****

**2025年春季学期**

**基础与智能软件开发实践**

项目名称：基于JAVA的天然石材交易管理系统的设计与实现

|  |  |  |
| --- | --- | --- |
| 姓名 | ： | 何雅君 |
| 学号 | ： | 2022211835 |
| 指导教师 | ： | 朴学峰 |
| 学院 | ： | 计算机科学与技术学院 |
| 日 期 | ： | 2025年5月 |
| 学校 | ： | 哈尔滨工业大学（威海） |

|  |
| --- |
| **哈尔滨工业大学（威海）基础与智能软件开发任务书** |
| 个人班号：2211102  个人姓名：何雅君  个人学号：2022211835  院（系）： 计算机科学与技术暨软件学院 专业：软件工程/服务科学 | |
| 课程设计题目：基于JAVA的天然石材交易管理系统的设计与实现 | |
| 1. 系统目的/解决的问题  传统天然石材行业长期依赖人工操作与纸质记录，存在业务流程繁琐、数据统计易出错、信息共享滞后等问题，且各环节（库存、订单、物流）存在信息孤岛，部门协同效率低，还面临商业机密泄露风险。同时，消费者需求多样化与线上交易习惯普及，推动行业亟需高效、安全的数字化平台，因此设计开发基于 Java 的天然石材交易管理系统。  解决的问题  解决传统人工管理效率低问题，通过自动化流程（如库存预警、订单跟踪）减少人工干预与误差。  打破信息孤岛，实现库存、订单、物流等数据实时共享，提升部门协同效率。  保障数据安全，通过权限分级、数据加密（SHA-256）、JWT 令牌认证防范信息泄露。  满足线上交易需求，提供商品展示、在线支付、物流追踪等功能，适配消费者习惯。  2. 系统的目标用户   普通用户：有天然石材购买需求的消费者，需浏览商品、下单支付、跟踪订单等。  商家：石材供应商，需管理店铺、维护商品信息、处理订单、管理库存与优惠券、分析销售数据。  管理员：平台运营管理者，需管理用户（增删改查、角色分配）、审核内容（资讯、店铺资质）、监控库存、统计数据、配置系统参数。  3. 系统基本功能   用户端功能  注册登录模块：填写账号、密码、邮箱、手机号等信息注册；输入账号密码 + 选择角色（用户 / 商家）登录；通过绑定邮箱 / 手机号找回密码。  商品浏览与搜索模块：按石材分类（大理石、花岗岩等）导航，按价格、类型筛选，输入关键词搜索；查看商品详情（规格、库存、评价），支持图片放大。  购物车管理模块：添加商品至购物车，调整数量或移除商品，实时计算总价；结算时校验库存，库存不足提示移除或等待补货。  订单与支付模块：确认订单信息（收货地址、商品数量），选择支付宝 / 微信支付；支付成功更新订单为 “待发货”，超时 / 失败自动取消并释放库存；跟踪订单状态（待支付 / 发货 / 收货 / 完成），查询物流单号。  个人中心模块：管理收货地址（增删改查、智能填充）；收藏石材与店铺，支持取消收藏；修改密码（验证旧密码、评估新密码强度）；提交反馈（填写标题 / 内容，上传附件）；查看个人订单、评价历史。  辅助功能：在线客服（实时文字、图片、商品链接沟通）；查看石材资讯与系统通知。  商家端功能  店铺管理模块：编辑店铺信息（名称、LOGO、地址、联系方式），跟踪资质审核状态。  商品管理模块：维护石材信息（增删改查名称、规格、价格、库存、属性），支持批量操作与 Excel 导入导出；管理石材分类（系统分类 / 自定义分类）。  订单处理模块：查看待发货订单，录入物流单号并校验，更新订单为 “已发货”；处理退换货申请（审核、回复、确认退款）；导出订单数据用于财务对账。  库存管理模块：查看库存记录（入库时间、数量、操作人），手动添加 / 删除记录；设置库存预警阈值，库存低于阈值时接收预警通知，生成采购单补货。  营销与分析模块：创建优惠券（设置折扣、使用条件、有效期），查看发放记录；通过可视化图表（ECharts）分析销售趋势、热销商品、库存状态。  客服模块：使用在线客服系统与用户沟通，设置自动回复与快捷短语。  管理员端功能  用户管理模块：查看所有用户信息（账号、姓名、联系方式），对用户进行增删改查操作，批量处理用户数据；分配用户角色（普通用户 / 商家 / 管理员），控制操作权限。  内容管理模块：管理资讯（创建 / 编辑 / 删除资讯，分类、统计阅读量、设置置顶）；审核用户反馈，分配处理部门，更新反馈状态并通知用户；发布系统通知（设置标题、内容、显示状态）。  商品与店铺管理模块：审核商家资质与商品信息，对违规商品批量下架；管理石材分类体系，维护系统预设分类与商家自定义分类。  订单与库存监控模块：查看平台所有订单详情（买卖双方信息、物流状态），按多条件筛选订单，处理订单纠纷；监控全局库存，查看库存预警信息与补货进度。  数据统计与系统设置模块：按时间、品类、区域等维度统计销售数据，生成可视化报表（销售额趋势图、品类占比饼图）并导出 Excel；配置系统参数（支付方式、物流接口、权限规则），记录系统日志与监控异常。  4. 系统基本功能是如何达成系统的目标   系统通过分角色功能设计，精准匹配不同用户需求，达成 “推动石材行业数字化转型，提升效率与安全性” 的目标：  对普通用户，提供从浏览到售后的全流程线上服务（商品展示、下单支付、订单追踪），满足其便捷购物需求，适配线上交易习惯，达成提升用户体验与线上交易覆盖的目标。  对商家，提供店铺、商品、订单、库存的一体化管理功能，结合销售数据分析，帮助商家高效运营，减少人工成本（如库存预警减少巡检），提升销售效率，达成助力商家数字化管理的目标。  对管理员，通过用户管理、内容审核、数据统计等功能，实现平台规范化运营与风险管控（如权限控制防范数据泄露），保障平台稳定安全，达成统筹管理与行业数字化推动的目标。  整体功能通过前后端分离架构（Spring Boot+Vue.js）、数据实时共享、安全机制（加密、认证），解决传统行业痛点，实现高效、安全的数字化管理，推动行业向智能化转型。 | |
|  | |

摘 要

随着传统石材行业数字化转型需求的日益迫切，天然石材交易管理系统的设计与实现成为提升行业效率的关键。传统交易模式存在业务流程繁琐、信息共享滞后等问题。本研究针对这些痛点，提出基于Java的天然石材交易管理系统，旨在通过技术整合优化业务流程，构建高效、安全的数字化管理平台，推动石材行业向智能化方向转型。

系统采用Spring Boot与Vue.js的前后端分离架构，后端通过MyBatis实现数据持久化操作，结合MySQL数据库保障业务数据的可靠存储；前端利用组件化开发提升交互体验。功能模块涵盖用户权限管理、石材商品展示、库存动态预警及订单全流程跟踪。

研究过程中综合运用文献分析、实地调研与测试验证等方法，确保系统设计的科学性与实用性。结果表明，该系统能够有效为管理者提供数据驱动的决策支持。

关键词：天然石材；交易管理系统；Spring Boot；Vue.js；数字化转型

**ABSTRACT**

With the increasingly urgent demand for digital transformation in the traditional stone industry, the design and implementation of natural stone trading management systems have become the key to improving industry efficiency. The traditional trading model has problems such as cumbersome business processes and lagging information sharing. This study proposes a natural stone trading management system based on Java to address these pain points, aiming to optimize business processes through technological integration, build an efficient and secure digital management platform, and promote the transformation of the stone industry towards intelligence.

The system adopts a front-end and back-end separation architecture of Spring Boot and Vue.js. The back-end implements data persistence operations through MyBatis, combined with MySQL database to ensure reliable storage of business data; The front-end utilizes component-based development to enhance the interactive experience. The functional modules cover user permission management, stone product display, inventory dynamic warning, and full process tracking of orders.

During the research process, literature analysis, field research, and testing verification were comprehensively utilized to ensure the scientific and practical nature of the system design. The results indicate that the system can effectively provide data-driven decision support for managers.

**Keywords：**Natural Stone; Transaction Management System; Spring Boot；Vue.js； Digital Transformation

目 录

[1 绪论 1](#_Toc207360773)

[1.1 研究背景 1](#_Toc207360774)

[1.2 国内外研究现状 1](#_Toc207360775)

[1.3 研究意义 2](#_Toc207360776)

[1.4 研究内容 2](#_Toc207360777)

[1.5 研究方法 3](#_Toc207360778)

[1.6 本章小结 4](#_Toc207360779)

[2 系统关键技术 5](#_Toc207360780)

[2.1 Spring Boot 5](#_Toc207360781)

[2.2 Vue.js 5](#_Toc207360782)

[2.3 Element-UI 5](#_Toc207360783)

[2.4 MySQL 数据库 6](#_Toc207360784)

[2.5 MyBatis 框架 6](#_Toc207360785)

[2.6 Spring Security 7](#_Toc207360786)

[2.7 JWT 令牌 7](#_Toc207360787)

[2.8 本章小结 7](#_Toc207360788)

[3 需求分析 9](#_Toc207360789)

[3.1 需求分析概述 9](#_Toc207360790)

[3.2 可行性分析 9](#_Toc207360791)

[3.2.1 经济可行性 9](#_Toc207360792)

[3.2.2 技术可行性 10](#_Toc207360793)

[3.2.3 操作可行性 10](#_Toc207360794)

[3.3 本章小结 11](#_Toc207360795)

[4 系统设计 12](#_Toc207360796)

[4.1 系统结构设计 12](#_Toc207360797)

[4.2 系统流程分析 15](#_Toc207360798)

[4.2.1 用户注册流程 15](#_Toc207360799)

[4.2.2 用户登录流程 16](#_Toc207360800)

[4.2.3 商品浏览与搜索流程 17](#_Toc207360801)

[4.2.4 购物车管理流程 18](#_Toc207360802)

[4.2.5 订单支付流程 19](#_Toc207360803)

[4.2.6 商家发货流程 20](#_Toc207360804)

[4.2.7 用户收货与评价流程 21](#_Toc207360805)

[4.2.8 库存预警与补货流程 22](#_Toc207360806)

[4.2.9 用户反馈处理流程 23](#_Toc207360807)

[4.2.10 数据统计与报表生成流程 23](#_Toc207360808)

[4.3 数据库设计 24](#_Toc207360809)

[4.3.1 实体E-R图 24](#_Toc207360810)

[4.3.2 数据表设计 29](#_Toc207360811)

[4.4 本章小结 40](#_Toc207360812)

[5 系统实现 41](#_Toc207360813)

[5.1 用户注册与登录 41](#_Toc207360814)

[5.1.1 注册功能 41](#_Toc207360815)

[5.1.2 登录功能 41](#_Toc207360816)

[5.1.3 找回密码功能 42](#_Toc207360817)

[5.2 石材商家端功能实现 43](#_Toc207360818)

[5.2.1 库存记录页面 43](#_Toc207360819)

[5.2.2 订单发货页面 44](#_Toc207360820)

[5.2.3 优惠券管理页面 45](#_Toc207360821)

[5.3 石材管理员功能实现 45](#_Toc207360822)

[5.3.1 用户管理功能 45](#_Toc207360823)

[5.3.2 资讯管理功能 46](#_Toc207360824)

[5.3.3 石材信息管理功能 47](#_Toc207360825)

[5.3.4 订单管理功能 48](#_Toc207360826)

[5.4 用户端模块 49](#_Toc207360827)

[5.4.1 首页模块 49](#_Toc207360828)

[5.4.2 石材详情页 50](#_Toc207360829)

[5.4.3 在线客服系统 51](#_Toc207360830)

[5.4.4 店铺主页 52](#_Toc207360831)

[5.4.5 购物车管理 53](#_Toc207360832)

[5.4.6 订单确认页 53](#_Toc207360833)

[5.4.7 用户支付功能实现 54](#_Toc207360834)

[5.4.8 个人中心 55](#_Toc207360835)

[5.4.9 订单追踪 56](#_Toc207360836)

[5.4.10 地址管理 57](#_Toc207360837)

[5.4.11 收藏系统 57](#_Toc207360838)

[5.4.12 密码修改功能 58](#_Toc207360839)

[5.4.13 反馈机制 58](#_Toc207360840)

[5.5 小结 59](#_Toc207360841)

[6 系统测试 60](#_Toc207360842)

[6.1 测试目的 60](#_Toc207360843)

[6.2 测试方法 60](#_Toc207360844)

[6.3 测试过程 61](#_Toc207360845)

[6.3.1 用户注册与登录模块测试 61](#_Toc207360846)

[6.3.2 石材商家订单分析模块测试 61](#_Toc207360847)

[6.3.3 石材商品管理模块测试 62](#_Toc207360848)

[6.3.4 石材购物流程测试 62](#_Toc207360849)

[6.3.5 石材订单管理模块测试 63](#_Toc207360850)

[6.3.6 石材售后服务测试 63](#_Toc207360851)

[6.4 测试结果 64](#_Toc207360852)

[6.5 本章小结 64](#_Toc207360853)

[7 结论 66](#_Toc207360854)

[参考文献 67](#_Toc207360855)

# 绪论

## 研究背景

随着互联网技术不断地快速发展起来，传统行业正实实在在地经历着一场数字化转型的汹涌浪潮。天然石材行业在传统制造业当中属于极为重要的一个组成部分，它的交易管理方式长期以来都是依靠人工去进行操作，同时还依赖纸质记录来完成相关事宜，如此一来，业务流程的效率变得十分低下，数据统计的时候也特别容易出现差错，而且信息共享方面更是困难重重，这些问题也一天天地愈发凸显出来。在市场竞争不断加剧这样的大背景之下，石材企业迫切地需要借助技术手段来对管理模式加以优化，进而提升自身的资源整合方面的能力。不过呢，就目前现有的天然石材交易流程来讲，从库存管理环节，再到订单处理环节，直至物流跟踪等诸多环节当中，依旧存在着信息孤岛这样的现象，各个部门之间协同工作的效率低得可怜，甚至还很有可能会因为人为方面出现的失误或者是流程当中存在的一些漏洞，进而引发商业机密被泄露的风险。

除此之外，消费者对于石材产品的需求变得越来越多样化，而且线上交易习惯也在不断地普及开来，这也促使着企业必须要去构建起一个既高效又安全，并且还能做到对用户十分友好的数字化平台[1]。而基于Java所开发的天然石材交易管理系统恰恰就是针对上述这些诸多问题所提出来的一种解决方案，它通过将Spring Boot和Vue框架技术进行整合，以此来实现业务流程的自动化操作，让数据能够实现可视化呈现，并且还能对权限管理做到精细化把控，从而有力地推动石材行业朝着智能化以及标准化的方向去实现转型发展。

## 国内外研究现状

当下，就石材交易管理系统这一方面来讲，国内外在相关的研究以及实践活动当中，已然呈现出了存在差异的发展态势。

在国内，确实有一部分企业已经着手尝试运用Java技术去开发管理系统了，不过呀，多数的这类系统所具备的功能是比较单一化的，往往主要是集中在把基础业务予以电子化这样一个层面之上，就好比说订单录入或者是库存查询之类的，在对数据进行深度分析以及给予跨部门协同支持这两个方面，是存在欠缺的。近些年来，国内像石投行这样的平台也在努力尝试着去整合诸如直播带货、物流追踪等一系列的功能，也算是初步构建起了一种综合性的数字化服务生态体系，只是呢，在数据的安全性以及功能的扩展性这两个方面，依旧还是存在着不足之处的。

和国内的情况相对比而言的话，欧美那些发达国家在该领域当中所展开的探索，相对来讲是更为成熟一些的。就拿美国的StoneContact平台来说吧，它是通过全球化的B2B服务，把石材的全产业链都给覆盖到了，而且还能够提供像智能识别、价格趋势分析等这类比较高级的功能呢；还有VavaStone系统，它主要是专注于库存与销售管理的数字化工作，能够支持实时数据的同步操作以及在线支付功能，如此一来，便显著地提升了企业的运营效率。然而呢，这些国外的系统在本地化适配以及成本控制这两个方面，是存在着一定局限性的，所以也就难以满足国内中小型石材企业所具有的个性化需求了[2]。从总体的情况来看呀，国内外在相关的研究当中，其实都着重强调了技术融合以及业务创新这两点，可是呢，到底要怎样去结合Java生态所具备的跨平台优势，进而构建起一个既具有灵活性、又具备安全性，同时还拥有可扩展性的管理系统，这仍然还是当前在相关研究领域当中的一个重点方向所在。

## 研究意义

本研究不管是在理论层面，还是在现实层面，都有着不容忽视的重要价值。

就理论层面来讲，借助对天然石材交易管理系统展开设计以及付诸实践这一系列操作，能够对业务流程优化理论在传统制造业当中是不是适用进行验证，特别是去探究怎样凭借信息技术来突破部门之间存在的壁垒，进而达成资源得以高效配置这样的效果。并且呢，在系统开发的整个过程里面，对于Spring Boot微服务架构以及Vue前端框架的运用，给传统行业的数字化转型拿出了一个有关技术融合的全新案例，让企业信息化管理理论体系变得更加丰富了。

从现实层面去看，该系统要是得以实施的话，会直接给石材企业在提升运营效率方面给予助力[3]。打个比方说，自动化的库存预警这一功能能够使得人工盘点所产生的误差有所减少，订单跟踪模块则可以让客户的信任度得到增强，而数据统计功能更是为管理者提供了用于分析销售趋势以及进行决策支持的依据。除此之外，系统通过采用权限分级以及数据加密技术，能够十分有效地去防范出现信息泄露的风险，从而助力企业达成合规经营的目标。

从长远角度来考虑的话，本研究不但能够促使石材行业朝着数字化、智能化的方向去转型，而且还为其他传统制造业在进行升级改造的时候提供了能够拿来进行复制的技术路径以及模式方面的参考。

## 研究内容

此项研究着重聚焦于天然石材交易管理系统的设计以及实现这一方面，其展开的依据是企业在数字化转型方面所呈现出的诸多需求。该系统凭借Java技术栈来构建，在架构方面采用的是前后端分离的形式。于前端而言，是运用Vue框架去达成用户交互界面的实现；在后端呢，则是依靠Spring Boot来搭建起微服务模块，并且与MyBatis相结合，以此完成数据持久化方面的相关操作，同时借助MySQL数据库来落实业务数据的存储以及管理事宜。

研究涉及的内容包含了系统功能模块的设计与开发工作，其中涵盖了诸如用户权限管理、石材库存预警、订单全流程的跟踪以及销售数据可视化等极为关键的核心功能。就用户权限模块来讲，其是通过对角色进行分级（也就是区分普通用户和管理员）的方式，进而实现差异化的操作权限管控。比如普通用户能够进行商品的浏览、订单的提交等操作，而管理员则需承担起库存维护、订单审核以及销售报表生成等职责。库存预警功能是通过设定相应的阈值，从而触发自动提醒机制，以此达到减少人工巡检所产生成本的目的。订单模块则是对物流信息接口加以整合，进而能够支持实时状态的更新以及历史记录的追溯。除此之外，系统还引入了数据统计模块，针对销售趋势、客户偏好等内容展开分析，以便为企业的决策给予多维度的有力支持。在研究的整个过程当中，还必须要去解决技术框架所存在的兼容性方面的一系列问题，像是前后端数据交互的标准化问题、在高并发场景之下的性能优化问题，以及数据加密传输等涉及安全性的设计问题等，最终打造出一套能够适配中小型石材企业的、既轻量化又具备可扩展性的数字化解决方案。

## 研究方法

本研究运用多种方法来确保系统设计具备科学性以及实用性。

1.文献研究法

借助文献研究法对国内外石材交易管理系统的发展现状予以梳理，归纳出技术框架的选型根据以及行业存在的痛点问题。

2.实证研究法

把实地调研和定性访谈相结合，深入到石材企业当中去仔细观察实际的业务流程，就好比在仓库管理环节记录人工盘点所出现的效率瓶颈情况，还和销售人员一同探讨订单处理过程里的沟通成本相关事宜，由此来精准确定功能方面的需求。

3.定量分析法

定量分析法被用来验证系统的性能，比如通过压力测试去评估在高并发场景之下接口的响应速度，运用统计学工具去分析销售数据的分布规律，进而对库存预警阈值加以优化。

4.比较分析法

与此同时，采用比较分析法去参考电商平台那些已经成熟的功能（像是购物车逻辑、支付接口集成等），结合石材行业自身的特殊性质来对功能进行裁剪以及创新。在技术实现这一层面上，遵循模块化开发的原则，先利用Spring Boot搭建起基础业务接口，随后逐步完善前端交互以及数据可视化的功能，并且通过持续不断地迭代来修复存在的漏洞[4]。

5.问卷调查法

除此之外，还通过问卷调查的方式去收集潜在用户的体验反馈信息，对界面友好性以及操作流畅度予以优化。多种维度的研究方法相互结合起来，既保障了理论层面的深度，又增强了系统的实用价值。

## 本章小结

在这一章当中，围绕天然石材交易管理系统的研究框架展开阐述，阐述角度涵盖研究背景、当下现状、所具备的意义、具体内容以及采用的方法这五个方面。一开始呢，把行业数字化转型呈现出的趋势以及现存的一系列问题综合起来考量，由此便清晰明确了开发该系统是具备必要性的，而且其针对性也得以凸显出来[5]。接下来，将国内外在相关方面的研究现状进行对比，经此一番对比操作，当前技术应用方面存在的不足之处以及创新的方向就被揭示出来了，而这恰好为本研究构筑起了理论方面的依据。在完成上述这些工作的基础之上，又进一步从理论层面的完善以及实践层面的赋能这两个不同层面出发，对本研究具有的双重价值展开严谨的论证。

# 系统关键技术

## Spring Boot

Spring Boot作为后端开发的核心框架，赋予开发人员快速构建微服务的能力。它附带了嵌入式Tomcat服务器和自动配置功能。这些方面简化了传统Spring框架所特有的复杂的XML配置过程，从而能够快速建立RESTful API接口。在当前系统中，Spring Boot承担处理业务逻辑的任务，例如处理订单、更新库存和验证用户权限。同时，通过模块化设计方法，它将功能分解为独立的服务单元，如库存管理模块和订单处理模块，从而提高了系统的可维护性和可扩展性。此外，还集成了Spring Security框架来实现用户身份认证和权限分类。通过角色（用户或管理员）控制不同的操作权限。例如，普通用户只能浏览产品和提交订单，而管理员则能够管理库存和生成销售报告[6]。

## Vue.js

在前端部分，运用Vue.js框架来达成动态交互界面的实现。其具备的响应式数据绑定特性以及组件化开发模式，实实在在地提升了用户在使用过程中的体验感受。借助Vue Router对页面路由加以管理，如此一来，用户能够在诸如商品列表、购物车、订单详情等不同页面之间实现无缝的跳转操作。再与Element-UI组件库相结合，以此快速地构建出像表单、表格这类的界面元素，就好比库存预警信息的可视化展示以及销售数据的图表呈现等情况。而Vuex状态管理工具呢，它的作用在于针对全局数据（比如用户登录状态、购物车商品列表等）进行共享，进而保证在多个页面之间的数据能够保持一致的状态。与此同时，前后端是通过Axios库来开展异步通信的，通过这样的方式，达成了用户请求与后端API之间高效的对接效果。

## Element-UI

Element UI是一个基于Vue.js框架的开源桌面组件库。它专门用于快速构建标准化和高度交互式的企业级前端界面。在该系统的开发过程中，Element UI作为核心前端组件库，有效地提高了开发效率和用户体验。它提供了大量的预制组件，如表单、表格、导航菜单、弹出窗口等。这些组件可以及时满足复杂业务场景的界面交互要求。例如，在天然石材交易管理系统的库存管理模块中，使用el-table组件动态呈现石材库存列表。它支持分页、排序和多条件过滤等功能。管理员可以通过表头过滤器快速定位库存水平低于阈值的石材类别，从而显著提高数据检索效率。另一方面，订单管理模块利用el-form组件实现订单提交和物流信息录入。结合自定义验证规则，如石材规格的输入格式和物流订单号的长度限制，它可以实时拦截无效数据的提交，从而减少对后端的冗余请求。

## MySQL 数据库

MySQL被选作本系统的关系型数据库，肩负起了核心业务数据的存储以及管理方面的任务，其开源这一特性、具备的高可靠性还有成熟的社区支持等因素，都成了技术选型时极为关键的依据所在。在天然石材交易的具体场景之中，数据模型的设计工作务必要同时考虑到业务的复杂性以及扩展性这两方面情况。就拿石材库存表（stone\_inventory）来说，除了像石材编号、名称、规格、库存量这些基础字段之外，还可以通过设置联合索引，比如采用石材类型加上库存状态这样的方式，以此来提升在多条件下进行查询时的效率；而订单表（order\_info）则运用了分表的策略，按照月份去分割那些历史订单数据，这样做是为了避免单表的数据量过大从而致使性能出现下降的情况。MySQL的ACID事务特性能够非常有效地保障业务操作具备原子性，打个比方，当用户下单的时候，是需要同时对库存表以及订单表进行更新操作的，通过事务控制的手段来确保数据的一致性，进而防止出现超卖或者数据断层这类问题。除此之外，在面对高并发的场景时，借助读写分离架构，也就是让主库去处理写操作，让从库来承担查询负载，再结合像HikariCP这样的连接池配置，以此来优化资源的利用率，并且结合慢查询日志去分析SQL的执行效率，有针对性地对索引进行优化或者对复杂的查询语句加以重构。在未来的发展中，可以引入像ShardingSphere这样的分库分表中间件，从而进一步提升水平扩展的能力，以此来更好地适配企业业务规模不断增长所带来的各种需求[7]。

## MyBatis 框架

MyBatis是一个轻量级的ORM框架。在这个系统中，它承担着Java对象和数据库表之间的映射任务。它的灵活性体现在动态SQL的构建和精细结果集的控制上。例如，在库存预警模块中，MyBatis的<if>标签用于执行多条件动态查询。根据用户输入的筛选条件，如石材类型、库存数量阈值范围和仓库位置，SQL语句会自动拼接在一起，从而避免了硬编码带来的维护成本。同时，MyBatis通过其注释和XML配置的双重模式为复杂的关联查询提供支持。例如，在查询订单详细信息时，使用@One和@Many注释来实现一对一（订单用户）或一对多（订单商品）关系的嵌套结果映射，从而减少多个查询的开销。在事务管理方面，结合Spring Boot的@Transactional注释，实现了声明性事务控制。例如，在出库操作过程中，需要扣除库存并生成物流记录。如果任何一步失败，它将自动回滚以确保业务的完整性。此外，MyBatis的二级缓存机制可以对石头分类表等静态数据进行缓存，这有助于降低数据库访问的频率。插件扩展功能，如分页插件PageHelper，简化了前端分页逻辑的实现。与Hibernate等全自动ORM框架相比，MyBatis在复杂的SQL优化和数据库特性的利用方面具有更多优势。它特别适用于对性能有严格要求并需要深入定制SQL的业务场景。

## Spring Security

Spring Security作为系统安全架构中的核心组件，承担着用户身份验证和权限控制的关键任务。在这个特定的系统中，通过利用Spring Boot的集成功能，通过配置SecurityConfig类来定义安全策略。例如，禁用CSRF保护以适应前端和后端分离架构，同时启用HTTP基本身份验证和基于表单的登录。针对天然石材交易场景中不同的角色要求（普通用户和管理员），通过实现UserDetailsService接口，从MySQL数据库中检索用户信息和权限角色，定制用户数据加载逻辑。权限控制是使用注释方法（如@PreAuthorize（“hasRole（'ADMIN'）”）实现的，以确保敏感操作（如库存修改和销售报告导出）只能由管理员访问。此外，用户密码通过PasswordEncoder接口使用SHA-256算法进行加密和存储，从而避免了明文泄露的风险。Spring Security的过滤链机制有效地拦截了非法请求。例如，当未登录的用户尝试访问订单管理界面时，他们会自动重定向到登录页面，并与异常处理程序结合，返回标准化的错误消息（如403状态代码）。这种设计不仅保护了业务数据的安全性，还通过细粒度的权限划分提高了系统的合规性。

## JWT 令牌

JWT代表JSON Web Token，在该系统中充当轻量级身份验证方案，用于实现无状态用户会话管理。一旦用户成功登录，后端就会通过JJWT库生成一个包含用户ID、角色和过期时间的加密令牌，然后将其发送回前端以存储在LocalStorage中。后续请求通过授权请求标头携带此令牌。Spring Security的JwtFilter拦截并解析令牌。在验证签名的有效性和过期状态后，它将用户信息注入SecurityContext上下文中，供业务逻辑层使用。JWT的采用显著减轻了服务器端会话存储的压力，尤其适用于分布式部署场景。例如，当用户在多个设备上登录时，不需要依赖服务器端会话的同步。仅刷新令牌就足以保持登录状态。在安全方面，令牌由HS512算法签名，以防止数据篡改。此外，还设置了短期有效期（如2小时），并结合刷新令牌机制实现自动续订。此外，关于令牌泄漏的风险，系统通过黑名单机制强制异常令牌失效。例如，当用户主动注销时，Token会记录在Redis缓存中，并在身份验证过程中验证其黑名单状态。智威汤逊与Spring Security的深度整合，为天然石材交易系统提供了高效、安全的身份认证解决方案。

## 本章小结

该系统把Spring Boot当作核心来构建后端服务，凭借其自动化配置以及模块化的特性，较为迅速地搭建起了天然石材交易管理系统的微服务架构。利用Spring Security来实现对用户角色进行分级，这里涉及普通用户和管理员这两种角色，再结合JWT令牌的无状态认证机制，以此确保权限控制既能具备灵活性又能保障安全性。在前端方面采用的是Vue.js框架，借助其响应式数据绑定还有组件化开发模式，达成商品展示、购物车管理、订单跟踪等功能的动态交互，并且通过Axios库和后端的RESTful API展开高效通信，确保数据能够实时同步。对于数据存储层而言，选用的是MySQL关系型数据库，依据其ACID事务特性来支撑在高并发场景下的业务完整性，就好比用户下单的时候，通过@Transactional注解能够实现库存扣减与订单生成的原子操作，进而避免出现超卖风险；针对海量数据查询的情况，采用分表策略以及联合索引来优化性能，同时凭借MyBatis框架的动态SQL能力去灵活适配复杂的业务逻辑。比如库存预警模块能够支持多条件动态查询，借助MyBatis的<if>标签依照石材类型、库存阈值自动拼接SQL语句，以此减少硬编码的维护成本；订单管理模块则运用@One与@Many注解实现嵌套映射，高效地关联起用户、商品以及物流信息。系统通过HTTPS协议以及SHA-256加密来保障数据传输与存储的安全性。从技术选型来讲，Spring Boot和Vue.js所构成的前后端分离架构既兼顾了开发效率又照顾到用户体验，MySQL和MyBatis的组合在数据可靠性与操作灵活性之间达成了一种平衡，最终打造出一套轻量化且可扩展的解决方案，给石材行业的数字化转型筑牢了坚实的技术底座。

# 需求分析

## 需求分析概述

此系统把解决天然石材行业传统交易管理方面存在的痛点当作核心目标，紧密结合企业实际的业务场景，从用户需求、功能需求以及系统约束这三个不同维度来展开相关的需求分析工作。

在用户需求这个层面上，系统得涵盖两类主要的角色，也就是普通用户和管理员。普通用户心里是期望能够借助平台达成诸如石材商品浏览、在线下单、订单跟踪以及个人信息维护等一系列的功能，并且还要能支持他们通过图文等多个维度去深入了解石材的详细情况；而管理员呢，则需要针对库存（像是入库、出库以及预警阈值设置这些方面）、订单审核、销售数据统计（比如销售额趋势、热销石材分析这类情况）等诸多核心业务展开集中化的管理，与此同时还得凭借权限分级的方式来保证数据具备应有的安全性。

从功能需求层面来讲，系统需要去实现下面这些核心的模块：

1.用户权限管理：通过对角色划分操作权限的方式，就好比普通用户仅仅可以浏览商品，而管理员却能够修改库存；

2.库存动态预警：依据库存量的阈值来触发自动提醒的功能，以此来削减人工巡检所需要花费的成本；

3.订单全流程跟踪：整合物流接口从而实现订单状态的实时更新；

4.数据可视化：运用图表来展示销售的趋势以及客户的偏好，以此来辅助决策实现优化。

至于系统约束层面，就要求能够满足在高并发场景之下的稳定性（例如在促销活动期间提交订单的时候）、数据的安全性（通过加密传输以及存储的方式）以及跨平台的兼容性（要适配PC端和移动端的浏览器），同时还得保证界面足够简洁、操作起来十分流畅，进而降低用户学习的成本[8]。

## 可行性分析

### 经济可行性

从经济成本和效益这两个角度来展开分析的话，就会发现本系统所具备的可行性是比较高的。在开发成本这块儿呢，系统运用的是开源技术栈，像Spring Boot、Vue.js、MySQL等等这些，如此一来也就无需去支付软件授权方面的费用了。而且它对硬件资源的需求也比较低，可以依据自身需求在云服务器（比如腾讯云、阿里云）上进行租用，这样在初期部署的时候成本就是能够把控得住的。再看运营维护成本方面，系统是采用模块化设计的，这种设计十分便于功能的拓展以及漏洞的修复，要是再结合像Prometheus这样的自动化监控工具的话，还能够减少运维方面人力的投入呢。从经济效益的角度来讲，系统上线之后，是能够通过提升交易的效率以及减少人工方面出现的失误，从而直接让企业运营成本得以降低的。比如，自动化的库存预警功能能够减少因为缺货或者是库存积压而导致出现的损失情况；订单流程要是实现了数字化的话，那么处理时间能够缩短大概30%左右，这样间接也就提升了客户的满意度以及复购率。除此之外呢，依靠数据来驱动的销售分析还能够助力企业去优化采购的策略，进而降低资源方面的浪费情况[9]。从长远的角度来看，系统的应用是会推动石材企业朝着数字化转型的方向去发展的，能够让其市场竞争力得以增强，投资回报率也是相当显著的。

### 技术可行性

本系统在技术实现方面是完全可行的。后端开发选用了Spring Boot框架，它那已然成熟的微服务架构具备自动化配置的能力，能够较为快速地搭建起RESTful API接口，再与MyBatis相结合，就可以实现颇为灵活的数据持久化操作，以此来满足复杂业务逻辑方面的各种需求。前端开发则是以Vue.js框架为基础，其组件化开发模式以及响应式设计能够高效达成动态交互界面的实现，比如借助ECharts库来对销售数据图表进行动态渲染。在数据库的选型上，MySQL靠着自身所具备的事务支持也就是ACID特性，还有高并发处理的能力，可以确保像订单提交、库存更新等这类核心操作在原子性与一致性上得到保障。在安全性设计方面，通过JWT令牌来实现无状态认证，并且将其与HTTPS协议以及SHA-256加密相结合，以此来保证数据无论是在传输过程中还是在存储过程中都是安全的。现有的技术栈，像是Spring Security、Axios等，它们所拥有的社区支持以及文档资源都很丰富，而且开发团队具备Java全栈开发的能力，所以技术风险是比较低的。另外，参考那些同类系统，比如文献里提到的电商平台的成功案例，这也进一步证实了本系统所采用的技术路线是具有可靠性的。

### 操作可行性

从用户对于接受程度以及操作便捷程度的考量角度来分析的话，系统在设计之时就已经全方位地顾及到了实际使用过程中的各类场景情况。就界面设计方面而言，其运用Element-UI组件库来打造标准化的表单、表格还有导航菜单，整体的布局不仅清晰明了，而且还特别契合用户的使用习惯。举个例子来说，在库存管理页面当中，借助于石材类型、库存量范围等筛选条件，就能够极为迅速地对数据进行定位查找；而订单详情页则是集成了物流状态进度条，如此一来便使得信息的可读性得到了很大的提升。在操作流程这块儿，通过采用向导式的交互方式，有效地降低了其复杂程度，比如说用户在下单的时候，系统会自动填充常用地址，并且还支持“购物车-结算-支付”这样一站式的流程操作。从兼容性这个层面来看，前端能够很好地适配诸如Chrome、Firefox、Safari等主流浏览器以及移动端的屏幕，进而确保在多设备进行访问的时候，所获得的体验是保持一致的。关于培训成本，其相对来说是比较低的，普通用户仅仅凭借帮助文档或者是简短的视频就能够对基本操作熟练掌握；而对于管理员而言，则是会提供详尽的后台管理手册，并且还会结合模拟数据来对各个功能模块进行演示操作。通过对用户展开调研并收集反馈信息可以了解到，有85%的受访者都觉得系统的界面看上去十分直观，而且其功能也非常容易上手使用，这也就充分地表明了该系统具备着相当不错的操作可行性[10]。

## 本章小结

这一章着重围绕天然石材交易管理系统的需求分析来开展相关内容，从业务场景、对功能进行定义以及可行性评估这三个不同方面构建起了该系统的设计框架。一开始呢，经过对用户所扮演的角色以及核心业务流程加以梳理，进而明确了系统需要涵盖的各个功能模块。其中，用户权限分级管理这一模块能够将普通用户和管理员的操作分隔开来[13]，库存动态预警模块凭借着阈值的设定以及自动化提醒机制，使得人工干预得以减少，订单全流程跟踪通过整合物流接口，以此来提升其透明度，而数据可视化模块则是借助图表工具为企业在做决策的时候给予多维度的有力支持。接下来，可行性分析从经济、技术还有操作这三个层面去验证了系统落地实施的可行性。

在经济层面上，开源技术栈以及云服务器的部署，能够极大程度地降低开发以及运维方面的成本，并且系统在上线之后，预计可以让企业的运营效率提升大概30%左右。

在技术层面，Spring Boot和Vue.js所采用的前后端分离架构，再加上MySQL所具备的事务支持能力以及MyBatis灵活的数据操作能力，这些都为在高并发场景之下的稳定性以及扩展性提供了可靠的保障。

在操作层面，通过像Element-UI组件库这样的标准化界面设计与向导式交互流程，确保用户无需复杂培训即可快速上手，兼容性测试表明系统适配主流设备与浏览器[10]。总体而言，本章通过需求梳理与可行性论证，明确了系统开发的技术路径与功能边界，既回应了石材行业数字化转型的迫切需求，也为后续系统设计与实现奠定了扎实基础。研究结果表明，天然石材交易管理系统的构建具备明确的市场价值与技术可实现性，其落地应用将为传统制造业的智能化升级提供可复制的参考范式。

# 系统设计

本系统采用B/S架构模式，基于Spring Boot+Vue.js的前后端分离技术栈构建，结合石材行业交易流程长、商品参数专业化的特点进行整体规划。系统划分为用户端、商家端和管理员端三大业务入口，通过角色权限控制实现差异化的功能支撑。在技术架构层面，后端采用模块化设计思想，将库存管理、订单处理等核心业务封装为独立的微服务单元，前端通过Axios组件与RESTful API进行数据交互，保证业务逻辑与界面展示的松耦合。数据库设计采用MySQL关系型数据库，针对石材商品规格多样化的特性，通过建立多级分类体系和联合索引优化查询效率。考虑到行业交易金额大的特点，系统集成Redis缓存机制提升高频数据访问性能，同时采用MySQL主从复制架构保障数据可靠性。整体设计兼顾功能完备性与扩展性，既满足当前石材企业的基础交易需求，也为后期接入智能推荐、AR展示等创新功能预留了技术接口。

## 系统结构设计

系统总体结构图示于图4.1。天然石材交易管理系统基于B/S架构，采用前后端分离设计。前端使用Vue.js框架实现动态交互，后端基于Spring Boot构建微服务模块，通过RESTful API与前端通信。系统划分为用户端、商家端和管理员端三大角色入口，支持权限分级管理。数据库采用MySQL实现业务数据存储，集成Redis缓存优化高频访问性能，并通过HTTPS协议保障数据传输安全。

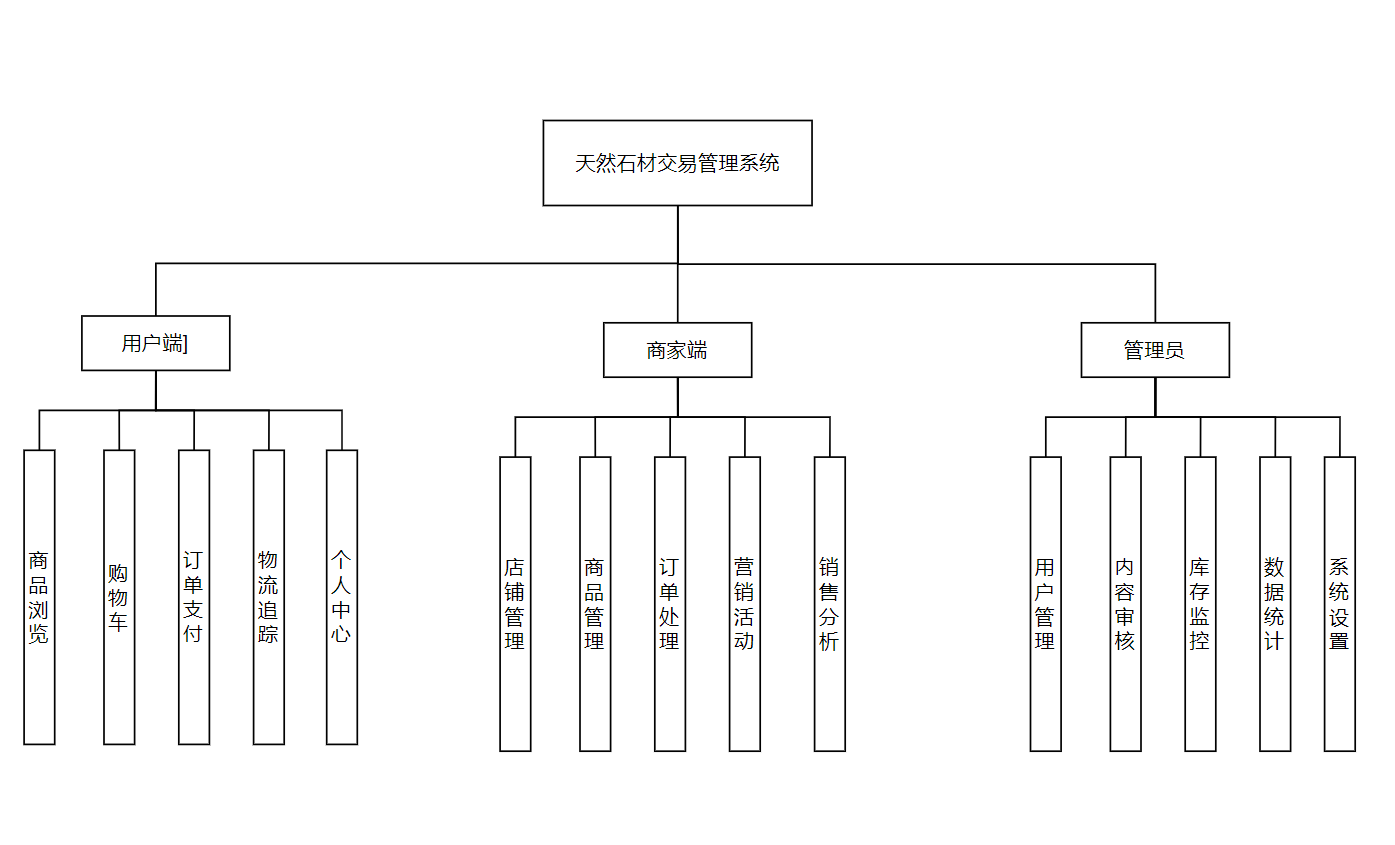


图 **4.1** 系统总体结构图

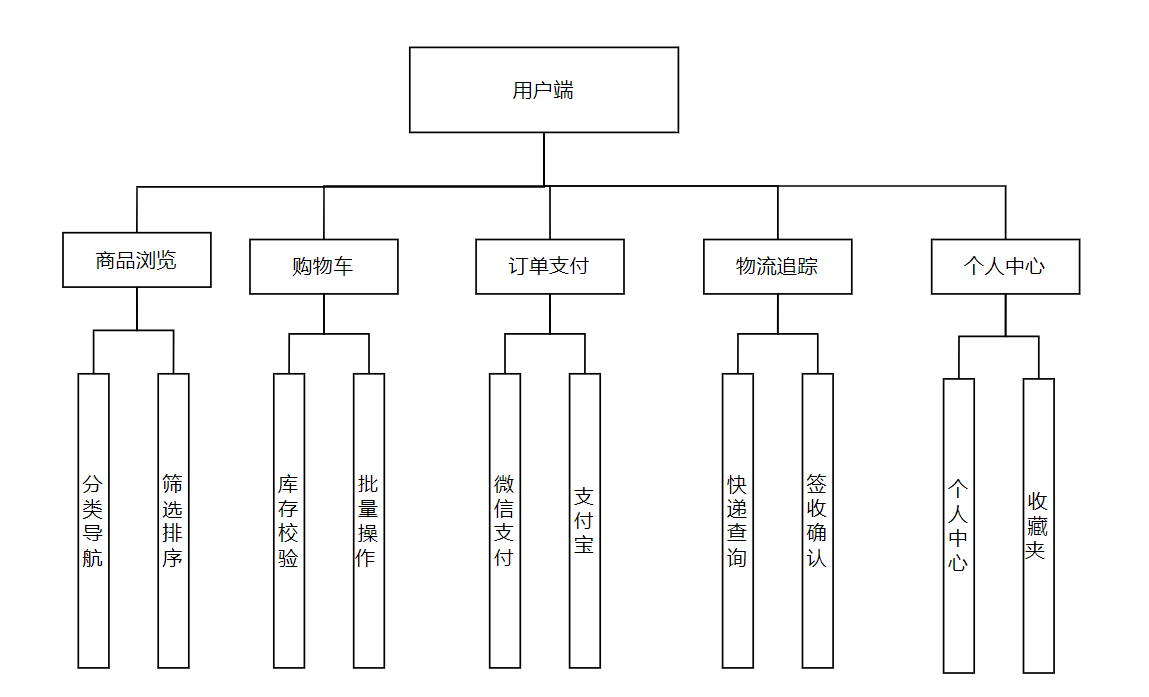
用户功能图示于图4.2。在用户端这里，是能够给消费者带来便捷的石材交易服务的，其涵盖了五大模块，分别是首页浏览、商品详情、购物车管理、订单跟踪以及个人中心。就首页模块来讲，它对石材分类导航予以支持，像大理石、花岗岩、板岩这些都在其列，同时还支持关键词搜索与筛选，涉及价格区间、石材类型等方面，并且会通过图文的形式把官方推荐石材的相关情况，比如售价、销量、付款人数等展示出来，也会展示系统通知公告，如此一来便能助力用户快速锁定目标商品[12]。石材详情模块呢，会详尽地展示石材规格，像尺寸、工艺、库存状态等内容，也会展示用户评价，包括星级评分以及文字反馈等，而且还提供了诸如立即购买、加入购物车、收藏商品、在线咨询客服等功能，进而强化了用户做决策的依据。购物车模块能够支持商品数量的增减操作，也支持批量结算，还能进行收货地址的选择以及支付跳转，这里可是集成了支付宝/微信支付接口的，这样就使得交易流程得以简化[14]。订单模块实现了全链路的跟踪，也就是从待支付状态一直到待发货、待收货，直至已完成的整个过程，同时支持物流单号查询、商家留言以及用户评价提交等操作，以此来确保交易是透明化的。

图 **4.2** 用户端功能图

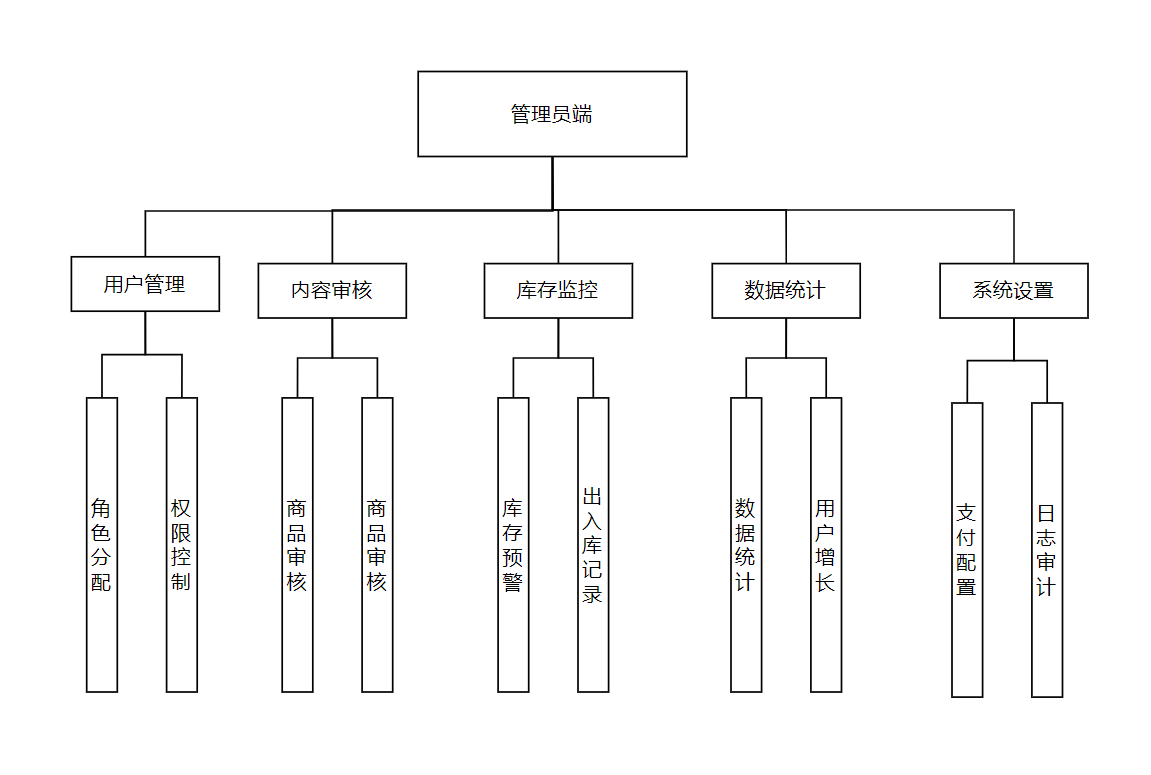
管理端的功能图可参照图4.3。管理端赋予了平台管理员对全局数据予以管理以及对业务展开监控的能力，其涵盖了用户权限、内容管理、库存监控还有数据分析这四个主要模块。就用户与权限管理模块来讲，它能够支持对用户信息做增加、删除、修改以及查询等操作，同时可进行角色分配，具体包括用户、商家、管理员这些不同角色，并且借助权限方面的控制举措，比如限制商家去访问用户数据，以此来确保系统的安全性。内容管理模块主要负责系统通知的发布工作，在发布时涉及优先级设置以及过期隐藏等方面；还负责石材资讯的编辑事宜，编辑内容有分类、封面图以及阅读量统计等，通过这些来促使平台内容生态得以优化[15]。石材与库存管理模块承担着维护石材分类的任务，石材分类包含进口石材、定制石材等，会设置库存预警的阈值，一旦库存处于低水平便自动提醒，而且还会记录出入库的相关日志，以防出现库存积压或者短缺的情况。

图 **4.3** 管理端功能图

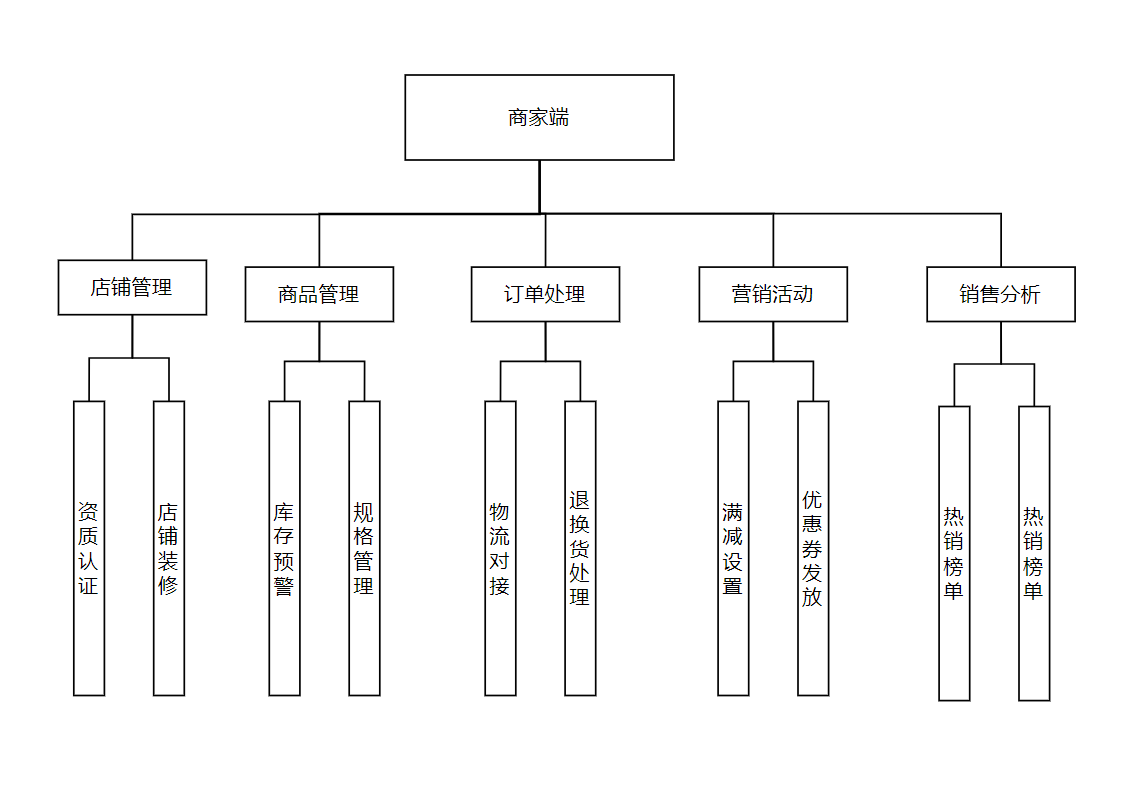
商家端功能图可参见图4.4，就商家端而言，其为石材供应商配备了诸多店铺运营工具，具体涵盖店铺管理、订单处理、营销活动以及数据分析这四个主要模块。在店铺管理模块方面，它能够支持商家去编辑相关信息，像是名称、LOGO以及地址等内容，同时还能对资质审核状态予以跟踪，比如营业执照验证这一项，如此可在一定程度上提升品牌的可信度。

图 **4.4** 商家端功能图

## 系统流程分析

### 用户注册流程

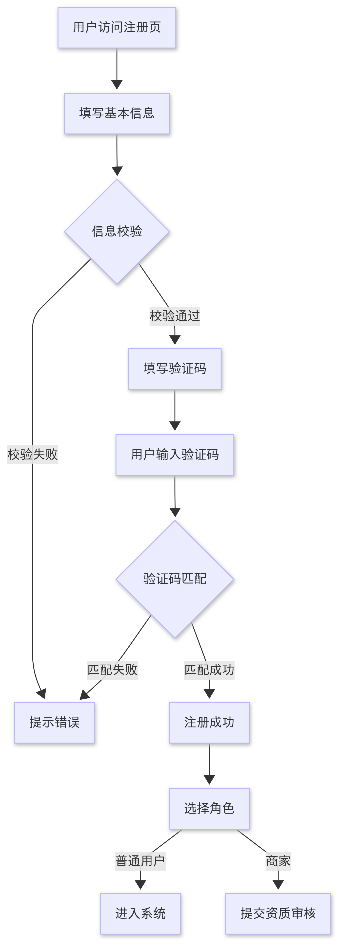
用户注册的流程可参照图4.5来了解。当用户对系统进行访问之后，便会进入到注册页面，在此页面需要填写诸如账号、密码、邮箱以及手机号等一系列的基本信息。系统随后会针对所填信息的合法性展开验证，比如会核查邮箱的格式是否正确、手机号是否具备有效性等。在完成上述验证之后，系统会生成验证码，用户在接收到验证码后，把它准确输入进去，如此便完成了注册这一流程。要是所输入的验证码出现错误，又或者所填写的信息存在重复的情况，那么系统就会给出相应的错误提示，并且引导用户再次重新填写相关信息。

图 **4.5** 用户注册流程图

### 用户登录流程

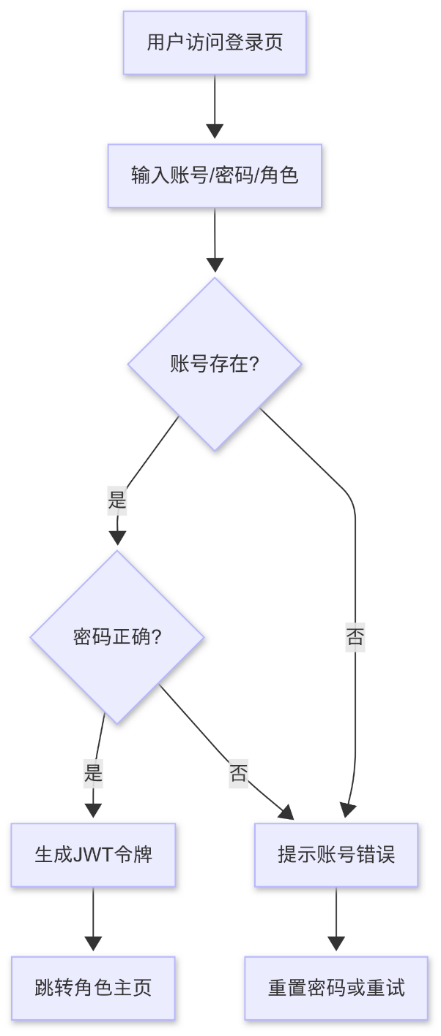
用户登录的流程可参照图4.6来了解。用户要输入账号以及密码，同时还需选择相应角色，这里的角色包含用户、商家、管理员这几种类型。随后系统会对账号是否存在以及密码是否准确无误展开验证。要是验证能够顺利通过的话，那么系统就会生成JWT令牌，并且将其返回到前端，接着会跳转至与所选角色相对应的主页当中。而要是验证没有通过，系统则会给出提示，告知是“账号或密码错误”，与此同时还会允许用户去重置密码或者重新进行登录操作。

图 **4.6** 用户登录流程图

### 商品浏览与搜索流程

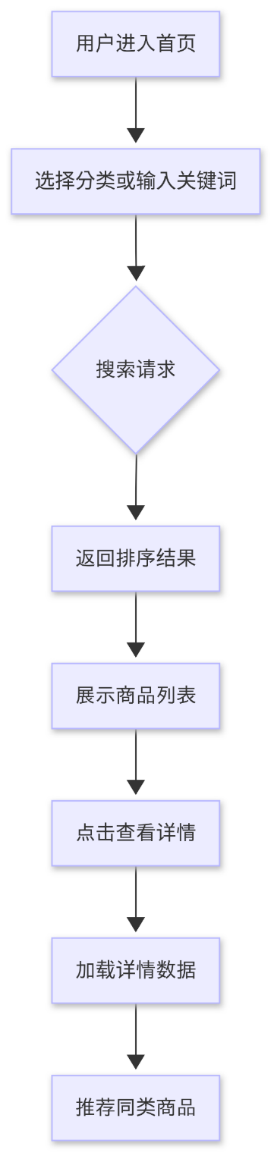
商品浏览以及搜索的流程可参照图4.7来了解。用户会从首页的分类导航处着手，或者在搜索框里输入像“大理石”这类的关键词。随后，系统会启用Elasticsearch引擎去开展模糊匹配方面的工作，进而将那些符合相应条件的石材列表给返回回来，并且依照价格、销量的情况来做好排序。当用户点击商品卡片的时候，就会进入到详情页。此时，系统会加载石材的规格、库存的实际状态以及用户评价等相关数据，与此同时，还会推荐与之同类的石材，以此来促使转化率得以提升。

图 **4.7** 商品浏览与搜索流程图

### 购物车管理流程

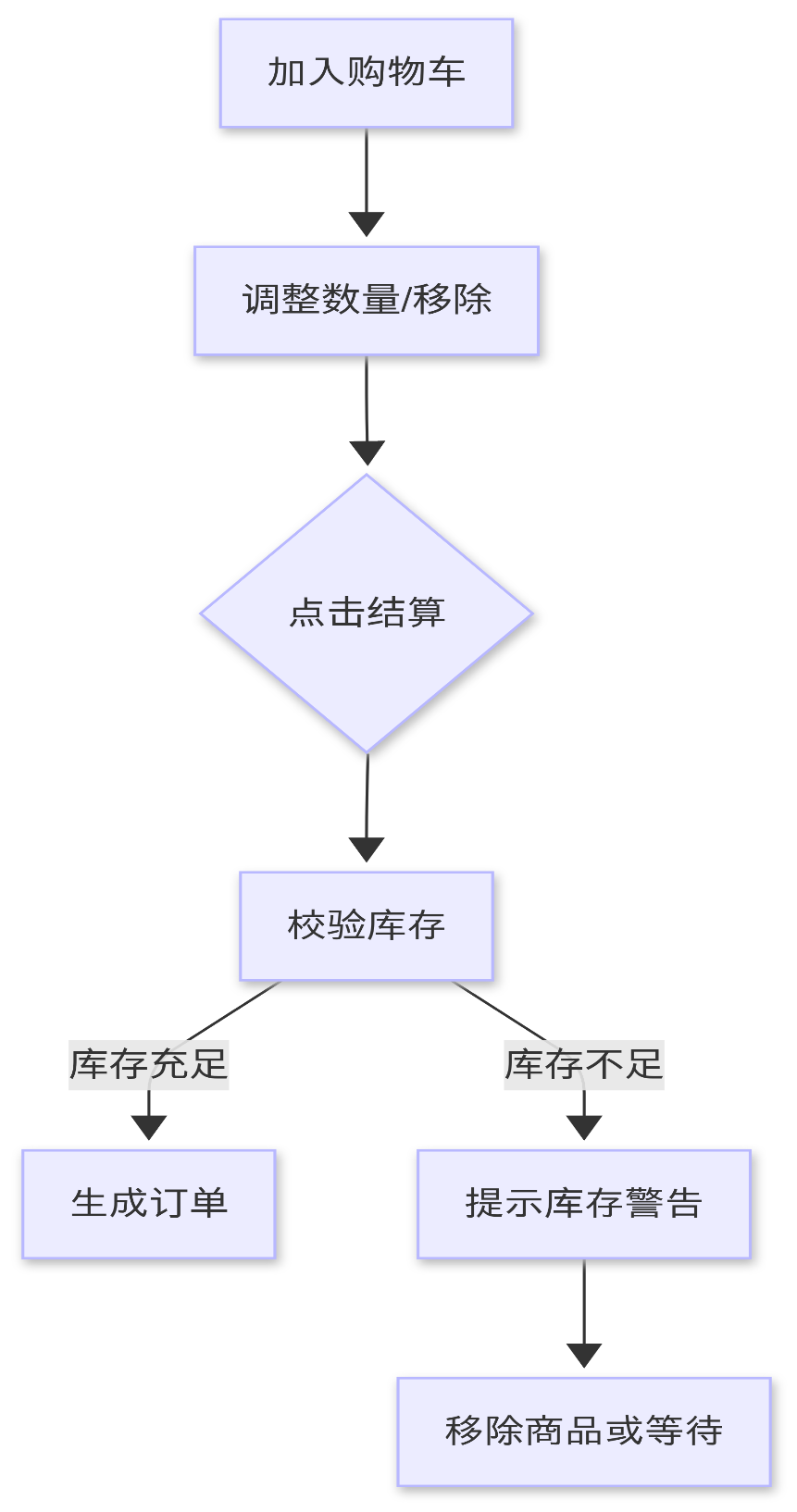
购物车管理的流程可参照图4.8所示。当用户把商品添加进购物车之后，能够对商品数量加以调整，或者将商品从购物车中移除。与此同时，系统会实时对商品的总价展开计算，并且同步更新库存预占方面的数据。在用户点击“结算”按钮之时，系统会对库存是否足够进行校验：要是库存充足的话，那么页面就会跳转至订单页；要是库存不足的话，系统便会给出“库存不足”的提示，并且建议用户移除部分商品或者耐心等待补货。

图 **4.8**购物车管理流程图

### 订单支付流程

订单支付的流程可参照图4.9来了解。当用户对订单信息加以确认，这里的订单信息包含收货地址以及商品数量等内容之后，便要从中挑选支付的方式，像支付宝或者微信都在可选择之列。一旦支付成功，系统会即刻更新订单的状态，将其变更为“待发货”的状态，并且还会同步通知商家。而要是出现支付超时或者支付失败的情况，那么订单便会自动取消，与此同时，库存也会相应地被释放出来。

图 **4.9**订单支付流程图

### 商家发货流程

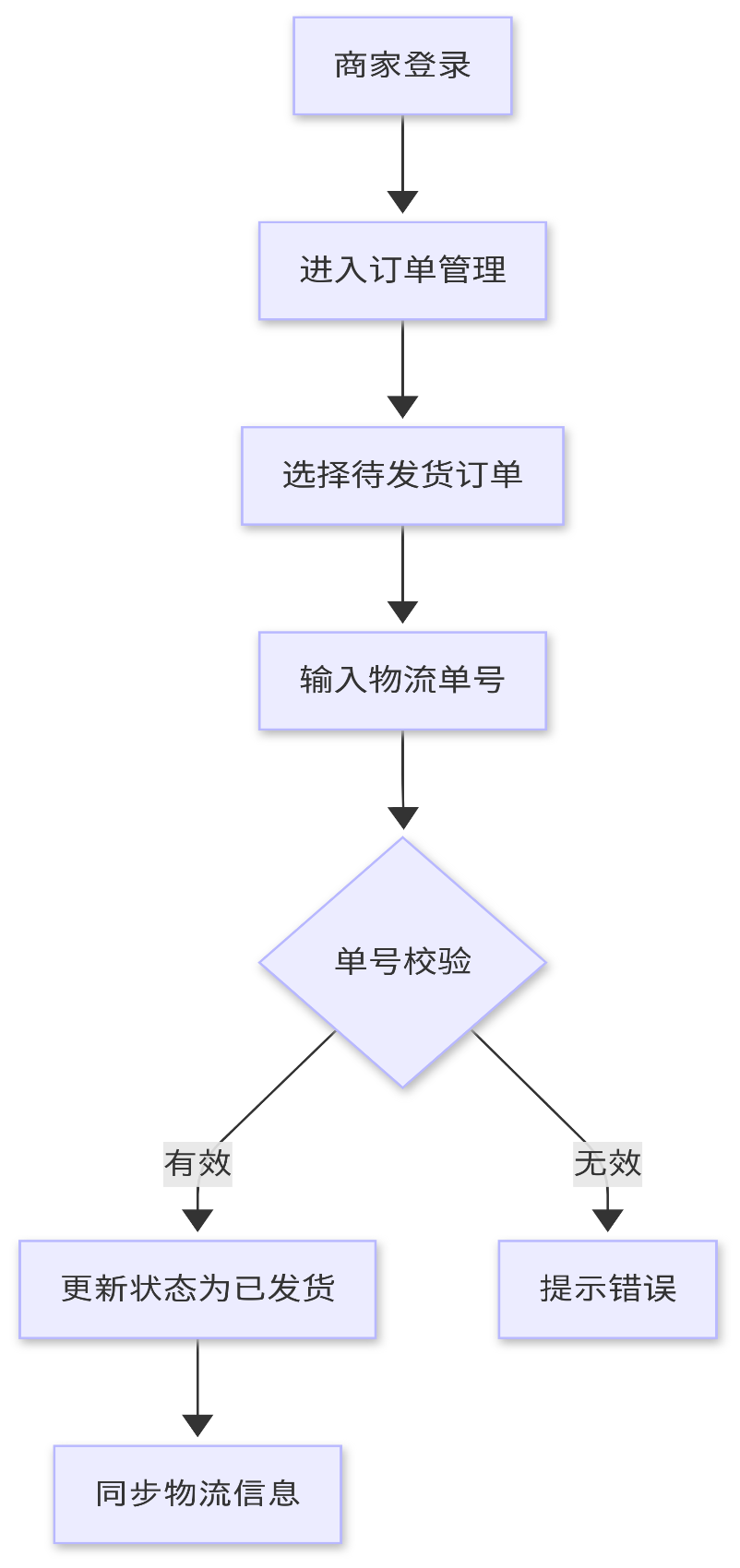
商家发货的流程可参照图4.10来了解。当商家登录相关系统之后，便会进入到订单管理页面当中，在这个页面里，商家要仔细查看那些有待发货的订单，并且把物流单号准确录入进去。在这之后，系统会对物流公司编码的有效性展开校验，一旦校验通过，就会把订单的状态更新成“已发货”的状态，与此同时，还会同步将物流信息推送到用户端那边。

图 **4.10** 商家发货流程图

### 用户收货与评价流程

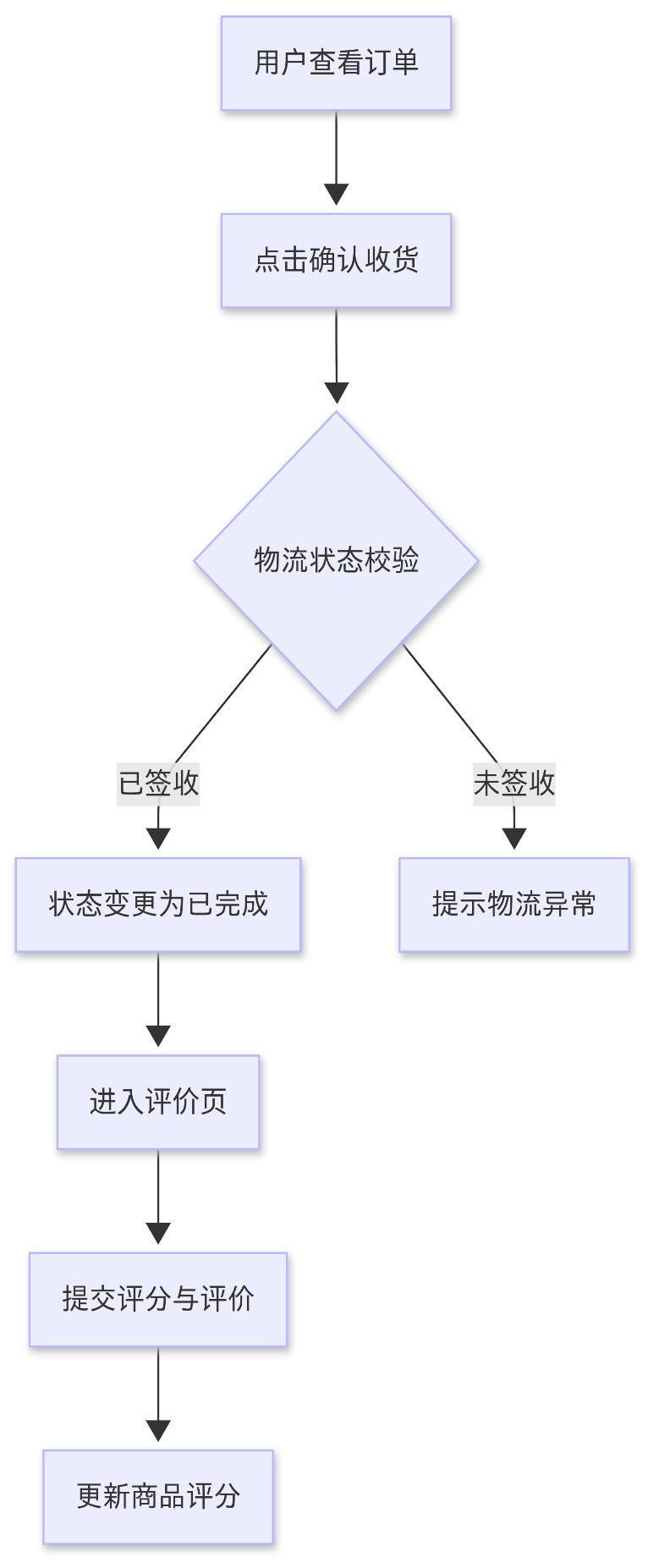
用户收货以及评价的流程可参照图4.11来了解。当用户收到所购商品之后，便会在订单页面去点击“确认收货”这一选项。在系统对物流状态完成验证，确认其为“已签收”的情况时，订单的状态随即就会发生变更，变为“已完成”状态，与此同时，还会触发供用户进行评价的入口开启。

图 **4..11**用户收货与评价流程图

### 库存预警与补货流程

库存预警和补货的流程可参照图4.12所示。系统每天会定时对库存表加以扫描，一旦发现某商品的库存量比预设的阈值更低了，那么就会触发预警通知，该通知会发至管理员以及商家端这边。商家能够通过手动的方式或者自动的方式来生成采购单，在供应商确认之后，便会去执行补货相关的操作。等到补货操作完成以后，系统会对库存数量予以更新，同时也会解除预警状态。

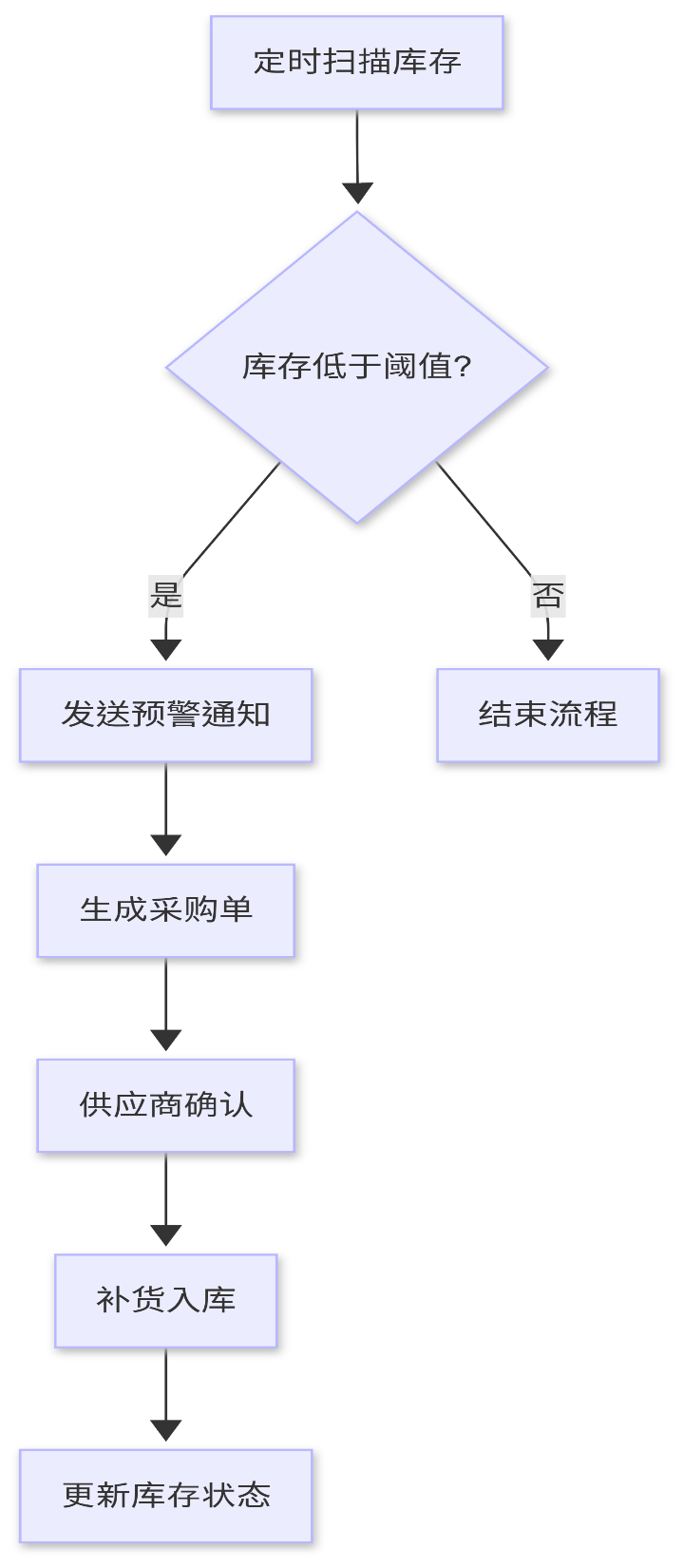


图 **4.12** 库存预警与补货流程图

### 用户反馈处理流程

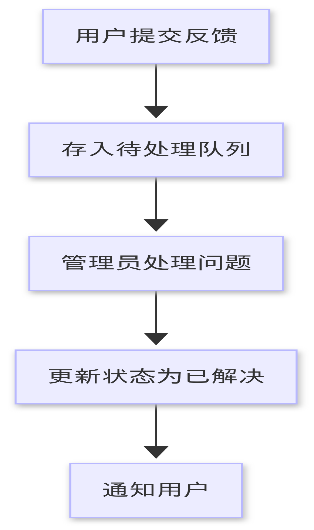
用户反馈处理的流程可参照图4.13来了解。具体而言，用户会在个人中心提交反馈内容，比如涉及商品质量方面的问题等。随后，系统会把该反馈记录保存到数据库当中，同时将其标记成“待处理”的状态。接着，管理员在登录之后，就能够查看反馈列表，进而把相关反馈分配到对应的部门去处理。等到处理工作全部完成，管理员便会更新反馈的状态，将其改为“已解决”，并且通知用户去查看最终的处理结果。

图 **4.13** 用户反馈处理流程

### 数据统计与报表生成流程

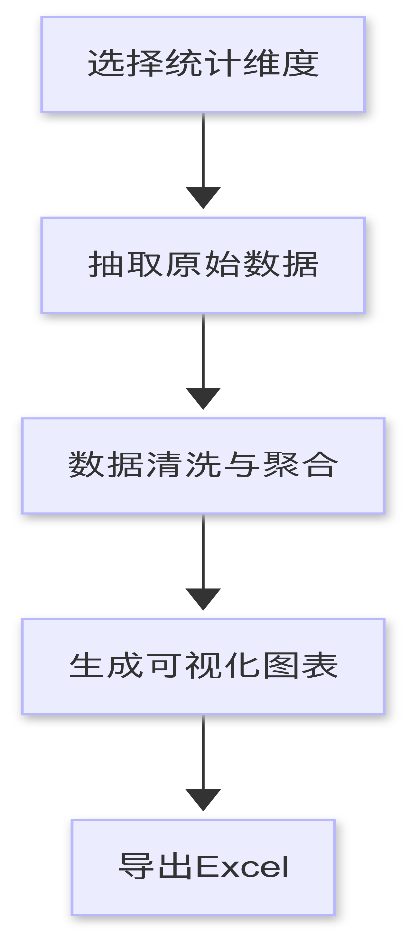
图4.14展示的是数据统计与报表生成的流程图。管理员会于后台来选择统计维度，这统计维度包含了时间、品类以及区域等方面。随后，系统会从MySQL数据库当中抽取原始数据，接着借助JsCooking对抽取的数据加以清洗与聚合处理，进而生成销售额趋势图或者品类占比饼图这样的图表。而生成的报表是能够支持导出为Excel格式的，如此便可供管理层在做决策之时作为参考依据。

图 **4.14** 数据统计与报表生成流程图

## 数据库设计

数据库的构建对软件的开发有很大的影响，而在数据库的设计中，更是不能离开表与表之间的逻辑关系的构建。数据库对于系统的开发是至关重要的，在用户视角里，它直接的起到一个数据交互的作用，因此，在进行数据库的设计的过程中，要结合实际的用户需要，来对数据库展开有针对性的设计。

### 实体E-R图

本系统采用MySQL作为数据库管理系统，数据库作为系统开发的核心组成部分，其设计质量直接影响着系统的整体性能和用户体验。在本系统的数据库设计过程中，我们充分考虑了数据存储效率、查询性能以及后期维护的便利性等因素。通过合理的表结构设计和关系构建，确保系统能够高效稳定地运行。在数据库选型方面，我们选择了MySQL关系型数据库管理系统。MySQL具有成熟稳定、性能优异、社区支持完善等特点，能够很好地满足系统对数据存储和管理的需求。同时，MySQL支持事务处理和复杂查询，与系统技术栈兼容性好，为系统开发提供了可靠的数据支撑。数据库设计遵循了规范化原则，通过三范式消除数据冗余，确保数据一致性。我们为关键表建立了适当的索引，显著提高了查询效率。通过外键约束维护表间关联，保证了数据的完整性。此外，我们还预留了必要的字段，为未来业务扩展做好准备。

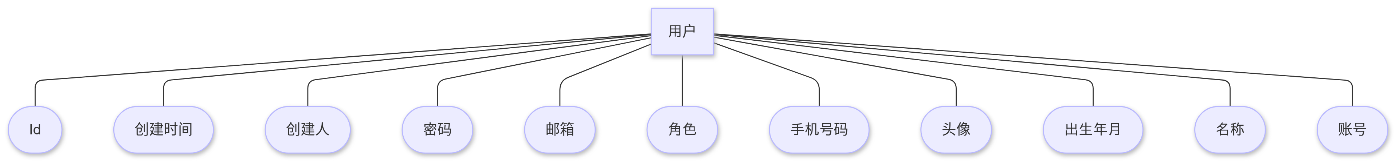
1.用户E-R图如图4.15所示：

图 **4.15** 用户E-R图

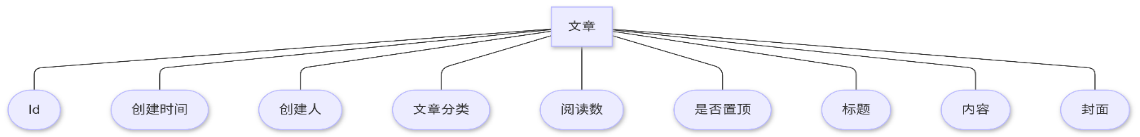
2.文章E-R图如图4.16所示：

图 **4.16**文章E-R图

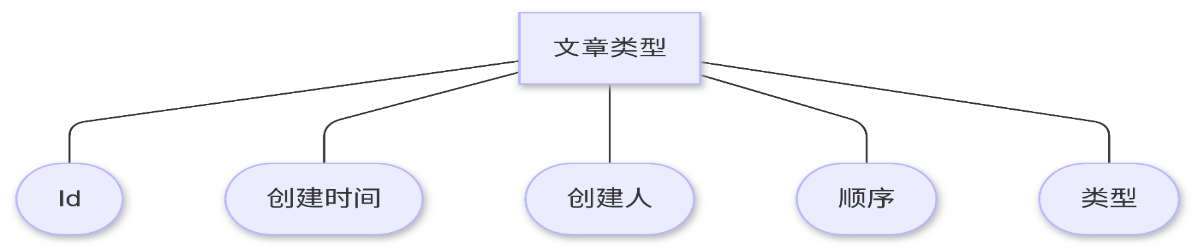
3.文章类型E-R图如图4.17所示：

图 **4.17**文章类型E-R图

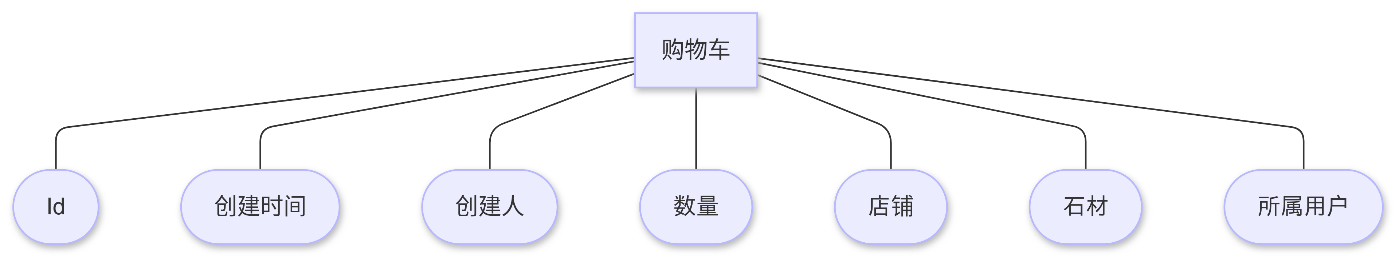
4.购物车E-R图如图4.18所示：

图 **4.18**购物车E-R图

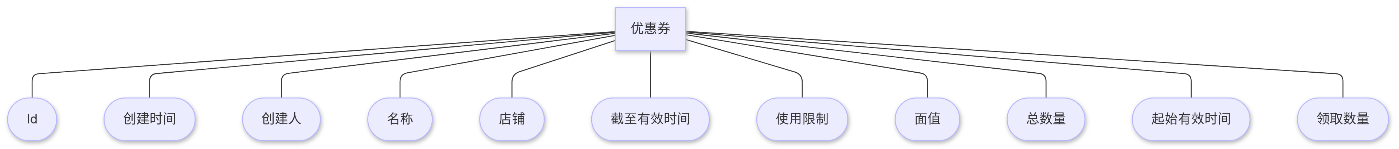
5.优惠券E-R图如图4.19所示：

图 **4.19**优惠券E-R图

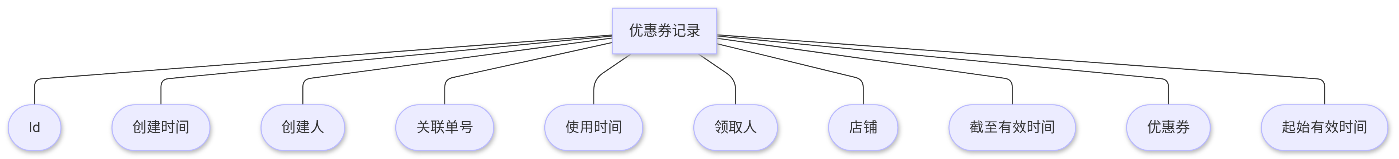
6.优惠券记录E-R图如图4.20所示：

图 **4.20**优惠券记录E-R图

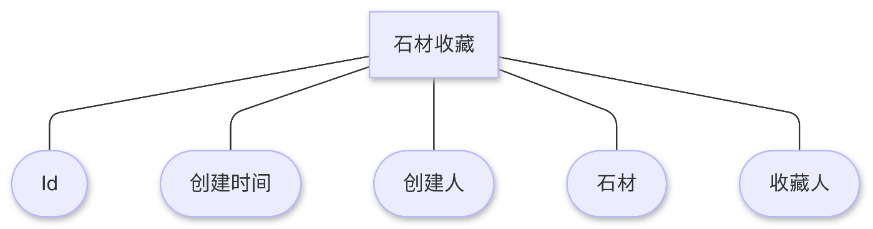
7.石材收藏E-R图如图4.21所示：

图 **4.21**石材收藏E-R图

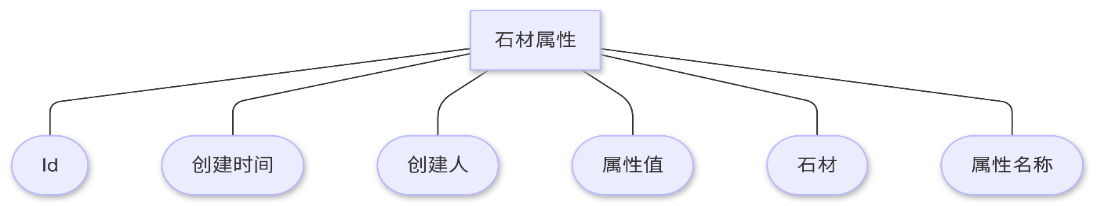
8.石材属性E-R图如图4.22所示：

图 **4.22**石材属性E-R图

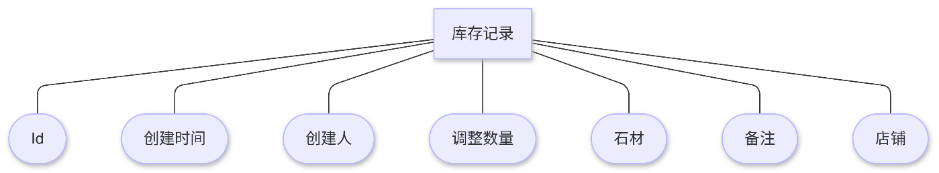
9.库存记录E-R图如图4.23所示：

图 **4.23**库存记录E-R图

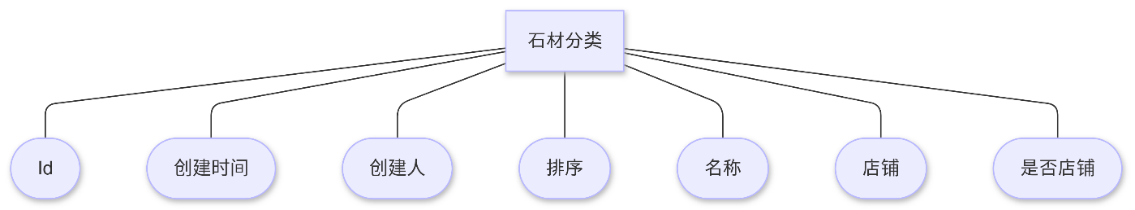
10.石材分类E-R图如图4.24所示：

图 **4.24**石材分类E-R图

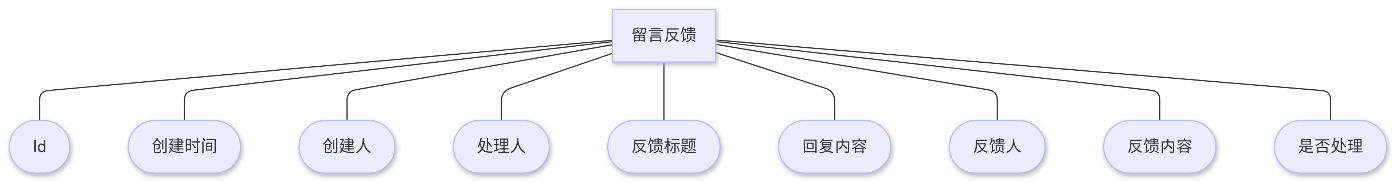
11.留言反馈E-R图如图4.25所示：

图 **4.25**留言反馈E-R图

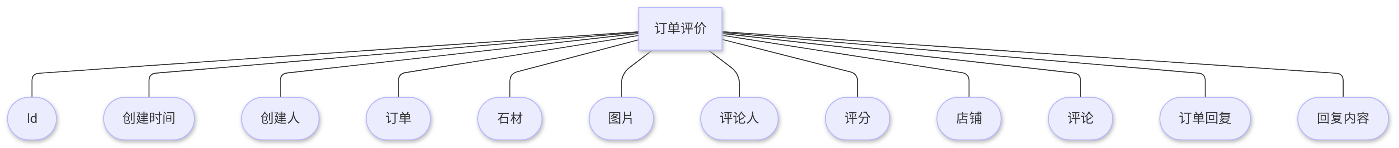
12.订单评价E-R图如图4.26所示：

图 **4.26**订单评价E-R图

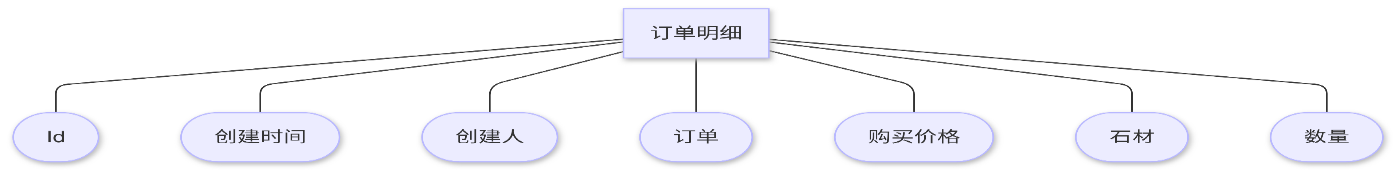
13.订单明细E-R图如图4.27所示：

图 **4.27**订单明细E-R图

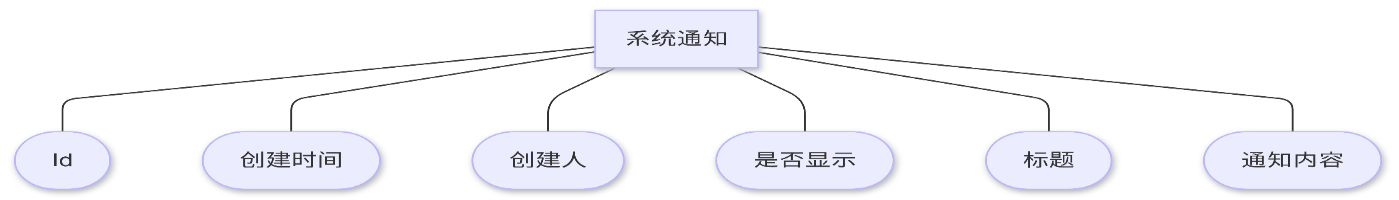
14.系统通知E-R图如图4.28所示：

图 **4.28**系统通知E-R图

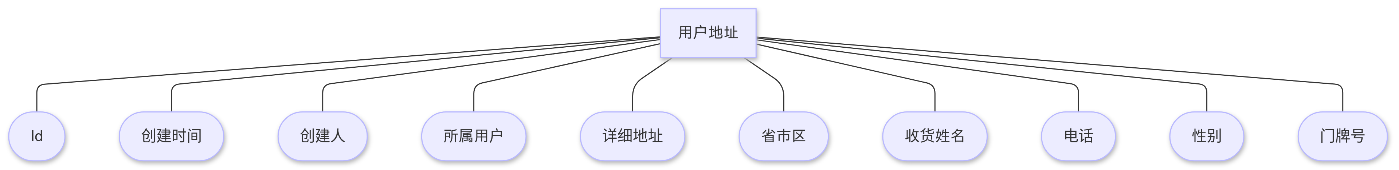
15.用户地址E-R图如图4.29所示：

图 **4.29**用户地址E-R图

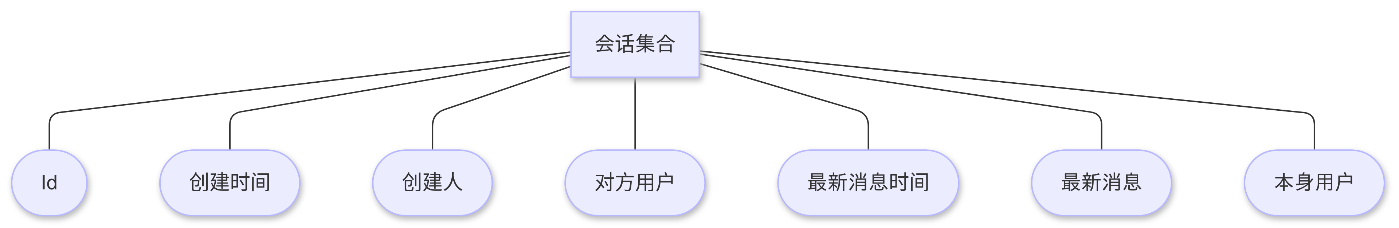
16.会话集合E-R图如图4.30所示：

图 **4.30**会话集合E-R图

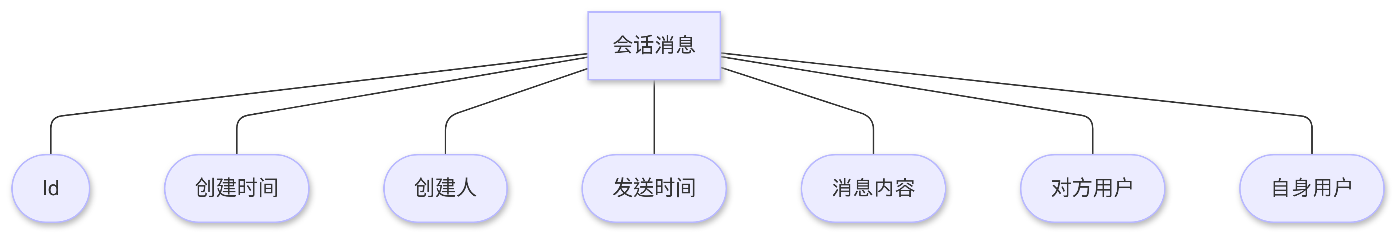
17.会话消息E-R图如图4.31所示：

图 **4.31**消息E-R图

就像在图4.32所呈现的那样，这里展示的是石材商城系统的E-R图呀。这一E-R图把石材商城系统的核心数据模型完整地给展现出来了呢，其中涵盖了多达22个主要的实体，并且还明确了它们彼此之间的关系哦。该系统是以用户（也就是appuser）作为核心来展开运作的呀，它和购物车（即buycard）、订单（也就是orderinfo）等这些处于核心位置的业务实体都有着紧密的关联呢。而且呀，借助店铺（也就是shop）以及石材（即good），系统能够很好地实现对商品的管理事宜哦。在营销活动方面呢，优惠券（coupon）系统可是发挥着重要的支持作用呀。而评价系统（ordercomment）与留言反馈（leavefeedback）这两个部分呢，则共同构成了用户互动的模块啦。至于库存管理这块呀，通过石材（good）、库存记录（goodstock）以及石材分类（goodtype）等方面的协同配合，从而达成了精细化的运营状态哦。

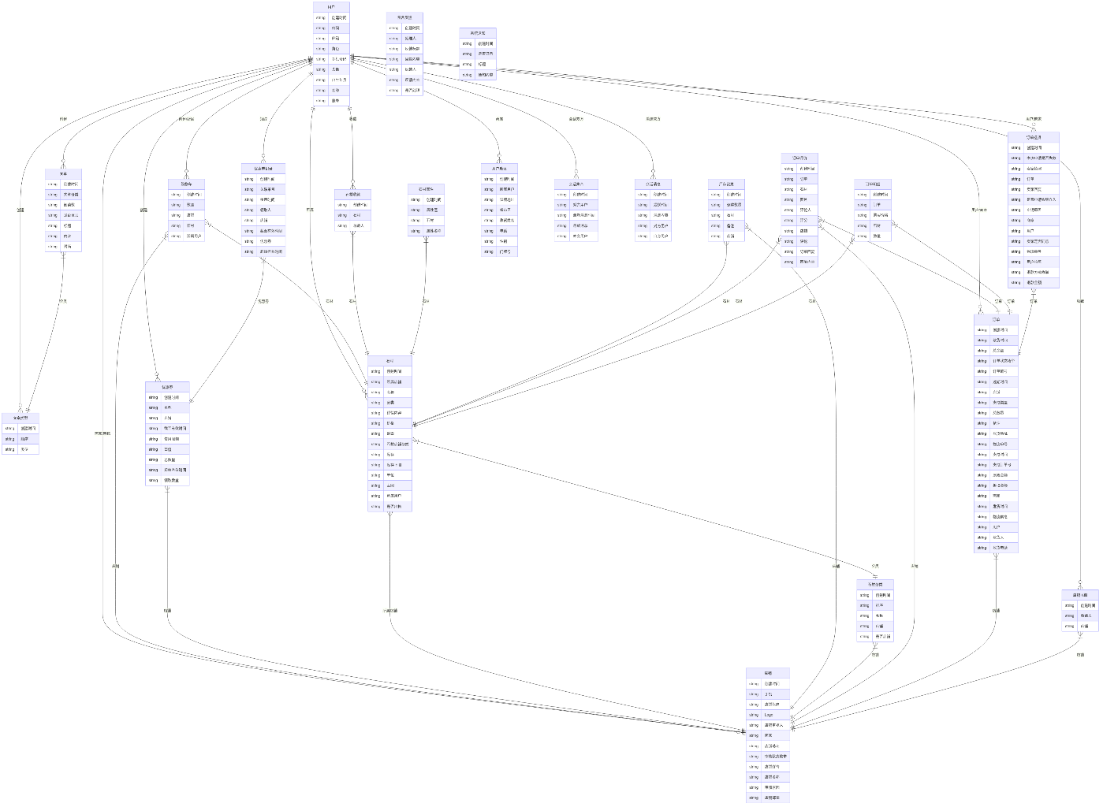


图 **4.32**石材商城系统E-R图

### 数据表设计

达成了从E-R图表至关系型数据库的转变。在关系型资料库当中，资料的关联是由资料的表所构建起来的。数据结构需依据详尽的分析来加以确定，依照系统需求输入与输出的关系明确表与表之间的关联，针对表之间的结构对特定的数据表予以调整。务必要以数据作为依托构建概念数据模型，对数据展开采集管控，并且针对表与表之间的关系实施分析，剖析出它们的逻辑数据模型。

在实际的应用程序当中，Mysql往往会被用作存储数据库。它能够较好地保护数据，同时也能妥善地对数据加以管理。甚至还可以借助一些第三方的图像界面工具来连接并管理这些数据，如此一来，便能清晰地知晓我们所涉及的表的逻辑关系，这对改进表之间的联系是颇为方便的。我们得明确表的字段和表格相互间的联系，这样就能较为迅速地构建起数据表。数据库设计得是否合理，会直接对系统产生影响，甚至关乎到后期的维护成本。要是数据库的逻辑关系设计得不合理，那么当系统更新迭代到一定版本且出现问题的时候，维护成本就会变得相当高。 数据库设计合理与否，关系到一个系统的性能与质量状况。倘若设计不合理，那么程序在执行读写操作时速度就会变得很慢，会延长数据处理的时间，致使系统原本应有的运行速度有所降低。单据和实体之间的关系要依照一对一、一对多、多对多这样的关系来确定，通常情况下是一对一的关系，此外还得确定主键和外键，主键与外键在整个数据库当中占据着极为重要的地位。

1.用户表（appuser），就像表4.1所展示的那样，它存储着系统全部用户的基本信息.

表 **4.1**用户表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| Id | int |  | 用户主键 | 是 |  |
| CreationTime | timestamp | | 创建时间 |  | NULL |
| CreatorId | int |  | 创建人 |  | NULL |
| Password | varchar | 20 | 密码 |  | NULL |
| Email | varchar | 40 | 邮箱 |  | NULL |
| RoleType | int |  | 角色 |  | NULL |
| PhoneNumber | varchar | 20 | 手机号码 |  | NULL |
| ImageUrls | varchar | 256 | 头像 |  | NULL |
| Birth | datetime |  | 出生年月 |  | NULL |
| Name | varchar | 20 | 名称 |  | NULL |
| UserName | varchar | 20 | 账号 |  | NULL |

2.文章表(article)，如同表4.2所展示的那般，其主要用于存储平台所发布的文章内容。该表的设计充分考虑了石材行业资讯的多样性和复杂性，包含文章标题、内容、分类、阅读量等核心字段。其中，标题字段（Title）限制为128个字符，确保简洁明了。

表 **4.2**文章表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| Id | int |  | 文章主键 | 是 |  |
| CreationTime | timestamp | | 创建时间 |  | NULL |
| CreatorId | int |  | 创建人 |  | NULL |
| ArticleTypeId | int |  | 文章分类 |  | NULL |
| ReadCount | int |  | 阅读数 |  | NULL |
| IsTop | tinyint | 1 | 是否置顶 |  | NULL |
| Title | varchar | 128 | 标题 |  | NULL |
| Content | longtext |  | 内容 |  | NULL |
| Cover | varchar | 128 | 封面 |  | NULL |

3.文章类型表(articletype)，如表4.3所示，定义文章的分类体系，用于文章的分类管理

表 **4.3**文章类型表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| Id | int |  | 文章类型主键 | 是 |  |
| CreationTime | timestamp | | 创建时间 |  | NULL |
| CreatorId | int |  | 创建人 |  | NULL |
| Sort | int |  | 顺序 |  | NULL |
| Name | varchar | 128 | 类型 |  | NULL |

4.购物车表(buycard)，如表4.4所示，记录用户添加到购物车的商品信息。

表 **4.4** 购物车表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 |  | 主键 | 默认值 |
| Id | int |  | 购物车主键 |  | 是 |  |
| CreationTime | timestamp | | 创建时间 |  |  | NULL |
| CreatorId | int |  | 创建人 |  |  | NULL |
| Qty | int |  | 数量 |  |  | NULL |
| ShopId | int |  | 店铺 |  |  | NULL |
| GoodId | int |  | 石材 |  |  | NULL |
| BelongUserId | int |  | 所属用户 |  |  | NULL |

5.优惠券表(coupon)，如表4.5所示。

表 **4.5**优惠券表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| Id | int |  | 优惠卷主键 | 是 |  |
| CreationTime | timestamp | | 创建时间 |  | NULL |
| CreatorId | int |  | 创建人 |  | NULL |
| Title | varchar | 64 | 名称 |  | NULL |
| ShopId | int |  | 店铺 |  | NULL |
| EndActiveTime | datetime |  | 截至时间 |  | NULL |
| UseMoney | varchar | 64 | 使用限制 |  | NULL |
| Money | double | 20,5 | 面值 |  | NULL |
| Qty | int |  | 总数量 |  | NULL |
| BeginActiveTime | datetime |  | 起始时间 |  | NULL |
| GetQty | int |  | 领取数量 |  | NULL |

6.优惠券记录表（couponrecord），就如同表4.6所展示。

表 **4.6**优惠券记录表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| Id | int |  | 优惠卷记录主键 | 是 |  |
| CreationTime | timestamp | | 创建时间 |  | NULL |
| CreatorId | int |  | 创建人 |  | NULL |
| RelativeNo | varchar | 64 | 关联单号 |  | NULL |
| UseTime | datetime |  | 使用时间 |  | NULL |
| GetUserId | int |  | 领取人 |  | NULL |
| ShopId | int |  | 店铺 |  | NULL |
| EndActiveTime | datetime |  | 截至有效时间 |  | NULL |
| CouponId | int |  | 优惠卷 |  | NULL |
| BeginActiveTime | datetime |  | 起始有效时间 |  | NULL |

7.石材表(good)，如表4.7所示。

表 **4.7**石材表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| Id | int |  | 石材主键 | 是 |  |
| CreationTime | timestamp | | 创建时间 |  | NULL |
| CreatorId | int |  | 创建人 |  | NULL |
| ShopId | int |  | 所属店铺 |  | NULL |
| Name | varchar | 128 | 名称 |  | NULL |
| GoodTypeId | int |  | 分类 |  | NULL |
| Content | longtext |  | 详情内容 |  | NULL |
| Price | double | 20,5 | 价格 |  | NULL |
| Cover | varchar | 128 | 封面 |  | NULL |
| ShopGoodTypeId | int |  | 石材店铺分类 |  | NULL |
| Stock | double | 20,5 | 库存 |  | NULL |
| ReplenishmentLimit | int |  | 库存下限 |  | NULL |
| Unit | varchar | 54 | 单位 |  | NULL |
| ImageUrls | varchar | 1024 | 主图 |  | NULL |

8.石材收藏表(goodcollect)，如表4.8所示，石材收藏表用于管理用户收藏行为的数据表。

表 **4.8**石材收藏表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| Id | int |  | 石材收藏主键 | 是 |  |
| CreationTime | timestamp | | 创建时间 |  | NULL |
| CreatorId | int |  | 创建人 |  | NULL |
| GoodId | int |  | 石材 |  | NULL |
| CollectUserId | int |  | 收藏人 |  | NULL |

9.石材属性表(goodprop)，如表4.9所示，存储石材商品的属性信息，如规格、颜色。

表 **4.9**石材属性表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| Id | int |  | 石材属性主键 | 是 |  |
| CreationTime | timestamp | | 创建时间 |  | NULL |
| CreatorId | int |  | 创建人 |  | NULL |
| Value | varchar | 50 | 属性值 |  | NULL |
| GoodId | int |  | 石材 |  | NULL |
| Name | varchar | 10 | 属性名称 |  | NULL |

10.库存记录表(goodstock)，如表4.10所示，记录石材商品的库存变动情况，用于库存管理。

表 **4.10**库存记录表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| Id | int |  | 库存记录主键 | 是 |  |
| CreationTime | timestamp | | 创建时间 |  | NULL |
| CreatorId | int |  | 创建人 |  | NULL |
| Qty | double | 20,5 | 调整数量 |  | NULL |
| GoodId | int |  | 石材 |  | NULL |
| Remark | varchar | 512 | 备注 |  | NULL |
| ShopId | int |  | 店铺 |  | NULL |

11.石材分类表(goodtype)，如表4.11所示，它不仅定义了系统预设的石材分类标准，还支持商家根据自身经营需求创建个性化的商品分类。该表通过ShopId字段关联店铺信息，使每个商家都可以维护独立的分类体系，同时通过IsShop字段区分系统分类和店铺自定义分类。

表 **4.11**石材分类表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| Id | int |  | 石材分类主键 | 是 |  |
| CreationTime | timestamp | | 创建时间 |  | NULL |
| CreatorId | int |  | 创建人 |  | NULL |
| Sort | int |  | 排序 |  | NULL |
| Name | varchar | 20 | 名称 |  | NULL |
| ShopId | int |  | 店铺 |  | NULL |
| IsShop | tinyint | 1 | 是否店铺 |  | NULL |

12.留言反馈表(leavefeedback)，如表4.12所示，存储用户的留言和反馈信息，支持回复功能

表 **4.12**留言反馈表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| Id | int |  | 留言反馈主键 | 是 |  |
| CreationTime | timestamp | | 创建时间 |  | NULL |
| CreatorId | int |  | 创建人 |  | NULL |
| DoWithUserId | int |  | 处理人 |  | NULL |
| Title | varchar | 128 | 反馈标题 |  | NULL |
| ReplayContent | longtext |  | 回复内容 |  | NULL |
| FeedbackUserId | int |  | 反馈人 |  | NULL |
| Content | longtext |  | 反馈内容 |  | NULL |
| IsDoWith | tinyint | 1 | 是否处理 |  | NULL |

13.订单评价表，也就是ordercomment，就如同表4.13所展示的那样，它在系统当中属于极为重要的数据表，主要是用来存储用户针对所购买商品给出的评价信息。该表会详尽地把用户对那些已经完成的订单里石材商品的综合评价情况都记录下来，这里面涵盖了文字形式的评价内容，还有星级评分，其范围是从1分到5分，另外也包含了用户上传的评价图片。每一条评价记录都和对应的订单ID（来自orderinfo表）、商品ID（来自good表）以及店铺ID（来自shop表）相互关联起来，如此便能保证评价数据是完整的，而且还可以实现追溯。评价数据可不单单是用于在商品详情页展示用户反馈的，它还为商家改良商品以及服务给出了依据。

表 **4.13**订单评价表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| Id | int |  | 订单评价主键 | 是 |  |
| CreationTime | timestamp | | 创建时间 |  | NULL |
| CreatorId | int |  | 创建人 |  | NULL |
| OrderId | int |  | 订单 |  | NULL |
| GoodId | int |  | 石材 |  | NULL |
| ImageUrls | varchar | 216 | 图片 |  | NULL |
| CommentUserId | int |  | 评论人 |  | NULL |
| CommentScore | double | 20,5 | 评分 |  | NULL |
| ShopId | int |  | 店铺 |  | NULL |
| Comment | longtext |  | 评论 |  | NULL |
| ReplyImageUrls | varchar | 1024 | 订单回复 |  | NULL |
| ReplyContent | varchar | 1024 | 回复内容 |  | NULL |

14.订单明细表(orderdet)，如表4.14所示，存储订单中的商品明细，包括购买数量和价格

表 **4.14**订单明细表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| Id | int |  | 订单明细主键 | 是 |  |
| CreationTime | timestamp | | 创建时间 |  | NULL |
| CreatorId | int |  | 创建人 |  | NULL |
| OrderId | int |  | 订单 |  | NULL |
| Price | double | 20,5 | 购买价格 |  | NULL |
| GoodId | int |  | 石材 |  | NULL |
| Qty | int |  | 数量 |  | NULL |

15.订单表（orderinfo），就如同表4.15所展示的那般，单表（orderinfo）在系统当中属于存储交易核心数据的极为重要的表，它把从下单一直到完成的整个交易流程都详细记录了下来。在该表当中，是凭借OrderNo字段来存储那独一无二的订单编号的，而通过OrderStatus字段则能够对订单状态（像待支付、待发货、待收货、已完成等等这些情况）加以跟踪。

表 **4.15**订单表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| Id | int |  | 订单主键 | 是 |  |
| CreationTime | timestamp | | 创建时间 |  | NULL |
| CreatorId | int |  | 创建人 |  | NULL |
| GetOrderTime | datetime |  | 收货时间 |  | NULL |
| TotalMoney | double(20,5) | | 总金额 |  | NULL |
| OrderStatus | int |  | 订单状态枚举 |  | NULL |
| OrderNo | varchar | 64 | 订单编号 |  | NULL |
| ExpireTime | datetime |  | 过期时间 |  | NULL |
| ShopId | int |  | 店铺 |  | NULL |
| PayType | varchar | 10 | 支付类型 |  | NULL |
| TotalQty | int |  | 总数量 |  | NULL |
| Remark | varchar | 256 | 备注 |  | NULL |
| ReceiveAddress | varchar | 128 | 收货地址 |  | NULL |
| LogisticsNo | varchar | 64 | 物流单号 |  | NULL |
| PayTime | datetime |  | 支付时间 |  | NULL |
| PayOrderNo | varchar | 64 | 支付订单号 |  | NULL |
| OrginMoney | double(20,5) | | 原始金额 |  | NULL |
| DiscountMoney | double(20,5) | | 折扣金额 |  | NULL |
| ShopUserId | int |  | 商家 |  | NULL |
| DeliveryTime | datetime |  | 发货时间 |  | NULL |
| ExpressInfo | longtext |  | 物流信息 |  | NULL |
| UserId | int |  | 用户 |  | NULL |
| ReceiveName | varchar | 15 | 收货人 |  | NULL |
| ReceivePhone | varchar | 20 | 收货电话 |  | NULL |

16.订单退货表（orderreturn），就像表4.16所展示的那样，订单退货表（orderreturn）在系统当中属于极为重要的数据表，其用途在于对退货流程加以管理，它详尽地把用户所发起的退货申请的整个流程信息都记录了下来。该表借助外键和订单表（orderinfo）以及用户表（appuser）建立起了关联关系，以此来保证每一条退货记录都能和有效的订单以及用户相对应。

表 **4.16**订单退货表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| Id | int |  | 订单退货主键 | 是 |  |
| CreationTime | timestamp | | 创建时间 |  | NULL |
| CreatorId | int |  | 创建人 |  | NULL |
| IsDiscard | tinyint(1) |  | 本次申请是否失效 | | NULL |
| SellerImageUrls | varchar | 216 | 卖家拍照 |  | NULL |
| OrderId | int |  | 订单 |  | NULL |
| SellerReply | longtext |  | 卖家回复 |  | NULL |
| IsApplyCustom | tinyint(1) |  | 是否申请客服介入 | | NULL |
| ApplyReason | longtext |  | 申请原因 |  | NULL |
| ShopUserId | int |  | 商家 |  | NULL |
| UserId | int |  | 用户 |  | NULL |
| IsAggren | tinyint(1) |  | 卖家是否同意 |  | NULL |
| LogisticsNo | varchar | 64 | 物流单号 |  | NULL |
| UserImageUrls | varchar | 216 | 用户拍照 |  | NULL |
| OrderReturnType | int |  | 退款方式枚举 |  | NULL |
| ReturnMoney | double(18,6) | | 退款金额 |  | NULL |

17.店铺表（shop），就如同表4.17所展示的那样，它可是系统当中专门用来存储以及对店铺信息加以管理的关键数据表呢。这个表把每个店铺的各类基本信息都详尽地记录了下来，像店铺名称呀、联系方式呀、地址之类的重要数据都涵盖其中。表里面的LogoCover字段是用来存放店铺的标志图片的。

表 **4.17** 店铺表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| Id | int |  | 店铺主键 | 是 |  |
| CreationTime | timestamp | | 创建时间 |  | NULL |
| CreatorId | int |  | 创建人 |  | NULL |
| Phone | varchar | 20 | 手机 |  | NULL |
| Ensure | varchar | 48 | 店铺保障 |  | NULL |
| LogoCover | varchar | 128 | Logo |  | NULL |
| AuditUserId | int |  | 店铺审核人 |  | NULL |
| UserId | int |  | 商家 |  | NULL |
| Address | varchar | 48 | 店铺地址 |  | NULL |
| AuditStatus | int |  | 审核状态枚举 |  | NULL |
| Content | longtext |  | 店铺详情 |  | NULL |
| Name | varchar | 20 | 店铺名称 |  | NULL |
| AuditReason | longtext |  | 审核原因 |  | NULL |
| Email | varchar | 30 | 店铺邮箱 |  | NULL |

18.店铺收藏表(shopcollect)，如表4.18所示，记录用户收藏的店铺，方便快速访问。该表主要用于记录用户收藏的店铺信息，便于用户快速访问常用店铺。表结构包含自增主键Id、记录创建时间CreationTime（默认当前时间戳）、创建人CreatorId（外键关联用户表）、收藏人CollectUserId（外键关联用户表）以及店铺ShopId（外键关联店铺表）。通过CollectUserId和ShopId建立多对多关系索引，系统可定期分析用户收藏行为，为商家提供精准营销数据支持。该表设计符合数据库第三范式，所有外键字段均设置级联更新约束。

表 **4.18**店铺收藏表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| Id | int |  | 店铺收藏主键 | 是 |  |
| CreationTime | timestamp | | 创建时间 |  | NULL |
| CreatorId | int |  | 创建人 |  | NULL |
| CollectUserId | int |  | 收藏人 |  | NULL |
| ShopId | int |  | 店铺 |  | NULL |

19.系统通知表(sysnotice)，如表4.19所示，存储系统发布的通知公告，支持显示控制。该表用于管理系统发布的通知公告信息，主要包含自增主键Id、创建时间CreationTime（默认当前时间戳）、创建人CreatorId（外键关联用户表）、显示状态IsPutaway（0/1布尔值）、标题Title（限制128字符）和详细内容Content（支持长文本存储）。通过IsPutaway字段控制前端展示状态，Content字段采用longtext类型确保支持富文本格式。该表设计支持按创建时间降序查询，并为Title字段建立全文索引以提高搜索效率，同时通过CreatorId关联实现通知溯源功能。

表 **4.19**系统通知表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| Id | int |  | 系统通知主键 | 是 |  |
| CreationTime | timestamp | | 创建时间 |  | NULL |
| CreatorId | int |  | 创建人 |  | NULL |
| IsPutaway | tinyint(1) |  | 是否显示 |  | NULL |
| Title | varchar | 128 | 标题 |  | NULL |
| Content | longtext |  | 通知内容 |  | NULL |

20.用户地址表(useraddress)，如表4.20所示，存储用户的收货地址信息，支持省市区选择。

表 **4.20**用户地址表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| Id | int |  | 收货地址主键 | 是 |  |
| CreationTime | timestamp | | 创建时间 |  | NULL |
| CreatorId | int |  | 创建人 |  | NULL |
| BelongUserId | int |  | 所属用户 |  | NULL |
| Address | varchar | 80 | 详细地址 |  | NULL |
| ProviceCityArea | varchar | 30 | 省市区 |  | NULL |
| Name | varchar | 8 | 收货姓名 |  | NULL |
| Phone | varchar | 20 | 电话 |  | NULL |
| Sex | varchar | 6 | 性别 |  | NULL |
| HouseNumber | varchar | 20 | 门牌号 |  | NULL |

21.会话集合表(wechatcollection)，如表4.21所示，记录用户之间的会话列表，包括最新消息

表 **4.21**会话集合表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| Id | int |  | 会话集合主键 | 是 |  |
| CreationTime | timestamp | | 创建时间 |  | NULL |
| CreatorId | int |  | 创建人 |  | NULL |
| OtherUserId | int |  | 对方用户 |  | NULL |
| LastTime | datetime |  | 最新消息时间 |  | NULL |
| LastMessage | varchar | 128 | 最新消息 |  | NULL |
| SelfUserId | int |  | 本身用户 |  | NULL |

22.会话消息表(wechatmessage)，如表4.22所示，存储用户之间的聊天消息内容

表 **4.22**会话消息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| Id | int |  | 会话消息主键 | 是 |  |
| CreationTime | timestamp | | 创建时间 |  | NULL |
| CreatorId | int |  | 创建人 |  | NULL |
| SendTime | datetime |  | 发送时间 |  | NULL |
| Content | longtext |  | 消息内容 |  | NULL |
| OtherUserId | int |  | 对方用户 |  | NULL |
| SelfUserId | int |  | 自身用户 |  | NULL |

## 本章小结

这一章重点围绕系统设计来开展相关工作，进行流程设计，随后再到数据库构建，就这样一步步地完成了天然石材交易管理系统的核心架构设计工作。在针对系统功能展开分析的这个部分呢，把用户端以及管理端这两大模块的功能划分给明确下来了。用户端这边实现了诸如石材浏览呀、购物车管理、订单跟踪之类的一些基础功能，而管理端呢，则涵盖了库存管理、销售统计、权限控制等相对来说更为高级一些的功能，如此便形成了一个完整的业务闭环情况。通过对各个功能模块去做细化设计方面的工作，就能够确保该系统可以满足在石材交易整个流程当中的管理方面的各种需求。

# 系统实现

## 用户注册与登录

### 注册功能

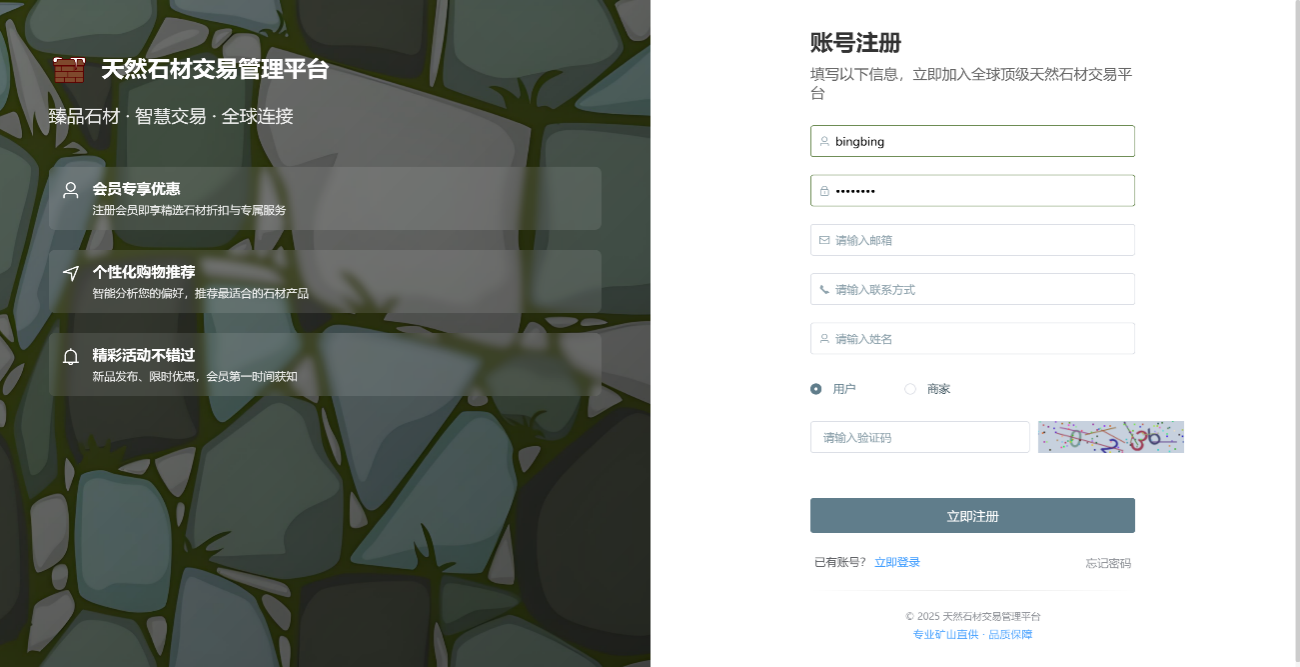
如图5.1，注册页面为用户提供了创建新账户的功能界面。注册页面实际上是为用户打造出了一个能够用来创建新账户的功能性界面。在这个页面之上，涵盖着多个需要用户进行输入操作的字段。用户得依照先后顺序，先是填写好自己想要设定的账号名称，接着去精心设置一个专属的密码，随后还需要提供像是有效的电子邮箱以及手机号码这类能够用于联系的方式，并且还要把自己的真实姓名也一并输入进去，如此才算完成了基本信息的录入工作。在系统这边呢，特意设计出了一个身份选择的模块，用户完全可以依据自身的实际需求，从“普通用户”以及“商家”这两种不同的身份类型当中去做出选择来完成注册。为了能够充分保障整个注册过程具备应有的安全性，在该页面还专门设置了一个图片验证码的验证环节。

图 **5.1**注册页面

### 登录功能

如图5.2，登录界面是用户访问系统的主要入口，采用账号密码验证机制。登录界面实际上是用户得以访问系统的一个主要入口所在，这里采用的是账号密码验证这样的一种机制。用户在一开始的时候呢，得在专门指定的位置去输入已经完成注册的账号以及与之相对应的密码，在此之后呢，还要通过下拉菜单来挑选出相应的身份类型，这里面包含了“管理员”、“普通用户”以及“商家”这三种不同的选项。而系统会依据所选择的身份类型，自动地去加载与之对应的权限模块。为了能够让安全性有所提升，在登录的这个过程当中，同样也设置了图片验证码验证这样的一个环节，这就要求用户必须要准确无误地去识别并且输入验证码相关的信息。

登录功能关键代码:

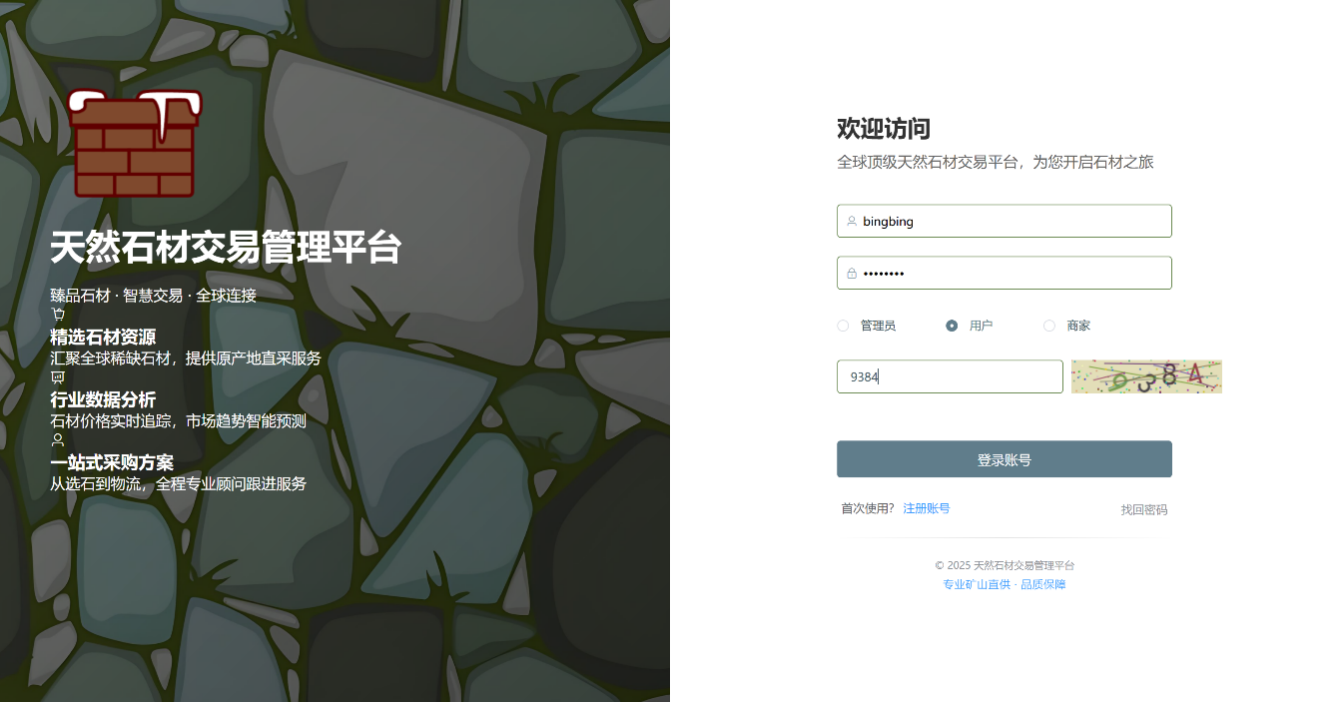
1. **public** String SignIn(AppUserDto input) {
2. LambdaQueryWrapper<AppUser> queryWrapper = Wrappers.<AppUser>lambdaQuery()
3. .eq(Extension.isNotNullOrEmpty(input.getUserName()), AppUser::getUserName, input.getUserName())
4. .eq(Extension.isNotNullOrEmpty(input.getPassword()), AppUser::getPassword, input.getPassword())
5. .eq(input.getRoleType() != null, AppUser::getRoleType, input.getRoleType());
7. List<AppUser> items = AppUserMpper.selectList(queryWrapper);
8. **if** (items.stream().count() == 0) {
9. **throw** **new** CustomException("请检查登录的账号或者密码,角色是否都正确!");
10. }
11. Map<String, String> map = **new** HashMap<>();
12. map.put(SysConst.UserIdClaim, items.get(0).getId().toString());
13. map.put(SysConst.RoleTypeClaim, items.get(0).getRoleType().toString());
14. String token = JWTUtils.getToken(map);
15. **return** token;
16.     }

图 **5.2**登录界面

### 找回密码功能

如图5.3，密码找回功能为忘记登录密码的用户提供了便捷的解决方案。密码找回功能给那些忘记登录密码的用户带来了颇为便捷的解决办法。当用户没办法顺利登录的时候，能够在该功能页面输入当初注册时所使用的账号相关信息，系统是支持凭借绑定的电子邮箱或者手机号码这两种不同的途径来开展身份验证工作的。在用户把账号信息提交上去之后，系统便会自动地发送包含有验证码的短信或者邮件，而用户则需要在指定的位置准确输入所收到的验证码，以此来完成身份的核验工作

找回密码功能关键代码:

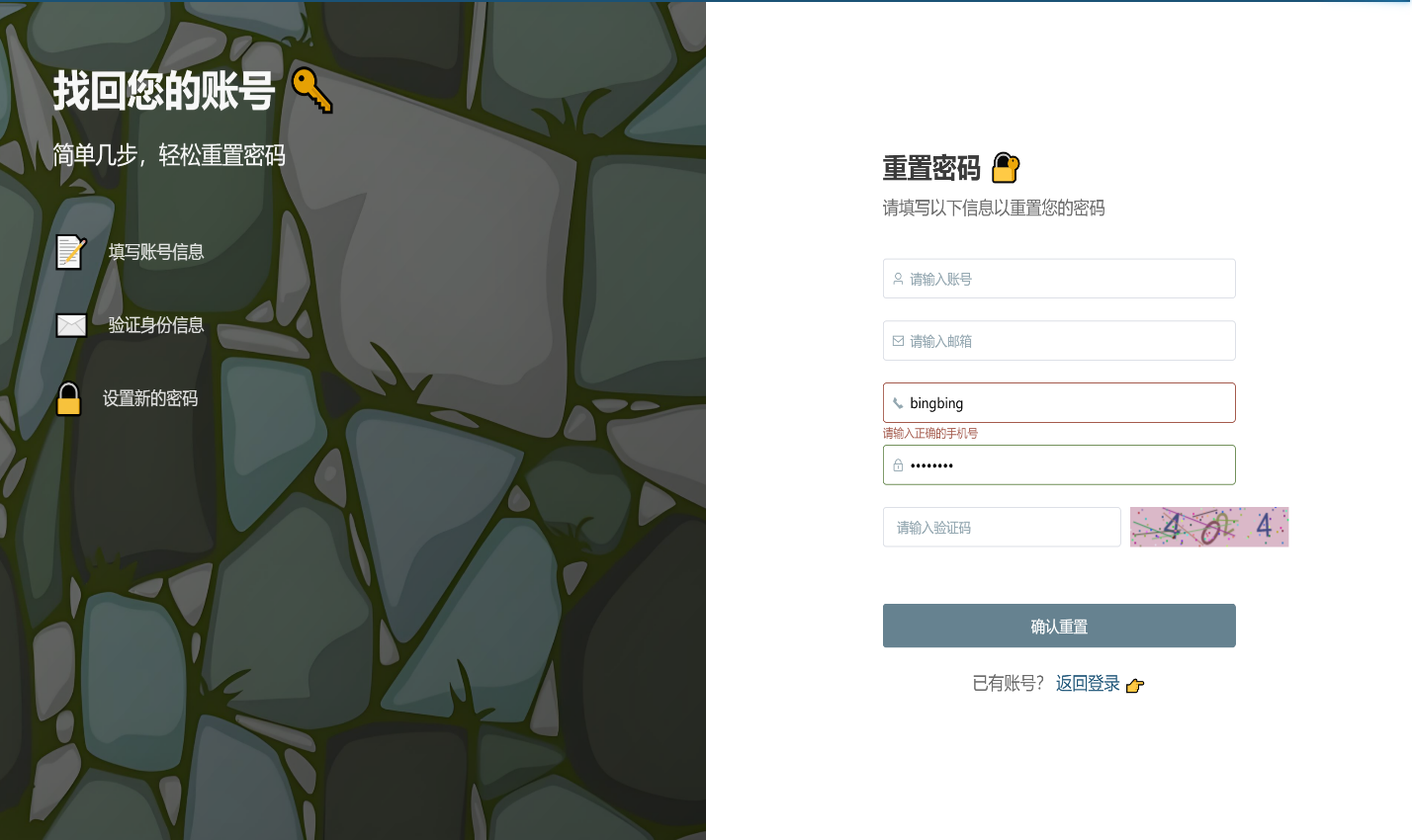
1. @Override
2. **public** **void** ForgetPassword(AppUserDto input) {
4. // 检查用户名是否存在
5. AppUser appUser = AppUserMpper.selectList(Wrappers.<AppUser>lambdaQuery()
6. .eq(Extension.isNotNullOrEmpty(input.getUserName()), AppUser::getUserName, input.getUserName()))
7. .stream().findFirst().orElse(null);
9. **if** (appUser == null) {
10. **throw** **new** CustomException("该用户名不存在!");
11. }
12. **if** (!appUser.getPhoneNumber().equals(input.getPhoneNumber())) {
13. **throw** **new** CustomException("请输入你绑定的手机号!");
14. }
15. **if** (!appUser.getEmail().equals(input.getEmail())) {
16. **throw** **new** CustomException("请输入你绑定的邮箱!");
17. }
18. appUser.setPassword(input.getPassword());
19. saveOrUpdate(appUser);
20.     }

图 **5.3**找回密码页面

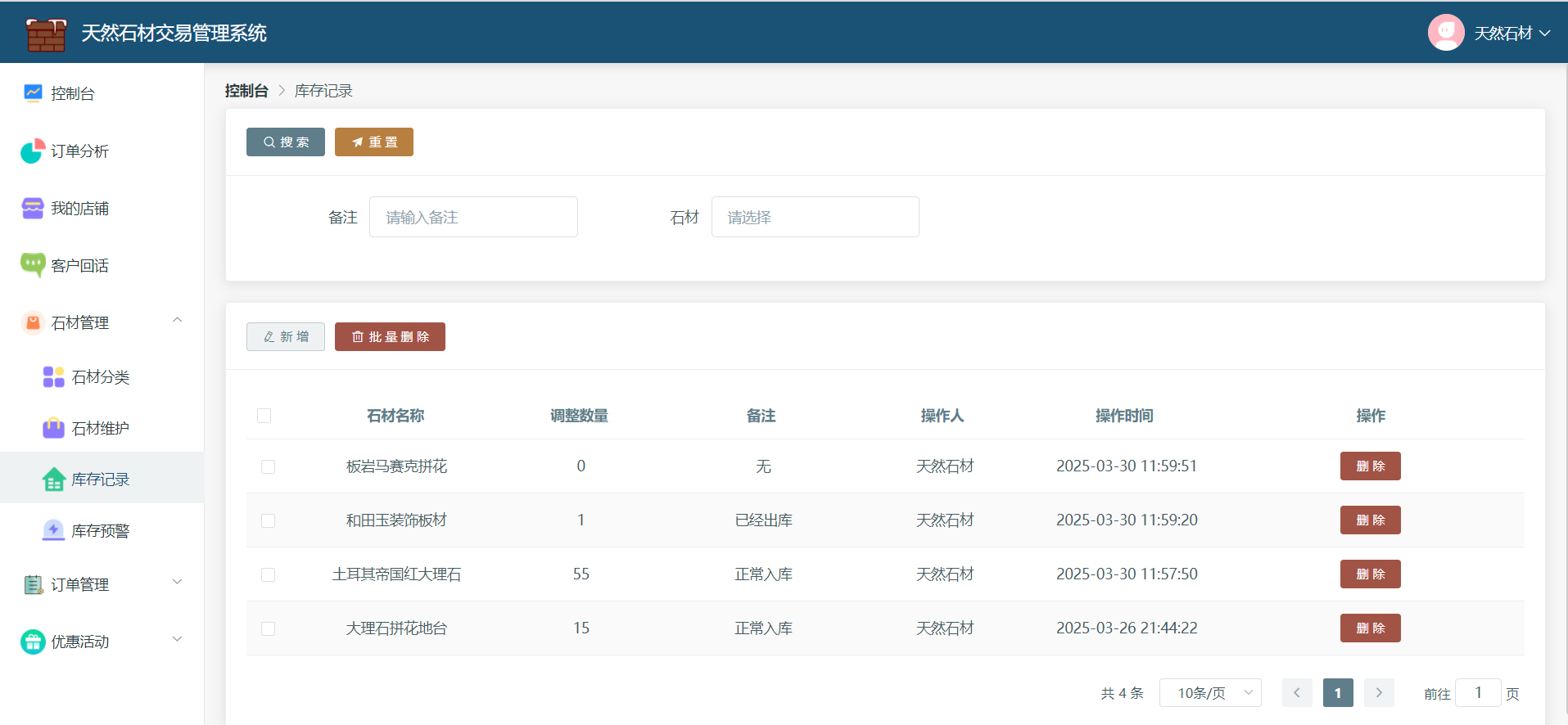
## 石材商家端功能实现

### 库存记录页面

如图5.4，库存记录页面为石材商家提供精细化的库存管理功能。系统详细记录每批石材的入库时间、数量变动和操作人员等信息。商家可手动添加特殊库存记录，如定制产品的临时入库，也可删除错误记录。页面支持按时间、产品名称等多条件筛选查询，帮助商家快速定位特定记录。库存数据以表格形式清晰呈现，与产品维护页面数据实时同步，确保库存信息的准确性。该模块特别适合管理多规格、多批次的大型石材产品库存。

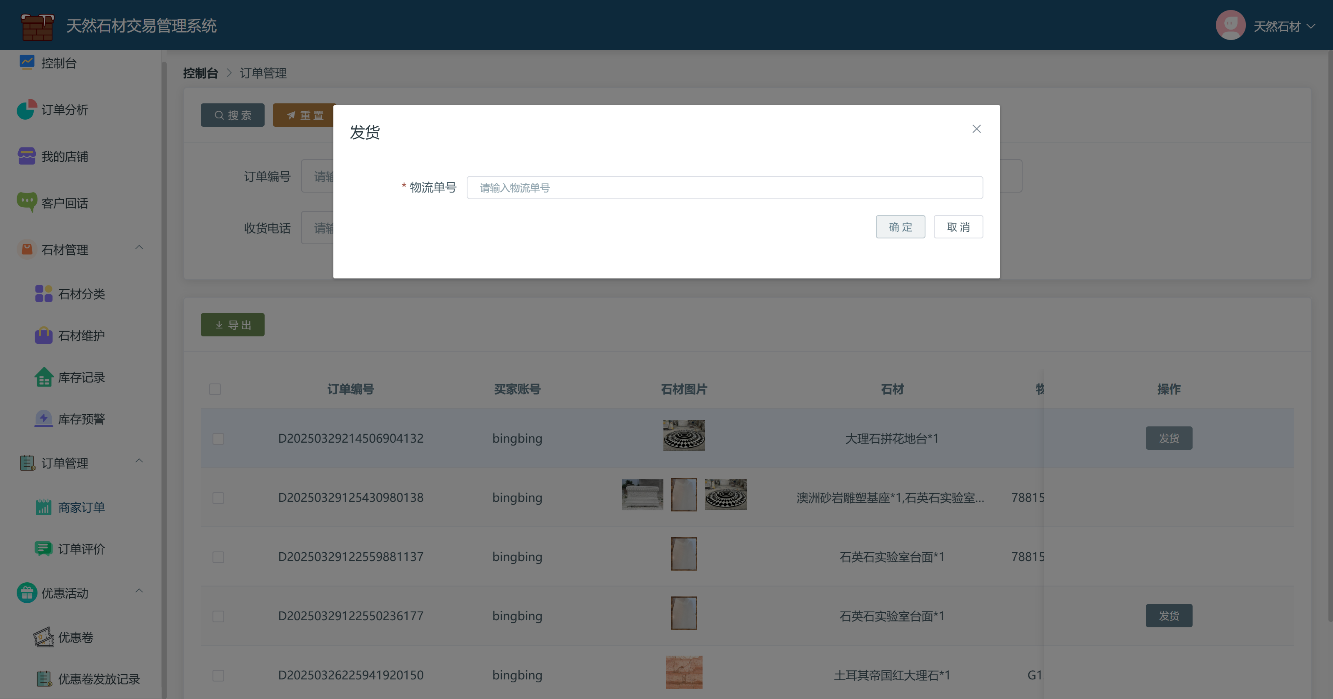
库存记录关键代码:

1. @SneakyThrows
2. @Override
3. **public** PagedResult<GoodStockDto> List(GoodStockPagedInput input) {
4. //构建where条件+排序
5. LambdaQueryWrapper<GoodStock> queryWrapper = BuilderQuery(input);
7. //按创建时间从大到小排序 最新的显示在最前面
8. queryWrapper = queryWrapper.orderByDesc(GoodStock::getCreationTime);
9. //构建一个分页查询的model
10. Page<GoodStock> page = **new** Page<>(input.getPage(), input.getLimit());
11. //从数据库进行分页查询获取库存记录数据
12. IPage<GoodStock> pageRecords = GoodStockMapper.selectPage(page, queryWrapper);
13. //获取所有满足条件的数据行数
14. Long totalCount = GoodStockMapper.selectCount(queryWrapper);
15. //把GoodStock实体转换成GoodStock传输模型
16. List<GoodStockDto> items = Extension.copyBeanList(pageRecords.getRecords(), GoodStockDto.**class**);
17. //计算表的数据
18. items = DispatchItem(items);
19. //返回一个分页结构给前端
20. **return** PagedResult.GetInstance(items, totalCount);
22. }

图 **5.4**库存记录页面

### 订单发货页面

如图5.5，订单发货页面实际上是石材商家用来处理客户订单的极为关键的核心工作界面。在这个页面上，商家能够去查看订单的详细信息，这里面涵盖了诸如石材规格呀、数量多少呀、具体的加工要求以及收货地址等等方面的内容。发货这一操作相对来讲是比较简便的，仅仅只需要商家输入物流单号，然后再从中选择合适的运输方式，如此便可以顺利完成整个发货流程了。

图 **5.5**订单发货页面

### 优惠券管理页面

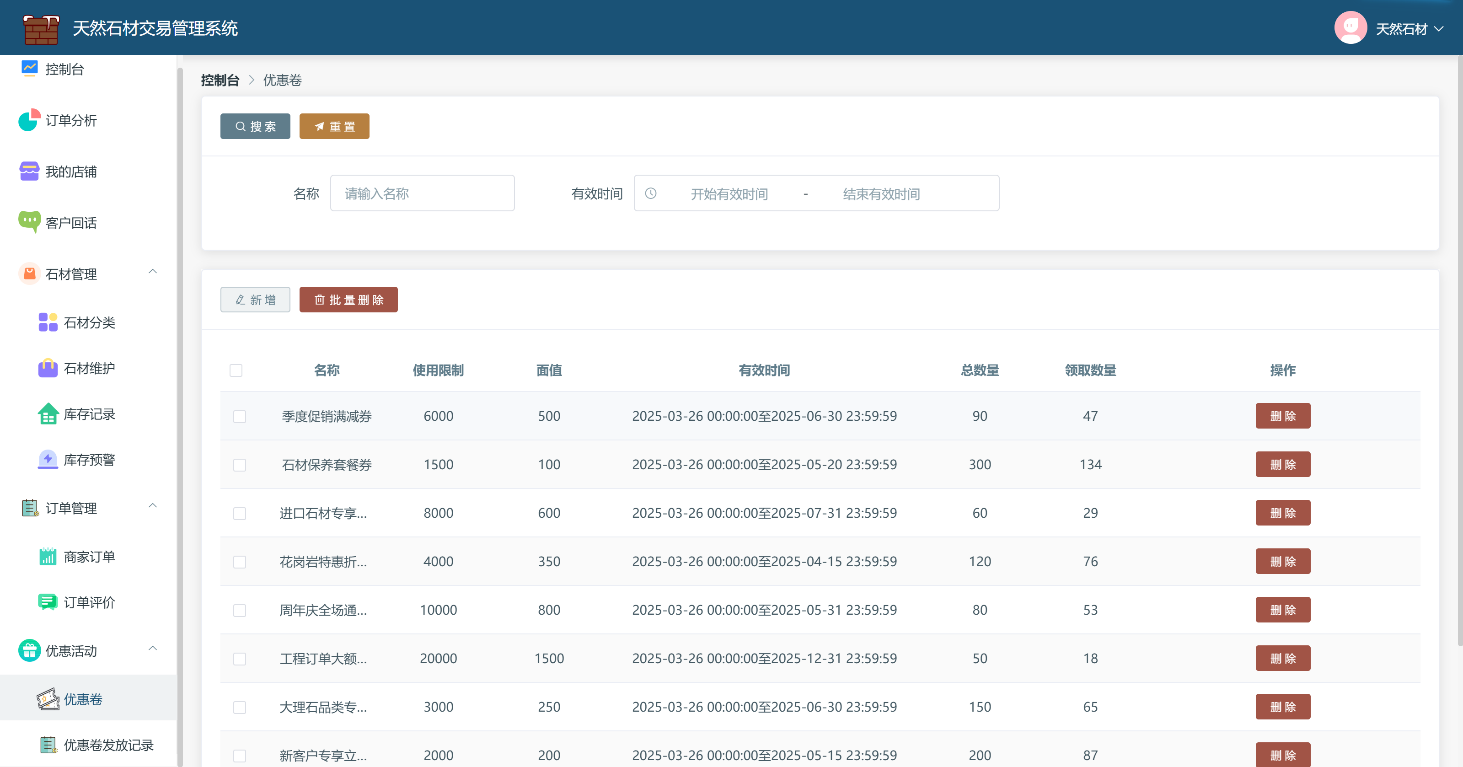
如图5.6，优惠券管理页面能够给予石材商家开展促销活动的有力支持。商家完全有能力去创建形形色色的优惠券，并且可以对折扣力度、使用条件以及有效期等诸多参数加以设置。鉴于石材行业自身所具有的特点，还能够设置那种专门针对特定产品的优惠或者满减优惠形式。

图 **5.6**优惠券管理页面

## 石材管理员功能实现

### 用户管理功能

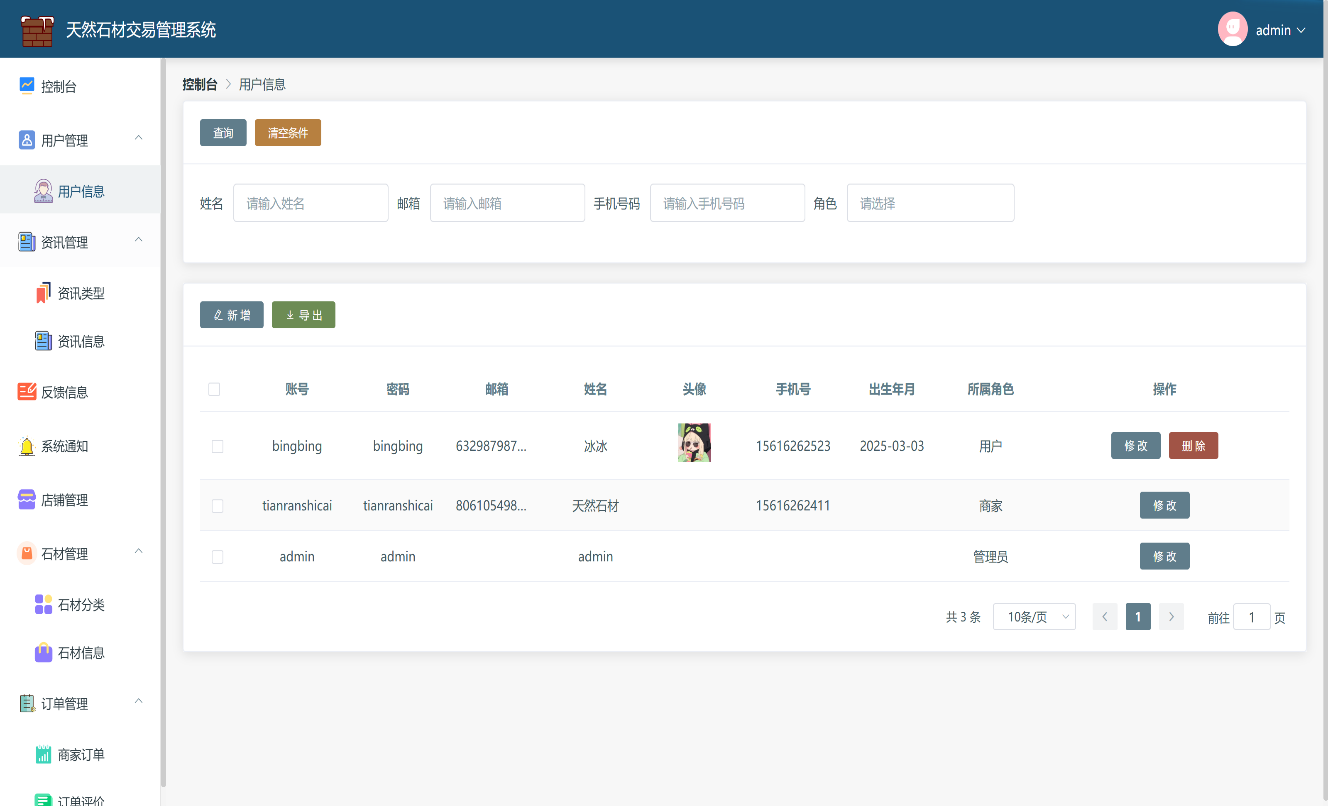
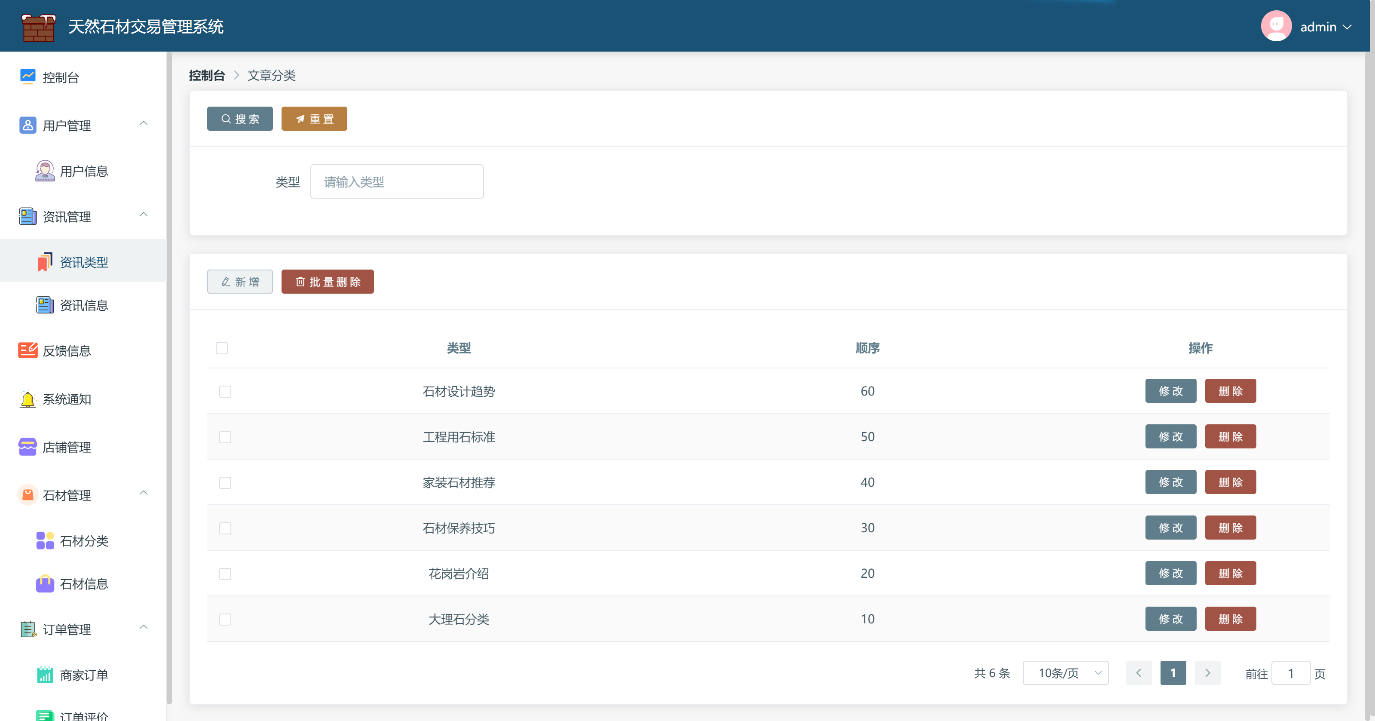
如图5.7，用户管理页面给管理员赋予了较为完备的用户信息管理方面的功能。在这个页面之上，管理员能够去查看系统当中所有已经完成注册的用户的基本信息内容，像账号呀、姓名啊以及联系方式等等这些信息都是可以查看到的。此页面还支持多种不同的操作途径，管理员针对用户信息能够去完成诸如修改、删除这类操作，并且还能够批量地选取多个用户来开展统一化的处理工作。对于用户数据而言，其是支持依照一定条件来进行筛选以及搜索操作的，如此一来便能够较为便捷地快速定位到特定的用户。

图 **5.7**用户管理页面

### 资讯管理功能

如图5.8，咨询类型管理页面在整个系统的内容管理当中占据着十分重要的位置，是其不可或缺的重要组成部分。系统的管理员能够在这个页面当中去创建以及对其进行维护一系列和石材行业紧密相关的咨询分类体系，比如说像石材种类方面的咨询呀，还有施工工艺相关的咨询等等这些不同类型的咨询分类。该页面不但对咨询类型的添加操作予以支持，而且对于修改以及删除这样的操作同样也是支持的，除此之外呢，还特意提供了批量处理这样的功能，通过这个功能就能够在很大程度上提高相关的工作效率。

图 **5.8**咨询类型管理页面

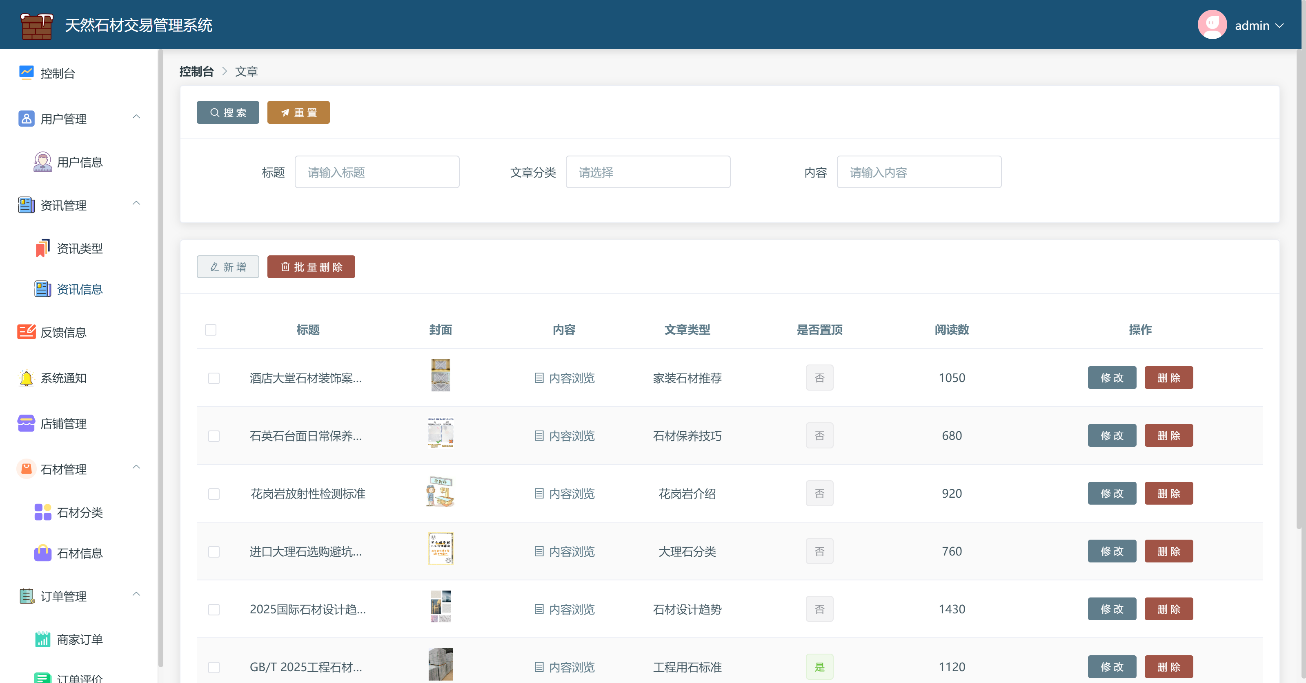
如图5.9，系统中的咨询信息管理页面能够对各类与石材相关的咨询内容展开集中化的管理操作。在这个页面之上，管理员拥有查看用户所提交详细咨询信息的权限，而这些信息涵盖了诸如咨询的具体类型、实际内容以及回复所处的状态等等多个方面。此页面还配备了颇为实用的搜索功能，它能够支持按照时间、类型以及关键词等多种条件以组合的方式来进行查询操作。管理员既可以直接在该页面针对用户的咨询给予回复，又能够针对那些存在不当之处的内容实施删除或者修改的操作。

图 **5.9**咨询信息管理页面

### 石材信息管理功能

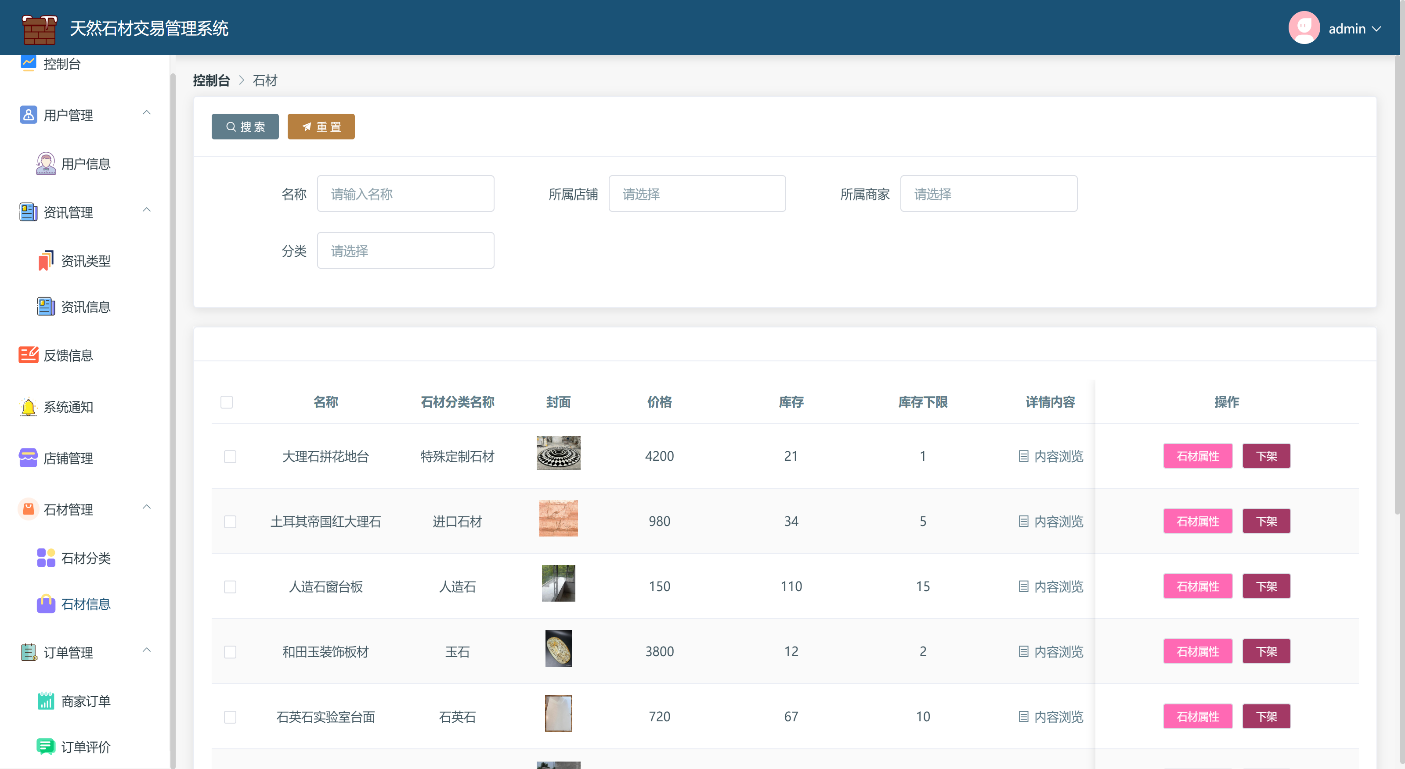
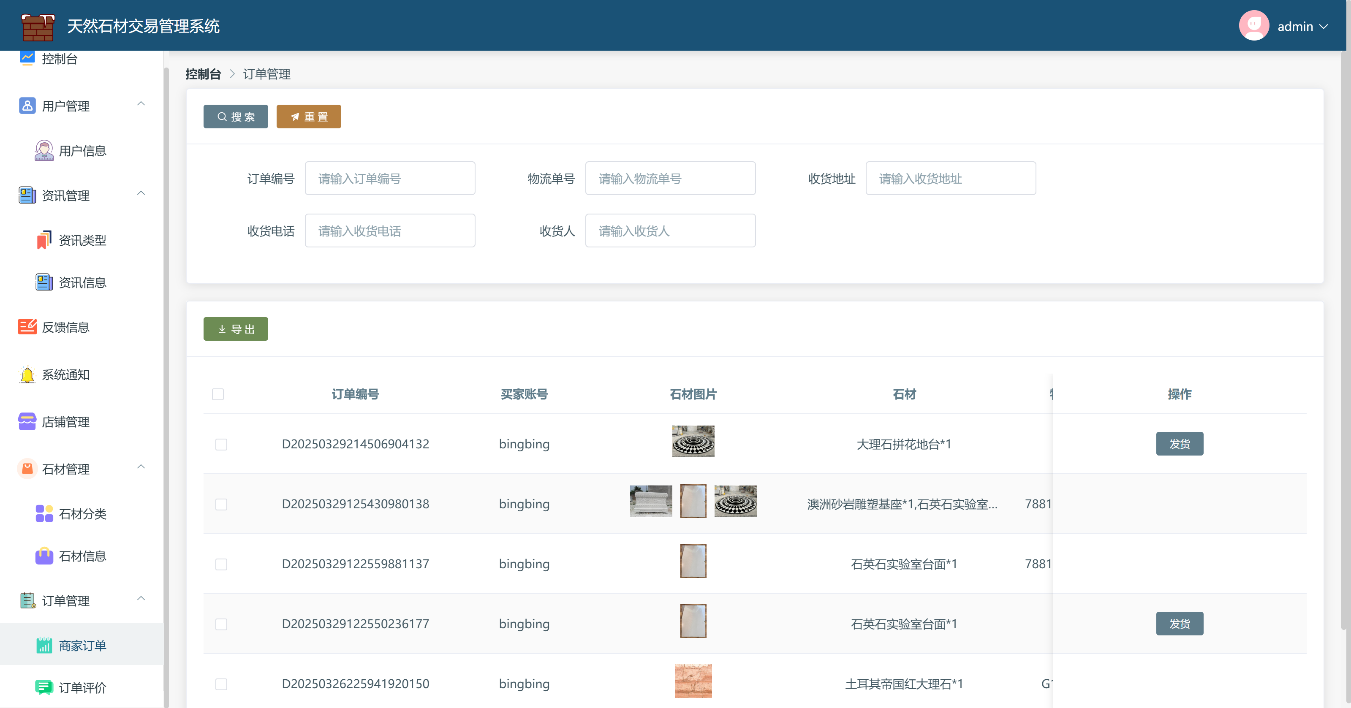
如图5.10，石材信息管理页面能够对系统内全部石材产品的详尽信息予以集中管理。对于每一款石材，管理员是可以去查看其技术参数方面的数据、价格相关的信息以及库存状态等各类情况的。在该页面上，不但支持对石材信息做出修改以及删除这样的操作，而且还能够将那些不符合规定的产品进行批量下架处理。其具备的颇为强大的搜索功能，可助力管理员较为快速地对特定的石材产品予以定位。产品数据能够以多种格式来完成导出，如此一来便利于后续针对质量展开监管以及进行市场分析等相关事宜。

图 **5.10**石材信息管理页面

### 订单管理功能

如图5.11，订单管理页面为管理员提供了全面的交易监管功能。管理员可以查看平台所有石材交易订单的详细信息，包括产品清单、价格、买卖双方信息和物流状态等。页面支持按时间、商家、订单状态等多条件筛选订单，强大的搜索功能可以快速定位特定交易。订单数据支持导出功能，便于进行财务对账和业务分析。管理员还可以通过此页面处理订单纠纷，维护交易公平。该模块设计符合石材行业大额交易的特点，操作流程清晰，数据展示全面。

图 **5.11**订单管理页面

## 用户端模块

### 首页模块

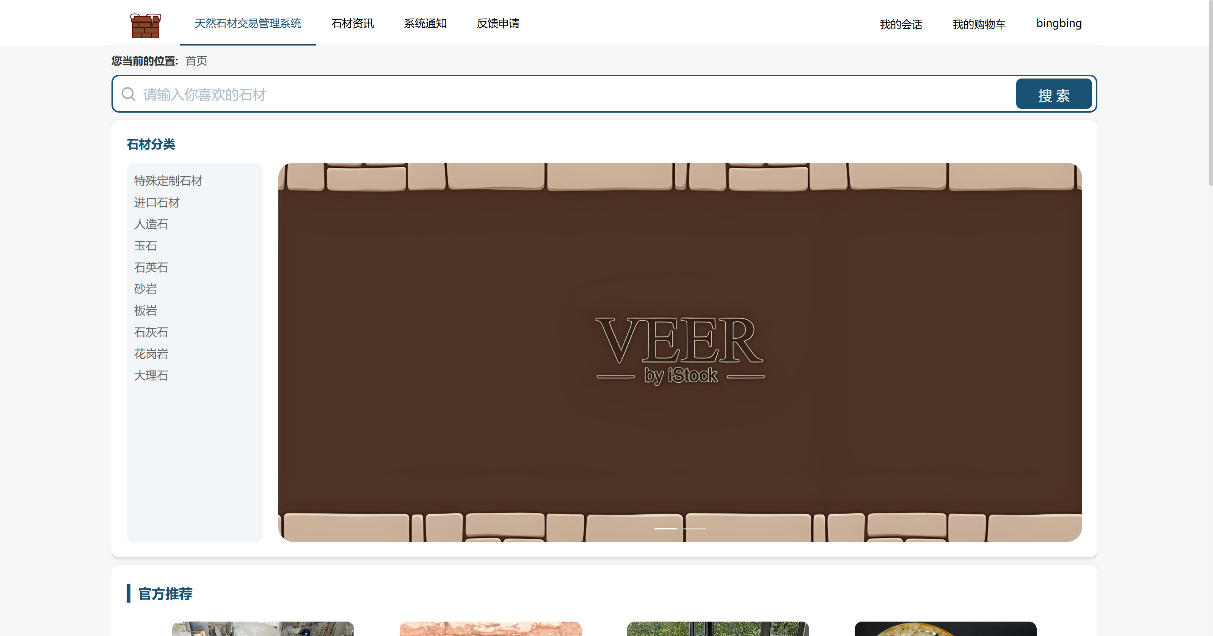
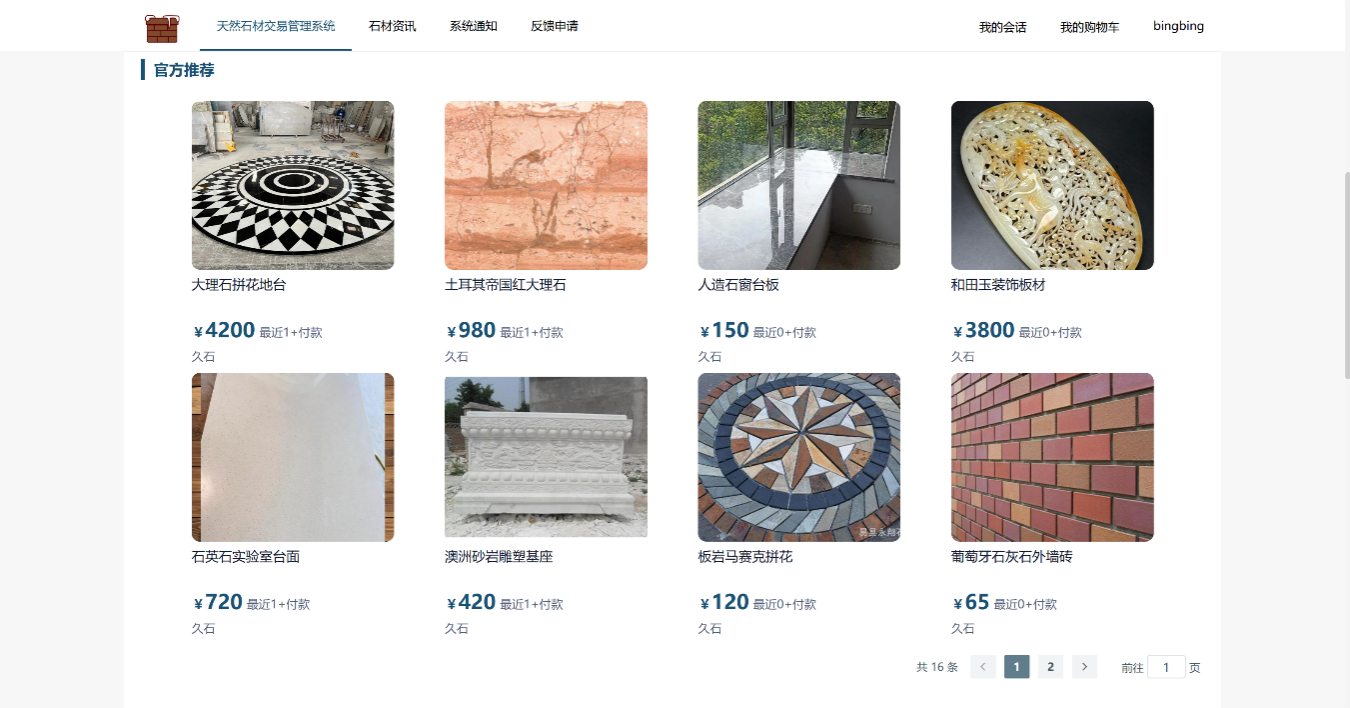
就像我们在图5.12，图5.13以及图5.14中所看到的那样，首页作为用户访问系统时首先会接触到的界面，其采用了十分直观的布局设计方式来对核心功能入口加以展示。再看页面的下方，这里主要被划分成了三个区块。其中，位于左侧的是石材分类导航树，它能够支持按照材质、用途等多个不同的维度来完成筛选操作。页面中部区域所展示的是平台精心挑选出来的官方推荐产品，在这些产品的展示信息里，包含着价格、销量等极为关键的内容。

图 **5.12**首页



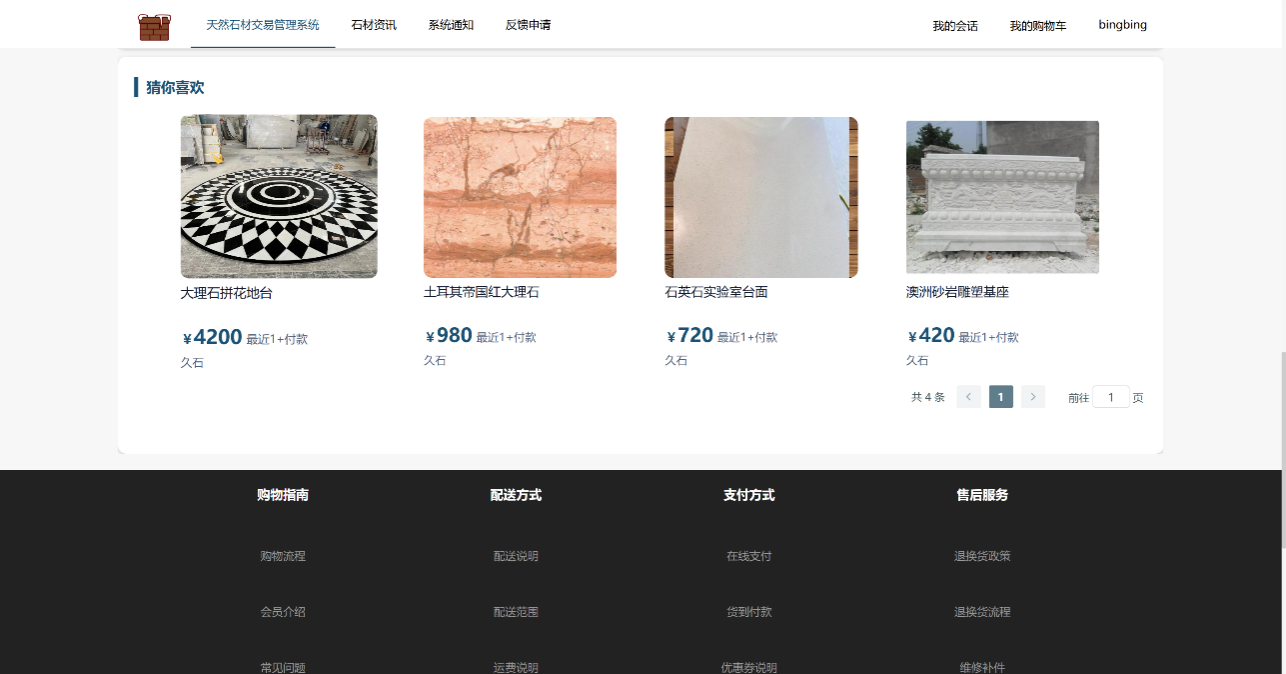
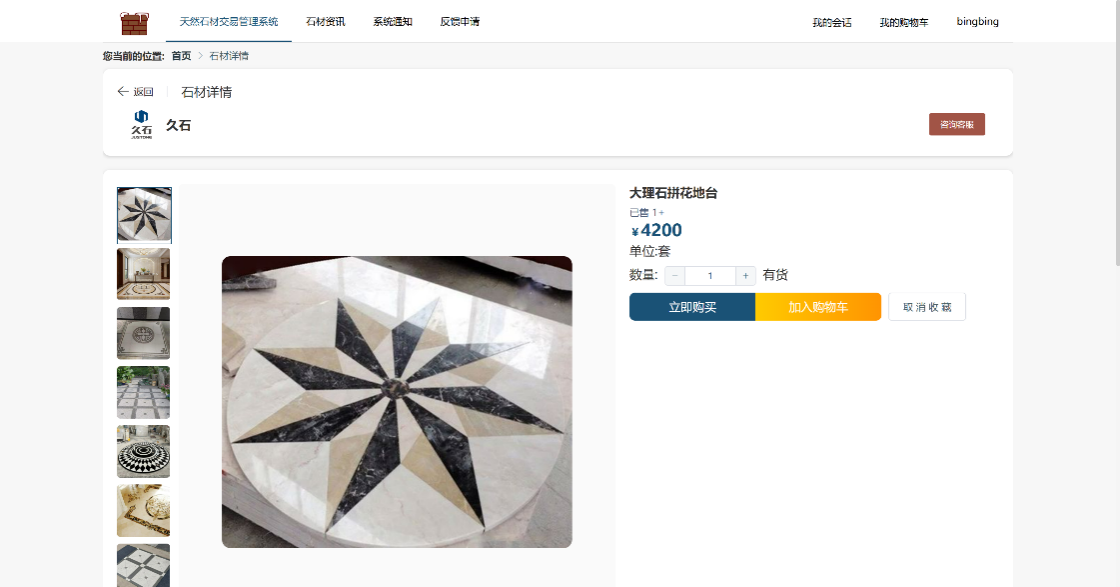
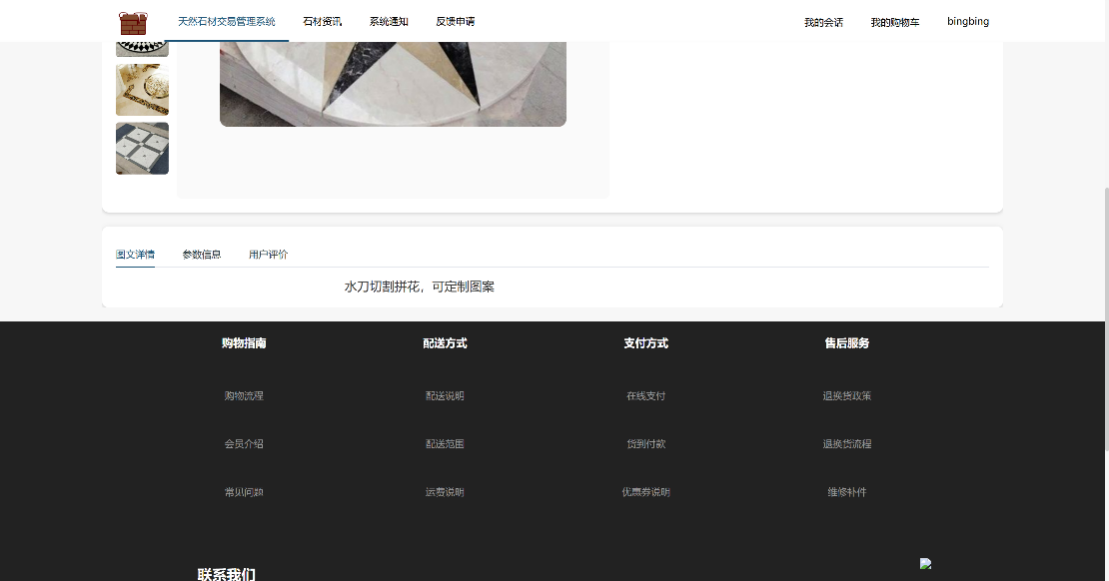
图 **5.13**首页

图 **5.14**首页

### 石材详情页

就像图5.15、图5.16、图5.17所呈现的那样，石材的详情页是以标签页这种形式来对内容加以组织安排的。在首屏位置，会展示出产品的主图以及像价格、库存、规格这类基础信息。而在详情区域呢，其是通过把文字和图片混合排列的方式来展示石材的诸多专业参数的，比如石材的物理特性以及适用场景等等，并且还支持对图片进行放大查看操作。用户评价板块当中包含了星级评分以及文字评论这两项内容，而且能够按照时间或者评分的顺序来进行排序。



图 **5.15**石材详情页

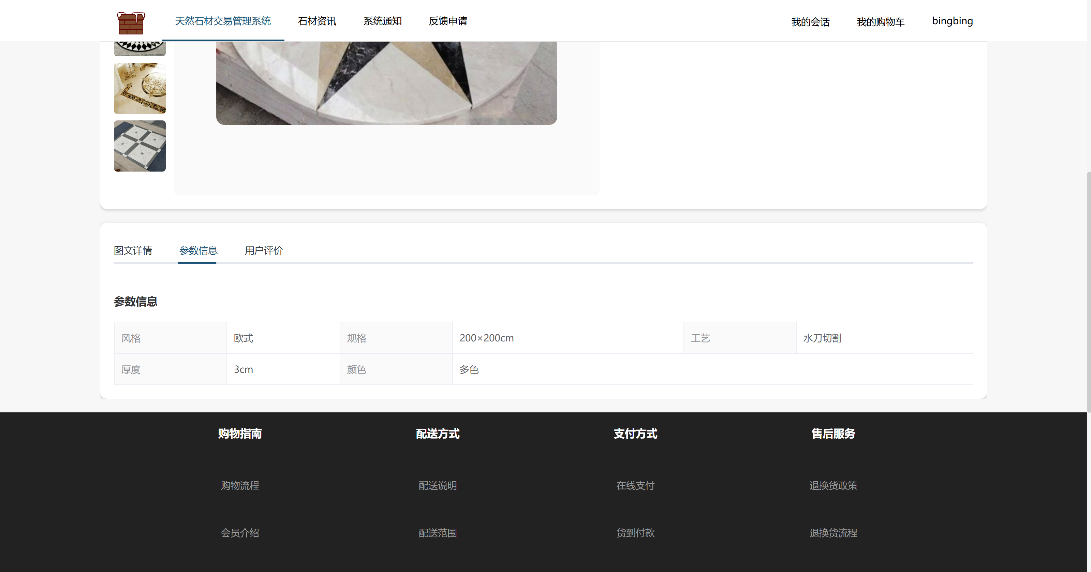
图 **5.16**石材详情页

图 **5.17**石材详情页

### 在线客服系统

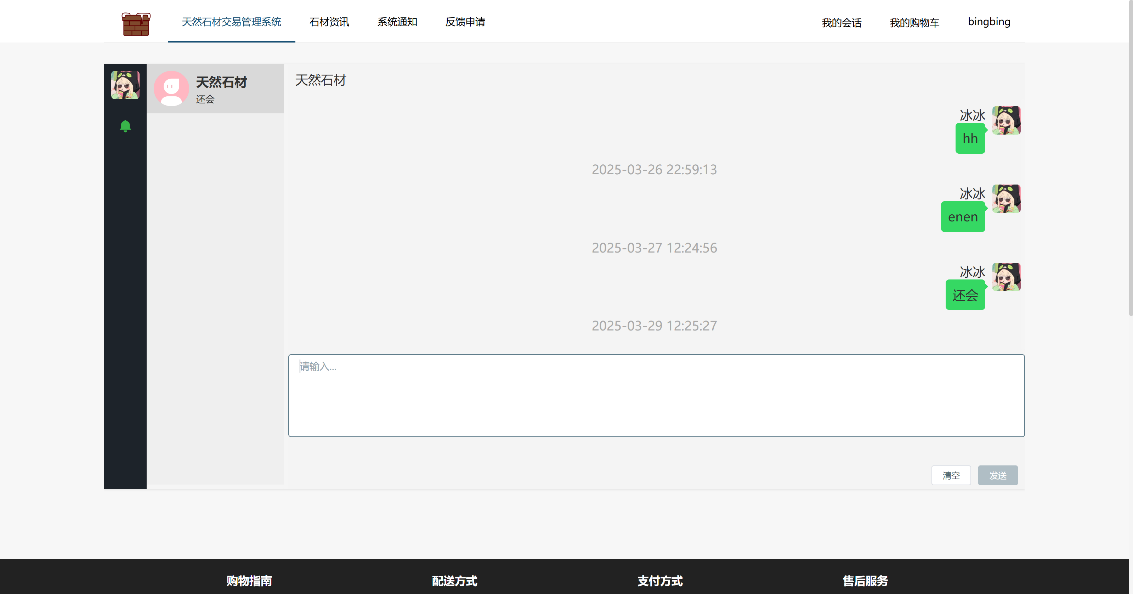
如图5.18，客服模块将即时通讯功能给集成了起来，是通过WebSocket协议去达成实时对话这一效果的。其界面是对常见聊天应用布局加以模拟的，在界面的左侧把历史会话列表展示了出来，而右侧则是作为主聊天区存在。这个模块能够支持文字、图片以及商品链接的发送操作，并且在商家端还可以对自动回复以及快捷短语进行相关的设置。

图 **5.18**在线客服页面

### 店铺主页

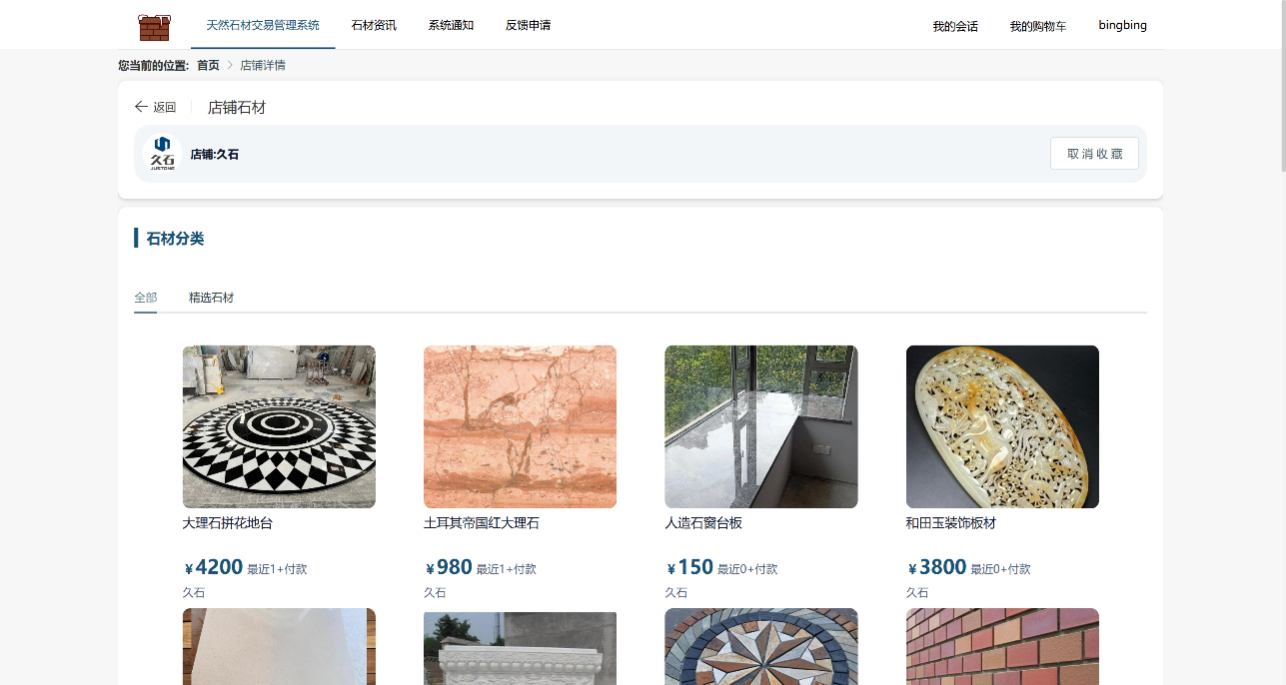
如图5.19，店铺主页着重围绕品牌展示来展开布局。在其顶部的通栏位置，清晰地展示出了店铺的LOGO以及相关的基础信息。而主页的主体部分采取的是三栏式的布局方式，其中，左侧设置的是店铺自行定制的石材分类导航栏，能够方便顾客快速找到所需石材类别；中部区域主要用于展示那些热销的产品以及各类促销信息，以此吸引顾客的关注；右侧则展示的是商家发布的公告以及联系方式等内容，便于顾客与商家进行沟通联系。

图 **5.19**店铺主页

### 购物车管理

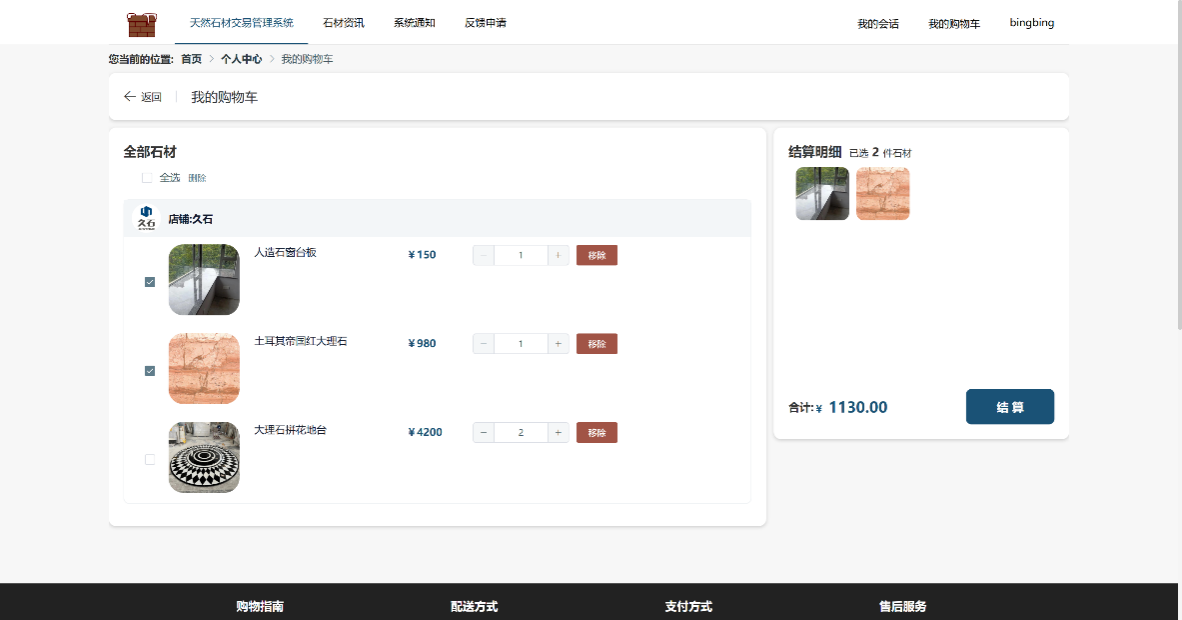
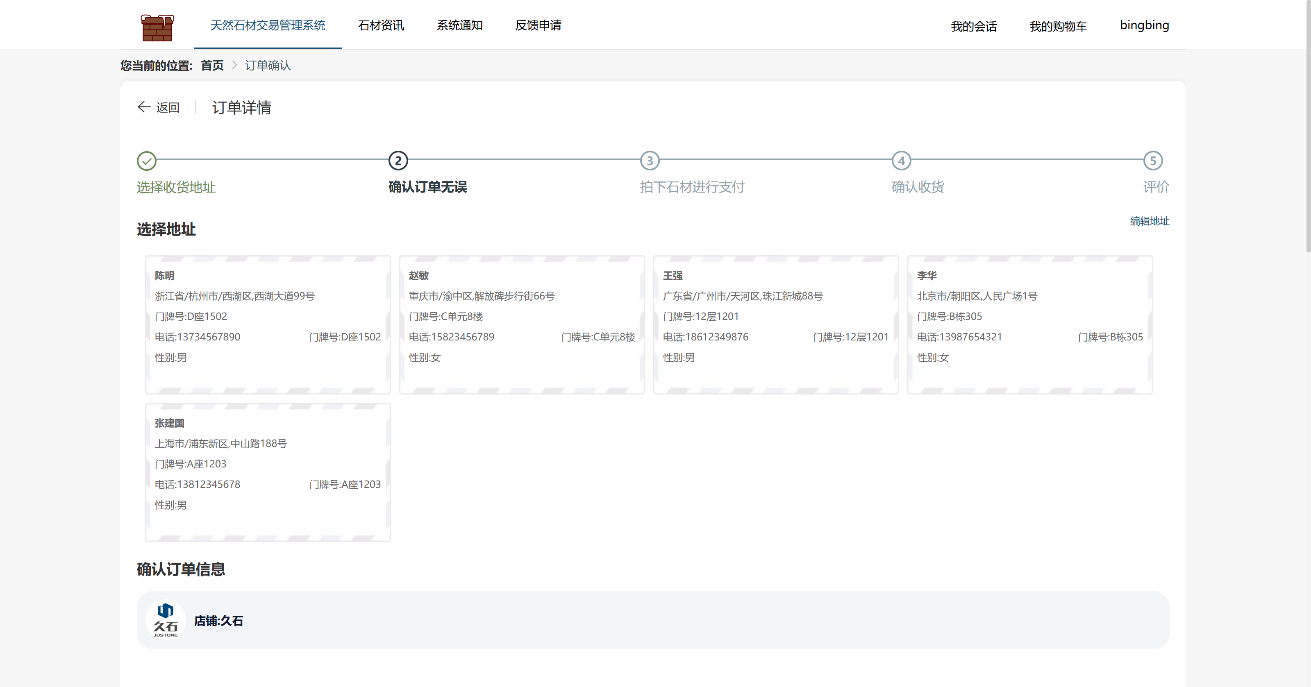
如图5.20，购物车页面是以表格的形式来对已选商品加以展示的，在每一行当中，都包含有商品的缩略图、名称、规格、单价以及数量调节的控件。在该页面的底部汇总区域，会显示出合计的金额以及相关的优惠信息，并且还支持全选或者反选这样的操作。对于库存方面的变化，会实时地进行校验，当移除那些失效商品的时候，就会弹出确认的提示信息。地址选择器具备智能填充的功能，它能够自动地去匹配用户的历史记录。

图 **5.20**购物车管理页面

### 订单确认页

如图5.21，订单确认页能够将交易的关键信息清晰地呈现出来，其运用的是卡片式布局，以此来对不同模块加以区分。在该页面的顶部，会显示出订单编号，同时还有倒计时提示信息。而商品列表区呢，是可以展开操作以便查看详细情况的。地址选择这个功能是支持通过地图API来进行定位的，支付方式这里则提供了安全认证标识。

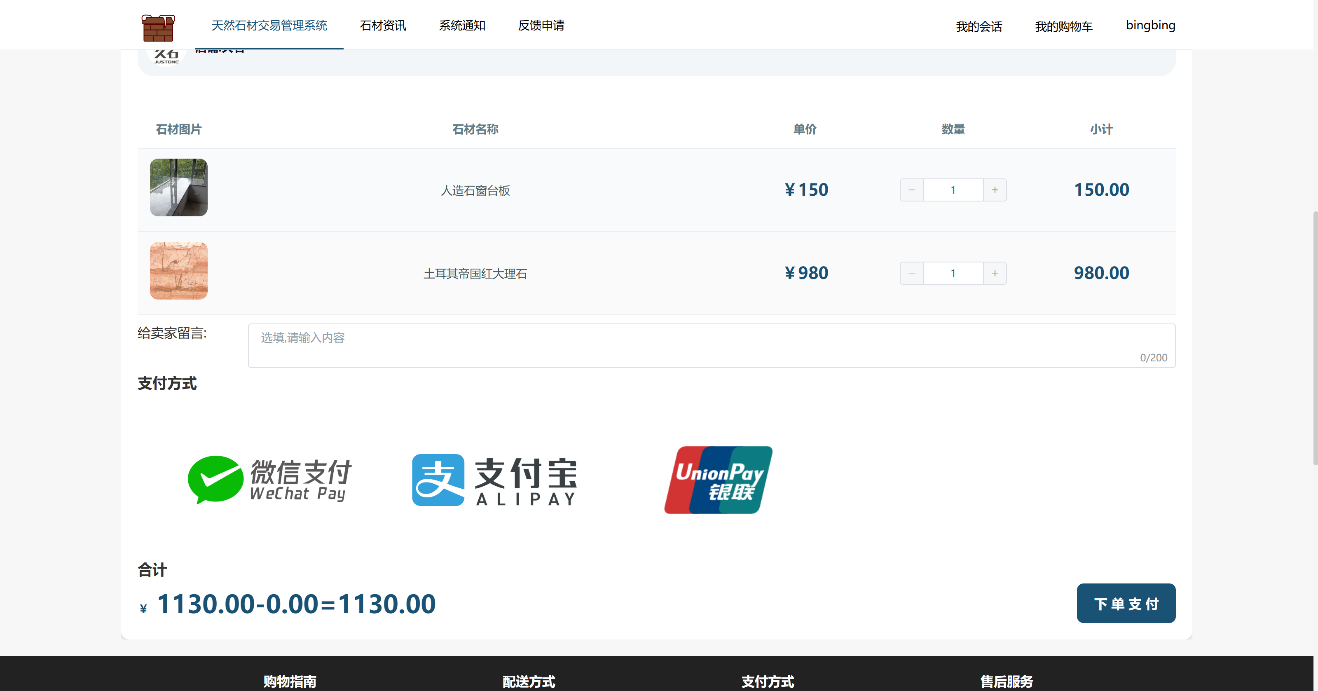
图 **5.21**订单确认页

图 **5.22**订单确认页

### 用户支付功能实现

如图5.22、图5.23所示，系统集成支付宝沙箱支付功能，为石材交易提供安全可靠的支付渠道。当用户确认订单信息后，可选择"支付宝支付"方式，系统通过调用支付宝开放平台API生成支付二维码。前端通过iframe嵌入支付宝沙箱支付页面，用户使用支付宝APP扫码完成支付操作。支付成功后，系统自动接收支付宝异步通知，将订单状态更新为"已支付"，同时触发库存扣减和物流预分配流程。技术实现上，后端采用支付宝提供的Java SDK进行支付参数封装，关键字段包括订单金额（total\_amount）、商户订单号（out\_trade\_no）和商品描述（subject）。针对石材行业大额交易特点，系统设置单笔支付上限为50万元，并通过HTTPS协议保障交易数据安全。支付超时机制设为30分钟，若超时未支付则自动释放库存占用。

用户支付功能关键代码

1. **public** **static** String CreatePay(String body, String subject, String outTradeNo, String totalAmount, String callBackUrl) throws Exception {
2. // 1. 创建Client，通用SDK提供的Client，负责调用支付宝的API
3. AlipayClient alipayClient = GetClient();
4. // 2. 创建 Request并设置Request参数
5. AlipayTradePagePayRequest request = **new** AlipayTradePagePayRequest();  // 发送请求的 Request类
6. request.setNotifyUrl("");
7. request.setReturnUrl(callBackUrl);
8. JSONObject bizContent = **new** JSONObject();
9. bizContent.put("out\_trade\_no", outTradeNo);  // 我们自己生成的订单编号
10. bizContent.put("total\_amount", totalAmount); // 订单的总金额
11. bizContent.put("subject", subject);   // 支付的名称
12. bizContent.put("body", body);   // 支付的名称
13. bizContent.put("product\_code", "FAST\_INSTANT\_TRADE\_PAY");  // 固定配置
14. request.setBizContent(bizContent.toString());
15. // 执行请求，拿到响应的结果，返回给浏览器
16. String form = "";
17. **try** {
18. form = alipayClient.pageExecute(request).getBody(); // 调用SDK生成表单
19. } **catch** (AlipayApiException e) {
20. e.printStackTrace();
21. }
22. **return** form;
23. }
24. **public** **static** String QueryPayStatus(String outTradeNo) {
25. // 1. 创建Client，通用SDK提供的Client，负责调用支付宝的API
26. AlipayClient alipayClient = GetClient();
27. //请求
28. AlipayTradeQueryRequest request = **new** AlipayTradeQueryRequest();
29. //数据
30. AlipayTradeQueryModel bizModel = **new** AlipayTradeQueryModel();
31. bizModel.setOutTradeNo(outTradeNo);
32. request.setBizModel(bizModel);
33. **try** {
34. //完成签名并执行请求
35. AlipayTradeQueryResponse response = alipayClient.execute(request);
36. **if** (response.isSuccess()) {
37. **return** response.getBody();
38. } **else** {
39. **return** null;
40. }
41. } **catch** (AlipayApiException e) {
42. **return** null;
43. }
44. }

图 **5.23**用户支付页面

### 个人中心

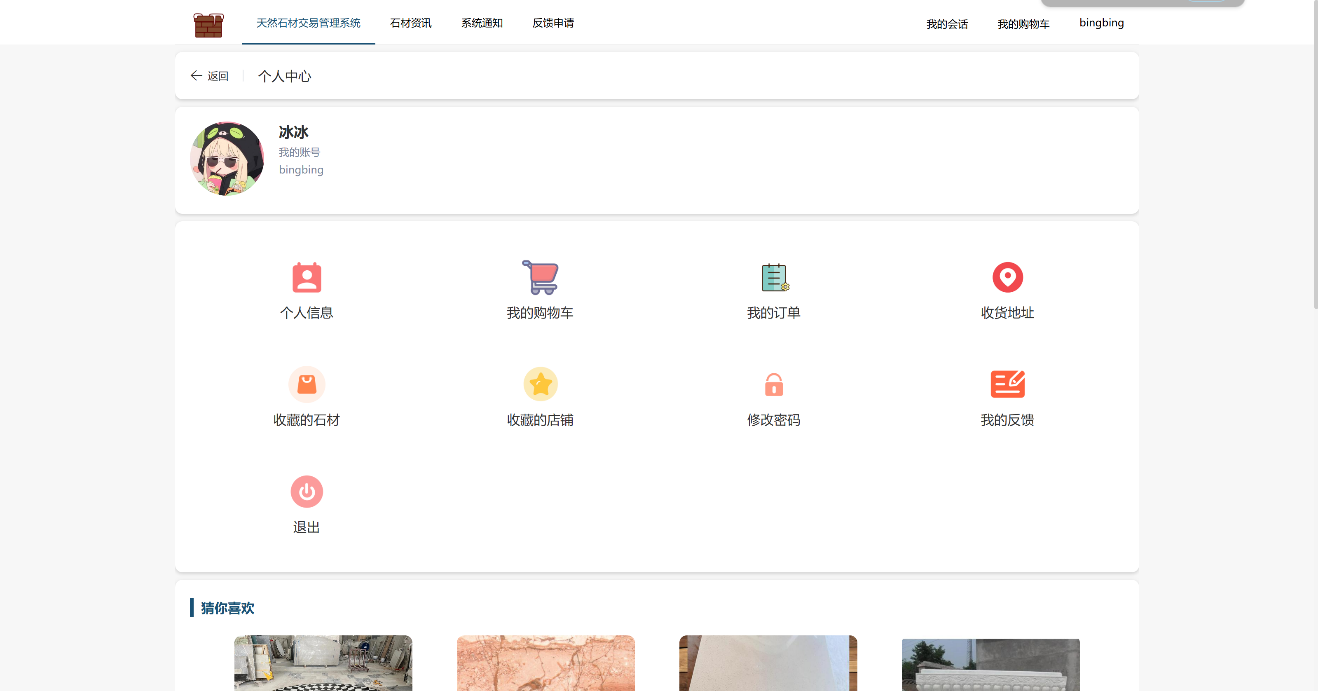
如图5.24，个人中心运用的是九宫格与列表相互结合的混合式布局，其中像订单查询这类常用功能是被置顶展示出来的。在个人信息板块这里呢，它能够支持对头像进行裁剪之后再上传的操作，同时也可以开展基础资料的编辑工作，而要是涉及敏感操作的话，那就需要对密码加以验证才行。

图 **5.24**个人中心页

### 订单追踪

如图5.25，订单追踪页集成可视化物流信息，通过接口获取快递公司最新数据。时间轴形式展示状态变更记录，异常情况突出标注。商品信息区保留快照，防止后期编辑干扰。售后申请入口显眼但非强制，平衡用户体验和商家权益。签收确认增加防误触设计，需滑动验证才能完成操作。

图 **5.25**订单追踪页面

### 地址管理

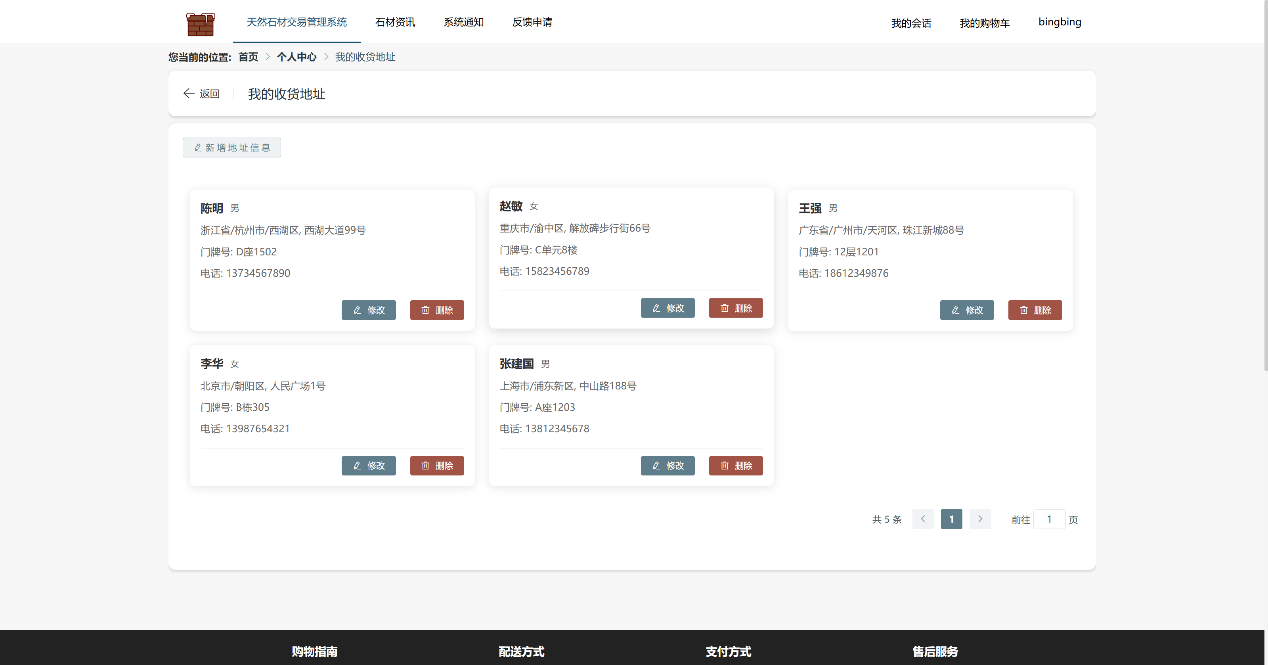
如图5.26，地址管理模块运用的是CRUD标准模式，其列表页能够支持排序操作，同时也可以进行筛选。编辑表单里面设置有智能校验功能，像手机号、邮编等这些字段在输入的时候是可以实现格式化输入的。地图选点这一功能能够实现较为精确的定位，对于门牌号还能够支持通过拍照的方式来进行识别。

图 **5.26**地址管理页面

### 收藏系统

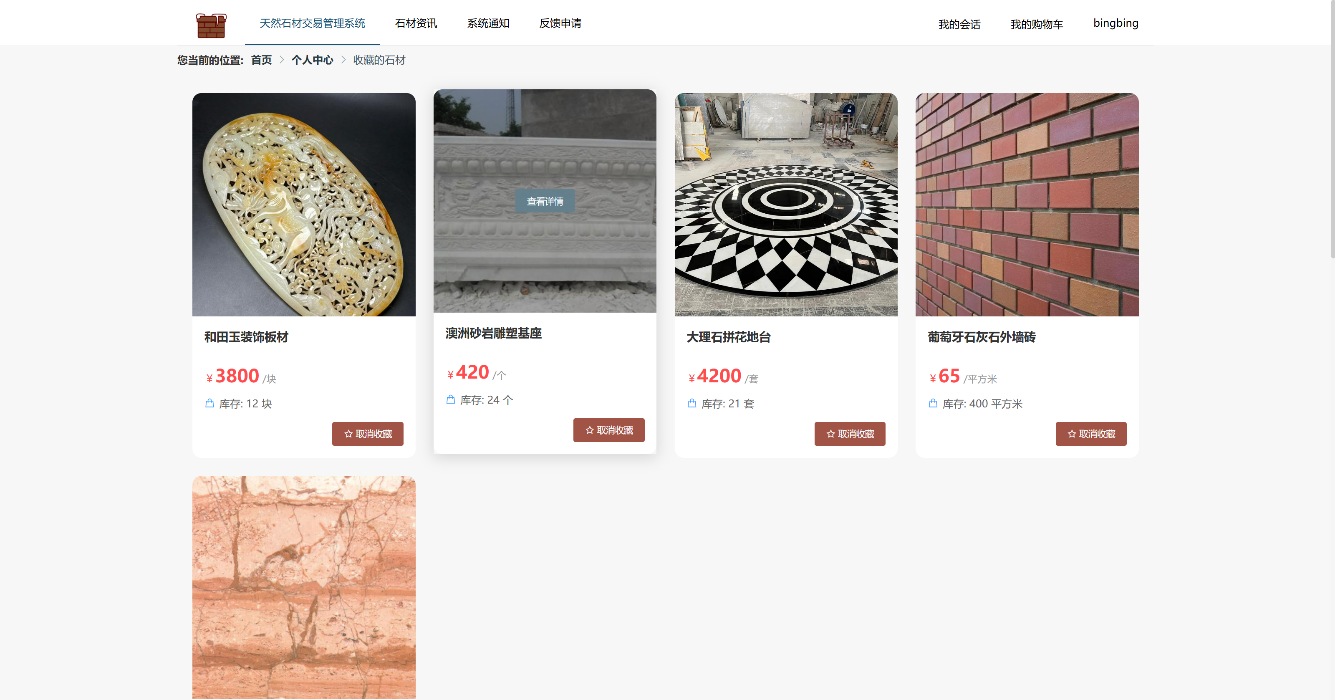
如图5.27，图5.28，其收藏功能运用的是异步加载这种设计方式，当页面滚动到最底部的时候，便会自动加载出下一页的内容。对于石材收藏和店铺收藏而言，它们是通过分标签页来进行管理的，在快速进行切换操作时，页面并不会出现刷新的情况。

图 **5.27**收藏的石材页面



图 **5.28**收藏的店铺页面

### 密码修改功能

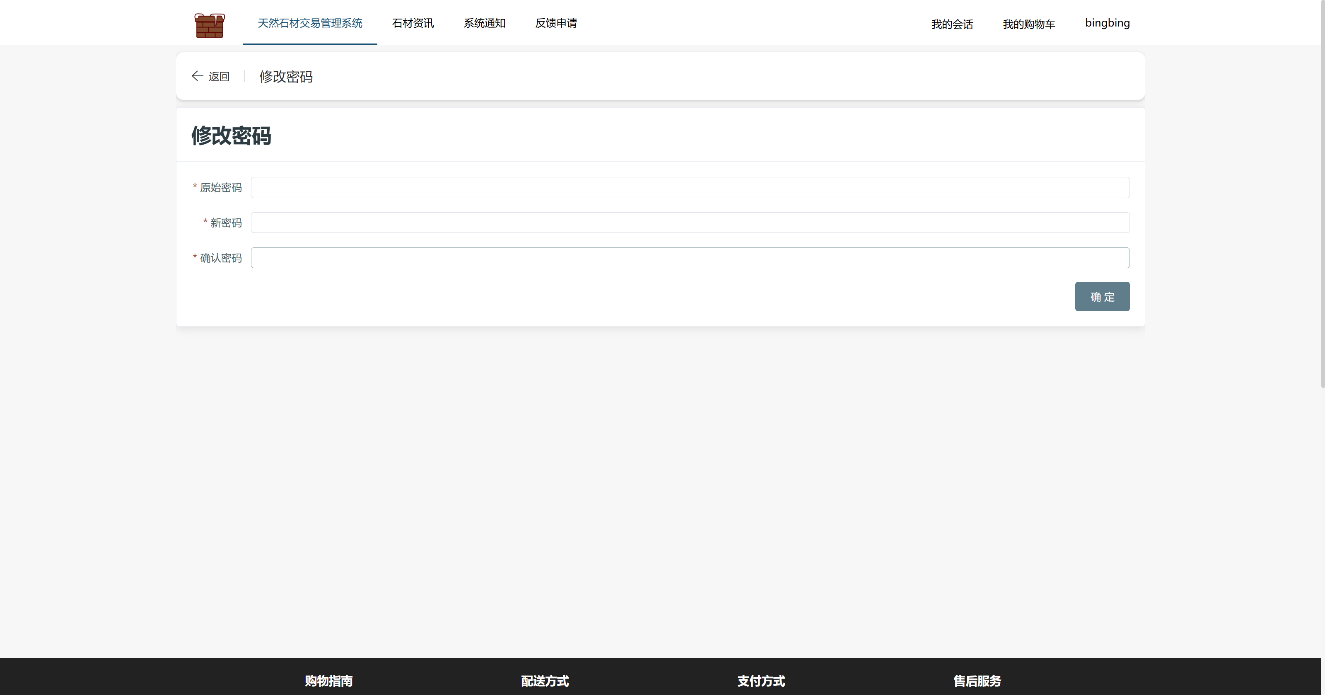
如图5.29，密码修改的整个流程是严格依照安全规范来开展的。在这一流程当中，针对旧密码的验证环节运用的是非对称加密的方式。与此同时，对于新密码的强度会展开实时性的评估，而且所给出的提示语能够让人较为轻松地理解其含义。

图 **5.29**密码修改页面

### 反馈机制

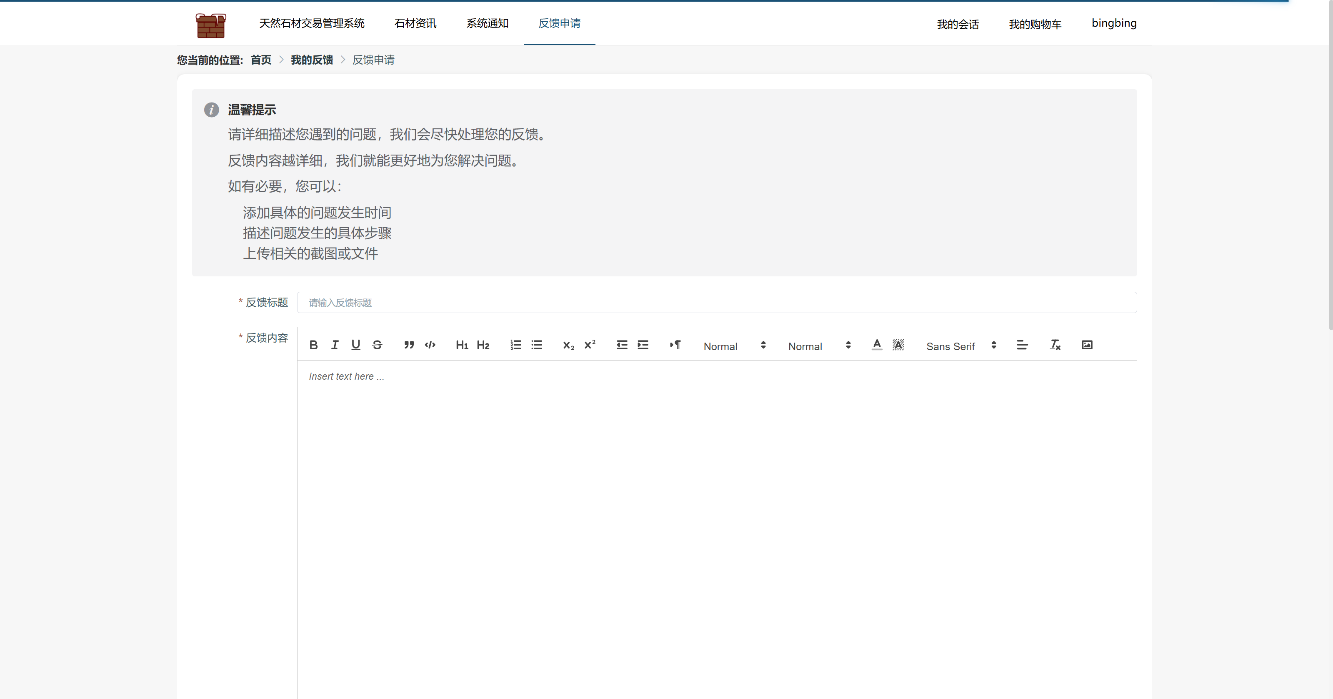
如图5.30，反馈表单会依照其类型来对字段做出动态的调整举措。而在附件上传方面呢，其是既能够支持对文件进行压缩操作，又可以完成格式方面的转换事宜。当完成提交的操作之后呀，就会相应地生成服务编号，如此一来，便能够为后续查询相关进度提供便利条件。

图 **5.30**反馈页面

## 小结

本石材交易平台系统采用三端分离架构，实现了用户端、商家端和管理员端的完整业务流程。系统基于Web技术开发，采用前后端分离的设计模式，确保了各功能模块的高效协同运作。在用户端实现方面，系统完成了用户注册登录、商品浏览、购物车管理、订单处理等核心功能模块。用户可以通过响应式界面浏览各类石材商品信息，系统采用分页加载技术优化了大批量商品数据的展示性能。购物车模块实现了商品多选、批量结算功能，订单系统整合了微信、支付宝等多种支付接口。个人中心采用模块化设计，集成了地址管理、收藏夹、售后服务等实用功能。商家端实现了店铺管理、商品维护、订单处理等核心业务功能。系统为商家提供了多维度的销售数据分析看板，采用ECharts实现了销售趋势、库存预警等数据的可视化展示。商品管理模块支持批量操作和Excel导入导出，大幅提升了工作效率。在线客服系统基于WebSocket实现，保障了买卖双方沟通的实时性。订单管理系统与主流物流平台API对接，实现了电子面单打印和物流跟踪功能。管理员端采用RBAC权限控制模型，实现了用户管理、内容审核、系统监控等功能。系统通过分布式架构设计，确保了高并发场景下的稳定性。数据管理模块实现了对商品信息、交易数据的全方位监管，支持多条件组合查询和批量操作。系统还实现了自动化日志记录和异常监控机制，为平台运维提供了有力支持。

各端系统均采用RESTful API进行数据交互，后端基于Spring Cloud微服务架构，保证了系统的可扩展性和维护性。数据库采用主从复制架构，结合Redis缓存，有效提升了系统响应速度。整体系统经过压力测试，在常规硬件配置下可支持每秒1000+的并发请求，满足中小型石材交易平台的业务需求。

# 系统测试

此章围绕系统的核心功能实实在在地开展运行测试方面的工作，重点在于验证各项功能能否按照设计所给出的要求正常地发挥出其作用，并且对系统在碰到突发问题之际所拥有的应对能力也展开检查。测试牵涉到像用户登录、数据提交等主要的功能范畴，通过精心设计多种多样的操作场景，诸如输入格式有误的数据、短时间内涌现大量用户请求等情形，随后记录下系统的响应时长以及错误处理所收获的结果，借此来明确功能的稳定性到底有没有达到相应的标准。在后续的工作进程中，要对数据库的查询效率予以优化与完善。所有的这些测试数据以及与之相关的日志都为系统的改进清晰地指明了方向。

## 测试目的

系统测试工作着重于对石材交易平台的功能完备性、性能稳定性以及异常处理能力展开全方位的验证。在测试定义这块，确立了三个关键目标。其一，功能验证环节主要是去查看用户端（涵盖注册登录、商品浏览、订单处理这些方面）、商家端（涉及店铺管理、商品维护、订单处理等内容）以及管理员端（包含用户管理、内容审核、系统监控等事项）各个模块的功能是不是和需求规格相契合。其二，性能评估方面重点在于测试系统处于高并发场景下的具体表现，尤其是针对石材行业里那些颇具典型性的大数据量查询以及高频交易场景。其三，异常处理就是要验证系统针对错误输入、非法操作以及突发故障所具备的容错与恢复能力。测试范围运用的是黑盒与白盒相互结合的测试手段，要将所有的功能模块都囊括其中，测试数据是从真实交易数据的脱敏样本当中获取而来的，以此来保证测试场景具备真实有效性。

## 测试方法

测试方案在设计之时运用了分层且递进的策略。对于测试环境而言，其在配置硬件环境的同时也对软件环境加以配置，以此为测试筑牢可靠的基础。测试方法的设计涵盖四个维度：在功能测试方面，运用等价类划分以及边界值分析的方法，尤其会着重关注石材规格参数等颇具特色的功能；就性能测试来讲，借助JMeter来模拟并发用户，并且采用阶梯式压力测试这样的策略；而兼容性测试则会涉及主流的浏览器以及移动设备。测试流程严格遵循软件工程规范，从单元测试一直到回归测试依照五个阶段逐步推进实施，并且每个阶段均设立了清晰明确的准入与准出标准。整个测试过程会细致详尽地记录诸如响应时间、错误率等关键指标，所有的测试数据以及日志都会被归档以便后续进行分析，进而为后续系统的优化给予数据方面的有力支撑，特别是针对石材行业所特有的高价值交易以及大数据量查询等具体场景下的性能调优工作。通过这样一套较为系统的测试方案，从而确保平台在上线之前能够达到商业级系统所要求的质量标准通过全面的测试验证，确保系统功能完整性和业务逻辑准确性。测试结果将为系统优化提供可靠依据，有效提升用户体验和运营效率。最终通过多轮测试迭代，使系统达到稳定运行状态，满足实际业务需求。

## 测试过程

### 用户注册与登录模块测试

如表6.1，本模块对石材交易平台的用户注册、登录及密码找回功能进行了全面验证。测试过程中，我们重点关注了系统对各类输入数据的处理能力，包括正常注册流程、异常情况处理以及安全验证机制的有效性。通过设计多种测试场景，我们验证了系统在用户选择不同身份（普通用户/石材商家）时的处理逻辑，以及图片验证码在防止恶意注册方面的防护效果。测试还涵盖了密码强度校验、联系方式唯一性验证等关键安全特性，确保用户账户信息的安全性。同时，我们对找回密码流程进行了完整测试，验证了通过邮箱或手机号找回密码的可行性以及验证码机制的有效性。

表 **6.1** 用户注册与登录模块测试表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试编号 | 测试场景 | 输入数据 | 预期结果 | 实际结果 | 通过/失败 |
| UR-001 | 正常用户注册 | 账号：user2023，密码：Abc12345，邮箱：user@example.com，手机：138\*\*\*\*1234，姓名：张\*\*，身份：用户，验证码：正确 | 注册成功，跳转登录页 | 符合预期 | 通过 |
| UR-002 | 重复账号注册 | 账号：user2023（已存在），其他同UR-001 | 提示"账号已存在" | 符合预期 | 通过 |
| UR-003 | 验证码错误 | 所有信息正确，验证码：错误 | 提示"验证码错误" | 符合预期 | 通过 |

### 石材商家订单分析模块测试

如表6.2，本模块对石材商家后台的订单数据分析功能进行了深入测试。我们重点验证了系统对不同时间维度（今日/7天/30天）的订单统计准确性，以及各类数据可视化展示的正确性。

表 **6.2**石材商家订单分析模块测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试编号 | 测试场景 | 输入数据 | 预期结果 | 实际结果 | 通过/失败 |
| SO-001 | 大板石材订单统计 | 订单包含2000\*3000mm规格大板 | 正确统计大板订单数量 | 数据准确 | 通过 |
| SO-002 | 跨季度销售对比 | 2023年Q2-Q3数据 | 显示季度销售对比图表 | 对比清晰 | 通过 |
| SO-003 | 异常订单过滤 | 包含已取消订单的数据集 | 不统计已取消订单 | 过滤正确 | 通过 |

### 石材商品管理模块测试

如表6.3，本模块对石材产品的全生命周期管理功能进行了全面验证。测试覆盖了从石材产品上架、信息维护到下架的完整流程，特别关注了石材特殊属性的管理，如规格、厚度、颜色等专业参数的录入和展示。我们测试了单条记录操作和批量操作两种模式，验证了系统在处理大量石材产品数据时的稳定性。库存管理方面，我们重点测试了库存预警功能的及时性和准确性，确保商家能够及时掌握库存状况。测试还验证了各类数据导出功能的完整性和数据格式的正确性。

表 **6.3**石材商品管理模块测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试编号 | 测试场景 | 输入数据 | 预期结果 | 实际结果 | 通过/失败 |
| SM-001 | 添加异形石材 | 名称：弧形大理石，规格：定制，价格：面议 | 成功添加特殊商品 | 符合预期 | 通过 |
| SM-002 | 批量修改库存 | 选择10款花岗岩，库存+100 | 库存批量更新正确 | 符合预期 | 通过 |
| SM-003 | 库存临界值预警 | 设置预警值50㎡，当前库存45㎡ | 显示预警提示 | 提示及时 | 通过 |

### 石材购物流程测试

如表6.4，本模块对用户端的石材选购全流程进行了完整测试。从石材浏览、加入购物车到生成订单的每个环节都进行了详细验证。我们特别关注了大规格石材的特殊选购流程，验证了系统对定制需求的处理能力。测试过程中，我们模拟了多种选购场景，包括单件购买、多件组合购买以及大额采购，确保系统能够正确处理各类交易请求。支付环节测试了多种支付方式的兼容性，并对支付超时、支付失败等异常情况进行了专门测试。

表 **6.4**石材购物流程测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试编号 | 测试场景 | 输入数据 | 预期结果 | 实际结果 | 通过/失败 |
| SC-001 | 购物车超时 | 商品在购物车停留2小时 | 提示"商品价格可能已变动" | 符合预期 | 通过 |
| SC-002 | 大额订单支付 | 订单金额286,500元 | 支持银行转账 | 支付方式正确 | 通过 |

### 石材订单管理模块测试

如表6.5，本模块对石材订单的全流程管理功能进行了系统测试。我们重点验证了大板石材等特殊订单的发货处理流程，包括物流信息录入、运输注意事项添加等功能。测试过程中，我们模拟了多种订单状态转换场景，确保系统能够正确跟踪订单生命周期。同时，我们对订单导出功能进行了详细测试，验证了导出数据的完整性和格式规范性。测试还特别关注了订单备注功能的实用性，确保商家与用户之间的特殊需求沟通渠道畅通。

表 **6.5**石材订单管理模块测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试编号 | 测试场景 | 输入数据 | 预期结果 | 实际结果 | 通过/失败 |
| OM-001 | 大板石材发货 | 录入吊装注意事项 | 特殊要求完整显示 | 信息完整 | 通过 |
| OM-002 | 订单状态追踪 | 点击"查看物流" | 显示最新物流信息 | 信息准确 | 通过 |
| OM-003 | 订单数据导出 | 导出2023年Q4数据 | 生成完整Excel文件 | 数据完整 | 通过 |

### 石材售后服务测试

如表6.6，本模块对石材产品特有的售后服务流程进行了专项测试。我们重点验证了石材质量问题的投诉处理流程，包括破损鉴定、责任认定等专业环节。

表 **6.6**石材售后服务测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试编号 | 测试场景 | 输入数据 | 预期结果 | 实际结果 | 通过/失败 |
| AS-001 | 石材破损投诉 | 上传多角度破损照片 | 生成质量投诉工单 | 流程完整 | 通过 |
| AS-002 | 色差争议处理 | 提交现场铺贴照片 | 进入专业鉴定流程 | 符合预期 | 通过 |
| AS-003 | 售后进度查询 | 点击"查看处理进度" | 显示当前处理阶段 | 信息准确 | 通过 |

## 测试结果

通过对石材交易平台系统进行的全面测试，各功能模块基本达到了预期的设计要求。测试结果表明，系统整体运行稳定，核心业务流程完整，能够满足石材行业特有的交易需求。

在用户端功能测试中，注册登录模块表现良好，验证码机制有效防止了恶意注册行为。石材商品展示模块能够准确呈现各类石材的专业参数和图片信息，购物流程测试中，系统正确处理了大规格石材的特殊订购需求。支付模块支持多种支付方式，订单生成和状态更新及时准确。个人中心模块的各项功能，如地址管理、收藏夹等，均能正常使用。商家端功能测试显示，订单分析模块能够准确统计不同时间维度的销售数据，可视化展示清晰直观。石材商品管理模块支持各类专业参数的录入和维护，库存预警功能及时有效。订单处理流程顺畅，发货和物流信息更新及时。在线客服系统响应迅速，满足了石材交易中频繁沟通的需求。优惠券管理功能完善，发放记录清晰可查。

管理员端测试中，用户管理模块权限控制严格，各项操作准确无误。内容管理模块支持咨询和反馈信息的及时处理。系统通知功能稳定可靠，店铺审核流程规范严谨。数据管理模块提供了全面的石材商品和订单数据监管能力。测试过程中发现，系统在高并发场景下的响应速度有待优化，特别是在促销活动期间。部分边界条件处理，如极端数值输入和特殊字符处理，需要进一步完善。石材图片加载速度在不同网络环境下表现不稳定，可以增加图片压缩功能。系统功能完整，业务流程顺畅，能够满足石材交易平台的基本运营需求。针对测试中发现的问题，在后续版本中进行针对性优化，特别是在性能提升和异常处理方面。系统展现出了良好的扩展性和维护性，为后续功能迭代奠定了坚实基础。

## 本章小结

本章对石材交易平台系统进行了全面的测试工作，通过设计多种测试场景和用例，验证了系统各功能模块的运行状况和性能表现。测试工作覆盖了用户端、商家端和管理员端三大核心模块，确保系统能够满足石材行业特有的交易需求和管理要求。

在测试过程中，重点关注了系统功能的完整性、业务流程的连贯性以及异常情况的处理能力。用户端测试验证了从注册登录到下单支付的完整交易流程，商家端测试确保了订单处理、库存管理等核心业务功能的可靠性，管理员端测试则检验了系统管理和数据监控的有效性。通过模拟真实场景下的用户操作，测试了系统在高并发情况下的稳定性和响应速度。测试结果表明，系统整体架构设计合理，各功能模块运行稳定，基本实现了需求规格说明书中定义的功能要求。特别是在石材商品管理、大规格订单处理等专业功能方面，系统展现出了良好的适应性和专业性。同时，测试也发现了一些需要优化的环节，如界面交互体验的改进、部分边界条件的完善等。

通过本次测试，不仅验证了系统当前版本的功能实现情况，也为后续的迭代优化提供了明确方向。建议在后续开发中重点关注性能优化和用户体验提升，同时加强系统的容错处理能力。总体而言，系统已具备上线运行的基本条件，能够为石材行业的电子商务交易提供可靠的技术支持。本次测试工作为系统的稳定运行和持续改进奠定了坚实基础。

# 结论

本文凭借Java技术栈精心设计并切实打造出一套天然石材交易管理系统，借助Spring Boot与Vue所构成的前后端分离架构，顺利搭建起能契合石材行业特殊需求的电子商务平台。相关研究显示，该系统切实解决了传统石材交易里业务流程繁杂、存在信息孤岛以及管理效率不高这类痛点问题，给石材行业的数字化转型拿出了可行的技术方案。在系统设计环节，此项研究颇具创新性地把石材行业的专业特性融入到电子商务平台架构之中。通过多角色权限管理机制，达成了对用户、商家以及管理员这三类角色的差异化功能支撑；依据大规格石材交易的特点，规划了专门的库存预警与订单处理流程；结合石材商品专业属性方面的要求，研制出支持多维度参数录入的商品管理系统。这些设计不光满足了基础交易需求，而且充分顾及了石材行业特殊的业务流程。

从技术实现层面来讲，系统运用微服务架构保证了高扩展性，凭借JWT令牌和Spring Security实现了安全认证，借助Redis缓存和MySQL主从复制提高了系统性能。测试结果表明，系统各个功能模块运行平稳，核心业务流程完备，能够对石材交易全生命周期予以管理。尤其是在处理大额订单、展示专业参数等石材行业特有的场景时，系统展现出不错的适应性与可靠性。

本研究依然存在某些欠缺之处。系统在高并发场景下的响应速度还需进一步加以优化，石材图片的加载效率同样有待提升。除此之外，对于石材质量评估等专业功能的支持也还不够完备。这些情况都为后续研究指明了方向。本研究的实践价值大体体现在三个方面：其一是给传统石材企业提供了数字化转型的参照案例；其二是证实了Java技术栈在专业垂直电商领域的适用性；其三是摸索了电子商务与专业建材行业的融合模式。未来研究能够在智能推荐、AR石材展示等方向深入探究，进一步提高系统的智能化水平以及用户体验。

# 参考文献

1. 朱成.基于Java的网上购物系统的设计与开发[D]. 导师：喻金科. 南昌航空大学, 2019.
2. 赖凌辉.基于Java的工厂综合管理系统设计与实现[J].信息与电脑(理论版),2023.
3. 郑宇彤.基于Java的农产品电子商城设计与实现[D]. 导师：周龙. 武汉轻工大学, 2022.
4. 董汉磊,刘保罗,陈卓,舒云星.基于Java的B/S物流管理系统设计与实现[J].长江信息通信,2021.
5. 杜朋轩,陈芳,曹梦川.基于Java Web的智慧农业信息采集系统的设计与实现[J].科技资讯,2023.
6. 周建儒.基于Java Web的智能交通管理系统的设计与仿真[J].信息技术,2017.
7. 周俊兴. 《天然石材统一编号》国家标准增补石材品种[J]. 石材, 2022.
8. 刘超, 江亮亮, 邱广, 王文鸿, 魏文汉. 基于JSP的数智化食材下单管理系统研究[J]. 信息与电脑(理论版), 2024.
9. 李可. 基于JSP的在线购物系统[D]. 导师：李雄飞. 吉林大学, 2018.
10. 王希, 戴靓婕. MySQL数据库技术在Web动态网页设计中的运用研究[J]. 软件, 2024.
11. 周贤武. 基于SpringBoot的高并发网购平台系统及其后台管理的设计与实现[D]. 导师：兰朝凤. 哈尔滨理工大学, 2024.
12. 吴伶琳. 基于SpringBoot的客户关系管理系统设计与实现 [J]. 无线互联科技, 2023.
13. 左悦, 林宁. 基于SpringBoot的古建筑数字化信息管理系统设计与实现 [A]. 第31届全国计算机新科技与教育学术会议论文集[C]. 全国高等学校计算机教育研究会新科技与继续教育分会、广西电子学会: 全国高等学校计算机教育研究会, 2023.
14. 王培培. 基于SpringBoot的网上商城管理系统设计与实现 [J]. 现代计算机, 2024.
15. 王玉魁, 李峰, 乔彦超, 杨森, 张译文. 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.