**武汉体育学院**

**信息系统分析与设计报告**

**题 目： 校园二手交易平台**

**学生姓名：李明树**

**专业班级： 信息系统与信息管理**

**学 院： 经济与管理学院**

2025年 6 月 7 日

**成绩评定表**

|  |  |  |
| --- | --- | --- |
| 一级指标 | 二级指标 | **成绩** |
| 1.选题  （5分） | 1.1选题背景及意义 (2分) |  |
| 1.2 题目难度及工作量 (3分) |
| 2.可行性分析  （5分） | 2.1 系统设计目标 （1分） |  |
| 2.2 系统开发初步方案 (2分) |
| 2.3 可行性分析 (2分) |
| 3.系统分析  （20分） | 3.1 组织与功能分析 (2分) |  |
| 3.2 业务与业务过程分析 (5分) |
| 3.3信息与信息过程分析 (5分) |
| 3.4 新系统逻辑模型建立 (8分) |
| 4.系统设计  （60分） | 4.1 系统体系结构设计 (5分) |  |
| 4.2 系统运行处理流程设计 (5分) |
| 4.3 系统功能与模块关系设计 (5分） |
| 4.4 系统物理配置方案设计 （5分） |
| 4.5 数据库设计 (8分) |
| 4.6 代码方案设计 (2分) |
| 4.7 输入与输出设计 (5分) |
| 4.8 系统界面设计 (5分) |
| 4.9 模块流程设计 (5分) |
| 4.10 系统原型设计 (15分) |
| 5.系统实施  （10） | 5.1 物理系统实施设计 (3分) |  |
| 5.3 系统测试设计 (5分) |
| 5.4 系统运行设计 (2分) |
| **合计** | |  |

目 录

[一、引言 7](#_Toc201176593)

[1.1 选题背景 7](#_Toc201176594)

[1.2 选题目的与意义 7](#_Toc201176595)

[二、可行性分析 8](#_Toc201176596)

[2.1 系统设计目标 8](#_Toc201176597)

[2.1.1 功能性目标 8](#_Toc201176598)

[2.1.3 非功能性目标 10](#_Toc201176599)

[2.2 系统开发初步方案 12](#_Toc201176600)

[2.2.1 技术架构 12](#_Toc201176601)

[2.2.2 开发模式 12](#_Toc201176602)

[2.2.3 开发周期 12](#_Toc201176603)

[2.2.4 团队分工 12](#_Toc201176604)

[2.3 可行性分析 12](#_Toc201176605)

[2.3.1 技术可行性 12](#_Toc201176606)

[2.3.2 经济可行性 13](#_Toc201176607)

[2.3.3 操作可行性 13](#_Toc201176608)

[2.3.4 法律可行性 13](#_Toc201176609)

[三、系统分析 13](#_Toc201176610)

[3.1 组织与功能分析 13](#_Toc201176611)

[3.1.1 组织结构分析 13](#_Toc201176612)

[3.1.2 系统功能模块 13](#_Toc201176613)

[3.2 业务与业务过程分析 14](#_Toc201176614)

[3.2.1 主要业务 14](#_Toc201176615)

[3.2.2 核心业务过程 15](#_Toc201176616)

[3.2.3 业务规则 15](#_Toc201176617)

[3.3 信息与信息过程分析 17](#_Toc201176618)

[3.3.1 主要信息流 17](#_Toc201176619)

[3.3.2 数据处理过程 17](#_Toc201176620)

[3.3.3 关键数据实体 17](#_Toc201176621)

[3.3.4 数据字典（部分） 18](#_Toc201176622)

[3.3.5 核心数据实体 18](#_Toc201176623)

[3.3.6 关键数据流 18](#_Toc201176624)

[3.3.7 数据约束 19](#_Toc201176625)

[3.4 新系统逻辑模型建立 19](#_Toc201176626)

[3.4.1 对象模型（类图） 19](#_Toc201176627)

[3.4.2 功能模型（用例图） 21](#_Toc201176628)

[四、系统设计 22](#_Toc201176629)

[4.1 系统体系结构设计 22](#_Toc201176630)

[4.1.1 总体架构 22](#_Toc201176631)

[4.1.2 层次结构 22](#_Toc201176632)

[4.1.3 系统交互流程 24](#_Toc201176633)

[4.1.4 技术选型 25](#_Toc201176634)

[4.1.5 架构优势 25](#_Toc201176635)

[4.2 系统运行处理流程设计 27](#_Toc201176636)

[4.2.1 顺序图 27](#_Toc201176637)

[4.2.2系统流程 28](#_Toc201176638)

[4.3 系统功能与模块关系设计 28](#_Toc201176639)

[4.3.1模块划分细化 28](#_Toc201176640)

[4.3.2 模块职责定义 29](#_Toc201176641)

[4.3.3 模块间接口定义 30](#_Toc201176642)

[4.3.4 模块依赖关系 32](#_Toc201176643)

[4.4 系统物理配置方案设计 33](#_Toc201176644)

[4.4.1硬件选型清单 33](#_Toc201176645)

[4.4.2软件选型清单 33](#_Toc201176646)

[4.4.3网络配置方案 34](#_Toc201176647)

[4.4.4部署拓扑详图 35](#_Toc201176648)

[4.4.5容量规划 36](#_Toc201176649)

[4.5 数据库设计 38](#_Toc201176650)

[4.5.1概念数据模型（CDM） 38](#_Toc201176651)

[4.5.2逻辑数据模型（LDM） 39](#_Toc201176652)

[4.5.3物理数据模型（PDM） 42](#_Toc201176653)

[4.6 代码方案设计 45](#_Toc201176654)

[4.6.1 用户管理模块代码设计 46](#_Toc201176655)

[4.6.2 商品管理模块代码设计 46](#_Toc201176656)

[4.6.3 交易管理模块代码设计 46](#_Toc201176657)

[4.6.4 消息管理模块代码设计 47](#_Toc201176658)

[4.6.5 后台管理模块代码设计 47](#_Toc201176659)

[4.7 输入与输出设计 47](#_Toc201176660)

[4.7.1输入设计 47](#_Toc201176661)

[4.7.2输出设计 48](#_Toc201176662)

[4.8 系统界面设计 50](#_Toc201176663)

[4.8.1 首页界面 50](#_Toc201176664)

[4.8.2 购物车界面 50](#_Toc201176665)

[4.8.3 个人中心页面 51](#_Toc201176666)

[4.8.4 搜索结果界面 51](#_Toc201176667)

[4.8.5 购买界面 52](#_Toc201176668)

[4.8.6 消息界面 52](#_Toc201176669)

[4.9 模块流程设计 53](#_Toc201176670)

[4.10 系统原型设计 55](#_Toc201176671)

[五、系统实施 58](#_Toc201176672)

[5.1 物理系统实施设计 58](#_Toc201176673)

[5.1.2基础设施规划 58](#_Toc201176674)

[5.1.2物理设备清单 59](#_Toc201176675)

[5.1.3系统模块部署方案 59](#_Toc201176676)

[5.1.4容灾与备份策略 59](#_Toc201176677)

[5.1.5运维监控体系 59](#_Toc201176678)

[5.1.6实施进度计划 60](#_Toc201176679)

[5.2 系统测试设计 60](#_Toc201176680)

[5.2.1测试目标 60](#_Toc201176681)

[5.2.2测试范围 60](#_Toc201176682)

[5.2.3详细测试策略 61](#_Toc201176683)

[5.2.4测试环境与数据 62](#_Toc201176684)

[5.2.5测试交付物 62](#_Toc201176685)

[5.2.6风险应对 62](#_Toc201176686)

[5.3 系统运行设计 62](#_Toc201176687)

[5.3.1运行环境配置 62](#_Toc201176688)

[5.3.2关键服务运行设计 63](#_Toc201176689)

[5.3.3持续运维机制 63](#_Toc201176690)

[5.3.4应急预案设计 64](#_Toc201176691)

[5.3.5运行成本控制 64](#_Toc201176692)

[5.3.6运行文档体系 64](#_Toc201176693)

[六、结束语 65](#_Toc201176694)

[七、参考文献 65](#_Toc201176695)

# 一、引言

## 1.1 选题背景

随着武汉体育学院校园规模的不断扩大以及师生数量的日益增多，校园内的二手物品交易需求逐渐凸显。学生在学习和生活中会不断产生各类闲置物品，如运动器材、教材、生活用品等，而目前缺乏一个便捷、安全、高效的官方二手交易平台。这不仅导致资源的浪费，也给有需求的师生带来诸多不便。目前校内二手交易主要通过微信群聊、校园表白墙或线下张贴广告等方式进行，存在信息不对称、交易效率低下、缺乏安全保障等诸多问题。因此，开发一个专门针对武汉体育学院校园的二手交易平台具有重要的现实意义，能够促进校园资源的循环利用，方便师生的生活，营造良好的校园交易氛围。

随着“共享经济”和“循环经济”理念的普及，高校二手交易市场潜力巨大。建立一个规范、安全、便捷的校内二手交易平台，能够有效整合资源，促进校园内闲置物品的流通，符合绿色校园建设的理念。

## 1.2 选题目的与意义

本需求说明书旨在详细阐述武汉体育学院二手交易平台的功能需求、性能要求以及运行环境等各项规定，为项目的开发、测试、维护以及后续的升级提供明确的规范与依据，确保项目能够满足校园内师生对于二手物品交易的实际需求，同时保证系统开发过程的顺利进行以及最终产品的稳定可靠。

1. 资源优化：促进校内资源循环利用，减少浪费。
2. 经济实惠：为学生提供经济实惠的二手物品交易渠道。
3. 安全保障：规范交易流程，提供信息评价机制，降低交易风险。
4. 便捷高效：提供统一交易平台，相较于表白墙、微信群聊的人数局限，大大提高了交易效率
5. 为学校了解学生需求提供数据支持

具体目标:

1. 建立统一的二手物品信息发布与管理平台
2. 实现用户身份认证与信息评价体系
3. 提供便捷的搜索与沟通功能
4. 设计安全的交易流程与纠纷处理机制
5. 实现数据统计与分析功能

# 二、可行性分析

## 2.1 系统设计目标

### 2.1.1 功能性目标

用户管理：注册、登录、身份认证、个人信息管理

商品管理：发布、编辑、下架、搜索、分类浏览

交易管理：订单生成、状态跟踪、交易评价

消息管理：站内信、交易提醒

信用管理：用户信用评价体系

数据统计：交易数据分析与可视化

**对性能的规定:**

**1）.精度**

商品信息的展示应准确无误，包括价格、新旧程度、所属分类、描述等，确保用户在浏览商品时能够获取真实的商品详情，避免因信息误差导致交易纠纷；交易记录的保存和统计也需保证数据的精确性，如交易金额、时间、双方用户信息等，为财务核算和数据分析提供可靠的数据基础。

用户评价数据的记录和展示应保持原汁原味，客观反映双方的交易体验和反馈，不得对评价内容进行随意修改或篡改，确保评价系统的公正性和可信度，为其他用户提供个性参考依据。

**2）.灵活性**

系统应具备良好的适应性和可扩展性，能够根据校园二手交易市场的发展变化以及用户需求的不断演变，方便地进行功能模块的增删改查和业务流程的调整优化。例如，随着新的商品类别出现或交易模式的创新，系统能够快速响应并进行相应的功能扩展和配置修改，无需对整个系统进行大规模的重构，降低系统的维护成本和升级难度。

界面布局和操作流程应具有一定的灵活性，允许用户根据个人使用习惯进行一定程度的个性化设置，如自定义首页展示内容、设置消息提醒方式、调整商品列表显示样式等，提高用户的使用满意度和操作便捷性。

**3）.输入输出**

输入方面，系统应提供友好的输入界面和提示信息，引导用户正确、快速地输入各类数据，如在商品发布页面，对必填项进行明确标注，并对输入的数据格式（如价格格式、手机号码格式等）进行实时校验，当用户输入错误时及时给予错误提示并指导用户进行修改；对于批量导入数据的情况，应提供详细的导入模板和示例说明，帮助用户准确填写导入数据，提高数据导入的成功率。

输出方面，无论是商品搜索结果、交易记录查询结果还是各类统计分析图表，都应以清晰、直观、美观的方式呈现给用户，确保用户能够快速理解和获取所需信息。对于导出的数据文件，应按照规定的格式规范进行生成，保证文件的兼容性和可读性，方便用户在其他软件中进行查看和处理。

**4）.数据管理能力**

系统应具备高效的数据存储和管理能力，能够妥善处理大量的用户信息、商品数据、交易记录等各类数据，确保数据存储的安全性、完整性和可靠性。采用合理的数据库设计和优化策略，如数据分表、索引优化等，保证数据的快速读写和查询性能，避免因数据量过大导致系统响应迟缓或出现卡顿现象。

建立完善的数据备份与恢复机制，定期对数据库进行备份操作，以防数据丢失或损坏。在出现系统故障、数据误操作或其他意外情况时，能够及时恢复数据至最近的可用状态，最大限度地减少数据损失，保障平台的正常运行和数据的可用性。

**5）.故障处理的要求**

系统应具备良好的稳定性和容错性，能够自动检测和处理常见的故障情况，如网络中断、数据库连接失败、服务器异常等。当出现故障时，系统应及时向管理员发出警报通知，同时在前端界面为用户显示友好的错误提示信息，说明故障原因和可能的解决方法，避免用户因系统故障而产生困惑或不满情绪。

针对不同的故障类型，制定相应的应急预案和恢复措施，确保系统能够在最短的时间内恢复正常运行，降低故障对用户使用和平台运营的影响。例如，在服务器出现故障时，能够快速切换到备用服务器，保证平台服务的不间断性；对于数据丢失或损坏的情况，通过数据备份及时进行恢复，确保数据的完整性和一致性。

**6）.其他专门要求**

系统应遵循学校的信息化建设规范和安全标准，与学校的其他信息系统（如校园统一身份认证系统、教务系统等）进行有效的集成和数据共享，实现单点登录和用户信息的互联互通，避免用户在不同系统之间重复注册和登录，提高校园信息系统的整体协同性和用户体验。

考虑到校园二手交易的特殊性，系统应具备一定的地域针对性和文化适应性，充分结合武汉体育学院的校园文化和特色，如在界面设计上融入学校的标志性元素和色彩风格，展示学校的运动特色和学科优势，使平台更具亲和力和归属感，更容易被师生所接受和使用。

### 2.1.3 非功能性目标

性能：响应时间<2秒，支持200并发

安全性：数据加密、防SQL注入、XSS防护

可用性：99%的正常运行时间

可扩展性：模块化设计，便于功能扩展

用户体验：简洁直观的界面设计

**运行环境规定:**

**1）.硬件设备**

服务器 ：推荐采用中高端服务器设备，具备多核处理器、较大容量的内存和高速存储设备，以满足系统在高并发访问情况下的数据处理和存储需求，保证系统的稳定运行和快速响应。同时，服务器应具备良好的散热性能和可靠性，配备冗余电源等硬件设施，防止因硬件故障导致系统停机。

客户端 ：支持常见的计算机设备（包括台式电脑、笔记本电脑）以及移动终端设备（如智能手机、平板电脑），用户可以通过浏览器在不同设备上访问平台，系统应保证在各种设备屏幕尺寸下的自适应显示效果，提供一致的用户体验。对于移动终端设备，考虑到校园网络的多样性和移动设备的性能差异，系统应进行优化，确保在不同网络环境下能够流畅访问和操作平台。

**2）.软件支持**

操作系统 ：服务器端推荐采用稳定可靠的 Linux 操作系统（Ubuntu），其具备良好的多任务处理能力、网络服务能力和安全性，能够为系统运行提供高效、稳定的底层支持；客户端则支持主流的浏览器软件，如谷歌 Chrome、火狐 Firefox、微软 Edge、360 浏览器等，确保用户在不同操作系统平台（Windows、macOS、Linux）下均能正常使用平台的各项功能。

数据库管理系统 ：选用成熟主流的关系型数据库（MySQL）来存储系统的各类数据，根据数据的特点和业务需求进行合理选择和搭配，确保数据库的高性能、高可用性和易扩展性。数据库管理系统应具备数据备份、恢复、事务管理、安全管理等功能，保障数据的完整性和安全性。

开发与运行环境 ：前端开发采用 HTML5、CSS3、JavaScript 等主流的 Web 开发技术框架，结合 Vue.js等前端框架进行界面构建和交互设计，确保界面的美观性、响应式布局以及良好的用户体验；后端开发基于 Python（Django 或 Flask 框架）、Java（Spring Boot 框架）等主流编程语言和框架进行业务逻辑的实现和数据处理，保证系统的高效开发和稳定运行。同时，系统应利用 Nginx 等 Web 服务器软件进行部署和运行，确保平台能够对外提供稳定可靠的 HTTP 服务。

**3）.接口**

系统应提供与校园统一身份认证系统的标准接口，实现用户的单点登录功能，通过调用该接口获取用户的基本身份信息（如姓名、学号 / 工号、邮箱等），完成用户的注册和登录流程，确保用户身份的真实性和合法性，同时也避免了用户在不同系统之间的重复注册和密码记忆问题。

为了便于后续的数据扩展和与其他系统的数据交互，系统应预留相应的数据接口规范，采用常见的数据交换格式（如 JSON、XML 等）和通信协议（如 HTTP、HTTPS 等），方便与其他校园信息系统（如教务系统、后勤管理系统等）进行数据共享和业务协同，实现校园信息资源的整合与优化利用。

**4）.控制**

系统应具备完善的用户权限控制机制，基于用户角色（学生、教职工、管理员）进行细粒度的权限分配和管理，确保用户只能访问和操作其被授权的功能模块和数据资源，防止非法访问和数据泄露等安全问题。例如，普通用户只能浏览和操作自己发布的商品信息和交易记录，而管理员可以对整个平台的数据和用户行为进行全面的监控和管理，但管理员的权限操作也应受到严格的审计和记录，确保系统的安全性和可控性。

对于数据的访问和操作，系统应实施严格的访问控制策略，采用数据加密、数字签名等技术手段对敏感数据（如用户的联系方式、交易记录等）进行保护，确保数据在传输和存储过程中的保密性和完整性。同时，系统应具备防sql注入攻击、XSS 攻击等常见的 Web 安全防护措施，防止恶意用户利用系统漏洞进行非法操作和数据破坏，保障平台的安全稳定运行。

## 2.2 系统开发初步方案

### 2.2.1 技术架构

* 1. 前端：Vue.js
  2. 后端：Spring Boot
  3. 数据库：MySQL
  4. 缓存：Redis
  5. 部署：Docker + Nginx

### 2.2.2 开发模式

采用敏捷开发模式，分阶段迭代交付

### 2.2.3 开发周期

* 1. 需求分析：2周
  2. 系统设计：3周
  3. 编码实现：8周
  4. 测试验收：3周
  5. 部署上线：2周

### 2.2.4 团队分工

项目经理；前端开发；后端开发；测试工程师；UI设计

## 2.3 可行性分析

### 2.3.1 技术可行性

1. 现有Web开发技术成熟（如Spring Boot、Vue.js等）

2. 移动端适配技术完善

3. 学校网络基础设施完备

4. 可与学校统一身份认证系统对接

### 2.3.2 经济可行性

1. 开发成本可控，可使用开源技术栈

2. 后期维护成本低

3. 可为学校节省部分资源管理成本

4. 可考虑引入适度广告实现部分收益

### 2.3.3 操作可行性

1. 用户群体（师生）信息化素养较高

2. 界面设计符合用户习惯

3. 操作流程简单直观

4. 提供必要的使用指导

### 2.3.4 法律可行性

1. 符合电子商务相关法律法规

2. 符合学校信息化管理规定

3. 用户隐私保护措施完善

4. 交易纠纷处理机制合规

# 三、系统分析

## 3.1 组织与功能分析

### 3.1.1 组织结构分析

系统涉及的主要组织单位：

1. 学校信息中心：提供基础设施支持
2. 学生工作处：参与用户管理
3. 后勤部门：参与线下交易点管理
4. 学生会：协助平台推广与运营

### 3.1.2 系统功能模块

1. 用户管理模块

注册与登录、身份认证、个人信息管理、信用评价

1. 商品管理模块

商品发布、商品搜索与浏览、商品分类管理、商品推荐

1. 交易管理模块

订单生成、交易状态跟踪、支付对接、评价系统

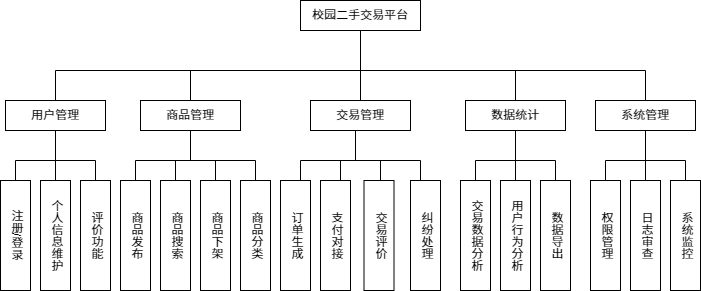
1. 消息管理模块

站内信、系统通知、交易提醒

1. 后台管理模块

用户管理、商品审核、交易监控、数据统计

**（附）功能模型图** ：



图表 1功能图

## 3.2 业务与业务过程分析

### 3.2.1 主要业务

* 1. 商品发布业务
  2. 商品搜索与浏览业务
  3. 交易协商业务
  4. 订单生成与支付业务
  5. 商品交付业务
  6. 评价反馈业务

### 3.2.2 核心业务过程

* 1. 用户注册与认证流程
  2. 商品发布与审核流程
  3. 商品搜索与浏览流程
  4. 交易协商与订单生成流程
  5. 支付与交付确认流程
  6. 评价与信用更新流程

（附）业务流程图：



图表 2业务流程图

### 3.2.3 业务规则

**1）. 商品管理规则**

**商品发布规则**：

用户发布商品时必须填写完整信息，包括商品标题（限制100字符）、商品分类（从预设分类中选择）、价格（需标注是否可议价）、商品描述（限制500字符）和至少上传3张实物图片。

商品价格不得超过原价的80%，若用户标注为"全新未拆封"商品，则需上传购买凭证。

系统自动过滤商品描述中的联系方式（如手机号、微信、QQ等），强制要求通过平台内置聊天工具沟通。

**商品展示规则**：

商品列表默认按发布时间倒序排列，支持按价格、发布时间、浏览量和信用评分等多维度排序。

商品搜索支持关键词模糊匹配和分类筛选，搜索结果分页显示，每页展示20条商品信息。

**商品状态变更规则**：

商品发布后需经过管理员审核，审核通过后方可展示。

卖家可随时下架商品，下架后历史订单仍保留。

商品售出后自动变更为"已售"状态，保留30天后自动归档。

**2）. 交易管理规则**

**订单生成规则：**

买家选择商品后，需确认交易方式（自提/快递）和交易时间（自提商品需约定具体时间地点）。

系统自动生成订单编号（规则： 6位随机数+时间戳），订单状态初始为"待支付"。

**支付规则：**

订单生成后，买家需在15分钟内完成支付，超时未支付订单自动取消。

支付方式仅支持平台集成的第三方支付（微信支付/支付宝），平台不处理现金交易。

支付成功后，系统自动通知买卖双方，并开始计算交易完成时限。

**交易完成规则：**

自提商品：买家需在约定时间内完成提货，提货时扫码确认收货。

快递商品：系统自动同步物流信息，签收后3天内无异议视为交易完成。

交易完成后7天内，买卖双方可互相评价，逾期评价入口关闭。

**3）. 用户管理规则**

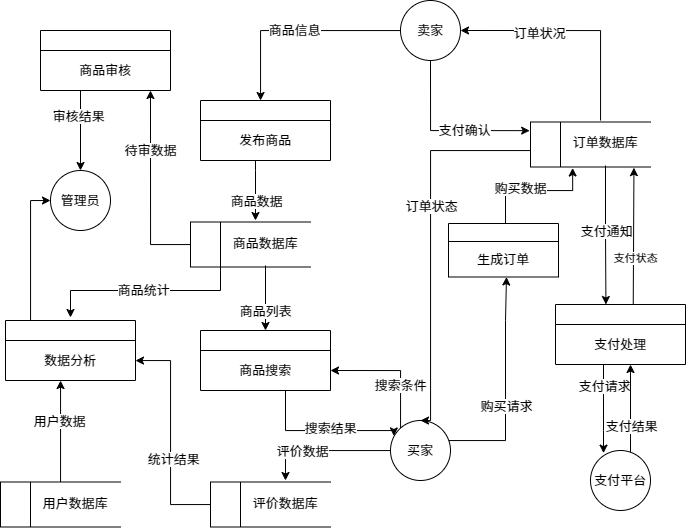
**信用评价规则：**

采用五星评分制（15星），包含商品描述相符度、卖家服务态度、物流服务三个维度。

用户信用分计算公式：（近30天好评数×1 + 中评数×0.5 差评数×1）/总交易数×100。

信用分低于60分的用户将限制部分功能使用（如每日发布商品数量减半）。

（附）数据模型



图表 3数据流程图

## 3.3 信息与信息过程分析

### 3.3.1 主要信息流

1. 用户信息流：注册信息、认证信息、信用信息

2. 商品信息流：商品基本信息、状态信息、分类信息

3. 交易信息流：订单信息、支付信息、交付信息

4. 评价信息流：评价内容、评分、回复

### 3.3.2 数据处理过程

1. 数据采集：用户输入、系统自动生成

2. 数据存储：结构化存储于数据库

3. 数据处理：业务逻辑处理、统计分析

4. 数据展示：界面呈现、报表生成

### 3.3.3 关键数据实体

1. 用户(Student/Teacher)

2. 商品(Product)

3. 订单(Order)

4. 评价(Review)

5. 消息(Message)

### 3.3.4 数据字典（部分）

1. 用户表：用户ID、学号/工号、姓名、联系方式、信用分等

2. 商品表：商品ID、名称、分类、价格、描述、状态等

3. 订单表：订单ID、商品ID、买家ID、卖家ID、状态等

4. 评价表：评价ID、订单ID、评分、内容、时间等

### 3.3.5 核心数据实体

**商品信息实体：**

商品ID（goods\_id）：字符型，20位，系统自动生成的唯一标识

商品标题（title）：字符型，最大长度100，必填

商品分类（category）：枚举型，取自预设分类（教材、体育器材、电子产品等）

价格（price）：数值型，保留2位小数，范围0.019999.99

商品状态（status）：枚举型（1待审核/2在售/3已下架/4已售出/5审核不通过）

**订单信息实体：**

订单ID（order\_id）：字符型，32位，生成规则： 6位随机数+时间戳

支付状态（payment\_status）：枚举型（0待支付/1已支付/2已取消/3已退款）

交易方式（trade\_type）：枚举型（1自提/2快递）

订单创建时间（create\_time）：时间戳，精确到毫秒

**用户评价实体：**

评价ID（comment\_id）：自增整数，主键

评分（score）：整型，15分

评价内容（content）：文本型，最大长度200

是否匿名（is\_anonymous）：布尔型，默认false

### 3.3.6 关键数据流

**商品数据流：**

商品发布数据流：用户→商品管理模块→商品数据库，包含商品基本信息和图片

商品搜索数据流：用户→搜索模块→商品数据库→返回商品列表

**交易数据流：**

订单生成数据流：用户→订单模块→订单数据库

支付通知数据流：支付平台→支付接口→订单数据库

**评价数据流：**

评价提交数据流：用户→评价模块→评价数据库

信用分更新数据流：评价数据库→用户数据库

### 3.3.7 数据约束

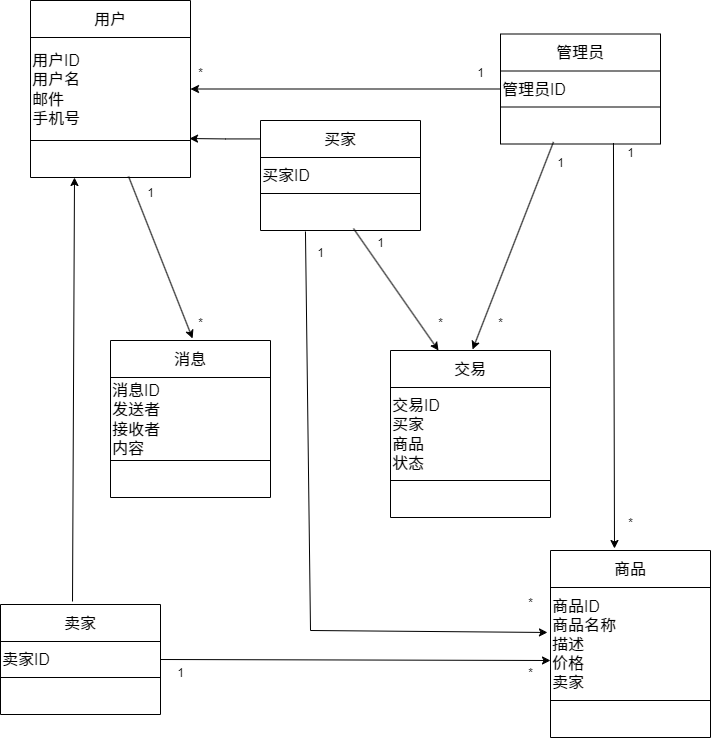
商品图片存储限制：每张不超过5MB，格式仅限JPG/PNG

订单数据保留策略：已完成订单保留5年，已取消订单保留1年

用户评价修改规则：提交后30分钟内可修改，逾期不可编辑

## 3.4 新系统逻辑模型建立

### 3.4.1 对象模型（类图）



图表 4类图

类图通过展示校园二手交易平台中核心类及其相互关系，全面描绘了系统的静态架构，为理解系统组成与功能提供了清晰的蓝图。各部分详细说明如下：

用户类（User）：作为系统最基础的类，封装了用户的核心信息与通用行为，是卖家类（Seller）和买家类（Buyer）的父类，实现了用户注册、登录及个人信息更新等关键功能，为系统中所有用户角色提供了统一的身份验证与信息管理机制。

卖家类（Seller）：继承自用户类，不仅拥有用户类的全部属性和方法，还扩展了与商品发布及管理相关的独特功能。卖家能够通过该类的方法将商品信息录入系统，对已发布的商品进行实时监控和调整，为校园二手交易市场提供了商品供给的核心动力。

买家类（Buyer）：同样继承自用户类，侧重于商品浏览、搜索、下单购买以及交易评价等功能。买家类为用户在平台上的消费行为提供了全方位的支持，从寻找心仪商品到完成交易的全流程覆盖，有效提升了用户的购物体验。

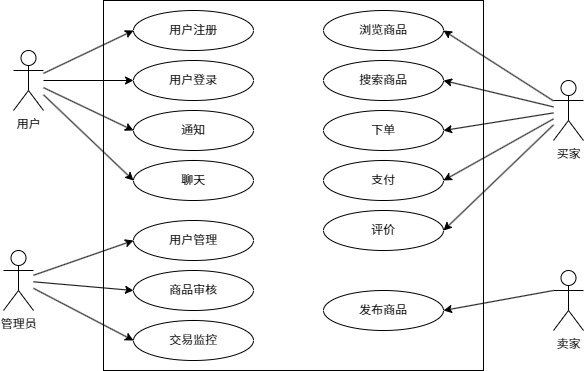
商品类（Product）：承载了商品的详细信息，包括名称、描述、价格、分类等关键属性，为交易提供了必要的商品数据支持。该类还封装了获取和更新商品信息的方法，确保商品数据的准确性和时效性，是连接卖家与买家、促成交易的核心数据纽带。

交易类（Transaction）：作为记录交易过程的关键类，关联了买家、商品等核心实体，涵盖了从下单、支付到收货确认、评价等交易全流程的状态管理。它不仅保障了交易流程的完整性和可追溯性，还通过与支付平台、物流系统的交互，实现了交易的自动化和信息化处理。

消息类（Message）：为用户之间的沟通互动搭建了桥梁，封装了消息的发送、接收以及内容管理等功能。消息类支持用户在交易前的沟通协商、交易中的问题处理以及交易后的反馈交流，极大地增强了平台的社交属性和用户粘性。

管理员类（Admin）：拥有系统的最高管理权限，能够对用户、商品、交易等进行全方位的管控和监督。管理员类通过审核用户注册信息、商品发布内容，监控交易动态，处理售后纠纷以及生成数据分析报表等操作，确保了平台的正常运营和交易秩序，为平台的稳定发展提供了坚实的管理保障。

### 3.4.2 功能模型（用例图）



图表 5用例图

用例图从用户视角出发，通过图形化的方式全面呈现了系统能够为不同角色提供的功能服务，以及各角色与系统之间的交互关系，以下是具体说明：

用户注册与登录：新用户访问平台时，可通过点击注册按钮进入注册页面，填写必要的注册信息完成注册流程。已注册用户则可通过输入用户名和密码等凭证进行登录验证，成功登录后即可享受平台提供的各类服务。这一用例确保了只有合法注册的用户能够访问平台的核心功能，保障了平台用户的身份合法性和数据安全性。

商品管理：卖家作为商品的提供者，拥有发布和管理商品的专属权限。卖家可随时根据个人闲置物品的情况，在系统中创建新的商品条目，输入商品的详细信息并上传图片。同时，卖家能够对已发布的商品进行多维度的管理操作，如修改商品价格、更新商品描述、调整商品库存状态以及下架已售罄或不再出售的商品，确保商品信息的实时性和准确性，满足校园二手交易市场的动态需求。

商品浏览与搜索：买家登录系统后，即可进入商品浏览界面，系统默认以列表形式展示所有在售的二手商品，并提供丰富的筛选条件，如商品分类、价格区间、新旧程度等，帮助买家快速定位到感兴趣的商品范围。此外，买家还可利用搜索框输入商品名称、关键词等信息进行精准搜索，系统将实时返回匹配的商品结果，提高买家查找商品的效率，优化购物体验。

交易管理：买家在确定购买意向后，可点击商品页面的购买按钮发起交易请求，系统将引导买家完成订单创建、支付等操作，并在交易过程中提供订单状态查询、物流信息跟踪等功能。交易完成后，买家有权对卖家及商品进行评价，评价结果将影响卖家的信誉评分。卖家则可在系统中查看订单详情、管理发货状态、处理售后申请等，确保交易流程的顺畅进行，维护良好的买卖双方关系。

消息沟通：平台为用户提供信息交流渠道，方便用户在交易前后进行沟通协商。用户可发送站内信、留言或通过聊天工具进行实时沟通，交流内容涵盖商品详情咨询、交易细节确认、售后问题处理等各个方面。系统将对消息进行实时传输和存储，确保信息的及时性和完整性，为交易双方搭建起高效沟通的桥梁，提升平台的互动性和用户满意度。

数据统计与分析：管理员借助系统提供的数据分析功能，能够对平台的交易数据、用户行为数据等进行深度挖掘和分析。系统以直观的图表形式（如柱状图、折线图、饼图等）展示各类统计数据，如商品交易量趋势、热门商品排行、用户活跃度分析等，帮助管理员全面了解平台的运营状况和市场动态。基于这些数据，管理员可制定更有针对性的运营策略，优化平台功能和服务，为平台的持续发展提供数据支持和决策依据。

# 四、系统设计

## 4.1 系统体系结构设计

### 4.1.1 总体架构

本二手交易平台采用浏览器 / 服务器（B/S）架构模式。客户端只需安装通用的浏览器（如 Chrome、Firefox、Safari 等），无需安装额外的软件，用户通过浏览器访问系统提供的 Web 界面进行操作。服务器端负责处理业务逻辑、数据存储与管理等核心任务。这种架构具有良好的跨平台特性，用户可以在多种操作系统（Windows、macOS、Linux、Android、iOS 等）上使用，降低了对客户端环境的依赖性，便于系统的推广和使用。

### 4.1.2 层次结构

**1. 表示层（用户界面层）**

用户通过浏览器与系统进行交互，所有的用户请求和操作都在此层发起。前端界面采用 HTML5、CSS3 和 JavaScript 技术进行构建，结合响应式设计，确保在不同设备（桌面电脑、平板电脑、手机等）上都能获得良好的显示效果和用户体验。同时，为了提高用户体验，使用了一些前端框架和库，如 Vue.js 或 React.js 等，实现页面的动态加载、交互效果和数据绑定等功能。

界面设计注重简洁性、直观性和易用性，以降低用户的学习成本。例如，在商品发布页面，提供简单易懂的表单，引导用户填写商品信息；在商品列表页面，采用清晰的布局和分类展示，方便用户快速浏览和筛选商品。

**2. 业务逻辑层**

位于表示层和数据访问层之间，是系统的核心部分。它负责处理系统的业务逻辑，包括用户注册与登录、商品管理（发布、编辑、查询、下架等）、交易流程处理（下单、支付、评价等）、消息通知、用户权限管理等。该层采用面向对象的编程语言（如 Java、Python 等）编写，通过代码实现各种业务规则和流程。

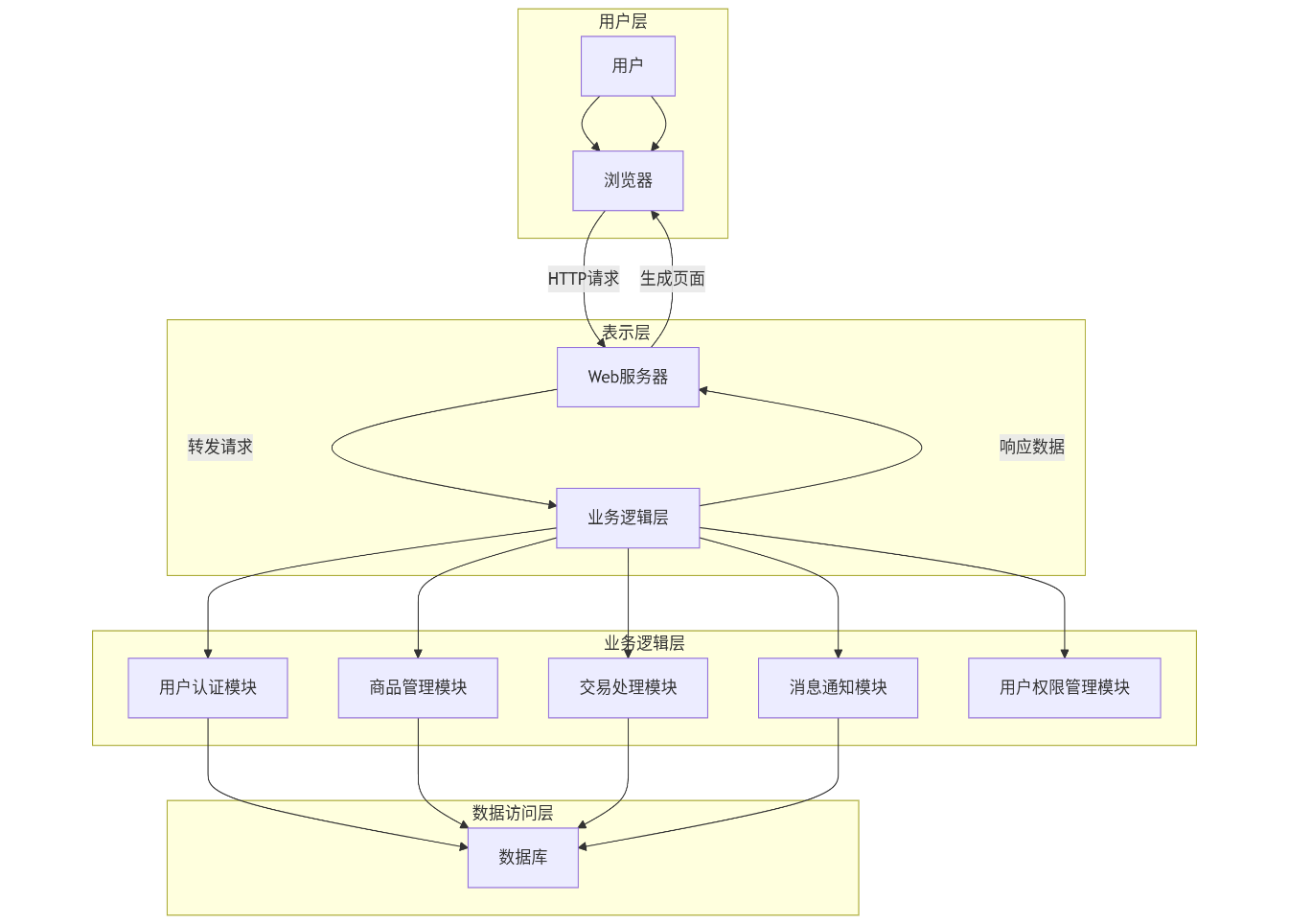
为了提高系统的可维护性和可扩展性，将业务逻辑划分为多个模块，每个模块对应特定的业务功能。例如，用户认证模块负责用户身份验证和权限管理；商品管理模块处理与商品相关的一切操作；交易处理模块管理交易的各个环节。模块之间通过定义良好的接口进行通信和协作，降低了模块之间的耦合度。

在业务逻辑层中，还引入了一些设计模式，如 MVC（Model - View - Controller）模式，将业务逻辑与用户界面分离，提高了代码的组织性和可读性；策略模式用于处理不同的业务算法，如不同的商品搜索策略、用户评价算法等，使得系统能够灵活地应对业务需求的变化。

**3. 数据访问层**

数据访问层负责与数据库进行交互，实现数据的持久化存储和检索。服务器端使用关系型数据库（如 MySQL、PostgreSQL 等）来存储系统中的各类数据，包括用户信息、商品信息、订单信息、交易记录等。通过编写 SQL 语句或者使用 ORM（Object - Relational Mapping）框架（如 Hibernate、Django ORM 等），将业务逻辑层中的数据操作请求转换为对数据库的操作，实现数据的增删改查等功能。

为了保证数据的安全性和一致性，在数据访问层采用了事务机制。对于涉及多个数据表更新的复杂操作（如订单创建、商品库存更新等），通过事务的 ACID（原子性、一致性、隔离性、持久性）特性，确保数据在操作过程中的完整性。同时，对数据库进行定期备份和恢复测试，以防止数据丢失。



图表 6层次结构图

### 4.1.3 系统交互流程

当用户在浏览器中输入系统的网址并访问时，首先由服务器端的 Web 服务器接收请求。Web 服务器根据请求的 URL 和方法，调用相应的后端业务逻辑模块进行处理。业务逻辑模块在处理过程中，可能会向数据访问层发送数据查询或更新的请求，数据访问层从数据库中获取数据或执行数据更新操作后，将结果返回给业务逻辑模块。业务逻辑模块根据处理结果生成相应的 HTML 页面内容或其他响应数据（如 JSON 格式的数据），通过 Web 服务器返回给用户的浏览器。浏览器接收到响应后，解析并渲染 HTML 页面，将结果展示给用户。

例如，当用户登录系统时，用户在浏览器中输入用户名和密码，点击登录按钮后，浏览器向服务器发送登录请求。服务器端的用户认证模块接收到请求后，调用数据访问层查询用户信息，验证用户名和密码是否正确。如果验证成功，服务器生成一个会话信息（Session）并返回给浏览器，同时将响应结果封装为 HTML 页面或 JSON 数据，其中包含用户的个人信息和系统主页面的导航链接等。浏览器收到响应后，渲染登录成功后的页面，用户就可以在系统中进行后续的操作，如浏览商品、发布商品等。

### 4.1.4 技术选型

**1. Web 服务器**

选择 Nginx作为 Web 服务器。这些服务器具有高性能、稳定性强、能够处理大量并发请求的特点，能够有效地将用户的请求分发到后端的应用服务器进行处理，并将处理结果返回给用户。

**2. 应用服务器**

根据系统的开发语言，选择合适的应用服务器。例如，如果使用 Java 开发，可以选用 Tomcat、Jetty 等；如果使用 Python 开发，则可选择 Gunicorn、uWSGI 等。应用服务器负责运行系统的业务逻辑代码，处理用户的请求并生成响应。

**3. 缓存技术**

为了提高系统的性能和响应速度，在系统中引入了缓存技术。使用 Redis 或 Memcached 等缓存工具，将一些频繁访问的数据（如热门商品信息、用户登录信息等）缓存到内存中，减少对数据库的访问次数。当用户请求这些数据时，优先从缓存中获取，大大缩短了响应时间。

**4. 消息队列**

为了实现异步处理和提高系统的吞吐量，引入了消息队列（如 RabbitMQ、Kafka 等）。例如，在用户下单后，订单信息通过消息队列发送给后端的订单处理模块和库存更新模块，这两个模块可以并行处理订单和更新库存，提高了系统的效率，避免了因同步处理导致的用户等待时间过长的问题。

**5. 数据库**

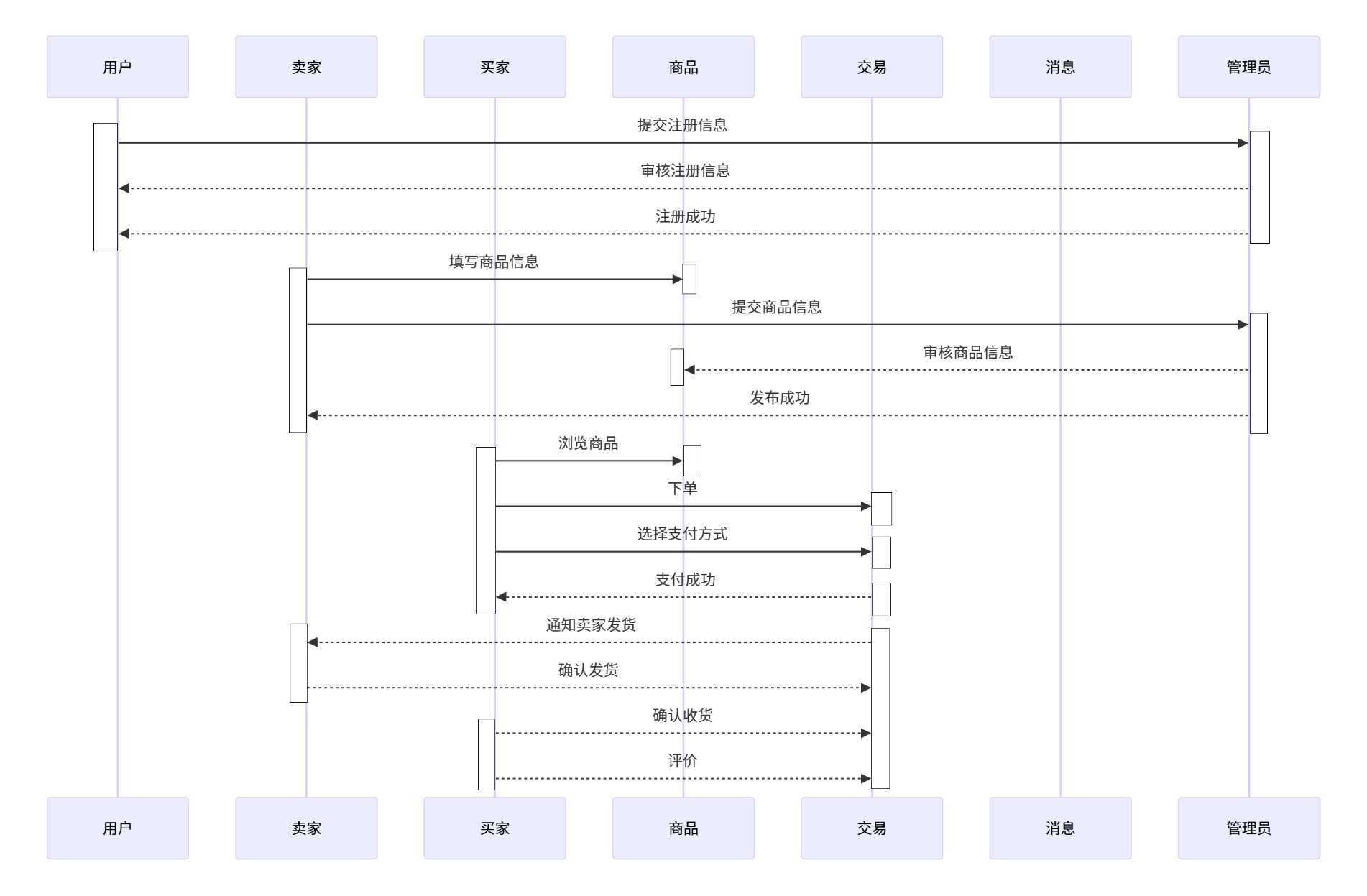
如前面所述，选用关系型数据库来存储系统数据。关系型数据库具有良好的数据一致性和复杂查询能力，能够满足系统对数据存储和检索的需求。同时，为了提高数据库的读写性能，可以对数据库进行优化，如创建索引、分表分库等操作。

### 4.1.5 架构优势

1. 易于维护和升级：由于业务逻辑集中在服务器端，系统的更新和维护只需在服务器端进行，无需对客户端进行任何修改和重新部署。例如，当需要对商品分类规则进行修改时，只需在服务器端更新相应的业务逻辑代码和数据库配置，用户下次访问系统时就能获取最新的分类信息，大大降低了维护成本。
2. 良好的可扩展性：B/S 架构能够方便地应对用户数量的增加和业务规模的扩大。通过增加服务器的硬件资源（如 CPU、内存、磁盘等）、优化服务器配置（如负载均衡、集群部署等），可以轻松地提高系统的处理能力和并发量，以满足不断增长的需求。
3. 跨平台性强：用户可以使用任何支持浏览器的设备访问系统，无需安装特定的客户端软件。这使得系统能够适应多种操作系统和设备类型，扩大了系统的用户群体，提高了系统的可用性和普及度。
4. 便于资源共享：服务器端集中存储和管理数据，便于对数据进行统一的安全控制和备份恢复。同时，系统中的公共资源（如商品图片、用户评价等）也可以在服务器端进行共享，减少了资源的重复存储和管理成本。

## 4.2 系统运行处理流程设计

### 4.2.1 顺序图



图表 7顺序图

### 4.2.2系统流程

顺序图通过描绘系统中不同对象在特定业务流程中的交互时序，生动展现了系统的动态行为和协作机制，是对系统运行时状态的精准刻画，以下是详细说明：

用户注册流程：新用户通过注册页面提交包含用户名、密码、邮箱等关键信息的注册请求，系统接收到请求后，首先对信息的完整性和格式进行自动验证。验证通过后，将注册信息发送至管理员处进行人工审核。管理员依据平台的注册规则和审核标准，判断注册信息是否合法合规。若审核通过，系统将向用户发送注册成功的通知，并自动创建用户账户，赋予用户相应的基础权限；反之，若审核不通过，系统则向用户发送详细的注册失败原因说明，引导用户修改信息后重新提交注册申请。

商品发布流程：卖家登录系统后，进入商品发布页面，按照系统提供的模板填写商品的详细信息，包括商品名称、描述、价格、分类、新旧程度等，并上传展示商品外观和细节的图片。填写完成后，卖家提交商品发布请求。系统对提交的商品信息进行初步的格式和完整性校验，校验通过后，将商品信息发送至管理员处进行内容审核。管理员重点审核商品信息是否真实准确、是否符合平台的交易规则以及是否包含违规内容。审核通过的商品将被正式发布到平台上，供买家浏览和购买；若审核不通过，系统将向卖家反馈具体的审核意见，卖家可根据意见修改商品信息后再次提交发布请求。

交易流程：买家在平台上浏览商品列表或通过搜索功能找到心仪的二手商品后，可选择商品并进入商品详情页面查看详细信息。确认购买意向后，买家点击购买按钮，系统自动生成订单，订单初始状态为“待付款”。买家需在系统设定的15分钟时间内选择支付方式（如微信支付、支付宝等）并完成支付操作。支付成功后，系统实时通知卖家有新的订单待处理，同时将订单状态更新为“待发货”。卖家收到通知后，需在规定时间内根据订单信息安排发货，发货后更新订单物流信息，此时订单状态变为“待收货”。买家收到商品并确认无误后，在系统中进行确认收货操作，订单状态最终更新为“已完成”，交易流程至此结束。若买家在收货过程中发现商品存在问题，可发起售后申请，系统将根据具体的售后规则和流程进行处理。

## 4.3 系统功能与模块关系设计

### 4.3.1模块划分细化

将子系统进一步分解为更具体的功能模块（或类/包），划分如下：

1）用户管理模块

用户注册与登录子模块：处理新用户的注册请求和已有用户的登录验证。

身份认证子模块：对用户进行身份认证，如学生证、身份证等信息的验证。

个人信息管理子模块：允许用户查看和编辑自己的个人信息。

信用评价子模块：记录和管理用户的信用评分和评价信息。

2）商品管理模块

商品发布子模块：负责处理用户发布新商品的信息录入和初步审核。

商品搜索与浏览子模块：提供搜索功能和商品列表展示，支持多种筛选条件。

商品分类管理子模块：管理商品的分类信息，方便用户进行分类浏览。

商品推荐子模块：根据用户的历史行为和偏好，推荐相关的商品。

3）交易管理模块

订单生成子模块：在用户确定购买意向后，生成相应的订单信息。

交易状态跟踪子模块：实时更新和跟踪交易订单的状态变化。

支付对接子模块：与第三方支付平台进行对接，处理支付相关操作。

评价体系子模块：允许买卖双方在交易完成后进行互相评价。

4）消息管理模块

站内信子模块：用户之间可以通过站内信进行一对一的消息交流。

系统通知子模块：系统向用户发送重要通知，如系统维护、活动信息等。

交易提醒子模块：在交易的关键节点，如订单生成、支付成功、发货通知等，向相关用户发送提醒消息。

5）后台管理模块

用户管理子模块：管理员对用户信息进行审核、查看和管理。

商品审核子模块：对用户发布的商品进行审核，确保符合平台规定。

交易监控子模块：监控交易的全过程，及时发现和处理异常交易。

数据统计子模块：对平台的各项数据进行统计分析，生成报表供管理员参考。

### 4.3.2 模块职责定义

明确每个模块的核心功能如下：

1）用户管理模块

负责用户在平台上的身份信息管理，包括注册、登录、身份认证以及个人信息的维护，同时记录用户的信用评价信息，为交易提供参考。

2）商品管理模块

涉及商品的发布、展示、分类和推荐等功能，确保用户能够方便快捷地找到自己需要的商品，并为卖家提供有效的商品管理工具。

3）交易管理模块

主要处理订单的生成、交易状态的跟踪、支付的对接以及交易后的评价等操作，保障交易流程的顺利进行和交易双方的权益。

4）消息管理模块

为用户提供深度的交流渠道，包括站内信、系统通知和交易提醒等功能，及时传递重要信息，促进用户之间的沟通和交易的顺利完成。

5）后台管理模块

提供用户管理、商品审核、交易监控和数据统计等功能，帮助管理员有效地管理平台的运营，确保平台的正常运转和健康发展。

### 4.3.3 模块间接口定义

定义模块之间如何交互如下：

1）用户管理模块与其他模块接口

与商品管理模块接口：在商品发布时，验证用户身份是否合法，是否具备发布商品的权限。传递用户身份信息和信用评价信息，以便商品管理模块对商品的展示和推荐进行相应的处理。

与交易管理模块接口：在订单生成时，提供用户的身份信息和信用状况，帮助判断交易的风险程度。在交易评价时，接收评价信息并更新用户的信用评价数据。

与消息管理模块接口：当用户注册、登录或个人信息发生变化时，触发消息管理模块向用户发送相关的欢迎消息、验证消息或提示消息。

2）商品管理模块与其他模块接口

与交易管理模块接口：在订单生成时，提供商品的详细信息，包括商品价格、库存等，以便交易管理模块创建准确的订单。在交易完成后，通知商品管理模块更新商品的销售状态和库存数量。

与消息管理模块接口：当有新的商品发布或商品推荐时，向用户发送相关的消息通知，吸引用户关注和购买。

3）交易管理模块与其他模块接口

与支付对接模块接口：在用户选择支付方式后，调用支付对接模块提供的接口，将订单信息传递给第三方支付平台，并接收支付结果反馈，更新订单状态。

与消息管理模块接口：在交易的各个关键节点，如订单生成、支付成功、发货通知、交易完成等，调用消息管理模块的接口，向买卖双方发送相应的提醒消息和通知消息，确保双方及时了解交易进度。

4）消息管理模块与其他模块接口

与用户管理模块接口：根据用户的身份信息和偏好设置，定制消息的发送内容和方式，确保消息能够准确地传达给目标用户。

与商品管理模块和交易管理模块接口：接收来自这两个模块的消息发送请求，根据请求内容生成相应的消息，并按照指定的方式发送给用户。

5）后台管理模块与其他模块接口

与用户管理模块接口：管理员可以通过后台管理模块查看和管理所有用户的信息，包括用户注册信息、身份认证状态、信用评价等。同时，可以对用户的权限进行调整和管理。

与商品管理模块接口：后台管理模块能够对商品的发布信息进行审核，查看商品的详细信息和销售情况，对违规商品进行下架处理等。

与交易管理模块接口：可以监控交易的全过程，查看订单的详细信息和交易状态，处理交易纠纷和投诉等。

与数据统计模块接口：收集和整理平台的各项数据，生成各类统计报表，为管理员提供决策支持。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **接口名称** | **函数/方法名** | **参数** | **返回值** | **功能描述** |
| 用户身份验证接口 | authenticateUser | 用户名，密码 | 布尔值，用户信息 | 验证用户身份是否合法，返回验证结果及用户详细信息 |
| 商品信息查询接口 | queryProduct | 商品ID | 商品对象 | 根据商品ID查询商品的详细信息 |
| 订单创建接口 | createOrder | 用户ID，商品ID，数量 | 订单对象 | 创建一个新的订单，并返回订单相关信息 |
| 支付接口 | payOrder | 订单ID，支付方式 | 支付结果状态 | 调用第三方支付平台进行支付操作，并返回支付结果 |
| 消息发送接口 | sendMessage | 发送者ID，接收者ID，消息内容 | 发送结果状态 | 向指定用户发送消息，并返回发送是否成功的结果 |

图表 8接口定义图

### 4.3.4 模块依赖关系

描述模块之间的调用或依赖关系如下：

**用户管理模块**是整个系统的基础模块，为其他模块提供用户身份认证和信用评价等关键数据支持。商品管理模块依赖用户管理模块来验证商品发布者的身份合法性；交易管理模块依赖用户管理模块获取交易双方的用户信息和信用状况，以确保交易的安全性和可靠性；消息管理模块根据用户管理模块提供的用户信息进行消息的精准推送；后台管理模块对用户管理模块进行全面的监控和管理，实现用户权限的分配和控制。

**商品管理模块**为交易管理模块提供商品的基本信息和库存状态，交易管理模块根据商品信息生成订单并处理交易流程。同时，消息管理模块依据商品管理模块中的商品动态信息，向用户推送相关的商品推荐和上下架通知等消息。后台管理模块对商品管理模块进行审核和监管，确保商品信息的真实性和合法性，维护平台的商品质量。

**交易管理模块**是系统的核心模块之一，与多个模块存在紧密的依赖关系。它依赖支付对接模块完成交易的支付操作，依赖消息管理模块向用户发送交易相关的提醒和通知消息，确保交易流程的顺利进行和用户及时了解交易状态。后台管理模块对交易管理模块进行全面的监控和数据分析，以便及时发现和处理交易中的异常情况，同时根据交易数据为平台的运营决策提供支持。

**消息管理模块**作为信息传递的枢纽，与用户管理模块、商品管理模块和交易管理模块等多个模块相互协作。它依赖用户管理模块获取用户的身份和联系方式等信息，确保消息能够准确无误地发送给目标用户；根据商品管理模块和交易管理模块的业务动态，生成并发送相应的消息通知，保障用户能够及时获取平台的各项重要信息。

**后台管理模块**作为系统的管理中枢，对其他所有模块具有管理和监控的职责。它依赖各个业务模块提供的数据进行综合分析和决策制定，同时对各模块的运行状态进行实时监控，确保系统的稳定运行和业务的正常开展。其他模块在运行过程中产生的数据和日志等信息会反馈给后台管理模块，以便进行数据统计、性能评估和问题排查等工作。

## 4.4 系统物理配置方案设计

### 4.4.1硬件选型清单

1. **服务器** ：Dell PowerEdge R750 服务器，CPU 采用 2 颗 Intel Xeon Gold 6338 处理器，每颗处理器具备 32 核心 64 线程，主频为 2.1GHz，可睿频至 3.9GHz；内存配置为 1TB DDR5 RDIMM 4800MT/s 内存，支持热插拔；硬盘配备 2 块 1.92TB U.2 7mm 3.84TB NVMe 固态硬盘，可满足系统对快速读写的需求，同时后续可通过添加硬盘笼进行扩展，支持高达 48 块 3.5 英寸热插拔硬盘，保障系统存储容量的弹性扩展能力，适用于承载校园二手交易平台的高并发访问与数据存储需求。
2. **网络设备** ：采用华为 AR6300 路由器，具备高性能的报文转发能力和丰富的协议支持，能够满足校园网络的复杂路由需求，确保数据包的高效传输。搭配华为 S6730 系列交换机，支持高密度的万兆接口以及多种先进的网络功能，如虚拟交换矩阵、MDC（多设备管理）等，可实现网络流量的灵活调度和设备的集中管理，为服务器之间以及服务器与外部网络的通信提供高速、稳定的网络连接。防火墙选用具备深度防御体系的天融信 AC8000-1000 型号，整机吞吐量可达 1000Mbps，具备防火墙、IPS、AV、DPI 等多种功能，能够有效抵御网络攻击和恶意流量的侵入，保障校园二手交易平台的安全稳定运行。
3. **存储设备** ：采用华为 OceanStor 5300 V3 中端磁盘阵列作为存储设备，支持 SAN 和 NAS 一体化存储，提供高可靠的存储服务。该设备具备双控架构，每个控制器配备 16GB 缓存，可支持最多 384 块硬盘，满足系统对存储容量和读写性能的要求，适用于存储平台的用户数据、交易记录、商品图片等海量数据，确保数据的安全存储与快速读取。
4. **备份设备** ：使用联想 ThinkSystem SR850 服务器作为备份服务器，搭载 1 颗 Intel Xeon Gold 6230 处理器，16GB 内存，配备 2 块 1TB 7.2K RPM SATA 3.5 英寸硬盘，能够满足数据备份的基本需求。采用磁带备份方式，选用 IBM LTO-9 磁带机及磁带，LTO-9 技术可提供高达 45TB 的未压缩容量和 180TB 的压缩容量（2:1 压缩比），传输速率最高可达 400MB/s，能够实现高效的数据备份和长期归档，为校园二手交易平台的数据提供可靠的冷备份保障。

### 4.4.2软件选型清单

1. **操作系统** ：选用 Ubuntu Server 22.04.2 LTS 64 位操作系统，该版本具有良好的稳定性、安全性以及强大的社区支持，能够满足校园二手交易平台服务器端的运行环境需求。它提供了丰富的软件包资源和灵活的配置选项，方便进行系统开发、部署和维护工作。
2. **数据库管理系统** ：采用 MySQL 8.0.30 社区版数据库管理系统，该版本具有高性能、高可用性和良好的兼容性，支持丰富的 SQL 功能和事务处理机制，能够满足平台对数据存储、查询和更新的需求。通过合理的数据库设计和优化，可确保系统的数据读写效率和数据一致性。
3. **中间件** ：选用 Redis 7.0.12 作为缓存中间件，能够提供高性能的键统的整体性能。同时，考虑到平台对搜索功能的需求，选用 Elasticsearch 8.6.1 作为搜索引擎中间件，其强大的全文检索能力和高效的索引机制能够为用户提供流量的搜索服务，帮助用户快速精准地找到所需的二手商品信息。
4. **开发工具** ：前端开发采用 Visual Studio Code 1.83.1 版本，该编辑器功能强大、插件丰富，支持多种前端开发语言和框架，如 HTML、CSS、JavaScript、Vue.js 等，能够为前端开发人员提供高效的代码编辑、调试和测试环境。后端开发选用 IntelliJ IDEA 2022.3.3 版本，集成了丰富的开发工具和插件，支持 Java 语言以及 Spring Boot 等主流框架的开发，提供了智能代码提示、自动补全、代码分析等功能，有助于提高后端开发的效率和代码质量。
5. **监控工具** ：采用 Prometheus 2.44.0 作为监控系统核心组件，搭配 Grafana 10.0.2 作为可视化展示工具，可实现对服务器的 CPU 使用率、内存占用、磁盘 I/O、网络流量等关键指标的实时监控和可视化展示。同时，结合 ELK Stack（Elasticsearch 8.6.1、Logstash 8.6.1、Kibana 8.6.1）进行日志的收集、解析和分析，通过 Kibana 可直观地查看日志信息，及时发现系统中的异常情况和潜在问题，为校园二手交易平台的稳定运行提供全方位的监控保障。
6. 值对存储服务，有效缓解数据库的压力，提升系统的响应速度。在高并发的校园二手交易平台场景下，Redis 可用于存储热点数据、会话信息等

### 4.4.3网络配置方案

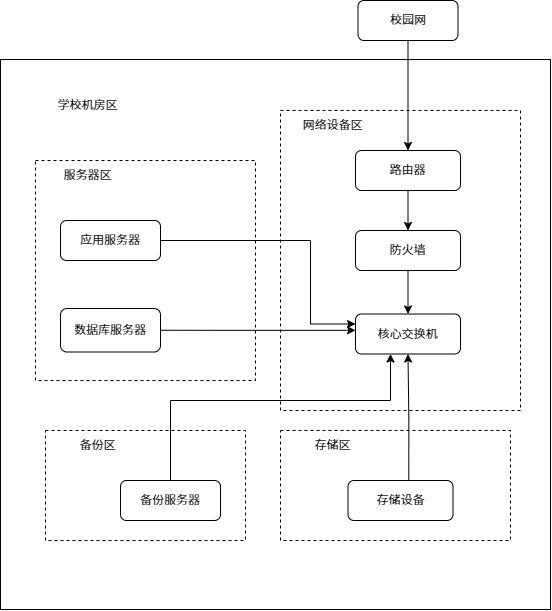
1. **带宽要求** ：考虑到校园二手交易平台预计的用户规模和并发量，出口带宽建议达到 1000Mbps，以满足师生在高峰时段的并发访问需求，确保用户能够流畅地浏览商品信息、进行交易操作等。同时，内部网络带宽应满足服务器之间的数据交互需求，建议采用万兆以太网连接服务器与存储设备，以保障数据传输的高效性。
2. **安全区域划分** ：将网络划分为互联网接入区、业务区和管理区等安全区域。互联网接入区主要负责与外部网络的通信，部署防火墙设备，对进入校园网络的流量进行严格过滤和监控，仅允许合法的用户请求通过。业务区部署应用服务器、数据库服务器等核心业务设备，负责处理平台的业务逻辑和数据存储。管理区则用于部署管理服务器、备份服务器等设备，供管理员进行系统的运维管理和数据备份恢复操作。通过安全区域的划分，实现不同安全级别的网络隔离，降低网络风险，保障平台的安全运行。

### 4.4.4部署拓扑详图

系统采用基于校园机房的物理部署方案，以下是详细的部署拓扑：

1. 在校园信息中心机房中，设置一个专门的机柜用于部署校园二手交易平台的硬件设备。
2. 网络设备：将华为 AR6300 路由器置于机柜顶部，连接校园网络的核心交换机，实现平台与外部网络的通信。在其下方部署华为 S6730 系列交换机，连接防火墙设备（天融信 AC8000-1000），通过防火墙对进出业务区的流量进行安全防护。然后利用交换机分别与应用服务器、数据库服务器、存储设备（华为 OceanStor 5300 V3）等硬件设备相连，构建内部高速局域网络，确保各设备之间的数据交互高效稳定。
3. 服务器部署：应用服务器（Dell PowerEdge R750）部署在机柜中间位置，用于运行平台的前端和后端应用程序，处理用户的请求和业务逻辑。数据库服务器紧邻应用服务器部署，负责存储平台的各类数据，如用户信息、商品数据、交易记录等。存储设备（华为 OceanStor 5300 V3）位于机柜底部，通过光纤通道与数据库服务器相连，为平台提供高容量、高性能的存储服务。
4. 备份设备：备份服务器（联想 ThinkSystem SR850）与磁带机（IBM LTO-9）部署在机柜的另一侧，通过网络与应用服务器和数据库服务器相连，定期进行数据备份操作，备份数据可通过磁带机进行离线存储，实现数据的多重保护。
5. 云服务区域：由于本次设计主要针对物理机房部署，暂不涉及云服务区域的部署。但考虑到平台未来的发展需求和扩展性，建议预留与校园云平台的对接接口，以便在必要时将部分业务迁移到云端，充分利用云计算资源的弹性和可扩展性。

以下是物理部署拓扑图：



图表 9物理部署拓扑图

### 4.4.5容量规划

1. **用户量估算** ：假设武汉体育学院有师生约 20000 人，预计校园二手交易平台的用户渗透率达到 60%，则平台的注册用户量约为 12000 人。
2. **并发量估算** ：根据用户行为分析，平台的并发量主要集中在课余时间和周末等时段。考虑到高峰时段可能有 20% 的注册用户同时在线进行浏览、交易等操作，预计并发量约为 2400 人。
3. **数据量估算**
   1. 用户数据：每位用户的基本信息（如用户名、联系方式、信用评价等）预计占用存储空间约 2KB，12000 用户则占用约 24MB 空间。
   2. 商品数据：假设每件商品平均有 5 张图片，每张图片压缩后约为 200KB，商品描述等文本信息约 5KB，每件商品数据量约为 1005KB。若平台每月发布的商品数量为 5000 件，每年预计发布 60000 件商品，商品数据总量约为 60000×1005KB=6030000KB=5889MB，约为 5.7GB。
   3. 交易数据：每笔交易记录包含交易双方信息、商品信息、支付信息等，预计占用存储空间约 5KB。假设每月交易量为 3000 笔，每年交易量为 36000 笔，交易数据总量约为 36000×5KB=180000KB=175.78MB。
   4. 日志数据：系统运行日志、用户操作日志等预计每天产生约 100MB 数据量，每月约为 3GB，每年约为 36GB。
4. 综上，平台每年的数据量增长预计约为 5.7GB（商品数据）+ 0.175GB（交易数据）+ 36GB（日志数据）+ 其他数据≈42GB 左右，为保障系统性能和数据安全，建议存储设备预留足够的存储空间以满足未来 3 - 5 年的数据增长需求。
5. **硬件资源需求估算**
   1. **CPU** ：根据并发量估算，采用并发量/单核处理能力的公式进行初步计算。假设单核 CPU 能够处理 50 个并发请求，平台在高峰时段 2400 并发量需要的 CPU 核心数为 2400/50=48 核。考虑到实际业务场景中可能存在复杂的业务逻辑处理和一定的性能冗余需求，选择 2 颗 Intel Xeon Gold 6338 处理器（共 64 核）的服务器配置，能够满足系统对 CPU 的需求。
   2. **内存** ：前端应用服务器运行 Vue.js 等前端框架以及后端 Spring Boot 应用，结合并发量和应用内存占用经验估算，前端服务器内存需求约为 8GB，后端服务器内存需求约为 16GB。数据库服务器运行 MySQL 数据库，根据数据量和缓存需求估算，内存需求约为 32GB。综合考虑，配置服务器内存为 1TB，能够满足系统运行过程中对内存的高要求，确保数据的快速读写和缓存处理。
   3. **存储** ：根据上述数据量估算，平台每年数据增长约 42GB，3 - 5 年后的数据量预计为 126 - 210GB 左右。考虑到存储设备的磁盘阵列配置以及数据冗余备份等因素（如 RAID 5 配置，一般占用约 1/3 的存储空间用于冗余），实际需要的存储容量约为 378 - 630GB 左右。选择华为 OceanStor 5300 V3 存储设备，初始配置 2 块 3.84TB 磁盘，可满足未来 3 - 5 年的数据存储需求，并且后续可通过添加硬盘进行扩展。
   4. **带宽** ：在并发量 2400 人的情况下，假设每个并发用户平均占用带宽为 50KB/s（主要考虑用户浏览商品图片、访问页面等操作），则总带宽需求约为 2400×50KB/s=120000KB/s=120MB/s，换算为网络带宽约为 960Mbps。为确保平台在高峰时段的流畅访问，出口带宽建议达到 1000Mbps，同时内部网络采用万兆以太网连接，满足服务器与存储设备之间的高速数据交互需求。

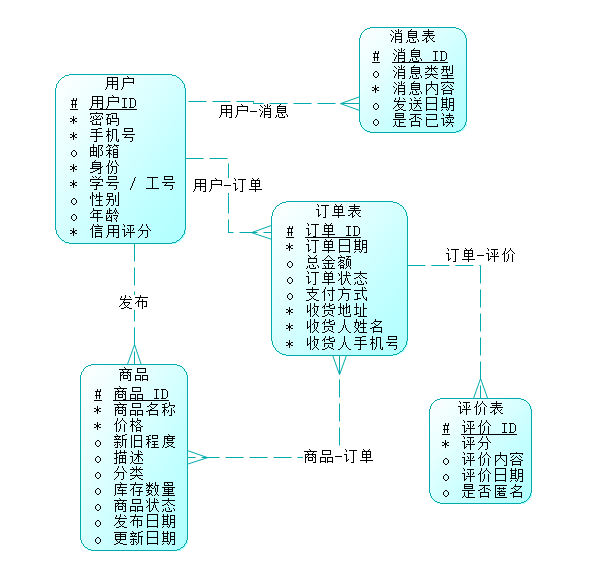
## 4.5 数据库设计

### 4.5.1概念数据模型（CDM）

概念数据模型（CDM）是从用户的角度对数据进行建模，主要描述了系统的业务概念和业务规则。在校园二手交易平台中，CDM 包括以下内容：

1. **实体**
   1. **用户（User）** ：唯一标识每个用户的属性是用户 ID，还包括用户名、密码、手机号、邮箱、身份（学生 / 教职工）、学号 / 工号、性别、年龄、信用评分等属性。
   2. **商品（Product）** ：唯一标识每件商品的属性是商品 ID，包括商品名称、价格、新旧程度、描述、分类、库存数量、状态（在售 / 已售 / 下架）、发布日期、更新日期等属性。
   3. **订单（Order）** ：唯一标识每个订单的属性是订单 ID，包括订单日期、总金额、订单状态（待付款 / 待发货 / 待收货 / 已完成 / 已取消）、支付方式、收货地址、收货人姓名、收货人手机号等属性。
   4. **评价（Review）** ：唯一标识每个评价的属性是评价 ID，包括评分、评价内容、评价日期、是否匿名等属性。
   5. **消息（Message）** ：唯一标识每条消息的属性是消息 ID，包括消息类型（站内信 / 系统通知 / 交易提醒）、消息内容、发送日期、发送者 ID、接收者 ID、是否已读等属性。
2. **关系**
   1. **用户 - 商品（发布）** ：一个用户可以发布多件商品，一件商品只能由一个用户发布。
   2. **用户 - 订单（买家）** ：一个用户可以下多个订单，一个订单只能由一个用户下单。
   3. **用户 - 订单（卖家）** ：一个用户可以有多个作为卖家的订单，一个订单只能由一个用户作为卖家。
   4. **商品 - 订单（包含）** ：一个商品可以属于多个订单（如果商品有库存），一个订单可以包含多件商品。
   5. **订单 - 评价（关联）** ：一个订单可以有多个评价（如买家对卖家的评价、卖家对买家的评价），一个评价只能关联一个订单。
   6. **用户 - 消息（接收）** ：一个用户可以接收多条消息，一条消息只能由一个用户接收（如果是系统通知，可能发送给多个用户，但这里简化为一条消息一个接收者）。

以下是概念数据模型（CDM）图：



图表 10概念数据模型（CDM）图

### 4.5.2逻辑数据模型（LDM）

逻辑数据模型（LDM）是在 CDM 的基础上，进一步细化和规范化数据结构，确定实体及其属性和关系，并为每个关系确定主键和外键。以下是 LDM 的详细内容：

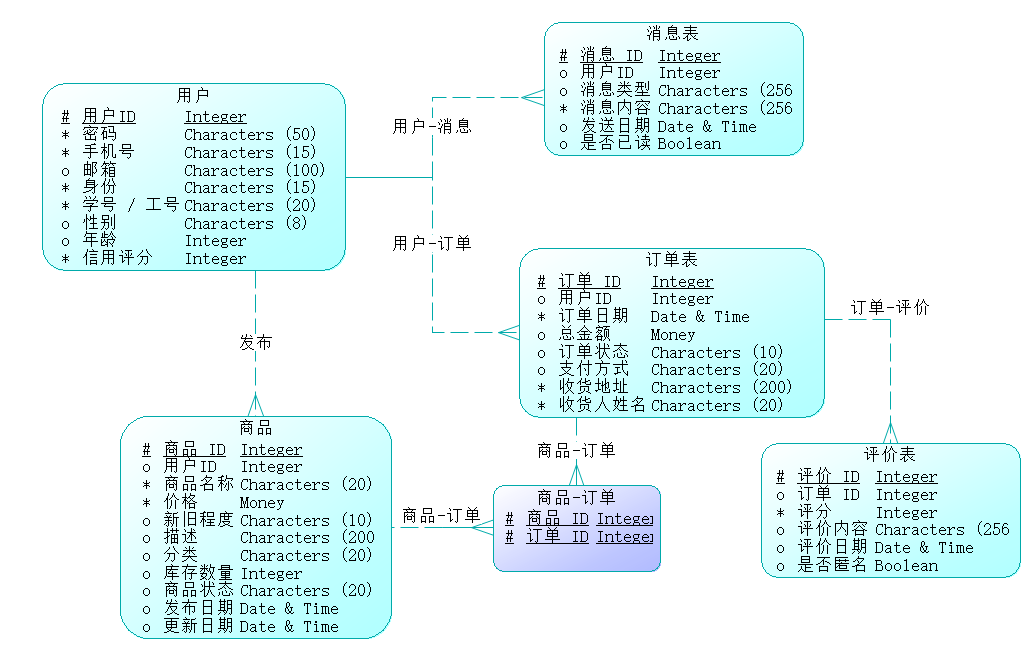
**表结构**

1. **用户表（User）**
   * 用户 ID（user\_id）：主键，整数类型，自增长。
   * 用户名（username）：字符串类型，唯一索引，非空。
   * 密码（password）：字符串类型，非空。
   * 手机号（phone）：字符串类型，唯一索引。
   * 邮箱（email）：字符串类型，唯一索引。
   * 身份（identity）：枚举类型（学生 / 教职工）。
   * 学号 / 工号（student\_no/employee\_no）：字符串类型，唯一索引。
   * 性别（gender）：枚举类型（男 / 女 / 保密）。
   * 年龄（age）：整数类型。
   * 信用评分（credit\_score）：整数类型，默认值为 100。
2. **商品表（Product）**
   * 商品 ID（product\_id）：主键，整数类型，自增长。
   * 商品名称（name）：字符串类型，非空。
   * 价格（price）：数值类型，保留两位小数，非空。
   * 新旧程度（condition）：枚举类型（全新 / 九成新 / 七成新 / 旧）。
   * 描述（description）：文本类型。
   * 分类（category）：字符串类型。
   * 库存数量（stock\_quantity）：整数类型，默认值为 1。
   * 状态（status）：枚举类型（在售 / 已售 / 下架），默认值为在售。
   * 发布日期（post\_date）：日期类型，默认值为当前日期。
   * 更新日期（update\_date）：日期类型，默认值为当前日期，每次更新时自动更新。
   * 发布者 ID（poster\_id）：外键，引用用户表的用户 ID。
3. **订单表（Order）**
   * 订单 ID（order\_id）：主键，整数类型，自增长。
   * 订单日期（order\_date）：日期类型，默认值为当前日期。
   * 总金额（total\_amount）：数值类型，保留两位小数，非空。
   * 订单状态（order\_status）：枚举类型（待付款 / 待发货 / 待收货 / 已完成 / 已取消），默认值为待付款。
   * 支付方式（payment\_method）：枚举类型（微信支付 / 支付宝支付 / 其他）。
   * 收货地址（shipping\_address）：字符串类型。
   * 收货人姓名（receiver\_name）：字符串类型。
   * 收货人手机号（receiver\_phone）：字符串类型。
   * 买家 ID（buyer\_id）：外键，引用用户表的用户 ID。
   * 卖家 ID（seller\_id）：外键，引用用户表的用户 ID。
4. **评价表（Review）**
   * 评价 ID（review\_id）：主键，整数类型，自增长。
   * 评分（rating）：整数类型，范围为 1 - 5，非空。
   * 评价内容（content）：文本类型。
   * 评价日期（review\_date）：日期类型，默认值为当前日期。
   * 是否匿名（is\_anonymous）：布尔类型，默认值为 false。
   * 订单 ID（order\_id）：外键，引用订单表的订单 ID。
   * 评价者 ID（reviewer\_id）：外键，引用用户表的用户 ID。
5. **消息表（Message）**
   * 消息 ID（message\_id）：主键，整数类型，自增长。
   * 消息类型（type）：枚举类型（站内信 / 系统通知 / 交易提醒），非空。
   * 消息内容（message \_content）：文本类型，非空。
   * 发送日期（send\_date）：日期类型，默认值为当前日期。
   * 发送者 ID（sender\_id）：外键，引用用户表的用户 ID。
   * 接收者 ID（receiver\_id）：外键，引用用户表的用户 ID。
   * 是否已读（is\_read）：布尔类型，默认值为 false。

**关系**

1. **用户 - 商品（发布）** ：用户表的用户 ID 作为外键（poster\_id）出现在商品表中，一个用户可以发布多件商品，一件商品只能由一个用户发布。
2. **用户 - 订单（买家）** ：用户表的用户 ID 作为外键（buyer\_id）出现在订单表中，一个用户可以下多个订单，一个订单只能由一个用户下单。
3. **用户 - 订单（卖家）** ：用户表的用户 ID 作为外键（seller\_id）出现在订单表中，一个用户可以有多个作为卖家的订单，一个订单只能由一个用户作为卖家。
4. **商品 - 订单（包含）** ：商品表的商品 ID 和订单表的订单 ID 可以通过关联表（订单商品表）建立多对多关系，但在此简化模型中，假设一个订单只能包含一件商品（可根据实际情况调整），因此在订单表中添加商品 ID（product\_id）作为外键。
5. **订单 - 评价（关联）** ：订单表的订单 ID 作为外键（order\_id）出现在评价表中，一个订单可以有多个评价（如买家对卖家的评价、卖家对买家的评价），一个评价只能关联一个订单。
6. **用户 - 消息（接收）** ：用户表的用户 ID 作为外键（receiver\_id）出现在消息表中，一个用户可以接收多条消息，一条消息只能由一个用户接收。

以下是逻辑数据模型（LDM）图：



图表 11逻辑数据模型（LDM）图

### 4.5.3物理数据模型（PDM）

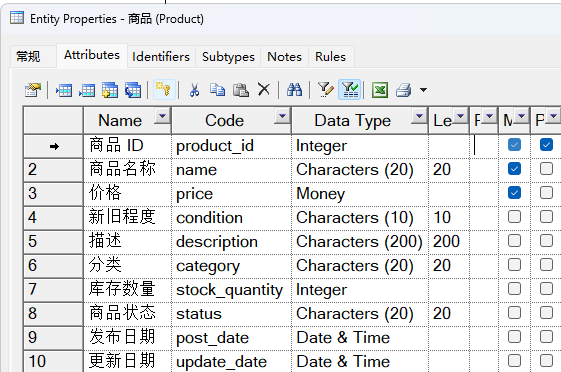
物理数据模型（PDM）是在 LDM 的基础上，进一步确定每个表的物理存储结构，包括数据类型、索引、存储引擎等。以下是 PDM 的详细内容：

**表结构**

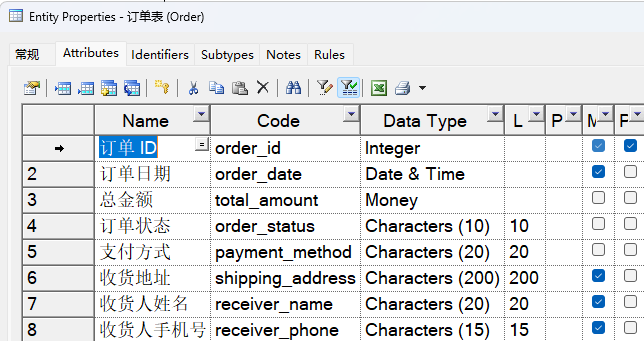
1. **用户表（User）**



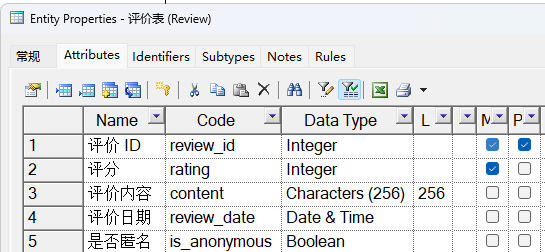
1. **商品表（Product）**



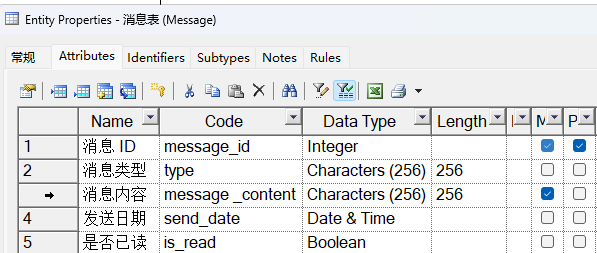
1. **订单表（Order）**



1. **评价表（Review）**



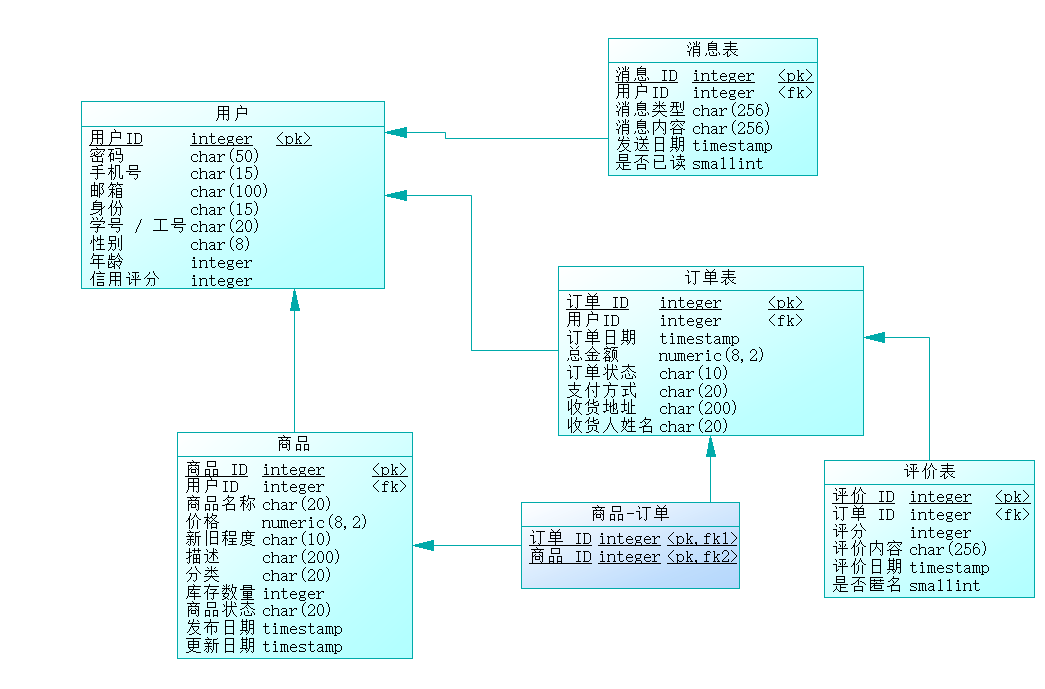
1. **消息表（Message）**



**索引**

1. **用户表（User）** ：为用户名、手机号、邮箱、学号 / 工号等字段创建唯一索引，提高查询效率。
2. **商品表（Product）** ：为商品名称、分类等字段创建索引，方便用户快速搜索商品。
3. **订单表（Order）** ：为订单状态、买家 ID、卖家 ID 等字段创建索引，便于按状态查询订单和按用户查询订单。
4. **评价表（Review）** ：为订单 ID、评价者 ID 等字段创建索引，方便查询订单相关的评价和用户发布的评价。
5. **消息表（Message）** ：为接收者 ID、消息类型等字段创建索引，便于按用户查询消息和按消息类型查询消息。

以下是物理数据模型（PDM）图



图表 12物理数据模型（PDM）图

## 4.6 代码方案设计

系统采用前后端分离架构，前端使用 Vue.js 框架构建用户界面，负责页面展示和用户交互逻辑；后端基于 Spring Boot 框架搭建，提供 RESTful API 服务，处理业务逻辑和数据操作。前端通过 HTTP 请求与后端交互，后端与 MySQL 数据库和 Redis 缓存进行数据存储和读取，确保系统的高性能和可靠性。

### 4.6.1 用户管理模块代码设计

用户注册接口：

@PostMapping("/api/user/register")

public ResponseEntity<?> register(@RequestBody User user) {

boolean result = userService.register(user);

if (result) {

return ResponseEntity.ok().body("注册成功");

} else {

return ResponseEntity.badRequest().body("注册失败");

}

}

### 4.6.2 商品管理模块代码设计

商品发布接口：

@PostMapping("/api/product/publish")

public ResponseEntity<?> publishProduct(@RequestBody Product product) {

boolean result = productService.publish(product);

if (result) {

return ResponseEntity.ok().body("商品发布成功");

} else {

return ResponseEntity.badRequest().body("商品发布失败");

}

}

### 4.6.3 交易管理模块代码设计

订单支付接口：

@PostMapping("/api/order/pay")

public ResponseEntity<?> payOrder(@RequestParam Long orderId) {

boolean result = orderService.pay(orderId);

if (result) {

return ResponseEntity.ok().body("支付成功");

} else {

return ResponseEntity.badRequest().body("支付失败");

}

}

### 4.6.4 消息管理模块代码设计

站内信发送接口：

@PostMapping("/api/message/send")

public ResponseEntity<?> sendMessage(@RequestBody Message message) {

boolean result = messageService.send(message);

if (result) {

return ResponseEntity.ok().body("消息发送成功");

} else {

return ResponseEntity.badRequest().body("消息发送失败");

}

}

### 4.6.5 后台管理模块代码设计

商品审核接口：

@PostMapping("/api/admin/product/audit")

public ResponseEntity<?> auditProduct(@RequestParam Long productId, @RequestParam Boolean approved) {

boolean result = adminService.auditProduct(productId, approved);

if (result) {

return ResponseEntity.ok().body("审核成功");

} else {

return ResponseEntity.badRequest().body("审核失败");

}

}

## 4.7 输入与输出设计

### 4.7.1输入设计

用户输入

|  |  |  |  |
| --- | --- | --- | --- |
| 用户输入表 | | | |
| 用户名 |  | 用户ID |  |
| 手机号 |  | 邮件 |  |
| 学号/工号 |  | | |
| 身份 | 教职工 / 学生 | 性别 | 男 / 女 / 保密 |
| 年龄 |  | 信用评分 | 分 |

图表 13用户输入表

商品输入

|  |  |  |  |
| --- | --- | --- | --- |
| 商品输入表 | | | |
| 商品名称 |  | 商品编号 |  |
| 商品所有用户 |  | 价格 | 元 |
| 分类 |  | 新旧程度 | 全新 / 九成新 / 八成新 |
| 存货数量 | 个 | 发布日期 | 年 月 日 |
| 描述 |  | | |

图表 14商品输入表

### 4.7.2输出设计

**用户信息表**

****

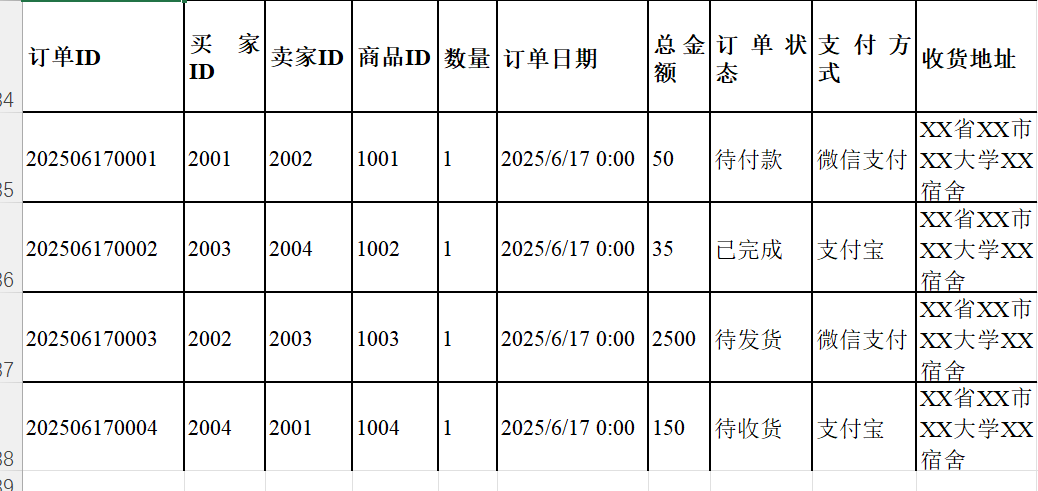
图表 15用户信息表

**商品信息表**

****

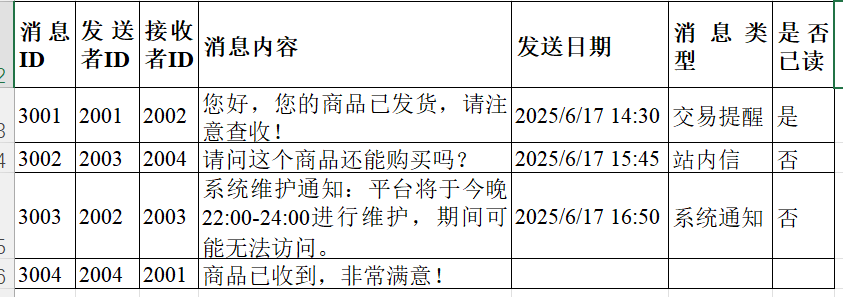
图表 16商品信息表

**订单信息表**



图表 17订单信息表

**消息信息表**



图表 18消息信息表

## 4.8 系统界面设计

### 4.8.1 首页界面

**功能目标**

首页旨在为用户提供关键词，自动联想相关内容，提高搜索效率。

**布局设计**

**顶部导航栏** ：固定在页面顶部，包含网站 logo、搜索框、导航菜单以及用户登录 / 注册入口。搜索框设计为智能提示，用户输入关键词时，自动联想相关内容，提高搜索效率。

**轮播图区域** ：展示平台的特色功能、热门活动、新品推荐等信息，吸引用户关注。轮播图自动播放，同时提供手动切换按钮和指示器，方便用户浏览。

**商品分类展示区** ：以图标加文字的形式展示主要商品分类，如教材、运动器材、电子产品、生活用品等。每个分类图标直观形象，鼠标悬停时有轻微放大效果，增加交互感。

**热门商品推荐区** ：根据用户的浏览历史、购买行为以及商品热度等因素，推荐个性化的热门商品。商品以卡片形式展示，包含商品图片、名称、价格、新旧程度、信用评分等关键信息，用户可快速浏览并点击进入商品详情页。

**校园特色模块** ：突出武汉体育学院的运动特色，设置如 “运动装备特惠”“赛事相关物品交易” 等特色模块，激发学生的购买兴趣，增强平台的吸引力。

### 4.8.2 购物车界面

**功能目标**

购物车界面用于展示用户已添加的商品，方便用户进行结算和调整购物清单。

**布局设计**

**商品列表区** ：以列表形式展示用户添加到购物车的商品，包括商品图片、名称、价格、数量、小计等信息。用户可随时修改商品数量或删除不需要的商品。

**优惠信息展示区** ：显示可用于当前购物车商品的优惠券、促销活动等信息，吸引用户使用优惠，提高购买转化率。

**结算信息区** ：位于页面底部，实时计算商品总价、优惠金额以及最终应付金额。提供 “去结算” 按钮，用户点击后进入订单确认页面，继续完成支付流程。

### 4.8.3 个人中心页面

**功能目标**

个人中心是用户管理个人账户信息、查看订单和进行其他个性化设置的中心枢纽。

**布局设计**

**个人信息概览区** ：展示用户的头像、昵称、学号 / 工号、身份信息以及信用评分等基本资料。用户可点击编辑按钮进入个人信息编辑页面，修改个人资料。

**我的订单区** ：以标签页的形式展示用户的全部订单，包括 “待付款”“待发货”“待收货”“已完成”“已取消” 等不同状态的订单。每个订单以卡片形式展示，包含订单编号、商品信息、订单金额、订单状态等关键信息，用户可快速查看订单详情并进行相应的操作，如付款、提醒发货、确认收货、申请售后等。

**我的发布区** ：展示用户发布的所有商品，包括在售、已售、下架等状态的商品。用户可查看商品的浏览量、购买量等数据，对商品进行编辑、下架等管理操作。

**我的收藏区** ：展示用户收藏的商品和店铺，用户可快速访问收藏的商品详情页或进入店铺首页，方便用户跟踪感兴趣的商品和优质卖家。

**我的评价区** ：展示用户发布的评价记录，包括评价的商品、评价内容、评价时间等信息。用户可查看和修改已发布的评价，确保评价内容真实有效。

### 4.8.4 搜索结果界面

**功能目标**

搜索结果界面旨在根据用户输入的关键词，快速准确地展示相关的商品信息。

**布局设计**

**搜索结果展示区** ：根据用户的搜索关键词，以网格或列表形式展示相关的商品搜索结果。每件商品以卡片形式展示，包含商品图片、名称、价格、信用评分等关键信息，用户可快速浏览搜索结果并点击进入感兴趣的商品详情页。

**筛选与排序工具栏** ：位于搜索结果上方，提供丰富的筛选条件，如商品分类、价格区间、新旧程度、商品状态等，帮助用户快速缩小搜索范围。同时，提供多种排序方式，如按发布时间、价格、销量、信用评分等排序，满足用户不同的搜索需求。

**搜索优化提示区** ：当搜索结果较少或无结果时，在页面显著位置提供搜索优化提示，如建议用户尝试使用同义词、扩大搜索范围、检查关键词拼写等，引导用户获取更准确的搜索结果。

### 4.8.5 购买界面

**功能目标**

购买界面引导用户完成商品购买流程，确保用户能够方便快捷地进行支付和确认订单信息。

**布局设计**

**商品信息确认区** ：再次展示商品的基本信息，包括商品图片、名称、规格、价格等，确保用户确认购买的商品无误。

**收货地址选择区** ：默认显示用户设置的常用收货地址，用户也可点击 “添加新地址” 按钮，填写新的收货地址信息。同时，提供地址修改、删除等功能。

**支付方式选择区** ：列举平台支持的支付方式，如微信支付、支付宝支付等，用户可选择其中一种支付方式。选中的支付方式以高亮显示，未选中的则为灰色显示。

**订单备注区** ：用户可在此区域填写对卖家的特殊要求或备注信息，如商品颜色、尺码的特殊指定等。

**订单金额汇总区** ：清晰地展示商品总价、运费（如有）、优惠金额以及最终的应付金额。用户可在此确认订单金额的明细。

**提交订单按钮** ：位于页面底部，设计为醒目且足够大的按钮，方便用户点击提交订单。按钮颜色采用与平台主色调一致的颜色，增加用户的视觉识别度。

### 4.8.6 消息界面

**功能目标**

消息界面用于展示和管理用户在平台上的各类消息通知，确保用户能够及时获取重要信息。

**布局设计**

**消息分类标签区** ：设置 “系统通知”“交易提醒”“站内信” 等消息分类标签，用户可快速切换查看不同类型的消息。

**消息列表区** ：根据所选消息类型，以列表形式展示相应的消息内容，包括消息标题、发送时间、消息摘要、是否已读等信息。未读消息以加粗、变色等方式突出显示，提醒用户查看。

**消息详情查看区** ：用户点击消息标题或内容后，在右侧弹出消息详情查看窗，展示完整的消息内容，包括图片、链接等。用户可对消息进行回复（针对站内信）或标记为已读操作。

## 4.9 模块流程设计

**4.9.1 用户管理模块流程设计**

1. **注册流程** ：

**开始** ：用户访问平台注册页面。

**输入信息** ：用户填写注册信息，包括用户名、密码、邮箱、手机号、身份（学生 / 教职工）、学号 / 工号等。

**信息验证** ：系统验证填写的信息是否完整、格式是否正确，如密码长度是否符合要求、邮箱格式是否正确等。

**重复性检查** ：系统检查用户名、邮箱、手机号、学号 / 工号是否已被注册。

**提交审核** ：若信息验证通过且无重复，系统将注册信息提交给管理员审核。

**审核结果通知** ：管理员审核后，系统向用户发送审核结果通知。若审核通过，用户注册成功，获得相应权限；若审核不通过，用户根据提示修改信息后重新提交注册申请。

1. **登录流程** ：

**开始** ：用户访问平台登录页面。

**输入账号密码** ：用户输入用户名和密码。

**身份验证** ：系统验证用户名和密码是否匹配且账号状态是否正常（未被锁定等）。

**登录结果** ：若验证通过，用户成功登录，系统记录登录状态并跳转到用户个人中心页面；若验证失败，系统提示错误信息，用户可重新输入账号密码。

**4.9.2 商品管理模块流程设计**

1. **商品发布流程** ：

**开始** ：卖家登录后，进入商品发布页面。

**填写商品信息** ：卖家填写商品名称、价格、新旧程度、描述、分类、上传商品图片等详细信息。

**信息验证** ：系统对商品信息的完整性、格式进行验证，如检查价格是否为有效数字、图片是否上传成功等。

**提交审核** ：验证通过后，卖家提交商品发布请求，系统将商品信息发送至管理员处进行审核。

**审核结果处理** ：管理员审核商品信息，若审核通过，商品正式发布到平台展示；若审核不通过，系统向卖家反馈审核意见，卖家修改后重新提交发布请求。

1. **商品搜索流程** ：

**开始** ：用户进入平台首页或商品列表页面。

**输入搜索关键词** ：用户在搜索框输入商品名称、关键词等。

**搜索匹配** ：系统根据搜索关键词在商品数据库中进行匹配查询，筛选出符合条件的商品。

**展示搜索结果** ：将匹配到的商品以列表形式展示给用户，用户可根据商品信息（如价格、信用评分等）进行进一步筛选和排序。

**4.9.3 交易管理模块流程设计**

1. **订单生成流程** ：

**开始** ：买家确定购买意向后，点击商品购买按钮。

**选择商品规格** ：若商品有不同规格（如颜色、尺码等），买家选择所需规格。

**确认收货地址与支付方式** ：买家选择或填写收货地址，选择支付方式（微信支付、支付宝支付等）。

**生成订单** ：系统根据买家选择的商品、收货地址、支付方式等信息生成订单，订单初始状态为 “待付款”，并分配唯一的订单编号。

**跳转至支付页面** ：系统跳转至相应的支付页面，等待买家完成支付操作。

1. **支付流程** ：

**开始** ：买家在支付页面确认支付信息。

**选择支付方式** ：买家再次确认选择的支付方式。

**支付操作** ：买家通过第三方支付平台完成支付操作，系统实时接收支付结果反馈。

**支付结果处理** ：若支付成功，系统将订单状态更新为 “待发货”，通知卖家有新的订单待处理，并向买家发送支付成功通知；若支付失败，系统提示买家支付失败原因，买家可重新进行支付操作或取消订单。

**4.9.4 消息管理模块流程设计**

1. **站内信发送流程** ：

**开始** ：发送者登录后，进入消息页面的站内信撰写界面。

**选择接收者** ：发送者选择要发送消息的用户（可搜索用户昵称或从联系人列表中选择）。

**撰写消息内容** ：发送者输入消息主题和具体内容。

**发送消息** ：发送者点击发送按钮，系统将消息发送给指定的接收者，并在消息列表中记录发送消息的信息（如发送时间、消息主题等）。

1. **系统通知发送流程** ：

**开始** ：系统根据预设的规则或事件触发通知条件，如平台活动通知、系统维护提醒等。

**确定通知内容与范围** ：系统确定通知的具体内容以及需要接收通知的用户范围（全体用户或特定用户群体）。

**发送通知** ：系统向指定范围的用户发送系统通知，通知以消息的形式显示在用户的系统通知列表中，并可通过消息详情页查看完整内容。

**4.9.5 后台管理模块流程设计**

1. **用户管理流程** ：

**开始** ：管理员登录后台管理系统，进入用户管理页面。

**用户信息查看与筛选** ：管理员可查看所有用户的信息列表，并可通过筛选条件（如用户名、身份、信用评分等）快速定位特定用户。

**用户操作** ：管理员可对用户进行审核、修改用户信息（如锁定 / 解锁账号、调整信用评分等）、查看用户交易记录等操作。

1. **商品审核流程** ：

**开始** ：管理员登录后台管理系统，进入商品审核页面。

**待审核商品列表展示** ：系统展示所有待审核的商品信息列表，包括商品名称、发布者、发布时间等。

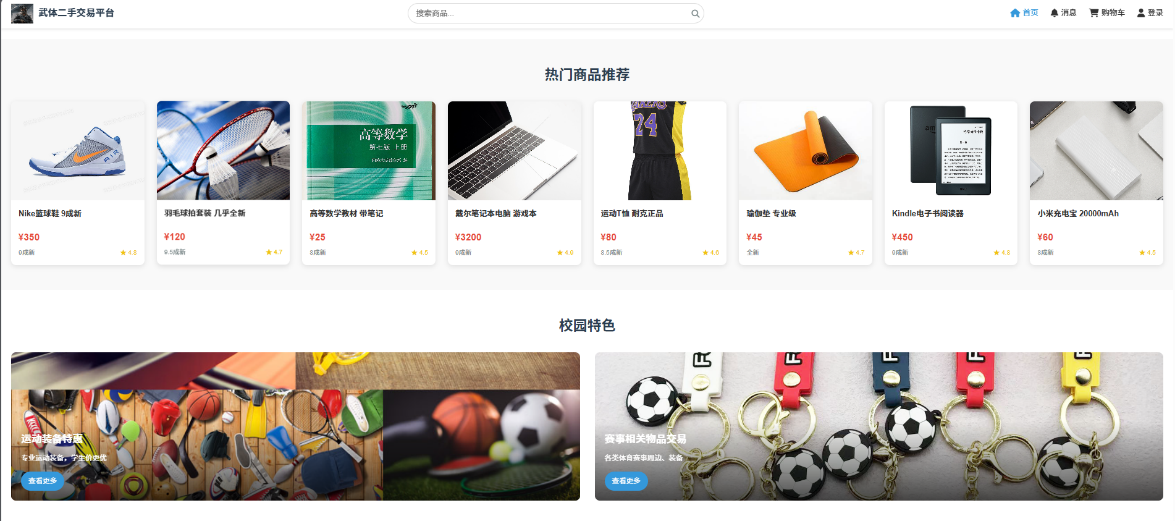
**审核操作** ：管理员逐个审核商品信息，判断商品是否符合平台规定（如商品信息真实性、合法性等）。若审核通过，点击 “审核通过” 按钮，商品正式发布；若审核不通过，填写审核不通过原因，点击 “审核不通过” 按钮，系统将审核结果反馈给卖家。

## 4.10 系统原型设计

首页界面

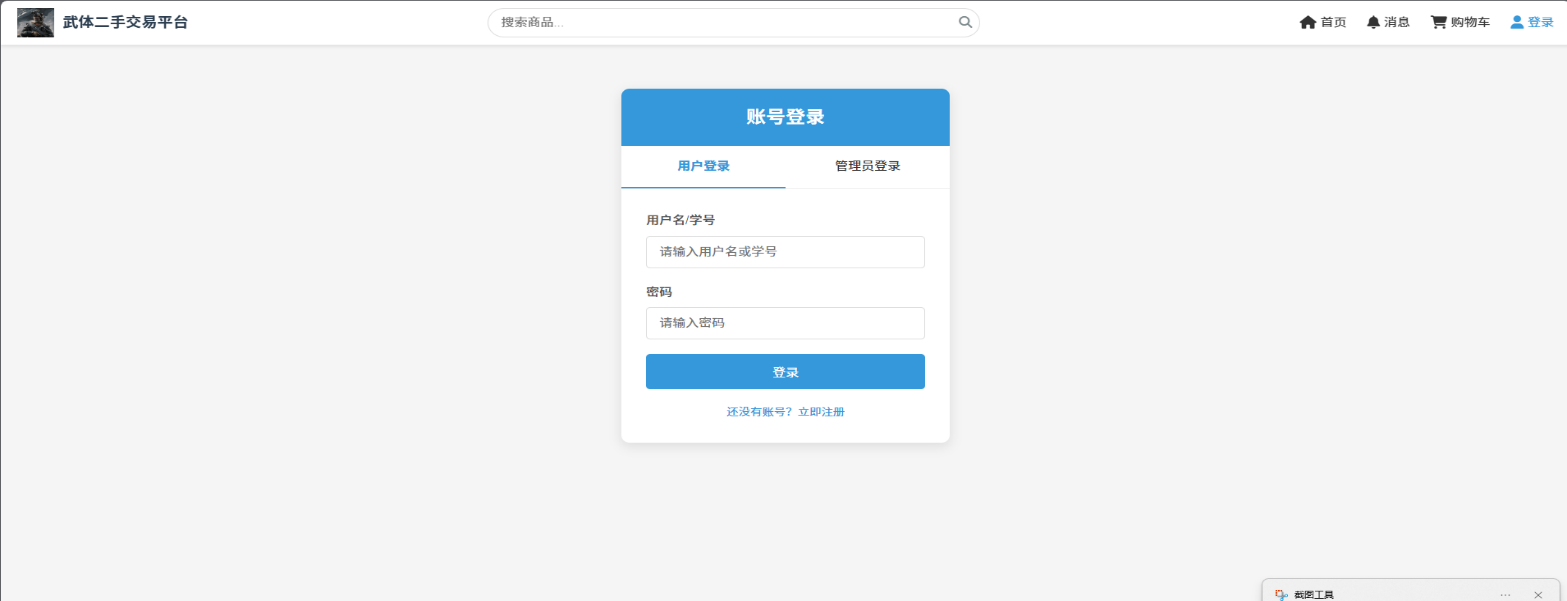


图表 19首页界面1



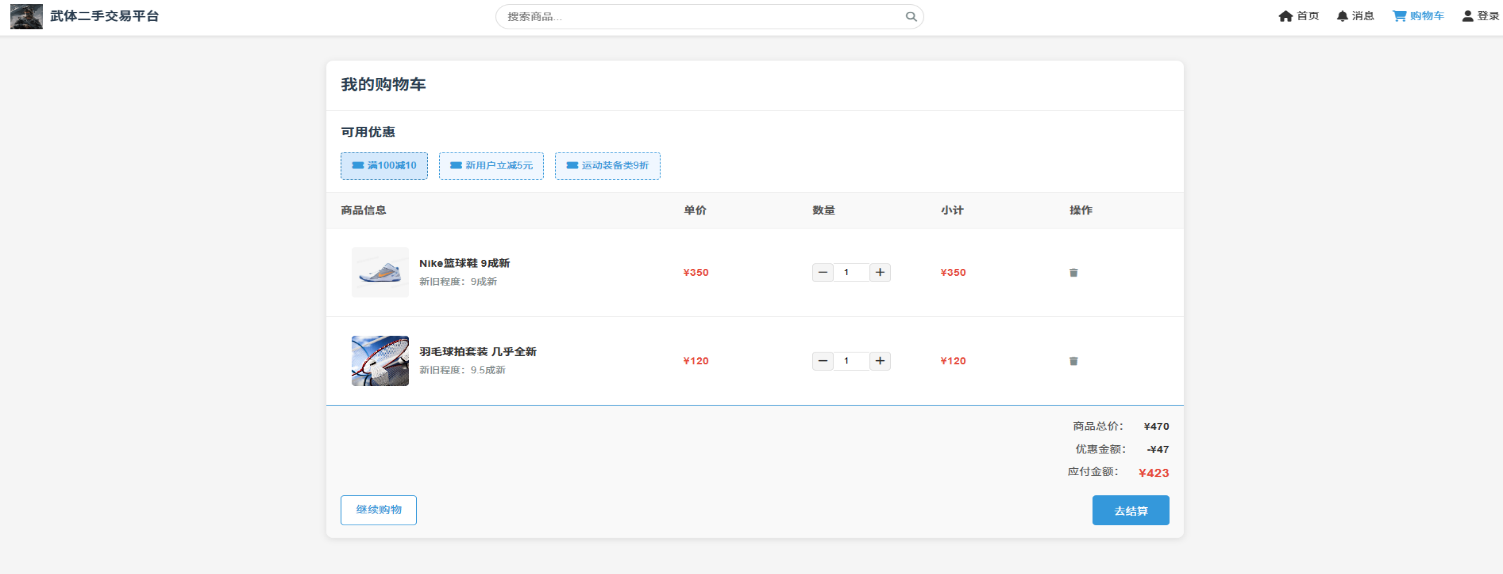
图表 20首页界面2

登录界面



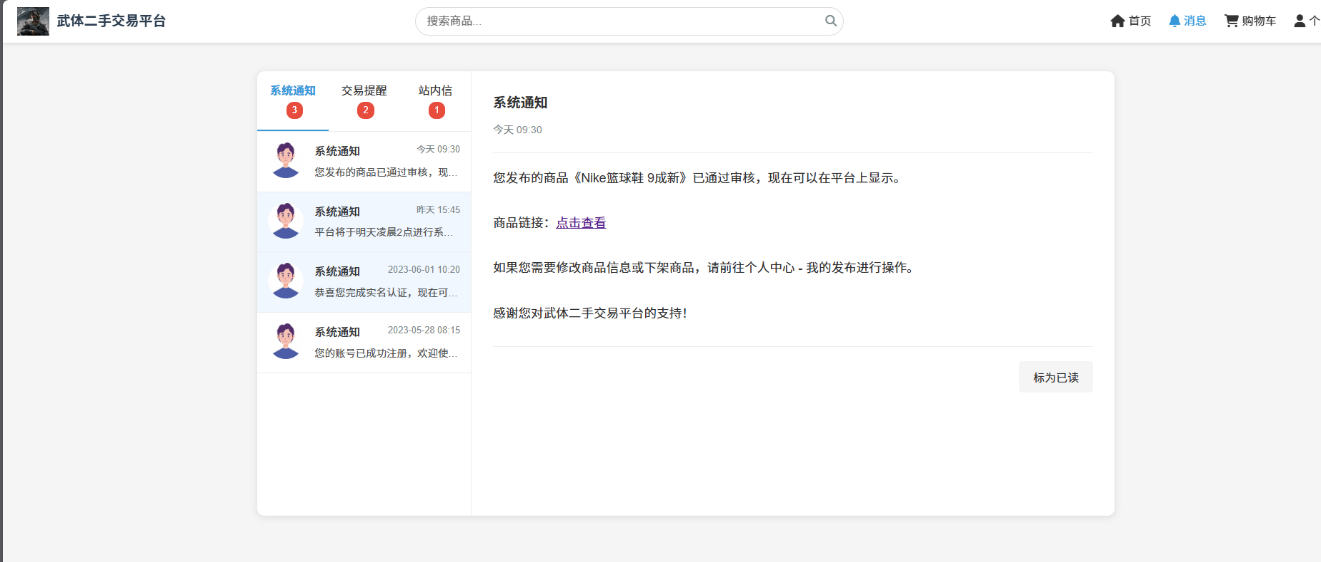
图表 21登录界面

购物车界面



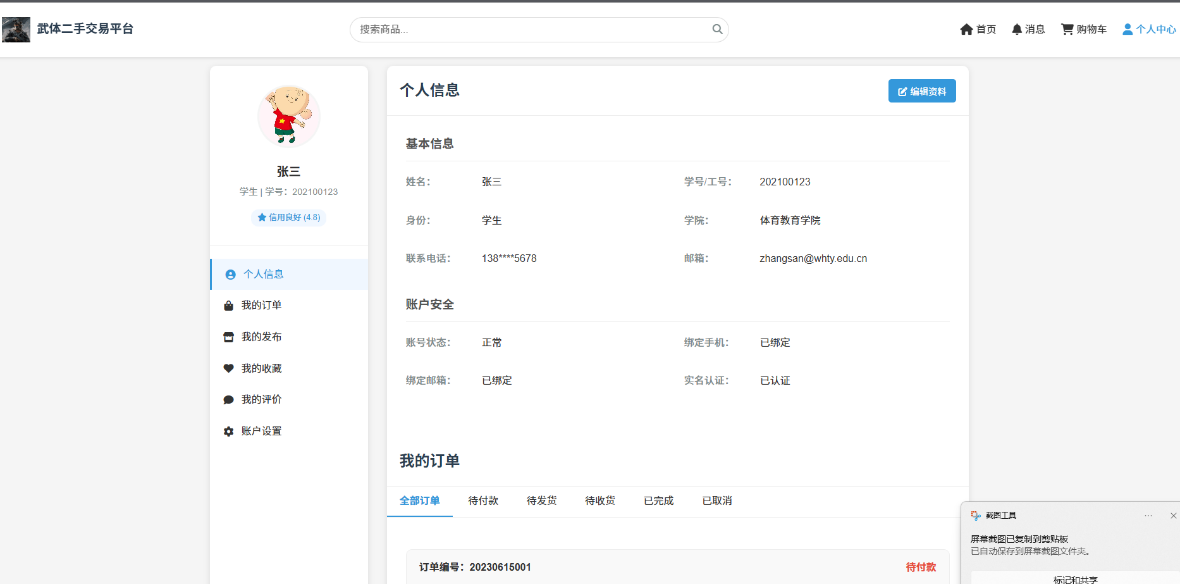
图表 22购物车界面

消息界面



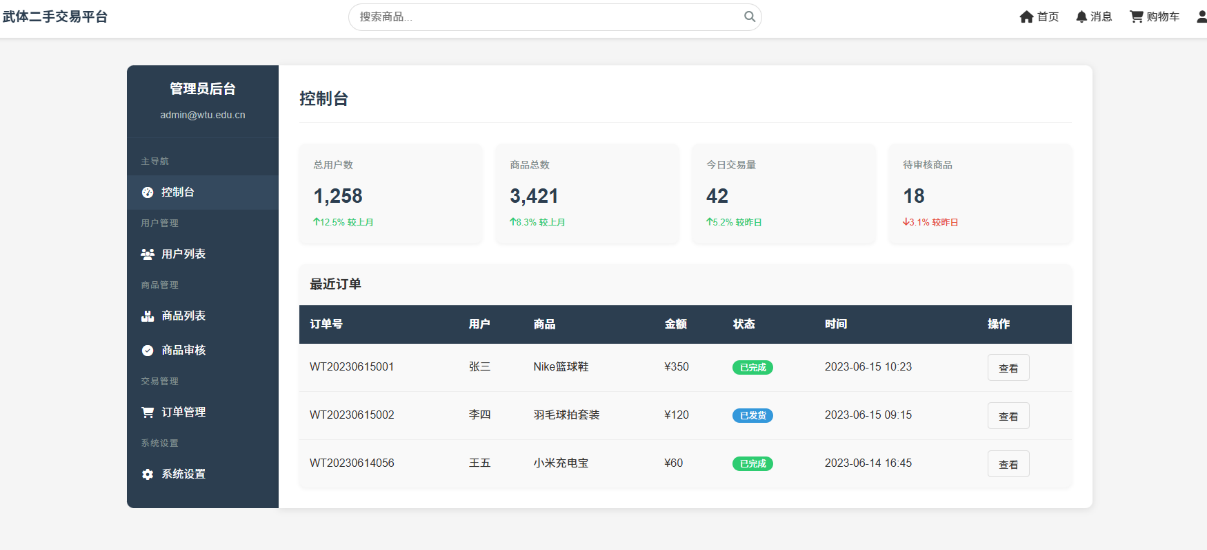
图表 23消息界面

个人中心界面



图表 24个人中心界面

管理员后台页面



图表 25管理员后台界面

# 五、系统实施

## 5.1 物理系统实施设计

以下是针对校园二手交易平台的物理系统实施设计方案，结合技术架构、基础设施及安全运维等关键要素，确保系统高效稳定运行：

### 5.1.2基础设施规划

**1） 服务器部署架构**

云服务选择：采用混合云架构（如阿里云/腾讯云），核心业务部署于公有云（弹性扩容），敏感数据（用户隐私）存储在私有云或校园本地服务器。

服务器配置：

Web服务器（Nginx/Apache）：部署Vue.js前端，处理静态资源与负载均衡，配置2核4GB×2（主备冗余）。

应用服务器（Tomcat）：运行SpringBoot后端，4核8GB×3（集群部署，支持5000+并发）。

数据库服务器（MySQL集群）：主从复制（1主2从），SSD存储，16GB内存，保障高可用。

CDN加速：静态资源分发至全球节点，提升访问速度。

**2）网络与安全**

防火墙：部署WAF（Web应用防火墙），防御SQL注入、XSS攻击。

隔离设计：划分DMZ区（对外服务）与内网区（数据库/管理后台），通过VPN接入运维。

HTTPS加密：全站启用SSL/TLS证书（Let's Encrypt），保障数据传输安全。

### 5.1.2物理设备清单

|  |  |  |
| --- | --- | --- |
| 组件 | 规格与数量 | 说明 |
| 应用服务器 | 4和8GB RAM/3台 | 部署SpringBoot应用  Docker容器化 |
| 数据接口服务器 | 16GB RAM/SSD 500GB\*3 | MySQL集群  每日自动备份 |
| 存储设备 | NAS网络存储(10TB) | 商品图片/文件存储 |
| 网络交换机 | 千兆企业级交换机x2 | 双机热备 |
| UPS电源 | 在线式2KVA | 保障30分钟应急供电 |

图表 26物理设备清单表

### 5.1.3系统模块部署方案

**1） 用户服务模块**

独立部署认证微服务（JWT Token鉴权），支持OAuth2.0登录（微信/校园统一认证）。

**2） 交易与支付模块**

对接第三方支付（微信支付/支付宝沙箱环境），交易请求隔离至独立安全区。

支付回调接口采用IP白名单+签名验证。

**3） 商品管理模块**

图片存储：对象存储（如阿里云OSS），压缩缩略图生成（节省带宽）。

搜索引擎：Elasticsearch实现商品关键词检索，部署于专用服务器。

### 5.1.4容灾与备份策略

数据备份：

每日增量备份（Binlog）+ 每周全量备份，异地存储（校园机房+云存储）。

故障转移：

数据库主从切换（Keepalived），应用层Nginx健康检查自动剔除故障节点。

灾备演练：每季度模拟服务器宕机，恢复时间目标（RTO）<15分钟。

### 5.1.5运维监控体系

**1）监控工具：**

Prometheus + Grafana：实时监控服务器CPU/内存/磁盘。

ELK日志分析：追踪API请求异常与性能瓶颈。

**2） 告警机制：**

企业微信/邮件告警（阈值：CPU>85%或响应延时>1s）。

### 5.1.6实施进度计划

|  |  |  |
| --- | --- | --- |
| 阶段 | 周期 | 任务 |
| 硬件部署 | 1周 | 服务器上架，网络拓扑搭建 |
| 系统安装 | 2周 | OS环境配置，中间件部署，集群初始化 |
| 压力测试 | 3天 | JMeter模拟2000并发用户 |
| 试运行 | 1个月 | 小范围灰度发布，修复漏洞 |

图表 27实施计划表

备注：硬件成本可优化——初期使用云服务按需付费，后期根据用户量扩展（参考：校园平台日均UV约3000-5000）。

此方案兼顾性能、安全与扩展性，符合高校实际运维能力，确保平台稳定服务校园二手交易生态。

## 5.2 系统测试设计

以下是为校园二手交易平台设计的系统测试方案，涵盖功能、性能、安全及兼容性等核心测试维度，结合用户需求与行业最佳实践：

### 5.2.1测试目标

1）验证系统功能符合需求规格（参考功能模块设计）

2）确保高并发场景下系统稳定性（日均UV 3000-5000）

3）保障交易流程安全性与数据隐私

4）验证跨平台兼容性（PC/移动端）

### 5.2.2测试范围

|  |  |  |
| --- | --- | --- |
| 测试类型 | 覆盖模块 | 依据 |
| 功能测试 | 用户注册/登录、商品发布/搜索、订单交易、支付回调、退货退款 | 需求文档 |
| 性能测试 | 商品搜索、支付接口、高并发下单 | 用户规模预测 |
| 安全测试 | 身份认证、支付安全、SQL注入防护、XSS攻击防御 | 安全设计 |
| 兼容性测试 | Chrome/Firefox/微信浏览器 | 跨平台需求 |
| 用户体验测试 | 界面响应速度、操作流程顺畅度 | 非功能性需求 |

图表 28测试表

### 5.2.3详细测试策略

**1）功能测试用例（核心示例）**

|  |  |  |
| --- | --- | --- |
| 模块 | 测试场景 | 预期结果 |
| 支付流程 | 用户支付成功后中断网络 | 订单状态自动更新为“已支付” |
| 退货退款 | 卖家拒绝退货后买家申诉 | 管理员后台触发仲裁流程 |
| 商品搜索 | 输入“教材 九成新”模糊查询 | 返回相关度排序结果 |
| 校园认证 | 使用无效学号注册 | 提示“学籍信息验证失败” |

图表 29功能测试表

**2）性能测试方案**

工具：JMeter + Prometheus监控

场景设计：

200用户并发浏览商品（响应时间<1s）

50用户同时支付（TPS≥20，错误率<0.1%）

持续压测30分钟（内存泄漏检测）

**3）安全测试重点**

支付安全：

模拟支付回调参数篡改（验证签名机制）

敏感信息脱敏（银行卡号、手机号）

漏洞扫描：

使用OWASP ZAP扫描XSS/SQL注入漏洞

会话超时机制验证（10分钟无操作强制退出）

**4）兼容性测试矩阵**

|  |  |  |
| --- | --- | --- |
| 平台 | 浏览器/OS | 验证动作 |
| 移动端 | iOS 15+/Android 12+ | 商品发布、扫码支付 |
| 微信生态 | 微信内置浏览器 | 小程序授权登录、消息推送 |
| PC端 | Chrome 110+/Edge 100+ | 多图上传、订单导出 |

图表 30兼容性测试

### 5.2.4测试环境与数据

|  |  |  |
| --- | --- | --- |
| 环境 | 配置 | 作用 |
| 测试数据库 | MySQL 8.0 主从复制 | 模拟生产数据一致性 |
| 模拟支付网关 | 支付宝/微信沙箱环境 | 支付流程验证 |
| 测试数据量 | 商品数据10万条+用户数据1万条 | 性能测试基础 |
| 网络模拟 | Charles模拟弱网（3G/高延迟） | 极端场景容错 |

图表 31环境测试

### 5.2.5测试交付物

1）测试报告：含缺陷分布图、性能瓶颈分析

2）缺陷清单：按严重等级排序（阻塞/严重/一般）

3）安全审计报告：OWASP TOP 10漏洞修复建议

4）上线建议：基于测试结果评估发布风险

### 5.2.6风险应对

支付回调丢失：设计Mock服务重发机制

高并发超时：数据库连接池优化+缓存策略调整

身份冒用：加强学籍信息二次验证（如人脸比对）

执行周期：建议2周（功能测试7天+性能/安全测试5天+回归测试2天）

参考依据：毕业设计测试标准+ 企业级系统安全规范

本方案确保覆盖校园场景核心需求，特别强化交易安全与高可用性，适配学生用户行为特点。

## 5.3 系统运行设计

以下是为校园二手交易平台设计的系统运行设计方案，涵盖运行环境、服务管理、监控维护及应急响应等核心环节，结合高校场景特性和技术最佳实践：

### 5.3.1运行环境配置

**1）服务器资源分配**

应用层：Tomcat集群（3节点），每节点配置4核8GB内存，Docker容器化部署，通过Nginx负载均衡（加权轮询策略）。

数据层：MySQL主从架构（1主2从），主节点16GB内存+SSD磁盘，从节点8GB内存，半同步复制保障数据一致性。

缓存层：Redis哨兵模式（3节点），缓存商品目录及会话信息，降低数据库压力。

**2）混合云部署架构**

前端静态资源托管公有云（阿里云OSS+CDN加速），后端API及数据库部署校内私有服务器，通过专线打通网络。

敏感数据（用户身份、支付信息）隔离存储于校内私有云，符合《网络安全法》要求。

### 5.3.2关键服务运行设计

|  |  |  |
| --- | --- | --- |
| 服务模块 | 运行策略 | 安全控制 |
| 用户认证 | 对接校园统一身份系统（OAuth 2.0），登录会话有效期24小时，JWT令牌自动续签 | 异地登录实时短信告警，连续失败5次锁定账户 |
| 交易支付 | 微信/支付宝沙箱环境对接，支付回调接口IP白名单限制，交易流水异步落库 | 金额变动需二次密码确认，敏感操作留痕审计 |
| 商品管理 | Elasticsearch索引每日增量更新（凌晨1点），图片存储OSS并压缩至200KB以内 | 敏感词过滤（政治/违禁品），AI图片内容审核 |

图表 32服务运行设计

### 5.3.3持续运维机制

**1）自动化运维栈**

部署：GitLab CI/CD流水线，代码合并后自动构建Docker镜像并滚动更新。

监控：

基础设施：Prometheus监控服务器CPU/内存/磁盘I/O，阈值告警（CPU>80%持续5分钟）。

业务层面：ELK收集API日志，分析接口成功率（要求≥99.9%）及慢查询（>500ms）。

备份：MySQL每日全备+Binlog实时同步，备份文件加密存储于异地机房（保留30天）。

**2）性能优化策略**

高并发场景：秒杀活动期间启用Redis队列限流（每秒500请求），预热商品缓存。

数据库优化：

查询分离：读写请求分流至主从库

索引优化：高频搜索字段（商品名、分类）添加B+Tree索引。

### 5.3.4应急预案设计

|  |  |  |
| --- | --- | --- |
| 故障类型 | 响应措施 | 恢复目标 |
| 数据库主节点宕机 | Keepalived触发从库升主，5分钟内切换流量，短信通知运维团队 | RTO<10分钟 |
| 支付回调丢失 | 消息队列重试机制（3次/2分钟间隔），失败订单人工核对 | 数据零丢失 |
| DDoS攻击 | 启用云WAF清洗流量，黑洞策略拦截异常IP，带宽扩容至1Gbps46 | 服务不中断 |

图表 33应急预案设计

## 5.3.5运行成本控制

弹性扩缩容：

日常流量：维持基础资源（CPU利用率40%以下）

开学季/毕业季：公有云自动扩容50%计算节点。

存储成本优化：

冷数据（6个月未交易商品）迁移至低价OSS归档存储。

### 5.3.6运行文档体系

1）《系统运维手册》：含服务启停命令、监控配置、备份恢复步骤

2）《故障处理SOP》：典型故障场景处置流程（附操作截图）

3）《权限管理规范》：分级管理员权限表（如客服仅可操作订单退款）。

运行周期规划：

试运行期（1个月）：每日巡检，记录性能基线

稳定运行期：周度健康检查，季度容灾演练

参考依据：校园系统安全规范+ 高可用架构设计

本方案强化了混合云架构的安全可控性，针对校园场景优化了身份认证与支付流程，并通过自动化运维降低人力成本，确保平台7×24小时稳定服务。

# 六、结束语

武汉体育学院二手交易平台的建设具有显著的实际意义和可行性。系统设计应以用户需求为中心，注重交易安全与便捷性，同时考虑与学校现有信息系统的整合。建议采用分阶段实施策略，先实现核心功能，再逐步扩展增值服务。系统上线后应建立有效的运营机制，确保平台的活跃度和安全性。

# 七、参考文献

[1]郭林盛,张晓艳,赵妍,等.基于微信云开发实现校园二手交易的平台设计[J].现代计算机,2024,30(22):185-190.

[2]周佳敏.二手交易平台用户感知价值对满意度的影响研究[D].哈尔滨商业大学,2024.DOI:10.27787/d.cnki.ghrbs.2024.000035.

[3]谢虓,潘秋群,裴宇成,等.基于波特五力模型高校二手交易平台的优化与改良[J].再生资源与循环经济,2024,17(05):10-14.

[4]刘珉彤,朱敏娜,潘柯文,等.社区闲置交易系统开发实施的前景分析[J].经济研究导刊,2024,(10):54-57.

[5]赵俊杰,葛敬军,朱文婷.基于微信小程序的校园二手书交易平台的设计与实现[J].科技与创新,2024,(09):7-11+15.DOI:10.15913/j.cnki.kjycx.2024.09.002.

[6]蒋瑞霞,王莉.基于地理位置的校园二手交易平台设计与实现[J].物联网技术,2024,14(04):73-76.DOI:10.16667/j.issn.2095-1302.2024.04.018.

[7]董美雯.信息生态视角下线上二手交易平台用户持续使用意愿研究[D].湖北大学,2024.DOI:10.27130/d.cnki.ghubu.2024.001714.

[8]孙丽,王皓,戴璐,等.大学校园二手交易平台构建与运营——以E大学“花梨闲转”微信小程序为例[J].科技与创新,2024,(04):12-16.DOI:10.15913/j.cnki.kjycx.2024.04.003.

[9]蒙金蓝,黄远香,高梦婷,等.废品收购嵌入校园二手交易平台的路径探究[J].中国市场,2024,(05):137-141.DOI:10.13939/j.cnki.zgsc.2024.05.032.

[10]唐瑛.服务设计思维下校园二手书籍交易平台设计研究[J].大众文艺,2023,(23):13-15.DOI:10.20112/j.cnki.ISSN1007-5828.2023.23.005.

[11]殷梅雨,肖月,周慧珍.基于“互联网+”的校园二手交易平台发展对策研究[J].现代商贸工业,2023,44(23):23-25.DOI:10.19311/j.cnki.1672-3198.2023.23.008.

[12]王川,高凤英,蔡则天,等.高校二手书籍交易平台研究与设计——以内蒙古大学为例[J].科技创业月刊,2023,36(05):176-179.

[13]李顺,康雄杰.当前经济背景下校园二手交易平台的构建——以邵阳学院为例[J].财富时代,2023,(05):88-90.

[14]饶锎月.可持续商业模式下垂直类二手交易平台服务设计研究[D].江南大学,2023.DOI:10.27169/d.cnki.gwqgu.2023.002036.

[15]李钱洪.贵阳市花溪大学城二手市场行情分析与发展建议[J].现代商贸工业,2023,44(08):78-81.DOI:10.19311/j.cnki.1672-3198.2023.08.026.

[16]林梦瑶,武光华,田爽,等.基于二手书交易平台校园纸质垃圾回收利用研究[J].再生资源与循环经济,2023,16(02):25-27.

[17]曹振兴,刘亚娇,邹敏,等.绿色校园背景下的校园二手交易平台[J].数字通信世界,2023,(01):75-77.

[18]高耀,许诺,李博,等.基于Web的新型校园二手交易平台实践研究——以“校易集市”为例[J].中国商论,2023,(01):130-132.DOI:10.19699/j.cnki.issn2096-0298.2023.01.130.

[19]李馨雅,吴茂琳,许婕益,等.互联网下大学生二手交易平台的物流网络优化研究[J].中国储运,2023,(01):197-198.DOI:10.16301/j.cnki.cn12-1204/f.2023.01.115.

[20]张翼飞,郭宇斌,高斌梁.关于大学生App的开发与现实意义——以校园二手交易平台“拾物”App为例[J].老字号品牌营销,2022,(22):88-90.