

学士学位论文

小区车辆管理系统设计与实现

姓 名 吴世棚

学 号 1605024428

院 系 数据科学与软件工程

专 业 软件工程

指导教师 李爱华

二零二零年五月四日

学位论文原创性声明

本人所提交的学位论文 ，是在导师的指导下，独立进行研究工作所取得的原创性成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中标明。

本声明的法律后果由本人承担。

论文作者（签名）： 指导教师确认（签名）：

年 月 日 年 月 日

学位论文版权使用授权书

本学位论文作者完全了解保定学院有权保留并向国家有关部门或机构送交学位论文的复印件和磁盘，允许论文被查阅和借阅。本人授权保定学院可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其它复制手段保存、汇编学位论文。

保密的学位论文在\_\_\_\_\_\_\_年解密后适用本授权书。

论文作者（签名）： 指导教师（签名）：

年 月 日 年 月 日

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。私家车虽然为人们的出行提供了便利，但也带来了一些管理方面的问题，特别是对于小区车辆的管理。由于小区是人们生活的场所也是车辆停留时间最长的地方，因此以更加方便快捷的方式对小区车辆进行管理是十分有必要的。现如今网络发展的速度越来越快，人工管理方式已成为过去式，只有利用网络进行管理才能更加高效。

本系统用Java语言实现，使用了MVC分层思想，采用了传统的B/S架构。利用Spring Boot框架以及结合模板引擎Thymeleaf来完成系统的开发和页面的渲染，使用MySQL作为系统的数据库。用户模块和管理员模块是本系统的两大核心模块，用户模块为用户提供了车位查询、停车缴费、订单查询等主要功能；管理员模块则提供了用户管理、车位管理等管理功能。

关键词：车辆管理；JAVA；Thymeleaf；MVC

ABSTRACT

With the rapid development of Chinese economy and the rapid improvement of people's quality of life, many families have their own car.Although private cars provide convenience for people to travel, they also bring some management problems, especially for the management of residential vehicles.As the community is the place where people live and where the vehicles stay the longest, it is necessary to manage the community vehicles in a more convenient and efficient way. Today's networks are growing faster and faster, manual management is a thing of the past and can only be more efficient if it is managed using the network.

The system is implemented in Java language, using the idea of MVC stratification and the traditional B/S architecture. Using the Spring Boot framework and the combination of template engine Thymeleaf to complete the development of the system and page rendering, using MySQL as the database for the system. The user module and the administrator module are the two core modules of the system, the user module provides users with the main functions such as parking space inquiry, parking payment, order inquiry; the administrator module provides user management, parking space management and other management functions.

**Key words：**vehicle management；JAVA；Thymeleaf；MVC

目 录

[第1章 引言 1](#_Toc7598)

[1.1 研究背景及意义 1](#_Toc10309)

[1.2 国内外研究现状分析 1](#_Toc1657)

[1.3 研究思路及方法 1](#_Toc2918)

[第2章 系统设计相关原理 2](#_Toc4532)

[2.1 Spring Boot 2](#_Toc9341)

[2.2 MySQL 2](#_Toc3130)

[2.3 MyBatis Plus 2](#_Toc297)

[2.4 Thymeleaf 2](#_Toc14486)

[2.5 MVC设计模式 3](#_Toc16270)

[第3章 系统需求分析 4](#_Toc4891)

[3.1 系统说明 4](#_Toc20101)

[3.1.1 系统的概述 4](#_Toc27590)

[3.1.2 经济可行性分析 4](#_Toc4215)

[3.1.3 技术可行性分析 4](#_Toc11314)

[3.1.4 操作可行性分析 4](#_Toc32264)

[3.2 系统用户用例图 4](#_Toc4643)

[3.2.1 普通用户用例图 4](#_Toc3820)

[3.2.2 管理员用户用例图 5](#_Toc11124)

[第4章 系统设计 6](#_Toc8867)

[4.1 系统总体架构 6](#_Toc1646)

[4.2 数据库的设计 6](#_Toc12280)

[4.2.1 数据库概述 6](#_Toc25035)

[4.2.2 E-R图 7](#_Toc31081)

[4.2.3 数据库表的设计 7](#_Toc16893)

[4.3 系统功能流程分析 11](#_Toc13799)

[4.3.1 用户注册 11](#_Toc15721)

[4.3.2 用户停车 12](#_Toc22676)

[4.3.2 车辆出库 13](#_Toc8115)

[第5章 系统实现 14](#_Toc14179)

[5.1 用户功能模块的实现 14](#_Toc17499)

[5.1.1 用户注册模块 14](#_Toc4041)

[5.1.2 用户登录模块 1](#_Toc19247)4

[5.1.3 用户展示模块 1](#_Toc8669)5

[5.1.4 用户充值模块 1](#_Toc7521)6

[5.1.5 用户停车模块 1](#_Toc7581)7

[5.1.6 车辆出库模块 1](#_Toc12356)7

[5.1.7 费用查询模块 1](#_Toc15151)9

[5.2 管理员功能模块的实现 2](#_Toc6909)0

[5.2.1 登录管理 2](#_Toc24151)0

[5.2.2 用户管理 2](#_Toc32425)0

[5.2.3 车位管理 2](#_Toc26239)1

[5.2.4 车辆管理 2](#_Toc31946)1

[5.2.5 公告管理 2](#_Toc6135)2

[5.2.6 费用管理 2](#_Toc5618)3

[第6章 系统测试 2](#_Toc12390)4

[6.1 测试环境 2](#_Toc23311)4

[6.2 测试用例及结果 2](#_Toc1912)4

[结论 2](#_Toc28095)5

[参考文献 2](#_Toc26193)5

[致谢 2](#_Toc28820)6

# 第1章 引言

## 1.1 研究背景及意义

汽车在2004年左右的时候才开始进入普通老百姓的家中，发展至今，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。这些汽车走进了人们的生活，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，依靠人工方式收取费用往往效率很低而且也有出错的可能性，同时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]。总的来说，我国的停车场管理系统目前还在应用的初期，不论是总体设计上还是整体功能上都存在着不足。

## 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，每个功能模块的开发都要在需求分析的基础上进行。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，还应该对系统整体功能进行测试，确保系统的健壮性、正确性以及是否满足需求。

# 系统设计相关原理

本章节对系统所使用到的主要技术Spring Boot、MySQL、MybatisPlus、Thymeleaf、MVC设计模式进行详细的介绍。

## 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软件，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文件上编写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做了，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，而且它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。

业务层的职责是实现业务逻辑，使用各个封装好的方法完成对数据的处理；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，MVC框架图如图2.1所示。

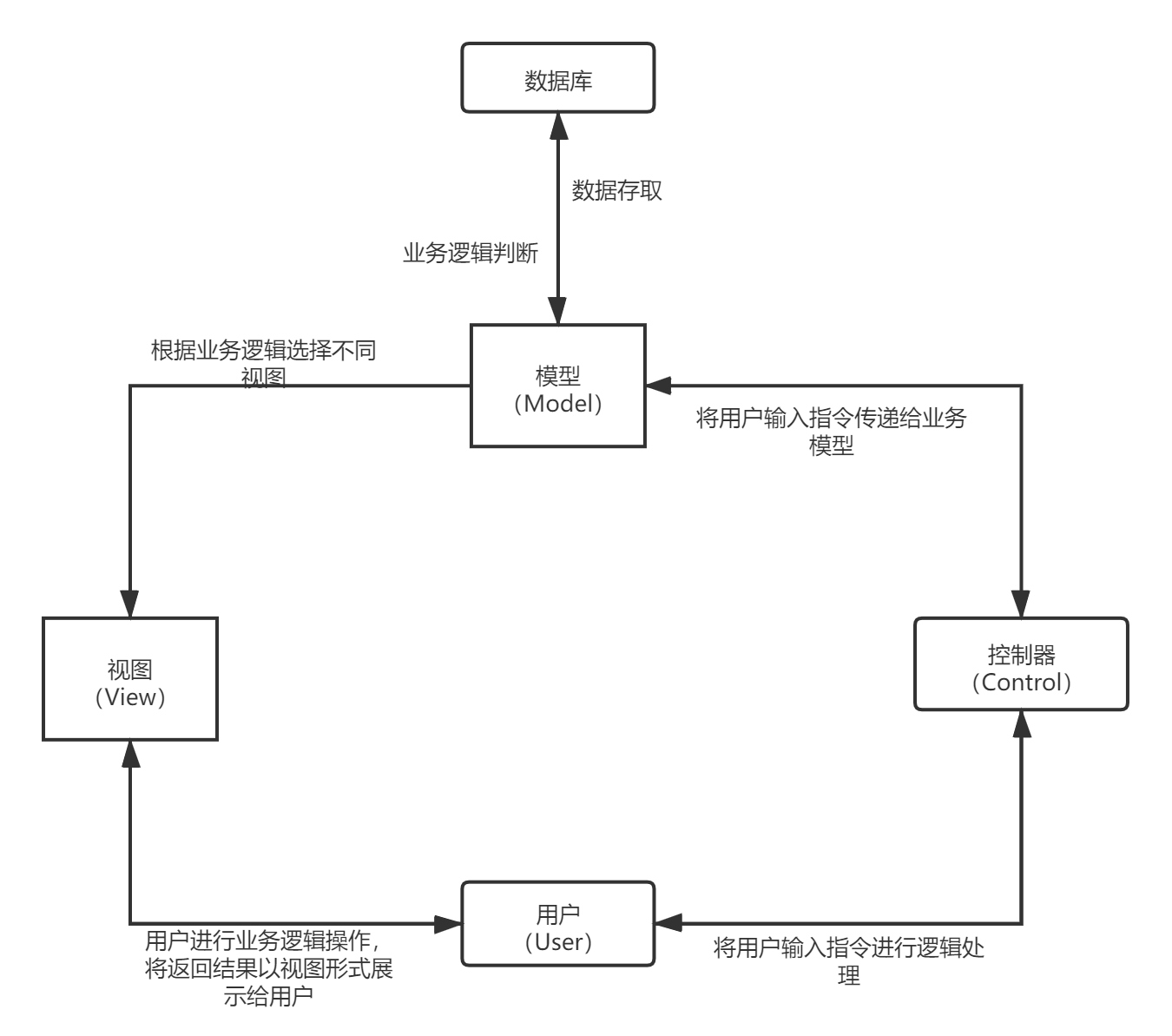


图2.1 MVC框架图

# 系统需求分析

可行性分析是系统开发前必不可少的工作。本章节将对系统进行简单的介绍并通过经济可行性、技术可行性、操作可行性四个方面对系统进行可行性分析。

## 3.1 系统说明

### 3.1.1 系统的概述

小区车辆管理系统是以B/S架构设计的小型管理系统，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，且系统开发完成后能持续稳定的运行不需要支付额为的费用，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，开发该系统从经济方面来说是可行的。

### 3.1.3 技术可行性分析

本系统以Java语言为基础进行开发，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、MyBatis Plus、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。

### 3.1.4 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

作为系统的使用者，普通用户代表的是小区中的住户或者是临时用户，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、充值缴费功能以及历史订单查询功能，普通用户的用例图如图3.1所示：

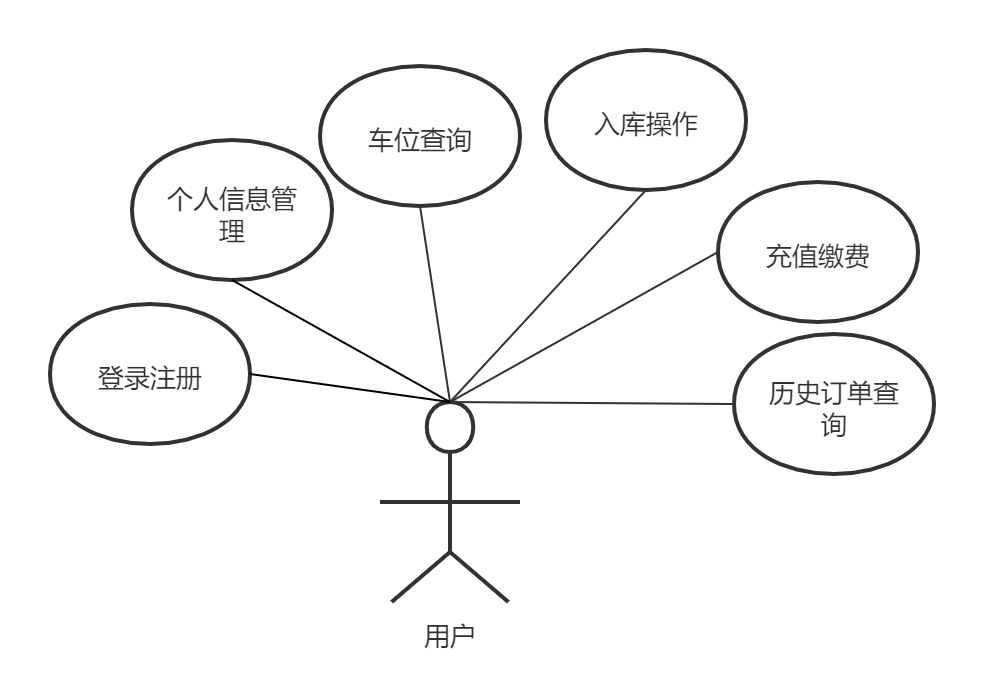


图3.1 普通用户用例图

### 3.2.2 管理员用户用例图

管理员用户作为系统的核心用户拥有着系统的最高权限，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3.2所示：

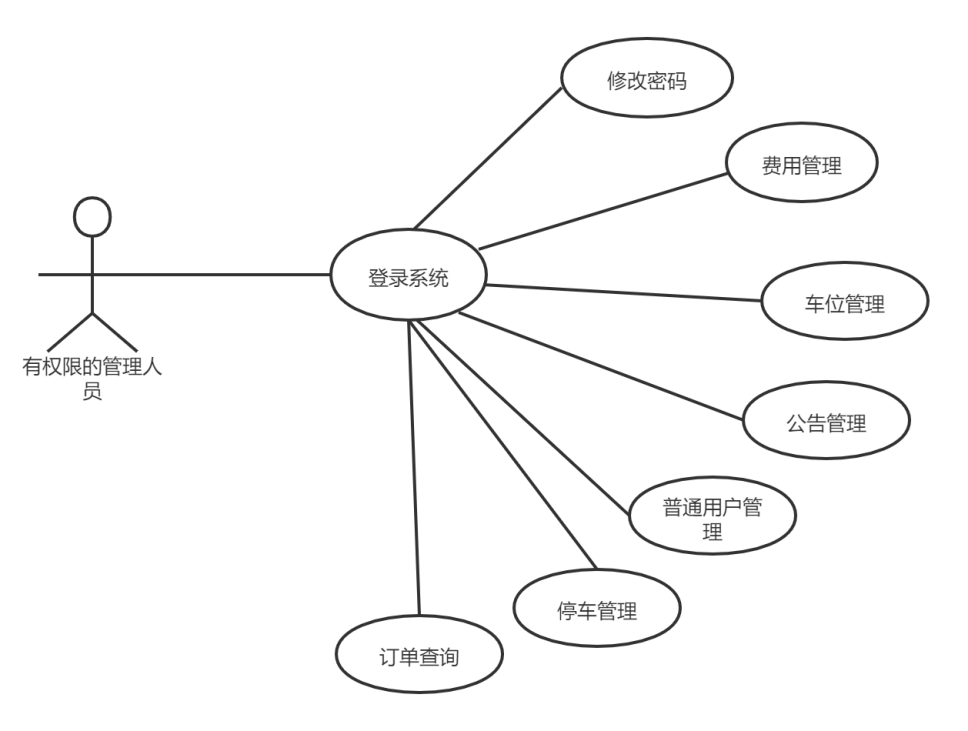


图3.2 管理员用户用例图

# 系统设计

本章主要对小区车辆管理系统的整体设计及数据库设计作详细的介绍。

## 4.1 系统总体架构

本系统的总体功能结构如图4.1所示：

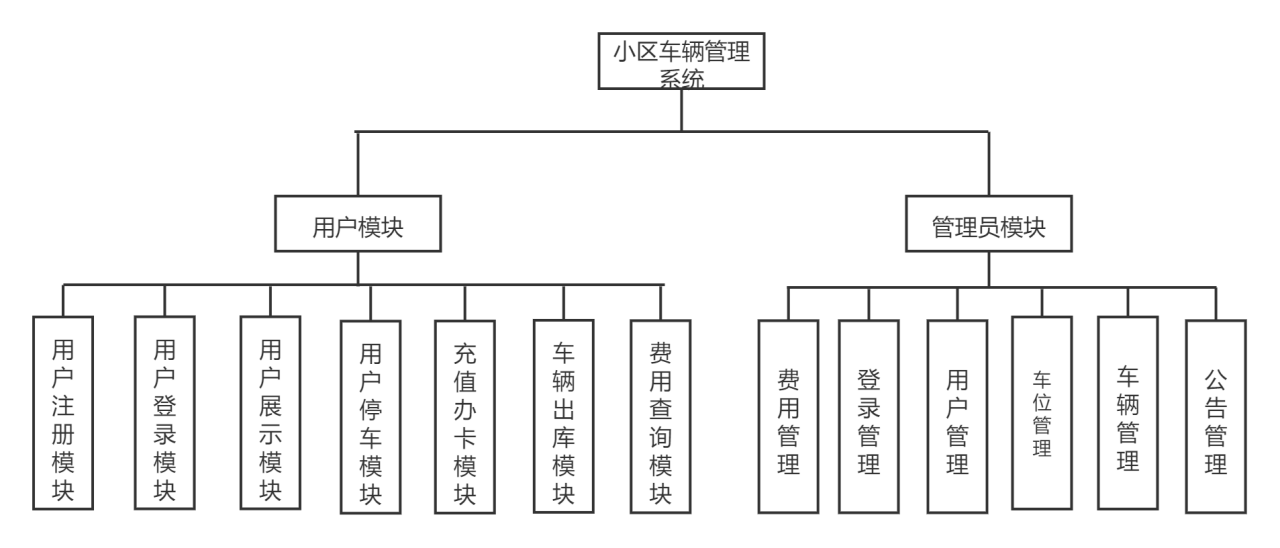


图4.1 系统功能结构图

## 4.2 数据库设计

### 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。仅仅完成对实体的设计是不够的，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

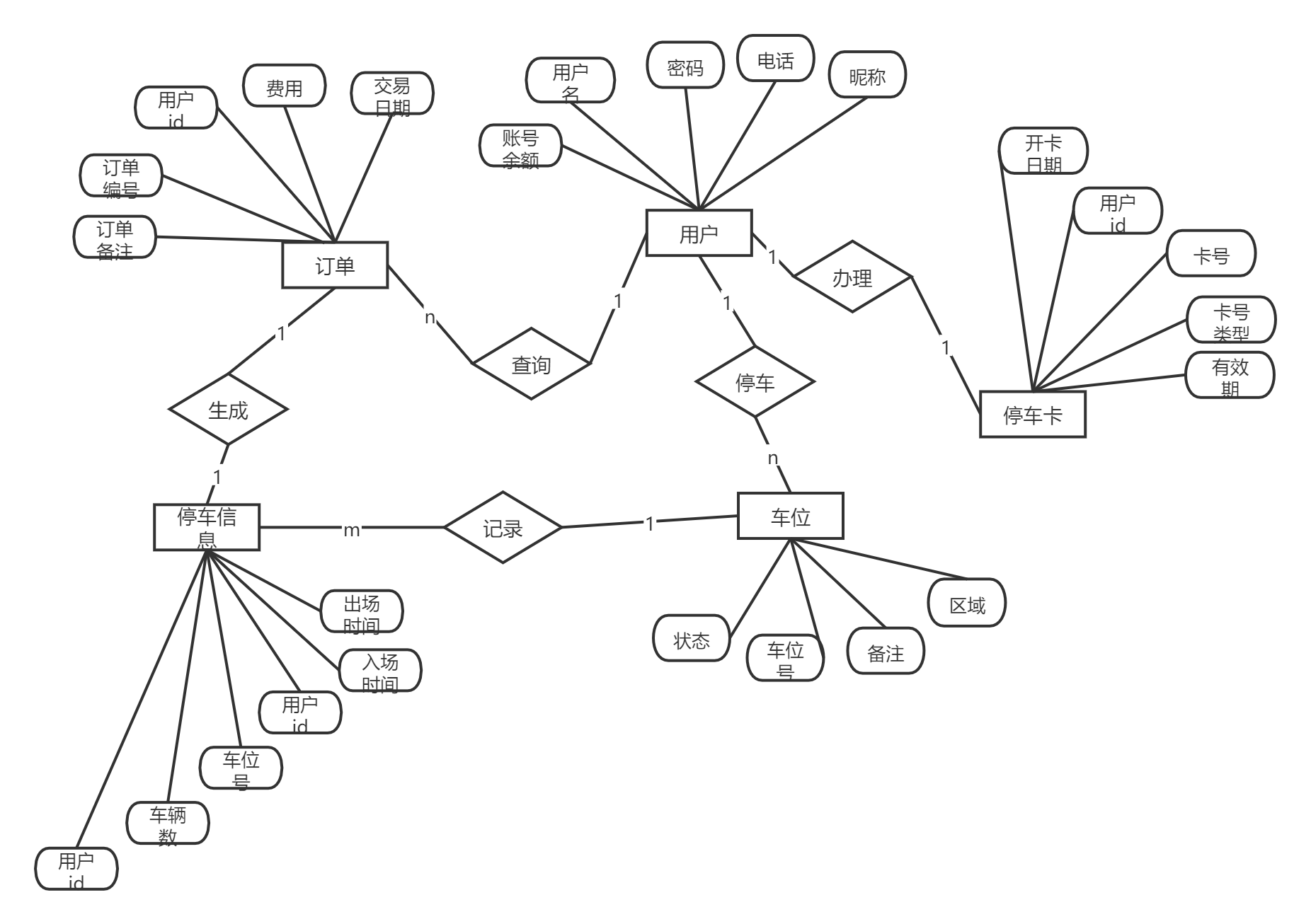


图4.2 系统E-R图

小区车辆管理系统的E-R图如图4.2所示，根据图中的关系可以看出，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；一条停车信息只能生成一个订单，所以是一对一的关系；一个车位能记录多条停车信息，是一对多的关系。

### 4.2.3 数据库表的设计

在本系统中，car\_manage是数据库的名字，在该数据库下共包含8张表。下面对这些表进行介绍。

表4.1 用户表（user\_user）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 | 5 | 是 | 否 | 账户余额 |

用户表：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（username）的作用是确保能够定位到唯一的一名用户，因此它是唯一的；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4.2 管理员用户表（user\_admin）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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：管理员的账号并不是通过注册获得的，而是提前准备好的。这张表存放的就是管理员的账号和密码，若是要进行登录则需要使用这张表的用户名和密码。

表4.3 用户卡号表（user\_card）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 | 6 | 是 | 否 | 开卡日期 |
| end\_time | datetime | 6 | 是 | 否 | 有效期 |

用户卡号表：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代表的则是年卡用户。同时，根据开卡的类型不同有效期也不同，表中的end\_time记录的是卡号的有效期。

表4.4 车位表（car\_carparks）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 |

车位表：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字段是车位的区域号，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，用来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，此时其他用户就不能再对这个车位进行停车操作。

表4.5 公告表（user\_announcement）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 | 6 | 是 | 否 | 公告日期 |

公告表：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的一些通知都可以当做公告。当管理员发布了这些内容，都会在这张表中保存着，同时这些公告信息都会向用户进行展示。

表4.6 用户停车表（car\_parking）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 | 6 | 是 | 否 | 入场时间 |
| end\_time | datetime | 6 | 是 | 否 | 出场时间 |
| sort | int | 2 | 是 | 否 | 车辆数 |

用户停车表：用户和车位本来是没有关联的，都是一个个独立的实体，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并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一辆时，超出数量的车就不能享受停车优惠。

表4.7 费用设定表（car\_parking\_cost）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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：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4.8 缴费订单表（user\_pay）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 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 | 5 | 是 | 否 | 费用 | |
| order\_number | varchar | 55 | 是 | 否 | 订单编号 | |
| time | datetime | 6 | 是 | 否 | 交易日期 | |
| context | varchar | 100 | 是 | 否 | 订单备注 | |

缴费订单表：缴费订单表记录的是用户的缴费记录，用户所有的消费记录、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。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 系统功能流程分析

### 4.3.1 用户注册

用户拥有账号后才能登录本系统，因此用户需要进行注册，用户注册流程图如图4.3所示。

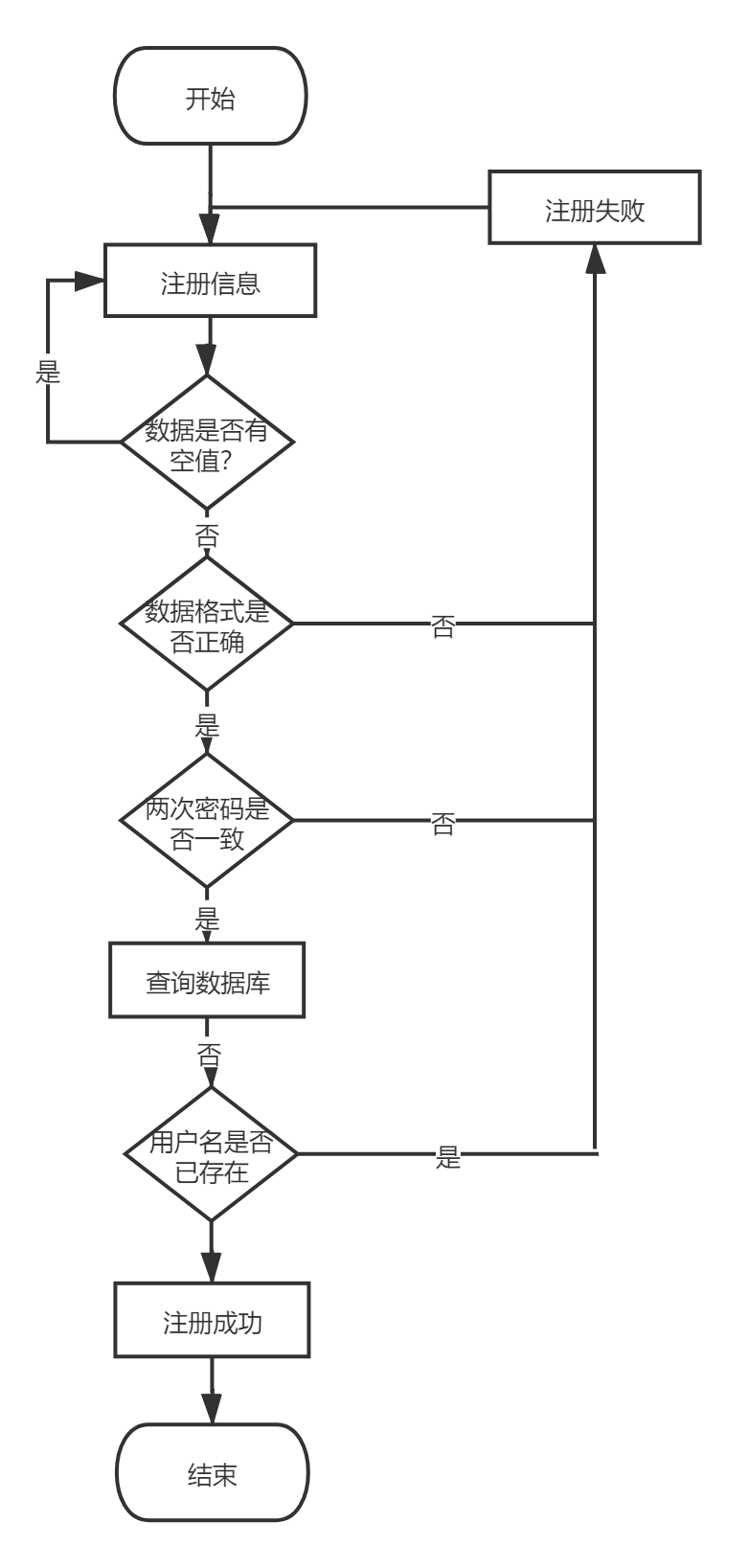


图4.3 用户注册流程图

### 4.3.2 用户停车

本系统为用户提供了便捷的停车操作，用户停车流程图如图4.4所示。

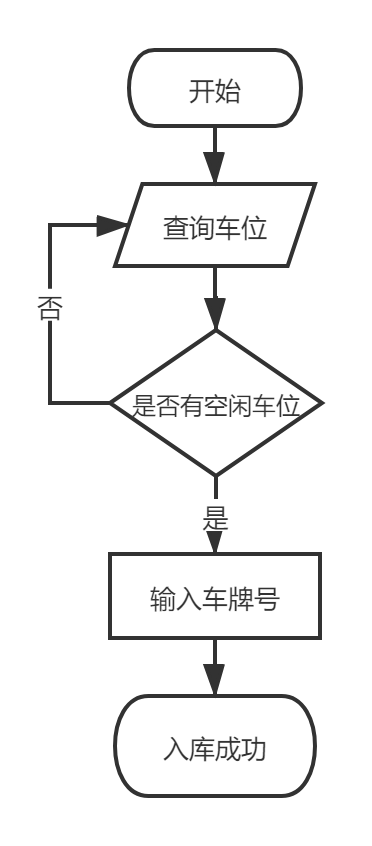


图4.4 用户停车流程图

### 4.3.2 车辆出库

用户通过本系统能进行快捷的出库缴费操作，车辆出库缴费流程图如图4.5所示。

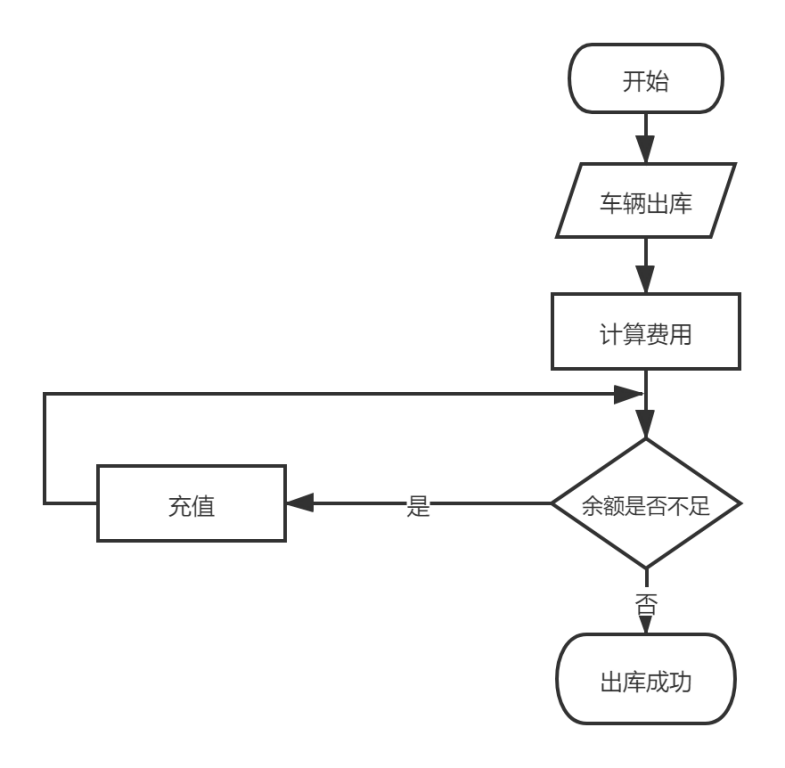


图4.5 车辆出库缴费流程图

# 系统实现

本章通过展示系统功能图片并结合关键代码对系统的管理员模块以及用户模块的具体实现做介绍。

## 5.1 用户功能模块的实现

### 5.1.1 用户注册模块

用户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5.1所示。

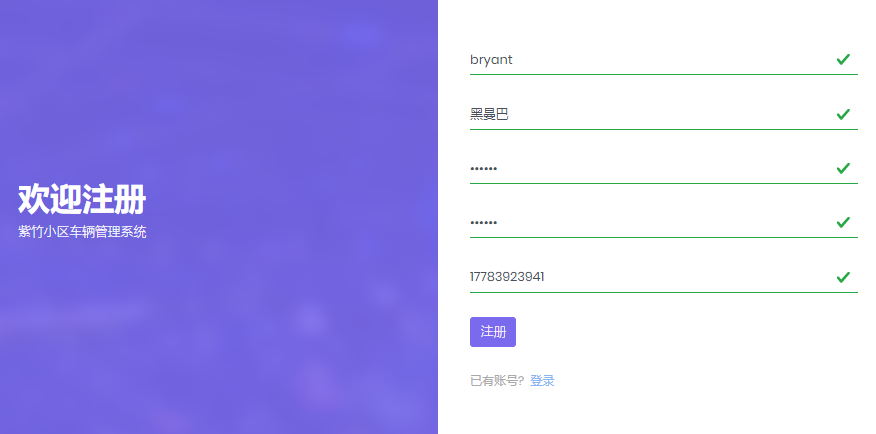


图4.3 注册界面

### 5.1.2 用户登录模块

当用户成功注册后系统后台会生成一条用户的信息存到数据库中，用户就能凭借这条个人信息登录系统。在用户输入完账号和密码并点击登录后，系统会对用户输入的登录信息进行后台校验，输入非法时会给出友好提示。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。用户登录页面如图5.2所示。

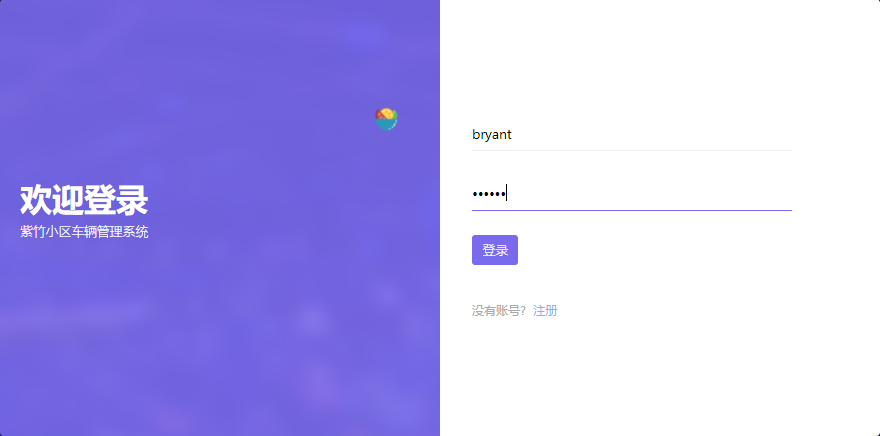


图5.2 用户登录界面

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);

}

### 5.1.3 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。首页个人信息如图5.3所示，首页通知信息如图5.4所示。



图5.3 首页个人信息



图5.4 首页通知信息

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";

### 5.1.4 用户充值模块

用户注册后账户是没有余额的，如果用户想要进行年卡月卡办理或是出库缴费则必须对账户进行充值。用户能够选择指定的充值金额或是自定义充值金额进行充值，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5.5所示。

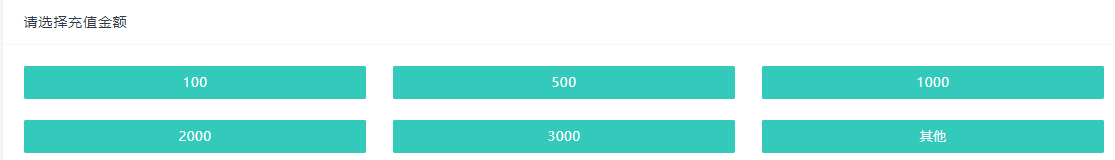


图5.5 用户充值

### 5.1.5 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5.6所示，车辆入库界面如图5.7所示。



图5.6 车位查询



图5.7 车辆入库

### 5.1.6 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对用户进行展示。由于用户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5.8所示。



图5.8 出库缴费

出库缴费关键代码如下：

// 普通用户

if (result == "普通用户") {

// 计算各个梯度的费用

if (hours < levelOneTime) {

cost = dayDiff \* dayCost + levelOne;

} else if (hours >= levelOneTime && hours < levelTwoTime) {

cost = dayDiff \* dayCost + levelTwo;

} else if (hours >= levelTwoTime && hours < levelThreeTime) {

cost = dayDiff \* dayCost + levelThree;

} else if (hours >= levelThreeTime) {

cost = dayDiff \* dayCost + dayCost;

}

} else {

// 办卡用户

cost = 0;

}

### 5.1.7 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5.9所示。



图5.9 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;

}

}

## 5.2 管理员功能模块的实现

### 5.2.1 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5.10所示。



图5.10 管理员登录

### 5.2.2 用户管理

管理员可以通过用户管理这个模块对用户信息进行管理，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，同时还能够对用户的账户余额进行充值。用户管理如图5.11所示。



图5.11 用户管理

### 5.2.3 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5.12所示。

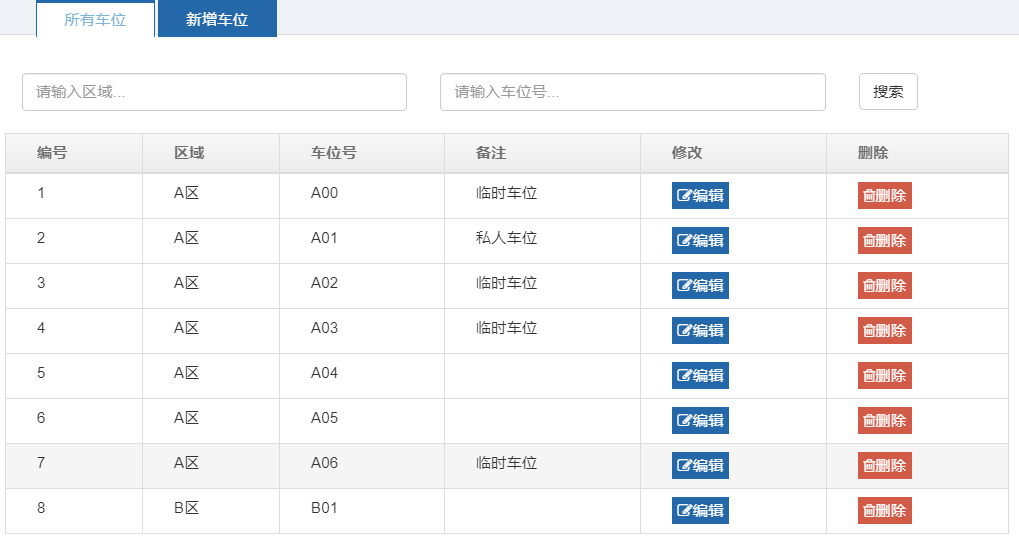


图5.12 车位管理

### 5.2.4 车辆管理

为了给用户提供方便，管理员也能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。用户车辆入库如图5.13所示，车辆管理如图5.14所示。



图5.13 用户车辆入库



图5.14 车辆管理

### 5.2.5 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已发布的公告进行修改，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5.15所示。

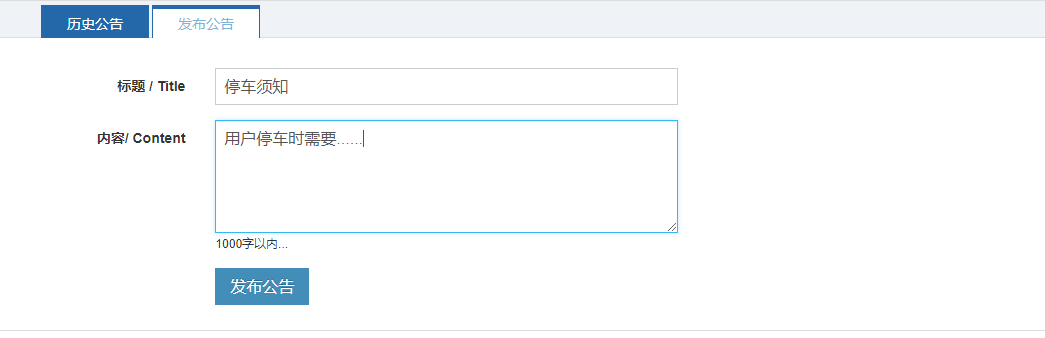


图5.15 公告管理

### 5.2.6 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5.16所示。

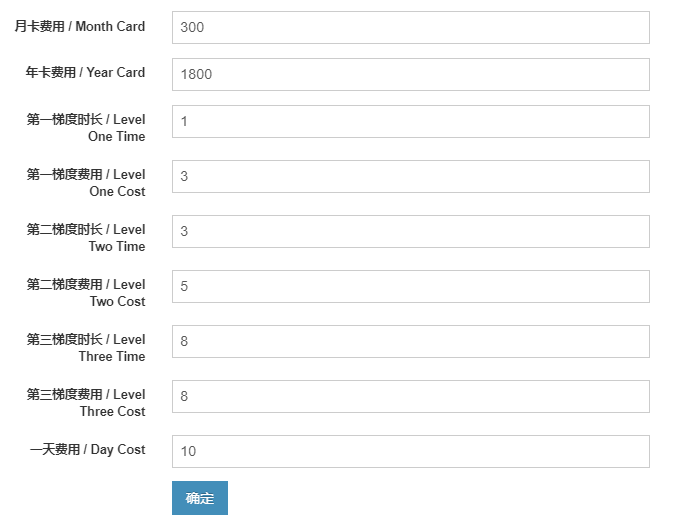


图5.16 费用管理

# 系统测试

系统测试是系统开发中较为重要的一个环节，通过对系统各个模块和系统整体进行测试，找出系统在设计中可能存在的缺陷。

## 6.1 测试环境

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

内存：12GB

系统：Windows 10

开发工具：Intellij IDEA

数据库：MySQL

数据库管理工具：DataGrip

## 6.2 测试用例及结果

|  |  |  |  |
| --- | --- | --- | --- |
| 测试模块 | 测试用例 | 预期结果 | 实际结果 |
| 用户注册模块 | 注册时用户名和手机号等必填信息为空 | 无法完成注册并给出必填信息不能为空的提示 | 与预期结果相符 |
| 用户注册模块 | 使用已经存在的用户名进行注册 | 无法完成注册，给出提示用户名已存在 | 与预期结果相符 |
| 用户登录模块 | 输入用户名和密码进行登录 | 用户名错误或者密码错误给出用户名或密码错误提示，如果用户名密码正确则进入首页 | 与预期结果相符 |
| 用户停车模块 | 输入车牌号为空 | 无法入库，提示请输入车牌号 | 与预期结果相符 |
| 充值办卡模块 | 进入充值界面点击充值 | 根据选择的充值金额跳转到支付宝充值界面从，充值成功后返回首页 | 与预期结果相符 |
| 车辆出库模块 | 选择已经入库的车辆点击出库 | 如果办理年卡或月卡的用户显示VIP用户并且出库费用为0，普通用户则根据停车时长计算费用 | 与预期结果相符 |
| 费用查询模块 | 选择缴费记录查询进行查询历史缴费 | 能显示用户的所有缴费记录并能根据条件进行过滤 | 与预期结果相符 |

# 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；将系统划分成多个子模块进行开发；各个模块开发完后进行功能测试，测试通过后再进行系统总体测试。在用户模块中，完成了用户的停车功能、充值功能、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。

经过测试系统能够稳定的运行，在完成系统功能的过程中解决了许多的问题。例如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将时间字符串转换成时间戳。除了那些已经解决的问题，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好奇、害怕以及喜悦。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。从选题初期到论文完成，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，当我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以及论文的写作。

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。