****

实习结课论文



|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于Python和Amap的智能车位管理系统 | | |
| 学 院 | 大数据与智能工程学院 | 专 业 | 计算机科学与技术 |
| 学生姓名 | 闫建槿 | 学 号 | 20221152022 |
| 学生姓名 | 郭思宇 | 学 号 | 20222601017 |
| 学生姓名 | 廖祥辉 | 学 号 | 20221152014 |
| 指导教师 | 鲁 宁 | | |

2024 年 12 月 20 日

基于Python和Amap的智能车位管理系统

闫建槿、郭思宇、廖祥辉

（西南林业大学 大数据与智能工程学院 昆明 650224）

摘 要：针对当前市面上已有的停车管理系统工作存在功能单一、车主找车位不易和停车场管理花费大量的时间和成本等问题，以Python语言为主要编程语言、MySQL8为核心数据库系统、结合Amap 地图API技术，设计并开发了一套基于B/S架构智能车位信息系统。该系统主要由车主用户模块，车位导引模块，管理员用户后台管理模块构成，实现了车位管理的数字化。实验证明，该系统工作良好，有效地解决了车主停车难、停车场管理烦等问题。

关键词：停车场管理；停车导引系统；数字化管理。

**Smart parking space management system based on Python and Amap**

Jianjin Yan; Siyu Guo; Xianghui Liao

(College of Big Data and Intelligent Engineering, Southwest Forestry University, Kunming 650224)

**Abstract:**In view of the problems of the existing parking management systems on the market, such as single functions, difficulty for car owners to find parking spaces, and a lot of time and cost for parking lot management, we designed and developed a smart parking space information system based on B/S architecture, using Python as the main coding language, MySQL v8 as the core of the database system and use Amap API technology. The system is mainly composed of the car owner user module, parking space guidance module, and user of administrator background management module, which realizes the digitalization of parking space management. Experiments have proved that the system works well and effectively solves the problems of car owners' difficulty in parking and parking lot management.

**Key Words:** parking lot management; parking guidance system; digital management.

目 录

[第1章 前言 1](#_Toc185608486)

[1.1 设计背景及意义 1](#_Toc185608487)

[1.2 成员分工 1](#_Toc185608488)

[第2章 系统分析与设计 2](#_Toc185608489)

[2.1 需求分析 2](#_Toc185608490)

[2.1.1 角色和功能需求 2](#_Toc185608491)

[2.1.2 非功能需求 2](#_Toc185608492)

[2.2 可行性分析 3](#_Toc185608493)

[2.2.1 经济可行性 3](#_Toc185608494)

[2.2.2 技术可行性 4](#_Toc185608495)

[2.2.3 时间可行性 4](#_Toc185608496)

[2.3 技术选型 5](#_Toc185608497)

[2.3.1 开发环境 5](#_Toc185608498)

[2.3.2 服务器选择 5](#_Toc185608499)

[2.3.3 Python语言 6](#_Toc185608500)

[2.3.4 Flask后端框架 6](#_Toc185608501)

[2.3.5 MySQL数据库 6](#_Toc185608502)

[2.3.6 Amap地图API 6](#_Toc185608503)

[2.4 体系架构设计 7](#_Toc185608504)

[2.4.1 总体架构 7](#_Toc185608505)

[2.4.2 前端部分 7](#_Toc185608506)

[2.4.3 后端部分 8](#_Toc185608507)

[2.5 数据库设计 8](#_Toc185608508)

[2.5.1 数据库实体关系设计 8](#_Toc185608509)

[2.5.2 数据库表 10](#_Toc185608510)

[第3章 车主端设计和实现 13](#_Toc185608511)

[3.1 车主端设计 13](#_Toc185608512)

[3.1.1 车主端总体设计 13](#_Toc185608513)

[3.2 界面功能和实现 15](#_Toc185608514)

[3.2.1 注册与登录 15](#_Toc185608515)

[3.2.2 预约功能 16](#_Toc185608516)

[3.2.3 车辆管理 18](#_Toc185608517)

[3.2.4 车位信息 21](#_Toc185608518)

[3.3 车位寻路与导航功能设计及实现 21](#_Toc185608519)

[3.3.1 驶入车位选择 21](#_Toc185608520)

[3.3.2 Amap API调用功能 23](#_Toc185608521)

[3.3.3 导航跳转 23](#_Toc185608522)

[第4章 管理端设计和实测展示 24](#_Toc185608523)

[4.1 管理端设计 24](#_Toc185608524)

[4.1.1 管理端总体设计 24](#_Toc185608525)

[4.2 界面功能设计及实测展示 25](#_Toc185608526)

[4.2.1 管理登录 25](#_Toc185608527)

[4.2.2 车辆管理 26](#_Toc185608528)

[4.2.3 车位管理 27](#_Toc185608529)

[4.2.4 预约管理 29](#_Toc185608530)

[4.2.5 车主管理 30](#_Toc185608531)

[4.2.6 综合管理可视化看板 31](#_Toc185608532)

[第5章 系统服务部署 33](#_Toc185608533)

[5.1 部署目标 33](#_Toc185608534)

[5.2 云服务器选择与配置 33](#_Toc185608535)

[5.2.1 服务器选择和设计 33](#_Toc185608536)

[5.2.2 安全配置 33](#_Toc185608537)

[5.3 系统部署地址 34](#_Toc185608538)

[第6章 总结与展望 35](#_Toc185608539)

[参考文献和资料 36](#_Toc185608540)

[指导教师介绍 37](#_Toc185608541)

[致谢 38](#_Toc185608542)

[开发小组成员介绍 39](#_Toc185608543)

# 前言

## 设计背景及意义

随着城市化进程的加快，城市交通问题日益严峻，尤其是“停车难”成为了许多城市的突出的民生问题，车主们常常面临寻找空余车位的困扰，导致停车场内拥堵。为了提高停车场管理能力、优化车主体验，我们开发了基于Python与Amap的智能车位管理系统。

该系统创新性地将Amap API的定位与导航功能与停车场管理相结合，系统获取实时的停车位信息使车主能够通过导航直接驶入到空闲车位，避免了传统停车管理方式中的拥堵和不便，有效减少了停车场内的交通压力，优化了停车资源的分配；帮助停车场管理者实时监控车位使用情况，进行精准的车位调度与管理，提升整体停车效率，为有效解决“停车难”等交通问题提供一种可能的方案[1]。

## 成员分工

开发组由3名成员构成，组长由闫建槿（学号20221152022）担任，郭思宇（学号20222601017）、廖祥辉（学号20221152014）是开发组成员。

闫建槿负责全组项目开发工作统筹安排并负责用户端侧开发设计和系统集成；郭思宇负责车位导引功能设计，提供云服务器及项目部署支持；廖祥辉负责项目管理端侧开发设计和数据库初步设计。数据库优化、系统功能优化、实习论文、展示PPT和视频由三人讨论合作完成。

# 系统分析与设计

## 需求分析

### 角色和功能需求

经根据项目组调研，智能车位管理系统目标使用对象是城市公共停车场，预留接入社区停车场的条件；咨询相关目标用户得知，车主用户普遍希望智能车位管理系统可以预约目标位置的车位、快速指引到达车位和停放状态查看，停车场管理用户希望系统可以进行在线综合管理、实时查看车位使用情况和相关信息的维护与监控。项目组综合分析认为，系统的主要使用对象是车主和停车场管理方。因此系统设计了车主用户、管理员用户角色；车主用户角色主要有车位预约、车位查询、停车指引、名下车辆管理和历史停车记录查看的需求；管理员用户角色主要有实时监控车位使用情况、管理车位和系统信息综合维护。此外，为满足部分停车场数据可视化的需求，项目组在管理员用户角色下设计了数据可视化看板功能[2]。

### 非功能需求

经根据项目组调研和对市面上相似的系统研究，从人机交互、安全控制、使用场景等角度出发，结合已有的项目经验分析研判，认为智能车位管理系统的非功能需求主要有以下几个：

1. 性能需求：系统需要能够支持大量用户同时使用，尤其是在停车高峰期，系统响应时间应足够快，尤其是在车位查询和导航引导时，保证用户体验。
2. 设计需求：系统需要能够提供便捷、快速的功能入口，界面设计模块化、对用户友好。
3. 安全性需求：系统需要保障用户的个人信息数据的安全；位置信息、账号信息、车辆信息等是用户敏感信息，获取需要用户同意。
4. 可用性需求：系统应保证高可用性，避免由于服务器中断或API服务中断导致系统不可用。提供完善的错误提示和恢复机制，确保用户能够正常操作。
5. 兼容性需求：系统应支持移动端设备和桌面端设备使用主流浏览器（如Chrome、Firefox等）访问，预留智能车机设备访问兼容性设计。
6. 扩展性需求：系统虽然以公共停车场为开发基础点，但是在实际工程中，社区等场景的停车场主要需求与公共停车场类似，系统应预留其他接口，满足后续针对特定场景的使用需求。

## 可行性分析

### 经济可行性

项目组结合自身资源、经济情况和项目条件，以“节省开发成本、最小化使用成本、降低综合运维成本”为出发点进行考虑，从技术成本、建设运维成本和人力成本三个角度出发论证经济可行性如下：

1. 技术成本：系统开发选择使用开源技术和工具，仅需按相关协议使用无需付费；Amap API对于个人开发者有一定量的免费调用额度，已能满足项目需求，综合开发成本低。
2. 建设运维成本：系统部署于云服务器上，可免去系统建设中用户投入的巨量服务器和网络基础设施的购买和配置成本，仅需支付日常网络费用、停车场基础设施建设费用和软件费用，综合建设成本低。
3. 人力成本：系统配置于云服务器上，极大减少硬件运维团队的培育、薪金成本，仅需支付开发团队开发和部署服务费用即可，综合人力成本适中。

需要说明的是，项目组已经有成员长期购买云服务器和域名资源并经过公安部、工信部备案且愿意在合理范围内免费提供，无建设和运维成本；系统为综合实习项目，初衷是为了让我们更加有效掌握技术和开发流程，因此无人力成本。

综上所述，项目组一致认为，开发和设计该智能车位管理系统在经济上是可行的。

### 技术可行性

项目组结合其他类似项目经验和已有技术、资源，从满足智能车位管理系统目标用户的功能性和非功能性需求角度出发，决定系统将基于现有的成熟技术（Python，Flask，MySQL，Amap API等）进行开发，下面是对系统采用的各技术的可行性研究说明：

1. Python与Flask：Python广泛应用于后端开发的编程语言，具有强大的库支持；而Flask作为轻量级的Web框架，可以被Python引用，非常适合开发小到中型的Web应用，易于与前端技术集成。项目组各成员熟悉Flask框架，可以高效地进行开发，完成既定的设计目标。
2. MySQL数据库：MySQL作为一款高效且稳定的关系型数据库，已经广泛应用于Web系统中，能够很好地满足智能车位管理系统的数据存储需求。项目组各成员熟悉MySQL原理和应用要求，可以高效地进行数据设计与管理开发，完成既定的设计目标。
3. Amap API：Amap提供了强大的地理位置服务，包括定位、路线规划和导航功能，能够有效地支撑停车位引导和车位管理的需求，已被其他相似应用稳定调用，且此API提供一定数额的免费使用额度，官方开发文档详尽可靠，项目组有成员有WebGIS开发经验，结合Amap文档和项目实例可以进行开发，完成既定的设计目标。
4. 应用部署：选用了阿里云ECS作为云服务器平台，搭载上述技术平台依赖，能够根据需求自动调整资源，确保在高并发情况下系统的稳定运行。开发组成员已有阿里云服务器和域名网址，可提供服务器资源和HTTPS（HyperText Transfer Protocol Secure，超文本传输协议安全版本）证书，满足技术要求。

综上所述，项目组一致认为，开发和设计该智能车位管理系统在技术上是可行的。

### 时间可行性

项目组结合其他类似项目经验，根据项目的复杂度和时间要求，综合研判开发周期为5天左右。计划花费1天时间及进行项目准备、分析与设计，计划花费3天时间进行系统编码和实现，花费1天时间进行调优和文档整理。故项目组可以在规定时间内完成系统的开发。

## 技术选型

### 开发环境

开发环境是代码编码的基础工具，行业常用由JetBrains公司开发的系列语言编程IDE（集成开发环境）套件、由Microsoft公司开发的Visual Studio studio系列IDE和Visual Studio code系列代码编辑器等。VS code轻量且高效，支持多种语言环境，较之其他工具卓越的性能、多编程语言支持、轻量化等优势，能够极大提升开发者的编程效率，提供舒适的编程体验。故智能车位管理系统拟使用VS Code v1.95.3版本作为主要开发工具，部署于项目组各成员计算机上。

### 服务器选择

行业中，服务器有本地化部署和云服务部署两种类型，本地化部署要耗费大量的财力购置，且维护部署需要相关专业人士进行、应用联网审批复杂，适用于超大型项目，综合成本高；而云服务器仅需支付较为平价的运行费用，配合智慧运维面板可大大简化工作流程，节省综合成本。

国内目前提供云服务器的企业有阿里巴巴、华为、腾讯等，其中阿里巴巴的阿里云服务是目前国内规模最大、设施最齐全的云服务。阿里云ECS是由阿里云提供的性能卓越、稳定可靠、弹性扩展的IaaS级别云计算服务，免去了采购IT硬件的前期准备，广泛应用于应用部署[4]。项目组成员已有部署Debian12 Linux系统、基于Ngnix的Openresty负载均衡代理器、MySQL8.4.2数据库和智能运维面板的阿里云ECS实例并配置了备案的域名，仅需申请HTTPS证书、开放子域名即可承载系统业务，有效提高应用可用性、降低综合成本。故智能车位管理系统拟选用项目组成员提供的阿里云服务器ECS实例承载系统。

### Python语言

行业中广泛应用Java、C#、Python和Go等编程语言进行后端服务开发和搭建工作。Python具有代码高可读性、第三方库和框架支持丰富等优势，支持快速构建高效、可维护、轻量级的Web应用，越发广泛地运用于人工智能、数据科学和后端开发领域。故智能车位管理系统拟使用Python作为主要后端开发语言，版本号3.8.6。

### Flask后端框架

Flask基于Python的轻量级Web框架，广泛应用于后端开发，特别适合构建中型的Web应用。Flask提供了基础的Web服务器功能，同时集成了Werkzeug作为底层库，提供了请求和响应处理机制，以及Jinja2模板引擎，用于高效的HTML渲染，适应现代前后端分离的开发模式，逐渐成为了Web开发中的重要工具。故智能车位管理系统拟使用基于Python的Flask作为主要后端开发框架，版本号3.0.3。

### MySQL数据库

MySQL是当前最流行的开源关系型数据库管理系统之一，具有体积小、速度快、总体使用成本低，具有跨平台和开放源代码的优势，广泛用于各类站点和信息管理系统开发[3]。而Navicat是一套快速、可靠和全面的数据库管理工具，提供图形界面直观管理页面，并支持通过SSH管理云端数据库。故智能车位管理系统拟使用系统开发使用MySQL作为数据库系统，版本号8.4.2，部署于项目服务器上，拟使用Navicat软件连接服务器数据库进行数据库管理和设计，版本号16.0.11，部署于项目组各成员计算机上。

### Amap地图API

Amap API是由阿里巴巴旗下的高德公司提供的一套开发者工具，用于将地图服务、定位服务、路线规划等集成到网页或移动应用中。Amap API提供了丰富的接口，支持地理信息的展示、位置搜索、路径规划等功能，适用Web应用开发等场景。故智能车位管理系统拟使用Amap API提供地理信息位置服务。

## 体系架构设计

### 总体架构

目前主流管理系统设计有C/S（Client/Server，客户端/服务器）架构和B/S（Browser/Server，浏览器/服务器）架构两大类。C/S架构采用的是客户端与服务器之间直接通信的模式，客户端需要安装专门的软件，而服务器则负责数据处理和存储，受限于特定操作系统和硬件环境，缺乏跨平台支持。B/S架构则通过浏览器作为客户端，所有应用和数据都集中在服务器端，用户只需通过浏览器访问系统，不仅简化了客户端的要求，无需安装任何专用软件，具备跨平台特性，用户可以在不同操作系统和设备上访问系统。此外，B/S架构支持集中管理和维护方式，使得系统的升级和维护更加高效，所有更新都可以通过服务器端进行，减少了客户端的维护负担；客户端甚至可以结合WebView技术独立封装，脱离浏览器依赖转为类C/S应用，满足其他本地化需求[4]。

综上所述，B/S架构可在智能车位管理系统中提供了更为简便的客户端体验、更强的跨平台能力、更高效的系统维护和更强的扩展性，满足系统要求。故项目组决定采用B/S架构（图2.1）进行开发。

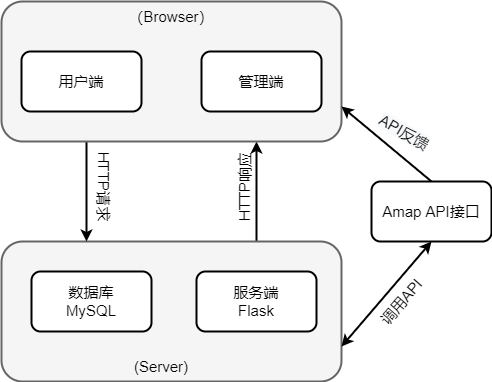


图 2.1 系统B/S架构图

### 前端部分

前端是智能车位管理系统直接面向用户的交互界面，是B/S架构的B端。前端的任务是通过图形化界面接收用户输入的请求与后端的数据交互，向API接口获取或提交数据，实现功能的实现和数据的动态更新。

项目组主要基于HTML5、CSS、JavaScript技术，以设计人机友好页面为目标，结合BootStrap5前端UI设计框架美化页面控件，集成Amap API JS请求接口，为车主用户和管理用户提供跨平台、跨终端、实时化、支持地理位置服务的前端功能页面。用户可在PC端、移动设备端甚至智能车机设备上通过浏览器联网登录系统、完成相应操作。

### 后端部分

后端是智能车位管理系统的核心处理部分，是B/S架构的S端，负责业务逻辑的实现和数据的存储管理。后端的任务是接收前端传来的请求，处理针对数据的增删改查请求和针对功能的调用请求并向前端反馈操作结果。

项目组主要基于Python语言搭载Flask框架处理前端请求，完成用户认证、信息查询与更新功能，集成Amap API请求接口提供地理位置服务，使用MySQL数据库存储信息和处理数据操作，使用云服务器和容器化技术部署系统，使用域名和HTTPS证书承载安全联网服务，保证系统服务稳定、高效运行。

## 数据库设计

### 数据库实体关系设计

针对智能车位管理系统，设计了“停车场”、“管理员”、“车主”、“车辆”四个实体。如图2.2所示，“车主”和“车辆”之间存在一对多的联系关系“拥有”；“停车场”和“车辆”之间存在一对一的联系关系“停放”和“预约”；“管理员”管理其他联系关系。

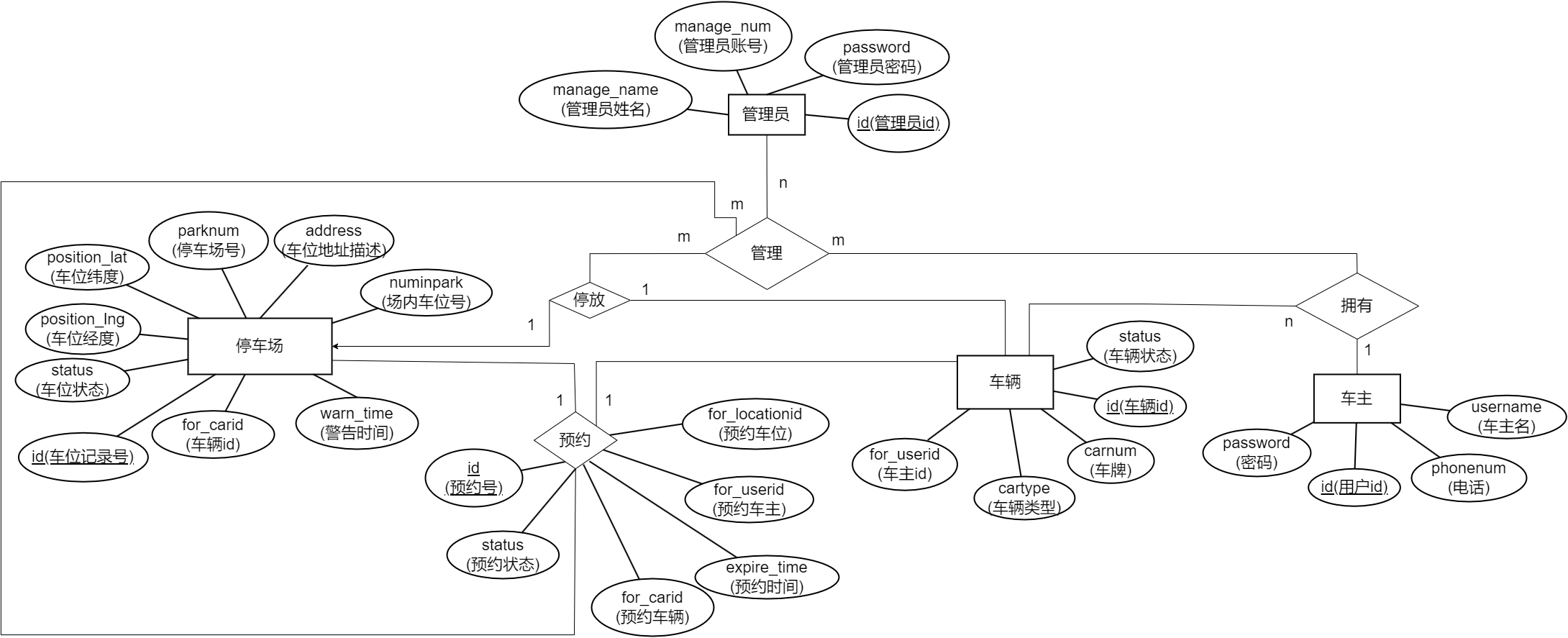


图 2.2 系统E-R图

### 数据库表

管理员信息表以id为主键，存储关于管理员的账号、密码、姓名信息，其中密码存储MD5加密数据；管理员信息表设计如表2.1所示。

表.1 管理员信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段描述 | 类型 | 长度 | 主键 |
| id | 管理员id | int |  | 是 |
| manage\_num | 管理员账号 | varchar | 50 | 否 |
| password | 管理员密码 | varchar | 32 | 否 |
| manage\_name | 管理员姓名 | varchar | 50 | 否 |

车位信息表以id为主键，存储关于车位的停车场编号、场内编号、车位经纬度位置信息、车位地址描述、车位状态码、车位停放车辆等；车位信息表设计如表2.2所示。

表2.2 车位信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段描述 | 类型 | 长度 | 主键 |
| id | 车位编号 | int |  | 是 |
| parknum | 停车场编号 | varchar | 50 | 否 |
| position\_lat | 车位纬度 | decimal | (3,6) | 否 |
| position\_lng | 车位经度 | decimal | (3,6) | 否 |
| status | 车位状态码 | smallint | 1 | 否 |
| for\_carid | 车位停放车辆 | int |  | 否 |
| warn\_time | 告警时间 | datetime |  | 否 |
| address | 车位地址 | text |  | 否 |
| numinpark | 场内编号 | varchar | 6 | 否 |

车主用户信息表以id为主键，存储关于车主的姓名、电话、密码信息，其中用户密码存储MD5加密字符串；车主用户信息表设计如表2.3所示。

表2.3 车主用户信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段描述 | 类型 | 长度 | 主键 |
| id | 用户id | int |  | 是 |
| username | 用户名 | varchar | 50 | 否 |
| phonenum | 用户电话 | varchar | 11 | 否 |
| password | 用户密码 | varchar | 32 | 否 |

车辆用户信息表以id为主键，存储关于车辆的车主、车牌、车辆类型、车辆状态码信息；车辆信息表设计如表2.4所示。

表2.4 车辆信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段描述 | 类型 | 长度 | 主键 |
| id | 车辆id | int |  | 是 |
| carnum | 车牌 | text | 10 | 否 |
| for\_userid | 车主 | int |  | 否 |
| cartype | 车辆类型 | text |  | 否 |
| status | 车辆状态 | smallint | 1 | 否 |

预约信息表以id为主键，存储预约用户、预约车辆、预约时间、预约车位号、预约状态码信息；预约信息表设计如表2.5所示。

表2.5 预约信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段描述 | 类型 | 长度 | 主键 |
| id | 预约id | int |  | 是 |
| expire\_time | 预约时间 | datetime |  | 否 |
| for\_carid | 预约车辆 | int |  | 否 |
| for\_locationid | 预约车位号 | int |  | 否 |
| status | 预约状态码 | smallint | 1 | 否 |
| for\_userid | 预约用户 | int |  | 否 |

# 车主端设计和实现

## 车主端设计

车主端是系统的重要组成部分之一，是车主使用系统功能的实现；设计充分考虑了高内聚和低耦合的行业要求，设计车辆管理和车位预约流程相对独立，用户可以单独进行账号注册、登录或管理车辆信息，而不会同时影响到预约流程。

### 车主端总体设计

车主用户使用智能车位管理系统的流程从启动车主端网页服务开始，新用户需先注册账号，然后所有用户通过输入账号信息登录系统。登录后，用户可以从停车场列表中挑选停车场，并检查所选停车场是否有空位：若有空位，用户则选择预约时间并提交预约信息，若无空位，用户预约失败，需重新开始流程。用户可取消预约，系统会自动修改预约状态和对应车位资源状态。停车结束后，用户驶离车位；系统自动修改车辆、车位状态。用户可以管理自己的车辆信息，包括添加新车辆、编辑现有车辆信息或删除不再使用的车辆（图3.1）。

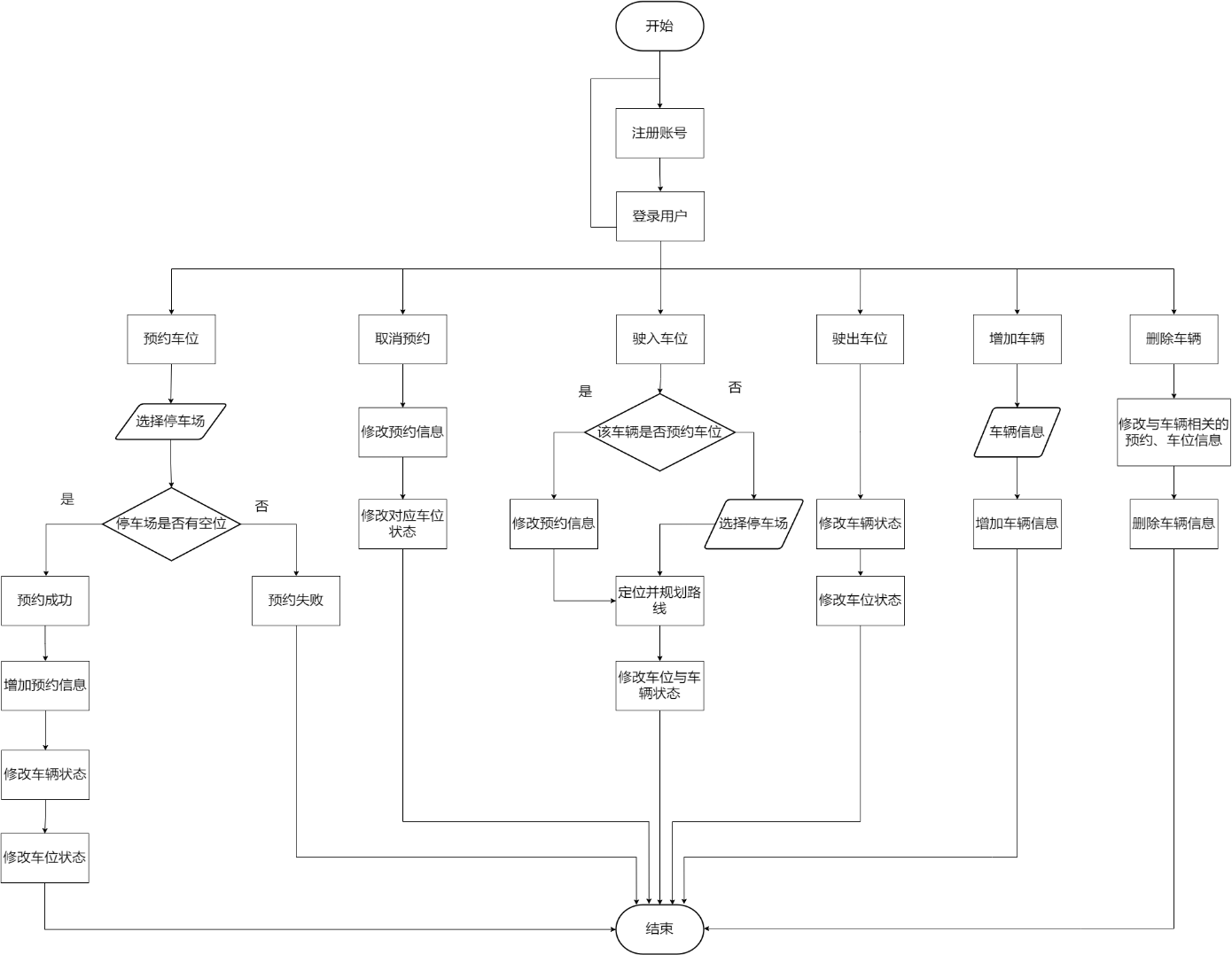


图 3.1 车主端用户流程图

车主端所有用例（图3.2）面向车主用户，用例及对应功能均需要登录验证后才能操作。

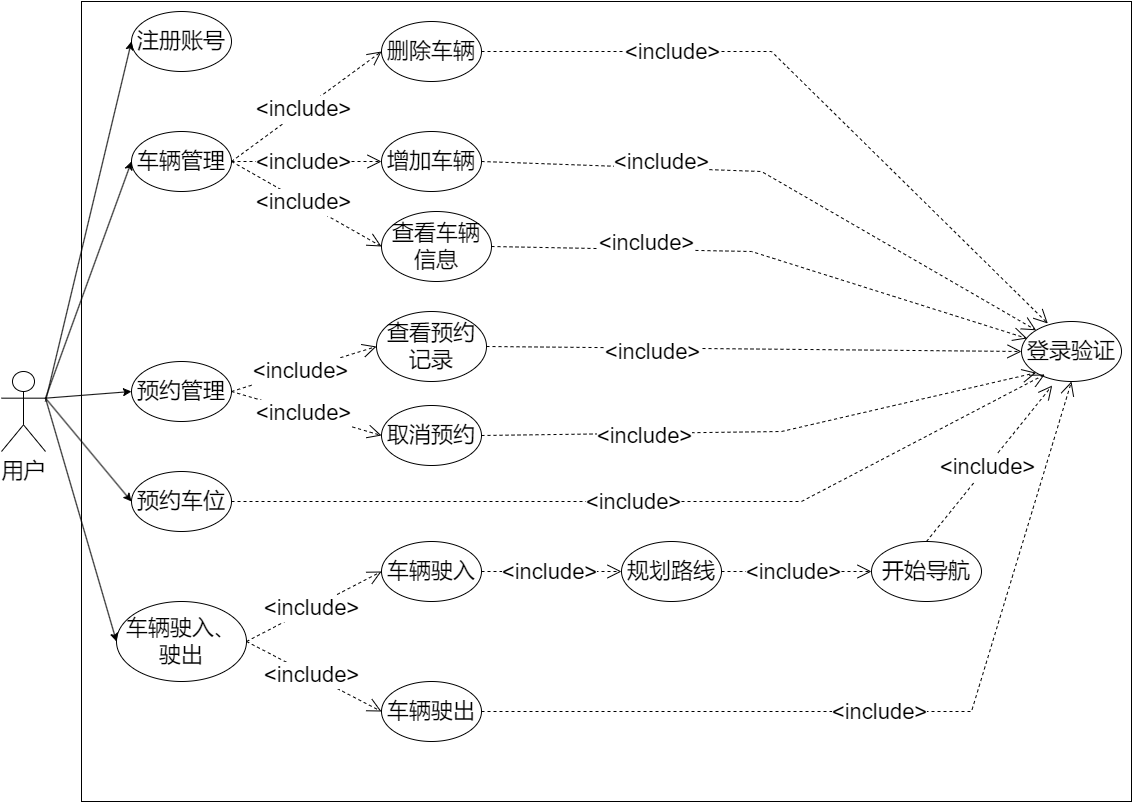


图 3.2车主端用户用例图

## 界面功能和实现

### 注册与登录

项目组在车主用户端设计了登录与注册功能（图3.3），车主用户可在此登录进入系统或注册新用户使用系统。车主使用手机号登录或注册，限制密码长度为16位，密码在后端存储为32位的MD5加密密码字符串。

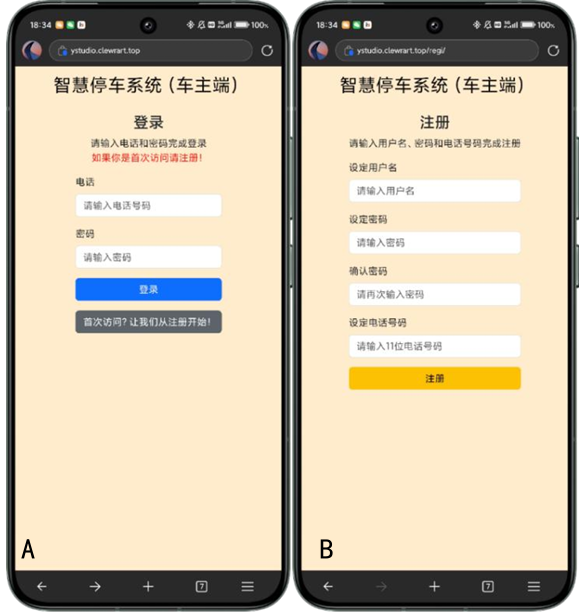


图 3.3 车主端登录与注册页面（A图：登陆页面，B图：注册页面）

车主用户成功登录后会进入仪表板（图3.4），显示当前车主、“预约记录”，“车辆管理”，“车辆信息”模块，并对名下车辆列表显示，可快捷进行驶入或预约功能。



图 3.4 车主用户仪表板

### 预约功能

项目组在车主用户端设计了预约功能（图3.5），当用户点击“预约车位”按钮时，会显示停车场选择框，用户可以选择要预约的停车场。

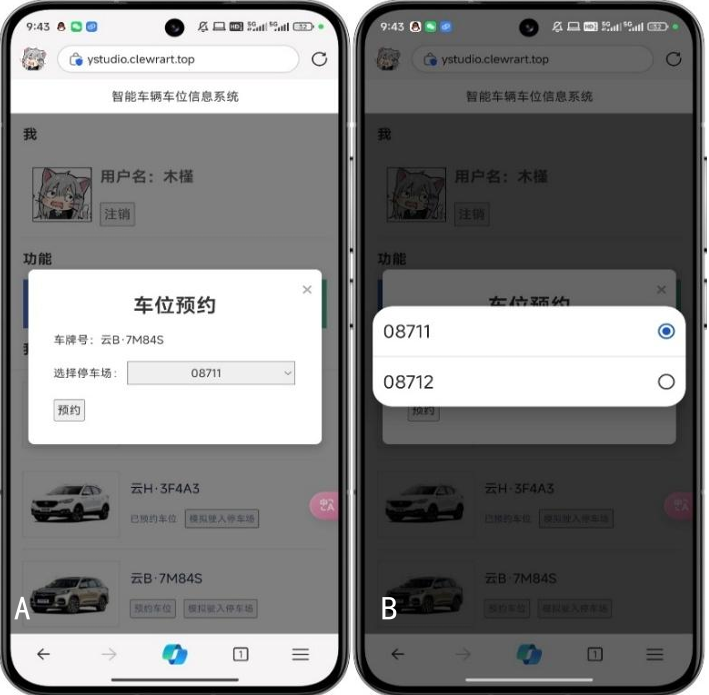


图 3.5 预约功能之停车场选择（A图：车位预约框；B图：停车场选择列表）

车主用户选好停车场并点击“预约”按钮后，如果由空闲车位则会提示用户预约成功，并且跳转至预约记录界面，显示预约记录（图3.6）。

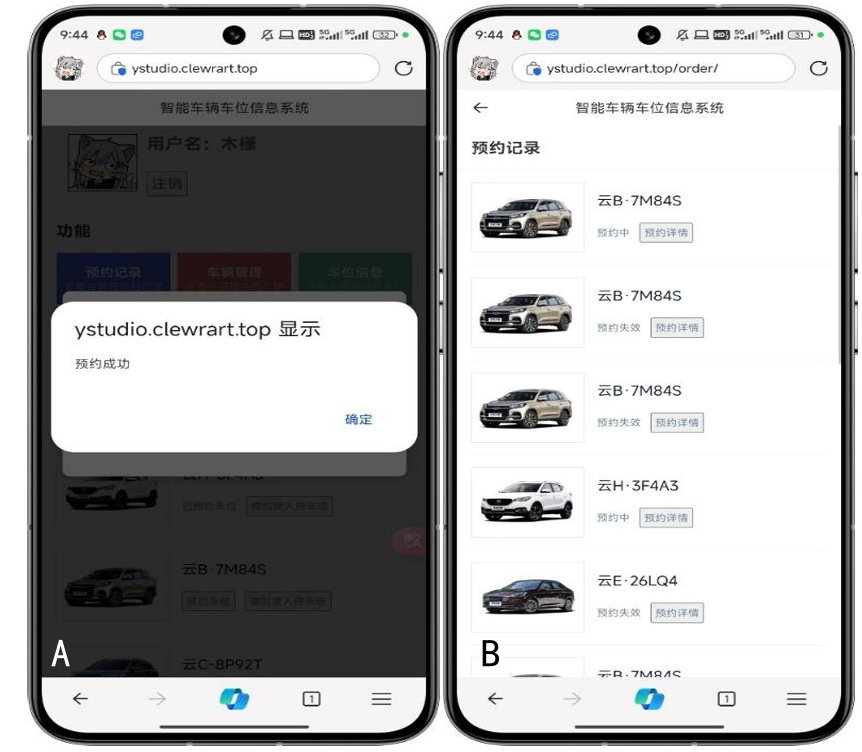


图 3.6 预约功能之预约结果（A图：弹窗提示，B图：预约记录）

车主用户在预约记录页面中，点击预约详情可以查看预约详情（图3.7），点击“取消预约”可取消预约。



图 3.7 预约功能之预约详情（A图：正常的预约详情，B图：取消状态的预约详情）

### 车辆管理

车主用户在仪表板页点击“车辆管理”按钮时，跳转至车辆管理页面（图3.8），在此界面可以查看车辆信息、删除车辆、增加车辆。当点击“车辆详情”按钮时，显示车辆详细框，可以看到车辆的信息和状态。

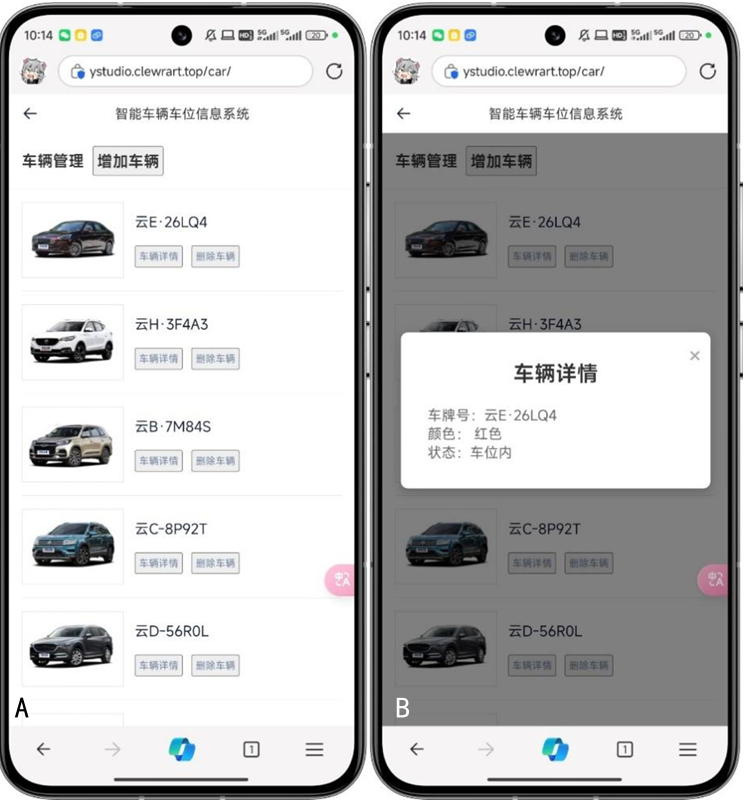


图 3.8 车辆管理功能（A图：车辆列表，B图：车辆详情）

车主用户点击“增加车辆”按钮时，会显示增加车辆的弹窗（图3.9），在这里可以填写车牌号以及选择车辆的颜色。

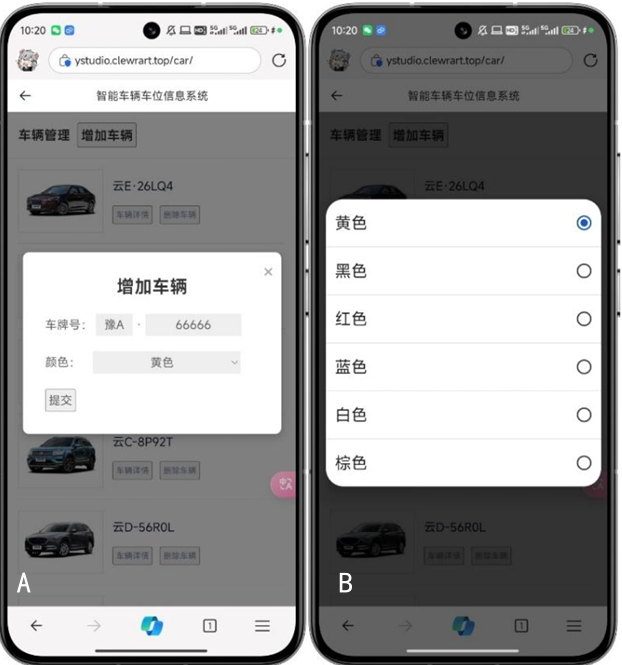


图 3.9 车辆增加功能（A图：增加车辆信息，B图：车辆颜色选择）

当点击“提交”后，如果格式正确则刷新页面，否则会提醒“车牌号格式不正确，请重新输入！”（图3.11）。为保证车牌号格式正确，提交信息部分设计了正则表达式（图3.10）检查检验车牌号格式，格式要求为文字＋字母+5或6位字母或数字组成的字符串。

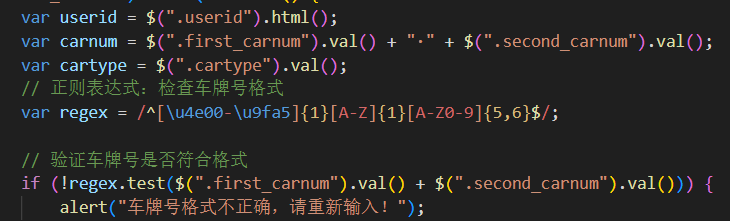


图 3.10 正则表达式检查检验车牌号格式代码



图 3.11 车辆信息增加结果（A图：车牌格式正确状态，B图：车牌格式错误提示）

用户点击“车辆删除”后，页面会弹窗提示“删除成功”，点击弹窗中确定按钮后会移除用户刚才删除的车辆信息（图3.12）。

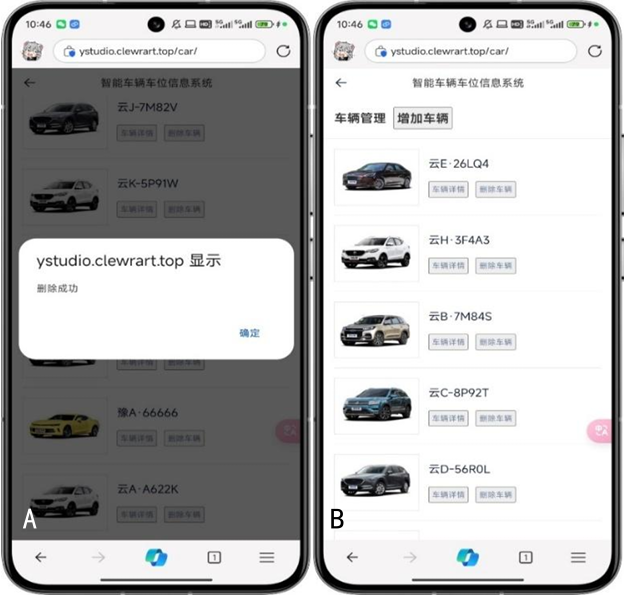


图 3.12 车辆信息删除（A图：删除成功提示，B图：更新后的车辆列表）

### 车位信息

用户在仪表盘点击“车位信息”按钮后，跳转到车位信息页面（图3.13），可以看到用户已预约或已占用的车位信息，点击“车位详情”即可查看车位详情。

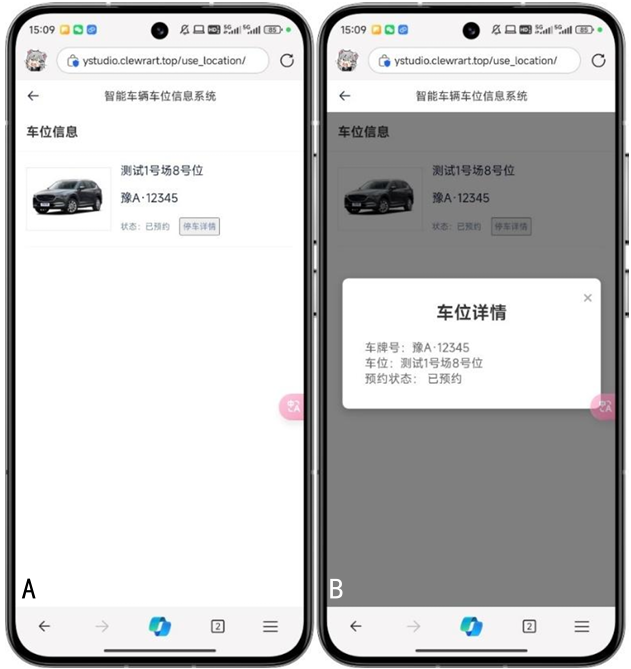


图 3.13 车位信息（A图：已预约的车位信息，B图：预约的车位详情）

## 车位寻路与导航功能设计及实现

### 驶入车位选择

车主用户在仪表盘点击某一车辆“模拟驶入停车场”的按钮时，将模拟车辆进场过程，如果车辆未预约车位，则会弹窗让用户选择停车场，选择停车场后点击“模拟驶入”，则会进入搭载Amap API的在线定位与路线规划界面并提示目标车位、当前位置和定位精度（图3.14）。

移动设备默认获取的是GPS坐标系，中国大陆地区需进行坐标转换成GCJ-02坐标才可在地图上正确显示当前位置，定位精度与设备GNSS定位系统频段配置、搜星能力、周边干扰、通信管制和卫星信号等软硬件因素有关[7]。

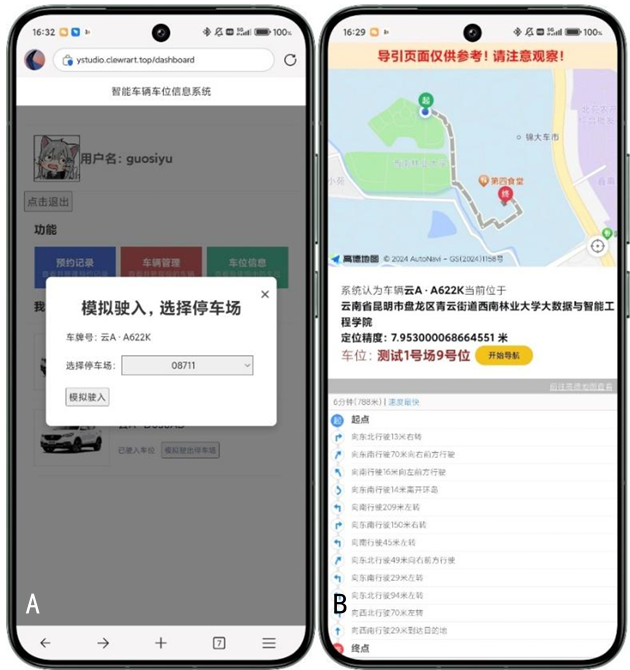


图 3.14 模拟驾驶驶入停车场（A图：停车场选择，B图：自动定位和路径规划页）

当车主用户点击下侧局部路段时，会在地图中放大化显示该段路线细节（图3.15）共供驾驶人参考。



图 3.15 路径规划局部路段查看

### Amap API调用功能

路径规划功能调用了Amap API，表3.1列出了启用的功能对象。

表.1 车位信息表

|  |  |
| --- | --- |
| 功能对象方法名 | 功能描述 |
| AMap.Map | 创建、加载地图对象 |
| Map.Geolocation | 进行定位和GCJ-02、GPS坐标转换 |
| AMap.Driving | 行车路线规划 |
| AMap.plugin | 加载控制插件和信息 |

### 导航跳转

导航功能受Amap API支持限制，当用户点击图3.14B所示的黄色“开始导航”按钮后，则会传递目标车位经纬度和当前用户位置至高德地图APP（图3.16），进行实时导航[7]。

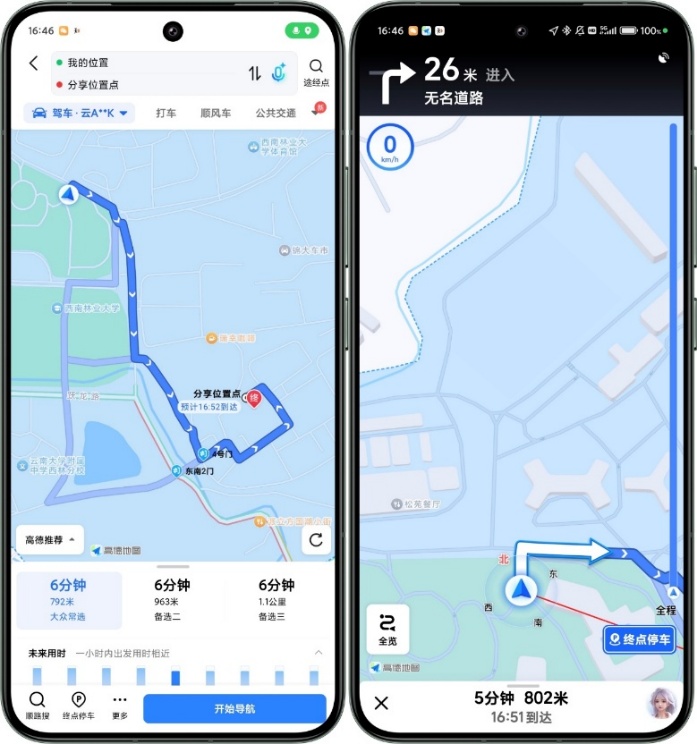


图 3.16跳转高德地图APP导航

# 管理端设计和实测展示

## 管理端设计

管理端也是系统的重要组成部分，用于实现对系统内各类资源和数据的有效管理与维护。通过功能模块划分，实现高内聚低耦合的行业要求，实现高效管理。

### 管理端总体设计

管理用户使用智能车位管理系统的流程从启动管理端网页服务开始，用户进入管理登录页面输入管理账号密码来判断用户角色，如果用户是管理员，可以新增、编辑、删除用户信息，查询、删除、查看和统计预约信息，新增、编辑、删除和查询车位信息，查看、编辑、删除车辆信息；如果用户不是管理员则登录失败（图4.1）。

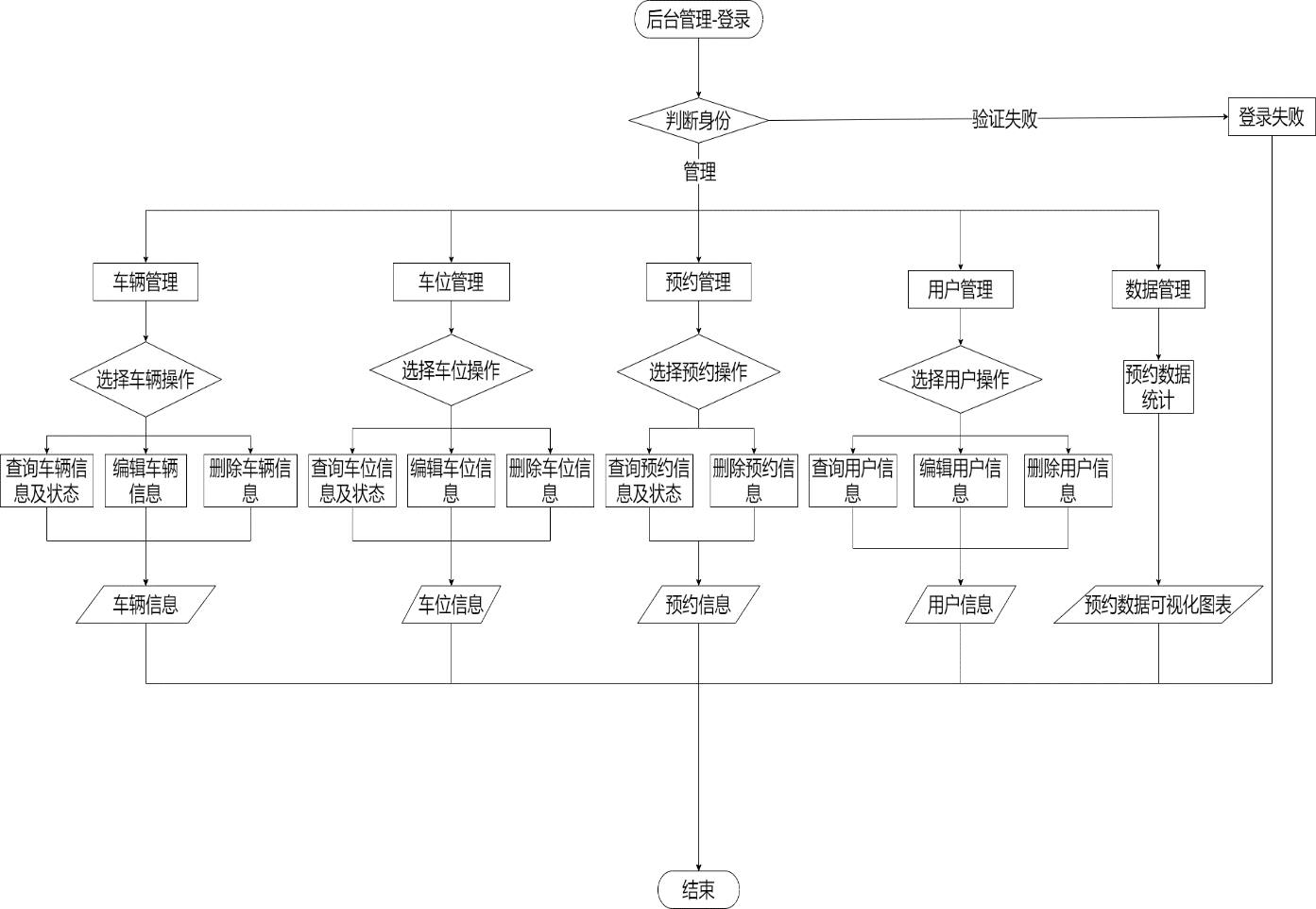


图 4.1 管理端用户流程图

管理端所有用例均面向管理员用户，所有用例需要管理员通过登录验证后方可进行操作，以确保系统的安全性和数据的有效管理（图4.2）。

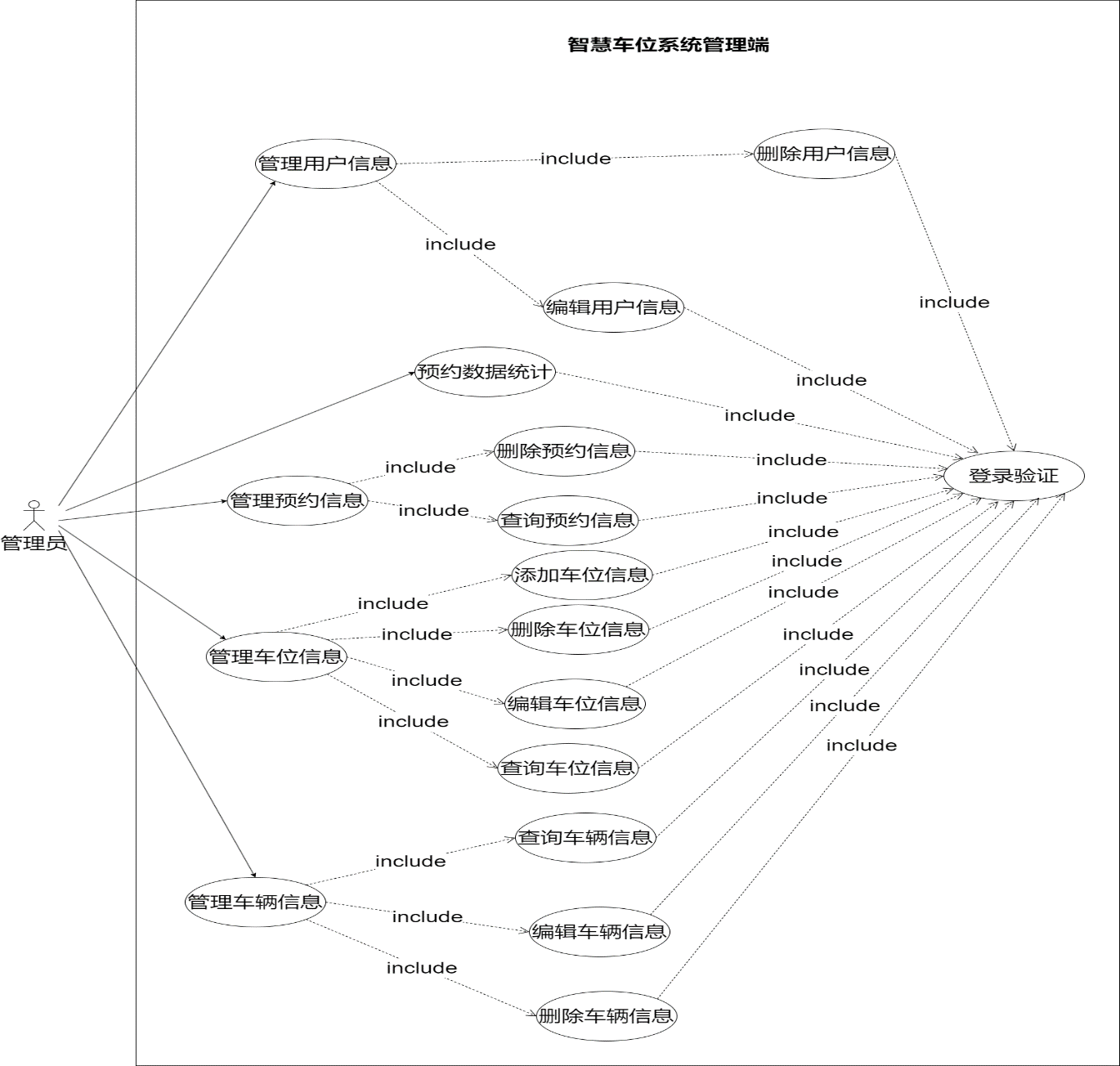


图 4.2 管理端用户用例图

## 界面功能设计及实测展示

### 管理登录

系统设计了管理登录页（图4.3），需要输入管理员账号密码才能进入管理页面。项目组在设计时出于安全管理考虑，管理端和车主用户端没有部署在同一网址下。



图 4.3 管理登录页

### 车辆管理

管理员用户登录后可以按报表类型查看已登记的车辆信息，显示车牌号、状态、车主姓名和电话信息，左侧导航栏可以快捷跳转各管理功能（图4.4）。

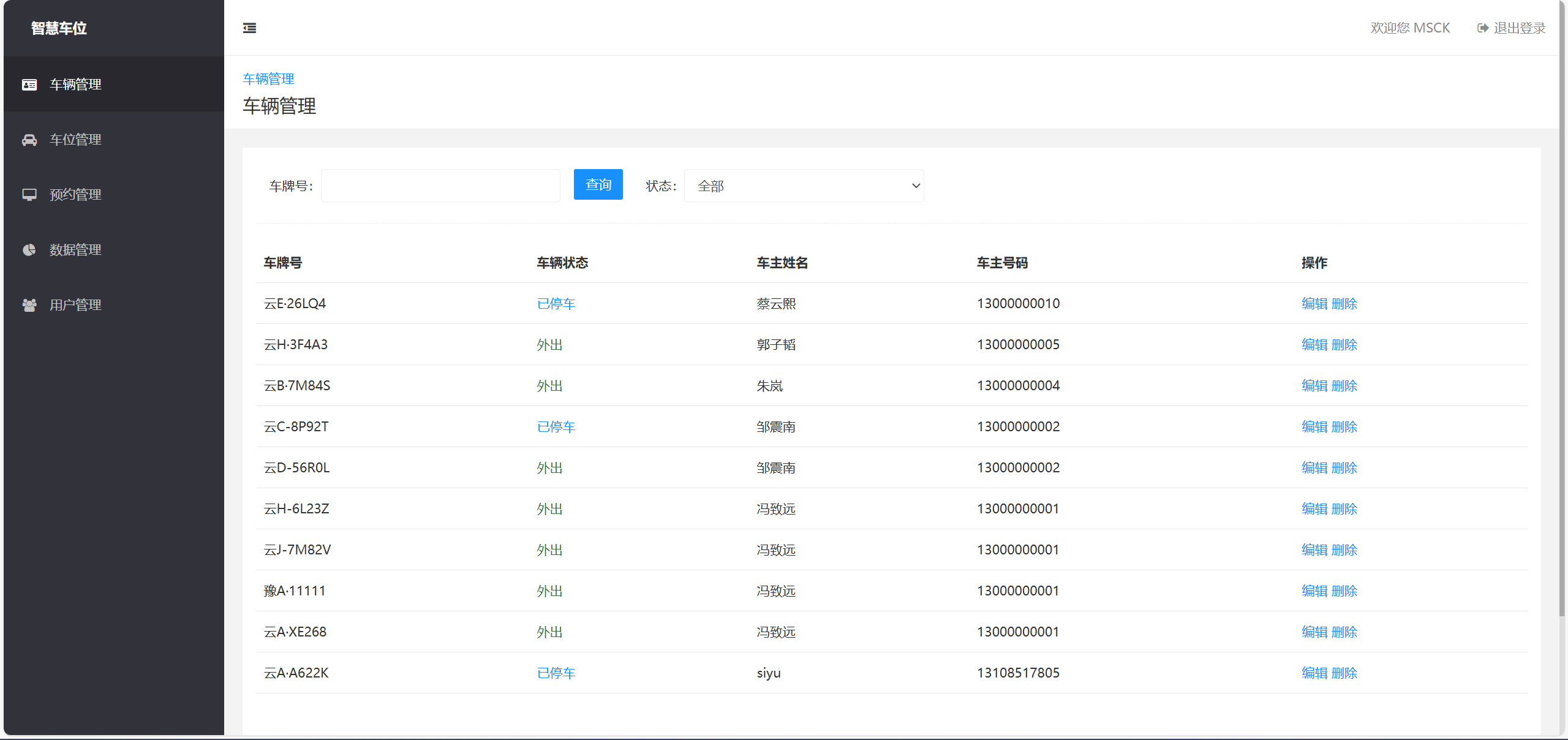


图 4.4 车辆信息管理查看功能

管理员用户在车辆信息查看页面点击编辑操作按钮后，可以编辑车牌号车主姓名和电话信息。（图4.5）

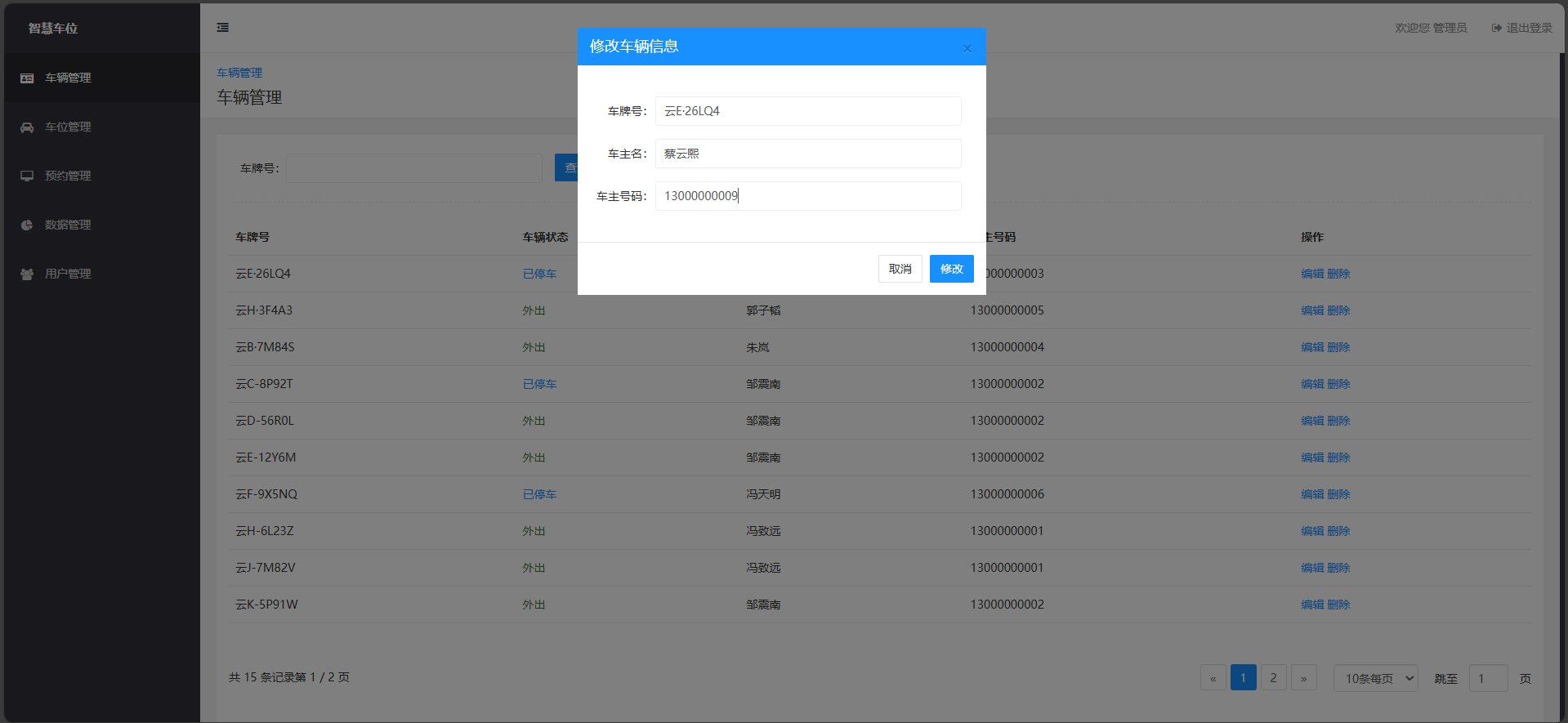


图 4.5 车辆信息管理修改功能

管理员用户在车辆信息查看页面点击删除操作按钮后，会弹窗提示删除确认信息，点击确定会删除用户信息（图4.6）。

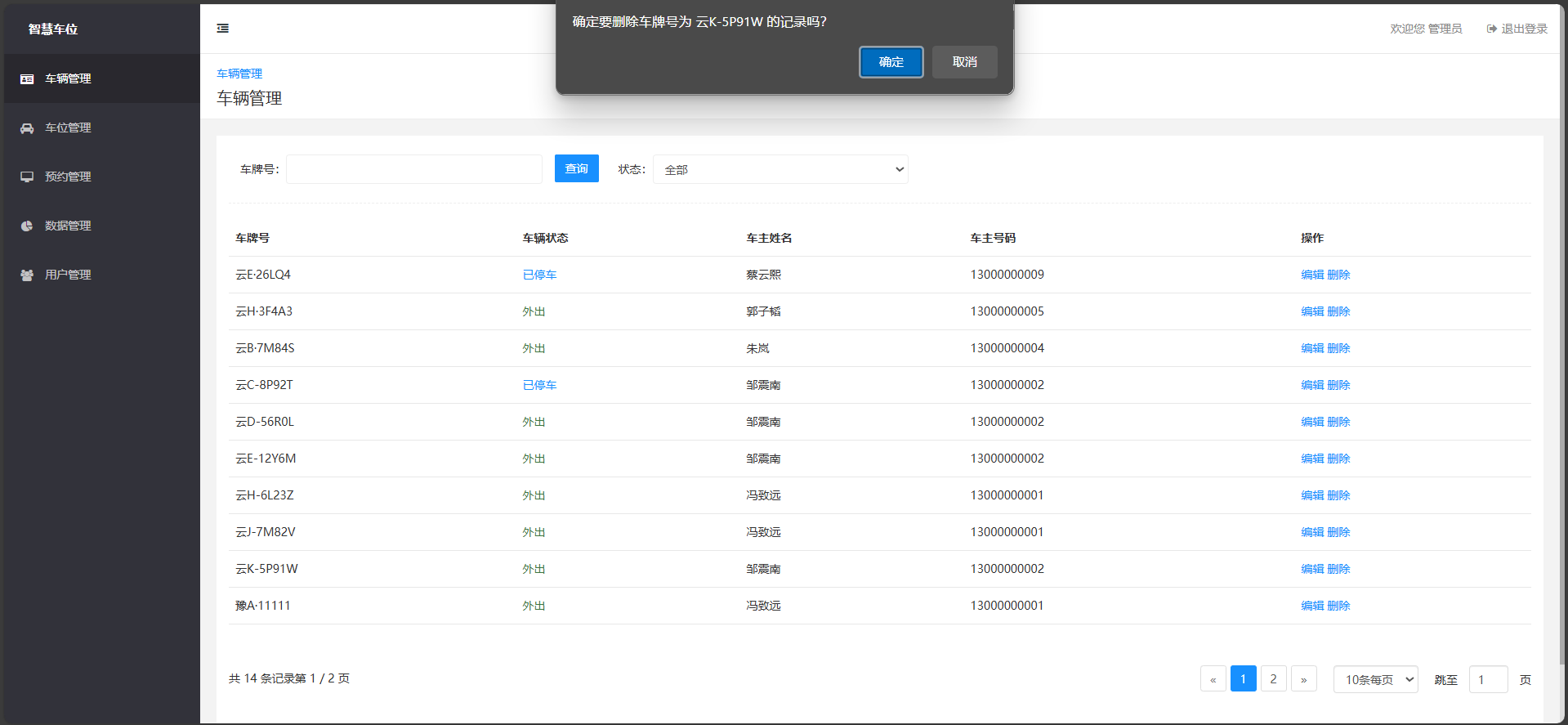


图 4.6 车辆信息管理删除功能

### 车位管理

管理员用户可以查看可用车位信息，会显示车位编号，对应停车场及场内编号、车位状态、占用状态、报警事件（图4.7）。

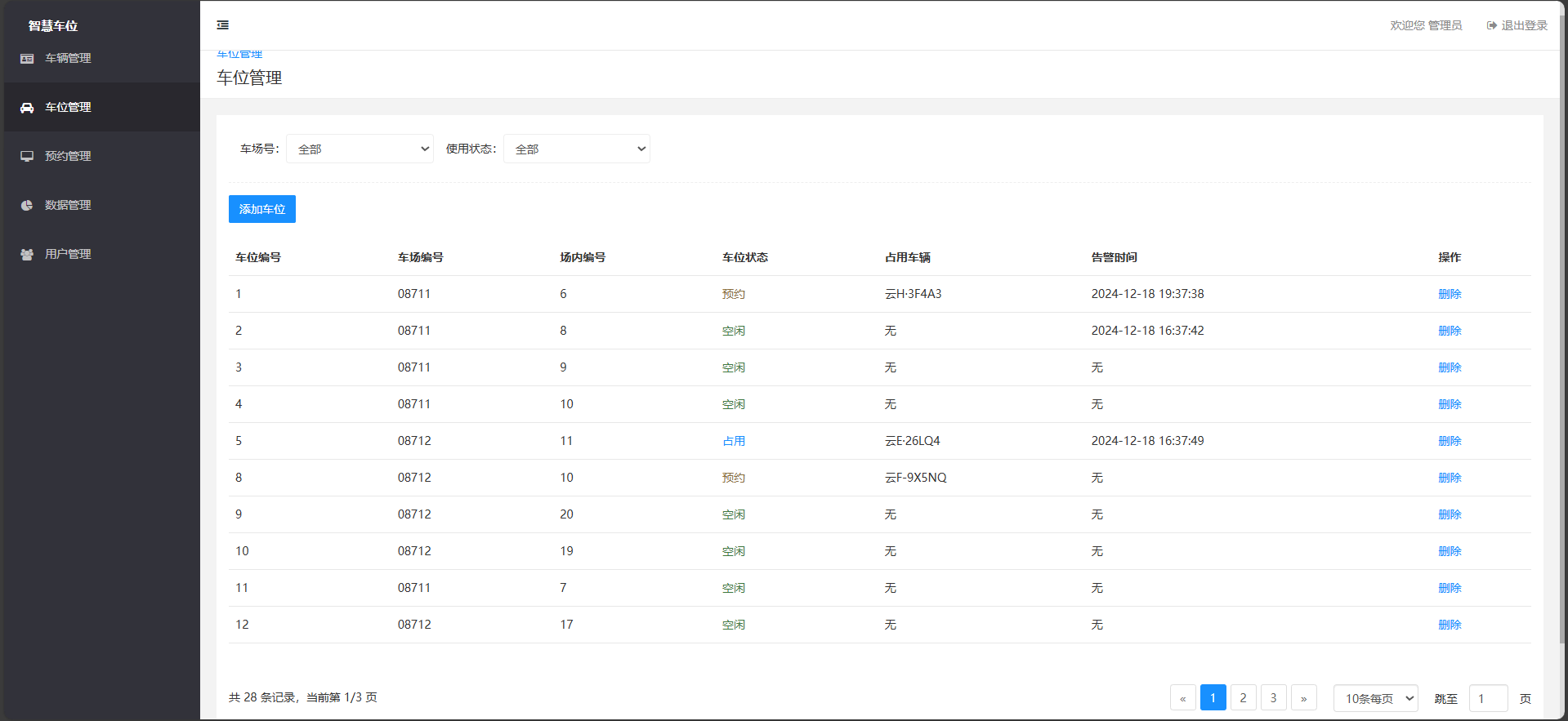


图 4.7 车位信息管理查看功能

管理员用户可以新增可用车位信息，需要输入车位编号，场内编号、基于GCJ-2的车位经纬度位置（图4.8）。



图 4.8 车位信息管理新增功能

管理员用户可以删除车位信息，此时会弹窗提示删除确认信息（图4.9），点击确定会删除对应车位。

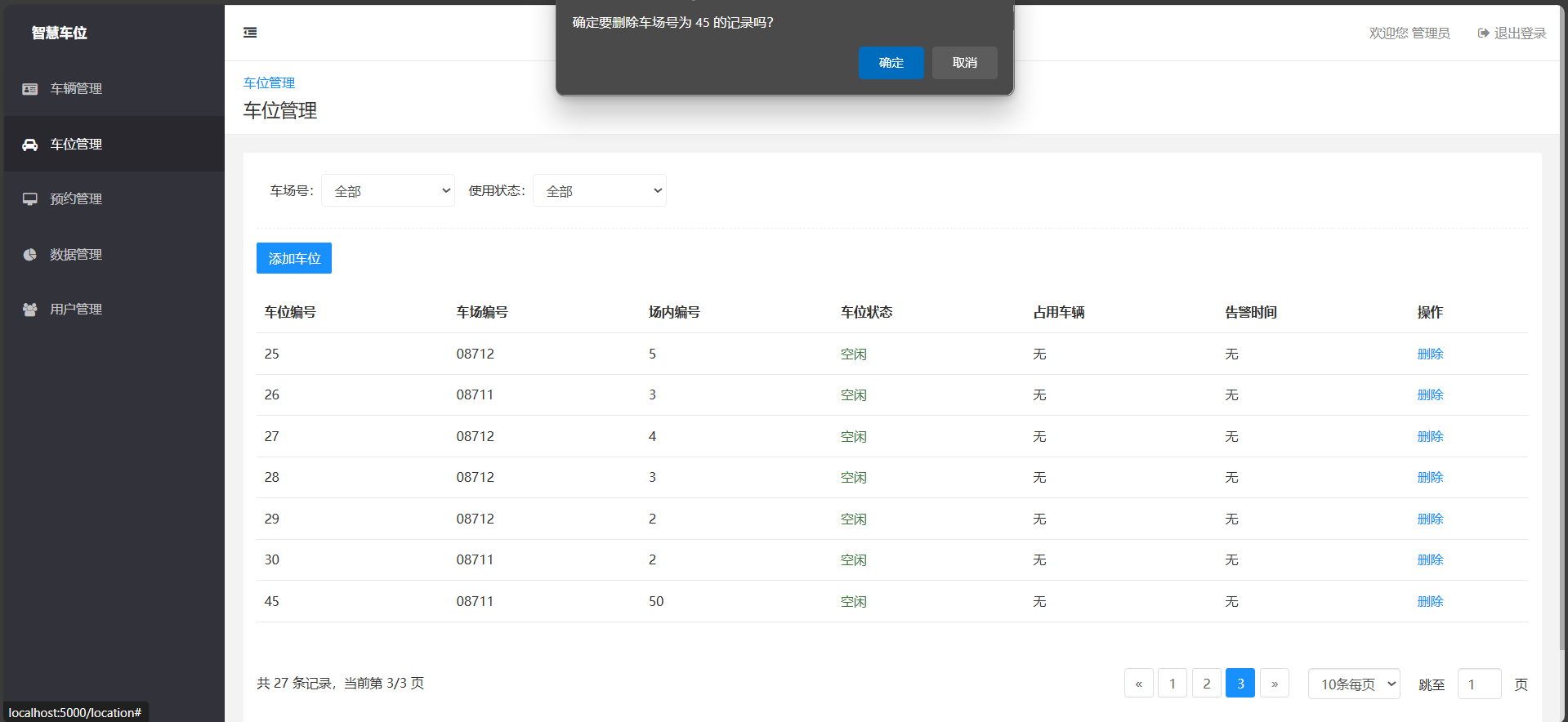


图 4.9 车位信息管理删除功能

### 预约管理

管理员用户可以查看各类预约信息，会显示预约ID、预约车辆、预约时间、预约状态、预约车场和车位号、预约人和电话号码（图4.10）。

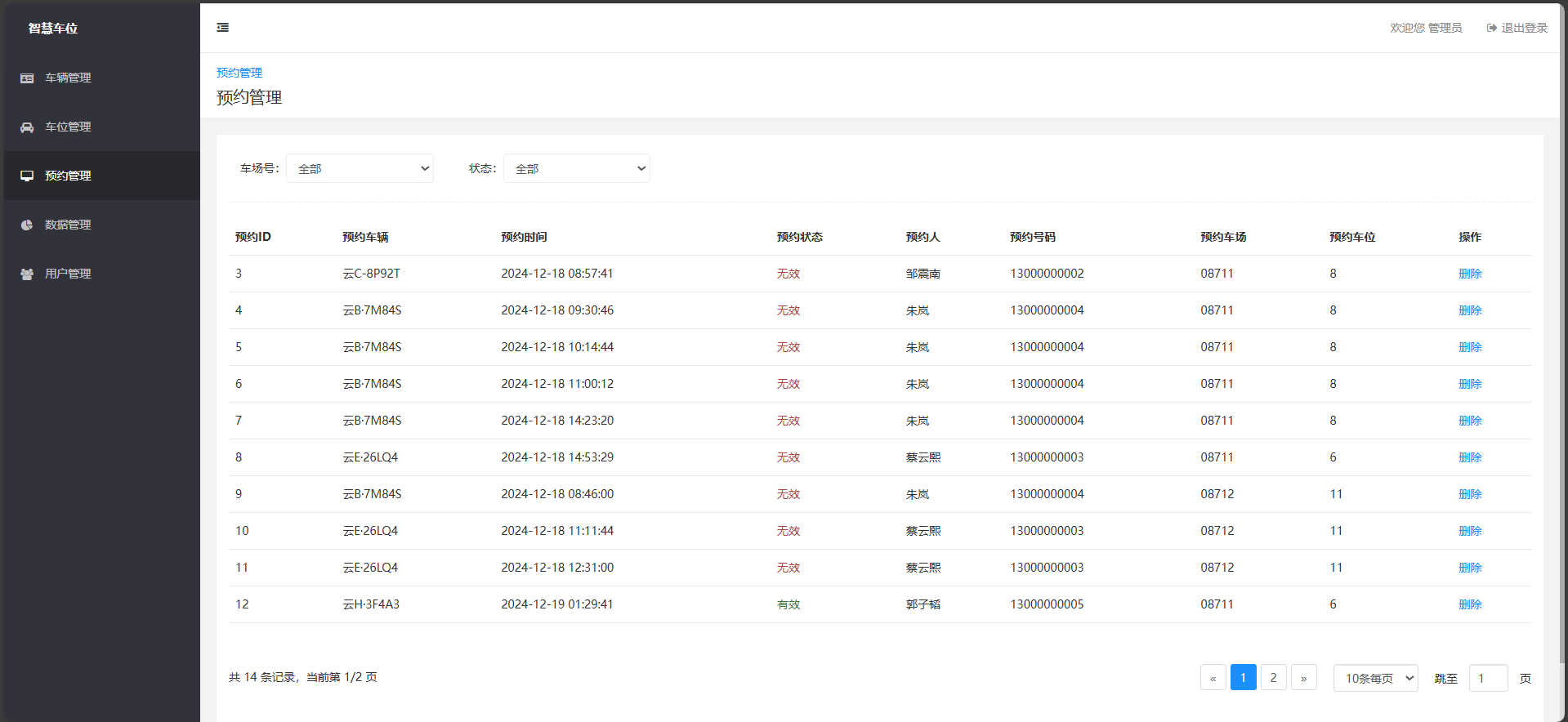


图 4.10 预约信息管理查看功能

管理员用户可以删除各类预约信息，此时会弹窗提示删除确认信息（图4.11），点击确定会删除对应预约信息。

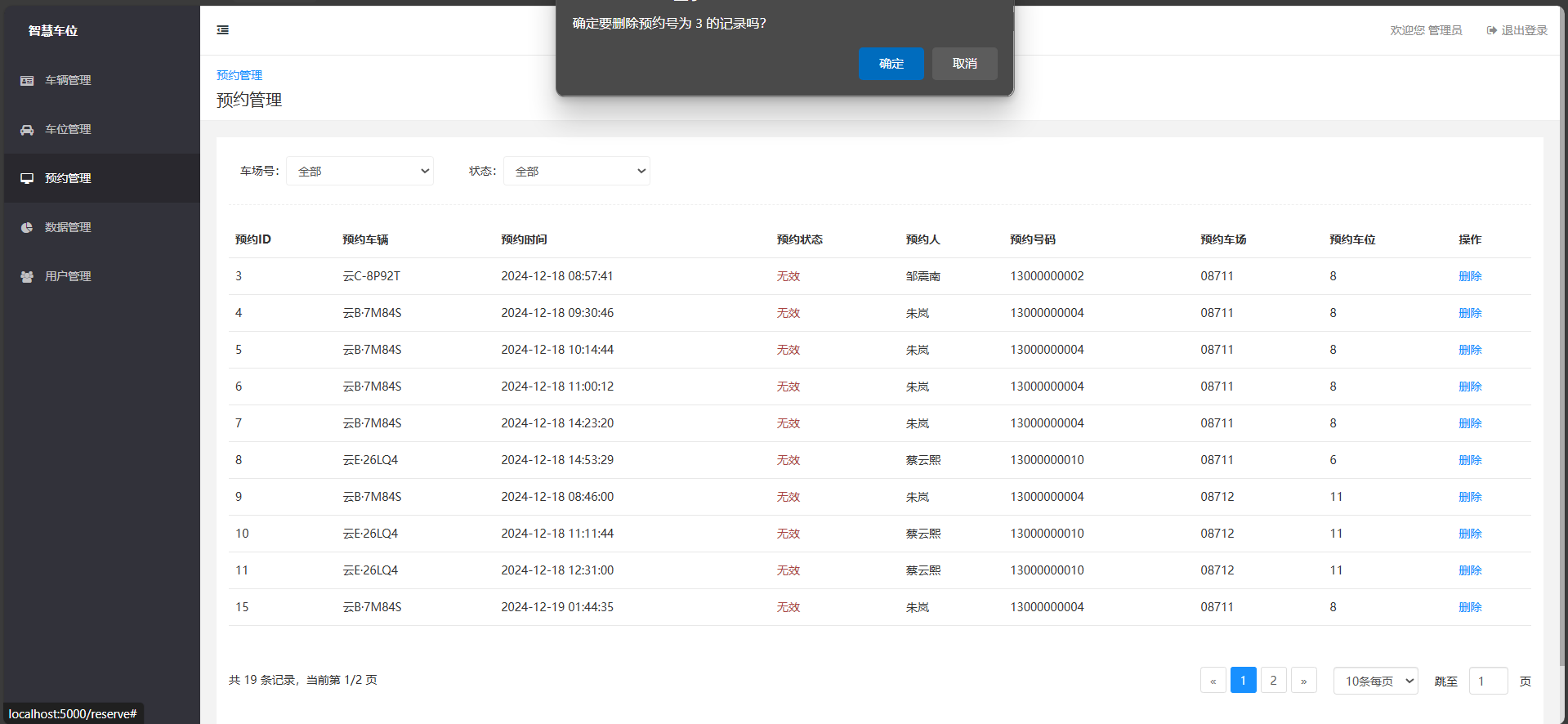


图 4.11 预约信息管理删除功能

### 车主管理

管理员用户可以查看车主信息，会显示用户ID、用户名、用户手机号信息（图4.12）。

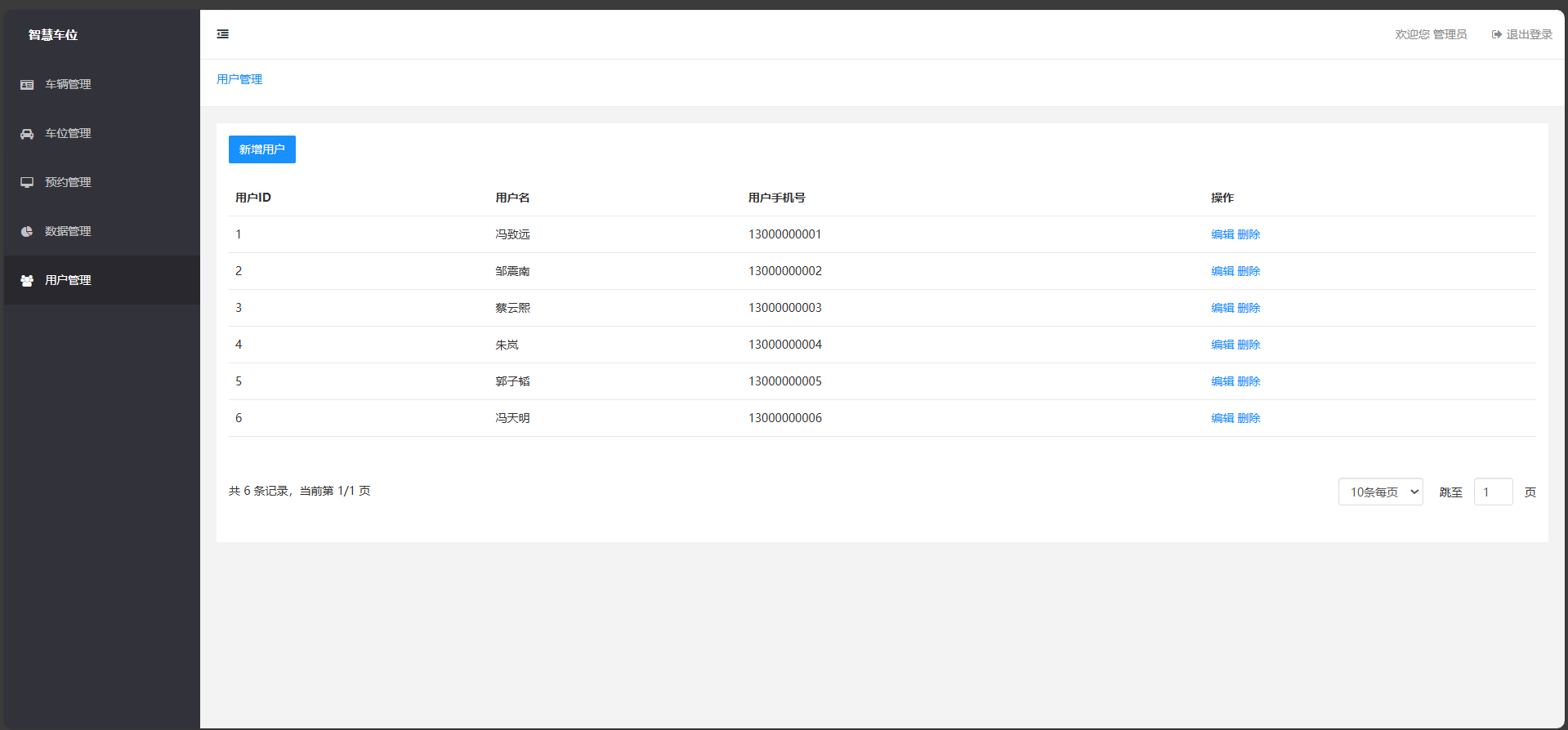


图 4.12 车主信息管理查看功能

管理员用户在车主信息查看页面点击编辑按钮后可以编辑车主姓名和电话信息（图4.13）。

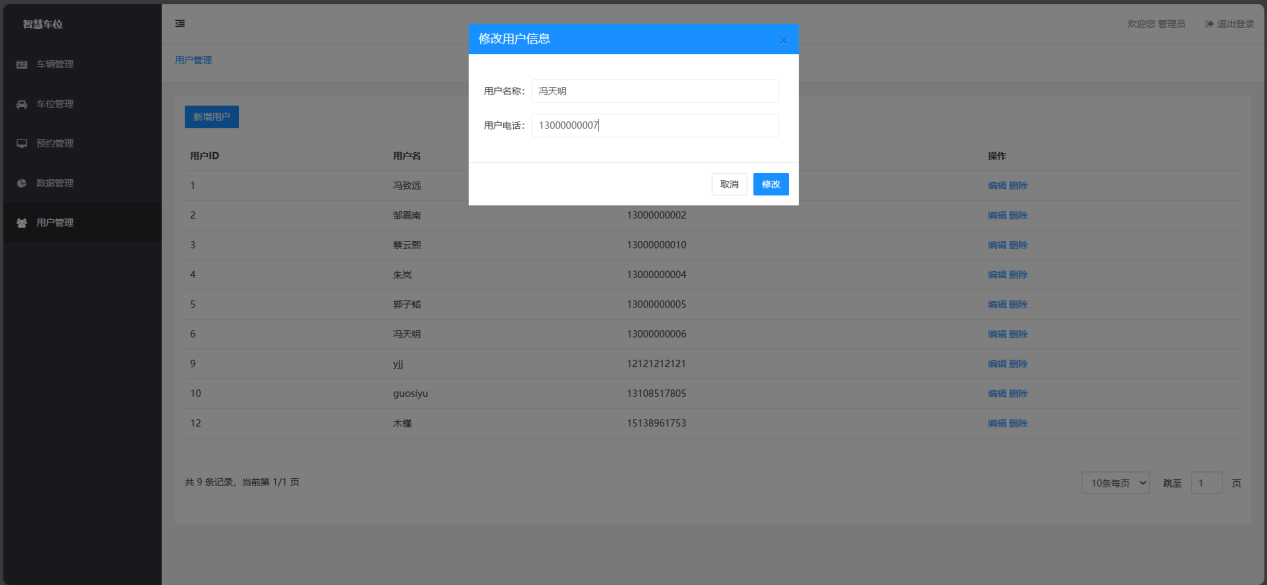


图 4.13 车主信息管理修改功能

管理员在车主信息查看页面点击删除操作按钮后，会弹窗提示删除确认信息（图4.14），点击确定会删除用户信息。系统执行车主信息删除时将关联删除其名下的车辆和预约数据。

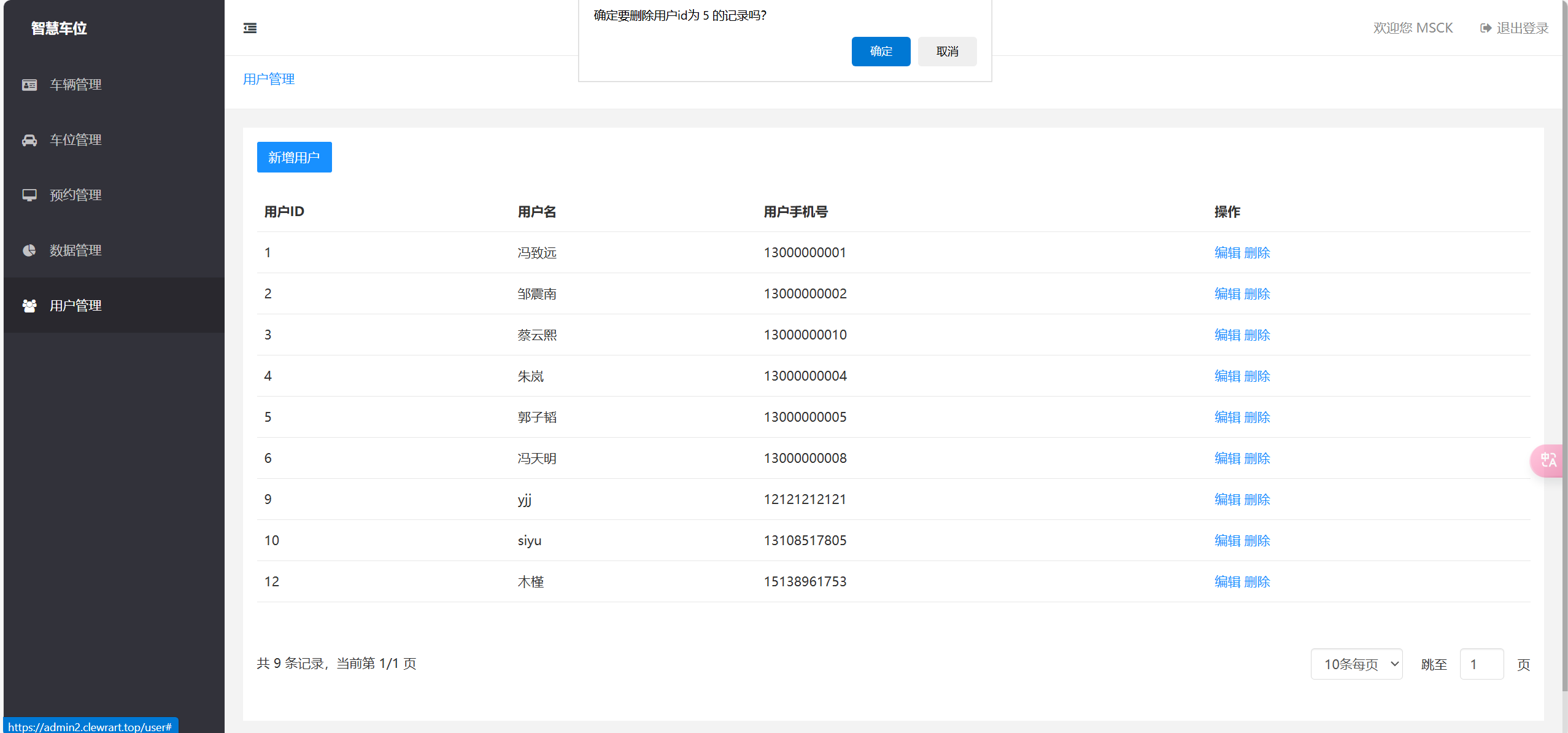


图 4.14 车主信息管理删除功能

### 综合管理可视化看板

综合管理可视化看板是智能车位管理系统的特色功能，支持按停车场、按时间序列对系统运营数据通过图表和统计信息进行可视化展示（图4.15）。

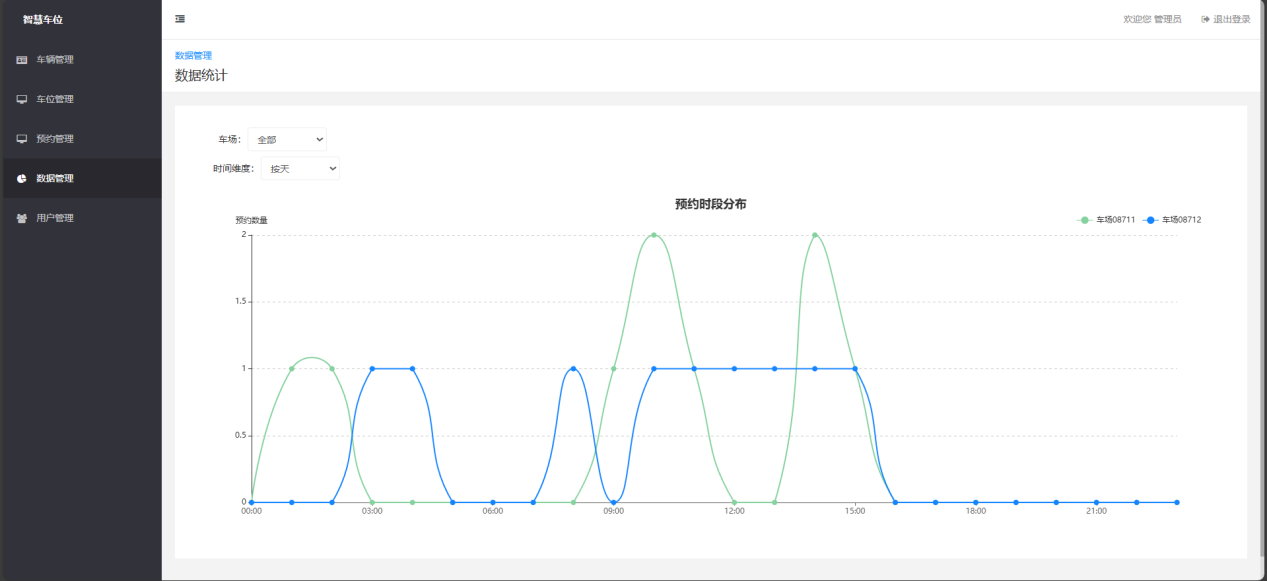


图 4.15 可视化看板功能

# 系统服务部署

## 部署目标

为了实现系统的公网访问，让课程设计项目更加贴近于实际运用，需要将系统部署到云服务器上，确保系统能够从互联网访问，并具备良好的性能，保障数据安全。

## 云服务器选择与配置

### 服务器选择和设计

本系统选择了阿里云ECS（Elastic Compute Service）作为部署平台，ECS具备高可用性、弹性扩展以及稳定可靠的计算性能。ECS服务器的选择依据项目的预期访问量和负载进行配置，确保系统能够稳定运行。服务器选择弹性实例e（2核心2GB内存40G ESSD存储），搭载Debian 12 Linux操作系统和OpenResty反向代理与负载均衡器，将系统管理端和用户端业务代码分别以镜像形式封装运行，安装MySQL 8.4.2作为数据库系统，配合智能运维面板进行系统的监控和管理。

### 安全配置

进行安全配置是为了确保系统服务的安全性和可靠性。通过设置防火墙和开放必要的端口，可以防止未授权访问，保护数据不被泄露或篡改。使用HTTPS证书加密数据传输，验证网站身份，防止中间人攻击，增强用户信任。部署系统车主端和管理端在不同子域名下，可实现逻辑功能隔离，降低攻击错误使用风险，提升整体安全性。智能车位管理系统在部署时主要做了以下安全配置：

1. 在云服务器上配置防火墙，并按规则开放管理端网址和客户端网址的80（HTTP）和443（HTTPS）端口。
2. 因项目已有可用域名，按照规定可以申请HTTPS证书。HTTPS证书（也称为SSL/TLS证书）是确保网站安全和保护用户隐私的重要工具。HTTPS是HTTP的安全版本，使用SSL/TLS协议加密数据传输，确保数据在客户端和服务器之间安全传输，可开启加密通信功能同时验证网站身份以及防止数据篡改。车主用户端和管理端页面HTTPS证书可浏览器成功识别并开启加密通信（图5.1）。
3. 将系统面向公众部分（车主端）和管理部分分别部署于两个子域名下，从划分上完全隔离不同用户群体的操作部分，减少潜在的攻击面和爬取，从而提高系统的安全性。

图 5.1车主用户端（左）和管理端页面（右）HTTPS证书已被浏览器成功识别

## 系统部署地址

用户端部署于https://ystudio.clewrart.top；

管理端部署于https:/admin2.clewrart.top；

因服务器资源和https证书有效期有限制，上述网址仅供在2024年12月21日23:59前可用。

# 总结与展望

随着智能车位管理系统的开发完成，系统已经实现了基于Python与Amap的车位导航与管理功能，并部署在云服务器上，具备了跨平台访问能力，完成了既定的设计目标。通过结合Amap的定位和导航功能，车主能够实时预约空闲车位，并通过系统定位并引导到车位，从而有效减少了停车场内的交通压力。同时，停车场管理者也能够通过系统实时监控车位的使用情况，进行高效的车位调度与管理，进一步优化了停车资源的分配。

然而，随着技术的不断进步和用户需求的变化，系统仍有很多提升空间。例如，系统可与微信等常用社交平台进行深度集成，实现无感登录和注册；引入地下引导系统，帮助车主在地下停车场等复杂环境中进行精准定位和导航甚至是像北京大兴国际机场那样的机器人自动化引导服务[5]；引入ETC车辆识别与无感支付技术，实现车辆信息自动识别和自动收费；未来还可接入全国各地正在建设的智慧城市平台接口，将本系统与智慧城市系统无缝连接，联动其他交通设施，对城市交通进行全面分析和预测，优化整体交通规划[6]，最终为城市的智慧化发展、提高城市交通运输效率、促进城市交通运输领域治理体系和治理能力现代化。

参考文献和资料

1. 张葳葳、刘静、王凯．基于AMAP API的网络地图应用服务研究[J]．《北京测绘》．2017，第01S期，322-324
2. 黄日明.停车场智能停车引导方法及系统研究[J].华南理工大学专业学位硕士学位论文,2021年，第02期.
3. 赵晓侠、潘晟旻、寇卫利 等.《MySQL数据库设计与应用（慕课版）》.2022.03.
4. 阿里云开发者平台. https://developer.aliyun.com
5. 明晨辉 等．大型交通枢纽停车场管理的思考——以杭州西站停车系统为例[J]．《中国物业管理》．2024，第11期，101-102
6. 张恒 等．城市级智慧停车管理系统设计与研究[J]．《智能建筑与智慧城市》．2024，第1期，170-172
7. 高德开放平台.Amap API开发参考手册. https://lbs.amap.com/api/javascript-api-v2/documentation

指导教师介绍

鲁宁：男，博士，教授，西南林业大学大数据与智能工程学院教师，计算机科学与工程系副主任，2020年毕业于南京农业大学，获农业信息学博士学位。曾获国家留学基金委西部项目资助，到美国犹他州立大学遥感与地理信息系统实验室访学1年。长期从事计算机、遥感教学和科研工作，教学上主讲《数据库原理与应用》、《遥感原理与遥感图像处理》等多门本科课程，主编教材4部，参编4部。科研上围绕3S技术在农林生长监测开展研究，主要专注于无人机遥感在农林作物理化参数反演研究。构建了基于无人机RGB影像小麦地上部生物量低成本、高效监测方法；提出了基于无人机多视角影像改进氮素营养指标估算的多角度观测方法。主持国家自然科学地区基金2项、云南省科技厅基金1项，以第一或通讯作者发表中科院2区SCI论文5篇。

致谢

不到一周的实习时间匆匆而过。从设计之始至完成归档，在这昆明的冬日时节，我们三人通力合作，智能车位管理系统开发初显成果。

在这段奋斗的日子里，我们都以无限的激情击溃昆明冬日冷风，获得了无数的收获。在完成系统的整个过程中，我们遇到了许许多多困难和障碍，不过在鲁老师的指导和帮助之下，这些障碍、困难终究从阻碍我前进的“绊脚石”成功的转化成系统运行的一个个“代码石”。从系统设计、开发、最后的论文格式校对，鲁老师都在悉心指导，我们也在整个过程中收获颇多。

山外有山，人外有人。由于我们三人自身学术造诣水平有限，本次设计难免存在不足之处，在此恳请各位老师和同学批评和指正。

开发小组成员介绍

闫建槿：男，本科在读，学号20221152022，2022年进入西南林业大学大数据与智能工程学院攻读计算机科学与技术专业，班级学习委员。积极参与国内各专业比赛，获省级、国家级奖项若干。兴趣方向为应用开发和人工智能技术。

郭思宇：男，本科在读，学号20222601017，2022年进入西南林业大学湿地学院就读，2023年转入大数据与智能工程学院攻读计算机科学与技术专业。兴趣方向为网络安全和机器学习技术。

廖祥辉：男，本科在读，学号20221152014，2022年进入西南林业大学大数据与智能工程学院攻读计算机科学与技术专业，班级安全委员。积极参与国内外各专业比赛，获国家级奖项若干。兴趣方向为算法和智能计算技术。