

**2018级综合实训课**

□嵌入式系统综合实训（计算机科学与技术专业学生适用）

☑软件开发综合训练（软件工程专业学生适用）

□网络工程综合训练（网络工程专业学生适用）

**课题研究报告**

**课题名称： 物业管理系统**

**专业班级及分组：软件工程专业6班第6组**

**学生团队成员：叶志坚、罗远彬、李嘉豪、魏兆庆、林健强**

**指导教师： 陈绪行**

**计算机科学与工程学院**

**2021年7月**

**综合实训课团队成员分工表(2018级)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 课  题  情  况 | 课题名称 | 物业管理系统 | | | | |
| 教师姓名 | 陈绪行 | 职称 | 讲师 | 学位 | 硕士 |
| 课程  类别 | □嵌入式系统综合实训（计算机科学与技术专业学生适用）  ☑软件开发综合训练（软件工程专业学生适用）  □网络工程综合训练（网络工程专业学生适用） | | | | |
| 团队成员分工安排 | **所在班级**：**2018级软件工程6班**  **队长：罗远彬**  **组员：****李嘉豪、魏兆庆、林健强、叶志坚**  **团队成员具体分工安排如下：（下文需根据项目的实际完成情况，对团队成员各自承担的任务进行详细描述，必须要细致，工作任务落实到每个人，不得简化叙述。）**  罗远彬任务内容:  1、编写需求分析的任务概述、需求规定  2、编写概要设计的总体设计  3、编写详细设计  4、编写数据库设计的外部设计  5、负责编码用户端的登录页面、主页面、商家页面及相关接口  李嘉豪任务内容：  1、编写需求分析的涉众分析、约束、性能需求  2、编写概要设计的接口设计  3、编写详细设计  4、编写数据库设计的结构设计  5、负责编码用户端的确认订单页面、订单页面、订单详情页面、搜索商家页面及相关接口  魏兆庆任务内容：  1、编写需求分析的接口需求、公司需求  2、编写概要设计的运行设计、系统数据结构设计、系统出错处理设计  3、编写详细设计  4、编写数据库设计的运用设计  5、负责编码商家端的注册页面、登录页面、忘记密码页面、主页面及相关接口  林健强任务内容：  1、编写项目简介  2、编写需求分析的系统用例、数据描述  3、编写详细设计的14-18  4、编写功能测试  5、负责编码商家端的商家信息页面、商家信息修改页面、添加餐品页面、餐品管理页面、商家订单页面及相关接口  6、负责整合全部代码，测试功能  叶志坚任务内容:  1.配合完成文档编写的部分工作，文档格式排版、文书说明等等  2.针对系统核心功能、设计测试用例并进行测试，写入测试文档  3.系统抗压力测试，配合完成系统接口压力测试工作  4.记录项目会议，跟进对接项目进度 | | | | | |

（续表）

**物业管理系统**

作者姓名：罗远彬、李嘉豪、魏兆庆、林健强、叶志坚 专业班级：18软件工程6班

指导教师：陈绪行

**摘 要**

当前社会是一个计算机普遍应用的社会，随着社会的不断发展和人们的生活水平的提高，人们逐渐步入的信息社会，同时，随着房地产行业的发展，住宅小区已称为居住的主流，但小区物业管理水平还相对落后，传统的人工方式难以应付大量的业主资料，复杂频繁的业务收费以及对各项资源的管理。小区物业管理系统就是为了为小区管理者更好的开展小区各项业务处理工作而开发的软件。

该项目是一个小区物业管理系统，目的是为了辅助物业管理公司进行小区的物业的管理工作正常运行。适用于中小型小区的物业管理系统，能够方便物业公司进行服务管理，可以在其中查询与缴纳水电费，停车费以及公共设施故障上报维修。

**关键词：**物业管理系统 SpringBoot

Food Ordering Platform

Author: Yuanbin Luo, Zhijian Ye, Jiahao Li, Zhaoqin Wei, Jianqiang Lin

Specialty: Software Engineering Tutor: Xuhang Chen

**Abstract**

The present era is one in which computers are extensively utilized. People are progressively moving into the information society as society continues to evolve and people's living circumstances improve. Simultaneously, the residential community has been dubbed the "mainstream of housing" due to the development of the real estate industry, but the level of property management in the community is still relatively low; the traditional manual method is difficult to cope with a large number of owner data, complex and frequent business charges, and the management of various resources. The community property management system is software designed to help community administrators better manage their community's business processes.

The project is a community property management system, with the goal of assisting the property management firm with the day-to-day operations of community property management. The property management system, which is ideal for small and medium-sized residential areas, may simplify property company service administration by allowing you to enquire about and pay water and energy costs, parking fees, and public facility failure reports and repair.

**Keywords：**Property Management System SpringBoot

**目录**

[1 绪论 1](#_Toc102550110)

[1.1 研究背景 1](#_Toc102550111)

[1.2 研究意义 1](#_Toc102550112)

[1.3 系统概述 1](#_Toc102550113)

[2 关键技术介绍 2](#_Toc102550114)

[2.1 spring框架 2](#_Toc102550115)

[2.1.1 spring框架简介 2](#_Toc102550116)

[2.1.2 特征 2](#_Toc102550117)

[2.2 相关技术栈简介 2](#_Toc102550118)

[3 需求分析 4](#_Toc102550119)

[3.1任务概述 4](#_Toc102550120)

[3.1.1目标用户分析 4](#_Toc102550121)

[3.1.2 应用环境 4](#_Toc102550122)

[3.2环境规定 5](#_Toc102550123)

[4 系统分析 6](#_Toc102550124)

[4.1 系统概述 6](#_Toc102550125)

[4.2系统处理流程 6](#_Toc102550126)

[4.3 系统结构 8](#_Toc102550127)

[5 系统设计 9](#_Toc102550128)

[5.1程序系统的结构 9](#_Toc102550129)

[5.2用户授权登录设计说明 10](#_Toc102550130)

[5.2.1程序描述 10](#_Toc102550131)

[5.2.2功能 10](#_Toc102550132)

[5.2.3性能 10](#_Toc102550133)

[5.2.4输入项 10](#_Toc102550134)

[5.2.5输出项 11](#_Toc102550135)

[5.2.6接口 11](#_Toc102550136)

[5.2.7存储分配 11](#_Toc102550137)

[5.2.8注释设计 11](#_Toc102550138)

[5.2.9限制条件 11](#_Toc102550139)

[5.3浏览餐品设计说明 11](#_Toc102550140)

[5.3.1程序描述 11](#_Toc102550141)

[5.3.2功能 11](#_Toc102550142)

[5.3.3性能 11](#_Toc102550143)

[5.3.4输入项 11](#_Toc102550144)

[5.3.5输出项 12](#_Toc102550145)

[5.3.6接口 12](#_Toc102550146)

[5.3.7存储分配 12](#_Toc102550147)

[5.3.8注释设计 12](#_Toc102550148)

[5.4选购餐品设计说明 12](#_Toc102550149)

[5.4.1程序描述 12](#_Toc102550150)

[5.4.2功能 12](#_Toc102550151)

[5.4.3性能 12](#_Toc102550152)

[5.4.4输入项 12](#_Toc102550153)

[5.4.5输出项 13](#_Toc102550154)

[5.4.6接口 13](#_Toc102550155)

[5.4.7存储分配 13](#_Toc102550156)

[5.4.8注释设计 13](#_Toc102550157)

[5.5取消选购设计说明 13](#_Toc102550158)

[5.5.1程序描述 13](#_Toc102550159)

[5.5.2功能 13](#_Toc102550160)

[5.5.3性能 13](#_Toc102550161)

[5.5.4输入项 13](#_Toc102550162)

[5.5.5输出项 14](#_Toc102550163)

[5.5.6接口 14](#_Toc102550164)

[5.5.7存储分配 14](#_Toc102550165)

[5.5.8注释设计 14](#_Toc102550166)

[5.6结算支付设计说明 14](#_Toc102550167)

[5.6.1程序描述 14](#_Toc102550168)

[5.6.2功能 14](#_Toc102550169)

[5.6.3性能 14](#_Toc102550170)

[5.6.4输入项 14](#_Toc102550171)

[5.6.5输出项 15](#_Toc102550172)

[5.6.6接口 15](#_Toc102550173)

[5.6.7存储分配 15](#_Toc102550174)

[5.6.8注释设计 15](#_Toc102550175)

[5.6.9限制条件 15](#_Toc102550176)

[5.7用户查询订单设计说明 15](#_Toc102550177)

[5.7.1程序描述 15](#_Toc102550178)

[5.7.2功能 15](#_Toc102550179)

[5.7.3性能 15](#_Toc102550180)

[5.7.4输入项 15](#_Toc102550181)

[5.7.5输出项 16](#_Toc102550182)

[5.7.6接口 16](#_Toc102550183)

[5.7.7存储分配 16](#_Toc102550184)

[5.7.8注释设计 16](#_Toc102550185)

[5.7.9限制条件 16](#_Toc102550186)

[5.8搜索商家设计说明 16](#_Toc102550187)

[5.8.1程序描述 16](#_Toc102550188)

[5.8.2功能 16](#_Toc102550189)

[5.8.3性能 16](#_Toc102550190)

[5.8.4输入项 17](#_Toc102550191)

[5.8.5输出项 17](#_Toc102550192)

[5.8.6接口 17](#_Toc102550193)

[5.8.7存储分配 17](#_Toc102550194)

[5.8.8注释设计 17](#_Toc102550195)

[5.9商家注册登录设计说明 17](#_Toc102550196)

[5.9.1程序描述 17](#_Toc102550197)

[5.9.2功能 17](#_Toc102550198)

[5.9.3性能 17](#_Toc102550199)

[5.9.4接口 18](#_Toc102550200)

[5.9.5存储分配 18](#_Toc102550201)

[5.9.6注释设计 18](#_Toc102550202)

[6 平台实现 18](#_Toc102550203)

[6.1 运行环境及开发工具 18](#_Toc102550204)

[6.2 系统界面 18](#_Toc102550205)

[7 系统测试 22](#_Toc102550206)

[7.1测试概要 22](#_Toc102550207)

[7.1.1测试目的 22](#_Toc102550208)

[7.1.2测试环境 22](#_Toc102550209)

[7.1.3测试执行 22](#_Toc102550210)

[7.2 测试结论 22](#_Toc102550211)

[8 结论与展望 23](#_Toc102550212)

[8.1 结论 23](#_Toc102550213)

[8.2 展望 23](#_Toc102550214)

[参考文献 24](#_Toc102550215)

[附录 25](#_Toc102550216)

# 1 绪论

## 研究背景

当前社会是一个计算机普遍应用的社会，随着社会的不断发展和人们的生活水平的提高，人们逐渐步入的信息社会，同时，随着房地产行业的发展，住宅小区已称为居住的主流，但小区物业管理水平还相对落后，传统的人工方式难以应付大量的业主资料，复杂频繁的业务收费以及对各项资源的管理。小区物业管理系统就是为了为小区管理者更好的开展小区各项业务处理工作而开发的软件。

## 研究意义

目前全国物业管理面积145.3亿㎡。 现在全国物业服务企业共11.8万家，全国物业管理从业人员约904.7万人。平均每个人需要管理1607平方米的面积，以每户一百平方米计算需要每人需要对接160名业主，但考虑到不同工作岗位的分工需要管理更多的单位。其次，根据恒大研究院院长任泽平的估算，预计2018-2030年中国城镇年均新增住房需求在11亿-13亿平方米。这代表着物业管理系统将会有巨大的空间可以发展。

## 1.3 系统概述

民以食为天，本项目的目标是致力于提供一个校园订餐平台，主要的服务对象是广大的学生和校园餐厅商户。一方面，协助校园餐厅商户改善运营方式，提高运营效率和服务质量。另一方面，服务于高校学生，为学生提供便捷的订餐方式，减少学生点餐后等待取餐的时间，优化学生的用餐体验。

该项目是一个小区物业管理系统，目的是为了辅助物业管理公司进行小区的物业的管理工作正常运行。适用于中小型小区的物业管理系统，能够方便物业公司进行服务管理，可以在其中查询与缴纳水电费，停车费以及公共设施故障上报维修。

# 2 关键技术介绍

本系统是基于spring框架进行设计的，因而有必要知道什么是spring框架，有什么优缺点,并了解一些相关的技术栈。

## 2.1 spring框架

### 2.1.1 spring框架简介

Spring框架是[Java平台](https://baike.baidu.com/item/Java%E5%B9%B3%E5%8F%B0/3793459" \t "https://baike.baidu.com/item/Spring%20Boot/_blank)上的一种开源应用框架，提供具有控制反转特性的容器。尽管Spring框架自身对编程模型没有限制，但其在Java应用中的频繁使用让它备受青睐，以至于后来让它作为[EJB](https://baike.baidu.com/item/EJB/144195" \t "https://baike.baidu.com/item/Spring%20Boot/_blank)（EnterpriseJavaBeans）模型的补充，甚至是替补。Spring框架为开发提供了一系列的解决方案，比如利用控制反转的核心特性，并通过依赖注入实现控制反转来实现管理对象生命周期容器化，利用面向切面编程进行声明式的事务管理，整合多种持久化技术管理数据访问，提供大量优秀的Web框架方便开发等等。

### 2.1.2 特征

SpringBoot所具备的特征有：

（1）可以创建独立的[Spring](https://baike.baidu.com/item/Spring/85061" \t "https://baike.baidu.com/item/Spring%20Boot/_blank)应用程序，并且基于其Maven或Gradle插件，可以创建可执行的JARs和WARs；

（2）内嵌Tomcat或Jetty等Servlet容器；

（3）提供自动配置的“starter”项目对象模型（POMS）以简化[Maven](https://baike.baidu.com/item/Maven/6094909" \t "https://baike.baidu.com/item/Spring%20Boot/_blank)配置；

（4）尽可能自动配置Spring容器；

（5）提供准备好的特性，如指标、健康检查和外部化配置；

（6）绝对没有代码生成，不需要XML配置。

## 2.2 相关技术栈简介

技术栈：springboot、mysql、spring sercurity、mybatis、thymeleaf、Bootstrap、 Redis、layui、ActiveMQ、MongoDB、editorMarkDown

* Web端使用thmeleaf、bootstrap来实现响应式页面渲染、后端使用mybatis来实现数据持久化和PageHelper实现数据分页，中间件Redis、ElasticSearch、kafka来集成加强项目功能
* 使用Spring Security 做权限控制，替代拦截器的拦截控制，并使用自己的认证方案替代Security 认证流程，使权限认证和控制更加方便灵活。
* 使用Redis进行数据缓存，缓解数据库压力，使用Redis的数据类型set实现点赞，zset实现关注，并使用Redis存储ticket和验证码。
* 使用Redis高级数据类型HyperLogLog统计访问量,使用Bitmap统计网站日流量。
* 使用Elasticsearch做全局搜索
* 使用editorMarkDown实现富文本编辑器
* 使用Quartz定时同步缓存数据。
* 期望优化使用Kafka处理发送评论、点赞和关注等系统通知，构建异步消息系统。

# 3 需求分析

## 3.1任务概述

### 3.1.1目标用户分析

用户是分为管理员身份与业主身份，一个管理员身份，是以物业公司的员工为主体，具有处理多种事务的管理系统，公司员工需要处理业主申请的服务以及管理业主的消费

另一种是业主身份，用户不需要多少复杂的功能，只需要能将系统满足自己的日常需要即可。

对于业主而言，则是可以在该系统上进行停车费、物业管理费以及水电费的缴纳、投诉以及设施保修。

对于管理员则是需要管理业主的账单是否有异常，以及对业主的投诉以及维修申请进行处理答复。此外，还要对楼宇，车位进行管理。最后，管理员还能对业主发送的消息。

### 3.1.2 应用环境

CPU：8核心cpu，两颗，主频2.0GZ以上

内存：32GB

存储：高速存储，250G，普通存储，500G及以上

软件运行环境：

操作系统：Linux / Windows 7及以上

支持环境：JAVA（JDK1.8以上） Tomcat6以上

数 据 库：Mysql 5.6 / JDBC及以上

## 3.2环境规定

系统环境技术如表3-3-1所示

表3-3-1 系统环境

|  |  |
| --- | --- |
| ***工具*** | ***版本号*** |
| JDK | 1.8 |
| Mysql | 5.7 |
| Redis | 3.2 |
| layui |  |
| AticeMQ | 5.10.0 |
| MongoDB | 4.4.6 |

支持环境：JAVA（JDK1.8以上） Tomcat9以上

开发软件：idea（2020）

# 4 系统分析

## 4.1 系统概述

系统的逻辑范围是，物业公司的员工进行社区物业管理，业主能在其上进行费用缴纳，意见反馈、投诉、申请维修。

系统的物理范围包括服务器，数据库，网页前端前台与后台，目前是用户在仅在web端进行交互。

该物业管理系统采用C/S结构，后端，数据库会在服务器上运行，前端的业主视图与管理员视图会在客户机的浏览器上运行。我们团队开发的物业管理系统可从管理员端与用户端将整个系统体系划分为两个子系统。

订餐小程序旨在帮助用户建立一个基于互联网的网络订餐平台，商家能够通过有效的方式，将自己的商品发布到订餐小程序系统中；而用户能够通过浏览、关键字检索等手段，获取自己所喜爱的餐品，然后进行预约。

其中，管理员子系统又可以再划分为核心业务管理，基础业务管理，数据统计分析3个功能模块，其中核心业务管理功能模块提供车位收费管理，物业收费管理，投诉信息管理，报修信息管理4个基础功能模块；基础业务管理功能模块提供楼宇管理，房屋管理，业主管理，车位管理，抄表管理，管理员管理6个基础功能模块。数据统计分析功能模块提供报修统计分析基础功能模块。

下文将通过流程图，详细描述。

## 4.2系统处理流程

1. **车位收费管理流程**

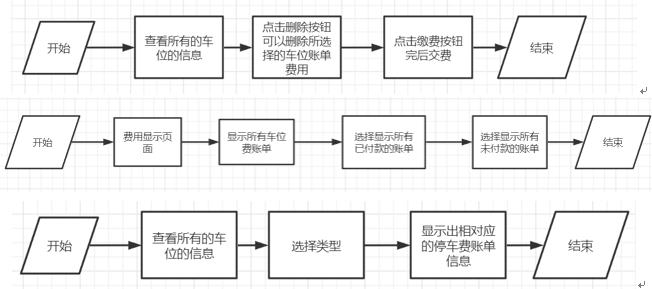


图4-2-1 车位收费管理流程图

其流程包括以下步骤：

1. 查询：管理员可以在车位费管理页面进行查询所有的业主的车停费账单信息，然后所有的
2. 处理：管理员可以对所有未缴费的停车费账单记录点击缴费按钮，对应的停车费账单的状态将会被改为已缴费
3. 删除：管理员可以删除调所有的停车费账单记录
4. **物业收费管理流程**

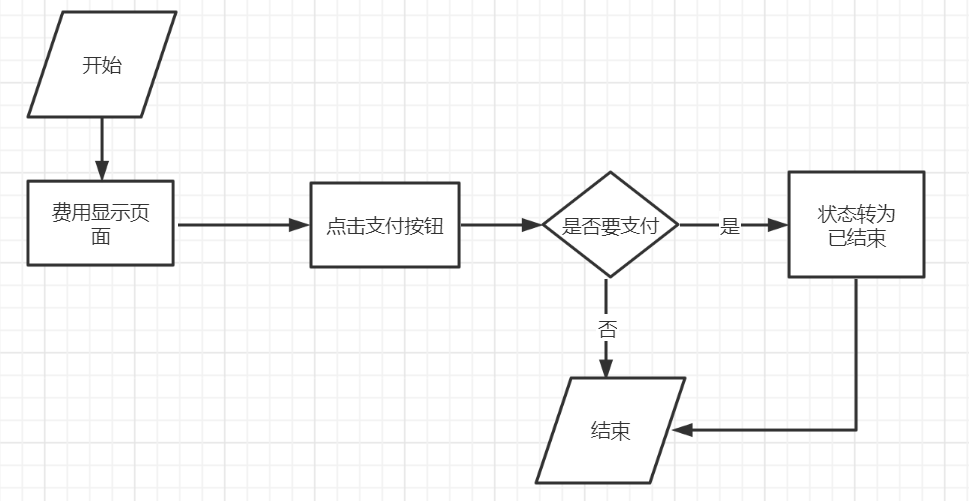
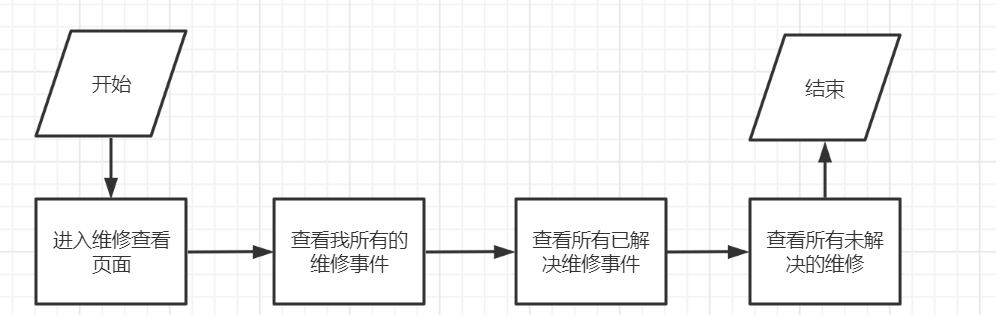


图4-2-2 物业收费管理流程图

其流程包括以下步骤：

（1）查看功能：查看所有的物业费用账单，或者按照已缴费和未交费为条件查看符合条件的物业费用账单。

（2）缴费功能：管理员手动点击缴费按钮进行缴费，完成后未交费的账单会变为已缴费状态。

（3）删除功能：点击删除按钮对所有被选中的物业费用账单信息进行删除。

1. **投诉信息管理流程**

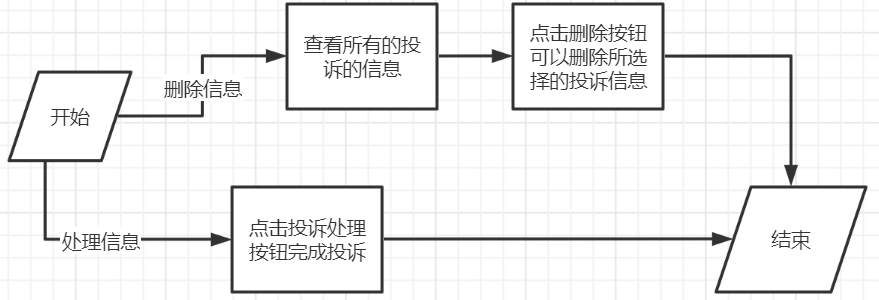
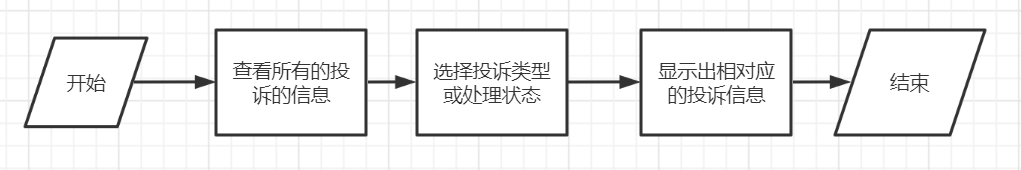


图4-2-3 投诉信息管理流程图

其流程包括以下步骤：

1. 投诉查看：管理员可以在投诉查看页面查看所有的投诉信息
2. 投诉处理：管理员点击完成完成投诉按钮后，该投诉即被完成
3. 投诉删除：管理员点击完成完成投诉按钮后，相应的投诉信息会被删除

## 4.3 系统结构

系统在软件使用模式上，为了方便用户使用，降低用户使用成本，采用B/S架构进行构建。同时，在系统软件架构上，采用MVC的分层模式对系统架构进行实现，确保系统安全稳定运行。其整体模块结构图如下：

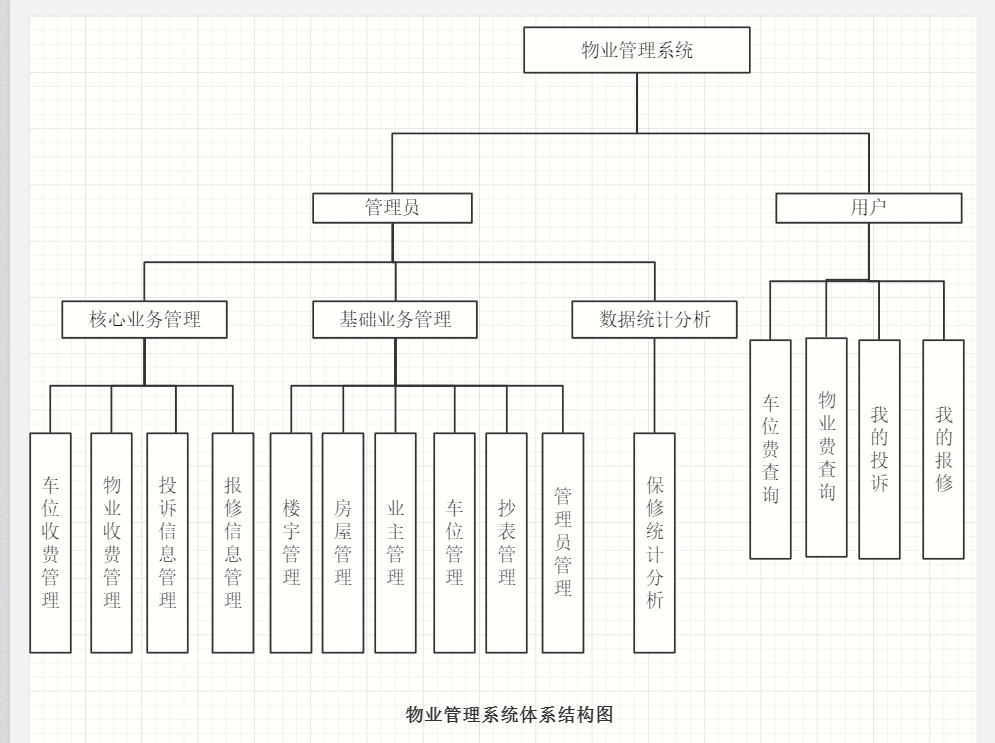


图4-3-1 系统模块结构图

整个系统分为五个层级：用户界面层、功能服务层，软件实现层，基础软件层和硬件服务层。

其中，功能服务层通过浏览器或微信小程序，对各类不同用户，提供不同的功能使用。

软件实现层提供了软件的MVC实现，它在每个层级都采用标准技术来完成层次的构建，并且因为层级存在，每个层次都可以根据需求不同进行调整而不影响其它层级运行。

基础软件层为系统的运行提供基本的环境，它包括软件运行的应用服务器中间件环境、JDK环境、数据库环境以及操作系统环境。

硬件服务层则为系统运行提供基本的硬件基础，它包括网络环境以及硬件服务器环境。

1）管理员子系统

在管理员使用流程中，管理员首先在浏览器打开该系统的登录首页，输入正确的账户信息进行登录，登录成功后进入物业管理系统后台，管理员可选择进行核心业务，基础业务或查看报修数据统计分析图。其中，选择进入核心业务管理页面后，可选择对车位费，物业费，投诉信息，报修信息进行查看或管理。选择进入基础业务管理页面后，可对楼宇信息，房屋信息，业主信息，抄表信息，管理员身份信息进行查看或管理。

2）用户子系统

在用户使用流程中，用户首先在浏览器打开该系统的登录首页，输入正确的账户信息进行登录，登录成功后进入系统，在系统中，用户可选择对车位费，物业费，我的投诉信息，我的报修信息进行查看或管理。

# 5 系统设计

## 5.1程序系统的结构

该博客系统采用多模块方式来开发，主要是一个主模块，里面包含多个子模块。每个子模块由页面包、公共包、数据库操作包、管理业务service包，实体类Model包组成，其包图如下

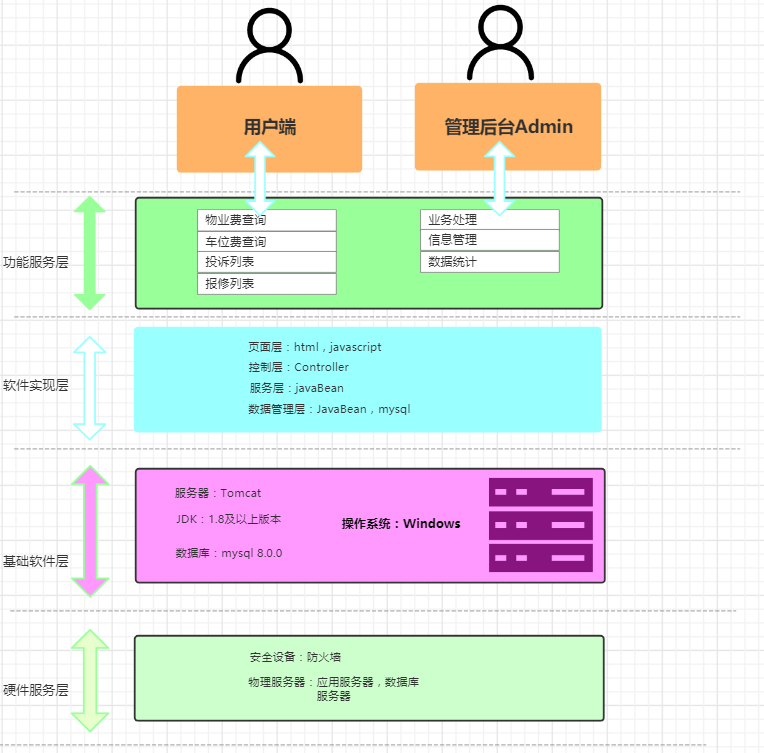


图5-1-1 系统结构图

## 5.2楼宇管理设计说明

### 5.2.1程序描述

用户能够查看楼宇信息，楼宇信息（ID，楼编号，单元，备注）。用户可选择性展示，导出，在线打印，分页等功能。除此之外能够进行高级查询，添加，删除，编辑楼宇信息功能等。

### 5.2.2功能

用户从楼宇列表界面开始，将输入的数据和指令，从页面通过http协议传送到IBuildingService的方法中，然后根据指令，将会使用具体业务方法get来完成数据的获取。

### 5.2.3性能

程序要求各功能用户发出操作指令后，能够在5秒内能返回操作结果。程序要求各层级之间的数据传输在0,1s内完成。

### 5.2.4输入项

楼宇编号、所属单元、备注信息

### 5.2.5输出项

操作成功/失败

### 5.2.6接口

本程序不使用外部接口。

### 5.2.7存储分配

本系统无存储描述。

### 5.2.8注释设计

本程序中安排的注释如下：

* 1. 加在类首部的注释；
  2. 加在方法首部的注释
  3. 加在各分枝点处的注释；
  4. 对各变量的功能、范围、缺省条件等所加的注释；
  5. 对使用的逻辑所加的注释等等。

### 5.2.9限制条件

暂无限制条件

## 5.3房屋管理设计设计说明

### 5.3.1程序描述

用户房屋进行管理，房屋信息（ID，房号，面积，入住时间，状态）本功能还支持用户选择性展示，导出，在线打印，分页等功能。除此之外能够进行高级查询，添加，删除，编辑房屋信息功能等。

### 5.3.2功能

本模块的核心程序如下分别为：

Dao层接口和实现：

HouseMapper：房屋高级查询接口，添加修改删除功能。

HouseMapper.xml：房屋查询实现功能

Service层接口和实现：

IHousService：分页查询接口，查询所有房屋。

HouseServiceImpl：分页查询实现功能。返回分页查询结果

Controller层：

HouseController：将查询结果包装Json数据返回页面。

### 5.3.3性能

程序要求各功能用户发出操作指令后，能够在5秒内能返回操作结果。程序要求各层级之间的数据传输在0,1s内完成。

### 5.3.4输入项

房屋编号、所属楼宇、房屋面积、入住时间、备注信息

### 5.3.5输出项

用户操作的反馈操作成功/失败

### 5.3.6接口

本程序不使用外部接口。

### 5.3.7存储分配

本程序无存储描述。

### 5.3.8注释设计

本程序中安排的注释如下：

* 1. 加在类首部的注释；
  2. 加在方法首部的注释
  3. 加在各分枝点处的注释；
  4. 对各变量的功能、范围、缺省条件等所加的注释；
  5. 对使用的逻辑所加的注释等等。

## 5.4业主管理设计说明

### 5.4.1程序描述

用户查询业主基本信息（ID，业主姓名，性别，联系方式，身份证号，房屋编号），以及编辑业主信息包括添加修改删除打印等。

### 5.4.2功能

本模块的核心程序如下分别为：

Dao层接口和实现：

OwnerMapper：业主高级查询接口，添加修改删除功能。

OwnerMapper.xml：业主查询实现功能

Service层接口和实现：

IOwnerService：分页查询接口，查询所有业主。

OwnerServiceImpl：分页查询实现功能。返回分页查询结果

Controller层：

OwnerController：将查询结果包装Json数据返回页面。

### 5.4.3性能

程序要求各功能用户发出操作指令后，能够在5秒内能返回操作结果。程序要求各层级之间的数据传输在0,1s内完成。

### 5.4.4输入项

姓名、电话、性别、身份证号、房子编号、备注信息

### 5.4.5输出项

用户操作的反馈信息。

### 5.4.6接口

本程序不使用外部接口。

### 5.4.7存储分配

本系统无存储描述。

### 5.4.8注释设计

本程序中安排的注释如下：

* 1. 加在类首部的注释；
  2. 加在方法首部的注释
  3. 加在各分枝点处的注释；
  4. 对各变量的功能、范围、缺省条件等所加的注释；
  5. 对使用的逻辑所加的注释等等。

## 5.5车位管理设计设计说明

### 5.5.1程序描述

用户车位进行管理（ID，车位号，业主姓名，业主电话，状态），还支持用户选择性展示，导出，在线打印，分页等功能。除此之外能够进行高级查询，添加，删除，编辑房屋信息功能等。

### 5.5.2功能

本模块的核心程序如下分别为：

Dao层接口和实现：

HouseMapper：车位高级查询接口，。

HouseMapper.xml：车位查询实现功能

Service层接口和实现：

IHouseService：分页查询接口，查询所有车位，添加修改删除功能。

HouseServiceImpl：添加修改删除及分页查询实现功能。返回结果

Controller层：

HouseController：将添加修改删除及查询结果包装Json数据返回页面。

### 5.5.3性能

程序要求各功能用户发出操作指令后，能够在5秒内能返回操作结果。程序要求各层级之间的数据传输在0,1s内完成。

### 5.5.4输入项

车位编号、关联户主、备注信息

### 5.5.5输出项

页面信息或提示。

### 5.5.6接口

本程序不使用外部接口。

### 5.5.7存储分配

本系统无存储描述。

### 5.5.8注释设计

本程序中安排的注释如下：

* 1. 加在类首部的注释；
  2. 加在方法首部的注释
  3. 加在各分枝点处的注释；
  4. 对各变量的功能、范围、缺省条件等所加的注释；
  5. 对使用的逻辑所加的注释等等。

## 5.6抄表管理设计说明

### 5.6.1程序描述

用户抄表信息管理（ID，费用类型，上次度数，本次度数，户主id，开始使用时间，截止时间，登记时间，登记人，备注） ，用户可选择性展示，导出，在线打印，分页等功能。除此之外能够进行高级查询，添加，删除，抄表信息功能等。

### 5.6.2功能

本模块的核心程序如下分别为：

Dao层接口和实现：

RecordsMapper：门派号抄表数据高级查询接口。

RecordsMapper.xml：门牌号抄表数据查询实现功能

Service层接口和实现：

IRecordsService：分页查询接口，查询所有门派抄表数据及增删。

RecordsServiceImpl：分页查询实现功能。返回分页查询结果以及增删结果

Controller层：

RecordsController：将查询结果及增删结果包装Json数据返回页面。

### 5.6.3性能

程序要求各功能用户发出操作指令后，能够在5秒内能返回操作结果。程序要求各层级之间的数据传输在0,1s内完成。

### 5.6.4输入项

房子编号、费用类型、数据值、抄表时间、抄表员、备注信息

### 5.6.5输出项

用户操作的反馈信息。

### 5.6.6接口

本程序不使用外部接口。

### 5.6.7存储分配

本系统无存储描述。

### 5.6.8注释设计

本程序中安排的注释如下：

* 1. 加在类首部的注释；
  2. 加在方法首部的注释
  3. 加在各分枝点处的注释；
  4. 对各变量的功能、范围、缺省条件等所加的注释；
  5. 对使用的逻辑所加的注释等等。

### 5.6.9限制条件

暂无限制条件

## 5.7车位收费设计说明

### 5.7.1程序描述

车位收费管理（ID，使用开始时间，使用截止时间，需要缴费的钱数，状态，所有者的id，备注，费用类型，车位id） 用户可选择性展示，导出，在线打印，分页等功能。除此之外能够进行高级查询，添加，删除，车位收费信息功能等。

### 5.7.2功能

本模块的核心程序如下分别为：

Dao层接口和实现：

CarchageMapper：车位收费高级查询接口。

CarchageMapper.xml：车位收费查询实现功能

Service层接口和实现：

ICarchargeService：分页查询接口，及增删查所有数据CarchargeServiceImpl：分页查询实现功能。返回分页查询结果以及增删查结果

Controller层：

CarchargeController：将查询结果及增删改查结果包装Json数据返回页面。

### 5.7.3性能

程序要求各功能用户发出操作指令后，能够在5秒内能返回操作结果。程序要求各层级之间的数据传输在0,1s内完成。

### 5.7.4输入项

用户的操作指令。

### 5.7.5输出项

用户操作的反馈信息。

### 5.7.6接口

本程序不使用外部接口。

### 5.7.7存储分配

本系统无存储描述。

### 5.7.8注释设计

本程序中安排的注释如下：

* 1. 加在类首部的注释；
  2. 加在方法首部的注释
  3. 加在各分枝点处的注释；
  4. 对各变量的功能、范围、缺省条件等所加的注释；
  5. 对使用的逻辑所加的注释等等。

### 5.7.9限制条件

暂无限制条件

## 5.8物业收费设计说明

### 5.8.1程序描述

物业收费管理（ID，使用开始时间，使用截止时间，需要缴费的钱数，状态，户主的id，备注，费用类型） ，用户可选择性展示，导出，在线打印，分页等功能。除此之外能够进行高级查询，添加，删除，物业收费信息功能等。

### 5.8.2功能

本模块的核心程序如下分别为：

Dao层接口和实现：

PropertyInfoMapper：房屋物业费用高级查询接口。

PropertyInfoMapper.xml：物业收费查询实现功能

Service层接口和实现：

IPropertyService：分页查询接口，及增删查所有数据

PropertyServiceImpl：分页查询实现功能。返回分页查询结果以及增删查结果

Controller层：

PropertyController：将查询结果及增删查结果包装Json数据返回页面。

### 5.8.3性能

程序要求各功能用户发出操作指令后，能够在5秒内能返回操作结果。程序要求各层级之间的数据传输在0,1s内完成。

### 5.8.4输入项

用户的操作指令。

### 5.8.5输出项

用户操作的反馈信息。

### 5.8.6接口

本程序不使用外部接口。

### 5.8.7存储分配

本系统无存储描述。

### 5.8.8注释设计

本程序中安排的注释如下：

* 1. 加在类首部的注释；
  2. 加在方法首部的注释
  3. 加在各分枝点处的注释；
  4. 对各变量的功能、范围、缺省条件等所加的注释；
  5. 对使用的逻辑所加的注释等等。

## 5.9商家注册登录设计说明

### 5.9.1程序描述

本模块主要在商家首次使用本系统时记录下商家的信息，作为商家以后信息查询和各项操作的依据。

### 5.9.2功能

商家在登录页面点击“登录”按钮上方的蓝色字体“点我注册为商家”进入商家在注册页面填写商家的信息后点击下方的“填写完毕”，前端验证填写的信息是否合法后，系统验证商家的账号是否已被注册，若是提示商家错误信息，若不是则保存商家信息并提示“注册成功”后跳转登录页面。

### 5.9.3性能

程序要求各功能用户发出操作指令后，能够在5秒内能返回操作结果。程序要求各层级之间的数据传输在0,1s内完成。

### 5.9.4接口

本程序不使用外部接口。

### 5.9.5存储分配

本系统无存储描述。

### 5.9.6注释设计

本程序中安排的注释如下：

* 1. 加在类首部的注释；
  2. 加在方法首部的注释
  3. 加在各分枝点处的注释；
  4. 对各变量的功能、范围、缺省条件等所加的注释；
  5. 对使用的逻辑所加的注释等等。

# 6 平台实现

## 6.1 运行环境及开发工具

服务器：基于Intel构架的企业服务器

支持环境：JAVA（JDK1.8以上） Tomcat9以上

数 据 库：MySQL5.7以上

编程语言：Java,html,js

设计工具：intellij idea、redis、powerdesign、archi等

## 6.2 系统界面

1） 主页面

主页面主要显示楼宇管理列表。界面如图6-1所示。

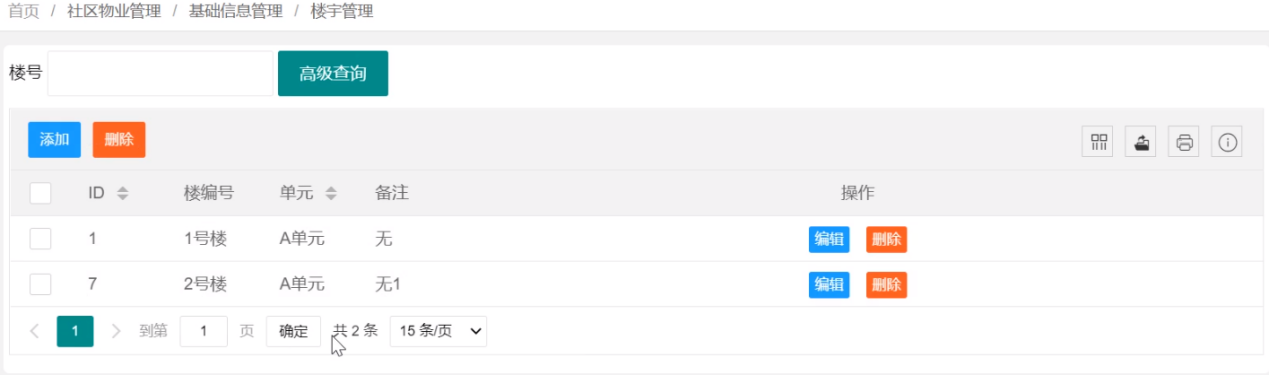


图6-1 用户主界面

1. 房屋管理界面

房屋管理页面中显示房屋信息。界面如图6-2所示。



图6-2 房屋管理页面

1. 业主管理页面

业主管理页面中显示业主基本信息，如图6-3所示。



图6-3 业主管理界面

1. 车位管理页面

页面中显示车位信息，如图6-4所示。



图6-4 车位管理页面

1. 抄表管理页面

页面中显示用户抄表信息管理，如图6-5所示。



图6-5 抄表管理页面

6）物业收费页面

页面中显示物业收费管理，如图6-6所示。



图6-6 物业收费管理

# 7 系统测试

## 7.1测试概要

### 7.1.1测试目的

在于执行测试提供用例，指导测试的实施，查找分析缺陷，评估测试质量并执行测试用例后，需要跟踪故障，以确保开发的产品适合需求。

测试声明：测试人员在软件开发过程中的任务

① 寻找BUG；

② 避免软件开发过程中的缺陷；

③ 衡量软件的品质；

④ 关注用户的需求。

### 7.1.2测试环境

系统测试环境如表7-1所示

表7-1 测试环境

|  |  |
| --- | --- |
| *环境* | *所有环境都是部属在单机上的* |
| 硬件配置 | CPU：奔腾G5500  内存：16G  硬盘：金士顿固态240G |
| 软件配置 | Win10、Jmeter测试工具 |
| 网络配置 | 无要求 |

### 7.1.3测试执行

主要采用黑盒测试，以需求分析文档中的功能模块为单位，测试的重点集中在基本数据录入、业务流程及各功能模块间的接口。

## 7.2 测试结论

1）功能

系统正确的实现了用户登录功能等功能，已存在一些账户和密码的约束问题。例如：要用户名不能重复……

2）性能

系统在单机部署的情况下，能承载20000个用户发起的20000次请求。但视不同的用户主机有一些小小的差异，当线程开太多的时候容易产生冲突，出现异常。

# 8 结论与展望

## 8.1 结论

当前社会是一个计算机普遍应用的社会，随着社会的不断发展和人们的生活水平的提高，人们逐渐步入的信息社会，同时，随着房地产行业的发展，住宅小区已称为居住的主流，但小区物业管理水平还相对落后，传统的人工方式难以应付大量的业主资料，复杂频繁的业务收费以及对各项资源的管理。小区物业管理系统就是为了为小区管理者更好的开展小区各项业务处理工作而开发的软件。

## 8.2 展望

一个完美的产品是希望它能适应各式各样的需求，本系统框架也是不尽完美的，也是有需要改进的地方，这也是展望之处：

1）减少不必要的冗余代码；

2）实现更多的个性化设置、功能；

3）优化前端页面；

# 参考文献

[1]黄文毅.springboot 微服务项目实战[M].

[2]王珊.数据库系统概论[M].

[3]软件设计开发文档【国家标准】-GB8567-88[S].

[4]朱少明.软件质量保证和管理[M].

[5]黑马.Java实用教程[M].

[6]尚硅谷.数据库系统概论[M].

[7]张奇思.数据库设计三大范式[R].

[8]钟鸣.MySQL必知必会[M].

[9]谢希仁.计算机网络[M].

[10]耿祥义,张跃平.Java 2实用教程[M].

[11]黄文毅.SpringBoot微服务项目实战[M].

[12]赵丰年.网页设计与制作HTML5+CSS3+JavaScript[M].

[13]黄文毅.一步一步学Spring Boot微服务项目实战[M].

附录