目录

[1 绪 论 1](#_Toc30519)

[1.1 研究背景及意义 1](#_Toc2831)

[1.2 国内研究状况及分析 1](#_Toc8859)

[1.3 研究目标、内容和关键技术 2](#_Toc29035)

[1.4 术语说明 3](#_Toc22670)

[2 详细设计 4](#_Toc11421)

[2.1 用户功能设计 4](#_Toc18301)

[2.1.1 用户搜索物品功能 4](#_Toc22372)

[2.1.2 用户留言功能 6](#_Toc15366)

[2.1.3 用户修改个人信息功能 7](#_Toc15270)

[2.1.4 用户添加物品功能 8](#_Toc26977)

[2.1.5 用户修改物品信息功能 10](#_Toc10450)

[2.1.6 用户删除物品功能 11](#_Toc10673)

[2.2 管理员功能设计 12](#_Toc27862)

[2.2.1 管理员审核用户信息功能 12](#_Toc1516)

[2.2.2 管理员审核物品功能 13](#_Toc18917)

[3 系统实现 14](#_Toc31336)

[3.1系统配置 14](#_Toc14545)

[3.1.1 数据库实现 14](#_Toc32618)

[3.1.2 前端实现 14](#_Toc17553)

[3.1.3 后端实现 16](#_Toc3832)

[3.2 用户功能实现 17](#_Toc450)

[3.2.1 用户搜索商品功能 17](#_Toc17151)

[3.2.2 用户修改个人信息功能 19](#_Toc2309)

[3.2.3 用户商品管理功能 20](#_Toc21000)

[3.2.4 用户留言功能 21](#_Toc8589)

[3.3 管理员功能实现 23](#_Toc2212)

[3.3.1 用户信息审核功能 23](#_Toc16266)

[3.3.2 物品审核功能 24](#_Toc20188)

[4 系统测试 26](#_Toc30205)

[4.1 单元测试 26](#_Toc137)

[4.1.1 login方法 26](#_Toc13359)

[4.2 集成测试 27](#_Toc4238)

[4.2.1 登陆功能和首页展示 27](#_Toc494)

[4.2.2 物品搜索功能 28](#_Toc5379)

[4.2.3 个人信息修改功能 28](#_Toc30234)

[4.2.4 添加商品功能 29](#_Toc21506)

[4.2.5 商品信息修改功能 30](#_Toc1585)

[4.2.6 商品删除功能 31](#_Toc23963)

[4.2.7 申请查询功能 31](#_Toc22635)

[4.2.8 处理接收申请功能 32](#_Toc17409)

[4.2.9 用户信息审核功能 32](#_Toc3999)

[4.2.10 用户信息审核功能 32](#_Toc23128)

[总结体会 33](#_Toc10079)

[参考文献 34](#_Toc22246)

# **1 绪 论**

## **1.1 研究背景及意义**

随着全球资源稀缺性问题不断升级，闲置二手交换作为一种资源可持续利用的方式，可以减少新资源的开采，延长产品生命周期，降低资源浪费，有助于可持续发展的实现。

在现代社会，许多人面临高昂的生活成本和经济压力。闲置二手交换能够为个人提供了经济实惠的途径，帮助他们以更低的成本获取所需的物品，有效地管理预算。闲置二手交换还可以在社区内促进资源共享和社交互动。个人和社区居民通过这种方式建立联系，分享资源，形成更为紧密的社区网络，这有助于社区建设和社交互动的提升，有助于减少废物的产生，降低废物处理和填埋的压力，有益于环境保护，减少污染，以及保护生态系统的健康。通过闲置二手交换，人们能够更有效地利用现有资源，提高资源利用效率，减少浪费，从而降低了对有限资源的过度依赖。

与此同时，互联网和移动技术的崛起为闲置二手交换提供了便捷的平台，促进了二手市场的蓬勃发展。研究如何更好地利用技术、数据分析和人工智能来推动闲置二手交换的发展，是当前数字时代的重要议题。目前，从开展二手交换平台、维修和翻新服务，到定制化的产品再制造，这些创新举措推动了经济增长和创业潜力的发挥。

人们越来越重视闲置二手交换还反映了人们对消费文化的变革和可持续生活方式的认知提升。这种文化转变强调物品的再利用和价值，鼓励个人以更负责任的方式处理物品，这有助于减少浪费和塑造更加可持续的社会价值观。

闲置二手交换已经成为一个全球性的现象，各国之间可以通过合作和知识分享，共同解决资源有限性、环境挑战和消费问题。国际间的经验交流有助于制定更加合理的政策和策略，推动全球可持续发展。此外，相关法律和政策的缺失，如市场监管、知识产权、产品安全等问题不容忽视。如何完善相关法律和政策有助于更好地规范和管理闲置二手交换市场，确保公众权益和产品质量。

## 1.2 国内研究状况及分析

“现在年轻人对于闲置交易的接受程度以及绿色消费的意识越来越强，闲鱼用户的年轻化趋势明显，90后用户占比达50%。”近日，闲鱼负责人季山表示。

闲鱼是国内规模领先的线上闲置交易平台。第三方数据显示，2022年全国闲置交易月度活跃用户为1.45亿，其中，闲鱼平台的月度活跃用户为1.25亿。

清华大学能源环境经济研究所等机构发布的《2021中国闲置二手交易碳减排报告》预测，到2025年中国的闲置物品交易有望突破3万亿元。

推动发展方式绿色转型，完善支持绿色发展的政策，发展循环经济也被写入今年的《政府工作报告》中，成为2023年重点工作之一。

顺应我国循环经济发展趋势，人们的消费观念也潜移默化改变，“变浪费为消费”逐渐成为共识，尤其在年轻人中蔓延。

据季山介绍，闲鱼发展的初衷是建立一个人人可参与、低门槛的闲置资源交易平台，鼓励社会闲置资源的流通和用户间的互动。在闲鱼上，用户每年挂出的闲置物品超10亿件，几乎涵盖了消费品的全部品类。

“从用户认知的角度我们能看到，绿色消费的理念在年轻人里面接受度非常高的，同时我们也发现，闲置交易主力军由最初的一二线人群，逐步向三四线乃至五线城市下沉，”季山介绍“五线城市用户占比显著增长，超过30%。”

究其背后的原因，季山认为，随着绿色消费观念的深入，闲置交易的目的发生了一些变化，“对于年轻人来讲，闲置物品交易不仅仅是换钱，更多是一种潮流生活、时尚方式”。

季山以闲鱼为例介绍，2022年闲鱼平台全年碳减量预计将超过240万吨，每年减少的碳排放量相当于中国17.9万户家庭一年用电量的碳排放。

全国政协委员李学林认为，只有让个人减碳“有标准、可量化、可感知”，才能真正激发民众更为强劲的绿色消费意愿。因此建立闲置二手商品交易减碳标准体系成为未来趋势。

## 1.3 研究目标、内容和关键技术

(1) 研究目标

闲鱼以 C2C 的交易模式为主导，发展 C2X，多模式并存助力商品品类逐步完善，背靠淘宝流量不断提升社交属性，在C2C 模式中平台作为信息与交易的载体，个人卖家向个人买 家出售商品完成交易。采用 C2C 商业模式的大多是综合型平台，一般不参与用户之间交易流程，也不提供商品的质检和物流等服务，主要是为买卖双方提供交 易平台，保障交易顺利进行。研究闲鱼的交易模式对网站的各处设计是有意义的。

1. 研究内容

本系统将研究一个二手交换平台应有的基础功能，模拟闲鱼的业务流程，设计出一个具备基础信息交换的网站。同时为了用户体验更好，我还将研究每个功能的交互方式，具体是如何实现的，采用哪种实现方式更加合理，页面设计是否合理等内容。

1. 关键技术

本系统采用B/S模式设计系统，采用Maven构建，使用Java语言，主要使用的技术有Spring Boot、Mybatis-plus、MySQL、Vue、Element UI等。

## 1.4 术语说明

1、Java:是一种跨平台、面向对象的编程语言，以其可移植性、安全性和广泛应用而闻名。Java的代码通过Java虚拟机（JVM）在各种操作系统上运行，它支持多线程编程、自动内存管理和强大的标准库，广泛用于Web应用、移动应用、嵌入式系统和大型企业级应用的开发。

2、Spring Boot:Spring Boot是一个开源的Java应用程序框架，它旨在简化和加速Spring应用程序的开发和部署过程。Spring Boot的设计理念是"约定优于配置"，因此它提供了默认配置和自动配置，减少了开发人员在项目设置上的时间和精力。Spring Boot还引入了快速启动器（Starter）的概念，允许你轻松添加项目所需的依赖库，从而简化了构建过程。Spring Boot的另一个显著特点是内嵌式Web服务器支持，包括嵌入式的Tomcat、Jetty和Undertow，这意味着你可以将Web应用程序打包成一个独立的可执行JAR文件或WAR文件，无需外部应用服务器，从而简化了部署和维护。Spring Boot还提供了丰富的功能，如自动配置、自动扫描、健康检查、外部属性配置、配置文件分割、日志管理等，使开发人员能够专注于业务逻辑而不必担心配置细节。此外，Spring Boot具有强大的生态系统，提供了各种插件和扩展，可用于构建各种类型的应用，包括Web应用、RESTful服务、批处理应用和微服务架构。

3、Mybatis-plus:MyBatis-Plus（MyBatis-Ext）是一个开源的Java持久层框架，它构建在MyBatis之上，提供了丰富的便捷功能，包括自动生成SQL、条件构造器、分页查询、通用CRUD操作、自动填充、逻辑删除、枚举处理、乐观锁等，使数据库操作更加高效和容易，是Java开发中的强大工具。

4、MySQL:MySQL是一款流行的开源关系型数据库管理系统（RDBMS），以其高性能、可扩展性和稳定性而著名。它支持多种操作系统，提供了广泛的SQL功能，用于存储和管理数据，被广泛用于Web应用、企业应用和许多其他领域，是全球最受欢迎的开源数据库之一。

5、Vue:Vue.js（通常简称Vue）是一款流行的开源JavaScript前端框架，以其轻量、易学和灵活的特点而广受欢迎。Vue专注于构建现代、交互式的用户界面，其中核心特点包括响应式数据绑定，允许数据与DOM元素之间保持同步；组件化开发，以可重用的组件构建用户界面；虚拟DOM，提高性能并减少DOM操作开销；指令系统，用于操作DOM元素；路由管理，实现客户端端路由；状态管理，通过Vuex集中管理应用状态；以及生命周期钩子函数，允许开发者在不同组件生命周期阶段执行自定义逻辑。Vue还拥有强大的社区支持和丰富的生态系统，是构建现代Web应用的强大工具。

6、Element UI:Element UI（也称为Element）是一款基于Vue.js的流行前端UI框架，提供了丰富的精美UI组件、布局系统、表单元素、数据可视化组件等，使前端开发者能够更便捷地构建现代化、具有良好用户体验的Web应用程序。Element UI强调了可定制性和响应式设计，支持国际化、自定义主题、适应多种屏幕尺寸，同时拥有活跃的开发社区，为开发者提供了丰富的文档和示例，使其成为前端开发的重要工具，适用于各种项目规模和类型。

# 2 详细设计

## 2.1 用户功能设计

### 2.1.1 用户搜索物品功能

1、功能描述

用户可以根据物品名、物品类型关键字搜索相关的物品。

2、性能描述

搜索物品的时间限制为3秒。

3、输入

用户搜索物品功能的输入如表2.1所示。

**表2.1 用户搜索物品功能的输入**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 位于页面：src\views\User\home\index.vue | | | | |
| 输入项 | 长度 | 类型 | 输入方式 | 备注 |
| 关键字 | 20 | 字符串 | 人工输入 | 关键字 |

4、输出

用户搜索物品功能的输出如表2.2所示。

**表2.2 用户搜索物品功能的输出**

|  |  |  |  |
| --- | --- | --- | --- |
| 无异常时跳转至页面：不跳转 | | | |
| 输出项 | 长度 | 类型 | 备注 |
| 物品名 | 32 | 字符串 | 32个字符以内 |
| 物品类型 | 32 | 字符串 | 32个字符以内 |
| 物品描述 | 256 | 字符串 | 256个字符以内 |
| 期待价格 |  | FLOAT | 价格 |
| 想换物品 | 32 | 字符串 | 愿意交换的物品 |
| 图片链接 | 128 | 字符串 | 物品图片的url |

5、流程图

用户搜索物品的流程图如图2.1所示。

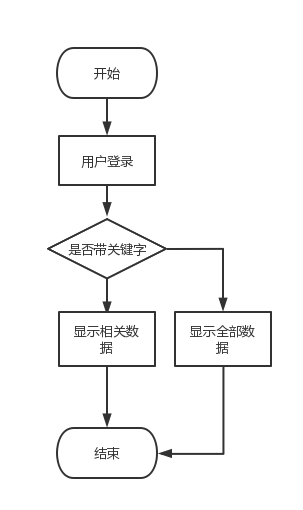


图2.1 用户搜索物品功能的流程图

### 2.1.2 用户留言功能

1、功能描述

用户可以点击按钮输入留言信息后确认发送，卖家能看到你的留言和联系方式。

2、输入

用户留言功能的输入如表2.3所示。

**表2.3 用户留言功能的输入**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 位于页面：src\views\User\home\index.vue | | | | |
| 输入项 | 长度 | 类型 | 输入方式 | 备注 |
| 留言 | 256 | 字符串 | 人工输入 | 留言信息 |

3、流程图

用户留言的流程图如图2.2所示。

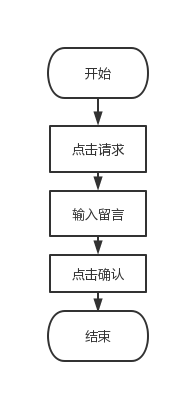


图2.2 用户留言功能的流程图

### 2.1.3 用户修改个人信息功能

1、功能描述

用户可以修改个人信息。

2、输入

用户修改个人信息功能的输入如表2.4所示。

**表2.4 用户修改个人信息功能的输入**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 位于页面：src\views\User\person\homepage.vue | | | | |
| 输入项 | 长度 | 类型 | 输入方式 | 备注 |
| 用户名 | 32 | 字符串 | 人工输入 | 用户名，不为空 |
| 密码 | 32 | 字符串 | 人工输入 | 密码，不为空 |
| 邮箱 | 32 | 字符串 | 人工输入 | 邮箱，不为空 |
| QQ | 32 | 字符串 | 人工输入 | QQ账号 |
| 微信 | 32 | 字符串 | 人工输入 | 微信账号 |
| 电话 | 32 | 字符串 | 人工输入 | 电话号码 |
| 个人描述 | 256 | 字符串 | 个人输入 | 个人描述 |

3、流程图

用户修改个人信息的流程图如图2.3所示。

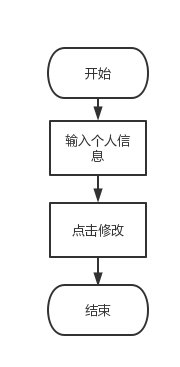


图2.3 用户修改个人信息功能的流程图

### 2.1.4 用户添加物品功能

1、功能描述

用户可以添加物品。

2、输入

用户添加物品功能的输入如表2.5所示。

**表2.5 用户添加物品功能的输入**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 位于页面：src\views\User\product\index.vue | | | | |
| 输入项 | 长度 | 类型 | 输入方式 | 备注 |
| 物品名 | 32 | 字符串 | 人工输入 | 物品名 |
| 物品类型 | 32 | 字符串 | 人工输入 | 物品类型 |
| 物品描述 | 256 | 字符串 | 人工输入 | 物品描述 |
| 目标价格 |  | FLOAT | 人工输入 | 接受的价格 |
| 目标物品 | 32 | 字符串 | 人工输入 | 愿意交换的物品 |

3、流程图

用户添加物品的流程图如图2.4所示。

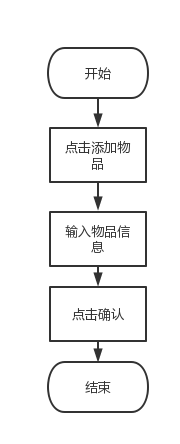


图2.4 用户添加物品功能的流程图

### 2.1.5 用户修改物品信息功能

1、功能描述

用户可以修改物品的信息。

2、输入

用户修改物品信息功能的输入如表2.6所示。

**表2.6 用户修改物品信息功能的输入**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 位于页面：src\views\User\product\index.vue | | | | |
| 输入项 | 长度 | 类型 | 输入方式 | 备注 |
| 物品名 | 32 | 字符串 | 人工输入 | 物品名 |
| 物品类型 | 32 | 字符串 | 人工输入 | 物品类型 |
| 物品描述 | 256 | 字符串 | 人工输入 | 物品描述 |
| 目标价格 |  | FLOAT | 人工输入 | 接受的价格 |
| 目标物品 | 32 | 字符串 | 人工输入 | 愿意交换的物品 |
| 物品状态 |  | SAMLLINT | 人工输入 | 物品上架0，物品下架1 |

3、流程图

用户修改物品信息的流程图如图2.5所示。

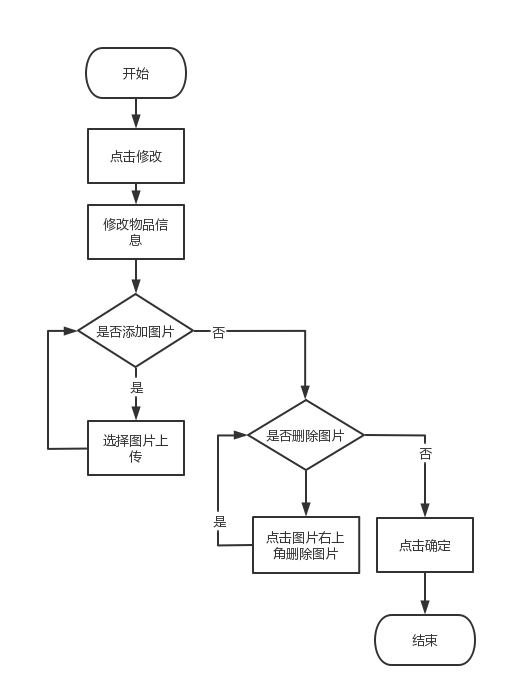


图2.5 用户修改物品信息功能的流程图

### 2.1.6 用户删除物品功能

1、功能描述

用户可以删除物品。

1. 流程图

用户删除物品的流程图如图2.6所示。

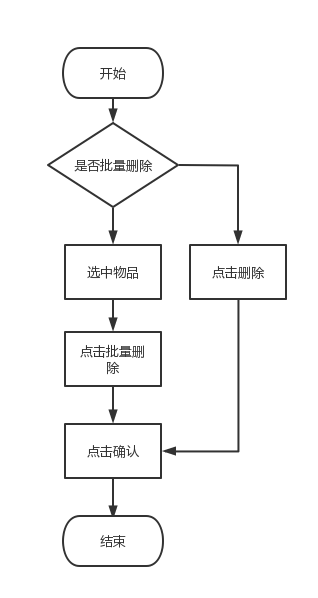


图2.6 用户删除物品功能的流程图

## 2.2 管理员功能设计

### 2.2.1 管理员审核用户信息功能

1、功能描述

管理员可以对用户修改的个人信息进行审核。

2、流程图

管理员审核用户信息功能的流程图如图2.7所示。

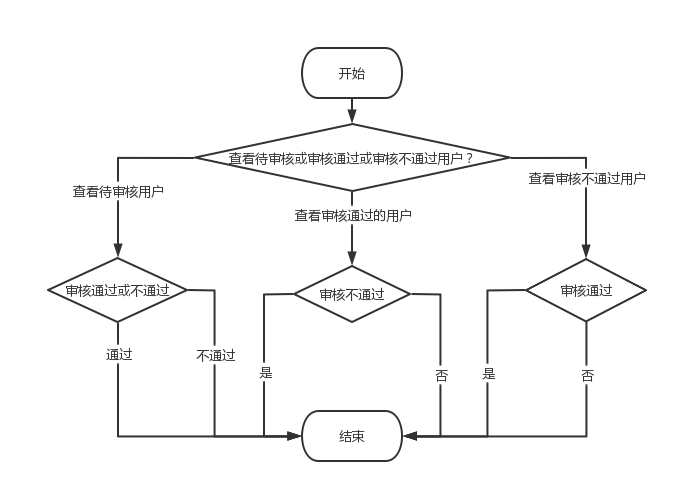


图2.7 管理员审核用户信息功能的流程图

### 2.2.2 管理员审核物品功能

1、功能描述

管理员可以对物品进行审核。

2、流程图

管理员审核物品功能的流程图如图2.8所示。

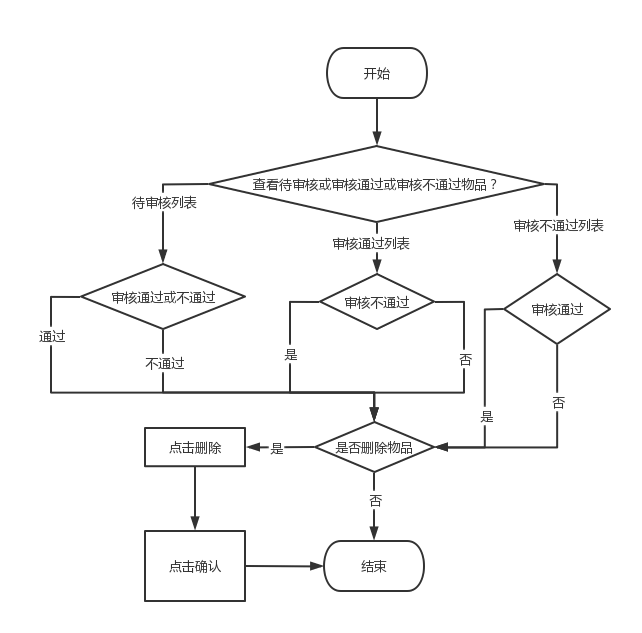


图2.8 管理员审核物品功能的流程图

# 3 系统实现

## 3.1系统配置

### 3.1.1 数据库实现

共有五个表，物品表(product)，用户表(customer)，照片表(photo)， 管理员表(admin)，申请表(trade)，其中用户表和物品表以外键连接，如图3.1所示。

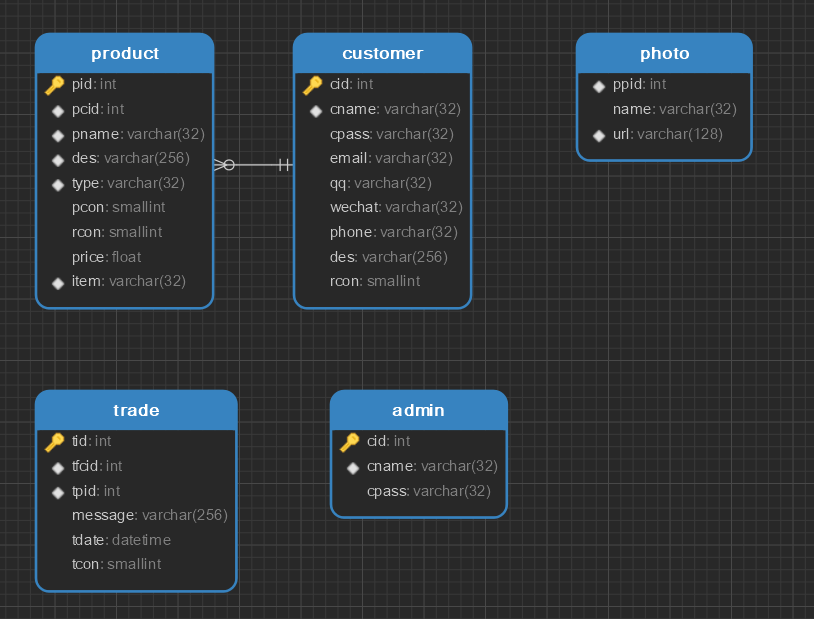


图3.1 数据库实现图

### 3.1.2 前端实现

1.在main.js里面引入ElementUI库，并设置反向代理，同时编写一个用于认证的钩子函数，如图3.2所示。

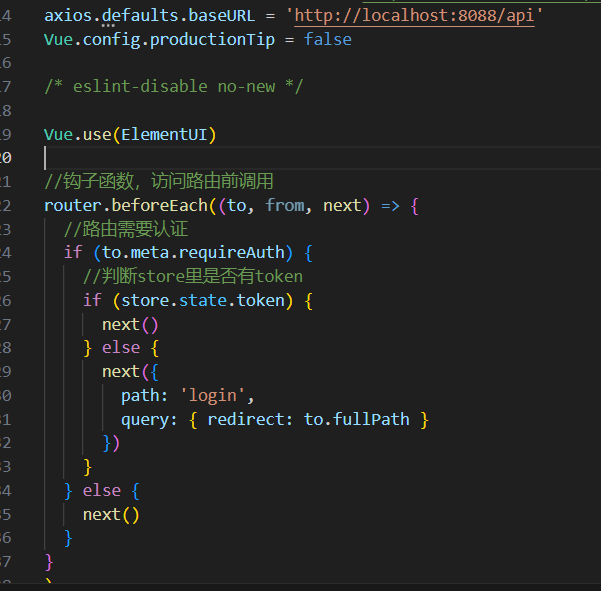


图3.2 main.js

1. 编辑store/index.js，将从后台获取到的token存储到state中，这样通过state.token就能获取到token添加到每一次请求的请求头中，如图3.3所示。



图3.3 store/index.js

1. 编辑utils/request.js，对每一次请求进行拦截，从会话中获取token添加到请求头中，如图3.4所示。

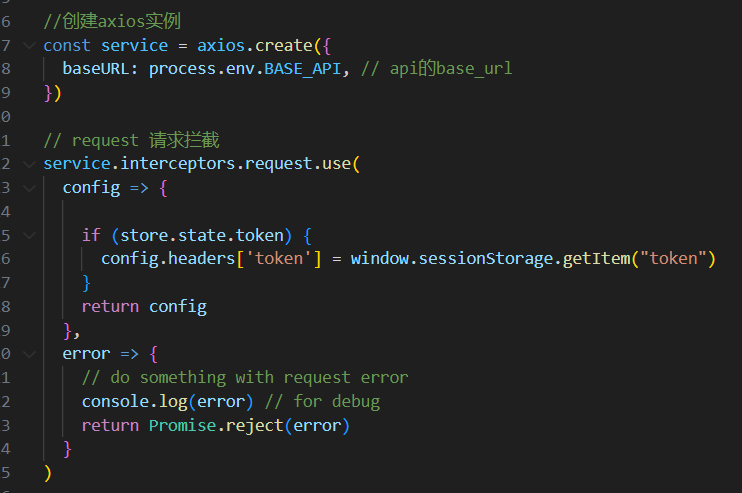


图3.4 utils/request.js

1. 通过导入utils/request，即可创建对后台进行请求的方法，这些方法被包含在api目录里面。

### 3.1.3 后端实现

1. 设置图片的保存目录以及uri，如图3.5所示。



图3.5 application.properties

1. 创建一个controller返回图片数据，如图3.6所示。

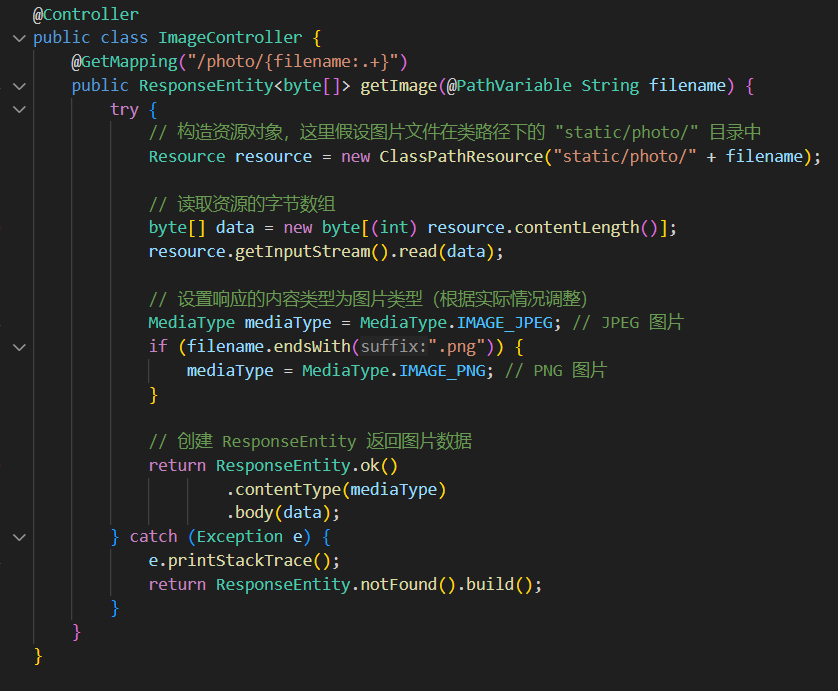


图3.6 ImageController.java

## 3.2 用户功能实现

### 3.2.1 用户搜索商品功能

1、功能实现思路

用户输入关键字后点击发送后,通过api/search中的productList方法从后台获得相应数据在页面进行渲染。

2、相关类，方法，属性

SearchService.selectProductPage(QueryDTO queryDTO)

3、核心代码

    public IPage<Item> selectProductPage(QueryDTO queryDTO) {

        String keyword = queryDTO.getKeyword();

        Integer pcid = queryDTO.getPcid();

        Integer pageNo = queryDTO.getPageNo();

        Integer pageSize = queryDTO.getPageSize();

        IPage<Item> page = new Page<>(pageNo, pageSize);

        // 查询记录总数

        int total = searchMapper.countProducts(keyword,pcid);

        // 设置总记录数

        page.setTotal(total);

        // 查询数据

        List<Item> itemList = new ArrayList<>();

        for(Integer pid:searchMapper.PidList(keyword,pcid,pageSize,(pageNo-1)\*pageSize))

        {

            Item i = new Item();

            ItemInfo itemInfo = searchMapper.getItemInfo(pid);

            List<String> srcList = searchMapper.getSrcList(pid);

            Image image = new Image();

            if(srcList!=null&&srcList.size()!=0){

                image.setUrl(srcList.get(0));

                image.setSrcList(srcList.toArray(new String[0]));

            }

            i.setItemInfo(itemInfo);

            i.setImage(image);

            itemList.add(i);

        }

        // 设置查询结果

        page.setRecords(itemList);

        return page;

    }

其中物品的基本信息被封装在ItemInfo中，物品的图片数据被封装在Image中，Item条目包含ItemInfo和Image。

4、界面

用户搜索物品界面如图3.7所示。



图3.7 用户搜索物品界面

### 3.2.2 用户修改个人信息功能

1、功能实现思路

用户修改个人信息后，点击修改，修改后的数据就会提交给后台对customer表进行更新，修改后的信息就会提交给审核，能在个人信息页面看到审核状态。

2、相关类，方法，属性

CustomerService.CustomerModify(Customer customer)

3、核心代码

 public Result CustomerModify(Customer customer) {

        //用户名查重

        QueryWrapper<Customer> wrapper = new QueryWrapper<>();

        wrapper.eq("cname", customer.getCname());

        int count=customerMapper.selectCount(wrapper);

        if(count>1) return new Result(400, "用户名已存在", null);

        int update = customerMapper.updateById(customer);

        if(update>0) return new Result(200, "上传成功", customer);

        return new Result(400, "上传失败", null);

    }

更新信息前会对用户名进行查重。

4、界面

用户修改个人信息界面如图3.8所示。

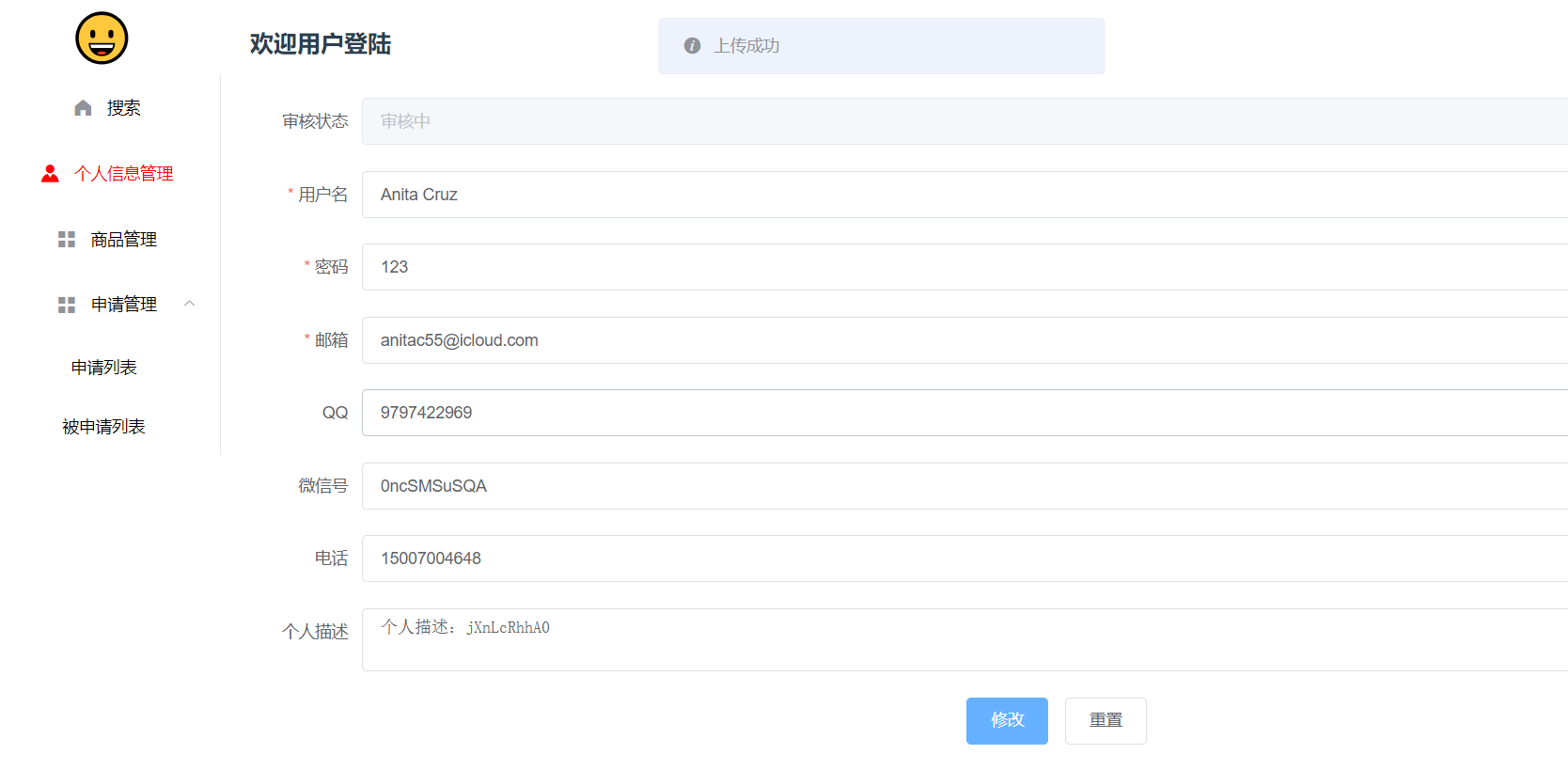


图3.8 用户修改个人信息界面

### 3.2.3 用户商品管理功能

1、功能实现思路

用户点击添加物品，输入物品信息确定后，对后台发送请求，插入数据到product表，点击修改按钮，对物品基本信息进行修改点击确定后，对后台发送请求，更新product表，点击添加图片或图片右上角删除图片，都会对后台发送请求，更新photo表。

2、相关类，方法，属性

对product表的操作：

ProductService.selectProductPage(QueryDTO queryDTO)

ProductService.addProduct(Product product)

ProductService.updateProduct(Product product)

ProductService.deleteProduct(Integer id)

ProductService.batchDelete(List<Integer> ids)

对photo表的操作：

PhotoService.photoList(Integer id)

PhotoService.uploadPhoto(MultipartFile file,String ppid)

PhotoService.deletePhoto(String url)

3、核心代码

    public String uploadPhoto(MultipartFile file,String ppid) throws IllegalStateException, IOException {

        String originalFilename = file.getOriginalFilename();

        if(originalFilename==null) return null;

        int lastIndex = originalFilename.lastIndexOf(".");

        String type = originalFilename.substring(lastIndex + 1);

        String uuid = UUID.randomUUID().toString().substring(0, 8);

        String filePath = uploadPath + uuid + "." + type;

        String url = uploadUri + uuid + "." + type;

        System.out.println(filePath);

        File dest = new File(filePath);

        file.transferTo(dest);

        photoMapper.insertPhoto(Integer.parseInt(ppid), originalFilename, url);

        productMapper.updateRcon(Integer.parseInt(ppid));

        return url;

    }

这是上传图片的代码片段，其中uploadPath是从配置文件获取的文件上传路径，uploadUri是从配置文件获取的uri。

4、界面

用户添加物品界面如图3.9所示，修改物品信息如图3.10所示。



图3.9 用户添加物品界面

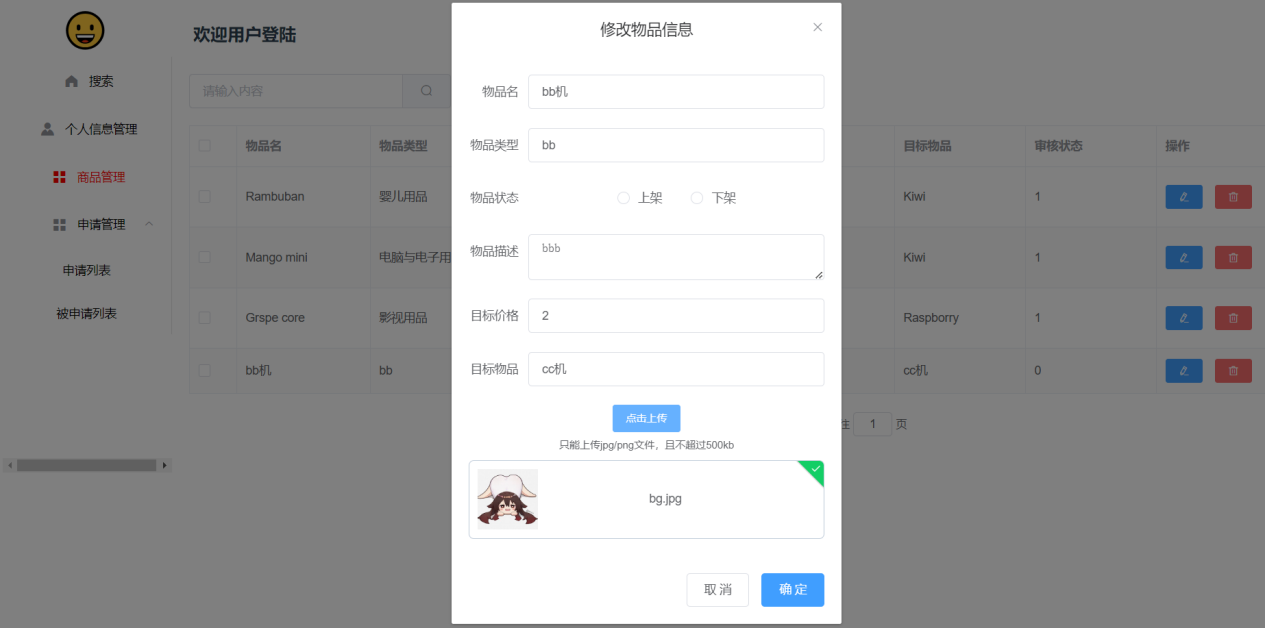


图3.10 用户修改物品信息界面

### 3.2.4 用户留言功能

1、功能实现思路

用户点击搜索页面对应物品的发送留言按钮，即可输入留言后点击确认发送，对后台发送请求，插入数据到trade表，然后我们可以在请求列表里看到发送的请求。

2、相关类，方法，属性

//插入留言到trade表中

TradeService.insertTrade(applyDTO applyDTO)

//查询收到的留言

TradeService.selectAppliedPage(QueryDTO queryDTO)

//查询发送的留言

TradeService.selectApplyPage(QueryDTO queryDTO)

//拒绝接受的留言

TradeService.deleteTrade(Integer id)

//批量拒绝

TradeService.batchDelete(List<Integer> ids)

3、核心代码

    <select id="selectAppliedPage" resultMap="ApplyMap">

        SELECT tid,cname,tpid,pname,message,

        CONCAT("邮箱：",email,"\_qq：",qq,"\_微信：",wechat,"\_电话：",phone) as contact

            from trade

            left join product on trade.tpid=product.pid

            left join customer on trade.tfcid = customer.cid

            where tpid in

                (SELECT pid from product where pcid = #{pcid})

                and trade.tcon=0

                <if test="keyword != null and keyword != ''">

                    AND pname LIKE CONCAT('%', #{keyword}, '%')

                </if>

            LIMIT #{pageSize} OFFSET #{num}

    </select>

查询收到的留言。

4、界面

用户留言界面如图3.11所示，用户发送留言查看界面如图3.12所示。



图3.11 用户留言界面



图3.12 用户发送留言查看界面

## 3.3 管理员功能实现

### 3.3.1 用户信息审核功能

1、功能实现思路

用户修改信息后，对后台发送请求，将状态码更新为0，待审核，管理员对待审核用户信息进行审核，管理员还可以查询审核通过或审核不通过的用户并更改状态。

2、相关类，方法，属性

//查询待审核用户，审核通过用户，审核不通过用户

AdminService.getCustomerPage(Map<String, String> map)

//更改审核状态

AdminService.review(Map<String, String> map)

3、核心代码

    public Integer review(Map<String, String> map) {

        Integer id = Integer.parseInt(map.get("id"));

        Integer rcon = Integer.parseInt(map.get("rcon"));

        String tableName = map.get("tableName");

        Integer num =0;

        switch(tableName){

            case "customer" : num = adminmapper.updateCustomerRcon(id, rcon);

            case "product" : num = adminmapper.updateProductRcon(id, rcon);

        }

        return num;

    }

根据tableName更新对应编号的用户或物品的状态。

4、界面

用户信息审核界面如图3.13所示。



图3.13 用户信息审核界面

### 3.3.2 物品审核功能

1、功能实现思路

管理员可以查询待审核，审核通过或审核不通过的物品，更改其状态，还可以删除物品。

2、相关类，方法，属性

AdminService.getProductPage(Map<String, String> map)

AdminService.review(Map<String, String> map)

AdminService.delete(Map<String, String> map)

3、核心代码

    public void delete(Map<String, String> map) {

        Integer id = Integer.parseInt(map.get("id"));

        adminmapper.deletePTrade(id);

        adminmapper.deleteProduct(id);

    }

删除物品的时候对应的申请也会被删除。

4、界面

物品审核界面如图3.14所示，物品删除界面如图3.15所示。

 图3.14 物品审核界面

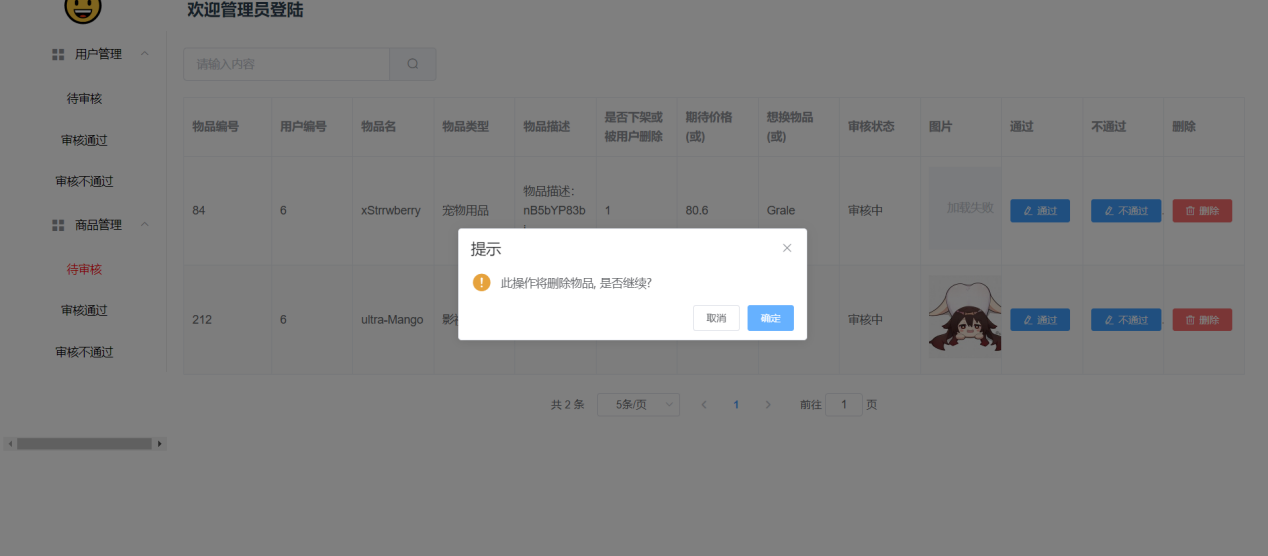


图3.15 物品删除界面

# 4 系统测试

## 4.1 单元测试

### 4.1.1 login方法

等价类表如表4.1所示，测试用例设计表如表4.2所示。

**表4.1 等价类表**

|  |  |  |
| --- | --- | --- |
| 输入条件 | 有效等价类 | 无效等价类 |
| cname | 1.0<长度<=32 | 3.长度<=0。4.长度>32 |
| cpass | 2.0<长度<=32 | 5.长度<=0。6.长度>32 |

**表4.2 测试用例设计表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 测试用例 | | | | 测试记录 |
| 用例编号 | 测试类型 | 覆盖等价类 | 输入数据/预置条件（按序输入） | 预期结果 | 结果 |
|  | 有效等价类 | 1 2 | “Joyce Kennedy”,”123” | 登陆成功 | 登陆成功 |
|  | 无效等价类 | 15 | ”Joyce Kennedy”,”” | 登陆失败 | 登陆失败 |
|  | 无效等价类 | 23 | “”,”123” | 登陆失败 | 登陆失败 |
|  | 无效等价类 | 16 | “Joyce Kennedy”,33个字符 | 登陆失败 | 登陆失败 |
|  | 无效等价类 | 24 | 33个字符,”123” | 登陆失败 | 登陆失败 |

## 4.2 集成测试

### 4.2.1 登陆功能和首页展示

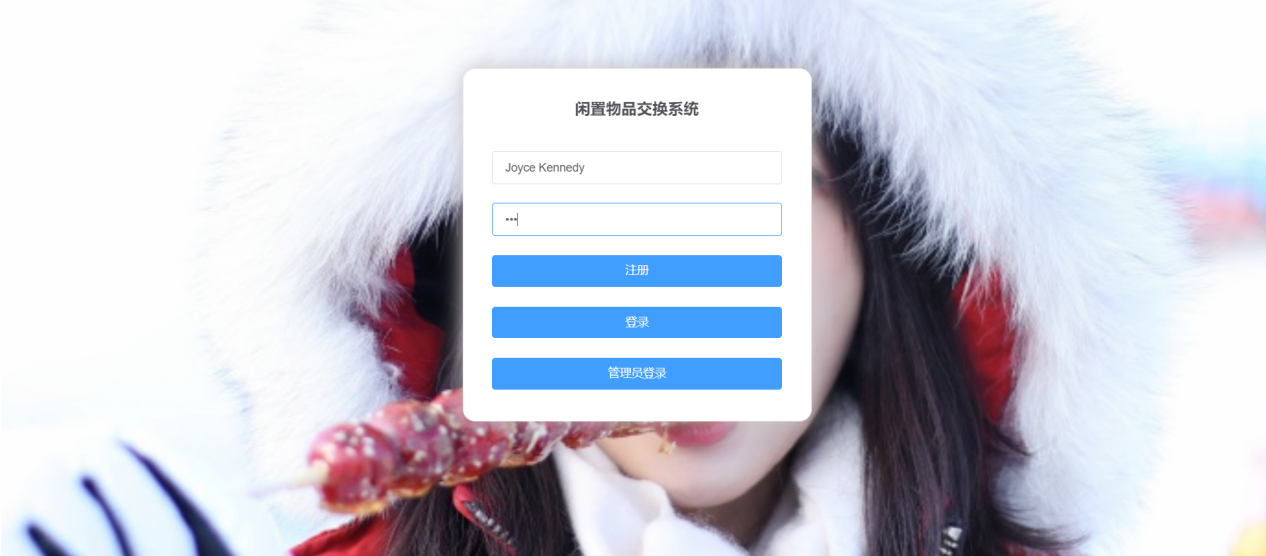


图4.1 登陆界面



图4.2 登陆成功进入主界面

### 4.2.2 物品搜索功能

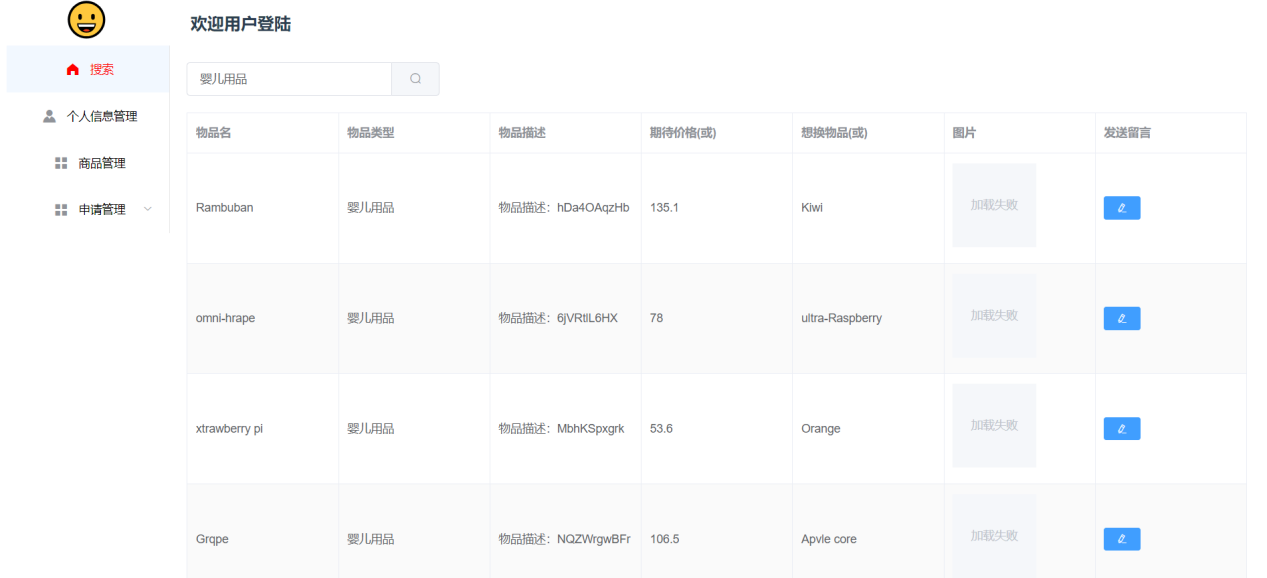


图4.3 搜索物品界面

### 4.2.3 个人信息修改功能

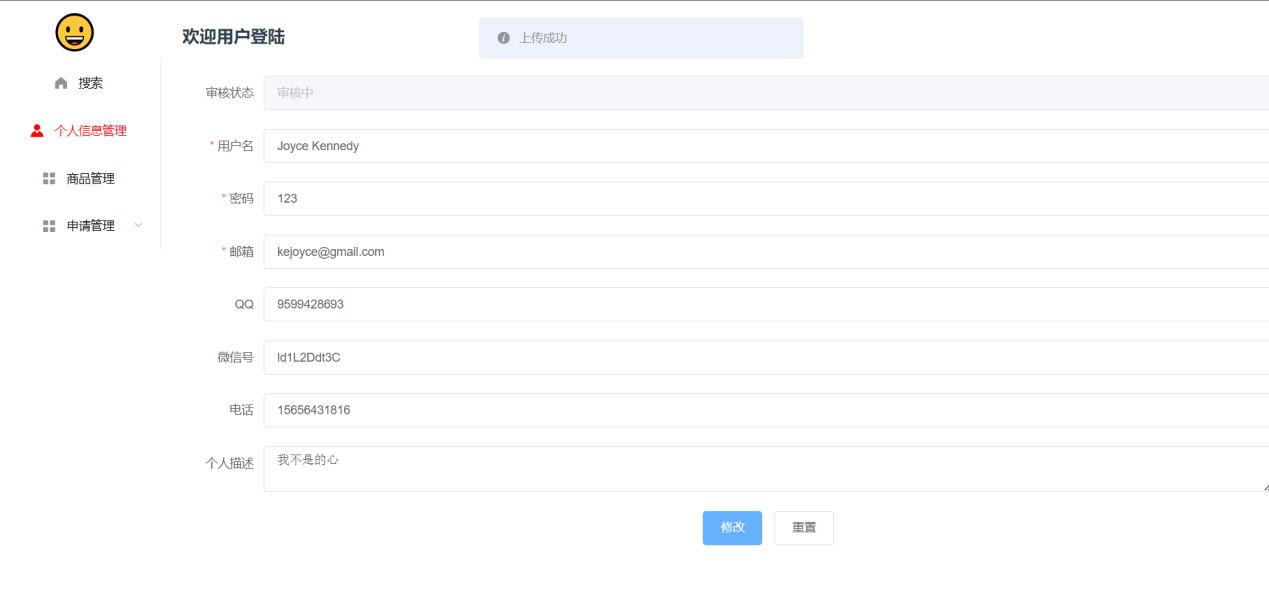


图4.4 个人信息修改界面

### 4.2.4 添加商品功能

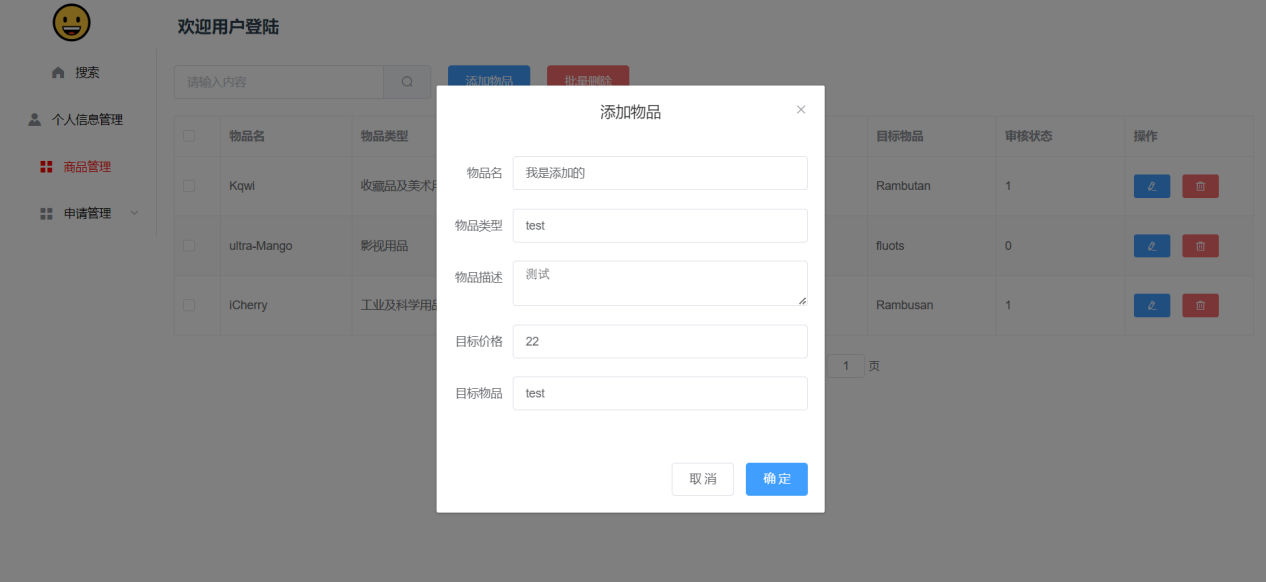


图4.5 商品信息输入界面



图4.5 商品添加成功界面

### 4.2.5 商品信息修改功能



图4.5 商品信息修改界面



图4.6 商品信息修改成功界面

### 4.2.6 商品删除功能

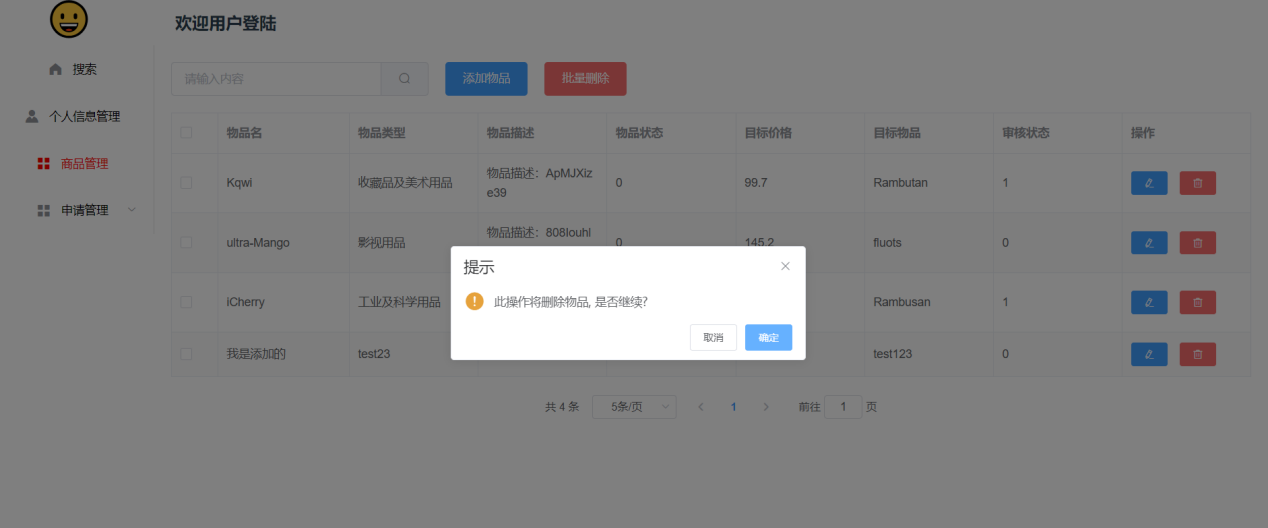


图4.7 商品删除界面



图4.8 商品删除成功界面

### 4.2.7 申请查询功能



图4.9 申请查询界面

### 4.2.8 处理接收申请功能



图4.10 处理接收申请界面

### 4.2.9 用户信息审核功能

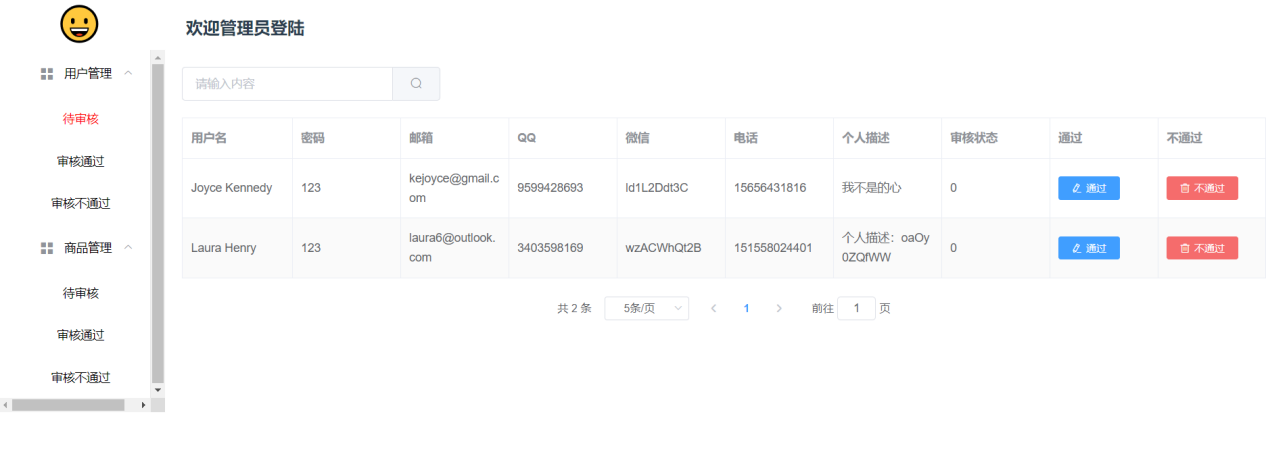


图4.11 用户信息审核界面

### 4.2.10 用户信息审核功能



图4.12 商品审核界面

# 总结体会

在本项目中，我独立设计并实现了一个基于Vue.js、Element UI、Spring Boot和MyBatis-Plus的闲置物品交换系统。这个项目不仅让我学会应用了多个技术栈，还得到了宝贵的经验和教训。

这个项目的初衷是创建一个方便用户交换不再需要的物品的平台，以减少浪费和促进可持续消费。在项目启动初期，我花了大量的时间搜集信息，以明确用户的需求和期望。这个过程中，我了解到用户需要一个简单易用的界面，能够快速找到感兴趣的物品，同时也需要一个安全的平台来进行物品交换。

这个项目要求我掌握多个技术栈，包括前端和后端技术。对于前端，我选择了Vue.js和Element UI来构建用户界面，这是我第一次使用这两个技术。通过阅读文档、教程和在线课程，我逐渐掌握了Vue.js的基本概念和Element UI的使用方法。对于后端，我采用Spring Boot和MyBatis-Plus来处理数据和业务逻辑，这需要我深入了解Java和数据库操作。这个技术学习的过程让我感到充实，也锻炼了自学和解决问题的能力。

虽然这是一个单人项目，但在整个过程中，我需要与老师和同学进行沟通和合作。我定期向老师汇报项目进展，接受老师的指导和建议。与同学的互动也让我受益匪浅，我们分享了关于技术、设计的经验。

测试是确保项目质量的关键一环。我进行了大量的单元测试、集成测试和端到端测试，以确保系统的稳定性。同时，我也学会了如何调试和排查问题，这是一个开发者必备的技能。通过不断地测试和调试，我能够确保系统平稳运行。

我积极寻求老师和同学的反馈，以便进一步改进系统。他们提供了宝贵的意见，帮助我发现和解决问题。这个过程让我明白了接受他人意见不断改进项目的重要性。

通过设计并实现闲置物品交换系统，我收获了丰富的经验和知识。这个项目不仅锻炼了我的技术能力，还提高了我的自学和解决问题的能力。我明白了软件开发的复杂性和团队合作的价值。在未来，我将继续应用这些经验，参与更多的项目，并不断学习和成长。这次课程设计也让我明白，只要有梦想和坚持，就可以创造出令人自豪的成果。

参考文献

1. 中国互联网络信息中心.第47次中国互联网络发展现状统计报告[R/OL].(2021年2月)[2021年10月28日]. http://www.cac.gov.cn/2021-02/03/c\_1613923423079314.htm.
2. （美）盖茨 著；童云兰等 译.《Java并发编程实战》.-北京:机械工业出版社
3. 许令波 著.《深入分析JavaWeb技术内幕》.-北京:电子工业出版社
4. （美）布洛克 著；杨春花，俞黎敏 译.《Effective Java》.-北京:机械工业出版社
5. （美）Craig Walls 著；张卫滨 译.《Spring实战》.-北京:人民邮电出版社
6. 姜承尧 著. 《MySQL技术内幕 InnoDB存储引擎(第2版)》.-北京:机械工业出版社(1-346)
7. （美）卡尔森 著；黄健宏 译.《Redis实战》.-北京:人民邮电出版社