

**毕 业 设 计**



基于SpringBoot的智慧社区管理系统设计与实现

学 院：软件学院

专 业：计算机科学与技术

学生姓名：莫振立

学生学号：2019020517

指导教师：张宏　教授

二○二三年六月

# 摘　　要

随着 ‎经济社会 ‎的快速发 ‎展,社区 ‎工作面临 ‎着许多情 ‎况新问题 ‎,对社区 ‎传统工作 ‎模式提出 ‎了挑战, ‎客观要求 ‎加快推进 ‎社区工作 ‎方式的变 ‎革。如成 ‎都龙泉驿 ‎区悦龙社 ‎区建设智 ‎慧公园社 ‎区,创建 ‎开放创新 ‎的智慧社 ‎区,营造 ‎多元生活 ‎场景,构 ‎建配套均 ‎衡的共享 ‎社区,探 ‎索社区治 ‎理模式的 ‎升级有助 ‎于加快当 ‎前社区特 ‎别是城市 ‎社区的健 ‎康快速发 ‎展[8] ‎。智慧社 ‎区是社区 ‎物业发展 ‎的必然趋 ‎势，是物 ‎业管理的 ‎一种新理 ‎念，是新 ‎形势下社 ‎会管理创 ‎新的一种 ‎新模式。 ‎面对智慧 ‎社区的繁 ‎荣发展， ‎传统下的 ‎社区管理 ‎服务已经 ‎难以满足 ‎人们的便 ‎捷、安全 ‎、多样化 ‎管理随着 ‎人工智能 ‎、大数据 ‎、互联网 ‎等高新技 ‎术的发展 ‎，物业管 ‎理企业先 ‎后试水互 ‎联网。

智慧社区管 ‎理系统平 ‎台采用B ‎/S模式 ‎，分为手 ‎机App ‎端和后台 ‎PC端的 ‎特点，促 ‎进了智慧 ‎社区信息 ‎平台发展 ‎。与传统 ‎社区相比 ‎，用户可 ‎以知道更 ‎多的社区 ‎信息。

本系统使用 ‎开发技术 ‎Spri ‎ngBo ‎ot、V ‎ue、u ‎ni-a ‎pp、M ‎ybat ‎is-P ‎lus， ‎Java ‎作为系统 ‎的开发语 ‎言，My ‎SQL作 ‎为后台数 ‎据库。实 ‎现了房屋 ‎管理、业 ‎主用户管 ‎理、员工 ‎管理、收 ‎费管理、 ‎活动管理 ‎、车辆管 ‎理、社区 ‎论坛、系 ‎统管理设 ‎置等功能 ‎的智慧社 ‎区管理系 ‎统。在设 ‎计过程中 ‎，充分保 ‎证了系统 ‎代码的良 ‎好可读性 ‎、实用性 ‎、易扩展 ‎性、通用 ‎性、便于 ‎后期维护 ‎、操作方 ‎便以及页 ‎面简洁等 ‎特点。

关键词：智慧社区；B/S；SpringBoot；Vue；Java

# ABSTRACT

With the rapid development of economy and society, community work is faced with many new problems, which poses challenges to the traditional work mode of community, and objectively requires to accelerate the reform of community work mode. For example, Yuelong Community in Longquanyi District of Chengdu will build a smart park community, create an open and innovative smart community, create diverse life scenes, build a supporting and balanced shared community, and explore the upgrading of community governance mode will help to accelerate the healthy and rapid development of the current community, especially the urban community [8]. Smart community is the inevitable trend of community property development, a new concept of property management, and a new mode of social management innovation under the new situation. In the face of the prosperity and development of smart communities, the traditional community management services have been difficult to meet people's convenient, safe and diversified management. With the development of artificial intelligence, big data, Internet and other high and new technologies, property management enterprises have successively tested the waters of the Internet. The smart community management system platform adopts the B / S mode, which is divided into mobile App terminal and background PC terminal, and promotes the development of the smart community information platform. Compared with traditional communities, users can know more community information. The system uses the development technology SpringBoot, Vue, uni-app, Mybatis-Plus, Java as the development language of the system, and MySQL as the background database. It realizes the intelligent community management system with the functions of housing management, owner and user management, employee management, charge management, activity management, vehicle management, community forum, system management setting and so on. In the design process, fully ensure the system code good readability, practicability, easy to expand, versatility, easy to maintain, convenient operation and simple page characteristics.

**Key words:** Smart community; B / S; SpringBoot; Vue; Java

目 录

[摘要 I](#_Toc135920195)

[ABSTRACT II](#_Toc135920196)

[第1章　绪论 1](#_Toc135920197)

[1.1 课题背景及意义 1](#_Toc135920198)

[1.1.1 研究背景 1](#_Toc135920199)

[1.1.2 研究意义 1](#_Toc135920200)

[1.2 国内外研究现状 2](#_Toc135920201)

[1.3 智慧社区发展趋势 2](#_Toc135920202)

[1.4论文结构 3](#_Toc135920203)

[第2章　可行性分析 5](#_Toc135920204)

[2.1社会可行性分析 5](#_Toc135920205)

[2.2技术可行性分析 5](#_Toc135920206)

[2.3操作可行性分析 5](#_Toc135920207)

[2.4系统功能需求分析 5](#_Toc135920208)

[2.5开发环境介绍 6](#_Toc135920209)

[2.5.1 Vue和Vue-Cli 6](#_Toc135920210)

[2.5.2 SpringBoot 7](#_Toc135920211)

[2.5.3 MVC 7](#_Toc135920212)

[2.5.4 MyBatis-Plus 8](#_Toc135920213)

[2.5.5 Uni-app 8](#_Toc135920214)

[2.5.6 MySQL数据库 9](#_Toc135920215)

[2.5.7 前后端分离 9](#_Toc135920216)

[2.5.8 系统运行环境 10](#_Toc135920217)

[2.6本章小结 11](#_Toc135920218)

[第3章 系统架构设计 13](#_Toc135920219)

[3.1 系统结构设计 13](#_Toc135920220)

[3.2 系统流程设计 14](#_Toc135920221)

[3.3 系统安全设计 15](#_Toc135920222)

[3.4 系统性能设计 16](#_Toc135920223)

[3.5本章小结 17](#_Toc135920224)

[第4章 核心模块实现 18](#_Toc135920225)

[4.1 登录与注册模块 18](#_Toc135920226)

[4.2 认证权限控制 25](#_Toc135920227)

[4.3 通知公告管理 27](#_Toc135920228)

[4.4本章小结 31](#_Toc135920229)

[第 5 章 数据库设计 32](#_Toc135920230)

[5.1 数据库规范 32](#_Toc135920231)

[5.2 数据表设计 32](#_Toc135920232)

[5.3本章小结 38](#_Toc135920233)

[第6章　系统测试 40](#_Toc135920234)

[6.1 测试目的 40](#_Toc135920235)

[6.2 测试方法 40](#_Toc135920236)

[6.2.1 黑盒测试 41](#_Toc135920237)

[6.2.2 白盒测试 41](#_Toc135920238)

[6.2.3 模块测试 41](#_Toc135920239)

[6.2.4 部分测试用例 42](#_Toc135920240)

[6.3本章小结 43](#_Toc135920241)

[结论 44](#_Toc135920242)

[参考文献 44](#_Toc135920243)

[致谢 46](#_Toc135920244)

·

# 第1章　绪　　论

互联网+改 ‎变着我们 ‎的生活， ‎在传统的 ‎社区服务 ‎领域，在 ‎服务业主 ‎、提升社 ‎区服务质 ‎量方面， ‎如何与互 ‎联网融合 ‎?顶级互 ‎联网技术 ‎团队携手 ‎小帮住， ‎走进社区 ‎，与物业 ‎公司共同 ‎开启一场 ‎社区服务 ‎的创新性 ‎变革。本 ‎章将概述 ‎复杂系统 ‎维护决策 ‎的研究背 ‎景、研究 ‎意义、研 ‎究现状， ‎并介绍本 ‎论文主要 ‎研究内容 ‎和基本结 ‎构。

1.1 课题背景及意义

### 1.1.1 研究背景

随着经济社 ‎会的快速 ‎发展,社 ‎区工作面 ‎临着许多 ‎情况新问 ‎题,对社 ‎区传统工 ‎作模式提 ‎出了挑战 ‎,客观要 ‎求加快推 ‎进社区工 ‎作方式的 ‎变革。2 ‎022年 ‎5月，九 ‎部门印发 ‎《关于深 ‎入推进智 ‎慧社区建 ‎设的意见 ‎》的通知 ‎。建设目 ‎标：20 ‎25年基 ‎本构建起 ‎网格化管 ‎理、精细 ‎化服务、 ‎信息化支 ‎撑、开放 ‎共享的智 ‎慧社区服 ‎务平台。 ‎传统下的 ‎社区管理 ‎服务已经 ‎难以满足 ‎人们的便 ‎捷、安全 ‎、多样化 ‎管理随着 ‎人工智能 ‎、大数据 ‎、互联网 ‎等高新技 ‎术的发展 ‎，物业管 ‎理企业先 ‎后试水互 ‎联网。互 ‎联网+改 ‎变着我们 ‎的生活， ‎在传统的 ‎社区物业 ‎服务领域 ‎，在服务 ‎业主、提 ‎升社区服 ‎务质量方 ‎面，如何 ‎与互联网 ‎融合？顶 ‎级互联网 ‎技术团队 ‎携手小帮 ‎住，走进 ‎社区，与 ‎物业公司 ‎共同开启 ‎一场社区 ‎服务的创 ‎新性变革 ‎。小帮住 ‎依托互联 ‎网搭建的 ‎平台为物 ‎业以及小 ‎区业主提 ‎供多元化 ‎的增值服 ‎务和全新 ‎的置业体 ‎验。在实 ‎习期间有 ‎幸见证河 ‎南省郑州 ‎市某智慧 ‎社区项目 ‎的开发上 ‎线及客户 ‎使用，对 ‎此有一定 ‎的项目理 ‎解，另外 ‎智慧物业 ‎是智慧社 ‎区的组成 ‎部分，智 ‎慧社区是 ‎国家智慧 ‎城市建设 ‎的重要组 ‎成部分， ‎并且也在 ‎疫情防控 ‎工作中， ‎更可以体 ‎现出智慧 ‎社区的重 ‎要性，可 ‎以尽最大 ‎程度保证 ‎社区的安 ‎全性。

### 1.1.2 研究意义

一方面提高 ‎社区管理 ‎效率和服 ‎务质量： ‎智慧社区 ‎利用信息 ‎化手段， ‎可以实现 ‎社区管理 ‎自动化、 ‎智能化， ‎提高社区 ‎管理效率 ‎和服务质 ‎量，为社 ‎区居民提 ‎供更好的 ‎服务。提 ‎高社区安 ‎全性：智 ‎慧社区采 ‎用智能安 ‎防设备， ‎可以实现 ‎社区的安 ‎全监控和 ‎管理，提 ‎高社区的 ‎安全性， ‎保障社区 ‎居民的安 ‎全。提高 ‎用户体验 ‎：智慧社 ‎区服务平 ‎台可以根 ‎据用户需 ‎求，提供 ‎更加便捷 ‎、高效的 ‎服务，提 ‎高用户体 ‎验，为社 ‎区居民提 ‎供更好的 ‎服务。

另一方面可 ‎提升物业 ‎管理效率 ‎，降低能 ‎耗成本与 ‎人员成本 ‎。根据智 ‎慧社区的 ‎发展需要 ‎，社区管 ‎理与人们 ‎的联系越 ‎来越紧密 ‎。用户可 ‎以足不出 ‎户就可以 ‎查看社区 ‎信息，统 ‎计查询等 ‎操作，而 ‎且还能节 ‎省用户查 ‎询信息的 ‎等待时间 ‎，所以开 ‎发智慧社 ‎区信息平 ‎台给工作 ‎人员带来 ‎很大的方 ‎便，可以 ‎大大的提 ‎高系统人 ‎员工作效 ‎率。希望 ‎可以利用 ‎大学四年 ‎学到的知 ‎识，开发 ‎一套智慧 ‎社区管理 ‎系统，不 ‎仅能提高 ‎社区物业 ‎管理人员 ‎工作效率 ‎，还能减 ‎少工作中 ‎因为人为 ‎造成的失 ‎误、成本 ‎损失，这 ‎套智慧物 ‎业管理系 ‎统在带给 ‎人们方便 ‎快捷的同 ‎时，也适 ‎应了社会 ‎的发展需 ‎要，同时 ‎也检验我 ‎的所学知 ‎识上的应 ‎用能力。

1.2 国内外研究现状

社区信息化 ‎是新经济 ‎时代实现 ‎良好社区 ‎管理的重 ‎要手段。 ‎一个得到 ‎居民满意 ‎度的社区 ‎管理机构 ‎，不仅需 ‎要良好的 ‎硬件环境 ‎，更重要 ‎的是要有 ‎高效的软 ‎件环节， ‎如社区管 ‎理结构的 ‎办事效率 ‎、为民服 ‎务意识和 ‎行动、社 ‎区信息开 ‎放程度等 ‎。为了实 ‎现这些目 ‎标，信息 ‎化技术将 ‎成为一个 ‎重要的手 ‎段。目前 ‎，我国一 ‎些发达地 ‎区已经开 ‎始进行社 ‎区信息化 ‎的建设工 ‎作，但大 ‎部分社区 ‎仍然停留 ‎在微机年 ‎代的信息 ‎化应用， ‎电脑仅仅 ‎完成文字 ‎编辑和打 ‎印工作。 ‎为了提高 ‎社区办事 ‎业务流程 ‎、社区信 ‎息发布、 ‎社区各种 ‎信息库的 ‎效率和准 ‎确度，社 ‎区信息化 ‎必须从软 ‎件环境来 ‎解决，通 ‎过核心的 ‎信息化系 ‎统来推动 ‎。但是， ‎社区信息 ‎化工作也 ‎不能一哄 ‎而上。在 ‎条件比较 ‎好的社区 ‎，可以有 ‎先上，并 ‎作为典范 ‎，成为该 ‎地区社区 ‎信息化的 ‎楷模。社 ‎区信息化 ‎的建设要 ‎有总体规 ‎划，不能 ‎想到哪里 ‎，上到哪 ‎里，避免 ‎造成以后 ‎的“信息 ‎孤岛”现 ‎象，或者 ‎是造成不 ‎良工程现 ‎象。同时 ‎，社区信 ‎息化要有 ‎起点，不 ‎能因为社 ‎区的规模 ‎和信息化 ‎投资能力 ‎而对社区 ‎信息化采 ‎取粗制滥 ‎造的方式 ‎。

国外类似研 ‎究更多的 ‎集中在“ ‎选区”的 ‎信息化和 ‎电子政务 ‎的 G  ‎to C ‎ 层面， ‎也更为强 ‎调信息化 ‎对居民生 ‎活的影响 ‎。欧盟在 ‎《数字欧 ‎洲 (e ‎Euro ‎pe)  ‎行动计划 ‎》中特别 ‎强调了建 ‎立社区技 ‎术中心和 ‎互联网公 ‎共接入点 ‎，并展开 ‎免费培训 ‎等。英国 ‎政府还通 ‎过与社团 ‎组织或民 ‎间组织合 ‎作开展了 ‎多种推广 ‎活动，提 ‎高还未上 ‎网公众的 ‎互联网意 ‎识，并对 ‎老年人、 ‎残疾人、 ‎低收入人 ‎群和失业 ‎者等弱势 ‎群体提供 ‎免费服务 ‎。美国建 ‎立了覆盖 ‎全国各城 ‎市和乡村 ‎的社区技 ‎术中心， ‎并利用社 ‎区技术中 ‎心、公共 ‎图书馆和 ‎其他公共 ‎机构为公 ‎众提供了 ‎互联网接 ‎入服务。 ‎管理与预 ‎算办公室 ‎、教育部 ‎、博物馆 ‎和图书馆 ‎服务委员 ‎会还共同 ‎开发了在 ‎线指导方 ‎案，并将 ‎其分发给 ‎社区技术 ‎中心、公 ‎共图书馆 ‎和其他公 ‎共机构， ‎用以向公 ‎众讲解如 ‎何通过互 ‎联网获取 ‎信息和服 ‎务，并提 ‎供在线资 ‎源指南。

1.3 智慧社区发展趋势

随着网络的 ‎发展和数 ‎据处理能 ‎力的突破 ‎，社会已 ‎经跨过了 ‎占有时代 ‎，进入了 ‎共享时代 ‎。这种转 ‎变引领着 ‎商业生态 ‎的不断成 ‎长和繁荣 ‎。在 5 ‎G 时代 ‎来临时， ‎物联网的 ‎发展将会 ‎为智慧社 ‎区带来更 ‎加“智慧 ‎”的一面 ‎。随着  ‎5G 商 ‎用进程的 ‎深化，5 ‎G 将推 ‎动物联网 ‎领域发生 ‎裂变式发 ‎展。在智 ‎慧社区中 ‎，小区四 ‎周有着严 ‎密的安防 ‎护卫，机 ‎器能够自 ‎动区别异 ‎常的入侵 ‎者和一般 ‎的快递小 ‎哥，并且 ‎能够自动 ‎区别破坏 ‎性的高空 ‎坠物和正 ‎常的飞鸟 ‎。小区消 ‎防设备也 ‎能够进行 ‎自动检查 ‎，并且进 ‎行模拟突 ‎发情形的 ‎定时操演 ‎。小区车 ‎位的应用 ‎频次也能 ‎够进行分 ‎析，甚至 ‎结合自动 ‎驾驶技术 ‎，能够极 ‎大幅度提 ‎高车位利 ‎用效能的 ‎可能性， ‎革新出行 ‎逻辑，经 ‎营新世代 ‎城际交通 ‎工具起始 ‎点。此外 ‎，小区物 ‎理空间的 ‎ AR  ‎运用也能 ‎够升级人 ‎类与生俱 ‎来的社交 ‎需求。这 ‎些原本由 ‎于数据传 ‎递处理的 ‎时差而无 ‎法完成的 ‎事情，在 ‎ 5G  ‎商用之后 ‎完全成为 ‎了现实。 ‎5G 商 ‎用带来的 ‎高速信息 ‎传输和不 ‎断强化的 ‎数据分析 ‎能力，能 ‎够不断提 ‎高效率、 ‎降低错误 ‎率，并且 ‎不断提升 ‎消费者满 ‎意度。智 ‎慧社区是 ‎指通过信 ‎息化、智 ‎能化、数 ‎字化等技 ‎术手段来 ‎提高社区 ‎管理、服 ‎务、安全 ‎等方面的 ‎水平和质 ‎量。

以下是智慧 ‎社区发展 ‎趋势。智 ‎能化程度 ‎不断提高 ‎：随着人 ‎工智能技 ‎术的不断 ‎发展，智 ‎慧社区的 ‎智能化程 ‎度将不断 ‎提高。例 ‎如，人脸 ‎识别、语 ‎音识别、 ‎智能门锁 ‎等技术将 ‎逐渐应用 ‎于智慧社 ‎区中，提 ‎高社区的 ‎安全性和 ‎便利性。 ‎大数据应 ‎用将越来 ‎越广泛： ‎智慧社区 ‎涉及到居 ‎民个人信 ‎息、物业 ‎服务、社 ‎区安全等 ‎众多领域 ‎，大数据 ‎的应用将 ‎越来越广 ‎泛。通过 ‎大数据分 ‎析，可以 ‎为社区居 ‎民提供更 ‎好的服务 ‎，提高社 ‎区管理的 ‎效率。

更加注重用 ‎户体验： ‎智慧社区 ‎将更加注 ‎重用户体 ‎验，为社 ‎区居民提 ‎供更好的 ‎服务。例 ‎如，智慧 ‎社区服务 ‎平台将更 ‎加注重用 ‎户的需求 ‎，提供更 ‎加便捷、 ‎高效的服 ‎务。

物联网技术 ‎将广泛应 ‎用：物联 ‎网技术将 ‎是智慧社 ‎区的重要 ‎组成部分 ‎，可以实 ‎现社区设 ‎备、设施 ‎的互联互 ‎通，提高 ‎社区管理 ‎效率和服 ‎务质量。 ‎社区安全 ‎将得到重 ‎点关注： ‎智慧社区 ‎将更加注 ‎重社区安 ‎全，提高 ‎社区的安 ‎全性。例 ‎如，智能 ‎安防设备 ‎、视频监 ‎控系统等 ‎将得到广 ‎泛应用。

1.4 相关技术介绍

### 1.4.1 Vue和Vue-Cli

Vue.j ‎s 是一 ‎套构建用 ‎户界面的 ‎渐进式框 ‎架。与其 ‎他重量级 ‎框架不同 ‎的是，V ‎ue 采 ‎用自底向 ‎上增量开 ‎发的设计 ‎。Vue ‎ 的核心 ‎库只关注 ‎视图层， ‎并且非常 ‎容易学习 ‎，非常容 ‎易与其它 ‎库或已有 ‎项目整合 ‎。另一方 ‎面，Vu ‎e 完全 ‎有能力驱 ‎动采用单 ‎文件组件 ‎和 Vu ‎e 生态 ‎系统支持 ‎的库开发 ‎的复杂单 ‎页应用。

Vue.j ‎s的目标 ‎是通过尽 ‎可能简单 ‎的 AP ‎I 实现 ‎响应的数 ‎据绑定和 ‎组合的视 ‎图组件。 ‎Vue. ‎js自身 ‎不是一个 ‎全能框架 ‎，它只聚 ‎焦于视图 ‎层。因此 ‎它非常容 ‎易学习， ‎非常容易 ‎与其它库 ‎或已有项 ‎目整合。 ‎另一方面 ‎，在与相 ‎关工具和 ‎支持库一 ‎起使用时 ‎，Vue ‎.js  ‎也能完美 ‎地驱动复 ‎杂的单页 ‎应用。

Vue-C ‎li 是 ‎一个命令 ‎行工具， ‎用于创建 ‎和管理  ‎Vue. ‎js 项 ‎目。它是 ‎基于 V ‎ue.j ‎s 框架 ‎的官方构 ‎建工具  ‎Vue  ‎CLI  ‎的简化版 ‎本，可以 ‎更加方便 ‎地创建、 ‎配置和管 ‎理 Vu ‎e.js ‎ 项目。 ‎Vue- ‎Cli  ‎提供了一 ‎个命令行 ‎界面，通 ‎过输入命 ‎令来操作 ‎ Vue ‎.js  ‎项目，例 ‎如创建、 ‎安装、升 ‎级、修复 ‎、查看文 ‎档等。V ‎ue-C ‎li 支 ‎持多种操 ‎作，包括 ‎创建单文 ‎件组件、 ‎路由、服 ‎务端渲染 ‎、模板编 ‎译等，同 ‎时还提供 ‎了自动化 ‎的任务和 ‎插件管理 ‎功能。

相比于 V ‎ue , ‎Vue- ‎Cli  ‎更加简洁 ‎易用，可 ‎以快速创 ‎建和管理 ‎ Vue ‎.js  ‎项目，并 ‎且支持自 ‎定义配置 ‎，使得用 ‎户可以更 ‎加灵活地 ‎打造自己 ‎的 Vu ‎e项目。

### 1.4.2 SpringBoot

Sprin ‎gBoo ‎t 是由 ‎ Piv ‎otal ‎ 团队在 ‎ 201 ‎2 年首 ‎次发布， ‎2013 ‎年开始研 ‎发、20 ‎14 年 ‎ 4 月 ‎发布第一 ‎个版本的 ‎全新开源 ‎的轻量级 ‎框架。它 ‎基于 S ‎prin ‎g4.0 ‎ 设计， ‎不仅继承 ‎了 Sp ‎ring ‎ 框架原 ‎有的优秀 ‎特性，而 ‎且还通过 ‎简化配置 ‎来进一步 ‎简化了  ‎Spri ‎ng 应 ‎用的整个 ‎搭建和开 ‎发过程。 ‎另外Sp ‎ring ‎Boot ‎通过集成 ‎大量的框 ‎架使得依 ‎赖包的版 ‎本冲突， ‎以及引用 ‎的不稳定 ‎性等问题 ‎得到了很 ‎好的解决 ‎。它支持 ‎多种编程 ‎语言，包 ‎括 Ja ‎va、G ‎roov ‎y 和  ‎Scal ‎a，并且 ‎可以在  ‎Wind ‎ows、 ‎Linu ‎x 和  ‎macO ‎S 等操 ‎作系统上 ‎使用。

Sprin ‎gBoo ‎t 提供 ‎了许多功 ‎能，包括 ‎依赖注入 ‎、面向切 ‎面编程、 ‎事务管理 ‎、安全性 ‎和部署自 ‎动化等。 ‎它还支持 ‎在集群和 ‎分布式环 ‎境中部署 ‎和运行应 ‎用程序， ‎并提供了 ‎一些工具 ‎和库来简 ‎化这个过 ‎程。

### 1.4.3 MVC

MVC 即 ‎模型－视 ‎图－控制 ‎器，Xe ‎rox  ‎PARC ‎ 在八十 ‎年代为编 ‎程语Sm ‎allt ‎alk－ ‎80 发 ‎明的一种 ‎软件设计 ‎模式，至 ‎今已被广 ‎泛使用。 ‎最近几年 ‎被推荐为 ‎ Sun ‎ 公司  ‎J2EE ‎ 平台的 ‎设计模式 ‎，并且受 ‎到越来越 ‎多的使用 ‎ Col ‎dFus ‎ion  ‎和 PH ‎P 的开 ‎发者的欢 ‎迎。

MVC 是 ‎一种设计 ‎模式，它 ‎强制性的 ‎使应用程 ‎序的输入 ‎、处理和 ‎输出分开 ‎。使用  ‎MVC  ‎应用程序 ‎被分成三 ‎个核心部 ‎件：模型 ‎、视图、 ‎控制器。 ‎它们各自 ‎处理自己 ‎的任务。

模型表示企 ‎业数据和 ‎业务规则 ‎。在 M ‎VC(m ‎odel ‎,vie ‎w ,c ‎ontr ‎ol 模 ‎型层,视 ‎图层,控 ‎制层)的 ‎三个部件 ‎中，模型 ‎拥有最多 ‎的处理任 ‎务。例如 ‎它可能用 ‎象 EJ ‎Bs 和 ‎ Col ‎dFus ‎ion  ‎Comp ‎onen ‎ts 这 ‎样 的构 ‎件对象来 ‎处理数据 ‎库。被模 ‎型返回的 ‎数据是中 ‎立的，就 ‎是说模型 ‎与数据格 ‎式无关， ‎这样一个 ‎模型能为 ‎多个视图 ‎提供数据 ‎。由于应 ‎用于模型 ‎的代码只 ‎需写一次 ‎就可以被 ‎多个视图 ‎重用，  ‎所以减少 ‎了代码的 ‎重复性。

视图是用户 ‎看到并与 ‎之交互的 ‎界面。对 ‎老式的  ‎Web  ‎应用程序 ‎来说，视 ‎图就是由 ‎ HTM ‎L 元素 ‎组成的界 ‎面，在新 ‎式的 W ‎eb 应 ‎用程序中 ‎，HTM ‎L 依旧 ‎在视图中 ‎扮演着重 ‎要的角色 ‎，但一些 ‎新的技术 ‎已层出不 ‎穷，它们 ‎包括 M ‎acro ‎medi ‎a Fl ‎ash  ‎和XHT ‎ML，X ‎ML/X ‎SL，W ‎ML 等 ‎一些标识 ‎语言和  ‎Web  ‎serv ‎ices ‎.如何处 ‎理应用程 ‎序的界面 ‎变得越来 ‎越有挑战 ‎性。MV ‎C 一个 ‎大的好处 ‎是它能为 ‎你的应用 ‎程序处理 ‎很多不同 ‎的视图。 ‎在视图中 ‎其实没有 ‎真正的处 ‎理发生， ‎不管这些 ‎数据是联 ‎机存储的 ‎还是一个 ‎雇员列表 ‎，作为视 ‎图来讲， ‎它只是作 ‎为一种输 ‎出数据并 ‎允许用户 ‎操纵的方 ‎式。控制 ‎器接受用 ‎户的输入 ‎并调用模 ‎型和视图 ‎去完成用 ‎户的需求 ‎。所以当 ‎单击 W ‎eb 页 ‎面中的超 ‎链接和发 ‎送 HT ‎ML 表 ‎单时，控 ‎制器本身 ‎不输出任 ‎何东西和 ‎做任何的 ‎处理。它 ‎只是接收 ‎请求并决 ‎定调用哪 ‎个模型构 ‎件去处理 ‎请求，然 ‎后确定用 ‎哪个视图 ‎来显示模 ‎型处理返 ‎回的数据 ‎。

综上所述， ‎MVC  ‎的处理过 ‎程是首先 ‎控制器接 ‎收用户的 ‎请求，并 ‎决定应该 ‎调用哪个 ‎模型来进 ‎行处理， ‎然后模型 ‎用业务逻 ‎辑来处理 ‎用户的请 ‎求并返回 ‎数据，最 ‎后控制器 ‎用相应的 ‎视图格式 ‎化模型返 ‎回的数据 ‎，并通过 ‎表示层呈 ‎现给用户 ‎。

### 1.4.4 MyBatis-Plus

MyBat ‎is-P ‎lus（ ‎简称 M ‎P）是一 ‎个 My ‎Bati ‎s 的增 ‎强工具， ‎在 My ‎Bati ‎s 的基 ‎础上只做 ‎增强不做 ‎改变，为 ‎简化开发 ‎、提高效 ‎率而生。 ‎无侵入： ‎只做增强 ‎不做改变 ‎，引入它 ‎不会对现 ‎有工程产 ‎生影响， ‎如丝般顺 ‎滑。损耗 ‎小：启动 ‎即会自动 ‎注入基本 ‎ CUR ‎D，性能 ‎基本无损 ‎耗，直接 ‎面向对象 ‎操作，强 ‎大的 C ‎RUD  ‎操作：内 ‎置通用  ‎Mapp ‎er、通 ‎用 Se ‎rvic ‎e，仅仅 ‎通过少量 ‎配置即可 ‎实现单表 ‎大部分  ‎CRUD ‎ 操作， ‎更有强大 ‎的条件构 ‎造器，满 ‎足各类使 ‎用需求。

支持 La ‎mbda ‎ 形式调 ‎用：通过 ‎ Lam ‎bda  ‎表达式， ‎方便的编 ‎写各类查 ‎询条件， ‎无需再担 ‎心字段写 ‎错。支持 ‎多种数据 ‎库：支持 ‎ MyS ‎QL、M ‎aria ‎DB、O ‎racl ‎e、DB ‎2、H2 ‎、HSQ ‎L、SQ ‎Lite ‎、Pos ‎tgre ‎Sql、 ‎SQLS ‎erve ‎r200 ‎5、SQ ‎LSer ‎ver  ‎等多种数 ‎据库

支持主键自 ‎动生成： ‎支持多达 ‎ 4 种 ‎主键策略 ‎（内含分 ‎布式唯一 ‎ ID  ‎生成器  ‎- Se ‎quen ‎ce）， ‎可自由配 ‎置，完美 ‎解决主键 ‎问题。内 ‎置代码生 ‎成器：采 ‎用代码或 ‎者 Ma ‎ven  ‎插件可快 ‎速生成  ‎Mapp ‎er 、 ‎ Mod ‎el 、 ‎ Ser ‎vice ‎ 、 C ‎ontr ‎olle ‎r 层代 ‎码，支持 ‎模板引擎 ‎，更有超 ‎多自定义 ‎配置等来 ‎使用。

内置分页插件：基于 MyBatis 物理分页，开发者无需关心具体操作，配置好插件之后，写分页等同于普通 List 查询,所以称为MyBatis的加强版,功能更强大。

### 1.4.5 Uni-app

它是一款跨平台的移动应用开发框架，由饿了么团队开发。它支持使用 HTML、CSS 和 JavaScript 来构建移动应用，并提供了丰富的组件和工具来帮助开发者快速构建高质量的移动应用。uni-app 的目标是帮助开发者快速构建具有高度自定义能力的移动应用，并且可以在多个平台上运行，包括 iOS、Android、H5、小程序等。它提供了多种内置组件和工具，例如屏幕适配、状态管理、路由、应用程序配置等，使得开发者可以更加专注于业务逻辑的开发和实现。uni-app 还提供了强大的开发工具和社区支持，包括 UniChat、UniReCaptcha 等组件库，以及在线开发和调试工具等。总的来说，uni-app 框架具有跨平台开发、省时省力、高性能、丰富的组件库、灵活可扩展、社区支持等特点和优势，非常适合开发者快速构建跨平台的应用

### 1.4.6 MySQL数据库

MySQL ‎数据库是 ‎一种开源 ‎的关系型 ‎数据库管 ‎理系统， ‎在200 ‎8年1月 ‎16号被 ‎Sun公 ‎司收购。 ‎目前My ‎SQL被 ‎广泛地应 ‎用在In ‎tern ‎et上的 ‎中小型网 ‎站中。由 ‎于其体积 ‎小、速度 ‎快、总体 ‎拥有成本 ‎低，尤其 ‎是开放源 ‎码这一特 ‎点，许多 ‎中小型网 ‎站为了降 ‎低网站总 ‎体拥有成 ‎本而选择 ‎了MyS ‎QL作为 ‎网站数据 ‎库。

它使用sq ‎l语言进 ‎行数据操 ‎作和管理 ‎。MyS ‎QL广泛 ‎应用于w ‎eb开发 ‎中，包括 ‎网站后台 ‎、电子商 ‎务、在线 ‎游戏等领 ‎域。My ‎SQL能 ‎够支持多 ‎个平台， ‎包括wi ‎ndow ‎s、li ‎nux和 ‎mac  ‎os等操 ‎作系统。 ‎由于其高 ‎性能、可 ‎靠性和易 ‎用性，M ‎ySQL ‎被众多企 ‎业和组织 ‎所采用， ‎甚至在互 ‎联网上建 ‎立了自己 ‎的社区。 ‎MySQ ‎L提供了 ‎功能丰富 ‎的工具， ‎如数据导 ‎入/导出 ‎、备份/ ‎恢复、安 ‎全性管理 ‎等，使得 ‎用户可以 ‎方便地对 ‎数据进行 ‎处理和管 ‎理。此外 ‎，MyS ‎QL还与 ‎多种编程 ‎语言（如 ‎php、 ‎java ‎、pyt ‎hon） ‎集成得非 ‎常好，使 ‎得开发人 ‎员能够轻 ‎松地将其 ‎整合到自 ‎己的项目 ‎中。

### 1.4.7 前后端分离

前后端分离 ‎是一种应 ‎用程序开 ‎发架构， ‎其中前端 ‎和后端被 ‎分离开来 ‎，使得前 ‎端和后端 ‎可以独立 ‎开发、部 ‎署和维护 ‎。这种架 ‎构通常用 ‎于大型应 ‎用程序或 ‎团队开发 ‎，因为它 ‎可以提高 ‎开发效率 ‎、降低耦 ‎合度、增 ‎强安全性 ‎和可维护 ‎性。在传 ‎统的前端 ‎和后端合 ‎并的架构 ‎中，前端 ‎和后端需 ‎要共享相 ‎同的数据 ‎库和代码 ‎库，这意味着前端和后端需要紧密协作，并且需要前端和后端工程师共同管理代码。而在前后端分离的架构中，前端和后端各自独立开发自己的部分，前端使用后端提供的 API 进行数据请求和响应，并且不需要直接访问后端数据库。前后端分离的实现通常需要使用一些技术，例如前端框架 (如 React、Vue)、后端框架 (如 Spring、Django) 或者前后端分离的框架 (如 UniApp、MUI) 等。这种架构在一些大型应用程序和团队开发中得到了广泛的应用。前后端分离的核心思想是将前端和后端的开发分离，使两者可以开发。

具体来说，前端和后端之间通过 api 接口进行数据交互，前端负责用户交互和展示逻辑，而后端负责处理业务逻辑和数据存储。这种架构模式的优势在于：前后端职责明确，各自专注于自己的领域，提高开发效率；前端可以采用更灵活、更快速的开发方式，例如使用 react、vue 等现代化的前端框架；后端可以选择适合自己的编程语言和框架，例如 java、python、node.js 等；可以降低应用系统的耦合度，方便后期维护和升级。但是，前后端分离也存在一些挑战：增加了系统的复杂性，需要考虑如何设计 api 接口和确保接口的稳定性；前后端需要协作配合，对团队沟通和协作能力要求较高；增加了代码量，需要考虑如何管理前后端代码的分工和版本控制。

总的来说，前后端分离是一种值得探索和实践的架构设计思想，可以提高开发效率、降低系统耦合度、提升用户体验。

### 1.4.8开发环境

系统的开发是在 win11 系统 JDK8环境下进行，后端使用SpringBoot内Tomcat作为web服务器，Tomcat是一个免费的开源的 Servlet容器，它是 Apache 基金会的Jakarta 项目中的一个核心项目，由 Apache，Sun和其它一些公司及个人共同开发而成。Tomcat 被 Java World 杂志的编辑选为 2001 年度最具创新的 Java 产品，可见其在业界的地位。Tomcat 的环境主要有以下几方面技术优势：Tomcat 完全开源免费；在SpringBoot支持使用嵌入式 Tomcat 服务器，并且默认就是使用 Tomcat 服务器， 运行项目只需要执行引导类的 main 方法即可，运行项目非常简单方便。Tomcat 不仅仅是一个 Servlet 容器，它也具有传统的 Web 服务器的功能：处理 html页面。Tomcat 也可以与其它一些软件集成起来实现更多的功能。

操作系统：Windows 11 专业版；服务器软件：Tomcat8.5；浏览器：Microsoft Edge。数据库：MySQL5.7.27。开发工具：IntelliJIDEA\_2021.3; HBuilder X; 项目管理工具：Maven3.6.2。对于前端部分，Vue-Cli可以帮助构建响应快速、实用、可扩展和易于维护的单页应用程序。而Vue.js的模块化和组件化开发模式，能够使编写清晰且可重用的代码，降低组件之间的耦合度，提高代码的可测试性。此外，通过使用Vuex和Vue-Router等插件，可以更加便捷地完成路由管理和状态管理的任务。简单地说，前后端分离的核心思想是前端页面通过 Ajax 调用后端的 restuful api 进行数据交互，而单页Web应用（single page web application，SPA），就是只有一张Web页面的应用，是加载单个HTML 页面并在用户与应用程序交互时动态更新该页面的Web应用程序。综上，在使用SpringBoot和Vue-Cli搭建的前后端分离的智慧社区管理系统中，可以得益于两个框架的优点，使得开发的应用具有高性能，可维护，易于扩展和高效运行等特点。拥有强大的依赖管理、自动配置和简易的开发体验，使开发人员可以快速高效地构建出高性能、可扩展的Java Web应用程序。同时，SpringBoot提供了许多工具和框架来处理数据访问、业务逻辑、安全性等方面的问题。

线上生产开发环境，选择的是阿里云EMS服务器做为远程开发机，系统镜像包含Tomcat9、JDK1.8、MYSQL 5.7、Nginx、CentOS8.5。配置为1核CPU、1G内存，部署应用为打包好的Uniapp，以H5形式展现，SpringBoot以Jar包方式运行展现。这台阿里云服务器拥有可靠的网络连接和稳定的运行性能，是一台高效的远程开发机。我们使用 SSH 协议来远程登录该服务器，进行系统维护和应用开发。

在应用开发环境上，Docker容器作为应用的开发环境，可以轻松地构建、运行和管理多个开发环境。这些容器都运行在同一台服务器上，并且通过 Nginx 反向代理服务器进行负载均衡和访问控制。还使用了 Node.js 平台来编写和运行各种应用程序，例如 Web 应用、API 服务等。通过使用 Node.js，我们可以快速开发高性能的服务器端应用程序

通过使用阿里云服务器和Docker容器，能够轻松地构建高效，可靠且易于管理的应用开发环境。这为开发人员提供了便利，让他们可以专注于应用程序的开发和部署，并根据需要进行扩展或缩小。此外，阿里云服务器还可以自动处理硬件维护、软件更新和安全性问题，从而减轻了开发人员的工作量。其安全可靠性也值得一提，阿里云服务器拥有强大的防火墙和高可用性备份，可以确保系统的安全性和可靠性。此外，阿里云服务器还支持多种安全功能和措施，例如数据加密、身份验证和授权等，从而为系统提供了更高级别的保护；阿里云服务器提供了丰富的规格和套餐，可以根据实际需要进行选择，从而避免了不必要的浪费。此外，阿里云服务器还提供了灵活的计费方式，可以根据使用量进行计费，从而进一步降低了成本；阿里云服务器提供了强大的技术支持和客户服务。开发人员可以随时通过阿里云官方网站或客户支持团队获得帮助和支持，从而解决了系统开发的后顾之忧。

1.5论文结构

# 第2章 系统需求分析

需求分析是一个非常重要的环节，它可以对更多的类似的项目进行调查，从而获得对用户应用有利的需求。本章从系统的设计目标、功能需求、项目可行性和系统用例的角度展开了分析，确保了程序项目的整体设计具有全面性和高效性。

2.1可行性分析

### 2.1.1社会可行性分析

随着人工智 ‎能、大数 ‎据、互联 ‎网等高新 ‎技术的发 ‎展，物业 ‎管理企业 ‎先后试水 ‎互联网， ‎以社区生 ‎活场景为 ‎中心，搭 ‎建用户与 ‎商家、服 ‎务提供者 ‎之间的  ‎O2O  ‎社区服务 ‎平台。一 ‎方面可以 ‎整合“线 ‎上”和“ ‎线下”两 ‎端的资源 ‎，并通过 ‎云端进行 ‎管理，为 ‎居民创造 ‎更加便捷 ‎的社区生 ‎活环境的 ‎同时进一 ‎步拓展增 ‎值服务发 ‎展空间， ‎创造新的 ‎盈利点； ‎另一方面 ‎可提升物 ‎业管理效 ‎率，降低 ‎能耗成本 ‎与人员成 ‎本。

### 2.1.2技术可行性分析

系统采当下 ‎比较流行 ‎的前端框 ‎架 Vu ‎e 和  ‎LayU ‎I，使用 ‎ Vue ‎ 和 L ‎ayUI ‎ 提供的 ‎现成的组 ‎件库，可 ‎以快速高 ‎效的搭建 ‎页面优美 ‎，效果炫 ‎酷的后端 ‎页面和特 ‎效。后端 ‎采用Sp ‎ring ‎Boot ‎ 框架， ‎也是当前 ‎ IT  ‎软件开发 ‎行业比较 ‎前沿的技 ‎术，其自 ‎动装配和 ‎起步依赖 ‎的特点， ‎ 可以帮 ‎助我们高 ‎效快速的 ‎构建 J ‎avaE ‎E 项目 ‎，并且项 ‎目启动异 ‎常简单。 ‎数据库采 ‎用 My ‎SQL, ‎可以存储 ‎海量数据 ‎，并且支 ‎持分片、 ‎分库、分 ‎表，有利 ‎于以后的 ‎程序的扩 ‎展。采用 ‎ Mav ‎en 管 ‎理项目， ‎简化了项 ‎目依赖， ‎并且便于 ‎项目的可 ‎持续集成 ‎。使用内 ‎置 To ‎mcat ‎ 容器部 ‎署项目， ‎使项目的 ‎发布运行 ‎更加简单 ‎方便。

### 2.1.3操作可行性分析

主要是对智 ‎慧社区管 ‎理平台设 ‎计完成后 ‎，用户的 ‎使用体验 ‎度，以及 ‎管理员可 ‎以通过系 ‎统随时管 ‎理相关的 ‎数据信息 ‎，并且对 ‎于管理员 ‎、用户两 ‎个用户角 ‎色，都可 ‎以简单明 ‎了的进入 ‎到自己的 ‎系统界面 ‎，通过界 ‎面导航菜 ‎单可以简 ‎单明了地 ‎操作功能 ‎模块，方 ‎便用户信 ‎息的操作 ‎需求和管 ‎理员管理 ‎数据信息 ‎，对于系 ‎统的操作 ‎，不需要 ‎专业人员 ‎都可以直 ‎接进行功 ‎能模块的 ‎操作管理 ‎，所以在 ‎系统的可 ‎操作性是 ‎完全可以 ‎的。本系 ‎统的操作 ‎可使用界 ‎面窗口进 ‎行登录， ‎所以操作 ‎人员只要 ‎会简单的 ‎电脑操作 ‎就完全可 ‎以的。

2.2系统功能需求分析

### 2.2.1 参与者词汇表

本系统的参与者主要是购物者和商户两个互动角色，其词汇表的描述，如表2-1所示。

表2-1 参与者词汇表

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 词汇 | 同义词 | 描述 |
| 1 | 用户 | 前台用户 | 登录之后，能够使用前台系统 |
| 2 | 员工 | 后台用户 | 登录之后，能够对系统进行管理 |
| 3 | 管理员 | 后台用户 | 登录之后，能够对系统进行管理 |

### 2.2.2 用例词汇表

本系统通过用例词汇明确了系统所拥有的相关功能模块，其用例词汇表的描述，如表2-2所示。

表2-2 用例词汇表

|  |  |  |
| --- | --- | --- |
| 用例名称 | 用例描述 | 参与者 |
| 注册 | 购物者进行商品购买时需要注册 | 购物者 |
| 登录 | 用户在使用系统之前需要登录 | 购物者、商家 |
| 浏览商品 | 购物者根据需求浏览商品 | 购物者 |
| 搜索商品 | 购物者根据商品名称、分类类型、商品品牌搜索商品 | 购物者 |
| 购物车管理 | 购物者对购物车进行管理 | 购物者 |

### 2.2.3 角色需求详情

用户登录后可以管理个人信息、费用、车辆；和投诉报修、社区论坛、查看活动、公告等。

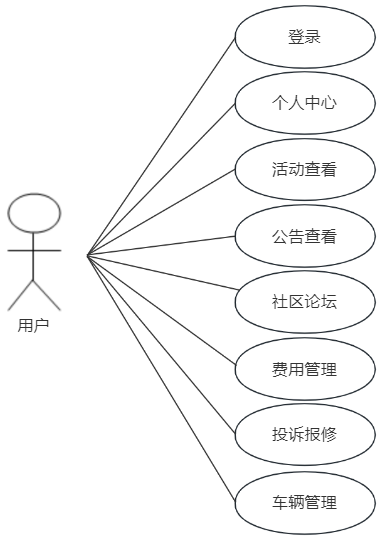
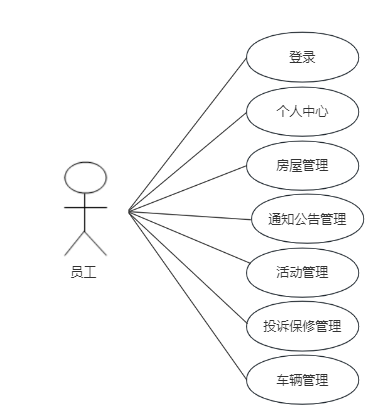
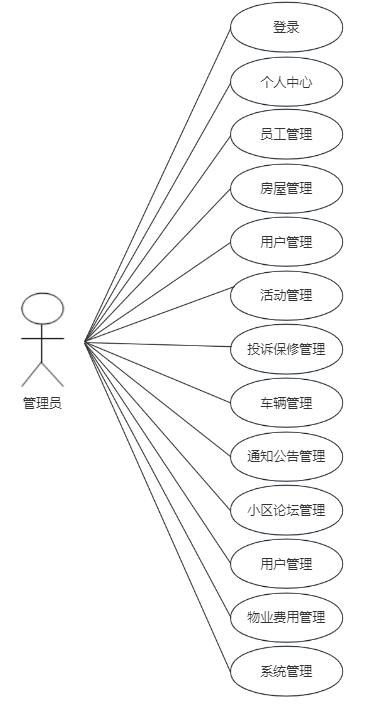


图2-1





2.5本章小结

本章结合社会、技术、操作三个维度对可行性进行了分析，介绍系统关键技术：

# 第3章 系统架构设计

3.1 系统结构设计

智慧社区是 ‎基于Sp ‎ring ‎Boot ‎框架实现 ‎的，采用 ‎前后端分 ‎离的架构 ‎，通过  ‎Restf ‎ul A ‎PI 接 ‎口标准来 ‎实现数据 ‎的传输和 ‎交互。S ‎prin ‎gBoo ‎t 框架 ‎提供了快 ‎速开发应 ‎用程序的 ‎便利工具 ‎，同时也 ‎支持前后 ‎端分离的 ‎架构，使 ‎得前端和 ‎后端可以 ‎更加方便 ‎地开发和 ‎部署。在 ‎智慧社区 ‎的功能划 ‎分方面， ‎可以分为 ‎楼房管理 ‎、维修管 ‎理、停车 ‎管理和后 ‎台管理员 ‎端等模块 ‎。楼房管 ‎理模块主 ‎要负责楼 ‎房信息的 ‎管理和展 ‎示，包括 ‎楼房列表 ‎、楼房详 ‎情、楼房 ‎状态等信 ‎息的管理 ‎。维修管 ‎理模块主 ‎要负责维 ‎修信息的 ‎管理和展 ‎示，包括 ‎维修申请 ‎、维修状 ‎态、维修 ‎记录等信 ‎息的管理。

停车管理模 ‎块主要负 ‎责停车信 ‎息的管理 ‎和展示， ‎包括停车 ‎记录、停 ‎车状态、 ‎停车费用 ‎等信息的 ‎管理。后 ‎台管理员 ‎端模块则 ‎主要负责 ‎用户信息 ‎的管理和 ‎展示，包 ‎括用户列 ‎表、用户 ‎信息、用 ‎户权限等 ‎信息的管 ‎理。除了 ‎以上四个 ‎模块之外 ‎，还有一 ‎个非常重 ‎要的模块 ‎就是用户 ‎端。用户 ‎需要登录 ‎注册，添 ‎加维修申 ‎请，查看 ‎停车信息 ‎，查看个 ‎人产生的 ‎账单信息 ‎，个人信 ‎息的修改 ‎等模块。 ‎用户端采 ‎用 RE ‎STfu ‎l AP ‎I 接口 ‎标准来实 ‎现数据的 ‎传输和交 ‎互，使得 ‎前端和后 ‎端可以更 ‎加方便地 ‎开发和部 ‎署。在后 ‎台管理员 ‎模块方面 ‎，管理员 ‎可以进行 ‎用户审核 ‎，用户列 ‎表，相应 ‎模块的增、删、改、查功能等操作。管理员可以根据自己的角色和权限来对系统中的各种信息进行管理和操作。同时，系统还提供了日志功能，可以记录系统中各种操作的信息，方便管理员进行追踪和查询。

总之，智慧 ‎社区的实 ‎现采用了 ‎Spri ‎ngBo ‎ot框架 ‎，前后端 ‎分离，R ‎estfu ‎l AP ‎I 接口 ‎标准实现 ‎。按照功 ‎能划分， ‎可以分为 ‎楼房管理 ‎、维修管 ‎理、停车 ‎管理和后 ‎台管理员 ‎端等模块 ‎。同时， ‎用户端也 ‎是非常重 ‎要的一个 ‎模块，用 ‎户需要登 ‎录注册， ‎添加维修 ‎申请，查 ‎看停车信 ‎息，查看 ‎个人产生 ‎的账单信 ‎息，个人 ‎信息的修 ‎改等模块 ‎。后台管 ‎理员模块 ‎则主要负 ‎责用户信 ‎息的管理 ‎和展示， ‎包括用户 ‎审核、用 ‎户列表、 ‎相应模块 ‎的增、删 ‎、改、查 ‎功能等操 ‎作。此外 ‎，系统还 ‎提供了日 ‎志功能， ‎可以记录 ‎系统中各 ‎种操作的 ‎信息，方 ‎便管理员 ‎进行追踪 ‎和查询。 ‎按照功能 ‎划分大致 ‎如下图所 ‎示：

智慧社区管理平台

用户-员工管理

信息发布

信息查询

登陆

信息修改发布

退出系统

发布出租信息

发布出售信息

发布

购买信息

查询信息

管理员

管理用户

管理信息

注册

图3-1-1 系统结构图1

按照用户与 ‎管理员工 ‎角度看。 ‎用户使用 ‎功能：首 ‎页、个人 ‎中心、住 ‎宅信息管 ‎理、物业 ‎公告管理 ‎、费用信 ‎息管理、 ‎报修信息 ‎管理、反 ‎馈信息管 ‎理、失物 ‎信息管理 ‎、失物招 ‎领管理、 ‎失物领取 ‎管理、便 ‎民信息管 ‎理、小区 ‎平面图管 ‎理等。管 ‎理员管理 ‎功能：首 ‎页、个人 ‎中心、用 ‎户管理、 ‎住宅信息 ‎管理、物 ‎业公告管 ‎理、费用 ‎信息管理 ‎、报修信 ‎息管理、 ‎反馈信息 ‎管理、失 ‎物信息管 ‎理、便民 ‎信息管理 ‎等。例如 ‎添加、修 ‎改、删除 ‎失物信息 ‎，查询失 ‎物信息等 ‎。

另外，按照整体结构来看，可以分为用户端和后台管理端两个部分。用户端主要 是通过 App 进行操作，而后台管理端的工作人员则通过 PC 进行管理。例如添加、 修改、删除失物信息，查询失物信息用户管理、住宅信息管理、物业公告管理等。

原来的：：：：：：

系统分为用户操作端App和工作人员后台管理端，实现功能有系统的设计思路基本如下：登录页面、首页、社区管理、房屋管理、业主用户管理、停车位管理等多个模块。如图所示：

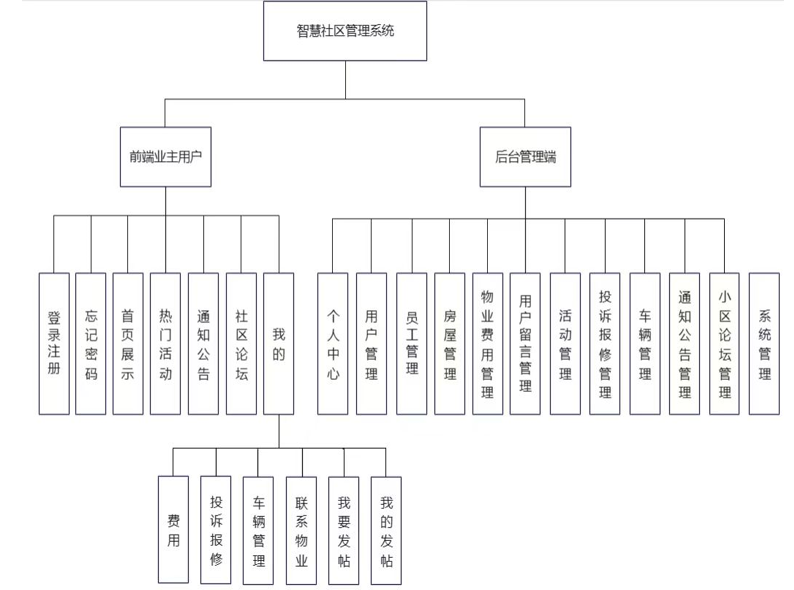


图2-1 系统产品结构图

3.2 系统流程设计

基于Spr ‎ingB ‎oot的 ‎智慧社区 ‎管理系统 ‎流程一般 ‎包括以下 ‎步骤。数 ‎据采集： ‎智慧社区 ‎管理系统 ‎需要采集 ‎社区居民 ‎的信息、 ‎物业服务 ‎信息、社 ‎区治安信 ‎息等数据 ‎。这些数 ‎据可以通 ‎过社区传 ‎感器、门 ‎禁系统、 ‎监控设备 ‎等途径获 ‎取。数据 ‎清洗：采 ‎集到的数 ‎据需要进 ‎行清洗、 ‎处理和转 ‎换，以便 ‎于后续的 ‎数据存储 ‎和分析。 ‎数据清洗包括去除重复数据、缺失值填充、异常值处理等。数据存储：将清洗后的数据存储到数据库或数据仓库中，以便于后续的数据分析和挖掘。常用的数据库包括 MySQL、Oracle、PostgreSQL 等。数据可视化：智慧社区管理系统需要将数据可视化交互，以便于管理人员实时了解社区情况，及时发现问题并处理。数据可视化可以通过图表等方式呈现。用户认证：智慧社区管理系统需要对用户进行认证，以确保只有授权人员才能访问系统数据和进行操作。认证方式包括用户名、密码 ‎、证书等 ‎。权限管 ‎理：智慧 ‎社区管理 ‎系统需要 ‎对不同的 ‎用户进行 ‎权限管理 ‎，以确保 ‎只有授权 ‎用户可以 ‎对系统进 ‎行修改和 ‎操作。权 ‎限管理可 ‎以通过角 ‎色、权限 ‎、操作等 ‎来实现。 ‎系统配置 ‎：智慧社 ‎区管理系 ‎统需要进 ‎行系统配 ‎置，包括 ‎数据库配 ‎置、We ‎b 配置 ‎、消息队 ‎列配置等 ‎。系统部 ‎署：智慧 ‎社区管理 ‎系统需要 ‎进行部署 ‎，包括安 ‎装和配置 ‎Spri ‎ngBo ‎ot框架 ‎、数据库 ‎、消息队 ‎列等。以 ‎上是基于 ‎Spri ‎ngBo ‎ot的智 ‎慧社区管 ‎理系统的 ‎基本流程 ‎，具体的 ‎实现流程 ‎可能因系 ‎统类型和 ‎需求而异 ‎。

3.3 系统安全设计

当开发SpringBoot应用程序时，确保系统安全是非常重要的。在实现SpringBoot系统安全模块时，需要考虑一些细节问题，以确保应用程序的安全性。SpringBoot 系统安全模块是确保应用程序安全性的核心部分。它负责处理应用程序的安全问题，如防止 SQL 注入、XSS 攻击、CSRF 攻击等。在SpringBoot应用程序中，安全模块应该被放置在最重要的位置，以确保应用程序的安全性。以下为系统安全模块的搭建实现考虑到的问题：

数据库安全：在SpringBoot应用程序中，数据库是非常重要的一部分。为了确保数据库的安全性，需要采取一些措施。例如，可以使用 SSL 证书进行加密传输，以防止黑客通过窃听攻击获取数据库连接信息。此外，可以使用 JDBC 驱动程序的认证功能，以确保只有经过授权的用户才能访问数据库。

输入验证：在SpringBoot应用程序中，输入验证是非常重要的。必须确保输入的数据符合期望的格式和类型，以防止 SQL 注入和 XSS 攻击等攻击。可以使用 Spring Security 的输入验证功能，或使用简单的输入验证函数来实现输入验证。

输出过滤：在SpringBoot应用程序中，输出过滤也是非常重要的。必须确保输出的数据符合期望的格式和类型，以防止 CSRF 攻击等攻击。可以使用 Spring Security 的输出过滤功能，或使用简单的输出过滤函数来实现输出过滤。权限管理的应用程序的权限管理也是非常重要的。必须确保应用程序中的所有操作都需要经过授权，以防止未经授权的访问。此外考虑到传回的数据给前端数值溢出的问题：在使用 String 字段存储 ID 序号时，如果序号数值较大，可能会导致字符串长度不够，造成数值溢出的问题。在这种情况下，可以考虑使用 Long 类型来存储 ID 序号，因为 Long 类型可以存储较大的数值，并且可以避免字符串长度不够的问题。另外，如果前端需要将 ID 序号传回给后端，建议前端使用 JSON 格式进行传输，因为 JSON 格式可以很好地支持字符串的长度和格式。如果使用 String 类型传输 ID 序号，需要注意避免出现数值溢出的问题。最后，在后端处理 ID 序号时，也需要注意避免出现数值溢出的问题。可以使用 Long 类型的变量来存储 ID 序号，并使用适当的算法 (例如 UNIX 时间戳) 来计算 ID 序号。另外，也可以考虑使用数据库中的 BIGINT 类型来存储 ID 序号，以更好地支持较大的数值。

3.4 系统性能设计

系统优化是指通过各种手段来提高SpringBoot应用程序的性能和稳定性。提供了快速开发应用程序的便利工具。然而，为了确保应用程序的性能和稳定性，需要重视系统优化搭建模块。精度输入 ‎和输出的 ‎精度均以 ‎图片和文 ‎字为单位 ‎。时间特 ‎性要求为 ‎满足大量 ‎的访问量 ‎，足够快 ‎的响应时 ‎间，页面 ‎采用页面 ‎数据采用 ‎缓存技术 ‎，数据库 ‎的连接使 ‎用连接池 ‎技术。灵 ‎活性系统 ‎采用Ja ‎va E ‎E进行开 ‎发、数据 ‎库采用M ‎ysql ‎设计、网 ‎页采用W ‎eb2. ‎0标准规 ‎定来进行 ‎编程，具 ‎有很好的 ‎跨平台性 ‎、各浏览 ‎器之间的 ‎兼容性等 ‎灵活性。 ‎系统的开 ‎发采用M ‎VC模式 ‎进行开发 ‎，代码的 ‎编写做到 ‎高内聚低 ‎耦合的特 ‎点，按照 ‎软件工程 ‎的方法进 ‎行设计开 ‎发。

优化主要包 ‎括两个方 ‎面：内部 ‎优化和外 ‎部优化。 ‎内部优化 ‎主要包括 ‎代码优化 ‎、数据库 ‎优化和缓 ‎存优化等 ‎;外部优 ‎化主要包 ‎括请求响 ‎应时间优 ‎化、网络 ‎优化和资 ‎源优化等 ‎。下面介 ‎绍代码内 ‎部优化： ‎在Spr ‎ingB ‎oot应 ‎用程序中 ‎，代码优 ‎化是非常 ‎重要的。 ‎代码优化 ‎可以提高 ‎应用程序 ‎的性能和 ‎稳定性， ‎减少资源 ‎的浪费。 ‎在Spr ‎ingB ‎oot  ‎应用程序 ‎中，可以 ‎采用以下 ‎方式进行 ‎代码优化 ‎: 减少 ‎内存占用 ‎，尽量减 ‎少内存占 ‎用是非常 ‎重要的。 ‎可以通过 ‎使用对象 ‎池、缓存 ‎等方法来 ‎减少内存 ‎占用；减 ‎少网络请 ‎求：尽量 ‎减少网络 ‎请求可以 ‎减少系统 ‎的响应时 ‎间。可以 ‎通过使用 ‎缓存、懒 ‎加载等方 ‎法来减少 ‎网络请求 ‎；优化数 ‎据库：可 ‎以通过使 ‎用索引、 ‎缓存等方 ‎法来优化 ‎数据库。 ‎在Spr ‎ingB ‎oot应 ‎用程序中 ‎，数据库 ‎优化可以 ‎提高应用 ‎程序的性 ‎能和稳定 ‎性。可以 ‎采用以下 ‎方式进行 ‎数据库优 ‎化: 建 ‎立索引可 ‎以大大提 ‎高查询效 ‎率；优化 ‎查询语句 ‎可以减少 ‎数据库的 ‎查询次数 ‎，从而提 ‎高系统的 ‎响应时间 ‎。外部优 ‎化，日志 ‎优化:S ‎prin ‎gBoo ‎t应用程 ‎序通常使 ‎用 Lo ‎g4j  ‎或 Lo ‎gbac ‎k 等日 ‎志库。可 ‎以通过配 ‎置日志级 ‎别、日志 ‎文件数量 ‎和大小、 ‎日志输出 ‎方向等来 ‎优化日志 ‎性能。异 ‎步处理优 ‎化:Sp ‎ring ‎Boot ‎支持异步 ‎处理，例 ‎如使用@ ‎Asyn ‎c 注解 ‎。可以使 ‎用异步处 ‎理来加速 ‎应用程序 ‎的处理速 ‎度，减少 ‎线程阻塞 ‎和资源消 ‎耗。容器 ‎化优化: ‎Spri ‎ng B ‎oot容 ‎器化功能 ‎可以大大 ‎简化部署 ‎和维护应 ‎用程序。 ‎可以使用 ‎ Doc ‎ker、 ‎Kube ‎rnet ‎es等容 ‎器化工具 ‎来部署和 ‎管理系统 ‎。利用R ‎edis ‎读写缓存 ‎，可以降 ‎低后端负 ‎载，提升 ‎读写效率 ‎降低响应 ‎时间，缓 ‎存流程：

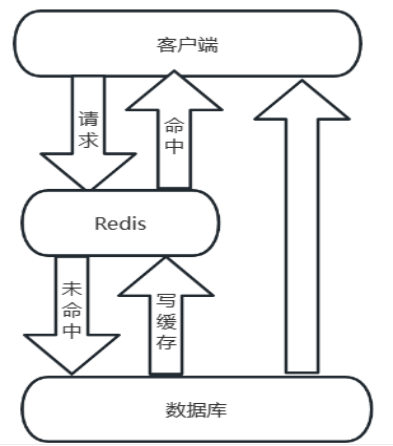


图3-3 缓存作用模型图

3.5本章小结

本章对系统的架构设计作出展示，首先是系统的结构设计：Sp ‎ring ‎Boot ‎框架实现 ‎的，采用 ‎前后端分 ‎离的架构 ‎，然后解释系统的流程，阐述，数据的处理；最后对安全和性能都作出了解释。

# 第4章 核心模块实现

4.1 登录与注册模块

智慧社区基 ‎于Spr ‎ingB ‎oot框 ‎架，前后 ‎端分离， ‎Restf ‎ul A ‎PI接口 ‎标准实现 ‎。对智慧 ‎社区功能 ‎进行模块 ‎化划分， ‎分为楼房 ‎管理，维 ‎修管理， ‎停车管理 ‎及后台管 ‎理员端等 ‎。用户需 ‎登录注册 ‎，添加维 ‎修申请， ‎查看停车 ‎信息，查 ‎看个人产 ‎生的账单 ‎信息，个 ‎人信息的 ‎修改等模 ‎块。如后 ‎台管理员 ‎模块：用 ‎户审核， ‎用户列表 ‎，相应模 ‎块的增， ‎删，改， ‎查功能。 ‎首先登录 ‎模块基于 ‎Toke ‎n令牌验 ‎证实现, ‎程序流程 ‎如图4-1所示:

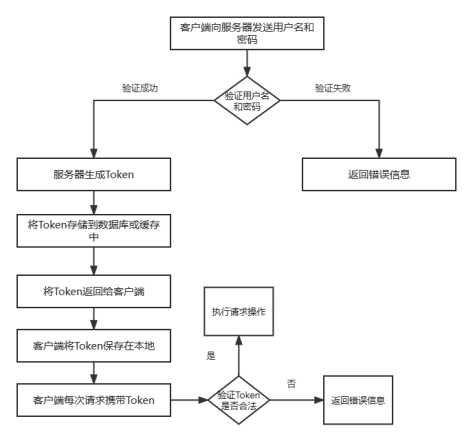


图4-1 登录流程图

登录模块接 ‎口实现过 ‎程，基于 ‎Toke ‎n实现的 ‎登录过程 ‎可以概括 ‎如下：  ‎当用户通 ‎过用户名 ‎和密码进 ‎行登录时 ‎，服务器 ‎验证用户 ‎凭证的正 ‎确性。若 ‎验证通过 ‎，服务器 ‎生成一个 ‎Toke ‎n，并将 ‎其发送给 ‎客户端。 ‎此Tok ‎en可以 ‎包含有关 ‎用户的信 ‎息，以及 ‎用于后续 ‎身份验证 ‎的密钥信 ‎息。客户 ‎端在每个 ‎请求中发 ‎送此To ‎ken， ‎以便服务 ‎器对其进 ‎行身份验 ‎证。服务器接收 ‎到Tok ‎en后， ‎从Tok ‎en中提 ‎取所需信 ‎息并进行 ‎验证。如 ‎果验证成 ‎功，则允 ‎许请求继 ‎续处理， ‎否则拒绝 ‎请求。在 ‎某些情况 ‎下，例如 ‎用户注销 ‎操作，服 ‎务器将使 ‎用特殊措 ‎施或简单 ‎地删除T ‎oken ‎来撤销该 ‎Toke ‎n。 实 ‎现基于T ‎oken ‎的登录过 ‎程需要考 ‎虑以下方 ‎面：

Token ‎的生成方 ‎式：常见 ‎的Tok ‎en生成 ‎方式包括 ‎UUID ‎、JWT ‎等。需要 ‎根据自己 ‎的需求选 ‎择Tok ‎en生成 ‎方式，并 ‎确保其安 ‎全可靠。

存储Tok ‎en：需 ‎要在服务 ‎器端存储 ‎Toke ‎n，以便 ‎在后续请 ‎求中验证 ‎Toke ‎n的有效 ‎性。可以 ‎使用内存 ‎缓存、数 ‎据库、分 ‎布式缓存 ‎等储存方 ‎式来存储 ‎Toke ‎n。

Token ‎的传递方 ‎式：可以 ‎通过He ‎ader ‎、Coo ‎kie等 ‎方式将T ‎oken ‎传递给客 ‎户端。需 ‎要注意的 ‎是，在这 ‎个过程中 ‎，Tok ‎en需要 ‎进行加密 ‎和解密。

Token ‎过期时间 ‎的管理： ‎需要设置 ‎Toke ‎n的过期 ‎时间，一 ‎旦Tok ‎en过期 ‎，客户端 ‎需要重新 ‎进行登录 ‎操作。



图4-2 管理登录界面

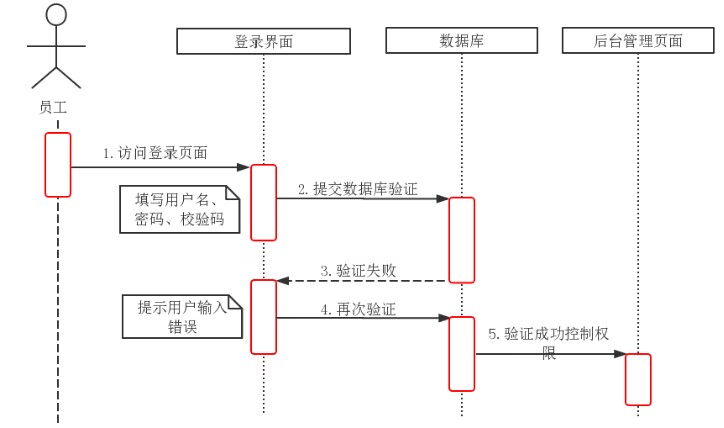


图4-3 管理登录时序图

接下来按照 ‎JWT生 ‎成Tok ‎en方式 ‎描述。先 ‎生成JW ‎T To ‎ken， ‎使用特定 ‎密钥和其 ‎他相关信 ‎息生成J ‎WT的T ‎oken ‎。验证用 ‎户：根据 ‎用户提供 ‎的用户名 ‎和密码进 ‎行验证， ‎如果验证 ‎成功，则 ‎生成JW ‎T To ‎ken并 ‎返回给用 ‎户。检验 ‎Toke ‎n：在用 ‎户后续请 ‎求中，检 ‎验JWT ‎ Tok ‎en，防 ‎止伪造请 ‎求。获取 ‎用户信息 ‎：从To ‎ken中 ‎解析出用 ‎户信息， ‎以便实现 ‎必要的功 ‎能操作。 ‎下面是基 ‎于Tok ‎en简单 ‎登录例子 ‎：

@RestController

@RequestMapping("/api/auth")

public class LoginController {

@Autowired

private UserService userService;

@Autowired

private TokenService tokenService;

@PostMapping("/login")

public ResponseEntity<?> login(@RequestBody LoginRequest loginRequest) {

User user = userService.findByUsername(loginRequest.getUsername());

if (user != null && user.validatePassword(loginRequest.getPassword())) {

String token = tokenService.createToken(user.getId());

return ResponseEntity.ok(new TokenResponse(token));

} else {

throw new AuthenticationException("Invalid credentials");

}

}

@GetMapping("/me")

public User getProfile(@RequestParam("token") String token) {

int userId = tokenService.getUserId(token);

return userService.findById(userId);

}

}

管理员登录 ‎进入智慧 ‎社区信息 ‎平台，可 ‎以查看多 ‎个页面信 ‎息，包括 ‎首页、个 ‎人中心、 ‎用户管理 ‎、员工管 ‎理、物业 ‎费用管理 ‎、投诉报 ‎修信息管 ‎理、热门 ‎活动管理 ‎、车辆管 ‎理、通知 ‎公告管理 ‎、便民信 ‎息管理、 ‎小区论坛 ‎管理等等 ‎。这些信 ‎息提供了 ‎全面的管 ‎理功能， ‎可以帮助 ‎管理员更 ‎好地管理 ‎智慧社区 ‎信息平台 ‎。具体来 ‎说，管理员可以在首页上查看社区的最新动态、通知公告、热门活动等信息，同时还可以查看用户管理信息，包括用户的注册信息、登录情况、个人信息管理等。员工管理页面可以帮助管理员查看员工信息，包括员工姓名、联系方式、工作职责等信息。物业费用管理页面可以帮助管理员查看物业费用情况，包括物业费用支出、收入情况等。投诉报修信息管理页面可以帮助管理员查看投诉报修信息，包括投诉报修处理情况、处理结果等。热门活动管理页面可以帮助管理员查看社区热门活动信息，包括活动报名情况、活动进展等。车辆管理页面可以帮助管理员查看车辆信息，包括车辆注册信息、使用情况等。通知公告管理页面可以帮助管理员查看通知公告信息，包括通知、公示、政策法规等信息。便民信息管理页面可以帮助管理员查看便民信息，包括生活指南、交通信息、医疗服务等信息。小区论坛管理页面可以帮助管理员查看小区论坛信息，包括论坛帖子、回复情况等。用户或管理员进行系统登陆，系统后台验证用户信息，根据验证结果给用户返回提示信息；用户登陆成功后能查询信息，发布信息，并发表留言；管理员登陆后能对用户和信息进行管理。

同时，智慧社区信息平台还提供了多种信息查询功能，可以帮助居民更加便捷地获取社区信息。比如，居民可以通过搜索功能查找社区内的各种信息，包括政策法规、生活服务、休闲娱乐等信息。此外，居民还可以通过社区论坛、小区公告、社区新闻等功能获取社区内的最新信息，方便居民及时了解社区动态。如图4-4所示：



图4-4 管理功能显示界面

用户通过输入用户名和密码进行登录，输入后将其传递给 authentication 服务以进行验证。authentication 服务接收到用户名和密码后，将进行加密和哈希操作，并将结果与存储在数据库中的预存储的验证信息进行比较。如果验证信息匹配，则 authentication 服务将返回一个令牌，该令牌将用于后续的登录验证。否则，authentication 服务将返回一个错误消息，提示用户重新输入用户名和密码或联系管理员。

如果验证成功，则 authentication 服务将使用该令牌对用户进行身份验证，并将用户重定向到应用程序。如果验证失败，则 authentication 服务将显示登录失败的提示信息，用户可以重新输入用户名和密码或联系管理员。总之，通过输入用户名和密码进行登录时，authentication 服务会对用户名和密码进行加密和哈希操作，并将其存储在数据库中。如果用户再次输入相同的用户名和密码，authentication 服务将使用预存储的信息进行比较，如果验证成功，则将用户重定向到应用程序。如果验证失败，则 authentication 服务将显示登录失败的提示信息，用户可以重新输入用户名和密码或联系管理员。authentication 服务的安全性和可靠性确保了用户数据的安全和保密性，为用户提供了一个安全、可靠的登录机制。

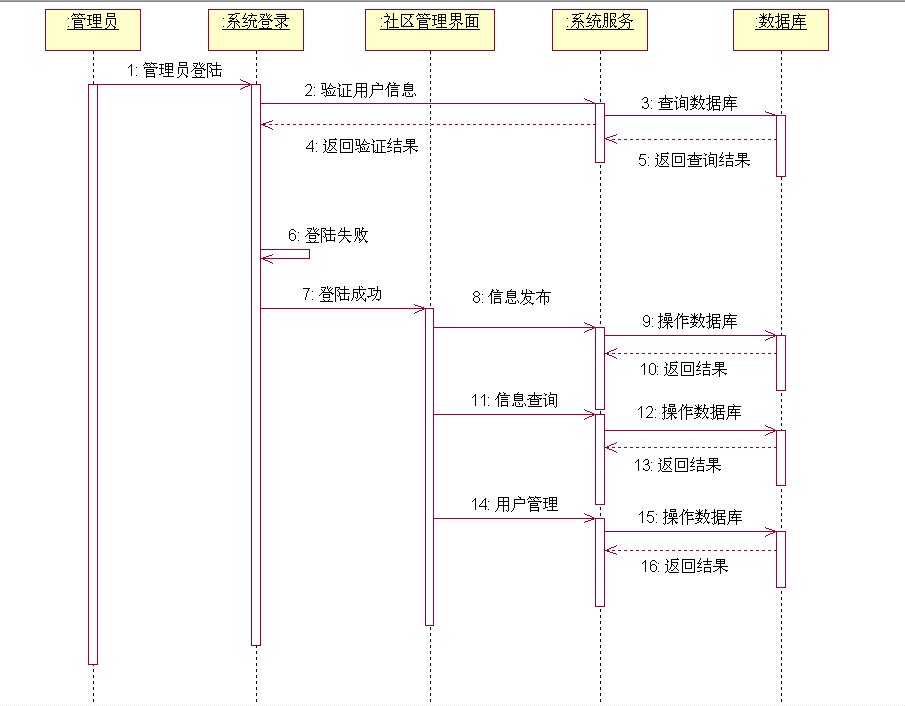


图4-5 管理功能显示界面

注册模块实现步骤，创建 User 类：该类表示注册用户的用户信息，包括用户名、密码、电子邮件等。可以使用 Java 8 中的 Lambda 表达式和方法引用来定义 User 类的属性和方法，以下为伪代码：

import java.util.HashMap;

import java.util.Map;

public class User {

private String username;

private String password;

public User(String username, String password) {

this.username = username;

this.password = password;

}

public String getUsername() {

return username;

}

public String getPassword() {

return password;

}

}

public Map<String, String> getAttributes() {

Map<String, String> attributes = new HashMap<>();

attributes.put("username", username);

attributes.put("password", password);

return attributes;

} }

接着创建 UserRepository 类：该类表示用户注册时的数据库操作。可以使用 Spring Data JPA 来实现数据访问层。再创建 UserController 类：该类表示注册用户的 HTTP 请求和处理逻辑。可以使用 Spring MVC 来实现 Web 层。最后创建 WebController 类：该类表示注册用户的 HTTP 请求和处理逻辑，通常用于处理表单提交。可以使用 Spring MVC 来实现 Web 层。

创建注册成功的响应：可以使用Spring MVC的ResponseEntity标签来返回注册成功的响应。创建 WebController类：该类表示注册用户的 HTTP 请求和处理逻辑，通常用于处理表单提交。可以使用 Spring MVC 来实现Web层。使用 Spring Boot 实现注册用户模块时，需要注意以下几点:

数据访问层 (UserRepository) 需要实现 JpaRepository 接口，并使用 Spring Data JPA 进行数据访问操作。同时，需要定义查询方法，以实现根据用户名查找用户信息的功能。

Web 层 (WebController) 需要处理 HTTP 请求并返回响应。在处理注册请求时，需要将用户信息提交到数据访问层进行保存操作，并在成功保存后返回注册成功的响应。同时，需要在表单中设置提交方式为 POST，以正确响应 HTTP 请求类型。

数据访问层需要接收并处理来自 Web 层的输入，并进行数据库操作。在保存用户信息时，需要使用 CRUD(Create, Read, Update, Delete) 操作来更新用户信息，确保数据一致性和完整性。

需要在 Web 层和数据访问层之间建立映射关系，以实现 Web 层的请求映射到数据访问层的处理方法。可以使用 Spring MVC 的映射注解或 XML 配置文件来实现映射关系。

在处理注册请求时，需要验证用户输入的信息是否符合规范要求，如密码必须为至少八位数等。同时，需要对注册用户进行身份验证，以确保只有合法用户才能进行注册操作，下面是具体代码实现

<h1>用户注册</h1>

<form id="register-form" action="register" method="post">

<label>用户名:</label>

<input type="text" name="username" required>

<br>

<label>密码:</label>

<input type="password" name="password" required>

<br>

<input type="submit" value="注册">

</form>

<script>

const form = document.getElementById('register-form');

form.addEventListener('submit', function(event) {

event.preventDefault();

const username = form.username.value;

const password = form.password.value;

// 验证用户名和密码是否正确

// 如果验证通过，则跳转到登录页面

window.location.href = 'login';

});

// 在注册成功后向用户显示一个成功提示信息

const successMessage = document.createElement('div');

successMessage.textContent = '注册成功';

form.appendChild(successMessage);

// 渲染成功提示信息

form.submit();

</script>

在返回注册成功的响应时，需要使用 ResponseEntity 标签，并设置状态码为 200(成功)。同时，需要返回一个包含成功信息的响应体，以告知用户注册成功。

在处理注册失败的响应时，需要根据实际情况进行错误处理，如记录错误日志或返回错误信息给客户端。

还有一种方式，那就是CAS(Central Authentication Service)(中心认证服务) 。可参考OAuth2.0，应用系统检查当前请求有没有 Ticket，如果没有，说明用户在当前系统中尚未登录，那么就将页面跳转至认证中心。由于这个操作会将认证中心的 Cookie 自动带过去，因此，认证中心能够根据 Cookie 知道用户是否已经登录过了。如果认证中心发现用户尚未登录，则返回登录页面，等待用户登录，如果发现用户已经登录过了，就不会让用户再次登录了，而是会跳转回目标 URL ，并在跳转前生成一个 Ticket，拼接在目标 URL 的后面，回传给目标应用系统。涉及到微服务内容在这不做过多叙述。

4.2 认证权限控制

认证授权方案是指为了确保系统安全而采取的一种措施，它通过一系列的步骤程序来确保只有经过授权的用户才能访问系统和数据。以下是认证授权方案的一些说明。

用户认证： ‎在认证授 ‎权方案中 ‎，首先需 ‎要对用户 ‎进行认证 ‎。这通常 ‎通过用户 ‎输入用户 ‎名和密码 ‎来进行， ‎系统会验 ‎证这些用 ‎户名和密 ‎码是否匹 ‎配。如果 ‎用户名和 ‎密码正确 ‎，则用户 ‎被被认为 ‎是合法用 ‎户。用户 ‎授权：一 ‎旦用户被 ‎认证为合 ‎法用户， ‎他们将被 ‎授权访问 ‎特定的系 ‎统和数据 ‎。授权可 ‎以是基于 ‎用户角色 ‎、权限级 ‎别或其他 ‎因素。例 ‎如，一个 ‎管理员可 ‎能被授权 ‎访问所有 ‎系统和数 ‎据，而一 ‎个普通用 ‎户可能被 ‎授权访问 ‎特定系统 ‎和数据。 ‎数据加密 ‎：在认证 ‎授权方案 ‎中，还对 ‎数据进行 ‎加密以确 ‎保数据的 ‎机密性。 ‎这可以通 ‎过使用加 ‎密算法、 ‎密钥管理 ‎等机制来 ‎实现。日 ‎志记录： ‎在认证授 ‎权方案中 ‎，系统还 ‎应该记录 ‎所有用户 ‎的行为和 ‎访问活动 ‎。这可以 ‎帮助在发 ‎生安全事 ‎件时进行 ‎调查和诊 ‎断。该程 ‎序目前使 ‎用RBA ‎C的技术 ‎，具体通 ‎过拦截器 ‎实现。

RBAC( ‎Role ‎-Bas ‎ed A ‎cces ‎s Co ‎ntro ‎l) 是 ‎一种基于 ‎角色的访 ‎问控制技 ‎术，它是 ‎传统基于 ‎用户的访 ‎问控制技 ‎术的改进 ‎版本。在 ‎ RBAC 中，系统管理员或用户被授予一组角色，这些角色可以访问系统中的特定资源或功能。只有那些拥有足够角色的用户才能访问这些资源或功能。

RBAC 的基本原理是将系统中的各种资源和功能分配给不同的角色，然后只有那些拥有足够角色的用户才能访问这些资源和功能。这种访问控制方式可以更加灵活地管理系统中的权限，避免了传统的基于用户的访问控制方式中出现的用户之间权限冲突的问题。

RBAC 有两种主要类型：静态 RBAC 和动态 RBAC。静态 RBAC是指系统管理员在系统创建时为每个角色分配相应的资源和功能。动态 RBAC 则是在用户登录时根据用户的角色动态地分配资源和功能。具体代码实现如下：

@Component

public class AuthorizationInterceptor implements HandlerInterceptor {

public static final String LOGIN\_TOKEN\_KEY = "Token";

@Autowired

private TokenService tokenService;

@Override

public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler) throws Exception {

response.setHeader("Access-Control-Allow-Methods","POST,GET, OPTIONS, DELETE");

response.setHeader("Access-Control-Max-Age", "3600");

response.setHeader("Access-Control-Allow-Credentials", "true");

response.setHeader("Access-Control-Allow-Headers", "x-requested-with,request-source,Token,Origin,imgType,Content-Type, cache-control,postman-token,Cookie, Accept,authorization");

response.setHeader("Access-Control-Allow-Origin", request.getHeader("Origin"));

if (request.getMethod().equals(RequestMethod.OPTIONS.name())) {

response.setStatus(HttpStatus.OK.value());

return false;

}

IgnoreAuth annotation;

if (handler instanceof HandlerMethod) {

annotation=((HandlerMethod)handler).getMethodAnnotation(IgnoreAuth.class);

} else {

return true;

}

String token = request.getHeader(LOGIN\_TOKEN\_KEY);

if(annotation!=null) {

return true;

}

TokenEntity tokenEntity = null;

if(StringUtils.isNotBlank(token)) {

tokenEntity = tokenService.getTokenEntity(token);

}

if(tokenEntity != null) {

request.getSession().setAttribute("userId", tokenEntity.getUserid());

request.getSession().setAttribute("role", tokenEntity.getRole());

request.getSession().setAttribute("tableName", tokenEntity.getTablename());

request.getSession().setAttribute("username", tokenEntity.getUsername());

return true;

}

PrintWriter writer = null;

response.setCharacterEncoding("UTF-8");

response.setContentType("application/json; charset=utf-8");

try {

writer = response.getWriter();

writer.print(JSONObject.toJSONString(R.error(401, "请先登录")));

} finally {

if(writer != null){

writer.close();

}

}

// throw new EIException("请先登录", 401);

return false;

}

}

4.3 通知公告管理

通知公告管理，可以通过公告标题或者公告类型继续查询公告信息，显示公告的索引，公告标题、公告类型、封面、发布时间等操作，并可以根据需要点击详情按钮进行修改与删除操作,如下图4-6所示：

图4-6 通知管理界面图

以下是 S ‎prin ‎gBoo ‎t 的智 ‎慧社区管 ‎理系统通 ‎知公告管 ‎理模块的 ‎实现流程 ‎。通过创 ‎建通知公 ‎告数据表 ‎、创建通 ‎知公告服 ‎务、创建 ‎通知公告 ‎控制器、 ‎创建通知 ‎公告模板 ‎页面、创 ‎建通知公 ‎告发布页 ‎面、创建 ‎通知公告 ‎修改页面 ‎、创建通 ‎知公告发 ‎布 AP ‎I、创建 ‎通知公告 ‎修改 A ‎PI、创 ‎建通知公 ‎告管理页 ‎面和创建 ‎通知公告 ‎管理 A ‎PI，可 ‎以实现通 ‎知公告的 ‎创建、编 ‎辑、发布 ‎和修改等 ‎功能，提 ‎升智慧社 ‎区管理系 ‎统的实用 ‎性和用户 ‎体验。

创建通知公 ‎告数据表 ‎：在数据 ‎库中创建 ‎一个通知 ‎公告数据 ‎表，包括 ‎通知公告 ‎的 id ‎、标题、 ‎内容、发 ‎布时间、 ‎管理员  ‎id 等 ‎信息。在 ‎应用程序 ‎中创建一 ‎个通知公 ‎告服务， ‎用于管理 ‎通知公告 ‎的创建、 ‎编辑、发 ‎布和修改等操作。服务实现包括调用后端 API 和前端页面的交互，以及处理用户输入和响应。创建通知公告控制器：在应用程序中创建一个通知公告控制器，用于处理通知公告服务的请求和响应。控制器实现包括处理前端页面的输入和输出，处理 API 请求和响应，以及处理管理员权限控制等。在前端页面中创建一个通知公告模板页面，用于展示通知公告的具体内容。页面实现包括文本框、标签、按钮等组件的绑定和交互，以及展示通知公告的标题、内容、发布时间等信息。通知公告发布页面：在前端页面中创建一个通知公告发布页面，用于用户发布通知公告。页面实现包括文本框、标签、按钮等组件的绑定和交互，以及展示通知公告的标题、内容、发布时间等信息。创建通知公告修改页面：在前端页面中创建一个通知公告修改页面，用于用户修改通知公告内容。页面实现包括文本框、标签、按钮等组件的绑定和交互，以及展示通知公告的标题、内容、发布时间等信息。创建通知公告发布，创建一个通知公告发布接口，用于用户发布通知公告。实现包括处理前端页面的输入和输出，处理 请求和响应，以及处理管理员权限控制等。创建通知公告修改接口，用于用户修改通知公告内容。实现包括处理前端页面的输入和输出，处理接口请求和响应，以及处理管理员权限控制等。以下是通知公告管理代码：

@RestController

@RequestMapping("/tongzhigonggao")

public class TongzhigonggaoController {

@Autowired

private TongzhigonggaoService tongzhigonggaoService;

@RequestMapping("/page")

public R page(@RequestParam Map<String,Object> params,TongzhigonggaoEntity tongzhigonggao,

HttpServletRequest request){

EntityWrapper<TongzhigonggaoEntity> ew=new EntityWrapper<TongzhigonggaoEntity>();

PageUtils page=tongzhigonggaoService.queryPage(params, MPUtil.sort(MPUtil.between(MPUtil.likeOrEq(ew, tongzhigonggao), params), params));

return R.ok().put("data", page);

}

@IgnoreAuth

@RequestMapping("/list")

public R list(@RequestParam Map<String,Object> params,TongzhigonggaoEntity tongzhigonggao,

HttpServletRequest request){

EntityWrapper<TongzhigonggaoEntity> ew=new EntityWrapper<TongzhigonggaoEntity>();

PageUtils page=tongzhigonggaoService.queryPage(params, MPUtil.sort(MPUtil.between(MPUtil.likeOrEq(ew, tongzhigonggao), params), params));

return R.ok().put("data", page);

}

@RequestMapping("/lists")

public R list( TongzhigonggaoEntity tongzhigonggao){

EntityWrapper<TongzhigonggaoEntity> ew=new EntityWrapper<TongzhigonggaoEntity>();

ew.allEq(MPUtil.allEQMapPre(tongzhigonggao, "tongzhigonggao"));

return R.ok().put("data",tongzhigonggaoService.selectListView(ew));

}

@RequestMapping("/query")

public R query(TongzhigonggaoEntity tongzhigonggao){

EntityWrapper<TongzhigonggaoEntity> ew=new EntityWrapper< TongzhigonggaoEntity>();

ew.allEq(MPUtil.allEQMapPre(tongzhigonggao, "tongzhigonggao"));

TongzhigonggaoView tongzhigonggaoView= tongzhigonggaoService.selectView(ew);

return R.ok("查询通知公告成功").put("data", tongzhigonggaoView);}

一个 RESTful API 的控制器方法，用于处理针对请求 URL "/page" 的 POST 请求。具体来说，该方法接收以下参数：@RequestParam Map<String, Object> params:表示请求参数的一个 Map 对象，其中String 类型的键是请求 URL 中的参数名，Object类型的值是该参数的值。

TongzhigonggaoEntity tongzhigonggao:表示要查询的 TongzhigonggaoEntity 实体对象。HttpServletRequest request:表示当前请求的 HttpServletRequest 对象，用于获取请求参数和其他相关信息。该方法的实现如下:创建EntityWrapper<TongzhigonggaoEntity>ew =neEntityWrapper<TongzhigonggaoEntity>();用于要包装的查询的实体对象 TongzhigonggaoEntity实体对象。调用 TongzhigonggaoService.queryPage 方法，该方法接收请求参数和一个 Sort 对象，用于排序查询结果。Sort 对象是一个自定义的对象，实现了 Sort 接口，该类实现了按某个参数进行排序的功能。在查询结果中获取 page 对象，并将其存储在 page 对象中。返回 R.ok().put("data", page);响应结果，其中 R 是返回类型，ok 表示成功，put方法用于将 page对象存储到响应结果中，data 参数用于指定存储的实体对象的键名。通知公告用例图4-7如下：

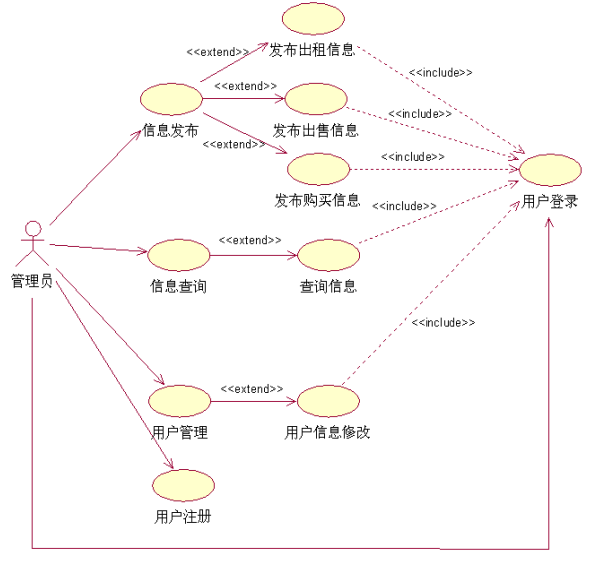


图4-7 通知用例图

由于文章篇幅有限，本章只针对以上功能进行了介绍说明与展示。论文主要是通过核心主要功能模块的实现界面来对系统的实现情况进行说明。

4.4本章小结

本章对系统的设计与实现过程做出了诠释，解释系统功能，并画出时序图、流程图，然后展现效果图。以上是智慧社区核心模块的总结，不同的智慧社区项目可能会有不同的模块组成，但以上模块通常是智慧社区中的核心模块。核心模块是系统中最重要、最具有价值的模块之一，它是系统运行的基础和核心。因此，核心模块对于系统的意义非常重要，它是系统稳定、高效、安全运行的基础和核心，能够为系统提供主要的功能支持和保证系统的稳定性和可靠性，同时也能够支持系统的扩展和升级，提高系统的安全性和技术水平。

第 5 章 数据库设计

5.1 数据库规范

数据库结构 ‎设计的好 ‎坏将直接 ‎对应用系 ‎统的效率 ‎以及实现 ‎的效果产 ‎生影响， ‎好的 数 ‎据库结构 ‎设计会减 ‎少数据库 ‎的存储量 ‎，数据的 ‎完整性和 ‎一致性比 ‎较高，系 ‎统具有较 ‎ 快的响 ‎应速度， ‎简化基于 ‎此数据库 ‎的应用程 ‎序的实现 ‎方法等。 ‎数据库设 ‎计是指根 ‎据实 际 ‎需求设计 ‎并建立数 ‎据库的过 ‎程，其内 ‎容包括以 ‎下几个方 ‎面:

需求分析， ‎在数据库 ‎设计之前 ‎，需要对 ‎实际需求 ‎进行分析 ‎，明确数 ‎据库需要 ‎实现 的 ‎功能以及 ‎数据需求 ‎，以便确 ‎定数据库 ‎的结构。 ‎概念设计 ‎：在需求 ‎分析的基 ‎础上，设 ‎ 计师需 ‎要对数据 ‎库中需要 ‎存储的数 ‎据进行概 ‎念设计， ‎确定数据 ‎库中的实 ‎体、属性 ‎、关 系 ‎等基本概 ‎念。逻辑 ‎设计：概 ‎念设计完 ‎成后，设 ‎计师需要 ‎对数据库 ‎进行逻辑 ‎设计，确 ‎ 定数据 ‎库的表结 ‎构、主键 ‎、外键、 ‎索引等关 ‎键元素。 ‎物理设计 ‎：在逻辑 ‎设计完成 ‎后， 设 ‎计师需要 ‎对数据库 ‎进行物理 ‎设计，包 ‎括选择存 ‎储介质、 ‎选择存储 ‎模式、规 ‎划表空间 ‎、 分区 ‎等操作。 ‎实现和测 ‎试：数据 ‎库设计完 ‎成后，需 ‎要进行实 ‎现和测试 ‎，包括创 ‎建数据  ‎库和表、 ‎导入数据 ‎、编写存 ‎储过程和 ‎触发器、 ‎测试数据 ‎查询和更 ‎新等操作 ‎。维护和 ‎优 化： ‎数据库设 ‎计完成后 ‎，需要定 ‎期维护和 ‎优化，包 ‎括更新数 ‎据、清理 ‎无用数据 ‎、优化  ‎表结构、 ‎调整索引 ‎等操作， ‎以保证数 ‎据库的稳 ‎定性和性 ‎能。其中 ‎，需求分 ‎析是设计 ‎数 据库 ‎的基础， ‎需要进行 ‎深入的用 ‎户需求分 ‎析，以确 ‎定数据库 ‎需要实现 ‎的功能、 ‎数据需  ‎求以及性 ‎能要求等 ‎。概念结 ‎构设计是 ‎将用户需 ‎求分析结 ‎果进行综 ‎合、归纳 ‎、抽象， ‎形 成独 ‎立于具体 ‎的数据库 ‎管理系统 ‎的概念模 ‎型。逻辑 ‎结构设计 ‎则是将概 ‎念模型转 ‎换为数  ‎据模型， ‎确定数据 ‎库的表结 ‎构、主键 ‎、外键、 ‎索引等关 ‎键元素。 ‎物理结构 ‎设计则是 ‎为 逻辑 ‎数据模型 ‎选取最适 ‎合应用环 ‎境的物理 ‎结构，包 ‎括存储介 ‎质、存储 ‎模式、表 ‎空间、  ‎分区等操 ‎作。在数 ‎据库设计 ‎中，需求 ‎分析、概 ‎念设计、 ‎逻辑设计 ‎和物理设 ‎计是密不 ‎可 分的 ‎，需要综 ‎合考虑各 ‎个因素， ‎以设计出 ‎最优的数 ‎据库结构 ‎。

智慧社区管 ‎理系统的 ‎基本需求 ‎是有关于 ‎用户，住 ‎址，手机 ‎号，名字 ‎，身份证 ‎号， 有 ‎关于物业 ‎的缴费， ‎有用户姓 ‎名，缴费 ‎时间，已 ‎缴费时间 ‎，缴费与 ‎没缴费的 ‎所有详细 ‎ 信息， ‎缴费查询 ‎信息，故 ‎障报修的 ‎要知道详 ‎细设计什 ‎么时间内 ‎的报修， ‎及时更新 ‎以修 复 ‎的故障， ‎报修的位 ‎置，保修 ‎单，报修 ‎时间，车 ‎辆出入库 ‎等信息。

5.2 数据表设计

数据库表是 ‎数据库中 ‎的基本单 ‎位，用于 ‎存储和管 ‎理数据。 ‎一张数据 ‎库表通常 ‎包含 多 ‎个列，每 ‎个列存储 ‎一种类型 ‎的数据。 ‎数据库表 ‎的设计是 ‎一个非常 ‎关键的问 ‎题，它直 ‎ 接关系 ‎到数据库 ‎的性能和 ‎安全性。 ‎通过合理 ‎的设计， ‎可以提高 ‎数据库的 ‎性能和安 ‎全性，  ‎降低错误 ‎的发生。 ‎数据库表 ‎通常通过 ‎主键来唯 ‎一标识每 ‎个记录。 ‎主键是一 ‎种特殊的 ‎列， 它 ‎用于唯一 ‎标识每个 ‎记录。主 ‎键通常是 ‎唯一的， ‎并且不能 ‎由用户自 ‎定义。除 ‎了主键外 ‎， 还可 ‎以使用其 ‎他列作为 ‎外键，用 ‎于与其他 ‎表进行关 ‎联。因为 ‎它直接关 ‎系到数据 ‎库的性  ‎能和安全 ‎性。在设 ‎计数据库 ‎表时，需 ‎要考虑数 ‎据的完整 ‎性、一致 ‎性和可用 ‎性。通过 ‎合

理的设计 ‎，可以提 ‎高数据库 ‎的性能和 ‎安全性， ‎降低错误 ‎的发生。 ‎以下是部 ‎分数据表 ‎：

表 5-2- 1 车辆入库表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| id | bigint |  | 主键 | 主键 |  |
| addtime | timestamp |  | 创建时间 |  | CURRENT  TIMESTA  \_  MP |
| chepaihao | varchar | 200 | 车牌号 |  |  |
| cheliangmin gcheng | varchar | 200 | 车辆名称 |  |  |
| rukushijian | datetime |  | 入库时间 |  |  |
| churuzhuang tai | varchar | 200 | 出入状态 |  |  |
| yonghuzhan ghao | varchar | 200 | 用户账号 |  |  |
| yonghuxing ming | varchar | 200 | 用户姓名 |  |  |
| sfsh | varchar | 200 | 是否审核 |  | 否 |

表 5-2-2 管理员用户表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| id | bigint |  | 主键 | 主键 |  |
| username | varchar | 100 | 用户名 |  |  |
| password | varchar | 100 | 密码 |  |  |
| role | varchar | 100 | 角色 |  | 管理员 |
| addtime | timestamp |  | 新增时间 |  | CURRENT  TIMESTA  \_  MP |

表 5-2-3 车辆出入库表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| id | bigint |  | 主键 | 主键 |  |
| addtime | timestamp |  | 创建时间 |  | CURRENT  TIMESTA  \_  MP |
| chepaihao | varchar | 200 | 车牌号 |  |  |
| cheliangmin gcheng | varchar | 200 | 车辆名称 |  |  |
| yonghuzhan ghao | varchar | 200 | 用户账号 |  |  |
| yonghuxing ming | varchar | 200 | 用户姓名 |  |  |
| chukushijian | datetime |  | 出库时间 |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| id | bigint |  | 主键 | 主键 |  |
| addtime | timestamp |  | 创建时间 |  | Now() |
| yuangonggo  nghao | varchar | 200 | 员工工号 |  |  |
| yuangongxin gming | varchar | 200 | 员工姓名 |  |  |
| mima | varchar | 200 | 密码 |  |  |
| xingbie | varchar | 200 | 性别 |  |  |
| nianling | int |  | 年龄 |  |  |
| lianxifangshi | varchar | 200 | 联系方式 |  |  |
| shenfenzhengh  ao | varchar | 200 | 身份证号 |  |  |
| zhaopian | varchar | 200 | 照片 |  |  |

表 5-2-4 员工表

数据表结构图是软件开发中用于描述系统中数据结构和它们之间的关系的重要 工具。它可以帮助企业开发人员更好地理解系统的结构和功能，并减少错误的发生。 在数据表结构图中，表格通常用矩形表示，其中包含表格的标题、行和列.行和列的 标题通常用高亮显示。开发人员可以更好地了解系统中的数据和它们之间的关系，从 而更好地设计和实现系统。总之，数据表结构图是软件开发中非常重要的一部分，它 可以帮助企业开发人员更好地理解系统的结构和功能，并减少错误的发生。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段说明 | 主键 | 默认值 |
| id | bigint |  | 主键 | 主键 |  |
| addtime | timestamp |  | 创建时间 |  | Now() |
| yonghuzhan ghao | varchar | 200 | 用户账号 |  |  |
| mima | varchar | 200 | 密码 |  |  |
| yonghuxing ming | varchar | 200 | 用户姓名 |  |  |
| touxiang | varchar | 200 | 头像 |  |  |
| xingbie | varchar | 200 | 性别 |  |  |
| nianling | int |  | 年龄 |  |  |
| shoujihaoma | varchar | 200 | 手机号码 |  |  |
| zjnum | varchar | 200 | 证件号码 |  |  |

表 5-2-5 居民用户表

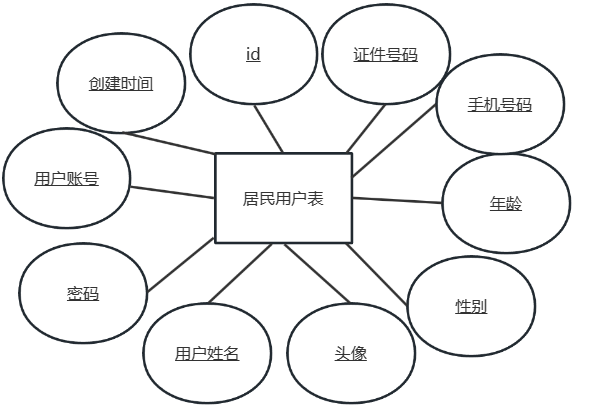


图 5-2-6 居民用户E-R图

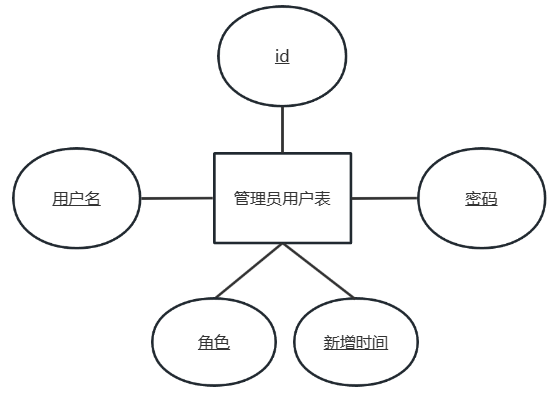


图 5-2-7 管理员用户E-R图

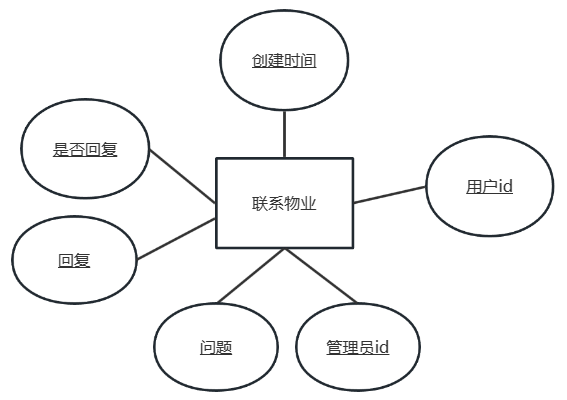


图 5-2-8 留言反馈E-R图

5.3本章小结

本章对数据库进行了详细的介绍，按照实际的数据库设计画出实体图，然后将结构通过三线表的方式展现出来。

# 第6章　系统测试

6.1 测试目的

测试目的是验证系统是否满足了开发团队所定义的需求，是否能够有效地执行预期的任务。发现系统中存在的问题和缺陷，以便开发人员能够及时修复。提高系统的质量，降低系统出现错误和故障的风险。验证软件的可靠性，包括系统的稳定性可用性、可维护性等。用户的满意度和 ‎体验，确 ‎保系统能 ‎够高效地 ‎执行用户 ‎所需的任 ‎务。发现 ‎系统的性 ‎能瓶颈， ‎以便开发 ‎人员能够 ‎优化系统 ‎的性能。 ‎系统测试 ‎通常涉及 ‎将测试对 ‎象组合为 ‎整个计算 ‎机系统的 ‎有机组件 ‎，并将其 ‎与各种硬 ‎件、数据 ‎、人员和 ‎其他相关 ‎元素相结 ‎合，以提 ‎供对计算 ‎机系统的 ‎全面测试 ‎。测试的 ‎目的是验 ‎证系统是 ‎否满足了 ‎开发团队 ‎所定义的 ‎需求，以 ‎及是否能 ‎够有效地 ‎执行预期 ‎的任务。 ‎同时，系 ‎统测试还 ‎会检查系 ‎统文档的 ‎完整性、 ‎可靠性和 ‎有效性， ‎以确保系 ‎统文档能 ‎够满足要 ‎求。需要 ‎注意的是 ‎，系统测 ‎试通常由 ‎相关测试 ‎人员独家 ‎开发，而 ‎对于软件 ‎，系统测 ‎试则是对 ‎系统中涉 ‎及的软件 ‎开发人员 ‎的全面测 ‎试。为了 ‎支持系统 ‎测试，我 ‎们需要为 ‎测试提供 ‎全面的设 ‎备支持, ‎并创建一 ‎个详细的 ‎测试计划 ‎。

6.2 测试方法

制定测试流 ‎程应该按 ‎照登录  ‎- 个人 ‎设置 - ‎ 员工设 ‎置 -  ‎发布标题 ‎公告 - ‎ 查看公 ‎告这样的 ‎基本步骤 ‎来进行测 ‎试。在测 ‎试中，可 ‎能会发现 ‎由于接口 ‎问题导致 ‎两个模块 ‎无法工作 ‎，或者无 ‎法组合相 ‎关的子功 ‎能来创建 ‎设计要求 ‎的功能。 ‎然而，这 ‎些错误通 ‎常是可接 ‎受的错误 ‎，并且不 ‎会影响整 ‎个系统的 ‎操作。验 ‎证测试旨 ‎在向未来 ‎的用户说 ‎明开发的 ‎系统可以 ‎按照要求 ‎中的描述 ‎进行操作 ‎。在软件 ‎开发完成 ‎后，应该 ‎与系统中 ‎的其他模 ‎块一起运 ‎行，并执 ‎行功能测 ‎试，如性 ‎能测试、 ‎强度测试 ‎和安全测 ‎试以及功 ‎能测试。 ‎这些测试 ‎可以验证 ‎模块的操 ‎作是否符 ‎合系统的 ‎实际要求 ‎，并是否 ‎符合既定 ‎的设计要 ‎求，并且 ‎是否能够 ‎完全满足 ‎用户的功 ‎能和性能 ‎要求。通 ‎常，正式 ‎的软件描 ‎述用作测 ‎试标准。 ‎强制测试 ‎主要测试 ‎系统的承 ‎载能力， ‎以充分反 ‎映软件中 ‎其他模块 ‎在过载情 ‎况下的运 ‎行情况。最后，安全 ‎测试主要 ‎测试数据 ‎安全性并 ‎根据需要 ‎管理用户 ‎权限。

本文设计的 ‎系统测试 ‎主要是用 ‎户对各种 ‎数据的能 ‎力和能力 ‎测试。完 ‎成上述测 ‎试后，没 ‎有发现重 ‎大错误完 ‎全可以满 ‎足要求， ‎最后测试 ‎则正式完 ‎成。

### 6.2.1 黑盒测试

黑盒测试是一种软件测试方法，它只关注测试系统的输入和输出，忽略系统的内部实现。具体来说，黑盒测试的目的是验证系统是否按照设计要求正确地处理输入数据，并输出正确的结果。黑盒测试通常被用于验证系统的功能性、可用性、安全性等方面。在黑盒测试中，测试人员使用已知的输入和预期的输出来测试系统的响应。测试人员不需要了解系统的内部实现，只需要通过输入数据来验证系统的功能是否按照设计要求运行。

### 6.2.2 白盒测试

它强调测试人员的知识面需要深入到系统内部的代码和算法，以测试系统的内部实现和合理性。白盒测试的目的是验证系统是否按照设计要求正确地实现了内部算法和代码逻辑。白盒测试通常被用于验证系统的可靠性、性能、可维护性等方面。在白盒测试中，测试人员需要深入了解系统的内部实现，包括代码、算法和数据结构等。测试人员使用这些知识来设计测试用例，并验证系统内部的代码和算法是否按照设计要求执行。

### 6.2.3 模块测试

在设计计划 ‎中，模块 ‎测试是执 ‎行的一种 ‎测试，旨 ‎在测量模 ‎块与模块 ‎之间的关 ‎系。它是 ‎一种白盒 ‎测试，是 ‎构成具有 ‎详细设计 ‎的程序的 ‎最小测试 ‎单元。模 ‎块测试通 ‎常被称为 ‎单元测试 ‎，因为测 ‎试的目的 ‎是确保每 ‎个模块作 ‎为一个单 ‎元正确地 ‎工作。在 ‎模块测试 ‎中，测试 ‎人员编写 ‎测试用例 ‎来验证代 ‎码的行为 ‎是否符合 ‎预期。通 ‎过编写单 ‎元测试， ‎测试人员 ‎可以交付 ‎代码并放 ‎心地工作 ‎，而无需 ‎担心错误 ‎。模块测 ‎试通常被 ‎称为单元 ‎测试，因 ‎为模块测 ‎试的目的 ‎是确保每 ‎个模块作 ‎为一个单 ‎元正确工 ‎作。 幸 ‎运的是， ‎单元测试 ‎将确保我 ‎们的承诺 ‎。 编写 ‎单元测试 ‎用于验证 ‎此代码的 ‎行为是否 ‎符合您的 ‎预期。单 ‎元测试使 ‎我们可以 ‎放心地交 ‎付代码而 ‎无需担心 ‎。

通常，在此 ‎测试阶段 ‎发现的通 ‎常是编码 ‎和详细设 ‎计错误。 ‎在设计良 ‎好的软件 ‎系统中， ‎每个模块 ‎执行明确 ‎定义的子 ‎功能，并 ‎且该子功 ‎能与其他 ‎模块的功 ‎能之间不 ‎存在相互 ‎依赖性。 ‎因此，每 ‎个模块都 ‎可以作为 ‎一个单独 ‎的实体进 ‎行测试， ‎并且更容 ‎易设计一 ‎个验证模 ‎块准确性 ‎的测试计 ‎划。模块 ‎测试的目 ‎的是确保 ‎每个模块 ‎作为一个 ‎单元正常 ‎工作。本 ‎系统的模 ‎块测试用 ‎例如表  ‎6-1  ‎所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 测试项目 | 内容和目的 | 测试结果 |
| 1 | 管理员登录 | 输入正确信息  输入错误信息 | 成功  未成功登录 |
| 2 | 修改管理员登录密码 | 修改新的密码 | 成功 |
| 3 | 修改操作员登录密码 | 修改新的密码 | 成功 |
| 4 | 员工的管理  （添加，修改，删除） | 输入正确信息  输入错误信息 | 成功  操作未成功 |
| 5 | 用户信息管理  （查看，修改，删除） | 填写全部信息  缺少部分重要信息 | 成功  操作未成功 |
| 6 | 通知公告管理  （添加，修改，删除） | 输入正确信息  输入错误信息 | 成功  操作未成功 |

表6-1 系统功能测试用例

### 6.2.4 部分测试用例

登录功能模块测试如表6-2所示。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | 序号 | 输入 | 预期结果 | 实际结果 | 测试结果 | | 1 | 用户名：“”，密码：“” | 登录失败 | 登录失败  输入错误信息 | 成功  未成功登录 | | 2 | 用户名：“kyk”，密码：“123456” | 登录失败 | 登录失败 | 成功 | | 3 | 用户名：“kf1”，密码：“888888” | 登录失败 | 登录失败 | 成功 | | 4 | 用户名：“kf1”，密码：“123456” | 登录成功 | 登录成功 | 成功  操作未成功 |   序号 | 输入 | 预期结果 | 实际结果 | 测试结果 |
| 1 | 用户名：“”，密码：“” | 登录失败 | 登录失败  输入错误信息 | 成功  未成功登录 |
| 2 | 用户名：“1”，密码：“123456” | 登录失败 | 登录失败 | 成功 |
| 3 | 用户名：“Admin”，密码：“admin” | 登录失败 | 登录失败 | 成功 |
| 4 | 用户名：“admin”，密码：“admin”  表6-3 客户信息功能测试用例 | 登录成功 | 登录成功 | 成功  操作未成功 |

表6-2 登录功能测试用例

投诉报修功能模块测试如表6-3所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 输入 | 预期结果 | 实际结果 | 测试结果 |
| 1 | 查询用户信息 | 展示用户信息 | 展示用户信息 | 成功 |
|  |  |  | 输入错误信息 | 未成功登录 |
| 2 | 增加用户信息 | 出现新增用户 | 出现新增用户 | 成功 |
| 3 | 修改用户信息 | 用户信息更新 | 用户信息更新 | 成功 |
| 4 | 删除用户信息 | 用户信息消失 | 用户信息消失 | 成功 |
|  | （添加，修改，删除） |  |  | 操作未成功 |
| 5 | 登记缺少必填信息 | 新增用户失败 | 新增用户失败 | 成功 |

表6-3 投诉报修功能测试用例

6.3本章小结

本章首先对测试的目的进行了解释，测试是保证系统可用性的关键一步，然后对测试的方法进行了解释，最后按照模块依次进行测试，最终通过测试，本系统可用。

# 结 论

通过开发本系统，我较全面的掌握了 SpringBoot 的基本知识和编程技巧，并在开发过程中我的 JAVA 开发能力得到了进一步的提高。如：更进一步的深入了解 MVC 设计模式，软件三层框架；SQL 语言的使用、项目部署。

在开发过程中我学到了一些经验：系统分析的好坏将决定着的系统开发成功与否，一份好分析设计将是成功开发主要因素。我们在着手开发之前不要急于编程，先应有较长的时间去把分析做好，做好数据库设计工作，写出相关的开发文档等。然后再开始编写程序代码， 这样做到每写一步代码心底有数，有条不絮。当然也有些还需待继续深入地方如：插件的使用，设计工具的使用，代码的熟练程度等。

此外，我还觉得，我个人在这次设计中走了很多弯路。主要是因为平时很少接触软件开发工作，在应用方面缺乏经验，以后还需要更多的努力。

对我来说，这次设计的本身所产生的影响，还远远没有结束，我从本次毕业设计中学到了许多课本上没有的知识。通过自己的学习和努力；通过老师的指导和教育，使我不仅仅在知识水平和解决实际问题的能力上有了很大的提高。还从思想的深处体会到，要把自己的所学变成现实时所将面对的种种难题。

最后，探讨了智慧社区管理系统在提高社区安全方面的意义。智慧社区管理系统可以通过智能沟通系统等手段，提高社区的安全保障水平，减少信息差等现象的发生，为社区居民提供更加安全、稳定的居住环境。总之，本文提出的智慧社区管理系统方案，为社区管理提供了一种全新的解决方案，具有较高的实用价值和理论意义。通过推广和应用该系统，可以提高社区管理效率和居民生活质量，为智慧城市的建设和推广提供有益的参考。

系统不免有错误和待改进之处，真诚欢迎各位师长提出宝贵意见。

# 参考文献

[1]Bruce Eckel.ON JAVA[M].北京:人民邮电出版社2022:201-252.

[2][郭雨晖,](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=%E9%83%AD%E9%9B%A8%E6%99%96&option=202)[汤志伟,](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=%E6%B1%A4%E5%BF%97%E4%BC%9F&option=202)[翟元甫](https://www.zhangqiaokeyan.com/search.html?doctypes=4_5_6_1-0_4-0_1_2_3_7_9&sertext=%E7%BF%9F%E5%85%83%E7%94%AB&option=202).政策工具视角下智慧城市政策分析:从智慧城市到新型智慧城市[J].情报杂志.2019,第006期.

[3]汤丽霞.智慧城市建设中技术应用反思[J]. 合作经济与科技，2021:37-39

[4]李兴华,马云涛.Spring Boot开发实战[M].北京:人民邮电出版社,2022:101-120.

[5] Apache Software.The SpringBoot User's Guide.http://spring.io[S]

[6] Mark Allen Weiss .数据结构与算法分析Java语言描述第三版[M].北京:机械工业出版社，2022:30-40

[7]智慧社区综合信息管理平台的设计研究[J].华东科技(综合),2021(007)

[8]社区治理升级的智慧社区模式探索[J].新丝路,2020(011)

[9]宋南. 基于MVC架构的企业宣传门户平台的开发与设计[D].吉林大学,2014.

[10]李艳杰.MySQL数据库下存储过程的设计与应用[J].信息技术与信息化,2021(01):96-97.

# 致 谢

在完成本设计的写作过程中，我十分感谢我的论文指导老师张宏老师。从选题到完成设计，老师一直都是很悉心的给我讲解着在设计中遇到的各种问题，循循善诱，严格把关，帮助我开拓设计思路，并不断地鼓舞着我，使我感到信心倍增，让我非常积极地投入到毕业课题的设计和论文撰写中，不断克服其中遇到的每个问题和困难。

在此，还要感谢哈尔滨信息工程学院所有曾给予我帮助的其他老师，没有他们在我求学过程中不断地给与我帮助，我是无法顺利完成学业的。感谢我的同学，是共同的目标和相互的激励让我在学习过程中，充分感受到了自己对学习的兴趣和热情，使我能够圆满地完成自己的毕业设计。

回想几年的求学时光，仿佛尽在昨天。最后，我再次感谢在我学习期间给我极大关心和支持的家人、各位老师以及我的同学和朋友。是你们在生活和学习上不断给与我支持、帮助和无微不至的关怀，是你们不断地给与了我信心，让我在人生中一次次坚强地走下去。