码	号码: 17307428595 密码: 123	用户名或密码错误	与预期 结果一致

找回密码功能测试主要针对手机号码、短信验证码和密码这三个测试点设计测试用例。测试用例及实际测试的结果, 如表 5.3 所示。

表 5.3 找回密码功能测试用例

编号	用例	测试数据	预期结果	实际结果
1	输入正确的手机号码、密码、确认密码和短信验证码	号码: 17307428595 新和确认密码: 123456 验证码: 512454(假设)	密码修改成功	与预期 结果一致

(续表)

编号	用例	测试数据	预期结果	实际结果
2	输入正确的手机号 码、空确认密码	号码: 17307428595 确认密码: Null	确认密码不能为空	与预期 结果一致
3	输入没有注册的手机号	号码: 17307428596 新密码: 123456 确认密码: 123456	手机号未注册	与预期 结果一致
4	输入正确的手机号 码、两次密码不一致	号码: 17307428595 新密码: 12356 确认密码: 12345	两次密码输入不一致	与预期 结果一致
5	输入正确的手机号 码、空密码	号码: 17307428595 新密码: Null	新密码不能为空	与预期 结果一致
6	获取短信验证码间隔未超过 60s	同用例	不能获取短信验证码	与预期 结果一致
7	60s 后再次获取短信验证码	同用例	短信验证码获取成功	与预期 结果一致
8	输入错误的短信验证码	验证码: 512454(假设) 输入: 555124	短信验证码有误	与预期 结果一致

5.2.2 购物车功能测试

购物车功能测试主要针对购物车登录验证、单选、多选、全选、商品数量加减等功能点设计测试用例。测试用例及实际测试的结果，如表 5.4 所示。

表 5.4 购物车功能测试用例

编号	用例	测试数据	预期结果	实际结果
1	未登录查看购物车、 商品加入购物车	同用例	提示请登录	与预期 结果一致
2	商品数增加超过库存	库存: 99	增加至 99, 无法继续增加	与预期 结果一致
3	商品数量减少为 0	商品数量: 0	购物车删除商品	与预期 结果一致
4	单选、多选、全选	同用例	订单总额随着勾选的商品及数量变化	与预期 结果一致

(续表)

编号	用例	测试数据	预期结果	实际结果
5	勾选商品，结算	同用例	跳转至结算页面，商品、数量及订单总额正确	与预期结果一致

5.2.3 订单结算及管理功能测试

订单结算及管理功能测试主要针对登录验证、立即支付、取消订单、申请退货、订单分类显示等功能点设计测试用例。测试用例及实际测试的结果，如表 5.5 所示。

表 5.5 订单结算及管理功能测试用例

编号	用例	测试数据	预期结果	实际结果
1	未登录会员/自提点管理员查看订单管理	同用例	会员：提示请登录 自提点管理员：跳转至登录页面	与预期结果一致
2	登录会员/自提点管理员查看不同类型的订单	全部、待支付、待提货、已提货	正常显示每个分类的订单	与预期结果一致
3	未支付订单点击取消订单	同用例	订单状态改变为已取消	与预期结果一致
4	未支付订单点击立即支付	同用例	订单支付成功，订单状态改变为已取消	与预期结果一致
5	已支付订单点击申请退货	同用例	等待后台审核	与预期结果一致
6	已支付、待提货、已提货、已取消订单点击继续购物	同用例	跳转至系统首页	与预期结果一致

5.2.4 选择自提点功能测试

选择自提点功能测试主要针对登录验证、进店、历史自提点、当前自提点等功能点设计测试用例。测试用例及实际测试的结果，如表 5.6 所示。

表 5.6 选择自提点功能测试用例

编号	用例	测试数据	预期结果	实际结果
1	未登录会员查看历史自提点	同用例	提示请登录	与预期结果一致

(续表)

编号	用例	测试数据	预期结果	实际结果
2	新注册会员查看历史自提点	同用例	当前自提点及历史自提点为空，提示请选择自提点	与预期结果一致
3	已购物会员查看历史自提点	同用例	当前自提点及历史自提点正常显示	与预期结果一致
4	选择一个自提点，点击进店	同用例	当前自提点修改为选择的自提点，历史自提点新增一个	与预期结果一致
5	反复切换自提点	同用例	当前自提点修改正常	与预期结果一致

5.2.5 佣金记录功能测试

佣金记录功能测试主要针对登录验证、佣金记录的显示以及订单结算后佣金记录的新增等功能点设计测试用例。测试用例及实际测试的结果，如表 5.7 所示。

表 5.7 佣金记录功能测试用例

编号	用例	测试数据	预期结果	实际结果
1	未登录管理员查看佣金记录	同用例	跳转至登录	与预期结果一致
2	查看佣金记录	同用例	当前管理员提现记录正常显示	与预期结果一致
3	订单结算后新增佣金记录	同用例	新增一条佣金记录	与预期结果一致

5.2.6 提现申请功能测试

提现申请功能测试主要针对登录验证、提现记录的查看、提现金额输入等功能点设计测试用例。测试用例及实际测试的结果，如表 5.8 所示。

表 5.8 提现申请功能测试用例

编号	用例	测试数据	预期结果	实际结果
1	未登录管理员查看提现记录	同用例	跳转至登录	与预期结果一致
2	查看提现记录	同用例	当前管理员提现记录正常显示	与预期结果一致

(续表)

编号	用例	测试数据	预期结果	实际结果
3	提现申请金额为空	提现金额: Null	提现金额不能为空	与预期 结果一致
4	提现申请金额大于余额	余额: 2.4 提现金额: 3	提现金额不能大于余额	与预期 结果一致
5	提现金额为负数、中文、字母、特殊符号	同用例	提现金额只能输入正数	与预期 结果一致
6	提现金额大于 0, 小于余额	余额: 2.4 提现金额: 1.3	提现申请提交成功, 管理员余额减少	与预期 结果一致

5.3 管理端功能测试

5.3.1 管理员管理功能测试

管理员管理主要针对门店入驻报名的管理员账号申请进行审核、自提点管理员信息修改、批量启用、批量禁用、批量导出以及对管理员信息进行模糊查询等功能点设计测试用例。测试用例及实际测试的结果, 如表 5.9 所示。

表 5.9 管理员管理功能测试用例

编号	用例	测试数据	预期结果	实际结果
1	单个/批量启用管理员账号	同用例	自提点管理员账号可以正常登录用户端	与预期 结果一致
2	单个/批量禁用管理员账号	同用例	自提点管理员账号不能登录用户端	与预期 结果一致
3	自提点管理员账号登录管理端	同用例	自提点管理员账号不能登录管理端	与预期 结果一致
4	批量导出	同用例	导出管理员信息 Excel 表	与预期 结果一致
5	信息修改任意必填项信息缺失	同用例	必填项为空	与预期 结果一致
6	模糊查询输入为空	关键字: Null	管理员信息全部展示	与预期 结果一致
7	模糊查询输入为 hh	关键字: hh	展示包含 hh 的管理员信息	与预期 结果一致

5.3.2 自提点管理功能测试

自提点管理主要针对自提点进行批量启用/禁用、批量导出以及对自提点地址进行模糊查询等功能点设计测试用例。测试用例及实际测试的结果，如表 5.10 所示。

表 5.10 自提点管理功能测试用例

编号	用例	测试数据	预期结果	实际结果
1	单个/批量启用自提点	同用例	会员能够使用自提点提货	与预期结果一致
2	单个/批量禁用自提点	同用例	会员不能使用自提点提货	与预期结果一致
3	批量导出	同用例	导出自提点信息 Excel 表	与预期结果一致
4	模糊查询输入为空	关键字: Null	自提点信息全部展示	与预期结果一致
5	模糊查询输入为湖南	关键字: 湖南	展示包含湖南的自提点信息	与预期结果一致

5.3.3 会员管理功能测试

会员管理主要针对会员进行批量导出以及对会员信息进行模糊查询等功能点设计测试用例。测试用例及实际测试的结果，如表 5.11 所示。

表 5.11 会员管理功能测试用例

编号	用例	测试数据	预期结果	实际结果
1	批量导出	同用例	导出会员信息 Excel 表	与预期结果一致
2	模糊查询输入为空	关键字: Null	会员信息全部展示	与预期结果一致
3	模糊查询输入为 hh	关键字: hh	展示包含 hh 的会员信息	与预期结果一致

5.3.4 商品类型管理功能测试

商品类型管理主要针对商品类型单个/批量添加、信息修改、批量启用/禁用、批量导出、对商品类型进行模糊查询以及时间查询等功能点设计测试用例。测试用例及实际测试的结果，如表 5.12 所示。

表 5.12 商品类型管理功能测试用例

编号	用例	测试数据	预期结果	实际结果
1	单个/批量启用商品类型	同用例	商品类型启用，用户端可以浏览分类下的商品	与预期 结果一致
2	单个/批量禁用商品类型	同用例	商品类型及商品全部禁用	与预期 结果一致
3	批量导出	同用例	导出商品类型信息 Excel 表	与预期 结果一致
4	信息添加/修改任意必填项信息缺失	同用例	必填项为空	与预期 结果一致
5	时间查询开始结束时间限制	同用例	提示：开始不能选择当前时间之后的日期，结束时间小于时间	与预期 结果一致
6	模糊查询输入为空	关键字：Null	商品类型信息全部展示	与预期 结果一致
7	模糊查询输入为水	关键字：水	展示包含水的商品类型信息	与预期 结果一致

5.3.5 商品管理功能测试

商品管理主要针对商品单个/批量添加、信息修改、批量上架/下架、批量导出、对商品进行模糊查询以及时间查询等功能点设计测试用例。测试用例及实际测试的结果，如表 5.13 所示。

表 5.13 商品管理功能测试用例

编号	用例	测试数据	预期结果	实际结果
1	单个/批量上架商品	同用例	商品启用，用户端可以浏览商品	与预期 结果一致
2	单个/批量下架商品	同用例	商品禁用，用户端不能浏览商品	与预期 结果一致
3	批量导出	同用例	导出商品信息 Excel 表	与预期 结果一致
4	信息添加/修改任意必填项信息缺失	同用例	必填项为空	与预期 结果一致

(续表)

编号	用例	测试数据	预期结果	实际结果
5	模糊查询输入为空	关键字: Null	商品信息全部展示	与预期 结果一致
6	模糊查询输入为云南	关键字: 云南	展示包含云南的商品类型信息	与预期 结果一致
7	时间查询开始结束 时间限制	同用例	提示: 开始不能选择当前 时间之后的日期, 结束时间 小于时间	与预期 结果一致

5.3.6 佣金管理功能测试

佣金管理主要针对批量导出、对佣金信息进行模糊查询以及按时间段查询等功能点设计测试用例。测试用例及实际测试的结果, 如表 5.14 所示。

表 5.14 佣金管理功能测试用例

编号	用例	测试数据	预期结果	实际结果
1	批量导出	同用例	导出会员信息 Excel 表	与预期 结果一致
2	模糊查询输入不为空	关键字: 202119492207103	展示订单编号为 202119492207103 佣金信息	与预期 结果一致
3	模糊查询输入为空	关键字: Null	佣金信息全部展示	与预期 结果一致
4	时间查询开始结束 时间限制	同用例	提示: 开始不能选择当前 时间之后的日期, 结束时间 小于时间	与预期 结果一致

5.3.7 订单管理功能测试

订单管理主要针对退货申请审核、批量导出、订单模糊查询或时间查询等功能点设计测试用例。测试用例及实际测试的结果, 如表 5.15 所示。

表 5.15 订单管理功能测试用例

编号	用例	测试数据	预期结果	实际结果
1	批量导出	同用例	导出订单信息 Excel 表	与预期 结果一致

(续表)

编号	用例	测试数据	预期结果	实际结果
2	退货审核通过	同用例	订单状态改变为已退货	与预期 结果一致
3	退货审核不通过	同用例	订单状态改变为不同意 退货	与预期 结果一致
4	时间查询开始结束 时间限制	同用例	提示：开始不能选择当前 时间之后的日期，结束时 间小于时间	与预期 结果一致
5	模糊查询输入为空	关键字：Null	订单信息全部展示	与预期 结果一致
6	模糊查询输入不为 空	关键字：20211949220 7103	展示订单编号为 2021194 92207103 订单信息	与预期 结果一致

5.3.8 系统统计功能测试

系统统计功能主要针对折线图、自提点数量地图、系统信息统计等功能点设计测试用例。测试用例及实际测试的结果，如表 5.16 所示。

表 5.16 提现管理功能测试用例

编号	用例	测试数据	预期结果	实际结果
1	查看折线图	同用例	对比数据库数据，折线图 数据正确	与预期 结果一致
2	查看自提点数量地图	同用例	对比数据库数据，自提点 数量地图数据正确	与预期 结果一致
3	系统信息统计	同用例	对比数据库数据，系统信 息统计数据正确	与预期 结果一致

5.3.9 提现管理功能测试

提现管理主要针对提现申请审核、批量导出、对提现记录进行模糊查询以及时间查询等功能点设计测试用例。测试用例及实际测试的结果，如表 5.17 所示。

表 5.17 提现管理功能测试用例

编号	用例	测试数据	预期结果	实际结果
1	批量导出	同用例	导出提现记录 Excel 表	与预期 结果一致

(续表)

编号	用例	测试数据	预期结果	实际结果
2	提现审核通过	同用例	提现状态由待审核改变 为已同意	与预期 结果一致
3	提现审核不通过	同用例	提现状态由待审核改变 为未通过	与预期 结果一致
4	时间查询开始结束 时间限制	同用例	提示：开始不能选择当前 时间之后的日期，结束时 间小于时间	与预期 结果一致
5	模糊查询输入为空	关键字：Null	提现记录信息全部展示	与预期 结果一致
6	模糊查询输入不为 空	关键字：1001	展示提现 ID 为 1001 提现 记录	与预期 结果一致

5.4 功能测试结果分析

(1) 测试用例统计

根据功能测试设计的测试用例、一轮测试和回归测试结果分析得到，系统在功能测试阶段，共设计测试用例 95 条，执行测试用例 95 条，未执行测试用例数为 0，成功的测试用例为 95 条，失败的测试用例为 0 条。测试用例统计结果，如表 5.18 所示。

表 5.18 测试用例统计

用例总数	执行数	未执行数	成功数	失败数	用例执行成功率
95	95	0	95	0	95/95

(2) 缺陷的统计与分析

根据系统测试记录的缺陷汇总可得，系统在功能测试阶段总缺陷数为 15 个，其中已关闭 Bug 数为 15 个，延期处理 Bug 数为 0 个。Bug 状态统计如图 5.1 所示。

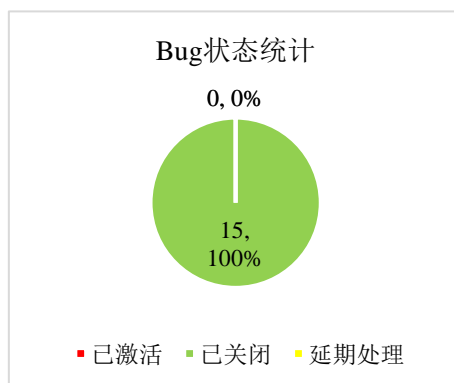


图 5.1 Bug 状态统计

按缺陷类型统计可得，大量 Bug 类型为功能问题，其次是页面问题。Bug 类型统计如图 5.2 所示。

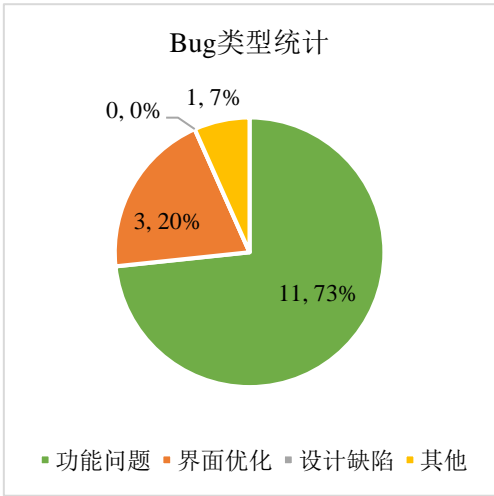


图 5.2 Bug 类型统计

按缺陷严重程度统计可得，大量 Bug 为三级 Bug，少量 Bug 为二级和四级 Bug。Bug 严重程度统计如图 5.3 所示。

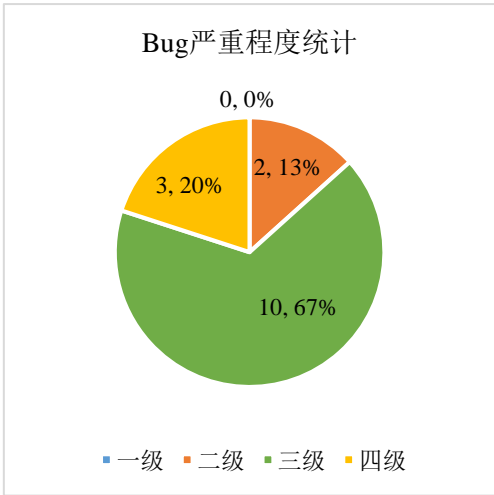


图 5.3 Bug 严重程度统计

综上所述，系统功能测试覆盖了功能测试所有的测试需求及案例，目前缺陷均已修复并回归关闭，系统达到项目预期需求，系统功能测试通过。

5.5 系统性能测试

系统进行的性能测试主要为系统接口的响应速度测试。系统遵循行业通用 2-5-8 原则，通过使用 Jmeter 工具对系统主要接口进行压力测试，得出以下结论：在并发用户为 2000 时，系统三个主要接口（注册、登录、添加订单）平均响应时间分别为 2195ms、2120ms、2963ms，总体平均响应时间为 2426ms，总体异常百分比为 0.2%，

符合 2.4.3 小节稳定性分析中接口请求的异常百分比小于 0.3%和平均响应时间在 3 秒内的需求。性能测试详细结果，如表 5.19 所示。

表 5.19 性能测试结果表

Label	样本	平均值	最小值	最大值	标准偏差	异常%	吞吐量	接收 KB/sec	发送 KB/sec	平均字 节数
注册	20000	2195	15	3257	1621	0.10%	222.568	99.33	98.24	457
登录	20000	2120	1	3163	1581	0.20%	129.895	59.87	44.52	472
下单	20000	2963	7	3591	1792	0.30%	128.625	61.61	44.59	490.5
总体	60000	2426	1	3337	1668	0.20%	385.381	178.07	145.27	473.2

5.6 本章小结

本章主要介绍了系统测试过程中所使用到的测试方法和相关工具，并从用户的角度出发对功能进行了详尽的测试，一轮测试及回归测试的测试结果表明，现阶段系统功能测试通过。在系统性能测试的过程中，通过使用 Jmeter 工具对系统主要接口进行两万并发量的压力测试，得出现阶段系统的接口平均响应时间符合需求（不超过 3 秒），该结果表明系统性能测试通过。

6 结 论

在设计和开发社区团购系统整个过程中，本人查阅了大量的期刊和硕博士论文，并综合分析了现有的社区团购系统，并针对现阶段传统电商无法解决的最后一公里配送、在大型电商平台上购物时效性受到限制以及很多农村地区快递存在的“长运距+低密度需求”导致的快递二次收费加价等问题，借助 IntelliJ IDEA 平台、SpringBoot 框架、Layui 模板、Bootstrap 模板、ECharts 可视化图表库和 Jmeter 工具开发、测试社区团购系统。最后经过不断的修改和完善，实现了集自提点管理、商品类型管理、商品管理、订单管理、佣金管理、提现管理、报表展示等功能于一体的社区团购系统，为会员提供了一个实用性强、功能丰富的社区团购平台，为社区和村镇门店（自提点）提供了一个分佣提现的平台。

由于目前实际开发经验的不足，系统还存在很多性能相关的问题，如在高并发情况下（超过两万并发量），大量用户在同一时间进入系统，系统资源瞬间消耗殆尽，从而会导致系统出现加载缓慢、甚至崩溃无法访问等问题。在完善好自身的专业知识后，将针对系统中存在的相关问题，我会不断的去优化、完善和丰富社区团购系统，对社区团购系统进行升级和改造。如在系统中接入实际的支付接口，使系统能够真实的支付和提现到账；在学习并掌握好微服务分布式技术后，将对系统重新进行架构，使系统的性能能够更加优越。

综合看来，通过本系统开展社区团购业务，可以达到降低人力成本和管理成本，并极大程度地让用户享受社区团购服务、获得商品让利，平台（系统运营方）和商家（社区自提门店）能够通过自身的服务获取相应的酬劳等。

参考文献

- [1] 徐婧.电子商务平台从C2B向C2M模式变革的机遇与挑战[J].企业改革与管理,2017(04):58.
- [2] 蹇芳莉.中国电子商务报告2019[R].北京:商务部电子商务和信息化司,2020.
- [3] 国家邮政局网站.关于推进邮政业服务乡村振兴的意见[EB/OL].http://www.spb.gov.cn/zc/flfgjzc_1/201905/t20190523_1839077.html,2019.05.
- [4] 启明.电商新趋势:社区团购来了[J].中国质量万里行,2018(11):39-41.
- [5] 艾媒新零售产业研究中心.2020上半年中国社区团购行业专题研究报告[EB/OL].<https://www.iimedia.cn/c400/74181.html>,2020.09.
- [6] 邓智敏,俞亚明.跨境电商平台商业模式的比较研究——以亚马逊与阿里巴巴为例[J].广西质量监督导报,2021(04):122-123.
- [7] 姜曼.基于波特价值链模型的Groupon类团购网站竞争力评价研究[D].成都:成都理工大学,2013.
- [8] 刘天雨.时间窗口约束下社区团购生鲜第三方平台仓配模式优化问题研究[D].郑州:郑州大学,2019.
- [9] 薛云建,高誉连.韩国团购网站Groupon Korea的盈利模式[J].企业研究,2014(10):4-5.
- [10] Matsuo T , Ito T . A group formation support system based on substitute goods in group buying.[J]. Systems and Computers in Japan, 2010, 35(10):23-31.
- [11] Varinder M. Sharma,Andreas Klein. Consumer perceived value, involvement, trust, susceptibility to interpersonal influence, and intention to participate in online group buying[J/OL].<https://doi.org/10.1016/j.jretconser>,2020.09.
- [12] Antonio Aguiar Costa,Luis Valadares Tavares. Social e-business and the Satellite Network model: Innovative concepts to improve collaboration in construction[J]. Automation in Construction, 2012, 22(3):387-397.
- [13] 程蔚.面向社区移动电商系统研究与实现[D].昆明:昆明理工大学,2016.
- [14] 匡晶.浅析社区社群电商发展现状及趋势[J].中国集体经济,2019(33):56-57.
- [15] 任小东.社区电商,中国零售第三极[J].销售与市场(管理版),2019(01):67-69.
- [16] 车新.面向智慧社区的电子商务系统的设计与实现[D].沈阳:东北大学,2013.
- [17] 龙慧,罗可利,胡利,赵娜.基于J2EE的社区电商平台系统的设计与开发[J].物联网技术,2020,10(06):71-75.
- [18] 王伟丽.智慧社区电子商务系统的设计与实现[D].北京:北京邮电大学,2018.
- [19] 万全.基于ThinkPHP框架的团购网站的设计与实现[D].北京:首都经济贸易大学,2018.

- [20] Chen C , Deng Y . A fair and secure group buying system based on arbitration computing mechanism[J]. Soft Computing - A Fusion of Foundations, Methodologies and Applications, 2018, 22(1):119-135.
- [21] Chen C L , Deng Y Y . A secure mobile group buying system based on the concept of spontaneity[J]. Engineering Computations, 2016, 33(7):2198-2215.
- [22] 陈恒.Spring Boot从入门到实战(微课视频版)[M].北京:清华大学出版社,2020:91-93.
- [23] 黑马程序员.Java EE企业级应用开发教程(Spring+Spring MVC+MyBatis)[M].北京:人民邮电出版社,2017:84-86.
- [24] 付磊,张益军.Redis开发与运维[M].北京:机械工业出版社, 2017:1-5.
- [25] 齐治昌,谭庆平,宁洪.软件工程[M].北京:高等教育出版社,2019:223-225.
- [26] 刘伶华.软件测试从小白到高手[M].天津:天津科学技术出版社,2020:5-7,42- 44.
- [27] 黑马程序员.软件测试[M].北京:人民邮电出版社,2019:69-76,102-107.
- [28] 陈志勇,马利伟,万龙,ROAD_TESTING软件测试组.全栈性能测试修炼宝典[M].北京:人民邮电出版社,2016:22-26.

致 谢

时光飞逝，像一首歌。转眼间，五年的大学生活即将划上圆满的句号。站在毕业季的大门口追溯往事，奋斗和拼搏都在欢声笑语中落下帷幕。五年间我收获了很多，在毕业论文即将完成之际，我真心感谢那些关心我成长、给予过我帮助的人，如果没有他们无私的帮助，就没有我今天的成长。

我由衷的感谢我的论文导师杨顺老师、陈振老师和郑治武老师，系统和论文的完成离不开三位老师的耐心指导。三位导师渊博的专业知识、严谨的治学态度、实事求是的学术作风和淳朴平易近人的人格魅力对我影响深远。从论文的选题，到系统的实现、论文的反复修改、润色直至定稿，三位导师始终认真负责地给予我深刻而细致地指导，将代码与论文的错误一一指出，并且给予了诸多意见。如果没有三位导师的建议和意见，我的系统和论文不可能顺利完成。在此，我再次向三位导师表示深深的感谢和祝福！

我要感谢我的辅导员张如老师、学业导师罗莉霞老师以及在求学路上给我授课的所有老师，感谢你们给予的学业指导和生活关爱。张如、罗莉霞两位老师无微不至，经常提醒我学习的重要性，时刻关注我们学习和顶岗实习期间出现的种种问题，并予以帮助协调解决。在老师们“学高为师，身正为范；学为人师，行为世范”的师德师风中，我学到了很多的专业及人文知识，学会了做人做事，更让我见识到了软件编码的魅力。我相信在未来工作和生活中，老师的教导将永远刻在心中。

我要感谢我的父母和祖父母，父母常年在外务工，感谢祖父母含辛茹苦把我抚养成人，感恩父母总是竭尽所能的把最好的东西给我，感谢父母在我的求学之路上的无私支持，让我圆满完成专科三年，本科两年的大学求学之路。养育之恩，无以为报，唯有不断发奋，成为家人眼中的骄傲。

我还要感谢感谢求学路上陪伴我成长的朋友、同学们，因为相遇我的人生才足够丰富，因为相遇我的求学之路才不孤单。

最后，祝福吾师桃李满天下，祝福吾亲身体健康，祝福吾友前程似锦。