



廣西農業職業技術大學
GUANGXI VOCATIONAL UNIVERSITY OF AGRICULTURE

职业本科毕业论文(设计)

题目： 基于 MVC 的医疗器械专

卖店管理系统的设计与

实现

学 院： 信息工程学院

专 业： 计算机应用工程

班 级：

学 号：

学生姓名：

指导教师姓名：

指导教师职称：

二〇二四年十二月

诚信声明书

本人郑重声明：所呈交的毕业论文（设计）是本人在导师的指导下独立进行研究、写作的成果，除文中已经注明引用的内容和致谢的地方外，本论文（设计）不包含任何其他个人或集体已经发表或撰写的研究成果。对本论文（设计）的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。

如本人的毕业论文（设计）涉及抄袭或剽窃等行为，本人愿承担由此造成的一切责任及后果。

论文作者签名： 日期：

摘要

在医疗技术持续进步、市场需求日益多元以及电商竞争愈发激烈的形势下，传统医疗器械专卖店管理模式陷入困境，存在管理效率低、资源配置不合理、运营成本高、客户服务差和数据安全性弱等诸多难题。为化解这些困境，基于 MVC 的医疗器械专卖店管理系统顺势而生。

技术层面上，后端以 Java 语言为基石，运用 Spring Boot 框架搭建起稳固架构，并通过 MyBatis 实现与 MySQL 数据库的高效精准通信，保障数据处理的高效性与准确性；前端采用 Vue.js 框架结合 Element UI 组件库，简化开发流程，同时借助 Axios 达成与后端 API 的无缝对接，为系统稳定运行筑牢根基。

从用户体验出发，系统贴合购买与管理流程。普通用户可前台浏览、加购、下单、收藏，管理个人信息、订单及申请售后维修等；管理员能实时掌控经营动态，精细管理器械等信息。前台有首页、查询购买、购物车等功能，后台含器械、库存等多模块，适配权限，促进线上线下融合。

该系统的应用，有力提升了专卖店管理效率，优化资源配置，削减运营成本，强化数据安全保障，助力专卖店于激烈竞争中脱颖而出，推动其向信息化迈进，具备推广价值与实践意义。

关键词：医疗器械专卖店;线上线下销售模式;Spring Boot;MVC;API

Design and Implementation of Medical Device Store Management System Based on Mvc

ABSTRACT

In the context of the continuous progress of medical technology, the increasingly diverse market demands and the increasingly fierce competition in e-commerce, the traditional management model of medical device specialty stores has fallen into difficulties, with problems such as low management efficiency, irrational resource allocation, high operating costs, poor customer service and weak data security. To solve these problems, a medical device specialty store management system based on the MVC architecture has emerged accordingly.

Technically, on the back end, taking the Java language as the foundation, the Spring Boot framework is utilized to build a solid architecture, and through MyBatis, efficient and accurate communication with the MySQL database is achieved, ensuring the high efficiency and accuracy of data processing. On the front end, the Vue.js framework combined with the Element UI component library is adopted to simplify the development process. Meanwhile, with the help of Axios, seamless connection with the back-end API is realized, laying a solid foundation for the stable operation of the system.

From the perspective of user experience, the system conforms to the purchase and management processes. Ordinary users can browse products, add items to the shopping cart, place orders, bookmark favorites, manage personal information, handle orders and apply for after-sales maintenance on the front end. Administrators can keep track of the business operations in real time and conduct meticulous management of medical equipment and other information. The front end of the system has functions such as a home page, product query and purchase, and shopping cart operations. The back end includes multiple modules like medical equipment management, inventory management and so on, which are adapted to different permissions and promote the integration of online and offline sales.

The application of this system has effectively improved the management efficiency of specialty stores, optimized resource allocation, reduced operating costs, strengthened data security, helped specialty stores stand out in the fierce competition, promoted their progress towards informatization, and thus has promotional value and practical significance.

Key words: Medical Device Specialty Stores;Online And Offline Sales Model;Spring Boot;MVC;API

目录

第一章 绪论	1
1.1 研究背景与意义	1
1.2 研究现状	1
1.3 本文主要工作	2
1.4 本文组织结构	2
第二章 关键技术	4
2.1 开发工具和环境	4
2.2 Spring Boot 框架	4
2.3 MyBatis 框架	4
2.4 Vue 框架	4
2.5 其他技术简介	5
2.6 MVC 架构	5
2.7 本章小结	6
第三章 系统需求分析	7
3.1 系统需求概述	7
3.2 功能需求分析	7
3.3 系统关键业务	11
3.4 本章小结	14
第四章 系统设计	15
4.1 概要设计	15
4.2 详细设计	16
4.3 数据库设计	23
4.4 本章小结	35
第五章 系统实现	36
5.1 前台模块	36
5.2 后台模块	47
5.3 本章小结	61
第六章 系统测试	62
6.1 测试环境	62
6.2 功能测试	62
6.3 本章小结	66

第七章 总结与展望	67
7.1 主要工作成果	67
7.2 工作展望	67
参考文献	68
致谢	70

第一章 绪论

1.1 研究背景与意义

随着医疗技术的飞速发展，医疗器械的种类与数量激增，传统专卖店管理方式已难以满足现代市场的多元化需求^[1]。当前，医药卫生体制改革深化、民众健康意识增强、人口老龄化加速以及电子商务的兴起，共同推动了医疗器械市场需求的不断攀升^[2]。消费者更倾向于线上购买医疗器械，这对管理系统提出了更高要求，我国医疗器械专卖店在经营和管理方面仍面临诸多挑战^[3]。

在此背景下，医疗器械专卖店管理系统的设计与实现至关重要。它融合线上销售，与线下无缝对接，为消费者提供便利。同时，线下门店提升购物体验。该系统简化销售流程，提高效率，降低成本，增强竞争力，优化资源配置，保障数据安全，推动行业健康发展。

1.2 研究现状

1.2.1 国外现状

在欧美等发达国家，电子商务的发展更为成熟和多元化，以 Amazon 和 eBay 为代表的电商平台已构建了完善的销售体系和技术架构^[4]。eBay 技术架构以 Java、Oracle、WebSphere 为核心，采用水平扩展与分片技术，有效应对高并发和海量数据处理，确保平台稳定运行。Amazon 则融合云计算、大数据、人工智能等技术，打造高效稳定的电商平台，注重高可用性和高性能，不断优化技术架构以提升可扩展性和灵活性。

这些平台深入了解消费者需求，利用大数据和人工智能技术优化销售策略，提升运营效率^[5]。同时，国外研究者不断探索新技术应用和创新模式，推动电商行业发展，技术选型上倾向于使用稳定且广泛应用的编程语言，如 Java，并注重系统的可扩展性和安全性设计^[6]。

这些成熟的经验和技术为国内基于 Java 的医疗器械专卖店管理系统的开发与实现提供了重要借鉴，有助于提升系统的稳定性和安全性，优化用户体验，推动医疗器械行业的持续发展。

1.2.2 国内现状

在国内电商行业的蓬勃发展下，医疗器械专卖店系统的研发与应用正加速推进。淘宝与京东作为电商巨头，凭借强大的技术架构和用户基础，成为该领域的领航者^[7]。淘宝通过微服务设计、Vue.js 前端技术、Java 后端技术，以及大数据与 AI 技术的广泛

应用，为用户提供了高效、智能的购物体验。京东则以云平台为核心，构建了集大数据处理、开放 API 服务于一体的技术平台，通过数智供应链架构推动数字经济与实体经济的深度融合，实现了高效稳定的运营。

然而，医疗器械专卖店在经营中仍面临诸多问题，如产品种类单一、反应速度慢、销售人员专业知识不足等，这促使越来越多的医疗器械企业开始重视线上渠道的建设^[8]。随着互联网技术的不断进步，“互联网+”模式正逐渐成为各行业转型升级的必然选择。医疗器械行业也不例外，信息化的发展推动了产品从传统的线下销售模式向线上线下混合销售模式的转变^[9]。

这一转变不仅有助于提升企业的品牌形象和竞争实力，更是医疗器械行业适应新时代发展的必由之路。通过借鉴电商巨头的技术架构和成功经验，医疗器械专卖店系统有望实现更加高效、智能、安全的发展，为消费者提供更优质的购物体验，推动整个行业的持续健康发展。

1.3 本文主要工作

本研究的主要目的是通过医疗器械专卖店管理系统，能够迅速获取各类销售数据，包括但不限于销售数量、销售价格、订单详情以及存货数量等关键信息。这一系统的核心优势在于，它利用信息化技术手段，使得专卖店管理者能够高效地实现医疗器械销售的全流程管理。

医疗器械专卖店管理系统不仅极大地提升了管理者的工作效率，还为消费者与专卖店之间搭建了一个便捷、高效的交互平台。借助这一系统，消费者可以随时随地浏览和选购所需的医疗器械，享受全天候的优质服务。

1.4 本文组织结构

医疗器械专卖店管理系统的组织架构可以被重新描述为一个全面而系统的框架，旨在详细记录和指导该系统从初步构想到最终实施的全过程。以下是该组织架构的重新描述：

第一章 绪论：开篇点明系统开发的起因，涵盖行业背景、传统管理问题，阐述开发目的与价值，明确项目范围与目标，为后续开发指引方向。

第二章 关键技术：详细介绍 Vue 2.0 与 Element UI 前端、Spring Boot 后端及 MySQL 数据库技术，阐释三者在系统中如何协同进行数据处理，构建技术底层。

第三章 系统需求分析：划分用户角色，梳理如商品管理、订单处理等功能需求，设定响应时间、并发量等性能指标，确保系统贴合业务与用户期望。

第四章 系统设计：展示系统架构布局、数据库 E-R 图与表结构及接口设计，保障系统扩展性、稳定性，为数据管理存储奠定基础。

第五章 系统实现：阐述编码规范，介绍前端界面与后端功能实现步骤，呈现系统

从设计到代码落地的过程与效果。

第六章 系统测试：进行全面测试，实施功能测试，设计用例、模拟场景，反馈问题并提出优化措施，确保上线质量。

第七章 总结与展望：总结与展望从成果回顾、问题反思、改进举措和未来规划四方面，对项目全面总结并指引后续发展，确保系统服务于专卖店运营。

此组织结构说明完整呈现系统开发流程与关键内容，为项目各阶段提供指引与参考。

第二章 关键技术

2.1 开发工具和环境

本课题基于 MVC 架构，致力于打造一个医疗器械专卖店管理系统。后端采用 Java 编程，结合 Spring Boot 框架与 MyBatis 持久层技术，实现业务逻辑与 MySQL 数据库的交互。前端则选用 Vue 框架，搭配 Element UI 组件库和 Axios 请求库，构建出用户友好的界面。开发过程中，运用 IntelliJ IDEA 工具。系统实现涵盖需求分析、设计、开发、测试优化及部署上线等关键环节，旨在打造出一个功能完备、性能稳定、易维护且可扩展的医疗器械专卖店管理系统，以满足业务需求并提升用户体验。

2.2 Spring Boot 框架

Spring Boot 由 Pivotal 团队打造，基于 Spring 框架，旨在简化开发流程。它遵循“约定大于配置”，凭借自动配置摆脱繁琐 XML 文件。内置 Tomcat 等 Web 容器，支持一键快速启动应用，搭配 starter POM 简化 Maven 配置，大幅缩短开发周期。

Spring Boot 特点鲜明：开发上，依“约定优于配置”简化流程，让开发者聚焦业务；启动时，内置容器实现快速搭建；生态方面，庞大社区催生丰富插件与第三方库，满足多样需求。

2.3 MyBatis 框架

MyBatis 作为一款出色的开源轻量级数据持久化框架，具备诸多优势。它不仅支持定制化 SQL、存储过程与高级映射，还对 JDBC 进行了内部封装，极大地简化了数据库操作流程，使得开发者能够将精力集中于 SQL 语句编写上。

在配置与映射方面，MyBatis 提供了 XML 文件和注解两种方式，以此实现原生信息的配置和接口与 Java 的 POJOs 到数据库记录的映射，这种灵活的映射机制确保了其易用性，且不会干扰应用程序或数据库既有的设计架构，因而成为 Java 开发领域常用的持久层框架。

2.4 Vue 框架

Vue.js 是一个构建用户界面的渐进式 JavaScript 框架^[11]。它易于上手，同时能够与其他库或已有项目轻松整合。Vue.js 的核心库专注于视图层，通过简单的 API 即可实现响应式的数据绑定和组合的视图组件。它采用虚拟 DOM 技术，提高页面渲染性能。Vue.js 还提供了丰富的插件和工具，如 Vue Router 和 Vuex，帮助开发者轻松构建复杂的应用。凭借其简洁的语法、高效的性能和良好的生态系统，Vue.js 已成为前端开发领域的热门选择。

Vue 框架的特点包括：

- 1.响应式编程：自动跟踪数据变化，即时更新相关 DOM，简化开发，增强交互性。
- 2.组件化：拆分界面为可复用组件，独立管理数据、模板与逻辑，提升开发效率与代码质量，便于团队协作、测试及复用。
- 3.数据驱动：实现双向数据绑定，视图与数据自动同步。
- 4.简单易学：API 与设计简洁直观，易于初学者掌握，入门难度低，能快速上手进行开发。

2.5 其他技术简介

2.5.1 MySQL

数据库选择 MySQL 8.0，MySQL 是开源的关系型数据库管理系统，支持多种操作系统，具有高性能、高可靠性和可扩展性，是目前运行速度最快的 SQL 数据库管理系统之一。MySQL 8.0 新增 JSON 支持、窗口函数等特性，优化了性能、安全性和事务管理，是快速且强大的数据库管理系统^[12]。MySQL 具有易于使用、高性能、可扩展性强、可靠稳定、安全且有大型社区支持等特点，其提供简洁界面与工具，具备高速读写和高并发性能，支持灵活扩展，保证数据完整持久，设有安全功能，还依托广大社区获得丰富资源与支持，适用于各类规模应用。

2.5.2 Maven

Maven 是 Java 项目管理与构建自动化的得力工具，基于 POM 管理项目生命周期，凭借自动化依赖管理（依 POM 文件精准解析下载依赖及传递依赖，保版本与依赖无误）、标准化构建流程（提供统一构建周期阶段，简化且提构建可靠性与重复性）以及丰富插件生态（含各环节插件并支持自定义，应项目特定需求）等特性，简化开发部署，其插件系统强大，是 Java 开发者的必备工具。

2.6 MVC 架构

本项目采用 MVC（Model-View-Controller）架构模式，即一种将应用程序分为模型、视图和控制器三个核心组件的设计模式^[13]。

模型（Model）：负责处理数据和业务逻辑，为系统提供数据支持与逻辑基础。

视图（View）：负责显示数据，搭建用户界面，提升用户交互体验。

控制器（Controller）：负责接收用户输入，调用模型和视图，协调系统响应，完成任务流程。

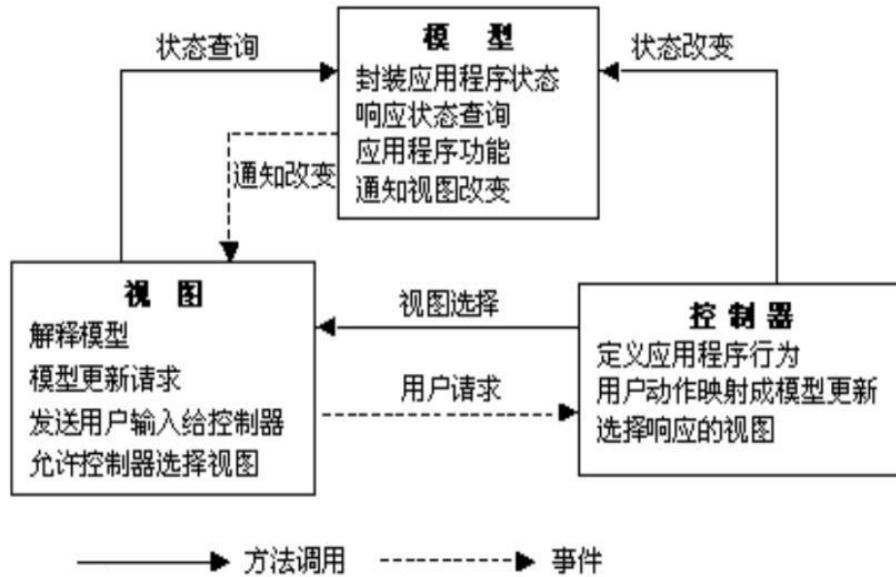


图 2-1 mvc 开发框架图

MVC 架构模式可分离应用关注点，提升代码的可维护、扩展与重用性，让开发者独立修改扩展各部分而互不干扰，在 Web、桌面应用、游戏开发等领域广泛应用。

2.7 本章小结

本章详述了医疗器械专卖店管理系统的关键技术与开发环境。开发工具为 IntelliJ IDEA，后端用 Java 配合 Spring Boot 与 MyBatis 分别实现高效开发及与 MySQL 交互，前端以 Vue 及相关库构建界面。Spring Boot 简化开发流程，MyBatis 灵活处理数据持久化，Vue 框架优势显著便于前端构建。MySQL 数据库性能强、扩展性佳，Maven 助力项目管理与构建自动化。MVC 架构分离应用关注点，提升系统性能，各技术协同合作，为系统稳定运行、满足业务需求提供保障，旨在打造优质管理系统。

第三章 系统需求分析

3.1 系统需求概述

在医疗技术迅猛进步以及民众健康意识大幅提升的背景下，医疗器械市场呈现出蓬勃的增长态势。紧跟线上线下融合的销售潮流，优化顾客的购物体验与商家的运营效率，现拟运用 Springboot + Mybatis 构建后端技术支撑体系，借助 Vue + Element UI 搭建前端框架，并以 MySQL 作为数据库，采用前后端分离的模式，开发一个综合性的医疗器械专卖店管理系统。该系统将涵盖商品展示、购买、销售、库存管理、订单处理、售后维修服务以及数据分析等一系列关键功能，旨在为医疗器械销售行业提供一套高效、智能且用户体验良好的信息化解决方案，助力企业在激烈的市场竞争中脱颖而出，实现可持续发展。

该系统通过实现线上线下业务的无缝对接，引入高效的数据管理机制，改善用户体验，明确员工权限与职责，以及利用协同过滤推荐算法提供个性化服务，旨在为管理员、商家、员工和用户打造一个高效、便捷、个性化的专卖店管理平台，从而提升整个行业的运营效率和服务质量，满足市场对专业化、规范化、智能化的需求，推动行业的健康发展，并为所有参与者提供必要的支持与保障。

3.2 功能需求分析

3.2.1 系统用例图

1. 总用例图

展示了医疗器械专卖店管理系统中不同用户角色及其对应的功能，主要分为四个角色，描述了系统的功能需求以及用户如何与系统交互。系统总用例图如图 3-1 所示：

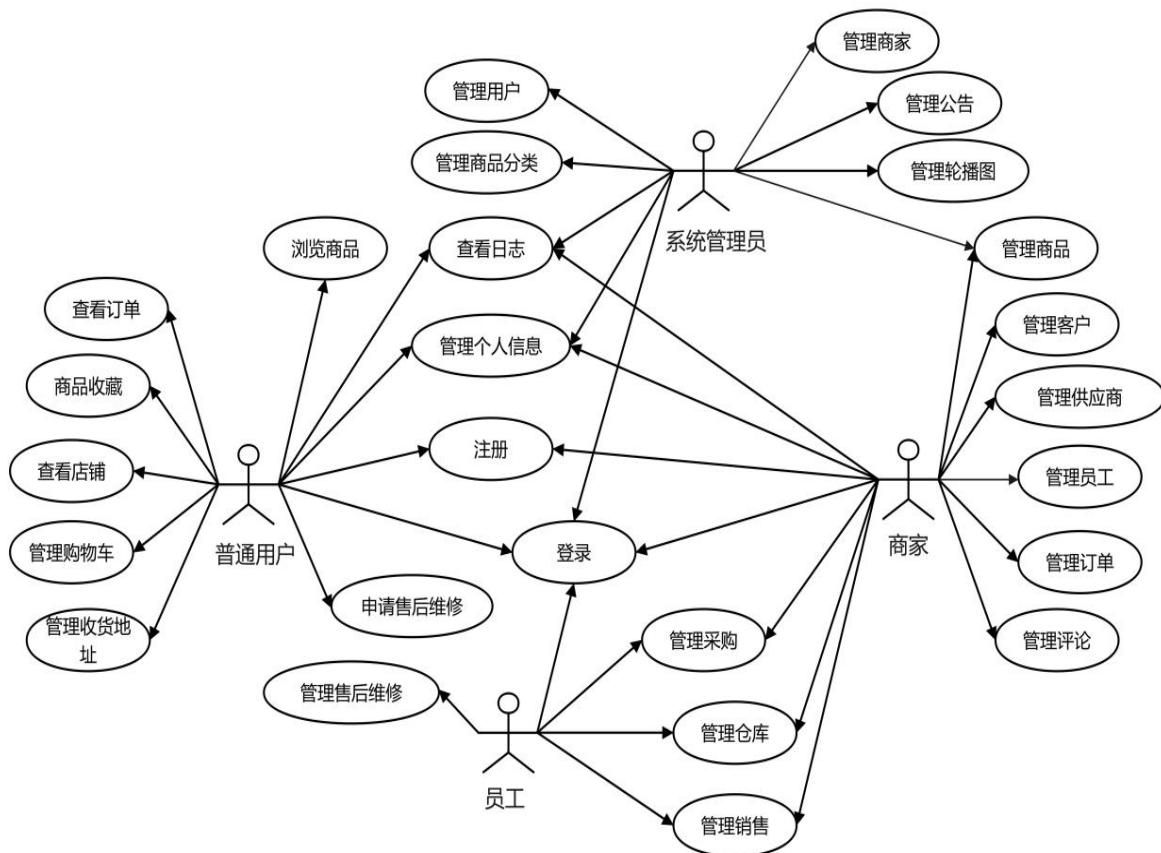


图 3-1 系统总用例图

用例角色描述:

系统管理员在系统中担任重要角色，主要负责维护系统的基本信息，拥有管理个人信息、管理商品分类、管理用户、管理商家、管理公告、管理轮播图、查看日志等操作。

普通用户拥有管理个人信息、查看订单、商品收藏、查看店铺、管理购物车、管理收货地址、浏览商品、申请售后维修等操作。

商家拥有管理个人信息、管理商品、管理客户、管理供应商、管理订单、管理评论、采购审核、销售审核、管理仓库等操作。

员工拥有管理售后维修、管理个人信息、管理采购、管理仓库、管理销售等操作。

3.2.1 系统管理员业务功能

- 1.个人信息：可查看和编辑个人信息。
 - 2.商品分类管理：对商品分类进行信息录入、删除，对已录入信息的商品分类可以进行查看、删除、修改商品分类信息。
 - 3.商品管理：查看和管理所有商品信息。
 - 4.管理员信息：查看和管理系统管理员账号信息。

5. 用户信息：查看和管理用户信息。
6. 商家信息：查看和管理商家信息。
7. 公告管理：发布行业新闻、产品知识等公告，提升用户黏性。
8. 轮播图管理：上传、编辑、删除首页轮播图，优化视觉效果，提升用户体验。
9. 日志信息：可查看所有登录日志并删除。

3.2.2 普通用户业务功能

1. 个人信息：可查看和编辑个人资料，包括姓名、联系方式等，确保信息准确，同时可修改密码以保护账号安全。
2. 商品详情：可查看商品的各项详细信息，如图像展示、价格明细、规格参数、库存数量以及用户评价等关键内容。可直接在该页面将商品添加至购物车，也可以收藏商品。
3. 购物车：用户将商品加入购物车后，可查看自己加入购物车的商品，可对商品进行数量修改、选中/取消选中、删除等操作，购物车实时计算商品总价，方便用户快速结算下单，同时支持购物车商品长期保存，便于用户随时回来继续购物。
4. 订单管理：用户可查看自己所有订单的详细信息，包括订单编号、金额、状态（待发货、待收货、已完成等）以及订单跟踪等信息。对于待收货，用户可进行确认收货。已完成的订单，可进行评价操作，评价内容包括文字和星级评分。
5. 我的地址：管理收货地址信息，用户能够添加、编辑、删除多个收货地址。
6. 我的收藏：查看自己收藏的商品信息。
7. 售后维修：查看和管理售后维修情况，查看审核情况。

3.2.3 商家业务功能

1. 个人信息：可查看和编辑个人资料，包括姓名、联系方式等，确保信息准确，同时可修改密码以保护账号安全。
2. 商品管理：可以对商品进行信息录入、删除，对已录入信息的商品可以进行商品基本信息查看、删除、修改商品信息。
3. 订单管理：可查看和管理自己店铺的订单，对订单进行发货并填写物流信息、删除订单操作，并且可以查看用户购买商品以及该订单单据历史。管理员可查看和管理全部订单。
4. 评论管理：可查看和删除自己店铺的商品评论，对违规评论进行删除。
5. 员工管理：可对自己店铺的员工信息进行管理，包括添加、编辑、删除员工账号，为员工分配相应的权限（如仓管、销售员、采购员）。
6. 供应商管理：可对自己店铺的供应商信息进行管理，包括添加、编辑、删除。
7. 客户管理：可对自己店铺的客户信息进行管理，包括添加、编辑、删除。

8. 库存管理：可管理自己店铺的仓库信息，包括仓库名称、地址、联系人等基本信息，可新增仓库，以满足业务拓展的需求。

9. 采购管理：审核采购员提交的采购预定信息和采购单信息，可查看单据审核历史，确保采购计划符合店铺的经营需求和库存情况。

10. 销售管理：审核销售员提交的销售单信息，可查看单据审核历史。

11. 日志信息：可查看自己的登录日志并删除。

3.2.4 员工业务功能

员工角色又细分为三种权限：仓管、销售员、采购员。每个权限可能拥有不同的操作权限。

1. 仓管：

（1）个人信息：可查看和编辑个人资料。

（2）库存管理：查看和管理店铺仓库信息，可查看和审核销售员提交的单据申请（销售出库、销售退货）以及采购员提交的单据申请（采购入库、采购退货）。

（3）日志信息：可查看自己的登录日志并删除。

2. 销售员：

（1）个人信息：可查看和编辑个人资料。

（2）销售单：可查看和管理自己店铺销售单信息，包括添加、编辑、删除。并向商家发起审核，可查看单据审核进度及回复。

（3）销售出库：可查看和管理自己店铺销售出库信息，包括添加、编辑、删除。

（4）销售退货：可查看和管理自己店铺的销售退货信息，包括添加、编辑、删除。

（5）线上售后维修：可查看和审核所属订单的用户发起的售后维修申请，如果符合售后维修条件，审核通过后可录入维修进度。

（6）线下售后维修：可查看和录入店铺线下售后维修记录存档，并录入维修进度等信息。

（7）日志信息：可查看自己的登录日志并删除。

3. 采购员：

（1）个人信息：可查看和编辑个人资料。

（2）采购预定：可查看和管理自己店铺的采购预定信息，包括添加、编辑、删除。

（3）采购单：采购员可查看和管理自己店铺的采购单信息，包括添加、编辑、删除。可调用采购预定信息，完成采购预定的采购。

（4）采购入库：可查看和管理自己店铺的采购入库信息，包括添加、编辑、删除。

（5）采购退货：可查看和管理自己店铺的采购退货信息，包括添加、编辑、删除。

（6）日志信息：可查看自己的登录日志并删除。

3.3 系统关键业务

3.3.1 用户购物业务

用户进入系统前台购物，流程如下：游客身份可先行浏览商品，若想将心仪商品加入购物车，则需先完成登录或注册账户操作。成功登录后，普通用户能自由浏览各类商品并挑选，选中后将其加入购物车。在购物车中，可以对商品进行编辑，如更改数量或删除商品。确认商品信息准确无误后，便可启动结算流程。接着，选定支付方式与收货地址，点击立即支付，一旦支付成功，系统随即生成订单。商家收到订单信息后进行商品的备货和发货，普通用户在收到商品后可以对商品进行评价。在整个流程中，还包括了个人信息管理、收货地址管理等功能模块。用户购物业务流程图如图 3-2 所示。

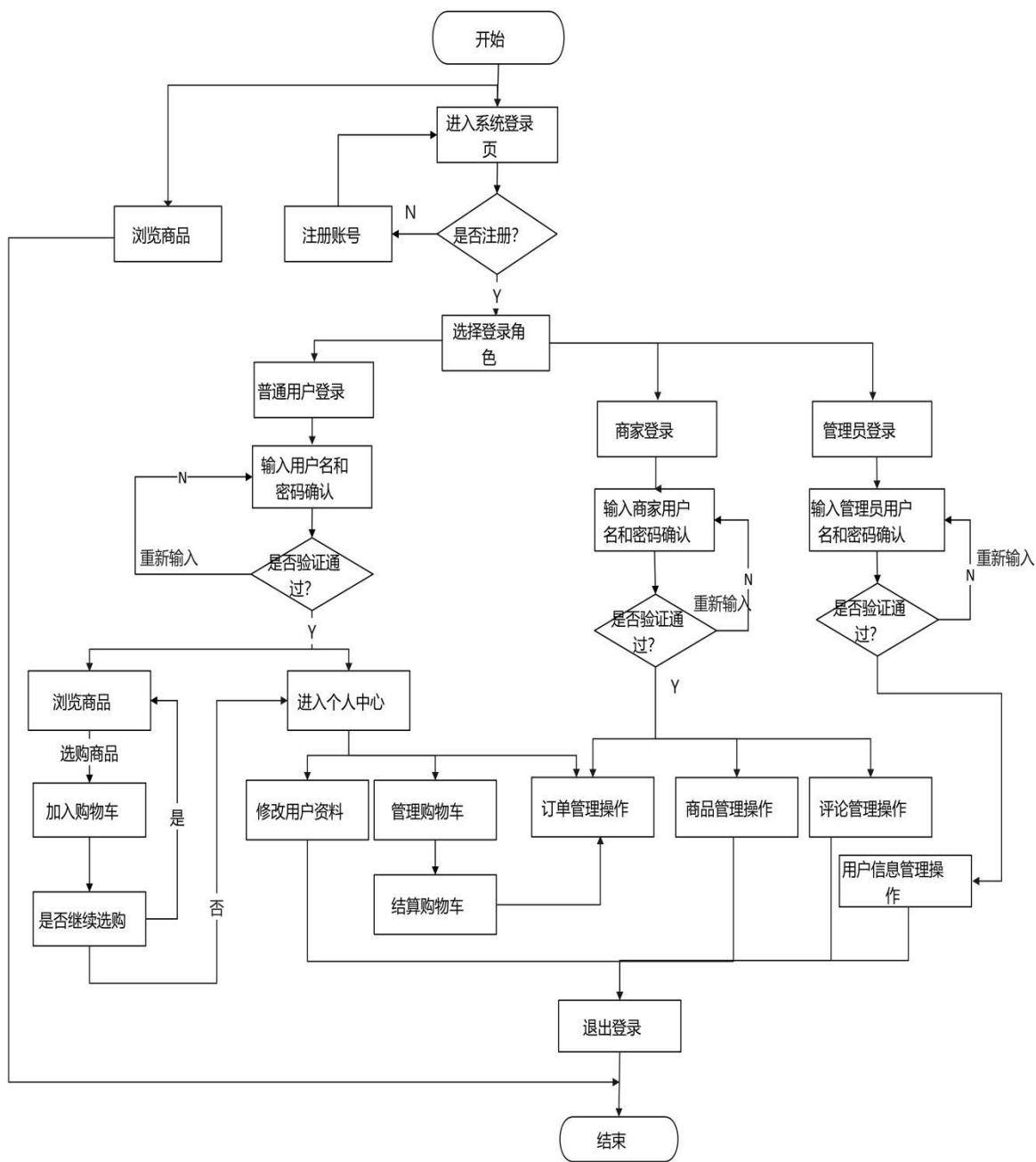


图 3-2 用户购物业务流程图

3.3.2 售后维修业务

售后维修业务流程如下：普通用户登录系统成功后，可对已完成且在保修期内的订单进行售后维修申请。在售后维修管理中，可对该售后维修申请进行编辑，如更改信息或删除。该订单商品的店铺员工接收到售后维修信息后，进行审核更新单据状态并回复。如果审核通过，店铺员工则录入维修进度。在整个流程中，还包括了线下售后维修管理等功能模块。售后维修业务流程图如图 3-3 所示。

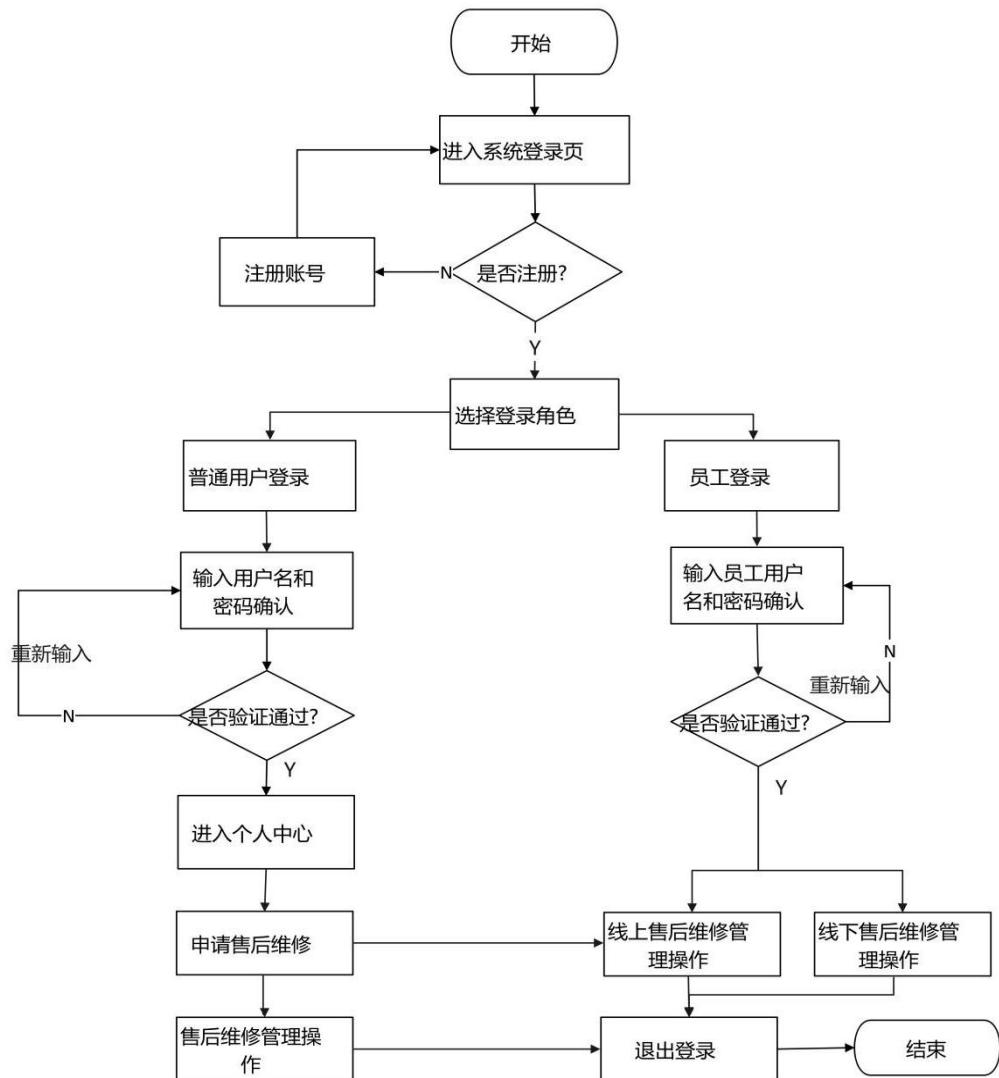


图 3-3 售后维修业务流程图

3.3.3 出入库业务

出入库业务流程包括销售与采购两方面。

在销售环节，手握销售权的员工登录系统，精准贴合店铺当下售卖动态填写销售单，提交给商家严格审核。一旦获批，即刻向仓管发起销售出库申请，仓管人员秉持严谨态度仔细核验，确保商品信息、数量等准确无误后放行出库，库存随之实时减少。倘若遭遇质量不过关、客户退货等状况，迅速申请销售退货单，交由仓管审核，通过后商品重新入库，库存相应调整。

在采购环节，采购权限员工登录系统后，可根据店铺规划申请采购预定，规划采购商品，经商家审核通过，再生成采购单完善采购价、数量、供应商等关键信息，二次提交审核；若遇紧急需求，则直接申请采购单。商家确认后，采购人员申请入库，

选定对应采购单与入库仓库提交仓管审核，审核通过，库存增加。若采购商品有问题，申请采购退货单，经仓管审核后出库，库存相应调整。在整个流程中，还包括了仓库管理等功能模块。出入库业务流程图如图 3-4 所示。

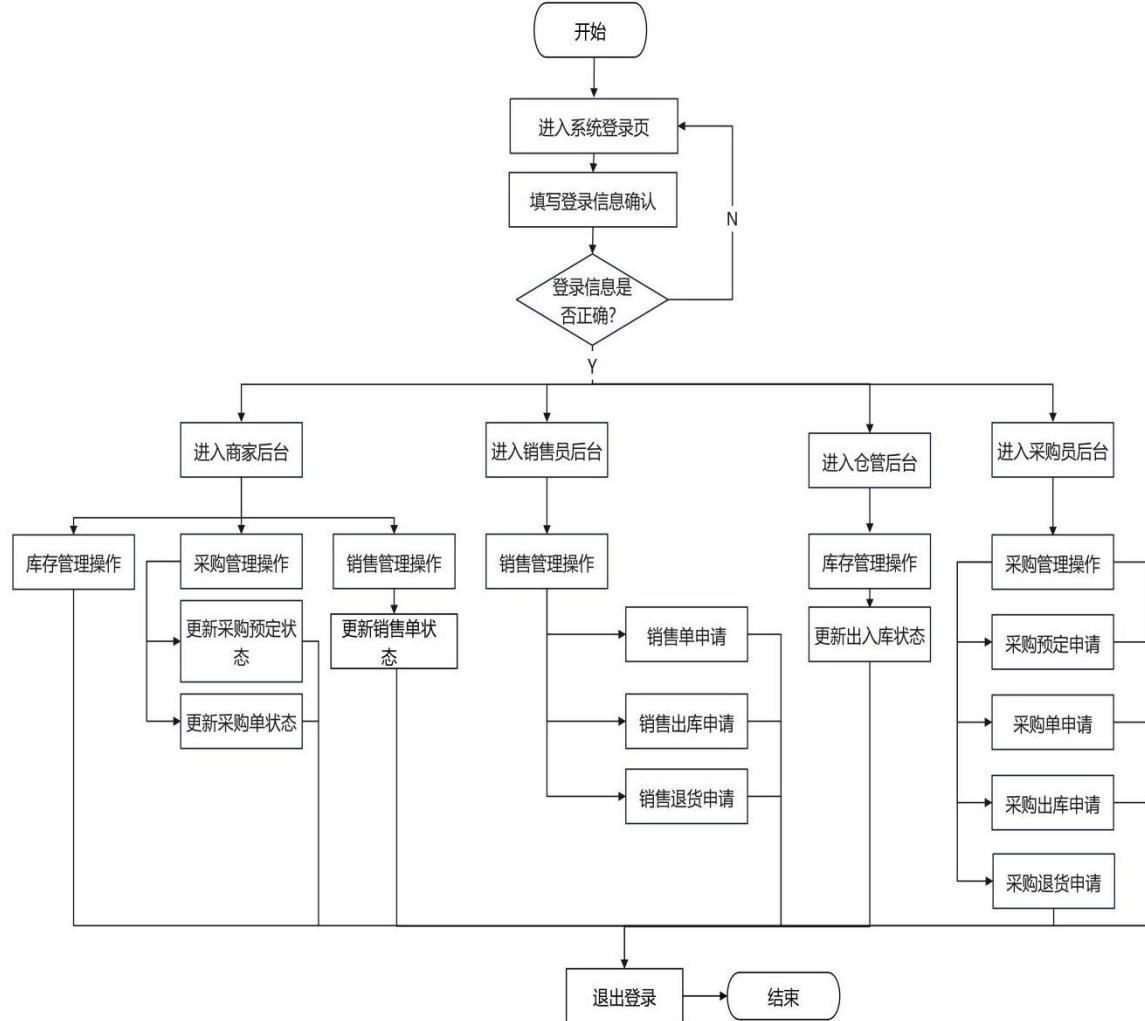


图 3-4 出入库业务流程图

3.4 本章小结

系统需求分析针对医疗器械专卖店管理系统，从用户、商家、管理员及员工角度划分功能，涵盖前台展示与后台管理模块，实现商品浏览、购买、库存管理、订单处理、售后维修及数据分析等功能，旨在提升顾客体验与商家运营效率，确保系统安全稳定运行。

第四章 系统设计

4.1 概要设计

医疗器械专卖店系统管理从总体上分为两大部分，分别是前台部分和后台部分，以下是系统功能结构图：

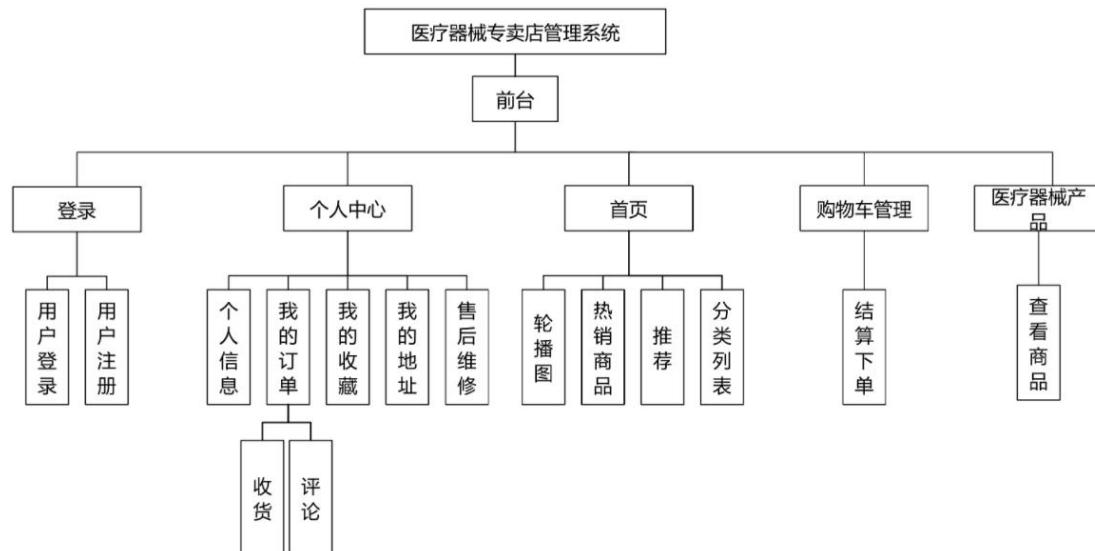


图 4-1 系统前台功能结构图

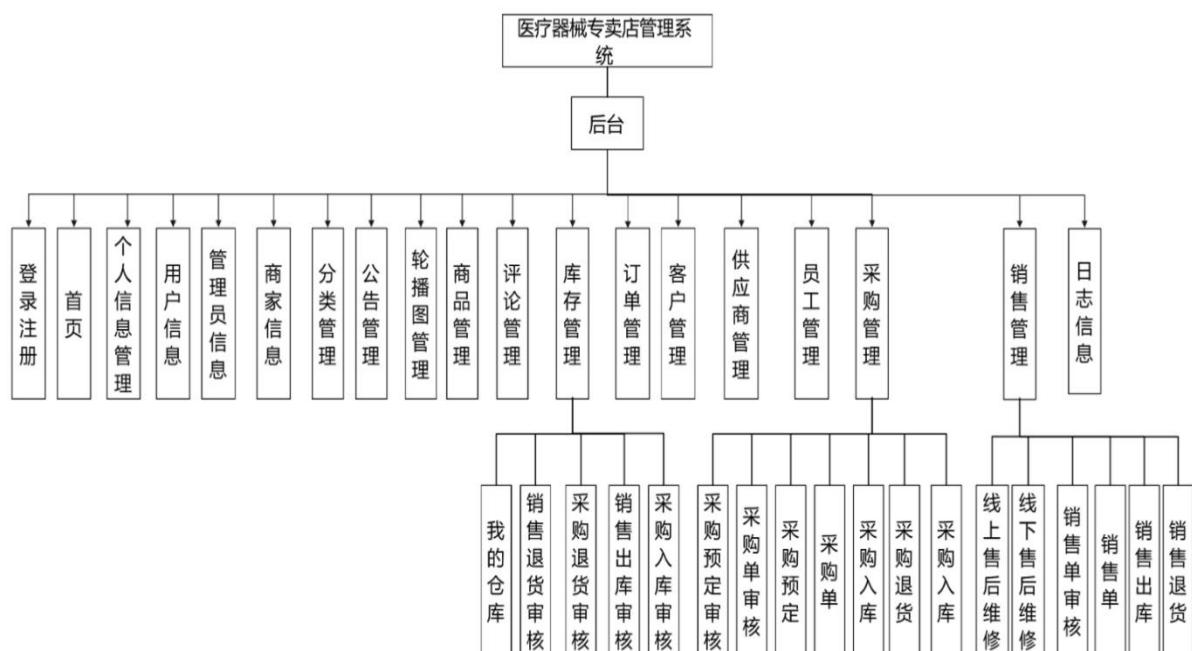


图 4-2 系统后台功能结构图

本系统主要包括前台展示、后台展示两个功能模块。

前台展示模块涵盖登录、首页、个人中心、购物车管理、医疗器械产品。其中登录实现用户登录和用户注册功能；个人中心实现个人信息、我的订单的收货和评论、我的收藏、我的地址、售后维修等功能；首页实现轮播图、热销商品、推荐、分类列表等功能；购物车管理实现结算下单功能；医疗器械产品实现查看商品等功能。

后台展示模块包含首页、登录注册、商品管理、订单管理、评论管理、员工管理、供应商管理、客户管理、库存管理、采购管理、销售管理等功能。其中库存管理涉及仓库信息维护和出入库审核；采购管理包括采购预定、采购单、采购入库、采购退货以及相应审核流程；销售管理涵盖销售单、销售出库、销售退货、销售单审核以及线上线下售后维修处理等功能；

4.2 详细设计

4.2.1 购物车模块设计

用户打开购物车界面，前端判断用户是否登录，如果未登录则拦截并提示登录信息。若用户已登录，则根据当前登录信息向后端发起 axios 请求后端接口，通过用户 id 获取对应购物车的信息，将返回的信息传至前端并展示。

用户可通过对应的商品进入商品详情页中，在商品详情页选择购买数量后点击“加入购物车”，商品数量不能大于库存，如果该商品已存在购物，则叠加商品数量。在购物车列表中，可对商品进行删除和选择商品数量，列表数据与后台数据一致，列表数据发生变化，前端将发送请求至后端修改相应数据。使用的接口有：加入购物车(add)、删除购物车(delete)、更新购物车商品数量(update)。

在购物车信息实体类（Cart）设计中中，包含商品实体（Goods.java），购物车信息类通过 goodsId 字段与商品实体类的 id 相联系，购物车-商品实体类图如图 4-3：

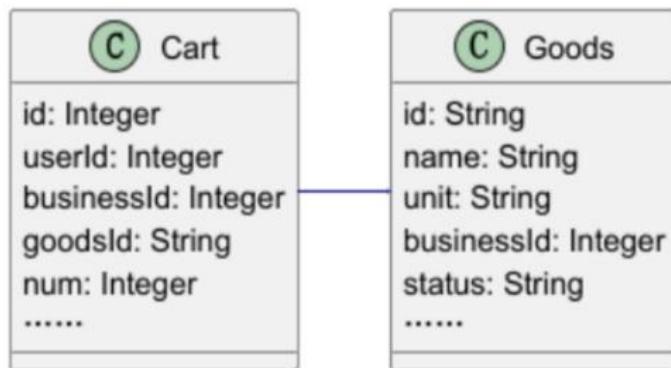


图 4-3 购物车-商品实体类图

加入购物车时序图如图 4-4：

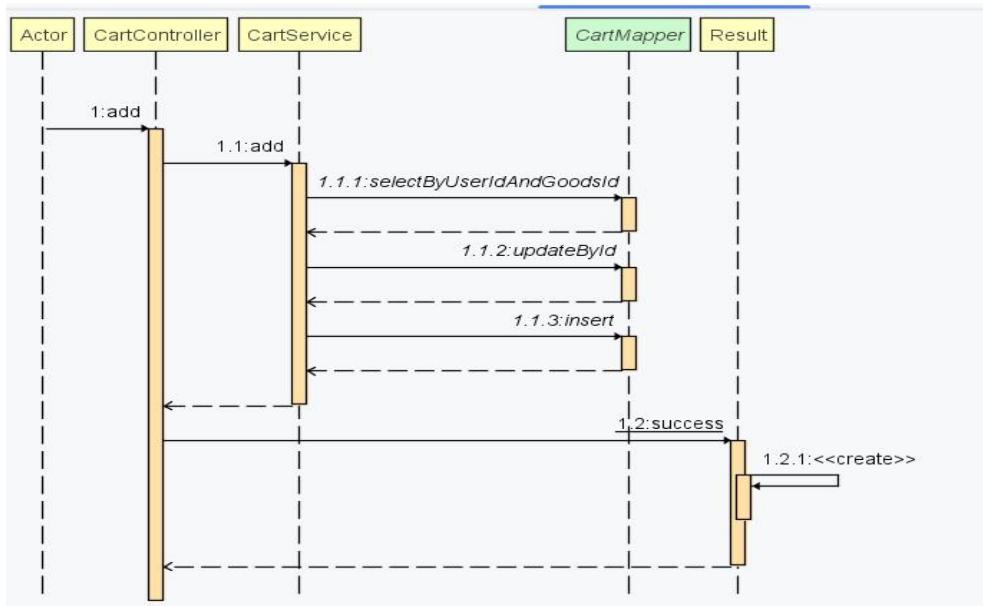


图 4-4 加入购物车时序图

4.2.2 订单管理模块设计

用户登录后选择商品加入购物车，在购物车中选择多个商品进行结算下单。用户点击“结算”后，跳转至确认订单页面。同时前端根据当前用户的 id 发送请求传至后端地址接口查询得到该用户的收货地址。用户确认收货地址和付款方式等信息后点击“立即支付”，前端发送订单信息传至后端接口生成订单，其中用到的接口有：下单（add）、删除订单（delete）、发货（update）等。

商家登陆后进入系统后台，可查看店铺订单详情和对订单进行操作。可点击订单列表的某行订单的“删除”按钮删除订单。也可对订单状态为待发货的订单进行发货。点击“发货”按钮，弹出发货弹窗，录入物流信息，点击“确认”发货成功。用户在前台确认收货后，可对订单进行评价。

在订单信息实体类（Order）设计中中，包含商品实体（Goods.java），订单信息实体类通过 goodsId 字段与商品实体类的 id 相联系，订单-商品实体类图如图 4-5：

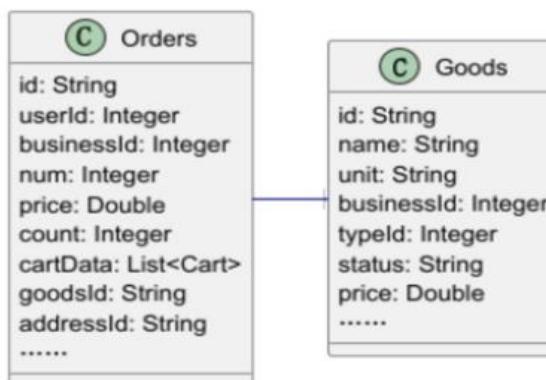


图 4-5 订单-商品实体类图

购物车下单的时序图如图 4-6：

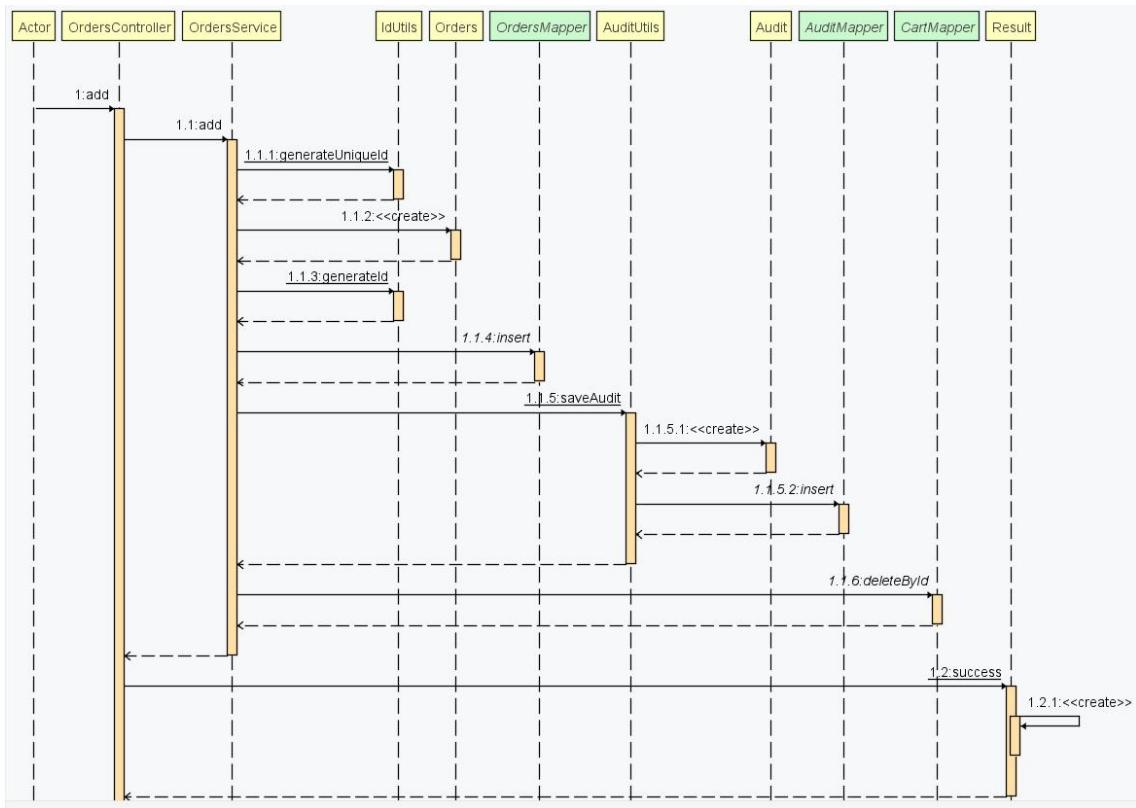


图 4-6 加入购物车时序图

发布评论的时序图如图 4-7：

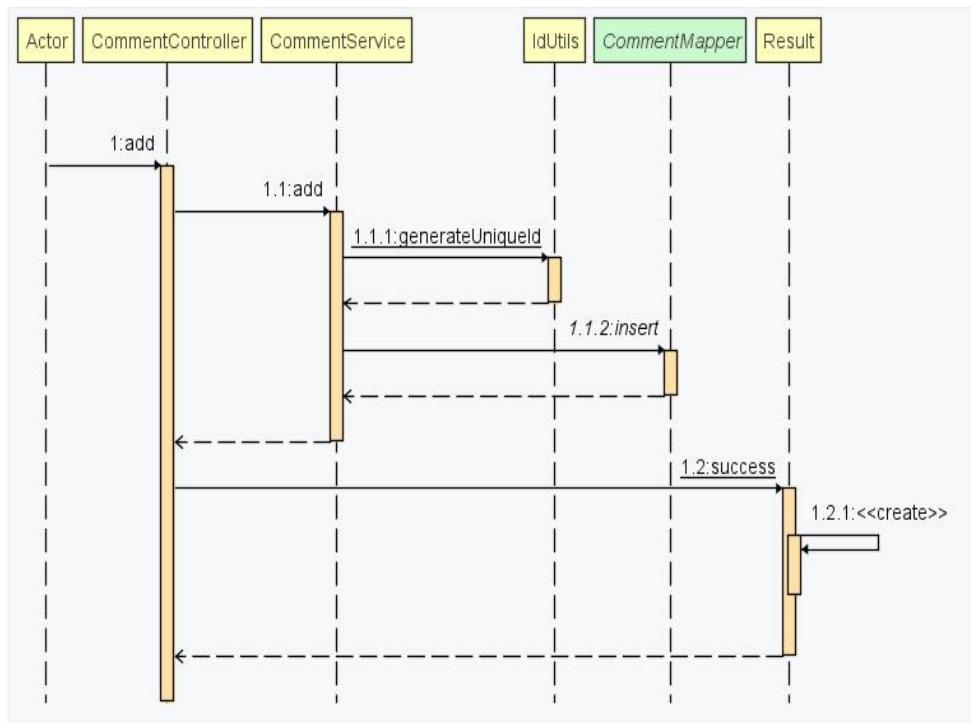


图 4-7 发布评论时序图

4.2.2 采购管理模块设计

拥有采购权限的员工成功登录系统后，可借助前端页面管控采购流程。无论是采购预定环节，还是采购单处理阶段，员工都有权进行信息的添加、编辑与删除操作。以添加采购预定为例，员工在页面填好商品明细、采购数量等相关信息后，点击“保存并发起审核”按钮，前端利用 AJAX 技术发送 POST 请求至“/procurement/add”接口，向商家正式发起审核申请，同时完成数据的更新。在整个采购流程管理中，涉及的关键接口包括用于提交申请的“/procurement/add”，以及负责审核环节的“/procurement/update”等，采购单的各项操作步骤及所依赖的接口与采购预定操作基本一致。

商家登录后台系统后，能够查看并审核采购预定或采购单，通过点击“审核”按钮，在审核页中选择通过或拒绝并提交，前端随即发送 PUT 请求至/audit/add 接口记录审核流程，其中用到的接口有：记录审核流程（/audit/add）。

在采购信息实体类（Procurement.java）设计中，包含采购商品实体列表（Batch.java），采购商品实体用于存放一个采购中多个的商品信息，采购-采购商品实体类图如图 4-8：

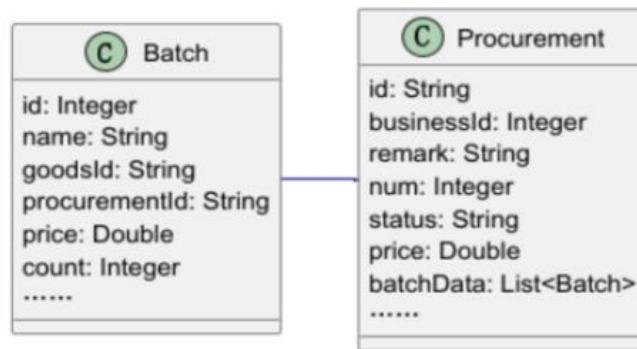


图 4-8 采购-采购商品实体类图

采购列表的时序图 4-9：

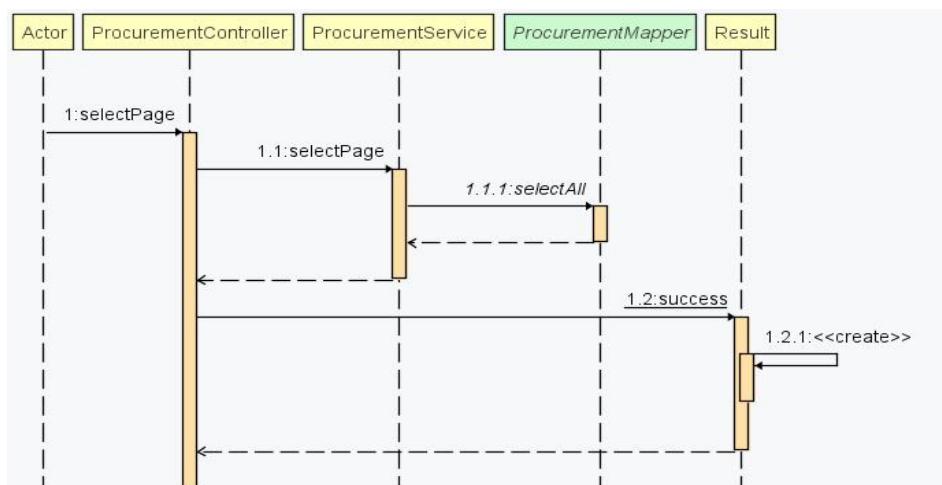


图 4-9 采购列表时序图

记录审核流程的时序图 4-10：

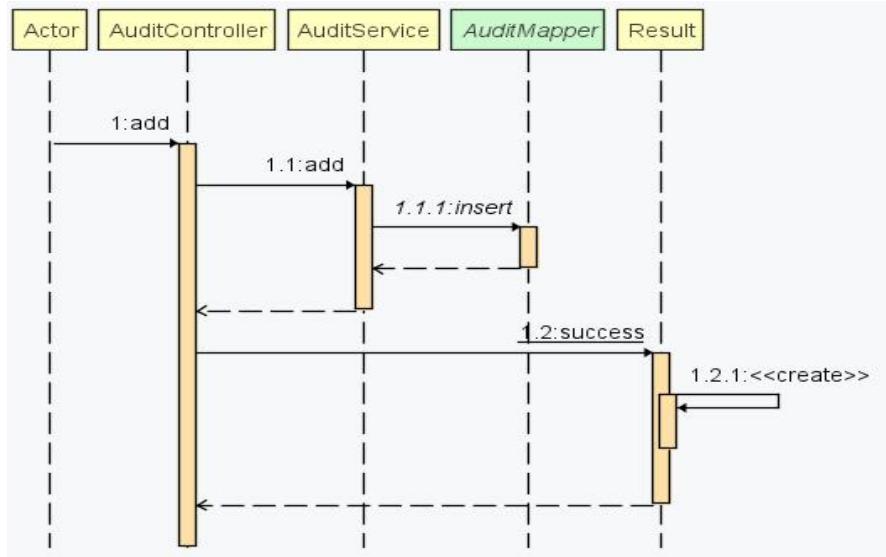


图 4-10 记录审核流程时序图

4.2.3 销售管理模块设计

有销售权限的员工可以通过点击销售单导航进入销售单页面。在这一过程中，前端会将登录信息中的店铺 ID 传至后端接口查。后端接收到店铺 ID 后，会查询并获取相应的销售单信息，并将这些信息返回给前端在页面渲染展示。员工还可以通过点击“新增”按钮来触发前端事件打开弹窗，允许员工录入新的销售单信息。当信息被确认并提交，前端会将数据发送到后端的相应接口，后端则会执行增加（/sale/add）、更新（/sale/update）、或删除（/sale/delete）销售单的操作。此外，后端还提供一个接口（/sale/selectAll）用于查看所有销售单信息。

在销售信息实体类（Storage.java）设计中，包含销售商品实体列表（Batch.java），销售商品实体用于存放一个销售中多个的商品信息，销售-销售商品实体类图如图 4-11：

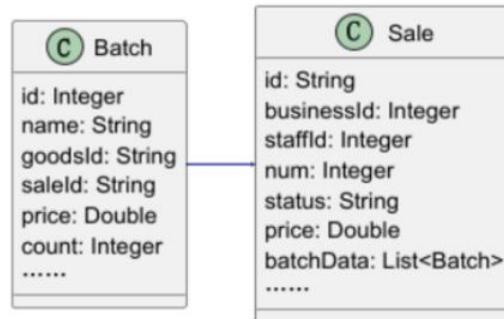


图 4-11 销售-销售商品实体类图

销售单展示的时序图 4-12：

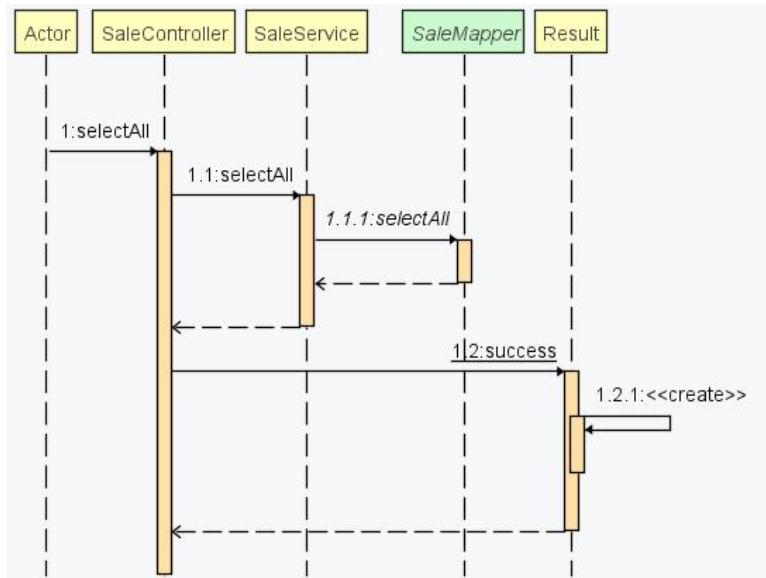


图 4-12 销售单展示时序图

4.2.4 出入库模块设计

拥有销售权限的员工登录系统后，可以通过前端页面管理销售和采购的出入库流程。在销售出库或退货处理等操作中，员工可以添加、编辑或删除信息。例如，添加销售出库时，员工输入信息并点击“保存并发起审核”，前端使用 AJAX 发送 POST 请求到“/storage/add”接口，提交审核申请并更新数据。关键接口包括提交申请的“/storage/add”和审核用的“/storage/update”。

仓管权限的员工登录后，可以查看并审核出入库单据。点击“审核”后，在弹窗中选择通过或拒绝，前端发送 PUT 请求到/audit/add 接口记录审核结果。主要接口是记录审核流程的/audit/add。其他关于出入库模块操作类似。

在出入库信息实体类（Storage.java）设计中，包含商品列表实体（Batch.java），商品列表实体用于存放一个出入库信息中多个的商品信息，其中商品列表实体的 storageId 用于存储出入库实体的 id，出入库-商品列表实体类图如图 4-13：

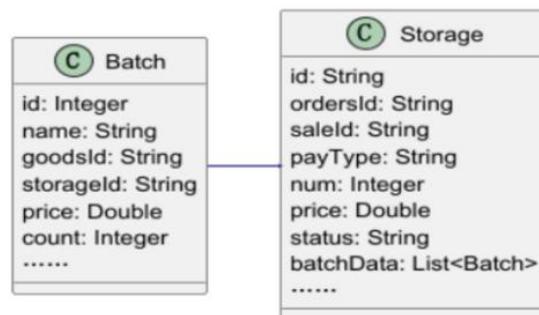


图 4-13 出入库-商品列表实体类图

出入库展示的时序图 4-14：

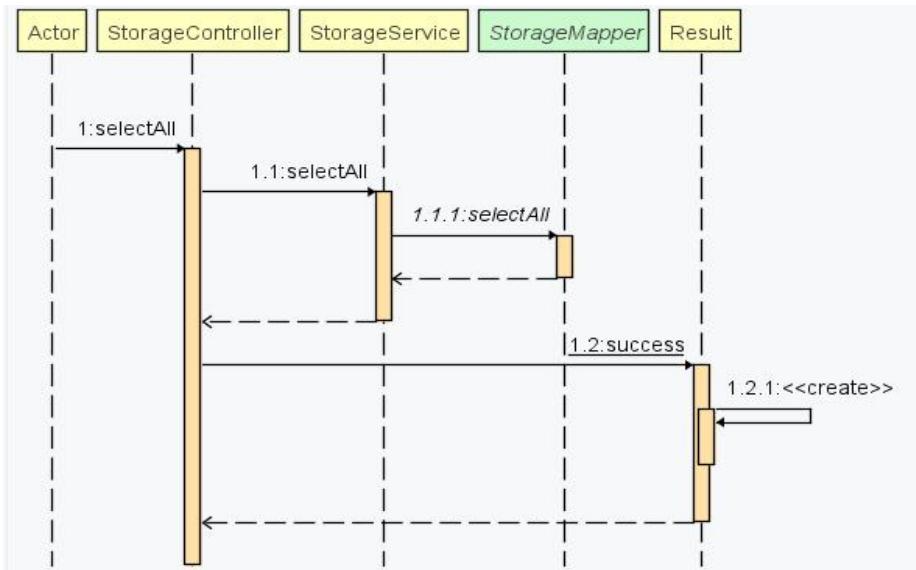


图 4-14 出入库展示时序图

4.2.5 审核模块设计

当商家及具有审核权限的员工点击“审核”按钮时，前端会弹出审核窗口，在此窗口录入审核信息并确认后，前端借助 axios 向后端的“/audit/add”接口发送请求，完成审核状态的更新并记录审核流程。对于已审核的信息，若需修改，可执行撤回审核操作，之后便能重新进行审核。此外，点击列表中的“单据历史”按钮，前端会将获取到的单据 id 发送至后端“/audit/selectAll”接口，后端依据单据 id 查询并返回审核历史记录信息，前端接收后将其渲染展示在页面上，以供查看单据的审核轨迹。

审核信息实体类（Audit.java）设计，其中 number 代表审核单据的 id，results 代表审核结果。审核实体类图如图 4-15：

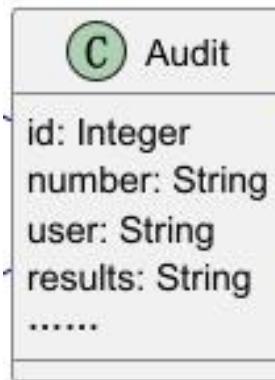


图 4-15 审核实体类图

单据历史展示的时序图 4-16：

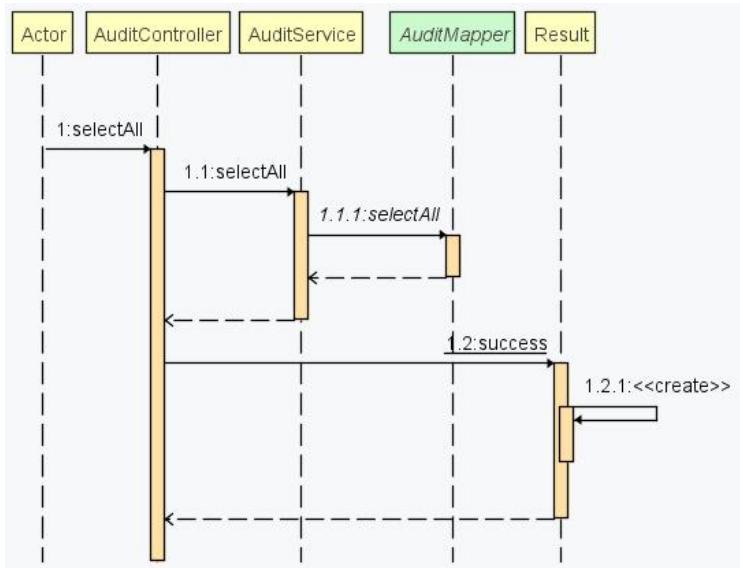


图 4-16 单据历史展示时序图

4.3 数据库设计

4.3.1 E-R 图

本节将呈现医疗器械专卖店管理系统数据库设计的 E-R 图，此图旨在清晰展现系统内各实体及其相互关系，直观地揭示数据库中各表之间的关联与约束情况，为系统的数据架构提供可视化的呈现方式，助力开发人员深入理解和构建稳定、高效的数据体系，以支撑整个医疗器械专卖店管理系统的稳定运行和数据交互。

系统 E-R 图如图 4-17 所示：

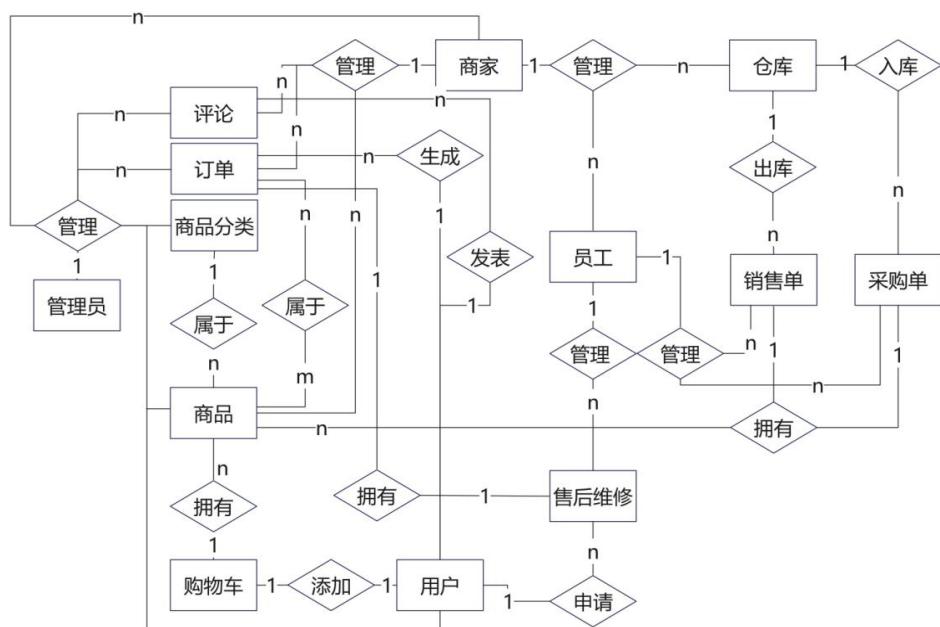


图 4-17 系统 E-R 图

4.3.2 数据库表设计

1. 用户表 user

存放用户登录所需要的信息，需要访问此表；设计如图 4-18：

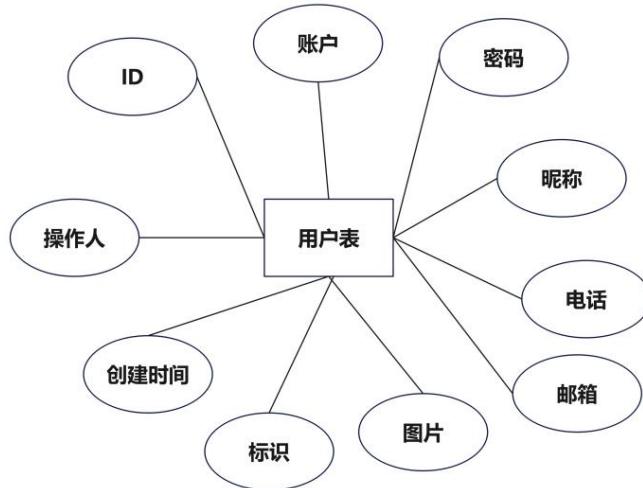


图 4-18 用户实体属性

用户表（ID，用户名，密码，姓名，头像，角色标识，电话，邮箱，创建时间，操作人 ID，账号，密码，昵称，电话，邮箱，图片，标识，创建时间，操作人）。如表 4-1：

表 4-1 用户表

字段	类型	长度	是否为主键	描述
id	int		是	ID
username	varchar	10	否	账号
password	varchar	10	否	密码
name	varchar	10	否	昵称
phone	varchar	20	否	电话
email	varchar	20	否	邮箱
avatar	varchar	100	否	图片
role	varchar	10	否	标识
time	datetime		否	创建时间
operator	varchar	10	否	操作人

2. 购物车信息表 cart

购物车表用于存储用户的购物车信息，其用户 id 外键是用户表的用户 id 主键，其商品 id 外键是商品表的商品 id 主键，其商家 id 外键是商家表的商家 id 主键；如图如图 4-19：

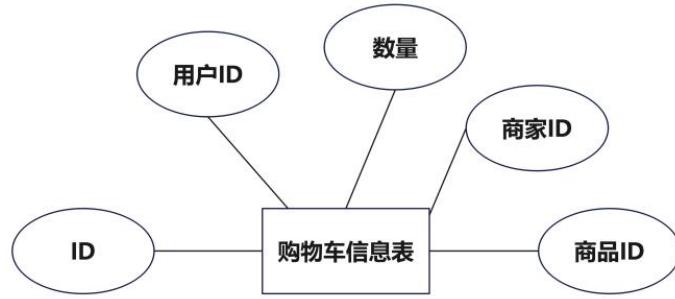


图 4-19 购物车信息实体属性

购物车信息表 (ID, 用户 ID, 商家 ID, 商品 ID, 数量)。如表 4-2:

表 4-2 购物车信息表

字段	类型	长度	是否为主键	描述
id	int		是	ID
user_id	int		否	用户 ID
business_id	int		否	商家 ID
goods_id	varchar	10	否	商品 ID
num	int		否	数量

3.商品信息表 goods

商品信息表中存储商品的基本信息，其分类 id 外键是商品分类表的商品分类 id 主键；如图 4-20：

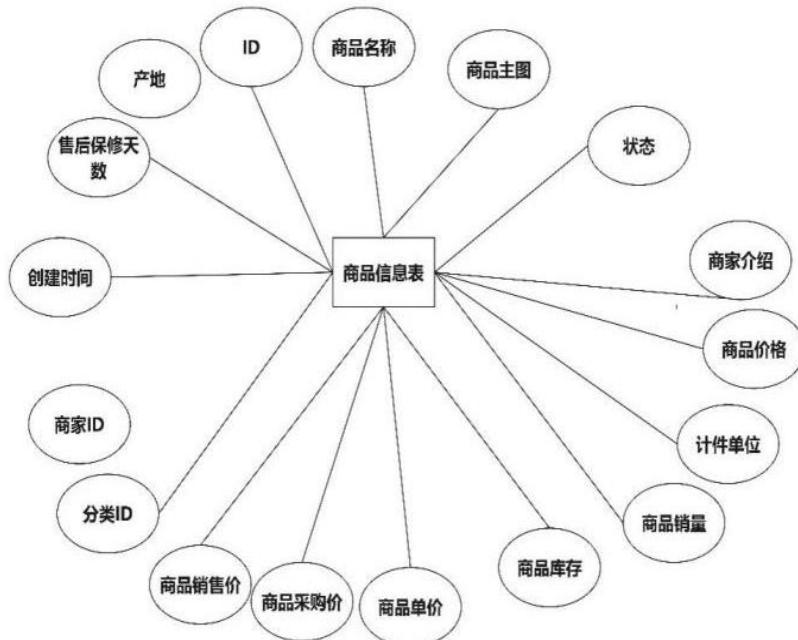


图 4-20 商品信息实体属性

商品信息表 (ID, 名称, 描述, 照片, 规格, 销量, 库存, 点击率, 类型 id, 专卖店 id, 单价, 售

价,采购价,产地,上新时间,售后保修天数,状态,操作人)。如表 4-3:

表 4-3 商品信息表

字段	类型	长度	是否为主键	描述
id	varchar	20	是	ID
name	varchar	20	否	名称
description	varchar	20	否	描述
img	varchar	100	否	照片
unit	varchar	20	否	规格
count	int		否	销量
stock	int		否	库存
click	int		否	点击率
type_id	int		否	类型 id
business_id	int		否	专卖店 id
price	double	10,2	否	单价
selling_price	double	10,2	否	售价
purchase_price	double	10,2	否	采购价
origin	varchar	50	否	产地
newtime	datetime		否	上新时间
warranty	int		否	保修天数
status	varchar	5	否	状态

4.评论信息表 comment

评论信息表存储用户对商品的评价信息，其用户 id 外键是用户表的用户 id 键，其商品 id 外键是商品表的商品 id 主键,其商家 id 外键是商家表的商家 id 主键；如图 4-21：

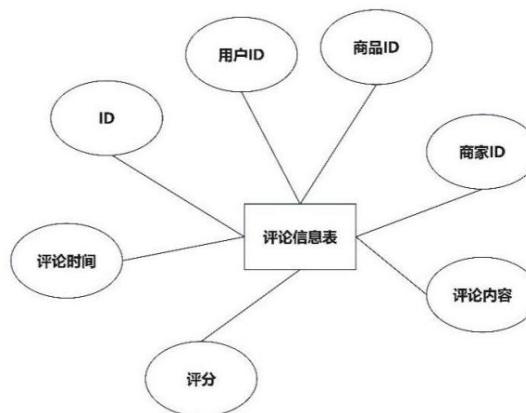


图 4-21 评论信息实体属性

评论信息表（ID, 用户 id, 商家 id, 商品 id, 时间, 评论, 评分）。如表 4-4：

表 4-4 评论信息表

字段	类型	长度	是否为主键	描述
id	varchar	20	是	ID
user_id	int		否	用户 id
business_id	int		否	商家 id
goods_id	varchar	20	否	商品 id
time	datetime			时间
content	varchar	255		评论
score	int		否	评分

5. 订单信息表 orders

订单信息表用于存储订单信息，其用户 id 外键是用户表的用户 id 键，其商品 id 外键是商品表的商品 id 主键，其商家 id 外键是商家表的商家 id 主键，其地址 id 外键是地址表的地址 id 主键；如图 4-22：

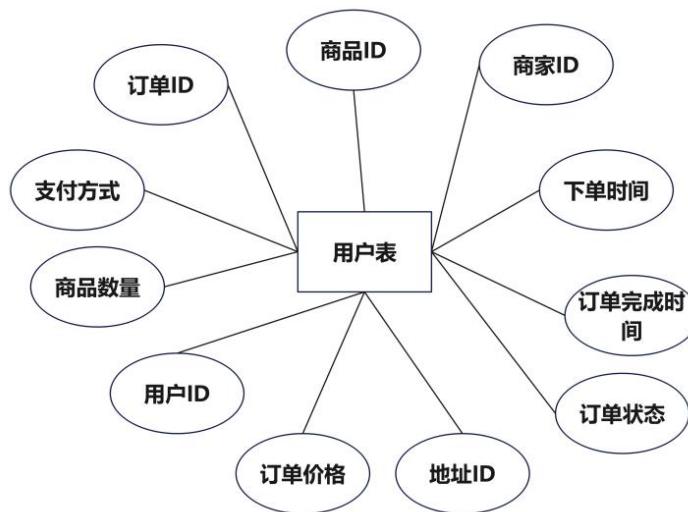


图 4-22 订单信息实体属性

订单信息表（ID, 用户 id, 商家 id, 商品 id, 地址 id, 公共订单号, 数量, 价格, 状态, 下单时间, 订单完成时间, 支付方式）。如表 4-5：

表 4-5 订单信息表

字段	类型	长度	是否为主键	描述
id	varchar	20	是	ID
user_id	int		否	用户 ID
business_id	int		否	商家 ID
goods_id	varchar	20	否	商品 ID

续表 4-5 订单信息表

字段	类型	长度	是否为主键	描述
address_id	varchar	20	否	地址 ID
order_id	varchar	20	否	公共订单号
num	int		否	数量
price	double	10,2	否	价格
status	varchar	5	否	状态
time	datetime		否	下单时间
finish_time	datetime		否	订单完成时间
pay	varchar	10	否	支付方式

6. 审核信息表 audit

审核信息表用于存储审核信息；如图 4-23：

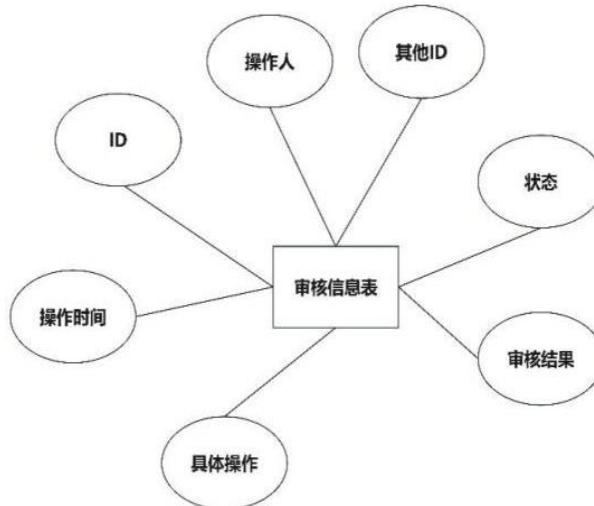


图 4-23 审核信息实体属性

审核信息表 (ID, 操作人, 对应表编号, 状态, 审核结果, 具体操作, 操作时间)。如表 4-6：

表 4-6 审核信息表

字段	类型	长度	是否为主键	描述
id	int		是	ID
user	varchar	10	否	操作人
number	varchar	20	否	对应表编号
status	int		否	状态
results	varchar	255	否	审核结果
operate	varchar	10	否	具体操作
time	datetime		否	操作时间

7. 采购信息表 procurement

采购信息表用于存储采购信息，其供应商 id 外键是供应商表的供应商 id 键，其员工 id 外键是员工表的员工 id 主键，其子类 id 外键是商表的商家 id 主键；如图 4-24：

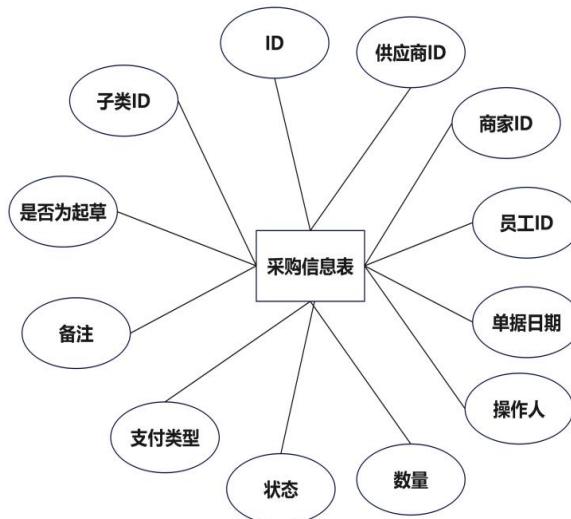


图 4-24 采购信息实体属性

采购信息表（ID, 供应商 ID, 商家 ID, 员工 ID, 单据日期, 操作人, 数量, 状态, 支付类型, 价格合计, 本次付款, 本次欠款, 采购类型, 备注, 是否为起草, 子类 ID）。如表 4-7：

表 4-7 采购信息表

字段	类型	长度	是否为主键	描述
id	varchar	20	是	主键
supplier_id	varchar	20	否	供应商 ID
business_id	int		否	商家 ID
staff_id	int		否	员工 ID
time	datetime		否	单据日期
operator	varchar	20	否	操作人
num	int		否	数量
status	varchar	10	否	状态
pay_type	varchar	255	否	支付类型
price	double	10,2	否	价格合计
payment	double	10,2	否	本次付款
deposit	double	10,2	否	本次欠款
type	varchar	10		采购类型
remark	varchar	255		备注

续表 4-7 采购信息表

字段	类型	长度	是否为主键	描述
draft	int			是否为起草
procurement_id	varchar	20		子类 ID

8. 销售信息表 sale

销售信息表用于存储销售信息，其客户 id 外键是客户表的客户 id 键，其员工 id 外键是员工表的员工 id 主键，其商家 id 外键是商表的商家 id 主键；如图 4-25：

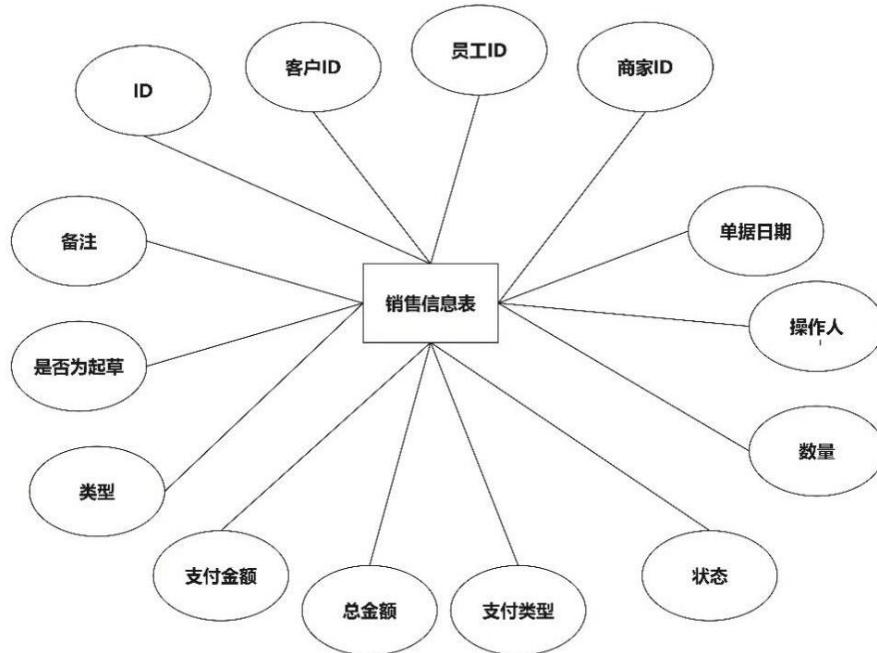


图 4-25 销售信息实体属性

销售信息表（ID,客户 ID,商家 ID,员工 ID,单据日期,操作人,数量,支付类型,总金额,支付金额,采购类型,状态,备注,是否为起草）。如表 4-8：

表 4-8 销售信息表

字段	类型	长度	是否为主键	描述
id	varchar	20	是	ID
customer_id	varchar	20	否	客户 ID
business_id	int		否	商家 ID
staff_id	int		否	员工 ID
time	datetime		否	单据日期
operator	varchar	10	否	操作人
num	int		否	数量
pay_type	varchar	10	否	支付类型

续表 4-8 销售信息表

字段	类型	长度	是否为主键	描述
price	double	10,2	否	总金额
payment	double	10.2	否	支付金额
type	varchar	10	否	采购类型
status	varchar	10	否	状态
remark	varchar	255		备注
draft	int			是否为起草

9.出入库信息表 storage

出入库信息表用于存储出入库信息，其客户 id 外键是客户表的客户 id 键，其供应商 id 外键是供应商表的供应商 id 键，其员工 id 外键是员工表的员工 id 主键，其销售 id 外键是销售表的销售 id 主键，其采购 id 外键是采购表的采购 id 主键，其仓库 id 外键是仓库表的仓库 id 主键，其订单 id 外键是订单表的订单 id 主键；如图 4-26：

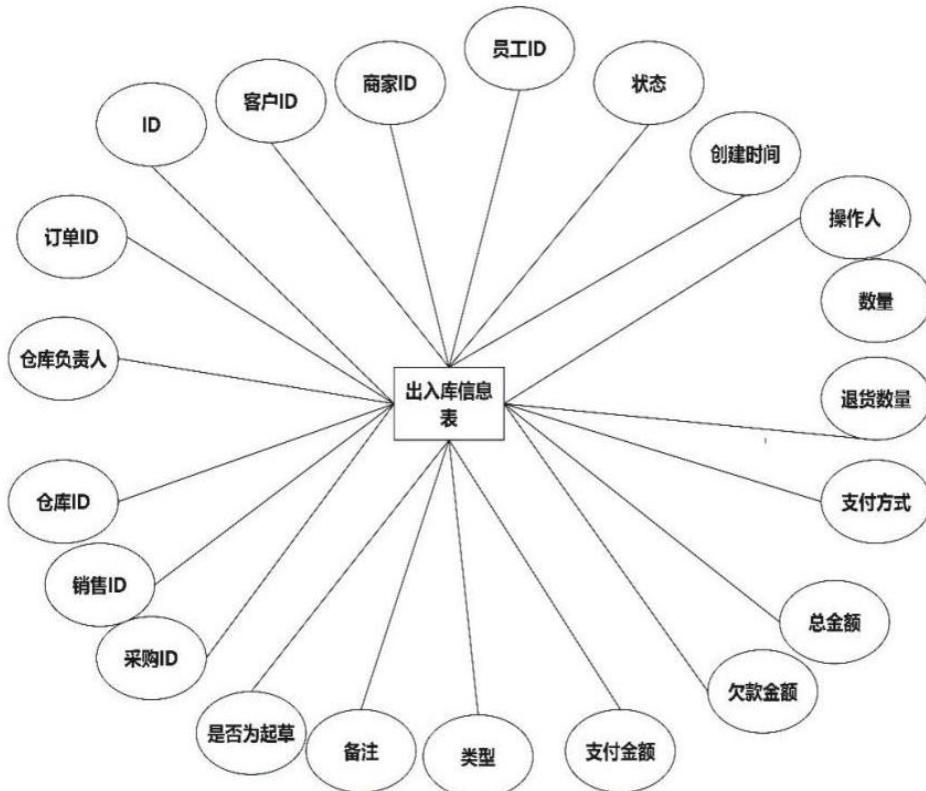


图 4-26 出入库信息实体属性

出入库信息表 (ID,客户 ID,供应商 ID,商家 ID,员工 ID,状态,单据日期,操作人,数量,退货,支付类型,总金额,欠款金额,支付金额,扣除金额,采购类型,备注,是否为起草,采购 ID,销售 ID,订单 id,仓库 id,仓库管理人)。如表 4-9：

表 4-9 出入库信息表

字段	类型	长度	是否为主键	描述
id	varchar	20	是	ID
customer_id	varchar	20	否	客户 ID
supplier_id	varchar	20	否	供应商 ID
business_id	int		否	商家 ID
staff_id	int		否	员工 ID
status	varchar	10	否	状态
time	datetime		否	单据日期
operator	varchar	10	否	操作人
num	int		否	数量
num_return	int		否	退货
pay_type	varchar	10	否	支付类型
price	double	10,2	否	总金额
deposit	double	10,2	否	欠款金额
payment	double	10,2	否	支付金额
deduct	double	10,2	否	扣除金额
type	varchar	10	否	采购类型
remark	varchar	255	否	备注
draft	int		否	是否为起草
procurement_id	varchar	20	否	采购 ID
sale_id	varchar	20	否	销售 ID
orders_id	varchar	20	否	订单 ID
depot_id	varchar	20	否	仓库 ID
depot_staff	int		否	仓库管理人

10.售后维修信息表 repair

售后维修信息表用于存储售后维修信息，其员工 id 外键是员工表的员工 id 主键，其出入库 id 外键是出入库表的出入库 id 主键，其用户 id 外键是用户表的用户 id 主键，其商家 id 外键是商家表的商家 id 主键，其商品 id 外键是商品表的商品 id 主键，其订单 id 外键是订单表的订单 id 主键；如图 4-27：

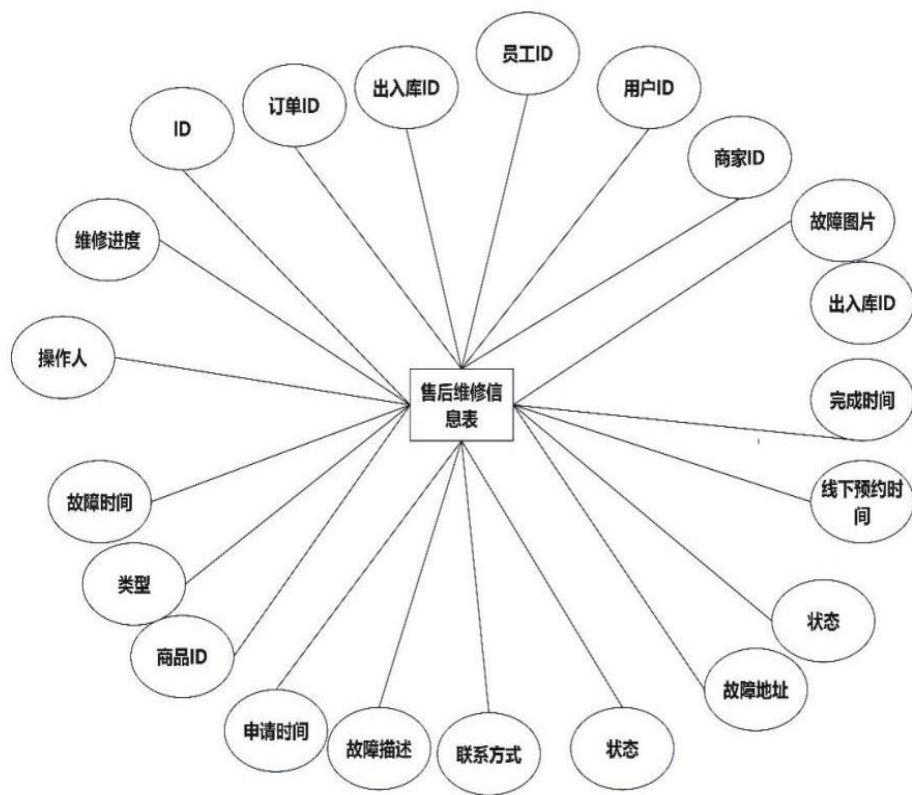


图 4-27 售后维修信息实体属性

售后维修信息表 (ID,订单 ID,出入库 ID,申请人,故障图片,完成时间,线下预约时间,申请状态,地址,联系方式,故障描述,申请时间,商家 ID,员工 ID,用户 ID,类型,售后商品,维修方式,故障时间,操作人,维修进度)。如表 4-10:

表 4-10 售后维修信息表

字段	类型	长度	是否为主键	描述
id	varchar	20	是	ID
order_id	varchar	20	否	订单 ID
storage_id	varchar	20	否	出入库 ID
user	varchar	20	否	申请人
fault_img	varchar	100	否	故障图片
finish_time	datetime		否	完成时间
booking_time	datetime		否	线下预约时间
status	varchar	5	否	申请状态
address	varchar	100	否	地址
phone	varchar	20	否	联系方式
fault_descr	varchar	255	否	故障描述
time	datetime		否	申请时间

续表 4-10 售后维修信息表

字段	类型	长度	是否为主键	描述
business_id	int		否	商家 ID
staff_id	varchar	20	否	员工 ID
user_id	int		否	用户 ID
type	varchar	10	否	类型
goods_id	varchar	20	否	售后商品
repair_type	varchar	10	否	维修方式
fault_time	date		否	故障时间
operator	varchar	20	否	操作人
pace	varchar	10	否	维修进度

11. 批量信息表 batch

批量信息表用于存储审核信息，用于存储每个单据中包含的商品信息，其销售 id 外键是销售表的销售 id 主键，其采购 id 外键是采购表的采购 id 主键，其出入库 id 外键是出入库表的出入库 id 主键，其商品 id 外键是商品表的商品 id 主键；如图 4-28：

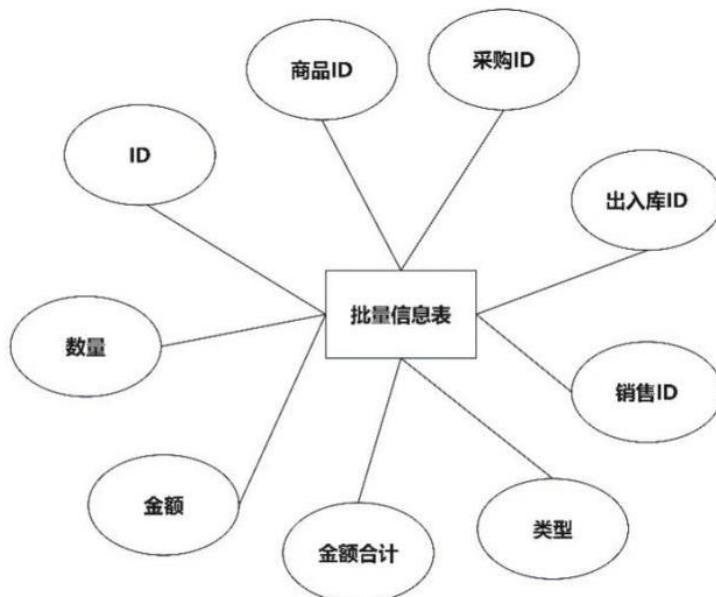


图 4-28 批量信息实体属性

批量信息表 (ID,商品 ID,数量,金额,合计,类型,采购 ID,出入库 ID,销售 ID)。如表 4-11：

表 4-11 批量信息表

字段	类型	长度	是否为主键	描述
id	int		是	ID
goods_id	varchar	20	否	商品 ID

续表 4-11 批量信息表

字段	类型	长度	是否为主键	描述
count	int		否	数量
price	double	10,2	否	金额
total	double	10,2	否	合计
type	varchar	10	否	类型
procurement_id	varchar	20	否	采购 ID
storage_id	varchar	20		出入库 ID
sale_id	varchar	20		销售 ID

4.4 本章小结

本章全面阐述了医疗器械专卖店管理系统的设计方案。在概要设计中，明确将系统分为前台与后台两大部分，并展示了系统功能结构图，使系统的整体架构一目了然。详细设计部分虽未具体展开，但为后续开发提供了方向指引。

数据库设计是本章重点，通过 E-R 图清晰呈现了系统内各实体及其相互关系，包括用户、订单、收货地址等实体，让开发人员对数据的关联和约束有了直观理解。在数据库表设计上，详细列出了用户表、收货地址表等关键表的字段、类型、长度、主键以及描述信息，为数据库的构建提供了精确蓝图。这些设计工作为系统的稳定运行和高效数据交互奠定了坚实基础，确保系统能够满足医疗器械专卖店复杂的业务管理需求，为后续的系统开发与实现提供了关键的技术支撑和保障，使整个系统的设计具有科学性、合理性与可扩展性。

第五章 系统实现

5.1 前台模块

5.1.1 首页展示实现

医疗器械专卖店系统的前台首页主要展示三项核心板块：热销商品板块，依据销售热度的前十款热销医疗器械；猜你喜欢板块，该板块运用协同过滤算法，分析用户收藏、加购、下单及评论等行为数据，推送用户喜好商品；分类列表板块，用户可以定位该分类商品。其数据交互流程为：前端通过 axios 向后端请求，后端将数据回传，前端负责对数据渲染。

首页前端核心代码如下：

```
//猜你喜欢商品列表显示
<el-col :span="5" v-for="item in goodsData">

<div>{{item.name}}</div>
<div>
<div style="font-size: 15px; color: #FF5000FF;">
    ¥ {{ item.sellingPrice }} / {{ item.unit }}
</div>
<div style="margin-left: 70px; color: #909399;">
    销量: {{item.count}}
</div>
</div>
</el-col>
```

前台首页后端核心代码如下：

```
//前台显示 10 件商品，热销商品
@GetMapping("/selectTop10")
public Result selectTop10() {
    List<Goods> list = goodsService.selectTop10();
    return Result.success(list);
}
```

```
//协同算法推荐接口 猜你喜欢
@GetMapping("/recommend")
public Result recommend(){
    List<Goods> list=goodsService.recommend();
    return Result.success(list);
}
```

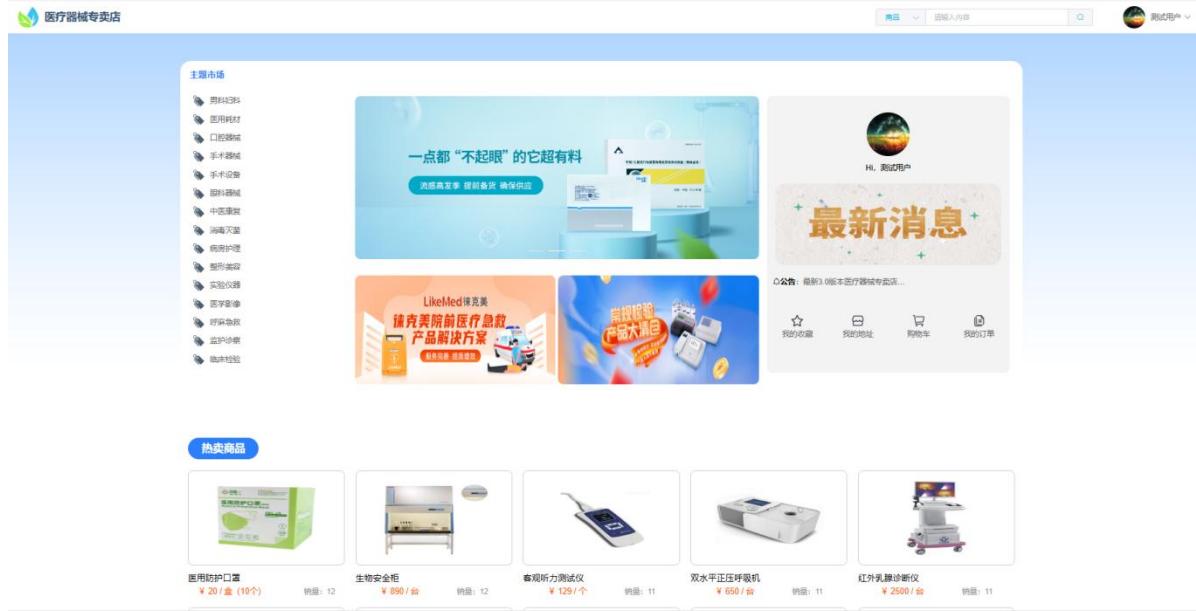


图 5-1 前台首页效果图

5.1.2 查找展示实现

通过用户选择的分类导航按钮或搜索框商品名搜索去查找对应商品信息并展示，也可根据搜索框商家名称查找对应的商家。

查找功能前端核心代码如下：

```
//搜索框查找按钮
<el-input placeholder="请输入内容" style="width: 400px" v-model="name" class="input-with-select">
    <el-select v-model="select" style="width: 90px" slot="prepend" placeholder="请选择">
        <el-option label="商品" value="1"></el-option>
        <el-option label="专卖店" value="2"></el-option>
```

```
</el-select>
<el-button slot="append" icon="el-icon-search" @click="search"></el-button>
</el-input>
```

查找功能后端核心代码如下：

```
//根据商品名查找商品
@GetMapping("/selectByName")
public Result selectByName(@RequestParam String name) {
    List<Goods> list = goodsService.selectByName(name);
    return Result.success(list);
}

//根据商家名称查找商家
@GetMapping("/selectPage")
public Result selectPage(Business business, @RequestParam(defaultValue = "1") Integer pageNum, @RequestParam(defaultValue = "10") Integer pageSize) {
    PageInfo<Business> page = businessService.selectPage(business, pageNum, pageSize);
    return Result.success(page);
}
```

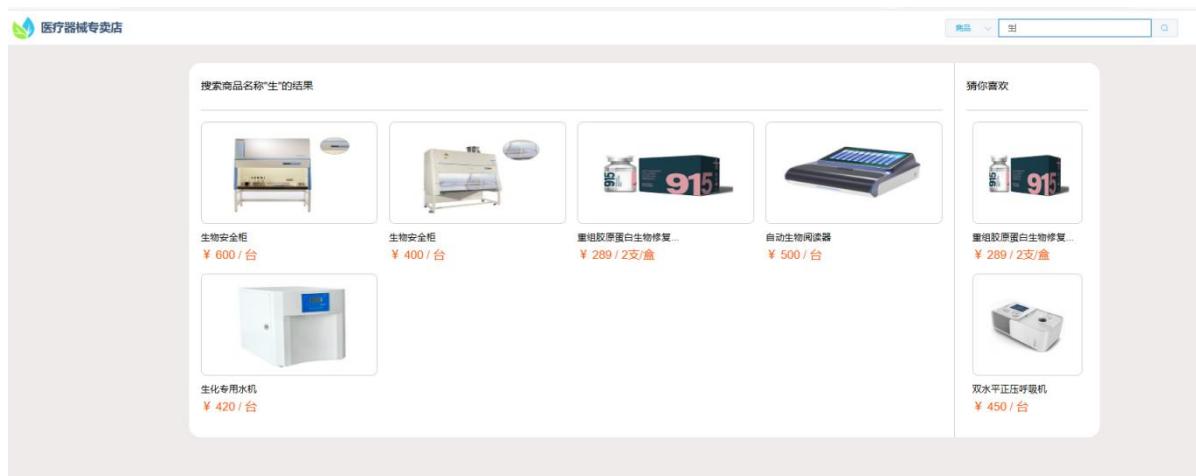


图 5-2 商品查找效果图

5.1.3 商品详情页展示实现

商品详情页呈现商品的详细信息，涉及多表查询商品类别以及商家信息等，涵盖商品库存、产地、销量等关键内容。用户可查看浏览，若想将商品加入购物车，可在

该页面选择商品数量并添加至购物车。此外，还能在此页面查看与该商品相关的评论信息。

商品详情页前端核心代码如下：

```
<el-row :gutter="20">
<el-col :span="12">
    
</el-col>
<el-col :span="12">
    <div>{{goodsData.name}}</div>
    <div>销量： {{goodsData.count}}</div>
    <div style="margin-top: 10px;color: #666666FF;">
        数量:<el-input-number v-model="num" :min="1" :max="goodsData.stock"
            @change="handleChange"/>库存:
        {{ goodsData.stock }}</div>
        //此处略商品详情其他属性代码
    </el-col>
</el-row>
```

商品详情页后端核心代码如下：

```
//商品详情
@GetMapping("/selectById")
public Result selectById(@RequestParam String id) {
    Goods goods = goodsService.selectById(id);
    return Result.success(goods);
}

//加入购物车
@PostMapping("/add")
public Result add(@RequestBody Cart cart) {
    cartService.add(cart);
    return Result.success();
}
```

商品评论详情后端核心代码如下：

```
//商品评论，根据商品 id 查询对应的评论
@GetMapping("/selectByGoodsId")
public Result selectByGoodsId(@RequestParam String id) {
    List<Comment> list = commentService.selectByGoodsId(id);
    return Result.success(list);
}
```



图 5-3 商品详情页效果图

5.1.4 购物车展示实现

用户在登录成功后可进入购物车页面，可对商品进行数量修改、选中/取消选中、删除等操作，购物车实时计算商品总价和优惠金额，并与订单系统无缝对接，方便用户快速结算下单，同时支持购物车商品长期保存，便于用户随时回来继续购物。

购物车前端核心代码如下：

```
<el-table :data="goodsData" strip @selection-change="handleSelectionChange">
    <el-table-column prop="goodsName" label="商品名称" >
        <template v-slot="scope">
            <a :href="/front/detail?id=' +
```

```

scope.row.goodsId">{{scope.row.goodsName}}</a>
        </template>
    </el-table-column>
<el-table-column prop="goodsPrice" label="商品价格" align="center">
    <template v-slot="scope">
        {{scope.row.goodsPrice}}元
    </template>
</el-table-column>

```

购物车后端核心代码如下：

```

//购物车更新
@GetMapping("/update")
public Result updateById(@RequestBody Cart cart) {
    cartService.updateById(cart);
    return Result.success();
}

//购物车列表显示
@GetMapping("/selectPage")
public Result selectPage(Cart cart,
    @RequestParam(defaultValue = "1") Integer pageNum,
    @RequestParam(defaultValue = "10") Integer pageSize)
{
    PageInfo<Cart> page = cartService.selectPage(cart, pageNum, pageSize);
    return Result.success(page);
}

//购物车删除
@DeleteMapping("/delete/{id}")
public Result deleteById(@PathVariable Integer id) {
    cartService.deleteById(id);
    return Result.success();
}

```



图 5-4 购物车效果图

5.1.5 我的订单展示实现

系统根据用户登录信息，获取其对应订单数据，通过 el-table 组件渲染列表。可对订单状态为“待收货”订单进行，确认收货和评价操作。

我的订单前端核心代码如下：

```
<el-table :data="filteredOrdersData" strip>
<el-table-column label="商品图片" width="120px">
  <template v-slot="scope">
    <el-image v-if="scope.row.goodsImg"
      :src="scope.row.goodsImg" :preview-src-list="[scope.row.goodsImg]"
    "></el-image>
  </template>
</el-table-column>
<el-table-column prop="orderId" label="订单编号" show-overflow-tooltip></el-table-column>
  //此处略订单其他属性代码
</el-table>
```

我的订单后端核心代码如下：

```
@PostMapping("/add")
public Result add(@RequestBody Comment comment) {
  commentService.add(comment);
  return Result.success();
}
```

```
//订单列表显示
@GetMapping("/selectPage")
public Result selectPage(Orders orders,
    @RequestParam(defaultValue = "1") Integer pageNum,
    @RequestParam(defaultValue = "10") Integer pageSize) {
    Account account= TokenUtils.getCurrentUser();
    if (RoleEnum.USER.name().equals(account.getRole())) {
        orders.setUserId(account.getId());
    }
    PageInfo<Orders> page = ordersService.selectPage(orders, pageNum, pageSize);
    return Result.success(page);
}
```

我的订单 (7个)											
商品图片	订单编号	商品名称	店铺名称	商品价格	商品数量	订单总价	支付方式	收货人	收货地址	联系电话	操作
	OR2024...	客视听...	贝贝医...	99 / 个	1	99	微信	米兰	广西南...	159774...	查看详情 单据历史 删除
	OR2024...	妇产科...	亚利医...	4500 / 台	1	4500	微信	米兰	广西南...	159774...	查看详情 单据历史 删除

图 5-5 我的订单效果图



图 5-6 订单评价效果图



图 5-7 单据历史效果图

5.1.6 结算下单展示实现

可选定购买商品，即时自动计算总价。点击结算后，页面将跳转至确认订单界面。在确认订单界面，选择收货地址；同时，页面显示已选商品的详细信息以及支付金额，用户既可选择返回购物车，也能点击“立即支付”按钮。用户完成下单，系统自动将对应商品从购物车中清除。前端采用 el-steps 步骤组件实时显示流程步骤，el-table 表格组件显示商品信息。

首页前端部分核心代码如下：

```

<el-steps :active="activeStep" finish-status="success">
  <el-step title="确认订单"></el-step>
  <el-step title="完成"></el-step>
</el-steps>
<el-table :data="tableData" stripe>
  <el-table-column prop="goodsName" label="商品名称" align="center"></el-table-column>
  //此处略商品信息其他属性代码
</template>

```

结算下单后端核心代码如下：

```

public void add(Orders orders) {
    orders.setOrderId(generateUniqueId("OR"));
    for (Cart cart : orders.getCartData()) { //遍历 购物车下单传输过来的数组
        Orders dbOrders = new Orders();
        BeanUtils.copyProperties(orders, dbOrders); //复制 orders 到 dbOrders 对象。
        dbOrders.setId(generateId("O"));
        dbOrders.setGoodsId(cart.getGoodsId());
        dbOrders.setBusinessId(cart.getBusinessId());
        dbOrders.setNum(cart.getNum());
        dbOrders.setPrice(cart.getNum() * cart.getGoodsPrice());
        dbOrders.setTime(DateUtil.now());
        ordersMapper.insert(dbOrders);
        AuditUtils.saveAudit(auditMapper, 0, dbOrders.getId(), null, "下单成功，等待发货");//记录单据流程
        cartMapper.deleteById(cart.getId());
    }
}

```

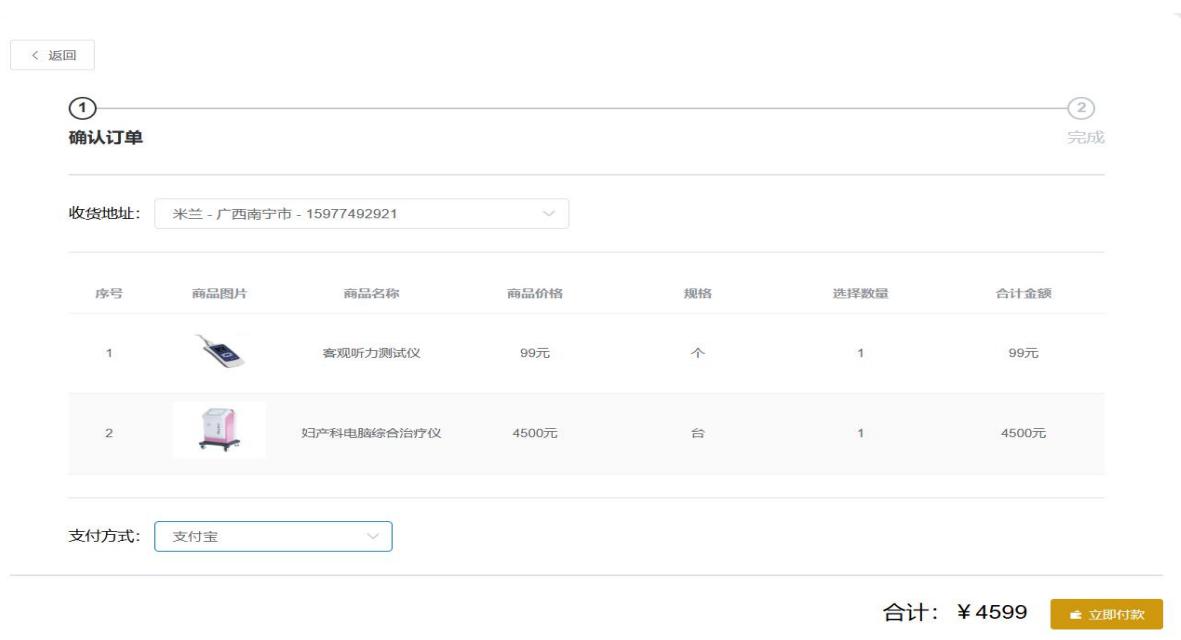


图 5-8 结算下单效果图

5.1.7 售后维修展示实现

前台普通用户查看个人售后维修信息。可编辑已有信息（如修正故障描述等），

也可新增申请，新申请初始为待审核状态。后台审核通过后信息锁定，用户无法修改；处于待审核或审核未通过状态时，用户可编辑。审核未通过时，用户依反馈修改单据后可重新发起审核，直至通过。

前端采用 el-dialog 拟态框组件显示表单组件，售后维修前端核心代码如下：

```
<el-dialog title=" 售 后 维 修 信 息 " :visible.sync="formVisible"
width="40%" :before-close="handleCancel" :close-on-click-modal="false"
destroy-on-close>
  <el-form label-width="100px" :model="form" :rules="rules" ref="formRef">
    <el-form-item prop="user" label="申请人">
      <el-input v-model="form.user" autocomplete="off" readonly></el-input>
    </el-form-item>
    //此处略售后维修申请其他表单代码
  </el-form>
</el-dialog>
```

售后维修后端核心代码如下：

```
//售后维修添加
@PostMapping("/add")
public Result add(@RequestBody Repair repair) {
  repairService.add(repair);
  return Result.success();
}

//售后维修列表显示
@GetMapping("/selectPage")
public Result selectPage(Repair repair,
                        @RequestParam(defaultValue = "1") Integer pageNum,
                        @RequestParam(defaultValue = "10") Integer pageSize)
{
  PageInfo<Repair> page = repairService.selectPage(repair, pageNum, pageSize);
  return Result.success(page);
}
```

The screenshot shows a table with columns: 商品名称 (Product Name), 申请人 (Applicant), 联系电话 (Phone Number), 地址 (Address), 故障图片 (Fault Picture), 维修方式 (Repair Method), 故障描述 (Fault Description), 预约到店时间 (预约 Time), 申请时间 (Application Time), 状态 (Status), 操作 (Operations). The rows represent three repair requests:

商品名称	申请人	联系电话	地址	故障图片	维修方式	故障描述	预约到店时间	申请时间	状态	操作
妇产科...	测试用户	159774...	广西南...		寄修	用不了了	2024-12...	2024-12...	已审核	<button>查看详情</button>
音响...	测试用户	159774...	广西南...		到店	1111 5 05:23:41	2024-12-1 2024-12-1	2024-12-1	审核未通过	<button>重新发起审核</button> <button>查看详情</button>
生物安...	测试用户	159774...	广西南...		寄修	坏了	2024-12...	2024-12...	待审核	<button>查看详情</button> <button>编辑</button>

图 5-9 售后维修列表效果图

5.2 后台模块

5.2.1 首页展示实现

首页加载时，前端通过 axios 向后端请求数据，利用 echarts 折线图与柱状图对获取的数据进行可视化处理，随后渲染至页面。

首页前端核心代码如下：

```
// 折线图显示
<div style="flex: 1" class="card">
  <div style="height: 300px" id="line1"></div>
</div>
<script>
  export default {
    mounted() {
      let lineDom1 = document.getElementById('line1')
      let lineChart1 = echarts.init(lineDom1)
      this.$request.get('/saleCountXx').then(res => {
        lineOption1.xAxis.data = res.data?.map(v => v.name)
        lineOption1.series[0].data = res.data?.map(v => v.value)
        lineChart1.setOption(lineOption1)
      })
    }
  }
</script>
```

首页后端核心代码如下：

```

    @GetMapping("/storeCount")
public Result storeCount() {
    Account account = TokenUtils.getCurrentUser();
    Goods g = new Goods();
    String role = account.getRole();
    if (RoleEnum.BUSINESS.name().equals(role)) {
        g.setBusinessId(account.getId());
    }
    List<Goods> goodsList = goodsService.selectAll(g);
    List<Dict> list = new ArrayList<>();
    List<String> nameList = goodsList.stream().map(Goods::getName).collect(Collectors.toList());
    for (String name : nameList) {
        Dict dict = Dict.create();
        Integer sum = goodsList.stream().filter(goods ->
            goods.getName().equals(name)).map(Goods::getStock).reduce(Integer::sum).orElse(0);
        dict.set("name", name).set("value", sum);
        list.add(dict);
    }
    return Result.success(list); // 返回商品库存比例数据
}

```



图 5-10 首页效果图

5.2.2 商品管理展示实现

商家与管理员登录系统后，能够对商品信息增删改查操作。利用 wangeditor 富文本框功能，插入文本信息和图片描述等信息，以图文并茂的形式描述商品。此外，能够依据特定条件，如商品名称关键词、商品分类，快速精准地查询到相关商品信息。

前端采用 wangeditor 富文本框，部分代码如下：

```
let editor

function initWangEditor(content) {  setTimeout(() => {
  if (!editor) {
    editor = new E('#editor')
    editor.config.placeholder = '请输入内容'
    editor.config.uploadFileName = 'file'
    editor.config.uploadImgServer = 'http://localhost:9090/files/wang/upload'
    editor.create()
  }
  editor.txt.html(content)
}, 0)
}
```

商品信息后端核心代码如下：

```
@PostMapping("/add")
public Result add(@RequestBody Goods goods) {
  Account account= TokenUtils.getCurrentUser();
  if (RoleEnum.BUSINESS.name().equals(account.getRole())){
    goods.setBusinessId(account.getId());//初始化为当前的商家 id
  }
  goodsService.add(goods);//商品新增
  return Result.success();
}

@PostMapping("/update")
public Result updateById(@RequestBody Goods goods) {
  goodsService.updateById(goods);
  return Result.success();//商品修改
}
```

```
// 商品根据 ID 查询  
@GetMapping("/selectById")  
public Result selectById(@RequestParam String id) {  
    Goods goods = goodsService.selectById(id);  
    return Result.success(goods);  
}
```

商品管理																
商品列表																
商品信息																
商品列表																
序号	ID	商品主图	商品名称	商品分类	商品描述	采购价	零售价	销售价	规格	商品库存	商品销量	商品产地	保修天数	所属专卖店	创建时间	操作
1	SP2024...		红外乳胶...	医学影像	点击查看	1200	1200	2500	台	9	11	上海	30	贝贝医疗...	2024-12-13 04:11:48	编辑 删除 详情
2	SP2024...		24小时...	监护诊察	点击查看	100	100	200	个	9	10	深圳	30	贝贝医疗...	2024-11-26 00:00:00	编辑 删除 详情
3	SP2024...		医用防护...	医用耗材	点击查看	10	10	20	盒(10...)	9	12	深圳	30	贝贝医疗...	2024-11-26 00:00:00	编辑 删除 详情
4	SP2024...		全自动血...	临床检验	点击查看	499	499	599	台	10	10	山东	30	贝贝医疗...	2024-11-26 00:00:00	编辑 删除 详情
5	SP2024...		客规听力...	监护诊察	点击查看	99	99	129	个	8	11	珠海	30	贝贝医疗...	2024-11-26 00:00:00	编辑 删除 详情

图 5-11 商品信息列表效果图

商品描述	<div style="border-bottom: 1px solid #ccc; padding-bottom: 5px;"> H B T F I U S ≡ ≡+ ↶ ↶ </div> <div style="border-bottom: 1px solid #ccc; padding-bottom: 5px;"> ⌚ ≡ ✓ ≡ “” 😊 🖼 ▶ ≡ >_> — </div> <div style="border-bottom: 1px solid #ccc; padding-bottom: 5px;"> ⌚ ⌚ ⌚ </div>
<p>商品参数</p> <p>通风系统 排气量m³/h(CFM): 530(130)</p> <p>热量输出 25°C下热能输出(kW): ~0.15</p> <p>噪音dB(A): <65</p> <p>光脚温度f(Lx): >65/700</p> <p>功率(kW): 0.2</p> <p>电流(A): 3.3</p> <hr/> <p>包装</p> <p>包装单位: 台</p> <p>包装体积: 152.00×130.00×80.20cm</p> <p>毛重: 195.00kg</p>	
<h2>产品优势</h2>	

图 5-12 富文本效果图

5.2.3 订单管理展示实现

商家登录系统后台后，在订单管理中查看和管理所属店铺的订单信息，对待发货的订单进行发货。在确认发货时，填写物流单号，用户可在单据历史中查看物流，了解订单的进度和物流状态。

前端订单发货实现采用弹窗录入，使用 Element ui 组件库中的 el-dialog 实现，

部分代码如下：

```
<el-dialog title="发 货" :visible.sync="dialogVisible" width="30%">
  @close="form={}"

  <el-form :model="form" label-width="80px" :rules="rules" ref="formRef">
    <el-form-item label="物流单号" prop="expressNumber">
      <el-input v-model="form.expressNumber" autocomplete="off"></el-input>
    </el-form-item>
  </el-form>

  <span slot="footer" class="dialog-footer">
    <el-button @click="dialogVisible=false">取 消</el-button>
    <el-button type="primary" @click="handleConfirm">确 定</el-button>
  </span>
</el-dialog>
```

订单管理后端核心代码如下：

```
//订单列表
@GetMapping("/selectPage")
public Result selectPage(Orders orders,
                        @RequestParam(defaultValue = "1") Integer pageNum,
                        @RequestParam(defaultValue = "10") Integer pageSize)
{
    Account account= TokenUtils.getCurrentUser();
    if (RoleEnum.BUSINESS.name().equals(account.getRole())) {
        orders.setBusinessId(account.getId());//只能查自己店铺的订单信息
    }
    PageInfo<Orders> page = ordersService.selectPage(orders, pageNum, pageSize);
    return Result.success(page);
}

@PutMapping("/update")
public Result updateById(@RequestBody Orders orders) {
    ordersService.updateById(orders);//发货状态更新
    return Result.success();
}
```

图 5-13 订单信息列表效果图

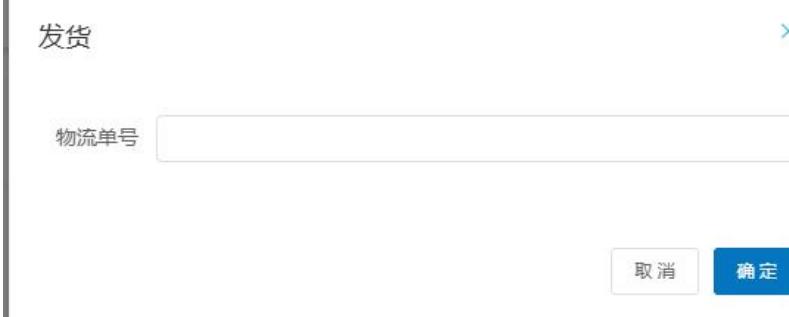


图 5-14 发货效果图

5.2.4 采购管理展示实现

1. 采购预定

采购员主要负责店铺商品采购信息的管理工作。通过开展采购预定工作，从商品清单中挑选多个商品录入采购数量。完成填写后，若选择“保存”，信息将处于草稿“待发起”状态，可供随时修改完善；若选择“保存并发起审核”，便提交给商家，随即进入“待审核”状态，此时无法再做改动。一旦审核未通过，采购员需依据审核意见对商品信息进行编辑修改，接着重新“保存并发起审核”，不断重复此流程，直至审核通过。而当该采购预定单在后续流程中被启用，其状态会自动更新为“完成采购”。采购员可通过单据历史查看该单据的历史审核记录。

前端商品清单和单据历史采用自定义对话框组件，部分代码如下：

```
<choose-goods-dialog
:visible.sync="isDialogVisible"
@save="handleGoodsSave"
/>
<LookAuditDialog :visible.sync="auditDialogVisible" :number="number" />
```

采购预定后端核心代码如下：

```
//添加
public void add(Procurement procurement) {
    procurementMapper.insert(procurement); //采购信息添加
    for (Batch batch : procurement.getBatchData()) {
        Batch dbBatch = new Batch();
        dbBatch.setGoodsId(batch.getGoodsId());
        dbBatch.setPrice(batch.getPrice());
        dbBatch.setCount(batch.getCount());
        dbBatch.setTotal(batch.getPrice()*batch.getCount());
        dbBatch.setType("CGYD");
        dbBatch.setProcurementId(procurement.getId());
        batchMapper.insert(dbBatch); //将采购所选商品存入采购商品列表中
    }
}
```

	序号	单据编号	商品	单据日期	数量	操作员	状态	操作
<input type="checkbox"/>	1	CGYD20241215235833	24小时动态血压监测仪...	2024-12-15 23:58:33	2	贝贝采购	待审核	<button>查看详情</button> <button>删除</button> <button>查看状态</button>
<input type="checkbox"/>	2	CGYD20241215235749	24小时动态血压监测仪...	2024-12-15 23:57:49	2	贝贝采购	待发起	<button>编辑</button> <button>查看详情</button> <button>删除</button> <button>查看状态</button>
<input type="checkbox"/>	3	CGYD20241212230220	红外乳腺诊断仪生物安...	2024-12-12 23:02:20	2	贝贝采购	完成采购	<button>查看详情</button> <button>删除</button> <button>查看状态</button>
<input type="checkbox"/>	4	CGYD2024121223802	双水平正压呼吸机(红外...	2024-12-12 23:08:02	4	贝贝采购	完成采购	<button>查看详情</button> <button>删除</button> <button>查看状态</button>

图 5-15 采购预定列表效果图

2. 采购单

采购员有权新增采购单，新增时需关联已审核通过的采购预定单并选定单号，系统自动带出商品信息并计算总金额与总数量，补充供应商信息、采购价格、付款方式等，结清付款金额，否则后续采购入库环节无法调用。填写完可保存为草稿或直接提交审核，审核未通过则按反馈意见针对性编辑，再次保存后重新提交，采购单启用后状态自动更新为“完成采购”。

前端采购单列表分页采用 Element ui 组件库中的 el-pagination，部分代码如下：

```
<el-pagination background @current-change="handleCurrentChange"
:current-page="pageNum" :page-sizes="[5, 10, 20]" :page-size="pageSize"
layout="total, prev, pager, next" :total="total">
</el-pagination>
```

采购单与采购预定公用后端代码。

序号	单据编号	供应商	商品	单据日期	数量	金额合计	支付金额	操作员	状态	操作
1	CG20241212230359	南宁医疗器械厂	生物安全柜...	2024-12-12	6	4300	0	贝贝采购	已审核	<button>查看详情</button> <button>查看状态</button>
2	CG20241212230317	北海安吉医疗器械厂	生物安全柜...	2024-12-12	2	1700	1000	贝贝采购	完成采购	<button>查看详情</button> <button>查看状态</button>
3	CG20241212224036	北海安吉医疗器械厂	医用防护口罩	2024-12-12	2	110	0	贝贝采购	已审核	<button>查看详情</button> <button>查看状态</button>

图 5-16 采购单信息列表效果图

3. 采购入库

采购员负责新增采购入库信息，操作时可直接选择商品，或选取已通过审核的采购单，系统会自动带出相关采购信息，采购员据此填写实际入库的详细情况，如实际到货数量、入库时间等，然后保存并提交审核。仓管人员负责对采购入库单进行审核，审核过程中严格比对实际到货与入库单信息是否相符，包括商品数量、质量和规格等关键要素。一旦审核通过，系统将自动更新商品库存，实现库存数量的增加。

采购入库前端核心代码如下：

```
<el-form label-width="100px" :model="form" :rules="rules" ref="formRef">
<el-col :span="6">
<el-form-item label="关联采购单" prop="time">
    <el-select v-model="form.procurementId" placeholder="请选择单据">
        <change="handleProcurementChange">
            <el-option v-for="item in
procurementData" :label="item.name" :value="item.id" :key="item.id"></el-option>
```

```

</el-select>
</el-form-item>
</el-col>
//此次采购入库其他表单信息省略
</el-form>

```

采购入库后端核心代码如下：

```

//出入库添加
@PostMapping("/add")
public Result add(@RequestBody Storage storage) {
    storageService.add(storage);
    return Result.success();
}

//出入库列表显示
@GetMapping("/selectPage")
public Result selectPage(Procurement procurement,
                        @RequestParam(defaultValue = "1") Integer pageNum,
                        @RequestParam(defaultValue = "10") Integer pageSize)
{
    PageInfo<Procurement> page = procurementService.selectPage(procurement,
                                                                pageNum, pageSize);
    return Result.success(page);
}

```

序号	单据编号	供应商	商品	单据日期	数量	总金额	待付金额	采购单金额	支付金额	欠款金额	入库仓库	操作员	状态	操作
1	CGRK20241212230535	北海安吉医疗器械厂	红外乳...	2024-12-...	2	1700	700	1000	700	0	仓库1	贝贝采购	已审核	详情
2	CGRK2024121224155	北海安吉医疗器械厂	双水平...	2024-12-...	4	2650	0	2650	0	0	仓库1	贝贝采购	已审核	详情
3	CGRK20241212221131	北海安吉医疗器械厂	生物安...	2024-12-...	4	2249	249	2000	249	0	仓库1	贝贝采购	待审核	详情

图 5-17 采购入库信息列表效果图

4.采购预定/采购单审核

商家在系统中可以查看和审核自己店铺的采购预定和采购单，对于不符合规范的

单据，商家可以拒绝并提供理由，帮助采购员改进。审核通过的单据会被记录。如果商家在审核过程中发现错误，可以在单据未被处理前撤回审核，恢复单据原状。系统还提供了查询单据历史的按钮，让商家能够查看单据的审核记录，包括时间、操作人和审核意见等信息。

前端审核采用自定义对话框 AuditDialog，部分代码如下：

```
<AuditDialog :visible.sync="auditDialog" :number="number" :onSubmit="handleSubmit" />
```

采购预定/采购单审核后端核心代码如下：

```
//采购审核状态修改
@PutMapping("/update")
public Result updateById(@RequestBody Procurement procurement) {
    procurementService.updateById(procurement);
    return Result.success();
}
```

序号	单据编号	商品	单据日期	数量	操作员	状态	操作
1	CGYD20241215235833	24小时动态血压监测仪 立夏系列...	2024-12-15 23:58:33	2	贝贝采购	待审核	<button>审核</button> <button>查看详情</button> <button>单据历史</button>
2	CGYD20241212230220	红外乳腺诊断仪 生物安全柜	2024-12-12 23:02:20	2	贝贝采购	完成采购	<button>查看详情</button> <button>单据历史</button>
3	CGYD20241212223802	双水平正压呼吸机 红外乳腺诊断...	2024-12-12 22:38:02	4	贝贝采购	完成采购	<button>查看详情</button> <button>单据历史</button>

图 5-18 采购预定审核列表效果图

序号	单据编号	供应商	商品	单据日期	数量	金额合计	支付定金	操作员	状态	操作
1	CG20241212230359	贝贝医疗器械专卖店	生物安全柜/红外...	2024-12-12 23:03...	6	4300	0	贝贝采购	已审核	<button>撤回审核</button> <button>查看详情</button> <button>单据历史</button>
2	CG20241212230317	贝贝医疗器械专卖店	生物安全柜/红外...	2024-12-12 23:03...	2	1700	1000	贝贝采购	完成采购	<button>查看详情</button> <button>单据历史</button>
3	CG20241212224036	贝贝医疗器械专卖店	医用防护口罩/24...	2024-12-12 22:40...	2	110	0	贝贝采购	已审核	<button>撤回审核</button> <button>查看详情</button> <button>单据历史</button>
4	CG20241212224013	贝贝医疗器械专卖店	生物安全柜/红外...	2024-12-12 22:40...	4	2650	2650	贝贝采购	完成采购	<button>查看详情</button> <button>单据历史</button>
5	CG20241212224013	贝贝医疗器械专卖店	生物安全柜/红外...	2024-12-12 22:40...	4	2650	2650	贝贝采购	完成采购	<button>查看详情</button> <button>单据历史</button>

图 5-19 采购单审核列表效果图

5.2.5 销售管理展示实现

1. 销售单

销售员负责管理店铺商品销售单信息，能够新增销售单，录入客户信息、支付方式、所售商品、销售价格、总金额以及支付金额等关键数据，并且必须结清款单，否则后续流程无法调用该单据。新增的销售单提交给商家审核通过后，销售员才能在销售出库环节创建对应的销售出库单，同时该销售单的状态则发生改变成“完成销售”。

销售单前端核心代码如下：

```
<el-form label-width="200px" :model="form" :rules="rules1" ref="formRef1">
  <el-form-item label="客户" prop="customerId">
    <el-select v-model="form.customerId" placeholder="请选择客户">
      <el-option v-for="item in customerData" :label="item.name" :value="item.id" :key="item.id"></el-option>
    </el-select>
  </el-form-item>
  <el-form-item label="单据编号" prop="id">
    <el-input v-model="form.id" autocomplete="off" readonly></el-input>
  </el-form-item>
  //此处省略其他表单信息
</el-form>
```

销售单后端核心代码如下：

```
//销售单添加
@PostMapping("/add")
public Result add(@RequestBody Sale sale) {
  saleService.add(sale);
  return Result.success();
}

//销售单修改
@PostMapping("/update")
public Result updateById(@RequestBody Sale sale) {
  saleService.updateById(sale);
  return Result.success();
}
```

图 5-20 销售单信息列表效果图

2. 销售出库

在销售流程中，销售员创建销售出库信息时，既可关联相应销售单带出相关数据，也能直接选定多个商品，此时系统会根据所售商品及其价格自动算出总金额、支付金额、商品数量以及当前欠款等信息，并且只有结清欠款才能将其提交给仓管审核。同时，销售员还需填好出库仓库与支付方式等关键内容后提交审核。仓管会进行核查，若信息准确无误便予以通过，审核通过的瞬间，系统会依据出库数量自动扣减相应商品库存，若存在对应的销售单，其状态将变更为“完成销售”。

销售出库前端核心代码如下：

```
<el-form label-width="100px" :model="form" :rules="rules" ref="formRef">
  <el-form-item label="关联销售单" prop="saleId">
    <el-select v-model="form.saleId" placeholder="请选择单据" @change="handlesaleChange">
      <el-option v-for="item in saleData" :label="item.name" :value="item.id" :key="item.id"></el-option>
    </el-select>
  </el-form-item>
  //此处省略其他表单信息
</el-form>
```

销售出库与采购入库公用后端代码。

序号	单据编号	客户	单据日期	数量	总金额	销售单金额	支付金额	欠款金额	出库仓库	操作员	状态	操作
1	XSCK20241212230951	桂林医疗...	2024-12-...	3	4040	1000	3040	0	仓库2	贝贝销售	已审核	<button>查看详情</button> <button>删除</button> <button>查看状态</button>
2	XSCK20241212224805	桂林医疗...	2024-12-...	2	220	200	20	0	仓库2	贝贝销售	已审核	<button>查看详情</button> <button>删除</button> <button>查看状态</button>
3	XSCK20241211174836	桂林医疗...	2024-12-...	4	2700	0	0	2700	仓库2	贝贝销售	待发起	<button>编辑</button> <button>查看详情</button> <button>删除</button> <button>查看状态</button>
4	XSCK20241211173843	桂林医疗...	2024-12-...	3	3279	0	3279	0	仓库2	贝贝销售	已审核	<button>查看详情</button> <button>删除</button> <button>查看状态</button>

图 5-21 销售出库信息列表效果图

3.线上售后维修

在售后维修服务流程中，前台普通用户提交线上售后维修申请后，销售员对这些申请进行审核。销售员评估申请内容，判断是否符合维修条件和范围。一旦审核通过，销售员便可着手录入维修进度信息。让用户能够实时了解维修进度。

前端线上售后维修列表操作采用作用域插槽和 Element UI 组件库的 Button 按钮，部分代码如下：

```
<template v-slot="scope">
<el-row>
  <el-col>
    <el-button size="mini" type="warning" v-if="scope.row.status==='已审核'" plain
@click="updatePace(scope.row)">维修进度</el-button>
    <el-button size="mini" type="success" v-if="scope.row.status==='待审核'" plain
@click="Audit(scope.row.id)">审核</el-button>
    <el-button size="mini" v-if="scope.row.status==='已审核'||scope.row.status==='
审核未通过'" plain @click="recallAudit(scope.row)">撤回审核</el-button>
    <el-button size="mini" icon="el-icon-document" plain
@click="detailRepair(scope.row)">详情</el-button>
    <el-button size="mini" type="warning" plain
<el-button size="mini" type="warning" plain @click="LookAudit(scope.row.id)"
icon="el-icon-timer">单据历史</el-button>
  </el-col>
</el-row>
</template>
```

线上售后维修和前台售后维修公用后端代码。

图 5-22 线上售后维修信息列表效果图

图 5-23 线上售后维修进度录入效果图

4.线下售后维修

销售员负责录入线下售后维修记录存档。当有线下商品完成维修后，销售员需及时将相关信息录入系统。包括客户的基本信息、所维修商品的基本等基础资料，还需详细录入维修进度信息，方便后期查看。

线下售后维修前端核心代码如下：

```
<el-table :data="repairData" strip @selection-change="handleSelectionChange">
<el-table-column prop="id" label="编号" show-overflow-tooltip></el-table-column>
<el-table-column prop="storageId" label="销售单据" show-overflow-tooltip></el-table-column>
<el-table-column prop="goodsName" label="商品名称" show-overflow-tooltip></el-table-column>
<el-table-column prop="type" label="类型" show-overflow-tooltip></el-table-column>
<el-table-column prop="user" label="申请人" ></el-table-column>
<el-table-column prop="phone" label="联系电话" show-overflow-tooltip></el-table-column>
<el-table-column prop="address" label="地址" ></el-table-column>
```

```
show-overflow-tooltip></el-table-column>
//此处省略其他表单属性
</el-table>
```

线下售后维修与线上售后维修公用后端核心代码。

编号	销售单据	商品名称	类型	申请人	联系电话	地址	故障图片	维修方式	故障描述	预约到店时间	申请时间	维修进度	维修完成时间	操作人	操作	
WX20241...	XSCK202...	奢观听力...	线下订单	小米	15977492...	广西南宁...		寄修	线路坏了	2024-12-...	已完成	2024-12-...	贝贝销售	<button>查看详情</button>	<button>编辑</button>	<button>删除</button>
WX20241...	XSCK202...	奢观听力...	线下订单	小妹	15977492...	广西南宁...		寄修	坏了	2024-12-1...	已完成	2024-12-1...	贝贝销售	<button>查看详情</button>	<button>编辑</button>	<button>删除</button>

图 5-24 线下售后维修信息列表效果图

5.3 本章小结

本章详述了医疗器械专卖店管理系统的实现过程。以商品模块的首页展示为例，前端 axios 与后端接口配合默契，实现了商品的精美呈现。同时，各模块在前端的协同下，功能得以完整、稳定地实现，满足了专卖店管理的多元需求，有力地提升了管理的信息化与高效化水平，也为系统的进一步完善与拓展提供了有力支撑，确保系统在实际应用中发挥重要作用。

第六章 系统测试

6.1 测试环境

硬件环境：

服务器：配置一台性能适中的测试服务器，处理器采用英特尔酷睿 i7 系列，具备多核多线程能力。

客户端设备：准备多台不同配置的电脑作为客户端，模拟真实用户操作环境。

软件环境：

操作系统：服务器端安装 Windows Server 2019 操作系统，确保稳定且能够高效支持各类服务的部署与运行。客户端电脑安装 Windows 10、Windows 11 等操作系统。

浏览器环境：客户端安装多种主流浏览器，包括谷歌、火狐、IE、Edge，确保系统在各类浏览器下的页面显示效果、交互功能以及兼容性均能达到预期标准。

数据库管理系统：以 MySQL 8.0 作为核心数据库支撑，结合功能强大、界面直观友好的 Navicat 工具实施精细化管理。借助 Navicat，轻松搞定数据库连接、表结构创建、SQL 查询执行以及数据备份与恢复等关键操作，全方位保障测试数据的可靠与完整，为系统稳定运行奠定坚实数据基础。

应用服务器：采用内嵌服务器（Tomcat 9.0），凭借自动配置轻松启动后端应用，稳定性佳且适配性强，无需额外部署，便于系统功能与接口测试。

测试工具：

接口测试工具：采用 Postman，可方便地构造各种 HTTP 请求，对系统提供的 API 接口进行功能、参数验证以及性能测试等，比如测试前端与后端交互获取热销商品数据的接口是否准确返回数据、响应时间是否在合理范围内。

自动化测试框架：选择 Selenium，搭配自动化测试脚本，实现对系统前端页面功能的自动化测试，像验证“猜你喜欢”模块是否根据用户行为准确推荐商品、分类列表能否正确跳转等功能的反复测试，提高测试效率和准确性。

通过搭建这样的测试环境，能够全面、细致地对医疗器械专卖店管理系统的各项功能等方面进行测试，确保系统上线后可为用户提供高质量的服务。

6.2 功能测试

全面验证医疗器械专卖店管理系统的各项功能是否符合要求，确保系统在各种正常和异常情况下都能正确、稳定地运行，为用户提供准确、高效且易用的服务体

验，发现并修复可能存在的功能缺陷，提升系统的整体质量和可靠性。

6.2.1 前台首页功能测试

前台首页是用户访问医疗器械专卖店管理系统的入口，提供了轮播图和热销商品、猜你喜欢等内容展示。该测试主要验证前台首页内容是否正常展示。前台首页功能测试结果表如 6-1 所示：

表 6-1 前台首页功能测试

序号	测试功能	测试用例	预期结果	实际结果
1	首页“热销商品”	进入前台首页，查看热销商品区域，点击商品（医用口罩）	展示十条热销商品，可跳转详情页	符合预期结果
2	首页“猜你喜欢”	进入前台首页，查看“猜你喜欢”区域，点击商品（生物安全柜）	展示相关推荐商品，可跳转详情页	符合预期结果
3	首页分类列表	进入前台首页，查看分类列表，点击口腔器械导航	展示分类导航，可跳转对应分类展示页面	符合预期结果

6.2.2 前台商品详情功能测试

前台商品详情功能测试旨在确保用户在浏览商品时能获取准确、完整，基本信息的展示是否清晰无误，且与实际商品相符，关注操作按钮功能是否正常，以及页面布局和加载速度是否优化。前台商品详情功能测试结果表如 6-2 所示：

表 6-2 前台商品详情功能测试

序号	测试功能	测试用例	预期结果	实际结果
1	商品详细信息	点击商品（生物安全感），进入商品详情页	展示商品详情信息	符合预期结果
2	加入购物车	在商品详情页，选择数量，点击加入购物车按钮	提示加入购物车成功，可在购物车中查看	符合预期结果

3	收藏商品	在商品详情页，点击收藏按钮	提示收藏成功，可在我的收藏查看	符合预期结果
4	查看评论	在商品详情页，点击评论按钮	显示评论页面	符合预期结果

6.2.3 购物车功能测试

购物车功能测试主要检验其功能的完整性与稳定性。检查商品添加到购物车后数量、价格等信息是否准确显示，修改商品数量时购物车总价能否实时更新，以及商品删除功能是否有效。购物车功能测试结果表如 6-3 所示：

表 6-3 购物车功能测试

序号	测试功能	测试用例	预期结果	实际结果
1	选择商品	选中购物车列表中的商品	选中的商品自动计算总价	符合预期结果
2	修改数量	进入购物车页面，修改商品数量	数量、总价信息修改更新	符合预期结果
3	删除商品	进入购物车页面，点击对应商品的删除按钮，确认删除	提示删除成功	符合预期结果
4	结算跳转	进入购物车页面，选中商品，点击结算	跳转确认页信息	符合预期结果

6.2.4 订单管理功能测试

订单管理功能测试是对订单管理功能展开全面测试，包括订单发货、查询等流程。重点检测各环节数据传递的准确性与及时性，如订单状态更新是否实时。订单管理功能测试结果表如 6-4 所示：

表 6-4 订单管理功能测试

序号	测试功能	测试用例	预期结果	实际结果
1	订单列表显示	商家登录后进入系统后台，进入订单管理页面	页面正确显示该商家店铺所有订单信息	符合预期结果

2	订单查询	在订单管理页面, 选查询条件 (商品名称: 生物) 进行查询	显示查询订单信息	符合预期结果
3	订单详情查看	在订单列表点详情按钮	弹出详情窗, 信息与订单一致, 布局及操作正常, 打印功能正常	符合预期结果
4	发货操作	在订单列表中, 选择状态为“待发货”订单, 点击发货按钮, 显示编辑弹窗, 填写物流信息, 确认发货	提示发货成功, 状态更新, 返回订单列表	符合预期结果

6.2.5 商品管理功能测试

商品管理功能测试是商家和管理员针对商品管理功能，测试商品信息录入、编辑、查询等操作。检查信息的准确性与完整性，确保商品图片、描述等展示无误。商品管理功能测试结果表如 6-5 所示：

表 6-5 商品管理功能测试

序号	测试功能	测试用例	预期结果	实际结果
1	商品添加	打开商品管理页点击添加按钮, 录入信息, 确认保存	提示添加成功, 返回列表	符合预期结果
2	商品编辑	点击商品列表中的编辑按钮, 修改信息, 确认保存	提示更新成功, 返回列表	符合预期结果
3	商品详情	在商品列表中点击详情按钮	弹出详情页面, 展示商品信息	符合预期结果
4	商品删除	在商品列表中点击删除按钮	提示删除成功	符合预期结果

6.3 本章小结

本章聚焦于医疗器械专卖店管理系统的测试环节，详细阐述了所运用的测试方法与策略，同时给出了各模块功能性测试以及非功能性测试的用例与结果汇总表。经系统且全面的测试流程，有力保障了该系统各项功能运行无误，并充分满足安全性标准与兼容性规范，为系统的稳定上线和可靠运行筑牢根基。

第七章 总结与展望

7.1 主要工作成果

本文旨在设计和实现一个医疗器械专卖店管理系统，通过详细的需求分析、系统设计和功能模块设计，成功完成了系统的开发和实现。深入探究其设计与实现。首先，分析行业背景，指出传统管理模式的不足，开发本系统对专卖店运营和市场竞争的重要意义。在技术选型上，后端采用 Java 语言，结合 SpringBoot 框架搭建服务架构，运用 MyBatis 与 MySQL 数据库交互，保障后端稳定高效；前端运用 Vue.js 框架与 Element UI 组件库，通过 Axios 实现与后端的数据通信，提升用户交互体验。通过需求分析明确系统功能与性能指标，在设计阶段详述架构规划、功能模块构建及数据库设计，各部分紧密协作。其中功能模块扎实构建，前台的商品展示、购物车、用户中心及后台的商品、订单、用户管理等模块完备。商品展示清晰且加载迅速，购物车操作流畅，后台管理高效，数据统计精准，有力支撑业务开展。

完成开发后，进行全方位严格的功能测试，通过反复优化，确保系统稳定可靠。完成了系统集用户登录注册、前后台管理、订单管控等功能于一体，经实践验证，显著提升专卖店运营效率与服务质量，有力推动医疗器械行业数字化进程，具有重要价值和广阔前景。

7.2 工作展望

尽管在系统开发和测试过程中取得了一定的成绩，但仍有一些方面需要进一步优化和完善，同时随着业务的发展和市场的变化，系统也需要不断升级和扩展以满足未来的需求。

首先，功能优化，提升前台用户体验，如优化搜索、完善评价晒单和购物流程；扩展系统功能，如增加会员积分、开发移动端应用和强化数据分析。

其次，性能提升，持续优化系统性能，包括提高响应速度、降低资源消耗，优化数据库和采用缓存技术；加强安全防护，定期漏洞扫描修复，加密敏感信息，完善权限管理。

再者，业务拓展，深化与供应商合作，建立供应链体系，拓展线下及跨境电商销售渠道，实现线上线下融合，开拓国际市场，提升竞争力与盈利能力。

最后，全力推进系统拓展延伸，用户的便利性和灵活性，使系统更具竞争力和市场价值。

参考文献

- [1]周艳芳,侯晓晨.医疗手术常用器械分类与维护保养问题探讨[J].中国防痨杂志,2024,46(10):1296.
- [2]陈胜伟.上市企业融资效率及影响因素的实证研究——以家医疗器械类上市企业为例[J].商展经济,2023,(18):91-96.DOI:10.19995/j.cnki.CN10-1617/F7.2023.18.091.
- [3]Yan J .Exploring the Development Model of Cross border E-commerce Economy from the Perspective of New Development Pattern[J].Computer Informatization and Mechanical System,2024,7(6):48-52.
- [4]闫雨萌.对亚马逊与 eBay 运营模式的分析和建议[J]. 2020.DOI:10.19373/j.cnki.14-1384/f.2020.22.029.
- [5]余永文.跨境电商平台运营模式分析及对策建议——以 eBay 平台为例[J].商场现代化,2021,(12):22-24.DOI:10.14013/j.cnki.scxdh.2021.12.009.
- [6]明日科技.JavaWeb 项目开发全程实录[M].北京:清华大学出版社,2019:1-69
- [7]韩雅雯,陈鹏.中国头部电商平台微博的品牌形象建设研究——以淘宝、京东、抖音、快手为例[J].传媒论坛,2024,7(18):66-68.
- [8]王瑛.基于商圈理论的兰州中小型专卖店营销模式研究[D].兰州大学,2013.
- [9]王爱东.线上线下融合的销售模式在新零售背景下的进一步探讨[J].山西财税,2023,(1):35-37.
- [10]黑马程序员, Spring Boot 企业级开发教程[M].北京:人民邮电出版社,2019:217-270.
- [11]方生.基于“Vue.js”前端框架技术的研究[J].电脑知识与技术,2021,17(19):59-60+64.
DOI:10.14004/j.cnki.ckt.2021.1863.
- [12]庞敏.MySQL 数据库的数据安全应用设计技术研究[J].数字通信世界,2024,(09):25-27.
- [13]郝蕊,李娜.基于 MVC 模式的医疗耗材管理系统设计[J].自动化技术与应用,2024,43(06):171-175.DOI:10.20033/j.1003-7241.(2024)06-0171-05.

- [14] 黄隆俊.自行车专卖店销售管理系统的分析与设计[D].云南大学,2015.
- [15] 李欣蔚.体育用品专卖店管理系统的设计与实现[D].电子科技大学,2014.
- [16] Arulsamy A ,Ong S J ,Watterson L J , et al.Medical Technology: A Systematic Review on Medical Devices Utilized for Epilepsy Prediction and Management[J].Current Neuropharmacology,2022,20(5):950-964.
- [17] 王玉魁,李峰,乔彦超,等.基于 Springboot 与 Vue 框架的仓储管理系统设计与实现[J].河南科技,2024,51(18):29-33.DOI:10.19968/j.cnki.hnkj.1003-5168.2024.18.006.
- [18] 高晓珠.长三角一体化下惠山医疗器械产业的发展前景[J].上海医药,2024,45(05):62-64+69.
- [19] 关传洋.医疗器械维修难点与现代化管理分析[J].中国医疗器械信息,2024,30(18):161-163.DOI:10.15971/j.cnki.cmdi.2024.18.043.

致谢

当医疗器械专卖店管理系统的开发画上句号，我的心中满是感激。

首先诚挚感谢我的导师，在项目的漫漫征途中，无论是系统架构的搭建、技术难关的攻克，还是测试阶段的优化完善，都倾囊相授，用深厚的专业知识和丰富的经验为我指引方向，让我得以在复杂的技术海洋中找准航向，稳步前行。

感恩我的同学们，那些一起在图书馆查阅资料、在教室反复调试代码的日子，因为有他们的陪伴而不再枯燥。我们各抒己见、相互切磋，共同攻克一个又一个难题，每一次思维的交汇都为项目推进注入新的活力。

尤其要把深深的感恩献给我的家人和朋友，在无数个为项目忙碌的日日夜夜，给予我无条件的支持和关爱，用温暖的港湾让我心无旁骛，全身心投入到系统开发中。

这份毕业设计的完成，是众人之力共同铸就。我会带着这份感恩，在未来的道路上坚定地走下去，用更好的成绩回报每一份厚爱。