

**本科毕业设计（论文）**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于Spring Boot的物品共享平台 | | |
|  | "ShareHub"的设计与实现 | | |
| 学 院 | 计算机学院 | | |
| 专 业 | 软件工程 | | |
| 学生姓名 | 燕子豪 | | |
| 学 号 | 202010111121 | 班级 | 2020级1班 |
| 指导教师 | 聂莉莎/郝明明 | 职称 | 讲师/高级工程师 |
| 完成时间 | 2024年 4月 15日 | | |

原创性声明

本人郑重声明：本人所呈交的毕业设计（论文），是在指导老师的指导下独立进行研究所取得的成果。毕业设计（论文）中凡引用他人已经发表或未发表的成果、数据、观点等，均已明确注明出处。除文中已经注明引用的内容外，不包含任何其他个人或集体已经发表或撰写过的科研成果。对本文的研究成果做出重要贡献的个人和集体，均已在文中以明确方式标明。

本声明的法律责任由本人承担。

论文作者签名： 日 期：2024 年 4 月15日

关于使用授权的声明

本人在指导老师指导下所完成的毕业设计（论文）及相关的资料（包括图纸、试验记录、原始数据、实物照片、图片、录音带、设计手稿等），知识产权归属成都大学。本人完全了解成都大学有关保存、使用毕业设计（论文）的规定，本人授权成都大学可以将本毕业设计（论文）的全部或部分内容编入有关数据库进行检索，可以采用任何复制手段保存和汇编本毕业设计（论文）。如果发表相关成果，一定征得指导教师同意，且第一署名单位为成都大学。本人离校后使用毕业设计（论文）或与该论文直接相关的学术论文或成果时，第一署名单位仍然为成都大学。

论文作者签名： 日 期：2024 年 4 月15日

指导教师签名： 日 期：2024 年 4 月15日

基于Spring Boot的物品共享平台"ShareHub"的设计与实现

专业：软件工程 学 号：202010111121

学生：燕子豪 指导教师：聂莉莎/郝明明

摘要：当下共享经济持续增长，社会发展呈现出数字化、智能化的趋势，人们的生活方式和资源利用模式也在不断演变。相对而言，物品共享这一模式为居民提供了更加便捷、绿色的资源利用方式。在这一背景下，物品共享平台“ShareHub”应运而生。该平台的建立的有助于减少社会资源的浪费，用户能够方便地共享、交换和购买物品，实现物品的充分利用。

在技术方面,作为一个多端跨平台项目，后端使用的Spring Boot框架来实现项目的基本功能，小程序端是用户主要的体验平台，基于uniapp的Color-UI框架实现，Web端是管理员的使用平台，基于Vue2的Element-UI框架实现。在数据存储方面，采用MySQL 数据库进行关系型数据库管理，以及阿里云的对象存储OSS存储和管理图片信息。在数据安全方面，使用了MD5算法进行数据库数据的加密，同时使用了 SSL 协议保证了网络的安全通信。在个性化推荐方面，使用了经典的基于物品内容的推荐算法，为用户进行恰当的物品推荐。该系统实现了合理的物品共享和交易流程，为用户实现了便利的物品共享平台。

关键词：物品共享；跨平台；Spring Boot；Vue

**Design and implementation of 'ShareHub ', an item sharing platform based on Spring Boot**

Major：Software Engineering. Student ID：202010111121

Student：Yan Zihao Instructor：Nie Lisha/Hao Mingming

**Abstract:** At present, the sharing economy continues to grow, social development shows a trend of digitization and intelligence, and people 's lifestyles and resource utilization patterns are constantly evolving. In contrast, the mode of item sharing provides residents with a more convenient and green way to use resources. In this context, the item sharing platform ' ShareHub ' came into being. The establishment of the platform helps to reduce the waste of social resources, users can easily share, exchange and purchase items, to achieve full utilization of items.

In terms of technology, as a multi-terminal cross-platform project, the back-end uses the Spring Boot framework to realize the basic functions of the project. The applet side is the user 's main experience platform, based on the Color-UI framework of uniapp. The Web side is the administrator 's use platform, based on the Element-UI framework of Vue2. In terms of data storage, MySQL database is used for relational database management, and Alibaba Cloud 's object storage OSS stores and manages image information. In terms of data security, the MD5 algorithm is used to encrypt the database data, and the SSL protocol is used to ensure the secure communication of the network. In terms of personalized recommendation, the classic item content-based recommendation algorithm is used to recommend items for users. The system realizes a reasonable item sharing and transaction process, and realizes a convenient item sharing platform for users.

**Key words：**goods sharing; cross-platform; Spring Boot; Vue

目 录

[1 绪论 1](#_Toc164169239)

[1.1 研究背景和意义 1](#_Toc164169240)

[1.2 国外研究现状 2](#_Toc164169241)

[1.3 国内研究现状 2](#_Toc164169242)

[1.4 研究内容 3](#_Toc164169243)

[2 系统开发环境 4](#_Toc164169244)

[2.1 后端开发环境 4](#_Toc164169245)

[2.1.1 IntelliJ IDEA 2021.1.3 4](#_Toc164169246)

[2.1.2 Spring Boot 5](#_Toc164169247)

[2.1.3 Git 5](#_Toc164169248)

[2.2 Web端开发环境 6](#_Toc164169249)

[2.2.1 Visual Studio Code 6](#_Toc164169250)

[2.2.2 Vue 7](#_Toc164169251)

[2.2.3 ECharts 7](#_Toc164169252)

[2.3 移动端开发环境 8](#_Toc164169253)

[2.3.1 HBuilder X 8](#_Toc164169254)

[2.3.2 Uni-app 8](#_Toc164169255)

[2.4 本章小结 8](#_Toc164169256)

[3 系统需求分析 9](#_Toc164169257)

[3.1 系统概述 9](#_Toc164169258)

[3.2 功能需求分析 9](#_Toc164169259)

[3.3 非功能需求分析 12](#_Toc164169260)

[3.3.1 市场需求分析 12](#_Toc164169261)

[3.3.2 安全需求分析 16](#_Toc164169262)

[3.3.3 性能需求分析 16](#_Toc164169263)

[3.4 本章小结 16](#_Toc164169264)

[4 系统设计 17](#_Toc164169265)

[4.1 系统架构设计 17](#_Toc164169266)

[4.1.1 总体架构设计 17](#_Toc164169267)

[4.1.2 逻辑架构设计 18](#_Toc164169268)

[4.1.3 物理架构设计 19](#_Toc164169269)

[4.2 功能模块设计 20](#_Toc164169270)

[4.2.1 Web端功能模块设计 20](#_Toc164169271)

[4.2.2 小程序端功能模块设计 21](#_Toc164169272)

[4.3 非功能模块设计 22](#_Toc164169273)

[4.3.1 安全设计 22](#_Toc164169274)

[4.3.2 性能保障设计 22](#_Toc164169275)

[4.3.3 界面设计 23](#_Toc164169276)

[4.4 本章小结 23](#_Toc164169277)

[5 数据库设计 24](#_Toc164169278)

[5.1 概念模型设计 24](#_Toc164169279)

[5.2 物理模型设计 25](#_Toc164169280)

[5.3 本章小结 28](#_Toc164169281)

[6 系统实现 29](#_Toc164169282)

[6.1 系统架构实现 29](#_Toc164169283)

[6.1.1 Web端多层架构实现 29](#_Toc164169284)

[6.1.2 小程序端架构实现 30](#_Toc164169285)

[6.2 功能模块实现 31](#_Toc164169286)

[6.2.1 Web端功能实现 31](#_Toc164169287)

[6.2.2 小程序端功能实现 34](#_Toc164169288)

[6.3 关键算法 40](#_Toc164169289)

[6.4 本章小结 42](#_Toc164169290)

[7 系统测试 44](#_Toc164169291)

[7.1.1 测试目的 44](#_Toc164169292)

[7.2 测试方法 44](#_Toc164169293)

[7.2.1 功能测试 44](#_Toc164169294)

[7.2.2 兼容性测试 47](#_Toc164169295)

[7.3 测试结论 48](#_Toc164169296)

[7.4 本章小结 48](#_Toc164169297)

[结 论 49](#_Toc164169298)

[参考文献 51](#_Toc164169299)

[致 谢 52](#_Toc164169300)

# 绪论

## 研究背景和意义

研究背景

随着社会科技的发展和经济结构的变革，共享经济作为一种新型的经济模式已经逐渐兴起。早在1978年，美国的学者就提出了共享经济的概念，共享经济被定义为“一个以社区为基础的互联网平台，促进产品和服务的对等共享”[1]。但直到最近几年，随着移动互联技术的快速发展，人们才真正开始利用过剩产能进行资源共享。共享经济是一种新兴的经济模式，它不仅改变了人们的生活方式，还在经济生活中带来了全新的组织方式和价值观。这种模式通过个人或机构将闲置资源或服务共享给需要的人，并从中获取回报。这种共享方式不仅让资源得到更有效的利用，还创造了新的经济价值，超越了传统的市场模式。

通过共享资源、信息和技术，共享经济促进了资源的有效利用和社会资源的共享，成为了全球经济的热点之一。共享经济包括三个要素，一是闲置资源，这些资源通过网络平台技术整合，并转化为可供交易的产品或服务。二是共享网络平台，它是共享经济的核心，通过网络技术分配和推广未被充分利用的资源。三是参与者群体，他们各自根据需求对共享经济进行创新，实现个性化和定制化服务[2]。共享模式涉及到的技术和服务不断创新，如移动支付、大数据分析、人工智能等，推动了互联网和科技行业的进步。共享经济也催生了新的商业模式和商业生态，促使传统行业进行转型升级，提高了市场竞争力和企业创新能力。根据普华永道的数据， 2014年全球共享经济规模已经达到150亿美元，并预计未来年均增长率将达到35%，到2025年全球规模将达到3350亿美元。在中国，共享经济市场规模更是惊人，2015年已经达到1.95万亿元，预计未来5年年均增长速度将高达40%，到2020年，共享经济规模将占GDP比重的10%以上[3]。这表明共享经济将成为中国经济的新的增长点，对经济结构的变革和发展具有重要意义。因此，共享经济不仅是一种经济模式，更是一种社会发展的新趋势，值得深入探讨和研究。

物品共享平台作为共享经济的一个重要分支，在这一大潮中也崭露头角，扮演着举足轻重的角色。

研究目的及意义

本研究的目的在于设计并实现基于Spring Boot的物品共享平台"ShareHub"，通过深入研究该平台的功能特点、技术架构和用户体验，探索共享经济模式在物品共享领域的应用与发展。该平台的优点在于其高效便捷的设计，充分利用了Spring Boot框架的特性，构建了一个稳定、高性能的共享平台，提升了资源共享的效率和便利性。此外，"ShareHub"平台还优化了用户体验和服务质量，通过深入了解用户需求，设计了三种交易方式，包括免费共享、以物换物，以及二手交易，可充分满足用户对各类限制物品的不同交易需求。同时完成了界面设计优化以及流程优化，以及增加了物品推荐算法等，提升了用户在平台上的体验感和满意度。与此同时，该系统也解决了当前其他类似系统的一些问题和痛点。相较于传统二手交易系统的繁琐沟通和交易流程，本系统的交易自由度更高，在系统中无需聊天与沟通就可轻松发布和达成交易，让用户不用耗费太多精力就可以妥善的处理闲置物品。同时增加了对用户隐私和数据安全的保护，平台不会泄露物品发布者的相关信息，是否与需求者进行沟通或者交易的权利在用户自己手中。这些优点和改进对于共享经济模式的发展具有重要意义，不仅提升了共享经济在物品共享领域的应用效果和用户体验，还为相关领域的研究和实践提供了新的思路和方法。

因此，本研究不仅关注于技术实现，更注重于共享经济模式在物品共享领域的应用与发展，对于推动经济社会发展和促进创新创业具有重要的实践意义和学术价值。

## 国外研究现状

当谈及共享经济的国外研究现状时，可以看到这一领域受到了广泛的关注和深入的探讨。举例来说，美国的共享经济专家杰里米·里夫金(Jeremy Rifkin)等人从社会学、经济学和技术学等多个角度对共享经济进行了深入研究，提出了许多深刻的见解和理论，对该领域的发展产生了积极的影响。他们探讨了共享经济的本质、对传统商业模式的挑战以及社会变革的潜力，为共享经济的理论研究提供了重要的实证案例。同时，国外的一些知名的共享经济平台如Airbnb、Uber等，也成为了研究的焦点。例如，Airbnb是全球最大的共享住宿平台，Airbnb采用的是P2P（Peer-to-Peer）模式，即点对点的共享经济模式，通过平台连接房东和房客，实现房屋资源的共享和利用。这种模式打破了传统旅馆业的垄断局面，让更多的个人房东有机会参与到旅游住宿市场中来，大大丰富了住宿选择，并促进了住宿价格的竞争和透明度。具体到物品免费共享平台，国外比较知名的平台是Freecycle Network，该平台的核心理念是“一个人的垃圾，另一个人的宝贝”。通过这种方式，用户可以将自己不需要的物品分享给他人，避免这些物品变成垃圾，减少了资源的浪费和环境的负担。这一理念与本系统的核心理念是一致的。

## 国内研究现状

转而谈及国内的研究现状，虽然共享经济在国内起步较晚，但是发展势头迅猛。越来越多的学者和机构开始关注共享经济领域并进行了一系列的研究工作。郑志来在《共享经济的成因、内涵与商业模式研究》分析了共享经济的发展原因，主要归因于移动互联网的迅速普及、第三方支付的普及以及多种新技术的涌现。同时，他指出了共享经济的本质在于去除中间环节。通过实例分析，他说明了这种经济模式的优势，包括提高产品利用率、降低成本、价格合理以及可持续发展等方面[4]。还有许多研究对滴滴出行、ofo等共享出行平台的案例进行分析，探讨了共享经济对城市交通和出行方式的有益影响。此外，共享经济对城市发展、环境保护和社会文化等方面的影响也成为了研究的热点之一。例如，共享单车在中国城市中的普及和发展，引发了学者们对城市交通运输、城市规划和环境影响等方面进行深入研究。国内免费物品共享平台在当前的市场中相对欠缺，主流的闲置物品处理平台如咸鱼、转转等更多偏向于商品的买卖交易，但这类二手交易平台普遍存在大量问题，如商品质量问题、交易信用体系不健全以及售后机制不完善等等[5]。总之，缺乏以免费共享为主要理念的平台。这也意味着国内的物品共享领域还有很大的发展空间和机会。由此可见免费物品共享平台的发展前景广阔，有助于满足用户对于免费资源共享的需求，促进社区互助合作，减少资源浪费和环境压力，是共享经济领域一个非常有潜力和发展空间的方向。

## 研究内容

本研究以Spring Boot和Vue核心框架为基础，旨在设计和实现一款名为"ShowHub"的物品共享平台，该平台将提供一系列全方位的服务和功能，以满足用户在物品共享领域的需求。Web端作为管理员管理平台的核心部分，拥有多项功能：物品管理，包括物品的添加、编辑、删除等操作，以及对物品信息的详细管理；用户管理，包括用户信息查看和编辑等；消息通知系统，用于向用户发送通知和提醒，以及管理纠纷和解决问题；数据分析和报告功能，通过对用户和物品的数据进行深度分析，为平台运营和决策提供有力支持。而小程序端则为广大用户提供了便捷的操作体验和丰富的功能选择，用户可以通过小程序进行注册、登录和个人资料的维护，同时可以选择多种交易方式进行物品的共享和交换，包括免费共享、以物换物和二手交易。用户还可以通过小程序进行物品的搜索与查看、留言和消息通知的接收，以及对交易过程中的不法行为进行申诉。其中，信誉分系统是平台的重要组成部分，通过对用户行为的评估，增强平台的安全性和信任度，提升用户体验。物品推荐算法则通过用户行为和偏好的分析，为用户提供个性化的物品推荐服务，增加平台的活跃度和用户黏性。

关键技术点方面，平台采用了跨平台设计，将Web端和小程序端有机结合，实现了信息的共享和互通；UniApp框架应用为小程序端的开发提供了便捷和高效的工具；信誉分系统和物品推荐算法则是平台智能化和个性化的体现，提升了用户体验和平台运营效率；通过这些技术和功能的整合，"ShowHub"旨在成为一个高效便捷、安全可靠的物品共享平台，为共享经济领域注入新的活力和动力。

# 相关技术简介

本章详细介绍了系统所采用的开发环境，包括后端开发环境、Web端开发环境和小程序端开发环境，旨在全面阐述系统设计过程中所涉及各个开发工具和框架。通过对这些开发环境的详细描述，可以更深入地了解系统设计的基础框架和技术支持。

## 后端开发环境

后端开发环境是指用于开发服务器端应用程序的工作环境和工具集合。在后端开发中，开发者主要负责处理数据逻辑、服务器端操作以及与数据库和前端交互的任务。常见的后端开发技术包括编程语言和框架、数据库、服务器、集成开发环境（IDE）、版本控制系统、测试工具以及部署工具。

首先是编程语言和框架，后端开发可以使用多种编程语言，如Java、Python、PHP、Ruby等，并结合各种框架如Spring Boot、Django、Laravel等来加快开发速度和提高代码质量。

其次是数据库，后端开发通常需要与数据库进行交互来存储和检索数据。常用的数据库包括MySQL、PostgreSQL、MongoDB等。

服务器也是后端开发环境中重要的组成部分，后端应用程序需要在服务器上运行，因此需要配置和管理服务器，例如使用Nginx或Apache等Web服务器。

集成开发环境（IDE）是开发人员常用的工具，用于编写、调试和测试后端代码。常见的IDE有PyCharm、Eclipse、Visual Studio Code等。

版本控制系统在后端开发中也是必不可少的，开发者使用版本控制系统如Git来管理代码的版本和协作开发。

### IntelliJ IDEA 2021.1.3(!!!放到系统实现 开发环境)

IntelliJ IDEA 2021.1.3是一款由JetBrains推出的集成开发环境（IDE），旨在为开发者提供高效、可靠的编程环境。该IDE支持多种主流编程语言，包括Java、Kotlin、Groovy和Scala等，使开发者可以在同一个环境下进行多种语言的开发工作。其核心特点之一是智能代码编辑功能，包括强大的代码分析和智能提示，有助于开发者编写高质量、高效率的代码。此外，IntelliJ IDEA 2021.1.3还具有丰富的插件支持。通过插件生态系统，开发者可以轻松集成各种主流框架和工具，如Spring、Hibernate、Maven和Gradle等，从而扩展IDE的功能和适用范围。此外，该IDE还内置了对版本控制系统的支持，包括Git、SVN等，有助于团队协作和代码管理。

安装步骤，访问JetBrains官网（https://www.jetbrains.com/idea/download/）。在该页面上，根据操作系统选择相应的版本，如Windows、macOS或Linux。点击下载按钮，系统会开始下载安装程序。下载完成后，双击运行这个安装程序，系统会提示开始安装IntelliJ IDEA。在安装过程中，可以选择安装路径和其他选项。通常建议保持默认设置。安装过程可能需要一些时间，等待安装完成后，可以看到安装成功的提示。安装完成后，桌面快捷图标大致如图2.1所示。

完成以上步骤后，就可以开始使用IntelliJ IDEA进行开发工作了。可以进行创建新项目、导入现有项目，使用IDE提供的各种工具和功能来编写、调试和管理代码。记得定期更新IntelliJ IDEA以获取最新的功能和性能优化。

窗体顶端



图2.1 IntelliJ IDEA图示（Windows）

### Spring Boot

Spring Boot是一个基于Spring Framework的开源框架，其目的在于简化Java应用程序的开发流程。采用了“约定优于配置”的理念，Spring Boot通过提供默认配置和快速启动的方式，使得开发者可以更加便捷地创建独立的、生产级别的Spring应用程序。该框架致力于减少开发者在项目配置上的工作量，以便他们能够更专注于业务逻辑的实现和功能开发[6]。

Spring Boot具备多项特性和优势。首先，它内置了许多常用的库和依赖，如Spring MVC、Spring Data、Spring Security等。这些库和依赖通常需要开发者手动配置和集成，而Spring Boot则通过自动配置的方式帮助开发者快速集成这些功能。其次，Spring Boot提供了嵌入式的Web容器，如Tomcat、Jetty等，使得开发者能够将应用程序打包成一个可执行的JAR文件，无需额外安装和配置独立的Web服务器。此外，Spring Boot还提供了多种开发工具和插件，如Spring Boot CLI、Spring Boot Actuator等，有助于简化开发流程、监控应用程序状态和性能。

在实际应用中，Spring Boot被广泛用于构建各种类型的Java应用程序，包括Web应用、微服务、批处理作业等。由于其简化的配置和快速启动特性，Spring Boot在企业级开发中备受认可和应用。开发者可以利用Spring Boot快速搭建项目骨架，集成各种功能模块，实现快速开发、部署和维护，从而提高开发效率和代码质量。

## Web端开发环境

Web端开发环境是指在进行Web应用程序开发时所需的一系列工具、软件和资源的综合集合。这个环境涵盖了开发人员在整个开发周期中所需的各种工具和技术，旨在提高开发效率、简化开发流程，并确保最终生成的Web应用程序具有良好的性能、可靠性和用户体验。首先在Web端开发环境中，开发人员需要一个适合自己的集成开发环境（IDE），如VSCode、WebStorm等。这些工具通常具有代码编辑、语法高亮、自动补全、调试等功能，能够帮助开发人员高效地编写和管理代码。其次，Web端开发环境需要包括用于前端开发的工具和技术。这包括HTML、CSS、JavaScript等前端语言的编写工具，如代码编辑器和预处理器。同时，前端开发还需要考虑到浏览器的兼容性和响应式设计，因此可能还需要使用浏览器开发者工具和移动设备模拟器等工具。

### Visual Studio Code(放到系统实现 开发环境!!!)

Visual Studio Code又称VS Code，是由微软开发的一款功能强大且备受欢迎的代码编辑器。其主要特点之一是免费、开源以及跨平台，这意味着无论是在Windows、MacOS还是Linux系统上都可以轻松使用，并且可以根据个人的喜好进行定制和扩展。VS Code的灵活性和可扩展性使得它成为了许多开发者的首选工具。由于VS Code比较轻量，不仅在打开时流畅快速，而且在安装上也比较容易。此外，VS Code还拥有强大的插件生态系统，开发者可以通过安装各种插件来扩展编辑器的功能，例如添加特定语言的语法检查器、增强调试功能、集成不同的开发工具等。这使得VS Code不仅适用于前端开发、后端开发、移动应用开发等多种场景，也能够满足个性化需求和不同项目的要求。

安装步骤：官网下载（https://code.visualstudio.com/download）与当前电脑系统相匹配的安装包版本，下载完成后双击即可打开安装包进行安装。在安装过程中，如没有特殊需求，所有都选择默认配置即可。安装完成后，打开VS Code就可以开始进行编码了。有需求也可以进一步安装相应的插件来提高工作效率。安装完成后，桌面快捷图标大致如图2.1所示。



图2.2 VS Code安装成功后图示（Windows）

### Vue.js

Vue.js是一款流行的JavaScript框架，专注于构建用户界面。它由华人开发者尤雨溪创建，于2014年首次发布。Vue的主要目标是通过提供简单易用的工具，使开发者能够快速构建交互式的Web界面。与其他框架如React和Angular相比，Vue更注重于可逐步采用、灵活性和轻量性。

Vue的核心是响应式数据绑定系统。这意味着当数据发生变化时，页面会自动更新，无需手动操作DOM。这种响应式的特性使得开发者能够专注于数据和业务逻辑，而不必过多关注页面的状态管理和更新。

除了响应式数据绑定外，Vue还提供了诸多其他功能，如组件化开发、虚拟DOM、路由管理、状态管理等。其中，组件化开发是Vue的一个重要特点，它允许开发者将页面拆分为多个独立的组件，每个组件都有自己的状态和逻辑。这种模块化的开发方式使得代码更易于维护和扩展。

Vue也有着活跃的社区和丰富的生态系统。开发者可以轻松地找到各种插件、工具和资源来辅助开发。此外，Vue还有着详细的官方文档和教程，为新手提供了学习和入门的便利。

### ECharts

ECharts是一个由百度开发的开源可视化库，是为了帮助用户创建丰富而动态的数据可视化图表。该框架支持各种常见的图表类型，包括折线图、柱状图、饼图、散点图等，以及更复杂的图形，如地图、雷达图和树图。其核心是一套声明式的可视化设计语言， ECharts利用基于 HTML5 画布的流式架构和高性能的图形渲染器，既确保了高扩展性，又保证了高性能[7]。

## 移动端开发环境

### HBuilder X

HBuilderX是一款由DCloud开发的集成开发环境（IDE），主要用于前端开发和移动应用开发。

对于移动应用开发而言，HBuilderX集成了对于原生App开发的支持，可以使用uni-app框架开发跨平台的移动应用。它还提供了丰富的插件和扩展机制，开发者可以通过安装相关插件来扩展HBuilderX的功能，满足不同项目的需求。

### Uni-app

Uni-app框架是由DCloud（HBuilderX的开发者）推出的一套跨平台应用开发框架。它基于Vue.js框架，并通过封装一套统一的API和组件，实现了一套代码可以同时运行在多个平台的目标。它是一个使用Vue .js开发所有前端应用的框架，开发者编写一套代码，可发布到iOS、Android、 Web (响应式)、以及各种小程序(微信/支付宝/百度/头条/QQ/钉钉/淘宝)等多个平台[8]。

使用Uni-app框架开发应用具有很多优势。首先，UniApp采用了基于Vue.js的开发方式，使得熟悉Vue.js的开发者能够快速上手，降低了学习成本。其次，UniApp提供了丰富的组件库和API，包括常见的UI组件、网络请求、本地存储、地理位置等功能，开发者可以通过简单的调用这些API来实现复杂的功能。此外，UniApp还支持插件市场，开发者可以通过安装插件来扩展框架的功能，满足不同项目的需求。

UniApp框架的另一个特点是支持原生应用打包，开发者可以通过UniApp框架开发出的应用打包成原生App，发布到App Store和各大应用商店，实现更广泛的应用分发。这使得UniApp不仅适用于小程序开发，也适用于移动应用的开发，为开发者提供了更多的选择和灵活性。

UniApp框架是一套强大的跨平台开发框架，通过统一的API和组件，实现了一次开发，多端运行的目标，为开发者提供了高效、便捷的跨平台开发体验。

## 其他技术

### Git

版本控制工具Git是一种广泛应用于软件开发领域的分布式版本控制系统，由Linus Torvalds在2005年创建。Git的主要功能是帮助开发团队有效地管理和协作开发项目的代码库。它通过记录代码的修改历史和分支管理，提供了一种可靠的方式来跟踪代码的演变，并允许开发者在不同的分支上进行独立的工作。

一个典型的Git项目包括一个中央仓库（repository）和多个本地仓库。开发者可以通过克隆（clone）操作将中央仓库的代码复制到本地，并在本地仓库中进行开发和修改。Git提供了一系列命令和工具，如add、commit、push、pull等，用于管理代码的提交、同步和版本控制。

Git最重要的特性之一是分支管理。开发者可以轻松地创建、切换和合并分支，这使得团队可以同时进行多个功能或修复的开发工作，而不会互相干扰。通过合并（merge）或变基（rebase）操作，可以将不同分支的代码合并到主分支或其他分支中，确保代码的一致性和稳定性。

除了分支管理，Git还提供了强大的历史记录和回溯功能。开发者可以查看代码的修改历史、提交信息和变更内容，并可以回退到任意历史版本，修复bug或恢复代码。这种历史记录的机制有助于追踪问题、审查代码质量，并且可以在需要时进行版本回退或撤销操作。

## 本章小结

本章主要介绍了物品共享平台的系统开发环境，包括后端开发环境、Web端开发环境和移动端开发环境。在后端开发环境方面，使用了IntelliJ IDEA 2021.1.3作为集成开发环境，Spring Boot作为后端框架，利用Maven进行项目管理和构建，同时Git作为版本控制工具，为团队协作提供了便利。在Web端开发环境方面，选择了Visual Studio Code作为主要的代码编辑工具，结合Vue作为前端框架，以及Element UI作为组件库，构建了Web端的界面和功能。而在移动端开发环境中，则采用了HBuilder X作为开发工具，利用Uni-app框架进行移动端应用的开发。

# 系统需求分析

本章详细介绍了物品共享平台系统的需求分析过程及结果。在功能需求方面，涵盖了用户在Web端和小程序端的各项操作，包括物品管理、用户管理、通知发送、纠纷解决以及数据分析等功能。在非功能需求方面，重点考虑了系统的安全性、可靠性、性能和可维护性等方面的要求。通过这些分析，为物品共享平台系统的设计和实现提供了清晰的指导和依据。

## 系统概述

本系统旨在设计和实现一个物品共享平台，名为"ShareHub"，目的是为用户提供一个便捷的平台，实现物品的共享、交易和管理。"ShareHub"平台分为Web端和小程序端，满足不同身份用户在不同场景下的需求。Web端由管理员管理物品和用户、发送通知和进行数据分析；小程序端为用户注册登录、个人资料维护、物品的共享和交易、物品搜索、留言和申诉举报等功能。

该系统的主要目标用户包括但不限于学生、居民社区成员、志愿者组织成员等具有物品共享需求的群体。这些用户希望通过平台便捷地共享自己的物品、找到需要的物品，并与其他用户进行交流和互动。通过"ShareHub"平台，用户可以实现共享物品、减少资源浪费、提高社区共享意识，从而促进社区资源的有效利用和社区氛围的建设。

## 功能需求分析

### 用例分析

1. 用户用例分析

分析用户能做什么事，用动词

1. 管理员用例分析

### 流程分析

发布物品流程分析

交易流程

事务处理：举报

事务处理：资格

(1)小程序端用户功能分析：

用户注册和登录：用户应能够方便地注册新账号并进行登录。这包括提供用户友好的注册界面和登录流程，实现了用户微信一键登陆及注册，确账号安全。

个人资料维护：用户需要能够编辑和更新个人资料，包括个人信息、联系方式等。这样可以让用户随时更新个人信息，保持资料的及时性和准确性。

物品浏览和上传：用户可以浏览平台上的物品列表，并通过搜索功能查找特定的物品。在物品详情页用户还可以进行。同时用户还可以上传自己闲置的物品，以供其他用户进行浏览和交易。

交易方式的选择：用户可以选择免费共享、以物换物或二手交易的方式进行交易。确保用户在交易过程中有多样化的选择，满足不同用户的需求和偏好。

留言和消息通知：用户可以在物品页面下留言，同时能够收到有关物品和交易的消息通知。这样可以增加用户与其他用户之间的互动，提高交易的透明度和信任度。

图3.1和图3.2展示用户功能用例图。

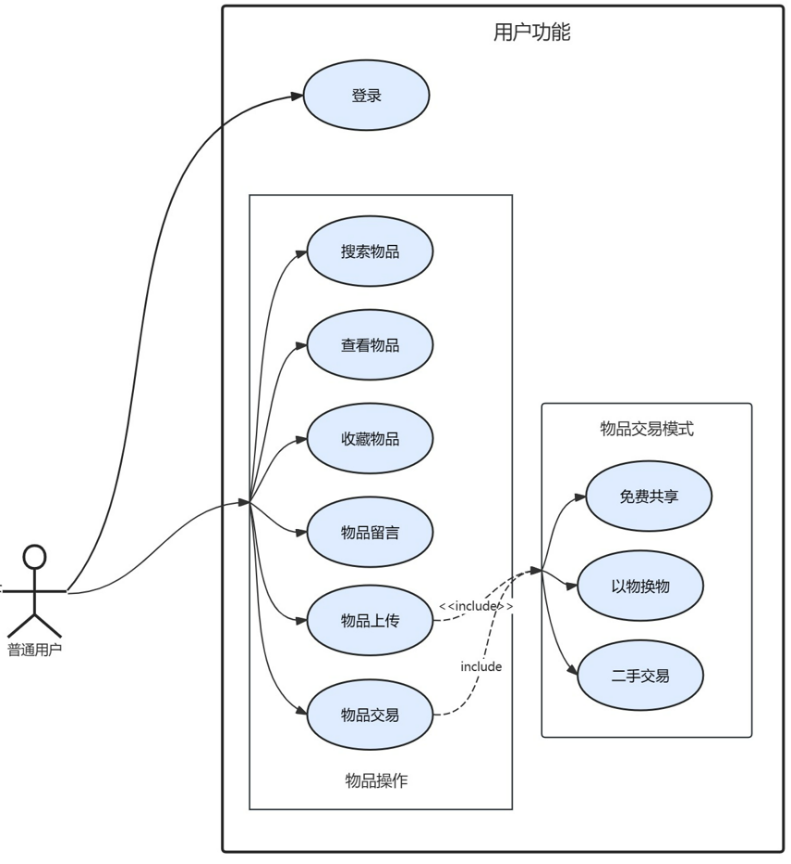


图 3.1

图 3.1 用户功能用例图

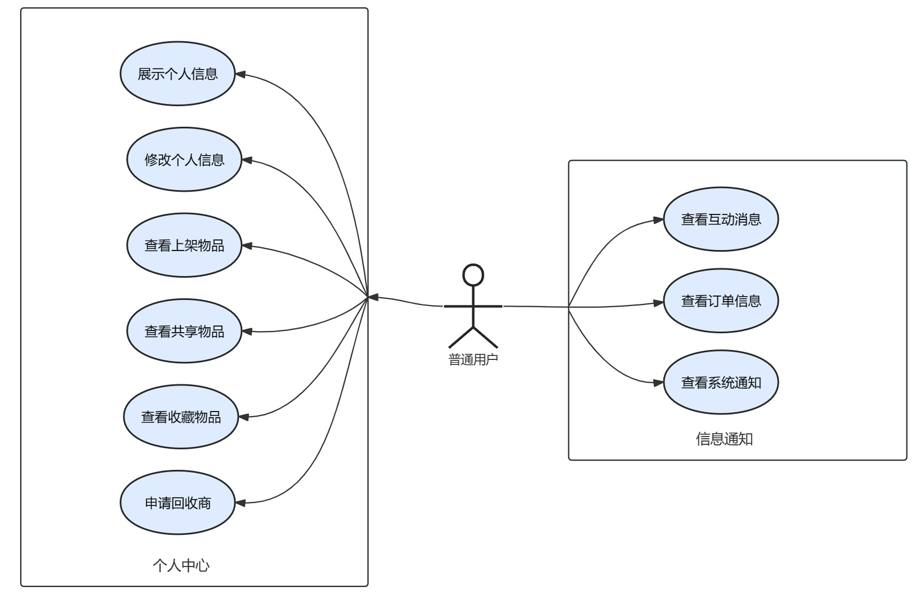


图3.2 用户功能用例图

物品推荐：根据用户的交易记录和行为偏好，系统可以推荐相关的物品给用户，提高用户体验。通过个性化的推荐算法，可以增加用户发现新物品的机会，提高交易成功率和用户满意度。

信誉分系统：每个用户都有一个初始信誉分，完成订单并收到好评可增加信誉分，收到差评则会降低。信誉分越高，用户可共享的物品次数上限越多。这样的系统可以鼓励用户诚实守信，增强交易的可靠性和稳定性。

申诉和举报功能：用户可以对订单进行申诉，并对违规物品进行举报，管理员会进行相应处理并反馈给用户。这样可以保障用户的权益，维护平台的公平和正义。

(1)Web端管理员功能分析：

物品管理：管理员需要能够对平台上的物品进行添加、编辑和删除操作，以确保物品信息的准确性和完整性。这包括对物品信息的查看、修改和删除，确保平台上展示的物品符合规范。

用户管理：管理员需要管理用户账号，包括用户的注册、编辑和查看。这涉及到用户信息的管理和维护，保障用户账号的安全和可靠性。

通知发送：管理员可以向用户发送重要消息和通知，以保持与用户的及时沟通。这包括对用户发送通知消息、提醒和公告等，确保用户能够及时获得平台的重要信息。

纠纷解决：管理员需要处理用户间的纠纷，包括订单申诉、举报审核等操作，以维护平台的秩序和公平，并善解决用户的问题。

资历审核：若有用户申请成为二手回收商，管理员会对其提交的相关资历进行审核，来决定其身份是否有效合法。

数据分析：管理员需要对用户和物品的数据进行分析，了解平台的运营情况和用户行为，为平台的进一步发展和改进提供参考依据。这包括对数据的收集、整理、分析和可视化展示等工作，以发现问题并提出解决方案。

如图3.3 Web端管理员功能用例图。

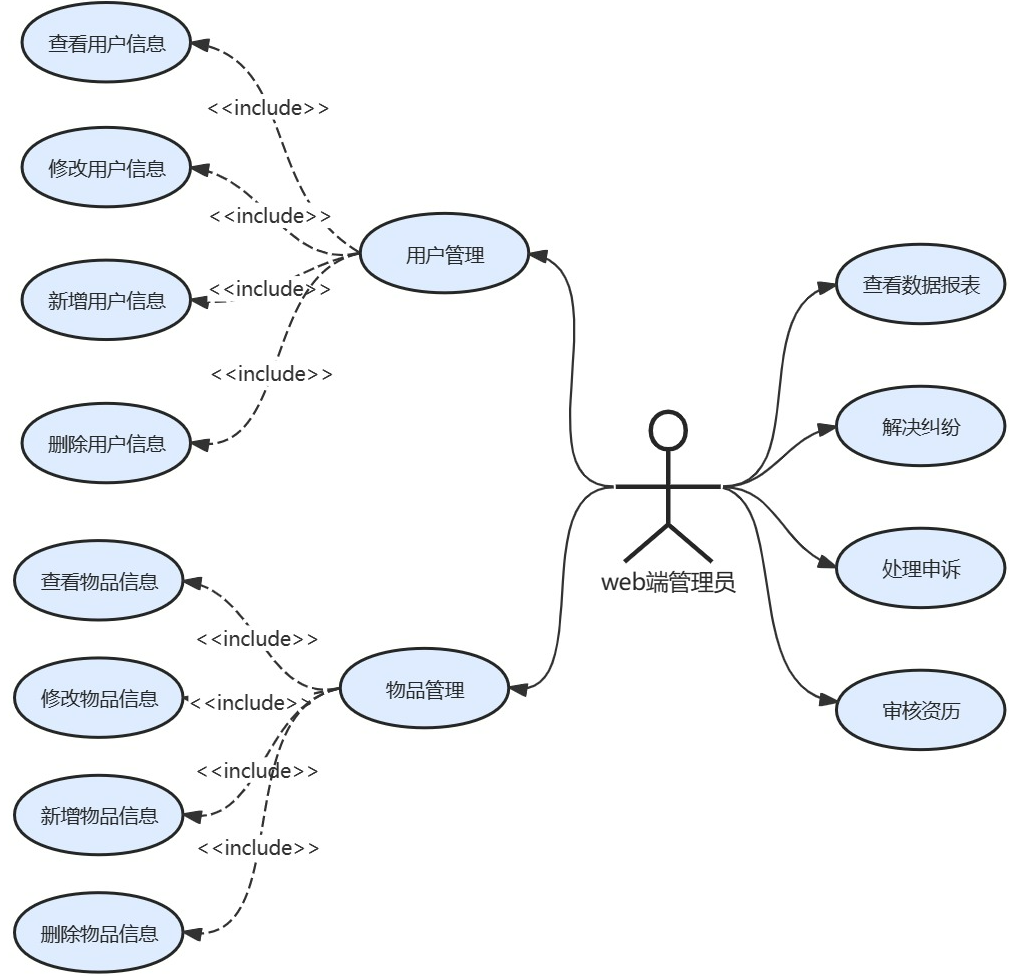


图3.3 Web端管理员功能用例图

## 非功能需求分析

非功能需求是指除了满足用户功能需求外，软件产品需要满足的其他特性，如性能、安全性、易用性等方面的要求。下面对物品共享平台的非功能需求进行分析。

### 市场需求分析

物品共享平台"ShareHub"的市场需求分析可以从几个方面来考虑。首先，需要对当前市场上的需求进行评估，包括物品共享的潜在用户数量、其需求类型等。主要借助网上调查问卷的方式对市场进行调研。具体调查问卷见图3.4。

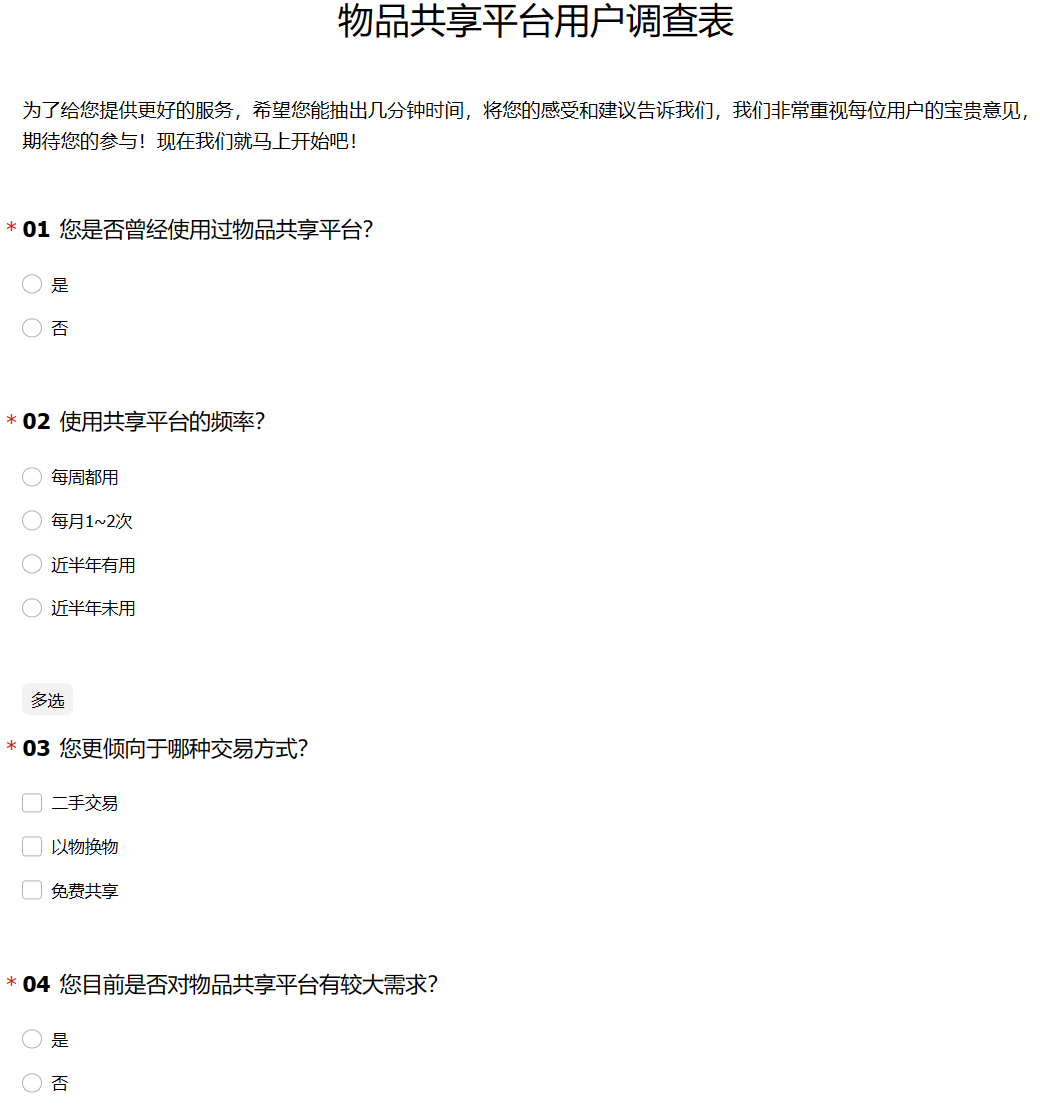


图3.4 用户问卷调查表

基于上述问卷调查的结果，针对每个问题绘制了饼状图，如图3.4、图3.5、图3.6和图3.7所示，发现约66.7%的用户曾经使用过物品共享平台，这表明物品共享平台在市场上已经有一定的认知度和普及度。然而，令人关注的是有33.33%的用户未曾使用过这类平台，这可能是由于缺乏相关信息、担心安全性或者其他原因造成的。因此，在推广和宣传方面可能需要加大力度，提高用户对物品共享平台的认知和信任度。

就共享平台的使用频率而言，有41.67%的用户近半年未使用过，这可能反映了用户对现有平台的使用体验或者物品种类的满意度不高。另一方面，有8.33%的用户近半年使用过，25%的用户每月1~2次使用，25%的用户每周都使用，这说明仍然有一部分用户对物品共享平台有着较高的需求和频繁的使用习惯。

对于用户倾向的交易方式，38.1%的用户倾向于免费共享，28.57%的用户倾向于以物换物，33.33%的用户倾向于二手交易。这显示了用户对于不同交易方式的偏好存在差异，免费共享和二手交易的比例较高，可能是由于用户更加注重经济性和环保性。因此，在平台设计和运营中需要考虑不同用户群体的需求，提供多样化的交易方式和服务。

最后，约66.67%的用户表示有较大的对物品共享平台的需求，这为"ShareHub"物品共享平台的市场前景提供了积极的信号。然而，仍有33.33%的用户表示没有较大需求，这可能与个人生活方式、消费习惯等因素有关。因此，在平台的推广和服务内容上需要不断改进和创新，以满足用户多样化的需求和期待。

。

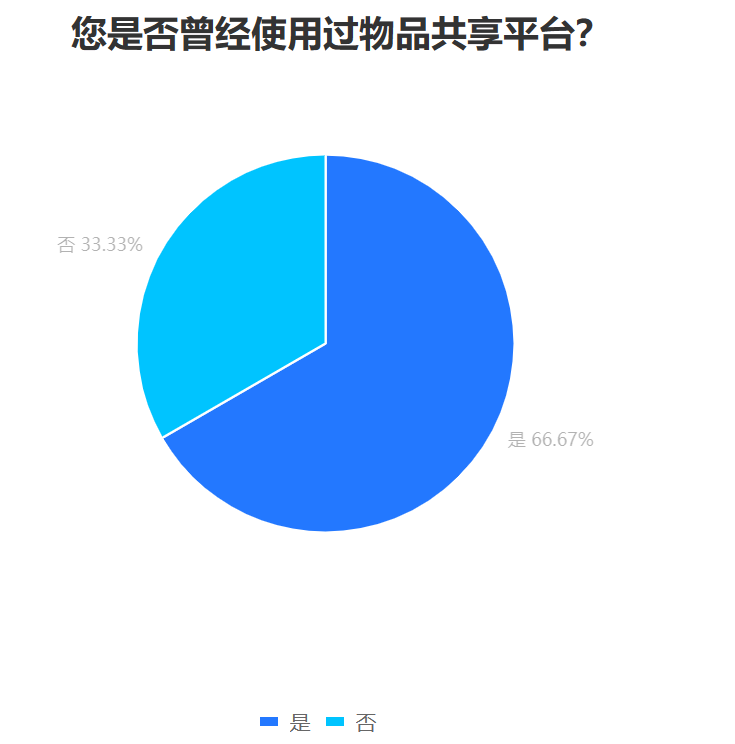


图3.4 平台调查饼状图

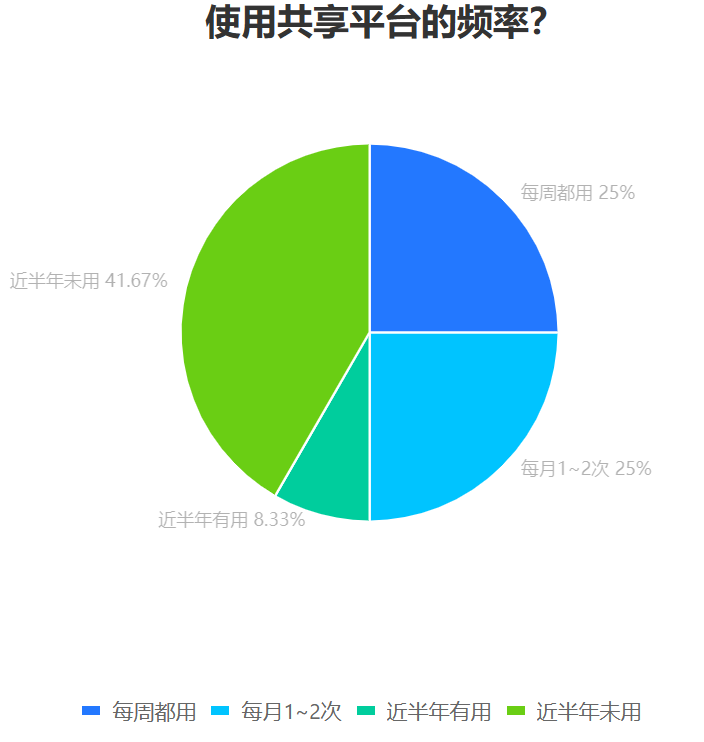


图3.5 使用频率饼状图

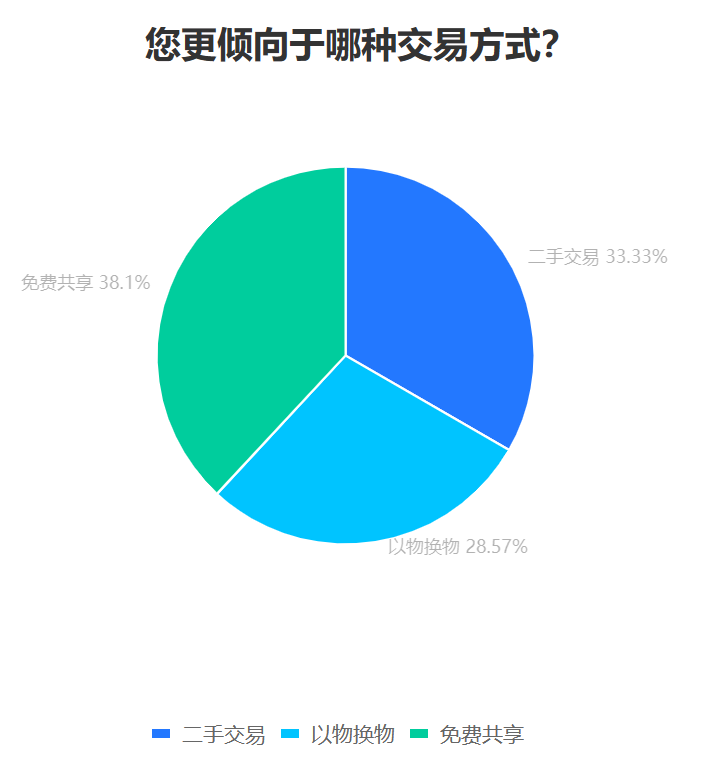


图3.6 倾向交易方式饼状图

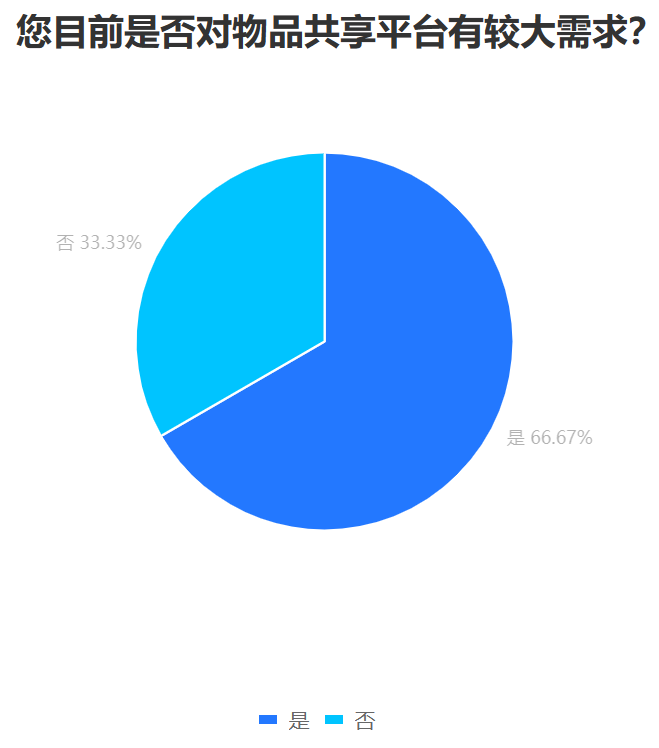


图3.7 平台需求饼状图

通过上述的市场调研和数据分析，可以确定是存在足够的市场需求来支撑物品共享平台的发展。

### 安全需求分析

安全性需求分析：系统需要确保用户身份认证和授权管理是安全可靠的，以防止未经授权的用户访问系统资源或执行操作。这意味着在用户登录和操作时，系统需要对用户进行有效的身份验证，并根据其权限级别授予适当的访问权限。其次，在数据传输过程中，特别是在用户登录、交易和个人信息传输时，系统需要使用加密协议来保护数据的机密性和完整性。通过使用相关加密技术来确保数据在传输过程中不被窃取或篡改。另外，系统还需要防范常见的安全威胁，如SQL注入和跨站脚本攻击。采用参数化查询和编码转义等技术可以有效地防止此类攻击。最后，系统应该具备异常监测和安全审计功能，及时检测到异常行为和安全事件，并记录相关日志以便事后分析和追踪，从而及时采取应对措施并提升系统的安全性。综上所述，安全性需求分析在系统设计和实现中至关重要，只有确保系统的安全性，才能有效保护用户的信息和系统的稳定运行。

### 性能需求分析

性能需求分析：在确保系统满足用户需求的同时，也着重于提升用户体验和系统的可用性。首先，系统需要尽量缩短响应时间，以保证用户的操作流畅和及时性。较短的响应时间能够提升用户满意度，使用户在系统交互过程中感受到更高的效率和便捷性。其次，系统的稳定性和可靠性是性能需求的关键，即系统在长时间运行过程中不应出现故障或崩溃，能够稳定提供服务。这种稳定性和可靠性不仅体现在系统的日常运行中，还包括系统的故障处理和恢复能力，确保用户的正常使用体验。另外，系统的可扩展性也是性能需求的重要考量，它保证系统能够持续支撑业务的进一步发展。通过合理的架构设计和技术选型，系统可以在业务需求不断变化和扩大的情况下，灵活地进行扩展和升级，确保系统能够满足未来的业务增长和用户增长。因此，性能需求分析需要综合考虑系统的响应时间、稳定性、可靠性、可扩展性等多个方面，以确保系统能够满足用户的需求并具备良好的性能表现。

## 运行环境需求

## 本章小结

本章主要介绍了系统需求分析，围绕系统概述、功能需求分析和非功能需求分析展开。在系统概述部分，详细介绍了物品共享平台的整体概况，包括项目的背景、目标和范围。而功能需求分析部分则着重于对平台功能的详细描述和需求分析，包括用户管理、物品管理、交易方式等方面的功能需求。在非功能需求分析中，将市场需求分析、安全需求分析和性能需求分析等方面进行了详细讨论。

# 系统设计

开展系统设计，可以帮助开发者深入理解项目的业务需求和用户期望，明确项目目标和需求。同时还可以提前发现和解决项目中可能存在的问题和隐患，减少后续开发过程中的调整和修改，从而降低了项目的开发成本和风险。通常在系统设计阶段开发者需要考虑系统整体的架构设计、功能模块设计以及非功能模块设计等

## 系统架构设计

### 总体架构设计

在设计物品共享平台的总体架构时，需要全面考虑不同用户对象之间的不同特点和功能需求，以及他们之间的协作和互动。采用多平台分层架构是一种有效的方式，可以将系统划分为不同平台和层级。每个平台服务相应的对象，每个层级负责特定的功能模块，这有助于提高系统的可维护性和扩展性，同时能够充分满足不同用户群体的需求，提供更便捷、灵活的共享体验。

对于小程序端，作为普通用户主要使用的平台，其优势在于其出色的便利性和用户友好性。微信小程序端的使用无需用户下载额外的应用程序，可以直接在微信中使用平台功能，这极大地降低了用户的使用门槛，提高了用户的使用便捷度，同时也增强了用户的黏性。用户只需在微信中搜索并打开小程序，即可随时随地享受到平台提供的各项功能和服务，无需进行繁琐的下载和安装操作，极大地方便了用户的日常使用。除此之外，微信小程序端还具有快速加载和操作简便的特点，用户可以迅速打开小程序并进行相关操作，无需等待长时间的加载过程，极大地提升了用户的使用体验。用户可以随时随地通过手机进行与物品的互动，例如浏览、收藏、上传等操作，使得用户更容易参与到平台的使用中，极大地方便了用户的生活。这种便捷和灵活的使用方式，使得小程序端成为用户首选的平台之一，也有助于增加平台的用户数量和活跃度。

窗体顶端

对于Web端，作为管理员主要平台的情况下，其优势不仅体现在功能的强大和操作的灵活性，还体现在提供了更为全面和细致的管理功能。管理员可以通过Web端实现对平台的全面管理和监控，这包括了对物品管理、用户管理、交易处理等方面的功能。Web端提供了更丰富和复杂的管理界面，相比于小程序端更适合处理繁琐的管理任务，使得管理员能够更轻松地查看和处理各种信息和数据。这一点对于管理员来说尤为重要，因为他们需要处理大量的数据和信息，并作出相应的决策。Web端的操作界面更为直观，功能更为丰富，管理员可以更有效地进行信息管理和决策，而不受到界面的限制。因此，Web端作为管理员使用的主要平台，其功能丰富、操作灵活，为管理员提供了更强大和便捷的管理工具，有助于提升平台的管理效率和运营水平。

对于开发者，使用双平台架构的设计不仅可以满足用户和管理员的不同需求，还可以为开发者提供更多的开发和运营策略，促进平台的持续发展和壮大。因此，对于开发者而言，使用双平台架构可以提供更多的发展机会和商业机遇，有助于实现平台的长期稳定运营和可持续发展。

如图 4.1展示了双平台架构系统设计：

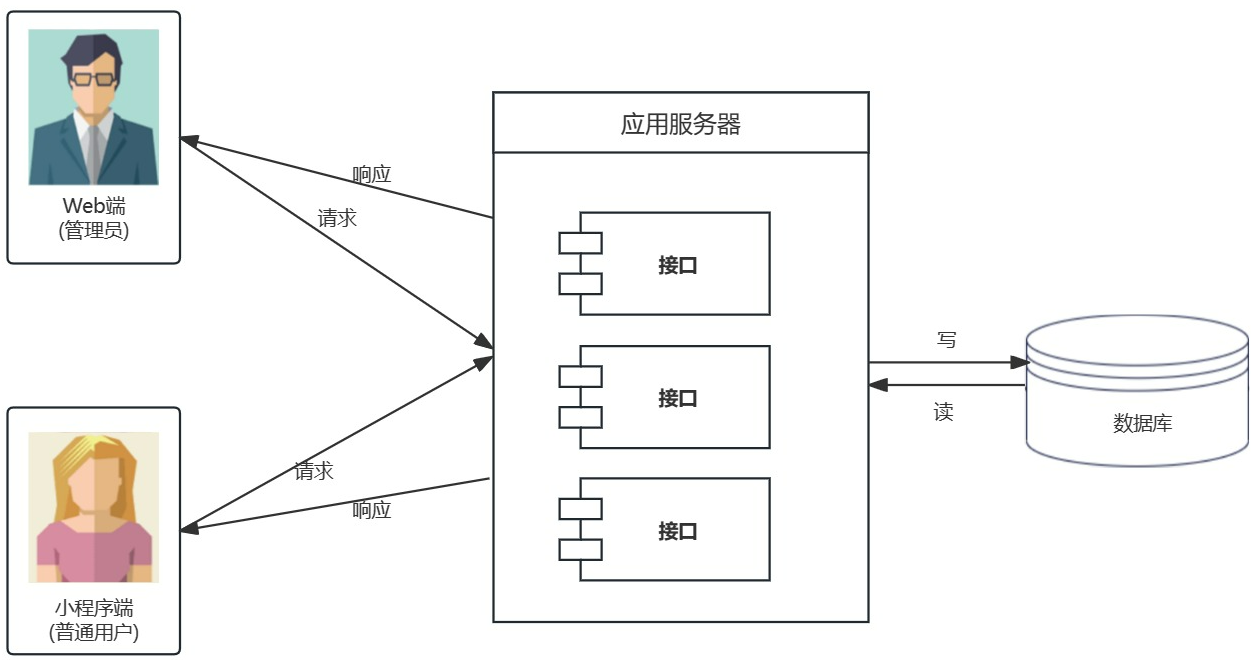


图 4.1双平台架构系统设计

### 逻辑架构设计

在“ShareHub”物品共享系统的整体逻辑架构中，采用了MVC（Model-View-Controller）软件架构设计，如图4.2 展示了MVC架构模式。MVC是一种常用的软件架构模式，用于组织代码以实现用户界面、业务逻辑和数据存取的分离。在MVC架构中，模型（Model）负责表示应用程序的数据和业务逻辑，视图（View）负责用户界面的呈现，而控制器（Controller）则负责处理用户输入并作出相应的响应[9]。

模型（Model）:在“ShareHub”物品共享系统中，模型层包括管理物品信息、用户数据、交易记录等各种数据的操作和管理。模型层会定义物品对象、用户对象、交易对象等实体类，并负责与数据库进行交互，执行数据的读取、写入和修改操作。此外，模型层还包括了用户的信誉分系统，用于记录和管理用户的信用情况。

视图（View）：在“ShareHub”中，视图层包括了Web端和小程序端的界面设计和实现。Web端的视图用于管理员的操作和管理，包括物品管理、用户管理、数据分析和报告等功能的展示。而小程序端的视图则主要面向普通用户，包括用户注册登录、个人资料维护、物品浏览与交易等功能的展示。

控制器（Controller）：在“ShareHub”中，控制器负责路由用户的请求，验证用户的输入数据，调用适当的模型方法进行数据处理，并将处理结果传递给视图进行展示。控制器还负责处理用户的申诉和举报请求，以及与管理员进行交互进行审核和反馈。

通过采用MVC架构模式，"ShowHub"物品共享平台能够实现数据与界面的分离，能够帮助维护人员从整体上了解软件系统的构成，以及有效地降低软件维护的难度和减少软件维护的成本。同时，MVC架构还能够使开发团队更好地分工合作，提高开发效率[10]。

窗体顶端

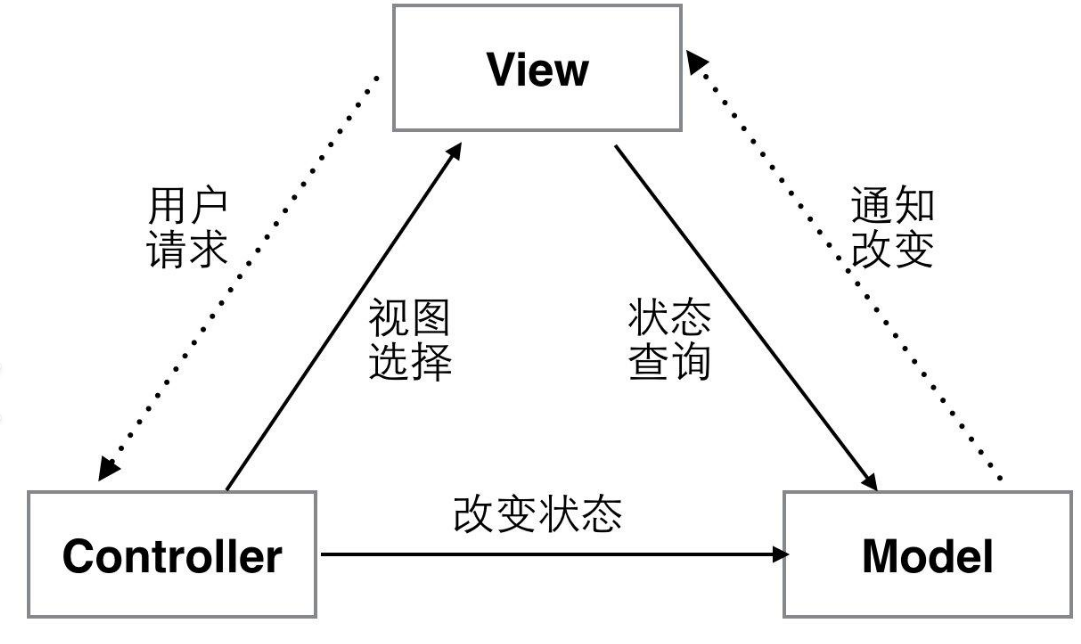


图 4.2 MVC架构示意图

### 物理架构设计

物理架构图是一种图形化的表示，用于展示软件系统的物理组成和部署结构。这种图通常描述了软件系统中各个模块、组件或服务是如何分布在物理设备上的，以及它们之间的连接方式。对于物品共享系统的物理架构主要有：用户端和管理端，其中用户端是基于移动设备，管理端是电脑设备。如图4.3物品共享系统的物理架构图。

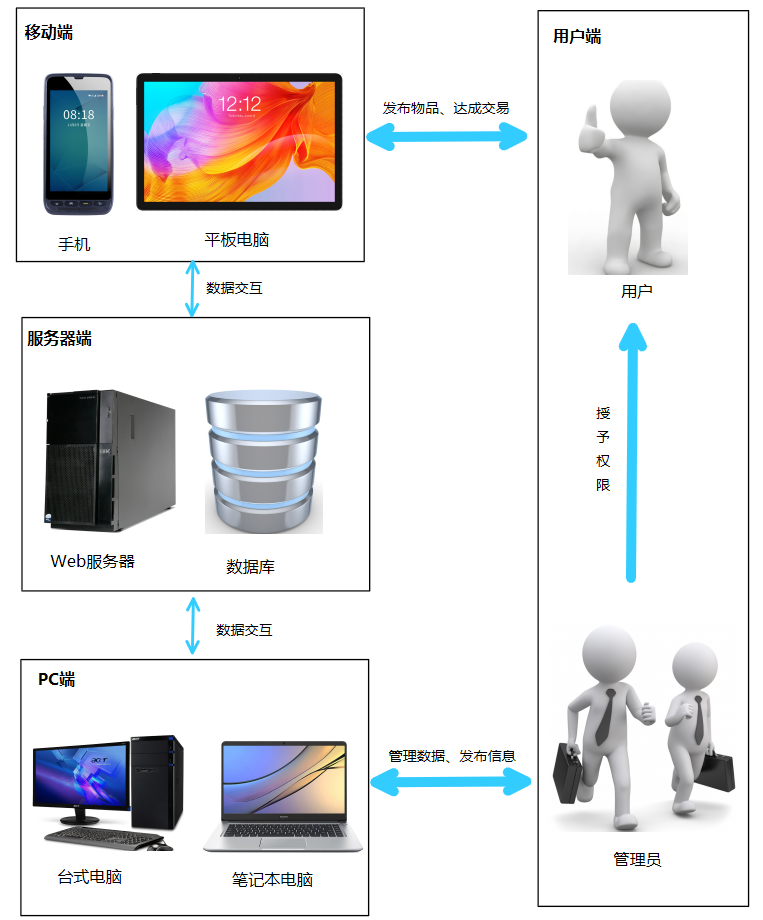


图 4.3物品共享系统物理架构图

## 功能模块设计

功能模块设计的目的在于对系统的功能进行细化和规划，将整个系统划分为多个相互独立、功能明确的模块。这样做的好处在于可以清晰地了解系统的各个功能点，为开发人员提供了具体的工作方向。“ShareHub”物品共享系统主要分为Web端和小程序端，下面小节里面主要是对Web端和小程序端分别进行功能模块的设计分析。

### Web端功能模块设计

物品共享系统中Web端的功能模块设计可以涵盖多个方面，以满足管理员对系统的全面管理需求。Web端功能分为三个大模块，分别是用户管理模块、管理模块、订单管理模块和公告与消息模块，如图4.4 Web端功能模块图。

用户管理模块中包括用户信息管理，即对用户的信息进行增、删、改、查等；以及用户资历验证，也就当有用户申请称为回收商时，需要提交相关资历证明，经管理员审核通过后，才能申请成功；申诉处理，当用户对订单有纠纷或者异议时，可提交申诉，管理员处理申诉并返回结果。

物品管理模块中包括物品信息管理，即对物品的信息进行增、删、改、查等；举报处理，当用户发现违规物品时，可对其进行举报，管理员来处理举报内容。

订单管理模块包括对订单的管理，以及数据可视化，也就是对不同的物品种类、物品交易模式对应着不同销售量的分析，会以可视化图来进行展示。

公告与消息模块，管理员可以控制修改小程序端主页的轮播图信息，以及主页的滚动公告的信息。

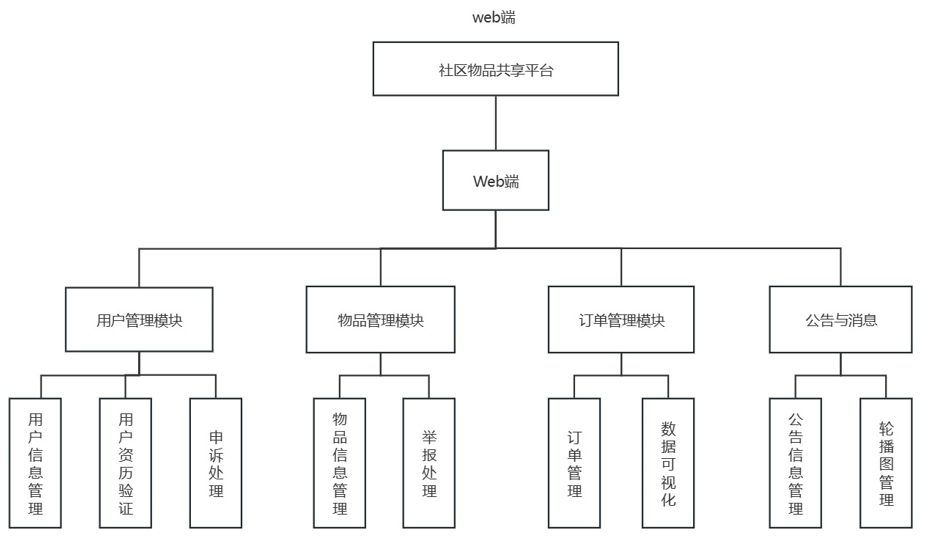


图 4.4 Web端功能模块图

### 小程序端功能模块设计

物品共享系统中小程序端的功能模块设计应该充分考虑用户的使用场景和需求。小程序端主要分为四个模块：用户模块、物品模块、通知模块和交易模块，如图4.5小程序端功能模块图。

用户模块包括用户的注册与登录，用户在使用小程序端的一些交互功能时，例如物品评论，申请共享等，就必须先进行登录；以及个人资料管理，用户点击‘我的’导航栏后会跳转到个人信息页面，点击个人头像就可进行个人资料的修改。同时用户还可在物品详情页面申请进行物品的交易和共享，如果是回收商用户，还可以在公告通知栏或者个人中心页申请资历验证称为回收商。

物品模块包括物品状态管理，即用户对于自己上架的物品可以进行状态的管理，当用户不想共享时，可以选择下架物品。物品搜索与筛选，用户可以对物品进行关键词。搜索物品分类与推荐，用户在挑选物品时，可以根据物品分类选择自己所需的物品种类进行挑选，同时系统还内置了推荐算法，会根据用户的行为和偏好来进行物品的推荐。查看物品详情，用户可以点击物品查看物品详情，包括对物品的叙述以及物品的详情图。

通知模块包括互动通知，主要用于促进用户之间的互动和交流。当其他用户对当前用户发布的物品感兴趣并进行留言时，系统会发送互动通知给当前用户，提醒用户及时回复和处理留言。此外，订单通知也是通知模块的重要功能之一。当有其他用户申请进行物品的交易或共享时，系统会向当前用户发送订单通知，以便用户及时确认订单并进行后续操作。这样的订单通知能够提高用户对交易状态的实时感知，有效促进交易的顺利进行。另外，系统通知也是通知模块的关键功能之一。系统通知主要包括系统公告和用户申诉结果等信息的发送。当系统发布重要公告或用户的申诉结果出炉时，系统会向相关用户发送系统通知，以便用户及时了解平台的最新动态和重要信息。通过这样的系统通知机制，用户可以及时获取平台的相关信息，保持对平台运营的关注和了解。

交易模块包括交易的流程，‘ShareHub’共享平台是一个免费便捷的共享物品信息平台，不涉及金钱的交易，交易过程中会给想要达成交易的双方提供彼此的联系方式，具体交易流程细节由用户私下完成。历史交易记录，每个用户可以查看自己的历史交易记录。评价和评分，用户完成交易后可以分别对双方的交易体验进行评分，较高和较低的评分都会影响对方信誉分的增减。

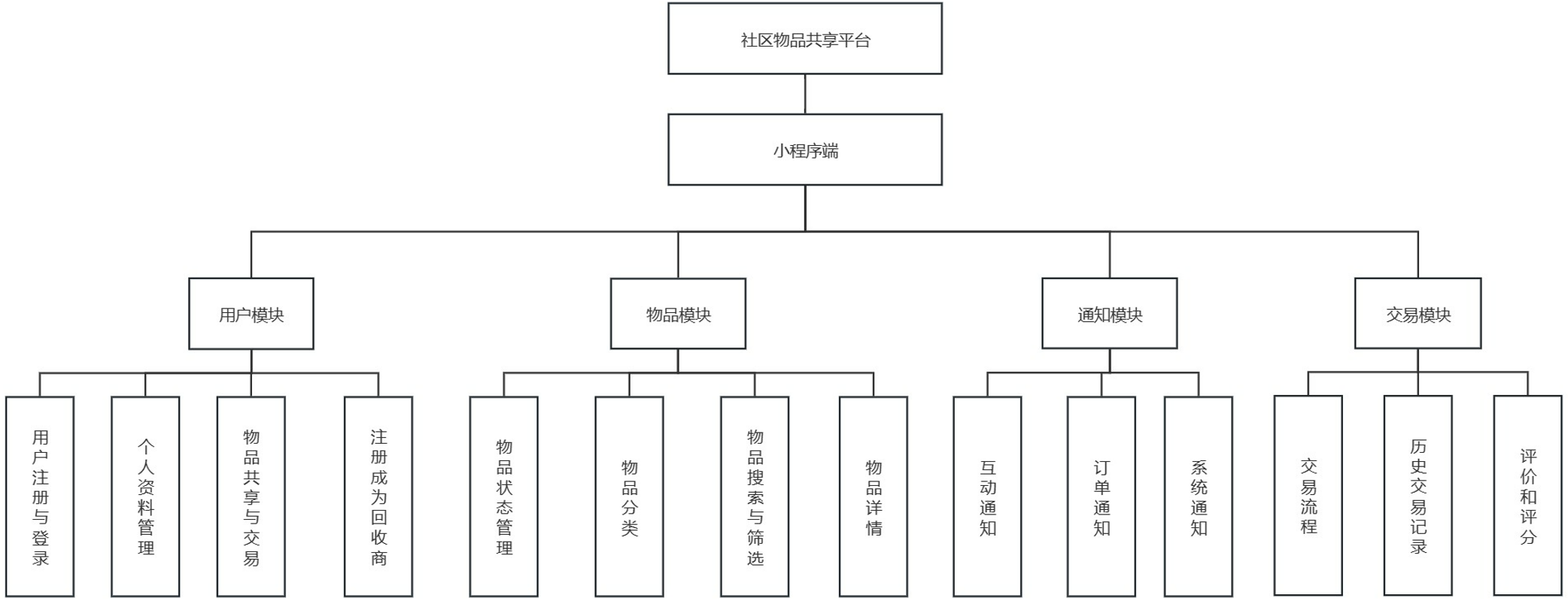


图 4.5小程序端功能模块图

## 非功能模块设计

在软件开发过程中，非功能模块设计扮演着至关重要的角色。其作用不仅在于确保系统的正常运行，还涉及到用户体验、系统性能、安全性以及可维护性等方面。

### 安全设计

对数据安全设计，项目应多项措施以确保用户隐私和数据安全。

采用JWT（JSON Web Token）进行用户身份认证和授权管理，通过JWT生成的令牌可以有效地验证用户身份，防止未经授权的访问。JWT作为一种安全的身份验证机制，具有无状态、跨平台等特点，在网络传输中不会暴露用户的敏感信息，有效地保护用户隐私。除了身份认证外，JWT还可以通过附加声明的方式实现授权管理，确保用户只能访问其具有权限的资源，从而进一步提高系统的安全性。

为了加强用户密码的安全性，项目应MD5等加密算法对用户敏感信息进行加密存储。MD5算法是一种常用的哈希算法，能够将用户的原始密码转换成固定长度的哈希值，使得用户密码在数据库中存储的是经过加密处理的密文，即使数据库被非法获取，也无法直接获取到用户的原始密码，有效地保护用户的账户安全。

为了防止SQL注入等数据库攻击，项目在数据库操作接口上实施了一系列安全措施。其中包括使用参数化查询等安全机制，通过对用户输入的数据进行严格的验证和过滤，确保用户输入的数据不会被误解析为恶意的SQL语句，从而有效地防止SQL注入攻击。

### 性能保障设计

针对性能保障设计，在保证了使用效率较高的SQL语句的基础上，针对数据库操作，项目使用了优化后的SQL语句，包括索引的合理利用、SQL查询语句的优化等，以确保数据库的查询效率和性能。通过对数据库表的设计和索引的建立，能够有效地减少查询时间，提高数据检索的速度，从而加快系统的响应速度和处理效率。

### 界面设计

在设计用户界面时，过度依赖单一文本和乏味色调可能导致视觉疲劳。相比之下，采用多彩色彩和多媒体元素能够增强用户的体验，降低他们长时间使用界面时的疲劳感，同时激发他们对软件的好奇心和探索欲[11]。本项目项目整合Uniapp框架和Element-UI框架进行界面设计。在小程序端界面设计方面，项目注重了用户体验和视觉效果的优化，以提升用户的操作体验和平台的吸引力。首采用了Uniapp框架提供的组件库和模板，实现了页面的快速开发和构建。Uniapp框架提供了丰富的组件和模板，包括导航栏、表单、列表、卡片等，可以快速构建出符合用户期望的界面布局和交互效果。针对Web端，项目采用了Element-UI框架提供的UI组件库，用于实现页面的样式设计和组件定制。Element-UI框架提供了丰富的UI组件，包括按钮、表格、对话框、菜单等，可以实现各种风格和样式的界面设计，满足不同用户群体的需求。通过对界面元素的选择和排版，使得用户可以直观、方便地操作平台，提高了用户的使用体验和满意度。

项目在界面设计方面充分利用了Uniapp框架和Element-UI框架提供的功能和特性，实现了界面的快速开发和定制化设计，以提供优质的用户体验和视觉效果。通过合理的布局和设计，使得用户可以方便、直观地使用平台。

## 本章小结

本章主要介绍了系统设计，涵盖了系统架构设计、功能模块设计和非功能模块设计三个方面。在系统架构设计中，通过总体架构设计、逻辑架构设计和物理架构设计三个层次的规划，构建了系统整体的结构框架，从宏观到微观全面考虑了系统的组织架构和技术实现。在功能模块设计方面，分别针对Web端和小程序端进行了功能模块的设计，明确了各个模块的功能和相互关系，为后续的开发工作提供了清晰的目标和方向。而在非功能模块设计中，则重点关注了安全设计、性能保障设计和界面设计等方面，通过多项措施确保了系统的安全性、性能和用户体验，为系统的稳定运行和用户满意度提供了保障。

# 数据库设计

本章主要是介绍了物品共享平台的数据库设计，数据库设计通常包括六个阶段：需求分析阶段、概念结构设计阶段、逻辑结构设计阶段、数据库物理设计阶段、数据库实施阶段以及数据库运行和维护阶段[12]。通过设计数据库表和建立表之间的关系，可以实现数据的结构化存储，提高数据的存取效率和查询速度。

## 概念模型设计

根据上文对物品共享系统的设计与分析，本系统的E-R图如图5.1所示。

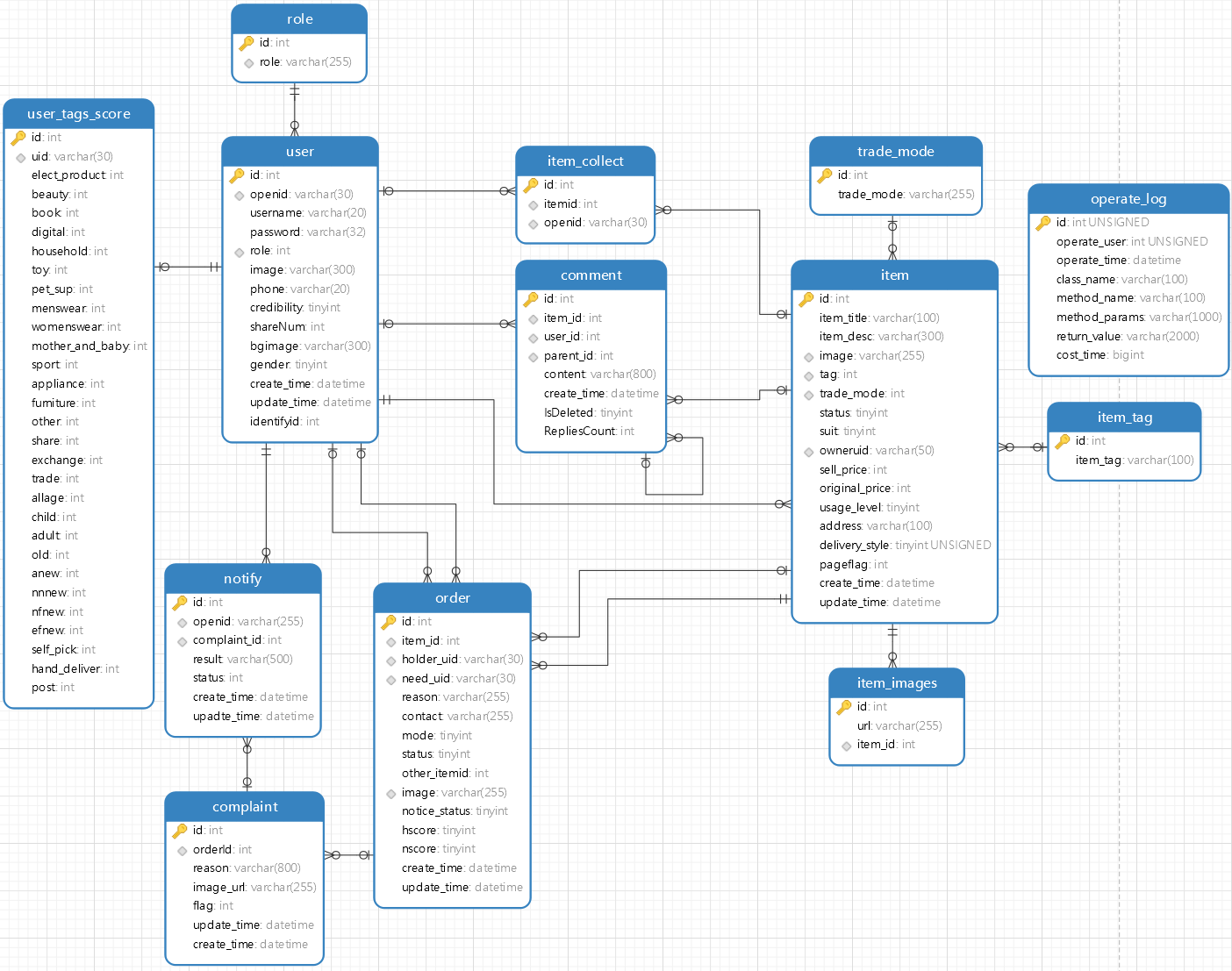


图 5.1 系统整体 E-R 图

上述E-R图中，物品表（item）是核心表之一，其中包含了各种物品的信息，例如物品名称、描述、图片等。物品表与物品分类表（item\_tag）之间建立了关联，通过物品表中的类型字段（tag）与物品标签表中的id字段进行关联，以便对物品进行分类和标记。这种关系允许一个物品属于一个或多个标签分类，从而更好地组织和管理物品信息。

用户表（user）与评论表（comment）、物品表（item）之间也存在着密切的关系。评论表记录了用户对物品的评论信息，通过评论表中的用户ID字段（user\_id）与用户表中的ID字段进行关联，以及物品ID字段（item\_id）与物品表中的ID字段进行关联，实现了评论、用户和物品之间的关联。这种关系允许一个用户可以发表多条评论，一个物品也可以收到多条评论。

订单表（order）与用户表（user）、物品表（item）、申诉表（complaint）、之间也存在着关联。订单表记录了用户对物品的交易订单信息，通过订单表中的物品ID字段（item\_id）与物品表中的ID字段进行关联，以及用户ID字段（holder\_uid、need\_uid）与用户表中的openid字段进行关联，实现了订单、用户和物品之间的关联。同时，投诉表记录了用户对订单的投诉信息，通过投诉表中的订单ID字段（orderId）与订单表中的ID字段进行关联，实现了申诉与订单之间的关联。

## 物理模型设计

（1）用户信息表，主要包括用户微信授权openid、用户名、密码、头像、性别、创建时间等等，如表4.1。

表 4.1 用户信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **数据类型** | **允许空** | **是否主键** | **说明** |
| id | int(11) | 否 | 是 | 序号 |
| openid | varchar(30) | 否 | 否 | 授权ID |
| username | varchar (20) | 否 | 否 | 用户名 |
| password | varchar (15) | 是 | 否 | 密码 |
| role | int | 否 | 否 | 角色类型 |
| image | varchar(200) | 是 | 否 | 头像 |
| credibility | tinyint | 否 | 否 | 信誉分 |
| shareNum | tinyint | 否 | 是 | 可共享数 |
| bgimage | varchar(200) | 是 | 否 | 背景图 |
| gender | tinyint | 是 | 否 | 性别 |
| create\_time | datetime | 否 | 否 | 创建时间 |
| update\_time | datetime | 否 | 否 | 更新时间 |

（2）物品信息表，主要包括物品标题、物品详情图、物品描述、物品种类、物品适用人群、物品交易模式、物品所在地址、物品交付方式、以及物品状态（在售、已下架）等等，如表4.2。

表 4.2 物品信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **数据类型** | **允许空** | **是否主键** | **说明** |
| id | int(11) | 否 | 是 | 序号 |
| item\_title | varchar(30) | 否 | 否 | 物品名称 |
| item\_desc | varchar (500) | 否 | 否 | 物品描述 |
| image | varchar(255) | 否 | 否 | 图片 |
| tag | tinyint | 否 | 否 | 分类 |
| trade\_mode | tinyint | 否 | 否 | 交易模式 |
| status | tinyint | 否 | 否 | 适用人群 |
| suit | tinyint | 是 | 否 | 物主id |
| owneruid | varchar(50) | 否 | 否 | 售价 |
| sell\_price | decimal | 否 | 否 | 原价 |
| usage\_level | tinyint | 否 | 否 | 使用程度 |
| address | varchar(100) | 是 | 否 | 地址 |
| delivery\_style | tinyint | 否 | 否 | 交付方式 |
| create\_time | datetime | 否 | 否 | 创建时间 |
| update\_time | datetime | 否 | 否 | 更新时间 |

（3）订单表，主要包括订单id、物品id、共享方id、需求方id、申请共享的理由、

订单交易模式等等，如表4.3。

表 4.3 订单表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **数据类型** | **允许空** | **是否主键** | **说明** |
| id | int(11) | 否 | 是 | 序号 |
| item\_id | int(11) | 否 | 否 | 物品id |
| holder\_uid | varchar (30) | 否 | 否 | 共享方id |
| need\_uid | varchar (30) | 否 | 否 | 需求方id |
| reason | varchar (255) | 是 | 否 | 共享理由 |
| contact | varchar (255) | 否 | 否 | 联系方式 |
| mode | tinyint | 否 | 否 | 交易模式 |
| notice\_status | tinyint | 否 | 否 | 读取状态 |
| hscore | tinyint | 是 | 否 | 共享方评分 |

表 4.3续 订单表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **数据类型** | **允许空** | **是否主键** | **说明** |
| nscore | tinyint | 是 | 否 | 需求方评分 |
| create\_time | datetime | 否 | 否 | 创建时间 |
| update\_time | datetime | 否 | 否 | 更新时间 |

（4）物品收藏表，主要包括自增id、物品id和用户的openid，如表4.4。

表 4.4 物品收藏表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **数据类型** | **允许空** | **是否主键** | **说明** |
| id | int(11) | 否 | 是 | 序号 |
| item\_id | int(11) | 否 | 否 | 物品id |
| openid | varchar (30) | 否 | 否 | 共享方id |

（5）物品留言表，主要包括主键id、物品id、评论者openid、父评论id、评论内容、评论时间等等,如表4.5。

表 4.5 物品留言表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **数据类型** | **允许空** | **是否主键** | **说明** |
| id | int(11) | 否 | 是 | 序号 |
| item\_id | int(11) | 否 | 否 | 物品id |
| user\_id | varchar (30) | 否 | 否 | 留言者id |
| content | varchar (255) | 否 | 否 | 评论内容 |
| isdeleted | tinyint | 是 | 否 | 是否被删除 |
| replies\_count | int | 是 | 否 | 回复数 |
| create\_time | datetime | 否 | 否 | 评论时间 |

（6）订单申诉表，主要包括主键id、订单id、申诉理由、申诉图片证明、处理状态、创建时间等等,如表4.6所示。

表 4.6 订单申诉表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **数据类型** | **允许空** | **是否主键** | **说明** |
| id | int(11) | 否 | 是 | 序号 |
| orderId | int(11) | 否 | 否 | 订单id |
| reason | varchar (255) | 否 | 否 | 申诉理由 |
| image\_url | varchar (30) | 是 | 否 | 图片证明 |
| flag | tinyint | 否 | 否 | 处理状态 |
| create\_time | datetime | 是 | 否 | 创建时间 |
| update\_time | datetime | 否 | 否 | 更新时间 |

（7）操作日志表，包括自增id、操作人id、操作时间、操作类名、操作方法、操作参数等等。

表 4.7 操作日志表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **数据类型** | **允许空** | **是否主键** | **说明** |
| id | int(11) | 否 | 是 | 序号 |
| operate\_user | int(11) | 否 | 否 | 操作人id |
| operate\_time | datetime | 否 | 否 | 操作时间 |
| class\_name | varchar (100) | 否 | 否 | 操作类名 |
| method\_name | varchar (100) | 否 | 否 | 操作方法 |
| method\_params | varchar (1000) | 是 | 否 | 方法参数 |
| return\_value | varchar (2000) | 是 | 否 | 返回值 |

## 本章小结

本章主要进行了数据库设计，主要包括概念模型设计和物理模型设计两个方面。在概念模型设计中，通过对系统的需求和功能进行分析，构建了数据库的概念模型，明确了系统中涉及的各种实体以及它们之间的关系。在物理模型设计中，则将概念模型转化为具体的数据库结构，包括数据表的设计、字段的定义、索引的建立等，以满足系统的实际运行需求。

# 系统实现

上述章节主要介绍了系统的相关设计，本章节开始通过相关核心代码和系统运行图介绍相关设计的具体实现，包括系统架构的实现以及系统具体功能模块的实现。

## 开发环境

## 系统架构实现

本系统采用了web端和小程序移动端双平台架构，分别使用了不同框架和架构进行了具体的实现。

### Web端多层架构实现

“ShareHub”物品共享系统的Web端架构的具体实现通过数据Model层、业务逻辑Service层、控制Controller层、持久Dao层和表示View层五层架构来实现，如图6.1项目目录结构。这与MVC（Model-View-Controller）架构有着密切的关联。

在项目目录中，pojo包和service包相当于MVC架构中的Model，负责管理系统中的数据和业务逻辑，包括对类的定义，如用户类、物品类，以及相关功能的逻辑处理，是系统的核心部分。controller包则相当于MVC架构中的Controller，负责调用相应的Service方法来执行业务逻辑，并将处理结果返回给用户，是系统与用户交互的入口。mapper包则同样可归纳于MVC架构中的Model，负责与数据库进行交互，实现了数据的读取、写入和修改等操作，保证了数据的安全性和一致性。

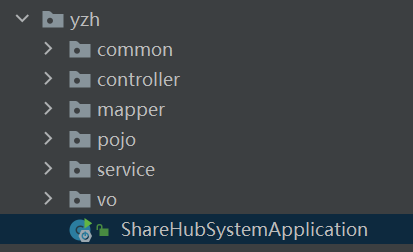


图 6.1 Spring boot项目目录结构

其中MVC中的表示层（View）是通过前端Vue项目来实现的，如图6.2 Vue项目目录结构。在物品共享系统中，针对管理员的前端Vue层负责展示Web端的界面和功能，包括物品管理、用户管理、通知发送、数据分析报告等。 Vue项目中的组件、路由、状态管理等模块，对应了MVC架构中View层的组件、界面控制、状态管理等功能，通过这些模块的配合，实现了Web端的各项功能，为用户提供了友好的界面和便捷的操作体验。

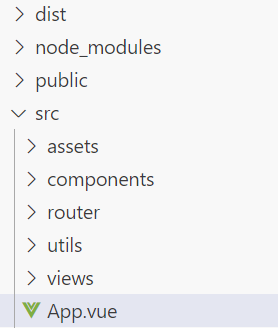


图 6.2 Vue项目目录结构

其他包的介绍，图6.1中的common包，包含了一些用于共享的通用工具类，如文件上传、字符串处理等等，全局异常处理类、常量类等。图6.1中的vo包用于存放视图对象（View Object），它们主要用于在前后端之间传递数据，通常用于前端展示或者接收前端请求时使用。VO对象通常包含了与前端交互的数据字段，与数据库实体类不同，它们更多地关注于展示的需求，而不是数据库存储的结构。VO对象可以根据前端的需求定制不同的字段组合，避免了将整个实体类的数据都暴露给前端，提高了数据的安全性和隐私性。

### 小程序端架构实现

小程序端在架构逻辑上位置与Vue项目相同，都是充当MVC架构中表示层（View）角色。小程序项目目录结构如图6.3所示。首先是components目录，用于存放项目中的自定义组件，可以将页面中的重复部分抽象成组件进行复用。然后是pages目录，其中存放着项目的页面文件，每个页面通常对应一个**.**vue文件，用于展示具体的页面内容和逻辑。static目录，用于存放项目中的静态资源文件，如图片、字体等。assets目录用于存放项目中的公共资源文件，如样式文件、图标等。由于本系统使用了Vuex进行状态管理，所以存在store目录，其中存放着Vuex相关的文件，包括state、mutations、actions等。main.js是uni-app项目的主入口文件，用于初始化Vue实例和加载各种配置和插件。App.vue是应用的根组件，用于定义应用的整体结构和样式。项目的配置信息通常存放在manifest.json和pages.json文件中，分别用于配置项目的基本信息、页面路径、全局样式等，最后是uni.scss全局样式文件，用于定义项目的全局样式，会被所有页面和组件共享。上述清晰的项目目录结构使得开发人员可以快速找到特定功能的代码文件，降低了维护和修改代码的难度。

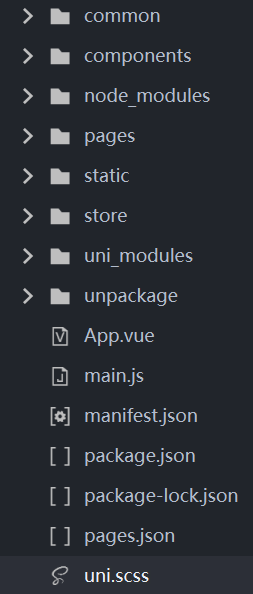


图 6.2 小程序项目目录结构

## 功能模块实现

本章通过将通过相关的核心代码、系统运行图对物品共享平台的Web端和小程序核心功能进行介绍

### Web端功能实现

管理员在Web端可通过账户密码进行登录，如果账户或密码不存在，系统会给出提醒，若账户密码正确则可成功进入管理页面，登录页面如图6.3。

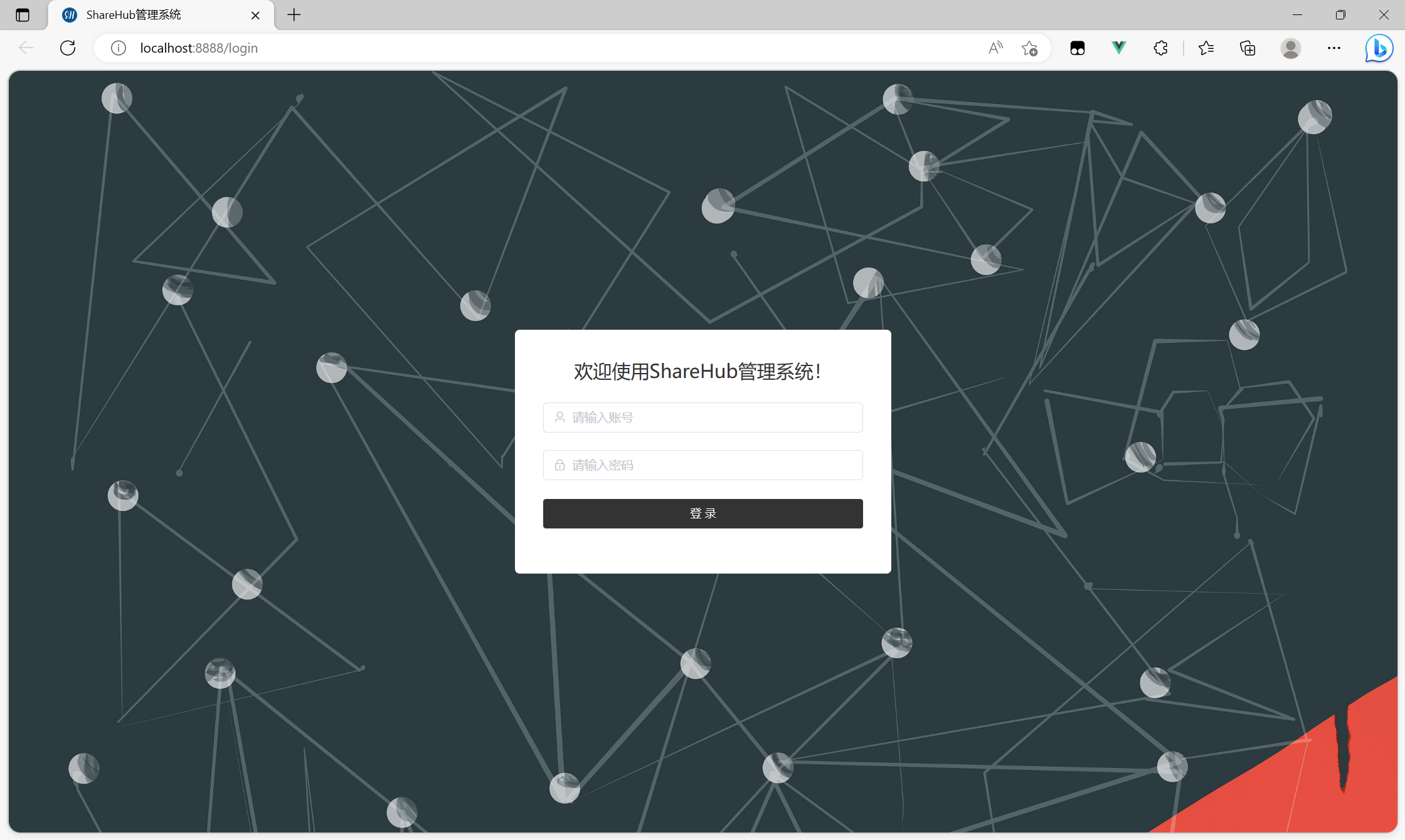


图 6.3 Web端登录界面

在Web端在管理员登录成功进入管理系统主页面后，就可以看到左侧的管理系统功能侧边栏，可以选择不同的功能，如系统首页、数据展示、管理公告信息、管理轮播图信息、以及管理用户信息和物品信息，同时管理员还可以处理用户的一些事务，如申诉或者进行二手商的资历验证等等，如图6.4Web管理系统主页面。

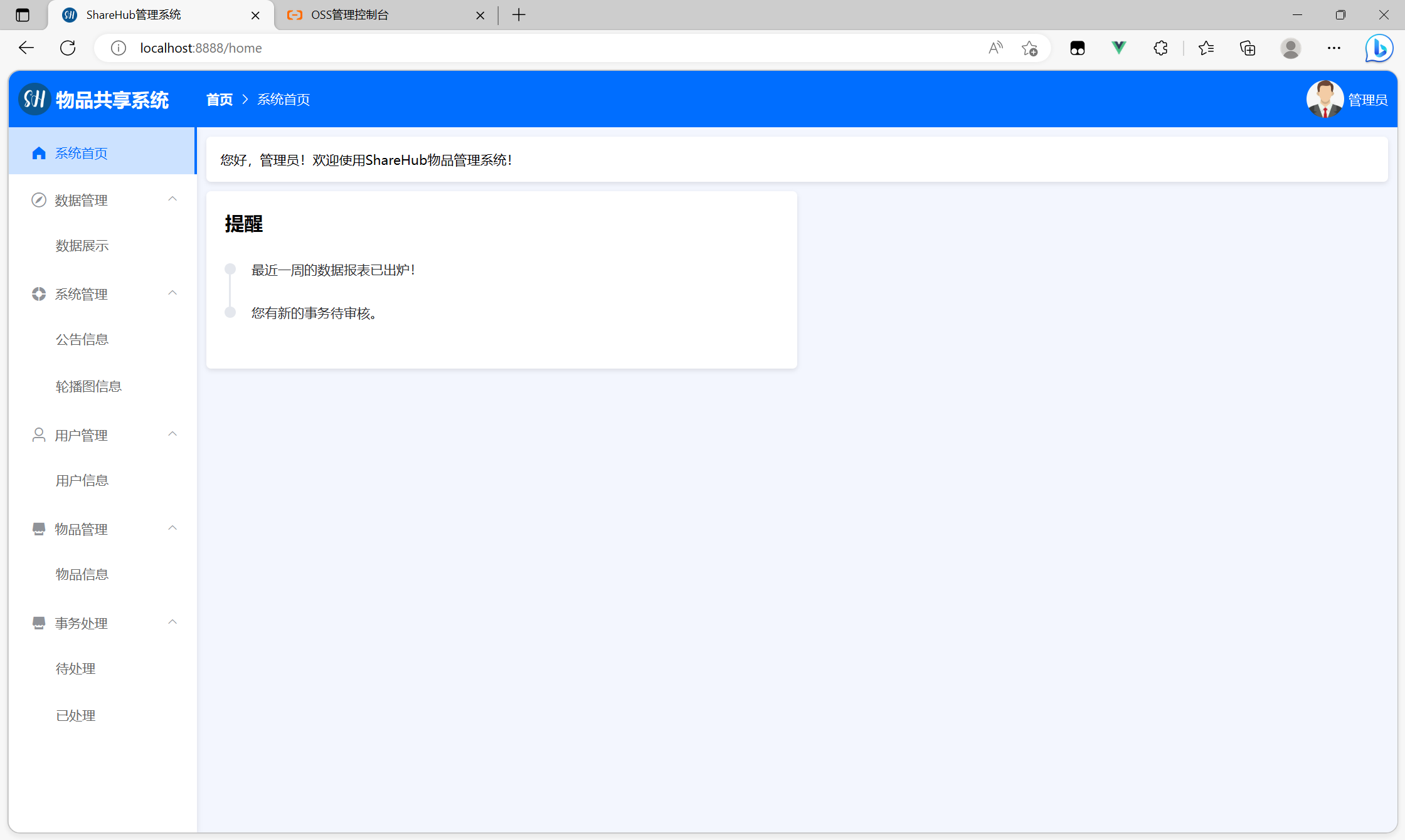


图 6.4 Web管理系统主界面

当管理员点击公告信息时，可查看所有当前正在发布的公告信息，当需要对公告信息进行修改时，管理员可点击编辑并进行修改，同理还可进行新增或删除，如图6.5公告信息管理。管理员对公告信息的操作会同步到用户的小程序端的主界面进行展示。对轮播图管理同理。

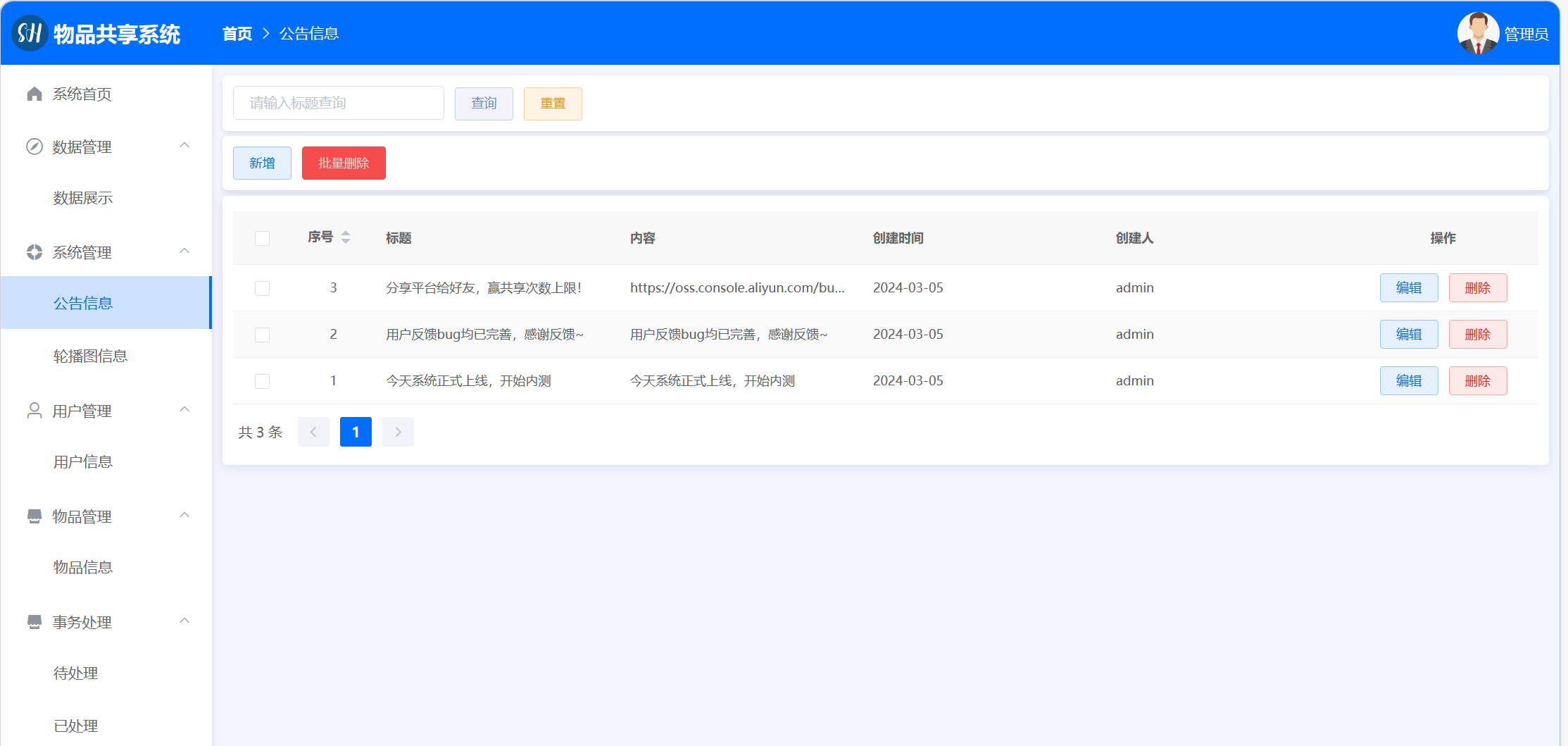


图 6.5 信息管理界面

当管理员点击用户管理时，系统会分页展示所有用户列表，每条数据都展示了用户的用户名、头像、用户角色、用户电话、信誉分和共享数等等，如图6.6用户管理界面。管理员可通过关键字检索想要查找的用户，还可新增或者批量删除用户数据，同时管理员还可编辑修改用户信息。

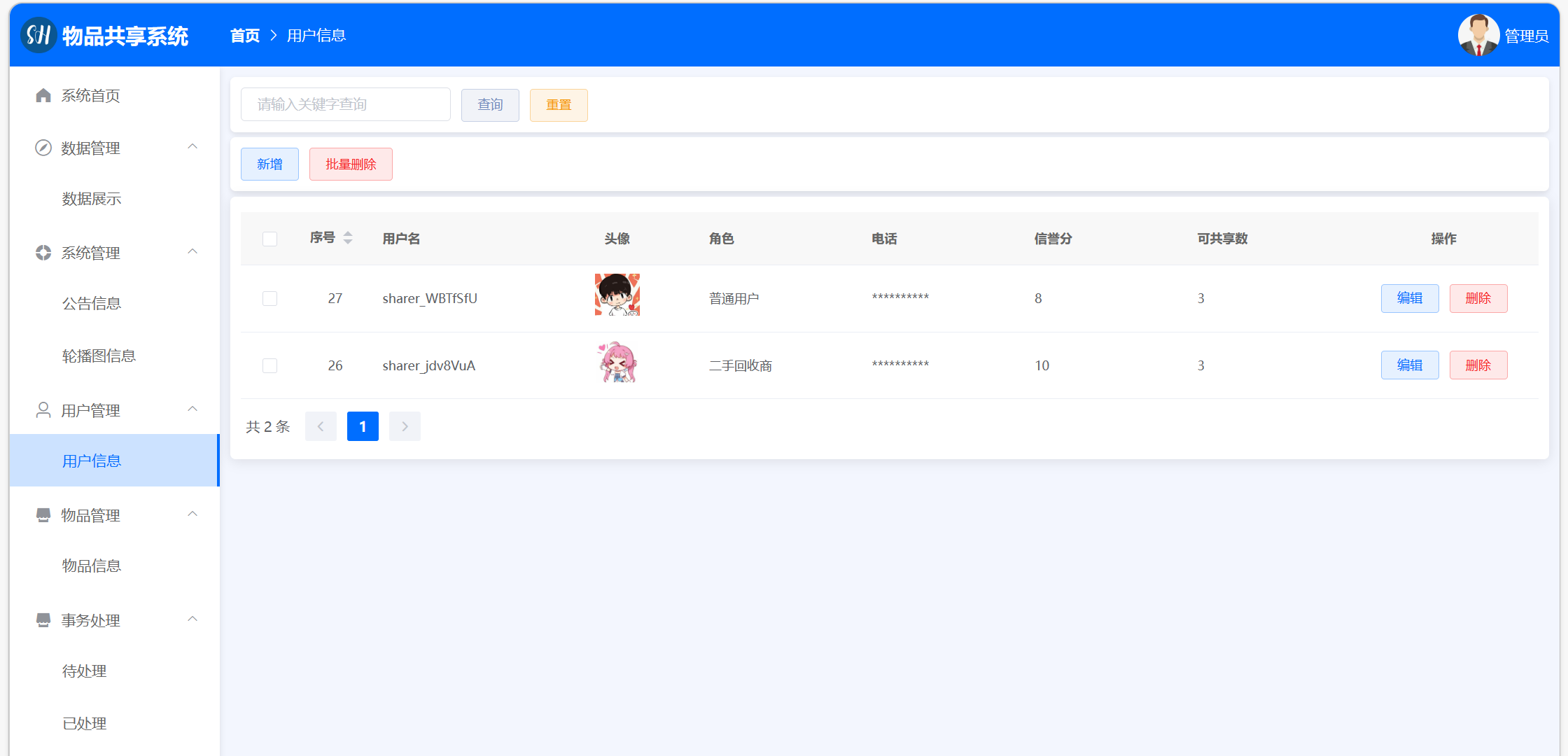


图 6.6 用户管理界面

当管理员选择物品管理时，系统会分页展示所有物品数据，每条物品数据分别展示了物品的物品标题、物品描述、物品图片、物品种类、物品交易模式、物品交付方式、物品状态以及上传时间等信息，如图6.7物品管理界面。管理员可通过关键字检索通过检索物品的标题和描述来查找的物品信息，还可新增或者批量删除物品数据，同时管理员还可编辑修改物品信息。

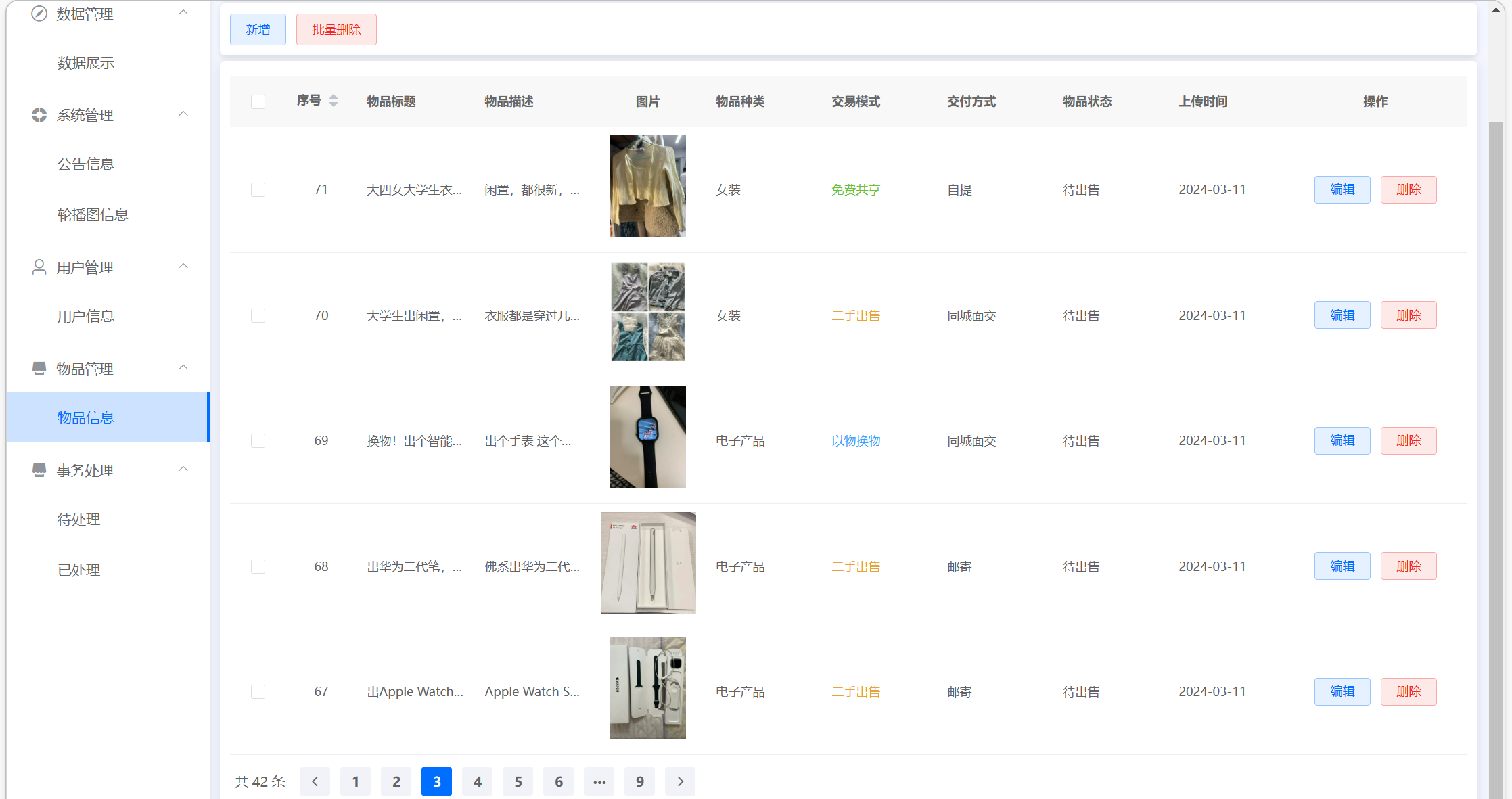


图 6.7 物品管理界面

当管理员点击数据展示功能时，系统会展示三个可视化图像，分别是今日物品交易方式占比饼状图、物品种类交易数前五柱状图以及近七天平台总交易数，如图6.8 数据可视化界面。当管理员将鼠标放置每个可视化图的具体位置时，会展示相关部分的详细数据以供管理员参考。

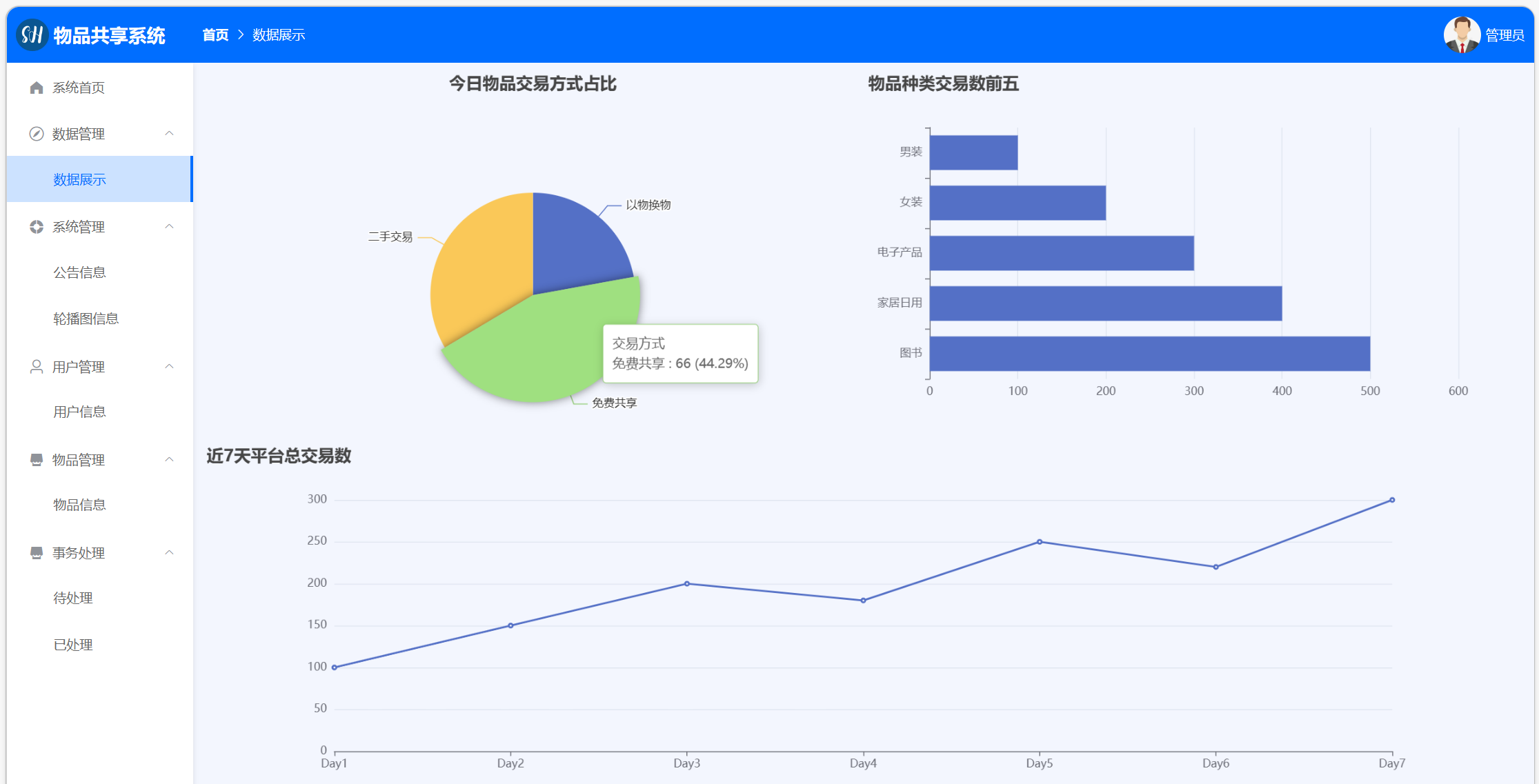


图 6.8 数据可视化界面

### 小程序端功能实现

小程序端是用户供用户使用的，用户在点击小程序后无需登录就可进入到小程序的主页面，如图6.9小程序主界面。该主界面从上到下分别由搜索框、轮播图、通知公告、物品种类Scroll-View、物品交易模式滑动条以及物品展示瀑布流。



图 6.9小程序主界面

其中搜索框是通过对物品的标题和描述进行关键字的模糊搜索，动态SQL如下, 其中的 similarity 列用于计算搜索结果与搜索关键词的相似度，并按照相似度降序排列结果。!!!（去掉）

1. <**select** id="searchItems" resultType="com.yzh.pojo.Item">
2. **SELECT** \*,
3. (CASE **WHEN** item\_title LIKE CONCAT('%', #{keywords}, '%') **THEN** 1 **ELSE** 0 **END** +
4. CASE **WHEN** item\_desc LIKE CONCAT('%', #{keywords}, '%') **THEN** 1 **ELSE** 0 **END**) **AS** similarity
5. **FROM** item
6. <**where**>
7. <if test="mode == 0 or mode == 1 or mode == 2">
8. trade\_mode = #{mode}
9. </if>
10. <if test="keywords != null and keywords != ''">
11. AND (item\_title LIKE CONCAT('%', #{keywords}, '%') OR item\_desc LIKE CONCAT('%', #{keywords}, '%'))
12. </if>
13. </**where**>
14. **ORDER** **BY** similarity **DESC**
15. </**select**>

接着是轮播图和通知栏，都是由管理员来管理和发布的，点击后均可跳转到相关详情页。然后是物品种类框以及物品交易模式框，点击对应的物品种类或交易模式后，可控制瀑布流只展示对应种类或交易模式的物品。最后是展示物品卡片的瀑布流，当滑动到底部时，会自动向后端请求新的物品列表进行展示，如图6.10 物品瀑布流展示。



图 6.9主页物品瀑布流展示

主页的底部的TableBa分为四个部分，分别为首页、发布、消息和我的，当点击发布时，用户需要先进行登录才能发布物品。小程序端的登录和注册分为两种，第一种是传统的账户密码登录，另外一种是更方便的微信一键登录/注册，如图6.10用户登录界面。

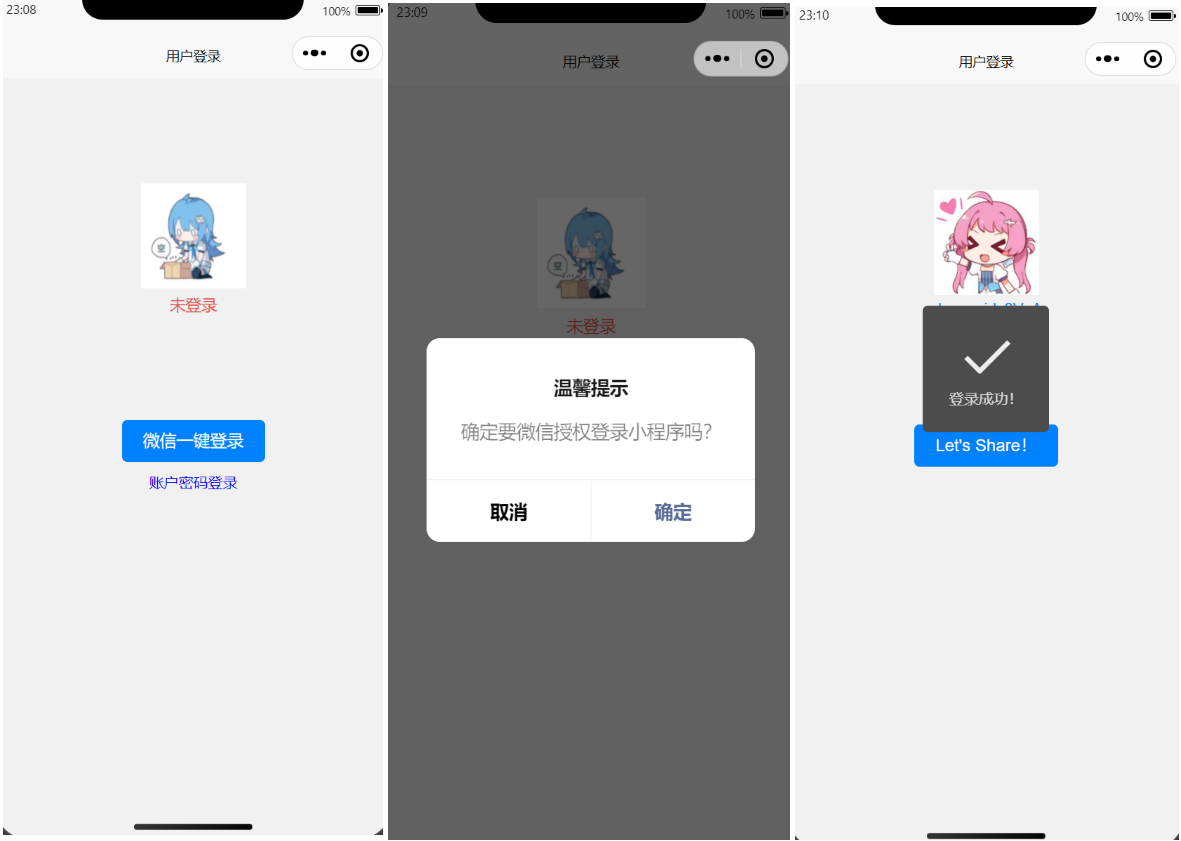


图 6.10用户登录界面

当用户点击发布后，可选择自己想要的发布方式，包括物品共享、以物换物和二手交易。当点击物品共享后，可填写自己物品的详细信息以及图片，其中图片的上传采用是阿里云的OSS云存储，相关代码如下：

1. **public** String upload(MultipartFile file, String flag) **throws** IOException {
2. String endpoint = aliOSSPropertise.getEndpoint();
3. String accessKeyId = aliOSSPropertise.getAccessKeyId();
4. String accessKeySecret = aliOSSPropertise.getAccessKeySecret();
5. String bucketName = aliOSSPropertise.getBucketName();
6. // 获取上传的文件的输入流
7. InputStream inputStream = file.getInputStream();
8. // 使用UUID生成文件名，避免文件覆盖
9. String originalFilename = file.getOriginalFilename();
10. String fileName = UUID.randomUUID().toString() + originalFilename.substring(originalFilename.lastIndexOf("."));
11. OSS ossClient = **new** OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
12. **if** (flag.contains("items")) {
13. //上传文件到 OSS,物品目录
14. fileName = "items/" + fileName;
15. } **else** **if** (flag.contains("users")) {
16. //上传文件到 OSS,物品目录
17. fileName = "users/" + fileName;
18. }
19. ossClient.putObject(bucketName, fileName, inputStream);
20. //文件访问路径
21. String url = endpoint.split("//")[0] + "//" + bucketName + "." + endpoint.split("//")[1] + "/" + fileName;
23. // 关闭ossClient
24. ossClient.shutdown();
25. **return** url;// 把上传到oss的路径返回
26. }

当物品发布成功后，会弹出发布成功的toast，然后返回到选择发布页。物品发布流程如图6.10。

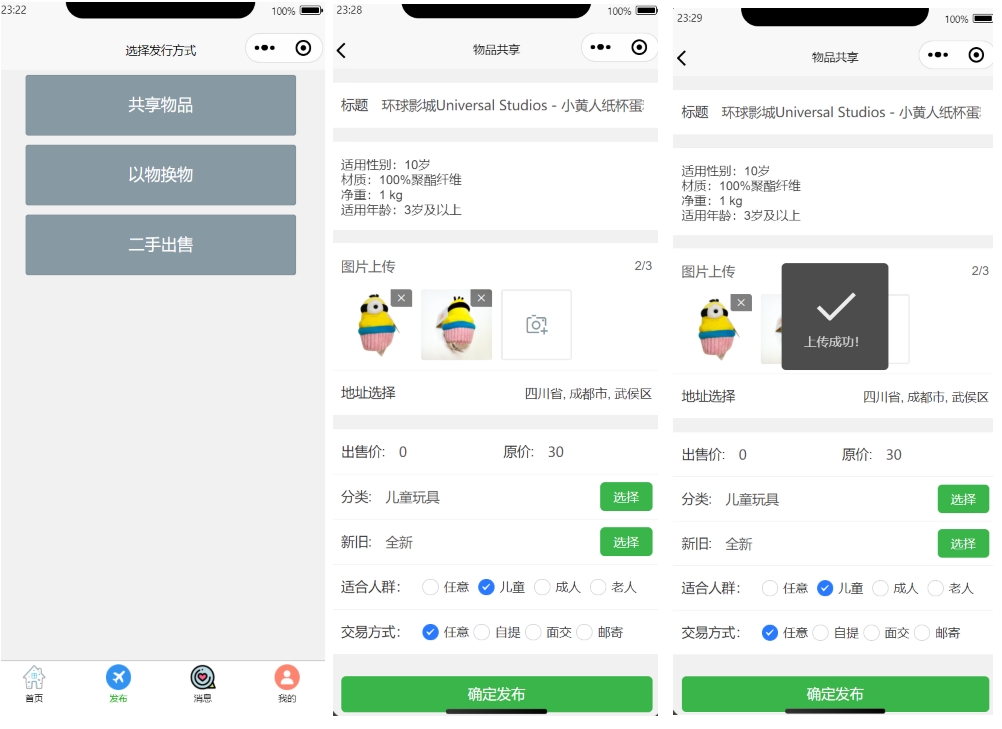


图 6.9物品发布流程

当点击TableBar的第三个按钮消息时，会展示系统的消息模块，消息模块分为三个部分，即互动消息、订单消息、系统消息，如图6.10。当用户收到三类消息中的任意一个时，TableBar中的消息上以及会出现一个红点，来提醒用户进行查看。



图 6.10 小程序消息模块图

点击TableBar的第四个按钮我的，会展示用户的个人中心页面，其中会展示用户头像，点击头像可进一步更改用户资料，同时还会展示用户的信誉等级，以及当前可共享物品数，接下来就是关于用户的一些列功能列表，包括查看正在上架的物品、已经送出的物品、已经领到的物品以及收藏的物品等等，如图6.11。

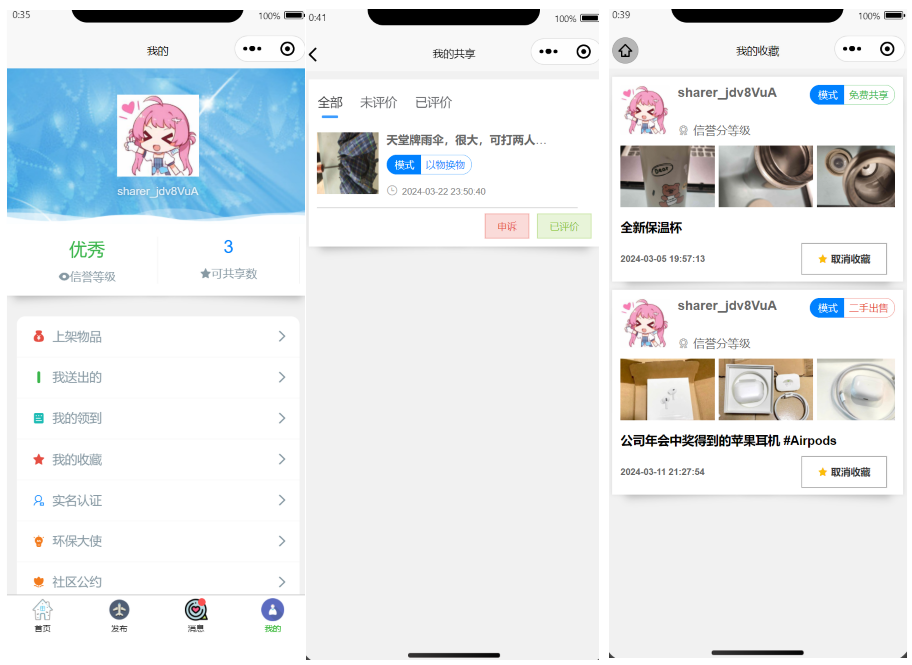


图 6.11 个人中心及相关功能

## 关键算法

本系统采用了基于物品内容的推荐算法（Content-Based Recommendations），通过分析物品的特征和用户的兴趣，实现更精准的物品推荐。基于TF-IDF的向量化方法，将物品的的类型和用户偏好转化为特征向量；然后通过计算它们之间的相似度，使用余弦相似度来衡量相关性；接下来，通过相似的大小来提高相关物品的推荐优先级[13]。

该算法是最早被使用的算法，年代久远，但至今仍被广泛使用效果良好。相对于当前较为流行的协同过滤推荐算法（Collaborative Filtering），使用基于内容的推荐算法可避免较为严重的冷启动问题。在项目创建早起缺乏用户数据时使用，仍具有较好的推荐性能。在系统上线一段时间后，拥有一定量的用户数据时，再使用协同过滤算法，才能起到较好的推荐效果。所以在本系统的创建早期，先使用基于内容的推荐算法进行过渡，是个恰当的选择。

基于内容的推荐算法的算法逻辑是首先要提取物品的特征信息，比如共享平台上的物品的特征标签可以包括服饰类、生活类、数码影音、儿童玩具等等与物品种类相关的标签，以及还有物品的交易方式（免费共享、以物换物、二手交易）相关的特征标签，还有可以含有物品的新旧程度、使用年龄等等相关标签。本系统提取了28个物品标签来刻画用户的偏好度。

然后使用提取的标签来表示物品以及用户信息，将提取好的所用标签创建为数据库表user\_tags\_score保证数据的持久化存储。后端可查询该表的每条数据然后将其转化为向量，方便后续的计算。

当新用户在平台完成注册后，系统会邀请用户填写一个‘选择偏好’，如图6.12 偏好选项界面，然后系统会根于用户选择的偏好选择，为该用户在数据库中的偏好表初始化一个偏好值，进而进行初步的推荐。



图 6.12 用户偏好选项界面

推荐算法的核心是计算用户数据库表的偏好值所得出的偏好向量与物品的属性向量的相似度来进一步推荐，其中物品的属性向量可通过物品所具有的标签来进行赋值，每个物品在发布后其属性向量就已经确定了，不会再发生变化，计算相似度核心的代码如下：

1. **public** **class** CosineSimilarity {
2. // 计算两个向量之间的余弦相似度
3. **public** **static** Double cosineSimilarity(Integer[] vectorA, Integer[] vectorB) {
4. // 计算向量的点积
5. Double dotProduct = dotProduct(vectorA, vectorB);
6. // 计算向量A的模
7. Double magnitudeA = magnitude(vectorA);
8. // 计算向量B的模
9. Double magnitudeB = magnitude(vectorB);
10. // 计算余弦相似度
11. **return** dotProduct / (magnitudeA \* magnitudeB);
12. }
13. // 计算向量的点积
14. **private** **static** Double dotProduct(Integer[] vectorA, Integer[] vectorB) {
15. **double** dotProduct = 0;
16. **for** (**int** i = 0; i < vectorA.length; i++) {
17. // 对应维度上的元素相乘并累加
18. dotProduct += vectorA[i] \* vectorB[i];
19. }
20. **return** dotProduct;
21. }
22. // 计算向量的模
23. **private** **static** Double magnitude(Integer[] vector) {
24. **double** sumSquared = 0;
25. **for** (**double** v : vector) {
26. // 对向量的每个元素求平方并累加
27. sumSquared += v \* v;
28. }
29. // 对平方和取平方根，即得到模
30. **return** Math.sqrt(sumSquared);
31. }
32. }

最后，后端会优先返回给前端相似度最近的几个物品，以达到对用户的推荐效果。

除此之外，用户的一系列行为都会影响到其偏好表中的数值，比如用户查看了某个物品、完成了订单、收藏了物品等等都会增加用户对相关物品标签的偏好值，若用户在主页长按某个物品选择不喜欢该物品，则会降低对该物品的偏好值，这些操作都会动态影响到后续系统对用户的物品推荐。

## 本章小结

本章主要介绍了系统架构的设计和实现、功能模块的开发以及关键算法的实现。详细介绍了Web端和小程序端的多层架构设计与相关功能的具体实现。此外，本章介绍的推荐算法同时对其核心代码进行了分析，推荐系统可对用户进行个性化推荐，提高用户体验。本章工作为物品共享平台的正常运行奠定了基础。

# 系统测试

本章是系统开发的最后一道工序，即系统测试。通过系统测试，可以评估系统的功能、性能、安全性等方面，确保软件能够满足用户需求并达到预期的质量标准。

## 测试目的

软件测试技术能够快速准确的找到计算机软件中存在的各种问题，以帮助软件开发人员能够对这些问题进行集中处理和解决，保证交给用户使用的软件，都是性能、功能、安全达标的软件[14]。通过系统测试，可以评估系统的稳定性、可靠性和可用性，确保系统能够满足用户需求并且达到预期的业务目标。系统测试还有助于提高系统的质量，减少在生产环境中出现的故障和风险，从而保障系统的正常运行和用户体验。

## 测试方法

系统测试包括多种测试方法，比如黑盒测试、白盒测试、灰盒测试、易用性测试、负载/压力测试、兼容性测试、回归测试和边界值测试等，其中黑盒测试，又被称为功能测试，是一种通过检验每个功能是否正常运作的测试方法。在进行这种测试时，将软件视作一个不可打开的黑盒子，而不考虑其内部结构和工作原理。测试重点在于通过程序的外部接口进行检查，以确认程序功能是否按照需求规格书的规定正常工作，能否正确接收输入数据并生成符合预期的输出信息[15]。

在物品共享系统中，测试员选择了采用黑盒测试和兼容性测试方法。黑盒测试聚焦于系统的外部行为，通过功能测试来验证系统的各项功能是否正常运行，而不需要了解系统的内部实现细节。这种方法可以帮助测试员发现系统中潜在的功能缺陷和错误，确保用户能够顺利地使用系统。另一方面，兼容性测试则是为了验证系统在不同环境下的正常运行情况。测试员关注的是系统在不同屏幕尺寸下页面显示是否正常、功能是否正常，以确保用户在不同设备上都能够良好地体验系统。

## 测试内容和结果

### 功能测试

### 兼容性测试

在系统测试中，兼容性测试是为了确保软件在各种不同环境下的正常运行和良好用户体验而进行的关键测试环节。兼容性测试的范围广泛，涵盖了浏览器、设备、分辨率和网络环境等多个方面。在操作系统方面，测试团队需要验证软件在不同操作系统版本下的兼容性。此外，兼容性测试还包括验证软件在不同浏览器中的兼容性，因为不同浏览器对网页的解析方式和支持的技术可能会有所不同，如HTML5、CSS3、JavaScript等。因此，测试人员需要确保软件在常见的浏览器中（如Chrome、Firefox、Safari、Edge等）都能够正确显示和运行。另外，兼容性测试还考虑了不同设备上的兼容性，包括桌面设备、笔记本电脑、平板电脑和智能手机等，以及在不同分辨率下的页面布局和显示效果。这意味着测试团队需要测试软件在各种设备上的显示效果，并确保在不同分辨率下都能够提供良好的用户体验。表7.8是 Web 端在主流浏览器下的兼容性测试结果展示。

表7.8 Web端浏览器兼容测试

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 浏览器 | 预期结果 | 实际结果 |
| 1 | Microsoft Edge | 页面展示与功能均正常 | 与预期相符 |
| 2 | Google Chrome | 页面展示与功能均正常 | 与预期相符 |
| 3 | 火狐浏览器 | 页面展示与功能均正常 | 与预期相符 |
| 4 | Opera | 页面展示与功能均正常 | 与预期相符 |
| 5 | IE浏览器 | 页面展示与功能均正常 | 与预期相符 |

表7.9是小程序端在不同设备下的兼容性测试结果展示。

表7.9 小程序端设备兼容测试

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 浏览器 | 预期结果 | 实际结果 |
| 1 | HUAWEI Mate 60 Pro | 页面展示与功能均正常 | 与预期相符 |
| 2 | Xiaomi 13 | 页面展示与功能均正常 | 与预期相符 |
| 3 | iPhone X | 页面展示与功能均正常 | 与预期相符 |
| 4 | Galaxy | 页面展示与功能均正常 | 与预期相符 |
| 5 | iPad | 页面展示与功能均正常 | 与预期相符 |

1. 小程序端用户登录测试

用户登录测试，包括测试微信一键登录以及账户密码登录的成功和失败的情况，如表7.1小程序端用户登录测试。

表7.1小程序端用户登录测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 测试用例 | 测试记录 | 预测结果 | 实际结果 |
| 1 | 用户微信一键登录 | 用户同意授权登录 | 提示登录成功 | 和预期相符 |
| 2 | 用户账户密码登录成功 | 填写正确的用户名和密码 | 提示登录成功 | 和预期相符 |
| 3 | 用户账户密码登录失败 | 填写用户名和错误密码 | 提示密码错误 | 和预期相符 |

(2)用户发布物品测试

用户发布物品主测试，主要是测试用户点击发布后，系统是否会弹出成功的提示框，然后用户在个人中心的已发布物品是否可以看到刚刚发布的物品信息，如表7.2 物品发布测试。

表7.2 物品发布测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 测试用例 | 测试记录 | 预测结果 | 实际结果 |
| 1 | 发布共享物品 | 用户选择物品共享交易模式，填写物品信息并发布 | 提示上传成功，发布列表出现物品 | 和预期相符 |
| 2 | 发布以物易物物品 | 用户选择以物易物模式，填写物品信息并发布 | 提示上传成功，发布列表出现物品 | 和预期相符 |
| 3 | 发布二手交易物品 | 用户选择二手交易模式，填写物品信息并发布 | 提示上传成功，发布列表出现物品 | 和预期相符 |

（3）查看物品详情及功能测试

用户点击物品查看详情测试，用户在主页面选择想要的物品，点击后可查看该物品的详细信息，测试该物品的相关信息是否能够正确显示，以及对应该物品的功能按钮是否可以点击，如表7.3物品详情展示测试。

表7.3 物品发布测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 测试用例 | 测试记录 | 预测结果 | 实际结果 |
| 1 | 查看物品详情 | 用户选择想要的物品并进行点击 | 物品相关信息及其图片均正常显示 | 和预期相符 |
| 2 | 进行物品收藏 | 在物品详情页点击收藏按钮 | 提示收藏成功，再次点击可取消收藏 | 和预期相符 |
| 3 | 申请物品共享 | 在物品详情页点击申请共享按钮，填写共享表单 | 提示申请成功，且物主收到订单消息 | 和预期相符 |

（4）修改用户个人信息测试

用户修改个人信息测试，主要测试当用户点击个人头像时，是否会弹出个人信息修改框，以及个人信息框中是否已经展示了当前用户现有信息内容。当用户修改好相关信息后，点击确认修改按钮后，用户信息是否成功被修改。

表7.4 物品发布测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 测试用例 | 测试记录 | 预测结果 | 实际结果 |
| 1 | 查看物品详情 | 用户选择想要的物品并进行点击 | 物品相关信息及其图片均正常显示 | 和预期相符 |
| 2 | 进行物品收藏 | 在物品详情页点击收藏按钮 | 提示收藏成功，再次点击可取消收藏 | 和预期相符 |
| 3 | 申请物品共享 | 在物品详情页点击申请共享按钮，填写共享表单 | 提示申请成功，且物主收到订单消息 | 和预期相符 |

（5）关键字模糊搜索查询测试

关键字迷糊查询测试，具体过程是在小程序主页的搜索框，搜选相关关键词，是否会查出包含相关关键词的物品。测试结果如表7.5。

表7.5 模糊查询测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 测试用例 | 测试记录 | 预测结果 | 实际结果 |
| 1 | 有效关键字查询 | 在搜索框中输入华为，并进行搜索 | 搜索后只展示华为品牌物品 | 和预期相符 |
| 2 | 无效关键字查询 | 在搜索框中输入乱码，并进行搜索 | 搜索后展示物品页为空 | 和预期相符 |

（6）用户订单申诉功能测试

订单申诉功能测试，主要是测试当用户对不满意的订单进行申诉，点击申诉按钮，申诉模态框是否能够正常展示；当提交申诉表后，管理员端是否能够查询到该申诉，并进行处理，测试结果如表7.6订单申诉功能测试。

表7.6 订单申诉功能测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 测试用例 | 测试记录 | 预测结果 | 实际结果 |
| 1 | 展示申诉模态框 | 在订单列表页点击订单申诉按钮 | 可正常弹出申诉模态框 | 和预期相符 |
| 2 | 提交申诉表单 | 填写完申诉表单后，点击提交按钮 | 提示成功，且管理员端出现申诉请求 | 和预期相符 |

（7）管理员端数据可视化功能测试

可视化功能测试主要是查看管理员端的是否能够正确展示相关可视化图，以及数据是否真确，具体测试结果如表7.7可视化功能测试。

表7.7 可视化功能测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 测试用例 | 测试记录 | 预测结果 | 实际结果 |
| 1 | 可视化图展示 | 测试员登录后点击数据展示按钮 | 可正常展示各个数据图 | 和预期相符 |
| 2 | 各可视化图详细数据展示 | 测试员将鼠标放置在可视化图的具体模块 | 该可视化图展示该部分的详细数据 | 和预期相符 |

## 测试结论

经过系统测试的功能测试和兼容性测试，物品共享平台的Web端和小程序端均表现出良好的稳定性和可靠性，能够正常运行。功能测试验证了平台各项功能的正确性和完整性，包括模糊查询、个人信息管理、查看物品详情以及数据分析报告等，结果显示所有功能均能够按照预期的方式运行，用户可以顺利完成各项操作。

同时，兼容性测试也得出了积极的结论，表明平台在不同浏览器和不同设备上均能够正常显示和操作，确保了用户在不同环境下的良好体验。

## 本章小结

本章主要介绍了系统测试的目的与常用方法，且对本系统进行了功能测试和系统兼容性测试。同时使用表格展示各个功能的测试结果，最后进行了系统测试总结，可以得出结论：物品共享平台的Web端和小程序端在功能性和兼容性方面均表现出良好的性能，用户可以安心使用。

结 论

本次毕业项目的设计是为了打造一个全新的物品共享平台，致力于解决用户闲置物品处理难的问题，为用户提供便捷、高效的共享服务。该平台包含了 Web 端和小程序端，采用双平台系统结构组成了整个项目所有功能。通过这个项目，用户可以方便地共享、交换、出售闲置物品，实现资源共享、减少浪费的目标。

本在项目开发初期，首先对相关文献进行了深入研究，发现了国内缺乏以共享为主要理念的线上平台，从而明确了物品共享平台的选题意义及目的，同时通过对国外相关共享平台案例的分析，了解了当前物品共享领域的发展趋势和存在的问题。接着，又详细介绍了系统的开发运行环境，包括后端开发环境和前端开发环境，以及相关软件和框架的安装和使用方式，为后续系统设计和开发提供了必要的准备工作。

在系统分析阶段，对系统的功能需求进行了深入分析，以确定系统需要实现的关键功能。这一过程包括对管理员和用户在平台上的各项操作需求进行详细梳理，并将其具体内容细化到不同的功能模块中。通过细致的功能分析，可以确保系统能够满足用户在物品共享平台上的各种需求，并提供便捷、高效的使用体验。除此之外，还利用用例图清晰地展示了系统的功能结构和用户交互流程，以帮助开发团队全面理解和把握系统的功能设计。接着，对系统的非功能性需求进行了分析，包括市场需求、安全需求以及性能需求等方面。通过对这些需求的梳理和确认，确保了系统能够在不同的环境和场景下稳定运行，并提供安全可靠、性能高效的服务。

接着进行了系统设计，将重点分为系统架构设计、功能模块设计和非功能模块设计三个方面。系统架构设计涵盖了总体架构设计、逻辑架构设计和物理架构设计，以全面规划系统的结构框架，从宏观到微观考虑了系统的组织架构和技术实现。功能模块设计针对Web端和小程序端进行，明确了各个模块的功能和相互关系，为后续的开发工作提供了清晰的目标和方向。而非功能模块设计主要关注了安全设计、性能保障设计和界面设计等方面，通过多项措施确保了系统的安全性、性能和用户体验，为系统的稳定运行和用户满意度提供了保障。窗体顶端

然后是数据库设计，分为概念模型设计和物理模型设计两个主要方面。在概念模型设计阶段，根据系统需求和功能分析，构建了数据库的概念模型，明确了系统中各种实体及其相互关系。而物理模型设计则将概念模型转化为具体的数据库结构，包括数据表设计、字段定义和索引建立等，以满足系统的实际运行需求。通过这一设计过程，数据库的结构和组织得到了合理规划和设计，为系统数据的存储、管理和操作提供了有效支持。

接着是关键步骤系统实现，包括系统架构、功能模块和关键算法的实现。在系统架构实现方面，针对Web端和小程序端分别设计了多层架构，确保了系统在不同平台下的稳定性和可扩展性。同时也实现了前后端的分离，提高了系统的灵活性和可维护性。功能模块的实现详细描述了Web端和小程序端各个功能模块的实现过程，包括用户管理、数据可视化和消息通知等，确保了系统功能的完整性和实用性。此外，在关键算法方面，对系统中的重要推荐算法算法进行了介绍和演示，为系统的核心功能提供了有效支持。以上功能的实施，使得物品共享平台的各项功能得到了有效的完善。

最后的步骤是系统测试，对系统进行了相关核心功能的测试以及在不同环境和设备下的兼容性的测试。窗体顶端通过使用表格展示各项功能测试结果，全面评估了系统的性能表现。最终，通过系统测试的总结，可以确定物品共享平台的Web端和小程序端在功能性和兼容性方面均表现出色，均达到了预期要求。

窗体顶端

参考文献

1. Jie Z ,Nan L ,Hong Z , et al.Role of sharing economy in energy transition and sustainable economic development in China[J].Journal of Innovation Knowledge,2023,8
2. 董成惠.共享经济:理论与现实[J].广东财经大学学报,2016,31(05):4-15.
3. 马强.共享经济在我国的发展现状、瓶颈及对策[J].现代经济探讨,2016(10):20-24.DOI:10.13891/j.cnki.mer.2016.10.004.
4. 何流. 共享经济背景下闲置物品共享平台的服务设计研究[D]. 景德镇陶瓷大学,2023.
5. 杨德艳,闫玫辰.二手平台运营存在问题分析与应对策略——以H企业为例[J].物流科技,2024,47(04):22-26.DOI:10.13714/j.cnki.1002-3100.2024.04.005.
6. 庞嵩昊,李盈,赵艺,等.基于Vue和SpringBoot前后端分离的宠物服务系统的设计与实现[J].电脑知识与技术,2023,19(21):42-45.DOI:10.14004/j.cnki.ckt.2023.1068.
7. Deqing L , Honghui M , Yi S ,et al.ECharts: A declarative framework for rapid construction of web-based visualization[J].Visual Informatics, 2018, 2:136-146.DOI:10.1016/j.visinf.2018.04.011.
8. 朱志慧,蔡洁.基于SpringBoot+Vue+Uni-app框架的校园失物招领系统[J].电子技术与软件工程,2022,(17):62-65.
9. Alan R ,Yobani M ,Samantha J , et al.inDev: A software to generate an MVC architecture based on the ER model[J].Computer Applications in Engineering Education,2021,30(1):259-274.
10. 尹强. 基于本体的MVC架构模式识别及重构技术[D].东南大学,2020.DOI:10.27014/d.cnki.gdnau.2019.003179.
11. 王星元. 人机交互软件界面设计的重要性探讨[J]. 美与时代（上旬刊）,2021(5):99-101.
12. 程雪松. 数据库设计探析[J]. 科技传播,2015(8):64-65.
13. 王宇哲.基于内容的电影推荐算法研究[J].信息系统工程,2023(12):117-120.
14. 张雅东,张清淘.软件工程中软件有关测试技术的思考[J].石河子科技,2022(01):17-19.
15. 张玮. 浅析白盒测试和黑盒测试[J]. 无线互联科技,2013(6):136-136. DOI:10.3969/j.issn.1672-6944.2013.06.102.

致 谢

大学四年匆匆而过，转眼间即将告别，这段时光是人生中最美好、自由度最高的四年。在这段时光里，我不仅积累了丰富的感悟和知识，还结识了许多优秀的师友。在此，我要由衷感谢母校对我四年来的培养，感谢在我成长路上给予帮助和指导的人们。

回顾紧张而充实的毕业论文设计过程，我首先要特别感谢聂莉莎老师的悉心指导和支持。她给予我宝贵的建议和启发，让我对项目和论文的思路和结构有了更清晰的认识。

其次，我要深深感谢我的家人。在这四年中，他们给予我不同视角的人生体验和领悟，永远是我坚强的后盾和无尽的动力。他们的爱、关心和支持让我倍感温暖，我会永远珍惜我们之间的每一次相处和交流。

最后，我向参与我的论文审稿和答辩的各位老师致以最诚挚的谢意。