|  |  |
| --- | --- |
|  |  |
| **学 号：** |  |

****

|  |  |
| --- | --- |
| **题 目** | 小区停车管理数据库综合实验 |
| **学 院** | 计算机科学与技术学院 |
| **专 业** | 计算机科学与技术 |
| **班 级** |  |
| **姓 名** |  |
| **指导教师** | 施继珉 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2019 | 年 | 12 | 月 | 25 | 日 |

**原创性声明**

所呈交的报告是我在老师指导下进行的研究工作及取得的研究成果。除了特别加以标注和致谢的地方外，报告中不包含其他人已经发表或撰写过的研究成果。

小组成员及分工：\*\*\*

签名：

日期：

1需求分析

* 1. 问题提出

应用环境：传统停车场采用人工管理的办法，有一些不可避免的缺点，如效率低，安全性差。为了解决这些不足，在智能小区中采用停车场自动管理系统。

用户：生活在小区中的居民和需要临时停车的客户

必要性和意义：IT行业的高速发展让计算机技术深入日常生活的每一个细节，在各个领域中，计算机技术的应用帮助人们减少劳动量，提高工作效率，发挥着越来越重要的作用。随着城市化程度的加深，房地产行业日益兴盛，越来越多的小区散布在城市里，小区中来往的车辆与日俱增，停车场是小区里不可缺少的一个部分，对停车场的管理也是非常重要，这就使得各停车场需要更加先进、更加完善的停车场管理系统，开发一个停车场管理系统迎合了现代的公司发展趋势，对居民合理管理和他人的练习信息起到重要意义。为车主带来方便。我结合本次实验开发以下的小区停车场管理系统，使小区里的车辆能得到有序的管理。

1.2系统分析

1.2.1数据流图

顶层数据流图是为了方便开发人员与用户更好的够同，以及位系统的设计提供帮助。

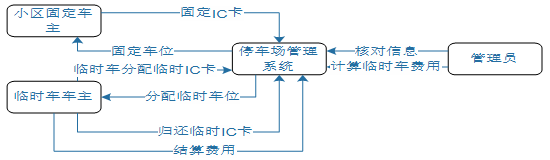


图1.1 顶层数据流图

各功能的数据流图可让用户更清晰的观察功能的实现。

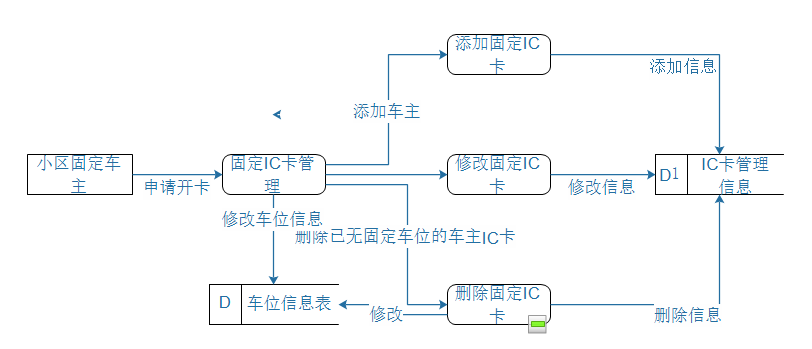


图1.2 IC卡管理数据流图

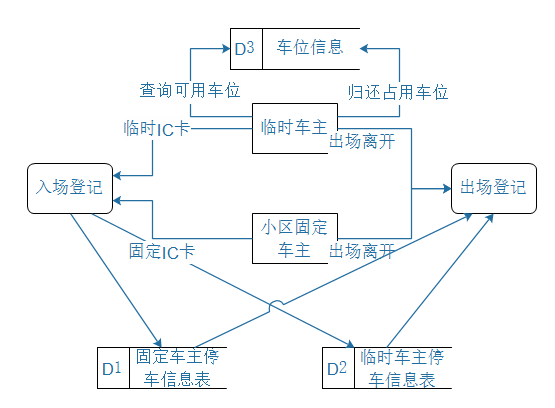


图1.3 出入停车场数据流图

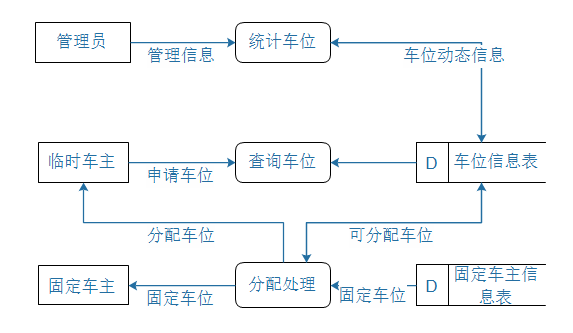


图1.4车位管理数据流图

1.2.2数据字典

（1）数据项的描述

数据项： IC卡号

含义说明： IC卡的编号

数据类型： 字符型

长度： 14

取值含义： 办卡年份+日期+6位随机数

取值范围：14位数字

与其他数据项的逻辑关系：无

数据项：停车位编号

含义说明： 给所有的停车位一个编号

数据类型： 字符型

长度： 3

取值含义： 400个停车位号中的一个

取值范围：0—400

与其他数据项的逻辑关系：与IC卡号是一一对应关系

数据项： 车主

含义说明： 车主的姓名

数据类型： 字符型

长度： 2-10

取值含义： 车主的姓名

取值范围：人的名字

与其他数据项的逻辑关系：无

数据项： 车主的住址

含义说明： 车主在小区中住在哪里

数据类型： 字符型

长度： 20以内

取值含义： 车主在小区中的具体住址

取值范围：中文字符、可见字符和数字

与其他数据项的逻辑关系：与IC卡号和车主姓名一一对应

数据项： 车牌号

含义说明： 车主的车牌号

数据类型： 字符型

长度： 7

取值含义： 车主的车牌号

取值范围：中文字符、大写英文字符和数字

与其他数据项的逻辑关系：以IC卡号和车主姓名一一对应

数据项： 停车区域

含义说明： 停车场的具体区域

数据类型： 字符型

长度： 2

取值含义： 停车区域有A区和B区

取值范围：A区、B区

与其他数据项的逻辑关系：无

数据项： 停车位标志

含义说明： 停车位的标志

数据类型： 字符型

长度： 6

取值含义： 车位是否有固定车主

取值范围：固定车主车位和空

与其他数据项的逻辑关系：停车位标志决定该车是否可停临时车

数据项： 停车位状态

含义说明： 停车位是否停了车

数据类型： int

长度： 1

取值含义： 停车位当前是否停了车

取值范围：0或1

与其他数据项的逻辑关系：停车位状态决定可使用车位数

数据项： 临时卡id

含义说明： 临时车的临时IC卡号

数据类型： 字符型

长度： 14

取值含义： 临时车的临时IC卡号

取值范围：开卡年份+日期+6位随机数

与其他数据项的逻辑关系：无

数据项： 车入场日期

含义说明： 车什么时候进入的停车场

数据类型： Date

长度： 10

取值含义： 车什么时候进入的停车场

取值范围：正常日期

与其他数据项的逻辑关系：决定临时车的停车费用

数据项： 车入场具体时间

含义说明： 车是几点几分进入的停车场

数据类型： time

长度： 8

取值含义： 车是几点几分进入的停车场

取值范围：正常的时间

与其他数据项的逻辑关系：决定临时车的停车费用

数据项： 车出场日期

含义说明： 车什么时候离开的停车场

数据类型： Date

长度： 10

取值含义： 车什么时候离开的停车场

取值范围：正常日期

与其他数据项的逻辑关系：决定临时车的停车费用

数据项： 车离开停车场具体时间

含义说明： 车是几点几分离开的停车场

数据类型： time

长度： 8

取值含义： 车是几点几分离开的停车场

取值范围：正常的时间

与其他数据项的逻辑关系：决定临时车的停车费用

数据项： 临时车费用

含义说明： 临时车离开停车场时要交的费用

数据类型：double

长度： 5

取值含义：临时车离开停车场是要交的费用

取值范围：大于0的双精度数

与其他数据项的逻辑关系：无

（2）数据结构的描述

数据结构：管理员

含义说明：定义了一位管理员的相关信息

组成： ID，角色，名字，密码，手机号

数据结构：IC卡

含义说明：是停车场管理系统的主题数据结构，是小区固定车主的相关信息

组成： 由IC卡的ID，车位号，车主名字，性别，车主住址，车主车牌号组成

数据结构：停车位

含义说明：停车场提供停车服务的必备设施，是停车场的重要组成部分

组成： 车位编号，车位状态，车位类型，车位区域

数据结构：临时车

含义说明：临时车主在停车场的相关信息

组成： 临时IC卡号，车位号，车牌号，进出停车场时间，停车费用

数据结构：小区车主

含义说明：小区车主在停车场的相关信息

组成： 固定IC卡号，进出停车场时间

（3）数据流的描述

数据流是数据结构在系统内传输的路径。对数据流的描述通常包括以下内容:数据流名、说明、数据流来源、数据流去向、组成等。

数据流编号： F1

数据流名称： 停车场车位信息表

简述： 供管理员查询停车场车位信息。

数据流来源： 小区停车场进出口闸道

数据流去向： 车主

数据流组成： 车位类型,车位编号,区域,车位状态

数据流量： 100/天

高峰流量： 300/天

数据流编号： F2

数据流名称： IC卡信息表

简述： 供管理员查询管理IC卡信息。

数据流来源： 小区停车场进出口管理处

数据流去向： 车主

数据流组成： IC卡号,车位编号，车主名字,车主性别,车主住址,车主车牌号

数据流量： 50/天

高峰流量： 150/天

数据流编号： F3

数据流名称： 小区车主停车信息表

简述： 登记小区车主进出停车场的信息

数据流来源： 小区停车场进出口管理处

数据流去向： 车主

数据流组成： IC卡号,进出停车场时间

数据流量： 50/天

高峰流量： 150/天

数据流编号： F4

数据流名称： 临时停车信息表

简述： 登记临时进出停车场的信息

数据流来源： 小区停车场进出口管理处

数据流去向： 临时车主

数据流组成： 临时IC卡号，临时车车牌号，停车位号，进出停车场时间，停车费用

数据流量： 50/天

高峰流量： 150/天

数据流编号： F5

数据流名称： 管理员信息表

简述： 记录管理员的相关信息和权限

数据流来源： 小区停车场进出口管理处

数据流去向： 管理员

数据流组成： 管理员ID，角色，名字，密码，手机号

数据流量： 2/天

高峰流量： 2/天

(4)处理过程

处理逻辑编号： P1

处理逻辑名称： 更新停车场车位信息

简述： 有停车的车主后，要及时更新车位信息

输入的数据流： 停车场车位信息表

处理描述： 对车位信息数据进行更新操作

输出的数据流： 新的车位信息

处理逻辑编号： P2

处理逻辑名称：IC卡的管理

简述： 若有小区车主要买固定车位，就添加固定车位IC卡

输入的数据流： IC卡信息表

处理描述： 对IC卡信息数据进行更新操作

输出的数据流： 新的IC卡信息

处理逻辑编号： P3

处理逻辑名称：固定车主信息管理

简述： 小区车主中由固定车位车主的相关信息管理

输入的数据流： 固定车主车位信息表

处理描述： 按车主相应信息进行停车管理，对数据表进行更新

输出的数据流： 新的固定车主车位信息表

处理逻辑编号： P4

处理逻辑名称：临时车主信息管理

简述： 临时车主的相关信息管理

输入的数据流： 临时车主车位信息表

处理描述： 按临时相应信息进行停车管理，对数据表进行更新

输出的数据流： 新的临时车主车位信息表

（5）数据存储的描述

数据存储编号： S1

数据存储名称： 停车场车位信息数据

简述： 记录车位信息

数据存储组成： 姓名,车位所在区域,车位状态,车位信息

关键字： 车位编号

输入数据流： 停车场车位信息表

输出数据流： 新的停车场车位信息表

存取频度： 300/天

数据存储编号： S2

数据存储名称： IC卡信息数据

简述： 记录IC卡信息

数据存储组成：IC卡号，车位编号，车主名字，性别，车主住址，车牌号

关键字： IC卡号

输入数据流： IC卡信息表，车位信息表

输出数据流： 新的IC卡信息表

存取频度： 150/天

数据存储编号： S3

数据存储名称： 固定车主停车信息数据

简述： 记录固定车主停车信息

数据存储组成： 固定车主IC卡号,进出停车场时间

关键字： 固定车主IC 卡号

输入数据流： 固定车主停车信息表，IC卡信息表

输出数据流： 新的固定车主停车信息表

存取频度： 150/天

数据存储编号： S4

数据存储名称： 临时车主停车信息数据

简述： 记录临时车主停车信息

数据存储组成： 临时车主IC卡号,停车位号，进出停车场时间，车牌号，停车费用

关键字： 临时车主IC 卡号

输入数据流： 临时车主停车信息表

输出数据流： 新的临时车主停车信息表

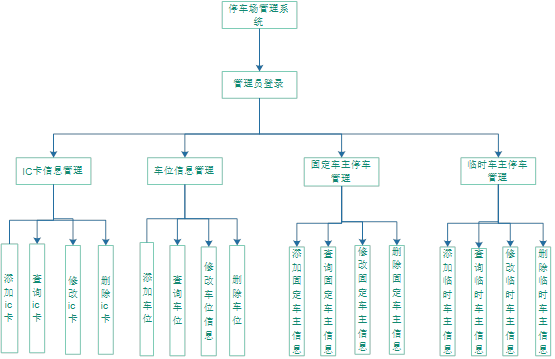
存取频度： 150/天

1.3功能需求

根据停车场管理系统的理念，此停车场管理系统必须满足以下需求：

1. IC卡信息管理：添加IC卡类型，管理IC卡类型。管理IC卡信息可以对车主的IC卡信息进行增、删、改、查。
2. 固定车主停车管理：出入场设置、停车场信息管理。出入场设置可以设用户的入场和出场，相应的在停车信息中有一个展现。停车信息管理可以对车主的停车信息进行查询和删除。
3. 临时车辆信息管理：车主入场信息，车主出场设置。车主入场在相应的表中展现，在车主出场时进行收费，临时车车主停车信息管理可以对临时车主的信息进行删除和查询。
4. 车位信息管理：管理车位、添加车位，管理车位信息模块可以对车主车位信息进行增、删、改。

功能结构图如下：



1.4非功能性需求

为了保证系统能够长期、安全、稳定、可靠、高效的运行，系统应该满足以下的非功能性需求：

1. 系统处理的准确性和及时性

系统处理的准确性和及时性是系统的必要性能。在系统设计和开发过程中，要充分考虑系统当前和将来可能承受的工作量，使系统的处理能力和响应时间能够满足用户对信息的处理。由于系统的查询功能对于整个系统的功能和性能完成很重要。从系统的多个数据来源来看，车位信息查询、IC卡信息查询、车辆信息查询，其准确性很大程度上决定了系统的成败。因此，在系统开发过程中，系统采用优化的SQL语句及安全扩展存储过程来保证系统的准确性和及时性。

1. 系统的开放性和系统的可扩充性

系统在开发过程中，应该充分考虑以后的可扩充性。例如系统权限和车位信息设置等模块也会不断的更新和完善。所有这些都要求系统提供足够的手段进行功能的调整和扩充。而要实现这一点，应通过系统的开放性来完成，既系统应是一个开放系统，只要符合一定的规范，可以简单的加入和减少系统的模块，配置系统的硬件。通过软件的修补、 替换完成系统的升级和更新换代。

1. 系统的易用性

本停车场管理系统面向的用户是小区工作人员，而有些使用人员往往对计算机并不 是非常熟悉，所以系统操作上要求简单、方便、快捷，便于用户使用。这就要求系统能 够提供良好的用户接口，易用的人机交互界面。

1.5开发与运行环境

开发工具：JDK-8、HTML、JavaScript、Eclipse集成开发环境、MySQL数据库管理系统、TomCat服务器运行环境：谷歌浏览器、火狐浏览器、IE浏览器效果最佳，其他主流浏览器也可。

1.5.1 eclipse for Java EE

Eclipse是著名的跨平台的自由集成开发环境（IDE）。最初主要用来Java语言开发，通过安装不同的插件Eclipse可以支持不同的计算机语言，比如C++和Python等开发工具。Eclipse的本身只是一个框架平台，但是众多插件的支持使得Eclipse拥有其他功能相对固定的IDE软件很难具有的灵活性。许多软件开发商以Eclipse为框架开发自己的IDE。

1.5.2 tomcat 9.0服务器

Tomcat是Apache 软件基金会（Apache Software Foundation）的Jakarta 项目中的一个核心项目，由Apache、Sun 和其他一些公司及个人共同开发而成。由于有了Sun 的参与和支持，最新的Servlet 和JSP 规范总是能在Tomcat 中得到体现。因为Tomcat 技术先进、性能稳定，而且免费，因而深受Java 爱好者的喜爱并得到了部分软件开发商的认可，成为目前比较流行的Web 应用服务器。

1.5.3 MySQL数据库管理系统

MySQL是一个关系型数据库管理系统，由瑞典MySQL AB 公司开发，属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件之一。

2数据库设计与实现

2.1概念设计

根据需求分析形成的数据字典和数据流图，抽象得到的实体有：

固定IC卡（卡号，车主姓名，车主住址，车牌号）

停车位（车位号，车位区域，车位状态，车位信息）

固定车主（进场日期，出场日期）

临时车主（卡号，车位号，车牌号，进场日期，出场日期，停车花费）

管理员（管理员id，姓名，密码，手机号）

实体之间的联系如下：

一个车主停在一个停车位。

一个固定车主有一个固定IC卡，一个临时车主有一条临时信息

每个车主都会分配一个停车位

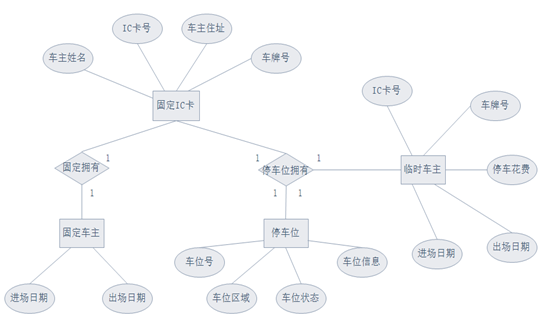


图2.1 停车场管理系统E-R图

2.2逻辑设计与物理设计

2.2.1逻辑设计

实体转换的关系模式有固定IC卡，停车位，固定车主，临时车主。

“固定拥有”联系与固定车主关系模式合并。

“停车位拥有”联系分别与固定IC卡和临时车主卡关系模式合并。

停车场管理系统的关系模式如下：

固定IC卡（卡号，车主姓名，车主住址，车牌号，停车位号）

停车位（车位号，车位区域，车位状态，车位信息）

固定车主（IC卡号，进场日期，出场日期）

临时车主（IC卡号，停车位号，车牌号，进场日期，出场日期，停车花费）

管理员（管理员id，姓名，密码，手机号）

说明：固定IC卡关系模式的外码为停车位号，被参照关系为停车位关系模式；

临时车主关系模式的外码为停车位号，被参照关系为停车位关系模式

固定车主关系模式的外码为固定IC卡号，被参照关系是固定IC卡关系模式

定义用户子模式如下：

空停车位（车位号，区域，车位信息）

2.2.1物理设计

一、在Parking数据库上，根据关系模式，定义基本表。表结构如下：

（1）车位表：存储小区车位的信息表。

表2.1 Seat（车位表）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性列名 | 属性说明 | 数据类型 | 码 | 外码 | 备注 |
| Seat\_id | 车位号 | Varchar(3) | 主码 |  |  |
| Seat\_section | 车位区域 | Varchar(2) |  |  | not null |
| Seat\_state | 车位状态 | Int(2) |  |  | 0或1（0为空） |
| Seat\_tag | 车位信息 | Varchar(50) |  |  | 反映是否为固定车位 |

（2）固定IC卡表：存储固定IC卡表。

表2.2 FixedCard（固定IC卡表）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性列名 | 属性说明 | 数据类型 | 码 | 外码 | 备注 |
| Card\_id | 卡号 | Varchar(14) | 主码 |  | 固定IC卡号 |
| Name | 车主姓名 | Varchar(10) |  |  |  |
| Address | 车主住址 | Varchar(20) |  |  |  |
| Car\_num | 车牌号 | Varchar(7) |  |  |  |
| Seat\_id | 停车位号 | Varchar(3) |  | 外码 | 参照Seat表 |

（3）临时车主表：存储临时车主表。

表2.3 TempCard（临时车主表）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性列名 | 属性说明 | 数据类型 | 码 | 外码 | 备注 |
| Card\_id | IC卡号 | Varchar(14) | 主码 | 外码 | 参照TempCard表 |
| Seat\_id | 停车位号 | Varchar(3) |  |  |  |
| Car\_nu | 车牌号 | Varchar(7) |  |  |  |
| Entry\_date | 进场日期 | Date |  |  |  |
| Entry\_time | 进场时间 | Time |  |  |  |
| Out\_date | 出场日期 | Date |  |  |  |
| Out\_time | 出场时间 | Time |  |  |  |
| Pay | 停车花费 | Double(5) |  |  |  |

（4）固定车主表：存储固定车主表。

表2.4 Fixed（固定车主表）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性列名 | 属性说明 | 数据类型 | 码 | 外码 | 备注 |
| Card\_id | IC卡号 | Varchar(14) | 主码 | 外码 | 参照FixedCard表 |
| Entry\_date | 进场日期 | Date |  |  |  |
| Entry\_time | 进场时间 | Time |  |  |  |
| Out\_date | 出场日期 | Date |  |  |  |
| Out\_time | 出场时间 | Time |  |  |  |

（5）管理员表：存储管理员信息。

表2.5 Admins（管理员表）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性列名 | 属性说明 | 数据类型 | 码 | 外码 | 备注 |
| Admin\_id | 管理员ID | Varchar(5) | 主码 |  |  |
| Name | 姓名 | Varchar(15) |  |  | 管理员的姓名 |
| Password | 密码 | Varchar(15) |  |  |  |
| Phonenum | 手机号 | Varchar(11) |  |  | 管理员的手机号 |

二、定义的视图如下：

（1）固定车主及其固定车位状态视图。

表2.6 voutfixed（视图）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性列名 | 属性说明 | 数据类型 | 码 | 外码 | 备注 |
| Card\_id | 卡号 | Varchar(14) |  |  |  |
| Name | 车主姓名 | Varchar(10) |  |  |  |
| Address | 车主住址 | Varchar(20) |  |  |  |
| Car\_num | 车牌号 | Varchar(7) |  |  |  |
| Seat\_id | 停车位号 | Varchar(3) |  |  |  |
| Seat\_state | 车位状态 | Int(2) |  |  |  |

（2）固定车主停车信息视图。

表2.7 vfixed（视图）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性列名 | 属性说明 | 数据类型 | 码 | 外码 | 备注 |
| Card\_id | IC卡号 | Varchar(14) |  |  |  |
| Entry\_date | 进场日期 | Date |  |  |  |
| Entry\_time | 进场时间 | Time |  |  |  |
| Out\_date | 出场日期 | Date |  |  |  |
| Out\_time | 出场时间 | Time |  |  |  |
| Seat\_id | 停车位号 | Int(2) |  |  |  |
| Name | 车主姓名 | Varchar(10) |  |  |  |
| Address | 地址 | Varchar(20) |  |  |  |
| Car\_num | 车牌号 | Varchar(7) |  |  |  |

三、根据处理需求，建立相关索引，如表2.8所示

(1)ALTER TABLE FixedCard ADD PRIMARY KEY(`Card\_id`)

(2)CREATE INDEX fcar\_num ON fixed(`car\_num`)

(3) CREATE INDEX tcar\_num ON tempcard(`car\_num`)

(5) ALTER TABLE Seat ADD PRIMARY KEY(`Seat\_id`)

表2.8 索引列表

|  |  |  |
| --- | --- | --- |
| 关系模式 | 索引属性列 | 索引类型 |
| 固定IC卡 | IC卡号 | 主键索引 |
| 固定车主 | 车牌号 | 普通索引 |
| 临时车主 | 车牌号 | 普通索引 |
| 停车位 | 车位号 | 主键索引 |

2.3数据库实现

定义Parking数据库：

CREATE DATABASE parking DEFAULT CHARACTER SET utf8 ;

（1）在Navicat中根据具体数据库设计建立数据库中的各个关系模式表。

（2）按照设计的数据库结构，使用Excel组织数据，将表格中的数据进行标准化处理，并对应的数据列的名称插入进表格上方第一行。

（3）在Navicat可视化操作的数据库表中找到“导入”，并在弹窗中找到对应的execl表格文件类型，点击下一步。

（4）选择需要导入的文件路径，并选择文件中的工作表，点击下一步。

（5）确定默认设置，确认目标表和源表，源栏位和目标栏位的对应关系，最后确定导入模式，若添加数据，默认选项即可实现需求，参数设置完成后，点击开始，等待文件添加完成，当出现Finished-Successfully，即导入成功。

（6）为防止测试数据违反有关约束，可先用Excel等工具产生数据库测试数据，然后利用Navicat软件导入这些数据。

3应用程序设计

3.1界面设计（部分）



图3.1车位信息管理界面



图3.2固定车主添加界面

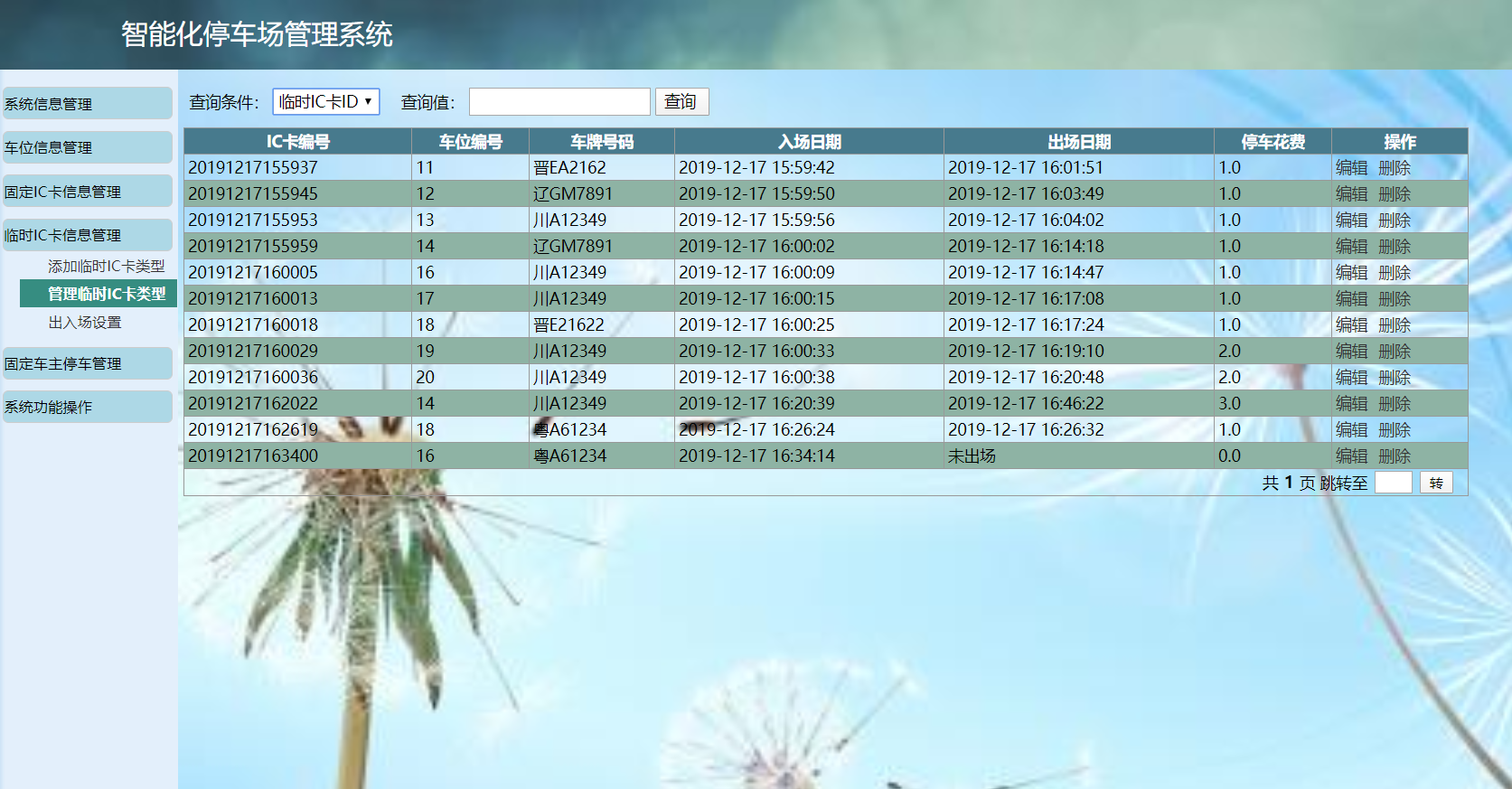


图3.3临时车主停车信息



图3.4固定车主出入场界面

3.2数据库存取技术

3.2.1 JDBC技术

JDBC（Java DataBase Connectivity,java数据库连接）是一种用于执行SQL语句的Java API，可以为多种关系数据库提供统一访问，它由一组用Java语言编写的类和接口组成。JDBC提供了一种基准，据此可以构建更高级的工具和接口，使数据库开发人员能够编写数据库应用程序。

JDBC API 既支持数据库访问的两层模型（C/S），同时也支持三层模型（B/S）。在两层模型中，Java applet或应用程序将直接与数据库进行对话。这将需要一个JDBC驱动程序来与所访问的特定数据库管理系统进行 通讯。用户的SQL语句被送往数据库中，而其结果将被送回给用户。数据库可以位于另一台计算机上，用户通过网络连接到上面。这就叫做客户机/服务器配置，其中用户的计算机为客户机，提供数据库的计算机为服务器。网络可以是 Intranet（它可将公司职员连接起来），也可以是 Internet。

在三层模型中，命令先是被发送到服务的"中间层"，然后由它将SQL 语句发送给数据库。数据库对 SQL 语句进行处理并将结果送回到中间层，中间层再将结果送回给用户。MIS 主管们都发现三层模型很吸引人，因为可用中间层来控制对公司数据的访问和可作的的更新的种类。中间层的另一个好处是，用户可以利用易于使用的高级API，而中间层将把它转换为相应的低级调用。最后，许多情况下三层结构可提供一些性能上的好处。

JDBC是面向对象的接口标准，一般有具体的数据库厂商提供。它的主要功能是存放在数据库中的数据，通过对象定义了一系列与数据库系统进行交互的类和接口。通过接口对象，应用程序可以完成与数据库的连接、执行SQL语句、从数据库中获取结果、获取状态及错误信息、终止事务和连接。

JDBC为java程序提供统一的、无缝地操作各种数据库的结构。因为实际应用中常常无法确定用户想访问什么类型的数据库，程序员使用JDBC编程时可以不关心它要操作的数据库是哪个厂家的产品，从而提高了软件的通用性。只要系统上安装了正确的驱动程序，JDBC应用程序就可以访问其相关的数据库。

3.3程序详细设计与实现

3.3.1主要算法框图

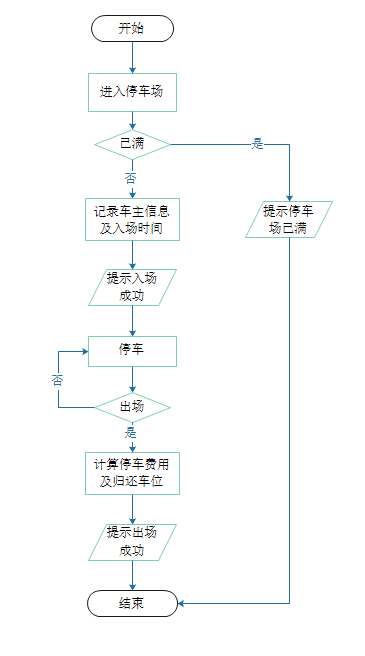
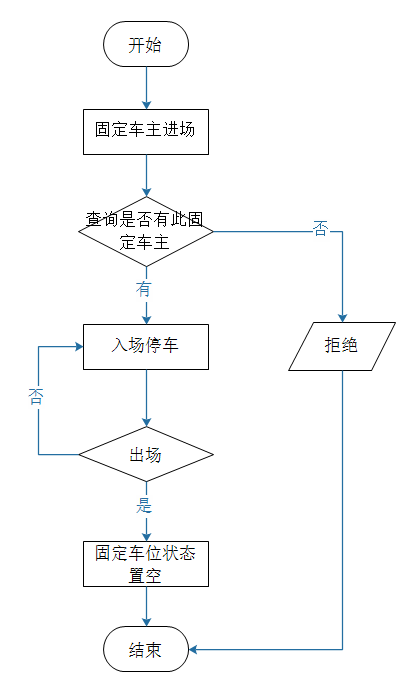
 

图3.5临时车主进出场流程图 图3.6 固定车主进出场流程图

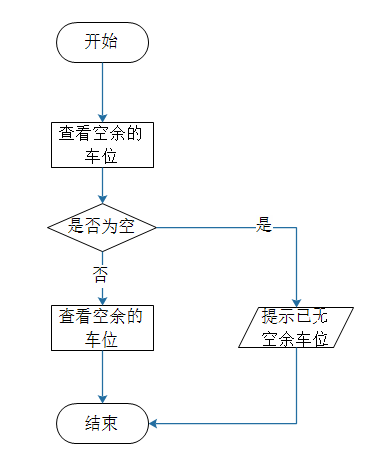


图3.7获取空余车位流程图

3.3.2典型程序代码

服务段代码：

临时车主入场代码：

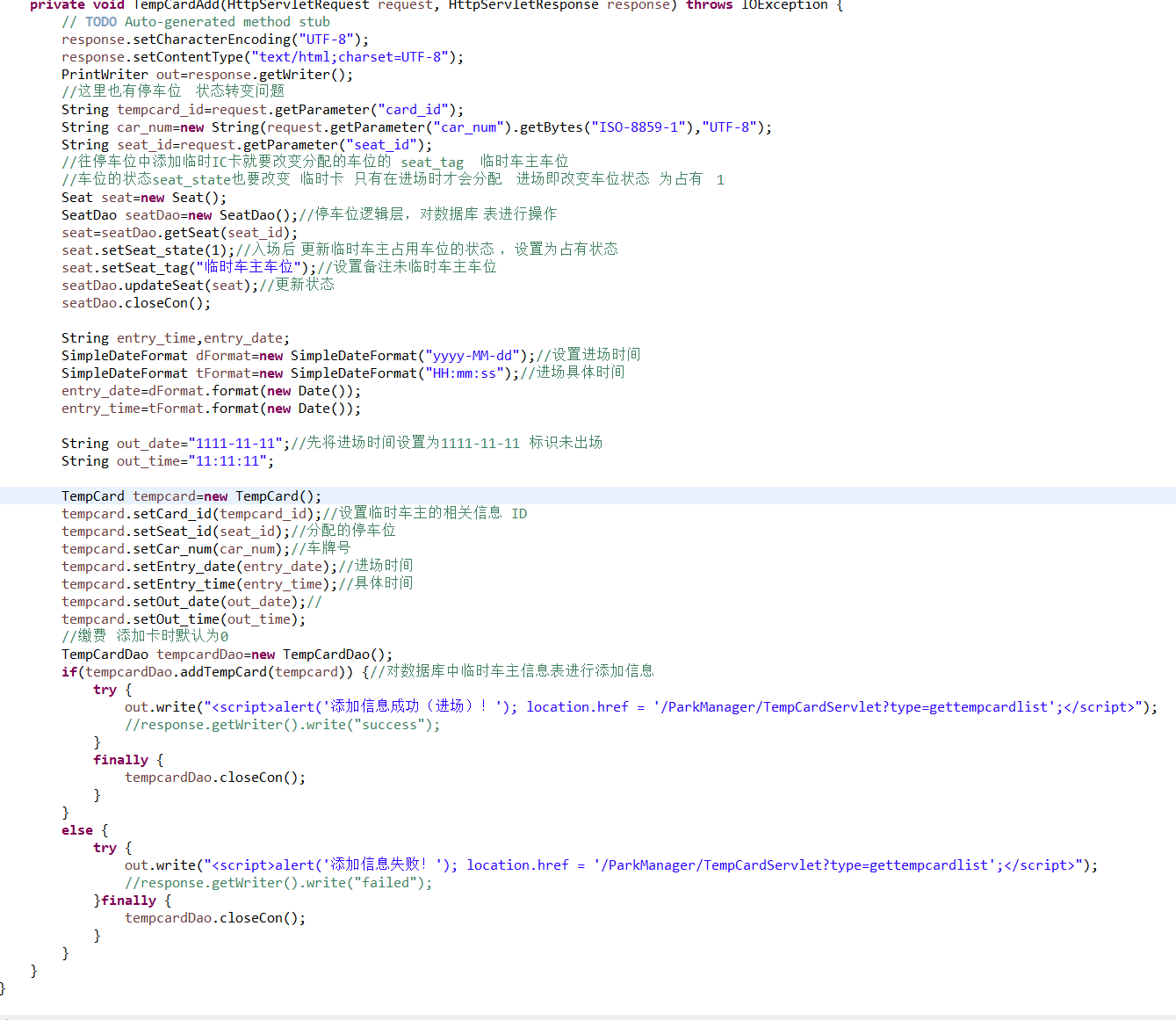


图3.8临时车主入场代码

临时车主信息展示代码：



3.9获取临时车主停车信息代码

临时车主出场代码：



图3.10临时车主出场代码

3.4测试及结果

典型测试（黑盒）：

临时车主入场添加信息：



图3.11临时车主入场测试图

入场后可看到临时车主信息表有了临时车主入场信息（最下面，未出场的）：

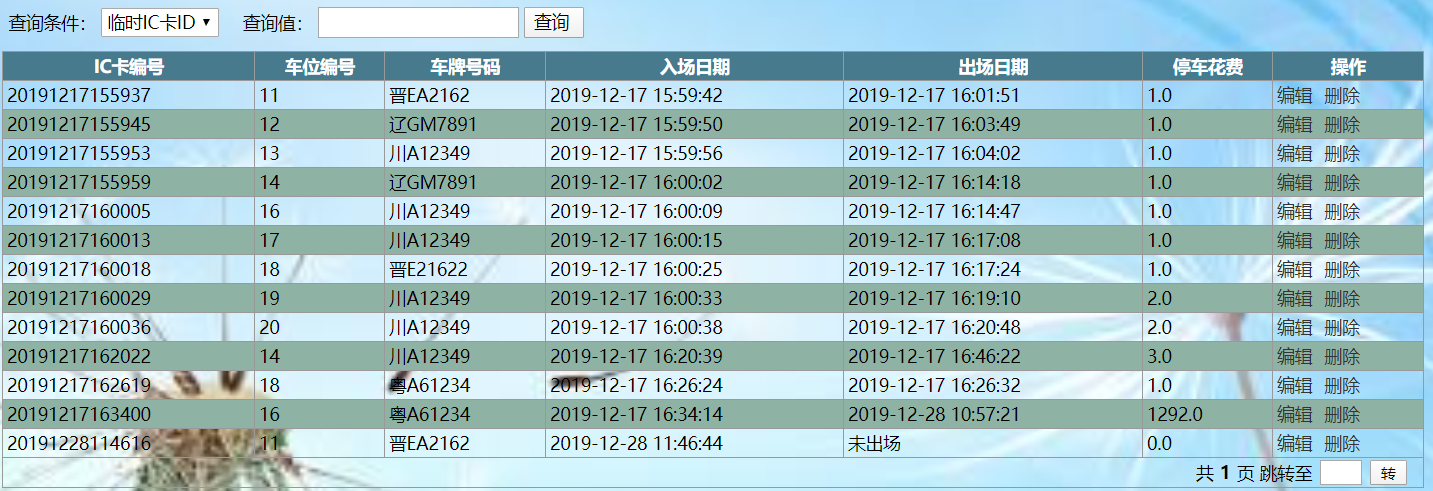


图3.12临时车主信息表测试图

进入出入场设置，选择该车出场：

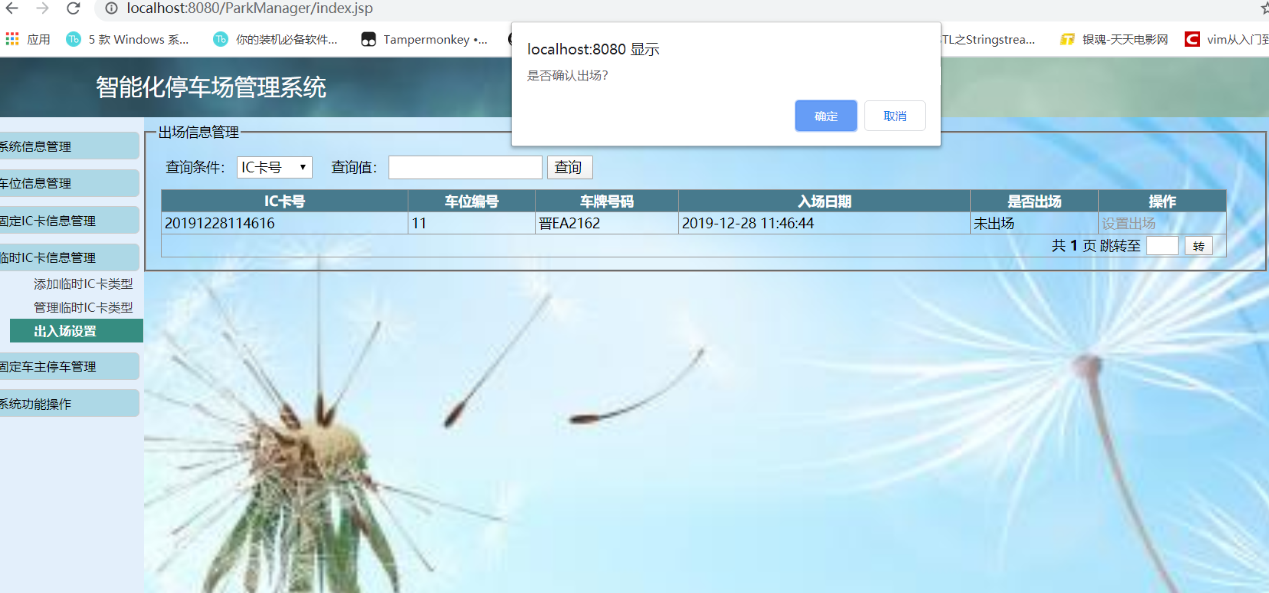


图3.13临时车主出场测试图

出场成功：

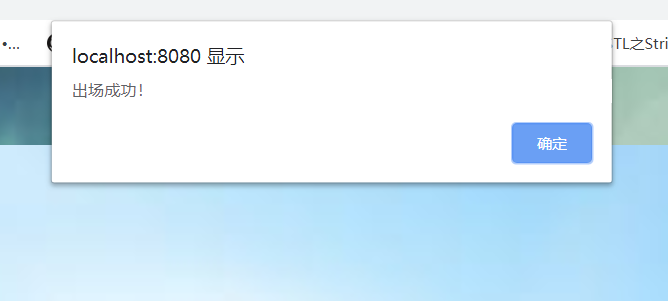


图3.14临时车主出场测试结果图

4参考文献

1. 数据库系统概论（第5版），王珊 萨师煊，高等教育出版社，2017.3
2. 软件工程导论（第6版），张海藩 牟永敏，清华大学出版社，2013.10
3. Java Web程序设计，明日科技，人民邮电出版社，2015.9
4. Java网络编程精解，孙玉琴，电子工业出版社，2007.6

**成绩评定表**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 评分项目 | 满分 | 实得分 |
| 1 | 学习态度认真、出勤情况 | 10 |  |
| 2 | 设计分析合理性 | 10 |  |
| 3 | 设计方案正确性、可行性、创造性、设计结果正确性 | 30 |  |
| 4 | 系统测试、运行情况 | 40 |  |
| 5 | 设计报告的规范性 | 10 |  |
|  |  | 总得分 |  |