

长沙理工大学

CHANGSHA UNIVERSITY OF SCIENCE & TECHNOLOGY

# 毕业设计（论文）

题目：基于 Web 的可视化停车场管理系统设计

---

学生姓名：范劲希

学 号：201908010214

班 级：计算机 19-2 班

专 业：计算机科学与技术

指导教师：郑斌

2023 年 6 月

# 基于 Web 的可视化停车场管理系统设计

学生姓名： 范劲希

学 号： 201908010214

班 级： 计算机 19-2 班

所在学院： 计算机与通信工程

指导教师： 郑斌

完成日期： 2023-06-16

## 诚信声明

本人声明：

- 1、本人所呈交的毕业设计（论文）是在老师指导下进行的研究（设计）工作及取得的研究（设计）成果；
- 2、据查证，除了文中特别加以标注和致谢的地方外，毕业设计（论文）中不包含其他人已经公开发表过的研究成果，也不包含为获得其他教育机构的学位而使用过材料；
- 3、我承诺，本人提交的毕业设计（论文）中的所有内容均真实、可信。

作者签名：范劲希

日期： 2023 年 06 月 16 日

## 基于 Web 的可视化停车场管理系统设计

### 摘要

随着现代停车场的发展，现有的停车场管理系统功能以及界面过于老旧不再能够很好的满足车主以及管理员的需求；如停车场管理系统的设备互不兼容，车主无法自主缴费或者自主选择车位，以及管理员无法管理多个停车场区域的问题。为了解决上述问题，同时满足用户对功能丰富、方便快捷的停车场管理系统的需求。本系统设计以下功能模块来实现需求；首先本系统为公司或者小区针对不同区域的停车场进行管理和收费并且进行统一的数据收集和展示，让停车场的各种数据能够一目了然；同时支持电脑以及手机多端登录和适配，配有可视化监控、停车管理、车主管理、临时车管理以及车主模块；本系统打通传统停车场管理系统只有管理员管理的局面，增加了车主角色，以及加强了车主对停车的感知和方便了车主缴费和选择车位的操作；通过按照多区域的停车场管理设计思想，建立区域表进行停车场区域的划分，以及为不同停车场区域配置可视摄像头来进行多区域的车辆监控管理。本系统采用了前后端分离的设计思想，其中前端部分使用了 Vue 框架，后端则采用了 node.js 的 express 框架，UI 采用了 ElementUI 开源组件库；这种设计选择能够提供方便、快速和便捷的服务。本系统的实施为用户提供了一个高效、可靠且功能齐全的停车场管理系统。

**关键词：**停车场；可视化；便捷；数据统计；多区域管理

# DESIGN OF A VISUAL PARKING LOT MANAGEMENT SYSTEM BASED ON THE WEB

## ABSTRACT

With the development of modern parking lots, the functions and interfaces of the existing parking lot management system are too old to meet the needs of car owners and administrators. For example, the equipment of the parking lot management system is incompatible with each other, the owner can't pay or choose the parking space independently, and the administrator can't manage multiple parking lots. In order to solve the above problems and meet the needs of users for a multifunctional, convenient and fast parking lot management system. This system has designed the following functional modules to meet the requirements; First of all, this system manages and charges parking lots in different areas for companies or communities and collects and displays unified data, so that all kinds of data of parking lots can be seen at a glance; At the same time, it supports multi-terminal login and adaptation of computers and mobile phones, and is equipped with visual monitoring, parking management, owner management, temporary car management and owner module; This system breaks through the situation that the traditional parking lot management system is only managed by the administrator, increases the role of the owner, strengthens the owner's perception of parking and facilitates the owner's payment and parking space selection. According to the design idea of multi-area parking lot management, the area table is established to divide the parking lot areas, and visual cameras are configured for different parking lot areas to carry out multi-area vehicle monitoring and management. This system adopts the design idea of separating the front end from the back end, in which the front end uses Vue framework, the back end uses the express framework of node.js, and the UI uses the ElementUI open source component library. This design choice can provide convenient, fast and convenient services.

The implementation of this system provides users with an efficient, reliable and fully functional parking lot management system.

**Key words:** parking lot; visualization; convenient; data statistics; multi-regional management

## 目 录

1 绪论 .....	1
1.1 课题背景与意义 .....	1
1.1.1 课题开发背景 .....	1
1.1.2 课题开发意义 .....	2
1.2 课题现状与发展趋势 .....	3
1.2.1 现状 .....	3
1.2.2 发展趋势 .....	4
1.3 使用工具和技术 .....	5
1.3.1 Vue.js .....	5
1.3.2 Node.js .....	5
1.3.3 MySQL 数据库 .....	5
1.3.6 图表组件库 .....	6
1.4 开发及运行环境 .....	6
1.5 课题主要内容 .....	7
2 停车场管理系统需求分析 .....	8
2.1 系统设计目标 .....	8
2.2 系统可行性分析 .....	8
2.3 系统功能性需求 .....	9
2.4 系统性能需求 .....	10
3 停车场管理系统总体设计 .....	10
3.1 停车场管理系统结构 .....	11
3.2 停车场管理系统功能模块 .....	11
3.3 数据库设计 .....	12
3.3.1 数据库概念结构设计 .....	12
3.3.2 主要数据表结构 .....	13
4 停车场管理系统详细设计及实现 .....	16

4.1 注册界面设计及实现 .....	16
4.2 登录界面的设计及实现 .....	17
4.3 车位管理功能模块设计及实现 .....	19
4.4 区域管理模块设计及实现 .....	21
4.5 临时车登记模块设计及实现 .....	22
4.6 车主信息模块设计及实现 .....	24
4.7 临时车管理模块设计及实现 .....	26
4.8 可视化监控摄像模块设计及实现 .....	27
4.9 个人资料界面设计及实现 .....	29
5 系统测试 .....	30
5.1 测试目标及任务 .....	30
5.2 系统功能测试 .....	30
5.2.1 注册页面功能测试 .....	31
5.2.2 登录页面功能测试 .....	32
5.2.3 车位管理模块测试 .....	34
5.2.4 车位管理模块测试 .....	36
5.2.5 临时车登记模块测试 .....	37
5.2.6 车主信息模块测试 .....	38
5.2.7 临时车管理模块测试 .....	39
5.2.8 可视化监控模块测试 .....	40
5.2.9 个人资料模块测试 .....	42
6 总结及展望 .....	44
参考文献 .....	45
致谢 .....	46
附录 部分程序源代码 .....	47



## 1 绪论

### 1.1 课题背景与意义

#### 1.1.1 课题开发背景

停车场管理系统是现代城市交通管理系统中的重要组成部分。随着城市化进程的加快，私家车数量快速增长，因此，智能停车场管理系统越来越受到人们的关注和追求。基于 Web 的可视化停车场管理系统应运而生。

随着信息技术的快速发展，许多传统的停车场管理模式已经无法满足人们的需求。传统的停车场管理需要大量的人员和物力投入，操作难度大，并且很难有效地应对突发情况。基于 Web 的可视化停车场管理系统通过物联网技术和云计算技术的应用，实现了停车场的信息化和智能化。

基于 Web 的可视化停车场管理系统可以实现多种管理功能，如安全监控、收费管理、车位预定等。使用这种系统可以大大提高停车场的运营效率，减少运营成本，为车主提供更加智能、便捷的停车服务<sup>[1]</sup>。

基于 Web 的可视化停车场管理系统作为一种最新的停车场管理方式，具有优越的管理和实时控制能力，使得停车运营效率更高。首先，停车场管理人员可以通过 Web 界面掌握停车场现状并进行有效管理。其次，车主可以使用智能设备进行预定停车位、在线缴纳停车费用等，并可通过 Web 界面实时了解停车场的状况和选择最优的停车位。

具体来说，基于 Web 的可视化停车场管理系统能够实现以下功能：

1.实时监控、管理停车场：通过 Web 界面，停车场管理员可以对停车场内的车位使用情况进行实时监控，了解空余车位数量、车辆停留时间、车位占用情况等信息，并进行有效的管理和调度操作，使车位利用率更高。

2.预订停车位服务：车主可通过 Web 界面进行预订车位，实现停车定位服务，并根据车辆类型和停车时间的不同，预订不同的停车位类型，并进行在线支付、退款等操作。

3.实现停车费用的在线缴纳：车主可以通过 Web 界面进行在线缴纳停车费用，并查询停车费用信息、历史账单等。数据分析、指标监控：停车场管理人员通过 Web 界面可以进行对停车收费、车位占用率等指标的查询，以及停车场运营效率的监控、数据分析等业务操作。

4.数据分析、指标监控：停车场管理人员通过 Web 界面可以进行对停车收费、车位占用率等指标的查询，以及停车场运营效率的监控、数据分析等业务操作。

5.数据分析、指标监控:停车场管理人员通过 Web 界面可以进行对停车收费、车位占用率等指标的查询，以及停车场运营效率的监控、数据分析等业务操作。

总的来说，基于 Web 的可视化停车场管理系统具有许多优点，如提高运营效率、实时监控和调度车位使用、简化操作流程等，可以缓解城市停车难问题，提高人们的生活质量。需要注意的是，开发该系统还需考虑车主信息、停车数据等的安全和保护。

### 1.1.2 课题开发意义

可视化停车场管理系统的开发有着极其重要的意义。随着现代交通工具的不断普及，停车场已成为城市交通建设中的重要配套设施。然而，传统的停车场管理方式存在许多问题，如车位管理困难、车主体验不佳、停车场资源浪费等。而可视化停车场管理系统正是为了解决这些问题而开发的。具体来说：

首先，可视化停车场管理系统可以提高停车场的管理效率。传统的停车场管理方式人工操作多，难以管理，容易发生车辆混乱、交通压力大等问题，严重影响城市交通效率。通过可视化停车场管理系统，可以对停车情况进行实时监控和管理，车位利用率得到了大幅提升，大大提高了停车场管理效率

其次，可视化停车场管理系统可以优化车辆管理和导航。系统不仅可以管理车辆，还可以为车辆提供车位导航、路径规划等服务，节省了车主寻找车位的时间和精力，提高了车辆管理水平和车主体验。

第三，可视化停车场管理系统可以实现智能化和安全性。通过使用传感器、摄像头等智能设备及数据分析技术来进行智能管理、车位识别，实现车辆安防智能化，提高了停车场的安全性和管理能力，进一步提高了城市交通治理水平。

最后，可视化停车场管理系统还能起到节能环保的作用。该系统可以将车位的重复使用率最大化，减少车辆拥堵和空转，节约了能源资源的使用，达到了环境友好和节能的目的，有助于实现城市可持续发展<sup>[2]</sup>。

因此，可视化停车场管理系统开发的意义不仅体现在提高城市交通治理水平、提高城市停车管理效率、优化城市道路资源利用等方面，同时也是推动城市智能化和可持续发展的一项重要举措。

## 1.2 课题现状与发展趋势

### 1.2.1 现状

可视化技术研究在海量医学数据、社交媒体数据或商业数据的可视化应用方面发挥着重要作用。这一需求源自车主终端计算能力的相对不足，比如手机处理能力不够，或者需要可视化的数据量太大。同时，由于传输带宽、延迟时间或者本地存储的限制，即使终端计算能力足够，这些海量的数据也难以实现高效传输。况且，在某些特殊场景下，一些敏感数据或者保密原始数据也并不适合直接向其他人开放。鉴于带宽、网络延迟等问题成为远程可视化的主要瓶颈，过去有大量研究和技術集中于解决这些问题。

尽管过去的研究成绩斐然，但数据量的不断增长和硬件设备的持续发展，使得远程可视化依然一直是研究的热点。由于天然的跨平台属性、以及成为未来协作平台的潜力，基于 Web 的可视化技术在该领域表现得尤为突出<sup>[3]</sup>。

基于 Web 的方式使得一套可视化工具代码可以跨平台执行，这不仅让团队间的协作和分享更加便捷，并且降低了程序的维护复杂度，使得可视化领域的研究者和其他应用行业领域的研究者可以更专注于研究各自领域的核心问题。更进一步，基于 Web 的方式使得各应用领域的研究者可以随时获得最新的数据（只需要刷新页面即可）。同时，这种可视化研究者和行业应用研究者的协同工作方式，更有助于促进可视化研究成果的落地应用<sup>[4]</sup>。

停车场管理其实也就是就是信息化管理。在停车场的信息化中，涉及停车场车位管理的信息化、车主管理的信息化、停车信息管理的信息化等，做好信息化工作有利于提高停车场的管理效率，提升停车场的竞争力<sup>[5]</sup>。

要提高竞争力，既要有好的硬件环境，同时也要有好的客户服务，而要实现这些目标的办法就是信息化。本选题，针对停车场过程中的一些信息，对其进行信息

化处理。在停车场过程中，主要涉及固定车位信息、自由车位信息、车主信息、停车信息等，通过对以上各个环节的信息化，提高档案的规范化管理，为以后的查询、统计作下基础。

传统的停车场管理系统无法实时的动态处理、对于车主的信息也无法进行很好地保护、对于管理者的权限处理不当等问题，本系统都有很好地处理这些问题。其中的系统信息管理功能、车位信息管理功能解决了这个问题。系统信息管理功能包括添加角色信息、管理角色信息、添加车主信息、管理车主信息、管理角色信息和管理车主信息模块可以对角色和车主进行查询、编辑和删除。车位信息管理包括添加车位信息、管理车位信息，管理车位信息模块可以对车主车位信息进行查询、编辑和删除

收费可以分为现金收费和月收费，一般是 VIP 车主进行月收费，临时车主进行现金收费，软件管理实行分级权限制。对出口值班员来讲，其登录后可进入收费管理，期间该出口所有收费均自动记入该值班员名下并存入电脑数据库。由于值班员受权限限制，不能进入系统中更高的软件菜单项，所以对电脑所记录的数据无法干涉；上级管理者可以随时查询，核对或打印一个值班段或任何一段时间乃至整个停车场的工作记录<sup>[6]</sup>。

### 1.2.2 发展趋势

1. 通过联网共享数据，实现停车场数据智能化，打破停车场信息孤岛，实现车位导航、车位预定、线上支付等。

2. 普及停车诱导、车位引导和反向寻车系统。通过诱导屏的实际投用，大大减少车主寻找车位时间，缓解部分交通拥堵，利用车位引导以及反向停车系统，引导车主短停快走，加大停车场车辆流通率。

3. 自动化程度增高，并趋向无人值守模式。传统停车管理，人工干预程度较高，智能化的停车管理随着智能化程度越来越高，逐渐减少人工参与，直到无需人工干预的情况下，通过智能化管理系统建立实现快速便捷的车辆进出场通道。

4. 移动智能终端技术的成熟，使得移动互联网车主超过了固定互联网车主，利用手机进行订餐、网购、订票等服务，已成为日常生活中必不可少的一部分，那么通过移动终端实现位预定、支付、寻车等功能也是必然趋势<sup>[7]</sup>。

## 1.3 使用工具和技术

### 1.3.1 Vue.js

Vue.js 是一款轻量级 JS 框架，用于构建优秀的车主界面和单页应用程序。相较于其他框架，Vue.js 更侧重于视图层的开发，提供非常简单、灵活的 API 和强大的响应式数据绑定，以实现高效、可维护的 Web 应用程序的构建。

Vue.js 具有易于上手、陡峭的学习曲线、良好的性能和简洁清晰的文档等优点，因此成为了众多开发人员首选的框架之一。此外，Vue.js 还有一个庞大的社区，开发者们可以通过社区获取与 Vue.js 相关的开发工具、插件、库、教程和示例等信息，从而提高开发的效率和质量，同时大大降低了构建 Web 应用程序的难度。

总之，Vue.js 的出现极大地推动了 Web 应用程序的开发，提供了更优异的开发体验、更高的编程效果。

### 1.3.2 Node.js

Node.js 是一种服务器端 JavaScript 运行时环境，采用事件驱动、非阻塞 I/O 模型，能高效地处理大量并发请求，提供出色的性能表现。Node.js 基于 Chrome V8 引擎构建，具备开发速度快、学习曲线低、可扩展性好、社区庞大和生态系统繁荣等优点。

相比传统的服务器端应用程序（如 Apache 和 Nginx）等，Node.js 在处理高并发、低延迟、数据密集型应用程序时提供了显著的性能优势。此外，它还能轻松地集成其他组件，例如数据库和缓存服务，具有良好的可扩展性和灵活性。

Node.js 在 Web 应用程序、命令行工具、网络工具和移动应用程序等领域得到广泛应用，被认为是开发效率高、代码易于维护、架构灵活、性能卓越的技术方案之一。许多知名公司，例如 Netflix、PayPal 和 LinkedIn 等，也采用 Node.js 来构建其 Web 应用程序。总之，Node.js 的出现为服务器端 JavaScript 开发提供了一种全新的范式，拓宽了开发者们的可能性，提升了开发效率和代码质量。

### 1.3.3 MySQL 数据库

MySQL 是一种开源关系型数据库管理系统，广泛应用于互联网应用领域。MySQL 基于客户端/服务器模式，支持多个操作系统和编程语言，同时提供高性能、高可靠性和高可用性等功能。MySQL 操作简便，可以轻松地安装和配置，同时还支持多种存储引擎和数据类型，具有高度的灵活性和可扩展性。MySQL 同时提供完善



的安全性控制功能，包括密码保护、数据加密、权限控制等，保证了数据的安全性和稳定性。MySQL 还支持分布式数据库环境和集群系统，可以处理海量的数据，并保证在多车主并发访问时的数据安全。总之，MySQL 是一个高性能、高可靠性、高灵活性和易于学习和使用的数据库管理系统<sup>[10]</sup>。

#### 1.3.4 ElementUI

ElementUI 是一种基于 Vue.js 框架的 UI 组件库，可以使 Web 应用程序的开发变得更加容易和高效。ElementUI 拥有丰富的 UI 组件、多种主题和国际化支持，使得它很容易被集成和使用。此外，ElementUI 采用 vue.js 风格的组件化开发方式，因此可以轻松嵌入到任何基于 vue.js 框架的项目中。另外，ElementUI 的图标库也是非常丰富的，可以增强页面的美观度和车主识别性。总之，ElementUI 是一种实用易用的 UI 组件库，为 Web 开发人员提供了更快、更高效的开发和设计工具。

#### 1.3.5 Webpack

Webpack 是一种现代化的打包工具，可以将多个前端资源文件打包成一个或多个文件，并提高 Web 应用程序的性能和效率。Webpack 支持各种不同的模块加载器和插件，可以打包各种类型的前端代码，并且可以自定义打包过程以满足不同的需求。Webpack 可以将应用程序的所有资源合并为几个文件，减少请求次数、提高整个应用程序的性能。Webpack 还支持 Code Splitting，即拆分大型应用程序为多个小文件，利用浏览器的并行加载机制加快应用程序的加载速度。

Webpack 的配置文件是一个 JavaScript 对象，开发人员根据需要自定义各种设置，例如入口文件、输出文件、加载器和插件等。Webpack 的高度可定制性和扩展性使得它成为编写现代 Web 应用程序的优秀工具之一。

#### 1.3.6 图表组件库

G2 是一个简洁的渐进式语法，主要用于制作基于网页的可视化。它提供了一套函数风格、声明形式的 API 和组件化的编程范式，希望能帮助车主能快速完成报表搭建、数据探索、可视化叙事等多样化的需求。

### 1.4 开发及运行环境

操作系统：Windows 11。

开发工具：VSCODE。

开发语言：node、JavaScript。

数据库：MySQL5.5。

浏览器：Chrome 100.0.4896.75。

摄像头：1080p FHD 摄像头

## 1.5 课题主要内容

在此次我的毕业设计期间，我不仅翻阅了很多的文献，完成了外文文献的翻译工作和撰写开题报告，还完成了整个系统的设计与实现。从需求分析入手，进行系统设计、详细设计、测试等工作，最后得出结论。

本次撰写的论文主要划分了六大章，每章下面又有很多小节，具体的结构安排如下：

（1）绪论：该部分讲述的是选题的背景、开发该项目的目的意义，以及车辆信息管理系统的发展现状和发展趋势，使用的工具技术，开发和运行环境，是论文的开端，为论文的下面部分奠定基础。

（2）需求分析：根据设计目标，完成可行性分析，同时进行详细的性能需求分析和功能性需求分析。

（3）总体设计：系统结构设计、功能模块设计和数据库的设计工作。

（4）系统详细设计及实现：对开发完成的程序先进行功能概述，完成页面设计图，最后加上具体的实现逻辑。

（5）系统测试：完成程序测试工作，对测试目的、方法等进行阐述，并对系统进行功能性测试，最后系统效果演示。

（6）总结及展望：对整个程序开发过程、论文写作过程进行归纳总结。

## 2 停车场管理系统需求分析

### 2.1 系统设计目标

基于 Web 的可视化停车管理系统的设计目标是提升停车场管理的效率和服务质量。它能够实现实时监控停车场、自动收费、方便车主预约车位、提供优惠奖励以及统计分析停车场数据等功能。此外，系统具有良好的可扩展性，可以方便地增加新的设备和功能，保障系统的长期使用。通过该系统，停车场管理者可以更好地管理车位，提高车主满意度和停车场的利润<sup>[7]</sup>。

### 2.2 系统可行性分析

#### （1）技术可行性

基于 Web 的可视化停车场管理系统的技术可行性是很高的，这主要是基于以下技术支持：

1. Web 技术：Web 技术是依赖于互联网的技术，使用 Web 技术可以轻松地在任何地方访问停车场管理系统，同时还支持多种设备上的使用。

2. 云计算技术：通过云计算技术，系统可以实现分布式管理，数据中心管理，存储分离等多种功能，使得停车场管理系统更为便捷和稳定。

3. 物联网技术：停车场管理系统可以通过物联网技术实现停车位的实时监控、车辆的快速检测和统计、停车费的自动计算等功能，从而提高管理效率。

4. 数据库技术：采用数据库等技术，实现对停车场数据的高效管理和快速查询，并且利用数据分析，提高系统的安全性和管理效率。

5. 可视化技术：采用可视化技术，可以实现停车位状态实时监测、车辆信息的实时显示等功能，使车主可以直观地了解停车场实时情况，提高管理效率。

因此，基于 Web 的可视化停车场管理系统的技术可行性非常高，并且已被实际应用于现实生活中的停车场管理中，取得了良好的效果<sup>[8]</sup>。

#### （2）经济可行性

基于 Web 的可视化停车场管理系统的经济可行性也是很高的，主要因为以下几点：



1. 降低开发成本：本系统为本人独立开发，无需支付额外人力费用，且服务器为本地搭建，无需支付云服务器的费用。

2. 使用开源软件：开发过程中全部使用的开源软件以及开源库，不需要支付额外的专利费用。

### （3）操作可行性

程序在操作上是否可行，这点是以使用者的立场来考虑的，使用者在使用操作该程序的时候，能不能无障碍式地来操作使用每一个功能模块，要做到这一点，就必须要保证程序的页面整齐明了，导航栏清晰，让人在进入程序之后，能够对程序的功能一目了然，并可以根据导航栏的提示来一步一步完成功能上的操作。对于设计的基于 Web 的可视化停车场管理系统，也充分考虑了这一点，界面简单易懂，通过侧边栏进行引导以及区分管理员和车主界面；操作无需消耗过多心智<sup>[9]</sup>。

## 2.3 系统功能性需求

经过调查实际车辆信息管理的情况，分析了车辆信息管理的功能需求，对要开发建设的基于 Web 的可视化停车场管理系统进行了详细的分析。本系统应具备以下功能：

（1）注册登录：车主或者管理员需要先登录系统才能对该系统进行操作；管理员有对应的邀请码。

（2）图表界面：管理员进入首页后即可看见图表展示，不同的图表展示不同的数据；一目了然且便于数据收集。

（3）可视化监控区域：管理员能够通过摄像头来监控到不同停车区域内的情况，且管理员可以截图摄像头内容进行保存。

（4）车位管理：作为停车场的基本功能，为每个区域划分车位。

（5）区域管理：对于不同的企业或者社区，停车场可能以区域作为区分，不同区域的收费或者车位数量上存在着差异；管理员可以对区域进行区分管理。

（6）临时车登记：作为停车场管理系统，临时车登记是不可缺少的部分，车辆进行登记后也会占用相关车位来达到一个车位同步效果。

（7）车主信息：车主在进行注册后，可以在移动端进行登录，进入后可以看见自己的跟人信息以及车位是否到期，同时在管理员侧可以查询所有车主的信息以及进行相关修改。

(8) 临时车管理：进入小区或者公司的车辆会进入该管理页面，可以统计该车辆驶入驶出时间、车辆和车位信息和停车产生的费用。

(9) 车主管理：系统车主繁多，将其分为两个角色，分别是管理员和车主；不同角色拥有不同的操作权限以及界面。

(10) 个人信息编辑：如果个人数据不正确，可以很容易地进行修改。

(11) 续费管理：车主可以自己选择自己的车位并在移动端进行缴费操作来延长车位有效期。

## 2.4 系统性能需求

1. 响应快速：系统需要具有快速响应的能力，确保车主能够快速预约车位、支付等操作，也能够及时地获取车位状态、停车场数据等信息。

2. 稳定可靠：系统需要具有高可靠性和稳定性，保证数据不丢失，功能正常；并且需要具备备份和恢复功能，保障数据安全。

3. 可扩展性：系统需要具备良好的可扩展性，能够根据实际需求方便地添加新功能、新设备，并支持分布式系统的部署。

4. 安全可靠：系统需要具备完善的安全性能，包括车主的身份验证、数据传输的加密等功能，保障车主信息不被泄露。

5. 高并发性：停车场常常会出现高并发的情况，系统需要具有高并发处理能力，以确保系统的稳定和快速响应。同时，系统需要具备负载均衡、缓存技术等手段，提高系统性能。

综上，基于 Web 的可视化停车场管理系统需要拥有快速响应、稳定可靠、可扩展性、安全可靠以及高并发性等性能需求，以提供高效的停车场管理服务<sup>[10]</sup>。

### 3 停车场管理系统总体设计

#### 3.1 停车场管理系统结构

基于 Web 的可视化停车场管理系统的角色划分有两种，管理员和车主，不同的角色拥有着完全不同的界面，在进入到自己的功能页面后可以根据导航栏来一一实现相应的功能的查看、管理。可以轻松构建出系统结构图，如图 3.1 所示：

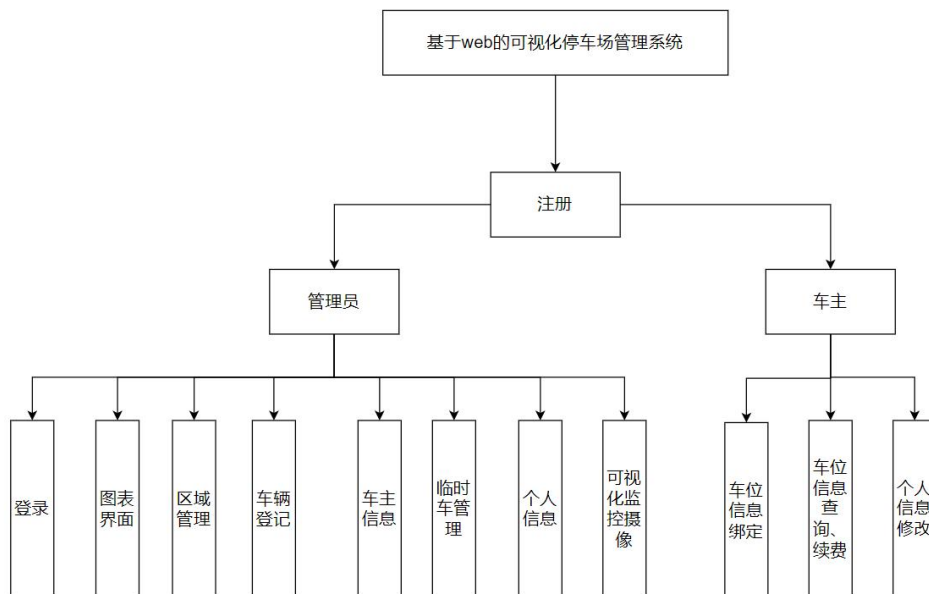


图 3.1 系统结构图

#### 3.2 停车场管理系统功能模块

本管理系统的设计初步规划为两个模块：

##### （1）管理员：

图表管理：通过收集的数据来进行图表的展示；包括停车位信息、收费信息、临时车出入信息等。

车位管理：管理员能够对车位根据不同区域进行新增，根据车位编号进行车位查询以及可以修改该车位的基本信息，包括是否空闲或者固定以及它所在的区域。

区域管理：管理员可以新增区域以及修改或者删除相关区域信息，包括区域内停车位的小时、月、永久的价位以及相关区域描述。

临时车登记：管理员对于临时车可以进行登记以及相关车位选择。

车主信息：车主信息会在车主进行注册后自动同步过来，管理员可以查看所有车主的相关信息以及可以修改和帮车主进行续费操作。

临时车管理：管理员可在该页面对所有驶入车进行出库操作，也可以查询所有已经出库的车的相关信息。

可视化监控摄像管理：管理员可在该页面进行摄像头的连接以及监控停车区域内情况，也可以对某个摄像头进行截图操作将该画面保存下来，供其余管理员查看。

个人资料：该页面管理员可以进行个人资料的修改。

## （2）车主：

车位信息：车主进入主界面后，如果未绑定相关车位的话则展示尽快绑定的提示信息，如果已绑定则会展示车位详细信息以及展示日历控件来直观的提示车主车位具体到期时间。

个人资料：车主可以修改自己的个人资料信息。

## 3.3 数据库设计

### 3.3.1 数据库概念结构设计

图表界面，包含的数据项目有：车位空闲状态、收费总额、停车区域、临时车驶入驶出信息。

车位管理，包含的数据项目有：车位编号、车牌号码、车主名称、是否占位、是否固定车位、到期时间。

区域管理信息，包含的数据项目有：区域名称、车位描述、小时收费、月租金、永久车位售价。

临时车登记，包含的数据项目有：车牌号码、占用车位、停车区域。

车主信息，包含的数据项目有：车主编号、头像、角色、车主名、密码、昵称、个性签名、性别、电话。

临时车管理信息，包含的数据项目有：编号、车牌号码、驶入时间、驶出时间、停车区域、车位编号、车辆状态、总价格。

管理员界面，包含的数据项目有：车主昵称、个性签名、头像、加入时间、住址。

可视化监控区域，包含的数据由：监控区域、摄像头信息、是否处于开启状态。

数据库的概念结构的设计是继需求分析结束后要进行的一步，它主要是将需求分析获得来的车主的需求，来转化为抽象的模型，这样做更加容易被理解。下面构建该基于 Web 的可视化停车场管理系统的概念模型——E-R 图，系统 E-R 图如图 3.2 所示。

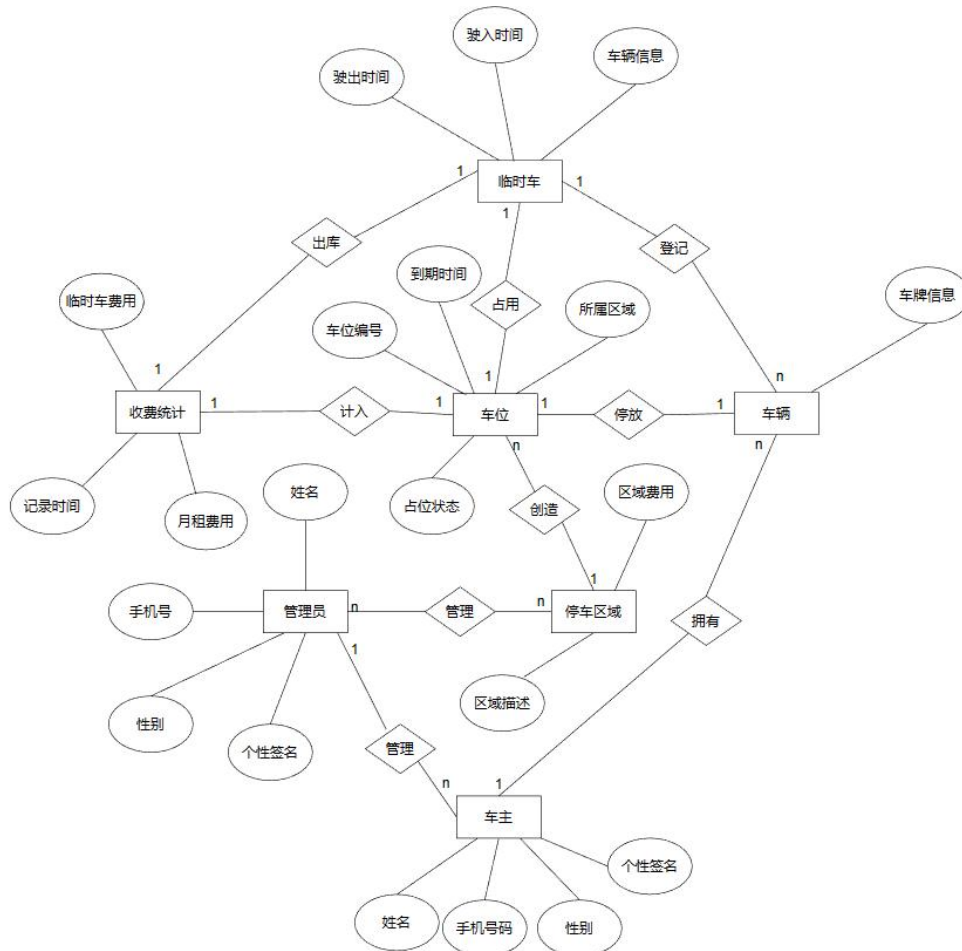


图 3.2 停车场管理系统 E-R 图

### 3.3.2 主要数据表结构

将前面做的概念模型图来一一转化为对应的数据库表结构，能够得到以下几个主要的表信息，包括人员登录表、车位信息表、车主信息表、进出记录表、收费详细表和停车区域表具体的表结构如下：

表 3-1 人员登录表

列名	数据类型	长度	允许空	主键	外键	说明
AdminId	int	11	否	是	否	唯一 ID
username	varchar	20	否	否	是	车主信息的 Phone
password	varchar	20	否	否	否	密码
usingimg	varchar	200	是	否	否	车主头像
nickname	datetime	255	是	否	否	昵称
sex	varchar	10	是	否	否	性别
address	varchar	255	是	否	否	地址
privatemes	varchar	255	是	否	否	个性签名
jointime	date	0	否	否	否	加入时间
IsAdmin	char	255	否	否	否	是否是管理员

表 3-2 车位信息表

列名	数据类型	长度	允许空	主键	外键	说明
CarportNumber	varchar	0	否	是	是	车位编号
HaveCar	tinyint	1	否	否	是	是否占位
CarCost	tinyint	255	否	否	否	收费标准
FixCar	tinyint	1	否	否	否	是否固定
CarpersonName	varchar	12	是	否	否	车主姓名
DueDate	datetime	0	是	否	否	到期时间
position	varchar	255	否	否	否	区域信息
CarNumber	varchar	255	是	否	否	车牌信息

表 3-3 车主信息表

列名	数据类型	长度	允许空	主键	外键	说明
CarNumber	varchar	255	是	否	是	车牌信息
CarportNumber	int	50	是	否	是	车牌编号
PersonName	varchar	1	是	否	否	删除状态(1 为已删除, 0 为未删除)
Adress	varchar	50	是	否	否	车主详细地址
Age	decimal	20	是	否	否	车主年龄
Phone	decimal	20	否	是	是	联系方式 人员登录的 username

表 3-4 进出记录表

列名	数据类型	长度	允许空	主键	外键	说明
Id	int	11	否	是	否	编号
CarNumber	varchar	20	否	否	是	车牌号码
ComeTime	datetime	6	是	否	是	进入日期
LeaveTime	datetime	0	是	否	否	离开日期
totalfare	varchar	50	是	否	否	费用
position	varchar	255	是	否	否	停车区域
CarportNumber	varchar	5	否	否	否	车位编号

续表 3-4

列名	数据类型	长度	允许空	主键	外键	说明
IsDone	int	0	否	否	否	是否完成
day	date	0	是	否	否	精确日期

表 3-5 收费详细表

列名	数据类型	长度	允许空	主键	外键	说明
id	int	10	否	是	否	编号
type	int	0	否	否	否	临时车或者固定车
fare	varchar	255	是	否	否	收费金额
date	date	0	是	否	否	收费日期

表 3-6 停车区域表

列名	数据类型	长度	允许空	主键	外键	说明
position	varchar	255	否	是	是	区域名称
fare	varchar	255	否	否	否	区域停车价格
decoration	varchar	255	是	否	否	区域描述
monthfare	varchar	40	否	否	否	每个月价格
fixedfare	varchar	255	否	否	否	永久车位价格

表 3-7 监控摄像表

列名	数据类型	长度	允许空	主键	外键	说明
id	int	0	否	是	否	唯一 id
url	varchar	255	否	否	否	监控地址
position	varchar	255	否	否	否	所属区域
isActive	int	0	否	否	否	是否开启
name	varchar	255	否	否	否	监控名称

表 3-8 监控图片表

列名	数据类型	长度	允许空	主键	外键	说明
id	int	0	否	是	否	唯一 id
url	varchar	255	否	否	否	图片地址
name	varchar	255	否	否	否	图片名称
time	datetime	255	否	否	否	截图时间



## 4 停车场管理系统详细设计及实现

### 4.1 注册界面设计及实现

注册功能是所有管理员和车主需要经历的步骤，让还没有自己账号的车主可以借助注册的方式得到属于自己的账号，密码不仅在前端页面做了隐私性设置，而且在数据库中采用了 MD5 信息摘要算法（MD5 Message-Digest Algorithm, MD5）对密码进行了加密，能最大程度地保证密码的安全性。车主注册页面由 Vue.js 的单页面搭建实现。具体流程图如图 4.1 所示，具体核心代码图如图 4.2 所示。

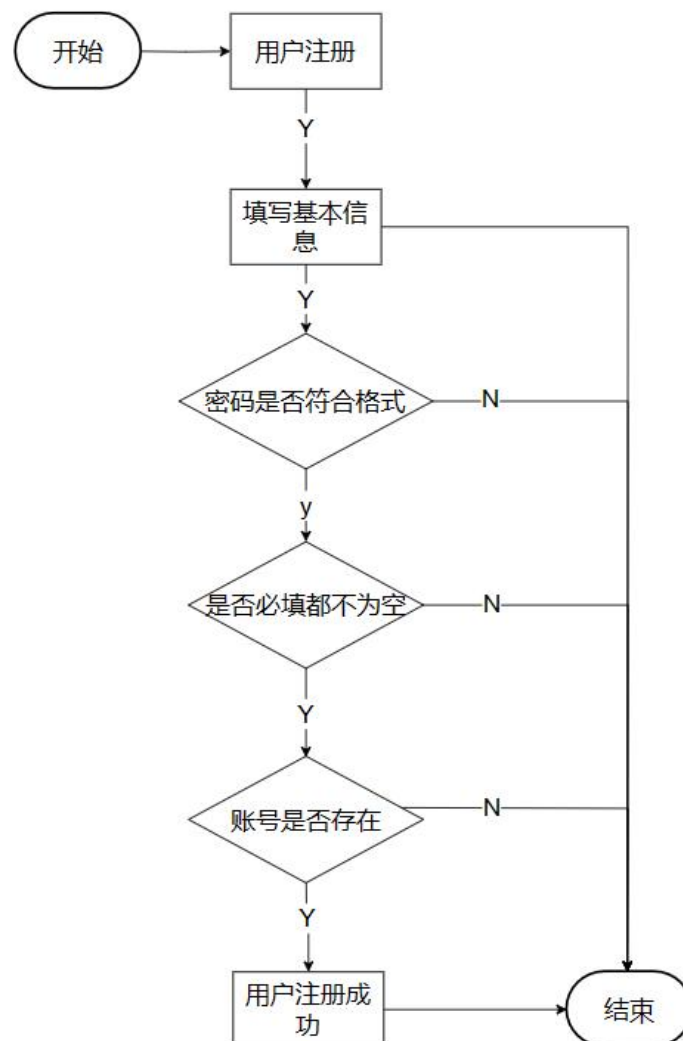


图 4.1 车主注册功能流程



```
//注册
register()
{
  if(this.username&&this.password1&&this.mycode)
  {
    if(this.password1.length>=8)
    {
      if(this.password1==this.password2)
      {
        this.$axios.post('/register',{
          username:this.username,
          password:this.password1,
          mycode:this.mycode
        }).then(res=>{
          console.log(res);
          this.messageBox(res.data.message,res.data.flag==1?'success':'error')
          if(res.data.flag==1)
          {
            this.$router.push('/login')
          }
        })
      }else{
        this.messageBox('两次输入密码不一致','warning')
      }
    }else{
      this.messageBox('密码长度要大于8位','warning')
    }
  }else{
    this.messageBox('信息填写不全','warning')
  }
}
```

图 4.2 车主注册核心代码

## 4.2 登录界面的设计及实现

基于 Web 的可视化停车场管理系统的登录页面，主要是用于来确认登录车主的身份的，系统中包含两类角色，选择输入登录账号都正确的情况下，服务端会通过该账号判断是否跳转管理员页面还是车主界面。流程图如图 4.2 所示，代码设计如图 4.3 所示。

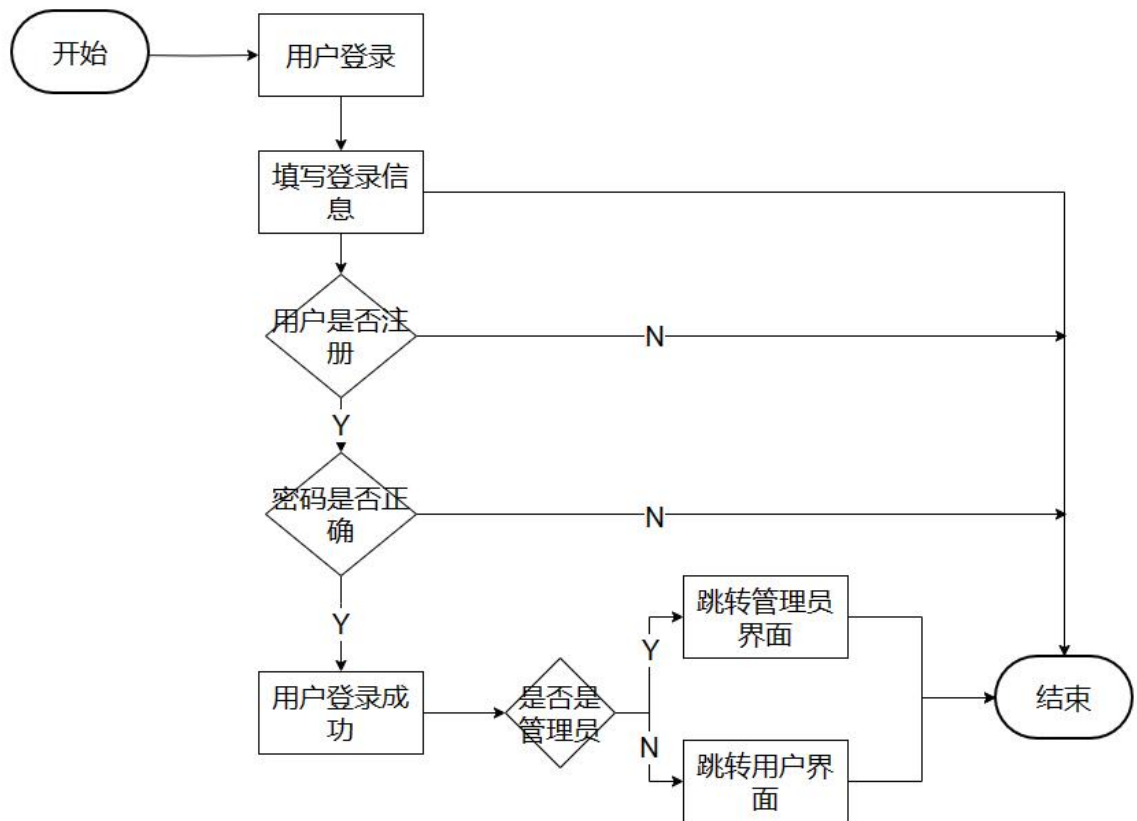


图 4.3 车主登录功能流程

```

this.$message({
  message: res.data.message,
  type: "success",
});
//sessionStorage存储
sessionStorage.setItem('adminname', res.data.result[0].nickname);
//存登录人员的id
sessionStorage.setItem('adminid', res.data.result[0].AdminId);
//存个人信息
sessionStorage.setItem('privatemes', res.data.result[0].privatemes);
//
sessionStorage.setItem('username', res.data.result[0].username);
//vuex里面存人员头像和名字
let getmyimg = 'http://localhost:3000/' + res.data.result[0].userimg.substring(6)
console.log(getmyimg);
this.$store.commit('changeimg', getmyimg)
this.$store.commit('changenname', res.data.result[0].nickname)
//也往sessionStorage里面存东西 防止页面刷新时候头像没了
sessionStorage.setItem('myimg', getmyimg);
//可以取到头像
// console.log(this.$store.state.myuserimg);
//给iphone发登录消息
// axios.get('https://api.day.app/sSA3xJGKscqrswYyeNgpvW/停车场网站/' + res.data.result[0].nickname + '登录了!')
console.log(res);
if (res.data.result[0].IsAdmin == 0) {
  console.log(1);
  this.$router.push("/user/home");
}
else this.$router.push("/mainshow");

```

图 4.4 车主登核心代码

### 4.3 车位管理功能模块设计及实现

车位管理功能模块是系统的管理者管理员所具有的功能，该功能模块用于停车场车位管理以及查找和修改车位信息，该模块可以在不同区域添加车位同时可以根据车位 id 进行车位查找；管理员也可以修改车位的信息以及车位是否处于空闲的状态，也可以删除现有的车位；车位的空闲或者占用状态由临时车登记或者车主的固定车位占有来改变；该模块由 Vue 以及 Node.js 进行前后端分离设计实现；图 4.5 展示的为车位含有的基本信息 图 4.6 为车位管理模块所具备的功能，图 4.7 为该模块的核心代码图。

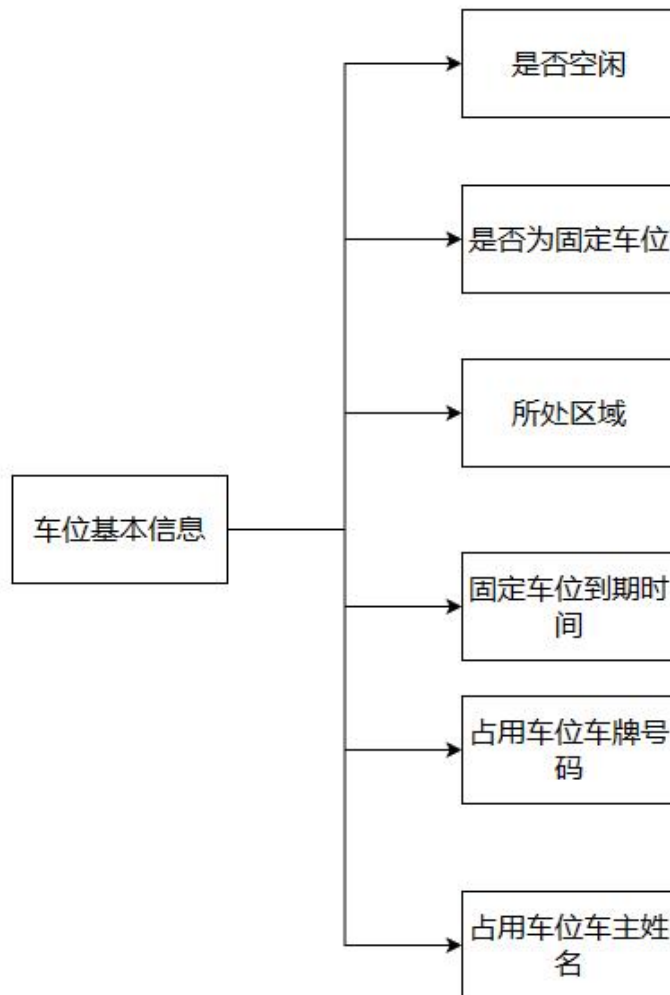


图 4.5 车位基本信息图

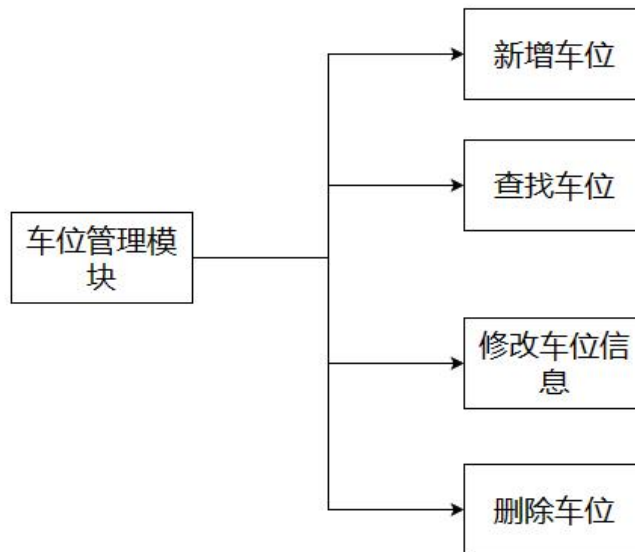


图 4.6 车位管理模块功能图

```

<div>
  <el-table :data="tableData"
    v-loading='loading'>
    <el-table-column prop="CarportNumber" label="车位编号" width="140">
    </el-table-column>
    <el-table-column prop="position" label="车位区域">
    </el-table-column>
    <el-table-column prop="CarCost" label="小时计费" >
    </el-table-column>

    <el-table-column prop="FixCar" label="是否固定车位" width="120">
      <template slot-scope="scope">
        <el-tag type="danger" v-if="scope.row.FixCar == 1">否</el-tag>
        <el-tag type="success" v-else>是</el-tag>
      </template>
    </el-table-column>

    <el-table-column prop="HaveCar" label="是否空闲" width="120">
      <template slot-scope="scope">
        <el-tag type="danger" v-if="scope.row.HaveCar == 0">否</el-tag>
        <el-tag type="success" v-else>是</el-tag>
      </template>
    </el-table-column>

    <el-table-column prop="CarNumber" label="车牌编号">
      <template slot-scope="scope">
        <span v-if="scope.row.CarNumber">{{scope.row.CarNumber}}</span>
        <span v-else style="color:grey">暂无车辆</span>
        <!-- {{ scope.row.CarNumber || "暂无车辆" }} -->
      </template>
    </el-table-column>

    <el-table-column prop="CarpersonName" label="车主姓名">
      <template slot-scope="scope">
        <span v-if="scope.row.CarpersonName">{{scope.row.CarpersonName}}</span>
        <span v-else style="color:grey">暂无车主</span>
        <!-- {{ scope.row.CarpersonName || "暂无车主" }} -->
      </template>
    </el-table-column>
  </div>
  
```

图 4.7 车位管理模块核心代码图

#### 4.4 区域管理模块设计及实现

车位可能会分布在不同的区域，而不同区域有着不一样的收费标准；管理员可以对区域进行增删改查；也可以修改不同区域的停车计费定价以及车位永久价格，修改后会同步到所有的该区域的车位；图 4.8 为区域基本信息图，图 4.9 为该模块功能图；图 4.10 为该模块核心代码图。

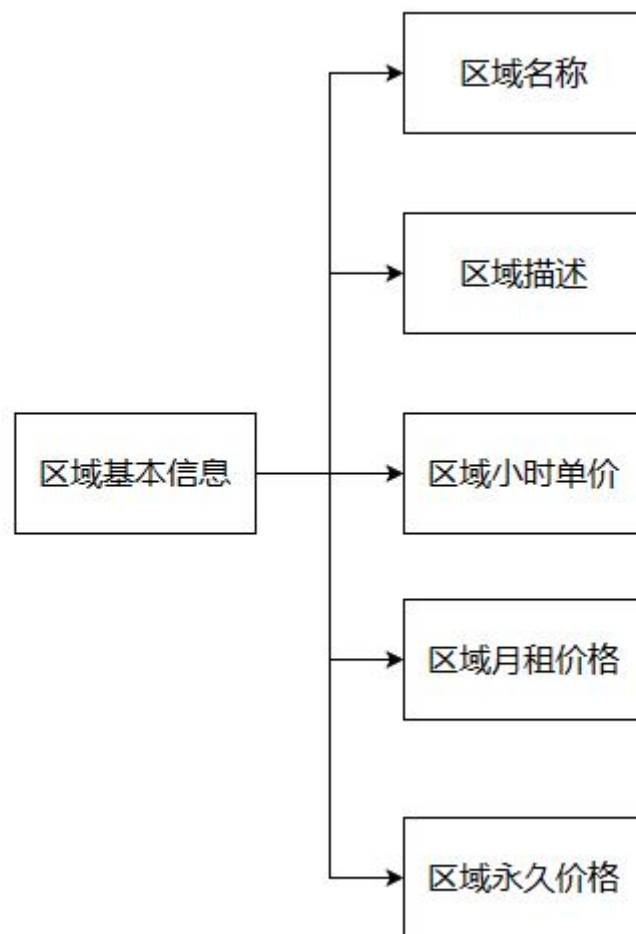


图 4.8 区域基本信息图

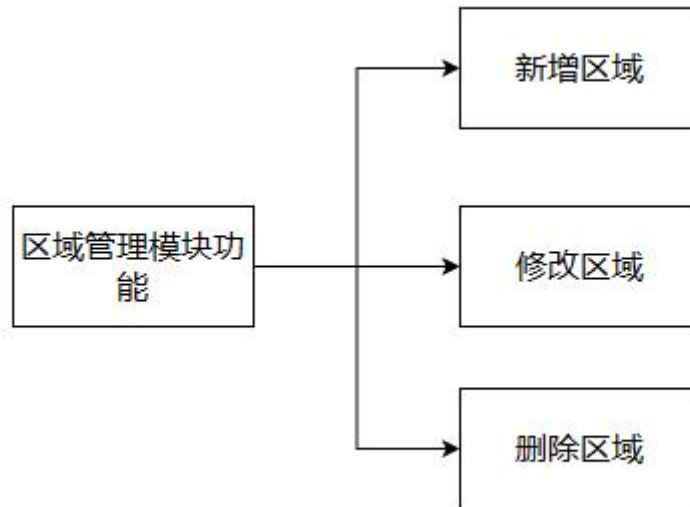


图 4.9 区域管理模块功能图

```

<!-- 停车区域展示表格 -->
<el-table :data="tableData">
  <el-table-column prop="position" label="停车区域" width="140">
  </el-table-column>
  <el-table-column prop="fare" label="区域价格">
    <template slot-scope="scope"> ¥{{ scope.row.fare }} </template>
  </el-table-column>
  <el-table-column prop="monthfare" label="月租价格">
    <template slot-scope="scope"> ¥{{ scope.row.monthfare }} </template>
  </el-table-column>
  <el-table-column prop="fixedfare" label="永久价格">
    <template slot-scope="scope"> ¥{{ scope.row.fixedfare }} </template>
  </el-table-column>
  <el-table-column prop="decoration" label="区域描述" width="350">
  </el-table-column>
</el-table>
  
```

图 4.10 区域管理模块核心代码图

## 4.5 临时车登记模块设计及实现

管理员可以通过该模块添加进入停车场的临时车，可以在不同区域内挑选出处于空闲状态下的车位来进行选择；后端通过过滤掉车位信息处于占位状态下的车位进行返回，同时在进行登记后将该车位也记录位占位状态；图 4.11 为临时车登记流程图；图 4.12 为该模块核心代码图。



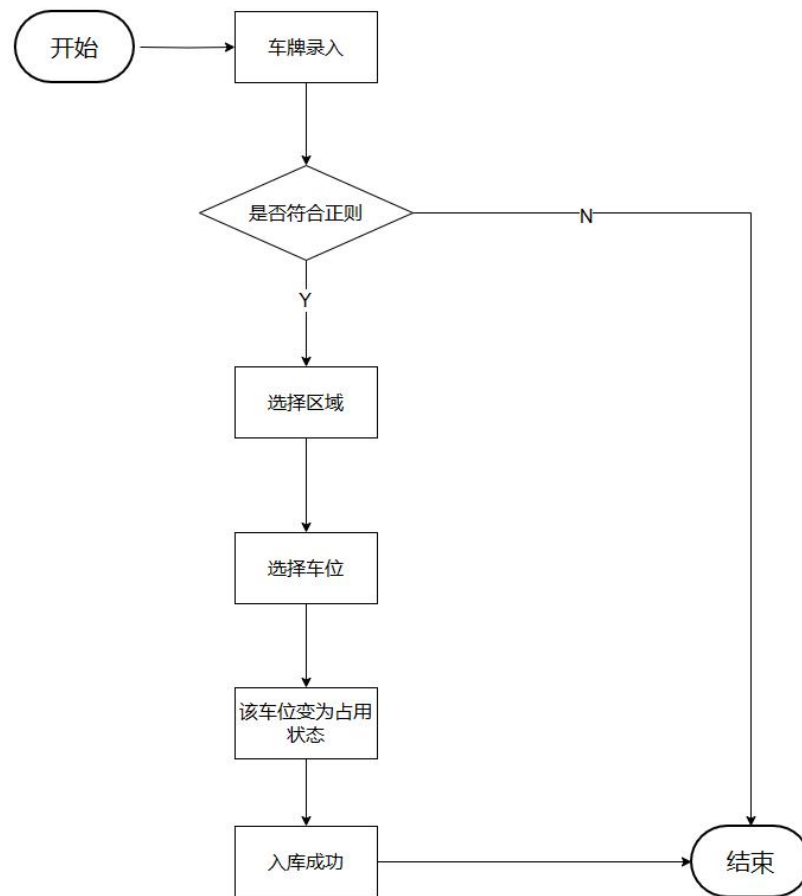


图 4.11 临时车登记流程图

```

<template>
  <div class="left">
    <p>进出车辆登记表</p>
    <div class="carjoinform">
      <table class="jointable">
        <tr>
          <td>车牌号码: </td>
          <td><el-input style="width:80px;float:left" v-model="carnumber1" placeholder="地区号"></el-input>
            <span>-</span>
            <el-input style="width:130px" v-model="carnumber2" placeholder="车牌号码"></el-input></td>
        </tr>
        <tr>
          <td>泊车区域: </td>
          <td>
            <el-select v-model="toaddposition" placeholder="请选择停车区域">
              <el-option
                v-for="item in positions"
                :key="item.description"
                :label="item.position"
                :value="item.position">
                <span style="float: left">{{ item.position }}</span>
                <span style="float: right; color: #8492a6; font-size: 13px">¥{{ item.fare }}</span>
              </el-option>
            </el-select>
          </td>
        </tr>
        <tr>
          <td>车位选择: </td>
          <td>
            <el-select v-model="postcarnumber" placeholder="请选择车位">

```

图 4.12 临时车登记模块核心代码图

#### 4.6 车主信息模块设计及实现

本模块提供管理员可见，用于统计所有的车主以及车主的车位信息，管理员也可以自己添加车主信息，但是添加的车主无法进行注册。注册的车主会自动同步到该表格，管理员可以对车主进行增删改查以及可以为车主进行车位的续费；前端在调用续费接口的时候需要先修改该车位的到期时间成功后再对本次收入进行统计；该模块基本信息图如图 4.13，功能图如图 4.1,4 所示，核心代码图如图 4.15 所示。

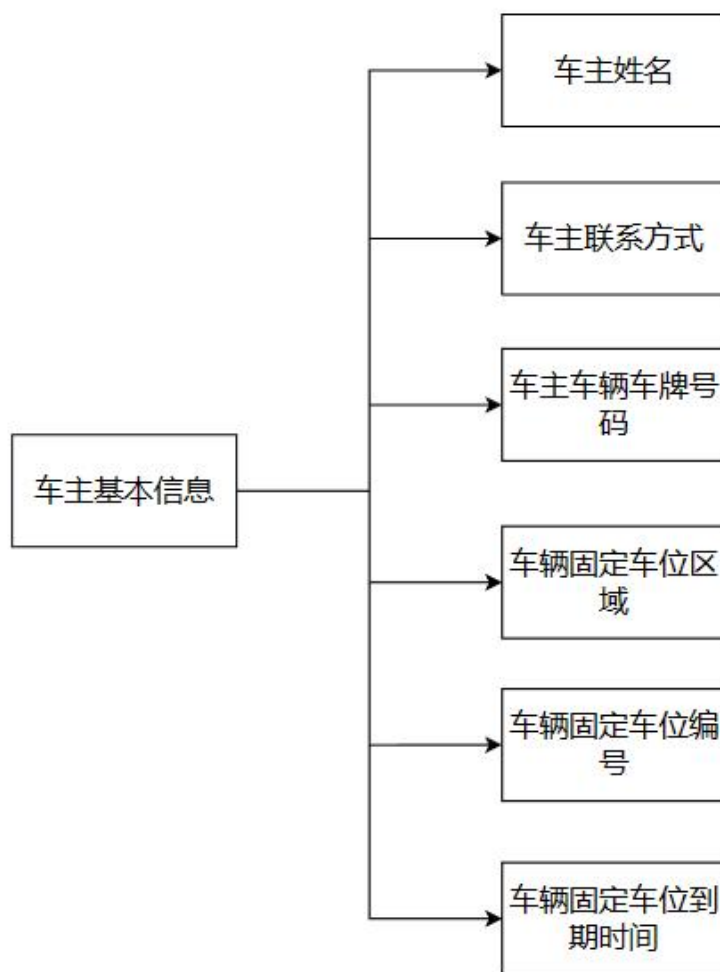


图 4.13 车主基本信息图



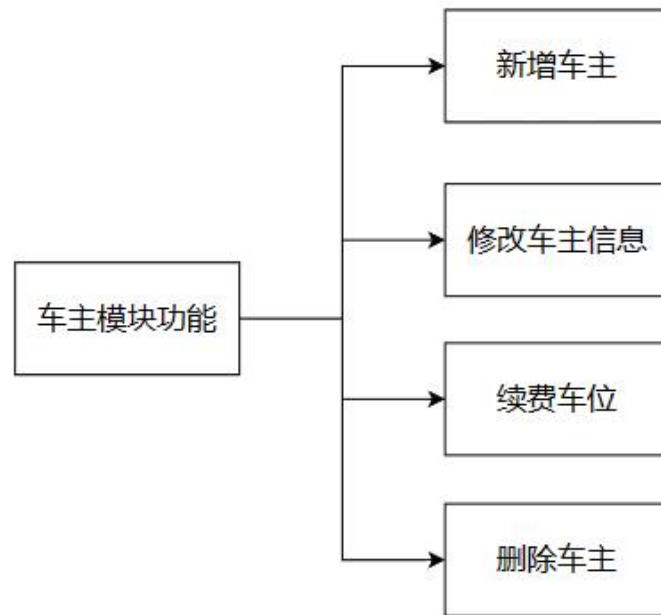


图 4.14 车主模块功能图

```

<el-table-column prop="PersonName" label="车主姓名" width="140">
</el-table-column>
<el-table-column prop="CarNumber" label="车牌号码" width="180">
</el-table-column>
<el-table-column prop="CarportNumber" label="车位编号" width="180">
</el-table-column>
<el-table-column prop="DueDate" label="到期时间" width="180">
  <template slot-scope="scope">
    <span v-if="scope.row.DueDate">{{
      scope.row.DueDate.substr(0, 10)
    }}</span>
    <span v-else><el-tag type="danger">已到期</el-tag></span>
  </template>
</el-table-column>
<el-table-column prop="address" label="地址"> </el-table-column>
<el-table-column prop="Age" label="年龄"> </el-table-column>
<el-table-column prop="Phone" label="手机号码"> </el-table-column>
<el-table-column label="操作" width="250">
  <template slot-scope="scope">
    <el-button size="mini" type="warning" @click="$event => toaddcartime(scope.row)"
      >续费</el-button>
    <el-button size="mini" type="primary">修改</el-button>
    <el-button
      size="mini"
      type="danger"
      @click="$event => deletepeoplemes(scope.row)"
      >删除</el-button>
  </template>
</el-table-column>

```

图 4.15 车主模块核心代码图

## 4.7 临时车管理模块设计及实现

管理员通过临时车登记模块登记的车辆会在本模块中展示，通过区分未出库和已出库的车辆信息，通过点击未出库车辆的出库按钮，会弹窗一个详细表单记录该车辆所有信息以及最终的金额，也可以根据车牌进行搜索；在进行出库操作前也会先保证该车位重置为空闲状态后才进行金额收费和统计；该模块流程图如图 4.16 所示，核心代码图如图 4.17 所示。

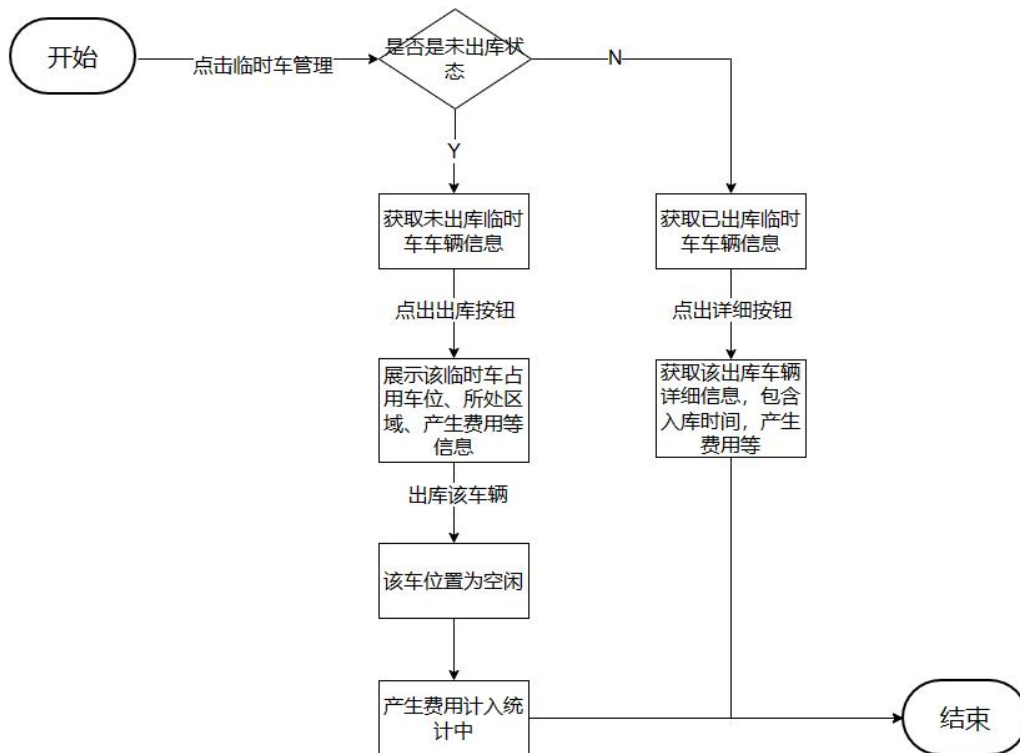


图 4.16 临时车管理流程图

```

<table id="carjointab">
  <tr><td><el-tag>价格</el-tag></td>
  <td><span style="color:grey">¥{{costlist.fare*costlist.hours}}</span></td>
</tr>
<tr><td><el-tag>小时单价</el-tag></td>
  <td><span style="color:grey">¥{{costlist.fare}}</span></td>
</tr>
</table>
<tr><td><el-tag>停车区域</el-tag></td>
  <td>{{costlist.position}}</td>
</tr>
<tr><td><el-tag>车位编号</el-tag></td>
  <td>{{costlist.carport}}</td>
</tr>
<tr>
  <td><el-tag>驶入时间</el-tag></td>
  <td>{{costlist.starttime}}</td>
</tr>
<tr>
  <td><el-tag>驶出时间</el-tag></td>
  <td>{{costlist.leavetime}}</td>
</tr>
</table>
<span slot="footer" class="dialog-footer">
  <el-button @click="$event => carjoinend = false">取消</el-button>
  <el-button type="primary" v-if="!DoneValue" @click="removecarjoin">驶出</el-button>
  <el-button type="primary" v-else @click="$event => carjoinend=false">确定</el-button>
</span>
</fan-dialog>
<div class="myslect">
  
```

图 4.17 临时车管理模块核心代码图

## 4.8 可视化监控摄像模块设计及实现

管理员通过进入监控摄像界面后可查看所有区域的监控摄像头信息，同时管理员也可以连接不同区域的摄像头；每个摄像头可以进行截图操作；管理员通过进入某个摄像头的拍照按钮后会将该图片从 base64 转化为 file 文件上传到服务端中，服务端将传递的文件进行存储后将该图片存储在服务端的地址落库，前端获取地址后对图片进行展示；具体流程图如图 4.18 所示；核心代码如图 4.19 、图 4.20 所示。

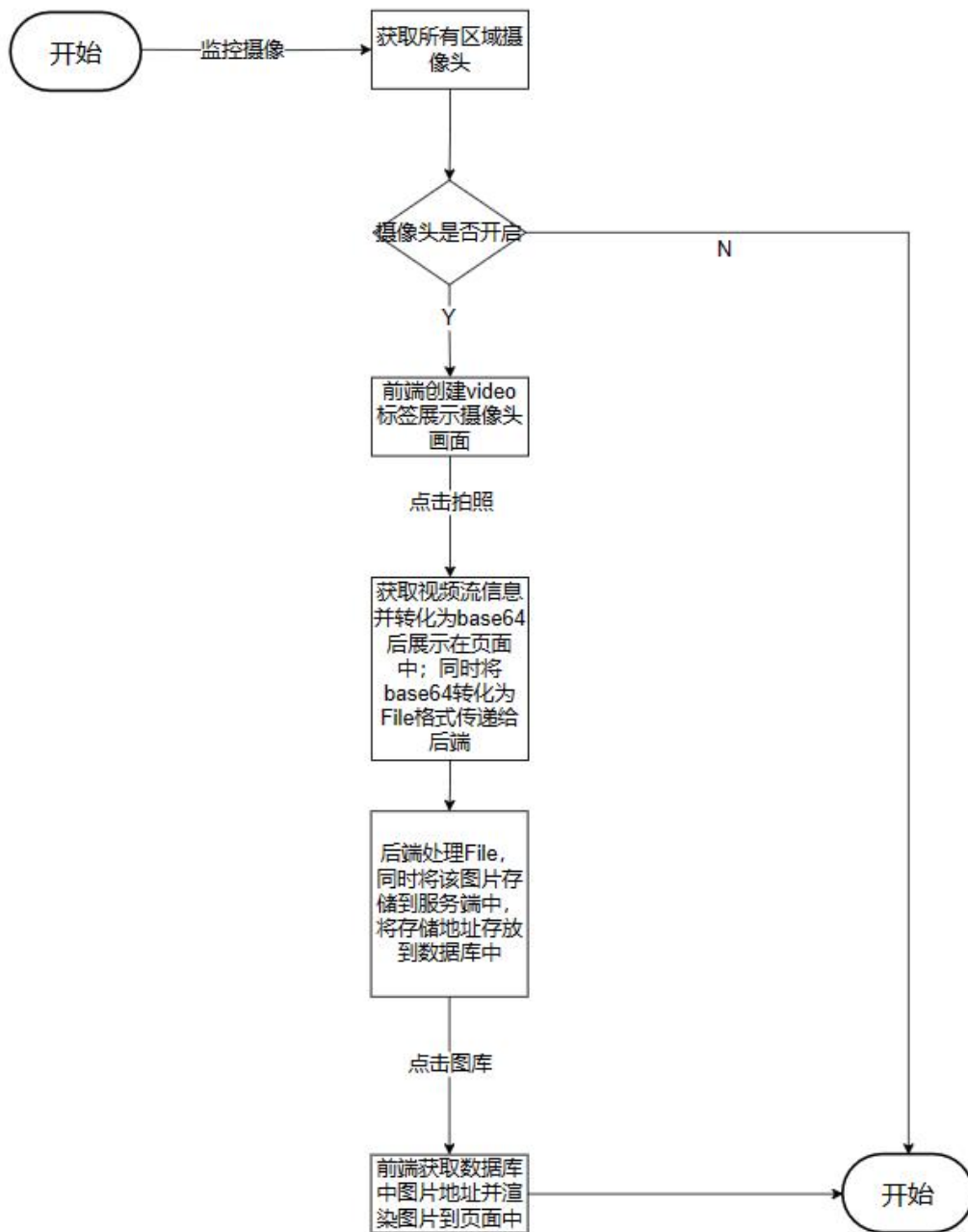


图 4.18 可视化监控摄像流程图

```
// 获取图片base64链接
var image = this.thisCancas.toDataURL("image/jpg");
// let file = new window.File([image], 'test', 'image/png')
// var file = new File([dataURLtoFile(image)], filename, {
//     type: 'application/json',
//     lastModified: Date.now()
// });
let file = this.dataURLtoFile(image, `${this.mediaMessage.name}.jpg`)

let data = new FormData();
data.append('file', file); //如果还有其他参数复制就行了
this.$axios.post('http://localhost:3000/uploadMedia', data)
    .then(res => {
        console.log(res);
        this.messageBox('保存截图成功', 'success')
    })
// this.$axios.post('/upload', file).then(res => {
//     console.log(res);
// })
```

图 4.19 可视化监控摄像前端核心代码图

```
//上传监控图片
router.post('/uploadMedia', multer({
    //设置文件存储路径
    dest: 'public/imgs'
}).array('file', 1), function (req, res, next) {
    let files = req.files;
    let file = files[0];
    let fileInfo = {};
    let path = 'public/imgs/' + Date.now().toString() + '_' + file.originalname;
    fs.renameSync('./public/imgs/' + file.filename, path);
    //获取文件基本信息
    fileInfo.type = file.mimetype;
    fileInfo.name = file.originalname;
    fileInfo.size = file.size;
    fileInfo.path = path;

    //把图片在静态文件夹下的路径记下来 放入数据库
    let sql = 'insert into 监控图片 set?'
    console.log(path, file.originalname);
    let data = {
        url: path,
        name: file.originalname.split('.')[0]
    }
}
```

图 4.20 可视化监控摄像后端核心代码图

## 4.9 个人资料界面设计及实现

管理员界面以及车主个人界面进行身份信息的修改，例如头像、名称以及其他信息；后端通过将前端上传的文件转化为 stream 流来将图片保存到服务端内，将最终的图片地址进行落库保存；代码核心图如图 4.21 所示。

```
<div class="basemesshow">
  <div class="mymesshow">
    <span class="basemes">姓名: </span> <span class="privateshow">{{myprivate.nickname}}</span>
  </div>
  <div class="mymesshow">
    <span class="basemes">性别: </span> <span class="privateshow">{{myprivate.sex}}</span>
  </div>
  <div class="mymesshow">
    <span class="basemes">住址: </span><span class="privateshow">{{myprivate.address}}</span>
  </div>
  <div class="mymesshow">
    <span class="basemes">注册时间: </span> <span class="privateshow">{{myprivate.jointime}}</span>
  </div>
  <div class="mymesshow">
    <span class="basemes">个性签名: </span> <span class="privateshow">{{myprivate.privatemes}}</span>
  </div>
</div>
```

图 4.21 个人资料界面核心代码图



## 5 系统测试

### 5.1 测试目标及任务

软件测试是为了确保软件系统能够按照需求和规格说明书的要求完成所有功能，保证系统的稳定性、安全性与易用性。测试任务主要包括需求分析、功能测试、性能测试、兼容性测试、安全性测试、车主体验测试和缺陷跟踪。其目的是发现并解决软件系统的缺陷与问题，提高软件系统的质量和稳定性，从而提高客户满意度。在测试过程中，测试人员需要编写测试计划和测试用例进行测试，并对测试过程中发现的问题进行记录和跟踪，及时修复问题以保证软件系统能够完整无缺地上线使用。通过测试，可以有效提高软件的市场竞争力，降低系统运维成本<sup>[11]</sup>。

测试是确保软件系统质量的重要手段，有着不可替代的必要性。经过测试，可以发现软件系统中的缺陷和问题，保证软件系统能够按照要求完成所有功能，并带来好的车主体验。测试可以提高软件的稳定性和安全性，降低维护成本和风险，提高软件的市场竞争力，获得车主信任。测试不仅对开发人员和软件质量保障人员具有帮助，还与企业、车主、产品和服务等方面息息相关。测试可以为企业带来更多的商业机会，在车主方面，可以为车主提供更好的车主体验，更好的服务。综上所述，测试是确保软件系统稳定运行的必要手段。

### 5.2 系统功能测试

以下对系统中各模块的主要功能进行了用例测试<sup>[12]</sup>。针对本平台的主要功能和模块进行了测试用例的填写以及最终页面效果的展示。

### 5.2.1 注册页面功能测试

表 5-1 注册页面测试用例

功能名称		注册功能		
测试目的		能够正常注册		
预置条件		无		
用例编号	操作步骤	输入数据	期望结果	执行结果
1.1	两次密码不一致	无	提示密码不一致	与期望结果一致
1.2	邀请码为'1234'	无	提示注册成功且注册身份为管理员	与期望结果一致

停车场管理系统登录功能效果图如图 5.1 所示。



图 5.1 车主注册图

## 5.2.2 登录页面功能测试

表 5-2 登录页面测试用例

功能名称		登录功能		
测试目的		能够正常登录		
预置条件		无		
用例编号	操作步骤	输入数据	期望结果	执行结果
1.1	输入未注册的账号	无	提示登录失败	与期望结果一致
1.2	管理员账号登录	无	提示注册成功 且注册身份为 管理员	与期望结果一致
1.3	车主账号登录	无	跳转到车主界面	与期望结果一致

停车场管理系统登录功能效果图如图 5.2、图 5.3、图 5.4 所示。

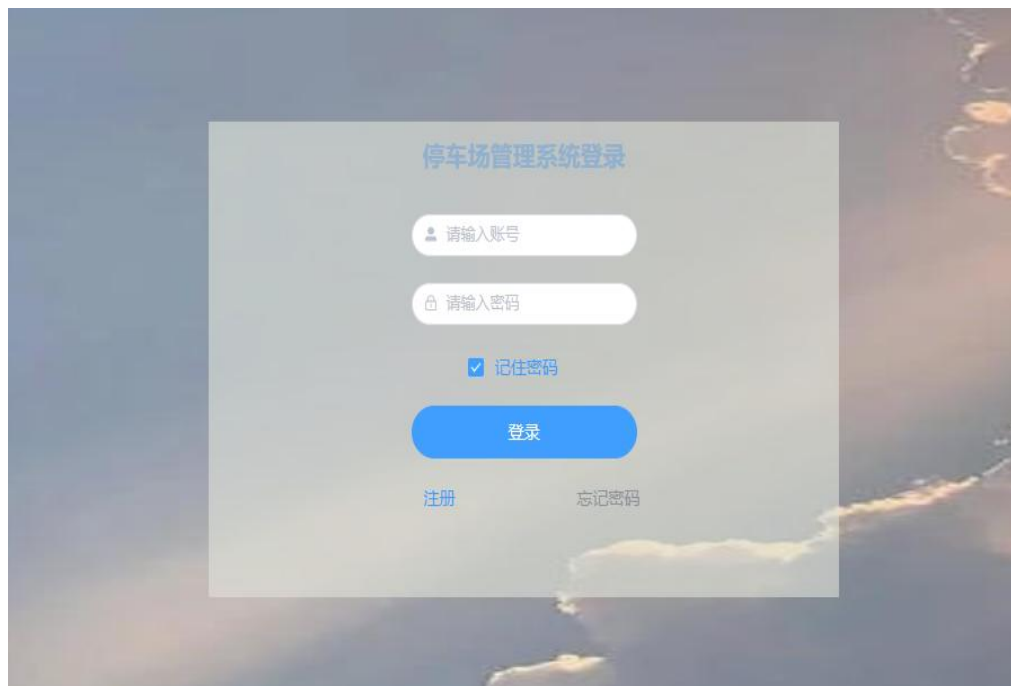


图 5.2 用户登录图



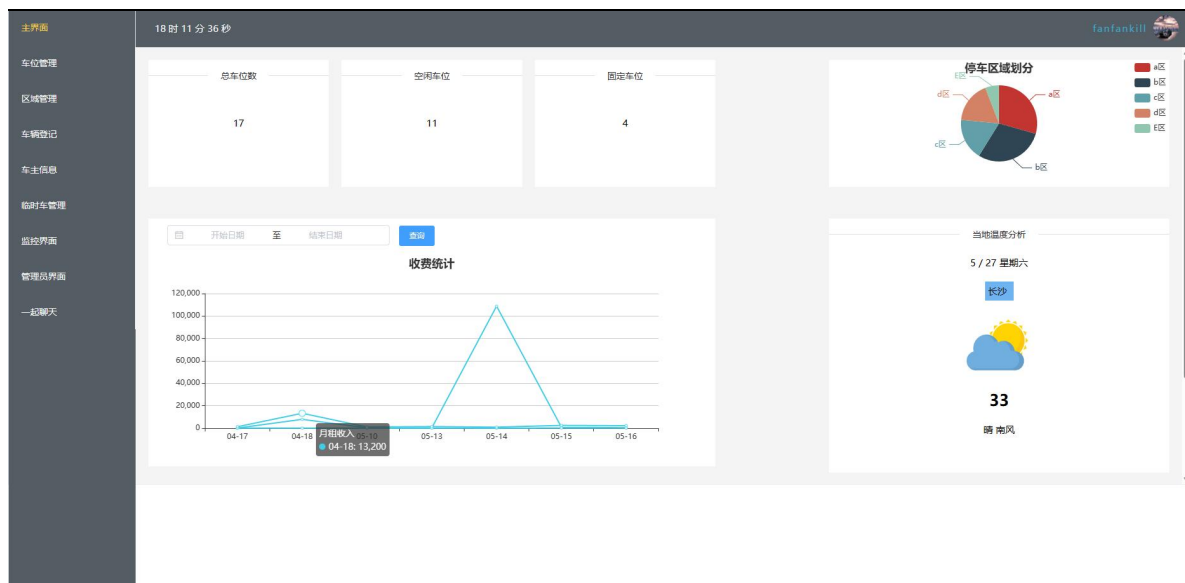


图 5.3 管理员工作视区图



图 5.4 车主主页面图

### 5.2.3 车位管理模块测试

表 5-3 车位管理模块测试用例

功能名称		车位管理功能		
测试目的		能够正常进行车位管理		
预置条件		无		
用例编号	操作步骤	输入数据	期望结果	执行结果
1.1	管理员点击该页面可正常获取数据	无	展示车位信息列表	与期望结果一致
1.2	点击“新增车位”可以正常弹窗且填写完信息后正常添加并且重新获取数据	无	提示添加成功且获取了最新数据	与期望结果一致
1.3	点击“编辑”修改车位的空闲状态后点击确认	无	提示修改成功且获取了最新数据	与期望结果一致
1.4	点击“删除”按钮后确认操作	无	提示删除成功且获取了最新数据	与期望结果一致
1.5	在搜索框中输入车位编号进行搜索	无	获取到对应编号的车位信息	与期望结果一致
1.6	点击分页按钮来切换数据	无	切换到其他页的数据	与期望结果一致

当切换到车位管理的 tab 下的时候会展示车位管理的内容，可以对车位进行增删改查操作，如图 5.5、图 5.6 所示。

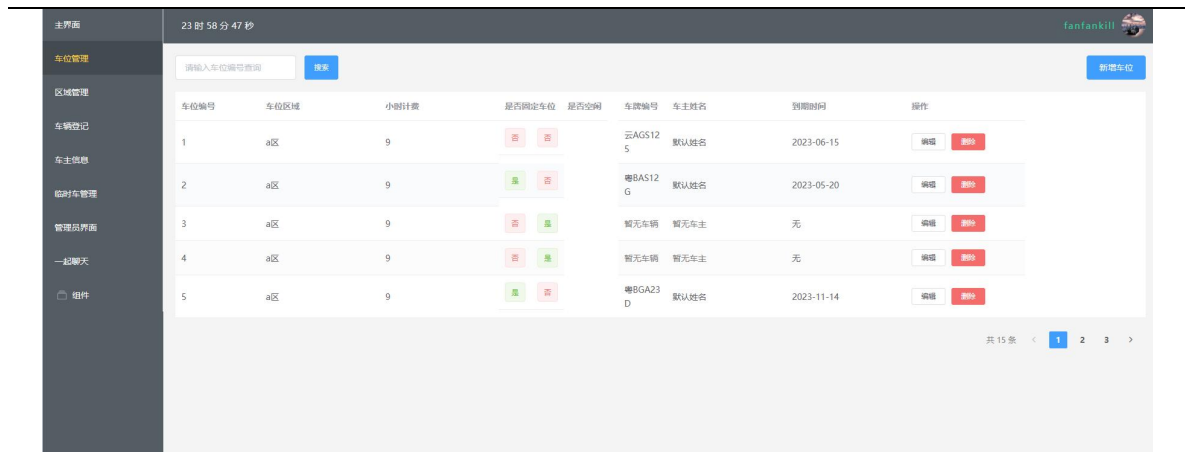


图 5.5 车位管理模块整体演示图

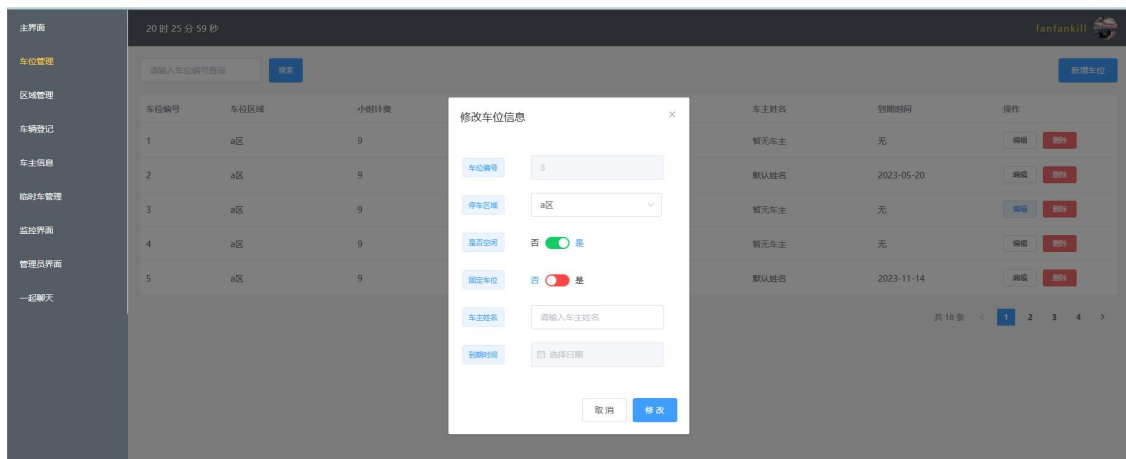


图 5.6 修改车位信息演示图

## 5.2.4 区域模块测试

表 5-4 区域模块测试用例

功能名称		区域管理功能		
测试目的		能够管理新增和修改区域信息		
预置条件		无		
用例编号	操作步骤	输入数据	期望结果	执行结果
1.1	点击“新增区域”后填写区域信息	无	提示新增成功且重新获取的列表含有新增区域	与期望结果一致
1.2	2. 点击区域编辑按钮修改信息	无	提示修改成功且该区域小时价格改为最新价格	与期望结果一致

区域模块展示如下图 5.7 图 5.8 所示。

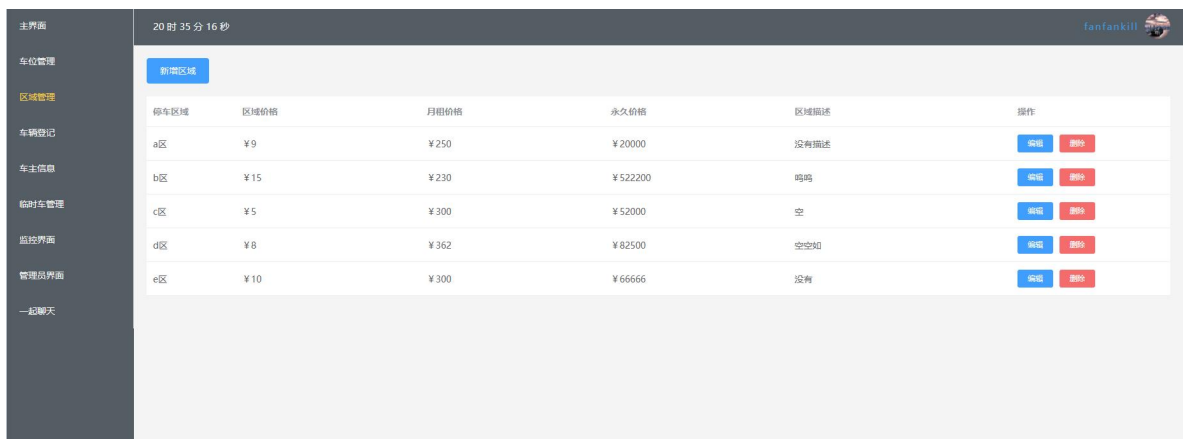


图 5.7 区域模块演示图

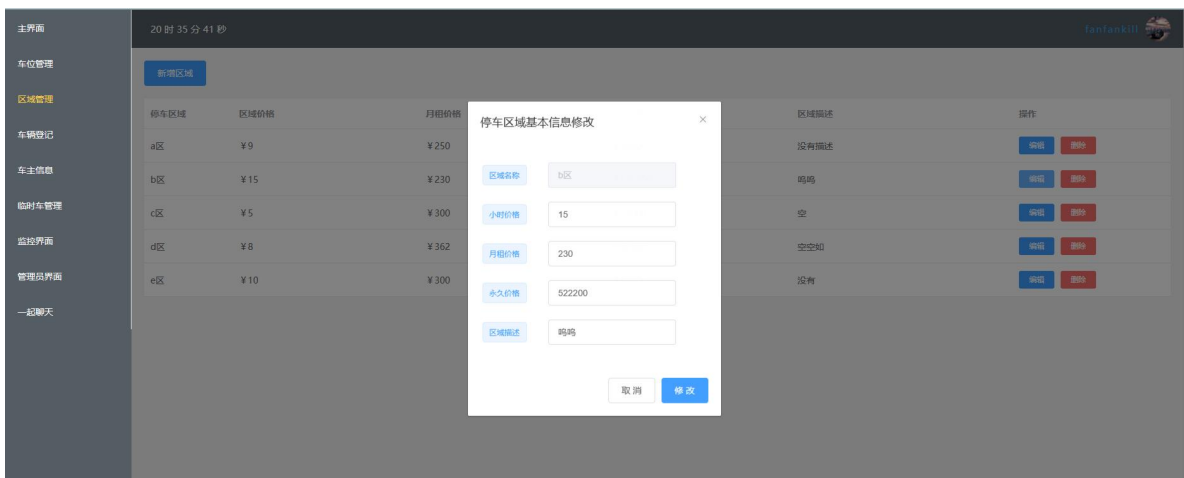


图 5.8 区域修改功能演示图

## 5.2.5 临时车登记模块测试

表 5-5 临时车登记模块测试用例

功能名称		临时车登记模块		
测试目的		保证能够正常录入临时车数据		
预置条件		无		
用例编号	操作步骤	输入数据	期望结果	执行结果
1.1	填写车辆车牌号码后登记，选择对应的停车区域，再筛选该区域下空闲的车位	无	该车进入临时车管理且选择的区域车位也显示为占用状态	与期望结果一致

临时车登记模块测试展示图如图 5.9 图 5.10 所示。

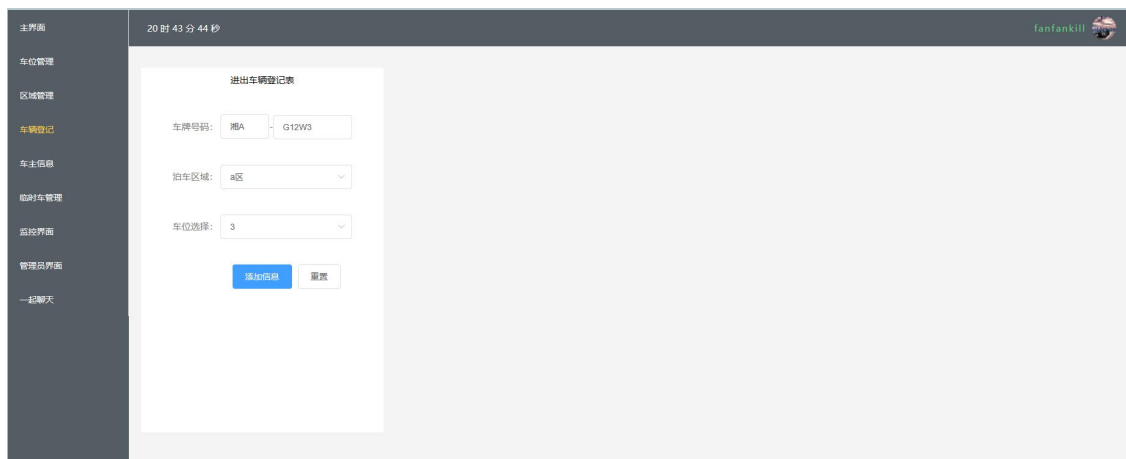


图 5.9 临时车登记模块演示图

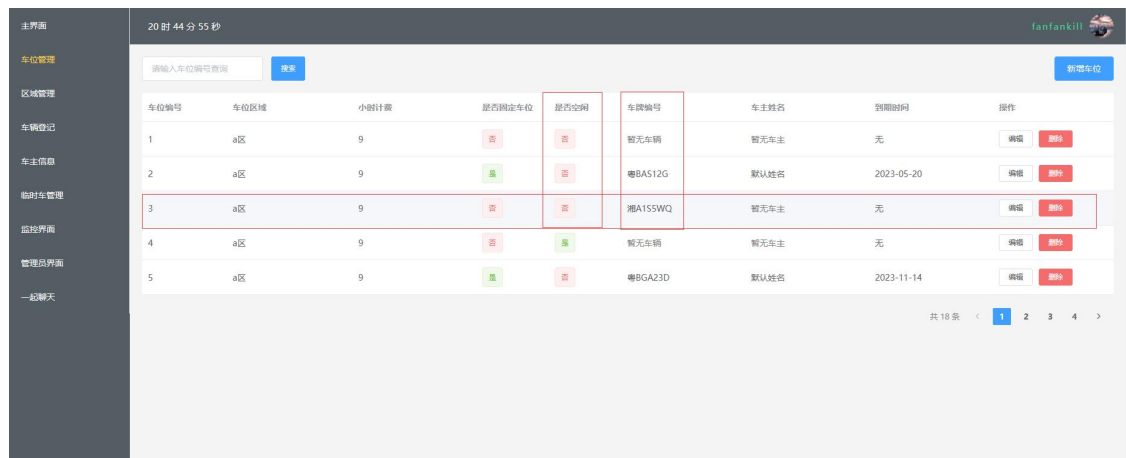


图 5.10 车辆入库占位演示图

### 5.2.6 车主信息模块测试

表 5-6 车主信息模块测试用例

功能名称		车主信息模块		
测试目的		测试车主信息是否能够同步过来且能够进行操作		
预置条件		存在车主信息		
用例编号	操作步骤	输入数据	期望结果	执行结果
1.1	管理员新增车主信息	无	车主信息会显示该新增车主	与期望结果一致
1.2	车主进行车位选择	无	相关信息会展示该车主的车位信息以及到期时间	与期望结果一致
1.3	管理员进行续费操作	无	能够正常的增加续费时长以及能够统计该笔金额	与期望结果一致

车主信息模块测试效果图如图 5.11 图 5.12 所示。

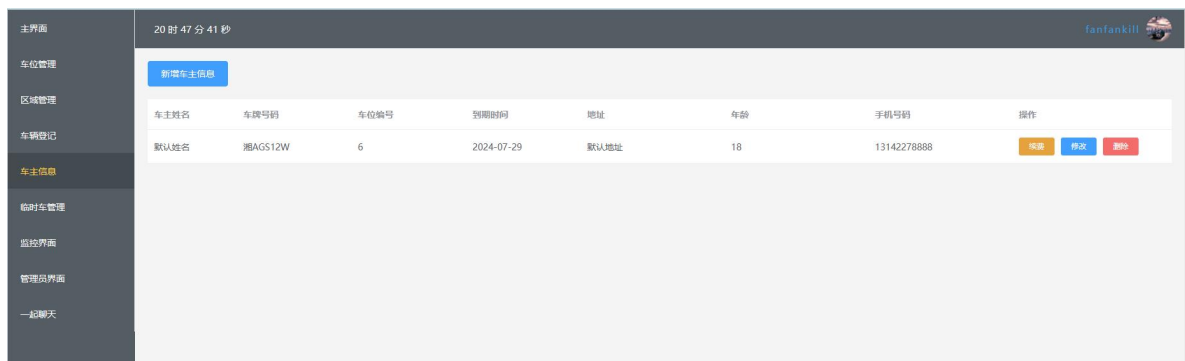


图 5.11 车主信息模块演示图

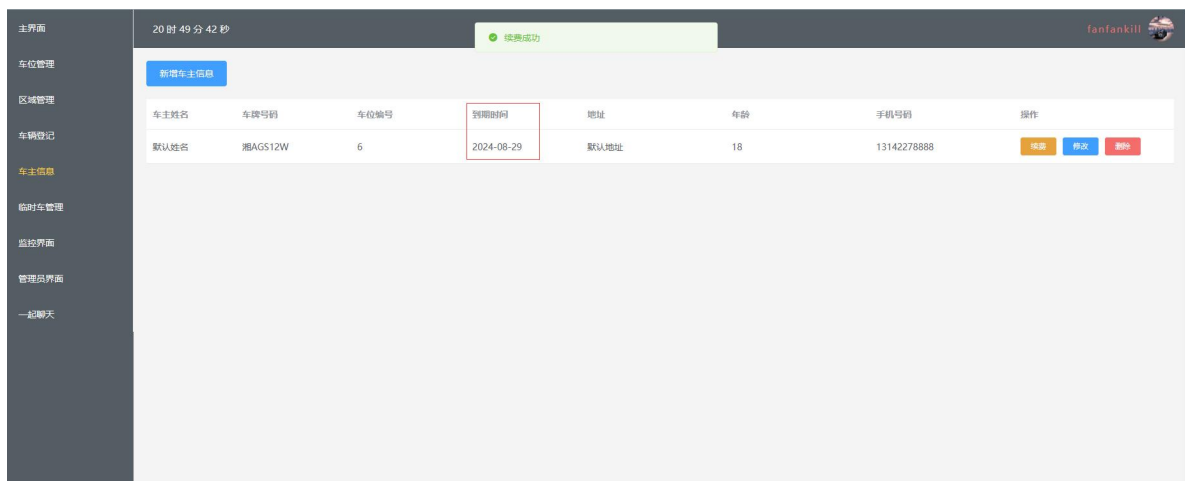


图 5.12 车主续费演示图

## 5.2.7 临时车管理模块测试

表 5-7 临时车管理模块测试用例

功能名称		临时车管理模块		
测试目的		能够正常获取临时车数据以及出库临时车		
预置条件		无		
用例编号	操作步骤	输入数据	期望结果	执行结果
1.1	点击临时车 tab	无	获取到所有临时车的车辆信息	与期望结果一致
1.2	点击出库按钮后将临时车出库	无	该车辆变为已驶出状态，占有车位也变为空闲状态，缴费金额计入图表	与期望结果一致

临时车测试效果图如图 5.13、图 5.14、图 5.15 所示。

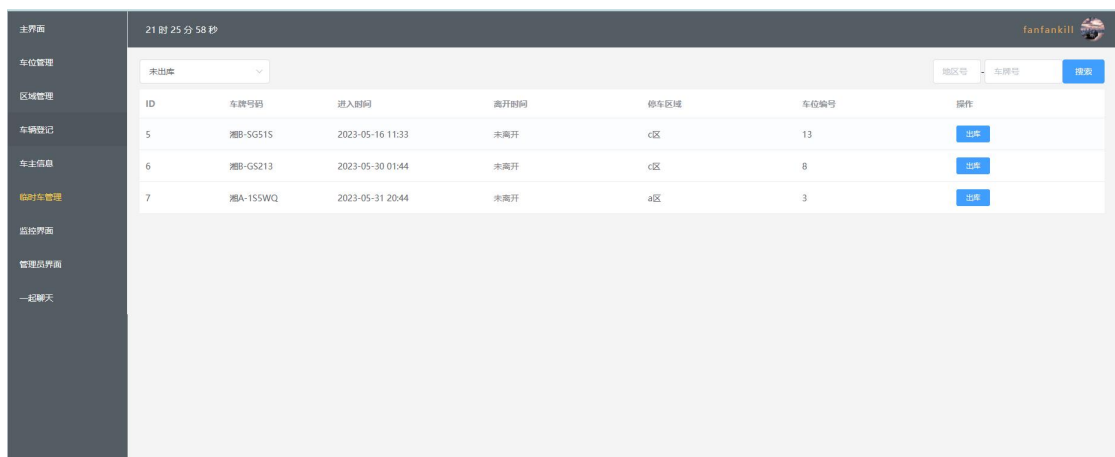


图 5.13 临时车管理模块演示图

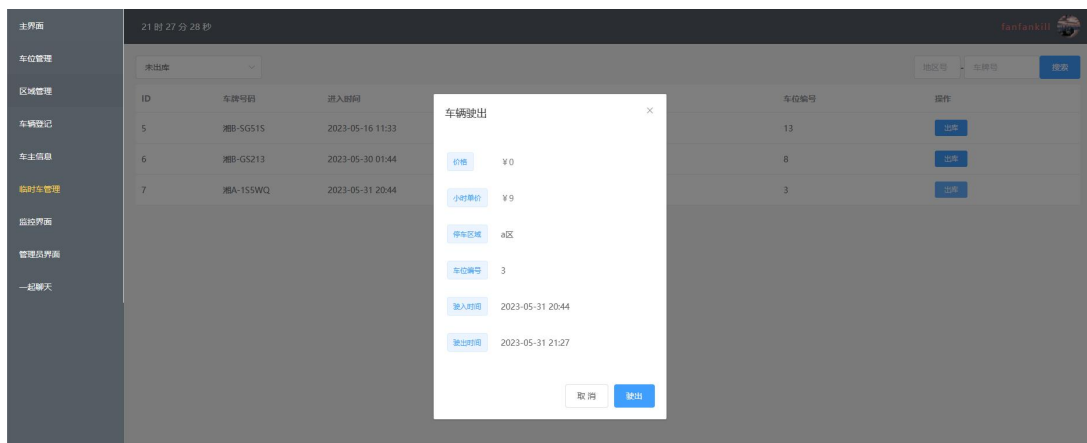


图 5.14 临时车出库演示图

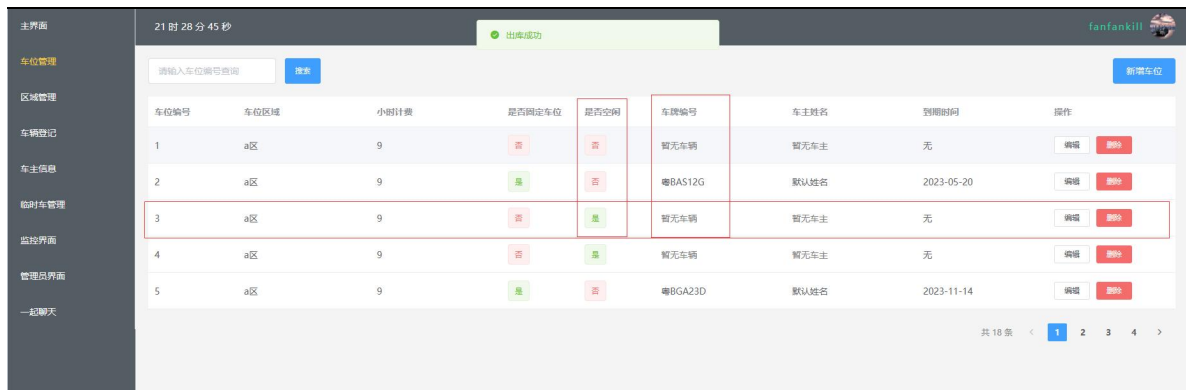


图 5.15 车位管理功能图

## 5.2.8 可视化监控模块测试

表 5-8 可视化监控模块测试用例

功能名称		可视化监控模块		
测试目的		能够看到所有区域摄像头信息以及可以截图保存		
预置条件		无		
用例编号	操作步骤	输入数据	期望结果	执行结果
1.1	切换到监控页面	无	获取到所有监控摄像头信息	与期望结果一致
1.2	增监控摄像头后页面出现相关摄像头信息	无	新增后获取最新的监控摄像头列表	与期望结果一致
1.3	点击拍照后生成截图信息并且保存到服务端中，可以在图库中查看	无	获取到所有监控截图	与期望结果一致

管理员切换到监控摄像头区域后可以看到所有区域的摄像头信息也可以连接不同区域的摄像头，对本地摄像头进行拍照截图展示（没有其他摄像头资源，以本地唯一摄像头作为演示效果）如图 5.16、图 5.17、图 5.18 所示。



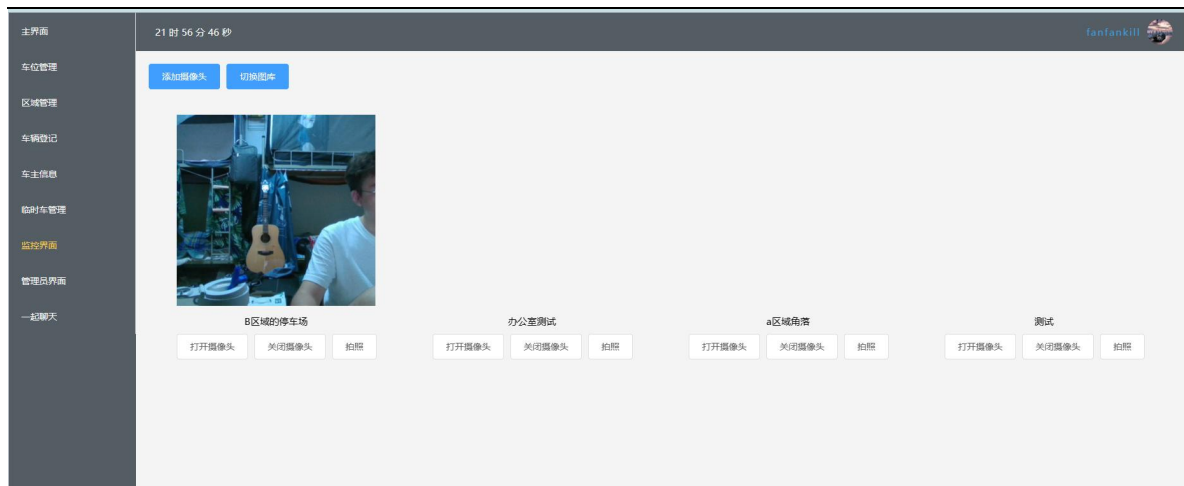


图 5.16 监控摄像头演示图

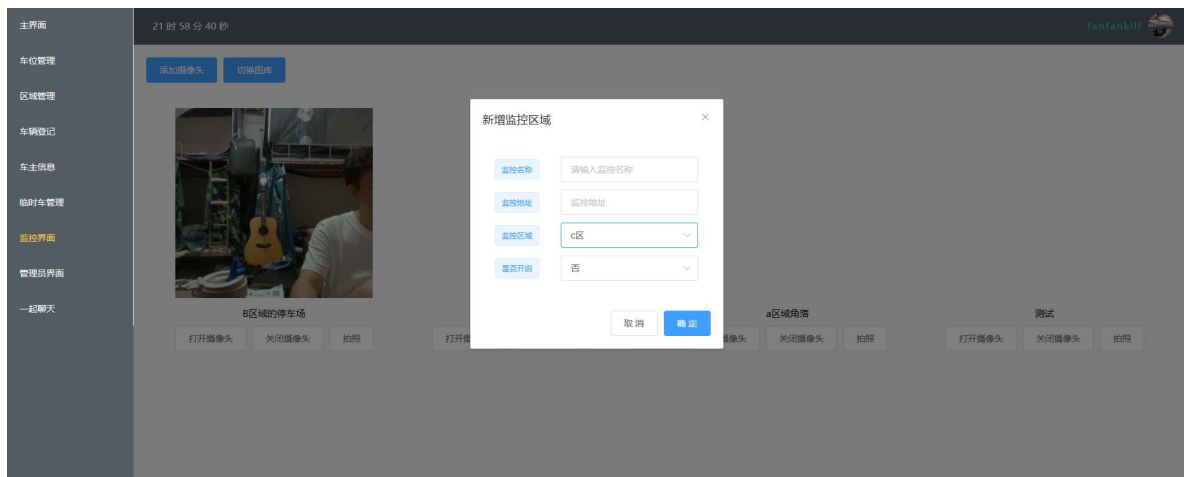


图 5.17 新增监控演示图

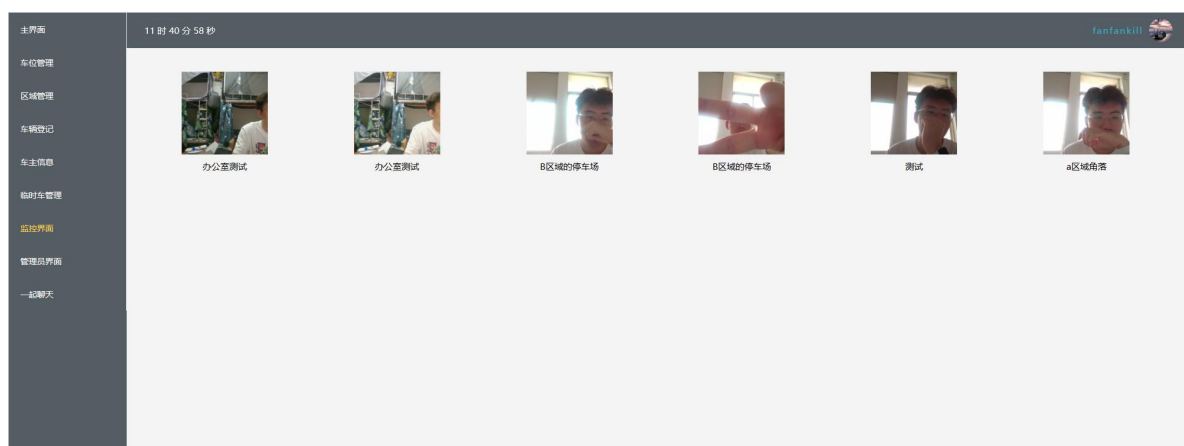


图 5.18 监控截图演示图

## 5.2.9 个人资料模块测试

表 5-9 个人资料模块测试用例

功能名称		个人资料模块		
测试目的		能够个人资料以及可以修改个人资料		
预置条件		无		
用例编号	操作步骤	输入数据	期望结果	执行结果
1.1	点击管理员界面	无	正常展示该管理员个人资料	与期望结果一致
1.2	点击车主个人资料	无	正常展示该车主个人资料	与期望结果一致
1.3	修改管理员头像	无	显示新头像	与期望结果一致

个人资料测试结果如图 5.19、图 5.20 所示。

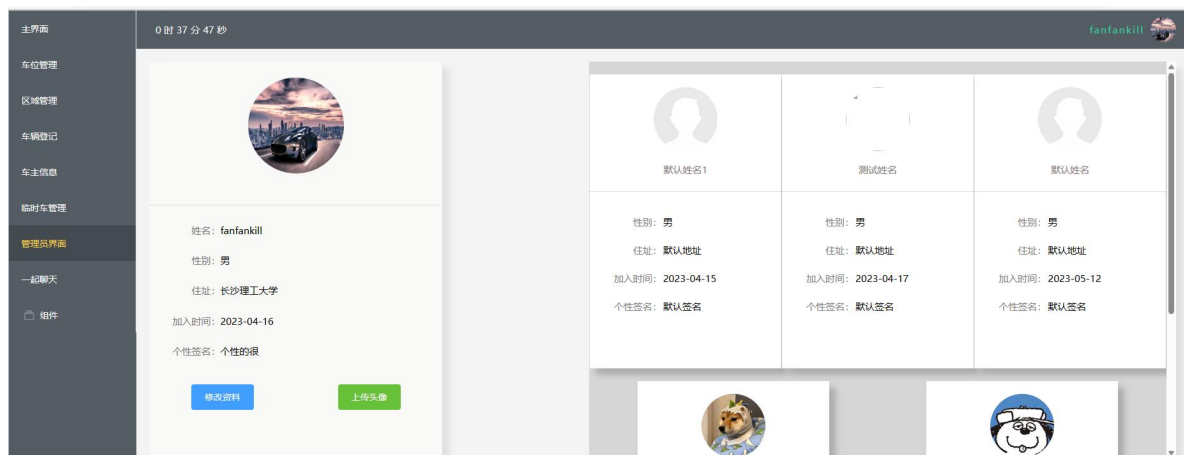


图 5.19 管理员个人资料演示图



姓名: 默认姓名

性别: 男

住址: 默认地址

注册时间: 2023-05-13

个性签名: 默认签名

修改资料

上传头像

图 5.20 车主个人界面演示图

## 6 总结及展望

该系统可以提高停车场的管理效率、优化车辆管理，并改善停车场的安全性，以提高城市交通治理水平。虽然该系统对传感器、摄像头等智能设备的依赖较大，可能在设备损坏或停电等情况下运行受到影响，但系统优点在于提高了停车场的管理效率，优化了车辆管理。需要解决一些在用户体验和系统架构设计上的问题，希望日后能够进行改进和提升。总体来说，这个可视化停车场管理系统有着重要的应用和发展价值。

随着 Web 技术的不断发展，数据分析技术的广泛应用，以及应用智能化、绿色化、集成化等技术的推广，这个系统将会得到进一步的完善和推广。例如，可以通过采集更多的停车场数据并进行分析，进一步提高停车场的管理水平和效率。同时，也可以将该系统与交通控制和城市规划等领域相结合，实现更加全面、高效的交通管理和城市智能化。此外，还可以通过开发智能车牌识别系统，进一步提高管理效率和安全性。总之，这个可视化停车场管理系统有着广泛的应用前景和巨大的发展空间，可以作为未来城市智慧交通管理的重要组成部分。

## 参考文献

- [1] 陈根. 互联网+智能家居:传统家居颠覆与重构[J]. 软件导刊, 2020, 10(02):212-213.
- [2] 海涛, 刘振语, 黄新迪, 陈永鉴, 李俊杰. 窄带宽物联网云平台技术在孵化系统中的应用[J]. 河南科技大学学报(自然科学版), 2020, 41(04):46-51+60+7.
- [3] 张水潮, 蔡逸飞, 黄锐等. 基于预约需求的共享停车平台泊位分配方法[J]. 交通运输系统工程与信息, 2020, 20(03):137-143+162.
- [4] 王亚茹. 车位智能管理系统设计与实现[J]. 无线互联科技, 2019, (24):60.
- [5] 杨勇, 董道文, 一种智能停车场车位预约管理平台[J]. 汽车实用技术, 2018, 43(18):49-50.
- [6] 陈礼辉, 吴振东, 尚诗嫣等. 基于 Lab VIEW 的停车场管理系统[J]. 山东工业技术, 2018(14):160.
- [7] 张秋艳, 刘艺. 基于 RFID 的智能停车场管理系统设计[J]. 榆林学院学报, 2022, 32(02):10-13.
- [8] 许卫洪. 基于 LabVIEW 和 RFID 技术的智能停车场管理系统[J]. 信息记录材料, 2019, 20(11):192-193.
- [9] 赵桂香. 混合停车场共享预约系统设计[J]. 交通企业管理, 2022, 37(02):72-75.
- [10] 薛文凯, 李梦洁, 共享经济下的停车位系统设计研究[J]. 时代汽车, 2021, 18(03):114-115.
- [11] Chen Z, Spana S, Yin Y et al. An advanced parking navigation system for downtown parking[J]. Networks and Spatial Economics, 2019, 19(3):953-968.
- [12] Kim M, Kim Y. Multi-blockchain structure for a crowdsensing-based smart parking system[J]. Future Internet, 2020, 12(5):90.
- [13] Said A M, Kamal A E, Afifi H. An intelligent parking sharing system for green and smart cities based Io T[J]. Computer Communications, 2021, 172(30):22-26.

## 致谢

我要感谢很多人对我论文的支持和帮助。

首先，我要感谢我的导师，在我完成这篇论文的过程中，对我的耐心指导和教导让我受益匪浅。导师您不仅是我的学者引路人，更是我的精神导师，您悉心的指导、严谨的治学态度、严格的工作要求，让我非常受益。在您的帮助下，我不仅对本篇论文的情况有了更深入的了解，在学术上也受到了很大的激励和启示。

其次，我要感谢指导委员会的其他成员，感谢他们的评议和评价，让我能够更好地理解自己的思路和写作方法。感谢委员们对我的支持和关心，让我对自己更有信心和勇气。

另外，我也要感谢实验室的老师 and 同学，感谢他们在学术上的指导和交流，让我不断深入研究问题。他们的帮助在本篇论文中起到了非常重要的作用，为我解决了不少问题。特别要感谢实验室的同学们，您们一直以来的友爱和鼓励，让我在学术上有了更多的见识和体会。

还要感谢我的家人和朋友，因为他们的支持和鼓励，我才有了完成这篇论文的勇气和决心。在我遇到困难和疑惑时，他们总是在我身边支持和鼓励着我，让我时刻感到被爱和温暖。尤其感谢我的父母，您们一直以来的无私支持和关爱，让我无论在何时何地都倍感温暖和幸福。

最后，我要感谢所有没有被提及的人对我的支持和帮助，你们的鼓励和耐心指导让我走向了更深入的学术道路。我将会铭记你们的友情和支持，在今后的学习和工作中，继续发挥自己的专业能力和素养，为建设和谐社会、服务人类的美好事业贡献力量。再次表达我最真挚的感谢和祝福！

## 附录 部分程序源代码

```
//返回首页的总车位数目 空闲车位数 固定车位数
exports.gettotalcar=(req,res)=>{
  let sql='select count(*) as totalcarport from 车位信息'
  let sql2='select count(*) as totalfreecar from 车位信息 where HaveCar=1'
  let sql3='select count(*) as totalfixcar from 车位信息 where FixCar=0'

  db.base(sql,null,(result)=>{
    let res1=result[0].totalcarport
    db.base(sql2,null,(result2)=>{
      let res2=result2[0].totalfreecar

      db.base(sql3,null,(result3)=>{
        let res3=result3[0].totalfixcar

        res.json({
          'message':'获取车位数据成功',
          'flag':'1',
          'totalcar':res1,
          'totalfreecar':res2,
          'totalfixcar':res3
        })
      })
    })
  })
}

//新增车主信息
exports.addpersonmes=(req,res)=>{
  let info=req.body
  console.log(info);
  let data={
    CarNumber:info.CarNumber,
    CarportNumber:info.CarportNumber,
    PersonName:info.PersonName,
    Address:info.Address,
    Age:info.Age,
    Phone:info.Phone,
    isUpdate: info.isUpdate ? true : false
  }
  let sql
  let dataArr
  if(data.isUpdate) {
    sql = `update 车主信息 set CarNumber=?, CarportNumber=?,PersonName=?, Age=?, Address=? where Phone =?`
    dataArr = [data.CarNumber, data.CarportNumber, data.PersonName, data.Age, data.Address, data.Phone]
  } else {
    sql='insert into 车主信息 set?'
  }

  console.log(info.duetime);
  //计算到期时间
  let nowtime=new Date()
  let duetime=info.duetime
  let lasttime= moment(nowtime).add(duetime, 'months')
  //格式化时间
  lasttime=moment(lasttime).format()
  console.log(lasttime);
  let sql2='update 车位信息 set FixCar =1 , HaveCar=0 ,CarpersonName=?,CarNumber=?,DueDate=? where CarportNumber=?'
  let data2=[info.PersonName,info.CarNumber,lasttime,info.CarportNumber]
  //先修改这个车位的固定状态
```



```
//驶出车辆计费登记
exports.removecarjoin=(req,res)=>{
  let info=req.body
  //进出记录表收费
  let sql='update 进出记录表 set LeaveTime=?,totalfare=?,IsDone=? ,day=?where Id=?'
  //当前日期
  let day=moment(Date.now()).format('YYYY-MM-DD')
  console.log(day);
  let nowtime=moment(Date.now()).format('YYYY-MM-DD HH:mm:ss')
  console.log("Id是"+info.CarPortNumber);
  console.log("时间"+nowtime);
  console.log("总费用"+info.totalfare);
  let data=[nowtime,info.totalfare,1,day,info.Id]
  console.log(info);
  db.base(sql,data,(result=>{
    //总收费表更新记录
    let sql2='insert into 收费详细 set ?'
    let data2={
      type:0,
      fare:info.totalfare,
      date:day
    }
    db.base(sql2,data2,result2=>{
      //关联表更新
      let sql3='update 车位信息 set HaveCar=?,CarNumber=? where CarPortNumber=?'
      let data3=[1,null,info.CarPortNumber]
      db.base(sql3,data3,(result3=>{
        res.json({
          'code':1,
          'message':'出库成功',
          result,
          result3
        })
      })
    })
  })
}
```

```
<template>
  <div class="camera_outen">
    <div>
      <el-button type="primary" @click="toAddMedia">添加摄像头</el-button>
    </div>
    <div class="showMediaContent" >
      <media v-for="item in medialist" :key="item.id" :mediaMessage="item"></media>
    </div>
    <!-- 新增监控弹窗 -->
    <fan-dialog title="新增监控区域" :visible="showAddMedia" width="80%">...
  </fan-dialog>
  </div>
</template>
<script>
import media from './media.vue'
export default {
  components: {
    media
  },
  mounted() {
    this.getallpositons()
    this.getAllmedia()
  },
  data() {
    return {
      showAddMedia: false,
      formData: {
        isActive: '0',
      },
      positions: [],
      medialist: [],
    }
  },
  methods: {
    //获取停车区域信息
```

```
<template>
  <div>
    <!-- 用于未加载出来数据时候的骨架模板 -->

    <el-skeleton :loading="loading" animated>
      <template slot="template">
        <div class="topdiv">
          <div class="leftdiv">
            <el-skeleton-item variant="div" class="topshowdiv" />
            <el-skeleton-item variant="div" class="topshowdiv" />
            <el-skeleton-item variant="div" class="topshowdiv" style="width:420px" />
          </div>
          <el-skeleton-item variant="div" class="rightdiv" style="height:230px;" />
        </div>
        <div class="bottomdiv">
          <el-skeleton-item variant="div" class="btn_leftdiv" style="height:400px" />
          <el-skeleton-item variant="div" class="btn_rightdiv" style="height:410px" />
        </div>
      </template>
    </el-skeleton>
  </div>
</template>

<!-- 数据部分 -->
<template> ...
</template>
</el-skeleton>
</div>
</template>
```