



廣東科技學院
GUANGDONG UNIVERSITY OF SCIENCE & TECHNOLOGY

葉選平

毕业设计

题目：光明社区物业管理系统的
设计与实现

姓 名：	陈华锋
学 号：	2019131838
层 次：	本科
专 业：	软件工程
班 级：	19 软件本科 8 班
指导教师：	蒋明奇

2023 年 5 月

广东科技学院教务处 制

毕业论文（设计）独创性声明

本人郑重承诺：所呈交的毕业论文（设计）是本人在指导教师的指导下，独立研究所取得的成果。本人在撰写毕业论文（设计）的过程中严格遵守学术规范，尽我所知，除致谢部分外凡文中引用的他人发表或未发表的学术成果均已注明出处，除引用部分外文中所有的数据及其他引证材料均真实客观。如有学术失范行为，本人愿意承担由此造成的一切后果。

论文（设计）作者签名：陈华锋

日期：2023 年 5 月 8 日

摘要 社区物业管理系统是物业管理发展的整体趋势，是物业管理的一种新发展，是现在社会形势下社区管理创新的一种新方式。随着社区物业管理的发展需要，物业管理将会与人们的联系越来越密切，光明社区物业管理系统在带给人们方便快捷的同时也适应了当今社会的发展需求。社区物业管理系统不仅能提高光明社区物业管理工作效率，而且还方便了业主们对物业了解社区的相关信息。

光明社区物业管理系统是一个基于 SpringBoot 框架的系统，结合 LayUI 框架、ECharts 图表、MySQL 数据库进行技术设计和开发。使用了 LayUI 框架来对系统前端页面进行快速编写；使用 ECharts 图表对各种的报修数据进行统计；MySQL 数据库对系统的数据进行储存。

光明社区物业系统设计由十个模块组成，分别是车位收费管理模块、物业收费管理模块、投诉信息管理模块、报修信息管理模块、房屋管理模块、业主管理模块、车位管理模块、报修统计模块、用户模块、访问拦截与权限控制模块。由此可以方便用户进行操作和管理员对信息进行管理。

关键词 管理系统；SpringBoot；物业

Abstract Community property management system is the overall trend of property management development,a new development of property management,and a new way of community management innovation under the current social situation. With the development needs of community property management,property management will be more and more closely connected with people,Guangming community property management system in the convenience and speed but also adapt to the development needs of today's society. The community property management system can not only improve the work efficiency of the property management personnel of Guangming Community,but also facilitate the owners to understand the relevant information of the community on the property.

Bright Community Property Management System is a system based on the SpringBoot framework,combined with LayUI framework,ECharts charts,Mysql database for technical design and development. The LayUI framework is used to quickly write the front-end pages of the system; Use ECharts charts to statistically report various repair data; The MySQL database stores the data of the system.

The design of Guangming Community property system consists of ten modules,namely parking space charge management module,property charge management module,complaint information management module,repair information management module,housing management module,owner management module,parking space management module,repair statistics module,user module,access interception and permission control module. This makes it easier for users to operate and administrators to manage information.

Keywords Manage the system; SpringBoot; Property

目 录

1 绪论	1
1.1 研究背景及意义	1
1.2 国内外研究现状	1
1.2.1 国内研究现状	1
1.2.2 国外研究现状	2
1.3 本章小结	2
2 相关技术	3
2.1 SpringBoot 框架	3
2.2 LayUI 框架	3
2.3 MySQL 数据库	3
2.4 ECharts 图表	3
2.5 本节小结	3
3 系统可行性和需求分析	4
3.1 系统功能需求分析	4
3.1.1 功能分析	4
3.1.2 系统用例分析	4
3.2 系统非功能需求分析	6
3.2.1 运行需求分析	6
3.2.2 性能需求分析	6
3.2.3 界面需求分析	6
3.2.4 其他需求分析	6
3.3 系统可行性分析	6
3.3.1 经济可行性	6
3.3.2 技术可行性	7
3.4 本章小结	7
4 系统设计	8
4.1 系统总体设计	8
4.1.1 需求规定	8
4.1.2 运行环境	9
4.1.3 处理过程	10

4.2 功能模块图	11
4.3 本章小结	12
5 数据库设计	13
5.1 图设计	13
5.1.1 总体 E-R 图	13
5.1.2 车位收费实体属性图	13
5.1.3 投诉实体属性图	13
5.1.4 投诉类型实体属性图	14
5.1.5 房屋实体属性图	14
5.1.6 业主实体属性图	15
5.1.7 车位实体属性图	15
5.1.8 报修实体属性图	15
5.1.9 报修类型实体属性图	16
5.1.10 用户实体属性图	16
5.1.11 收费类型实体属性图	16
5.1.12 物业收费实体属性图	17
5.2 表设计	17
5.3 功能模块与表的关系	20
5.4 本章小结	21
6 系统实现	22
6.1 车位收费模块	22
6.2 物业收费模块	23
6.3 投诉信息模块	24
6.4 报修信息模块	25
6.5 房屋模块	27
6.6 业主信息模块	28
6.7 车位信息模块	29
6.8 用户信息模块	31
6.9 数据统计模块	32
6.10 访问拦截与权限控制模块	33
6.11 本章小结	33

7 系统测试.....34

7.1 测试概述 34

7.2 系统测试用例 34

7.2.1 访问拦截与权限控制模块 34

7.2.2 车位收费模块 34

7.2.3 物业收费模块 35

7.2.4 投诉信息模块 35

7.2.5 报修信息模块 36

7.2.6 房屋信息模块 36

7.2.7 业主信息模块 37

7.2.8 车位信息模块 37

7.2.9 用户信息模块 38

7.2.10 数据统计模块 38

7.3 测试结论 39

7.4 小结 39

8 总结和展望.....40

8.1 总结 40

8.2 展望 40

参考文献41

1 绪论

1.1 研究背景及意义

社区物业管理系统是物业管理发展的整体趋势，是物业管理的一种新发展，是现在社会形势下社区管理创新的一种新方式，物业管理在人居生活中的作用也变得越来越重要^[1]。随着人工智能、大数据、物联网等高新技术的高速发展，给物业管理带来了巨大的挑战和契机，并且在当今严峻的疫情防控工作中，更可以体现出光明社区物业管理的重要性，可以尽最大程度保证小区的安全性，物业管理企业也先后试水互联网。如今互联网快速发展和计算机的普及，社区物业管理使用计算机进行管理成为可能，要积极开展社会建设^[2]。随着社区物业管理的发展需要，物业管理将会与人们的联系越来越密切，光明社区物业管理系统在带给人们方便快捷的同时也适应了当今社会的发展需求，管理的服务与业主息息相关，会直接影响到业主的归属感和幸福感等^[3]。社区物业管理系统不仅能提高光明社区物业管理人员工作效率，而且还方便了业主们对物业了解社区的相关信息。物业管理系统服务以业主为中心，为业主提供优质的服务，追求服务至上的态度，是可以满足业主对服务的质量和物业的信息管理需求的^[4]。

1.2 国内外研究现状

1.2.1 国内研究现状

随着我国科技技术发展和城市开发，物业管理已形成一定范围的规模，其效益也显现出来了。但是，目前我国有相当一部分物业管理公司还停留在人工管理的基础上，这样的物业管理机制已经不能适应当今时代的发展，其管理方法存在着不少的缺点，如：效率低且较为繁琐，另外随着物业管理项目的增加，工作量也将大幅度的增加，这必然增加物业管理人员工作量和劳动强度，这将给物业管理信息带来很大的困难^[5]。社区中存在的问题管理人员未能及时发现时，业主可以提出反馈，从而让管理人员及时进行处理。随着科学技术的不停的提高，这种传统的人工手工管理方法肯定被现代的信息管理方法所取代^[6]。由于手工管理信息各种变革因素的影响，手工的管理信息系统必定向现代管理信息系统发展。

我国虽然在物业管理系统的技术方面有了较大的突破和进步，但是各个方面仍然有所欠缺，不够完善，因此在整体的来看还是处于一个摸索、探索的处境。如何让国内的技术和国外的技术相结合，如何可以给我国物业管理方面带来便捷^[7]，如何给

业主在物业方面提供优质的服务，这些已经成为我国现在信息化物业管理系统建设的难题。

1.2.2 国外研究现状

目前，在发达国家的人数较少，而物业管理在 19 世纪 60 年代在英国兴起，物业管理信息技术发展较早，技术相对先进，软件种类齐全，物业管理系统已被广泛应用于各大企业中^[8]，通过管理信息系统实现具有检索迅速、查找方便、可靠性高、存储量大等优点，给社区的管理人员带来极大的方便。美国在信息化物业管理的过程中一直作为领导者的角色，也一直走在物业管理的领先地位^[9]，人们使用专用的信息物业管理业务在美国和一些欧洲的国家地区应用的十分普遍，节省了大量资源和人工的成本。在美国和欧洲这些发达国家中^[10]，人们计算机技术起步早和快速的发展，已经在社区物业管理系统方面拥有相当成熟以及完善的一套规范和标准要求，可以更好的服务业主们，也改变了管理人员的服务模式和业主的被服务模式^[11]。

1.3 本章小结

本章节主要介绍光明社区物业管理系统的研究背景及其意义，并分别讲述国内外的研究现状，说明社会对物业管理系统所产生的需求，以此对系统的功能架构进一步的优化，并说明本系统的开发设计思想。

2 相关技术

2.1 SpringBoot 框架

SpringBoot 框架目的是用来简化新 Spring 应用的初始搭建以及开发过程。该框架使用了特定的方式来进行配置，从而使开发人员不再需要定义样板化的配置。SpringBoot 只是一个整合 Spring 各项功能的中间件，用于快速的开发 Spring 应用，同时针对于常用的第三方库也提供了默认的配置，几乎可以实现 0 配置开发的目的。SpringBoot 采用 JavaConfig 的方式，对 Spring 进行配置，并且提供了大量的注解，极大的提高了工作效率。

2.2 LayUI 框架

LayUI 框架是一款采用自身模块规范编写的前端 UI 框架，遵循原生 HTML/CSS/JS 的书写与组织形式。它属于轻量级框架，简单美观，基于 DOM 驱动，它在服务端页面上有非常好的效果。

2.3 MySQL 数据库

MySQL 完全适用于网络，用其建造的数据库可在因特网上的任何地方访问，因此，可以和网络上任何地方的任何人共享数据库。MySQL 具有功能强、使用简单、管理方便、运行速度快、可靠性高、安全保密性强等优点^[12]。

2.4 ECharts 图表

ECharts 是一个纯 Javascript 的图表库，可以流畅的运行在 PC 和移动设备上，兼容当前绝大部分浏览器，底层依赖轻量级的 Canvas 类库 ZRender，提供直观，生动，可交互，可高度个性化定制的数据可视化图表。ECharts 提供了常规的折线图、柱状图、饼图等，用于统计的盒形图。

2.5 本节小结

本章节对系统的开发所使用的 SpringBoot 开发框架、LayUI 框架、ECharts 图表以及所使用的 MySQL 数据库进行了详细的介绍，向用户揭示本系统的相关技术构成，前端使用 LayUI 框架，后端使用 SpringBoot 框架。

3 系统可行性和需求分析

3.1 系统功能需求分析

3.1.1 功能分析

为实现社区物业管理规模化发展的需求，本系统需要支持管理员管理和业主管理，为了提高运营和管理效率，本系统将业主端和管理员端整合，提供一站式服务^[13]。

系统具体功能如下：

（1）车位收费模块：管理员查看业主车位信息，以及缴费情况并进行处理，业主可查看自己的车位信息。

（2）物业收费模块：管理员查看物业信息，以及缴费情况并进行处理，业主可查看自己的物业信息。

（3）投诉信息模块：管理员查看业主的投诉信息，以及进行处理，业主可查看自己的投诉信息并添加投诉信息。

（4）报修信息模块：管理员查看保修信息，以及处理情况并进行处理，业主可查看自己的报修信息并添加报修信息。

（5）房屋信息模块：管理员可以添加、修改、删除、查询房屋的信息。

（6）业主信息模块：管理员可以添加、修改、删除、查询业主的信息。

（7）车位信息模块：管理员可以添加、修改、删除、查询车位的信息。

（8）用户信息模块：管理员可以添加管理员、修改用户密码、删除和查询用户的信息。

（9）数据统计功能：统计业主对各种报修情况。

（10）访问拦截与权限控制功能：验证账号密码是否正确，并检查用户权限。

3.1.2 系统用例分析

根据上述的系统功能分析可将本系统根据用户进行划分，分别是管理员和业主，以下对用户进行系统用例分析。

其中管理员共有 11 用例，如图 3.1 所示：

（1）登录：管理员可以根据自己的账号和密码进行登录。

（2）车位收费管理：管理员查看业主车位信息，以及缴费情况并进行处理。

（3）物业收费管理：管理员查看物业信息，以及缴费情况并进行处理。

（4）投诉信息管理：管理员查看业主的投诉信息，以及进行处理

- (5) 报修信息管理：管理员查看保修信息，以及处理情况并进行处理
- (6) 房屋管理：管理员可以添加、修改、删除、查询房屋的信息。
- (7) 业主管理：管理员可以添加、修改、删除、查询业主的信息。
- (8) 车位管理：管理员可以添加、修改、删除、查询车位的信息。
- (9) 用户管理：管理员可以添加管理员、修改用户密码、删除和查询用户的信息。
- (10) 报修统计：查看业主对各种报修数量情况。
- (11) 退出登录：管理员注销退出系统。

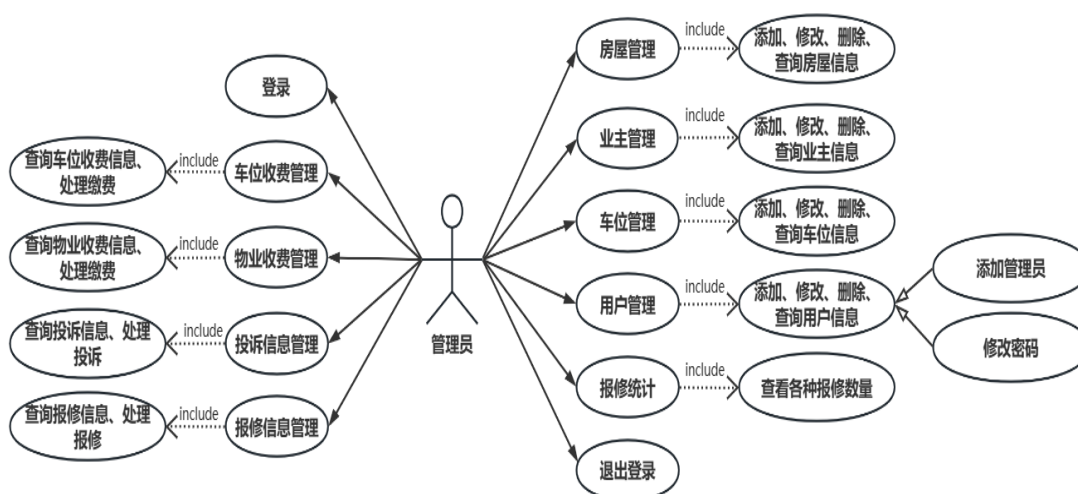


图 3.1 管理员用例图

其中业主共有 6 用例，如图 3.2 所示：

- (1) 登录：管理员可以根据自己的账号和密码进行登录。
- (2) 车位收费管理：业主可查看自己的车位信息。。
- (3) 物业收费管理：业主可查看自己的物业信息。
- (4) 投诉信息管理：业主可查看自己的投诉信息并添加投诉信息。
- (5) 报修信息管理：业主可查看自己的报修信息并添加报修信息。
- (6) 退出登录：业主注销退出系统。

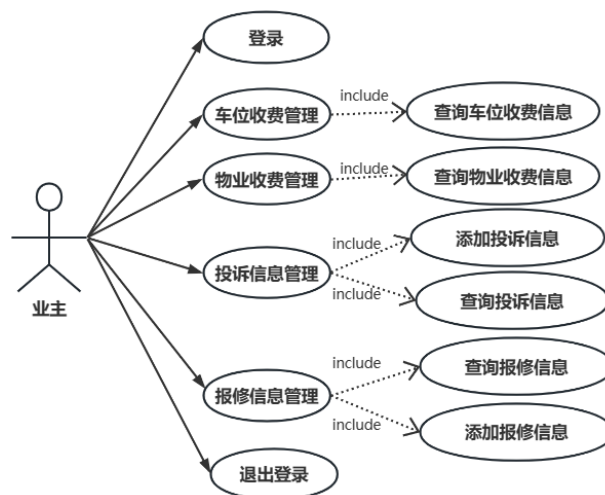


图 3.2 业主用例图

3.2 系统非功能需求分析

3.2.1 运行需求分析

本系统可运行在 Window 系统，服务器硬件最低要求 1 核 2G，带宽 1M，以保障流畅运行，并且需要 MySQL5.7 及以上版本、JRE1.8 环境支持。前端用浏览器就能使用。

本系统对硬件要求较低，适合大部分设备运行。

3.2.2 性能需求分析

本系统所有接口请求时间会控制在 1 秒内，除非发生异常或受数据库性能限制。并且支持多线程并发访问，本系统通过事务、锁保证并发安全^[14]。

本系统在低配置硬件中也能流程运行，满足一般社区使用需求。

3.2.3 界面需求分析

本系统前端界面使用 LayUI 设计，简洁、美观、大方、功能清晰有序，易于上手对操作人员要求低。

3.2.4 其他需求分析

安全性需求：员工无法访问管理员接口，管理员也无法访问员工接口，从而实现权限隔离，避免越权操作^[15]。

可靠性需求：使用统一异常处理拦截异常，保证系统正常运行。

3.3 系统可行性分析

3.3.1 经济可行性

随着当前的社会发展，各类社区也在快速的发展，在一些老城区也存在不少的社

区，然而这类的社区都还没有建立社区物业管理系统，。这些社区绝大部还在处于人工管理的阶段，并不具备电子信息化管理的条件，需要消耗大量的人力和资源。因此引入社区物业管理系统成为一个好的选择，社区物业管理系统可以降低人力资源，提高社区的服务质量。综上所述，设计开发一个社区物业管理系统在经济上可行。

3.3.2 技术可行性

随着计算机硬件水平的快速发展，当前软件技术也在迅速的发展，各方面都在迅速的发展。在系统的开发语言中也有不少的选择了，例如：Java、PHP、C#等语言，在数据库的存储上也有不同的选择，例如：MySQL、Oracle 等，在前端页面上的选择，例如：html 页面，jsp 页面等。综上所述，开发一个社区物业管理系统在技术上是可行的。本系统决定采用 SpringBoot 框架进行开发，可以对系统进行快速开发，前端使用 LayUI 框架提高开发效率。

3.4 本章小结

本节从系统的各方面对系统的可行性进行了介绍，也对系统的功能需求进行了说明本系统的功能模块的主要内容，并说明了管理员和业主的用例，以用例图体现管理员和业主的具体用例，以确保本系统的顺利开发。

4 系统设计

4.1 系统总体设计

4.1.1 需求规定

1. 车位收费管理

车位收费管理由管理员进行操作，用户要先以管理员身份登录本系统后方可使用添加业主的车位收费，并进行缴费处理或查询车位号、缴费状态，进行删除操作。

处理过程描述：先检查用户是否具有管理员身份，然后管理员添加业主的车位的收费金额，可进行缴费处理，并保存信息到数据库。

2. 物业收费管理

物业收费管理由管理员进行操作，用户要先以管理员身份登录本系统后方可使用添加业主的物业收费，并进行缴费处理或查询车位号、缴费状态，进行删除操作。

处理过程描述：先检查用户是否具有管理员身份，然后让用户添加业主的物业费的开始时间和结束时间，显示业主应交金额，可进行缴费处理，并保存信息到数据库。

3. 投诉信息管理

投诉信息管理由管理员进行操作，用户要先以管理员身份登录本系统后方可使用对业主的投诉信息进行处理功能或查询投诉类型和处理状态，进行删除操作。

处理过程描述：先检查用户是否具有管理员身份，如果是管理员可以对业主的投诉信息进行处理，并保存信息到数据库。如果是业主可以添加投诉信息，并保存到数据库。

4. 报修信息管理

报修信息管理由管理员进行操作，用户要先以管理员身份登录本系统后方可使用对业主的报修信息进行处理功能或查询报修类型和处理状态，进行删除操作。

处理过程描述：先检查用户是否具有管理员身份，如果是管理员可以对业主的报修信息进行处理，并保存信息到数据库。如果是业主可以添加报修信息，并保存到数据库。

5. 房屋管理

房屋信息管理由管理员进行操作，用户要先以管理员身份登录本系统后方可使用修改房屋信息、删除房屋、查看房屋列表、增加房屋功能。

处理过程描述：先检查用户是否具有管理员身份，根据房间号筛选并显示房间号

列表，管理员可在列表上进行增加、删除或修改操作，并把数据保存到数据库。

6. 业主信息管理

盘点管理由管理员进行操作，用户要先以管理员身份登录本系统后方可使用增加业主、修改业主信息、删除业主、查看业主列表功能。

处理过程描述：先检查用户是否具有管理员身份，根据电话或身份证号筛选并显示业主列表，管理员可在列表上进行增加、删除或修改操作，并把数据保存到数据库。

7. 车位信息管理

盘点管理由管理员进行操作，用户要先以管理员身份登录本系统后方可使用增加车位、修改车位信息、删除车位、查看车位列表功能。

处理过程描述：先检查用户是否具有管理员身份，根据车位筛选并显示车位列表，管理员可在列表上进行增加、删除或修改操作，并把数据保存到数据库。

8. 用户信息管理

盘点管理由管理员进行操作，用户要先以管理员身份登录本系统后方可使用增加用户、修改用户密码信息、删除用户、查看用户列表功能。

处理过程描述：先检查用户是否具有管理员身份，根据用户类型筛选并显示用户列表，用户可在列表上进行增加、删除或修改密码操作，并把数据保存到数据库。

9. 报修统计

盘点管理由管理员进行操作，用户要先以管理员身份登录本系统后方可使用查看报修类型统计功能。

处理过程描述：先检查用户是否具有管理员身份，然后可以显示业主各种报修的统计。

10. 访问拦截与权限控制

访问拦截与权限控制是系统的安全模块，此模块负责拦截未登录请求、越权请求，保证接口的合法调用。

处理过程描述：先检查用户输入的用户名和密码以及类型是否填写正确，如果验证正确，则跳转到相应的页面。如果验证错误，则进行拦截，重新进行登录。

4.1.2 运行环境

操作系统：Window10。

处理器：1 核 2G。

带宽：1M。

数据库：MySQL5.7。

4.1.3 处理过程

用户进入系统后，验证是否登录成功，如没有登录成功则跳转到登录界面再次进行验证登录。如登录成功则判断是否为管理员，是管理员则跳转至管理员的界面，如是业主则跳转至业主界面。并进行具体操作。登录验证处理过程如图 4.1 所示：

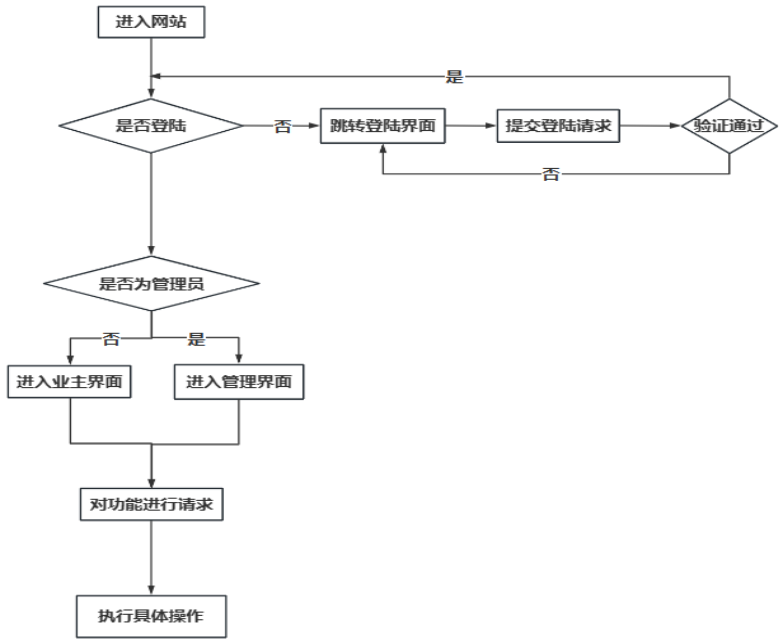


图 4.1 登录验证流程图

用户请求接口，判断是否登录，并查看权限，向系统发送请求，检验参数，向数据库获取数据，并返回结果，如图 4.2 所示：

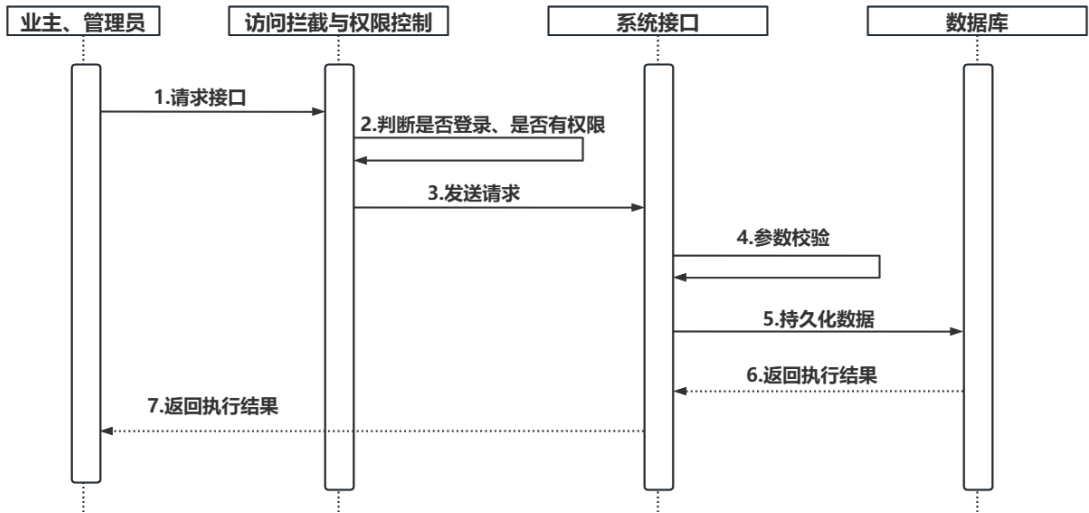


图 4.2 验证顺序图

用户请求登陆成功登陆后跳转至相应界面，用户打开车位收费管理界面，业主可查看自己的车位收费情况，管理员查看业主车位信息、添加车位收费信息，以及业主

缴费情况，可以对未缴费进行缴费处理，可以删除车位收费信息操作，如图 4.3 所示

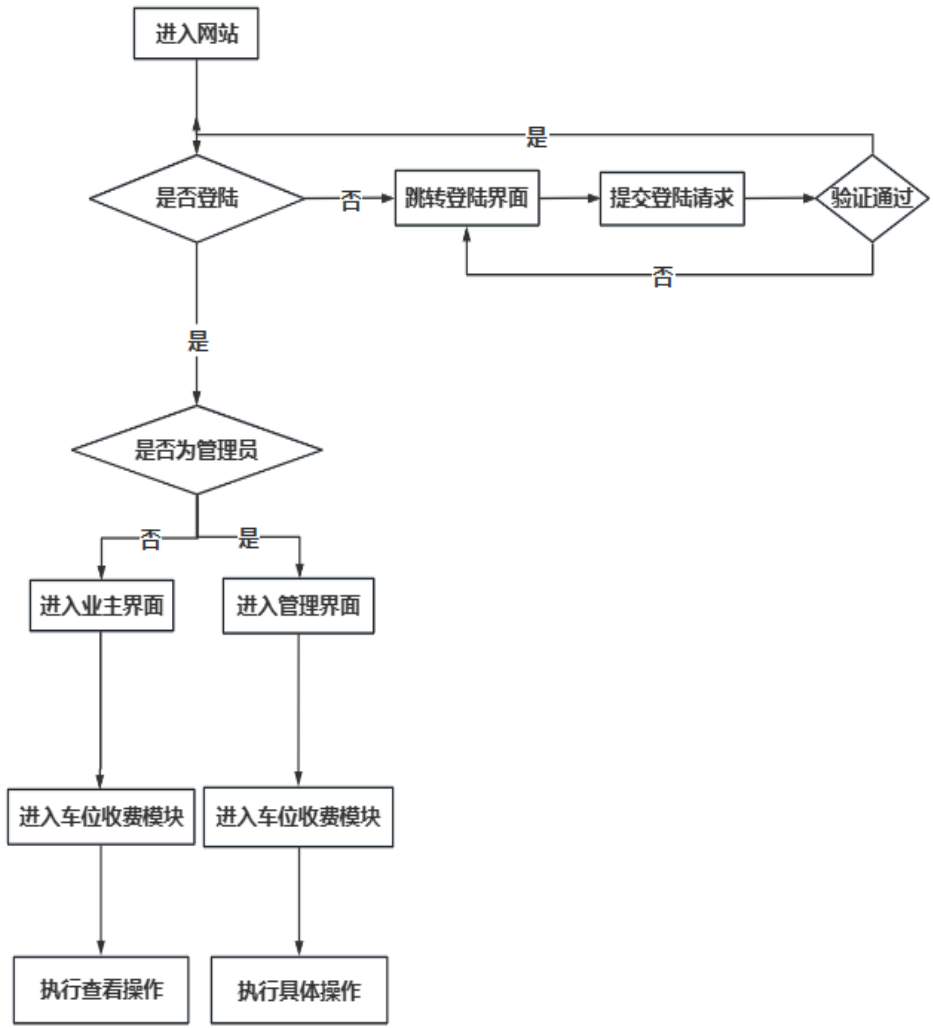


图 4.3 车位收费流程图

4.2 功能模块图

光明社区物业系统由十个模块组成，分别是车位收费管理模块、物业收费管理模块、投诉信息管理模块、报修信息管理模块、房屋管理模块、业主管理模块、车位管理模块、报修统计模块、用户模块、访问拦截与权限控制模块。

从总体结构中可了解到本系统的大致功能，光明社区物业管理系统的总体结构如图 4.4 所示：

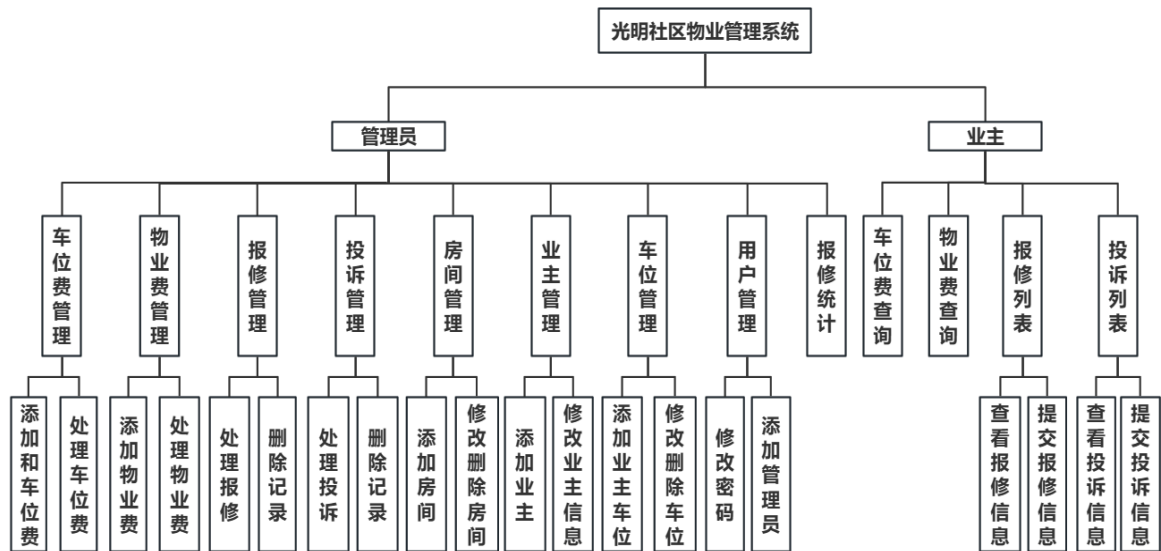


图 4.4 系统总体结构图

4.3 本章小结

本节从系统总体设计的需求规定说明本系统的各个模块的功能设计需求，以了解系统各模块的主要功能。在运行环境的基础上，进一步说明访问拦截与权限控制模块的认证流程。

5 数据库设计

数据库设计阶段主要考虑如何存储系统所需的数据，设计良好的表结构和适当的使用索引可以很大程度上提高系统的性能和系统所能存储的数据量。同时在数据库设计时也要遵循一些规范，这样可以降低后期维护的难度^[15]。

5.1 图设计

5.1.1 总体 E-R 图

本系统共有 11 个实体，分别是车位收费、车位、物业收费、投诉、报修、报修类型、房屋、业主、用户、投诉类型、收费类型，具体关系如图 5.1 所示：

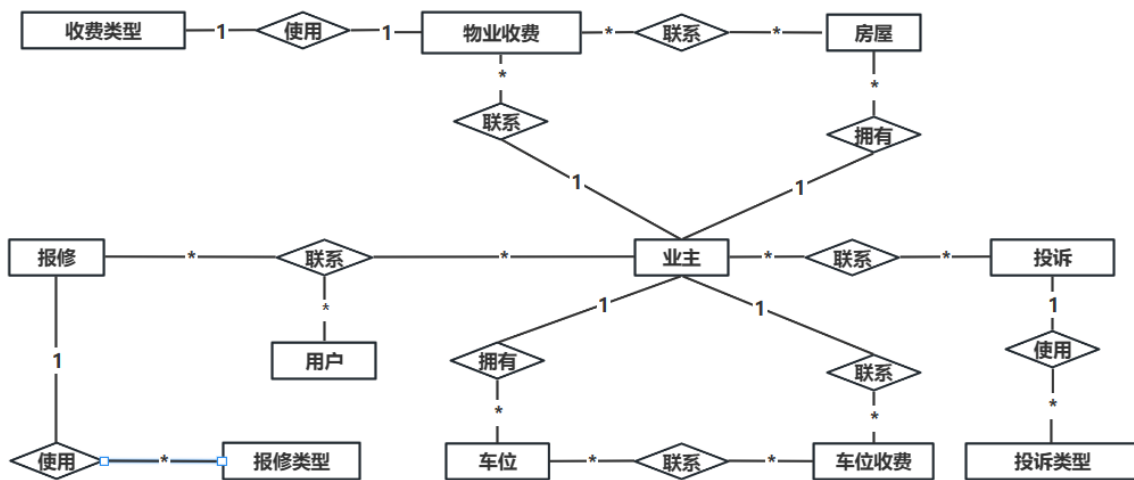


图 5.1 总体 E-R 图

5.1.2 车位收费实体属性图

车位收费由字段为编号、开始时间、结束时间、金额、状态、业主 id、备注、收费类型、车牌号 id 组成，具体内容如图 5.2 所示：

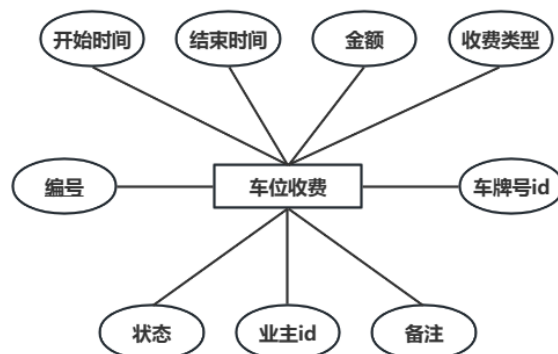


图 5.2 车位收费实体属性图

5.1.3 投诉实体属性图

投诉由字段为编号、投诉类型 id、投诉时间、处理时间、业主 id、状态、处理

人、备注，组成，具体内容如图 5.3 所示：

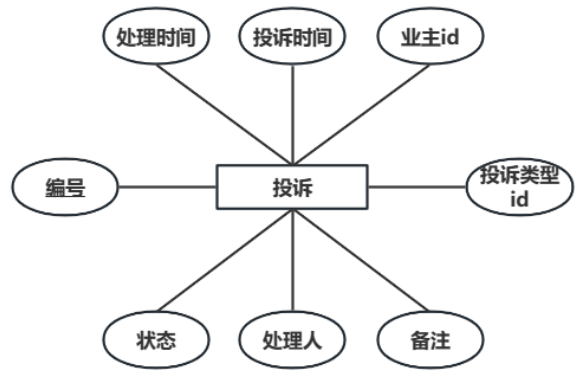


图 5.3 投诉实体属性图

5.1.4 投诉类型实体属性图

投诉类型由字段为编号、名称、备注、状态组成，具体内容如图 5.4 所示：

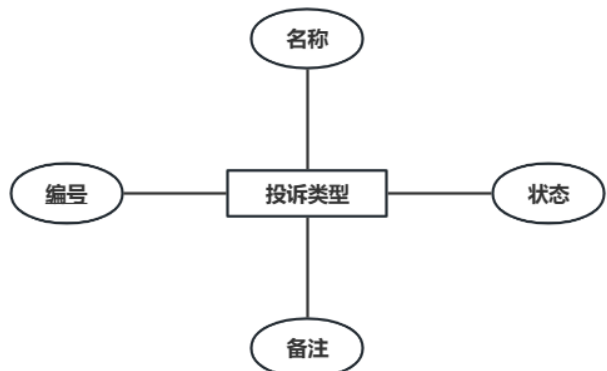


图 5.4 投诉类型实体属性图

5.1.5 房屋实体属性图

房屋由字段为编号、楼层、房屋号、状态、入住时间、楼栋 id、备注、面积组成，具体内容如图 5.5 所示：

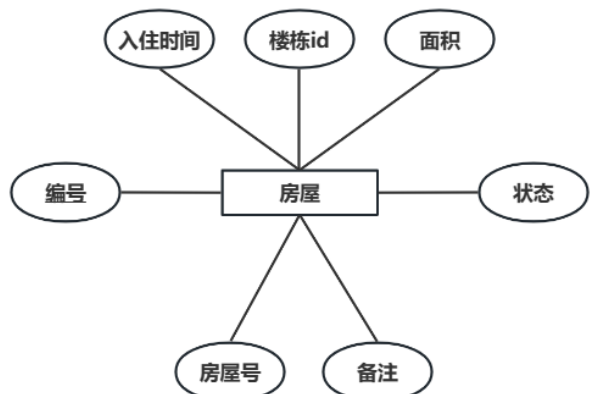


图 5.5 房屋实体属性图

5.1.6 业主实体属性图

业主由字段为编号、业主名、联系电话、性别、身份证号、房屋 id、备注、密码组成，具体内容如图 5.6 所示：

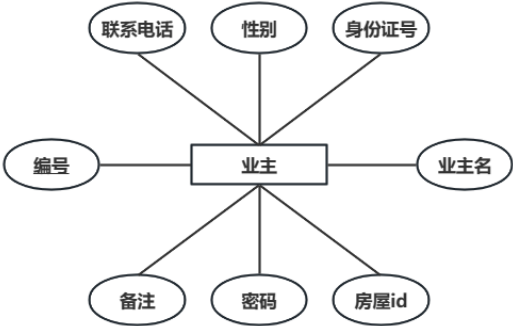


图 5.6 业主实体属性图

5.1.7 车位实体属性图

车位由字段为编号、车位号、状态、业主 id、备注组成，具体内容如图 5.7 所示：

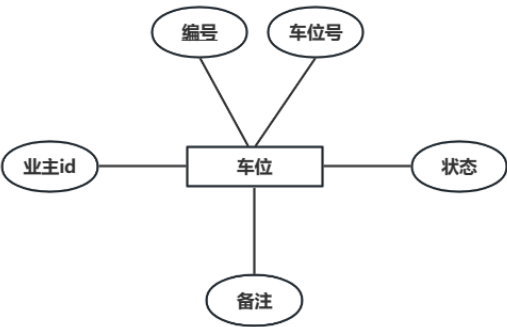


图 5.7 车位实体属性图

5.1.8 报修实体属性图

报修由字段为编号、报修类型 id、报修时间、处理时间、业主 id、状态、处理人、备注组成，具体内容如图 5.8 所示：

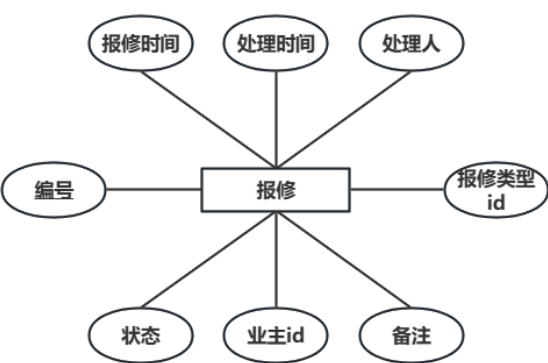


图 5.8 报修实体属性图

5.1.9 报修类型实体属性图

报修类型由字段为编号、名称、备注、状态组成，具体内容如图 5.9 所示：

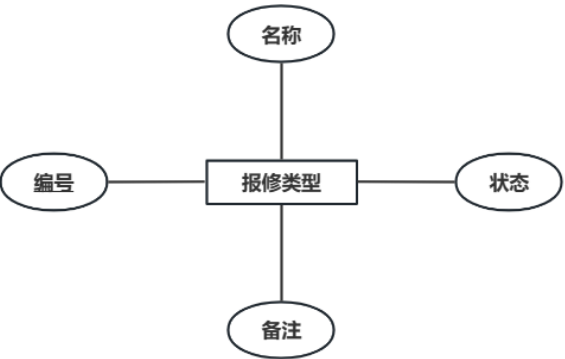


图 5.9 报修类型实体属性图

5.1.10 用户实体属性图

用户由字段为编号、用户名、密码、类型、备注组成，具体内容如图 5.10 所示：

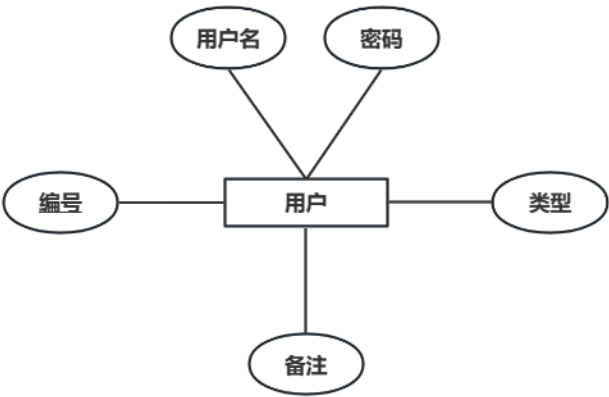


图 5.10 用户实体属性图

5.1.11 收费类型实体属性图

收费类型由字段为编号、名称、单位、备注组成，具体内容如图 5.11 所示：

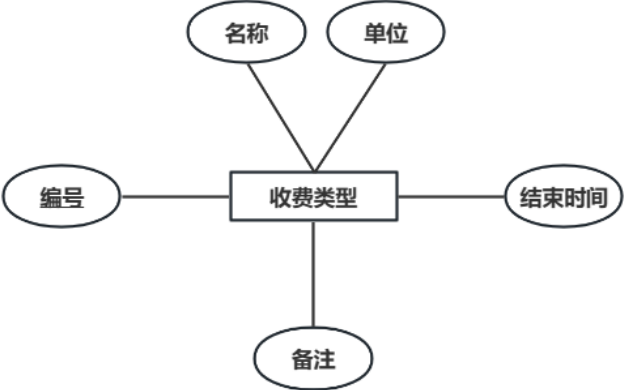


图 5.11 收费类型实体属性图

5.1.12 物业收费实体属性图

物业收费由字段为编号、类型 id、金额、开始时间、结束时间、状态、房屋 id、备注组成，具体内容如图 5.12 所示：



图 5.12 物业收费实体属性图

5.2 表设计

本系统采用 MySQL5.7 作为数据库，所有表使用 InnoDB 引擎，其提供了事务和行锁支持，具体表设计如下。

1. 车位收费表 (carcharge)

车位收费表用于保存车位收费记录，字段为编号、开始时间、结束时间、金额、状态、业主 id、备注、收费类型、车牌号 id，具体内容如表 5.1 所示：

表 5.1 车位收费表

列名	数据类型	长度	允许空	是否主键	说明
id	int	10	×	是	编号
pay_date	datetime	/	×	否	开始时间
end_date	datetime	/	×	否	结束时间
money	double	10, 2	×	否	金额
status	int	2	×	否	状态
owner_id	int	10	×	否	业主 id
remarks	varchar	255	√	否	备注
type	varchar	10	×	否	收费类型
park_id	int	10	×	否	车牌号 id

2. 投诉表 (complaint)

投诉表用于投诉信息，字段为编号、投诉类型 id、投诉时间、处理时间、业主 id、状态、处理人、备注，具体内容如表 5.2 所示：

表 5.2 投诉表

列名	数据类型	长度	允许空	是否主键	说明
id	int	11	×	是	编号
com_id	varchar	25	×	否	投诉类型 id
com_date	datetime	/	×	否	投诉时间
handle_date	datetime	/	×	否	处理时间
owner_id	int	11	×	否	业主 id
status	int	2	×	否	状态
clr	int	11	×	否	处理人
remarks	varchar	255	√	否	备注

3. 投诉类型表 (complaint_type)

投诉类型表用于投诉类型信息，字段为编号、名称、备注、状态，具体内容如表

5.3 所示：

表 5.3 投诉类型表

列名	数据类型	长度	允许空	是否主键	说明
id	int	10	×	是	编号
name	varchar	50	×	否	名称
remarks	varchar	255	√	否	备注
status	int	10	√	否	状态

4. 房屋表 (house)

房屋表用于房屋信息，字段为编号、楼层、房屋号、状态、入住时间、楼栋 id、

备注、面积，具体内容如表 5.4 所示：

表 5.4 房屋表

列名	数据类型	长度	允许空	是否主键	说明
id	int	11	×	是	编号
numbers	varchar	50	×	否	房屋号
status	int	10	√	否	状态
into_date	datetime	/	×	否	入住时间
remarks	varchar	255	√	否	备注
area	double	10, 2	×	否	面积

5. 业主表 (owner)

业主表用于业主信息，字段为编号、业主名、联系电话、性别、身份证号、房屋

id、备注、密码，具体内容如表 5.5 所示：

表 5.5 业主表

列名	数据类型	长度	允许空	是否主键	说明
id	int	5	×	是	编号
username	varchar	25	×	否	业主名
tel	varchar	25	×	否	联系电话
sex	varchar	5	×	否	性别

续表 5.5

列名	数据类型	长度	允许空	是否主键	说明
identity	varchar	25	×	否	身份证号
house_id	int	11	×	否	房屋 id
remarks	varchar	255	√	否	备注
password	varchar	28	×	否	密码

6. 车位表 (parking)

车位表用于保存车位信息, 字段为编号、车位号、状态、业主 id、备注, 具体内容如表 5.6 所示:

表 5.6 车位表

列名	数据类型	长度	允许空	是否主键	说明
id	int	11	×	是	编号
numbers	varchar	25	×	否	车位号
status	int	11	√	否	状态
owner_id	int	11	×	否	业主 id
remarks	varchar	255	√	否	备注

7. 报修表 (repair)

报修表用于保存物业收费信息, 字段为编号、报修类型 id、报修时间、处理时间、业主 id、状态、处理人、备注, 具体内容如表 5.7 所示:

表 5.7 报修表

列名	数据类型	长度	允许空	是否主键	说明
id	int	11	×	是	编号
com_id	varchar	25	×	否	报修类型 id
com_date	datetime	/	×	否	报修时间
handle_date	datetime	/	×	否	处理时间
owner_id	int	11	×	否	业主 id
status	int	11	√	否	状态
clr	int	11	√	否	处理人
remarks	varchar	255	√	否	备注

8. 报修类型表 (repair_type)

报修类型表用于保存报修类型信息, 字段为编号、名称、备注、状态, 具体内容如表 5.8 所示:

表 5.8 报修类型表

列名	数据类型	长度	允许空	是否主键	说明
id	int	11	×	是	编号
name	varchar	25	×	否	名称
remarks	varchar	255	√	否	备注
status	int	11	√	否	状态

9. 用户表 (userinfo)

用户表用于保存用户信息, 字段为编号、用户名、密码、类型、备注, 具体内容如表 5.9 所示:

表 5.9 用户表

列名	数据类型	长度	允许空	是否主键	说明
id	int	11	×	是	编号
username	varchar	20	×	否	用户名
password	varchar	20	×	否	密码
type	int	11	√	否	类型
remarks	varchar	255	√	否	备注

10. 收费类型表 (property_type)

收费类型表用于保存物业收费信息, 字段为编号、名称、单位、备注, 具体内容如表 5.10 所示:

表 5.10 收费类型表

列名	数据类型	长度	允许空	是否主键	说明
id	int	11	×	是	编号
name	int	11	×	否	名称
price	double	10, 2	×	否	单位
unit	datetime	/	×	否	结束时间
remarks	varchar	255	√	否	备注

11. 物业收费表 (property_info)

物业收费表用于保存物业收费信息, 字段为编号、类型 id、金额、开始时间、结束时间、状态、房屋 id、备注, 具体内容如表 5.11 所示:

表 5.11 物业收费表

列名	数据类型	长度	允许空	是否主键	说明
id	int	11	×	是	编号
type_id	int	11	×	否	类型 id
money	double	10, 2	×	否	金额
start_date	datetime	/	×	否	开始时间
end_date	datetime	/	×	否	结束时间
status	int	11	√	否	状态
house_id	int	11	×	否	房屋 id
remarks	varchar	255	√	否	备注

5.3 功能模块与表的关系

本系统各个功能模块与数据库表的对应关系如下表 5.12 所示:

表 5.12 功能模块对应表

功能名称	数据表
车位收费模块	车位收费表、车位表

续表 5.12

功能名称	数据表
物业收费模块	物业收费表
投诉信息模块	投诉表
报修信息模块	报修表、报修类型表
房屋管理模块	房屋表
业主管理模块	业主表
车位管理模块	车位表
用户模块	用户表
报修统计模块	报修表
访问拦截与权限控制模块	用户表

5.4 本章小结

本章节主要阐述了光明社区物业管理系统的数据库的设计思想，通过 E-R 图说明各表之间的联系，并以实体属性图展示了实体和属性之间的关系。对数据库表的设计进行详细字段说明。

6 系统实现

6.1 车位收费模块

管理员用于查看业主车位收费信息、以及缴费情况，业主可查看自己的车位费信息。管理员查看车位收费列表如图 6.1 所示：

车位号	车主	开始时间	结束时间	金额	状态	操作
A10003	mary	2020-10-01	2020-12-31	1200	已缴费	删除
A10003	mary	2020-11-01	2020-11-30	300	未缴费	缴费 删除
A10003	mary	2022-02-25	2022-06-26	50	未缴费	缴费 删除
A10003	mary	2022-05-31	2022-05-31	1000	未缴费	缴费 删除
A10003	mary	2022-06-01	2022-06-01	500	未缴费	缴费 删除
A10003	mary	2023-01-19	2023-01-19	200	未缴费	缴费 删除
A10002	Mike	2023-01-19	2023-01-19	200	未缴费	缴费 删除
A10002	Mike	2023-03-10	2023-03-30	122	未缴费	缴费 删除

图 6.1 车位费收费列表界面图

管理员选择费用的初始日期和截至日期，填入收取的金额，选择确认，即可添加成功，管理员添加车位费如图 6.2 所示：

添加车位

费用起始日期: yyyy-MM-dd

费用截至日期: yyyy-MM-dd

金额: 请输入金额

备注信息: 请输入备注信息

确认

图 6.2 添加车位费界面图

管理员点击缴费，确认缴费，缴费状态变为已交费状态，管理员缴费如图 6.3 所示：

车位号	车主	开始时间	结束时间	金额	状态	操作
A10003	mary	2020-10-01	2020-12-31	1200	已缴费	删除
A10003	mary	2020-11-01	2020-11-30	300	已缴费	缴费 删除
A10003	mary	2022-02-25	2022-06-26	50	未缴费	缴费 删除
A10003	mary	2022-05-31	2022-05-31	1000	未缴费	缴费 删除
A10003	mary	2022-06-01	2022-06-01	500	未缴费	缴费 删除
A10003	mary	2023-01-19	2023-01-19	200	未缴费	缴费 删除
A10002	Mike	2023-01-19	2023-01-19	200	未缴费	缴费 删除
A10002	Mike	2023-03-10	2023-03-30	122	未缴费	缴费 删除

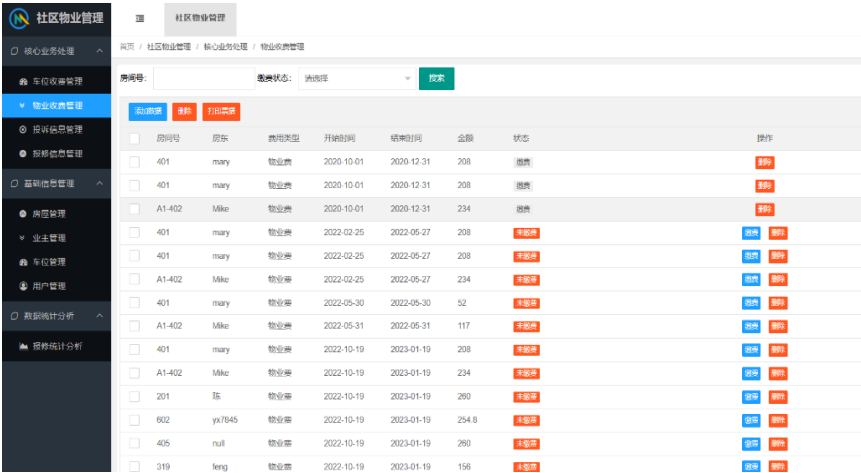
图 6.3 管理员缴费界面图

部分相关核心代码：

```
Carchargecarcharge=newCarcharge(); //处理缴费
carcharge.setId(id);
carcharge.setStatus(1);
intnum=carchargeService.updateData(carcharge);
if(num>0){
returnR.ok();}
returnR.fail("失败");
```

6.2 物业收费模块

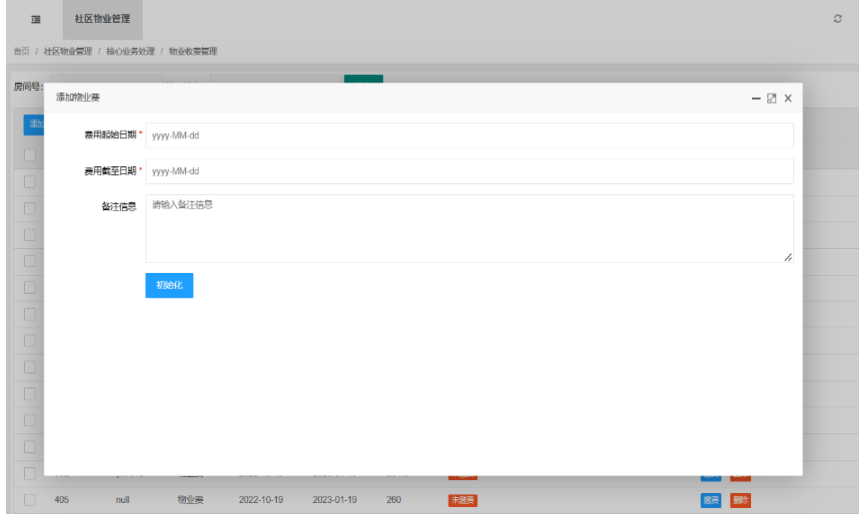
该模块用于管理员查看业主物业费信息，以及缴费情况，业主可查看自己的物业信息。管理员查看物业收费列表如图 6.4 所示：



选择	房间号	房东	费用类型	开始时间	结束时间	金额	状态	操作
<input type="checkbox"/>	401	mary	物业费	2020-10-01	2020-12-31	208	缴费	缴费
<input type="checkbox"/>	401	mary	物业费	2020-10-01	2020-12-31	208	缴费	缴费
<input type="checkbox"/>	A1-402	Mike	物业费	2020-10-01	2020-12-31	234	缴费	缴费
<input type="checkbox"/>	401	mary	物业费	2022-02-25	2022-05-27	208	未缴费	缴费 删除
<input type="checkbox"/>	401	mary	物业费	2022-02-25	2022-05-27	208	未缴费	缴费 删除
<input type="checkbox"/>	A1-402	Mike	物业费	2022-02-25	2022-05-27	234	未缴费	缴费 删除
<input type="checkbox"/>	401	mary	物业费	2022-05-30	2022-05-30	52	未缴费	缴费 删除
<input type="checkbox"/>	A1-402	Mike	物业费	2022-05-31	2022-05-31	117	未缴费	缴费 删除
<input type="checkbox"/>	401	mary	物业费	2022-10-19	2023-01-19	208	未缴费	缴费 删除
<input type="checkbox"/>	A1-402	Mike	物业费	2022-10-19	2023-01-19	234	未缴费	缴费 删除
<input type="checkbox"/>	201	陈	物业费	2022-10-19	2023-01-19	260	未缴费	缴费 删除
<input type="checkbox"/>	602	yx7845	物业费	2022-10-19	2023-01-19	254.8	未缴费	缴费 删除
<input type="checkbox"/>	405	null	物业费	2022-10-19	2023-01-19	260	未缴费	缴费 删除
<input type="checkbox"/>	319	feng	物业费	2022-10-19	2023-01-19	156	未缴费	缴费 删除

图 6.4 物业收费列表界面图

管理员选择费用的初始日期和截至日期，选择确认，即可添加成功，管理员添加物业收费如图 6.5 所示：



添加物业费

费用起始日期 * yyyy-MM-dd

费用截至日期 * yyyy-MM-dd

备注信息 请输入备注信息

初始化

图 6.5 添加物业收费界面图

管理员点击缴费，确认缴费，缴费状态变为已交费状态，管理员缴费如图 6.6 所

示：

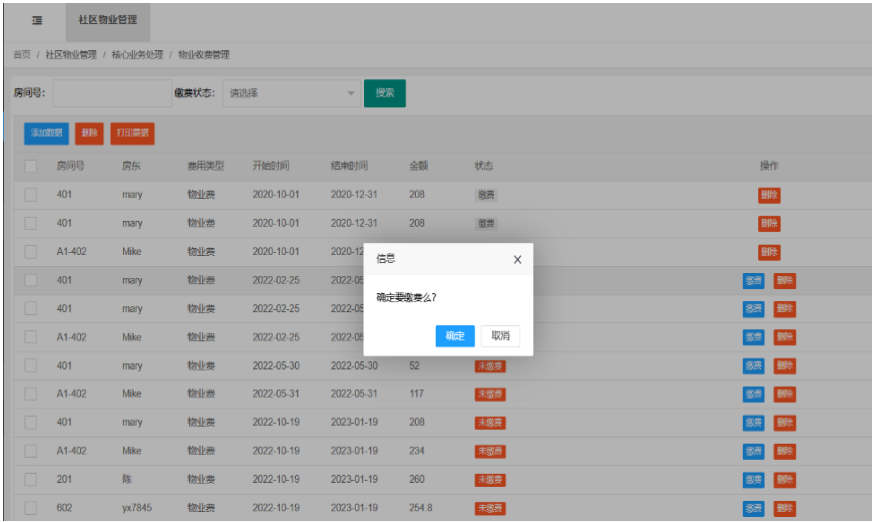


图 6.6 管理员缴费界面图

部分相关核心代码：

```
for(Househouse:list) {  
    if(status!=null || status!=0) {  
        doublemoney=house.getArea()*price;  
        propertyInfo.setMoney(money);  
        propertyInfo.setHouseId(house.getId());  
        propertyInfoService.add(propertyInfo);  
    }  
}
```

6.3 投诉信息模块

该模块用于管理员查看业主的投诉信息，以及处理状态，可以删除投诉信息，业主可查看自己的投诉信息。管理员查看投诉列表如图 6.7 所示：

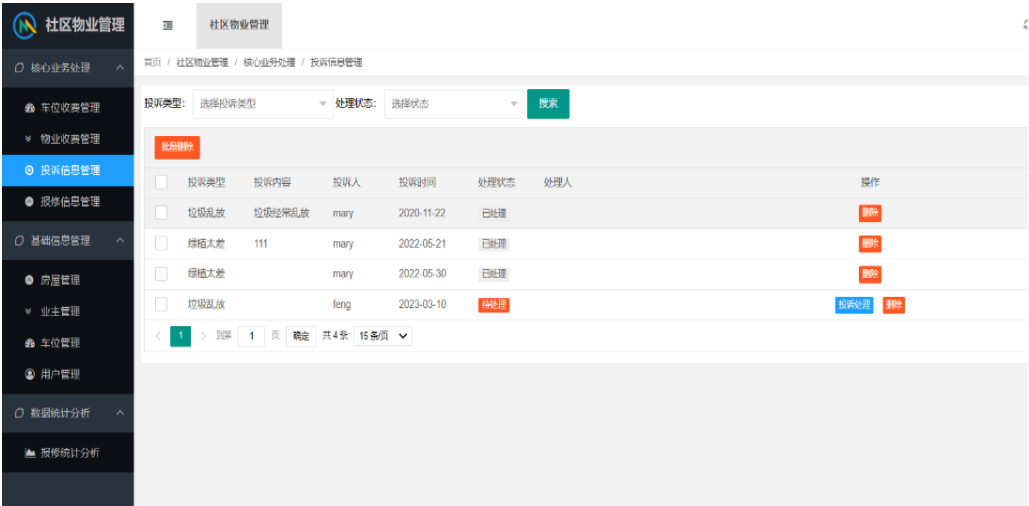


图 6.7 管理员投诉列表界面图

管理员点击缴费，确认处理，处理状态变为已处理状态，管理员处理投诉信息如图 6.8 所示：

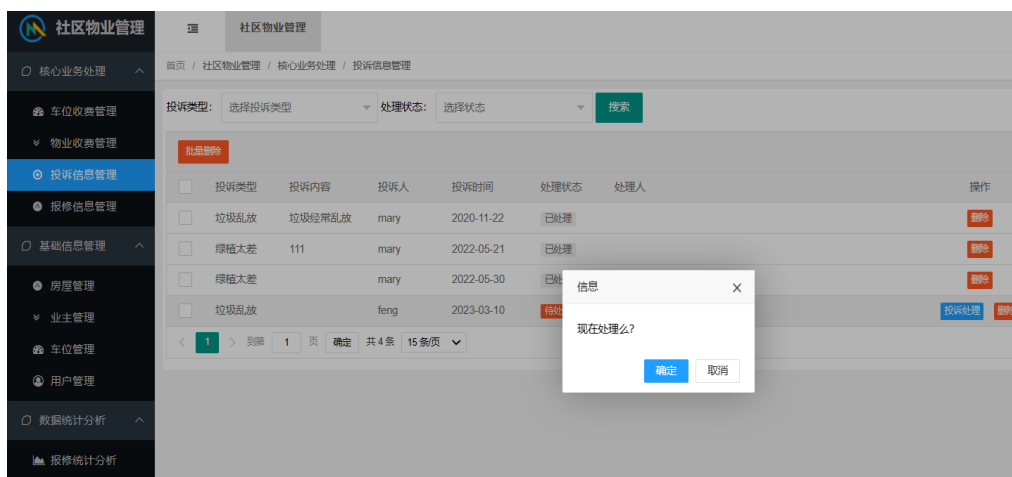


图 6.8 管理员处理投诉信息界面图

业主选择需要投诉的类型，选择确认，即可添加成功，业主添加投诉如图 6.9 所示：

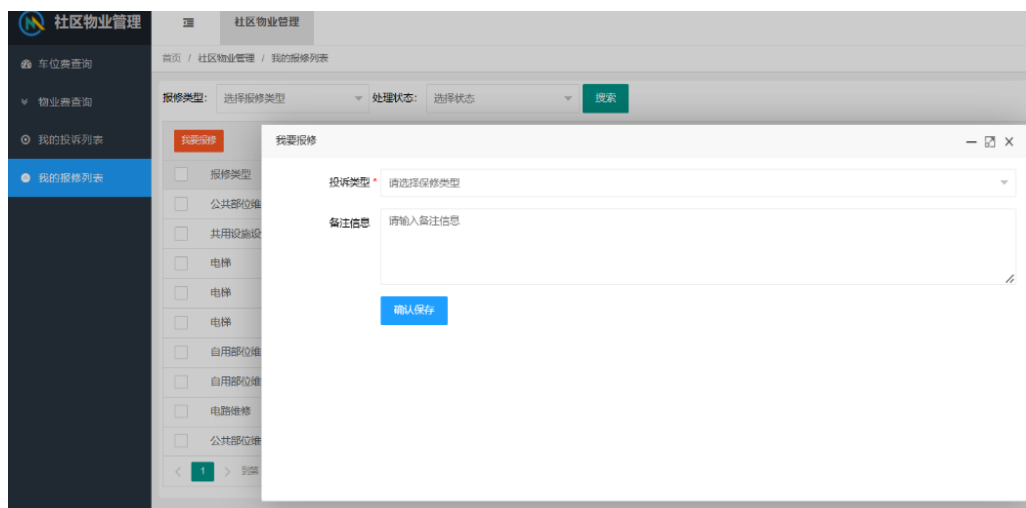


图 6.9 业主添加投诉界面图

部分相关核心代码：

```
Complaintcomplaint=newComplaint();//处理投诉
complaint.setId(id);
complaint.setStatus(1);
intnum=complaintService.updateData(complaint);
if(num>0){
returnR.ok();
}else{
returnR.fail("处理失败");
}
```

6.4 报修信息模块

该模块用于管理员查看报修信息，以及可以对未处理的进行处理，可以删除报修信息，业主可查看自己的报修信息。管理员报修列表如图 6.10 所示：

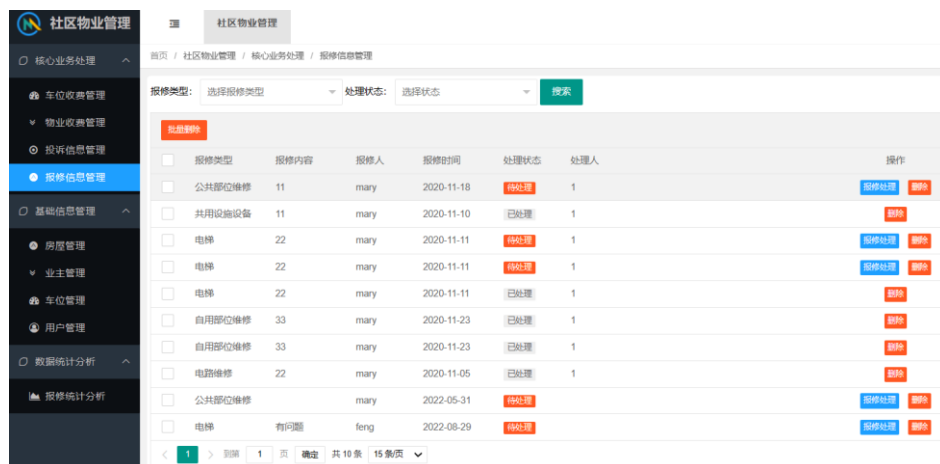


图 6.10 管理员报修列表界面图

管理员点击缴费，确认处理，处理状态变为已处理状态，管理员处理报修信息如图 6.11 所示：

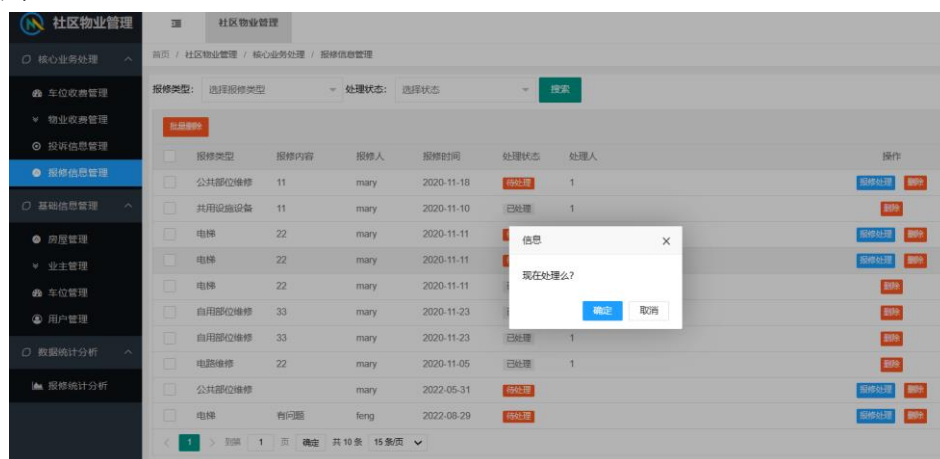


图 6.11 管理员处理报修界面图

业主选择需要报修的类型，选择确认，即可添加成功，业主添加报修如图 6.12 所示：

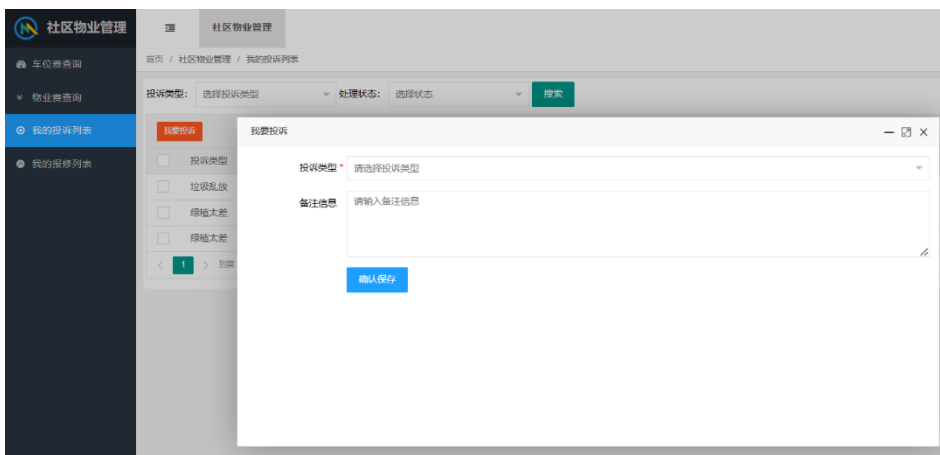


图 6.12 业主添加报修界面图

部分相关核心代码：

```
Userinfouserinfo=(Userinfo)request.getSession().getAttribute("user");
```

```
Stringusername=userinfo.getUsername();
Ownerowner=ownerService.queryOwnerByName(username);
repair.setOwnerId(owner.getId());
repair.setComDate(newDate());
intnum=repairService.add(repair);
```

6.5 房屋模块

管理员用于查看房屋的具体信息，以及入住状态，管理员房屋信息列表如图 6.13 所示：

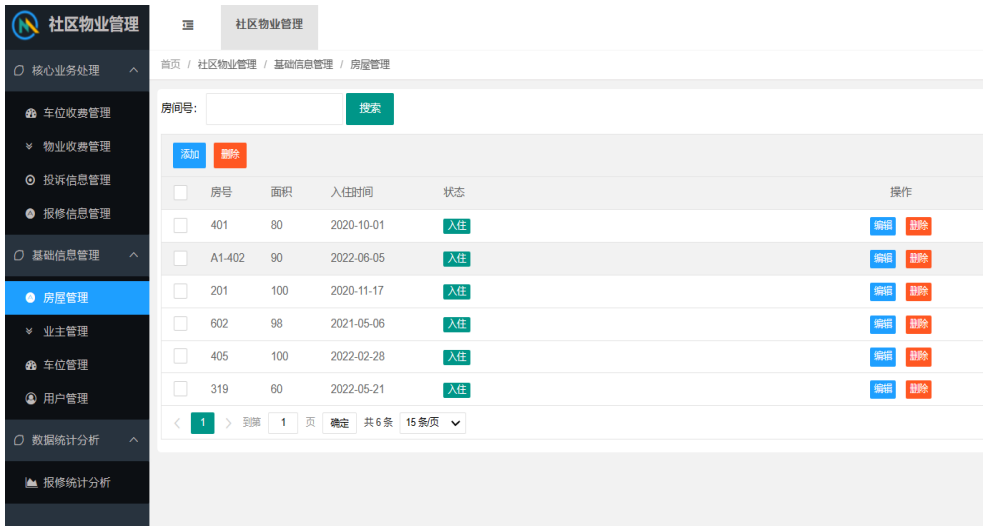


图 6.13 房屋信息列表界面图

管理员填写房屋编号、面积、选择入住时间，点击确认添加，管理员添加房屋信息如图 6.14 所示：

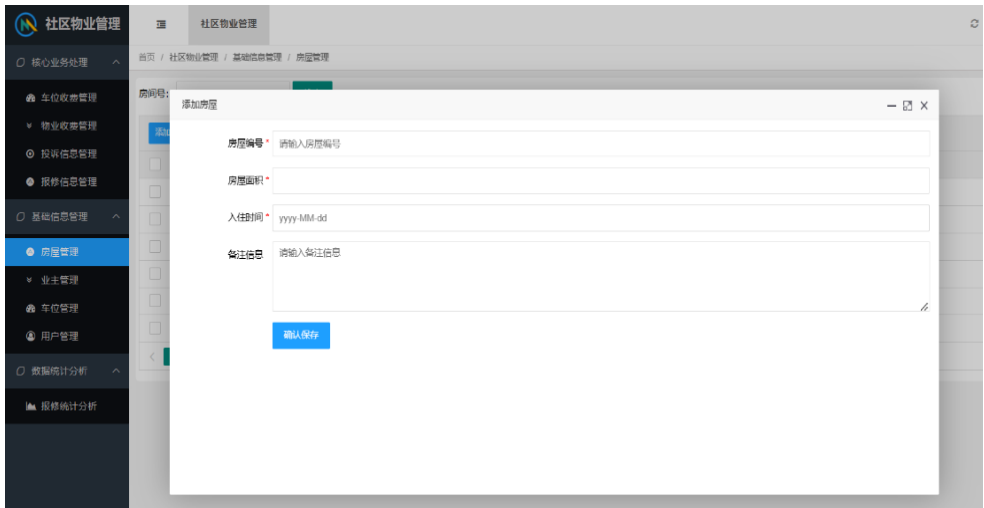


图 6.14 添加房屋信息界面图

管理员可修改房屋编号、面积、选择入住时间，点击确认进行修改，管理员修改房屋信息如图 6.15 所示：

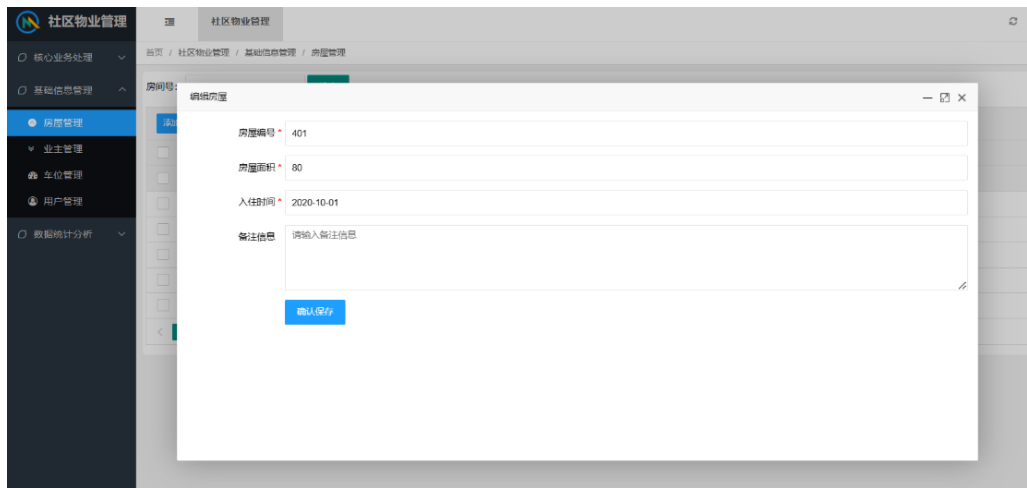


图 6.15 修改房屋信息界面图

部分相关核心代码:

```
if (house.getIntoDate() != null) { // 添加房屋信息
    house.setStatus(1);
} else {
    house.setStatus(0);
}
int num = houseService.add(house);
if (num > 0) {
    return R.ok();
} else {
    return R.fail("添加失败");
}
```

6.6 业主信息模块

管理员用于查看业主得具体信息，管理员查看业主信息如图 6.16 所示：

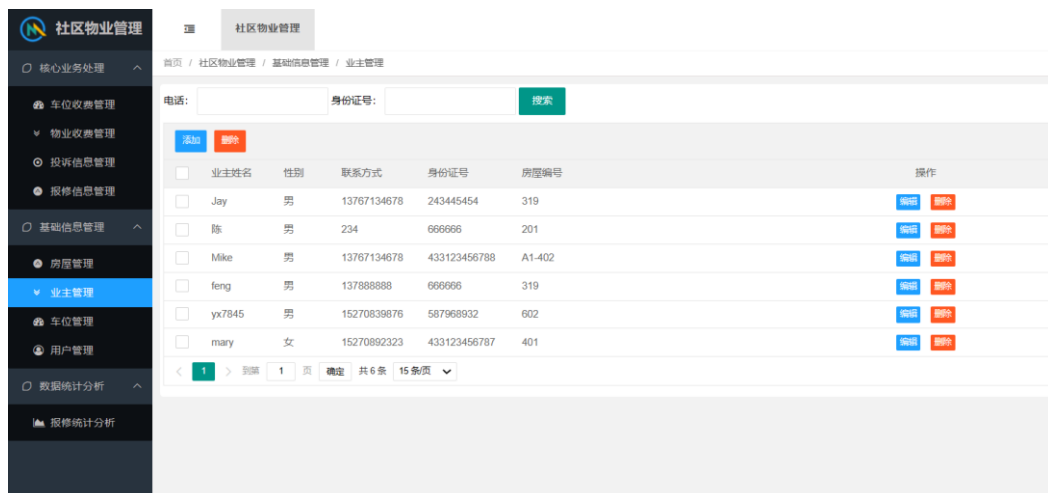


图 6.16 业主信息列表界面图

管理员填写业主姓名、业主电话、性别、身份证、房屋编号，点击确认添加，管理员添加业主信息如图 6.17 所示：

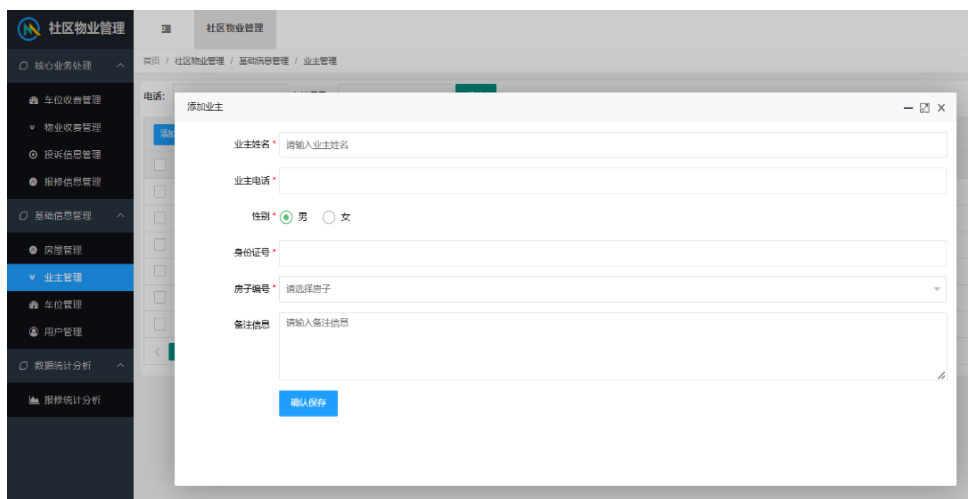


图 6.17 添加业主信息界面图

管理员可修改业主姓名、业主电话、性别、身份证、房屋编号，点击确认修改，管理员修改业主信息如图 6.18 所示：

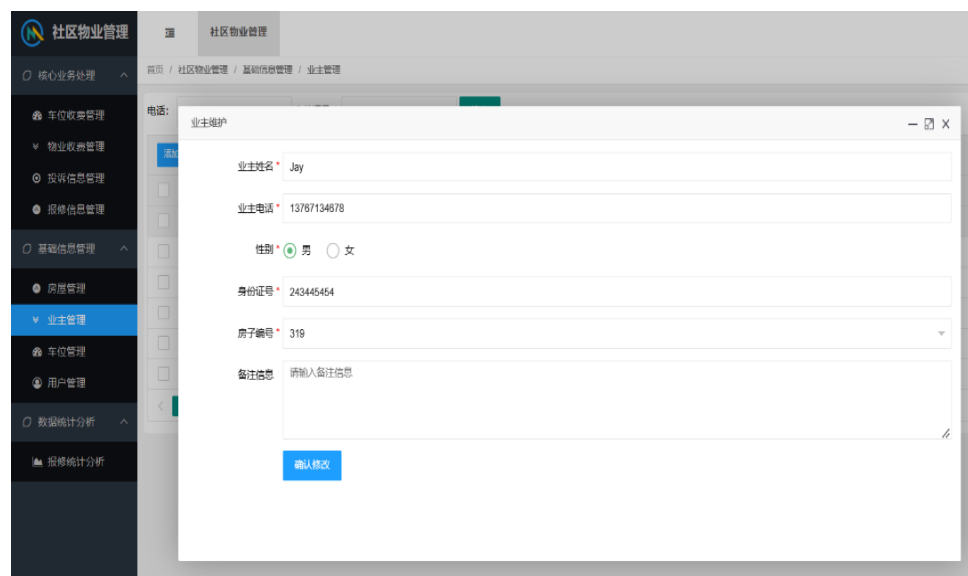


图 6.18 修改业主信息界面图

部分相关核心代码：

```
owner.setPassword("123456");
Userinfouser=newUserinfo();
user.setPassword("123456");
user.setRemarks(owner.getRemarks());
user.setType(0);
user.setUsername(owner.getUsername());
userinfoService.add(user);
```

6.7 车位信息模块

管理员用于查看业主车位得具体信息，管理员车位信息列表如图 6.19 所示：

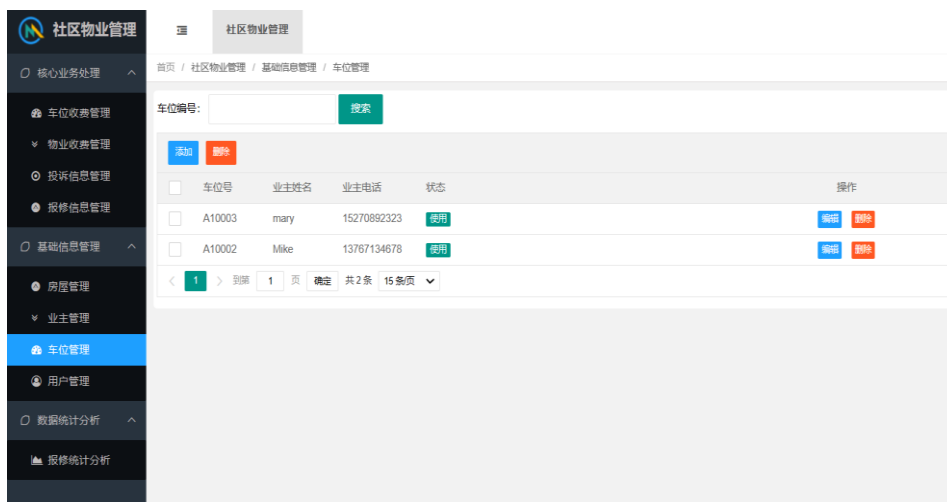


图 6.19 车位信息列表界面图

管理员填写车位编号、车位关联业主，点击确认添加，管理员添加车位信息如图 6.20 所示：

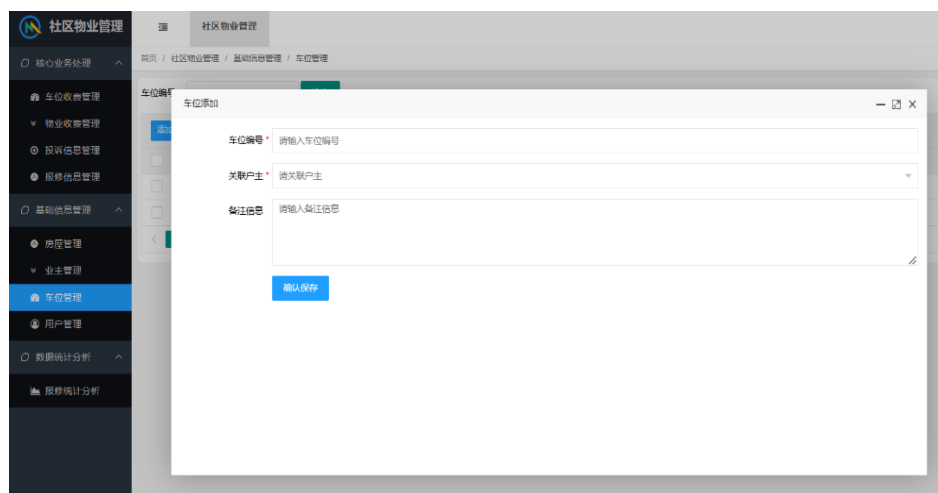


图 6.20 添加车位信息界面图

管理员填写车位编号、车位关联业主，点击确认添加，管理员添加车位信息如图 6.21 所示：

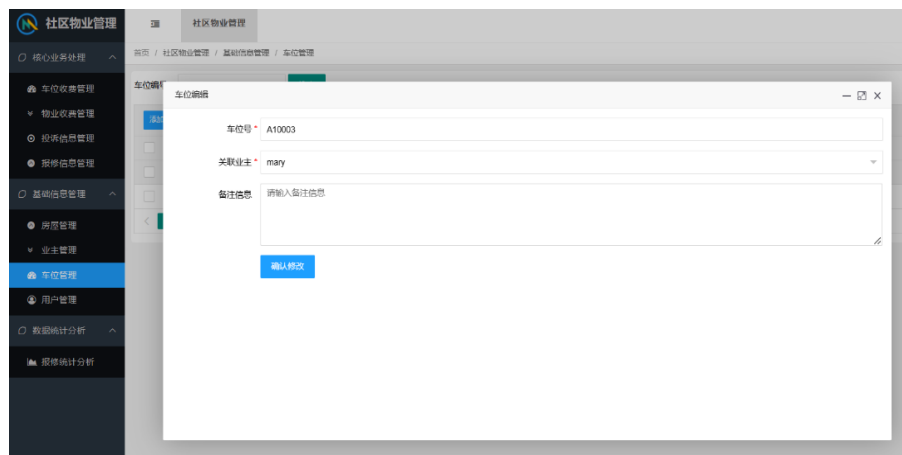


图 6.21 修改车位信息界面图

部分相关核心代码：

```
if(parking.getOwnerId()!=null){//添加车位信息
    parking.setStatus(1);
}else{
    parking.setStatus(0);
}
intnum=parkingService.add(parking);
if(num>0){
    returnR.ok();
}else{
    returnR.fail("添加失败");
}
```

6.8 用户信息模块

管理员用于查看用户信息，以及用户权限，管理员用户信息列表如图 6.22 所示：

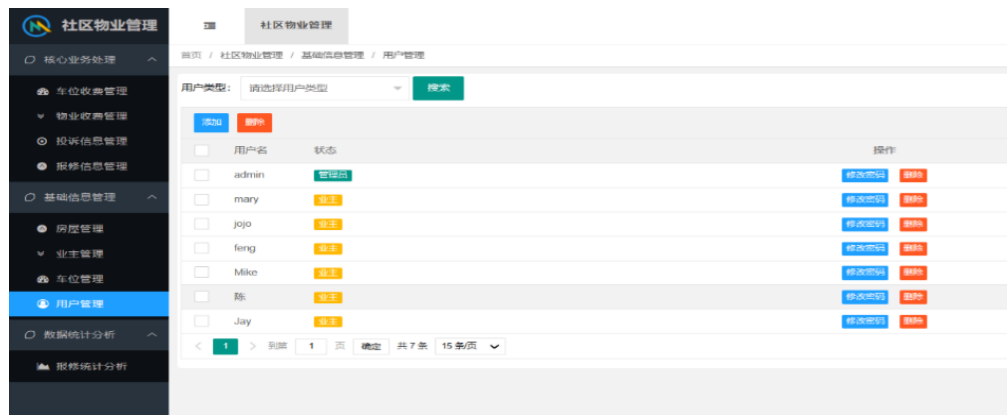


图 6.22 用户信息列表界面图

管理员用于添加管理员，填写管理员登录名称和密码，点击确认添加，管理员用户信息列表如图 6.23 所示：

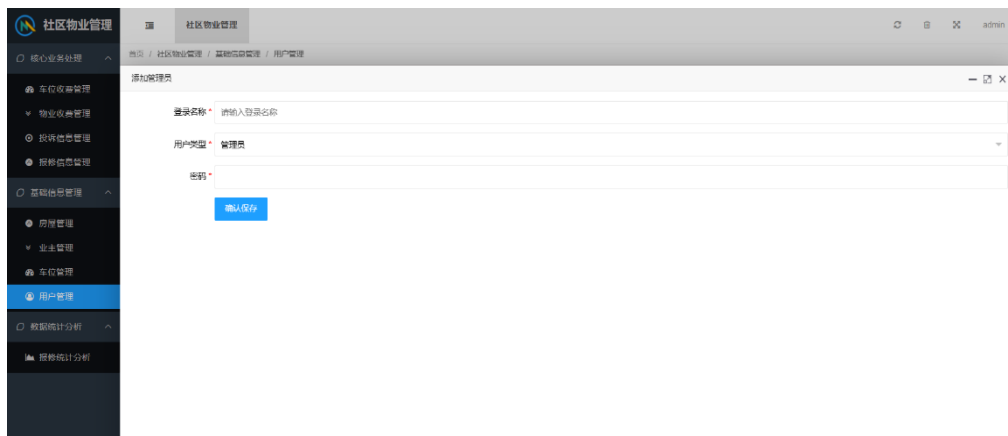


图 6.23 添加管理员信息界面图

管理员用于修改用户密码，填写新旧密码和确认密码，点击确认修改，管理员用户信息列表如图 6.24 所示：

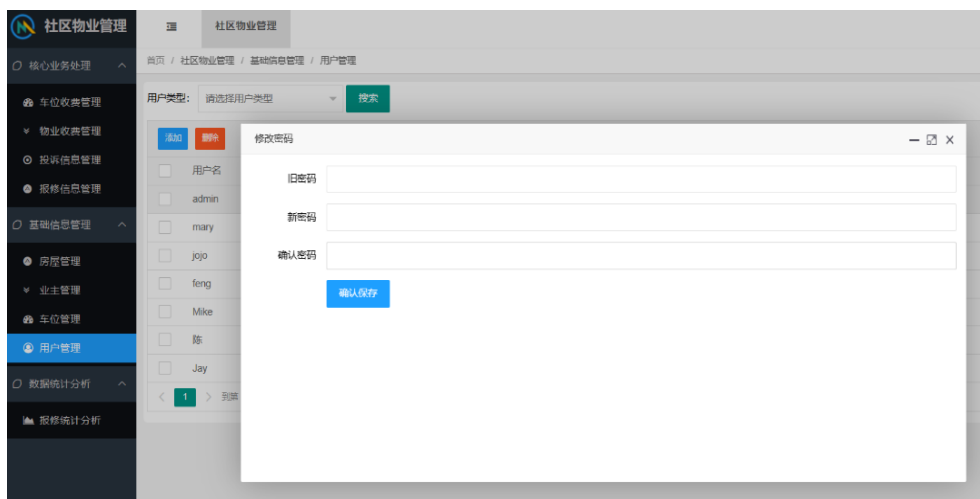


图 6.24 修改用户密码界面图

部分相关核心代码:

```
JsonObject object = new JsonObject ()
object.setCode (0);
object.setMsg ("ok");
object.setCount (pageInfo.getTotal ());
object.setData (pageInfo.getList ());
return object;}
```

6.9 数据统计模块

该模块用于统计业主各种报修的数量。管理员可以查看业主报修的各种类型的数量，管理员查看报修统计如图 6.25 所示：

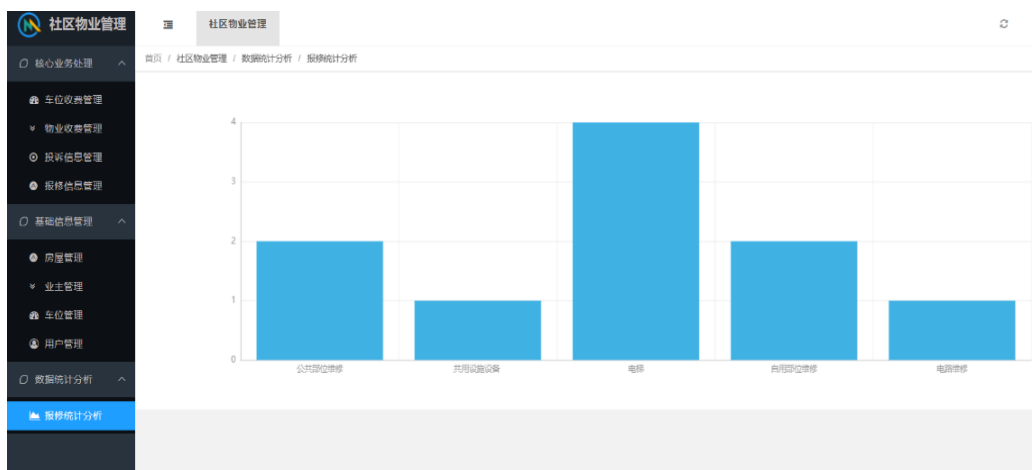


图 6.25 报修统计分析界面图

部分相关核心代码:

```
SELECT count (info. id) AS counts, type. NAME//查询报修数量的 sql 语句
FROM
`repair` info
LEFT JOIN repairtype type ON type. id=info. com_id
GROUP BY
info. com_id
```

6.10 访问拦截与权限控制模块

访问拦截与权限控制模块是本系统的安全性保证，此模块保证业主不能访问管理员界面，管理员不能访问业主界面，实现接口隔离，防止越权操作。此模块依赖登陆表，用户登录界面如图 6.26 所示：

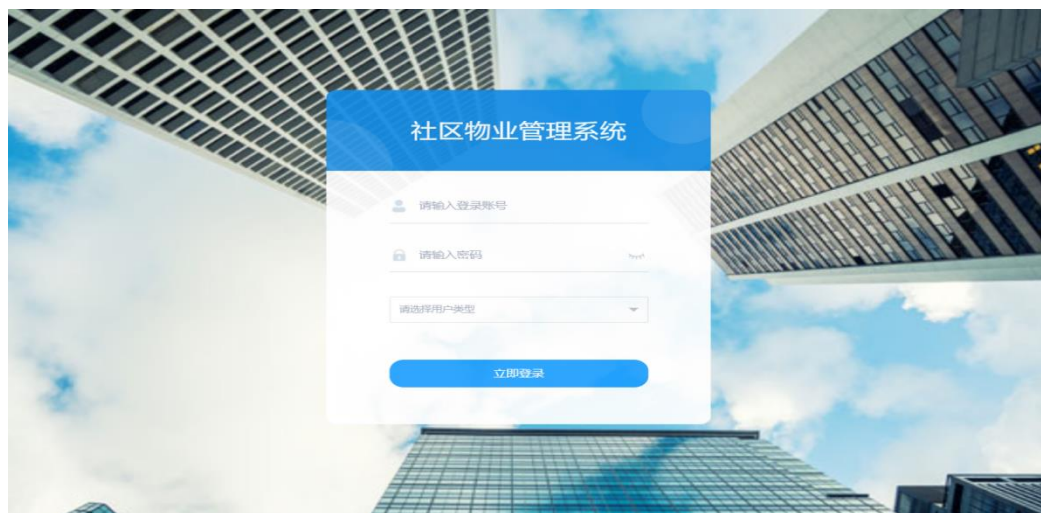


图 6.26 访问拦截与权限控制界面图

相关 controller 层代码：

```
{if(user==null) { //用户名密码判断
map.put("code", 404);
map.put("msg", "用户名或者密码错误");
return map; } else {
session.setAttribute("user", user);
map.put("code", 200);
map.put("user", user);
map.put("username", user.getUsername());
return map; }}
```

6.11 本章小结

本章节主要介绍了光明社区物业管理系统各个模块的功能的实现，展示了光明社区物业管理系统的运行结果，针对系统的各个模块的功能讲述管理员和业主对系统的可用功能，可以对本系统进行充分的了解。

7 系统测试

7.1 测试概述

系统测试是为了发现错误而执行程序的过程，希望能以最少的人力和时间发现潜在的各种错误和缺陷。应根据开发各阶段的需求、设计等文档或程序的内部结构精心设计测试用例，并利用这些实例来运行程序，以便发现错误。将通过确认测试的软件，作为整个基于计算机系统的一个元素，与计算机硬件、外设、某些支持软件、数据和人员等其他系统元素结合起来，在实际运行（使用）环境下，对计算机系统进行的测试。是为了发现缺陷并度量产品质量，对功能进行测试。

7.2 系统测试用例

7.2.1 访问拦截与权限控制模块

通过输入用户名和密码进行测试访问拦截与权限控制模块，测试结果如表 7.1 所示：

表 7.1 访问拦截与权限控制模块测试

功能点	验证点	具体步骤	测试结果
系统登录页面	验证是否可以正常进入系统登录页面	1. 打开浏览器，输入前台的登录地址： <code>http://localhost:8888</code>	成功跳转到系统的登录页面
登录权限验证	验证未选择登录类型是否登录成功	1. 在登录页面上输入用户账号：“admin”和密码：“12345”登录，未选择类型。	登录失败，提示类型不能为空
登录验证	验证错误的用户名是否登录成功	1. 在登录页面上输入用户账号：“feng”和密码：“12345”登录，选择管理员类型。	登录失败，提示用户名或者密码错误
正常登录验证	验证是否登录成功	1. 在登录页面上输入用户账号：“admin”和密码：“12345”登录，选择管理员类型。	成功跳转到管理员页面
退出系统验证	验证是否退出用户界面成功	1. 点击用户姓名的退出系统	退出成功，返回登录界面

7.2.2 车位收费模块

通过输入日期信息进行测试车位收费模块，测试结果如表 7.2 所示：

表 7.2 车位收费模块测试

功能点	验证点	具体步骤	测试结果
添加车位收费信息验证	验证是否可以添加车位收费信息	1. 在管理员车位收费页面，点击添加数据，输入相应的日期：“2022-12-12”和“2022-11-12” 金额：“100”	车位收费信息添加成功
缴费处理验证	验证是否对未缴费的缴费成功	1. 管理员在车位收费页面对未缴费的，点击缴费	缴费成功，未缴费状态变成已缴费
删除信息验证	验证是否删除信息成功	1. 管理员在车位收费页面，点击删除	删除成功，刷新页面
查询信息验证	验证是否查询成功	1. 管理员在车位收费页面，输入车位号：“A10001”和选择“未缴费” 点击查询	查询成功，显示查询结果

7.2.3 物业收费模块

通过输入日期信息进行测试物业收费模块，测试结果如表 7.3 所示：

表 7.3 物业收费模块测试

功能点	验证点	具体步骤	测试结果
添加物业收费信息验证	验证是否可以添加物业收费信息	1. 在管理员物业收费页面，点击添加数据，输入相应的日期：“2022-12-12”和“2022-11-12”	物业收费信息添加成功
缴费处理验证	验证是否对未缴费的缴费成功	1. 管理员在物业收费页面对未缴费的，点击缴费	缴费成功，未缴费状态变成已缴费
删除信息验证	验证是否删除信息成功	1. 管理员在物业收费页面，点击删除	删除成功，刷新页面
查询信息验证	验证是否查询成功	1. 管理员在物业收费页面，输入房屋号：“319”	查询成功，显示查询结果

7.2.4 投诉信息模块

通过对选择投诉类型进行测试投诉信息模块，测试结果如表 7.4 所示

表 7.4 投诉信息模块测试

功能点	验证点	具体步骤	测试结果
业主添加投诉信息验证	验证是否可以添加投诉信息	1. 在业主投诉管理页面，点击添加投诉信息，选择投诉类型为“垃圾乱放”	投诉信息添加成功

续表 7.4

功能点	验证点	具体步骤	测试结果
信息处理验证	验证是否对未处理的信息处理成功	1. 管理员在投诉信息页面对未处理的, 点击处理	处理成功, 未处理状态变成已处理
删除信息验证	验证是否删除信息成功	1. 管理员在投诉信息页面, 点击删除	删除成功, 刷新页面
查询信息验证	验证是否查询成功	1. 管理员在投诉页面, 选择投诉类型: “垃圾乱放” 和选择 “未处理”	查询成功, 显示查询结果

7.2.5 报修信息模块

通过选择报修类型进行测试报修信息模块, 测试结果如表 7.5 所示

表 7.5 报修信息模块测试

功能点	验证点	具体步骤	测试结果
业主添加报修信息验证	验证是否可以添加报修信息	1. 在业主投诉管理页面, 点击添加报修信息, 选择投诉类型为 “电梯”	报修信息添加成功
信息处理验证	验证是否对未处理的信息处理成功	1. 管理员在报修信息页面对未处理的, 点击处理	处理成功, 未处理状态变成已处理
删除信息验证	验证是否删除信息成功	1. 管理员在报修信息页面, 点击删除	删除成功, 刷新页面
查询信息验证	验证是否查询成功	1. 管理员在投诉页面, 选择投诉类型: “电梯” 和选择 “未处理”	查询成功, 显示查询结果

7.2.6 房屋信息模块

通过输入日期、房屋编号和面积进行测试房屋模块, 测试结果如表 7.6 所示:

表 7.6 房屋模块测试

功能点	验证点	具体步骤	测试结果
添加房屋信息验证	验证是否可以添加房屋信息	1. 在管理员房屋管理页面, 点击添加房屋信息, 填写的日期为 “2022-12-12”, 填写的房屋编号为 “918”, 填写的房屋面积为 “100”	房屋信息添加成功
删除信息验证	验证是否删除信息成功	1. 管理员在房屋信息页面, 点击删除	删除成功, 刷新页面

续表 7.6

功能点	验证点	具体步骤	测试结果
修改信息验证	验证是否修改成功	1. 管理员在房屋页面，点击编辑，填写的日期为“2022-12-12”，填写的房屋编号为“319”，填写的房屋面积为“80”	房屋信息修改成功
查询信息验证	验证是否查询成功	1. 管理员在房屋页面，输入房间号“319”	查询成功，显示查询结果

7.2.7 业主信息模块

通过输入姓名、房屋编号、电话、性别、身份证进行测试业主模块，测试结果如表 7.7 所示：

表 7.7 业主模块测试

功能点	验证点	具体步骤	测试结果
添加业主信息验证	验证是否可以添加业主信息	1. 在业主管理页面，点击添加业主信息，填写的业主姓名为“kiki”，填写的房屋编号为“918”，填写的电话为“13712645277”，选择性别为“男”填写的身份证为“6888888”	业主信息添加成功
删除信息验证	验证是否删除信息成功	1. 管理员在业主信息页面，点击删除	删除成功，刷新页面
查询信息验证	验证是否查询成功	1. 管理员在房屋页面，输入电话“13712645277”	查询成功，显示查询结果
修改信息验证	验证是否修改成功	在业主管理页面，点击添加业主信息，填写的业主姓名为“jojo”，填写的房屋编号为“918”，填写的电话为“13712645277”，选择性别为“男”填写的身份证为“6888888”	业主信息修改成功

7.2.8 车位信息模块

车位信息模块，测试结果如表 7.8 所示：

表 7.8 车位信息模块测试

功能点	验证点	具体步骤	测试结果
添加车位信息验证	验证是否可以添加车位信息	在管理员车位管理页面，点击添加信息，填写的车位编号为“A10003”，选择业主为“feng”	车位信息添加成功
删除信息验证	验证是否删除信息成功	管理员在车位信息页面，点击删除	删除成功，刷新页面
查询信息验证	验证是否查询成功	管理员在用户页面，输入车位编号：“A10002”	查询成功，显示查询结果
修改车位信息	验证是否修改成功	在管理员车位管理页面，点击添加信息，填写的车位编号为“A10002”，选择业主为“feng”	车位信息修改成功

7.2.9 用户信息模块

通过输入管理员登录名称和密码进行测试用户信息模块，测试结果如表 7.9 所示：

表 7.9 用户信息模块测试

功能点	验证点	具体步骤	测试结果
添加管理员信息验证	验证是否可以添加管理员信息	1. 在管理员用户管理页面，点击添加信息，填写的登录名称为“feng”，填写的密码为“12345”，	管理员信息添加成功
删除信息验证	验证是否删除信息成功	1. 管理员在用户信息页面，点击删除	删除成功，刷新页面
查询信息验证	验证是否查询成功	1. 管理员在用户页面，选择用户类型：“管理员”	查询成功，显示查询结果
修改密码验证	验证是否修改成功	1. 管理员在用户页面，点击修改密码，填写的旧密码为“12345”，填写的新密码为“123456”，填写的确认密码为“123456”	密码修改成功

7.2.10 数据统计模块

通过增加和删除报修信息进行测试数据统计息模块，测试结果如表 7.10 所示：

表 7.10 数据统计模块测试

功能点	验证点	具体步骤	测试结果
报修类型数量信息验证	验证是否可以显示报修类型的数量	业主添加“电梯”报修类型信息	显示“电梯”报修类型数量增加
报修类型数量信息验证	验证是否可以显示报修类型的数量	管理员删除“电梯”报修类型信息	显示“电梯”报修类型数量减少

7.3 测试结论

根据上面的测试用例对光明社区物业管理系统的功能模块进行测试，该系统的功能模块在设计限制条件下能够正常运行，可以对数据表中的数据进行显示，可以对各个模块进行的添加、删除、修改、查询操作，显示结果为正确。通过权限对用户登录进行限制，未登录的用户和没有相应权限的用户无法跳转到相应的用户页面。总而言之，光明社区物业管理系统的正常使用和运行，无明显的 BUG 和错误。

7.4 小结

光明社区物业管理系统的功能测试后，并未发现明显或严重的问题，在测试过程中的一些小问题也都被及时发现并解决。在应对非法操作时，系统也能及时阻止并抛出异常，同时用户在发起合法请求时，系统能够迅速响应并返回正确结果。但本次测试仍有一些缺点需要后续完善，比如本次测试全过程都是人工测试，人工测试难以覆盖系统测试的各个方面，也可能会有遗漏问题暂未发现。

8 总结和展望

8.1 总结

本文介绍了光明社区物业管理系统的开发过程，该系统采 SpringBoot 框架进行开发，该系统基本上体现了社区物业管理信息化的功能，借助该系统社区物业可以有效的提高对社区的管理效率，使业主可以用在网上就了解自己所产生的费用，系统对投诉信息和报修信息的添加和处理能够进行有效的管理，这使得管理员可以在对物业管理的过程中可以高效的投诉和报修进行处理，这也减少了管理人员的工作量，提高业主的服务体验。充分体现了该系统的物业管理信息化的便捷和快捷。

本论文是对光明社区物业管理系统的详细设计实现的详细介绍，展示本系统制作的具体内容。在编写本文中对设计的系统进行一步步的解析，从系统的背景意义，具体功能，使用的相关技术等方面都进行了详细的阐述。

本文通过对系统的详细分析，全面的展示本系统的，尽管本系统基本上实现了功能的需求，但是还是存在一些问题和不足之处，还需要日后进行改善和完善。

8.2 展望

在本文的最后，本系统在实际应用中一些问题和缺陷，例如本系统设计的时候只考虑到网页端，没有考虑到移动端的开发。在测试方面不够全面也可能存在一些问题和缺陷暂时还未发现。此后会增加移动端的开发实现网页端的功能，并增加线上收费功能，以增加系统的实用性，还需进行全面的测试，减少系统出现的问题和漏洞。但是在当今社会背景下本系统还是具有很大的使用潜力的，会在今后的交流和学习中使两个角色用户完美的符合使用需求。

参考文献

- [1]胡贺. 智能小区物业管理系统的的设计实现[D]. 导师: 王志慧. 内蒙古大学, 2020
- [2] “加强物业管理共建美好家园” 典型案例[J]. 中国物业管理, 2022 (02):46-147.
- [3]殷闽华. 社区治理中的物业管理:服务质量对居民归属感的影响[J]. 东南学术, 2020 (03):162-171. DOI:10.13658/j.cnki.sar.2020.03.018.
- [4]张鸣春. 探寻城市社区物业管理的发展之道:研究综述与未来展望[J]. 复旦城市治理评论, 2021, (01):3-39.
- [5]闫浩, 李合明. 提高物业管理水平提升居民的幸福——基于河南省鹤壁市淇滨区物业管理工作的实践与思考[J]. 信访与社会矛盾问题研究, 2020, (06):69-80.
- [6]邓国胜, 程令伟. 物业管理融入城市社区治理的理论逻辑与路径创新[J]. 城市发展研究, 2021, 28 (09):87-91+124.
- [7]侯金彪. 基于 JSP 的小区物业管理系统设计实现[J]. 陇东学院学报, 2020, (05):15-19.
- [8]Ooma Boosts Hotel Guest Experiences and Employee Productivity Through Expanded Integration with Property Management Systems[J]. Manufacturing Close - Up, 2022.
- [9]Whimstay Integrates with Streamline and HomeAway Software Property Management Systems[J]. Wireless News, 2019.
- [10]Condo Bridge Offers a Complete Property Management Solution[J]. M2 Presswire, 2021.
- [11]Saba Rock Resort Using Agilysys Property Management System and Point of Sale Solutions[J]. Manufacturing Close - Up, 2020.
- [12]戴亚峰. 基于 Javaweb 的住宅小区物业管理系统设计实现[D]. 导师: 杨勇; 黄集慧. 江西财经大学, 2018.
- [13]方水平. 智慧物业管理系统的的设计与实现[J]. 北京工业职业技术学院学报, 2020, (01):37-41.

- [14]赵越超. 基于 JavaEE 的小区物业管理系统设计与实现[J]. 物联网技术, 2020, (07):107-110.
- [15]潘冀松. 基于 J2EE 的物业管理信息系统的设计与实现[J]. 电子世界, 2019, (11):110-129.

致谢

转眼间大学的时间即将结束，毕业设计也接近了尾声，在这期间很多人都对我给予了帮助，在他们的无私帮助下我才能完成毕业设计。

本文是在蒋明奇老师的精心指导和大力支持下完成的。论文从选题到完成，得到了蒋老师的尽心的指导和热情帮助，在这个过程中蒋老师给了我很多珍贵的意见和建议。蒋老师严谨的学术态度、深厚的专业知识、卓越的精神给我留下了很深的印象，并对我的学习和工作做出了巨大贡献。在这里，特向蒋老师表示深深的尊重和感谢！

在此同时，大学四年在老师的激情的教学下，让我学会了很多的专业知识理论，这些知识理论在我论文写作的过程中发挥了极大的作用，使得我可以顺利完成系统的研究设计和论文的写作。感谢在大学四年学习生活中给与我帮助的老师 and 同学们。

最后，亲人和朋友们为我清除了生活上的各种麻烦，让我可以专心学习，感谢我的亲人朋友们，感谢他们给与我精神上 and 物质上的鼓励和支持，非常感激你们的帮助！