

分类号

密级

中国地质大学（北京）
专业硕士学位论文

基于 Spring Boot 框架的
社区管理系统设计与实现

研究生 包添文 学 号 2110200045

专业类别 工程硕士 专业领域 软件工程

校内导师 姚国清 校外导师 卢亿雷

学习方式 全日制

2023 年 05 月

**A Dissertation Submitted to
China University of Geosciences for Master of
Professional Degree**

**Design and Implementation of
Community Management System
Based on the Spring Boot framework**

Master Candidate: BaoTianwen

Professional Degree: Master of Professional Degree

Dissertation Supervisor: Prof.YaoGuoqing

Associate Supervisor: LuYiLei

China University of Geosciences (Beijing)

摘要

随着我国成功全面脱贫，广大次发达地区的人民生活也向着数字化迈进。人民的生活以社区为单位，部分发达地区社区已经完成了数字化社区建设。但是在疫情期间，次发达地区人民所在社区暴露出了管理效率低下，管理混乱等问题。因此，该课题目标是开发一个体量小，成本低，容易适配的社区管理系统。可以为次发达地区人民提供社区服务上的便利，由此看来该课题的研究具备一定的社会效益和经济效益。

本研究旨在探讨当前国内外智能社区建设的发展趋势，为此，我们采用 B/S 架构，并结合前后端分离的开发技术，运用 Spring Boot、Vue.js、MyBatis、MySQL 以及 Tomcat 等关键技术，实现了多种功能模块，包括居民管理、房屋管理、车位管理、缴费、社区服务以及社区居民操作系统。呈现出一个功能齐全、交互性良好的社区管理系统。论文首先从功能性和非功能性需求两方面对社区管理系统进行了分析，确定了目标用户群体及其用例；其次，对系统的体系架构、功能结构进行了总体设计，以及功能模块和数据库的详细设计，并列出了主要的数据库表；然后，在设计的基础上对系统的各个功能模块进行了实现，并展示了部分功能的实现代码和实现后的效果图；最终，对系统进行了黑盒测试，确保系统在满足用户需求的同时能够稳定流畅地运行。

经过精心开发，我们的系统能够满足主要用户的需求，为社区工作者提供有效的社区管理服务。对系统进行了分析和设计，达到了本课题的目的。

关键词：数字化社区，管理系统，社区管理

Abstract

With the successful and comprehensive poverty alleviation in China, people's lives in the vast underdeveloped areas are also moving towards digitization. People's lives are based on communities, and some developed communities have completed the construction of digital communities. However, during the epidemic, communities where people live in underdeveloped areas have exposed issues such as low management efficiency and chaotic management. Therefore, the goal of this project is to develop a small, low-cost, and easily adaptable community management system. It can provide convenience in community services for people in underdeveloped areas, indicating that the research on this topic has certain social and economic benefits.

This thesis introduces the current situation and development trends of the construction of intelligent community at home and abroad. The system adopts B/S architecture and uses front-end and back-end separation development techniques, and utilizes key technologies such as Spring Boot, Vue.js, MyBatis, MySQL, and Tomcat to achieve main functional modules such as resident management system, housing management system, parking management system, payment system, community service system, and community resident operating system. Present a fully functional and interactive community management system. Firstly, the thesis analyzes the community management system from two aspects of functional and non-functional requirement, and determines the target user groups and their use cases; Secondly, the overall structure and functional structure of the system are designed, and the functional modules and databases are designed in detail., and the main database tables were listed; Then, on the basis of the design, each functional module of the system were implemented, and some functional realization codes and effects are displayed; Finally, a black box test was conducted on the system to ensure that it can run stably and smoothly while meeting user needs.

The system currently being developed can meet the needs of key users and assist community workers complete community management. The analysis and design of the system reach the purpose of this study.

Key words: Digital community, management system, community management

目 录

1 绪论.....	1
1.1 课题背景及研究意义.....	1
1.2 国内外研究现状.....	3
1.2.1 国外研究现状.....	3
1.2.2 国内研究现状.....	4
1.3 本文主要研究内容.....	6
1.4 论文的组织结构.....	7
2 系统相关开发技术介绍	8
2.1 Spring Boot 框架技术	8
2.1.1 Spring Boot 框架的发展	8
2.1.2 Spring Boot 框架特点	9
2.1.3 Spring Boot 运行原理	10
2.2 Vue.js 框架技术.....	12
2.3 数据库技术介绍.....	13
2.4 Mybatis 框架技术	15
2.5 本章小结.....	17
3 系统需求分析	18
3.1 系统功能需求概述.....	18
3.2 系统用户及用例分析.....	19
3.3 系统功能需求分析.....	20
3.4 系统非功能需求分析.....	21
3.5 本章小结.....	22
4 社区管理系统的设计	23
4.1 系统总体架构设计.....	23
4.2 系统功能设计.....	24
4.2.1 居民管理系统设计	24
4.2.2 房屋管理系统设计	26
4.2.3 车位管理系统设计	28
4.2.4 缴费管理系统设计	30
4.2.5 社区服务管理系统设计	31
4.2.6 社区居民操作系统设计	33
4.3 数据库设计.....	34
4.4 本章小结.....	39
5 社区管理系统的实现与测试	40

5.1 系统开发环境.....	40
5.2 系统功能实现.....	40
5.2.1 居民管理系统实现.....	40
5.2.2 房屋管理系统实现.....	43
5.2.3 车位管理系统实现.....	45
5.2.4 缴费管理系统实现.....	47
5.2.5 社区服务管理系统实现.....	49
5.2.6 社区居民操作系统实现.....	50
5.3 系统测试.....	50
5.4 本章小结.....	54
6 总结与展望.....	55
参考文献.....	56

1 绪论

城市社区化能够有效解决城市的经济发展建设中出现的一部分社会问题，作为加快中国城市化建设的重要工具，作为提高人民幸福指数的迫切需要，我国的数字化社区发展历经近 20 年的光阴，现在已经获得了令人瞩目的发展和成绩。现阶段我国社区建设主攻方向就是建设数字化、智能化的社区。本论文的论题就是设计一个数字化社区管理系统，为社区建设普遍达到数字化，智能化等目的予以助力。

1.1 课题背景及研究意义

一方面，城市的建设规模和水平日新月异，人们的生活也被一个个社区所划分，社区的地位和重要性在人们心中日趋增加，社区为居民提供的服务也逐渐成为了人们生活中必不可少的一部分，人们对于社区服务的关注也与日俱增。社区管理的工作内容包括但不限于稳定社区秩序，满足社区居民生活中的基础需求等。

另一方面，虽然新冠肺炎疫情已经告一段落，但是新冠肺炎病毒疫情是新中国成立以来最难预防和控制的重大公共卫生事件之一，对我国和全球公共卫生安全提出了严峻的挑战。社区做为疫情联防联控和群防群治工作的前沿，是外防输入、内防扩散最为有效的首道防线^[1]。疫情爆发后，国家对社区进行严格网格化管理，切实减少人员互相接触，而社区的人员管理，材料购买，居民身体健康情况收集等都是社区管理者面临的重大难题，微信群交流复杂，人员接触不方便为社区工作人员的工作造成了很大的麻烦。因此，智能化，数字化社区管理系统建设以数据化“互联网+”思维和科学化工作运行机制，促进社区治理服务信息化，破解居民档案收集，日常健康打卡，日常物资统计难题，使疫情防控及社区管理工作朝着整体化，系统性，科学化，规范化，标准化的方向发展、以有序化为导向，促进街道社区管理信息规范、统一、协同、共享，利用网络、数字等技术帮助打造智慧社区平台具有较好的实用性^[2]。

过去的社区建设主要关注基础设施建设，轻视社区的服务功能，在 20 年的时间里社区建设重心逐渐转移到增强服务功能，推进社区服务网络化、产业化上来。我国多数城市中社区前身多为本地各居委会及其他机构，社区服务中心也多是居委会牵头、社区内居民响应居委会的号召，由居委会和物业共同协作，为社区内居民提供基础的生活保障，便民服务如生活配套、医疗服务、治安保障等等服务。

目前我国多数社区服务中心为居民能够提供的服务涉及行业广泛，覆盖内容丰富。同时，随着近年来物联网技术和社区数字化管理等技术的发展和日趋融合，数字化社区已经

从一个最初的设想逐渐被人们所实现。数字化社区成功地在社区管理层面上基本达成了信息化的目的。数字化社区在逐步提高信息收集和管理能力以及丰富各种数字化办公手段，为了能够不断地提升社区管理系统的数字化管理水平与服务水平^[3]。

建设信息化、数字化的社区需要对应数字化平台作为基础，在这个平台上，数字化社区才能够以更加高效的业务处理能力应对更多更复杂的社区业务。在当今的后疫情时代，整个房地产市场对数字化、智能化的社区管理系统的需求量很大。但是因为不同的社区之间的管理模式、社区硬件设施、地理位置、社区服务内容等等均存在较大的差异性，所以不同的社区对于数字化社区管理系统的需求也不尽相同。这些种种因素导致了，很难有一个企业能设计一个适用于所有类型小区的系统。除此之外，在商业层面上，数字化社区行业并没有出现一个能够拥有一定的行业水准，能够作为行业风向标的公司。所以目前数字化社区的发展仍然采用多方共同参与的方式，成本高，效率低下。

所以开发一个适用性强，泛用性搞的数字化社区管理系统具有一定研究意义。本论题的研究是为了实现社区管理服务的数字化建设，为了更加清晰的分配社区公共资源同时满足全体社区居民的生活和娱乐需求。为了我国数字化社区的发展做出贡献。

本论题研究意义有以下几点：

（1）伴随着我国人民的生活质量的不断提高，在计算机技术和软件工程科学不断发展的今天，我国人民对生活品质的要求也在逐步升高。而社区作为人民管理的基本单位，首先承担了居民大部分的生活、工作、安全等方面的需求。正因如此，我国的社区管理模式正在发生全方面的转变。社区管理工作正在逐步由人工方式转为半自动方式，由手工式办公转为数字化办公，信息管理也从个体管理逐步转为社区群体式管理。这样才能够提升社区工作的办事效率，提高社区服务的整体质量，满足社区居民的各方面需求的同时提高国民的整体幸福度。

（2）数字化的社区管理系统可以保证社区工作人员的办公高效性，社区工作人员的大部分工作可以在电脑上完成，社区工作人员只需在办公室就能对居民所需要的社区服务进行操作，对居民的信息进行检索，对社区管理系统的功能进行调用。这样的数字化社区管理系统能够满足社区工作人员和社区居民的大多数需求。

（3）社区管理系统的数字化工作方式能够最大程度的帮助社区管理人员避免工作量大、工作内容重复、工作效率低下的大量收集性、审核性劳动。在减少了人工成本的同时，还可以最大程度的降低由人工行为导致的发生工作失误甚至是工作事故的可能性。保证了

在能够满足大部分社区居民的各项基础需求的同时，还能够不断提高社区工作人员的工作效率和社区服务的整体质量。

1.2 国内外研究现状

国内外对数字化社区的研究都在逐渐增多。国内研究主要关注数字化社区的社会影响、用户行为、平台治理、商业模式等方面。国外研究则更注重数字化社区的跨文化交流、全球化影响、技术创新等方面。总体来说，数字化社区的研究范围广泛，包括社会、文化、政治、经济等多个领域。在未来，数字化社区的研究将继续深入，以更好地理解数字化社区在人类发展进程中的作用和意义。

1.2.1 国外研究现状

社区管理的数字化研究目前来说国外较为先进，西方国家出现社区信息化管理思想最早可以追溯至 20 世纪 60 年代。在西方国家中，美国、英国、德国的数字化社区发展起步最早，普及率也相对较高。在亚洲国家中，建设数字化社区的成功代表则是新加坡和日本。这些国家在多年的研究和发展后，在社区管理方面取得了很大的成果，建立了相对成体系的社区管理理论。

美国的数字社区的理念最早由美国副总统艾伯特·戈尔提出，艾伯特·戈尔提出的数字地球理论，该理论期望收集整个地球的所有的信息，借助数字化手段，达到促进经济快速发展和人类文明进步的目的。数字地球这一理念希望能够通过全方位、全区域的收集对应社区的信息并进行数字化整合，借助计算机和物联网等技术提高社区服务的整体质量。此后艾伯特·戈尔副总统又相继提出了建设数字化城市，构建数字化舒适社区的建议。当前，美国多数城市已经进行了多年的数字化社区的建设，已经完成数字化建设的示范社区也覆盖了美国大大小小数十个城市。这些建设完毕的示范社区不仅包括服务于大部分民众的综合型服务类社区，还包括一些具有政治、经济、安全等方面针对性设计的专业社区。比如由华盛顿管培中心设计和构建的华盛顿特区社区服务系统就是一个具有政治、安全等方面针对性设计的专业型社区。社区内涵盖了多种社区服务，包括但不限于升学、就业指导、居民培训等等专业型服务。而与华盛顿特区不同，纽约社区服务系统则是一个综合性社区服务体系，能够为社区居民提供十多种综合服务内容^[4]。

德国在 20 世纪 90 年代就成功建造了智能化楼宇系统，该系统通过结合软件工程科学和物联网技术，成功实现了数字化房屋系统的设计，该系统能够实时监控房屋内所有设备的实时信息，并加以控制。大大地提高了管理房屋的效率，保证了房屋内的各项事务的管

理质量。

在亚洲，新加坡在 20 世纪 90 年代开始建设数字化社区管理系统。旨在通过信息技术手段提高国民生活质量，希望将新加坡建设成高度信息化的城市。随后在新加坡上线的新加坡综合网便做到可以在线服务于新加坡的所有家庭，学校和公司。新加坡综合网历经多年建设与发展，现能提供的各项服务数量过百，这些服务涵盖子女教育、升学、卫生保洁、娱乐服务、购物帮助、甚至是商业投资建议、政坛时讯等领域，可以满足大部分新加坡市民的生活和工作所需^[5]。当这一体系和新加坡非常完善的线上线下服务中心结合时，新加坡已成为亚洲智能社区的建设范例。

目前，自治模式、行政模式和混合模式是国外社区发展中主要存在的三种模式。在欧美地区国家主要采用的是自治模式，这些地区的政府不会参加到社区的管理工作中，但是会通过制定相关的法律法规来规范社区的管理发展，相互支持与监督，保证社区自制管理的有序发展。而行政模式管理则是由当地政府对社区的发展进行管控，当地政府直接干预社区管理。最后一种是混合模式，最具有代表性的是日本，当地政府对社区采取发展较轻的管理，对社区给予物质等支持^[6]。

综上所述，国外的社区管理系统从理论体系的建立、优化、直至落实到设计与实现层面上都已经相对比较完备。国外的社区管理理论体系比起国内发展的更早，普及范围更广。目前来说其主要优势包括：有成熟的开发技术和开发环境，有较大的市场环境和政策支持。大多数的社区管理系统已经在西方落地，并且随着软件工程科学和物联网技术的发展，欧美等发达国家在这一方面的领先优势还有扩大的可能性。

1.2.2 国内研究现状

在我国，数字化社区管理系统的研究与建设还处于初级阶段，我国既需要借鉴于其他国家的先进技术，也会在对先进技术进行参考的前提下进行适应于我国国情的改进。我国智能社区管理系统在 20 世纪 90 年代开始建设，并且国内的社区信息管理系统在迅速涌现的同时，快速实现了和社区实际业务的结合。智能化社区管理系统的建设也随着物联网技术的发展得到了快速的发展，大数据等技术的发展又大大提高了系统的科学管理水平。目前来说我国的智能社区管理系统正在飞速发展阶段^[7]。

虽然我国近年来智能社区的发展速度迅速。但是，智能社区建设水平参差不齐。底层社区管理功能多为以传统通信方式为主的社区服务网络。这种情况也是我国最初构想的建设数字化社区的方式，这种社区系统采用设立社区服务热线在社区设立社区热线的方式，

居民可以通过电话或手机向社区服务中心咨询或求助。社区居民间，社区居民和社区管理者，社区服务者间的连接是以传统的通信方式进行的^[8]。中层社区管理系统是一个以互联网为基础的社区服务网络。社区服务已由传统的通信方式向万维网转移，居民通过网络连接社区服务者及管理者，是目前国内智能社区建设分布最广，使用最频繁的一种模式。最高一级的社区管理系统以物联网为基础构建社区服务网络，将社区内的资源，社区服务的相关方全部融入社区管理系统，服务内容丰富方便，极大地提升服务质量和管理效率^[9]。

总体来说，我国的智能社区管理系统还处于研究和发展的初级阶段^[10]，相对于发达地区还有很多学习和改进的地方：

第一，社区管理系统的适用对象应该具备相应的资质和条件，必须是经过一定法定程序认证的物业公司。但是，国内很多物业公司缺乏规范化的管理，忽视社区的管理建设，业务水平相对有限。这些公司只是单纯地收费提供服务，缺乏能力带动社区发展，无法提高社区品质。

第二，社区管理系统的运行模式不够清晰，会出现资金投入在先，业主却得不到相应的物业服务的情况。社区的管理经常发生避重就轻等不作为现象。调控运作方面缺少宏观思想。

第三，社区管理系统的时效性难以保证，很多社区内设施信息和居民的信息无法做到及时的更新，如果发生特殊事件，社区服务机构大都无法在第一时间发现并解决。存在安全隐患，影响社区发展。

最后，社区管理功能较为单一，社区服务也仅限于保洁、维修等一般性工作。很难从社区管理系统的角度提高社区居民的满意度。除此之外，从业人员不重视服务质量，在低效率、低服务质量的传统社区工作方式下，无法满足社区居民日益增长的物质文化需求。

随着互联网技术和软件工程技术的迅速发展，随着我国社区管理模式的研究进程的推进，之前落后的社区管理模式已经与我国现今日益提高的国民生活水平不符。我国的社区管理模式的转变已经发生，现在正处于传统社区管理模式向数字化社区管理模式转化的阶段，在当今数字化社区发展飞速的时期，许多社区的硬件水平都已经达到相对来说比较完善的条件^[11]。为了提高社区工作人员办公效率、降低人工成本、快速收集整理社区资源等功能，许多数字化社区引入了物联网技术，建立了网格化管理结构，建立了个社区数据库，社区网络安全中心等设施，大大提高了社区居民的生活质量，有效提高了社区工作人员的工作效率和服务能力，有效的推进了我国建设数字化社区的研究进程。

社区管理系统的设计和实现属于软件工程科学，在我国已经有了多年的研究和沉淀。我国绝大部分社区内也都已经完成了网络铺设，所以在这些基础上，本课题所研究的社区智能管理思想完全可以依托软件工程科学理论的发展建设一个信息数据互通的管理系统，并且满足绝大部分需求，尤其利用 B/S 架构软件工程管理技术，将进一步推动数字化社区的研究进程，进一步在我国普及数字化社区的思想。

1.3 本文主要研究内容

本论题在我国社区管理研究和发展的基础上，分析数字化社区的建设和普及工作中所遇到的问题，并对国内大部分社区管理系统研究成果进行分析，在结合全球社区管理系统的发展过程和未来发展方向，结合我国国情，利用软件工程科学知识和 SpringBoot 框架工具，设计并实现了一款适用于我国国情的，功能相对完备的社区管理系统。并完成了整部论文的撰写工作。

(1) 本论题采用 Spring Boot 框架，以 B/S 的结构模式，开发了一个服务器控制台系统。社区管理人员通过该系统对系统所应用的社区进行控制和管理，并通过调用系统内各个模块的功能实现对社区的信息更新、业务办理、居民管理等操作。

(2) 本论题主要工作是对社区管理系统的设计和实现。在设计时，本论题严格遵循 Java 面向对象模式。根据我国目前社区管理系统的发展现状，对社区管理系统进行了全面的需求分析，使用系统部署图与顺序图表达系统架构，并对系统静态结构进行详细设计使用关系类图。实现部分采用前后端分离的开发方式，对系统使用 Spring Boot 框架和 Vue.js 框架技术。使用 MySQL 数据库、Tomcat 技术和 MyBatis 技术。在系统使用界面，用户只需在浏览器上进行可视化操作即可，同时也可在窗口内进行多种操作，最大限度地减少了系统功能界面，合理得对相似操作功能加以概括与融合，降低了系统的学习成本和使用门槛，以便于更好的使用 and 普及。

(3) 本论题在设计社区管理系统各个基本功能的过程中，也针对性的设计了如查询、筛选、排序、打印等辅助功能。目的是，本论题旨在设计一个功能较为完备的社区管理系统，在主要功能实现的同时，也需要有符合用户操作习惯和便于进行数据库管理的其他辅助性功能。本论题主要功能模块包括：居民管理模块、房屋管理模块、车位管理模块、社区服务模块等等。各个主要功能模块内部又包含着很多的子功能和辅助功能，保证功能全面性的同时也能方便用户的使用。

1.4 论文的组织结构

论文总共分为六章，各章结构及内容概述如下：

第1章 绪论。本章首先阐述了社区管理系统的开发背景，并根据目前国内外关于社区管理系统的基本研究状况。对社区管理系统的基本思想及研究意义进行介绍，并对本课题的论文结构进行概述。

第2章 系统开发相关技术介绍。在这一章中，对系统开发过程中使用的开发工具进行了介绍，并且详细描述了该系统所应用的开发框架、相关技术和数据库系统等开发工具。并对这些因素的原理和使用方法进行了概述。最终，确定了本次系统开发所需要的开发工具及相关技术支持等。

第3章 系统需求分析。本章根据我国目前社区管理系统的发展现状，对社区管理系统进行了全面的需求分析。首先进行了功能性分析，对社区管理系统的各项功能进行了详细的阐述。其中包括居民信息管理、楼栋管理、车位管理、社区服务等多个方面。其次，本章详细分析了系统用例，通过对系统用例的描述，进一步明确了系统功能的具体实现方式和操作流程。最后进行了如性能、数据安全、可靠性等方面的非功能性需求分析。

第4章 系统设计。在本章中，在进行设计之前，明确了设计的原则。之后基于系统需求分析，对系统的架构、功能和数据库等进行了设计。

第5章 系统实现与测试。在系统设计的基础上，对系统的功能进行了实现。在实现过程中，采用了逐步迭代的方法，不断进行调试和优化，以确保系统的稳定性和可靠性。在功能实现完成后，对系统进行了黑盒测试，以验证系统的正确性和完整性。黑盒测试主要针对系统的功能进行测试，通过模拟用户的操作，测试系统是否能够正确地响应用户的请求，并且能够正确地处理各种异常情况。通过测试，可以发现系统中存在的问题，并及时进行修复和优化，以提高系统的质量和性能。

第6章 总结和展望。本章总结了系统开发的过程和成果，并对系统的优缺点进行了评价和总结。同时，也对系统的进一步优化和改进提出了展望和建议，以期能够更好地满足用户的需求和提高系统的效率和可靠性。

2 系统相关开发技术介绍

社区管理系统采用的是 B/S 结构下的前后端分离的开发模式，业务逻辑主要发生在服务端，而为了便于对系统进行升级维护和版本更新等操作，客户端载体为 web 浏览器。此外，基于广域网建立的 B/S 结构对网络硬件环境的要求相对较低，因此 B/S 结构适用范围更广泛。社区管理系统使用的前后端分离开发模式好处有两点，第一是双端同步进行的开发模式能够避免双端高耦合问题。第二是，前端能够过滤和预处理用户的输入，服务端的压力能够得到很大减轻。

2.1 Spring Boot 框架技术

Spring Boot 是一个基于 Spring 框架的开源应用程序开发框架，它通过提供自动配置和约定大于配置的方式，简化了 Spring 应用程序的开发过程。Spring Boot 的特点包括快速构建、自动配置、无代码生成、嵌入式 Web 服务器等，可以帮助开发人员快速构建高效、可扩展的应用程序和微服务^[13]。Spring Boot 的运行原理是通过 Spring 框架的依赖注入和控制反转机制，大大降低了程序的耦合性，为后续对程序进行维护和扩展留出技术空间。

2.1.1 Spring Boot 框架的发展

Javaweb 应用的雏形是 Servlet，在随着框架技术水平不断更新和提高时，经过数次的更新，最终形成了相对完备的开源组件。在这一套生态系统相对完整的开源组件中，Spring Boot 框架凭借着其丰富的功能性、便捷的开发性等特点，逐步获得了绝大部分开发人员的欢迎^[14]。

Spring 由若干个组件组成，比如 IOC、AOP；还包括多种框架如常用的 Spring MVC 框架、持久化框架；也包括用于单元测试的工具如 Junit 等^[15]。其中部分功能由于能够实现系统设计的松散耦合，所以能够为系统的维护和升级提供帮助，能够减少测试工作的工作量比如控制反转功能。也正因为这些种种优点，大部分开发人员都会在选择使用功能性和稳定性兼备的 Spring 框架。但是 Spring 框架也有一定的缺点，比如在开发过程中随着开发内容的深入和扩展，开发人员需要配置大量的 XML 文件和注解，这些大量繁琐的任务使得开发人员在 Spring 框架上浪费了过多的精力和时间。所以，随着这个问题的带来的副作用越来越大，开发人员迫切需要一个更轻量级更便捷的框架工具使他们从繁琐枯燥的配置工作中解脱出来。这就也促使了 Spring Boot 技术的诞生。在 2012 年 10 月，Mike Youngstrom 首次提出了在 Spring 框架中支持无容器 Web 应用程序体系结构的想法。

他认为，为了简化 Spring 框架的 Web 应用体系结构，可以通过 Spring 组件和配置工具内嵌一些常用的 Web 配置文件，实现通过 main() 方法引导 Spring 容器的目标。这一想法促成了 Spring Boot 的诞生。2014 年，Spring Boot 1.0 首次推出，此后 Spring Boot 的各种版本伴随着更加齐全的功能如雨后春笋般涌现。2018 年，Spring Boot 2.0 上线，2.0 版本能够同时支持 Java 9.0 兼容 Quartz 和嵌入式 Netty 等。Spring Boot 2.0 的推出，进一步提高了 Spring Boot 框架的性能和可扩展性。它不仅提供了内嵌式的 Web 容器服务，还支持多种数据源和多种安全认证方式。此外，Spring Boot 还支持各种开发工具和插件，如 Spring Tool Suite 和 Spring Initializr，使得开发人员可以更加轻松地进行开发和部署。因此，Spring Boot 已经成为了目前最受欢迎的 Java Web 开发框架之一。

2.1.2 Spring Boot 框架特点

目前大多数企业的后台搭建技术框架大都采用了 Spring Boot 框架，正如上文中所提到的，Spring Boot 框架能够使开发人员摆脱大多数开发过程中涉及到的繁琐枯燥的 XML 和注解的配置工作。这样能够大幅度的提高项目的开发效率。同时，由于 Spring Boot 的 starter 特性能够禁止重复依赖，所以开发过程无需担心重复依赖导致的版本冲突。总而言之，Spring Boot 的特点就是能够大幅度减轻开发人员的工作量，使得开发重点能够集中在核心业务的编写上去^[16]。以下将简单介绍 Spring Boot 的几个具体特性：

自动配置：Spring Boot 能够根据应用程序的依赖关系自动配置应用程序，大大减少了开发人员的工作量。

起步依赖：Spring Boot 提供了一系列的“起步依赖”，这些依赖包含了常用的库和框架，**嵌入式 Web 服务器：**Spring Boot 内置了 Tomcat、Jetty 和 Undertow 等 Web 服务器，开发人员可以选择其中的任意一种作为应用程序的 Web 服务器。

Actuator：Spring Boot 提供了 Actuator，可以实现对应用程序的监控和管理，包括监控应用程序的健康状况、性能指标、日志、配置信息等^[17]。

简化配置：Spring Boot 采用约定优于配置的方式，通过提供默认配置和自动配置，大大简化了应用程序的配置过程，同时也方便了应用程序的维护和升级。

易于集成：Spring Boot 提供了丰富的集成能力，可以轻松地集成其他 Spring 项目、第三方库、数据库、消息队列等。

生产就绪：Spring Boot 提供了多种生产环境下的部署方式，包括打包成可执行的 JAR/WAR 文件、Docker 容器化部署等，同时也提供了多种监控和管理工具，如 Spring Boot

Admin、Micrometer 等，方便开发人员对应用程序进行监控和管理。

无需代码生成：Spring Boot 采用注解和约定的方式，可以大大减少代码的编写量，同时也避免了代码生成带来的问题，如代码重复、维护难度等。

多环境支持：Spring Boot 提供了多种环境配置文件，如 `application.properties`、`application.yml` 等，可以根据不同的环境（如开发、测试、生产）配置不同的参数。

总之，Spring Boot 是一个快速、简单、灵活、生产就绪的框架，可以帮助开发人员快速搭建一个高质量、可维护、易扩展的应用程序。

Spring Boot 和 Spring 的关系在上文已经进行了描述，他们之间的关系可用如下架构图描述，Spring Boot 架构图^[18]如图 2-1 所示。

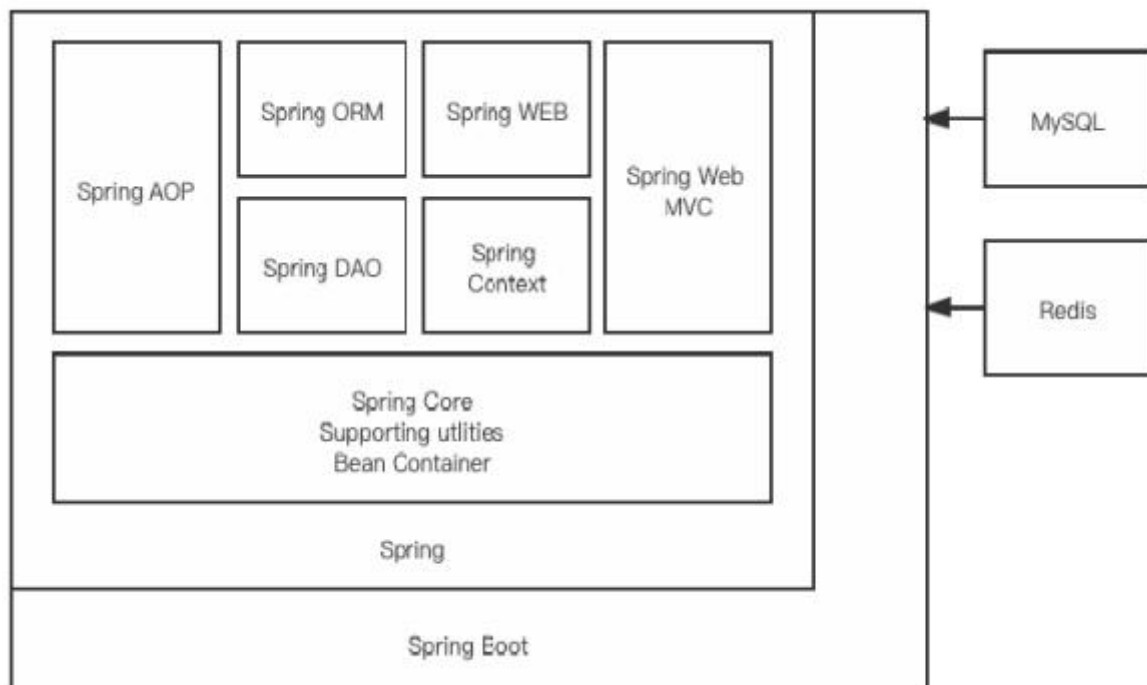


图 2-1 Spring Boot 架构图

2.1.3 Spring Boot 运行原理

上文中介绍了 Spring Boot 的特点，本节重点介绍 Spring Boot 的初始化流程和其基本运行原理，以便于读者更容易理解 Spring Boot 的特点。Spring Boot 的初始化流程如下：

1) 加载 Spring Boot 核心配置文件：Spring Boot 会加载默认的配置文件 `application.properties` 或 `application.yml`，其中包含了一些默认的配置项，例如应用程序的端口号、数据库连接等。

2) 加载 Spring Boot 的 Auto-Configuration: Spring Boot 会根据项目中所依赖的 jar 包，自动配置一些常用的 Bean，例如 `DataSource`、`JdbcTemplate` 等。

3) 加载用户定义的 Bean: Spring Boot 会扫描项目中的 `@Component`、`@Service`、`@Repository` 等注解, 将这些类注册为 Bean。

4) 执行 Spring Boot 的 `CommandLineRunner`: 如果项目中有实现 `CommandLineRunner` 接口的类, Spring Boot 会在初始化完成后执行这些类中的 `run` 方法。

5) 启动 Spring Boot 应用程序: Spring Boot 会创建一个内部 Tomcat 服务器, 并将所有的 Bean 注册到 Spring 容器中, 然后启动 Tomcat 服务器, 使得应用程序可以响应客户端的请求。

在整个初始化流程中, Spring Boot 的核心配置文件和 Auto-Configuration 起到了关键作用, 它们可以帮助开发人员快速地搭建一个基于 Spring 的 Web 应用程序, 同时也可以根据需要进行自定义配置。

SpringBoot 是一个开源的 Java 框架, 它可以帮助开发人员快速构建基于 Spring 框架的应用程序。SpringBoot 的运行原理可以简单概括为以下几个步骤:

1) 应用程序启动: SpringBoot 应用程序的运行是通过 `main` 方法启动的, 这个 `main` 方法会启动一个内嵌的 Tomcat 服务器。

2) 自动配置: SpringBoot 框架会根据应用程序的类路径、注解和配置文件等信息, 自动配置各种功能模块, 包括数据源、Web MVC、安全等。

3) Spring 框架集成: SpringBoot 框架会自动集成 Spring 框架的各种功能模块, 包括 IoC 容器、AOP、事务管理等, 开发人员可以直接使用 Spring 框架的相关功能^[19]。

4) 运行应用程序: SpringBoot 应用程序启动后, 内嵌的 Tomcat 服务器会启动, 同时应用程序会加载自动配置的各种功能模块, 然后等待客户端请求。

5) 处理请求: 当客户端发送请求时, Tomcat 服务器会将请求转发给应用程序, 应用程序会根据请求的 URL 和方法等信息, 调用相应的控制器方法进行处理, 并返回响应结果。

6) 关闭应用程序: 当应用程序不再需要运行时, 可以通过调用 SpringBoot 提供的关闭方法来关闭应用程序, 这个方法会关闭内嵌的 Tomcat 服务器, 并释放资源, 结束应用程序的运行。

总的来说, SpringBoot 的运行原理可以概括为自动配置、集成 Spring 框架、内嵌 Tomcat 服务器、处理请求和关闭应用程序等几个步骤^[20]。这些步骤使得开发人员可以快速构建基于 Spring 框架的应用程序, 并且可以更加专注于业务逻辑的实现, 而不需要过多关注底层

的技术细节。

2.2 Vue.js 框架技术

Vue 是一种先进的渐进式前端框架，它对基于原生 JavaScript、HTML、CSS 的前端开发模式进行了封装^[21]。Vue 的核心库只关注视图层，使得它易于上手，同时也方便与第三方库或既有项目整合。

Vue.js 是一种对 MVVM 架构的实践，本质上是一个 JavaScript MVVM 库，用于构建用户界面的渐进式框架^[22]。Vue.js 专注于 MVVM 中的 ViewModel，实现了数据双向绑定特性。采用 MVVM 响应式编程模型，避免直接操作 DOM，从而降低了 DOM 操作的复杂性。MVVM 实现了页面输入改变数据，数据改变影响页面数据展示与渲染，简化了前端开发的流程与业务处理^[23]。

Vue 框架的优点在于它的轻量级、易学易用、高效灵活、可扩展性强等特点，使得它成为了前端开发的首选框架之一。Vue 框架的组件化开发模式，使得前端开发者可以将复杂的页面拆分成多个组件，各个组件之间相互独立，易于维护和扩展。同时，Vue 框架还提供了丰富的插件和工具，可以帮助开发者更快速、更高效地完成前端开发工作。

总之，Vue 框架的出现，极大地提高了前端开发的效率和质量，使得前端开发者可以更加专注于业务逻辑的实现，而无需过多关注底层的实现细节^[24]。同时，Vue 框架还具有很好的可维护性和可扩展性，可以帮助开发者更好地管理和维护前端代码。因此，Vue 框架已经成为了前端开发的主流框架之一，受到了广泛的关注和应用。

具体的 MVVM 开发框架图如下图 2-2 所示。

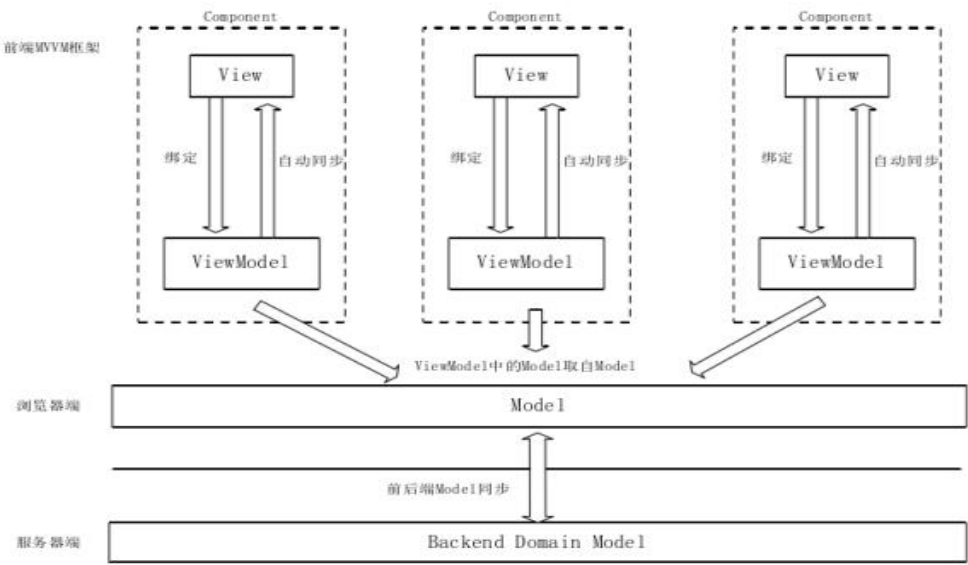


图 2-2 MVVM 框架图

与其他前端框架相比，Vue.js 具备以下优点：

（1）易用性

Vue.js 提供了诸多风格与 HTML 类似的语法，能够使得前端开发人员快速上手，其具有的数据双向绑定特性也极大改善了前端业务代码的开发体验。传统前端开发中，开发人员通常需要基于 Ajax 对服务端进行异步请求，在得到响应数据后，再通过选择器与 JavaScript 代码对相应页面组件的值进行修改，即页面组件中的值与 Ajax 发送请求后得到的响应值并不直接相关^[25]。而在 Vue.js 的数据双向绑定特性下，开发人员可以像使用变量一样为页面组件填充数据，一旦数据变量的值被修改则页面组件中的值被同步修改，反之亦然，极大减少了先前对二者数据进行同步修改所需编写的代码，同时使得业务代码变得简洁且模块化。

（2）灵活性

Vue.js 提供了丰富的条件控制语法，如 v-if 与 v-for 便可实现条件判断与循环逻辑，避免了开发人员使用 JavaScript 直接操作 DOM 的传统开发方式。同时，Vue.js 还提供了属性绑定、事件触发、路由跳转等功能，使得前端项目开发具有了部分传统后端开发才具有的特性，极大提高了前端项目开发的灵活性。

（3）高效性

压缩版 Vue.js 核心库仅有 33KB 大小，且 Vue.js 具有良好的执行效率。Vue.js 具备虚拟 DOM 特性，先采用 JavaScript 进行 DOM 预处理计算，后将最终的 DOM 操作推演出来并进行优化，由于其中采用了 DOM 预处理操作，仅是一种推演计算，因而具有较高性能，也因此称为虚拟 DOM。

2.3 数据库技术介绍

数据库技术的起源可以追溯到 20 世纪 60 年代，当时计算机存储和处理数据的能力开始快速发展。在此期间，IBM 公司的研究人员开发了第一个商业数据库管理系统（DBMS），名为 IMS。随着计算机技术的不断发展，数据库技术也得到了不断的完善和发展。在 20 世纪 70 年代，关系型数据库管理系统（RDBMS）开始出现，这是一种基于关系模型的数据库系统，它使用 SQL（结构化查询语言）进行数据管理和查询。在 80 年代和 90 年代，随着互联网的兴起，数据库技术得到了广泛应用和发展。此时，数据库技术开始涉及到更多的领域，如数据仓库、数据挖掘、大数据等^[26]。同时，非关系型数据库（NoSQL）也开始出现，提供了一种不同于传统关系型数据库的数据存储和管理方式，适用于不同的数据

类型和应用场景。现在，数据库技术已经成为了现代计算机系统中不可或缺的一部分，支持着各种应用程序和业务流程。随着云计算和人工智能等新技术的发展，数据库技术也在不断地创新和发展，为更广泛的应用场景提供更好的数据管理和分析解发展已经经历了多个阶段，从最初的关系型数据库到现在的多种数据库技术^[27]，每一次的发展都是为了更好地满足不同应用场景下的数据管理和分析需求。数据库技术的未来发展方向包括但不限于以下几个方面：一是更加智能化的数据库管理和优化，包括自动化管理、自动化决策等；二是更加高效的分布式数据库技术，以满足大规模数据处理和分析需求；三是更加安全的数据库技术，包括数据隐私保护、数据安全管理等。

本课题所选用的数据库为 MySQL 数据库，MySQL 是一种广泛使用的关系型数据库管理系统，它是开源软件，可以在多个操作系统上运行。MySQL 支持多种编程语言，如 C、C++、Java、Python 等，因此在开发 Web 应用程序时，MySQL 是一个非常流行的选择^[28]。

MySQL 数据库的主要特点之一是其可靠性和稳定性。它可以处理大量数据，并且在高负载情况下也能保持稳定性。此外，MySQL 还具有高度可扩展性，可以轻松地扩展到多台服务器上。

MySQL 的安全性也非常高。它支持多种身份验证和加密技术，以确保数据的安全性。此外，MySQL 还具有备份和恢复功能，可以轻松地备份和还原数据，以保护数据免受不良事件的影响。

MySQL 还具有高度的灵活性和可定制性。它支持多种存储引擎，如 InnoDB、MyISAM 等，可以根据特定的应用程序需求进行选择。此外，MySQL 还支持多种插件和扩展，可以轻松地添加新功能和功能。

MySQL 的基本架构包括以下几个组件：

连接器（Connection Manager）：负责管理客户端和服务端之间的连接，处理客户端的连接请求，验证客户端身份，并维护连接状态。

查询缓存（Query Cache）：在查询被执行之前，MySQL 会先检查查询缓存，如果缓存中已经存在相同的查询，MySQL 会直接返回缓存结果，避免了重复执行查询的开销。

分析器（Parser）：对 SQL 语句进行解析，检查语法和语义是否正确，并生成内部数据结构，以供后续处理。

优化器（Optimizer）：语句进行优化，选择最优的执行计划，以提高查询性能和效率。

执行器（Executor）：执行优化器生成的执行计划，读取数据，执行操作，并将结果返回给客户端。

存储引擎（Storage Engine）：负责管理数据的存储和检索，MySQL 支持多种存储引擎，如 InnoDB、MyISAM 等，每种存储引擎都有其独特的特性和优缺点^[29]。

MySQL 的基本架构可以帮助我们了解 MySQL 如何工作，以及如何优化 MySQL 的性能。对于开发人员和数据库管理员来说，了解 MySQL 的基本架构非常重要，MySQL 的基本架构图如图 2-3 所示。

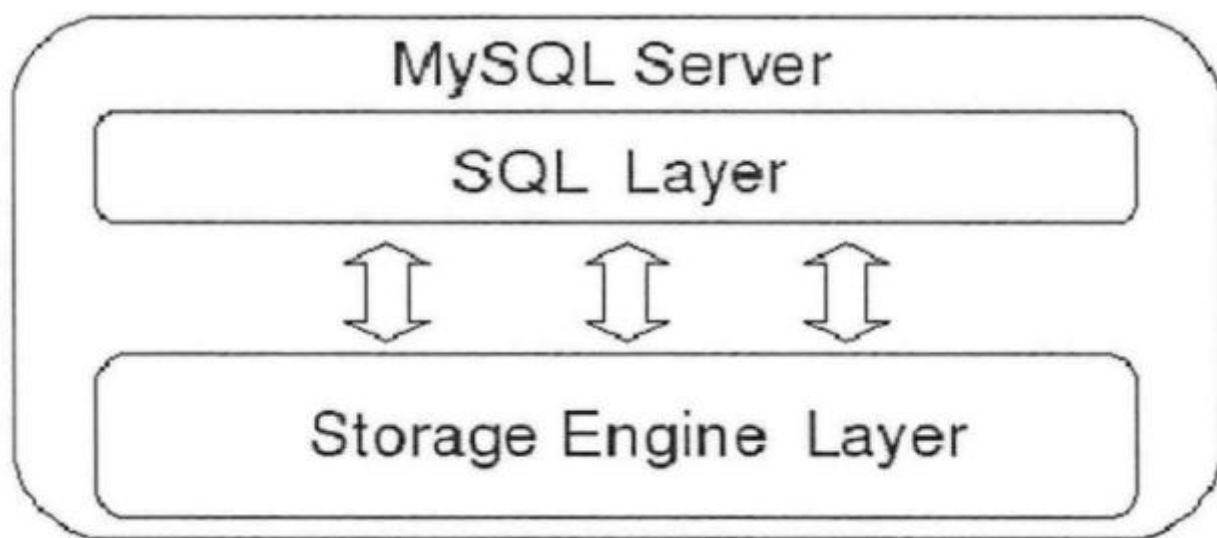


图 2-3 MySQL 架构图

从模块逻辑组成上分析，MySQL 主要由两层架构组成。第一层是服务层，服务层包括连接器、查询缓存、分析器和优化器等模块，这些模块负责接收和处理客户端请求，并将请求转换为存储引擎可以理解的格式^[30]。服务层的主要功能包括：连接管理、查询处理、缓存管理、权限管理。第二层是存储引擎层，存储引擎层是 MySQL 的底层核心，负责管理数据的存储和检索等功能。MySQL 支持多种存储引擎，如 InnoDB、MyISAM 等，每种存储引擎都有其独特的特性和优缺点。存储引擎层的主要功能包括：数据存储、数据检索、索引管理、锁管理等功能。

2.4 MyBatis 框架技术

MyBatis 是一种 Java 持久层框架，它可以将面向对象语言中的对象与数据库查询或存储语句耦合在一起，使用 XML 语言或其他注解语言^[31]。该框架源自 iBATIS 创建者团队，是 iBATIS 第三版的一个分支，并由该团队维护。MyBatis 的设计理念是将 SQL 语句和 Java 代码分离，使开发人员能够更好地维护和管理应用程序。

MyBatis 的核心组件^[32]包括 SqlSessionFactory、SqlSession、Mapper 和 Configuration 等。

其中, `SqlSessionFactory` 是 MyBatis 的入口点, 它负责创建 `SqlSession` 对象, `SqlSession` 是 MyBatis 的核心组件, 它负责执行 SQL 语句和管理事务。Mapper 是 MyBatis 的映射器, 它负责将 Java 对象映射到数据库中的表中, Mapper 通常使用 XML 或注解的方式进行配置。Configuration 是 MyBatis 的配置对象, 它负责管理 MyBatis 的全局配置信息, 如数据源、事务管理器、插件等^[33]。MyBatis 的使用流程通常包括以下几个步骤:

- 1) 配置 `SqlSessionFactory`: 通过 MyBatis 的配置文件或 Java 代码创建 `SqlSessionFactory` 对象。
- 2) 配置 Mapper: 通过 XML 或注解的方式配置 Mapper, 将 Java 对象映射到数据库中的表中。
- 3) 获取 `SqlSession`: 通过 `SqlSessionFactory` 对象获取 `SqlSession` 对象。
- 4) 执行 SQL 语句: 通过 `SqlSession` 对象执行 SQL 语句, 如查询、插入、更新和删除等操作。
- 5) 提交事务: 如果需要进行事务管理, 则需要通过 `SqlSession` 对象进行事务的提交或回滚。

MyBatis 的优点包括灵活性、易用性、易于调试、高性能、易于测试、安全性、易于维护、可扩展性和易于集成等。它是一种功能强大的 Java 持久化框架, 被广泛应用于各种类型的 Java 应用程序中^[34]。MyBatis 架构图如图 2-4 所示。

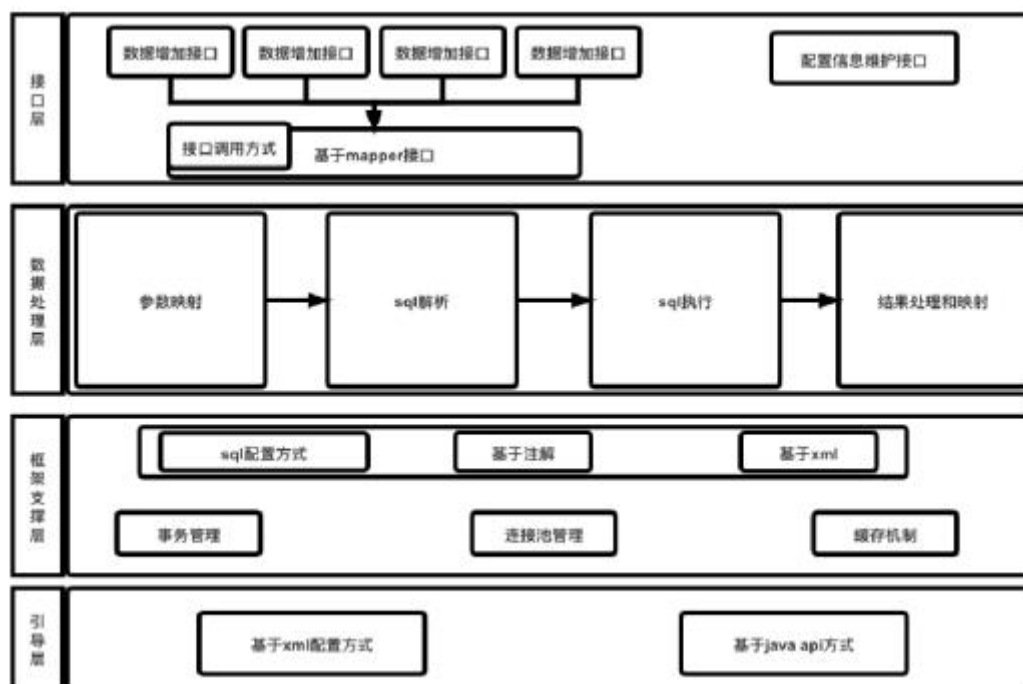


图 2-4 MyBatis 架构图

2.5 本章小结

本章介绍了在开发社区管理系统过程中所涉及到的技术框架，并对这些技术框架的运行原理和优点进行了介绍，主要有：前后端分离开发技术，搭建系统前端的 Vue.js 框架技术和系统的后台搭建框架 Spring Boot 以及持久层框架 MyBatis，除此之外还介绍了使用的 MySQL 数据库技术。通过对各种技术的介绍，为后期项目开发提供理论铺垫和技术可行性基础。

3 系统需求分析

需求分析是软件开发过程中非常重要的一步，它涉及到对用户需求的收集、分析、梳理和确认，为后续的软件设计、开发、测试和维护提供了基础和指导。进行需求分析可以确定用户需求，这是软件开发的基础。只有了解用户的需求，才能为用户提供满意的软件产品；可以防止开发人员在软件开发过程中偏离用户需求，从而降低开发成本和时间；可以帮助开发人员更好地理解用户需求，从而设计出更加符合用户期望的软件产品，提高软件质量；可以帮助开发人员尽可能地理解用户需求，避免在后续开发过程中频繁修改需求，减少开发成本和时间；可以帮助项目开发人员更好地预测和规避项目风险。通过对用户需求的深入分析，可以发现需求之间的冲突、不一致或不完整，从而及时进行调整和修正，降低项目风险；可以为测试人员提供测试用例和测试场景，为维护人员提供软件功能和性能的参考，从而提高测试和维护的效率和质量。

因此本章将在使用者的角度剖析用户在本系统中的需求，并将这些需求根据功能分类，使用软件工程语言阐述各个功能模块所能实现的具体功能和关键需求点在什么地方。目的是为了更好的理解用户需求，避免后续开发过程中偏离需求或是频繁的修改需求。最后在考虑应用的性能、可靠性、安全性、可维护性、可用性等非功能性需求，为接下来系统的设计和实现提供蓝图。

3.1 系统功能需求概述

近年来，国内已经出现了数字化系统成功应用在社区管理方向的案例。很多地区的社区工作也从纸质化办公向着数字化办公转化。与此同时，这些社区所能涵盖的工作和服务也扩展到了管理社区居民信息，管理社区楼栋信息，管理社区内车位，为社区内居民提供社区服务等多方面内容。由此可见，在国内部分地区，数字化社区得到了很大程度的普及和实现^[35]。相比于过去的人工工作方式，数字化社区的实现可以大大提高社区工作人员的办事效率和服务质量。同时，社区的管理在管理效率和深度方面还是有很大的发展潜力和发展空间。比如在国内社区内各项信息的交互仍然缺少信息化管理的身影，现有社区大都是采用张贴纸质通知的方式进行社区内的信息公示公告工作，疫情期间，部分社区进行消息通知的方式竟然是社区工作人员上门通知，这种情况不在少数。这样低效的方式不仅会占用了社区居民的时间，也会耽误物业工作人员的工作，降低工作效率。所以本论题针对以上情况，首先分析本系统的用户用例，之后根据整体制定特定功能的需求分析。

3.2 系统用户及用例分析

1) 社区居民用例

社区居民是系统的主要用户之一，他们的主要目的是使用系统来获取社区内的各种信息、服务和资源。以下是社区居民的主要用例：

注册和登录：居民需要注册并登录系统才能使用系统的各种功能。

查看社区信息：居民可以查看社区的公告、通知、活动等信息。

提交服务请求：居民可以通过系统提交各种服务请求，如维修、清洁、保安等。

支付费用：居民可以通过系统支付社区的各种费用，如物业费、水电费等。

社区论坛：居民可以在系统的论坛中发表观点、提出建议、交流信息等。

投诉建议：居民可以通过系统提交投诉和建议。

2) 社区管理员用例

社区管理员是系统的管理者，他们的主要目的是管理社区和居民。以下是社区管理员的主要用例：

登录和权限管理：管理员需要登录系统并管理用户权限，以确保只有授权用户才能访问系统。

社区信息管理：管理员可以管理社区的公告、通知、活动等信息，包括发布、修改和删除等。

服务请求管理：管理员可以查看和处理居民提交的各种服务请求，如维修、清洁、保安等。

费用管理：管理员可以管理社区的各种费用，包括收取、审核、退款等。

居民管理：管理员可以管理社区的居民信息，包括注册、审核、修改和删除等。

社区论坛管理：管理员可以管理社区的论坛，包括发表、修改、删除和审核等。

投诉建议管理：管理员可以管理居民提交的投诉和建议，包括查看、处理和回复等。

统计分析：管理员可以通过系统对社区的各种信息进行统计分析，如费用收支、服务请求处理情况等。

用例图如下图 3-1 所示：

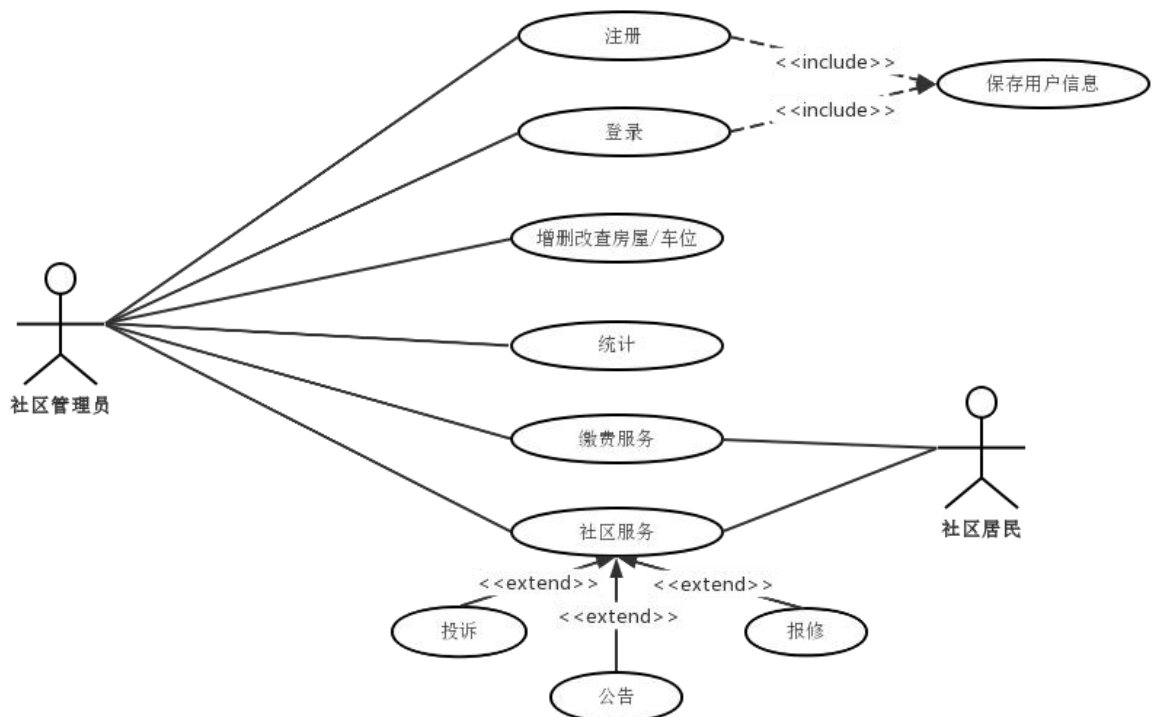


图 3-1 社区管理系统用例图

总的来说，社区居民用例和社区管理员用例都是系统的核心用例，通过这些用例的设计和实现，可以帮助管理员更好地管理社区和居民，提高社区管理的效率和质量。

3.3 系统功能需求分析

本文的社区管理系统分为 6 个功能模块，这 6 个模块分别对应着社区数字化管理过程中关键的业务内容，下面将各个模块的使用场景对各个模块的功能做简要的内容介绍。

1) 居民管理系统：社区居民业主用户在第一次使用居民管理系统时需要使用注册功能，经过社区管理员核对后将居民加入业主系统才能正常使用居民管理系统。社区管理员也可以使用居民管理系统，社区管理员通过该系统查询用户、修改用户基本信息、添加新用户，删除搬离社区的用户。

2) 房屋管理系统：房屋管理系统的主要服务对象是针对社区的楼栋、房屋和单元等单位。在进入房屋管理系统的主界面后，管理员用户可以进行楼栋管理、单元管理、和房屋管理操作。其中，楼栋管理模块主要针对楼栋的信息进行功能上的拓展，管理员可以在楼栋管理系统对楼栋信息进行增加、删除、修改信息、查找信息等操作；单元管理模块则是针对楼栋的各个单元进行管理，主要包括对各个楼栋的单元进行标号、分配、搜索、添加、修改删除等功能；房屋管理模块则是管理社区内所有房间的模块，主要功能包括更新房屋

名称、编号、面积、隶属楼栋和单元等，房屋的入住情况及入住人员和入住时间等等^[40]。此模块还包括对社区内所有房屋的增删改查等功能。

3) 车位管理系统：车位管理系统主要服务于管理员对于社区车位的管理等功能。在进入车位管理系统的主界面后，管理员用户可以对社区的车位进行管理，主要包括浏览社区内的所有车位，还可以在社区新建设或者裁撤车位时对车位进行增加和删除。除此之外管理员用户可以查看、修改已有的车位信息、车位分配信息、车位分配类型和具体的车位分配人员以及分配时间等。

4) 生活缴费系统：生活缴费系统主要针对社区居民生活缴费等需求进行分析。在进入生活缴费系统主界面后，管理员用户可以设置社区内所有已有的缴费类型，比如增加新的收费项目，删除不需要的收费项目，或者对收费项目进行调整等功能。同时，管理员用户还可以针对用户的缴费情况进行管理，比如设定特定用户的缴费类型和该类型下所对应的缴费数值，缴费截止时间，缴费状态等。

5) 社区服务系统：社区服务系统针对于社区居民在社区内的基本服务型需求进行分析。作为管理员用户，可以在社区服务系统发布新的公告，还可以浏览和查询所有公告的详细信息，删除公告、修改公告等操作；作为社区居民用户，可以在社区服务系统进行投诉操作并描述投诉内容。而作为管理员用户则可以浏览所有社区居民提交的投诉，并对这些投诉进行修改删除等操作；作为社区居民用户，可以在社区服务系统进行设施报修操作并描述报修内容。作为管理员用户则可以浏览所有社区居民的报修内容包括报修内容、报修人、报修时间等，并可以对报修内容进行修改等操作

6) 社区居民操作系统：这一系统是针对社区居民的操作模块，社区居民用户需要使用注册和登录功能来进入社区居民的功能界面，用户登陆后可以查看自己的基本信息，包括姓名、性别、住址、联系方式、电子邮箱等^[41]；社区居民用户可以在社区服务系统子模块以直接看到社区发布的公告信息如公告标题，公告详细信息；还可以查看或者提交当前社区居民用户的投诉、报修信息等；社区居民用户可以在缴费系统子模块查询自己的缴费账单，缴费信息，截止时间等；

3.4 系统非功能需求分析

对于任何软件系统而言，系统的非功能性需求的重要性都不低于系统的功能性需求，因为非功能性需求将直接影响到用户使用系统时的体验和系统使用时的稳定性。所以本论题在确定了以上功能性需求的同时，还需要满足以下非功能性的需求，包括：

- 1) **可靠性**: 系统应该具有高可靠性, 能够在任何情况下保持稳定运行, 避免出现宕机、数据丢失等问题。
- 2) **安全性**: 系统应该具有高度的安全性, 能够对用户数据进行保护, 防止未经授权的访问或恶意攻击。
- 3) **可扩展性**: 系统应该具有良好的可扩展性, 能够随着用户量的增加而扩展, 避免出现性能瓶颈。
- 4) **易用性**: 系统应该具有良好的用户体验, 包括易于使用的界面、清晰的操作流程、易于理解的提示信息等, 以使用户能够方便地使用系统。
- 5) **性能**: 系统应该具有高性能, 能够快速响应用户的请求, 避免出现卡顿、延迟等问题。
- 6) **可维护性**: 系统应该具有良好的可维护性, 能够方便地进行维护和升级, 以确保系统的稳定性和可靠性。
- 7) **可用性**: 系统应该具有高可用性, 能够保证 24 小时不间断地运行, 避免出现系统故障或停机等问题。

3.5 本章小结

在本章中, 我们对社区管理系统的需求进行了全面的分析和概述。首先, 我们对社区管理系统的需求进行了总体概述, 以确定该系统的目标和功能。接着, 我们详细分析了该系统的用例, 以确定系统的主要功能和用户需求。这些用例包括居民管理系统, 房屋管理系统, 车位管理系统, 生活缴费系统, 社区服务系统, 社区居民操作系统等。

在确定了系统的用例之后, 我们进一步确定了该系统的功能性需求。这些需求包括了各个用例的具体功能, 例如居民管理系统需要包括居民信息的管理和查询, 房屋管理系统需要包括房屋信息的管理和查询, 车位管理系统需要包括车位信息的管理和查询, 生活缴费系统需要包括缴费信息的管理和查询, 社区服务系统需要包括社区服务的发布和查询, 社区居民操作系统需要包括居民个人信息的管理和查询等。

除了功能性需求之外, 我们还对该系统的非功能性需求进行了分析。这些需求包括系统的可靠性、可用性、安全性、性能等方面。例如, 系统需要保证数据的安全性和隐私性, 同时也需要保证系统的可用性和性能, 以满足用户的需求。

4 社区管理系统的设计

上一章已经完成了对社区管理系统的需求分析，接下来将根据对各个模块的需求分析进行相对应功能上的设计，并针对各个功能模块的设计点进行相应的阐述，然后站在用户的视角上，具体说明社区管理系统 6 个功能模块的操作逻辑和设计方式。为接下来系统的开发实现打下设计基础。

4.1 系统总体架构设计

社区管理系统是一种基于 B/S 架构的软件系统，它由三个主要组件组成：Web 浏览器、Web 服务器和数据库服务器。这种架构使得用户可以通过浏览器轻松访问系统，并与服务器进行通信。Web 服务器充当中间人的角色，它接收来自浏览器的请求并将其发送到数据库服务器进行处理。数据库服务器则负责存储和管理系统中的所有数据。这种架构的优点在于它可以实现跨平台和跨浏览器的兼容性，并且能够提供高度安全性和可靠性，以确保用户的数据和隐私得到保护^[42]。此外，B/S 架构还具有灵活性和可扩展性，可以根据需要添加新的功能或模块。同时，由于 Web 浏览器已经成为了人们生活中不可或缺的工具，因此社区管理系统采用 B/S 架构也能够极大地提高用户的使用体验和便捷性。总之，采用 B/S 架构的社区管理系统是一种高效、安全、可靠、灵活和易用的解决方案，能够满足不同规模和需求的社区管理需求。系统总体架构如下图 4-1 所示。

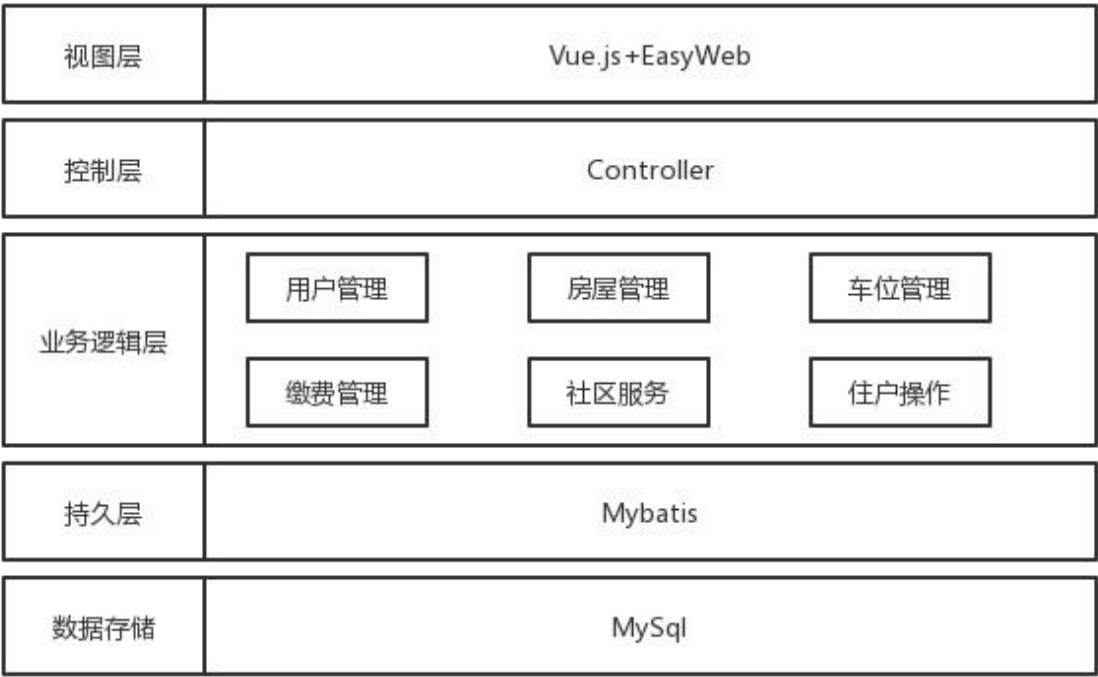


图 4-1 系统逻辑结构图

社区管理平台的架构图如上，该系统被划分为四个逻辑层：视图层、控制层、业务逻辑层和持久层。视图层负责展示用户界面，包括网页、移动应用等。视图层主要处理用户的请求和响应，将用户的操作转化为控制层可以理解的指令。控制层负责处理用户请求和响应，将用户请求转化为业务逻辑层可以处理的数据格式，同时将业务逻辑层处理的数据转化为用户可以理解的格式。业务逻辑层负责处理业务逻辑，包括社区管理平台的各种功能，如用户管理、话题管理、帖子管理、评论管理等等。业务逻辑层是整个系统的核心，它处理所有的业务逻辑，包括数据的处理、计算、验证、存储等等。持久层负责数据的存储和访问。持久层通常使用数据库来存储数据，包括用户信息、帖子、评论等等。持久层还负责数据的访问，包括数据的读取、写入、更新等等^[43]。

这四个逻辑层相互独立，各自承担不同的职责，通过接口进行交互和通信。这种分层的架构可以使系统更加灵活、可扩展和易维护。例如，如果需要增加一个新的功能，只需要在业务逻辑层中添加相应的代码，而不需要修改其他层的代码；如果需要更换数据库，只需要修改持久层的实现，而不需要修改其他层的代码。这种分层的架构也可以使系统更加可测试和可维护，因为每个层都可以单独进行测试和维护。同时，这种分层的架构也可以使不同的开发人员或团队分别负责不同的层，提高开发效率和代码质量。

4.2 系统功能设计

该社区管理系统主要由 6 个模块组成，包括：居民管理系统、房屋管理系统、车位管理系统、缴费系统、社区服务系统和社区居民操作系统模块。下面是这 6 个模块的具体功能设计和阐述。

4.2.1 居民管理系统设计

该系统的主要目的是帮助管理员对社区居民进行管理，包括访问控制和居民信息维护。其中，访问控制是系统的基础，它需要管理员对用户进行登录和授权访问的管理，以确保系统的安全性和保密性。同时，维护居民信息也是很重要的，因为社区居民的信息需要被妥善管理，以便管理员能够快速准确地找到需要的信息。

在访问控制方面，管理员需要对用户进行身份验证，以确保只有授权用户才能访问系统中的信息。管理员可以通过用户名和密码等方式对用户进行身份验证，以确保只有授权用户才能访问系统中的信息。此外，管理员还可以设置不同的用户权限，以限制他们能够访问的信息和功能。

在居民信息维护方面，管理员需要能够对居民信息进行增删改查等操作。管理员可以

通过系统界面方便地添加新的居民信息，修改已有的居民信息，或者删除不再需要的居民信息。同时，管理员还可以通过搜索和过滤等功能快速找到需要的居民信息，以便更好地管理社区居民。该系统的功能结构如下图 4-2 所示。

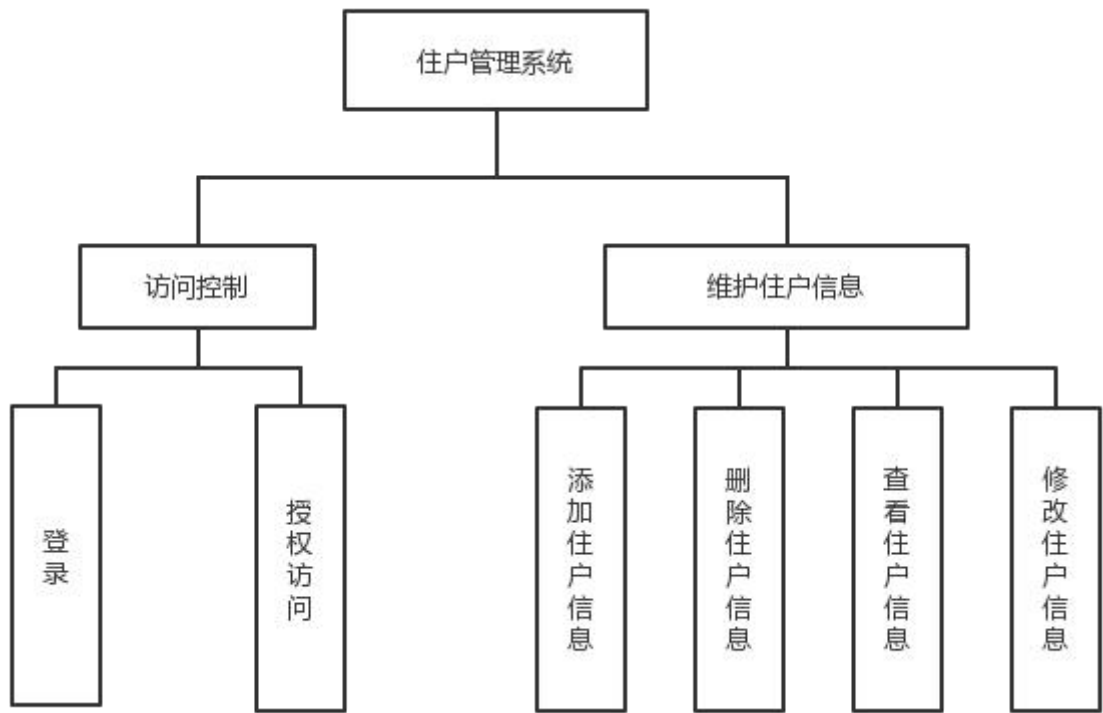


图 4-2 居民管理系统结构图

登录是指用户在使用系统前需要通过身份验证，以便系统能够确定用户的身份并授权其访问系统。在社区管理系统中，要求用户输入用户名和密码等信息进行身份验证。如果验证通过，用户就可以获得系统的访问权限，并进入系统进行操作。如果验证失败，则用户将无法访问系统。

授权访问是指管理员可以对用户的访问权限进行控制，以限制他们能够访问的信息和功能。在社区管理系统中，管理员可以设置不同的用户权限，以确保只有授权的用户可以访问系统中的敏感信息。例如，管理员可以设置某些用户只能查看居民信息，而不能进行修改或删除操作。管理员还可以设置某些用户只能访问特定的模块或功能，而不能访问其他模块或功能。

维护用户信息。社区管理员可以查询社区居民的具体信息，也可以在系统中添加新的居民信息包括姓名、地址、电话号码等信息。如果居民退出了社区，管理员可以在系统中删除不再需要的居民信息。删除操作需要谨慎，管理员需要确保删除的信息是正确的，并且不会影响到其他相关信息。同时如果居民对自己的个人信息有更新，也需要联系管理员

进行个人信息的更新，由管理员核实后再对居民的信息进行更新，管理员修改社区居民信息的流程如下图 4-3 所示。

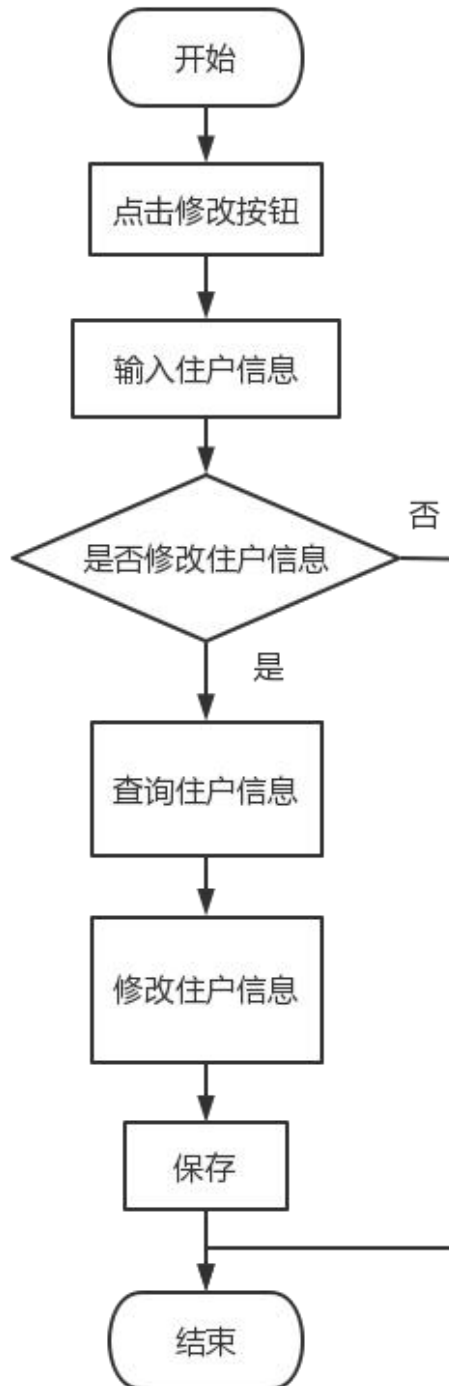


图 4-3 居民信息修改流程图

4.2.2 房屋管理系统设计

房屋信息管理功能，管理员可以轻松地管理社区内的房屋信息，包括房屋的内部信息、

房屋的具体位置、房屋的维护历史、房屋的出租情况等等。管理员可以通过增删改查等功能，对房屋信息进行灵活的管理，以满足社区内不同房屋的管理需求。

房屋分配功能则是针对社区内的房屋资源进行分配，包括添加房屋居民信息和删除房屋居民信息等功能。管理员可以通过该功能，对社区内的房屋进行合理的分配，确保房屋资源的最大化利用。同时，该功能还可以帮助管理员更好地管理社区内的人员信息，包括居民的基本信息、租赁信息等等。其功能如下图 4-4 所示。

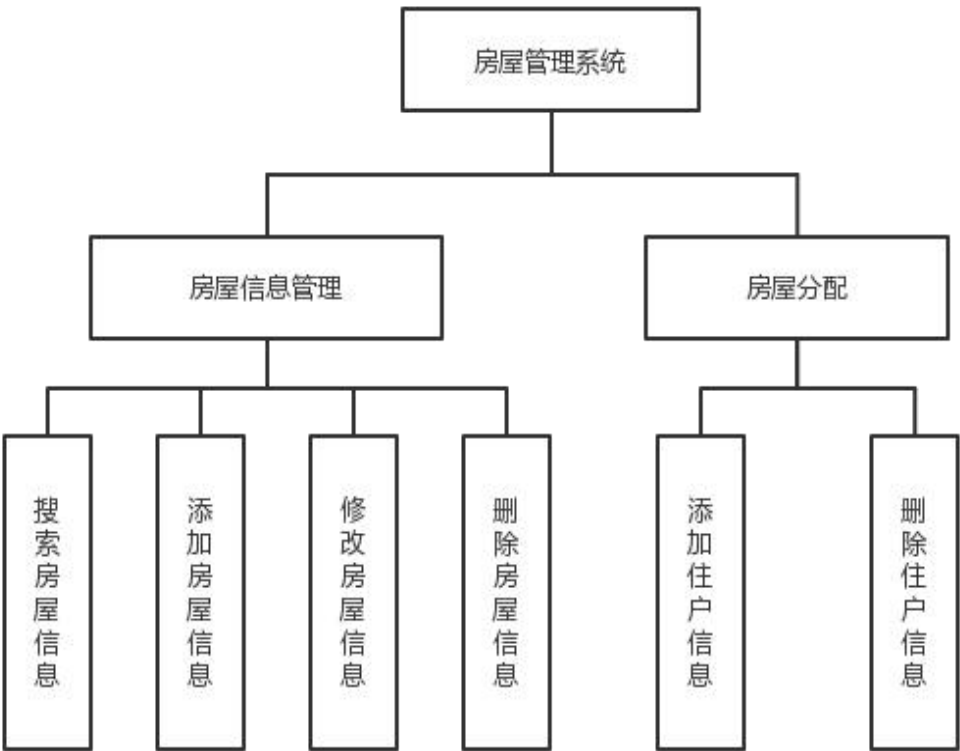


图 4-4 房屋管理系统结构图

房屋信息管理。社区管理员在使用该模块功能时，根据该系统所应用的社区的实际情况进行配置工作，具体工作则是添加或删除楼栋信息或单元信息，针对社区管理系统所应用的社区具体包含多少楼栋，各个楼栋包括多少单元进行详细的配置，适配好社区管理系统所应用的社区信息。一个社区中最最重要的就是社区内的房屋信息，在配置好楼栋和单元信息后，社区管理员用户进行各个房屋信息的配置，包括房屋所隶属的楼栋、单元，房屋面积，房屋入住状态等的配置。

房屋分配。社区管理员用户除了能够对社区服务系统进行房屋信息管理外，在有新居民入住时，还可以根据新居民所入住的房屋进行房屋户主信息的配置，包括户主的详细个人信息，和户主的入住时间，如果是出租房屋还有房屋的到期时间。

社区管理员用户在查看房屋信息时，需要使用房屋信息接口来获取房屋的基本信息，例如楼栋和单元。然而，由于房屋信息模块返回的信息不包括与房屋相关的居民信息，因此需要调用居民信息模块接口来获取这些信息。通过这种方式，管理员可以获得完整的房屋信息，包括房屋的位置、大小、居民信息等。在将返回的数据填充到房屋信息查看页面之前，管理员还需要确保数据的准确性和完整性，以便提供最准确的房屋信息给用户。这种方法不仅可以提高管理员的工作效率，还可以提高用户的满意度和信任度。房屋管理系统增删查改操作的时序图如下图 4-5 所示

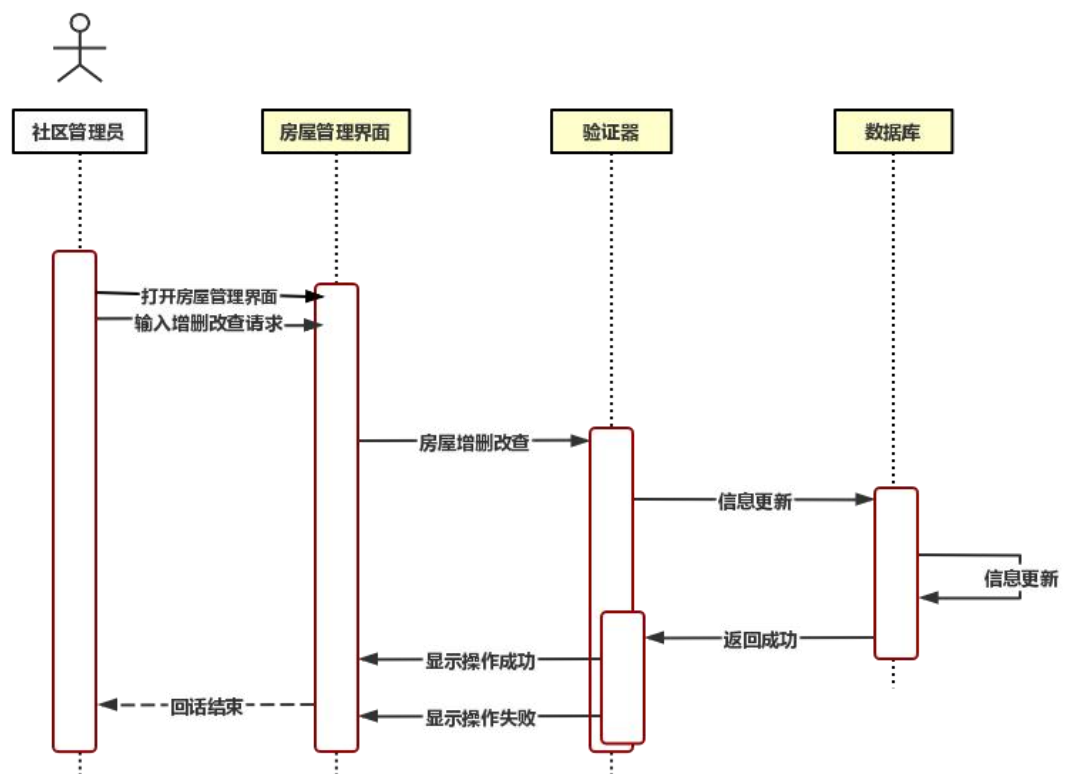


图 4-5 房屋管理时序图

4.2.3 车位管理系统设计

车位管理系统，管理员可以自由管理社区内的车位，即对居民的车位进行统一的信息化管理，只有社区管理员拥有操作这一系统功能的权限。其中具体功能包括车位信息管理和车位分配功能，其功能如下图 4-6 所示。

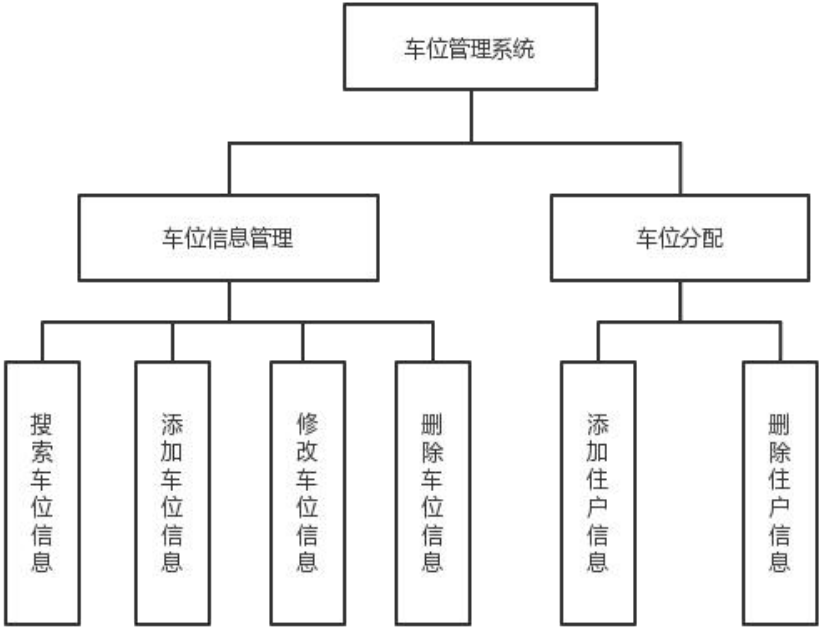


图 4-6 车位管理系统结构图

车位信息管理。社区管理员用户可以针对社区管理系统所应用的社区具体车位数量及车位类型进行详细的配置，具体工作是增加或者修改车位的相关信息，适配好社区管理系统所应用的社区的具体情况，包括社区所包含的车位数量，序号，车位类型，车位分配状态等的配置。

车位分配。社区管理员用户除了能够对社区服务系统进行车位信息管理外，在有居民新申请车位时，还可以对其已申请到的车位进行配置，包括车位的使用人和的车位的起始使用时间，如果是出租车位还有车位的到期时间。

社区管理员用户在进行车位信息查看时，需要先调用车位模块的车位信息接口，获取车位的基本信息，包括车位序号、车位名称、车位分配状态、车位类型等信息。之后判断车位是否已分配。如果车位未分配，则无需调用居民信息模块的接口，直接将车位基本信息填充到车位信息查看页面即可。但是如果车位已分配，则需要调用居民信息模块的查看居民车位具体信息接口，获取车位相关联的居民信息包，包括车位的起始使用时间和截止使用日期等。最后将车位的基本信息和居民信息包合并，并填充到车位信息查看页面中，以展示车位的具体信息。

需要注意的是，在调用居民信息模块的接口时，需要提供车位的唯一标识符（例如车位序号），以便该接口能够正确地返回与该车位相关联的居民信息包。修改车位信息详情时序图如下图 4-7 所示。

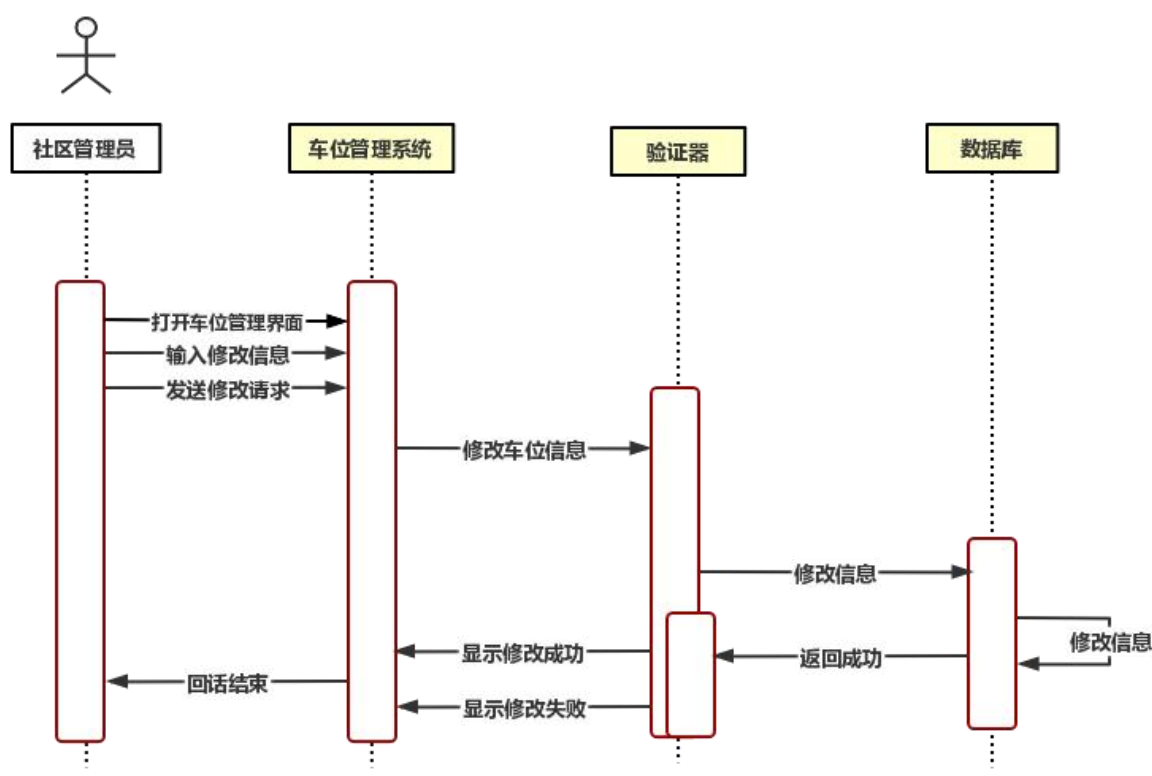


图 4-7 修改车位信息时序图

4.2.4 缴费管理系统设计

缴费管理系统的功能是管理员对整个社区的缴费项目进行管理，只有社区管理员拥有操作这一系统功能的权限。其中具体功能包括缴费类型管理、居民账单管理和居民缴费等功能，其功能如下图 4-8 所示。

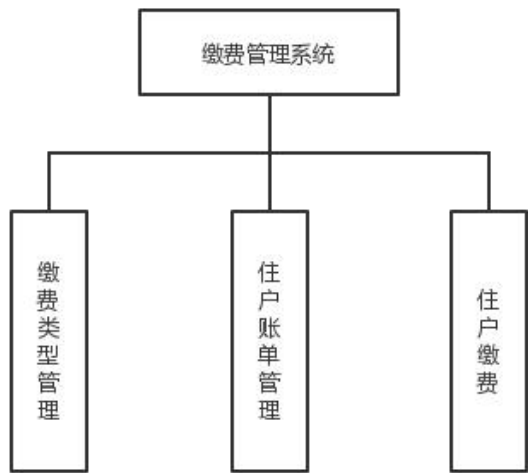


图 4-8 缴费管理系统结构图

缴费类型管理。社区管理员用户可以针对社区管理系统所应用的社区进行缴费类型的详细配置，具体工作是增加、删除、修改社区内需要缴费的项目类型，适配好社区管理系

统所应用的社区的具体情况。如水电费、煤气、物业费等的配置。

居民账单管理。社区管理员用户除了能够对社区服务系统进行缴费类型的配置外，还可以对社区内居民应缴费用相关信息进行统计，包括缴费类型、欠费用户名称、应缴费用、缴费截止时间、缴费状态等。如果居民已经缴费，在社区管理员核对过后，可以进行确认缴费操作，更改缴费状态为已缴费。

社区管理员可以通过缴费信息模块来查看社区居民的缴费情况。当管理员查看某名居民的缴费信息时，可以调用居民信息模块的居民缴费信息接口来获取相关数据。这些数据包括缴费的序号、缴费名称、居民名、欠费用户名称、应缴费用、缴费截止时间、缴费状态等信息。然后将返回数据填充到缴费信息查看页面中，使管理员可以方便地查看该居民的缴费情况。发布缴费信息时序图如下图 4-9 所示。

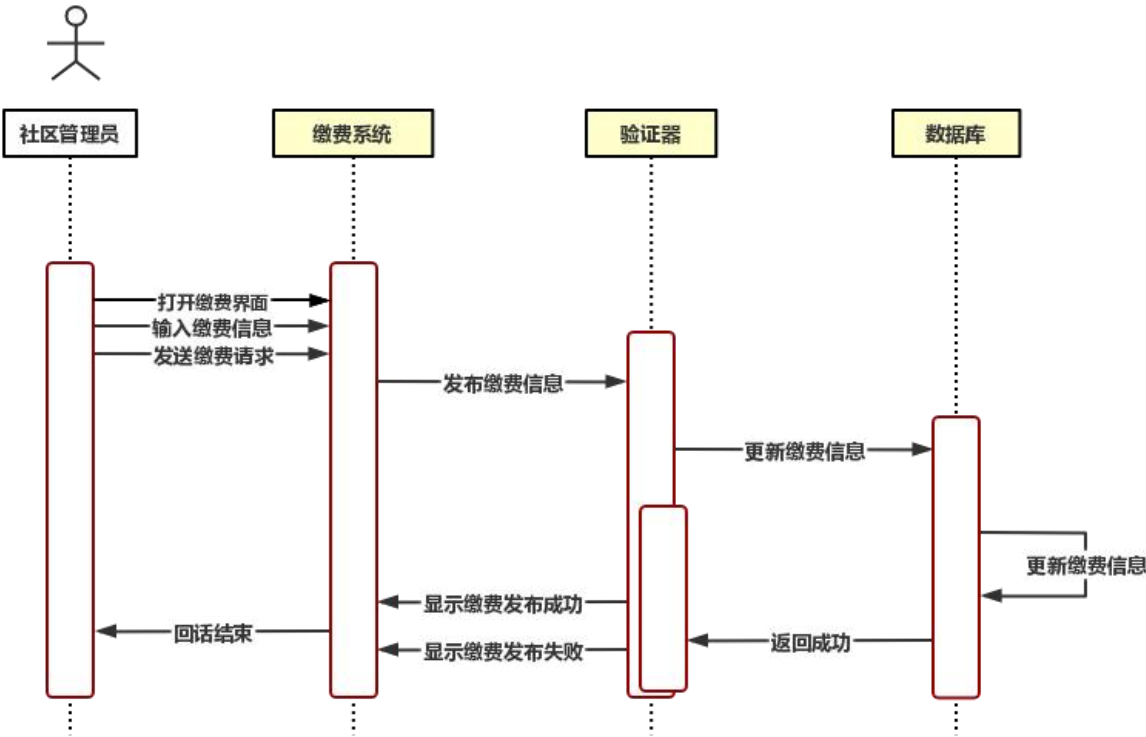


图 4-9 发布缴费信息时序图

4.2.5 社区服务管理系统设计

社区服务管理系统模块的功能是为整个社区的居民提供一个数字化的社区服务平台，对社区居民用户所需要的的社区服务项目进行管理，对社区服务流程使用数字化的记录方式，只有社区管理员拥有操作这一系统功能的权限。其中具体功能包括社区居民服务模块和维护社区服务信息等功能，其功能如下图 4-10 所示。

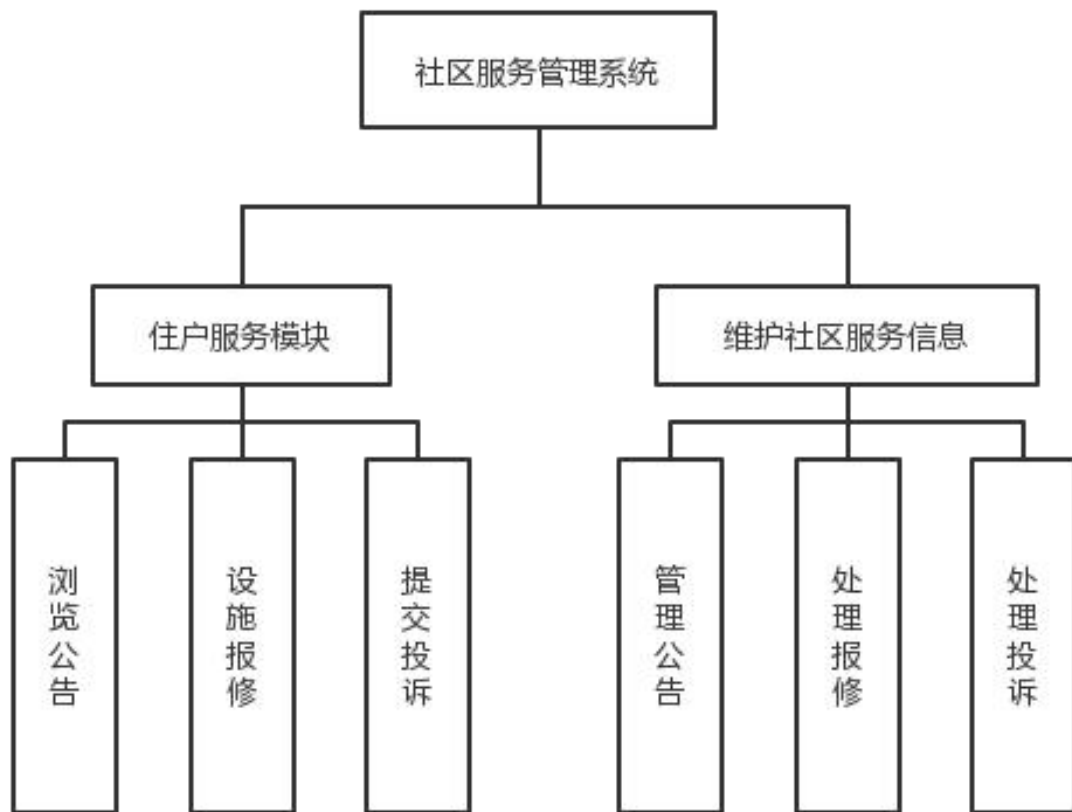


图 4-10 社区服务管理系统结构图

居民服务功能。社区居民用户在登录系统时，可以浏览社区管理员发布的最新公告。在相关设施需要维修时，可以进行设备报修操作。在对社区内其他居民或是社区工作人员的工作有不满时还可以向管理员进行投诉操作。

维护社区服务信息功能。社区管理员可以看到管理员发布的所有公告信息包括公告的大标题、公告的详细内容、公告发布者、发布时间等等信息。还可以看到用户提交的社区服务申请，包括社区服务的内容，申请社区服务的用户，申请社区服务的时间，社区服务的处理结果等信息。如果居民退出了社区，为了避免不必要的风险，管理员需要将对应的账号给注销，并清理该账号申请的相关的社区服务。

社区管理员用户在进行社区服务信息查看时，包含公告信息，报修信息，投诉信息等。社区管理员用户调用社区服务模块的社区服务接口，获取投诉或报修的内容、申请的用户、申请时间、服务状态、服务结果等基本信息。然后将返回数据填充社区服务对应的投诉或者报修相关查看页面中，以便查看相关信息。管理员查看社区服务信息的时序图如下图 4-11 所示。

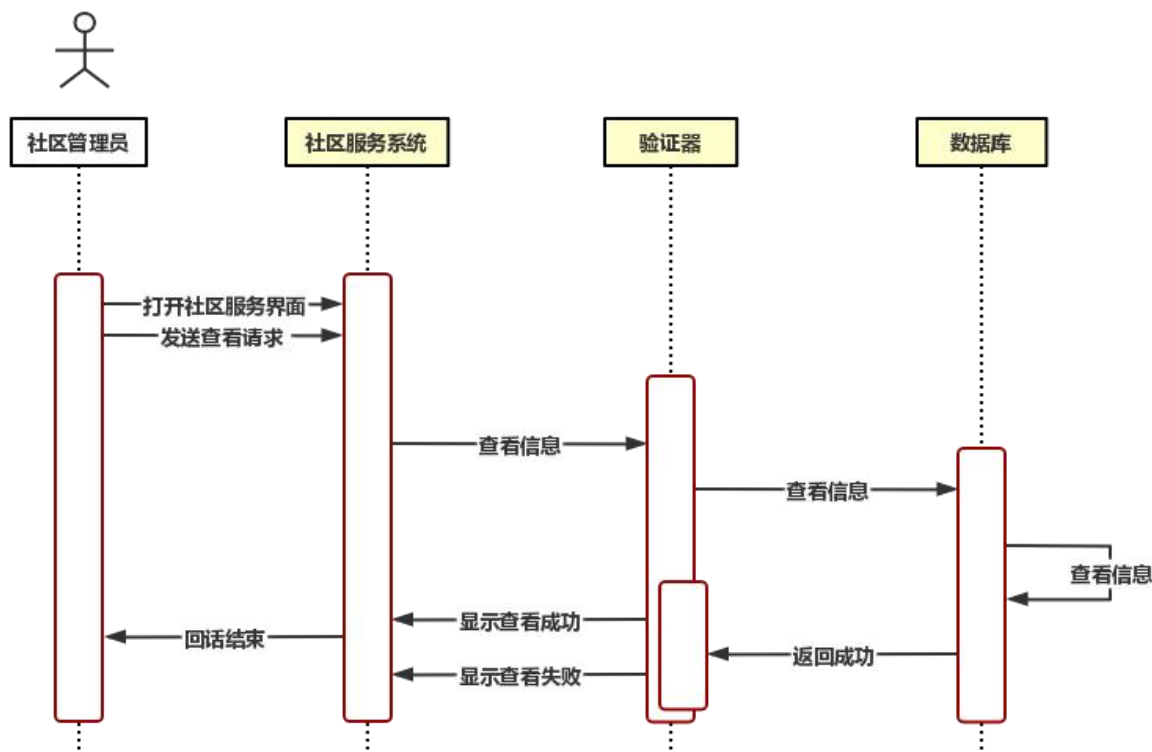


图 4-11 查看社区服务信息时序图

4.2.6 社区居民操作系统设计

社区居民操作系统是本论题中最重要的模块之一，这一模块是针对社区居民用户的操作模块，社区居民用户需要使用注册和登录功能来进入操作界面，居民可以在操作界面查看自己的基本信息，包括姓名、性别、住址、联系方式、电子邮箱等；社区居民用户可以在社区服务系统子模块以直接看到社区发布的公告信息如公告标题，公告详细信息；还可以查看或者提交当前社区居民用户的投诉、报修信息等；社区居民用户可以在缴费系统子模块查询自己的缴费账单，缴费信息，截止时间等；其功能如下图 4-12 所示。

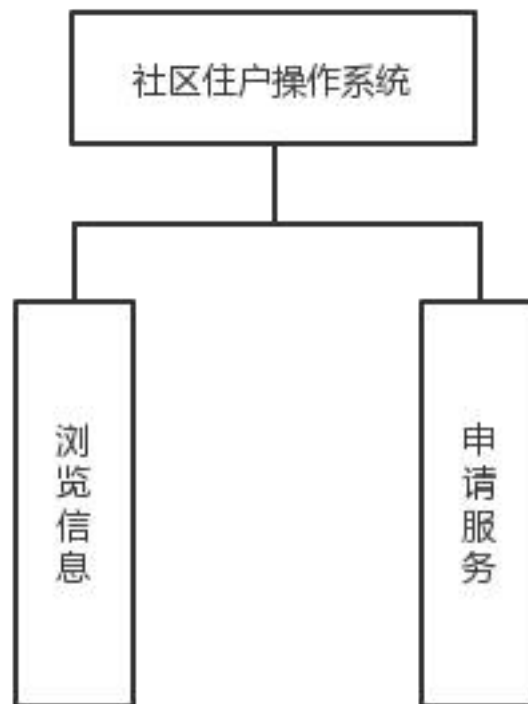


图 4-12 社区居民操作系统结构图

居民服务模块。社区居民在登录系统时，可以浏览社区管理员发布的最新公告。在相关设施需要维修时，可以进行设备报修操作。在对社区内其他居民或是社区工作人员的工作有不满时还可以向管理员进行投诉操作。

当社区管理员用户需要查看某个社区居民用户提交的社区服务信息时，可以通过调用社区服务模块的社区服务接口来获取相关信息。社区服务接口可以提供投诉或报修的内容、申请的用户、申请时间、服务状态、服务结果等基本信息。获取到这些信息后，社区管理员用户可以将数据填充到社区服务对应的投诉或报修相关查看页面中，以便查看相关信息。

4.3 数据库设计

为了更好理解数据之间的关系和属性，从而更好的设计和优化数据库模型，绘制该系统的实体关系图。该系统包括了管理员、车位管理、房屋管理、社区服务和社区居民这些实体。管理员实体包含管理员的姓名、用户名和密码等属性，与车位管理、房屋管理和社区服务实体存在一对多的关系。车位管理实体包含车位的编号、车位类型和所属房屋等属性，与管理元实体存在多对一的关系。房屋管理实体包含房屋的编号、楼栋、单元和房间等属性，与管理元实体存在多对一的关系。社区服务实体包含缴费服务、报修服务和投诉服务等属性，与管理元实体存在多对一的关系。社区居民实体包含居民的个人信息、所属房屋、车位和社区服务等属性，与管理元实体存在多对一的关系。系统主要业务 ER 图如图 4-13 所示。

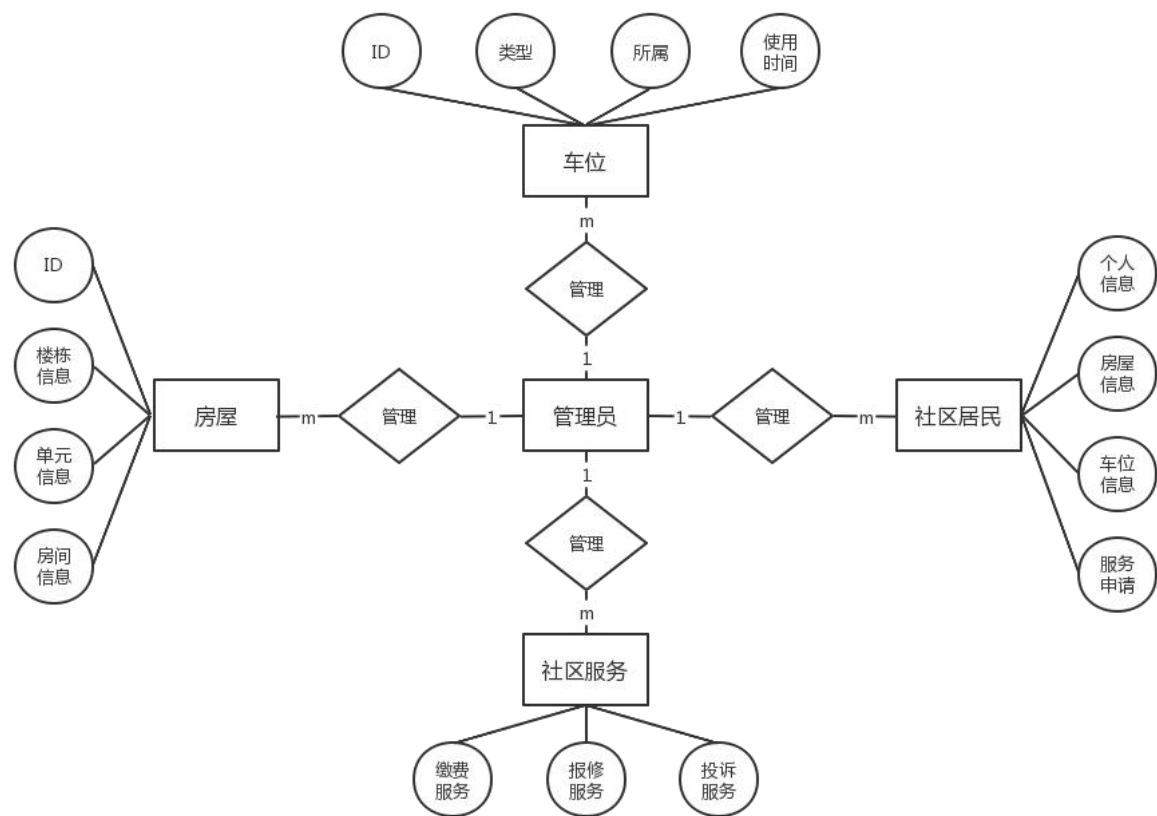


图 4-13 社区管理系统 E-R 图

该系统的表结构核心为社区管理员，主要的表的字段都是社区管理员和社区居民，本系统涉及的表共有 13 张，其中重点表格及其具体内容已经详细在本节列出。

(1) 管理员信息表。该表记录了本社区管理系统中管理员的基本信息，管理员用邮箱作为管理员账号，密码自行设置，用邮箱登入系统。该表包括管理员 id、姓名、密码、电话号码、性别、邮箱等字段。表格结构如下表 4-1 所示。

表 4-1 管理员信息表

字段名	数据类型	约束	备注
id	int	主键	管理员 id
username	varchar	非空	管理员姓名
password	varchar	非空	管理员密码
phone	varchar	非空	管理员电话号码
sex	int	非空	管理员性别
email	varchar	非空	管理员邮箱

(2) 社区居民信息表。该表记录了本社区管理系统中社区居民的个人信息，社区居民申请账号登录系统。该表包括社区居民 id、姓名、密码、电话号码、性别等字段。表格结构如下表 4-2 所示。

表 4-2 社区居民信息表

字段名	数据类型	约束	备注
id	int	主键	居民 id
username	varchar	非空	居民姓名
password	varchar	非空	居民密码
phone	varchar	非空	居民电话号码
sex	int	非空	居民性别
user_room	varchar	外键	居民住址信息
user_car	varchar	外键	居民车辆信息
user_payment	varchar	外键	居民缴费信息

(2) 楼栋信息表。该表记录了本社区管理系统中存入的楼栋信息，该表包括楼栋 id、楼栋名称、是否为电梯房等字段。表格结构如下表 4-3 所示。

表 4-3 楼栋信息表

字段名	数据类型	约束	备注
id	int	主键	楼栋 id
name	varchar	非空	楼栋名称
type	Int	非空	是否为电梯房

(3) 单元信息表。该表记录了本社区管理系统中存入的单元信息，该表包括单元 id、单元名称、楼栋 id 等字段。表格结构如下表 4-4 所示。

表 4-4 单元信息表

字段名	数据类型	约束	备注
id	int	主键	单元 id
name	varchar	非空	单元名称
building_id	int	外键	楼栋 id

(4) 房间信息表。该表记录了本社区管理系统中存入的单元信息，该表包括房间 id、房间名称、房间面积、房间是否入住、单元 id 等字段。表格结构如下表 4-5 所示。

表 4-5 房间信息表

字段名	数据类型	约束	备注
id	int	主键	房间 id
name	varchar	非空	房间名称
area	int	非空	房间面积
status	int	非空	房间是否入住
danyuan_id	int	外键	单元 id

（5）社区居民车位信息表。该表记录了本社区管理系统中的居民所对应的车位信息，该表包括车位列表序号、社区居民 id、车位 id、车位生效时间、车位到期时间等字段。表格结构如下表 4-6 所示。

表 4-6 车位信息表

字段名	数据类型	约束	备注
id	int	主键	车位列表序号
user_id	int	外键	社区居民 id
car_id	int	外键	车位 id
in_Time	datetime	非空	车位生效时间
out_Time	datetime	非空	车位到期时间

（6）社区居民缴费信息表。该表记录了本社区管理系统中的居民所对应的缴费信息，该表包括缴费表序号、社区居民 id、缴费类型 id、缴费金额、缴费截止时间、缴费状态等字段。表格结构如下表 4-7 所示。

表 4-7 社区居民缴费信息表

字段名	数据类型	约束	备注
id	int	主键	缴费表序号
user_id	int	外键	社区居民 id
payment_id	int	外键	缴费类型 id
value	varchar	非空	缴费金额
time	datetime	非空	缴费截止时间
status	int	非空	缴费状态

（7）社区公告信息表。该表记录了本社区管理系统中公告信息，该表包括公告表序号、

公告标题、公告详情、公告发布人、公告发布时间、公告修改人、公告修改时间、公告发布状态等字段。表格结构如下表 4-8 所示。

表 4-8 社区公告信息表

字段名	数据类型	约束	备注
id	int	主键	公告表序号
title	varchar	非空	公告标题
content	varchar	非空	公告详情
createBy	int	外键	公告发布人
createTime	datetime	非空	公告发布时间
updateBy	int	外键	公告修改人
updateTime	datetime	非空	公告修改时间
status	int	非空	公告发布状态

(8) 报修信息表。该表记录了本社区管理系统中用户提交的报修信息，该表包括报修表序号、报修详情、报修人、报修状态、报修时间、处理结果等字段。表格结构如下表 4-9 所示。

表 4-9 报修信息表

字段名	数据类型	约束	备注
id	int	主键	报修表序号
content	varchar	非空	报修详情
user_id	int	外键	报修人
status	int	非空	报修状态
time	datetime	非空	报修时间
result	int	非空	处理结果

(9) 投诉信息表。该表记录了本社区管理系统中用户提交的投诉信息，该表包括投诉表序号、投诉详情、投诉人、投诉状态、投诉时间、处理结果等字段。表格结构如下表 4-10 所示。

表 4-10 投诉信息表

字段名	数据类型	约束	备注
id	int	主键	投诉表序号
content	varchar	非空	投诉详情
user_id	int	外键	投诉人
status	int	非空	投诉状态
time	datetime	非空	投诉时间
result	int	非空	处理结果

4.4 本章小结

这一章节主要介绍了社区管理系统的总体架构设计以及各个模块的详细设计。在上一章需求分析的基础上，分别对居民管理、房屋管理、车位管理、缴费管理、社区服务管理、社区居民操作等 6 个模块进行了功能上的设计，确立了各个模块的功能和相互之间的关系。此外，还进行了数据库设计，绘制了系统 E-R 图，列出了数据库表，为后续的开发工作提供了重要的参考依据。

5 社区管理系统的实现与测试

上一章已经完成了对社区管理系统的关键功能模块的设计，接下来将根据对各个功能的设计部分进行编程实现。实现的效果图将以 Web 页面的形式直观展示出来。通过这种方式，用户可以更好地了解系统的整体架构和核心功能。除此之外，本章将对系统的核心功能代码进行阐述，以帮助用户更好地理解系统的实现方式。为了确保系统的质量，本章还会结合现有的资源对系统进行测试。我们将使用各种测试方法来测试系统的各个方面，包括性能、可靠性和安全性。这将确保系统能够在实际使用中表现出良好的性能和稳定性。系统功能的实现分为六个模块：居民管理系统，房屋管理系统，车位管理系统，缴费管理系统，社区服务管理系统，社区居民操作系统，接下来本论题将逐一介绍这些模块的实现及测试部分。

5.1 系统开发环境

本论题所采用的的开发环境都是开源的，包括：Java 集成开发工具 IntelliJ IDEA 2021，它支持多种编程语言和框架，可以提高开发效率和质量；JDK 1.8，作为 Java 平台的一个版本，它包含了 Java 编译器、Java 虚拟机和 Java 类库等组件，可以用于编写和运行 Java 应用程序；Spring Boot，一种基于 Spring 框架的快速开发框架，可以帮助开发人员快速构建 Web 服务和应用程序；Mybatis，作为一种轻量级的 ORM 框架，可以帮助开发人员将 Java 对象映射到数据库表中；Mysql 8.0，作为一种开源的关系型数据库，可以用于存储和管理应用程序的数据；Navicat，作为一款数据库管理工具，可以帮助开发人员设计和管理数据库表；Tomcat9.0，作为一种常见的 Web 服务器，可以用于部署和运行 Java Web 应用程序；Maven，作为一种项目管理工具，可以帮助开发人员管理项目依赖和构建过程。

5.2 系统功能实现

5.2.1 居民管理系统实现

居民管理系统的主要功能是维护系统的用户以及用户信息，本模块只有社区管理员才有权利进行操作。社区管理员的身份信息存入数据库中，登录时进行验证，核心代码如下图 5-1 调用 findAdmin 和图 5-2 登录验证所示。

```
public Admin findAdmin(String email, String password) {  
    List<Admin> list = template.query(sql: "select * from admin where email = ? && password = ?" ,  
        new Object[]{email,password}, new BeanPropertyRowMapper(Admin.class));  
    if (list!=null && list.size()>0){  
        return list.get(0);  
    }else{  
        return null;  
    }  
}
```

图 5-1 调用 findAdmin

```
@PostMapping("/admin/loginByPassword")  
public ResBody loginByPassword(@RequestBody Map<String, Object> params,  
    HttpSession session) {  
    ResBody resBody = new ResBody();  
    String email = params.get("email").toString();  
    String password = params.get("password").toString();  
    Admin admin = service.findAdmin(email,password);  
    if (admin == null){  
        resBody.setCode(500);  
        resBody.setMsg("登录失败，请重新登录");  
    }else {  
        session.setAttribute("admin",admin);  
        LOG.info(admin.toString());  
        resBody.setCode(200);  
        resBody.setMsg("登录成功");  
    }  
    return resBody;  
}
```

图 5-2 登录验证

管理员登录界面如下图 5-3 所示。



图 5-3 管理员登录界面

管理员可以通过点击用户管理菜单进入用户管理首页，然后发送请求到用户管理查询接口来获取用户信息及角色相关信息。查询接口会读取数据库中的用户表，并返回第一页的数据，以 JSON 格式进行返回。前端会将数据以分页列表的形式展示出来。这个系统的好处是管理员可以方便地管理用户信息和角色信息，而且可以通过分页列表方便地查看大量数据。同时，由于数据以 JSON 格式进行返回，前端可以轻松地处理数据并进行展示。

用户管理界面如下图 5-1 所示

用户名: 输入用户名

搜索

+ 添加

刷新

打印

导出

#	用户名	手机号	性别	车位	房屋	操作					
1	包源文	15827481111	<div><div></div>女</div>	暂无	A1单元样板间	<div>停用车位</div>	<div>停用房屋</div>	<div>房屋分配</div>	<div>车位分配</div>	<div>数据管理</div>	<div>修改</div>
2	2	15827481112	<div><div></div>男</div>	暂无	暂无	<div>停用车位</div>	<div>停用房屋</div>	<div>房屋分配</div>	<div>车位分配</div>	<div>数据管理</div>	<div>修改</div>
3	周杰伦	15827481113	<div><div></div>男</div>	暂无	暂无	<div>停用车位</div>	<div>停用房屋</div>	<div>房屋分配</div>	<div>车位分配</div>	<div>数据管理</div>	<div>修改</div>
4	4	15827481114	<div><div></div>男</div>	暂无	暂无	<div>停用车位</div>	<div>停用房屋</div>	<div>房屋分配</div>	<div>车位分配</div>	<div>数据管理</div>	<div>修改</div>
5	5	15827481115	<div><div></div>男</div>	暂无	暂无	<div>停用车位</div>	<div>停用房屋</div>	<div>房屋分配</div>	<div>车位分配</div>	<div>数据管理</div>	<div>修改</div>
6	6	15827481116	<div><div></div>男</div>	暂无	暂无	<div>停用车位</div>	<div>停用房屋</div>	<div>房屋分配</div>	<div>车位分配</div>	<div>数据管理</div>	<div>修改</div>
7	7	15827481117	<div><div></div>男</div>	暂无	暂无	<div>停用车位</div>	<div>停用房屋</div>	<div>房屋分配</div>	<div>车位分配</div>	<div>数据管理</div>	<div>修改</div>
8	包包	15827484086	<div><div></div>男</div>	暂无	暂无	<div>停用车位</div>	<div>停用房屋</div>	<div>房屋分配</div>	<div>车位分配</div>	<div>数据管理</div>	<div>修改</div>

< 1 >

到第 1 页

确定

共 8 条

10 条/页

图 5-4 用户管理界面图

用户管理系统提供了比较完善的用户管理功能，包括基本信息展示、分页列表、搜索、新增、编辑和删除等功能，可以帮助管理员方便地管理用户信息和角色信息，提高管理效

率。

除了基本的用户信息展示和分页列表功能外，还提供了搜索用户和新增用户的功能。在每个用户信息的右侧还提供了编辑和删除的操作标签。如果管理员点击删除按钮，系统会弹出一个提示页面，要求用户确认是否删除此用户。如果用户确认删除，系统会将该用户的使用状态改为不可用，实现逻辑上的删除。这种实现方式可以避免直接从数据库中删除数据，从而保留了数据的完整性和可追溯性。同时，系统还提供了编辑用户的社区相关信息的功能，这可以方便管理员对用户信息进行修改和更新。

5.2.2 房屋管理系统实现

房屋管理类包含控制类、服务组合类、数据持久化类和实体类，用于协调和控制系统中的各个组件，提供各种服务，存储和管理数据。房屋管理控制类是 `BuildingController`、`DanyuanController`、`RoomController` 和 `User_RoomController`。`BuildingController`、`DanyuanController`、`RoomController` 类用来接收社区管理员用户对房屋信息管理的请求并作出响应，`User_RoomController` 类用于接受社区管理员用户的请求，处理社区居民的房屋相关信息，并与数据持久化类和实体类交互以管理房屋信息。而服务类包括 `BuildingService`、`DanyuanService`、`RoomService` 和 `User_RoomService`，`BuildingService`、`DanyuanService`、`RoomService` 类用来处理房屋信息管理相关操作逻辑，`User_RoomService` 用来处理社区管理员处理社区居民的房屋相关信息的相关逻辑。房屋管理数据持久化类包括 `BuildingDao`、`DanyuanDao`、`RoomDao` 和 `User_RoomDao`，`BuildingDao`、`DanyuanDao`、`RoomDao` 用来提供房屋信息管理事项相关数据持久化能力，`User_RoomDao` 用来提供社区管理员处理社区居民的房屋相关信息相关数据持久化能力。房屋管理实体类包括 `Building`、`Danyuan`、`Room` 和 `User_Room`，`Building`、`Danyuan`、`Room` 是房屋信息管理实体类，包括楼栋名称、楼栋类型、单元、房屋所属单元、房屋面积、等属性，`User_Room` 是社区管理员处理社区居民的房屋相关信息实体类，包括新建分配房屋所有人、设置社区居民入住时间和房屋到期时间等属性。

房屋管理类图如下图 5-5 所示。

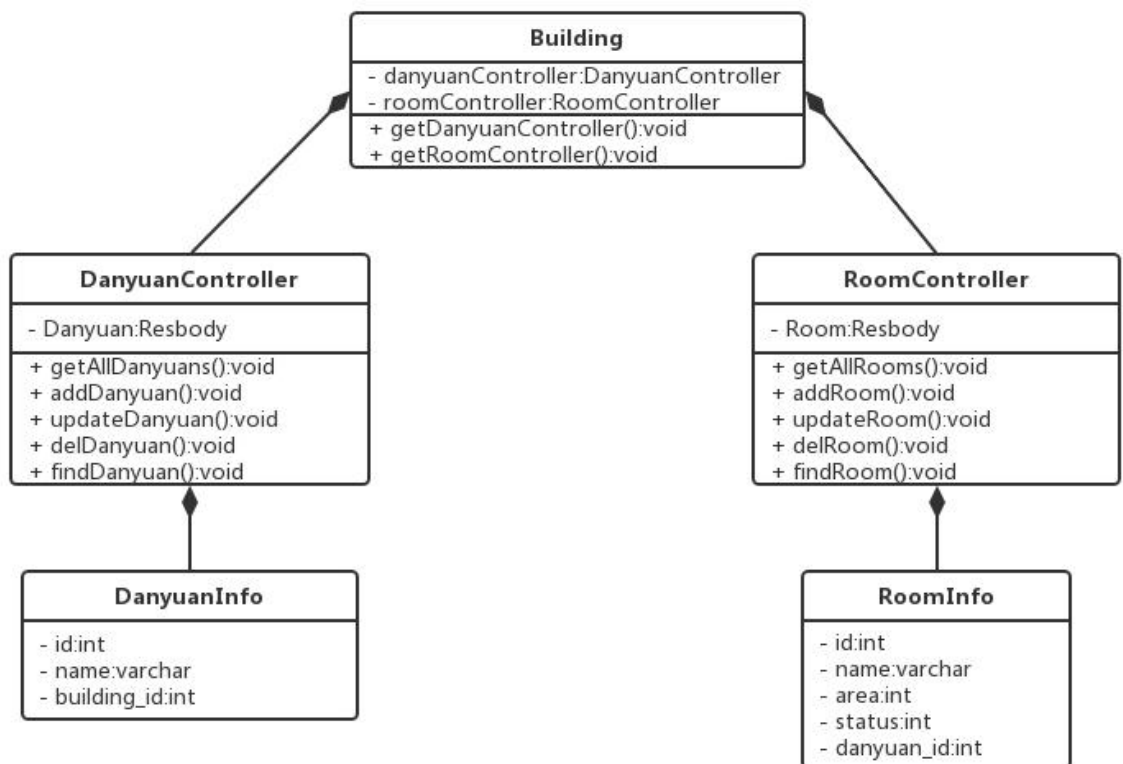


图 5-5 房屋管理类图

社区管理员通过房屋信息管理系统添加房屋信息，包括楼栋、单元和房屋等信息。系统会在管理员填写信息后进行检查，确保信息无误后将其保存到数据库中。最后，系统会返回保存结果并提示管理员添加成功。房屋管理业务界面如下图 5-6 所示。

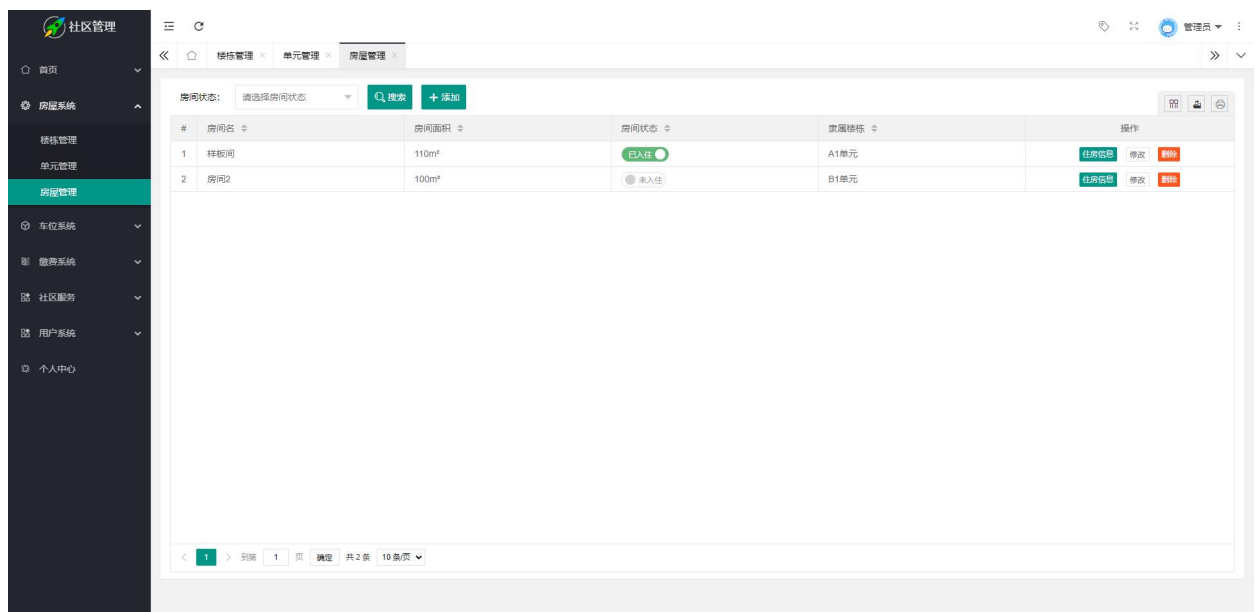


图 5-6 房屋管理业务界面

本节实现了房屋信息管理系统中四个主要功能：房屋信息增加、删除、修改和查询。

通过这些功能，社区管理员可以方便地进行房屋信息管理和维护，并且保证信息的准确性和可靠性。在增加功能中，管理员可以新建楼栋、单元和房屋，并且可以查看本环节所有房屋信息；在删除功能中，管理员可以单个或批量删除房屋信息；在修改功能中，管理员可以修改楼栋、单元和房屋信息，并且修改成功后数据库进行更新；在查询功能中，管理员可以查看本系统内所有房屋信息。这些功能的实现提高了房屋信息管理的效率和便捷性，同时也提高了信息的准确性和可靠性。

5.2.3 车位管理系统实现

车位管理类包含控制类、服务组合类、数据持久化类和实体类，用于协调和控制系统中的各个组件，提供各种服务，存储和管理数据。车辆管理控制类是 `CarController` 和 `User_CarController`。`CarController` 类用来接收社区管理员用户对车位信息管理的请求并作出响应，`User_CarController` 类用于接受社区管理员的请求，处理社区车位相关信息，并与数据持久化类和实体类交互以管理车位信息。而服务类包括 `CarService` 和 `User_CarService`，`CarService` 类用来处理车位信息管理相关操作逻辑，`User_CarService` 用来处理社区管理员处理社区居民的车位相关信息的相关逻辑。车位管理数据持久化类包括 `CarDao` 和 `User_CarDao`，`CarDao` 用来提供车位信息事项相关数据持久化能力，`User_CarDao` 用来提供社区管理员处理社区居民的车位相关信息相关数据持久化能力。车位管理实体类包括 `Car` 和 `User_Car`，`Car` 是车位信息管理实体类，包括车位序号、车位名称、车位类型、车位状态，`User_Car` 是社区管理员处理社区居民的车位相关信息实体类，包括新建分配车位所有人、设置车位起始时间和车位到期时间等属性。

车位管理类图如下图 5-7 所示。

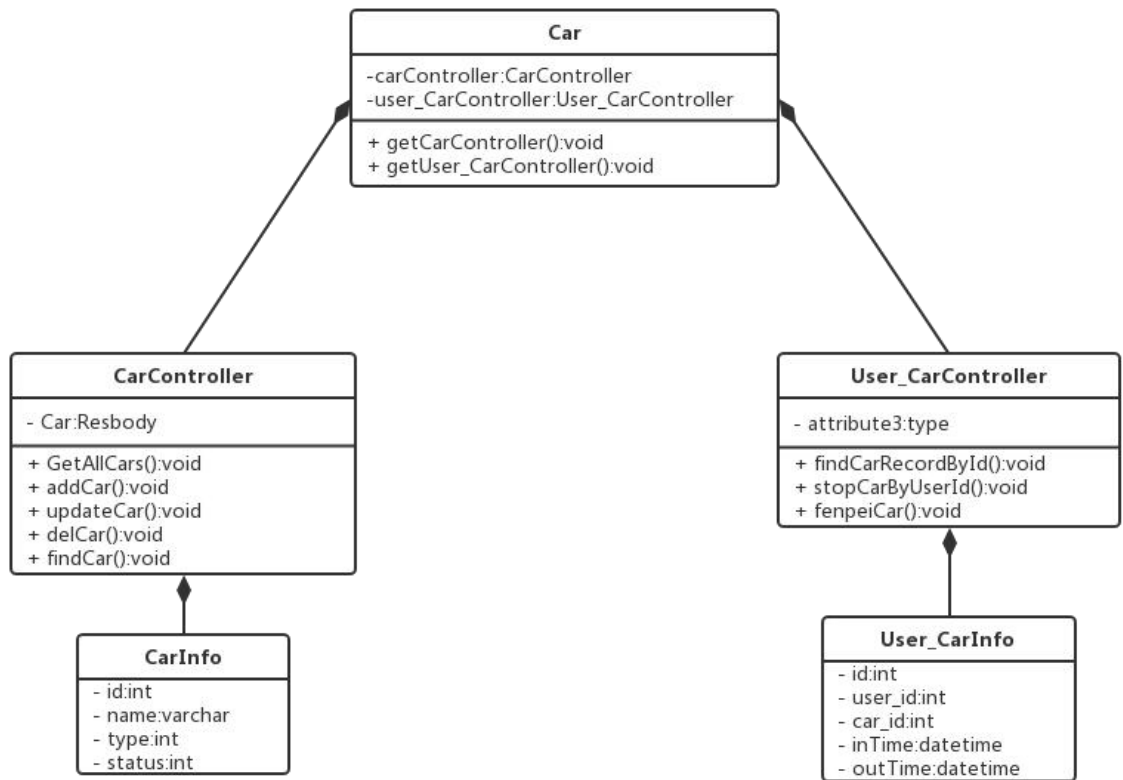


图 5-7 车位管理类图

社区管理员请求访问车位信息管理系统页面，然后请求添加车位。系统响应并要求管理员填写车位具体信息。管理员在操作界面填写信息，系统反馈无误。管理员提交保存信息请求，车位数据被录入系统并存储到数据库中。数据库完成操作并返回保存结果，管理员操作界面显示添加成功。这个流程可以帮助管理员有效地管理车位信息。在这个流程中，管理员首先需要访问车位信息管理系统，并且需要有相应的权限才能进行相关操作。添加车位需要填写车位的具体信息，例如车位编号、位置、大小等。系统需要对这些信息进行验证，以确保填写的信息无误。在管理员提交保存信息请求之后，系统会将车位信息存储到数据库中，以便以后进行查询和管理。如果管理员填写的信息有误，系统会提示管理员进行更正。如果操作成功，系统会返回添加成功的信息。这个流程需要确保管理员填写的信息准确无误，以便更好地管理车位信息。车位管理业务界面如下图 5-8 所示。

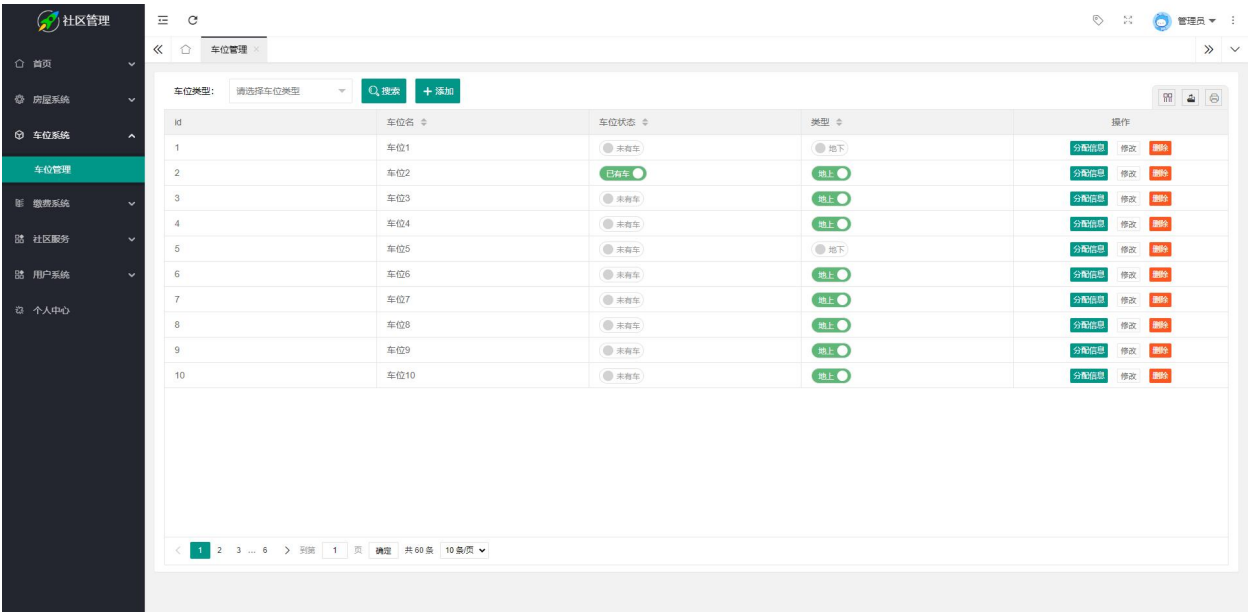


图 5-8 车位管理业务界面

本节实现了车位管理系统中四个主要功能：车位增加、删除、修改和查询。通过这些功能，社区管理员可以方便地进行车位管理和维护，并且保证信息的准确性和可靠性。

5.2.4 缴费管理系统实现

缴费管理类包含控制类、服务组合类、数据持久化类和实体类，用于协调和控制系统中的各个组件，提供各种服务，存储和管理数据。缴费管理控制类是 `PaymentController` 和 `User_PaymentController`。`PaymentController` 类用来接收社区管理员用户对缴费信息管理的请求并作出响应，`User_PaymentController` 类用于接受社区管理员的请求，处理缴费相关信息，并与数据持久化类和实体类交互以管理缴费信息。而服务类包括 `PaymentService` 和 `User_PaymentService`，`PaymentService` 类用来处理缴费信息管理相关操作逻辑，`User_PaymentService` 用来处理社区管理员处理社区居民的缴费相关信息的相关逻辑。缴费管理数据持久化类包括 `PaymentDao` 和 `User_PaymentDao`，`PaymentDao` 用来提供缴费信息管理事项相关数据持久化能力，`User_PaymentDao` 用来提供社区管理员处理社区居民的缴费相关信息相关数据持久化能力。缴费管理实体类包括 `Payment` 和 `User_Payment`，`Payment` 是缴费信息管理实体类，包括缴费序号，缴费名称，`User_Payment` 是社区管理员处理社区居民的缴费相关信息实体类，包括设置用户的缴费的序号、缴费名称、居民名、欠费用户名称、应缴费用、缴费截止时间、缴费状态等信息等属性。

缴费管理类图如下图 5-9 所示。

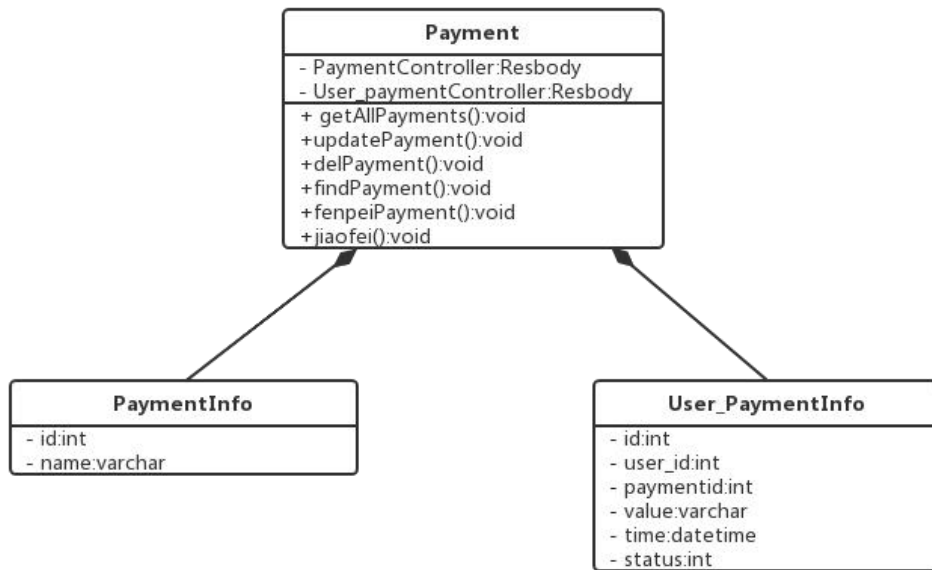


图 5-9 缴费管理类图

管理员对缴费管理系统进行操作的流程如下，管理员首先请求访问缴费信息管理页面，然后请求添加缴费类型。系统响应并要求管理员填写缴费类型的具体信息。管理员在操作界面填写信息，系统反馈无误。管理员提交保存信息请求，缴费类型被录入系统并存储到数据库中。数据库完成操作并返回保存结果，管理员操作界面显示添加成功。这个流程需要确保管理员填写的信息准确无误，以便更好地管理缴费信息。在这个流程中，管理员需要确保填写的缴费类型信息是准确的，例如缴费类型名称、费用、缴费周期等。系统需要对这些信息进行验证，以确保填写的信息无误。在管理员提交保存信息请求之后，系统会将缴费类型信息存储到数据库中，以便以后进行查询和管理。如果管理员填写的信息有误，系统会提示管理员进行更正。如果操作成功，系统会返回添加成功的信息。这个流程需要确保管理员填写的信息准确无误，以便更好地管理缴费信息。缴费管理业务界面如下图 5-10 所示。

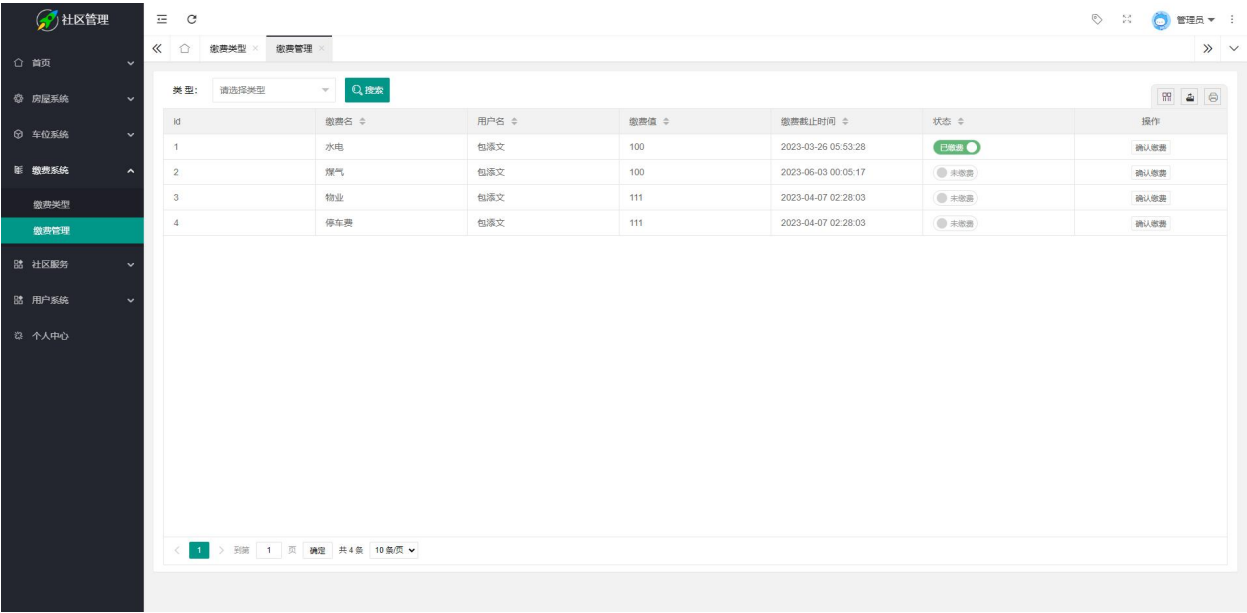


图 5-10 缴费管理业务界面

缴费确认功能用于确认社区居民的缴费行为，如果确认成功，则会更新数据库，并在社区管理员界面上显示修改后的内容。缴费信息管理查询功能用于查看本系统内所有缴费信息。

5.2.5 社区服务管理系统实现

社区服务管理类包含控制类、服务组合类、数据持久化类和实体类，用于协调和控制系统中的各个组件，提供各种服务，存储和管理数据。社区服务管理控制类是 GonggaoController、TousuController、RepairController，用来接收社区管理员用户对社区服务信息管理的请求并作出响应。社区服务管理服务类包括 GonggaoService、TousuService、RepairService，用来处理社区服务信息管理相关操作逻辑。社区服务管理数据持久化类包括 GonggaoDao、TousuDao、RepairDao，用来提供社区服务信息事项相关数据持久化能力。社区服务管理实体类包括 Gonggao、Tousu、Repair，是社区服务信息管理实体类，包括公告名称、公告详情、公告发布人、公告修改人、公告发布时间、公告修改时间、投诉维修具体内容、提交时间、处理结果等属性。社区服务类图如下图 5-11 所示。

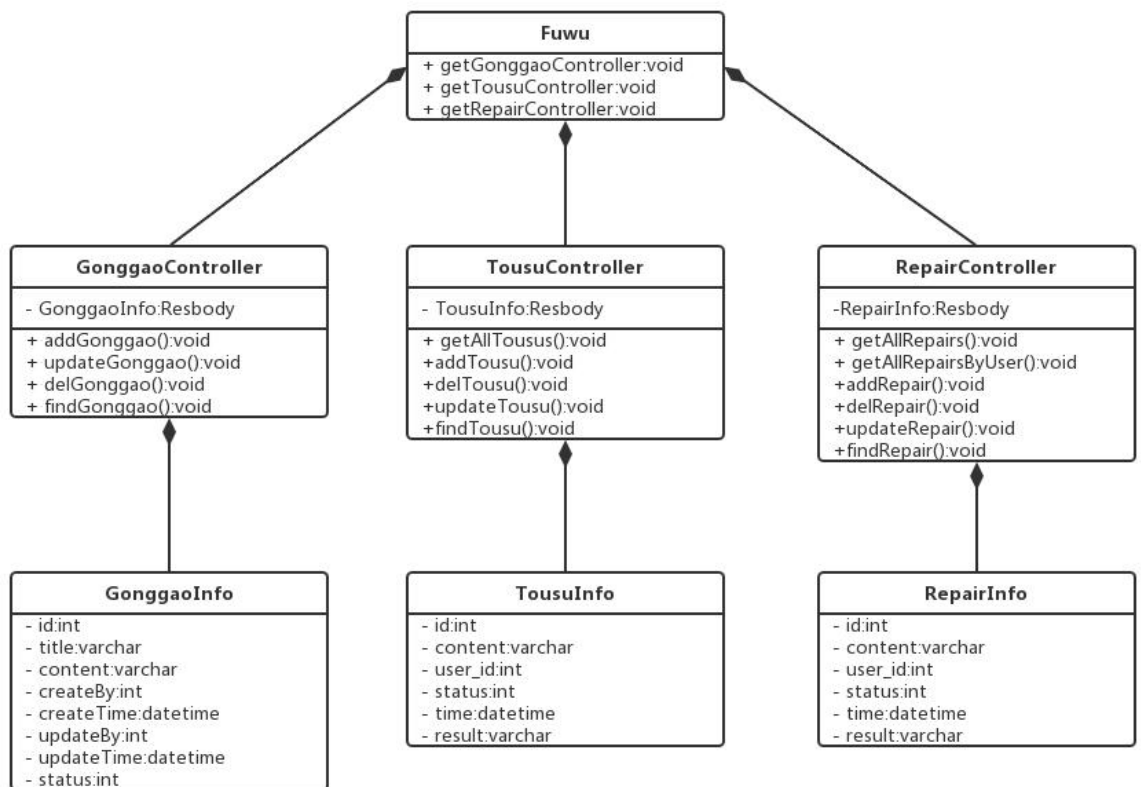


图 5-11 社区服务类图

5.2.6 社区居民操作系统实现

社区居民操作界面如下图 5-12 所示

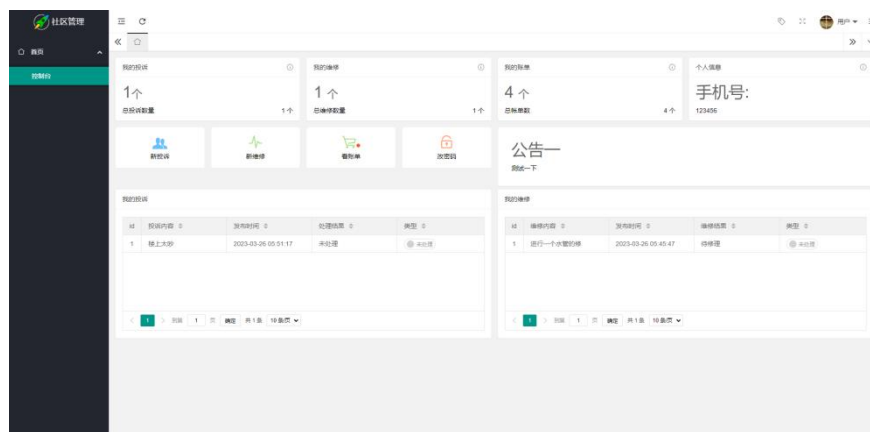


图 5-12 社区居民操作界面

5.3 系统测试

系统功能测试主要是模拟用户对系统的使用，以确保软件系统能够正常运行，并满足用户的需求和期望。下面将对已经实现的 6 个模块分别进行功能测试。

本节采用黑盒测试方法进行测试。黑盒测试是一种基于功能需求和规格说明书的测试方式，不需要了解程序的内部结构和实现细节。测试人员只需要将输入数据传入被测试的软件系统中，然后观察输出结果是否符合预期即可。黑盒测试可以大大提高测试效率。

接下来主要针对部分重要功能进行测试主要包括管理员登录功能、居民管理功能、房屋管理功能、车位管理功能、缴费管理功能、公告管理功能、投诉管理功能、用户操作功能等。

（1）管理员登录测试用例

测试模块：登录模块

测试用例名称：管理员身份登录

前置条件：数据库中已存入至少一条管理员账号数据

用例步骤：

- 1) 打开管理员登录界面，输入账号、密码。
- 2) 拖动验证滑块。
- 3) 账号密码输入错误信息后，点击登录按钮。
- 4) 账号密码输入正确信息后，点击登录按钮。

预期结果：

若用户输入的账号/密码不符合格式要求、或者输入了错误的账号/密码，则登录失败，并给出准确的提示，如“账号格式错误”或者“账号/密码错误”；输入信息账号密码后，成功登录到管理员系统中。

实际结果：测试通过

（2）删除房屋信息测试用例

测试模块：房屋管理模块

测试用例名称：删除房屋信息

前置条件：数据库中已存入至少一条房屋数据

用例步骤：

- 1) 打开房屋管理界面。
- 2) 展示房屋列表界面。
- 3) 单击列表中一个房屋
- 4) 展示房屋编辑界面

5) 单击“删除”按钮后，点击保存按钮。

预期结果：

若列表中并未存入房屋数据，则不能进行删除操作；若操作正确，该项数据将从数据库中删除。

实际结果：测试通过

(3) 修改车位信息测试用例

测试模块：车位管理模块

测试用例名称：修改车位信息

前置条件：社区管理系统数据库汇总已存在至少一条车位数据

用例步骤：

1) 进入车位管理系统界面，点击“查看车位信息”按钮。

2) 点击“更新车位信息”按钮。

3) 展示更新车位信息的界面。

4) 在车位界面提示的文本框中输入错误信息（如车位所属人输入“abc”、车位地址输入为“123”、车位使用时间输入“abc”等）后，点击保存按钮。

5) 每个文本框输入正确信息（如车位所属人输入用户 id、车位起始使用时间输入“2023-4-20”）后，点击保存按钮。

预期结果：

若管理员在输入车位所属人时，输入了“abc”、输入车位位置时为空、输入车位类型时输入“地上”，则在点击“保存”时会收到提示，“无法保存，请按正确格式输入”；若管理员输入格式正确的数据，数据则会被正确存入系统，可在数据库中查看到此项数据。

实际结果：测试通过

(4) 居民缴费管理测试用例

测试模块：缴费管理模块

测试用例名称：增加缴费项并

前置条件：无

用例步骤：

1) 进入缴费管理界面，点击“增加缴费项目”按钮。

2) 展示所有缴费项目界面。

3) 缴费项目输入内容为空后，点击保存按钮。

4) 输入符合系统要求的缴费项目信息后，点击保存按钮。

预期结果：

若用户输入的新的缴费项目内容为空或者长度超出了文本框的范围，则提示“输入不可为空”或者“输入内容过长，请重新输入”；输入了恰当的缴费项目内容后，数据正确更新并存入系统，可在数据库中查看到此项数据。

实际结果：测试通过

(5) 社区服务测试用例

测试模块：居民投诉模块

测试用例名称：居民进行社区投诉测试

前置条件：社区管理系统数据库中已存在至少一名居民账号数据

用例步骤：

1) 居民登录系统，点击“投诉”按钮。

2) 展示居民投诉的界面。

3) 每个文本框输入错误信息（如投诉内容输入为空）后，点击投诉按钮。

4) 每个文本框输入正确信息后，点击投诉按钮。

预期结果：

若用户在投诉文本框中输入了超出文本框容量的文字，或者并未在该文本框中输入任何文字，则在用户点击“投诉”按钮时会收到“投诉内容过长”或是“投诉内容不可为空”的提示；若用户在投诉模块的标题文本框和具体内容文本框按照系统要求输入了投诉信息后，点击“投诉”按钮，则会提示投诉成功，请等待社区工作人员反馈。数据则会正确存入到系统中，社区管理员可以在投诉管理界面中查看到此条投诉。

实际结果：测试通过。

5.4 本章小结

这一章主要介绍了社区管理系统的各个功能的实现和测试。本章首先详细介绍了系统的开发环境和使用对应开发环境的理由。接着在上一章对社区管理系统设计的基础上对各个功能模块进行编码实现。并展示了部分关键代码和系统实现后的界面图。最后为了确保系统能够顺利运行，编写了测试用例，对系统进行了黑盒测试。

6 总结与展望

本文介绍了基于 Springboot 框架的社区管理系统的设计与实现。首先在第一章介绍了社区管理系统在国内外的研究现状并根据我国当前现状分析了该论题的研究意义。在第二章介绍了本论题使用到开发技术,如 Springboot 框架、vue.js 框架、MySQL 数据库和 Mybatis 框架等等。在第三章对社区管理系统进行了整体需求分析、用例分析、功能性需求分析和非功能需求分析等工作。在社区管理系统的设计中,实现了系统总体架构的设计、系统功能设计和数据库设计。在系统实现与测试中,我们介绍了本论题开发过程中采用的开发环境、在设计的基础上进行了系统功能的实现。并编写了测试用例对系统进行了黑盒测试,验证了系统的正确性和可靠性。

总的来说,本文设计与实现了一个基于 Springboot 框架的社区管理系统。该系统具有良好的用户体验和高效的数据管理能力。通过本文的研究,我们发现 Springboot 框架具有很好的可扩展性和易用性,能够帮助开发者更快速地构建高效的 Web 应用程序。我们也发现,使用 vue.js 框架能够使前端开发更加简单和高效。

未来,该论题可以进一步优化该系统,例如增加更多的功能和改进用户体验。同时,也可以将该系统应用到实际的社区管理中,以提高社区管理的效率和质量。除此之外也可以探索其他开发框架和技术,以进一步提高系统的性能和可靠性。

总之,该论题的研究为基于 Springboot 框架的社区管理系统的设计与实现提供了有益的经验 and 启示。笔者相信,该系统能够为社区管理者和用户带来更好的体验和服务。

参考文献

- [1] 夏婉,李全,张晶晶.疫情下智慧社区管理系统构建[J].信息技术与信息化,2020(08):67-70.
- [2] 以“融治理”理念引领社区治理现代化[J].中国民政,2022(19):29-30.
- [3] 马建峰,徐启建.基于物联网的数字社区关键技术及应用[M].人民邮电出版社:, 2017:10.249.
- [4] 霍俊玲.关于智慧社区商圈经济发展的对策研究[J].现代商业,2021(06):9-11.DOI:10.14097/j.cnki.5392/2021.06.003.
- [5] 任兵,陈志霞,张茂茂.迈向数智时代的城市元宇宙:概念界定与框架构建[J/OL].电子政务:1-12[2023-05-14].<http://kns.cnki.net/kcms/detail/11.5181.TP.20230210.0906.002.html>
- [6] 查皓骞.智慧社区建设状况研究[D].河北大学,2021.DOI:10.27103/d.cnki.ghebu.2021.000669.
- [7] 李景雷.我国城市社区数字化管理问题研究[D].新疆大学,2018.
- [8] 张杰,徐永健.智能化小区设备优化建设与应用[J].电子测试,2022,36(18):116-118.DOI:10.16520/j.cnki.1000-8519.2022.18.035.
- [9] 徐亮.基于物联网技术应用的智慧社区应用系统实现[D].复旦大学,2013.
- [10] 李亚萍.我国智慧社区的研究现状与前沿热点——基于 CiteSpace 的文献计量可视化分析[J].互联网周刊,2023(05):50-53.
- [11] 刘萍,范蕤.智慧社区管理平台构建研究[J].电脑知识与技术,2022,18(21):54-56.DOI:10.14004/j.cnki.ckt.2022.1252.
- [12] 杨雪妍,冯丹娃.智慧社区信息化建设研究热点及发展趋势探析[J].情报科学,2021,39(12):187-193.DOI:10.13833/j.issn.1007-7634.2021.12.027.
- [13] 熊永平.基于 SpringBoot 框架应用开发技术的分析与研究[J].电脑知识与技术,2019,15(36):76-77.DOI:10.14004/j.cnki.ckt.2019.4290.
- [14] 肖铭轩.基于元数据的软件框架研究及优化[D].西南交通大学,2017.
- [15] 谭明亮,代玲林.基于微信小程序的医院医疗设备管理系统设计与实现[J].价值工程,2023,42(03):137-140.
- [16] 赵哲.监狱监督管理系统的设计与实现[D].北京交通大学,2019.
- [17] 王强.智能化社区的管理思路[J].企业改革与管理,2014(05):13.DOI:10.13768/j.cnki.cn11-3793/f.2014.05.179.
- [18] 刘敏.基于 SpringBoot 框架社交网络平台的设计与实现[D].湖南大学,2018.
- [19] 王开阳.基于微服务的智慧保险平台的设计与实现[D].北京交通大

学,2022.DOI:10.26944/d.cnki.gbfju.2022.000792.

[20] 黄志伟. 基于 SSH 框架的智能社区信息管理系统设计与实现[D].厦门大学,2019.

[21] 张伟. 基于 SpringBoot 和 Vue 的综合教学管理平台设计与实现 [D]. 重庆大学,2021.DOI:10.27670/d.cnki.gcqdu.2021.003900.

[22] 刘启伟. 基于 Vue.js 框架的 Web 前端开发工具的设计与实现 [D]. 北京邮电大学,2021.DOI:10.26969/d.cnki.gbydu.2021.002714.

[23] 周园园. 基于 Vue.js 和 JavaScript 技术设计与实现的多功能 Web 计算器 [J]. 科技资讯,2022,20(12):10-13.DOI:10.16661/j.cnki.1672-3791.2203-5042-2320.

[24] 刘勇芳. 基于文本相似度的版权保护系统的设计与实现[D].北京交通大学,2019.

[25] 毛炎. 基于 Vue.js 框架的 Web 方言地图的设计与开发[D].武汉大学,2018.

[26] 唐汉明,翟振兴,关宝军,王洪权,黄潇. 深入浅出 MySQL[M].人民邮电出版社:, 201401.655.

[27] 张彬彬. 基于数据库优化的实验室信息管理系统设计与实现[D].北京邮电大学,2017.

[28] 李学龙,龚海刚.大数据系统综述[J].中国科学:信息科学,2015,45(01):1-44.

[29] 刘智慧,张泉灵.大数据技术研究综述[J].浙江大学学报(工学版),2014,48(06):957-972.

[30] 魏钢.Web 前后端分离模式下 Spring MVC 在高职 Java Web 教学中的研究 [J]. 福建电脑,2019,35(02):107-108.DOI:10.16707/j.cnki.fjpc.2019.02.035.

[31] 徐雯,高建华. 基于 Spring MVC 及 MyBatis 的 Web 应用框架研究 [J]. 微型电脑应用,2012,28(07):1-4+10.

[32] 杨文韬. 基于 SSH 框架的智能社区信息管理系统的设计与实现[D].中山大学,2013.

[33] 霍福华,韩慧.基于 SpringBoot 微服务架构下前后端分离的 MVVM 模型[J].电子技术与软件工程,2022(01):73-76.

[34] 荣艳冬.关于 Mybatis 持久层框架的应用研究[J].信息安全与技术,2015,6(12):86-88.

[35] 赵瑞,黄思怡,韩朝桂,陈秋婷,邹明珠.新冠疫情下社区人员管理系统的设计与实现[J].现代信息科技,2022,6(24):14-17+22.DOI:10.19850/j.cnki.2096-4706.2022.24.004.

[36] 吴越菲.技术如何更智慧:农村发展中的数字乡村性与智慧乡村建设[J].理论与改革,2022(05):94-108+150.DOI:10.13553/j.cnki.llygg.2022.05.009.

[37] 刘少杰. 中国网络社会的交往实践和发展逻辑 [J]. 学术月刊,2022,54(08):105-117.DOI:10.19862/j.cnki.xsyk.000468.

[38] 朱先平. 社区居家养老服务的嵌入性情境及困境研究 [D]. 吉林大

学,2022.DOI:10.27162/d.cnki.gjlin.2022.000097.

[39] 曹 春 节 . 创 业 孵 化 社 区 管 理 系 统 的 设 计 与 实 现 [D]. 华 中 科 技 大 学,2020.DOI:10.27157/d.cnki.ghzku.2020.006669.

[40] 魏 婷. 社区管理系统的设计与实现[D].哈尔滨工业大学,2016.

[41] 柴彦威,郭文伯.中国城市社区管理与服务的智慧化路径[J].地理科学进展,2015,34(04):466-472.

[42] 陈夫真. 基于 SSM 的某高校教室管理信息系统的设计与实现[D].苏州大学,2012.

[43] 蒋朝惠,许石青.数字社区管理与服务信息系统的设计与实现[J].计算机工程,2005(24):228-230.