归 档 号：

武汉工商学院毕业论文(设计)

学院：人工智能学院

专业：计算机科学与技术　　年级：2017 级

学生：　　学号：

指导教师：　　职称：

题目：基于 SpringCloud 的聚美购物商城的设计与实现

2021 年 5 月 9 日

武汉工商学院
本科毕业论文（设计）原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的成果作品。本人完全意识到本声明的法律后果由本人承担。

作者签名：

年　　月　　日

目 录

摘　要.....I
关键词.....I
Abstract.....II
Key words.....II
1 绪论.....1
1.1 选题背景.....1
1.2 选题目的.....1
1.3 选题意义.....1
2 可行性分析.....3
2.1 技术可行性.....3
2.2 经济可行性.....3
2.3 操作可行性.....3
3 系统需求分析.....4
3.1 系统功能概述.....4
3.2 功能需求.....4
3.3 数据需求.....6
4 系统总体设计.....13
4.1 系统架构设计.....13
4.2 数据库设计.....13
4.2.1 逻辑结构设计.....13
4.2.2 数据库表结构设计.....14
5 系统详细设计与实现.....20
5.1 用户注册功能模块.....20
5.1.1 用户注册功能模块概述.....20
5.1.2 用户注册功能模块流程设计.....20
5.1.3 用户注册功能模块核心代码.....21
5.2 用户登录功能模块.....21
5.2.1 用户登录功能模块概述.....21

5.2.2 用户登录功能模块流程设计.....21
5.2.3 用户登录功能模块核心代码.....22
5.3 用户信息管理功能模块.....23
5.3.1 用户信息管理功能模块概述.....23
5.3.2 用户信息管理功能模块流程设计.....23
5.3.3 用户信息管理功能模块核心代码.....23
5.4 收货地址管理功能模块.....24
5.4.1 收货地址管理功能模块概述.....24
5.4.2 收货地址管理功能模块流程设计.....24
5.4.3 收货地址管理功能模块核心代码.....25
5.5 购物车商品管理功能模块.....25
5.5.1 购物车商品管理功能模块概述.....25
5.5.2 购物车商品管理功能模块流程设计.....25
5.5.3 购物车商品管理功能模块核心代码.....26
5.6 购买商品功能模块.....27
5.6.1 购买商品功能模块概述.....27
5.6.2 购买商品功能模块流程设计.....27
5.6.3 购买商品功能模块核心代码.....27
5.7 用户订单管理功能模块.....28
5.7.1 用户订单管理功能模块概述.....28
5.7.2 用户订单管理功能模块流程设计.....29
5.7.3 用户订单管理功能模块核心代码.....29
5.8 商品管理功能模块.....30
5.8.1 商品管理功能模块概述.....30
5.8.2 商品管理功能模块流程设计.....30
5.8.3 商品管理功能模块核心代码.....30
5.9 商家订单管理功能模块.....31
5.9.1 商家订单管理功能模块概述.....31
5.9.2 商家订单管理功能模块流程设计.....31
5.9.3 商家订单管理功能模块核心代码.....32

6 系统功能测试.....33
6.1 测试环境.....33
6.2 功能测试.....33
6.2.1 用户注册功能测试.....33
6.2.2 用户登录功能测试.....34
6.2.3 用户信息管理功能测试.....35
6.2.4 收货地址管理功能测试.....36
6.2.5 购物车商品管理功能测试.....37
6.2.6 购买商品功能测试.....38
6.2.7 用户订单管理功能测试.....40
6.2.8 商品管理功能测试.....41
6.2.9 商家订单管理功能测试.....42
7 总结.....43
参考文献.....44
致谢.....45

基于 SpringCloud 的聚美购物商城的设计与实现

摘　要

随着互联网技术的日渐成熟和不断发展，人们的生活方式也在逐渐改变，营销关系也逐渐由实体店面销售转变为网络营销。越来越多的企业都选择建立自己的网上购物平台。与传统营销方式相比，网络营销不仅大大降低了实体店面的营销成本，还突破了时间和空间的限制，在降低成本的同时，大大增加了企业的收益。

聚美购物商城是专门为年轻人购物而设计的平台，页面融入了很多年轻的元素，也提供了很多年轻化的商品。本系统实现了用户进行在线搜索、浏览商品，在购物车内进行结算以及最后的支付操作、订单管理等功能；同时实现了商家对商品信息管理、订单管理等一系列的管理功能。

本系统以 J2EE 技术为基础，使用目前较为流行的微服务框架 Spring Cloud 搭建的一个分布式系统，遵循 MVC 的模式理念，使系统开发和维护更方便。系统在开发之前进行了大量的技术储备，在充分了解网上商城系统的基础上，对系统进行需求分析的前提下开发。

关键词：网上购物；微服务；互联网

Design and Implementation of Jumei Shopping Mall Based on SpringCloud

Abstract

With the maturity and continuous development of Internet technology, people's lifestyles are gradually changing, and marketing relationships have gradually changed from physical store sales to online marketing. More and more companies are choosing to build their own online shopping platform. Compared with traditional marketing methods, network marketing not only greatly reduces the marketing costs of physical stores, but also breaks through the limitations of time and space. While reducing costs, it greatly increases the revenue of the enterprise.

Jumei Shopping Mall is a platform specially designed for young people to shop. The page incorporates many young elements and also provides many young products. This system realizes functions such as online search, browsing products, settlement in the shopping cart, final payment operation, order management, etc.; at the same time, it also realizes a series of management functions such as business information management and order management.

This system is based on J2EE technology, uses the currently more popular microservice framework Spring Cloud to build a distributed system, and follows the MVC model concept to make system development and maintenance more convenient. Before the development of the system, a large amount of technical reserves were carried out. On the basis of a full understanding of the online mall system, the system was developed under the premise of a demand analysis of the system.

Key words: Online Shopping; Microservice; Internet

1 绪论

1.1 选题背景

随着互联网技术迅猛发展，人们的工作、学习和生活也因此发生了巨大的变化。互联网不仅成为搜索信息和生活娱乐的最佳途径，而且正在逐步进入商品流通领域，从而使在线购物成为一种新的营销模式，因此涌现出大量的电商平台。针对互联网时代的消费特征，企业基于互联网技术和现有的市场营销理论进行了营销模式的创新。而相比于传统的营销方式，网络营销最大的特点在于消费者主导，人们可以随时随地的购买商品，给用户提供一种更加便捷的购物体验，因此越来越多的人选择网络购物。伴随着网络购物的持续发展，热度增加，新型的交易方式也在迅速的发展壮大，并且取得了令人满意的成绩。自从电子商务发展至今，网络购物已经融入到人们的日常生活，人们不仅能在购物上享受足够的便利，还能在经济上获得足够的优惠。企业也针对当前的背景状况采取适当的措施来应对网络营销这种模式，来更好的适应现代化市场的需求。

1.2 选题目的

由于社会经济的发展，人们把越来越多的时间和精力放在工作上，用来娱乐和购物的时间相对减少，如何能够方便快捷地购买到所需要的商品，已经日益成为人们需要解决的问题。所以，越来越多的企业在网上建立在线商城，模拟和改进传统商店的经营模式，来吸引更多互联网客户。相比于传统商城，网上商城不需要进货就可以把所有的产品都展示在网站上面，且能够实现商品信息化、数据化，满足着用户购物方面的便捷性，迎合电商的发展趋势，抓住了用户对于消费购物方面的痛点以及需求。网上购物商城作为电子商务的一种新形式正在以其高盈利、低成本的优势逐步成为新型的经营模式和理念。但随着用户量和访问量的激增和网上商城的需求也越来越复杂，对网上商城系统带来了极大的考验。所以传统的架构模式渐渐难以满足现状，因此出现了垂直应用架构，将一个系统拆分为多个独立的服务，每个服务各司其职，而 Spring Cloud 就是用来管理这些独立的服务，负责各个服务间的治理和通信。随着互联技术的不断迭代，目前电商系统已经非常强大和稳定，足以满足用户各大需求。

1.3 选题意义

在生活与工作节奏非常快的当今社会，很多上班族没有那么多时间去实体店购物。但网上商城不同，用户可以随时随地进行购物而不受时间和地点的影响，

大大提高了购物的效率和体验，也节省了消费者购物的时间和精力。网上购物商城在依托于互联网发展的基础上，将商品实体信息转化为信息化的数据，通过多媒体展示给不同的消费者，无论是在时间上还是空间上，都是一种营销模式的升级。这种线上的购物模式比线下购物更能给用户带来一种便捷的消费体验，并且这种购物模式正以其全新的运营方式改变着人们的生活方式。因此，研究出一款能让用户满意和喜爱的购物商城的是一件非常有意义且有必要的事情。与淘宝、京东等其他各大电商平台不同的是，聚美购物商城是专门为年轻人而设计的购物平台。在这里不仅可以购买到日常所需要的物品，而且聚美商城还推出商家版，方便商家在平台上发布自己的商品。

2 可行性分析

2.1 技术可行性

聚美购物商城前端使用 Vue+Element UI 框架作为页面开发，功能强大且页面精美。后端是由微服务框架 Spring Cloud 构建的一个分布式系统，Web 层采用 Spring MVC 框架，数据持久层采用 MySQL+Redis 架构，系统整体分工明确，并发能力和扩展性较高。

（1）Vue：Vue 是一个基于 JavaScript 开发的渐进式前端框架，功能强大且容易上手，而且能和第三方库完美兼容。

（2）Spring Cloud：Spring Cloud 是微服务一站式解决框架，它对市面上比较成熟且完善的服务框架进行了封装，提供给开发者一套简单且功能强大的组件，如配置中心、服务调用、负载均衡等。

（3）Spring MVC：Spring MVC 是 Spring 提供的 MVC 框架，它解决 Web 开发中的常见问题，比如参数接收、文件上传、表单验证等。

（4）MySQL：MySQL 是最流行的开源数据库之一，其以简单、高效、可靠的特点被各大网站使用。

（5）Redis：Redis 作为一个内存数据库，读写效率非常高效，目前已被各大公司作为分布式缓存中间件。

2.2 经济可行性

虽然系统在开发和运营上需要投入比较多的资金，但若是能完成开发，将有利于帮助企业完成快速转型。不仅能节约线下运营成本，还能更好的预测市场和及时调整营销策略，让企业能在激烈的市场竞争中取得优势。所以从长远的利益考虑，该系统的效益将远远高于投入。

2.3 操作可行性

用户不需要具备专业知识，也不需要用户去学习相关内容，只需要通过 Google 浏览器就可以轻松访问系统。系统具备优质的界面和良好的交互，操作简单且容易使用，不需要太多复杂的操作，只需要点点鼠标既可轻轻松松的完成在线购物。

3 系统需求分析

3.1 系统功能概述

本系统主要实现的是一个购物商城，主要分为用户前台和商家管理后台。用户前台主要实现了登录注册、找回密码、用户信息管理、收货地址管理、搜索商品、浏览商品、购买商品、购物车管理、用户订单管理、订单支付、支付记录管理等功能。商家管理后台的主要实现了登录注册、商家信息管理、商品信息管理、活动商品管理、商家订单管理等功能。系统功能结构图如图 3-1 所示。

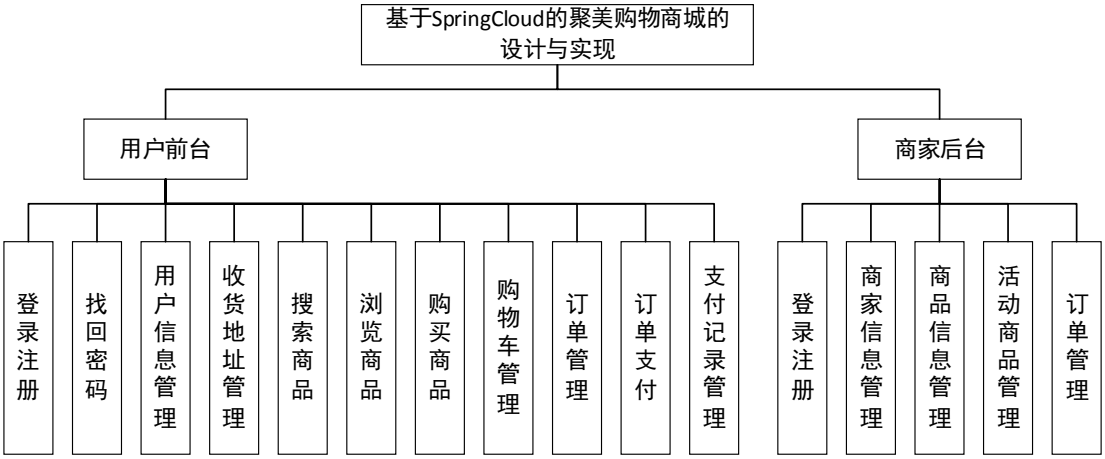

图 3-1 系统功能结构图

3.2 功能需求

聚美购物商城主要分为用户前台系统和商家后台系统，系统核心功能如下。

（1）登录注册

用户可以通过注册功能来创建一个属于自己的账号，其需要提供邮箱号和验证码进行注册。然后可以通过登录功能登录系统，系统会验证其邮箱密码并给出对应的提示。登录注册用例图如图 3-2 所示。

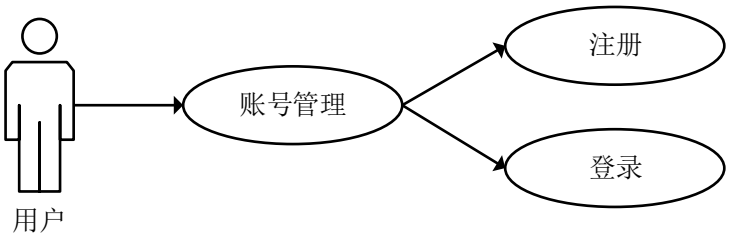

图 3-2 登录注册用例图

（2）用户信息管理

用户可以通过用户信息管理功能管理自己的信息，主要信息包括用户名、生

日、密码以及收货地址等。用户信息管理用例图如图 3-3 所示。

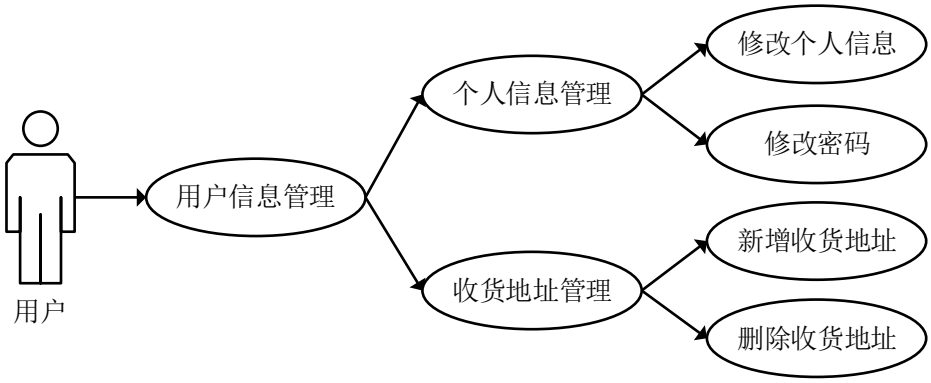

图 3-3 用户信息管理用例图

（3）购物车管理

用户可以在商品详情页将商品添加至购物车，也可以对购物车中的商品进行修改库存、删除商品、下单等操作。购物车管理用例图如图 3-4 所示。

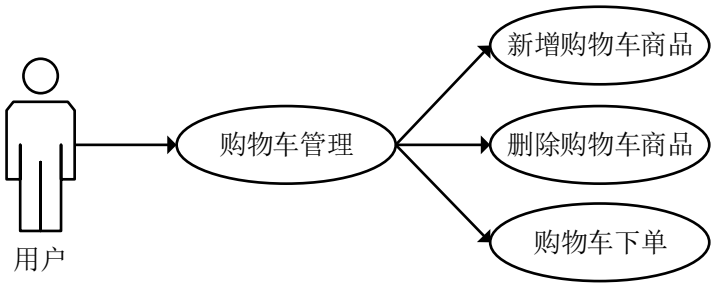

图 3-4 购物车管理用例图

（4）购买商品

用户可以在商品列表、活动商品列表、购物车列表中购买对应的商品，系统会根据用户选择的商品信息生成订单，用户确认订单信息无误后，选择收货地址即可付款购买。购买商品用例图如图 3-5 所示。

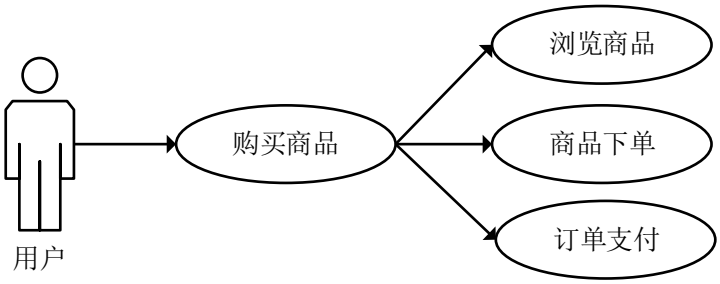

图 3-5 购买商品用例图

（5）用户订单管理

用户可以在订单管理界面查找到所有与该用户有关的订单，也可以支付还未支付的订单或删除已完成的订单。用户订单管理用例图如图 3-6 所示。

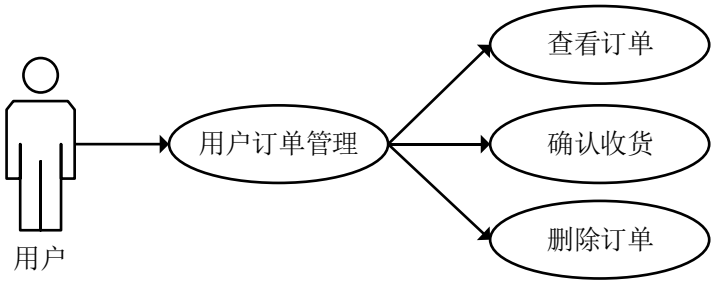

图 3-6 用户订单管理用例图

（6）商品管理

商家可以在商品管理界面查找到所有与该商家有关的商品，也可以对商品或活动商品进行添加、修改、删除等操作。商品管理用例图如图 3-7 所示。

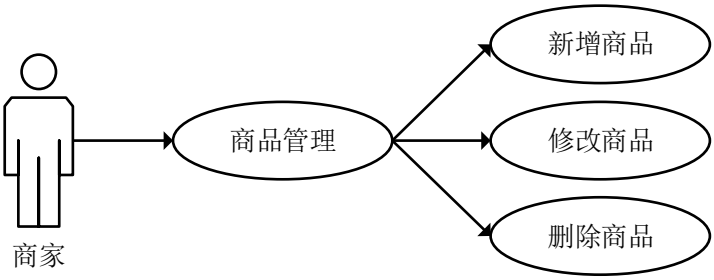

图 3-7 商品管理用例图

（7）商家订单管理

商家可以在订单管理界面查找到所有与该商家有关的订单，也可以对未发货的订单进行发货。商家订单管理例图如图 3-8 所示。

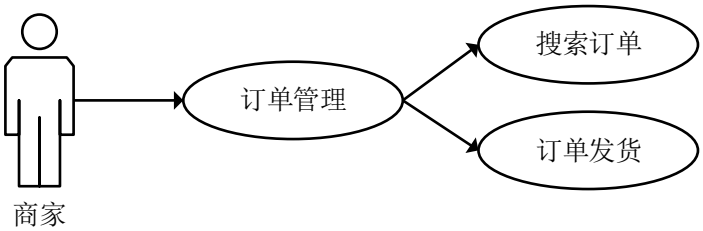

图 3-8 商家订单管理用例图

3.3 数据需求

在聚美购物商城系统中，主要的实体有商家信息，用户信息，收货地址，商品目录，商品公有属性，商品私有属性，活动商品，购物车商品，订单信息，订单商品，支付记录共 11 个实体。

（1）商家信息实体属性

商家信息实体属性包括商家编号，店名，邮箱，手机号，密码，头像，创建时间以及更新时间，商家信息实体属性图如图 3-9 所示。

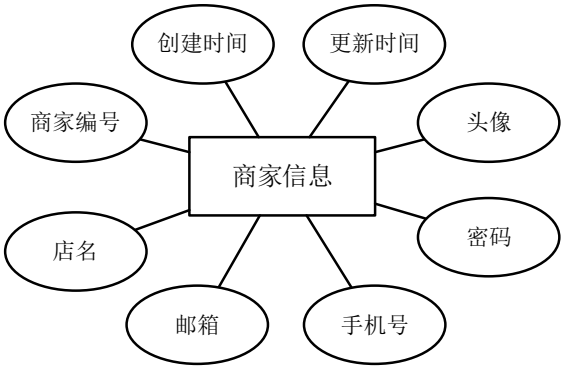

图 3-9 商家信息实体属性图

（2）用户信息实体属性

用户信息实体属性包括用户编号，邮箱，手机号，用户名，头像，密码，性别，生日，创建时间以及更新时间，用户信息实体属性图如图 3-10 所示。

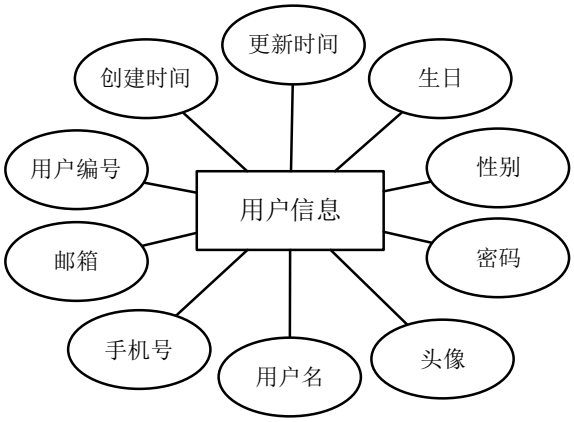

图 3-10 用户信息实体属性图

（3）收货地址实体属性

收货地址实体属性包括收货地址编号，用户编号，收货人姓名，手机号，收货地址，详细地址，邮编，是否默认地址，更新时间以及创建时间，收货地址实体属性图如图 3-11 所示。

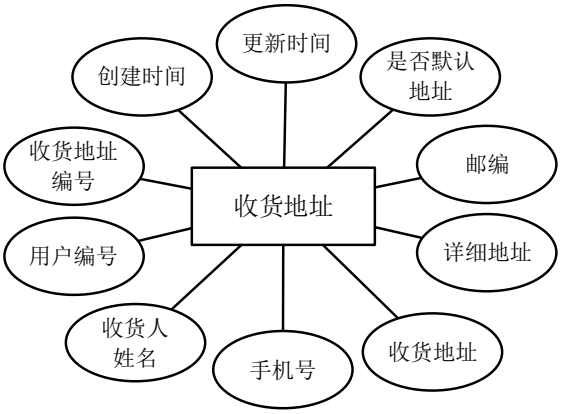

图 3-11 收货地址实体属性图

（4）商品目录实体属性

商品目录实体属性包括目录编号，目录名，上级目录编号，目录图标，目录链接，是否显示，是否为菜单，商品属性模板，更新时间以及创建时间，商品目录实体属性图如图 3-12 所示。

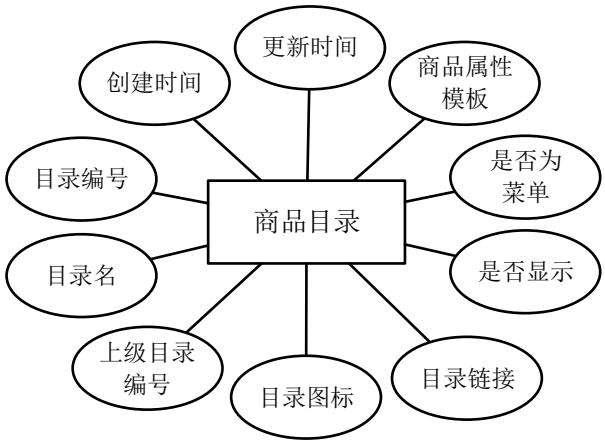

图 3-12 商品目录实体属性图

（5）商品公有属性实体属性

商品公有属性实体属性包括商品公有属性编号，商品名称，商家编号，店名，一级目录，二级目录，三级目录，图片，私有属性模板，销量，是否已删除，更新时间以及创建时间，商品公有属性实体属性图如图 3-13 所示。

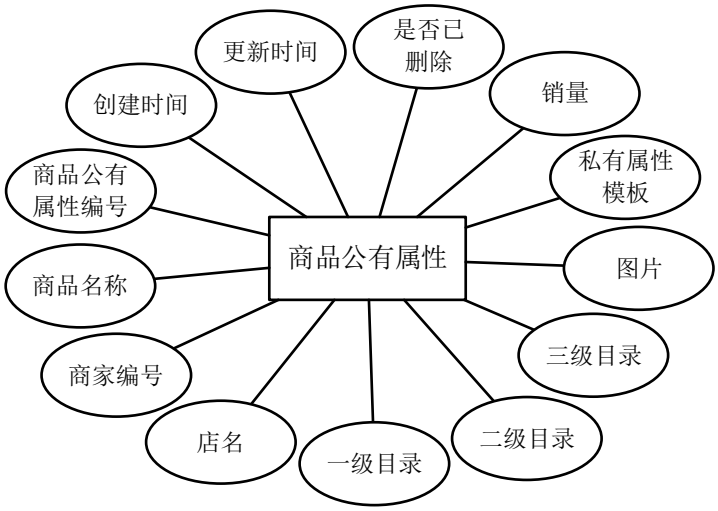

图 3-13 商品公有属性实体属性图

（6）商品私有属性实体属性

商品私有属性实体属性包括商品私有属性编号，商品公有属性编号，商品私有属性，图片，现价，原价，库存，预警库存，销量，冻结库存，更新时间以及创建时间，商品私有属性实体属性图如图 3-14 所示。

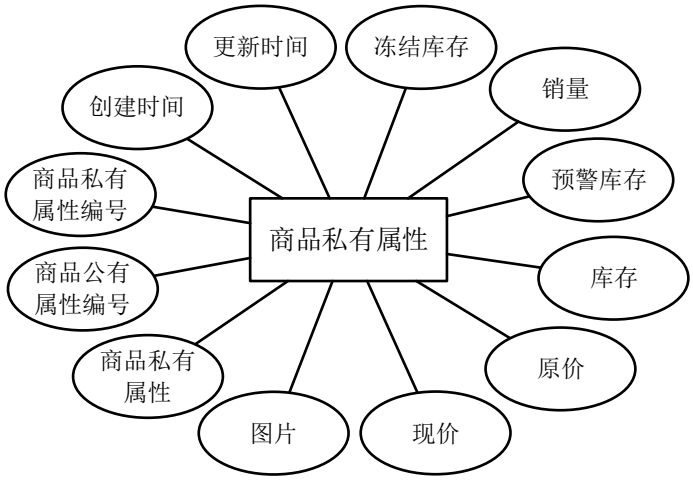

图 3-14 商品私有属性实体属性图

（7）活动商品实体属性

活动商品实体属性包括商品私有属性编号，商品公有属性编号，商家编号，商品私有属性，图片，原价，现价，活动开始时间，活动结束时间，数量，更新时间以及创建时间，活动商品实体属性图如图 3-15 所示。

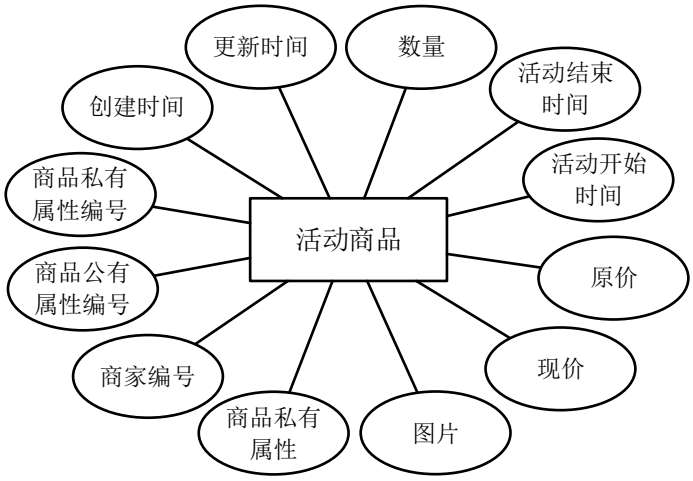

图 3-15 活动商品实体属性图

（8）购物车商品实体属性

购物车商品实体属性包括编号，商品私有属性编号，用户编号，商品名称，原价，数量，更新时间以及创建时间，购物车商品实体属性图如图 3-16 所示。

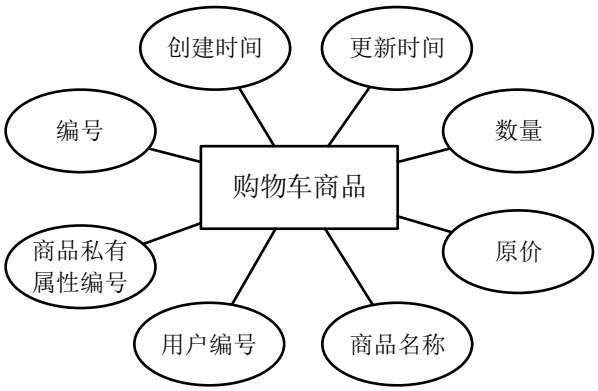

图 3-16 购物车商品实体属性图

（9）订单信息实体属性

订单信息实体属性包括订单编号，商家编号，用户编号，收货地址编号，总价格，订单备注，订单类型，订单状态，支付记录编号，支付类型，是否已删除，更新时间以及创建时间，订单信息实体属性图如图 3-17 所示。

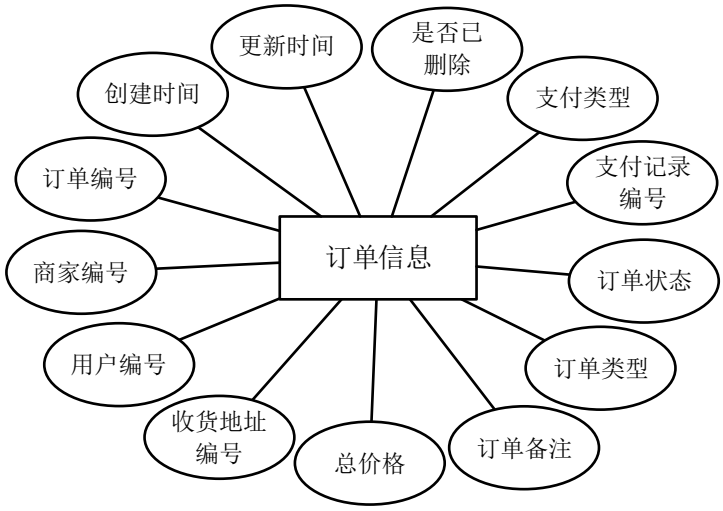

图 3-17 订单信息实体属性图

（10）订单商品实体属性

订单商品实体属性包括有编号，订单编号，商品公有属性编号，商品私有属性编号，商品名称，商品私有属性，图片，单价，数量，总价格，更新时间以及创建时间，订单商品实体属性图如图 3-18 所示。

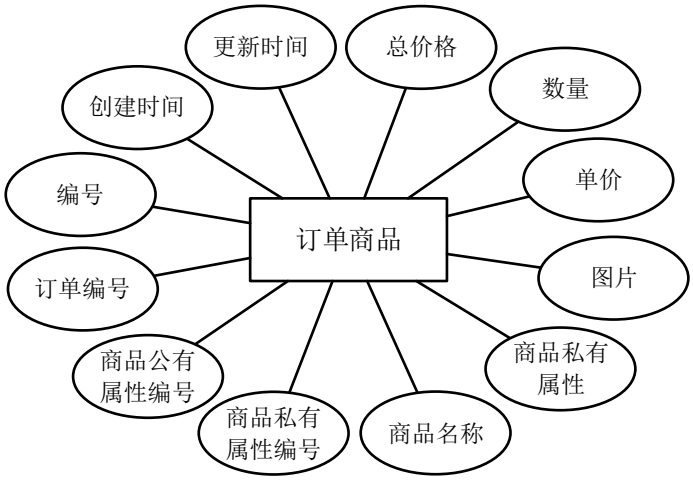

图 3-18 订单商品实体属性图

（11）支付记录实体属性

支付记录实体属性包括支付记录编号，用户编号，支付金额，支付状态，支付类型，支付时间，支付流水号，备注，更新时间以及创建时间，支付记录实体属性图如图 3-19 所示。

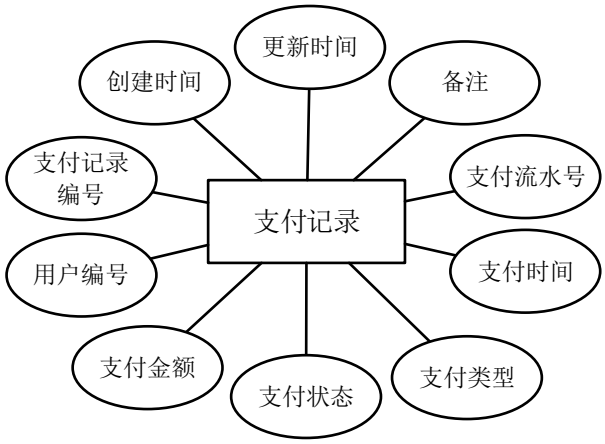

图 3-19 支付记录实体属性图

（12）总体 E-R 图

一个商家可有多个商品公有属性，一个商品公有属性只属于一个商家，所以商家和商品公有属性之间是一对多关系。一个商品分类可有多个商品公有属性，一个商品公有属性只属于一个商品分类，所以商品分类和商品公有属性是一对多关系。一个商品公有属性可有多个私有商品属性，一个商品私有属性只属于一个商品公有属性，所以商品公有属性和商品私有属性是一对多关系。一个商家可给多个商品私有属性添加活动，一个商品私有属性只能被一个商家添加活动，所以商家和商品私有属性是一对多关系。一个用户可对多个商品私有属性加入购物

车，一个商品私有属性也可以被多个用户加入购物车，所以用户和商品私有属性之间的加入购物车关系是多对多。一个用户可对多个商品私有属性下订单，一个商品私有属性也可以被多个用户下订单，所以用户和商品私有属性之间的下订单关系是多对多。一个订单可有多个订单商品，一个订单商品只属于一个订单，所以订单和订单商品之间是一对多关系。一个订单商品只属于一个商品私有属性，一个商品私有属性可属于多个订单商品，所以订单商品和商品私有属性是多对一关系。一个用户可有多个收货地址，一个收货地址只属于一个用户，所以用户和收货地址之间是一对多关系。一个订单只有一个收货地址，一个收货地址可属于多个订单，所以订单和收货地址是多对一关系。一个用户可以支付多个订单，一个订单只能被一个用户支付，所以用户和订单之间的支付关系是一对多。系统总体 E-R 图如图 3-20 所示。

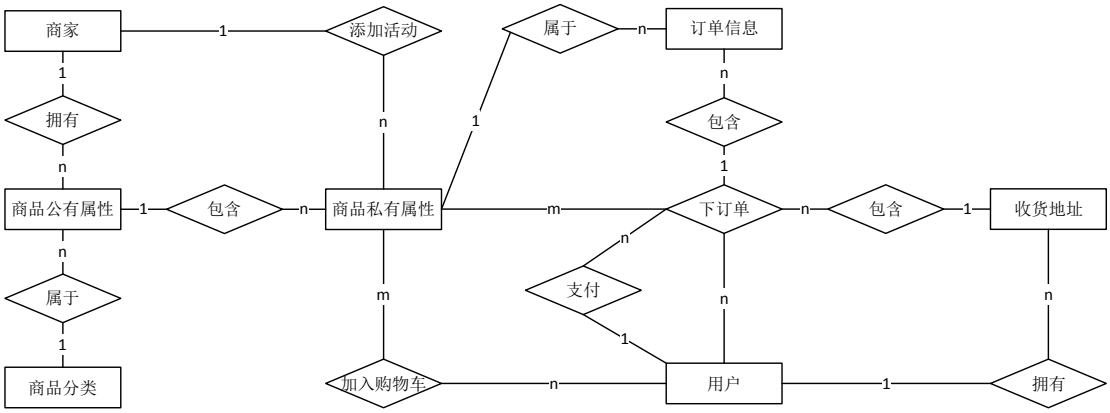

图 3-20 系统总体 E-R 图

4 系统总体设计

4.1 系统架构设计

聚美购物商城基于 Spring Cloud+Vue 实现，一共分为五个微服务，分别为用户服务、商品服务、活动商品服务、订单服务、支付服务，具体系统架构设计如图 4-1 所示。

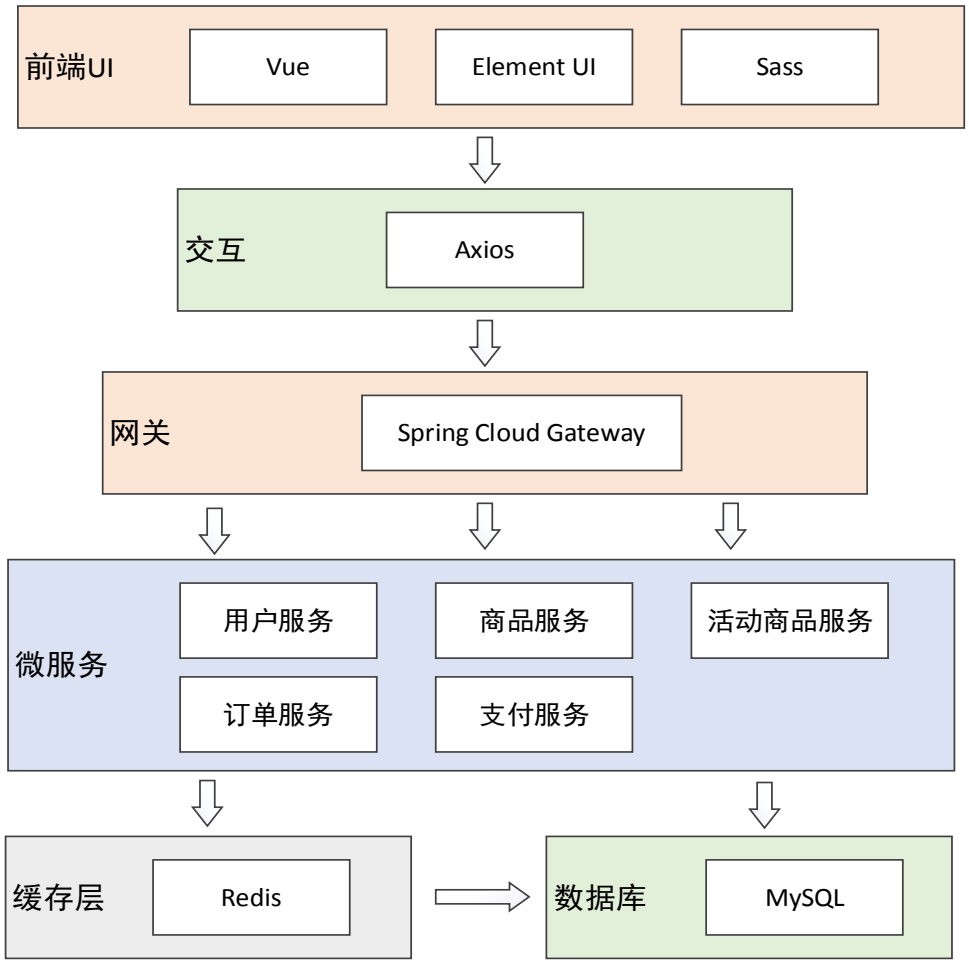

图 4-1 系统架构设计图

4.2 数据库设计

4.2.1 逻辑结构设计

商家信息（商家编号，店名，邮箱，手机号，密码，头像，更新时间，创建时间）

用户信息（用户编号，邮箱，手机号，用户名，头像，密码，性别，生日，更新时间，创建时间）

收获地址（收货地址编号，用户编号，收货人姓名，手机号，收货地址，详细地址，邮编，是否默认地址，更新时间，创建时间）

商品目录（目录编号，目录名，上级目录编号，目录图标，目录链接，是否显示，是否为菜单，商品属性模板，更新时间，创建时间）

商品公有属性（商品公有属性编号，商品名称，商家编号，店名，一级目录，二级目录，三级目录，图片，私有属性模板，销量，是否已删除，更新时间，创建时间）

商品私有属性（商品私有属性编号，商品公有属性编号，商品私有属性，图片，现价，原价，库存，预警库存，销量，冻结库存，更新时间，创建时间）

活动商品（商品私有属性编号，商品公有属性编号，商家编号，商品私有属性，图片，原价，现价，活动开始时间，活动结束时间，数量，更新时间，创建时间）

购物车商品（编号，商品私有属性编号，用户编号，商品名称，原价，数量，更新时间，创建时间）

订单信息（订单编号，商家编号，用户编号，收货地址编号，总价格，订单备注，订单类型，支付类型，订单状态，支付记录编号，是否已删除，更新时间，创建时间）

订单商品（编号，订单编号，商品公有属性编号，商品私有属性编号，商品名称，商品私有属性，图片，单价，数量，总价格，更新时间，创建时间）

支付记录（支付记录编号，用户编号，支付金额，支付状态，支付类型，支付时间，支付流水号，备注，更新时间，创建时间）

4.2.2 数据库表结构设计

（1）商家信息表

表 4-1 商家信息表

字段	类型	约束条件	说明
merchant_id	char(32)	主键	商家编号
shop_name	varchar(64)	非空	店名
email	char(32)	唯一、非空	邮箱
telephone	char(11)		手机号
password	varchar(32)	非空	密码
avatar	varchar(96)	非空	头像
update_time	datetime	非空	更新时间

create_time	datetime	非空	创建时间

（2）用户信息表

表 4-2 用户信息表

字段	类型	约束条件	说明
user_id	char(32)	主键	用户编号
email	varchar(32)	唯一、非空	邮箱
telephone	char(11)		手机号
username	varchar(32)	非空	用户名
avatar	varchar(96)	非空	头像
password	varchar(32)	非空	密码
gender	varchar (2)		性别
birthday	datetime		生日
update_time	datetime	非空	更新时间
create_time	datetime	非空	创建时间

（3）收货地址表

表 4-3 收货地址表

字段	类型	约束条件	说明
shipping_addr_id	char(32)	主键	收货地址编号
user_id	char(32)	外键、非空	用户编号
consignee	varchar(16)	非空	收货人姓名
telephone	char(11)	非空	手机号
address	varchar(32)	非空	收货地址
detail	varchar(64)	非空	详细地址
postcode	char(6)		邮编
is_default	tinyint	非空	是否默认地址
update_time	datetime	非空	更新时间
create_time	datetime	非空	创建时间

（4）商品目录表

表 4-4 商品目录表

字段	类型	约束条件	说明
category_id	char(32)	主键	目录编号
name	varchar (64)	非空	目录名
parent_id	char (32)	非空	上级目录编号
icon	varchar(96)	非空	目录图标
url	varchar (96)	非空	目录链接
is_show	tinyint	非空	是否显示
is_menu	tinyint	非空	是否为菜单
template	json		商品属性模板
update_time	datetime	非空	更新时间
create_time	datetime	非空	创建时间

（5）商品公有属性表

表 4-5 商品公有属性表

字段	类型	约束条件	说明
spu_id	char(32)	主键	商品公有属性编号
spu	varchar(32)	非空	商品名称
merchant_id	char(32)	外键、非空	商家编号
shop_name	varchar (32)	非空	店名
category1_id	char(32)	外键、非空	一级目录
category2_id	char(32)	外键、非空	二级目录
category3_id	char(32)	外键、非空	三级目录
images	json	非空	图片
sku_template	json		私有属性模板
sale_num	int	非空	销量
is_deleted	tinyint	非空	是否已删除
update_time	datetime	非空	更新时间
create_time	datetime	非空	创建时间

（6）商品私有属性表

表 4-6 商品私有属性表

字段	类型	约束条件	说明
sku_id	char(32)	主键	商品私有属性编号
spu_id	char(32)	外键、非空	商品公有属性编号
sku	json	非空	商品私有属性
images	json	非空	图片
price	decimal(10,2)	非空	现价
old_price	decimal(10,2)	非空	原价
num	int	非空	库存
alter_num	int	非空	预警库存
sale_num	int	非空	销量
freeze_num	int	非空	冻结库存
update_time	datetime	非空	更新时间
create_time	datetime	非空	创建时间

（7）活动商品表

表 4-7 活动商品表

字段	类型	约束条件	说明
sku_id	char(32)	主键	商品私有属性编号
spu_id	char (32)	外键、非空	商品公有属性编号
merchant_id	char(32)	外键、非空	商家编号
sku	json	非空	规格
image	json	非空	图片
price	decimal (10,2)	非空	现价
old_price	decimal (10,2)	非空	原价
start_time	datetime	非空	活动开始时间
end_time	datetime	非空	活动结束时间
num	int	非空	数量
update_time	datetime	非空	更新时间

create_time	datetime	非空	创建时间

（8）购物车商品表

表 4-8 购物车商品表

字段	类型	约束条件	说明
id	bigint	主键，自增	编号
sku_id	char(32)	外键、非空	商品私有属性编号
user_id	char(32)	外键、非空	用户编号
spu	varchar(32)	非空	商品名称
old_price	decimal (10,2)	非空	原价
num	int	非空	数量
update_time	datetime	非空	更新时间
create_time	datetime	非空	创建时间

（9）订单信息表

表 4-9 订单信息表

字段	类型	约束条件	说明
order_id	char(32)	主键	订单编号
merchant_id	char(32)	外键、非空	商家编号
user_id	char(32)	外键、非空	用户编号
shipping_address_id	char(32)	外键、非空	收货地址编号
total_price	decimal (10,2)	非空	总价格
remark	varchar(256)		订单备注
type	tinyint	非空	订单类型
status	tinyint	非空	订单状态
payment_id	char(32)	外键、非空	支付记录编号
pay_type	tinyint		支付类型
is_deleted	tinyint	非空	是否删除
update_time	datetime	非空	更新时间
create_time	datetime	非空	创建时间

（10）订单商品表

表 4-10 订单商品表

字段	类型	约束条件	说明
id	bigint	主键，自增	编号
order_id	char(32)	唯一，非空	订单编号
spu_id	char(32)	外键、非空	商品公有属性编号
sku_id	char(32)	外键、非空	商品是有属性编号
spu	varchar(64)	非空	商品名称
sku	json	非空	商品规格
image	varchar(96)	非空	图片
price	decimal (10,2)	非空	单价
num	int	非空	数量
total_price	decimal (10,2)	非空	总价
update_time	datetime	非空	更新时间
create_time	datetime	非空	创建时间

（11）支付记录表

表 4-11 支付记录表

字段	类型	约束条件	说明
payment_id	char(32)	主键	支付记录编号
user_id	char(32)	外键、非空	用户编号
money	decimal (10,2)	非空	支付金额
status	tinyint	非空	支付状态
pay_type	tinyint		支付类型
pay_time	datetime		支付时间
transaction_id	char(32)		支付流水号
remark	varchar(96)		备注
update_time	datetime	非空	更新时间
create_time	datetime	非空	创建时间

5 系统详细设计与实现

5.1 用户注册功能模块

5.1.1 用户注册功能模块概述

用户可以通过注册功能来创建属于自己的账号，首先点击右上角头像进入注册界面，根据表单提示输入个人信息，然后点击获取验证码，用户收到验证码后，将其填入表单后点击注册。此时后端会判断验证码是否正确，以及邮箱是否被使用，通过验证后完成注册。

5.1.2 用户注册功能模块流程设计

具体流程如图 5-1 所示。

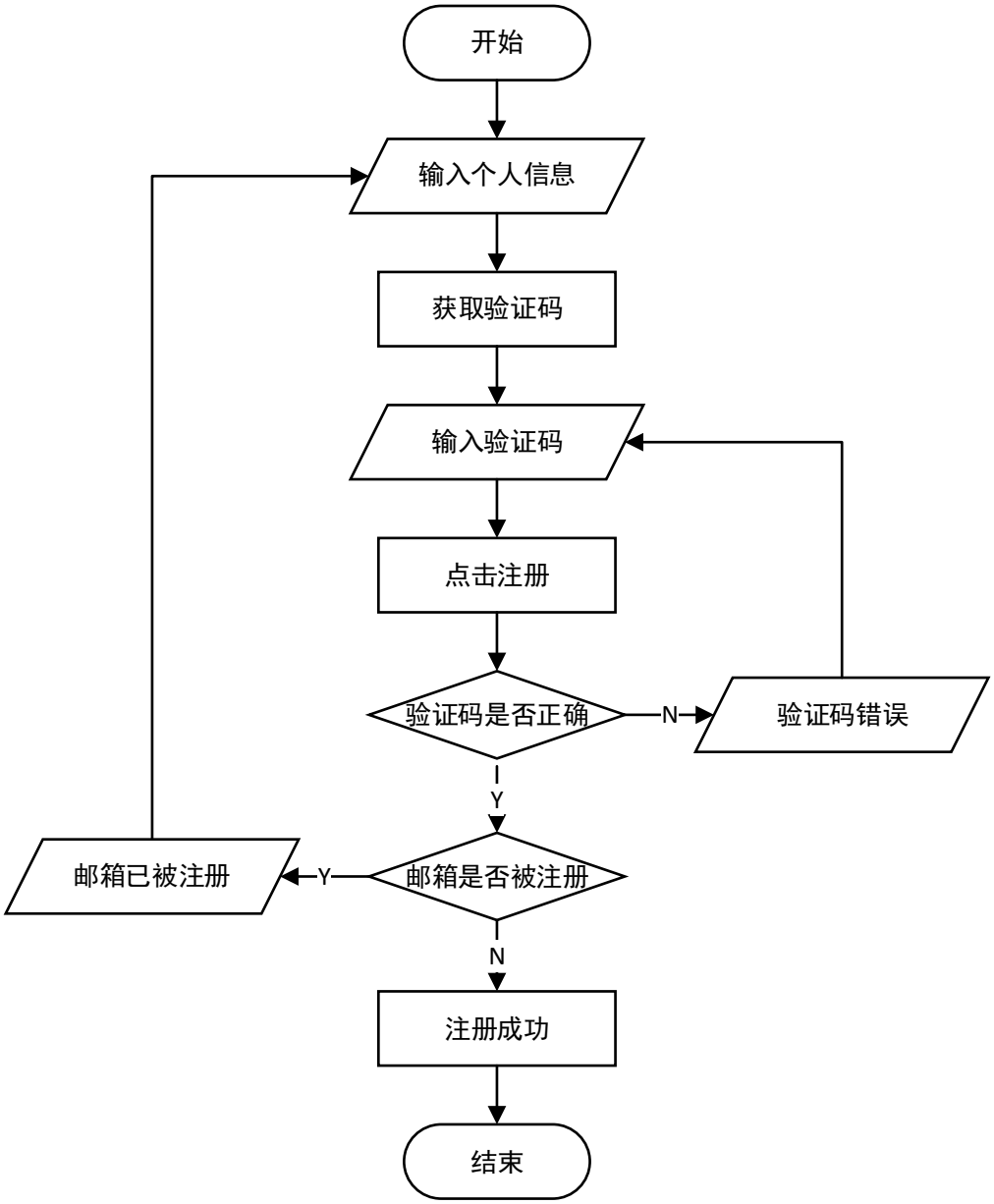

图 5-1 注册流程图

5.1.3 用户注册功能模块核心代码

核心代码如下所示：

```
public void register(RegisterRequest request) {
    String code = redisManager.get(
        RedisKey.REGISTER_USER_CODE.format(request.getEmail()));
    if (!Objects.equals(code, request.getVerifyCode())) {
        CastException.cast("验证码错误");
    }
    UserInfo isExist = userInfoMapper.selectByEmail(request.getEmail());
    if (isExist != null) {
        CastException.cast("邮箱已被注册");
    }
    UserInfo userInfo = BeanUtils.copy(request, UserInfo.class);
    userInfo.setUsername(request.getName());
    userInfo.setUserId(IdUtils.getId("user"));
    userInfoMapper.insert(userInfo);
}
```

5.2 用户登录功能模块

5.2.1 用户登录功能模块概述

用户可以通过注册的账号登录系统。首先点击右上角头像进入登录界面，根据提示输入邮箱和密码后点击登录。后端会根据邮箱和密码与数据库进行匹配，如果匹配成功，则返回 token 用来表示用户登录凭证，前端进行保存，后续的请求都会携带 token；若匹配不成功则返回账号和密码错误。

5.2.2 用户登录功能模块流程设计

具体流程如图 5-2 所示。

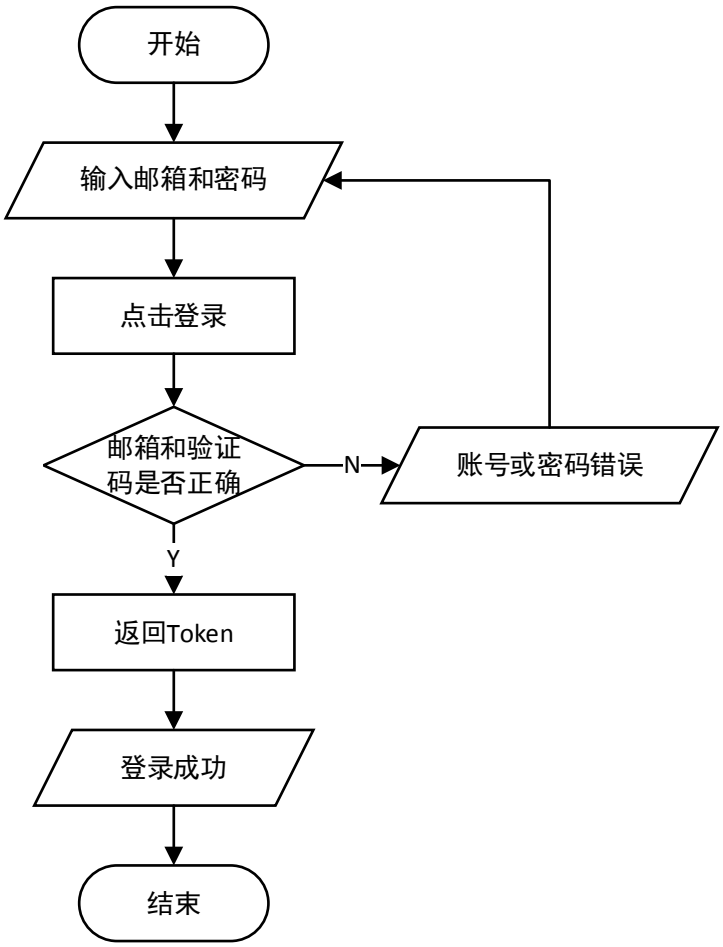

图 5-2 登录流程图

5.2.3 用户登录功能模块核心代码

核心代码如下所示：

```
public AuthToken login(LoginRequest request) {
    UserInfo user = userInfoMapper.login(request.getAccount(), request.getPassword());
    if (user == null) {
        CastException.cast("账号或密码错误");
    }
    String token = TokenUtils.generateToken(user.getUserId());
    AuthToken authToken = new AuthToken();
    authToken.setAccessToken(token);
    return authToken;
}
```

5.3 用户信息管理功能模块

5.3.1 用户信息管理功能模块概述

用户可以在个人信息页面查看并修改个人信息，前端会验证用户输入的数据是否符合要求，如果符合则提交表单保存至数据库，否则给出对应提示。

5.3.2 用户信息管理功能模块流程设计

具体流程如图 5-3 所示。

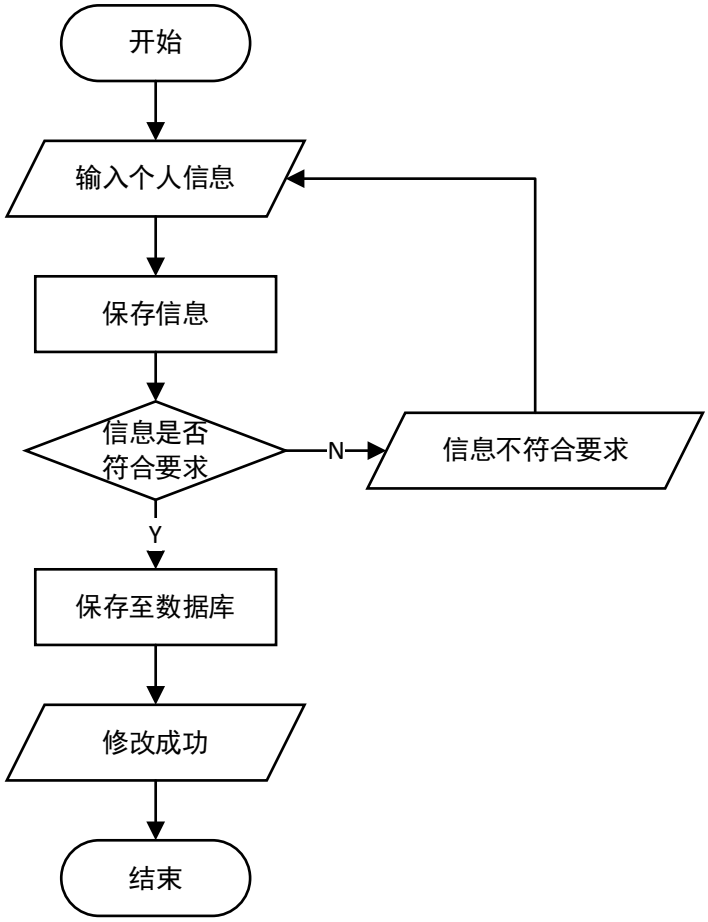

图 5-3 用户信息管理流程图

5.3.3 用户信息管理功能模块核心代码

核心代码如下所示：

```
public void updateUserInfo(UpdateUserInfoRequest request) {
    UserInfo userInfo = new UserInfo();
    userInfo.setUsername(request.getUsername());
    userInfo.setGender(request.getGender());
    userInfo.setBirthday(request.getBirthday());
```

```
        userInfoMapper.update(userInfo, new QueryWrapper<UserInfo>()
            .eq("user_id", AuthContext.get().getUserId())
        );
    }
```

5.4 收货地址管理功能模块

5.4.1 收货地址管理功能模块概述

在创建订单时，需要填写收货地址，收货地址主要内容包括地址信息、收货人、手机号等。用户可以进入个人信息收货地址管理页面，查看用户已有的收货地址，也可以对收货地址进行新增、编辑和删除操作。创建收货地址时，如果新增地址为默认地址，则会取消之前设置的默认地址。否则直接添加。

5.4.2 收货地址管理功能模块流程设计

具体流程如图 5-4 所示。

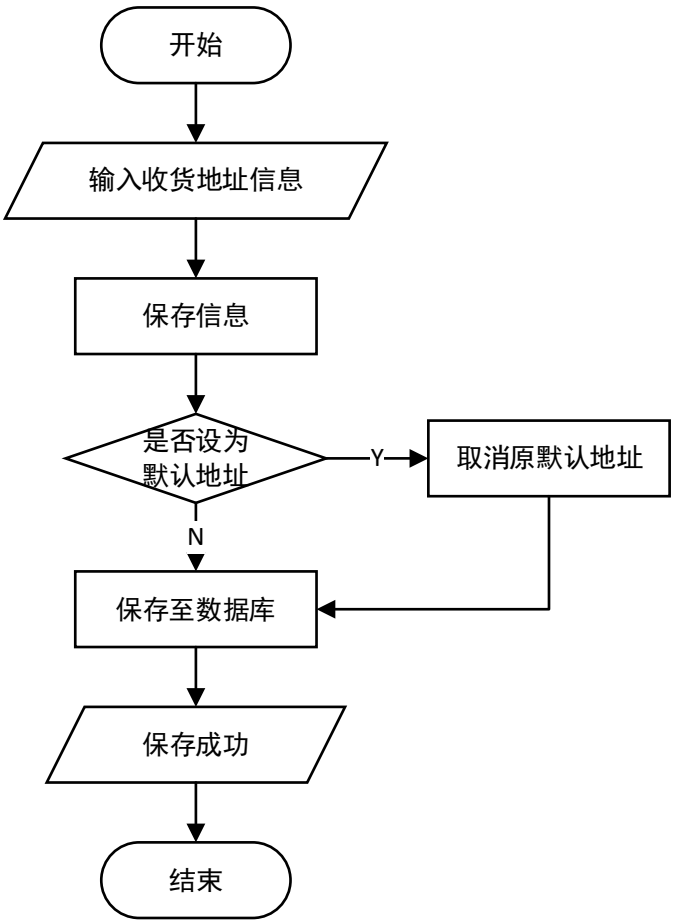

图 5-4 收货地址管理流程图

5.4.3 收货地址管理功能模块核心代码

核心代码如下所示：

```
public void addAddress(AddShippingAddressRequest request, String userId) {
    ShippingAddress address = BeanUtils.copy(request, ShippingAddress.class);
    if (address.getIsDefault()) {
        // 设置其他为 false
        shippingAddressMapper.cancelDefault(userId);
    }
    address.setUserId(userId);
    address.setShippingAddrId(IdUtils.getId("s_a"));
    // 若不是默认，判断是否存在其他默认地址
    if (!address.getIsDefault()) {
        int count = shippingAddressMapper.selectCount(
                    new QueryWrapper<ShippingAddress>()
                .eq("user_id", userId).eq("is_default", true)
        );
        address.setIsDefault(count == 0);
    }
    shippingAddressMapper.insert(address);
}
```

5.5 购物车商品管理功能模块

5.5.1 购物车商品管理功能模块概述

购物车可用来储存用户所需要购买的商品，也方便多个商品同时下单。用户可在商品详情页点击加入购物车按钮，也可以在购物车页面中修改商品库存、删除购物车商品，购买商品等操作。加入购物车时，系统会判断该条数据是否已存在，如果存在则只对库存进行递增操作，否则将对应的商品信息保存至数据库。

5.5.2 购物车商品管理功能模块流程设计

具体流程如图 5-5 所示。

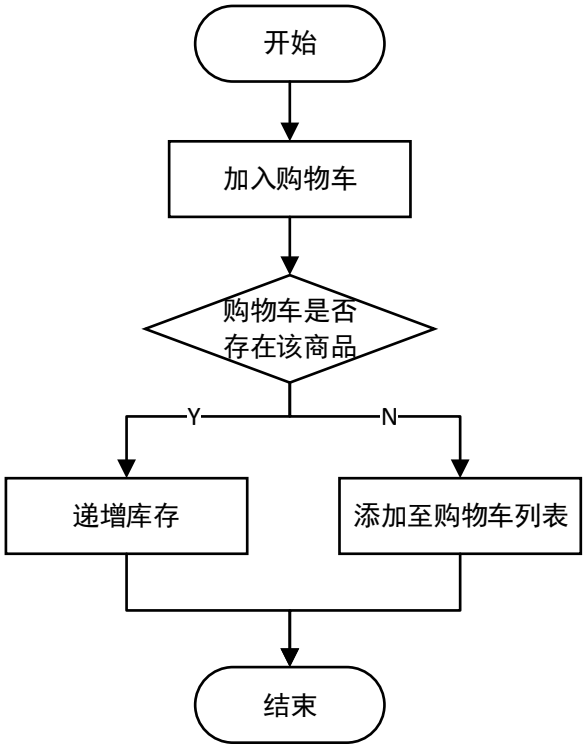

图 5-5 购物车管理流程图

5.5.3 购物车商品管理功能模块核心代码

核心代码如下所示：

```
public void addCart(AddCartRequest request, String userId) {
    CartDO select = cartMapper.selectByUserIdAndSkuId(userId, request.getSkuId());
    if (select != null) {   // 商品存在，则只添加库存
        select.setNum(select.getNum() + request.getNum());
        cartMapper.updateById(select);
        return;
    }
    RpcResult<SkuDTO> skuResult = skuFeign.getBySkuId(request.getSkuId());
    RpcResultUtils.checkNotNull(skuResult);
    SkuDTO skuDTO = skuResult.getData();
    CartDO cartDO = cartConvert.toCart(request);
    cartDO.setUserId(userId);
    cartDO.setOldPrice(skuDTO.getPrice());
    cartMapper.insert(cartDO);
}
```

5.6 购买商品功能模块

5.6.1 购买商品功能模块概述

用户可在商品详情页或者购物车中点击购买商品，然后进入确认下单页面，用户需要确认订单信息并选择收货地址，点击确认购买进入付款界面，扫码付款成功后，跳转至用户订单列表，完成下单操作。

5.6.2 购买商品功能模块流程设计

具体流程如图 5-6 所示。

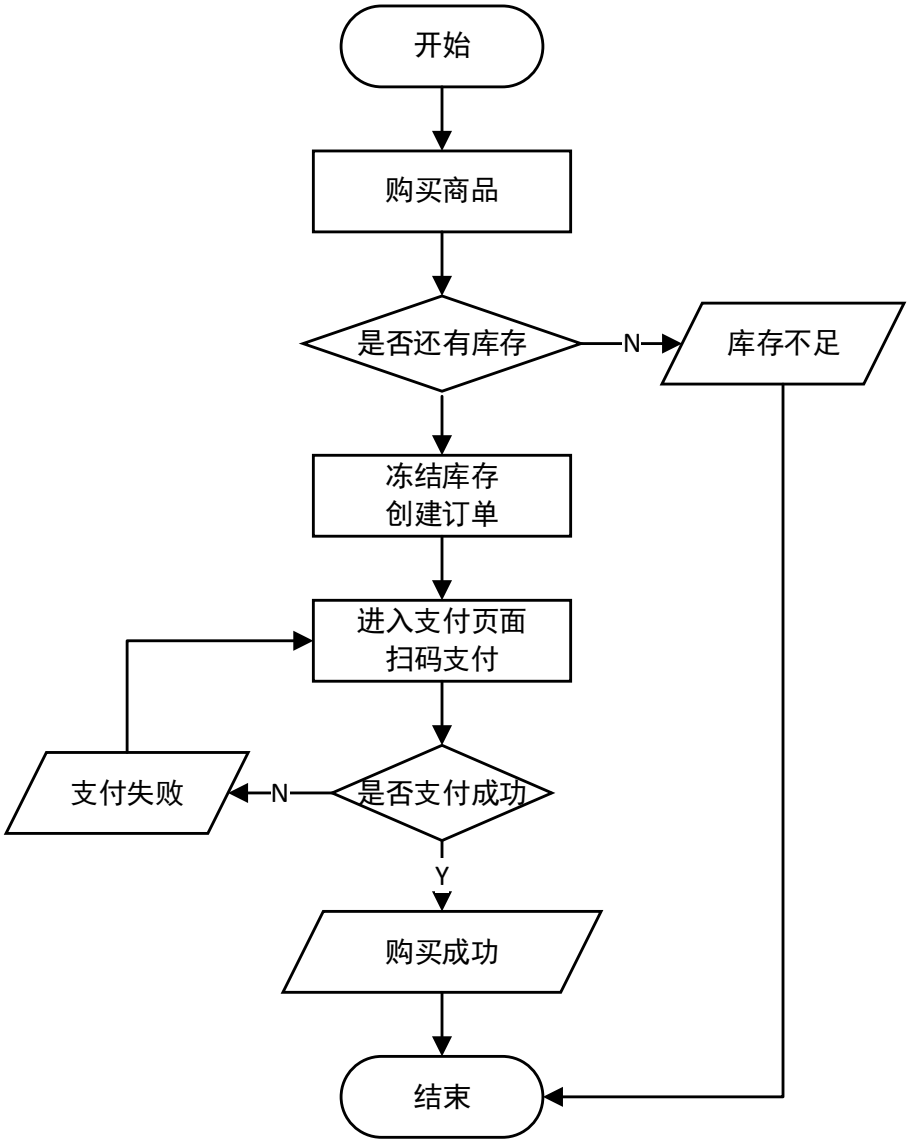

图 5-6 购买商品流程图

5.6.3 购买商品功能模块核心代码

核心代码如下所示：

```
public void createOrder(CreateOrderRequest message) {
```

```
        // 1 冻结库存
        RpcResult<Boolean> freezeResult = skuFeign.batchFreezeScore(argList);
        if (!freezeResult.isSuccess()) {
            // 订单创建失败
            redisManager.set(OmcRedisKey.ORDER_STATUS.format(
                    message.getPaymentId()), OrderStatus.FAILED);
            return;
        }
        // 2 保存至数据库
        for (OrderDO orderDO : orderDOList) {
            orderDO.setPaymentId(message.getPaymentId());
        }
        orderMapper.batchInsert(orderDOList);
        orderItemMapper.batchInsert(orderItemDOList);
        // 3 删除购物车
        if (message.getIsCart()) {
            cartMapper.batchDelete(message.getUserId(), skuIds);
        }
        // 4 添加支付记录
        RpcResult<String> paymentResult = orderPayRecordFeign.addRecord(addArg);
        // 5 修改订单状态为已创建
        redisManager.set(OmcRedisKey.ORDER_STATUS.format(
                    message.getPaymentId()), OrderStatus.PAYMENT);
    }
```

5.7 用户订单管理功能模块

5.7.1 用户订单管理功能模块概述

用户可以在订单管理界面查看用户订单信息，包括未付款、已付款、已完成等状态的订单信息。并可以支付未付款的订单，也可以删除已完成的订单等操作，删除的订单可以到回收站恢复。

5.7.2 用户订单管理功能模块流程设计

具体流程如图 5-7 所示。

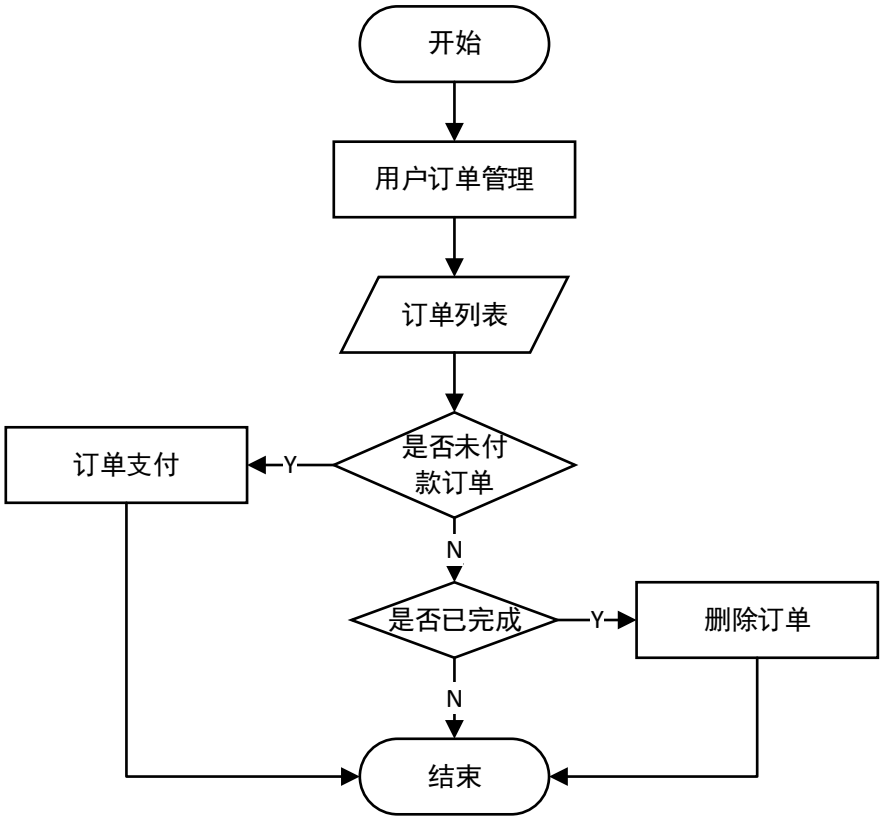

图 5-7 用户订单管理流程图

5.7.3 用户订单管理功能模块核心代码

核心代码如下所示：

```
// 查询订单
public PageInfo<OrderData> getUserOrder(GetUserOrderRequest request) {
    String userId = AuthContext.get().getUserId();
    PageHelper.startPage(request.getPage(), request.getSize());
    List<OrderDO> orderDOList = orderMapper
        .selectByUserIdAndStatus(userId, request.getStatus(), request.isDeleted());
    return convertToPage(orderDOList);
}
// 删除订单
public void deleteOrder(String orderId) {
    orderMapper.deleteOrder(orderId);
```

```
}
<update id="deleteOrder">
    update omc_order set is_deleted = true where order_id = #{orderId}
</update>
```

5.8 商品管理功能模块

5.8.1 商品管理功能模块概述

商家登录后台管理系统后，可以在商品管理界面，查看与该商家有关的所有商品。并且可以上架新商品和对已上架的商品进行修改、删除操作。此外，还可以点击活动按钮，填写活动时间、活动价格等信息，将商品设置为活动商品。

5.8.2 商品管理功能模块流程设计

具体流程如图 5-8 所示。

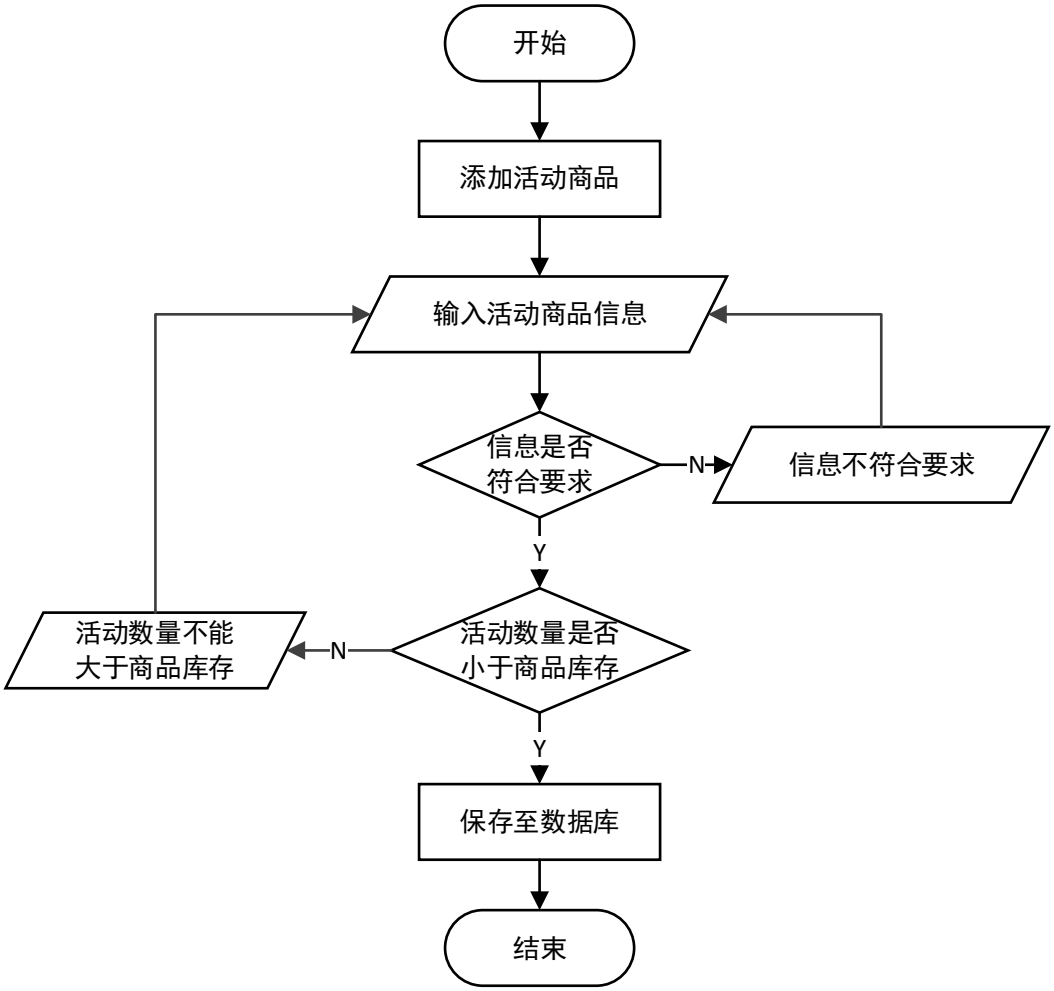

图 5-8 商品管理流程图

5.8.3 商品管理功能模块核心代码

核心代码如下所示：

```
public void addProduct(AddOrUpdateSkuRequest request) {
    RpcResult<SpuSkuDTO> spuSkuResult = skuFeign.getSpuSku(request.getSkuId());
    RpcResultUtils.checkNotNull(spuSkuResult);
    SpuSkuDTO spuSku = spuSkuResult.getData();
    if (request.getNum() > spuSku.getNum()) {
        CastException.cast("活动数量不能大于总库存");
    }
    SkuDO skuDO = BeanUtils.copy(request, SkuDO.class);
    skuMapper.insert(skuDO);
}
```

5.9 商家订单管理功能模块

5.9.1 商家订单管理功能模块概述

商家登录后台管理系统后，可以在订单管理界面，查看到与该商家有关的所有用户订单信息，并且可以对未发货的订单进行发货。

5.9.2 商家订单管理功能模块流程设计

具体流程如图 5-9 所示。

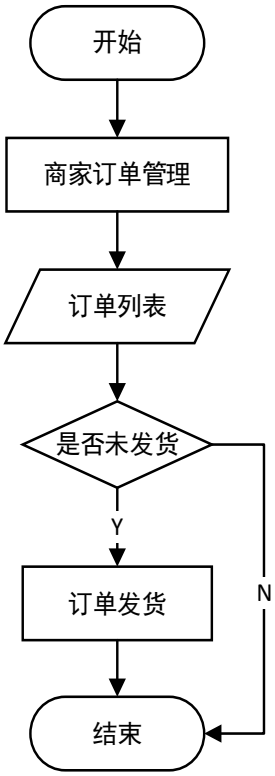

图 5-9 商家订单管理流程图

5.9.3 商家订单管理功能模块核心代码

核心代码如下所示：

```
@PostMapping("/delivery")
public void delivery(@RequestBody IdArg arg) {
    orderService.updateStatusByMerchant(arg.getId(), OrderStatus.RECEIVING);
}
public void updateStatusByMerchant(String orderId, Integer status) {
    String merchantId = AuthContext.get().getMerchantId();
    orderMapper.updateStatusByMerchant(orderId, merchantId, status);
}
// 更新数据库
<update id="updateStatusByMerchant">
    update omc_orderset status = #{status}
    where order_id = #{orderId} and merchant_id = #{merchantId}
</update>
```

6 系统功能测试

6.1 测试环境

本系统具体的测试环境如表 6-1 所示。

表 6-1 测试环境列表

系统：Window 10
处理器：Inter(R) Core (TM) i7-6700HQ
内存：16.00GB
软件运行环境：JRE 1.8.0_212
数据库：Redis、MySQL
语言：Java
浏览器：Chrome 79.0.4390.115

6.2 功能测试

6.2.1 用户注册功能测试

进入系统首页，点击右上角头像，会弹出一个登录框，点击注册按钮切换至注册页面，根据表单输入用户名、邮箱、密码和验证码进行注册。测试用例如表 6-2 所示。

表 6-2 注册功能验证

操作	预期结果	实际结果
输入不完整的信息	提示信息不完整	如图 6-1 所示
输入完整的信息和错误验证码	提示验证码错误	如图 6-2 所示
输入的邮箱已存在	提示邮箱已被注册	如图 6-3 所示
输入完整的信息和正确验证码	注册成功	如图 6-4 所示

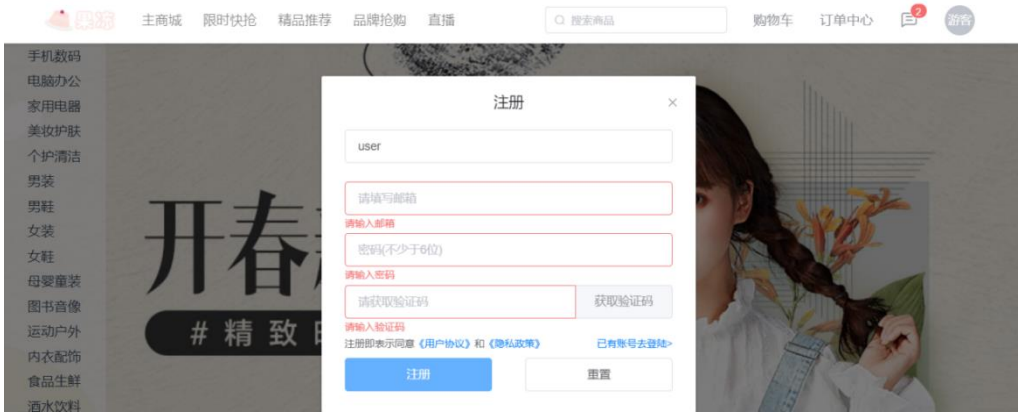

图 6-1 提示信息不完整

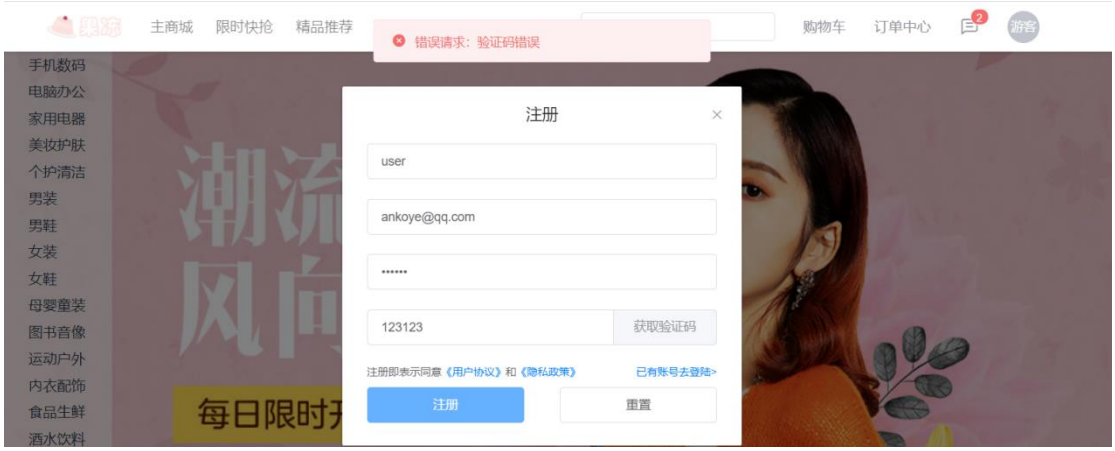

图 6-2 验证码错误

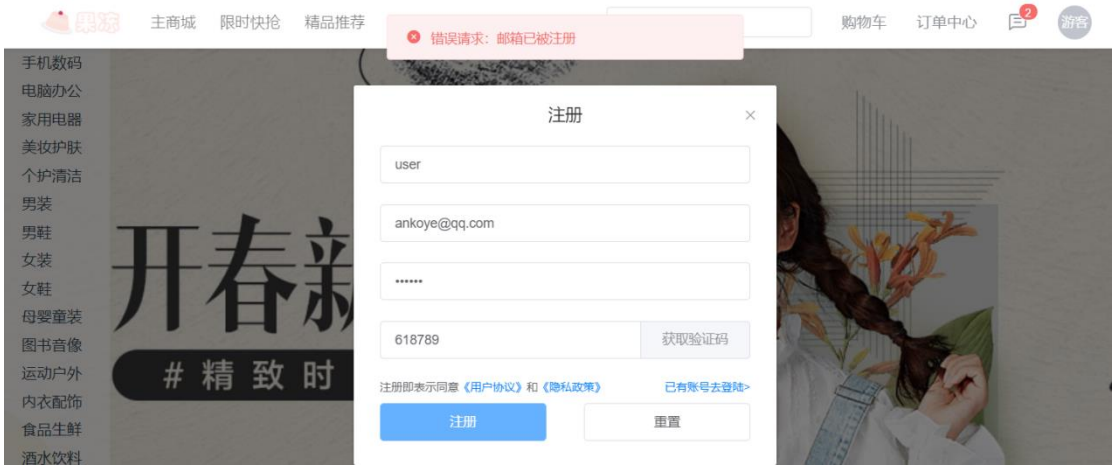

图 6-3 邮箱已被注册

图 6-4 注册成功

6.2.2 用户登录功能测试

进入系统首页，点击右上角头像，会弹出一个登录框，根据提示输入账号密码进行登录，如果邮箱和密码正确，则返回登录成功，否则返回用户名或密码错误。测试用例如表 6-3 所示。

表 6-3 登录验证

操作	预期结果	实际结果
输入错误的账号或密码	用户名或密码错误	如图 6-5 所示
输入正确的账号或密码	登录成功	如图 6-6 所示

图 6-5 用户名或密码错误

图 6-6 登录成功

6.2.3 用户信息管理功能测试

用户登录系统，点击右上角头像，进入到个人信息页面，用户可修改的信息包括用户名、性别和生日。如果输入的信息合法，则保存成功；否则页面提示某项信息不完整。测试用例如表 6-4 所示。

表 6-4 用户信息管理验证

操作	预期结果	实际结果
输入不完整的信息	页面提示信息不完整	如图 6-7 所示
输入完整的信息	更新成功	如图 6-8 所示

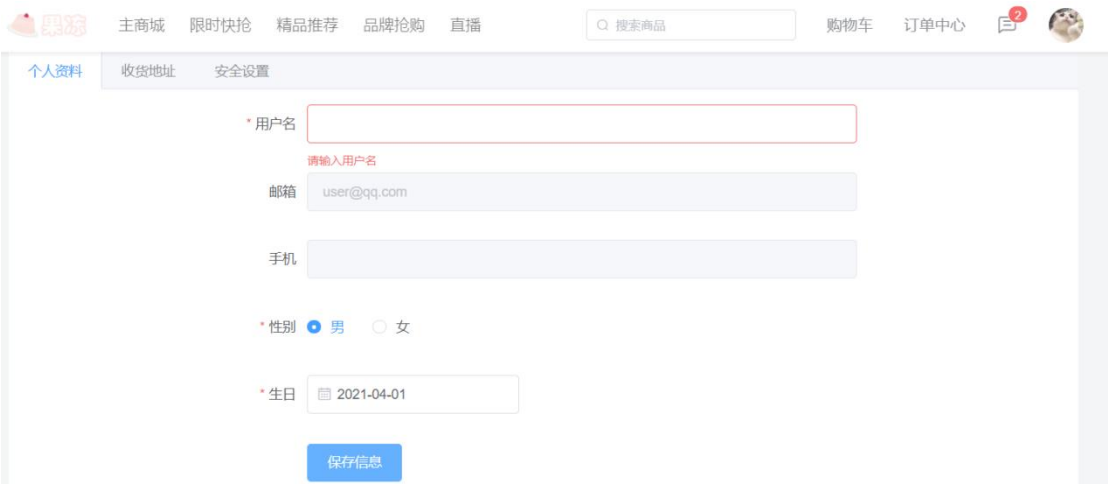

图 6-7 用户信息不完整

图 6-8 修改成功

6.2.4 收货地址管理功能测试

用户登录后点击右上角头像进入个人信息收货地址页面，如图 6-9 所示。首先根据提示输入完整的地址信息，若勾选了默认地址选项，则系统会取消原默认地址，设置当前新增的为默认地址；否则直接添加。测试用例如表 6-5 所示。

表 6-5 收货地址管理验证

操作	预期结果	实际结果
新增地址是默认地址	撤销原默认地址，设置新增地址为默认地址	如图 6-10 所示
新增地位不是默认地址	原默认地址信息不变	如图 6-11 所示

图 6-9 修改成功

图 6-10 新增默认地址

图 6-11 新增地址

6.2.5 购物车商品管理功能测试

首先进入商品详情页面，如图 6-12 所示，点击加入购物车。如果购物车中不存在该商品，将该商品添加至购物车列表中；否则直接递增该商品库存。测试用例如表 6-6 所示。

表 6-6 购物车商品管理验证

操作	预期结果	实际结果
购物车中不存在该商品	新增商品	如图 6-13 所示
购物车中存在该商品	递增商品库存	如图 6-14 所示

图 6-12 商品详情页

图 6-13 新增商品

图 6-14 递增商品库存

6.2.6 购买商品功能测试

从商品详情页或者从购物车中选择商品点击立即购买，进入确认订单信息页面，如图 6-15 所示；确认信息并选择对应的收货地址，点击确认并付款，进入

选择支付方式页面，如图 6-16 所示；选择支付宝支付，进入支付宝支付页面，如图 6-17 所示；扫描支付完成跳转至用户订单列表页，如图 6-18 所示。

图 6-15 确认订单信息

图 6-16 选择支付方式

图 6-17 支付宝支付

图 6-18 用户订单列表页

6.2.7 用户订单管理功能测试

登录系统后，点击右上角订单中心，进入用户订单列表页面，如图 6-19 所示；点击确认收货按钮可将未收货状态的订单修改为已完成状态，如图 6-20 所示；点击删除按钮，既可将完成或交易关闭状态的订单删除，被删除的订单存入订单回收站中，如图 6-21 所示。

图 6-19 用户订单列表页

图 6-20 确认收货

图 6-21 订单回收站

6.2.8 商品管理功能测试

登录商家后台系统，进入商品管理，既可看到当前商家所有的商品信息，如图 6-22 所示；点击商品活动按钮，根据表单填写活动信息，点击立即添加，如果活动商品数小于库存数量，则将商品添加至活动列表，如图 6-23 所示，否则提示添加失败，如图 6-24 所示。

图 6-22 商品列表

图 6-23 活动列表

图 6-24 活动商品数不能大于库存数

6.2.9 商家订单管理功能测试

登录商家后台系统，进入订单管理界面，既可看到所有与商家有关的订单，如图 6-25 所示；点击去发货按钮，既可对未发货状态的订单进行发货，如图 6-26 所示。

图 6-25 商家订单列表

图 6-26 订单发货

7 总结

随着毕业季的来临，我的毕业设计也画上了圆满的句号。从系统的设计到实现，总共花了三个月的时间，在这段时间里，我学到的东西却很多。首先，巩固了以前学过的基础知识和技术框架，也通过查阅专业书籍和相关资料，学到了很多新知识。其次，培养了自己整体规划的能力，能够对系统整体有较大的把控能力，感觉自己收获颇丰。

当然，在整个项目的开发过程中也遇到了一些技术难题和自己的一些不足之处，比如对前面所学的知识掌握的不够牢固，不能灵活应用等。这让我体会到只有理论知识还是远远不够的，要做到熟练使用，还需要多多练习，从而提高自己的实际动手能力。同时我也体会到，设计一个系统一定要规划好每个功能模块，考虑每个细节，要细心更要有耐心，不懂思考才能做得更好。

聚美购物商城在目前阶段，已经具备了购物系统核心的功能和能力，整个系统包括商家后台系统和用户前台系统，以及实现了整个购物流程的闭环。系统基于 Spring Cloud 微服务架构设计，设计之初考虑到了高可用、高扩展、高并发等场景。但仍然还有许多功能需要实现，比如搜索引擎和推荐系统等，也希望能在今后的学习和工作中掌握更多的知识来将它完善。

参考文献

[1] 凯•S.霍斯特曼.Java 核心技术卷 I[M].北京：机械工业出版社，2019.

[2] 明日科技.Java Web 从入门到精通[M].北京：清华大学出版社，2019.

[3] 杨开振.Java EE 互联网轻量级框架整合开发[M].北京：电子工业出版社，2017.

[4] 克雷格•沃斯.Spring Boot 实战[M].北京：人民邮电出版社，2016.

[5] 翟永超.Spring Cloud 微服务实战[M].北京：电子工业出版社，2017.

[6] 张工厂.MySQL 5.7 从入门到精通[M].北京：清华大学出版社，2019.

[7] 梁灏.Vue.js 实战[M].北京：清华大学出版社，2017.

[8] 耿庆阳.基于 Spring Boot 与 Vue 的电子商城设计与实现[D].西安：西安石油大学，2020.

[9] 张新磐.基于 O2O 模式的商城购物平台[D].武汉：华中师范大学，2019.

[10] 刘易.网上商城及推荐系统的设计与实现[D].哈尔滨：哈尔滨工业大学，2017.

[11] 马静.基于微信小程序的购物商城系统的设计与实现[J].微型电脑应用,2021,37(03):31-34.

[12] 张慧敏,王飒,张腾达.基于 SSM 框架的网上购物商城设计与实现[J].软件,2020,41(06):143-146.

[13] 童二宝,彭战军.基于分布式集群技术的 SSM 购物商城系统设计[J].软件,2019,40(08):123-126.

致谢

时光荏苒岁月如梭，转眼间四年的大学生活已经结束，我们即将告别安逸的校园，为生活游走于大江南北。回首望去，仿佛看见大一初入校园的样子，眼神充满了对大学生活的期待和向往，在这里有着被历史沉淀的书籍，被时间验证过的真理，无不充满着诱惑力。现如今，我们已经抵达了大学的终点站，即使内心有着万般的不舍，但这段人生经历终将结束，人生的列车还在呼啸前行，我们来不及多停留。

在这里我真诚的感谢大学四年帮助过我的所有老师和同学，感谢×××老师，教会了我学习方法，使我受益终身；感谢×××老师，临近毕业这段时间的指导和照顾；感谢一些志同道合的朋友，让我的大学生活不再孤单；感谢青葱岁月，感谢你们的陪伴，祝大家前程似锦。