**停车场收费系统系统V1.0**

**详细设计说明书**

课 程 名 称：图像处理编程实践

项 目 名 称：基于web的停车场收费系统

组        长：李鹏飞

小 组 成 员：高佳豪 高佳欢 田旭雷 王伟韬 马腾 杨哲 周帅

日 期：**2022-11-1**

1. 编写目的

随着计算机和网络的普及，以及选择买车的人越来越多，如今停车场的收费要求也越来越高，如果收费效率过低不仅会导致车辆堆积，影响停车场面容，还会使得车主等待时间较长失去市场竞争力。那么就要求一种能够快速收费的系统来辅助工作人员进行停车场收费。该项目所做的停车场收费管理系统就是为此而生的。在现代化的行业中，计算机已经广泛应用于各种应用，它能给传统的停车场管理带来重大的革命。

编制此文档的目的，是向本系统开发相关人员明确描述系统结构、功能及逻辑等各个角度的设计内容，保证大家对系统理解的一致性；同时是后续完成系统研发、测试、上线等工作的依据之一。本文同时作为课程的纲要类文档，既是教师备课的重要参考，也是学员理解系统、完成学习及开发任务的指导性框架文件。

该文档重点描述了成绩管理系统的系统架构，数据实体，类的定义，类功能的实现，部署要求，运行环境等内容，该文档可以用来指导程序员完成程序代码实现。文档的主要读者包括：项目分析人员，项目设计人员，项目经理，程序员，测试人员，部署人员，项目技术支持人员等。

1. 参考文档
2. 项目辅导老师的线上答疑
3. 学习网站的教学视频以及相关课题资料
4. 从CSDN官网搜索相关的学习资料
5. 相关术语解释

（1）运行环境:系统运行的环境及数据库管理软件的版本称为为运行环境。

（2）数据库建模:在设计数据库时，对现实世界进行分析、抽象、并从中找出内在联系，进而确定数据库的结构，这一过程称为数据库建模。它主要包括两部分内容:确定最基本的数据接口，对约束建模。。

（3）概念模型:概念模型是用于信息世界的建模，是现实世界的第一层抽象。

（4）实体关系图:也成为E-R图，提供描述实体型、属性和联系的方法，用来描述现实世界的概念模型。实体关系同表示在信息系统中概念模型的数据存储。

1. 系统架构综述

停车场收费系统系统主要有两个端口，分别是停车场收费人员和停车场管理人员。系统功能架构主要是两个大部分，一部分为收费员模块，包含入场，出场和会员注册功能，另一部分为管理员模块，核心功能有设置修改，信息查看和统计与生成图表，而具体的图像识别技术又可以划分为三个模块的功能。

每个模块的核心功能如下：

1. **收费员模块核心功能（车辆入场）：** 车辆一旦进入停车场的入口岗亭区，系统可以正确的识别车牌号，即便车辆照片有不同角度的测斜、有一定的模糊程度，系统也可以实现精准识别，将其以及拍摄照片保存在系统数据库中，并将识别的结果返回到停车场收费人员的页面上，停车场收费人员确定无误后可点击按钮实现智能抬杆放行。如图2.1所示：



图2.1

**（2） 收费员模块核心功能（车辆出场）：**车辆一旦进入停车场的出口岗亭区，系统可以正确的识别车牌号，将其车牌号和拍摄照片保存在系统数据库中，并将入口和出口的照片及车牌号输出至收费人员端，停车场收费人员可确定识别是否正确，如图所示。

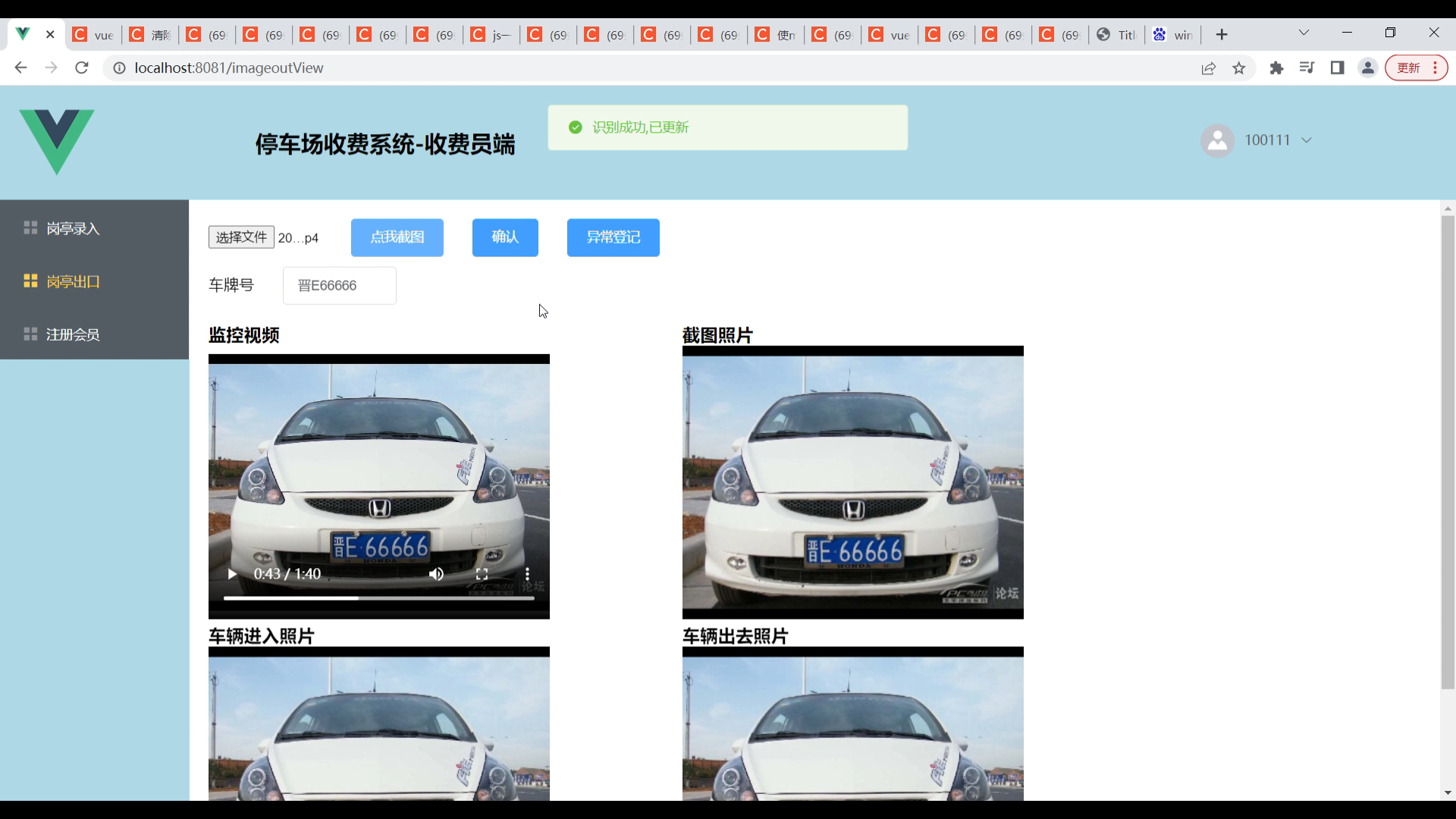
 **（3） 收费员模块核心功能（会员注册）：**收费人员可以对所有会员进行查询，并且对缴费的会员的信息进行上传。如图2.3所示



图2.3

**（4）管理员模块核心功能（修改设置）：**



图2.4

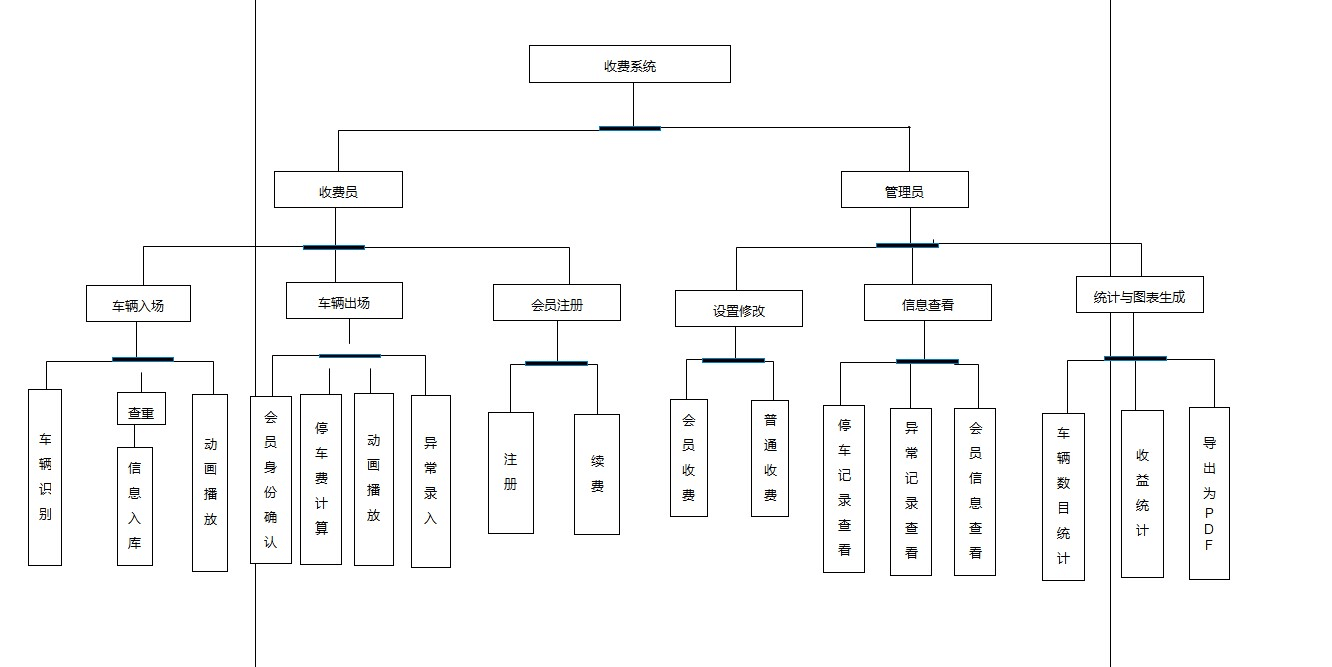
**（5）管理员核心模块功能（信息查看）：**

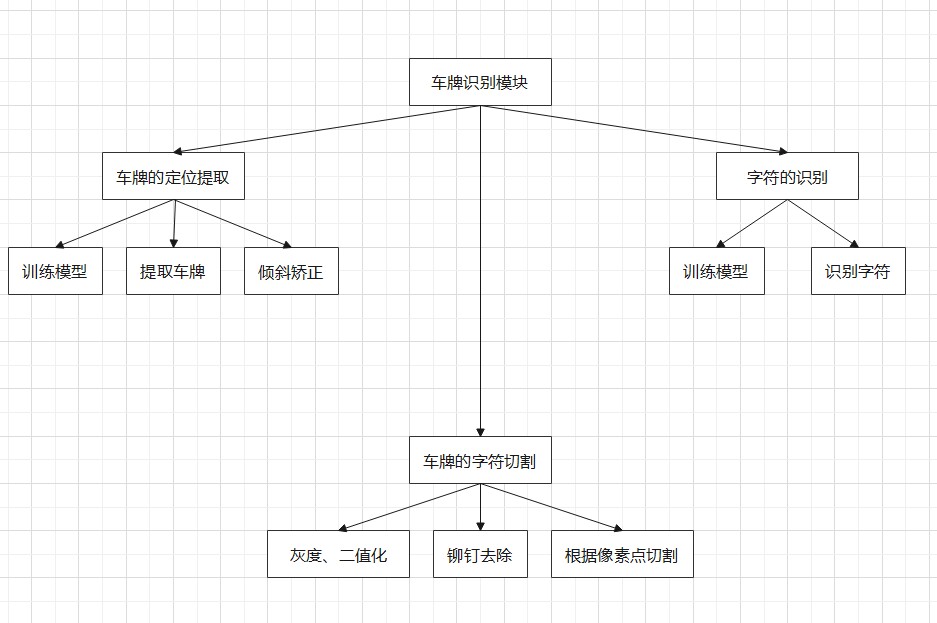


**（6）管理员核心模块功能（统计与生成图表）：**



**（7）功能模块E\_R图：**





1. 静态结构设计
   1. 数据实体类

根据需求分析，我们设计出以下数据实体类：

1. LogRec：登入/登出记录类，用来表示一条原始的用户登入/登出记录。
2. MLogRec：匹配的登录记录类，用来表示一条匹配的用户登录记录。

以下分别对这两个实体类的设计作出说明：

* + 1. 登入/登出记录数据类

1，登入/登出记录类（LogRec）类图：

2，登入/登出记录类（LogRec）成员变量说明：

|  |  |  |
| --- | --- | --- |
| **成员变量定义** | **成员变量说明** | **备注** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

* + 1. 匹配的用户登录记录类

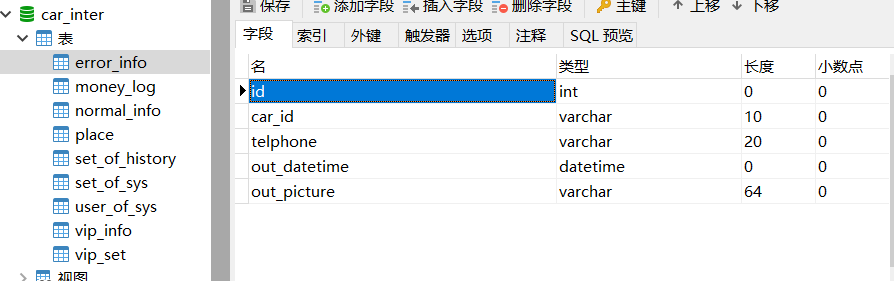
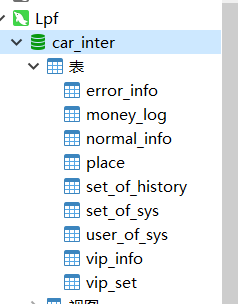
1，匹配的用户登录记录类（MLogRec）类图：

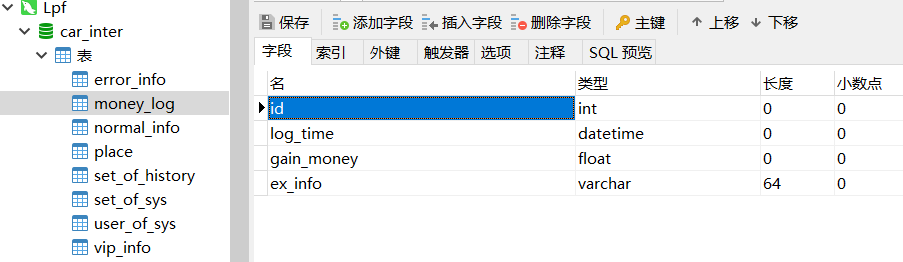
2，匹配的用户登录记录类（MLogRec）成员变量说明：

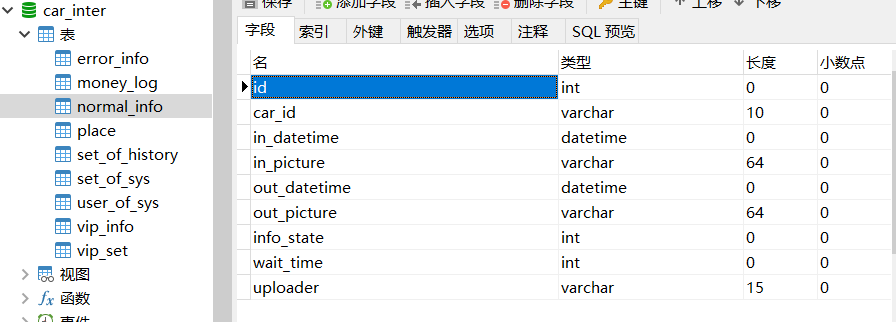
|  |  |  |
| --- | --- | --- |
| **成员变量定义** | **成员变量说明** | **备注** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

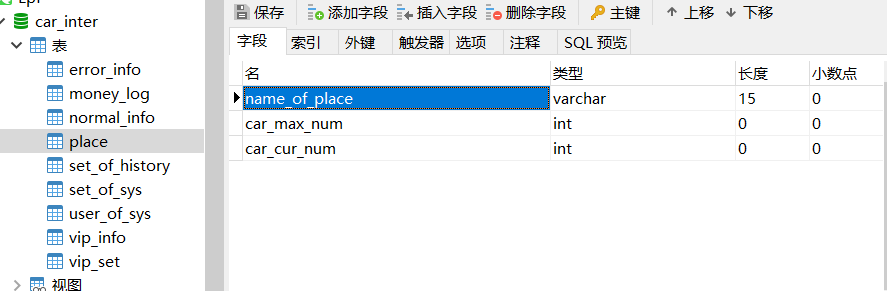
* 1. 数据库表设计

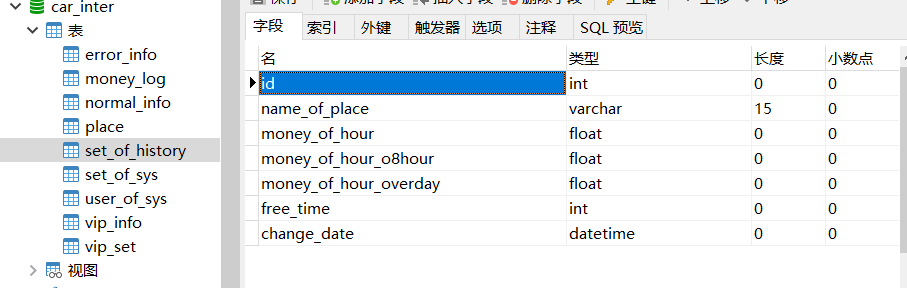
根据需求，设计出如下数据库表结构：

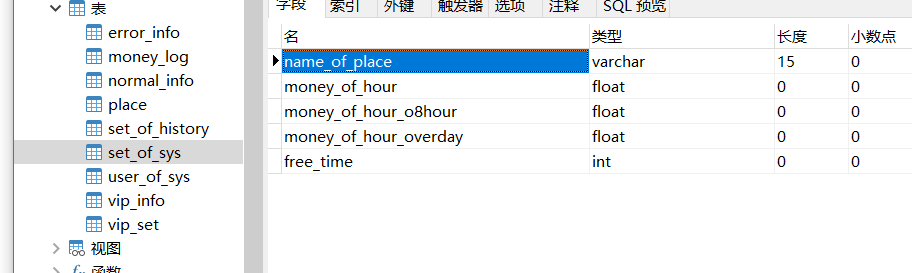


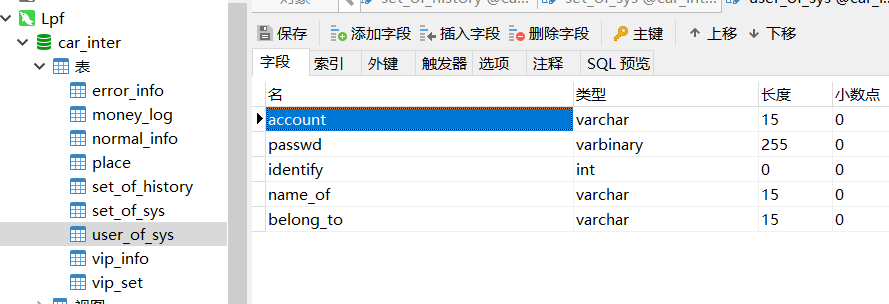


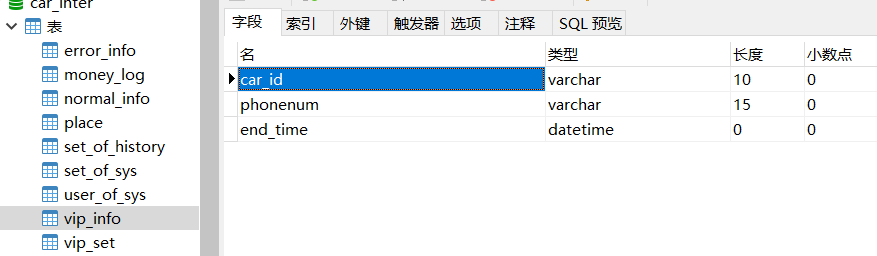


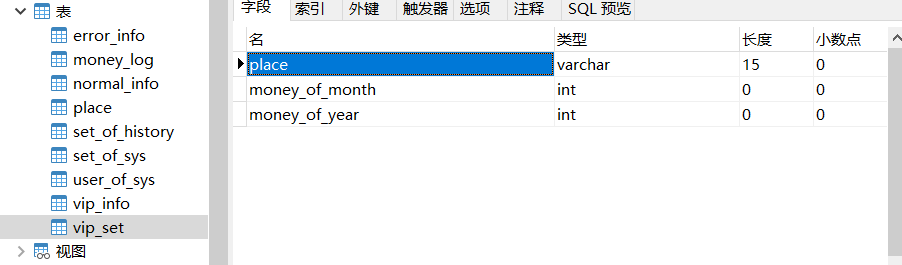












* 1. Client 类（一直到5.15.3为非具体系统部分的类的介绍，5.15.3以后为具体类的介绍，需要可以直接跳转）

Client**类封装数据采集系统客户端的所有功能。**

* + 1. Client类图

* + 1. Client类成员变量说明

|  |  |
| --- | --- |
| **成员变量定义** | **成员变量说明** |
| private:LogReader& m\_reader; | 日志读取器完成日志读取的功能 |
| private:LogSender& m\_sender; | 日志发送器完成日志发送的功能 |
|  |  |

* + 1. Client类方法说明

1. 构造函数

|  |  |
| --- | --- |
| 函数原型 | Client(LogReader& reader, LogSender& sender); |
| 参数说明 | reader:日记读取器  sender:日志发送器(SocketSender或ConsoleSender对象) |

2）dataMine方法定义：

|  |  |
| --- | --- |
| 方法原型 | public:void dataMine(void) throw(ClientException); |
| 方法功能 | 启动客户端数据采集和发送 |
| 参数说明 | 无 |
| 返回类型 | void |
| 抛出异常 | ClientException |

* 1. LogReader类  
     logReader类封装了日志读取的所有功能
     1. LogReader类图
     2. LogReader成员变量说明

|  |  |
| --- | --- |
| **成员变量定义** | **成员变量说明** |
| private: string m\_logFile | 原始日志文件路径名称 |
| private: string m\_loginsFile | 上一次未匹配成功的登入日志保存文件名 |
| private: string m\_failFile | 保存发送失败的日志记录的文件路径名 |
| private:list<LogRec> m\_logins | 登入日志记录集 |
| private:list<LogRec> m\_logouts | 登出日志记录集 |
| private:list<MLogRec> m\_logs | 匹配好的日志记录集 |

* + 1. LogReader类方法说明

1. 构造函数

|  |  |
| --- | --- |
| 函数原型 | LogReader(string const& logFile, string const& loginsFile,  string const& failFile); |
| 参数说明 | logFile:日志管理器生成的原始日志文件  loginsFile:保存上一次未匹配成功的登入日志记录文件  failFile:保存上一次未发送出去的匹配好的日志记录文件 |

1. readLog方法：

|  |  |
| --- | --- |
| 方法原型 | public list<MLogRec>& readLog(void) throw(ClientException) |
| 方法功能 | 备份原始日志文件、读上一次未匹配的登入日志文件，  读取备份日志文件，匹配，保存本次未匹配的登入日志记录 |
| 参数说明 | 无 |
| 返回类型 | 匹配好的日志记录链表引用(成员变量) |
| 抛出异常 | 调用函数中可能抛出的异常 |

1. backup方法：

|  |  |
| --- | --- |
| 方法原型 | private void backup(void) |
| 方法功能 | 备份日志文件 |
| 参数说明 | 无 |
| 返回类型 | 无 |
| 抛出异常 | BackupException |

1. readLoginsFile方法：

|  |  |
| --- | --- |
| 方法原型 | Private: void readLoginsFile(void) |
| 方法功能 | 读取上一次没有匹配成功的、剩余的登入日志记录 |
| 参数说明 | 无 |
| 返回类型 | 无 |
| 抛出异常 | ReadException |

1. readBackupFile方法：

|  |  |
| --- | --- |
| 方法原型 | private:void readBackupFile(void) |
| 方法功能 | 读取备份日志文件，分别将登入、登入日志记录  存入到相应的日志记录集中 |
| 参数说明 | 无 |
| 返回类型 | 无 |
| 抛出异常 | ReadException |

6) match方法：

|  |  |
| --- | --- |
| 方法原型 | private void match(void) |
| 方法功能 | 匹配登入、登出日志记录，形成匹配好的日志记录并存入  到匹配日志记录集 |
| 参数说明 | 无 |
| 返回类型 | 无 |
| 抛出异常 | 无 |

6) saveLoginsFile方法：

|  |  |
| --- | --- |
| 方法原型 | private: void saveLoginsFile(void) |
| 方法功能 | 保存没有匹配成功的登入日记记录 |
| 参数说明 | 无 |
| 返回类型 | 无 |
| 抛出异常 | SaveException |

* 1. LogSender类
     1. LogSender类方法说明

1）sendLog方法：

|  |  |
| --- | --- |
| 方法原型 | public:virtual void sendLog(list<MLogRec>& logs)  throw(ClientException) = 0; |
| 方法功能 | 定义发送日志记录的接口原型 |
| 参数说明 | 匹配好的日志记录集的引用(LogReader中的成员变量) |
| 返回类型 | 无 |
| 抛出异常 | ClientException |

1. 析构函数

|  |  |
| --- | --- |
| 函数原型 | virtual ~LogSender(); |

* 1. SocketSender类
     1. SocketSender类图
     2. SocketSender成员变量说明

|  |  |
| --- | --- |
| **成员变量定义** | **成员变量说明** |
| string m\_failFile | 记录上一次发送失败的日志记录文件名 |
| string m\_ip | 服务器的IP地址 |
| short m\_port | 网络通信需要的端口号 |
| int m\_sockfd | 网络套接字描述符 |

* + 1. SocketSender类方法说明

1. 构造函数

|  |  |
| --- | --- |
| 参数说明 | 1. SocketSender(string const& failFile, string const& ip, short port); |

1. sendLog方法说明

|  |  |
| --- | --- |
| 方法原型 | public:void sendLog(list<MLogRec>& logs)  Throw(ClientException); |
| 方法功能 | 通过网络发送日志记录 |
| 参数说明 | 匹配好的日记记录集 |
| 返回类型 | 无 |
| 抛出异常 | ReadException, SocketException, SendException |

3）connectServer方法：

|  |  |
| --- | --- |
| 方法原型 | pvivate:void connectServer(void)  throw(SocketException); |
| 方法功能 | 建立客户机和服务器的连接 |
| 参数说明 | 无 |
| 返回类型 | 无 |
| 抛出异常 | SocketException |

4）readFailFile方法：

|  |  |
| --- | --- |
| 方法原型 | pvivate: void readFailFile(list<MLogRec>& logs)  throw(ReadException); |
| 方法功能 | 读取上一次发送失败的日志记录文件，将读取到的日志  记录存入到匹配好的日志记录集中 |
| 参数说明 | 匹配好的日志记录集的引用(LogReader的成员变量m\_logs) |
| 返回类型 | 无 |

5）sendData方法

|  |  |
| --- | --- |
| 方法原型 | private:void sendData(list<MLogRec>& logs)  throw(SendException); |
| 方法功能 | 通过网络把容器中的数据发送给服务器 |
| 参数说明 | 存放匹配好的日志记录集的引用(LogReader的成员变量m\_logs) |
| 返回类型 | 无 |
| 抛出异常 | SendException |

6）saveFailFile方法：

|  |  |
| --- | --- |
| 方法原型 | Private:void saveFailFile(list<MLogRec>& logs)  Throw(SaveException); |
| 方法功能 | 保存发送失败的日志记录 |
| 参数说明 | 未发送出去的日志记录集(LogReader的成员变量m\_logs) |
| 返回类型 | 无 |
| 抛出异常 | SaveException |

* 1. ConsoleSender类
     1. ConsoleSender类方法说明

1)sendLog方法说明

|  |  |
| --- | --- |
| 方法原型 | public:void sendLog(list<MLogRec>& logs) throw(); |
| 方法功能 | 将日志记录输出到控制台 |
| 参数说明 | 匹配好的日志记录集的引用(LogReader::m\_logs) |
| 返回类型 | void |

* 1. Server类
     1. Server类图
     2. Server类成员变量说明

|  |  |
| --- | --- |
| **成员变量定义** | **成员变量说明** |
| ServerSocket m\_socket | 服务器套接字 |
| StoreThread m\_strore | 存储线程 |

* + 1. Server类方法说明

1. 构造函数

|  |  |
| --- | --- |
| 参数说明 | Server(LogDao& dao, string const& ip, short port); |

1. dataMine方法说明

|  |  |
| --- | --- |
| 方法原型 | Public：void dataMine(void) |
| 方法功能 | 启动服务器端数据采集 |
| 参数说明 | 无 |
| 返回类型 | void |

* 1. ServerSocket类
     1. ServerSocket类图
     2. ServerSocket成员说明

|  |  |
| --- | --- |
| **成员变量定义** | **成员变量说明** |
| int m\_sockfd | 建立网络连接需要的套接字 |
|  |  |

* + 1. ServerSocket方法说明

1）构造函数

|  |  |
| --- | --- |
| 参数说明 | ServerSocket(string const& ip, short port); |

1. acceptClient方法说明

|  |  |
| --- | --- |
| 方法原型 | public:void acceptClient(void) |
| 方法功能 | 等待客户端的连接，如有客户端连接上来，  创建客户线程接收客户端的数据 |
| 参数说明 | 无 |
| 返回类型 | void |

* 1. LogThread类说明

1. start方法说明

|  |  |
| --- | --- |
| 方法原型 | public:void start(void); |
| 方法功能 | 创建并启动线程 |
| 参数说明 | 无 |
| 返回类型 | 无 |

1. run方法说明

|  |  |
| --- | --- |
| 方法原型 | private:virtual void run(void)=0; |
| 方法功能 | 提供统一的，抽象的接口 |
| 参数说明 | 无 |
| 返回类型 | void |

1. task方法说明

|  |  |
| --- | --- |
| 方法原型 | private:staic void\* task(void\* arg); |
| 方法功能 | 创建线程所需要的、不依赖于对象来调用的、具体一  定全局属性的的静态函数，获取到子类对象的指针，  调用体现子类特殊的run函数 |
| 参数说明 | void\*(此处为子类对象的指针) |
| 返回类型 | void\* |

1. ~LogThread方法说明

|  |  |
| --- | --- |
| 方法原型 | virtual ~LogThread(void); |
| 方法功能 | 当用户使用多态的时候，调用基类析构的时候，也调用子类的析构函数 |
| 参数说明 | void\*(此处为子类对象的指针) |
| 返回类型 | void\* |

* 1. ClientThread类说明
     1. ClientThread类图
     2. ClientThread成员说明

|  |  |
| --- | --- |
| **成员变量定义** | **成员变量说明** |
|  |  |
|  |  |

* + 1. ClientThread方法说明

1. 构造函数

|  |  |
| --- | --- |
| 参数说明 |  |

1. run方法说明

|  |  |
| --- | --- |
| 方法原型 |  |
| 方法功能 |  |
| 参数说明 |  |
| 返回类型 |  |

1. 析构函数

|  |  |
| --- | --- |
| 方法原型 |  |
| 方法功能 |  |
| 参数说明 |  |
| 返回类型 |  |

* 1. StoreThread类
     1. StoreThread类图
     2. StoreThread成员说明

|  |  |
| --- | --- |
| **成员变量定义** | **成员变量说明** |
| LogDao& m\_dao | 将数据插入到数据库中的DAO |

* + 1. StoreThread方法说明

1）.构造函数

|  |  |
| --- | --- |
| 参数说明 | StoreThread(LogDao& dao) |

2）run方法说明

|  |  |
| --- | --- |
| 方法原型 | Private:void run(void) |
| 方法功能 | 从日志队列中取出数据，交给DAO |
| 参数说明 | 无 |
| 返回类型 | void |

~StoreThread方法说明

|  |  |
| --- | --- |
| 方法原型 |  |
| 方法功能 |  |
| 参数说明 | 无 |
| 返回类型 | 无 |

* 1. LogDao类
     1. LogDao类方法说明

1）insert方法说明

|  |  |
| --- | --- |
| 方法原型 | Public：virtual void insert(MLogRec const& log)  =0; |
| 方法功能 | 为插入数据提供统一，抽象的接口 |
| 参数说明 | 匹配好的日志记录 |
| 返回类型 | void |

* 1. OracleDao类
     1. OracleDao类图
     2. OracleDao成员说明

|  |  |
| --- | --- |
| **成员变量定义** | **成员变量说明** |
|  |  |
|  |  |

* + 1. OracleDao方法说明

1）.构造函数

|  |  |
| --- | --- |
| 参数说明 |  |

2）insert方法说明

|  |  |
| --- | --- |
| 方法原型 |  |
| 方法功能 |  |
| 参数说明 |  |
| 返回类型 |  |

* 1. LogQueue类
     1. LogQueue类图
     2. LogQueue成员说明

|  |  |
| --- | --- |
| **成员变量定义** | **成员变量说明** |
| list<MLogRec> m\_logs |  |
| pthread\_mutex\_t m\_mtx; |  |
| pthead\_cond\_t m\_notFull; |  |
| pthread\_cond\_t m\_notEmpty; |  |

* + 1. LogQueue方法说明

1）.构造函数

|  |  |
| --- | --- |
| 参数说明 | 无 |

2）operator<< 方法说明

|  |  |
| --- | --- |
| 方法原型 | public LogQueue& operator<<(MLogRec const& log); |
| 方法功能 |  |
| 参数说明 |  |
| 返回类型 |  |

3）operator>> 方法说明

|  |  |
| --- | --- |
| 方法原型 | public LogQueue& operator>>(MLogRec& log); |
| 方法功能 | 从日志队列中取出一条记录 |
| 参数说明 | 匹配日志记录 |
| 返回类型 | 日志队列的引用 |

5.16 ErroeInfo类

5.16.1 ErroeInfo类图

5.16.2ErroeInfo类成员说明:

同数据库error\_info表封装

5.16.3ErroeInfo类方法说明

1）.构造函数

|  |  |
| --- | --- |
| 参数说明 | 有参与无参构造器封装在实体类中 |

2）count << 方法说明

|  |  |
| --- | --- |
| 方法原型 | public ResponseBean count(){} |
| 方法功能 | 查看当前存在多少条异常记录 |
| 参数说明 | 无 |
| 返回类型 | Json格式 |

3）addExceptiont << 方法说明

|  |  |
| --- | --- |
| 方法原型 | Public responseBean addException(@RequestBody ErrorInfo errorInfo){ |
| 方法功能 | 添加异常信息 |
| 参数说明 | 网页传参 |
| 返回类型 | Boolean |

4）selectAllError>> 方法说明

|  |  |
| --- | --- |
| 方法原型 | public ResponseBean selectAllError(){} |
| 方法功能 | 查询所有异常 |
| 参数说明 | 无 |
| 返回类型 | Json格式 |

5）findErrorBycarId>> 方法说明

|  |  |
| --- | --- |
| 方法原型 | public ResponseBean findErrorBycarId(@RequestParam String carId){} |
| 方法功能 | 根据汽车的id来查询该车的异常 |
| 参数说明 | CarId |
| 返回类型 | Json格式 |

5.17moneyLog类

5.17.1 moneyLog类成员变量说明

由数据库moneyLog封装

5.17.2 moneyLog类方法说明

1）addMoney>> 方法说明

|  |  |
| --- | --- |
| 方法原型 | public ResponseBean addMoney(@RequestParam String carId){} |
| 方法功能 | 添加收费信息 |
| 参数说明 | Carid |
| 返回类型 | 布尔 |

2）day>> 方法说明

|  |  |
| --- | --- |
| 方法原型 | public ResponseBean day(@RequestParam String dateTime){} |
| 方法功能 | 日统计 |
| 参数说明 | Datetime |
| 返回类型 | Json格式 |

5.18 NormalInfo类

5.18.1 NormalInfo类成员变量

由数据库表格封装为实体类

5.18.2 NormalInfo类方法

1）findByData>> 方法说明

|  |  |
| --- | --- |
| 方法原型 | public ResponseBean findByData(@RequestParam String data){} |
| 方法功能 | 指定时间查询停车记录 |
| 参数说明 | 网页传参 |
| 返回类型 | Json格式 |

2）getMoney>> 方法说明

|  |  |
| --- | --- |
| 方法原型 | public ResponseBean getMoney(@RequestParam String carId){} |
| 方法功能 | 指定汽车id查询收费 |
| 参数说明 | Carid |
| 返回类型 | Json格式。Money |

3）selectAllNormal1>> 方法说明

|  |  |
| --- | --- |
| 方法原型 | public ResponseBean selectAllNormal1(@RequestParam long pageNo,@RequestParam long size){} |
| 方法功能 | 所有未离开车辆查询 |
| 参数说明 | 分页尺寸 |
| 返回类型 | Json格式 |

4）selectAllNormal2>> 方法说明

|  |  |
| --- | --- |
| 方法原型 | public ResponseBean selectAllNormal2(){} |
| 方法功能 | 查看符合模糊条件的未离开车辆 |
| 参数说明 | 分页尺寸，模糊条件 |
| 返回类型 | Json格式 |

5）updateImage>> 方法说明

|  |  |
| --- | --- |
| 方法原型 | public ResponseBean updateImage() { |
| 方法功能 | 视频上传 |
| 参数说明 | 地址 |
| 返回类型 | Json格式 |

6）update2>> 方法说明

|  |  |
| --- | --- |
| 方法原型 | public ResponseBean update2(){ |
| 方法功能 | 修改车辆记录，并计算停车时长 |
| 参数说明 | Carid picture uploader |
| 返回类型 | Json格式 |

5.19Place类

5.19.1 Place类成员变量

由数据库封装

5.19.2 Place类方法变量

1）find1>> 方法说明

|  |  |
| --- | --- |
| 方法原型 | public ResponseBean find1(){} |
| 方法功能 | 查询当前车位占用数与最大车位数 |
| 参数说明 |  |
| 返回类型 | Json格式 |

5.20 SetOfHistory类

5.20.1 SetOfHistory类成员变量

由数据库表封装

5.20.2 SetOfHistory类方法变量

1）selectAllHistory>> 方法说明

|  |  |
| --- | --- |
| 方法原型 | public ResponseBean selectAllHistory (){} |
| 方法功能 | 查询历史收费管理记录 |
| 参数说明 |  |
| 返回类型 | Json格式 |

5.21 SetOfSys类

5.21.1 SetOfSys类成员变量

由数据库表封装

5.21.2 SetOfSys类方法变量

1）selectAllHistory>> 方法说明

|  |  |
| --- | --- |
| 方法原型 | Public response UpdateSystem(){} |
| 方法功能 | 修改系统设置 |
| 参数说明 | moneyOfHou, moneyOfHourO8hout, moneyOfHourOverday, freeTime |
| 返回类型 | Json格式 |

5.22 UserOfSys类

5.22.1 UserOfSys类成员变量

由数据库表封装

5.22.2 UserOfSys类方法变量

1）Login>> 方法说明

|  |  |
| --- | --- |
| 方法原型 | public ResponseBean Login() {} |
| 方法功能 | 登录 |
| 参数说明 | loginUser |
| 返回类型 | Json格式 |

2）registration>> 方法说明

|  |  |
| --- | --- |
| 方法原型 | public ResponseBean registration(){} |
| 方法功能 | 注册 |
| 参数说明 | User网页传参 |
| 返回类型 | Json格式 |

3）findUserByIdentify>> 方法说明

|  |  |
| --- | --- |
| 方法原型 | public ResponseBean findUserByIdentify(){} |
| 方法功能 | 根据identify查找 |
| 参数说明 | identify |
| 返回类型 | Json格式 |

4）getCountByIdentify>> 方法说明

|  |  |
| --- | --- |
| 方法原型 | public ResponseBean getCountByIdentify(){ |
| 方法功能 | 根据identify统计 |
| 参数说明 | identify |
| 返回类型 | Json格式 |

5.23VipInfo类

5.23.1 VipInfo类成员变量

由数据库表封装

5.23.2 VipInfo类方法变量

1）VIPMonthFirst>> 方法说明

|  |  |
| --- | --- |
| 方法原型 | public ResponseBean VIPMonthFirst(){ |
| 方法功能 | 初次申请包月VIP |
| 参数说明 | Carid phonenumber month |
| 返回类型 | Json格式 |

2）find2>> 方法说明

|  |  |
| --- | --- |
| 方法原型 | public ResponseBean find2(){ |
| 方法功能 | 查看某个指定车牌号VIP的剩余时间（单位为天） |
| 参数说明 | Carid |
| 返回类型 | Json格式 |

3）VIPMonthAfter>> 方法说明

|  |  |
| --- | --- |
| 方法原型 | public ResponseBean VIPMonthAfter(){ |
| 方法功能 | 会员续费（单位月） |
| 参数说明 | Carid month |
| 返回类型 | Json格式 |

4）infoUpdTotal>> 方法说明

|  |  |
| --- | --- |
| 方法原型 | public ResponseBean infoUpdTotal(){ |
| 方法功能 | 用于异常处理界面的接口，根据ID进行检索，可用于修改包括车牌号在内的大部分属性 |
| 参数说明 | Carid id outpicture putdatetime uplpader |
| 返回类型 | Json格式 |

**5.24VipSet类**

**5.24.1 VipSet类成员变量**

由数据库表封装

**5.24.2 VipSet类方法变量**

1）getOne>> 方法说明

|  |  |
| --- | --- |
| 方法原型 | public ResponseBean getOne(){ |
| 方法功能 | 获取vip设置 |
| 参数说明 |  |
| 返回类型 | Json格式 vipset |

2）updateOne>> 方法说明

|  |  |
| --- | --- |
| 方法原型 | public ResponseBean updateOne (){ |
| 方法功能 | 更新vip设置 |
| 参数说明 | vipSet网页传参 |
| 返回类型 | Json格式 vipset |

1. 异常设计
   1. 客户端异常类：

6.1.1客户端异常类图

6.1.2客户端异常类说明

|  |  |  |
| --- | --- | --- |
| **异常类名** | **异常类责任说明** | **备注** |
| **ClientException** |  |  |
| **BackupException** |  |  |
| **ReadException** |  |  |
| **SocketException** |  |  |
| **SendException** |  |  |
| **SaveException** |  |  |

* 1. 服务端异常类

6.2.1服务端异常类图

6.2.2服务端异常类说明

|  |  |  |
| --- | --- | --- |
| **异常类名** | **异常类责任说明** | **备注** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |