**杭州电子科技大学信息工程学院**

**本科毕业设计**

**（2026届）**

|  |  |
| --- | --- |
| **题 目** | 基于Spring Boot的智能书城管理系统设计与实现 |
| **学院名称** | 计算机学院 |
| **专 业** | 计算机科学与技术（专升本） |
| **班 级** | 240935105 |
| **学 号** | 249350532 |
| **学生姓名** | 杨帆 |
| **指导教师** | 李舒琴 |
| **完成日期** | 2026年11月 |

摘 要

随着电商的不断发展和人工智能的快速崛起，实体的销售方式正需要迫切的数字化转型。本文实现设计了一个基于SpringBoot的智能商城管理系统，使用了现代化的前后端分离架构，也融合了现在发展成熟的生成式人工智能来节约客服的成本和提升的用户的体验。

管理系统的后端使用了SpringBoot框架来组成restful api服务，利用了mybatis-plus 进行数据持久层的设计，mysql来存储订单、商品、用户、管理员、等重要的业务数据和记录，redis作为一个缓存层来提升系统的整体性能。前端则用Vue3的框架开发，采用了ant design vue组件库来进行ui界面的构建，通过axios来和后端的数据进行交互。

主要的系统创新点在于装载了火山引擎DeepSeek V3多模态大语言模型，实现了书城的客服模块，能够24小时为用户提供在线书籍咨询服务。而且，管理系统具备了完整的电商功能，有用户登入注册、商品的浏览搜索、购物车的管理、订单的跟踪，评论功能等。经过性能的测试和功能的测试验证，系统稳定，ai客服响应准确，满足了电子购物的基本需求。本管理系统为智能化电商平台提供了实践经验和技术参考。

关键词：智能书城；人工智能；前后端分离；电子商务

**ABSTRACT**

With the continuous development of e-commerce and the rapid rise of artificial intelligence, the sales methods of physical book stores are in need of urgent digital transformation. This paper designs an intelligent store management system based on SpringBoot, uses a modern front- and back-end separation architecture, and integrates the now developed mature generative artificial intelligence to save customer service costs and improve user experience.

The backend of the management system uses the SpringBoot framework to form a restful API service, uses mybatis-plus to design the data persistence layer, and mysql stores important business data and records such as orders, products, users, administrators, etc. Redis serves as a cache layer to improve the overall performance of the system. The front-end is developed using the Vue3 framework, using the ant design vue component library to build the ui interface, and interact with the back-end data through axios.

The main system innovation is that it is equipped with the volcanic engine DeepSeek V3 multi-modal large language model, which realizes the bookstore's customer service module, and can provide users with online book consultation services 24 hours a day. Moreover, the management system has complete e-commerce functions, including user login registration, product browsing and search, shopping cart management, order tracking, commenting functions, etc. After performance testing and functional testing and verification, the system is stable and the AI ​​customer service responds accurately, meeting the basic needs of e-shopping. This management system provides practical experience and technical reference for intelligent e-commerce platforms.

**Key words：**Smart Book City；Artificial Intelligence；Front-end and Back-end Separation；E-commerce

目 录

[1 绪论 1](#_Toc209790816)

[1.1 课题的研究背景 1](#_Toc209790817)

[1.2 国内外发展和现状 1](#_Toc209790818)

[1.2.1 国外研究现状 1](#_Toc209790819)

[1.2.2 国内研究现状 1](#_Toc209790820)

[1.3 系统研究内容概述 1](#_Toc209790821)

[1.4 论文结构 1](#_Toc209790822)

[2 系统相关技术的介绍 2](#_Toc209790823)

[2.1 SpringBoot框架 2](#_Toc209790824)

[2.1.1 SpringBoot简介 2](#_Toc209790825)

[2.1.2 SpringBoot核心特性 2](#_Toc209790826)

[2.1.3 在本系统的应用 2](#_Toc209790827)

[2.2 Vue3前端框架 2](#_Toc209790828)

[2.2.1 Vue3核心特性 2](#_Toc209790829)

[2.2.2 配套技术栈 2](#_Toc209790830)

[2.3 数据库技术 2](#_Toc209790831)

[2.3.1 MySQL数据库 2](#_Toc209790832)

[2.3.2 Redis缓存 2](#_Toc209790833)

[2.4 人工智能技术 2](#_Toc209790834)

[2.4.1 大语言模型技术 2](#_Toc209790835)

[2.4.2 火山引擎DeepSeekV3 2](#_Toc209790836)

[2.4.3 在本系统的应用 2](#_Toc209790837)

[3 系统可行性 3](#_Toc209790838)

[3.1 可行性分析的目的 3](#_Toc209790839)

[3.2 经济可行性分析 3](#_Toc209790840)

[3.3 操作可行性分析 3](#_Toc209790841)

[3.4 技术可行性分析 3](#_Toc209790842)

[4 系统需求分析 4](#_Toc209790843)

[4.1 系统用户需求分析 4](#_Toc209790844)

[4.2 系统功能需求分析 4](#_Toc209790845)

[4.3 数据流分析 4](#_Toc209790846)

[5 系统设计 5](#_Toc209790847)

[5.1 系统总体机构设计 5](#_Toc209790848)

[5.1.1 机构设计原则 5](#_Toc209790849)

[5.1.2 系统架构图 5](#_Toc209790850)

[5.1.2 技术架构栈 5](#_Toc209790851)

[5.2 技术选型 5](#_Toc209790852)

[5.2.1 后端技术选型 5](#_Toc209790853)

[5.2.2 前端技术选型 5](#_Toc209790854)

[5.2.3 数据库技术选型 5](#_Toc209790855)

[5.3 数据库设计 5](#_Toc209790856)

[5.3.1 设计规则 5](#_Toc209790857)

[5.3.2 表结构设计 5](#_Toc209790858)

[5.4 系统接口设计 5](#_Toc209790859)

[5.4.1 RESTful API设计规范 5](#_Toc209790860)

[5.4.2 核心API接口设计 5](#_Toc209790861)

[5.4.3 数据传输格式设计 5](#_Toc209790862)

[6 系统实现 6](#_Toc209790863)

[6.1 数据库连接的实现 6](#_Toc209790864)

[6.2 前台功能模块实现 6](#_Toc209790865)

[6.2.1 注册登录模块 6](#_Toc209790866)

[6.2.2 首页模块 6](#_Toc209790867)

[6.2.3 详情页模块 6](#_Toc209790868)

[6.2.4 用户中心模块 6](#_Toc209790869)

[6.2.5 人工智能客服模块 6](#_Toc209790870)

[6.3后台管理模块实现 6](#_Toc209790871)

[6.3.1 用户管理 6](#_Toc209790872)

[6.3.2 分类管理 6](#_Toc209790873)

[6.3.3 标签管理 6](#_Toc209790874)

[6.3.4 商品管理 6](#_Toc209790875)

[6.3.5 订单管理 6](#_Toc209790876)

[6.3.6 日志管理 6](#_Toc209790877)

[6.3.7 运营管理 6](#_Toc209790878)

[6.3.8 系统信息 6](#_Toc209790879)

[6.3.9 总览模块 6](#_Toc209790880)

[6.4系统测试 6](#_Toc209790881)

[6.4.1测试目的 6](#_Toc209790882)

[6.4.2后台测试用例 7](#_Toc209790883)

[6.4.2前台测试用例 7](#_Toc209790884)

[6.4.3 测试结论 9](#_Toc209790885)

[7 结论 10](#_Toc209790886)

[参考文献 11](#_Toc209790887)

[致谢 12](#_Toc209790888)

# 1 绪论

## 课题的研究背景

随着互联网技术的快速发展和人工智能技术的日益成熟，电子商务行业正经历着前所未有的变革。传统的电商系统已经无法满足现代消费者对个性化、智能化服务的需求。消费者对于商品推荐的准确性、客服响应的及时性、购物体验的便捷性提出了更高的要求。

在这种背景下，构建一个集成人工智能技术的智能商城管理系统具有重要的现实意义：

1. 技术创新意义：通过将大语言模型等前沿AI技术与传统电商系统相结合，探索AI技术在垂直电商领域的应用模式，为行业发展提供技术参考。
2. 商业价值意义：智能化的客服系统可以显著降低人工成本，提高服务效率；个性化推荐算法能够提升用户购买转化率；自动化的数据分析为经营决策提供科学依据。
3. 用户体验意义：24小时不间断的智能客服服务、精准的商品推荐、便捷的购物流程，能够显著提升用户的购物体验和满意度。
4. 学术研究意义：该系统的设计与实现为高校电子商务、软件工程等相关专业的教学实践提供了完整的案例参考。

## 1.2 国内外发展和现状

### 1.2.1 国外研究现状

在国外，电子商务系统的发展已经相对成熟，许多知名企业都构建了先进的电商平台：

1. Amazon：作为全球最大的电商平台之一，Amazon在个性化推荐、智能客服、供应链管理等方面都处于行业领先地位。其推荐系统基于协同过滤和深度学习算法，能够精准预测用户需求。
2. eBay：在拍卖和二手商品交易方面具有优势，其智能定价系统和欺诈检测技术值得借鉴。
3. Shopify：提供完整的电商解决方案，其模块化的架构设计和丰富的API接口为第三方开发者提供了良好的扩展性。

### 1.2.2 国内研究现状

国内电商行业起步较晚，但发展迅速，涌现出一批优秀的电商平台：

1. 阿里巴巴：淘宝、天猫等平台在移动支付、物流配送、数据分析等方面具有显著优势。其千人千面的个性化推荐技术和阿里小蜜智能客服系统代表了国内的先进水平。
2. 京东：在自营电商模式、供应链管理、无人配送等领域进行了深入探索，其智能客服和智能仓储系统具有较高的技术含量。
3. 拼多多：通过社交电商模式创新，在下沉市场获得成功，其基于社交关系的推荐算法和团购机制具有独特性

## 系统研究内容概述

本系统旨在构建一个基于 Spring Boot 技术栈的智能化商城管理平台，通过整合商品管理、订单处理、库存调度、用户行为分析与智能推荐等功能，为中小型电商业务提供一体化、可扩展、低维护成本的后台支撑。系统从业务流程规范化、运营效率提升、数据价值挖掘与用户体验优化四个方面入手，解决传统分散式、低自动化、缺乏智能分析的商城管理方式的痛点。

## 1.4 论文结构

第1章 绪论：介绍研究背景、意义、现状和目标。

第2章 相关技术研究：分析系统涉及的关键技术原理和特点。

第3章 系统需求分析：详细分析系统的功能需求和非功能需求。

第4章 系统总体设计：描述系统的整体架构和设计思路。

第5章 数据库设计：设计系统的数据模型和数据库结构。

第6章 系统详细设计与实现：详细介绍各功能模块的设计和实现。

第7章 系统测试：描述测试策略、测试用例和测试结果。

第8章 总结与展望：总结研究成果，展望未来发展方向。

# 2 系统相关技术的介绍

## 2.1 SpringBoot框架

### 2.1.1 SpringBoot简介

SpringBoot是由Pivotal团队提供的全新框架，其设计目的是简化Spring应用的初始搭建以及开发过程。SpringBoot采用"约定优于配置"的原则，通过自动配置功能，极大地简化了Spring应用的开发复杂度。

### SpringBoot核心特性

(1)自动配置：根据项目依赖自动配置Spring应用

(2)起步依赖：通过starter依赖简化Maven配置

(3)内嵌服务器：支持内嵌Tomcat、Jetty等服务器

(4)生产就绪特性：提供健康检查、指标监控等功能

(5)无代码生成：不需要生成代码和XML配置

### 2.1.3 在本系统的应用

本系统使用SpringBoot作为后端开发框架，主要应用了以下特性：

(1)使用@RestController构建RESTful API

(2)集成MyBatis-Plus进行数据访问

(3)使用Spring Security进行安全控制

(4)利用Spring Cache整合Redis缓存

## 2.2 Vue3前端框架

Vue3是Vue.js的最新版本，采用了全新的Composition API设计，提供了更好的TypeScript支持，并在性能方面有显著提升。Vue3采用了Proxy作为响应式系统的基础，解决了Vue2中的一些限制。

### 2.2.1 Vue3核心特性

(1)Composition API：提供更灵活的逻辑复用方式

(2)性能提升：更快的渲染速度和更小的包体积

(3)更好的TypeScript支持：原生TypeScript支持

(4)多根节点组件：Fragment支持

(5)Teleport：组件可以渲染到DOM的任何位置

### 2.2.2 配套技术栈

(1)Vue Router 4：用于前端路由管理

(2)Pinia：状态管理库，替代Vuex

(3)Ant Design Vue：UI组件库

(4)Axios：HTTP客户端库

(5)Vite：构建工具

## 2.3 数据库技术

### 2.3.1 MySQL数据库

MySQL是世界上最流行的开源关系型数据库管理系统，具有以下特点：

高性能：支持大数据量的快速访问

高可靠性：提供完善的数据备份和恢复机制

易于使用：简单的安装和管理

开源免费：降低了开发成本

跨平台：支持多种操作系统

### 2.3.2 Redis缓存

Redis是一个开源的内存数据结构存储系统，可以用作数据库、缓存和消息代理：

高性能：基于内存存储，读写速度极快

丰富的数据类型：支持字符串、哈希、列表、集合等

持久化：支持RDB和AOF两种持久化方式

高可用：支持主从复制和哨兵模式

分布式：支持集群模式

## 2.4 人工智能技术

### 2.4.1 大语言模型技术

大语言模型是基于Transformer架构的深度学习模型，通过大规模文本数据训练，具备强大的自然语言理解和生成能力。

### 2.4.2 火山引擎DeepSeekV3

火山引擎DeepSeek V3是字节跳动推出的大语言模型，具有以下特点：

强大的理解能力：能够理解复杂的自然语言问题

流畅的对话能力：支持多轮对话和上下文理解

丰富的知识储备：涵盖多个领域的知识

快速的响应速度：适合实时对话应用

API接口：提供简单易用的API接口

### 2.4.3 在本系统的应用

本系统通过调用火山引擎DeepSeek V3 API实现智能客服功能：

@Service

public class AiChatServiceImpl implements AiChatService {

private static final String AI\_API\_URL

= "https://ark.cn-beijing.volces.com/api/v3/chat/completions";

@Override

public String processAiChat(String userMessage, String userId) {

Map<String, Object> requestBody = buildAiRequest(userMessage, userId);

ResponseEntity<Map> response = restTemplate.postForEntity(AI\_API\_URL, request, Map.class);

return extractAiResponse(response);

}

}

# 系统可行性

## 3.1 可行性分析的目的

可行性分析是系统开发前期的关键环节，旨在综合评估项目的必要性与可实施性：一方面通过分析市场环境和用户需求，判断是否值得立项（如在电商迅速发展与AI技术成熟的背景下，传统电商已难满足个性化与智能化需求，开发集成AI的智能商城管理系统具有明确必要性）；另一方面从经济、技术、运营等维度论证资源是否充足、技术难点能否攻克、组织环境是否支持，同时系统识别潜在风险与问题，为规划与风险管理提供依据；在此基础上为投资决策、资源配置与实施方案提供科学参考，并进一步明确项目目标、范围与约束条件，为后续需求分析与系统设计奠定基础。

## 3.2 经济可行性分析

通过引入智能客服系统替代人工客服，每年可节约约12万元的人力成本（相当于2名客服人员）；借助订单处理与库存管理等自动化功能，整体运营效率有望提升20%以上，进而间接创造可观的经济价值；同时，系统在运营过程中沉淀的用户与销售数据将成为重要的数据资产，为精准营销与业务决策提供持续支持。综合来看，虽然前期投入较大，但智能化技术的应用能够显著降低运营成本、提升服务质量并带来良好的经济回报，且随着用户规模扩大与业务发展，项目的经济效益将愈发凸显。

## 3.3 操作可行性分析

面向18–45岁、已熟悉网购的普通消费者（图书爱好者、学生、教师、专业人士等），目标用户整体数字素养较好，接受度高。系统基于Vue3提供简洁直观的界面与3–4步完成的购物流程，配合AI智能客服与PC/移动多端适配，显著降低上手门槛；普通用户几乎无需培训即可使用，管理员经过1–2天培训即可掌握商品管理、订单处理与数据分析等功能，并配备完善的在线帮助文档与视频教程。 在组织适应性上，系统的商品、订单、库存与数据统计功能契合图书零售的标准业务流程，支持灵活定价与促销策略以适应多样化营销需求；人员配置上，1–2名系统管理员、1名技术维护人员即可支撑运营，AI客服可处理80%以上常见问题，仅需1名人工客服应对复杂场景。引入系统后将推动线上线下融合，客服工作重心转向复杂与个性化服务，管理决策也将更多依托数据分析而非经验判断。

面向18–45岁、已熟悉网购的普通消费者（图书爱好者、学生、教师、专业人士等），目标用户整体数字素养较好，接受度高。系统基于Vue3提供简洁直观的界面与3–4步完成的购物流程，配合AI智能客服与PC/移动多端适配，显著降低上手门槛；普通用户几乎无需培训即可使用，管理员经过1–2天培训即可掌握商品管理、订单处理与数据分析等功能，并配备完善的在线帮助文档与视频教程。

在组织适应性上，系统的商品、订单、库存与数据统计功能契合图书零售的标准业务流程，支持灵活定价与促销策略以适应多样化营销需求；人员配置上，1–2名系统管理员、1名技术维护人员即可支撑运营，AI客服可处理80%以上常见问题，仅需1名人工客服应对复杂场景。引入系统后将推动线上线下融合，客服工作重心转向复杂与个性化服务，管理决策也将更多依托数据分析而非经验判断。

## 3.4 技术可行性分析

1. 后端技术

SpringBoot框架：成熟稳定的Java开发框架，拥有丰富的生态系统和庞大的开发者社区。开发团队对SpringBoot技术栈有深入的了解和实践经验，技术风险低。

MyBatis-Plus：优秀的持久层框架，简化数据库操作，提高开发效率。文档完善，社区活跃，技术支持充分。

MySQL数据库：开源的关系型数据库，性能稳定，可靠性高，完全能够满足系统的数据存储需求。

Redis缓存：成熟的缓存解决方案，可以显著提升系统性能，减轻数据库压力。

2. 前端技术

Vue3框架：最新的渐进式JavaScript框架，性能优异，开发效率高。开发团队对Vue生态系统（Vue Router、Vuex/Pinia、Element Plus等）有丰富的实践经验。

响应式设计：采用现代CSS技术和UI框架，实现多端适配，技术实现难度适中。

前后端分离：通过RESTful API进行数据交互，架构清晰，便于开发和维护。

3. AI技术

DeepSeek V3 API：国内领先的大语言模型服务，提供稳定的API接口，文档详细，集成简单。

智能推荐算：采用协同过滤和基于内容的推荐算法，技术成熟，实现难度适中。

情感分：基于自然语言处理技术，可以采用开源的情感词典或机器学习模型实现。

# 4 系统需求分析

## 4.1 系统用户需求分析

### 4.1.1 用户痛点分析

通过对图书电商行业的调研和用户访谈，我们识别出以下主要痛点：

痛点1：信息过载，难以找到合适的图书

问题描述：图书市场品类繁多，用户面临海量的图书信息，往往不知道该选择哪本书。传统电商平台的搜索功能只能按关键词匹配，无法理解用户的真实需求和阅读兴趣。用户需要花费大量时间浏览和筛选，购物体验不佳。

解决方案：本系统引入AI智能推荐系统，基于用户的浏览历史、购买记录、评价偏好等多维度数据，运用协同过滤算法和深度学习技术，为用户提供个性化的图书推荐。系统能够理解用户的阅读喜好，主动推荐可能感兴趣的图书，大大提高了用户发现好书的效率。同时，智能客服可以通过对话了解用户需求，直接推荐合适的图书，实现精准匹配。

痛点2：客服响应慢，咨询问题得不到及时解答

问题描述：用户在购买图书前通常有很多疑问，如图书内容介绍、适用人群、版本差异、配送时效等。传统电商平台依赖人工客服，存在工作时间限制（通常只在工作日的8小时内服务）、响应速度慢（高峰期需要排队等待）、服务质量不稳定（依赖客服人员的专业水平和服务态度）等问题。

解决方案：本系统集成基于DeepSeek V3大语言模型的AI智能客服，提供7×24小时全天候在线服务。智能客服具备强大的自然语言理解能力，能够准确理解用户问题，从图书信息库和FAQ知识库中快速检索答案，实时响应用户咨询。对于复杂问题，智能客服可以引导用户逐步明确需求，或转接人工客服处理。经测试，智能客服可以处理80%以上的常见问题，平均响应时间不超过3秒，大大提升了服务效率和用户满意度。

痛点3：购买决策困难，缺乏可靠的参考信息\*\*

问题描述：用户在选购图书时，往往需要参考其他读者的评价。然而，传统电商平台的评价系统存在诸多问题：评价数量少、评价内容质量参差不齐（大量简单的"好评"、"不错"等无效信息）、虚假评价难以识别、评价情感倾向不明确等。用户难以从海量评价中提取有价值的信息，无法做出准确的购买决策。

解决方案：本系统提供智能评价分析功能，运用自然语言处理和情感分析技术，自动分析用户评价的情感倾向（正面、中性、负面），提取评价中的关键信息（如图书内容质量、印刷质量、物流服务等维度），并生成可视化的评价摘要。用户可以快速了解一本书的整体口碑和具体优缺点，做出更明智的购买决策。同时，系统会识别异常评价模式，帮助过滤虚假评价，提高评价的可信度。。

### 4.1.2 用户需求总结

基于以上痛点分析，智能商城管理系统的核心用户需求可归纳为：

1. 智能化的商品发现：通过AI推荐算法帮助用户快速找到感兴趣的图书，降低信息搜索成本。

2. 即时化的咨询服务：提供7×24小时智能客服，随时解答用户疑问，提升购物体验。

3. 可靠化的决策支持：通过智能评价分析，为用户提供客观、全面的商品信息，支持购买决策。

4. 便捷化的购物流程：简化从浏览、选购到支付、收货的全流程，提高购物效率。

5. 个性化的用户体验：基于用户行为数据，提供个性化的界面、推荐和服务。 一段

## 4.2 系统功能需求分析

### 4.2.1 用户功能需求

1. 用户注册与登录

用户可以通过邮箱或手机号注册账号

支持邮箱和手机号登录

提供密码找回功能

支持第三方登录（预留接口）

2. 商品浏览与搜索

用户可以浏览商品列表，支持分页显示

提供多种商品分类浏览方式

支持关键词搜索商品

提供高级搜索功能（价格区间、分类筛选等）

支持商品排序（价格、销量、评分等）

3. 购物车功能

用户可以将商品添加到购物车

支持购物车商品数量修改

支持购物车商品删除

提供购物车商品总价计算

4. 订单管理

用户可以提交订单

支持多种支付方式（支付宝、微信支付等）

用户可以查看订单状态

支持订单取消功能

提供订单历史记录查询

5. 评价系统

用户可以对已购买商品进行评价

支持评分和文字评价

用户可以查看其他用户的评价

提供评价的情感分析展示

6. 个人中心

用户可以修改个人信息

支持收货地址管理

提供收藏夹功能

显示购买历史和推荐商品

7. AI智能客服

提供24小时智能问答服务

支持商品咨询和购买建议

具备上下文理解能力

能够推荐相关商品

支持多轮对话

### 4.2.2 管理员功能需求

1. 用户管理

查看用户列表和详细信息

支持用户状态管理（启用/禁用）

用户行为统计分析

用户权限管理

2. 商品管理

商品信息的增删改查

商品分类管理

库存管理和预警

商品上架/下架管理

批量导入导出功能

3. 订单管理

订单列表查看和搜索

订单状态管理

订单统计分析

退款处理

4. 系统管理

系统配置管理

数据备份与恢复

日志管理

性能监控

5. 数据统计

销售数据统计

用户活跃度统计

商品热度分析

收入统计分析。

## 4.3 数据流分析

通过对本系统的需求分析，系统的基本功能已经大体确定，需要面向数据流的分析将数据分解化简问题，将物理与逻辑表示分开，对系统进行数据与逻辑的抽象[20]。

此小节给出系统的数据流程，系统的数据流图，如图4-1所示。然后再根据数据流图展开介绍系统的数据流。

图4-1系统的数据流图

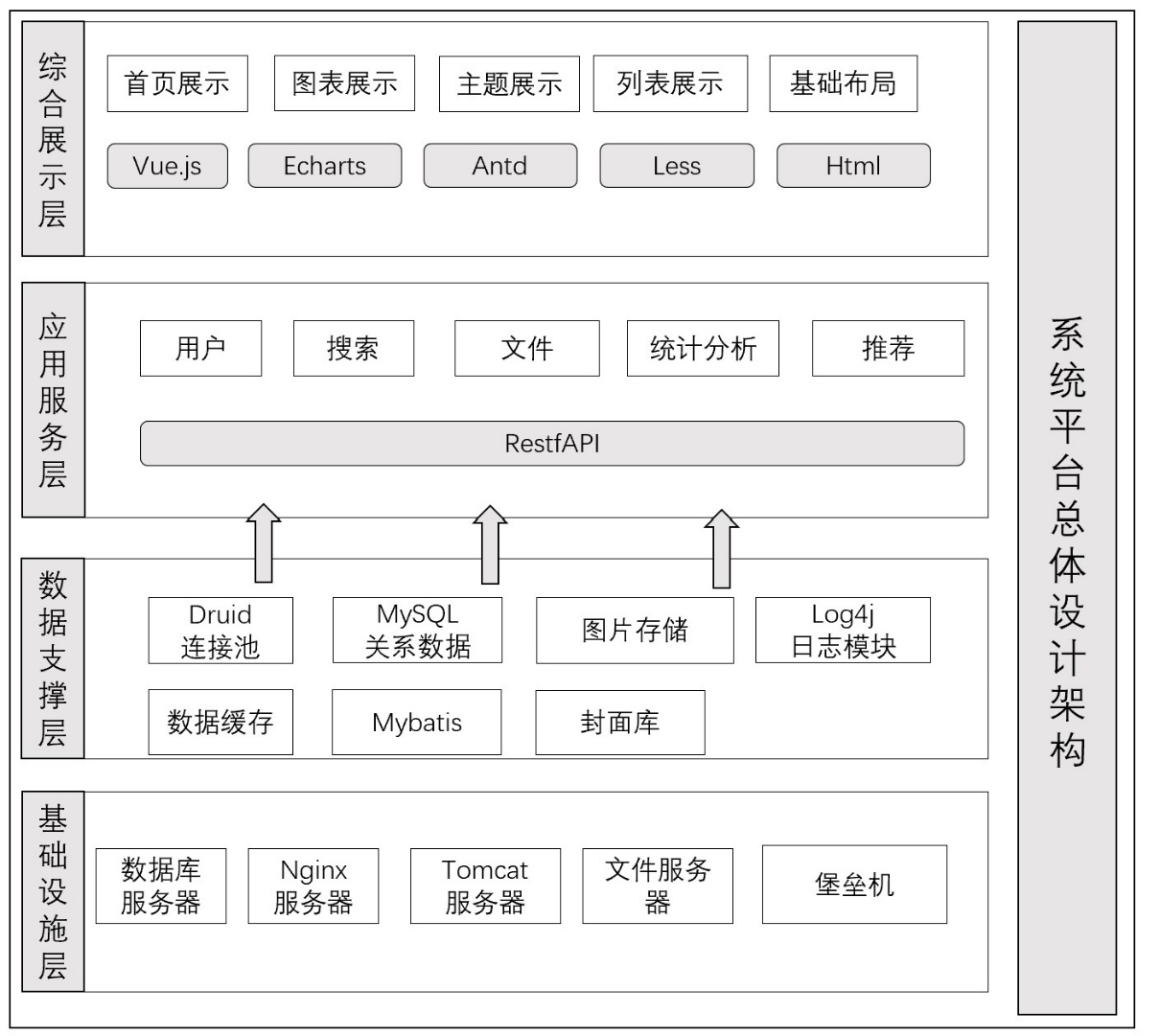
# 5 系统设计

## 5.1 系统总体机构设计

### 5.1.1 机构设计原则

本系统的设计遵循以下原则：首先，采用阶段开发原则，即在对系统整体框架和数据结构进行全面设计的基础上，将具体功能的实现划分为若干阶段，逐步推进开发过程。其次，强调易用性原则，在设计过程中充分考虑最终用户的特点和操作习惯，通过自动化处理部分业务流程，最大程度减轻用户操作负担。系统还坚持业务完整性原则，能够对业务流程中的特殊情况及时、准确响应，确保业务数据的完整性。此外，系统建设过程中同步制定了较为完善的业务规范，体现了业务规范化原则，为实际操作提供了有力的规范和支持。最后，系统设计贯彻可扩展性原则，各功能模块耦合度低，便于后续扩展和维护；同时对可能存在的旧有数据库系统做好兼容性考虑，以满足未来业务发展的需求。

### 5.1.2 系统架构图



### 5.1.2 技术架构栈

前端技术栈：Vue 3.2.45、Vue Router 4.1.6、Pinia 2.0.28、Ant Design Vue 3.2.20、Vite 4.0、Axios 1.2.2、TypeScript 4.9

后端技术栈：SpringBoot 2.5.5、Spring Security 5.5.2、MyBatis-Plus 3.5.2、Redis 6.0、Maven 3.9、Jackson 2.12、Lombok 1.18

数据库技术栈：MySQL 8.0 带 InnoDB 引擎，Redis 6.0 用于缓存

AI服务技术栈：通过REST API的DeepSeek V3大语言模型

## 5.2 技术选型

### 5.2.1 后端技术选型

SpringBoot 2.5.5：开发快，生态丰富，部署方便

Spring Security 5.5.2：完整的身份验证和授权

MyBatis-Plus 3.5.2：CRUD 自动化、代码生成、分页支持

Redis 6.0：高性能缓存、分布式锁定

Maven 3.9：依赖管理，构建自动化

### 5.2.2 前端技术选型

Vue 3.2.45：响应式系统、Composition API、TypeScript 支持

Vue Router 4.1.6：路由、延迟加载、导航防护

Pinia 2.0.28：现代状态管理，TypeScript 友好

Ant Design Vue 3.2.20：具有 60+ 组件的企业级 UI

Vite 4.0：快速开发服务器、HMR、优化构建

Axios 1.2.2：带有拦截器的基于 Promise 的 HTTP 客户端

### 5.2.3 数据库技术选型

MySQL 8.0：成熟、高性能、ACID 事务、JSON 支持

InnoDB 存储引擎：事务安全、行级锁定、MVCC

Redis 6.0：会话管理、数据缓存、计数器、分布式锁定

数据库设计原则：规范化、索引、分区、备份策略

## 5.3 数据库设计

### 5.3.1 设计规则

数据库表命名采用多段式命名，各段之间用下划线分隔，如b\_xx（一律小写）。其中：第一段表示模块或者主题，如图书模块为b；第二段表示此表的实体类型，如文件表为file。

字段名规则：列名采用多段式命名，各单词间用下划线分隔，如xx\_xx（一律小写）。应尽可能地采用简洁明了的列名以准确描述列的内容含义,根据需要可以一个单词或者多个单词进行命名。

### 5.3.2 表结构设计

用户信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 字段类型 | 长度 | 是否为空 | 是否为主键 | 默认值 | 备注 |
| id | bigint |  | NO | Y |  |  |
| username | varchar | 50 | YES |  |  | 用户名 |
| password | varchar | 50 | YES |  |  | 密码 |
| role | varchar | 2 | YES |  |  | 角色 |
| status | varchar | 1 | NO |  |  |  |
| nickname | varchar | 20 | YES |  |  | 昵称 |
| avatar | varchar | 100 | YES |  |  |  |
| mobile | varchar | 13 | YES |  |  |  |
| email | varchar | 50 | YES |  |  |  |
| gender | varchar | 1 | YES |  |  |  |
| description | longtext | 4.29E+09 | YES |  |  |  |
| create\_time | varchar | 30 | YES |  |  |  |
| score | int |  | YES |  | 0 |  |
| push\_email | varchar | 40 | YES |  |  | 推送邮箱 |
| push\_switch | tinyint |  | YES |  | 0 | 推送开关 |
| token | varchar | 32 | YES |  |  |  |
| 注1：用户角色role分为普通用户和管理员。  注2：用户状态, 1-停用，0-正常 | | | | | | |

分类表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 字段类型 | 长度 | 是否为空 | 是否为主键 | 默认值 | 备注 |
| id | bigint |  | NO | Y |  |  |
| title | varchar | 100 | YES |  |  | 标题 |
| create\_time | varchar | 30 | NO |  |  |  |

标签表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 字段类型 | 长度 | 是否为空 | 是否为主键 | 默认值 | 备注 |
| id | bigint |  | NO | Y |  |  |
| title | varchar | 100 | YES |  |  |  |

商品信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 字段类型 | 长度 | 是否为空 | 是否为主键 | 默认值 | 备注 |
| id | bigint |  | NO | Y |  |  |
| title | varchar | 100 | YES |  |  | 名称 |
| cover | varchar | 100 | YES |  |  | 封面 |
| description | longtext | 4.29E+09 | YES |  |  | 描述 |
| price | varchar | 50 | YES |  |  | 价格 |
| status | varchar | 1 | NO |  |  |  |
| repertory | int |  | NO |  |  | 库存 |
| score | int |  | YES |  | 0 | 分数 |
| create\_time | varchar | 30 | YES |  |  |  |
| pv | int |  | YES |  | 0 |  |
| recommend\_count | int |  | YES |  | 0 | 推荐数 |
| wish\_count | int |  | YES |  | 0 |  |
| collect\_count | int |  | YES |  | 0 |  |
| classification\_id | bigint |  | YES |  |  |  |

订单表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 字段类型 | 长度 | 是否为空 | 是否为主键 | 默认值 | 备注 |
| id | bigint |  | NO | Y |  |  |
| status | varchar | 2 | YES |  |  |  |
| order\_time | varchar | 30 | YES |  |  |  |
| pay\_time | varchar | 30 | YES |  |  |  |
| thing\_id | bigint |  | YES |  |  |  |
| user\_id | bigint |  | YES |  |  |  |
| count | int |  | NO |  | 0 |  |
| order\_number | varchar | 13 | YES |  |  | 订单编号 |
| receiver\_address | varchar | 50 | YES |  |  |  |
| receiver\_name | varchar | 20 | YES |  |  |  |
| receiver\_phone | varchar | 20 | YES |  |  |  |
| remark | varchar | 30 | YES |  |  |  |

地址表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 字段类型 | 长度 | 是否为空 | 是否为主键 | 默认值 | 备注 |
| id | bigint |  | NO | Y |  |  |
| name | varchar | 100 | YES |  |  | 姓名 |
| mobile | varchar | 30 | YES |  |  | 手机号 |
| description | varchar | 200 | YES |  |  | 地址 |
| def | varchar | 10 | YES |  |  | 默认 |
| create\_time | varchar | 30 | YES |  |  |  |
| user\_id | bigint |  | YES |  |  | 用户id |

评论表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 字段类型 | 长度 | 是否为空 | 是否为主键 | 默认值 | 备注 |
| id | bigint |  | NO | Y |  |  |
| content | varchar | 200 | YES |  |  | 评论内容 |
| comment\_time | varchar | 30 | YES |  |  | 评论时间 |
| like\_count | int |  | NO |  | 0 |  |
| user\_id | bigint |  | YES |  |  |  |
| thing\_id | bigint |  | YES |  |  |  |

消息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 字段类型 | 长度 | 是否为空 | 是否为主键 | 默认值 | 备注 |
| id | bigint |  | NO | Y |  |  |
| title | varchar | 100 | YES |  |  |  |
| content | varchar | 1000 | YES |  |  |  |
| create\_time | varchar | 30 | YES |  |  |  |

日志表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 字段类型 | 长度 | 是否为空 | 是否为主键 | 默认值 | 备注 |
| id | bigint |  | NO | Y |  |  |
| re\_ip | varchar | 100 | YES |  |  |  |
| re\_time | varchar | 30 | YES |  |  |  |
| re\_ua | varchar | 255 | YES |  |  |  |
| re\_url | varchar | 200 | YES |  |  |  |
| re\_method | varchar | 10 | YES |  |  |  |
| re\_content | varchar | 200 | YES |  |  |  |
| access\_time | varchar | 30 | YES |  |  |  |

## 5.4 系统接口设计

### 5.4.1 RESTful API设计规范

RESTful设计：遵循REST架构风格，使用标准HTTP方法

统一响应格式：所有接口返回统一的JSON格式

错误处理标准化：提供清晰的错误码和错误信息

版本控制：支持API版本管理，确保向后兼容

安全认证：使用JWT令牌进行身份验证

### 5.4.2 核心API接口设计

**用户相关接口**

| **方法** | **路径** | **描述** | **参数** |
| --- | --- | --- | --- |
| POST | /api/user/register | 用户注册 | UserRegisterDTO |
| POST | /api/user/login | 用户登录 | UserLoginDTO |
| GET | /api/user/profile | 获取用户信息 | Authorization Header |
| PUT | /api/user/profile | 更新用户信息 | UserUpdateDTO |
| POST | /api/user/logout | 用户退出 | Authorization Header |

**商品相关接口**

| **方法** | **路径** | **描述** | **参数** |
| --- | --- | --- | --- |
| GET | /api/thing/list | 商品列表 | keyword, sort, c, tag, page, size |
| GET | /api/thing/{id} | 商品详情 | id |
| POST | /api/thing/create | 创建商品 | Thing对象 |
| PUT | /api/thing/{id} | 更新商品 | id, Thing对象 |
| DELETE | /api/thing/{id} | 删除商品 | id |

**AI智能服务接口**

| **方法** | **路径** | **描述** | **参数** |
| --- | --- | --- | --- |
| GET | /api/ai-chat/message | AI对话 | message, sessionId |
| GET | /api/ai/recommend | 商品推荐 | userId, limit |
| POST | /api/ai/analyze | 情感分析 | content |
| GET | /api/ai/analytics | 数据分析 | type, dateRange |

# 6 系统实现

## 6.1 数据库连接的实现

本项目采用SpringBoot框架集成MyBatis-Plus实现与MySQL数据库的连接。SpringBoot通过自动配置机制，简化了数据库连接池和ORM框架的配置过程。下面详细介绍系统实现数据库连接的具体步骤和关键代码。代码格式如表6-1所示。

表6-1 MySQL数据库连接配置

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url=jdbc:mysql://localhost:3306/base\_zxsc?characterEncoding=utf-8&serverTimezone=GMT%2B8&useSSL=false&nullCatalogMeansCurrent=true&allowPublicKeyRetrieval=true

spring.datasource.username=root

spring.datasource.password=123456;

## 6.2 前台功能模块实现

### 6.2.1 注册登录模块

本系统的后端开发框架采用的是 SpringBoot，所以使用 SpringBoot 来新建一个Maven 项目，应用 SpringMVC 的分层设计思想进行开发。用户需要通过注册和登录后的操作后，方能进入商城系统，首先用户在注册页面填入自己的基本信息，包括用户名和密码等等。用户注册界面如下图所示：



用户注册成功之后，可以通过登录界面输入用户名和密码进行登录，便可进入到商城平台首页。登录界面如下图所示。用户进入登录界面输入账号和密码后，登录的 url 的请求发给处理器后，对用户输入的用户名和密码进行非空校验，判断账号和密码是否已经填写，并从数据库中询是否有匹配的记录，如果查询成功则登录成功，如果未查询到则提示“用户名或密码错误”，系统将返回到登陆界面继续操作。



登录和注册部分代码如下所示：

表6-1 DB\_Utils.jsp部分代码

@RequestMapping(value = "/userLogin", method = RequestMethod.POST)

public APIResponse userLogin(User user){

user.setPassword(DigestUtils.md5DigestAsHex((user.getPassword() + salt).getBytes()));

User responseUser = userService.getNormalUser(user);

if(responseUser != null) {

return new APIResponse(ResponeCode.SUCCESS, "查询成功", responseUser);

}else {

return new APIResponse(ResponeCode.FAIL, "用户名或密码错误");

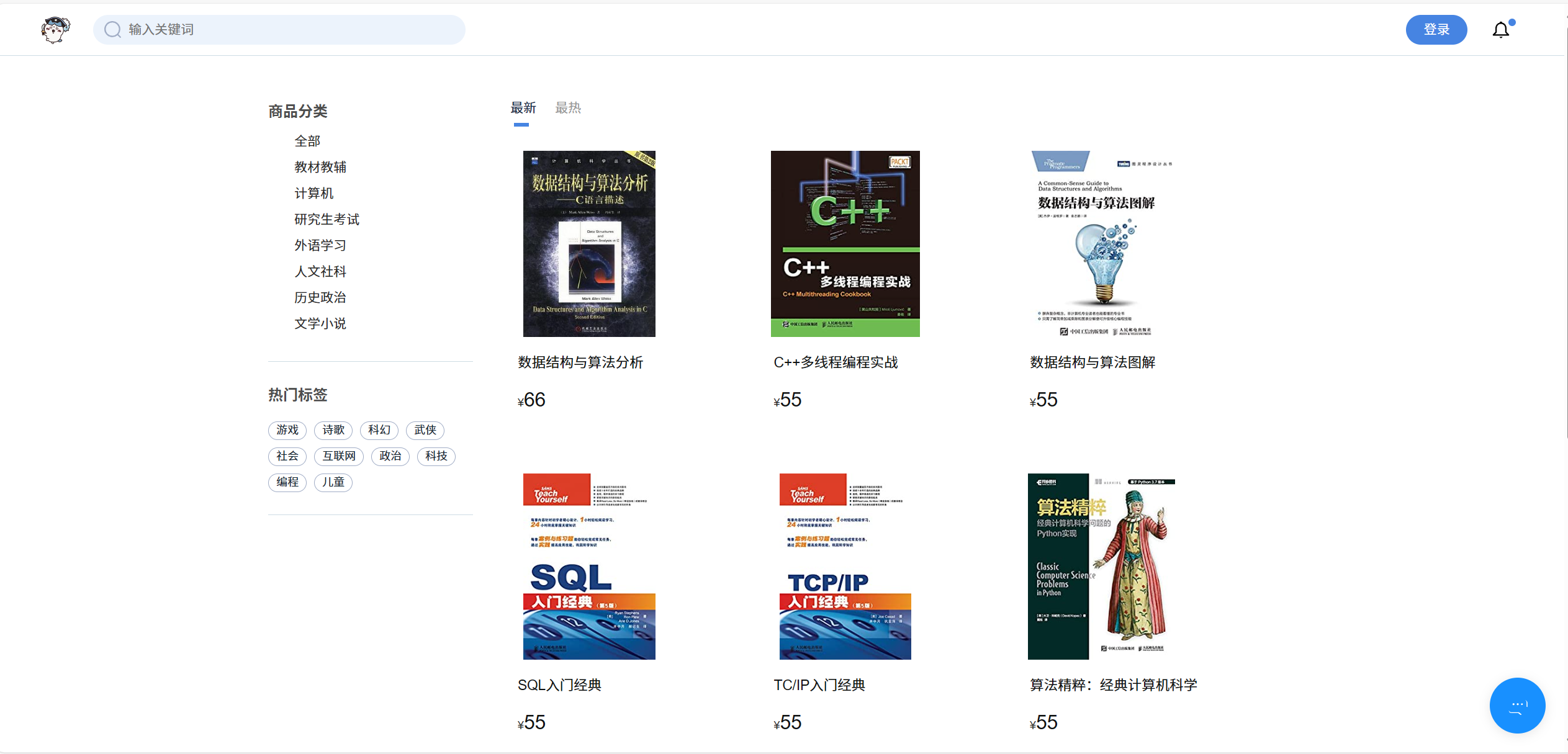
}

}

### 6.2.2 首页模块

用户进入到首页后，即可看到商品信息的列表，整个首页的布局分为三部分，顶部是搜索栏，左侧是分类和标签栏，右侧是列表栏。默认情况下，右侧列表栏会展示全部商品，当用户点击左侧分类栏的某一项时，右侧列表将会筛选出对应的商品。当用户点击左侧某标签时，右侧列表将会筛选出当前标签的所有商品。右侧列表栏顶部还有最新、最热、推荐的tab栏，当点击最新的时候，当前列表将按照时间倒序展示列表，当点击最热的时候，当前列表将按照pv数倒序展示列表，点击推荐的时候，列表将请求推荐接口，按照后端推荐逻辑获取列表。如果用户有搜索的需求，可以在顶部搜索栏输入关键词进行搜索。

首页模块界面如下：



页列表展示的关键代码如下：

表6-1 DB\_Utils.jsp部分代码

@RequestMapping(value = "/list", method = RequestMethod.GET)

public APIResponse list(String keyword, String sort, String c, String tag){

List<Thing> list = service.getThingList(keyword, sort, c, tag);

return new APIResponse(ResponeCode.SUCCESS, "查询成功", list);

}

### 6.2.3 详情页模块

用户点击首页列表某个商品时，系统将会跳转到对应商品的详情页，详情页展示的是商品的所有基本信息，包括名称、价格、上架状态、库存、封面、简介、收藏数、推荐列表等信息。用户通过点击立即购买按钮，可以购买当前商品，然后进入到确认页。页面右侧是交互栏，包括心愿单、收藏、分享等功能。页面下面是简介和评论tab页面，用户可以通过点击简介和评论标签来实现切换。如果用户想要评论当前商品时，可以输入关键词，然后点击发送按钮，系统会调用发送评论的接口，将评论内容发送到后端，发送成功后，系统会给出发送成功的提示。除此之外，页面右侧还有热门推荐列表，该列表运用了后端推荐接口将数据获取到前端展示。

详情页模块界面如下：



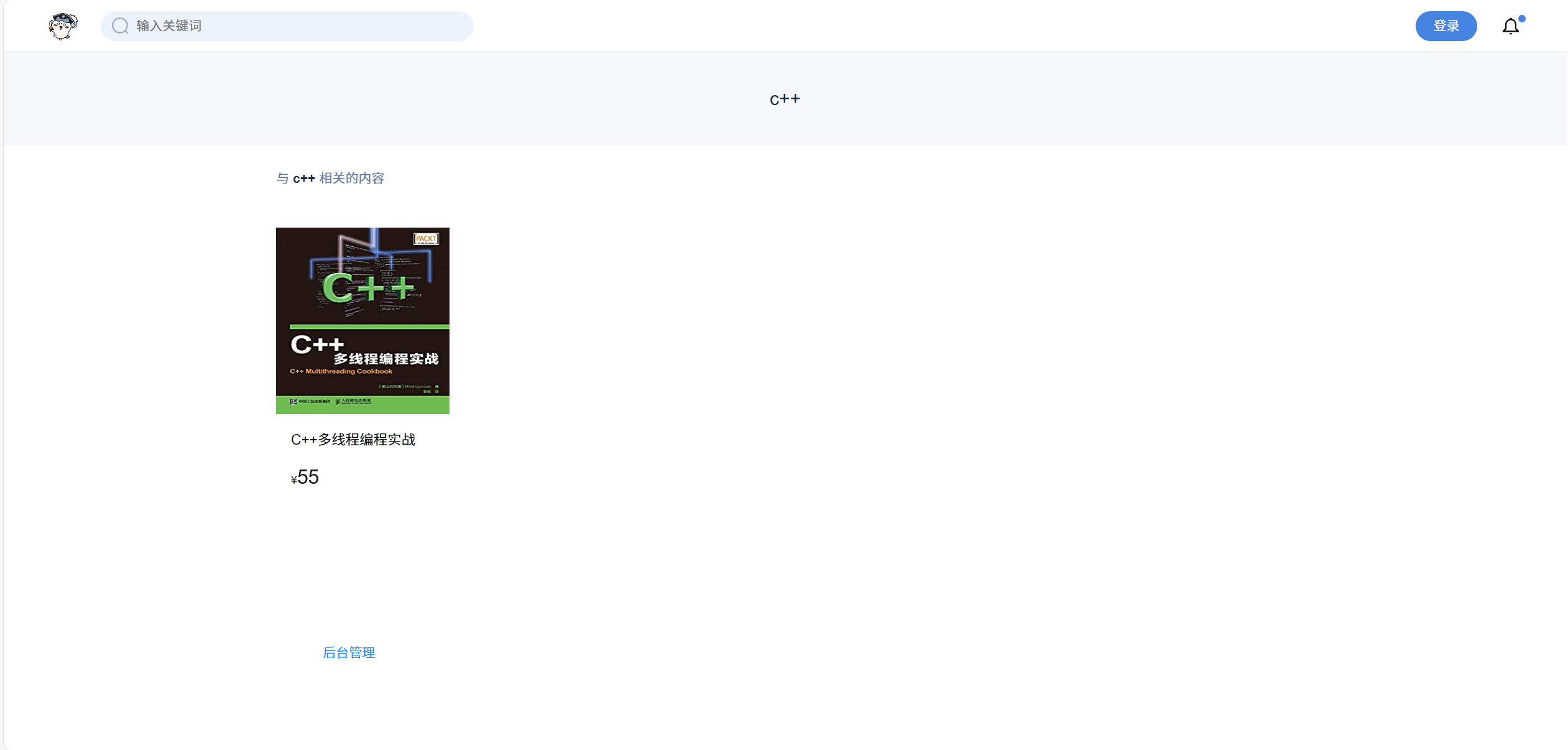
详情页关键代码如下

表6-1 DB\_Utils.jsp部分代码

@RequestMapping(value = "/detail", method = RequestMethod.GET)  
public APIResponse detail(String id){  
 Thing thing = service.getThingById(id);  
  
 return new APIResponse(ResponeCode.SUCCESS, "查询成功", thing);  
}

### 6.2.4 搜索模块

搜索模块实现了搜索功能，在页面顶端的搜索栏，用户输入关键词后，系统会自动跳转到搜索结果页面，在搜索结果页面，系统会调用后端search接口，将关键词内容传给后端，后端通过查询数据库，将列表信息返回给前端，前端页面拿到数据后，在将数据渲染到搜索结果页面。界面如下所示：

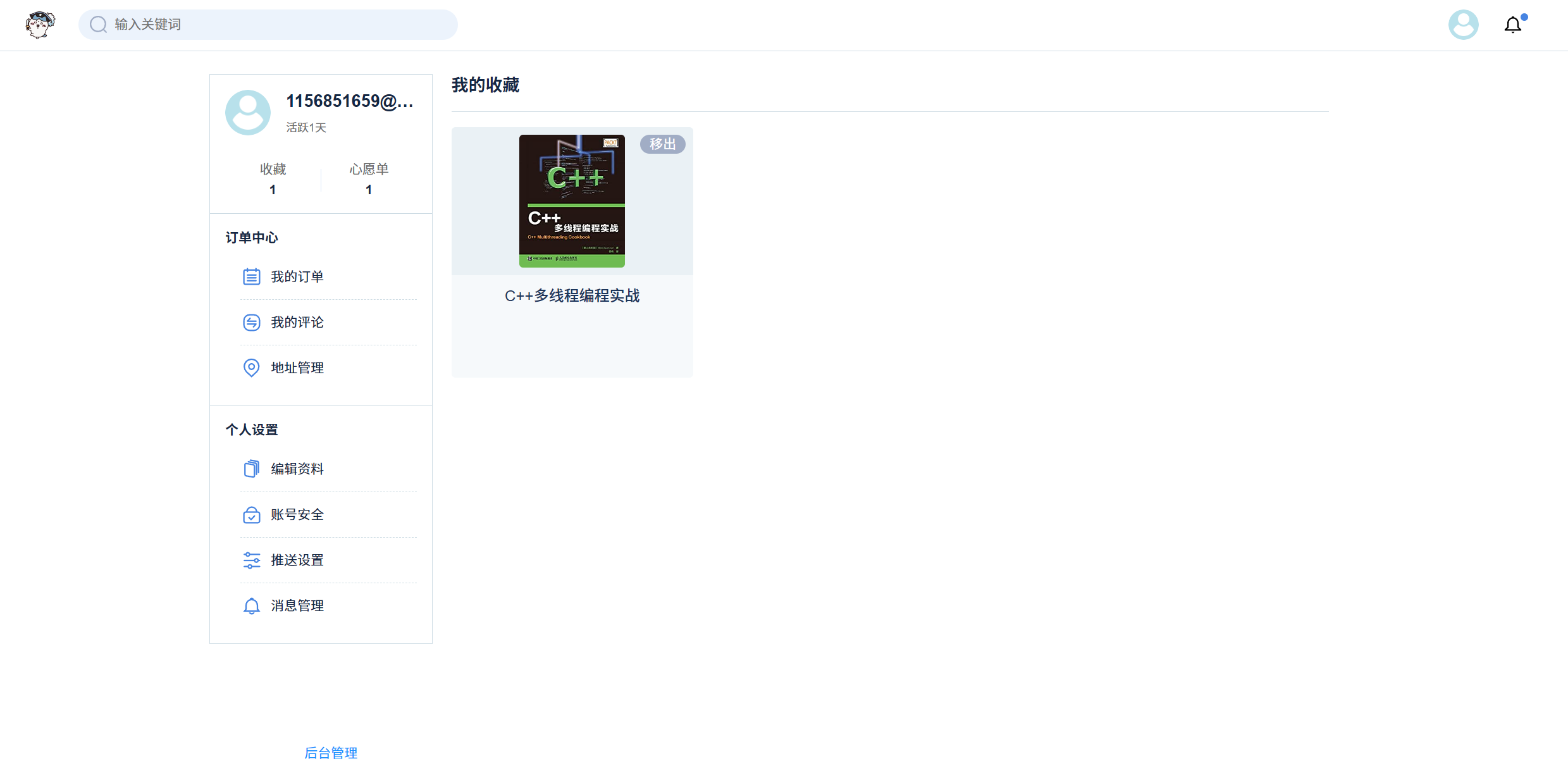


搜索接口关键代码如下：

@RequestMapping(value = "/list", method = RequestMethod.GET)  
public APIResponse list(String keyword, String sort, String c, String tag){  
 List<Thing> list = service.getThingList(keyword, sort, c, tag);  
  
 return new APIResponse(ResponeCode.SUCCESS, "查询成功", list);  
}

### 6.2.5 用户中心模块

点击顶部用户头像后，系统会跳转到用户中心模块。用户中心模块，集成实现了和用户相关的所有信息的展示，包括订单中心、地址管理、收藏、心愿单、个人资料等信息展示。其中，收藏和心愿单实现了收藏列表和心愿单列表的展示，我的订单实现了我的所有订单的展示，包括订单状态、订单物流信息、支付状态的显示。我的评论模块实现了用户对于所有商品的评论列表。地址管理模块实现了当前用户的所有地址列表。编辑资料模块实现了用户基本信息显示，包括头像、昵称、手机号、邮箱、个人简介等。账号安全模块实现了用户密码的修改功能。消息管理模块实现了后台推送消息的显示。整个界面预览如下：



其中订单列表关键代码如下：

@RequestMapping(value = "/userOrderList", method = RequestMethod.GET)

public APIResponse userOrderList(String userId, String status){

List<Order> list = service.getUserOrderList(userId, status);

return new APIResponse(ResponeCode.SUCCESS, "查询成功", list);

}

编辑资料接口关键代码如下：

@Access(level = AccessLevel.ADMIN)  
@RequestMapping(value = "/update", method = RequestMethod.POST)  
@Transactional  
public APIResponse update(User user) throws IOException {  
 // update不能修改密码，故置空  
 user.setPassword(null);  
 String avatar = saveAvatar(user);  
 if(!StringUtils.isEmpty(avatar)) {  
 user.avatar = avatar;  
 }  
 userService.updateUser(user);  
 System.out.println(user);  
 return new APIResponse(ResponeCode.SUCCESS, "更新成功");  
}

### 6.2.6 人工智能客服模块

## 6.3后台管理模块实现

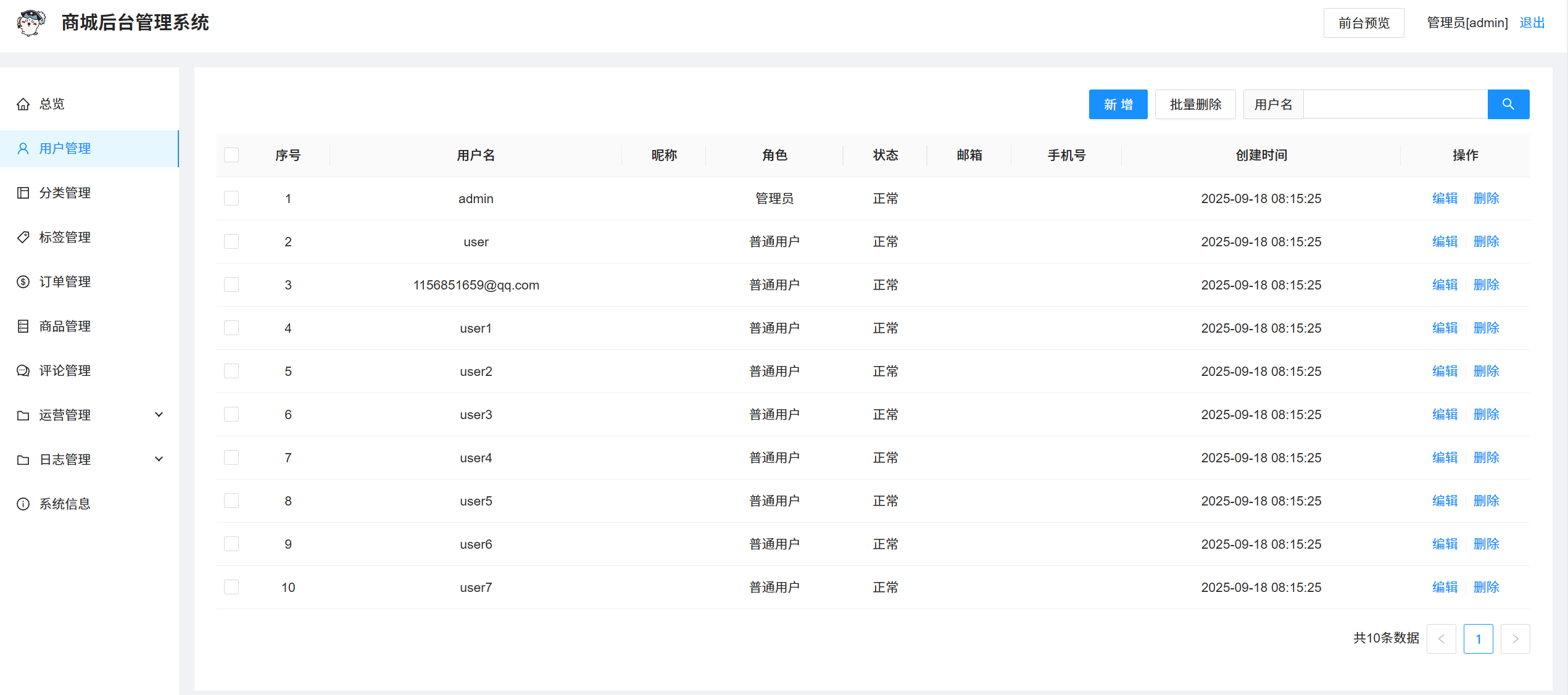
当使用管理员账号登录后台管理系统之后，即可进入后台管理主页。管理员在后台有对各个模块的功能的详细管理信息、并拥有对数据的管理的权限，可以对数据进行输入、现有数据进行修改和输出、添加新数据、管理权限等等。

界面如下所示：

### 6.3.1 用户管理

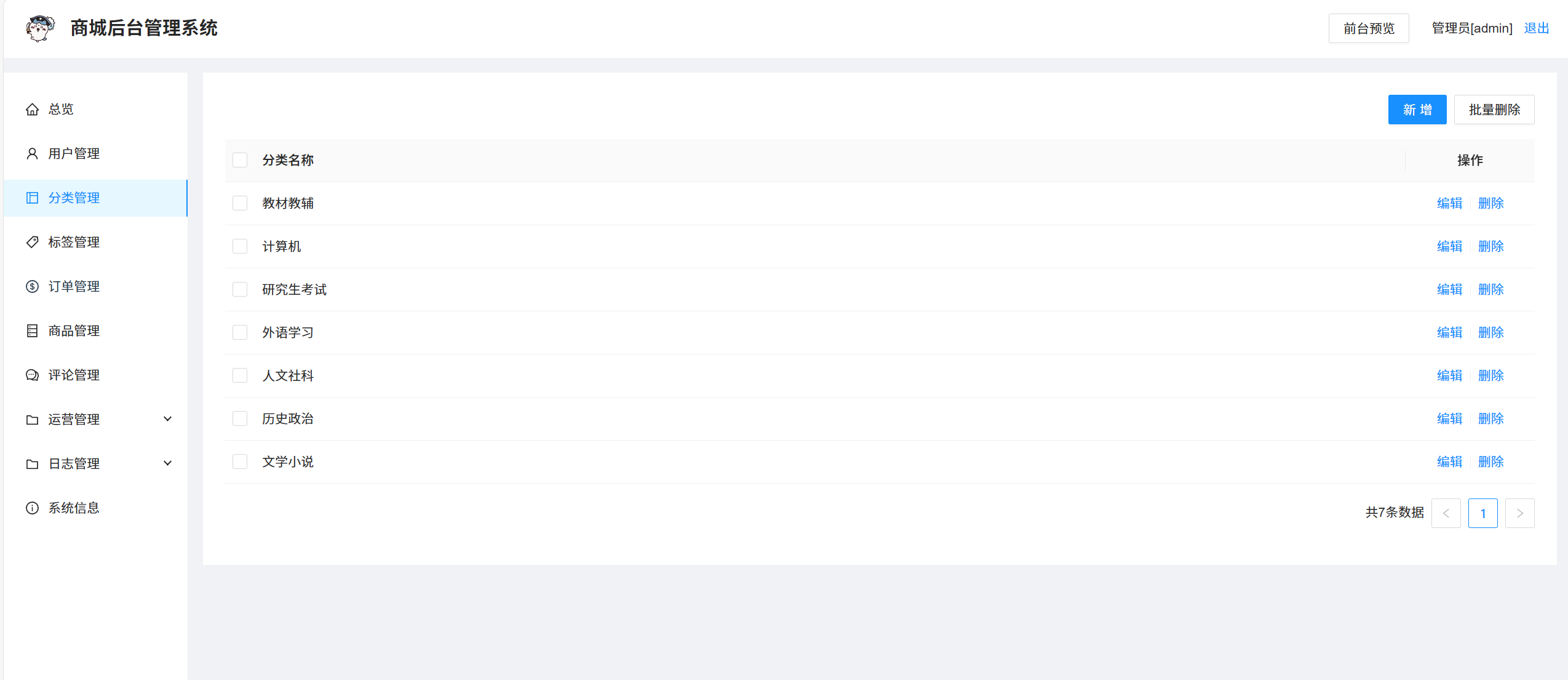
在用户信息管理模块，管理员能够修改用户信息，并根据用户身份的不同，给予当前用户相应的角色权限，并且修改状态、邮箱、手机号等信息。如果想新增用户，可以通过点击新增按钮进行新增，如果要删除用户，点击列表右侧的删除按钮进行删除，如果想查询用户，需输入用户名关键词进行搜索。

界面如下所示：



### 6.3.2 分类管理

点击左侧的分类管理，右侧将展示当前系统的所有分类信息，如果管理员想对当前分类进行编辑，可以点击编辑按钮进行编辑，如果管理员想对当前分类进行删除，可以点击删除按钮就行删除。界面如下所示：



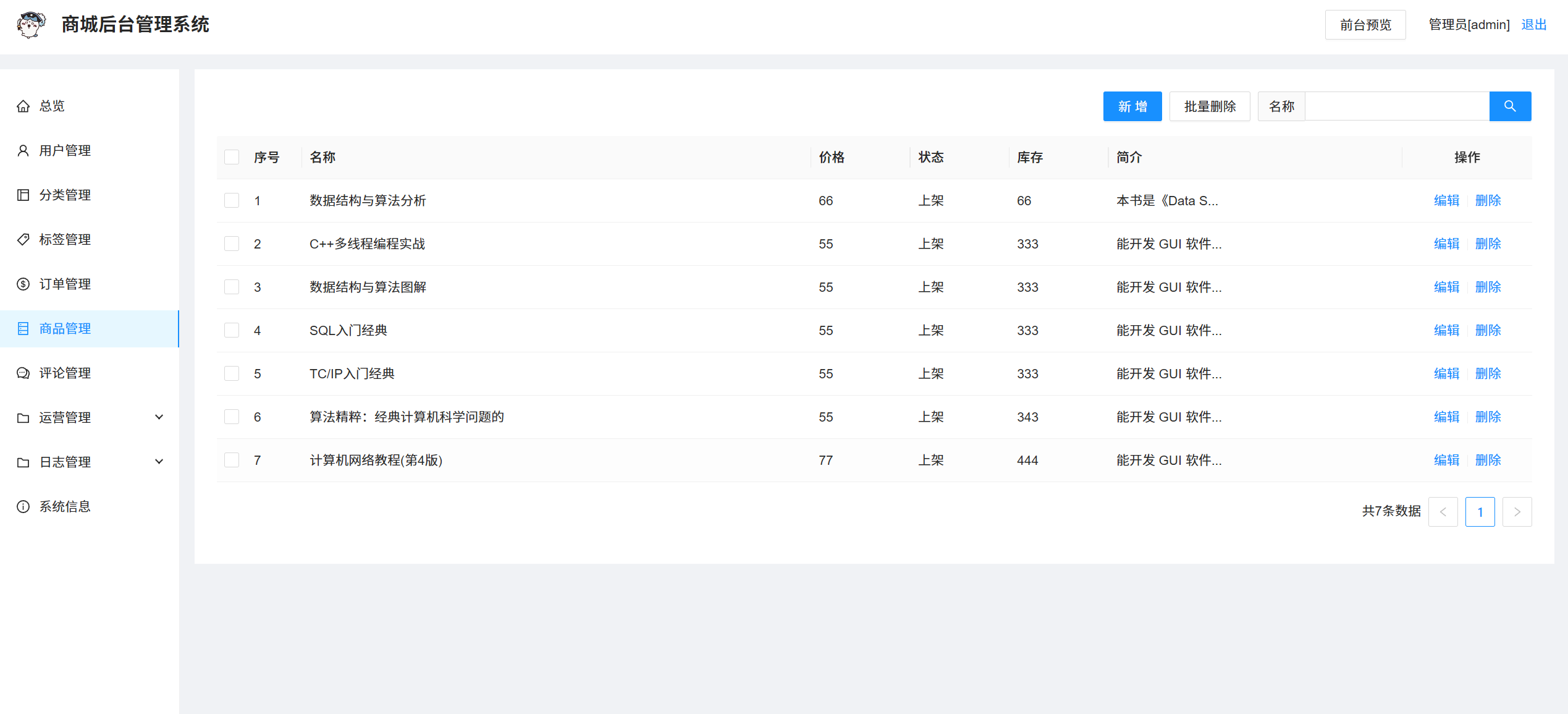
### 6.3.3 标签管理

点击左侧的标签管理，右侧将展示当前系统的所有标签信息，如果管理员想对当前标签进行编辑，可以点击编辑按钮进行编辑，如果管理员想对当前标签进行删除，可以点击删除按钮就行删除。界面如下所示：



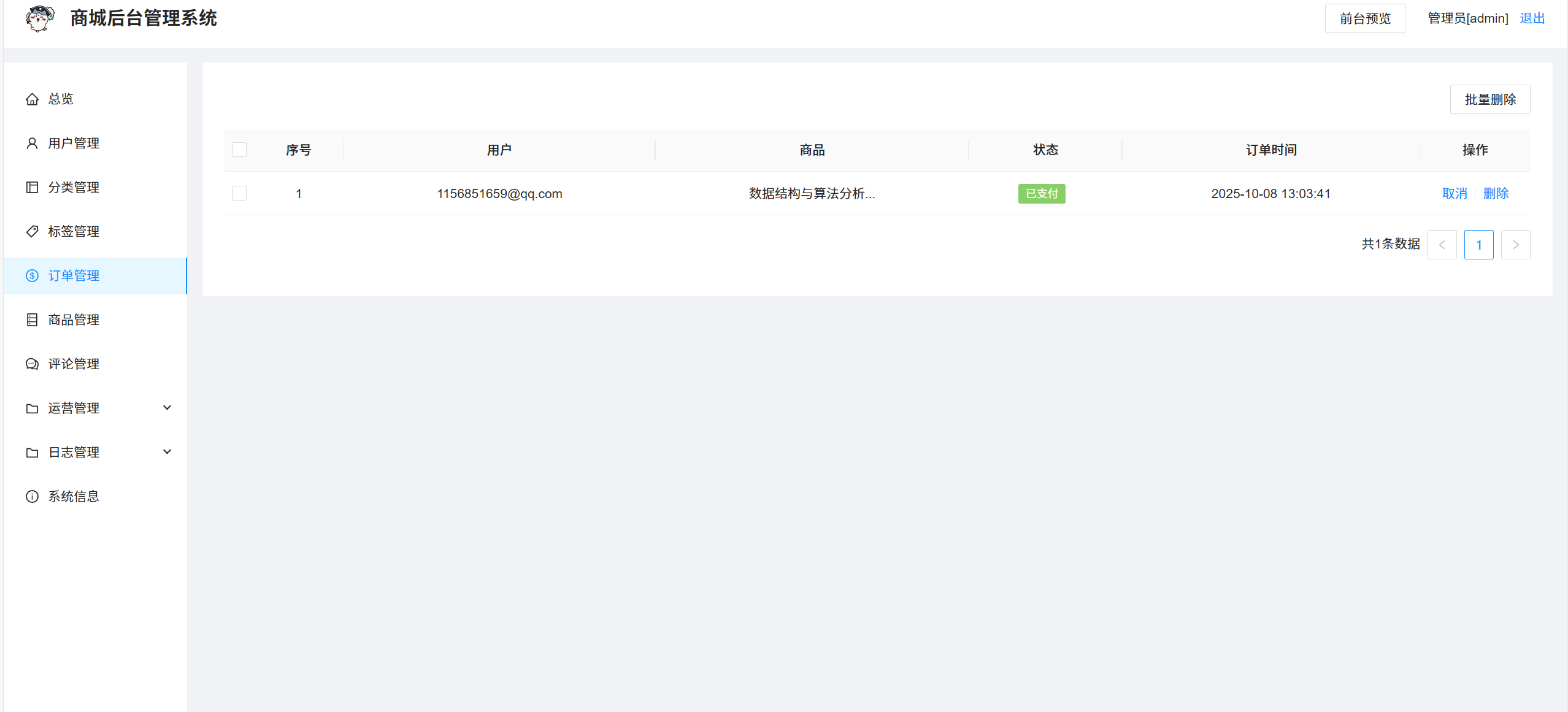
### 6.3.4 商品管理

在商品信息管理模块，管理员能够修改商品信息，包括修改商品状态、商品名称、价格、库存、封面、分类、标签等信息。如果想新增商品，可以通过点击新增按钮进行新增，如果要删除商品，点击列表右侧的删除按钮进行删除，如果想查询商品，需输入商品关键词进行搜索。界面如下所示：



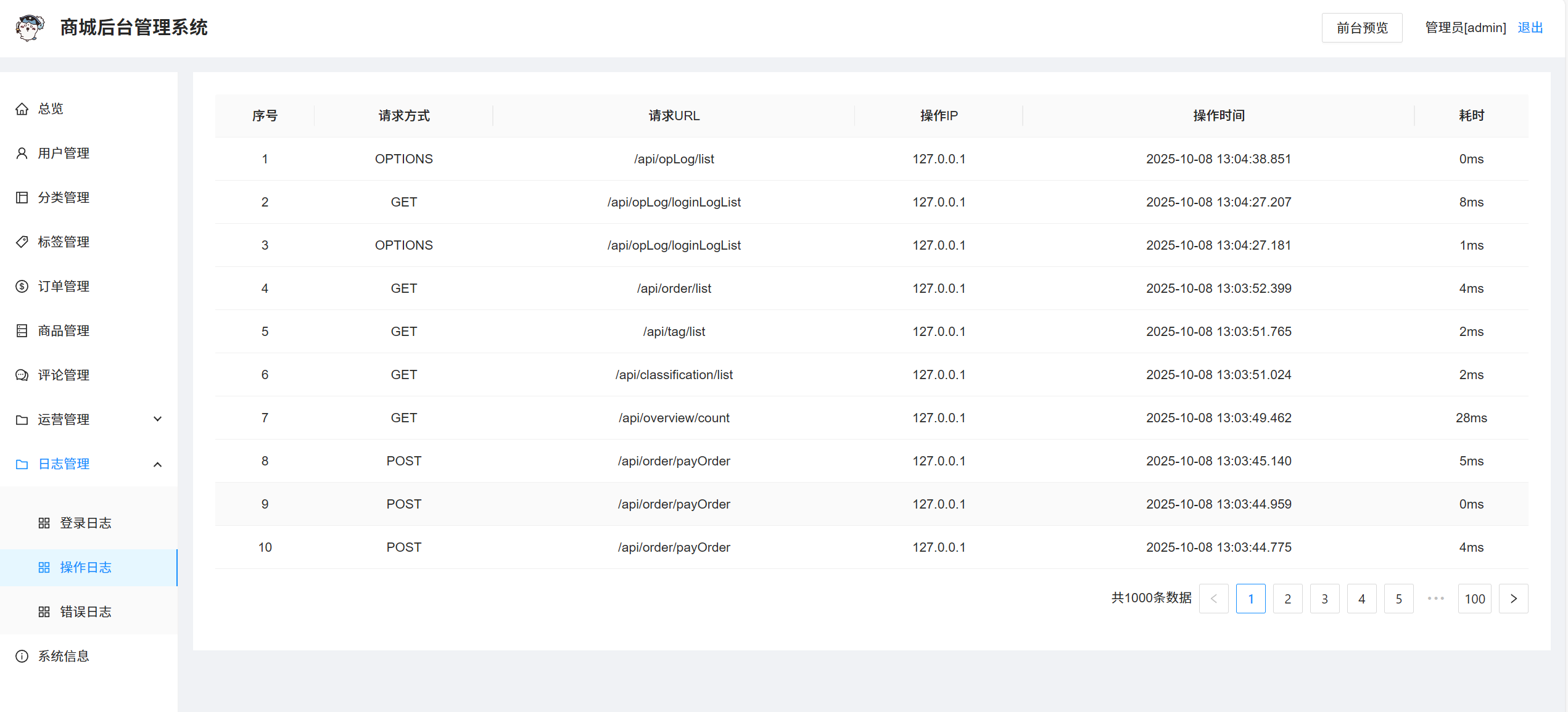
### 6.3.5 订单管理

在订单信息管理模块，管理员能够修改订单状态，如果想取消订单，可以通过点击取消按钮进行取消，如果要删除订单，点击列表右侧的删除按钮进行删除。界面如下所示：



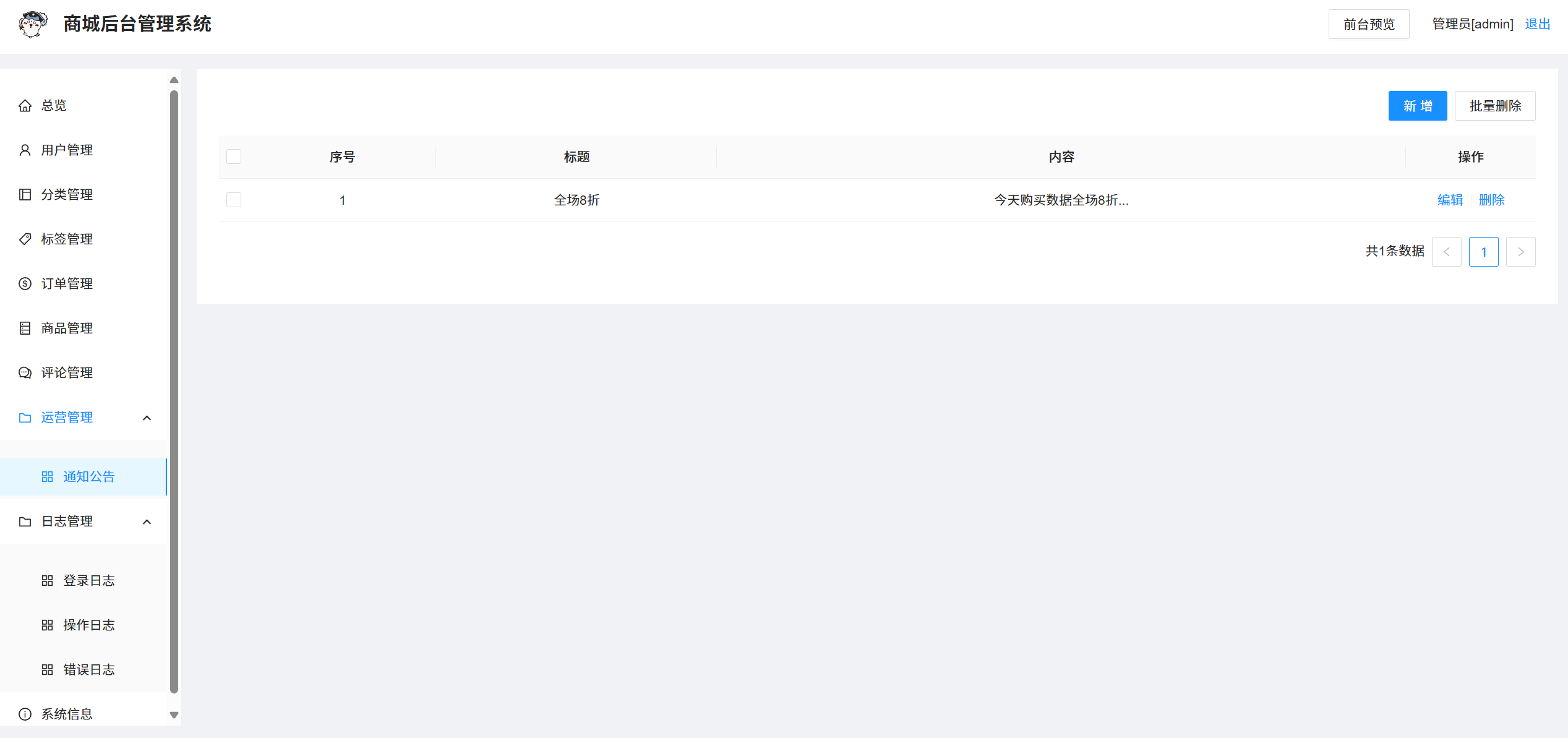
### 6.3.6 日志管理

在日志信息管理模块，包括三部分：登录日志、操作日志、异常日志。管理员通过点击这三项，在右侧列表栏将会显示对应的日志列表，日志基本信息有请求方式、ip地址、操作时间等信息。界面如下所示：



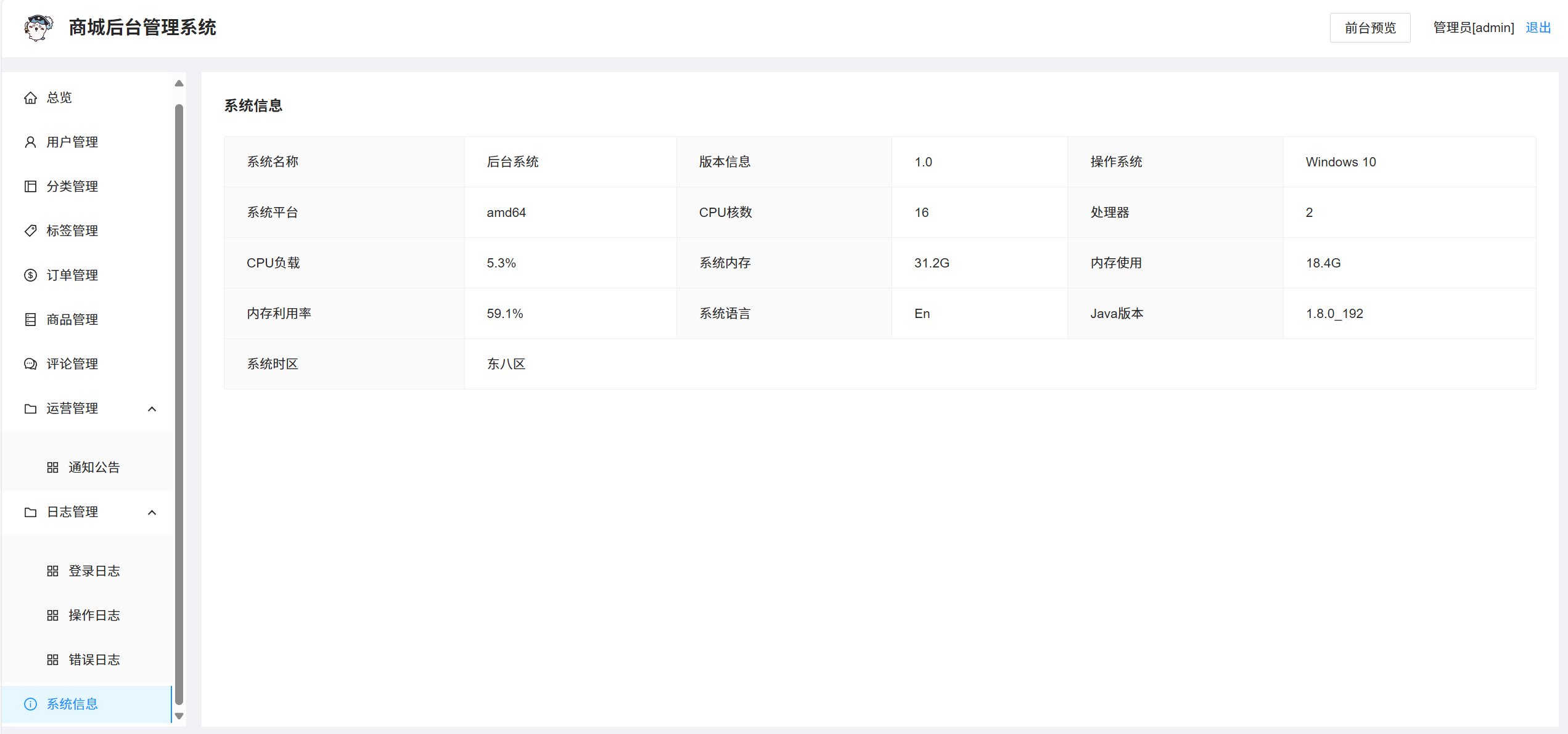
### 6.3.7 运营管理

在运营管理模块，包括两部分：广告管理和通知管理。其中广告管理实现了网站广告设置功能，通知管理实现了整个系统的通知的显示与删除。界面如下所示：



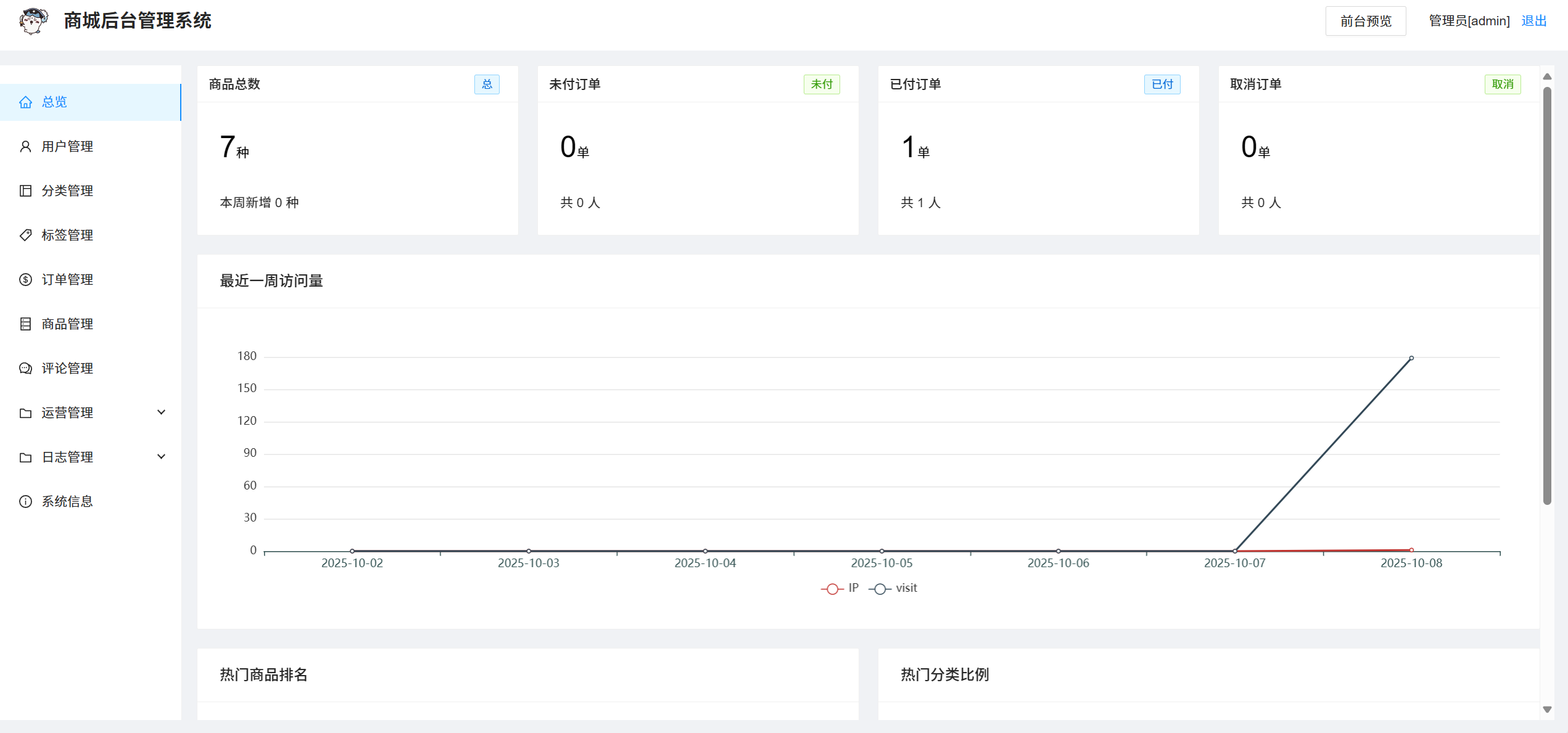
### 6.3.8 系统信息

在系统信息模块，页面通过请求后端系统信息接口，后端通过系统工具类来实现系统名称、版本信息、操作系统、系统平台、CPU核数 、处理器、CPU负载、系统内存、内存使用、内存利用率、系统语言、系统时区等信息，并返回给页面进行渲染显示。界面如下所示：



### 6.3.9 总览模块

在总览模块，系统实现了系统数据的统计分析功能，分别统计了商品数、订单数、网站访问量、热门商品排名、热门分类排名等数据。界面如下所示



## 6.4系统测试

### 6.4.1测试目的

系统测试是软件开发过程中的关键环节，其主要目的是对系统的功能、性能、稳定性、用户体验、安全性和兼容性等方面进行全面验证，通过发现并修复潜在缺陷，确保软件按照需求正常运行，满足性能和安全要求，提升用户满意度，从而保障系统的稳定上线与可靠运行。

### 6.4.2后台测试用例

表6-3 后台登录测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 用例名称 | 用例步骤 | 预期结果 | 实际结果 |
| 1 | 正常登录 | 1. 进入登录界面。 2. 输入正确的用户名和密码 3. 点击登录按钮。 | 成功进入首页，并在右上角显示登录人姓名。 | 与预期结果一致 |
| 2 | 用户名或密码错误 | 1. 进入登录界面。 2. 输入错误的用户名或密码 3. 点击登录按钮。 | 登录失败，提示“登录失败，用户名或密码错误”并返回登录界面。 | 与预期结果一致 |
| 3 | 用户名为空 | 1. 进入登录界面。 2. 不输入用户名 3. 点击登录按钮 | 登录失败，跳出提示，并返回登录界面。 | 与预期结果一致 |
| 4 | 密码为空 | 1. 进入登录界面。 2. 不输入密码 3. 点击登录按钮 | 登录失败，跳出提示，并返回登录界面。 | 与预期结果一致 |

表6-4 用户管理测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 用例名称 | 用例步骤 | 预期结果 | 实际结果 |
| 1 | 查看用户列表 | 1. 登录后台管理系统 2. 点击"用户管理"菜单 | 显示用户列表，包含用户名、邮箱、角色、状态等信息 | 与预期结果一致 |
| 2 | 搜索用户 | 1. 进入用户管理页面 2. 在搜索框输入用户名 3. 点击搜索按钮 | 显示符合搜索条件的用户列表 | 与预期结果一致 |
| 3 | 添加用户 | 1. 进入用户管理页面 2. 点击"添加用户"按钮 3. 填写用户信息 4. 点击提交 | 提示"添加成功"，用户列表中显示新添加的用户 | 与预期结果一致 |
| 4 | 编辑用户 | 1. 进入用户管理页面 2. 点击某用户的"编辑"按钮 3. 修改用户信息 4. 点击提交 | 提示"更新成功"，用户信息更新显示 | 与预期结果一致 |
| 5 | 删除用户 | 1. 进入用户管理页面 2. 点击某用户的"删除"按钮 3. 确认删除 | 提示"删除成功"，该用户从列表中消失 | 与预期结果一致 |
| 6 | 更改用户状态 | 1. 进入用户管理页面 2. 点击用户状态开关 | 用户状态在启用/禁用之间切换，提示"状态更新成功" | 与预期结果一致 |

表6-5 商品管理测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 用例名称 | 用例步骤 | 预期结果 | 实际结果 |
| 1 | 查看商品列表 | 1. 登录后台管理系统 2. 点击"商品管理"菜单 | 显示商品列表，包含商品标题、价格、库存、状态等信息 | 与预期结果一致 |
| 2 | 添加商品 | 1. 进入商品管理页面 2. 点击"添加商品"按钮 3. 填写商品信息并上传图片 4. 点击提交 | 提示"创建成功"，商品列表中显示新添加的商品 | 与预期结果一致 |
| 3 | 编辑商品 | 1. 进入商品管理页面 2. 点击某商品的"编辑"按钮 3. 修改商品信息 4. 点击提交 | 提示"更新成功"，商品信息更新显示 | 与预期结果一致 |
| 4 | 删除商品 | 1. 进入商品管理页面 2. 点击某商品的"删除"按钮 3. 确认删除 | 提示"删除成功"，该商品从列表中消失 | 与预期结果一致 |
| 5 | 商品上下架 | 1. 进入商品管理页面 2. 修改商品状态为上架或下架 3. 点击保存 | 商品状态更新，前台页面显示或隐藏该商品 | 与预期结果一致 |
| 6 | 按分类筛选商品 | 1. 进入商品管理页面 2. 选择商品分类 3. 点击筛选 | 显示该分类下的所有商品 | 与预期结果一致 |

表6-6 订单管理测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 用例名称 | 用例步骤 | 预期结果 | 实际结果 |
| 1 | 查看订单列表 | 1. 登录后台管理系统 2. 点击"订单管理"菜单 | 显示订单列表，包含订单号、用户、商品、金额、状态等信息 | 与预期结果一致 |
| 2 | 按状态筛选订单 | 1. 进入订单管理页面 2. 选择订单状态 3. 点击筛选 | 显示对应状态的订单列表 | 与预期结果一致 |
| 3 | 查看订单详情 | 1. 进入订单管理页面 2. 点击某订单的"详情"按钮 | 显示订单的详细信息，包括商品详情、收货地址等 | 与预期结果一致 |
| 4 | 更新订单状态 | 1. 进入订单管理页面 2. 点击某订单的"发货"按钮 3. 确认操作 | 订单状态更新为"已发货"，提示"操作成功" | 与预期结果一致 |

表6-7 分类管理测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 用例名称 | 用例步骤 | 预期结果 | 实际结果 |
| 1 | 查看分类列表 | 1. 登录后台管理系统 2. 点击"分类管理"菜单 | 显示分类列表，包含分类名称、图片、描述等信息 | 与预期结果一致 |
| 2 | 添加分类 | 1. 进入分类管理页面 2. 点击"新增分类"按钮 3. 填写分类信息 4. 点击提交 | 提示"创建成功"，分类列表中显示新添加的分类 | 与预期结果一致 |
| 3 | 编辑分类 | 1. 进入分类管理页面 2. 点击某分类的"编辑"按钮 3. 修改分类信息 4. 点击提交 | 提示"更新成功"，分类信息更新显示 | 与预期结果一致 |
| 4 | 删除分类 | 1. 进入分类管理页面 2. 点击某分类的"删除"按钮 3. 确认删除 | 提示"删除成功"，该分类从列表中消失 | 与预期结果一致 |

表6-8 系统总览测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 用例名称 | 用例步骤 | 预期结果 | 实际结果 |
| 1 | 查看系统统计数据 | 1. 登录后台管理系统 2. 进入总览页面 | 显示用户总数、商品总数、订单总数、销售额等统计数据 | 与预期结果一致 |
| 2 | 查看订单趋势图 | 1. 进入总览页面 2. 查看订单趋势图表类"按钮 | 显示近7天的订单数量趋势图 | 与预期结果一致 |
| 3 | 查看分类统计 | 1. 进入总览页面 2. 查看分类统计图表 | 显示各分类下的商品数量统计 | 与预期结果一致 |

### 6.4.2前台测试用例

表6-9 登录用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 用例名称 | 用例步骤 | 预期结果 | 实际结果 |
| 1 | 正常登录 | 1. 进入登录界面。 2. 输入正确的用户名和密码 3. 点击登录按钮。 | 成功进入首页，并在右上角显示登录人姓名。 | 与预期结果一致 |
| 2 | 用户名或密码错误 | 1. 进入登录界面。 2. 输入错误的用户名或密码 3. 点击登录按钮。 | 登录失败，提示“登录失败，用户名或密码错误”并返回登录界面。 | 与预期结果一致 |
| 3 | 用户名为空 | 1. 进入登录界面。 2. 不输入用户名 3. 点击登录按钮 | 登录失败，跳出提示，并返回登录界面。 | 与预期结果一致 |
| 4 | 密码为空 | 1. 进入登录界面。 2. 不输入密码 3. 点击登录按钮 | 登录失败，跳出提示，并返回登录界面。 | 与预期结果一致 |

表6-10 前台注册测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 用例名称 | 用例步骤 | 预期结果 | 实际结果 |
| 1 | 正常注册 | 1. 进入注册页 2. 输入正确信息 3. 点击注册按钮 | 跳出提示“注册成功”，并返回主页面 | 与预期结果一致 |
| 2 | 用户名为空 | 1. 进入注册界面 2. 不输入用户名 3. 点击注册按钮 | 跳出提示"名称不能为空！"并返回注册界面 | 与预期结果一致 |
| 3 | 密码为空 | 1. 进入注册界面 2. 不输入密码 3. 点击注册按钮 | 跳出提示"密码不能为空！"并返回注册界面 | 与预期结果一致 |
| 4 | 手机号为空 | 1. 进入注册界面 2. 不输入手机号 3. 点击注册按钮 | 跳出提示"手机号不能为空！"并返回注册界面 | 与预期结果一致 |
| 5 | 用户名已存在 | 1. 进入注册界面 2. 输入已存在的用户名 3. 点击注册按钮 | 跳出提示"用户名已存在"并返回注册界面 | 与预期结果一致 |

表6-11 首页功能测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 用例名称 | 用例步骤 | 预期结果 | 实际结果 |
| 1 | 浏览商品列表 | 1. 进入首页 | 显示商品列表，每个商品显示封面、标题、价格等信息 | 与预期结果一致 |
| 2 | 商品分类筛选 | 1. 进入首页 2. 点击某个分类 | 显示该分类下的所有商品 | 与预期结果一致 |
| 3 | 商品搜索 | 1. 进入首页 2. 在搜索框输入关键词 3. 点击搜索 | 显示包含关键词的商品列表 | 与预期结果一致 |
| 4 | 商品分页 | 1. 进入首页 2. 点击分页按钮 | 显示下一页的商品列表 | 与预期结果’一致 |

表6-13 用户中心测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 用例名称 | 用例步骤 | 预期结果 | 实际结果 |
| 1 | 查看个人信息 | 1. 登录系统 2. 进入用户中心 3. 点击"个人信息" | 显示用户的个人信息，包括用户名、昵称、邮箱等 | 与预期结果一致 |
| 2 | 修改个人信息 | 1. 进入用户中心个人信息页 2. 修改昵称或邮箱 3. 点击"保存修改" | 提示"更新成功"，个人信息更新显示 | 与预期结果一致 |
| 3 | 查看我的订单 | 1. 进入用户中心 2. 点击"我的订单" | 显示用户的所有订单列表 | 与预期结果一致 |
| 4 | 按状态筛选订单 | 1. 进入我的订单页面 2. 点击"待支付"标签 | 显示所有待支付的订单 | 与预期结果’一致 |
| 5 | 查看我的收藏 | 1. 进入用户中心 2. 点击"我的收藏" | 显示用户收藏的所有商品 | 与预期结果一致 |
| 6 | 取消收藏 | 1. 进入我的收藏页面 2. 点击某商品的取消收藏按钮 | 提示"取消收藏成功"，该商品从收藏列表中移除 | 与预期结果一致 |

表6-14 AI客服测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 用例名称 | 用例步骤 | 预期结果 | 实际结果 |
| 1 | 打开AI客服 | 1. 进入系统任意页面 2. 点击AI客服图标 | 打开AI客服对话窗口，显示欢迎消息 | 与预期结果一致 |
| 2 | 发送消息 | 1. 打开AI客服窗口 2. 输入问题 3. 点击发送 | AI回复相关答案，消息显示在对话框中 | 与预期结果一致 |
| 3 | 商品咨询 | 1. 打开AI客服窗口 2. 询问"有哪些编程类的书？" 3. 点击发送 | 显示用户的所有订单列表 | 与预期结果一致 |
| 4 | 连续对话 | 1. 打开AI客服窗口 2. 连续发送多条消息 | AI能理解上下文，提供连贯的回复 | 与预期结果’一致 |

### 6.4.3 测试结论

本系统在测试过程中，对前台和后台的所有功能模块进行了全面的测试。测试覆盖了用户登录注册、商品浏览购买、订单管理、后台管理、AI智能客服等核心功能。

经过严格的测试，所有核心功能均能正常运行，用户体验良好。测试过程中发现的问题已全部修复并通过回归测试验证。系统的稳定性、性能和安全性均达到预期目标。

# 7 结论

通过几年的大学学习，我。。。。

参考文献

[1]何为,迟文恒Java服务端研发知识图谱[M].机械工业出版社:信息科学与技术丛书, 201812.430.

[2]杨开振. 深入浅出Spring Boot 2.x[M].人民邮电出版社:, 201808.442.

[3]古凌岚,张婵,罗佳. Java系统化项目开发教程[M].人民邮电出版社:, 201802.313.

[4]肖睿,喻晓路,朱微,张超,戴立坤. Java Web应用设计及实战[M].人民邮电出版社:大数据开发实战系列, 201801.238.

[5]李宗花,朱林. 软件工程原理与实践[M].南京大学出版社:, 202008.252.

[6]陆惠恩,褚秋砚. 软件工程[M].人民邮电出版社:, 201712.284.

[7]肖睿,郭泰,王丁磊. SSH框架企业级应用实战[M].人民邮电出版社:大数据开发实战系列, 201801.344.

[8]付森,石亮,吴起立,刘冰. MySQL开发与实践[M].人民邮电出版社:, 201408.261.

[9]陈陆扬. Vue.js前端开发快速入门与专业应用[M].人民邮电出版社:, 201702.207.

[10]陈晓兵. Web前端开发技术[M].南京大学出版社:, 202008.252.

[11]杨洋,刘全. 软件系统分析与体系结构设计[M].南京东南大学出版社:, 201710.207.

[12]王昊,刘友华. 信息系统分析与设计[M].南京大学出版社:, 202109.323.

[13]张永奎. 数据库原理与设计[M].人民邮电出版社:, 201909.260.

[14]佟伟光,郭霏霏. 软件测试[M].人民邮电出版社:, 201501.281.

[15]郑钢,贺亚涛,尤胜涛. 大型网站服务器容量规划[M].人民邮电出版社:201608.204.

[16]姚庆涛.浅析计算机网络应用安全[J].网络安全技术与应用,2023(05):169-171.

[17]熊威.Java编程基础课程类和对象的教学设计[J/OL].中国教育技术装备:1-4

[18]齐研科,杨颖,贺喜,蔡雨耕,赵家骏,王一先. 软件开发系统及方法[P]. 重庆市：2023-05-02.

[19]朱若榕.人机交互软件界面设计的重要性[J].文化产业,2023(12):153-155.

[20]刘德山,金百东. Java设计模式深入研究[M].人民邮电出版社:, 201407.222.

致谢

在这几个月的时间里，从对课题的理解，。。。。