# 摘要

在城市化进程中，城市拥堵，停车难等问题日益突出，传统停车场已无法完全满足智慧城市的需求。因此，如何解决这些问题并优化城市停车场管理信息系统成为一个亟需解决的问题。本研究将对整体系统做系统分析，并对各功能模块进行设计，使用C++，QT，MySQL，OpenCV等技术对停车场管理信息系统进行功能实现。分析、设计并实现一个包含注册登陆，用户管理，预约车位，车牌识别等功能的停车场管理信息系统。

**关键字：**停车产管理系统，系统分析，系统设计，MySQL

# Abstract

In the process of urbanization, urban congestion, parking difficulties and other problems are becoming increasingly prominent, and traditional parking lots can no longer fully meet the needs of smart cities. Therefore, optimizing the urban parking management information system has become an urgent task.This study will analyze the overall system, design each functional module, and use C++, QT, MySQL, OpenCV and other technologies to implement the functions of the parking lot management information system. Analyze, design and implement a parking lot management information system that includes functions such as registration and login, user management, parking reservation, license plate recognition, etc.

**Keywords:** Parking property management system, Systems analysis, System design, MySQL

**目录**

[摘要 I](#_Toc133004418)

[Abstract II](#_Toc133004419)

[引言 1](#_Toc133004420)

[一、系统分析 1](#_Toc133004421)

[（一）业务流程 1](#_Toc133004422)

[（二）数据流程分析 2](#_Toc133004423)

[二、系统设计 6](#_Toc133004424)

[（一）功能模块式设计 6](#_Toc133004425)

[（二）数据库设计 10](#_Toc133004426)

[三、系统实现 15](#_Toc133004427)

[（一）界面设计 15](#_Toc133004428)

[（二）核心代码 20](#_Toc133004429)

[结束语 24](#_Toc133004430)

[参考文献 26](#_Toc133004431)

[谢辞 27](#_Toc133004432)

[附录 28](#_Toc133004433)

# 引言

根据国家发改委城市交通中心数据显示，2020年我国停车位缺口已超过8000万个，车位分布非常不均，停车场数据缺乏可视化，找车位，停车难等问题仍然是城市交通和居民生活中的痛点。随意停车不仅加剧城市交通拥堵与压力，而且深刻的影响城市居民生活幸福感。在智慧城市不断推动的背景下，路灯倒计时已经应用在地图导航APP当中，传统智慧停车场车位以专属APP融入城市专属APP当中，推动基础设备进入传统停车场。以专属APP的形式并不利于外地用户查询停车场数据，给智慧停车场提出更高的挑战。本课题将以停车场管理信息系统的分析与设计为主要研究对象，实现停车场管理信息系统的前后端实现，并通过小程序解决停车场车位查询和预约难问题。

# 一、系统分析

系统分析是指对一个系统进行全面的研究和评估，以确定其功能，性能，质量和效率等方面的优点和缺陷，并提出改进和优化建议的过程。该过程包括对现行系统的文档化研究，需求分析，设计和开发方案等步骤。本章将通过业务流程和数据流程分析两个部分来对停车场管理信息系统进行研究与评估。

## **（一）业务流程**

业务流程是由不同的人共同完成一系列活动，以实现特定的价值目标。停车场管理信息系统，需要用户，终端设备，管理员，停车场，管理信息系统多方面共同去最快的解决用户停车问题。

业务流程图(Transaction Flow Diagram, TFD)描述了管理系统内各成员之间的业务管理，作业顺序和管理信息流向的图表。通过业务流程图将能够更好的描述停车场管理信息系统的业务处理过程为后面的开发实现提供基础。

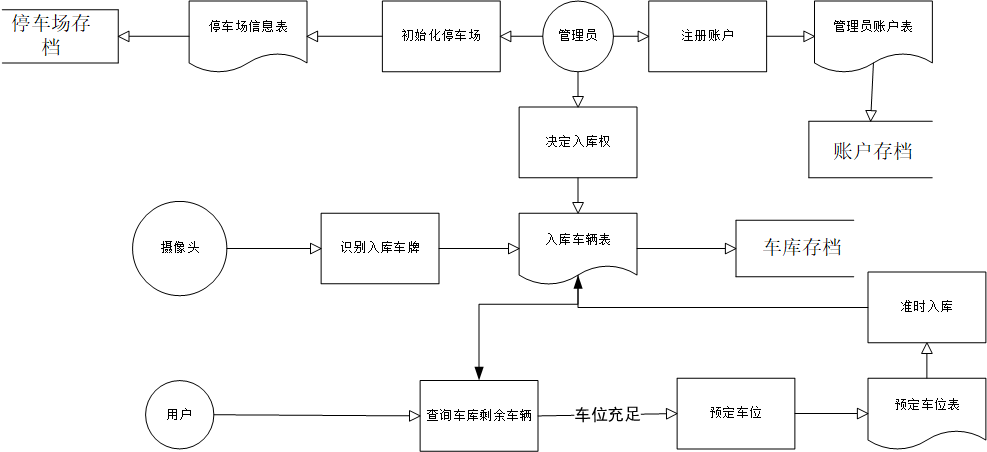
用户在初次使用时可以通过注册系统注册管理员账号，输入相关注册信息之后，系统将生成管理员账户表，并将管理员账户表以表格的形式存入数据库当中生成账户存档。当管理员登录系统时，首先要进行数据库信息初始化，添加相关停车场信息，并生成停车场信息表，并将相关信息生成停车场存档。当用户开车入库的过程中，摄像头作为一个主体将为停车场管理信息系统提供车牌信息，将车牌输入到入库车辆表之后，管理员将决定车牌能否正确入库，并生成数据库表格，存入数据库车库存档当中。为满足用户的预定需求，通过停车场管理信息系统生成的入库车辆表生成对应的车位剩余数量，反馈给用户进行查询，当剩余车位充足的前提下即可进行预定车位并生成预定车位表，当用户准时到达车库时，并将用户车牌信息传入入库车辆表，根据管理员最后决定车辆信息并入库生成对应的车库存档。若用户并没有准时入库，相应的预定信息将在一定时间内被注销。而当用户出库时，也将通过摄像头识别用户车牌信息，对照入库车辆表，根据前后对应的时间，计算出用户需要支付的价格，最终完成整个停车场管理信息系统的业务流程。停车场管理信息系统业务流程图如图1所示。

图1 停车场管理信息系统数据流程图

## （二）数据流程分析

为了准确的将整个停车场管理信息系统的数据流动清晰的表示出来，本节将采用数据流程图的方式进行分析整个停车场管理信息系统。

数据流程图(Data Flow Diagram , DFD/Data Flow chart)简称数据流图，是描述整个系统数据流程的主要方式，通过外部实体，数据存储，数据流和数据加工的形式来综合的反映信息在系统中的流动。

1.停车场管理信息系统数据流程图标总体设计

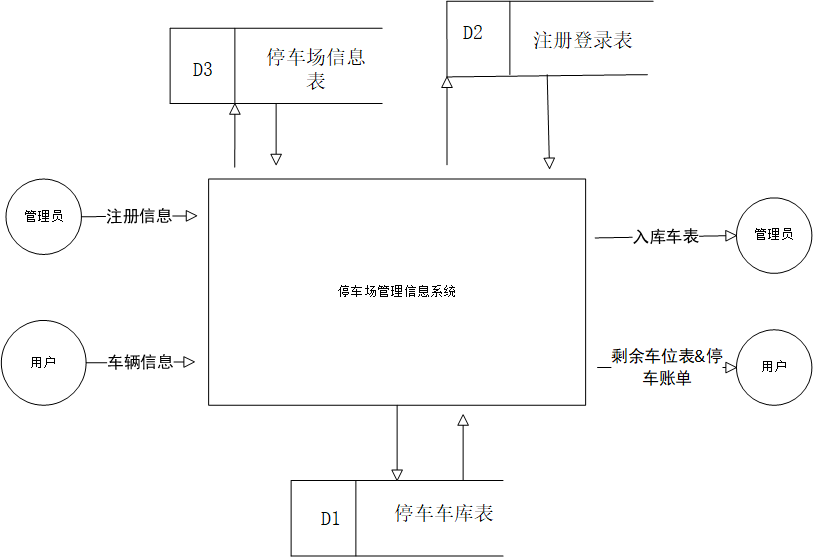
停车场管理信息系统总数据流程图如图2所示。若要登录停车场管理信息系统，首先需要登录管理员账号信息，用户可以将注册信息输入到停车场管理信息系统，注册系统将生成注册登录表存储到D2数据库当中。在初始化数据库的过程中，将停车场信息初始化并存储到D3数据库停车场信息表当中。当用户开车入库的时候，停车场管理信息系统会识别所需要的车辆信息，生成管理员所需要的入库信息表，并根据车库总体车位情况，将剩余车位情况传送给需要查询车位剩余情况的用户，并根据用户的总体停车时间生成停车账单传递给用户。最后停车场管理信息系统将生成的停车车库表D1传入到数据库当中。

图2 停车场管理信息系统系统数据流程图

2.停车场管理信息系统第一层分解数据流程图

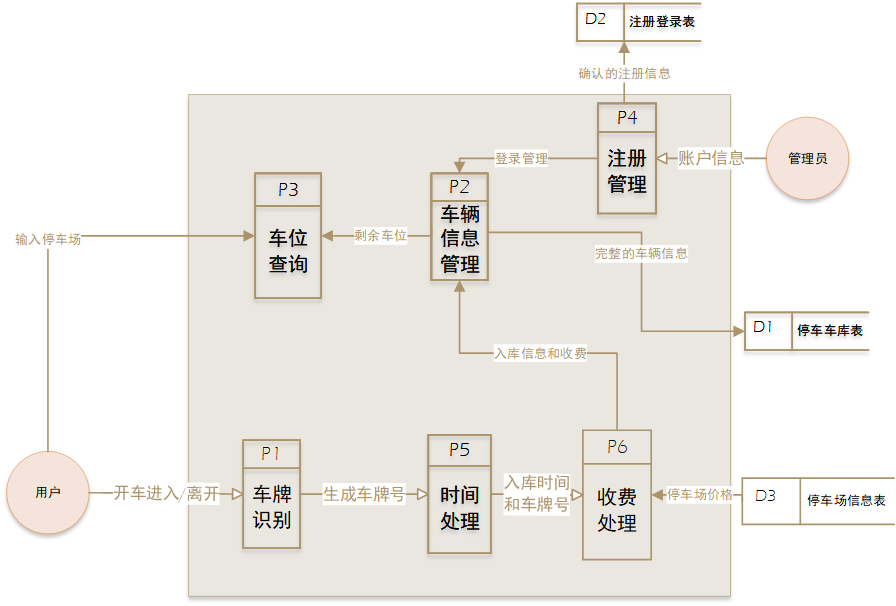
停车管理信息系统数据流程图如图3所示。该系统首先将用户输入的账户信息进行注册管理并将生成的注册信息上传到D2注册登录表当中，当用户登录成功之后才能够进行车辆信息管理审核。用户开车进入停车区域首先要进行车牌识别，然后进行时间处理P5记录当前的时间并将入库的时间和车牌号都交给收费处理P6，根据进入或离开时间完善收费信息，并将所有信息都传递给车辆信息管理进行整合生成完整的车辆信息，由管理员确认之后将车辆信息上传到停车车库表D1当中。当用户想要查询停车场剩余车位情况时，将进入车位查询系统，由车辆信息管理将剩余车位反馈给车位查询系统并最终返回到用户当中。

图3 停车管理信息系统第一层数据流程图

3.收费处理框分解

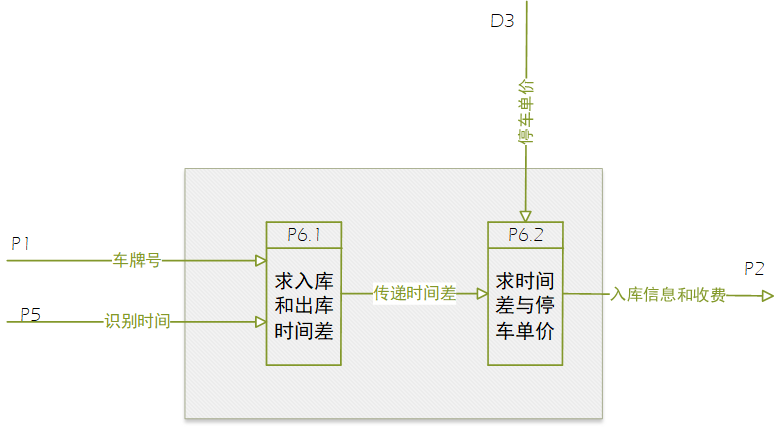
收费处理数据流程图如图4所示。收费管理首先根据用户入库识别出来的车牌号，以及系统记录的识别时间，经过P6.1计算出入库时间和出库的时间差，然后从D3数据库当中读取到停车场停车单价，通过P6.2求出停车需要支付的价格，最后将入库信息和收费情况传递给P2车辆信息管理进行处理。

图4收费处理数据流程图

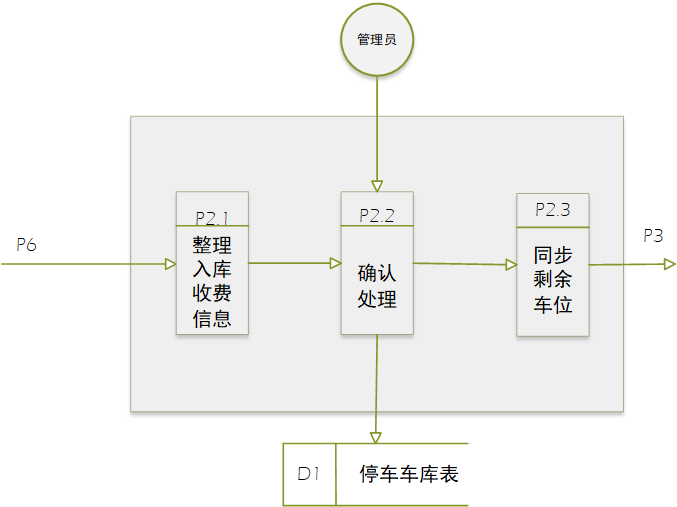
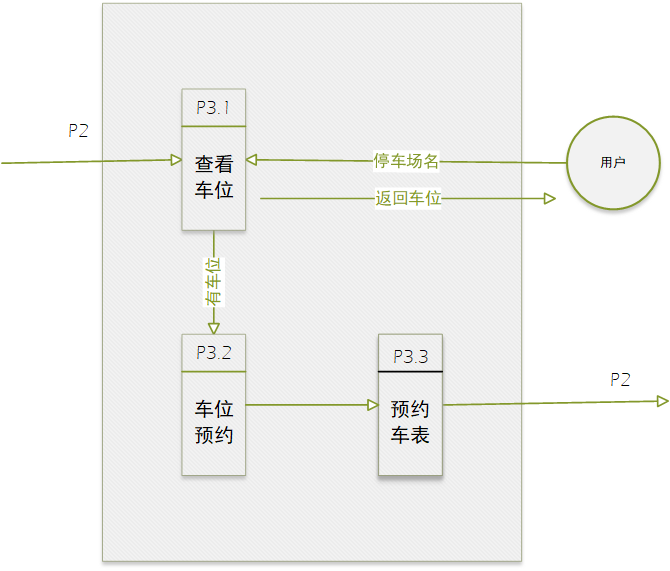
4.车辆信息管理框分解

图5 车辆信息管理数据流程图

车辆信息管理数据流程图如图5所示。车辆信息管理信息会根据P6收费处理传回的数据进行格式化整理，为形成数据库数据做准备，由管理员确认信息之后，将停车车辆信息上传到D1停车车库表当中，在上传停车车库表的同时并且同步数据库车库现有的车位情况，并将现有车位传递给P3车位查询功能模块，方便用户实时的查询车位情况。

5.车位查询框分解

车位查询数据流程图如图6所示。根据P2车辆信息管理返回的车位数量，根据用户提交的停车场名即可查看停车场的具体车位数量，然后将车位数量返回给用户，如果有车位的情况下，用户可以根据停车场名称去预定车位，并生成预约信息表，将生成的预约信息表传递给车辆信息管理并进行车辆信息处理。

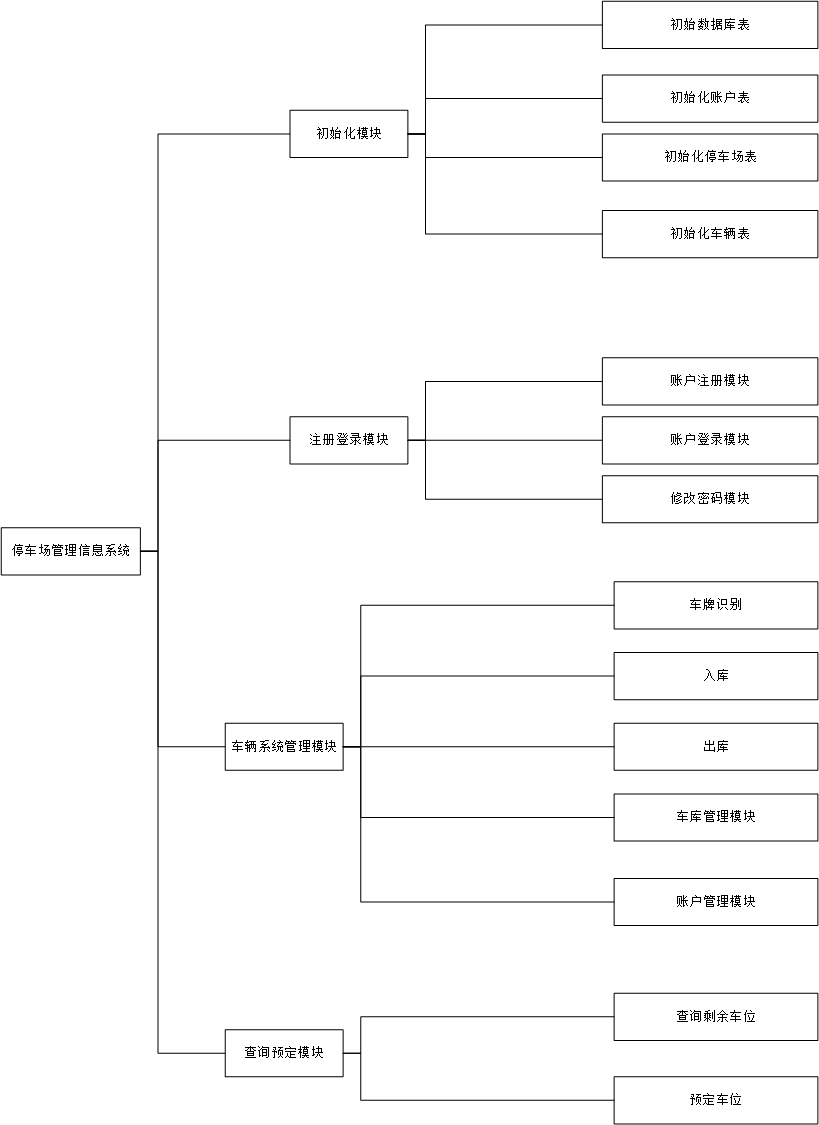
图6 车位查询数据流程图

# 二、系统设计

## （一）功能模块式设计

1.整体模块设计

根据停车场管理系统的使用场景，将停车场管理信息系统分为4个大模块 ，一个为首次登录系统所需要的初始化操作，包括数据库初始化，数据库相关表格与数据初始化。初始化之后，管理员需要登录相关账户方可进入停车场管理信息系统的主页面，即注册登录模块，以及防止管理员忘记密码所需要的修改密码模块。登录系统之后，进入车辆系统管理模块，此模块作为停车场管理信息系统的主体，车牌识别，入库出库，管理员确认识别车牌成功，并将相关车牌信息传入数据库。在基本出入库管理的基础上添加车库信息查询模块，管理员可以根据车牌号，根据时间段去查询车库停车的具体信息。于此同时管理员可以实时的查看停车场的车位状况，以及用户预约状况的可视化车库模块，还有方便管理管理员的账户管理模块。在用户未到达停车场之间，可以通过小程序实时查看停车场剩余车位的查询预定模块。停车场管理信息系统的整体功能设计图如图7。

图7 停车场管理信息系统整体模块图

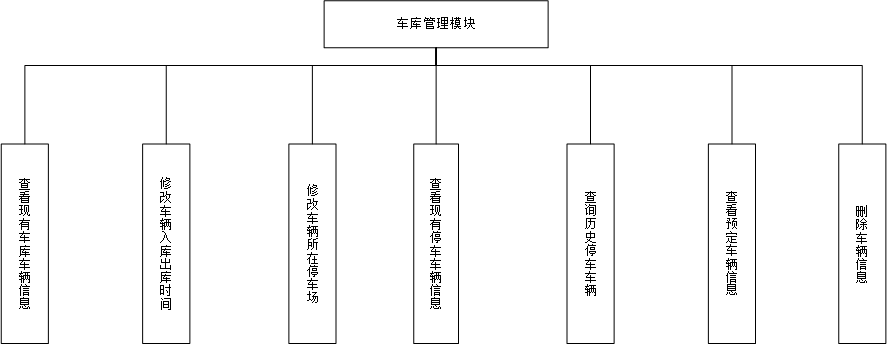
2.车库管理模块

图8车辆管理模块图

车库管理模块，管理员可以直接在停车场管理信息系统查看现有车库剩余车位以及历史停车数量，并且管理员可以根据实际情况去修改车辆所在停车场以及出入库时间等信息。并且可以根据需要删除车辆信息。通过车库管理模块实现管理员以可视化的方式对车库管理信息的全部修改操作，车辆管理模块图如图8所示。

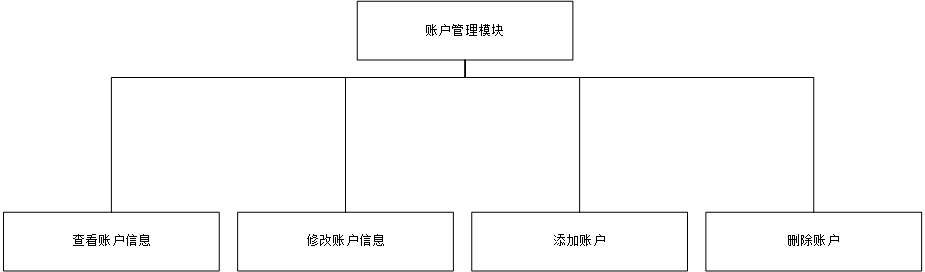
3.账户管理模块

图9账户管理模块图

账户管理模块分为遍历查看账户信息，修改账户，添加账户，删除账户等操作，通过此模块让管理员可以通过界面可视化的形式一键管理所有用户账户信息。密码账户作为加密显示，管理员并不能直接看到原密码，只能看到加密后的密码(密码不可逆)，保护其它管理员的用户隐私。账户管理模块图如图9所示。

4.出库入库描述图

车辆入库和出库作为停车场管理信息系统的重要组成成分，本节将站在用户的角度用流程图描述车辆入场和出场过程，以更加清晰的描述当车辆出入库时后台系统的运作过程。

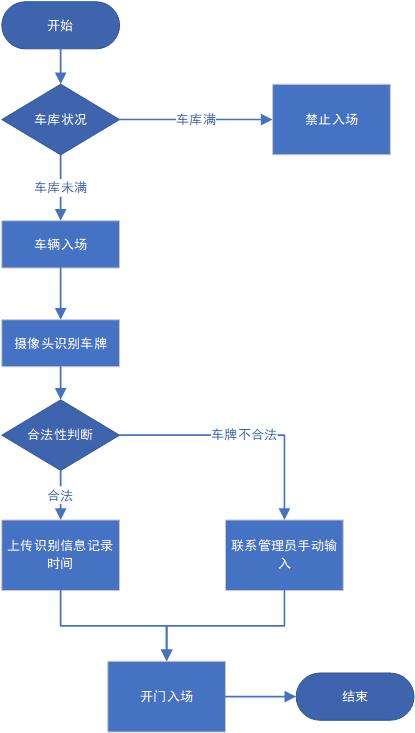
（1）入库流程描述

图10车辆入库流程图

停车场管理信息系统首先将检查停车场车库状况，若车库已满，则禁止车辆入场。当车辆未满时将运行车辆入场，使用摄像头识别车牌作为其唯一标识，并对车牌合法性进行判断，如果车牌信息合法则将车牌和对应的时间点上传到数据库。如果车牌不合法则通知管理员进行手动输入处理，处理完成之后，则开门允许车辆入场。车辆入库流程图如图10所示。

（2）车辆出场流程描述

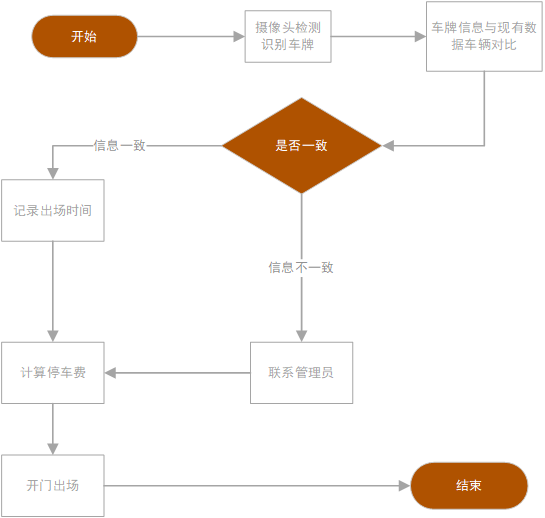
当出场的车辆进入出库区域时，摄像头将自动检查并识别车辆车牌信息，并将车辆车牌信息与现有数据车辆进行对比，如信息一致将记录出场时间并通过与入场时间的差进行计算用户需要支付的费用，如信息不一致则通知管理员进行相关操作。车辆出库流程图如图11所示。

图11车辆出库流程图

## （二）数据库设计

数据库设计(Database Design)是指根据实际的开发需求利用计算机软件进行数据库的设计与构建从而形成完整度较高的数据库管理信息系统。优秀的数据库设计具有如下特点:

冗余数据少

良好的日志信息便于分析与处理

兼顾数据库日后的修改，提高数据扩展性高

内存分配合理

1.需求分析

在管理员登录页，设计账户表，包含账户，密码，姓名，手机等相关信息。在停车场管理信息系统运行的过程中，主要包括用户账户密码的对应确认，以及用户表的注册，修改等操作。在车辆入场时需要将用户的车辆信息进行记录，并且要记录其入库的时间，等到其出库时还要记录出库时间，根据其出库时间与入库时间的差来计算用户停车费用。用户车辆信息表主要涉及创建，查询和添加等功能。而设计的停车场价格，就要根据不同的停车场具体而设定，所以需要设置一个停车场表，记录停车名称以及停车价格，为了方便用户通过小程序查询停车场车位状况，并且方便用户预约，需要再设置专属的预约表，当用户停车场满时，判断其车牌号是否在预约表里，如果在则入场，如果预约但30分钟内仍未入场，将调动触发器自动删除用户预约记录，当用户在预约表删除添加时会根据触发器更新停车场表内预约停车位数量，前端QT和小程序也可以根据此实时的反映停车场车位使用状况。

2.概念设计

E-R图(Entity-relationship model)为实体联系图，向停车场管理信息系统提供了表达表示实体，属性以及实体之间相互联系的方法，是用来描述现实世界的概念模型。根据数据库需求分析得出ER图如图12所示:

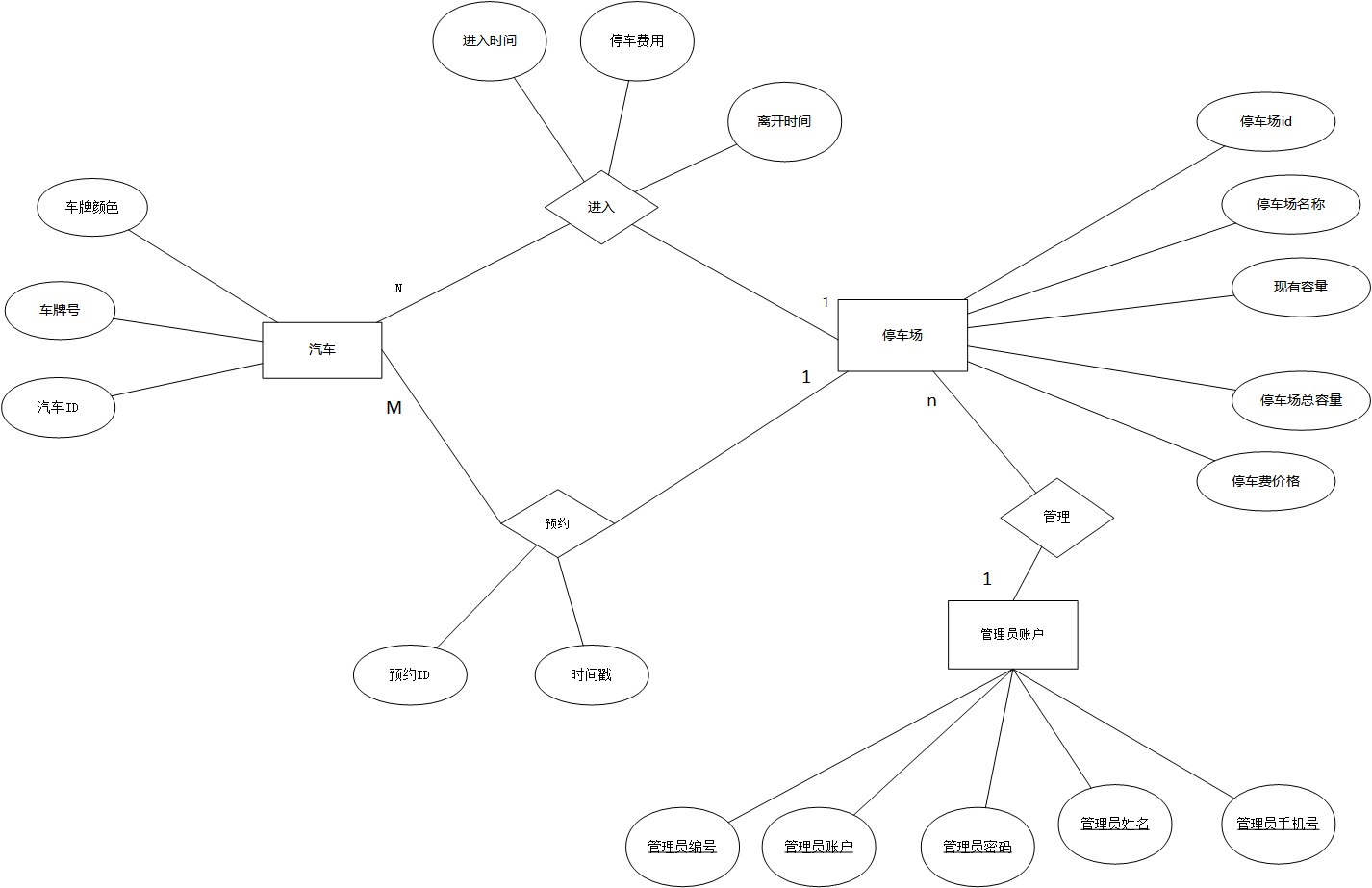


图12停车场管理信息系统ER图

汽车拥有汽车ID和车牌号和车牌颜色三个属性，汽车与停车场形成一比一的关系，在同一时间段内一个汽车只能停在一个停车场内，但是一个停车场却可以停n辆车。停车场的属性包括停车场ID，车库简称，停车场现有容量，停车场总容量，停车费费用。于此同时的管理员账户，一个管理员可以同时管理多个停车场，而管理员拥有管理员编号，管理员账户，管理员密码，管理员姓名，管理员手机号多个属性，其共同构成了停车场管理信息系统的概念设计ER图。

3.逻辑设计

将概念设计对应的ER图转换为关系模型如下:

管理员表(管理员ID，账户，密码，姓名，手机号) 主键:管理员ID

汽车表(汽车编号，车牌号，车牌颜色) 主键:汽车编号

停车场表(停车场编号，停车场名称，现有容量，停车场走过容量，停车费价格)主键：停车场编号

车辆进出记录表(汽车编号，停车场编号，入场时间，出场时间，停车费用)主键:汽车编号，停车场编号

预约车库表(预约编号,车牌号,停车场，时间戳) 主键:预约编号

4.物理设计

停车场管理信息系统后端数据存储采用中国移动云服务器CenOS6.7操作系统，宝塔面板PHP对数据库进行后端操作。数据库物理设计将采用MySQL数据库，在处理大规模车辆入库信息与检索时，MySQL能够轻松地扩展以处理更大的数据量和更高的负载。并且MySQL数据库支持多种索引技术，对于车牌号，管理员账户等经常需要访问的数据，使用其对应的索引技术拥有将会有更好的性能。

5.数据表设计

根据整个停车场管理信息系统对于停车信息管理的需求以及ER图，设计数据表如下:

（1）车位预约表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 【车位预约表表】表名：user | | | | | |
| 字段名 | 描述 | 数据类型 | 数据长度 | NULL | Key |
| id | 预约编号 | INT |  | N | PRI |
| license\_plate | 车牌号 | VARCHAR | 10 | N | UNI |
| P\_name | 停车场名称 | VARCHAR | 255 | N | N |
| created\_at | 记录时间 | datetime |  | Y | N |

（2）管理员信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 【管理员信息表】表名：user | | | | | |
| 字段名 | 描述 | 数据类型 | 数据长度 | NULL | Key |
| id | 账户编号 | INT |  | N | PRI |
| username | 账户用户名 | VARCHAR | 255 | N | UNI |
| password | 账户密码 | VARCHAR | 255 | N | N |
| telephone | 账户手机号 | INT | 11 | N | N |
| truename | 账户真实姓名 | VARCHAR | 255 | N | N |

（3）汽车表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 【停车表】表名：car | | | | | |
| 字段名 | 描述 | 数据类型 | 数据长度 | NULL | Key |
| id | 车编号 | INT |  | N | PRI |
| license\_plate | 车牌号 | VARCHAR | 20 | N | MUL |
| color | 车牌颜色 | VARCHAR | 10 | N |  |

（4）汽车停车表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 【停车信息表】表名：car\_park | | | | | |
| 字段名 | 描述 | 数据类型 | 数据长度 | NULL | Key |
| id | 车编号 | INT |  | N | PRI |
| license\_plate | 车牌号 | VARCHAR | 20 | N | MUL |
| check\_in\_time | 入库时间 | INT |  | N |  |
| check\_out\_time | 出库时间 | INT |  | Y |  |
| fee | 停车费用 | decimal | 10,2 | Y |  |
| location | 停车场 | VARCHAR | 20 | N |  |

（5）停车场数据表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 【停车场信息表】表名：parking | | | | | |
| 字段名 | 描述 | 数据类型 | 数据长度 | NULL | Key |
| P\_id | 停车场编号 | INT |  | N | PRI |
| P\_name | 停车场名称 | VARCHAR | 255 | N | UNI |
| P\_reserve\_count | 停车场预约量 | INT |  | Y |  |
| P\_now\_count | 停车场现有量 | INT |  | Y |  |
| P\_all\_count | 停车场总量 | INT |  | N |  |
| P\_fee | 停车场单价 | decimal | 10,2 | N |  |

6.实现

由于在实现过程中，车辆进出记录的经常与汽车账户一起使用，则在实现的过程中，不再将汽车账户与车辆进出记录分开设计，将汽车账户与车辆进出记录进行合并，共计四个表格，管理员账户表，停车场账户表，车辆信息表，车位预约表以及辅助的触发器。数据库建表详见附录，辅助触发器实现代码如下：

-- 添加事件调度器，当预约车辆30分钟自动清理  
SET GLOBAL event\_scheduler = ON;  
-- 创建清理任务，每隔一分钟检测是否时间大于30  
CREATE EVENT IF NOT EXISTS clean\_reservations  
ON SCHEDULE EVERY 1 MINUTE  
DO  
 DELETE FROM reservations WHERE TIMESTAMPDIFF(MINUTE, created\_at, NOW()) > 30;  
-- 触发器，当插入到预约表时,停车场表预约总数+1  
DELIMITER //  
CREATE TRIGGER update\_reserve\_count AFTER INSERT ON reservations  
FOR EACH ROW   
BEGIN   
 UPDATE PARKING SET P\_reserve\_count = P\_reserve\_count + 1 WHERE P\_name = "新工停车场";   
END//  
DELIMITER ;  
-- 触发器，当删除预约表数据时,停车场表预约总数-1  
DELIMITER //  
CREATE TRIGGER IF NOT EXISTS update\_reserve\_count2 AFTER DELETE ON reservations  
FOR EACH ROW   
BEGIN  
 UPDATE PARKING SET P\_reserve\_count = P\_reserve\_count -1 where P\_name = "新工停车场";  
END//  
DELIMITER ;

# 三、系统实现

由于停车场管理信息系统对于摄像头等硬件设备的依赖，以及对底层多线程，高并发的要求。本停车场管理信息系统实现端将基于window操作系统，采用C++11语言，使用QT5.12.1桌面开发框架，以QCSS对桌面进行美化处理，使用OpenCV3.4.5和EasyPR1.6车牌识别开源库对车牌进行识别，MySQL数据库存储数据。同时使用微信小程序作为用户端连接数据库查询剩余 停车场车位。

## （一）界面设计

界面作为信息传递和展示的平台，界面的设计与色彩图片视频的使用都在整个用户体验上发挥着举足轻重的作用。停车场管理信息系统，即有面向管理人员的停车场管理信息系统以及面向用户的停车场查询订阅系统。无论是管理页面还是用户端都要突出界面设计的易用性，便捷性，高效性，以及让用户眼前一亮的创新性。色彩的合理运用不仅提高整体视觉效果，也将给用户带来更好的用户体验。本系统主要采用QT Designer和QCSS来优化前端界面设计，以绿色为主色调。

1.数据库初始化

数据库初始化页面如图13所示。当管理员首次登录停车场管理信息系统时，系统会自动判断是否存在所需要的停车场JSON数据以及所在停车场的基本信息，如果存在则会直接进入登录页面，如果不存在，停车场管理信息系统会调用数据库初始化页面要求管理员输入相关IP，端口号，数据库名，账户，密码，停车场名，停车场价格，停车场容量信息。等到用户提交之后，会自动将用户提交的数据以JSON格式保存到本地根目录下。

图13数据库初始化页面

2.登录注册页面

登录注册页面如图14，15所示。登录和注册页面均采用左图片右内容的布局形式，改变单一的登录页面形式，使用图片的形式去表达一种意境，美化UI设计，使界面显得不那么单调。用户通过输入账号和密码即可进入主页，若没有账户密码可以点击注册按钮，并输入相关信息即可成功注册账户。使用QT的connect连接信号与槽，当用户在输入密码的时候，密码输入完毕用户会点击确定，而确认键将和登录键连接到一起，起到便捷登录的作用。

图14数据库初始化页面

图15数据库初始化页面

3.主页面

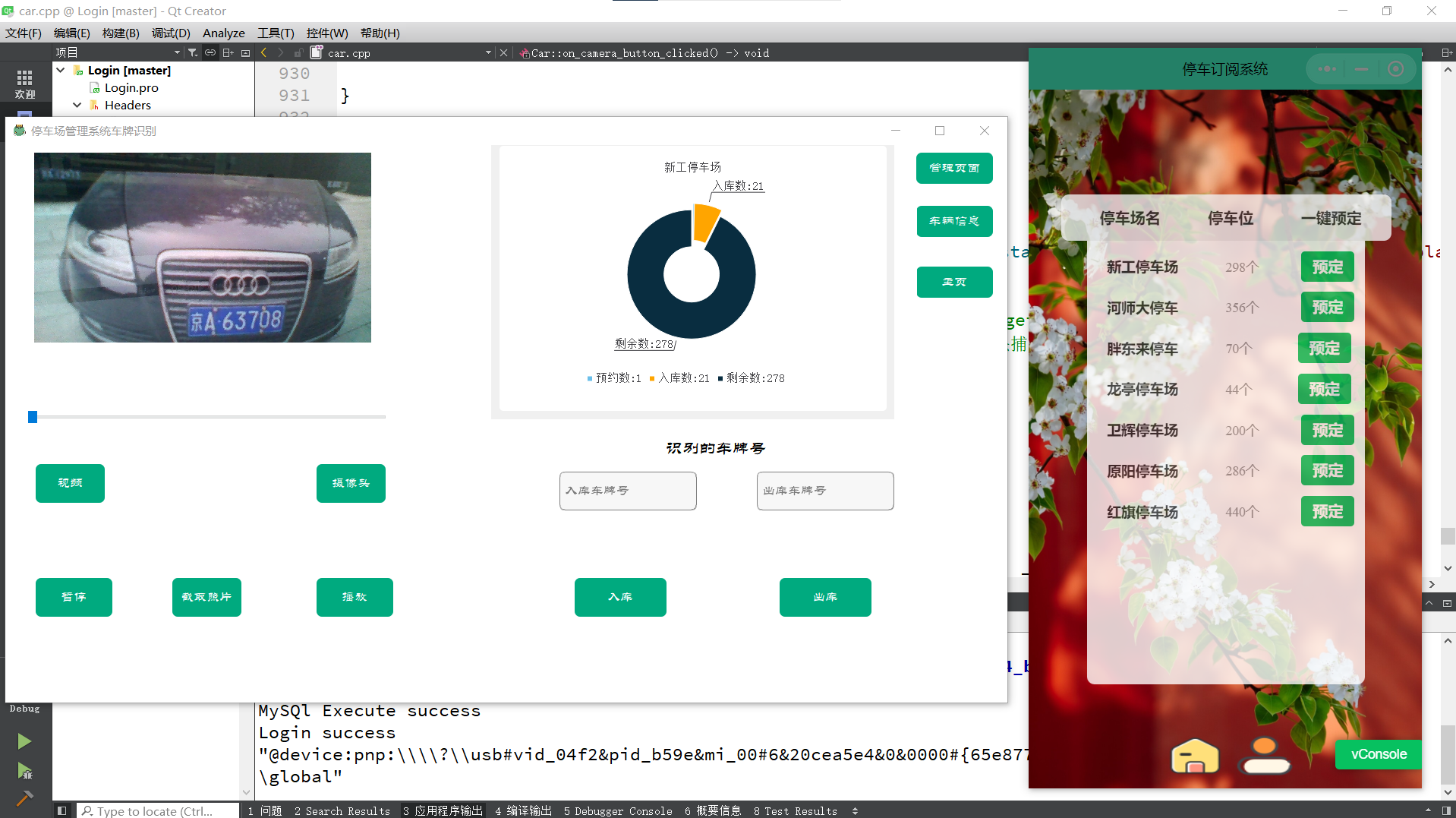
管理员在确认车牌号入库或者出库时会自动检测用户输入的车牌号是否符合规定(即车牌号要满足7位，首字母省份，第二个字母为市缩写)，若不符合规定将禁止入库。当车辆在停车场内部仍未出库的车牌禁止入库。主页将摄像头数据以视频的形式将车牌信息呈现到主页左上角。并根据识别的车牌决定入库与出库。在主页的上方将以饼图可视化的形式呈现出停车场车位使用情况。而右边是功能选项，可以选择车辆信息和管理页面。车库信息以饼图可视化的形式呈现在主页，当用户在停车订阅系统小程序，点击预定，即可在特定的停车场实时呈现出预约数量。停车场管理信息系统主页面如图16所示。

图16停车场管理信息系统主页面

4.账户管理页面

账户管理页面管理员可以直接在管理员信息表格当中修改用户数据，并点击修改按钮即可修改相关用户具体信息，而在最下面 的空行，管理员可以直接输入对应信息，点击添加新用户之后即可将添加的用户信息上传到停车场管理信息系统的管理员数据表当中。而当用户用鼠标点击某一行即可将某一行的数据进行删除，在管理员后端以最大的便捷性方便管理员管理账户信息。同时对用户输入的手机号进行限制，当添加或修改的手机号大于11位时将报错，无法执行。由于密码涉及用户隐私，仍将MD5加密呈现。账户管理页面如图17所示。

图17账户管理页面

5.停车信息管理页面

图18车库管理页面

当管理员进入车辆信息页面，将自动将所有入库车辆信息全部呈现在表格内部，以入库时间为次序进行排序，管理可以直接下滑查看数据库中所有入库信息，入库信息将包括ID，车牌号，入库时间，出库时间，费用(如果停车时间在半小时以内将不收费)，停车地点，车费单价信息。在车辆管理信息页面的上面，有车牌号和时间段，管理员可以根据时间直接查询在特定时间段内所有车辆信息。于此同时管理员也可以根据车牌号查询特定用户的出入库信息情况。当管理员想要删除某车牌的出入库信息时，可以点击对应的车牌号，然后点击删除键即可删除相关车辆的出入库信息。为了方便新管理员了解相关操作，当管理将鼠标悬浮在删除键时，将会提示如何删除数据的操作。由于出入库的数据涉及到计费，而且出入库时间和系统对应，所以车辆信息管理页面将不允许管理员对数据进行修改。车库管理页面如图18所示。

6.小程序查询预约设计

用户可以通过扫码或搜索的形式打开小程序，即可看到停车场的总停车数量以及现有的停车位数。用户可自行决定是否预定，若剩余停车位数量较多，用户无需进行任何操作，直接去停车场即可，若停车位紧张，用户可以预定停车位，但前提是用户必须在30分钟内到达停车场并入场，否则将自动放弃此停车位。小程序UI下标将替换微信小程序官方Taber，以图片背景透明的形式呈现在底部，添加简洁和美感。用户初次进入即可进行查询。但预约的前提是必须在我的页面输入自己的车牌号，车牌号输入之后将自动存储到微信小程序本地cookie文件里，用户再此启动应用将自动调用。首页通过设置透明效果，即可看到背景，也能在整体上避免大片白的美感。小程序预约页面如图19所示。

图19小程序预约页面

## （二）核心代码

//简单展示现有的车位图   
void Car::park\_num()   
{   
//qDebug()<<"num进入"<<mysql\_C.reserve;   
//根据本地停车场名称获取对应数据库停车场相关信息赋值到本地   
QString park\_name = mysql\_C.Parking\_name;   
   
 q.prepare("SELECT P\_now\_count,P\_all\_count,P\_reserve\_count FROM parking WHERE P\_name = :park\_name;");   
 q.bindValue(":park\_name", park\_name);   
 q.exec();   
 q.next();   
 QString now\_count = q.value(0).toString();   
 QString all\_count = q.value(1).toString();   
 QString reserve\_count = q.value(2).toString();   
 //将停车场数据呈现到图表中   
 mysql mysql\_instance;   
 mysql\_instance.reserve = reserve\_count.toInt();   
 mysql\_instance.parking\_now\_count = q.value(0).toInt();   
 //创建饼图   
 // 创建一个QPieSeries对象并添加数据//为每个分块设置颜色   
 series = new QPieSeries();   
 series->append("预约数", q.value(2).toInt());   
 series->append("入库数", q.value(0).toInt())->setColor("#FFA500");   
 series->append("剩余数", mysql\_instance.parking\_count-q.value(2).toInt()-q.value(0).toInt());   
 series->setHoleSize(0.3); //设置中间 空洞大小   
 series->pieSize();   
 //为每个分块设置标签文字   
 for(int i = 0;i<=2;i++)   
 {   
 slice = series->slices().at(i); //获取分块   
   
 slice->setLabel(slice->label()+QString::asprintf(":%.0f",slice->value()));   
 if(slice->value()<10)   
 {   
 slice->setLabelVisible(false);   
 }   
 else {   
 slice->setLabelVisible(true);   
   
 }   
 connect(slice,SIGNAL(hovered(bool)),this,SLOT(on\_PieSliceHighlight(bool)));   
   
 }   
 slice->setExploded(true); //最后一个设置为exploded,设置分裂效果   
 slice->setExplodeDistanceFactor(0.1);   
 // 创建一个QChart对象，并将QPieSeries对象添加到图表中   
 QChart \*chart = new QChart();   
 chart->addSeries(series);   
 chart->setTitle(park\_name);   
 chart->legend()->setVisible(true);//图例   
 chart->legend()->setAlignment(Qt::AlignBottom);   
 chart->setBackgroundBrush(QBrush(QColor(255, 255, 255))); //设置背景为白色   
 // 创建一个QChartView对象并设置图表   
 QChartView \*chartView = new QChartView(chart);   
 chartView->setRenderHint(QPainter::Antialiasing);   
 chartView->resize(532,371);   
 chartView->backgroundBrush();   
   
 // 将QChartView对象转换为QPixmap对象   
 QPixmap pixmap = chartView->grab();   
   
 // 将QPixmap对象显示到QLabel中   
 ui->label\_pie->setPixmap(pixmap);   
   
}   
//点击摄像头后，摄像头开始工作   
void Car::on\_camera\_button\_clicked()   
{   
   
 if(player->state() == QMediaPlayer::PlayingState||player->state() == QMediaPlayer::StoppedState||player->state() == QMediaPlayer::PausedState)   
 {   
 player->stop(); //停止播放文件   
 videowidget->setVisible(false); //隐藏文件播放的QVideoWidget   
 camera->setViewfinder(viemfinder); //将显示区域设置为摄像头捕获的QCameraViewfinder   
 }   
 viemfinder->setVisible(true); //显示摄像头显示区域   
 camera->start(); //让摄像头开始工作   
}   
 //当摄像头成功捕获一张图片后进行车牌识别   
 connect(imageCapture, &QCameraImageCapture::imageCaptured, this, [=](int /\*id\*/, const QImage& img)   
 {   
 Mat src;   
 Mat rgbImg;   
 //src = Mat(img.height(), img.width(), CV\_8UC4, (uchar\*)img.bits(), img.bytesPerLine()).clone();   
 src = Mat(img.height(), img.width(), CV\_8UC4, const\_cast<unsigned char\*>(img.bits()), static\_cast<size\_t>(img.bytesPerLine())).clone();   
   
 cvtColor(src, rgbImg, CV\_BGRA2RGB); //将OpenCV中颜色通道排列方式与Qt中不同的图片格式转换为Qt中正常的图片格式   
 //QImage qImg(rgbImg.data, rgbImg.cols, rgbImg.rows, rgbImg.step, QImage::Format\_RGB888);   
 QImage qImg(rgbImg.data, rgbImg.cols, rgbImg.rows, static\_cast<int>(rgbImg.step), QImage::Format\_RGB888);   
   
 //缩放图片到指定大小   
 QSize newSize(200,100);   
 QImage scaledImg = qImg.scaled(newSize,Qt::KeepAspectRatio);   
   
 //使用easyPR获取车牌信息   
 std::vector<easypr::CPlate> plates;   
 m\_plateRecognize.plateRecognize(rgbImg,plates,0);   
 for (auto plate : plates) {   
 std::cout << "plate: " << plate.getPlateStr() << std::endl;   
 }   
 qDebug()<<"内部输出";   
 if(plates.size()>0)   
 {   
 //显示车牌图片和号码   
 easypr::CPlate plate = plates[0];   
 Mat plateImg = plate.getPlateMat();   
 QImage qPlateImg(plateImg.data,plateImg.cols,plateImg.rows,static\_cast<int>(plateImg.step), QImage::Format\_RGB888);   
 ui->screen\_label->setPixmap(QPixmap::fromImage(qPlateImg));   
   
 QString plateStr = QString::fromLocal8Bit(plate.getPlateStr().c\_str());   
 // 获取车牌号码的部分字符串   
 QStringList list = plateStr.split(":");   
 if (list.size() >= 2) {   
 plateStr = list[1].trimmed();   
 }   
 ui->Car\_idinput->setText(plateStr);   
 ui->Car\_output->setText(plateStr);   
 qDebug()<<plateStr;   
 std::cout << "plate: " << plate.getPlateStr() << std::endl;   
 }   
 });

# 结束语

城市化进程不断加快，汽车保有量逐年增加，而城区停车空间日益狭小，停车难仍然是一个比较棘手的问题。随着智慧城市的不断推进，我们的城市也越来越智能化，能够根据人们的需要实时分析调整相关城市管理。本文主要分析并设计了停车场管理信息系统，并实现了简单的客户端停车场管理信息系统，并通过后台服务器和小程序实现了停车场剩余车位实时查询和预约系统。相比于综合完善的并能实现智慧城市全国覆盖的智慧停车场管理信息系统仍有很多不足，结合地图或者像如今的红绿灯倒计时一样也在地图软件中实时的显示停车场车位信息，并根据地图导航实时的提前规划停车场，彻底解决找车位问题，真正实现智慧停车场管理信息系统还有一定距离。

# 参考文献

[1]王思为. 企业用户停车服务管理信息系统设计与分析[D].南京邮电大学,2015.

[2]路漫漫.基于交互优化的城市商业智能停车场管理系统[J].软件,2021 (12):158-162.

[3] Yao R , Yang J , Liu M . Cloud Intelligent Parking Management System Based on Internet of Things Technology[J]. Journal of Physics: Conference Series, 2021 (4):042025-.

[4]新华社. (2023年1月11日). 我国新能源汽车保有量达1310万辆 呈高速增长态势. [新闻稿]. 国务院网站. <http://www.gov.cn/xinwen/2023-01/11/content_5736281.htm>

[5]叶允英. 福安市停车诱导信息系统设计[D].福州大学,2014.

[6] 王维波, 栗宝鹃, 侯春望. (2018). Qt 5.9 C++开发指南. 北京: 人民邮电出版社.

[7]计算机的潜意识. (2014, 9月 30日). EasyPR--中文开源车牌识别系统 [Blog post series]. 博客园. <https://www.cnblogs.com/subconscious/p/3979988.html>

[8]吴春辉.基于物联网的城市级智慧停车综合解决方案研究[J].时代汽车,2022(24):160-162.

[9]丁浩洪.动态链接库技术的应用探讨[J].信息系统工程,2019 (01):40.

[10]丁姝萌,辜林.基于分布式设备及智慧决策的停车场管理系统[J].软件,2021 (06):13-15.

[11]戴娴子,杨棚瑞.基于认知心理学探析APP界面设计中的色彩[J].湖南包装,2022 (06):93-95+143.

[12]徐立艳.计算机软件数据库设计的原则及问题研究[J].软件,2023 (01):141-143.

[13]尚兆功. 基于视频识别技术的智能车库管理系统[D].山东建筑大学,2016.

[14]徐成. 基于智能终端的免停自动支付停车系统的设计与实现[D].北京邮电大学,2017.

[15]喻正佳. 城市停车位智能服务平台的设计与实现[D].吉林大学,2021.

# 谢辞

虽然前方拥堵，当是您仍在最优的路线上。在完成研究生考试后开始学习C++，学习STL，QT即为了就业做准备也是为了毕业论文停车场管理信息系统的分析与设计。感谢这一路上身边每一个人的支持与陪伴，同时也非常感谢自己能够一直坚持下去，一直写下去，勇于放弃该放弃的勇气。也非常感谢老师在论文中的指导以及给与充足的时间去准备。感谢身边的每一个人还有爸妈对就业的宽恕，对我的包容，感谢这个时代，这个时代在进步，也给了这个时代的我们新的任务。好好加油，我也一定会坚信道路，勇敢的走下去。

# 附录

-- 创建停车信息管理信息系统数据库  
CREATE DATABASE car;  
-- 创建管理表  
CREATE TABLE  
IF  
 NOT EXISTS USER (  
 id INT PRIMARY KEY AUTO\_INCREMENT,  
 username VARCHAR ( 255 ) UNIQUE NOT NULL,  
 PASSWORD VARCHAR ( 255 ) NOT NULL,  
 telephone VARCHAR ( 11 ) NOT NULL,  
 truename VARCHAR ( 255 ) NOT NULL   
 );  
-- 创建停车场信息表  
CREATE TABLE  
IF  
 NOT EXISTS PARKING (  
 P\_id INT NOT NULL AUTO\_INCREMENT,  
 P\_name VARCHAR ( 255 ) NOT NULL,  
 P\_reserve\_count INT,  
 P\_now\_count INT,  
 P\_all\_count INT,  
 P\_fee DECIMAL ( 10, 2 ),  
 PRIMARY KEY ( P\_id )  
 );  
-- 创建汽车表  
CREATE TABLE  
IF  
 NOT EXISTS CAR (  
 id INT NOT NULL AUTO\_INCREMENT,  
 license\_plate VARCHAR ( 20 ) NOT NULL,  
 check\_in\_time DATETIME NOT NULL,  
 check\_out\_time DATETIME DEFAULT NULL,  
 fee DECIMAL ( 10, 2 ) DEFAULT NULL,  
 location VARCHAR ( 20 ) NOT NULL,  
 PRIMARY KEY ( id ),  
 UNIQUE INDEX ( license\_plate )   
 );  
-- 预约表  
CREATE TABLE IF NOT EXISTS reservations   
 (   
 id INT PRIMARY KEY AUTO\_INCREMENT,   
 license\_plate VARCHAR ( 10 ) NOT NULL,   
 P\_name VARCHAR ( 255 ) NOT NULL,  
 created\_at DATETIME DEFAULT CURRENT\_TIMESTAMP,  
 UNIQUE INDEX(license\_plate)  
 );