保定学院

本科毕业论文（设计）

题目：**小区车辆管理系统设计与实现**

学 院 **数据科学与软件工程**

学科门类 **工学**

专 业 **软件工程**

学 号 **1605024428**

姓 名 **吴世棚**

指导教师 **李爱华**

职 称 **讲师**

**2020**年 **4**月 **13**日

小区车辆管理系统

摘 要

伴随着中国经济的飞速发展和人民生活质量的快速提高，许多家庭都有了自己的爱车。私家车为人们的出行提供了方便，但也带来了一些管理方面的问题，而小区又是车辆停留时间最长的地方，所以如何更加方便快捷的对小区车辆进行管理是十分有必要的。

本系统用Java语言实现，使用了MVC分层思想。通过Spring Boot框架以及结合模板引擎Thymeleaf来完成系统的开发。利用本系统使小区管理人员能够更高效的对小区中的车辆以及小区中的车位进行管理。

关键字：车辆管理 JAVA Thymeleaf MVC

**ABSTRACT**

With the rapid development of China's economy and the rapid improvement of people's quality of life, many families have their own cars. Private cars provide convenience for people to travel, but also bring some management problems, and the residential area is the longest place where vehicles stay, so how to more convenient and fast management of residential vehicles is very necessary.

The system is implemented in Java language, using the idea of MVC stratification. Through the Spring Boot framework and combined with template engine Thymeleaf to complete the development of the system. This system enables the community managers to manage the vehicles and parking spaces more efficiently.

**Keywords:** vehicle management JAVA Thymeleaf MVC

目 录

[1 引言](#_Toc27887_WPSOffice_Level1) [1](#_Toc27887_WPSOffice_Level1)

[1.1 研究背景及意义](#_Toc10930_WPSOffice_Level2) [1](#_Toc10930_WPSOffice_Level2)

[1.2 国内外研究现状分析](#_Toc7053_WPSOffice_Level2) [1](#_Toc7053_WPSOffice_Level2)

[1.3 研究思路及方法](#_Toc23689_WPSOffice_Level2) [1](#_Toc23689_WPSOffice_Level2)

[2 系统设计相关原理](#_Toc10930_WPSOffice_Level1) [1](#_Toc10930_WPSOffice_Level1)

[2.1 Spring Boot](#_Toc31962_WPSOffice_Level2) [1](#_Toc31962_WPSOffice_Level2)

[2.2 MySQL](#_Toc22603_WPSOffice_Level2) [2](#_Toc22603_WPSOffice_Level2)

[2.3 MyBatis Plus](#_Toc7738_WPSOffice_Level2) [2](#_Toc7738_WPSOffice_Level2)

[2.4 Thymeleaf](#_Toc23382_WPSOffice_Level2) [2](#_Toc23382_WPSOffice_Level2)

[2.5 MVC设计模式](#_Toc28727_WPSOffice_Level2) [2](#_Toc28727_WPSOffice_Level2)

[3 系统需求分析](#_Toc7053_WPSOffice_Level1) [3](#_Toc7053_WPSOffice_Level1)

[3.1 系统说明](#_Toc8745_WPSOffice_Level2) [3](#_Toc8745_WPSOffice_Level2)

[3.1.1 系统的概述](#_Toc10930_WPSOffice_Level3) [3](#_Toc10930_WPSOffice_Level3)

[3.1.2 经济可行性分析](#_Toc7053_WPSOffice_Level3) [3](#_Toc7053_WPSOffice_Level3)

[3.1.3 技术可行性分析](#_Toc23689_WPSOffice_Level3) [3](#_Toc23689_WPSOffice_Level3)

[3.1.3 操作可行性分析](#_Toc31962_WPSOffice_Level3) [3](#_Toc31962_WPSOffice_Level3)

[3.2 系统用户用例图](#_Toc31490_WPSOffice_Level2) [3](#_Toc31490_WPSOffice_Level2)

[3.2.1 普通用户用例图](#_Toc22603_WPSOffice_Level3) [3](#_Toc22603_WPSOffice_Level3)

[3.2.2 管理员用户用例图](#_Toc7738_WPSOffice_Level3) [4](#_Toc7738_WPSOffice_Level3)

[4 系统设计与实现](#_Toc23689_WPSOffice_Level1) [5](#_Toc23689_WPSOffice_Level1)

[4.1 系统总体架构](#_Toc8983_WPSOffice_Level2) [5](#_Toc8983_WPSOffice_Level2)

[图3 系统功能结构图](#_Toc23382_WPSOffice_Level3) [5](#_Toc23382_WPSOffice_Level3)

[4.2 数据库的设计](#_Toc22249_WPSOffice_Level2) [5](#_Toc22249_WPSOffice_Level2)

[4.2.1数据库的分析](#_Toc28727_WPSOffice_Level3) [5](#_Toc28727_WPSOffice_Level3)

[4.2.2数据库表的设计](#_Toc8745_WPSOffice_Level3) [5](#_Toc8745_WPSOffice_Level3)

[4.2.3 E-R图](#_Toc31490_WPSOffice_Level3) [9](#_Toc31490_WPSOffice_Level3)

[4.3 用户功能模块的实现](#_Toc11124_WPSOffice_Level2) [10](#_Toc11124_WPSOffice_Level2)

[4.3.1 注册功能](#_Toc8983_WPSOffice_Level3) [10](#_Toc8983_WPSOffice_Level3)

[4.3.2 登录功能](#_Toc22249_WPSOffice_Level3) [10](#_Toc22249_WPSOffice_Level3)

[4.3.3 用户首页](#_Toc11124_WPSOffice_Level3) [11](#_Toc11124_WPSOffice_Level3)

[4.3.4 停车功能](#_Toc16702_WPSOffice_Level3) [12](#_Toc16702_WPSOffice_Level3)

[4.3.5 充值及办卡功能](#_Toc23379_WPSOffice_Level3) [13](#_Toc23379_WPSOffice_Level3)

[4.3.6 出库缴费功能](#_Toc31146_WPSOffice_Level3) [14](#_Toc31146_WPSOffice_Level3)

[4.3.7 历史缴费查询](#_Toc7183_WPSOffice_Level3) [15](#_Toc7183_WPSOffice_Level3)

[4.4 管理员功能模块的实现](#_Toc16702_WPSOffice_Level2) [16](#_Toc16702_WPSOffice_Level2)

[4.4.1 登录](#_Toc13072_WPSOffice_Level3) [16](#_Toc13072_WPSOffice_Level3)

[4.4.2 用户管理](#_Toc20528_WPSOffice_Level3) [17](#_Toc20528_WPSOffice_Level3)

[4.4.3 车位管理](#_Toc31833_WPSOffice_Level3) [17](#_Toc31833_WPSOffice_Level3)

[4.4.4 车辆管理](#_Toc12144_WPSOffice_Level3) [18](#_Toc12144_WPSOffice_Level3)

[4.4.5 公告管理](#_Toc26859_WPSOffice_Level3) [18](#_Toc26859_WPSOffice_Level3)

[4.4.6 费用管理](#_Toc6119_WPSOffice_Level3) [19](#_Toc6119_WPSOffice_Level3)

[5 结论](#_Toc31962_WPSOffice_Level1) [19](#_Toc31962_WPSOffice_Level1)

[参考文献](#_Toc22603_WPSOffice_Level1) [20](#_Toc22603_WPSOffice_Level1)

[致 谢](#_Toc7738_WPSOffice_Level1) [21](#_Toc7738_WPSOffice_Level1)

# 1 引言

## 1.1 研究背景及意义

我们国家的汽车是从上世纪九十年代开始进入家庭的，从原来平均每一千人中不到十个人有小汽车到现在的每一千人就有一百八十多人拥有小汽车。大多数的汽车都是私家车，它们停放在各个小区当中，因此人们进一步提高了物业对小区车辆的管理[1]。伴随着私家车数量的不断增加，小区中车辆乱停乱放的现象也明显增多。随意停放的车辆经常堵塞居民小区消防通道，挡住消防栓等设施，严重危害人民群众的生命财产安全[2]。

不仅如此，在停车场收费中，使用传统人工方式管理停车场收费不仅效率低下，而且容易出错，同时在实际操作中对车辆信息和个人信息的查询不太方便。因此，开发一个便捷、高效的管理系统，使用网络化对车辆进行管理变得十分有必要[3]。

## 1.2 国内外研究现状分析

从目前的情况来看，我国的停车场管理系统现状还是比较落后的，我们可以看到许多地方对于停车场管理还是采用比较传统的人工方式。在停车场的进出口都需要大量的人工去管理，这使得人工费用在整个管理费用中占的比例比较大，和国外一些发达国家相比，差距还是比较大的[4]。由于我国新型停车场管理系统还处在应用的初级阶段，许多的管理系统在设计上不太合理或是未考虑到实际情况从而导致了它并不能发挥出真正的作用。

## 1.3 研究思路及方法

本系统采用B/S架构进行开发，同时使用了响应式布局的方案来设计，即只需要开发一套界面来适配不同的设备。要设计该系统的相关功能，在设计中以需求分析为基础，实现流程以及相关问题的实现方法。要对系统的各个功能进行测试，要从不同角度，不同数据来完成。在系统开发完成后，还应该对整体功能进行测试，确保系统的健壮性、正确性以及是否满足需求。

2 系统设计相关原理

## 2.1 Spring Boot

Spring Boot是近几年特别流行的框架，它使得开发更加的简单快捷，在Web开发中起了很大的作用[5]。Spring Boot最大的特点就是避免了大量模板代码和配置文件的编写，它的出现大大提高了开发的效率。

## 2.2 MySQL

现如今各种数据库层出不穷，MySQL不能说是最杰出的，但是它确实是一款非常优秀的数据库。MySQL是关系型数据库，在数据量不大时往往它是第一选择。它有着优化的SQL查询算法同时还支持多操作系统，更重要的是它是开源，这意味着用户不需要支付额外的成本在数据库上，越来越多的互联网公司开始使用MySQL作为自己公司架构的数据库，使用MySQL显然已经成为了一种主流[6]。

## 2.3 MyBatis Plus

MyBatis Plus简称MP是国人团队苞米豆在MyBatis的基础上开发的框架，也就是MyBatis的增强版，它是为简化开发而生。以前使用MyBatis进行开发时，一些比较简单的查询也需要在xml文件编写SQL语句，这样比较浪费时间。而用了MP后我们就不需要再这样了，我们只需要调用那些已经封装好的方法就能完成一些操作，像增删改查、分页它都做了封装。不仅如此，它还能动态生成代码，而且他同样也能使用配置文件来编写自定义方法[7]。

## 2.4 Thymeleaf

Thymeleaf和JSP都是属于模板引擎，但是Thymeleaf的渲染性能更加强大。以往JSP能够将JAVA代码内嵌到标记语言中，这显得非常的杂乱，非常不方便维护。Thymeleaf模板技术是一种相对比较新的技术，不论从性能方面还是还是维护方面Thymeleaf都有着独特的优点[8]。

## 2.5 MVC设计模式

MVC是指数据模型（Model），用户界面（View）以及控制器（Controller）三层结构化的设计方法[9]。MVC的主要思想就是分层开发，它把应用程序分成了三个核心模块，业务层、视图层、控制层。

业务层负责的是实现业务逻辑，使用各个封装好的方法对数据进行处理；视图层主要负责的是展示，将视图渲染完成后展示给用户看；控制层的主要作用是控制着整体的流程。使用MVC模式开发能够减少耦合性，并且维护起来比较方便。

# 3 系统需求分析

## 3.1 系统说明

### 3.1.1 系统的概述

本系统是以B/S架构设计的小型管理系统——小区车辆管理系统，该系统可以为小区中的住户以及临时来访的用户提供快捷的停车方式以及缴费方式。用户在系统中能够实现登录注册，对车位进行查询、停车操作、快捷缴费等；管理员能在系统中对车位、用户进行管理以及对费用进行设定。

### 3.1.2 经济可行性分析

本系统的开发用到的技术、运行环境以及开发工具绝大多数都是免费的，因此从费用成本方面来说很低。从时间成本来看，该系统的主要功能为用户的停车功能以及缴费功能，系统功能并不复杂开发时间不会太长，所以时间成本也不高。综合来看，本系统从经济方面分析是可行的。

### 3.1.3 技术可行性分析

本系统是采用Java语言开发，Java是面向对象语言它被广泛使用，因此有许多其他技术来适配它[10]。像本系统开发所使用到的Spring Boot框架、MyBatisPlus、Maven、JavaScript等，这些技术都是比较成熟且容易入手，所以从技术方面来看是能够适应本系统的开发。

### 3.1.3 操作可行性分析

本系统是通过浏览器跟用户进行交互的，用户不需要额外进行学习。用户只需要通过网页的方式就能完成一系列操作，而且界面简洁明了，操作起来没有任何的困难。

## 3.2 系统用户用例图

### 3.2.1 普通用户用例图

普通用户是系统的使用者，他们代表的可以是小区中的住户也可以是临时用户，他们都能在系统中进行注册登录。普通用户拥有车位查询功能、停车功能、办卡功能、缴费功能以及历史缴费查询功能，普通用户的用例图如图1所示：

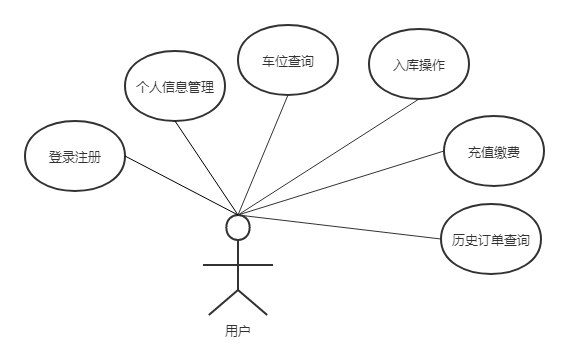


图1 普通用户用例图

### 3.2.2 管理员用户用例图

管理员用户既拥有普通用户的功能，同时他还拥有着普通用户没有的功能，作为系统的核心用户他拥有着系统一切权限，他能够对用户、车位、停车费用等进行管理，管理员用户的用例图如图2所示：

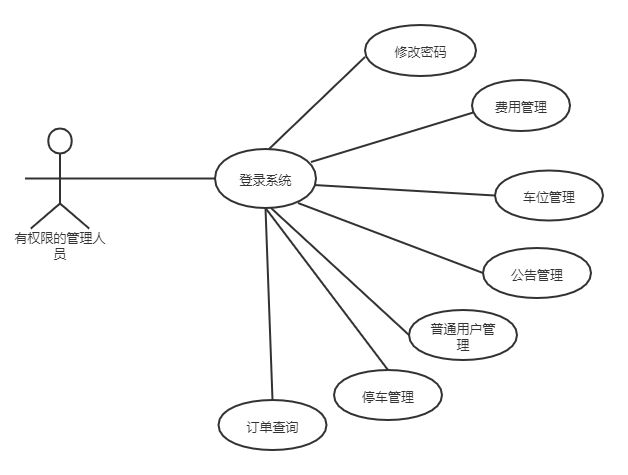


图2 管理员用户用例图

## 3.3 系统功能流程分析

### 3.3.1 用户注册

本系统所有的功能都需要用户登录后才能进行使用，用户需要进行注册后才能拥有账号，用户注册流程图如图3所示。

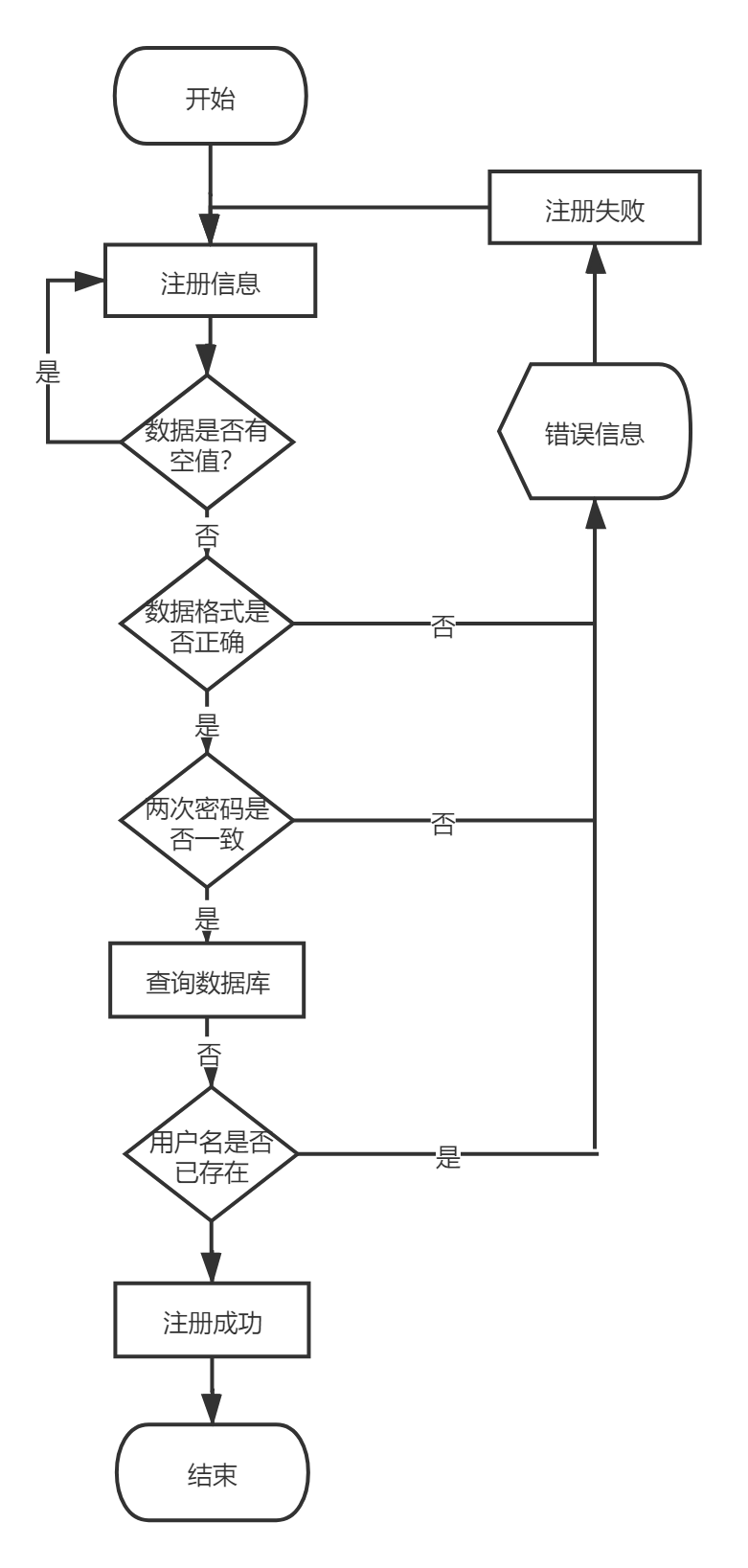


图3 用户注册流程图

### 3.3.2 用户停车

本系统能进行便捷的停车操作，用户停车流程图如图4所示。

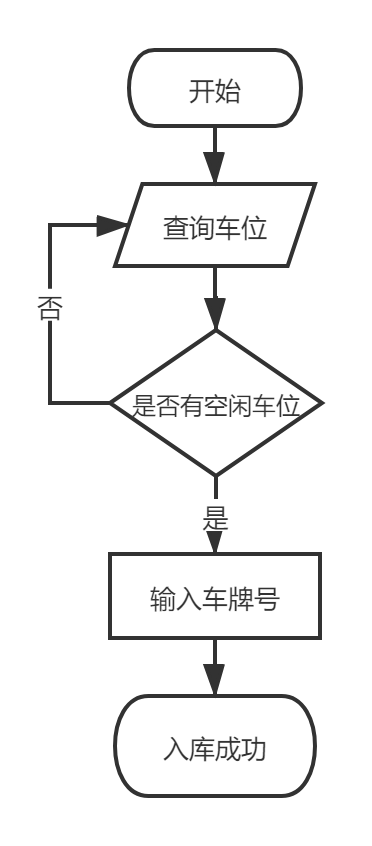


图4 车辆入库流程图

### 3.3.2 车辆出库

用户通过本系统能进行快捷的出库缴费操作，车辆出库流程图如图5所示。

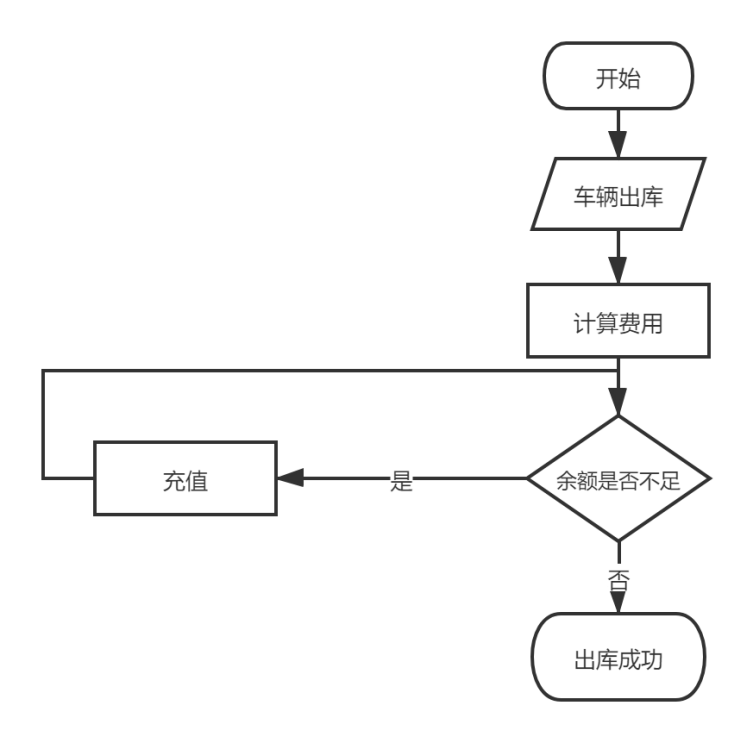


图4 车辆出库流程图

# 4 系统设计与实现

## 4.1 系统总体架构

本系统的总体功能结构如图3所示：

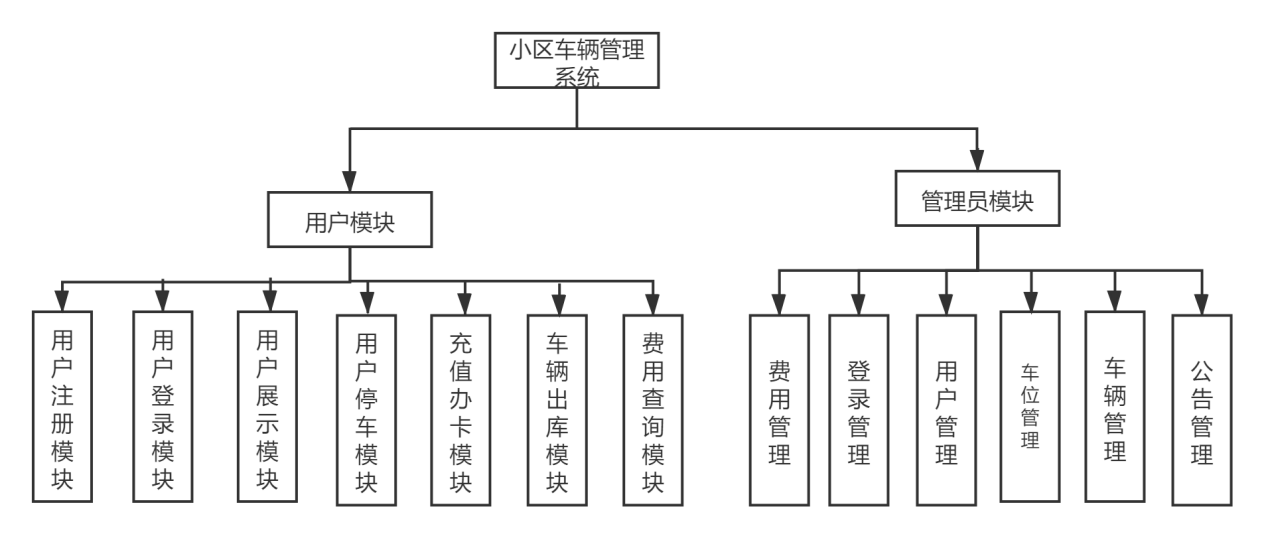


图3 系统功能结构图

## 4.2 数据库的设计

### 4.2.1数据库概述

数据库是系统的核心部分，数据库设计的好坏直接影响了系统的功能，一个好的系统应该与数据库系统协调一致地进行工作[11]。如果数据库中的表设计的合理能大大提高系统的工作效率。

通过对本系统的功能需求进行分析，定义出系统相对应的一些实体。例如：登录功能。这时我们所定义的实体至少要包含用户这个实体。不仅如此，在定义完实体之后，我们还需要定义出实体所包含的一些属性，像用户名、密码之类的属性。仅仅完成对实体进行设计是不够的，实体和实体之间有时候也会有着关系，我们还需要根据实体与实体间的关系画出对应的E-R图。

### 4.2.2 E-R图

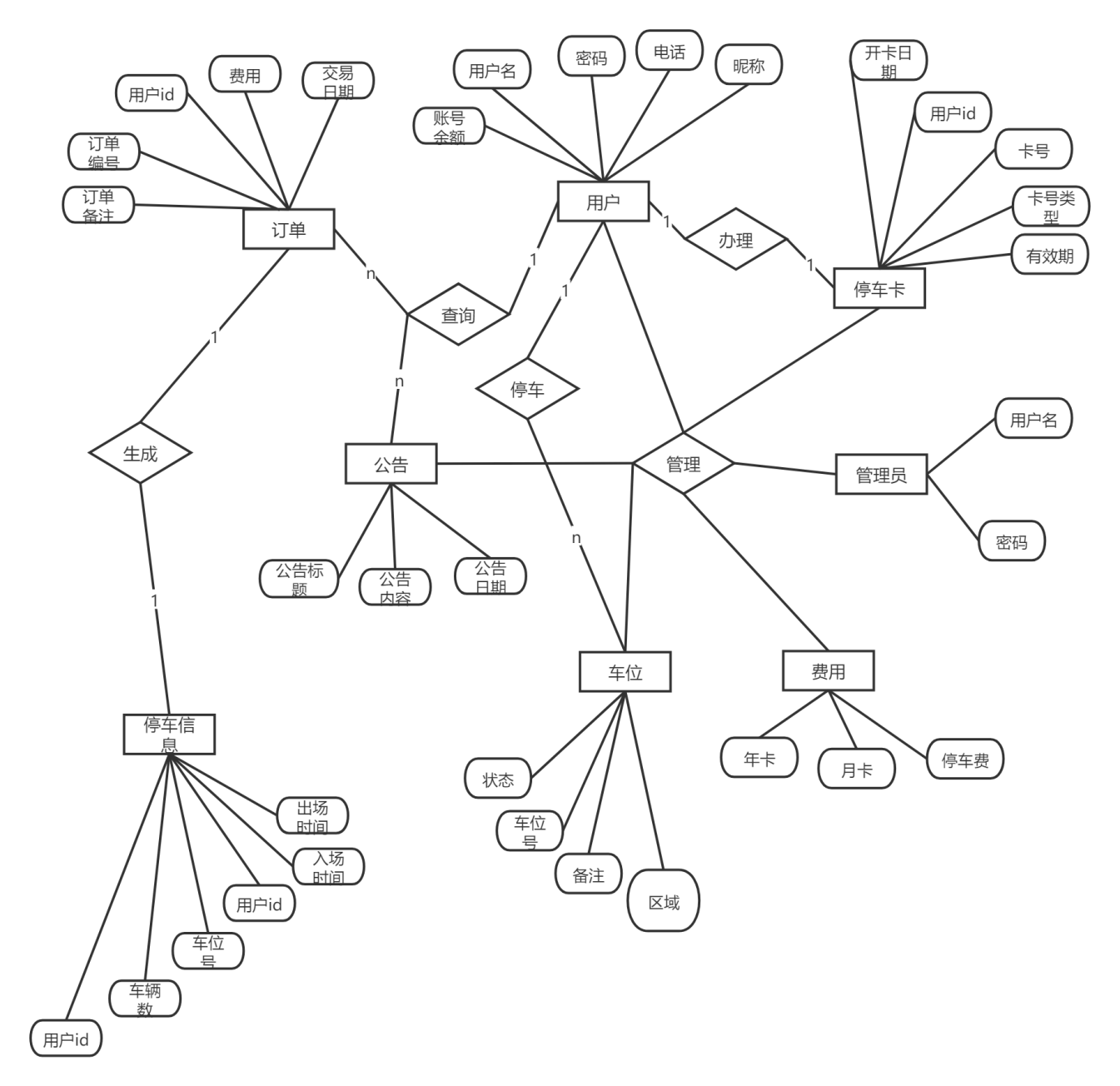


图4 系统E-R图

图4是本系统的E-R图，从图中我们可以看出，一个用户能够停多辆车，是一对多的关系。一个用户办理的卡号只能是一个，因此是一对一的关系。一条停车信息就能生成一个订单，所以是以

### 4.2.3数据库表的设计

本系统使用的是MySQL数据库，数据库名叫car\_manage，共包含8张表。数据库中的表和实体是一一对应的关系，即一个表对应着一个实体。下面对这些表进行介绍。

表1 用户表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 能否空 | 主键 | 注释 |
| id | varchar | 32 | 否 | 是 | 用户id |
| username | varchar | 32 | 否 | 否 | 用户名 |
| password | varchar | 32 | 否 | 否 | 密码 |
| phone\_number | varchar | 11 | 否 | 否 | 手机号 |
| nicheng | varchar | 55 | 是 | 否 | 昵称 |
| money | double | 0 | 是 | 否 | 账户余额 |

用户表记录的是用户的一些个人信息，用户注册的时候这些信息就会在表中记录下来。表中的id字段是作为表的主键，它采用的生成策略是uuid（由容器自动生成的一个32位的字符串），这样能确保主键的唯一性提高索引的效率。用户名字段不允许重复的它的作用是确保能够定位到唯一的一名用户；手机号是为了确保能够联系到用户；账号余额是在本系统中用来缴费的虚拟货币，用户注册时这个默认值为0。

表2 管理员用户表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 能否空 | 主键 | 注释 |
| id | varchar | 32 | 否 | 是 | 管理员id |
| username | varchar | 32 | 否 | 否 | 用户名 |
| password | varchar | 32 | 否 | 否 | 密码 |

管理员用户表存放的是管理员的账号信息，因为管理员的账号是不允许注册的，他的账号信息是提前放到此表中。若是要登录则需要使用这张表的用户名和密码登录。

表3 用户卡号表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 能否空 | 主键 | 注释 |
| id | varchar | 32 | 否 | 是 | 卡号id |
| card\_number | varchar | 32 | 是 | 否 | 卡号 |
| user\_id | varchar | 32 | 是 | 否 | 用户id |
| card\_type | int | 1 | 是 | 否 | 卡号类型 |
| start\_time | datetime | 55 | 是 | 否 | 开卡日期 |
| end\_time | datetime | 0 | 是 | 否 | 有效期 |

本系统考虑到用户可能会办理包月或包年的停车业务，因此设计了这个卡号表。当用户办理了年卡或者是月卡时，就会生成卡号在此表中保存着。card\_type代表着卡号类型，1代表的是月卡用户，2则代表的是年卡用户。同时，根据开卡的类型不同有效期也不同，表中的end\_time则记录的是卡号的有效期。

表4 车位表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 能否空 | 主键 | 注释 |
| id | varchar | 32 | 否 | 是 | 车位id |
| area | varchar | 15 | 否 | 否 | 车位区域号 |
| park\_number | varchar | 15 | 否 | 否 | 车位号 |
| remark | varchar | 55 | 是 | 否 | 备注信息 |
| status | varchar | 1 | 否 | 否 | 车位状态 |

停车位表是用来保存车位的具体信息，用户查询车位时就是从这张表中取的数据。area代表的是车位的区域号，代表的是能够停车的一块区域；park\_number是车位号，通过车位号能唯一确定具体位置；remark是一些备注信息，告诉用户关于这个车位的一些信息；status是车位的状态，它是一个标志位用来标识这个车位当前能不能停，当用户停到某个具体的车位后，这个标志位就会被置为0，这时其他用户就不能再对这个车位进行停车操作。

表5 公告表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 能否空 | 主键 | 注释 |
| id | varchar | 32 | 否 | 是 | 公告id |
| title | varchar | 55 | 是 | 否 | 公告标题 |
| content | varchar | 255 | 是 | 否 | 公告内容 |
| time | datetime | 0 | 是 | 否 | 公告日期 |

公告表所对应的是管理员发布的一些公告，像一些停车的注意事项或是小区一些通知类的消息。当管理员发布了这些通知，都会在这张表中保存着，而向用户展示的通知公告也同样是从这张表中进行获取的。

表6 用户停车表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 能否空 | 主键 | 注释 |
| id | varchar | 32 | 否 | 是 | 停车表id |
| user\_id | varchar | 32 | 是 | 否 | 用户id |
| carparks\_id | varchar | 32 | 是 | 否 | 车位id |
| plate\_number | varchar | 32 | 是 | 否 | 车位号 |
| start\_time | datetime | 0 | 是 | 否 | 入场时间 |
| end\_time | datetime | 0 | 是 | 否 | 出场时间 |
| sort | int | 2 | 是 | 否 | 车辆数 |

用户和车位本来是没有关联的，他们都是一个独立的实体，如果要让它们有联系就需要通过中间表来建立关系。这张用户停车表就是将用户和车位建立起关系的中间表，通过用户的id和车位号的id就能让它们产生关联。除此之外，表中的plate\_number字段代表的是用户停车车辆的车牌号；start\_time则是用户将车辆入库的具体时间；end\_time是用户车辆的出场时间，当用户的车辆缴费成功出库时就会记录下这个时间。sort字段代表的是用户当前停车的车辆数，当办理了月卡或年卡的用户停车数超过1时，超出数量的车就不能享受停车优惠。

表7 费用设定表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 能否空 | 主键 | 注释 |
| id | varchar | 32 | 否 | 是 | 表id |
| day\_cost | int | 10 | 是 | 否 | 一天费用 |
| month\_cost | int | 10 | 是 | 否 | 月租费用 |
| year\_cost | int | 10 | 是 | 否 | 年租费用 |
| level\_one | int | 10 | 是 | 否 | 第一梯度 |
| level\_two | int | 10 | 是 | 否 | 第二梯度 |
| level\_three | int | 10 | 是 | 否 | 第三梯度 |
| one\_time | int | 10 | 是 | 否 | 第一梯度时长 |
| two\_time | int | 10 | 是 | 否 | 第二梯度时长 |
| three\_time | int | 10 | 是 | 否 | 第三梯度时长 |

费用设定表是用来存储管理员所设定的费用，根据不同的梯度以及每个梯度对应的时长进行费用的设定，同时还能对年卡以及月卡的费用进行设定。

表8 缴费订单表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 能否空 | 主键 | 注释 |
| id | varchar | 2 | 否 | 是 | 表id |
| user\_id | varchar | 32 | 是 | 否 | 用户id |
| pay | double | 0 | 是 | 否 | 费用 |
| order\_number | varchar | 55 | 是 | 否 | 订单编号 |
| time | datetime | 0 | 是 | 否 | 交易日期 |
| context | varchar | 100 | 是 | 否 | 订单备注 |

缴费订单表记录的是用户的缴费记录，用户所有的消费、充值记录都会在这张表中保存着。通过user\_id将用户和缴费订单联系起来，订单备注对应的是缴费的用途，如：办理月卡。当用户查询自己缴费历史时系统就会从这张表中获取相对应的记录反馈给用户。

## 4.3 用户功能模块的实现

### 4.3.1 用户注册模块

用户想要使用本系统首先需要做的就是进行注册，用户在进行注册的时候需要填写的字段有用户名、昵称、密码、确认密码以及手机号。当用户对这些必填信息不填写或者填写的不符合规范时系统会对用户进行友好提示，只有校验通过时账号才能有效。注册页面如图5所示。



图5 用户注册界面

### 4.3.2 用户登录模块

当用户注册后数据库中就有了用户的信息，这时用户就能够进行登录操作。用户登录时需要输入的信息为用户名和密码，同样系统也对用户输入信息进行了校验，输入非法时会进行友好提示。当用户输入了正确的用户名和密码就能进入系统，同时系统会保存用户的信息到session中。登录页面如图6所示。

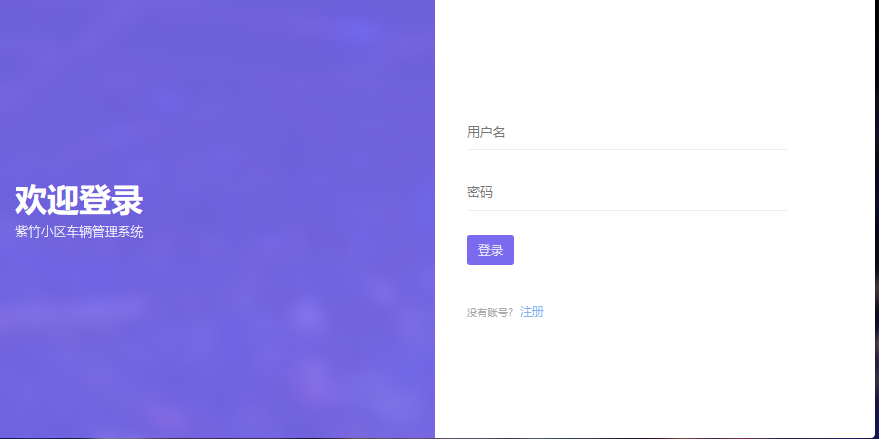


图6 用户登录界面

登录功能关键代码如下：

// 查询用户是否存在

QueryWrapper<User> queryWrapper = new QueryWrapper<>();

queryWrapper.eq("username", user.getUsername());

queryWrapper.eq("password", user.getPassword());

User u = userMapper.selectOne(queryWrapper);

//如果存在则保存用户session

if (u != null) {

request.getSession().setAttribute("user", u);

}

### 4.3.3 用户展示模块

用户成功登录系统后会进入到首页，首页会对用户展示一些个人的信息以及车位的信息，例如账户余额、车位剩余数，同时管理员发布的公告以及收费规则都会在首页向用户进行展示。首页如图7所示。



图7 用户首页

首页关键代码如下：

// 获取费用说明信息

List<UserType>userTypeInfoList=this.userService.getUserTypeInfo(); model.addAttribute("userTypeInfoList", userTypeInfoList);

// 获取公告栏

List<Announcement> announcementList = this.userService.getAnnouncementList();

model.addAttribute("announcementList", announcementList);

return "/user/index";

### 4.3.4 用户停车模块

用户能够对车位进行条件查询来查找特定区域的空闲车位，当选择完自己想要的车位点击入库就能进行停车操作。用户进行停车操作时只需要输入车牌号信息就能完成车辆的入库。车位查询界面如图8所示，车辆入库界面如图9所示。



图8 车位查询

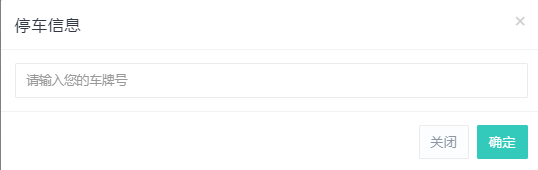


图9 车辆入库

### 4.3.5 充值办卡模块

本系统考虑到了小区中的住户可能会对车位进行包月或包年操作，因此提供了办理月卡以及年卡的功能，用户可以选择办理。用户注册后账户是没有余额的，必须对账户进行充值后才能进行办卡以及对车辆出库时进行缴费。用户能够选择充值金额或是自定义充值金额，同时为了方便用户进行充值操作，本系统使用了支付宝接口来提供对系统的充值。充值功能如图10所示，办卡功能如图11所示。

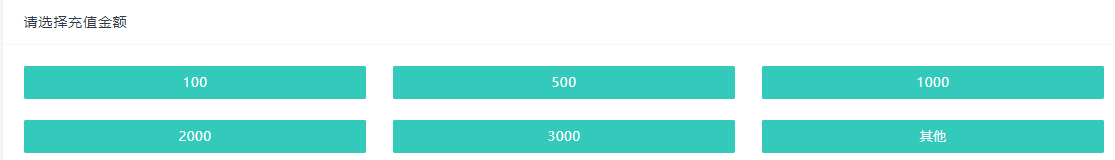


图10 用户充值



图11 用户办卡

### 4.3.6 车辆出库模块

用户停车成功后会在“我的车位”中找到自己的停车信息，当用户点击出库时系统会弹出车辆的停车信息给用户展示并进行缴费，由于用户有可能办理了月卡或是年卡，因此系统会根据用户是否办卡来区分普通用户和VIP用户并进行相对应的费用计算。当用户点击确定时就会进行缴费，如果用户的余额不足以缴纳停车费用时系统就会提示用户充值后再进行交费。同时系统还对办卡用户进行了停车限制，如果办理了年卡或者月卡的用户想要停多辆车那么他们缴费时只有停的第一辆车能够享受优惠，多余的车辆只能是按照普通计费方式进行缴纳。缴费页面如图12所示



图12 出库缴费

### 4.3.7 费用查询模块

为了能给用户更好的体验方便用户对账单进行核算，用户每一次充值、缴费后系统都会生成一个订单，用户能够在系统中对这些订单进行查询。历史缴费查询如图13所示。



图13历史缴费

历史缴费查询关键代码如下：

// 历史缴费查询

public UserVO<UserPay> getUserCost(String userId, String start, String end, Long currentPage, Long size) {

Date startDate = null;

Date endDate = null;

if (!StringUtils.isEmpty(start)) {

startDate = DateUtil.parse(start);

}

if (!StringUtils.isEmpty(end)) {

endDate = DateUtil.parse(end);

}

IPage<UserPay> iPage = new Page<>(currentPage, size);

QueryWrapper<UserPay> queryWrapper = new QueryWrapper<>();

// 获取用户信息

queryWrapper.eq("user\_id", userId);

// 起始日期和结束日期都不为空

queryWrapper.ge("time", startDate);

queryWrapper.le("time", DateUtil.offsetDay(endDate, 1));

// 分页查询用户订单信息

IPage<UserPay> page = userPayMapper.selectPage(iPage, queryWrapper);

UserVO<UserPay> userPayVO = new UserVO<>();

userPayVO.setPages(page.getPages());

userPayVO.setCurrent(page.getCurrent());

userPayVO.setSize(page.getSize());

userPayVO.setTotal(page.getTotal());

userPayVO.setList(page.getRecords());

return userPayVO;

}

}

## 4.4 管理员功能模块的实现

### 4.4.1 登录管理

管理员的账号是不能通过注册获得的，只能使用数据库留下的特定账号进行登录。管理员登录界面如图14所示。



图14历史缴费

### 4.4.2 用户管理

管理员能在系统中对用户进行管理，所有已经在本系统注册过的账号都能被查询的到。不仅如此，管理员还能对用户进行新增、修改、删除操作，同时还能够对账户的余额进行充值。用户管理如图15所示。

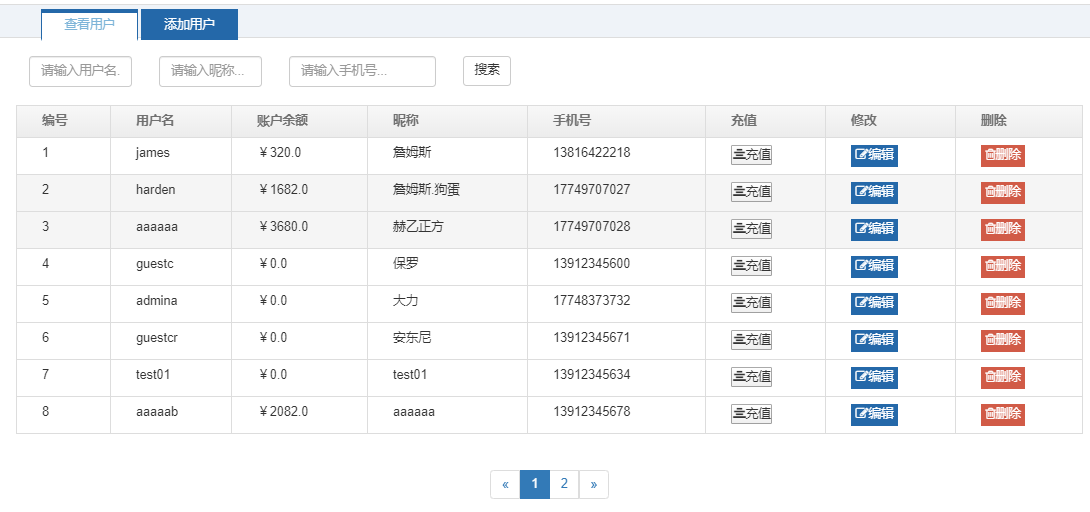


图15用户管理

### 4.4.3 车位管理

管理员能够发布新的车位以及对已有的车位进行编辑修改，同时也能对现有的车位进行删除操作，但是为了安全，如果某个车位当前被用户所占用时，对这个车位进行删除操作会失败。车位管理如图16所示。

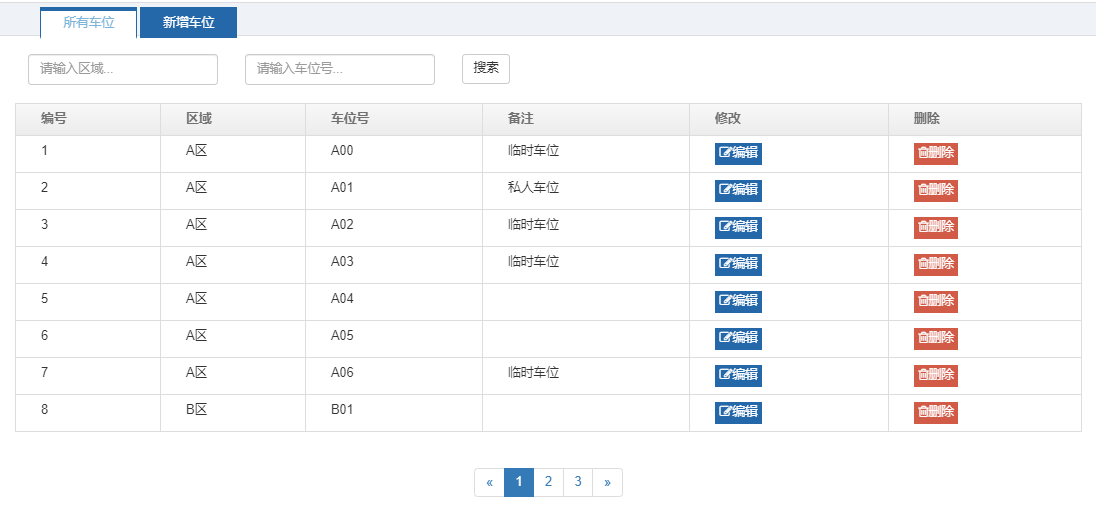


图16车位管理

### 4.4.4 车辆管理

为了方便用户的使用，管理员也能够在空闲车位完成对用户车辆的入库操作。管理员只需要输入用户的用户名以及车牌号就能对车辆完成入库，如果用户不存在那么系统就会提示给管理员并且不能完成对用户车辆的入库。除此之外，管理员能够查询到所有正在停车的车辆信息以及用户信息，还能为在场的车辆完成出库操作。车位管理功能如图17和18所示。

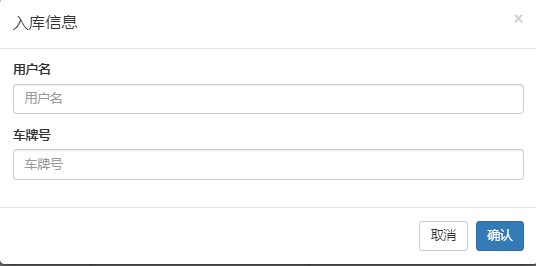


图17车位管理



图18车位管理

### 4.4.5 公告管理

“小区公告”是小区物业给业主提供一些重要信息以及重要规定的方式。同样，管理员想要对所有用户进行通知或是发布一些注意事项都是在公告管理这个模块进行的，管理员能在这发布新的公告对发布的公告内容进行编辑，还能查询和删除一些历史公告。发布的这些公告在用户的首页对用户进行展示，公告管理如图19所示。



图19公告管理

### 4.4.6 费用管理

办理月卡和年卡时所需要的费用，用户停车时收取多少费用，这些关于费用的设置都是在费用管理这个模块进行的。比如想要设定用户第一个小时的停车费用免费，那么就可以设定第一梯度时长为1第一梯度费用为0。费用管理如图20所示。

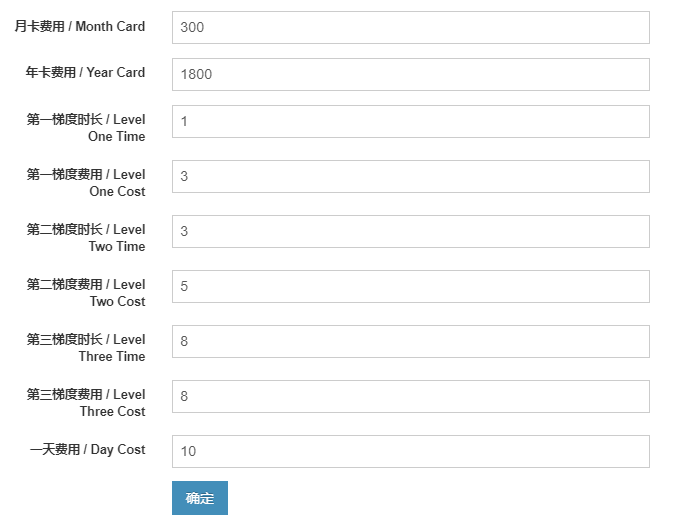


图20费用管理

# 5 系统测试

## 5.1 测试环境

处理器：Inter(R) Core (TM) i5- 6200U CPU @2.30GHz

内存：12GB

系统：windows10

开发工具：Intellij IDEA

数据库：MySQL

数据库管理工具：Navicat

## 5.2 测试用例及结果

表8 武馆官网操作部分测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试模块 | 测试用例 | 详细操作 | 预期结果 | 实际结果 |
| 首页链接跳转 | 页面跳转错误 | 首页展示其他项目，点击调转其他页面 | 正常跳转 | 与预期结果相符 |
| 产品信息更新 | 选择其他产品分类无法显示相应页面 | 在某一产品分类页面点击其他产品分类选项 | 正常跳转至其他产品分类页面 | 与预期结果相符 |
| 产品购买 | 无法跳转产品链接 | 点击产品 | 跳转微店界面 | 与预期结果相符 |
| 武馆课程更新 | 选择其他课程无法显示相应页面 | 在某一课程页面点击其他课程选项 | 正常跳转至其他课程页面 | 与预期结果相符 |
| 武师信息更新 | 选择其他武师分类无法显示相应页面 | 在某一武师分类页面点击其他武师分类选项 | 未正常跳转至其他武师分类页面 | 未与预期结果相符 |
| 关于武馆界面更新 | 选择其他分类无法显示相应页面 | 在某一分类页面点击其他武师分类选项 | 正常跳转至其他分类页面 | 与预期结果相符 |

# 6 结论

小区车辆管理系统主要提供了一个便于用户在小区中停车缴费的方式以及方便小区物业对小区车辆进行管理的平台。通过前期的调查分析，设计出了系统的主要功能；对系统做可行性分析确定是否能够完成，一步步完成这个系统的设计。在用户模块中，完成了用户的停车功能、充值办卡功能、出库缴费等功能。在管理员模块中，完成了用户管理功能、车位管理功能、公告管理功能以及费用管理功能。

目前系统能够完整的运行，在完成系统功能的过程中也遇到了许多的问题。在用户订单查询这个模块，当用户增加了日期条件进行筛选的时候如何将字符串转化成时间戳；如何对办卡用户控制他们优惠停车的车辆限制。除了这些已经解决的问题，本系统还存在着一些瑕疵，系统不能满足高并发和高可用，这些都是以后需要完善的地方。

# 参考文献

1. 王鹏. 智能小区物业管理系统设计与实现[D].江西财经大学,2019.
2. 文平. 多管齐下缓解车辆乱停乱放现象[N]. 益阳日报,2019-08-19(002).
3. 苟坤炎,韩利凯.基于web停车场车辆管理系统的设计与实现[J].电子世界,2019(18):193-194.
4. 吕田. 智能停车场管理系统的设计与实现[D].西安工业大学,2016.
5. 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.
6. 郑智方,李彬,刘世坤,李鹏.探究mysql的运用实例-对数据库的宏观把握[J].科技风,2020(06):129.
7. ]叶刚,王立河,王英明,谷国栋.基于Mybatis Plus的动态生成代码设计与实现[J].电脑编程技巧与维护,2019(07):7-8.
8. 唐炜.Spring Data、MongoDB、Thymeleaf的数据持久化方案及分页技术实现[J].陇东学院学报,2017,28(05):9-13.
9. 丁浩.基于MVC模式的购物网站设计研究与实现[J].电脑知识与技术,2019,15(33):27-29.
10. 王志辉.JAVA语言在计算机软件开发中的应用[J].电子技术与软件工程,2019(20):42-43.
11. 吕艳辉.软件开发中如何高效使用数据库[J].数字技术与应用,2016(11):147-147.

# 致 谢

时光飞逝，从论文选题到开题报告，从开始系统的设计到完成论文，在这过程中我经历了喜悦、害怕以及兴奋。论文得以完成，要衷心的感谢我的指导老师李爱华老师，她给了我很大的帮助。从选题到论文，李老师一步步指导着我并给我很大的鼓励。系统设计有不合理的地方李老师会给我建议，当态度放松时她会对我进行批评。她有着严谨认真的治学态度同时还有着很深厚的专业修养，正是有了她的不畏辛苦我才能成功完成系统的设计及论文的写作。

从初入大学校园到即将步入社会，四年的时间让我学到了很多。感谢所有在大学期间传授我知识的老师，是你们教会了我各种不同的专业技能；感谢所有的大学同学，是你们给我带来了帮助给我带来了快乐；感谢我的父母和家人，是你们的支持和鼓励让我能勇往直前。

大学生活即将结束，而我也将进入人生的下一关，在以后的日子里我要加倍努力来报答学校、老师对我的帮助。