**公车监管与服务平台详细设计**



安徽皖通科技股份有限公司

2015年07月

修订记录:

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 修订人 | 修订日期 | 修订内容 |
| V1.0 | 于鹏、张亚 | 2015-7-30 | 初稿 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目 录

[1 引言 1](#_Toc426933486)

[1.1 编写目的 1](#_Toc426933487)

[1.2 背景 1](#_Toc426933488)

[1.3 术语定义 1](#_Toc426933489)

[1.4 参考资料 3](#_Toc426933490)

[2 公车管理服务平台系统结构 3](#_Toc426933491)

[2.1 系统结构图 3](#_Toc426933492)

[2.2 慧管车平台系统结构概述 4](#_Toc426933493)

[2.3 系统部署图 5](#_Toc426933494)

[2.4 开放接口设计 5](#_Toc426933495)

[3 慧管车平台角色定义 5](#_Toc426933496)

[3.1 运营平台管理员 6](#_Toc426933497)

[3.2 机构管理员 6](#_Toc426933498)

[3.3 稽查管理员 6](#_Toc426933499)

[3.4 公车管理员 7](#_Toc426933500)

[3.5 公车驾驶员 7](#_Toc426933501)

[3.6 公车使用者 7](#_Toc426933502)

[3.7 机构领导用户 7](#_Toc426933503)

[4 慧管车平台详细设计 8](#_Toc426933504)

[4.1 用户管理详细设计 8](#_Toc426933505)

[4.1.1 用户权限认证详细设计 8](#_Toc426933506)

[4.1.2 用户session保持与令牌管理 10](#_Toc426933507)

[4.1.3 组织架构详细设计 11](#_Toc426933508)

[4.1.4 用户管理详细设计 12](#_Toc426933509)

[4.1.5 驾驶员信息管理详细设计 13](#_Toc426933510)

[4.2 车辆管理详细设计 15](#_Toc426933511)

[4.2.1 实时位置详细设计 15](#_Toc426933512)

[4.2.1 车辆日常信息管理详细设计 17](#_Toc426933513)

[4.2.2 人车匹配 18](#_Toc426933514)

[4.3 调度管理详细设计 19](#_Toc426933515)

[4.3.1 订单录入详细设计 19](#_Toc426933516)

[4.3.2 工作流详细设计 20](#_Toc426933517)

[4.3.1 行车记录详细设计 21](#_Toc426933518)

[4.3.2 区域管理详细设计 23](#_Toc426933519)

[4.4 地图管理详细设计 24](#_Toc426933520)

[4.5 事件提醒详细设计 26](#_Toc426933521)

[4.5.2 实时消息检查 28](#_Toc426933522)

[4.5.3 定时消息检查 29](#_Toc426933523)

[4.5.4 消息系统详细设计 30](#_Toc426933524)

[4.5.5 应急救援详细设计 31](#_Toc426933525)

[4.6 设备管理详细设计 32](#_Toc426933526)

[4.6.1 OBD设备管理详细设计 32](#_Toc426933527)

[4.6.2 OBD数据分表详细设计 33](#_Toc426933528)

[4.7 统计分析详细设计 34](#_Toc426933529)

[4.7.1 统计表详细设计 34](#_Toc426933530)

[4.7.2 统计图详细设计 35](#_Toc426933531)

[4.8 第三方接口详细设计 35](#_Toc426933532)

[4.8.1 信息交换接口 35](#_Toc426933533)

[4.8.2 数据共享接口 35](#_Toc426933534)

[4.9 稽查管理详细设计 35](#_Toc426933535)

[4.9.1 行程稽查 36](#_Toc426933536)

[4.9.2 费用稽查 36](#_Toc426933537)

[4.9.3 专题稽查 36](#_Toc426933538)

[4.10 运营管理需求 36](#_Toc426933539)

[4.10.1 系统运行监控 36](#_Toc426933540)

[4.10.2 组织架构管理 38](#_Toc426933541)

[4.10.3 日志管理 40](#_Toc426933542)

[4.10.4 系统版本更新与管理 40](#_Toc426933543)

[4.11 移动应用（APP）功能设计 41](#_Toc426933544)

[4.11.1 公车使用者 43](#_Toc426933545)

[4.11.2 部门领导 43](#_Toc426933546)

[4.11.3 公车管理员 44](#_Toc426933547)

[4.12 车载终端设备 46](#_Toc426933548)

[4.12.1 OBD设备 46](#_Toc426933549)

[4.12.2 ETC设备 47](#_Toc426933550)

[4.13 其他 48](#_Toc426933551)

[4.13.1 注释设计 48](#_Toc426933552)

[4.13.2 测试用例设计 48](#_Toc426933553)

[4.13.3 限制条件 48](#_Toc426933554)

[附件1 用户权限分配表 50](#_Toc426933555)

# 引言

## 编写目的

详细设计的主要任务是设计每个模块的实现算法、所需的局部数据结构。详细设计的目标有两个：实现模块功能的算法要逻辑上正确和算法描述要简明易懂。预期读者为该系统开发人员，维护人员，测试人员等。

## 背景

该文档基于公车管理服务平台的开发需求编写。

## 术语定义

**API：**（Application Programming Interface,应用程序编程接口）是一些预先定义的函数，目的是提供应用程序与开发人员基于某软件或硬件得以访问一组例程的能力，而又无需理解内部工作机制的细节。

**Web API：**是网络应用程序接口。

**HTTP**：超文本传输协议（HTTP，Hyper Text Transfer Protocol)是互联网上应用最为广泛的一种网络协议。所有的WWW文件都必须遵守这个标准。

**JavaScript：**一种直译式脚本语言，是一种动态类型、弱类型、基于原型的语言，内置支持类型。为浏览器的一部分，广泛用于客户端的脚本语言。

**OBD：**OBD是英文On-Board Diagnostic的缩写，中文翻译为“车载诊断系统”。这个系统随时监控发动机的运行状况和尾气后处理系统的工作状态，一旦发现有可能引起排放超标的情况，会马上发出警示。

**序列化**：(Serialization)将对象的状态信息转换为可以存储或传输的形式的过程。在序列化期间，对象将其当前状态写入到临时或持久性存储区。以后，可以通过从存储区中读取或反序列化对象的状态，重新创建该对象。

**队列**：是一种特殊的线性表，特殊之处在于它只允许在表的前端（front）进行删除操作，而在表的后端（rear）进行插入操作和堆栈一样，队列是一种受限制的线性表。

**缓存**：是数据交换的缓冲区（称作Cache），当系统要读取数据时，会首先从缓存中查找需要的数据，如果找到了则直接复制，找不到的话则从其他介质中查找。缓存工作的原则，就是“引用的局部性”，这可以分为时间局部性和空间局部性。

**AJAX**：AJAX即“Asynchronous JavaScript And XML”（异步JavaScript和XML），是指一种创建交互式网页应用的网页开发技术。

**Servlet**：是 Java 编程语言中的一个类，它被用来扩展服务器的性能，服务器上驻留着可以通过“请求-响应”编程模型来访问的应用程序。虽然 Servlet 可以对任何类型的请求产生响应，但通常只用来扩展 Web 服务器的应用程序。

**JSON：**JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式。它基于ECMA Script的一个子集。 JSON采用完全独立于语言的文本格式，但是也使用了类似于C语言家族的习惯。这些特性使JSON成为理想的数据交换语言。 易于人阅读和编写，同时也易于机器解析和生成（网络传输速率）。

**Protocol Buffer：**是Google 的一种数据交换的格式，它独立于语言，独立于平台。由于它是一种二进制的格式，比使用 XML进行数据交换速度快。可以把它用于分布式应用之间的数据通信或者异构环境下的数据交换。作为一种效率和兼容性都很优秀的二进制数据传输格式，可以用于诸如网络传输、配置文件、数据存储等诸多领域。

## 参考资料

《计算机文化（第五版）》June jamrich Parsons，Dan Oja．

《计算机信息处理》Steven L. Mandell，Sachi Sakthivel

《计算机网络（第四版）》谢希仁

# 公车管理服务平台系统结构

## 系统结构图



OBD服务的作用是将kafka队列中的数据取出并进行处理的后台服务，该服务采用常驻后台的形式运行。Obd服务收到kafka队列中的数据后按顺序执行以下操作：

将数据存入数据库

更新redis中的实时数据

判定车辆是否有异常

更详细的操作流程见下3.2.1章节。

业务服务集群内部分割出多个业务模块，如业务模块、obd设备模块、监控模块等。整个系统以数据库为中心，从而驱动整个系统的数据运行。

数据库维护服务是对数据库的维护操作，主要是定期清理数据，转移历史数据等操作。

## 慧管车平台系统结构概述

慧管车平台业务服务主要分为前台展现层，业务支撑层和后台支持层。前台展现包括移动端和网页端两种服务。业务支撑层分为多个模块，每个模块都可以独立部署或成为分布式的服务集群。后台支持服务为前台提供实时的数据支持或数据库维护等多种维护功能。

慧管车平台数据库分为

* 业务数据库

存储主要的业务数据，包括用户、驾驶员、订单等等数据。

* 主数据库

存储基础数据、组织架构、车辆信息等变动较小的数据。

* obd数据库

存储从obd设备获取到的信息，包括位置、车辆状态、车辆警报等信息。

* 第三方数据库

存储第三方数据信息

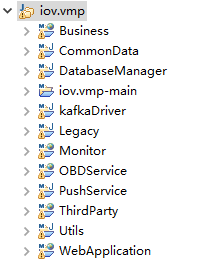
* 授权数据库

存储用户登录的授权信息

除了数据库和服务，平台内还有多个为提高服务质量存在的缓存服务、队列服务。

## 代码结构

项目代码结构如下图所示。将项目分为12个分项目，除一个顶层的配置项目外，包含11个功能性项目。每个项目的具体作用见下。



### iov.vmp-main

该工程是项目的父工程，在该工程中没有实际代码，但是包含了系统中的所有关键代码。同时，系统的部署脚本、安装脚本也存放于该工程文件夹下。该工程不生成任何可部署文件，所有的子工程都依赖于该工程以获得正确的配置。

### Business

主业务工程，该项目下包含了所有的接口代码，部分java数据结构文件也在该项目中。该项目依赖于CommonData，Utils，PushService三个代码工程。

整个项目的业务逻辑，为前台提供的接口服务都在该工程中实现，是整个系统的业务核心，包含了大部分的代码。

该工程最终以war包的形式发布并部署到http服务器上，本项目内使用tomcat服务器。

### OBDService

该项目提供OBD数据的查询，包括车辆实时位置、运动轨迹以及与OBD数据相关的统计等均在该工程中实现。该项目依赖于Utils，CommonData工程。

OBDService接受来自于Business的http请求，OBDService不直接提供对外的http接口服务，而只对系统内部的查询请求作出响应。

该工程最终以war包的形式部署为http应用服务。

### ThirdParty

第三方数据接口工程，该项目实现了服务于第三方的接口。其中之一是实时路况信息。实时路况信息通过http请求该项目中实现的接口，从而通过主动推送的方式将信息推送到系统中。

另一种是系统主动请求的信息，如ETC信息，通过该项目中的接口实现包装为本系统中统一的接口，再去请求第三方接口。该类接口只对系统内的项目开放。

该工程最终部署为war包。

### Utils

系统中共享的实用工具类。将常用的工具，如类型转换、坐标转换等相关的类统一到该工程下，从而整个项目都能使用。

### WebApplication

该工程为电脑客户端的实现。包含了电脑客户端所需的资源文件、脚本等文件。该工程依赖于Utils工程。

该工程部署为war包。

### KafkaDriver

该项目用于处理kafka队列中的数据，其中就包括来自于OBD设备的数据包。该项目依赖于Utils，CommonData工程。该项目将生成一个可运行的jar包，并在后台运行，从而可以实时处理OBD数据。

### DatabaseManager

数据库管理后台软件。该项目将生成一个可运行jar包，并在后台运行。通过定时启动来整理OBD数据库中的数据。将OBD数据库中的历史数据进行迁移。迁移策略详见下面详细设计。

### Monitor

系统监控项目，在该项目中实现了监控系统工作状态的接口，包括数据库、后台服务、系统软件等运行状态的监控。通过接入阿里云平台监控API，做到对系统中的部署机器的实时监控。

该项目生成可部署的war包，在webapplication中以链接的形式连接到该工程中。

### PushService

信鸽推送的相关服务代码，来自于信鸽官方的SDK。使用官方的http推送接口进行信息推送。具体的推送代码在business中实现，该项目仅作为依赖项目的形式存在于项目中。

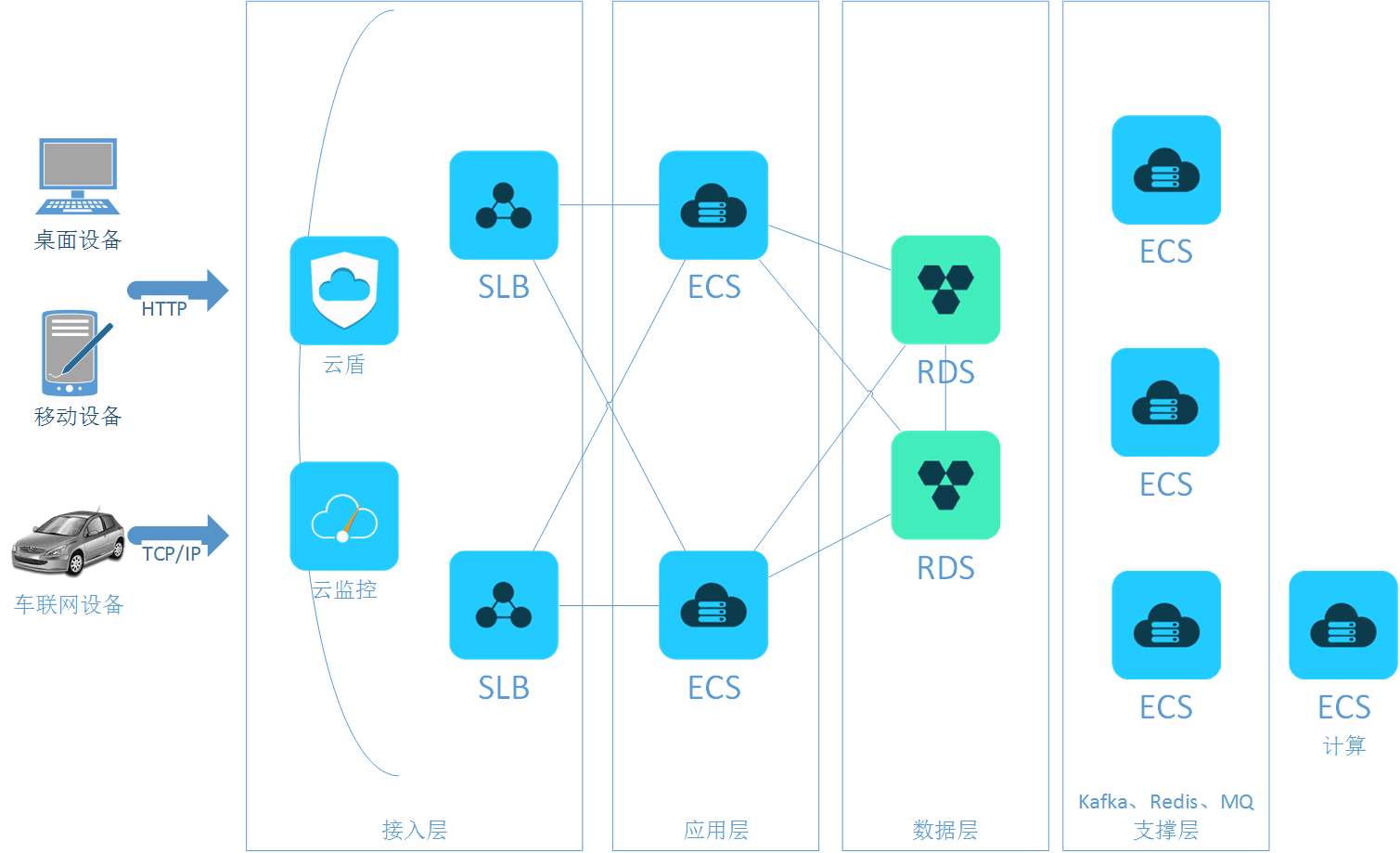
### CommonData

共享的数据类型工程。用户在工程之间共享部分数据结构定义，如关于车辆信息的OBD数据定义等。该项目被系统中的多个工程引用。

### 其他

项目文件夹下还包含多个工程，包括移动客户端appclient，遗留无效代码legacy，obd接入服务obdhub三个文件夹。

## 系统部署图



慧管车平台部署利用阿里云进行平台部署，我们申请了多个应用服务器，配合阿里云数据存储和负载均衡服务器，可方便地部署应用集群。

## 开放接口设计

系统的接口均采用开放式设计，我们的网页端和移动端作为应用接入开放接口。为保证系统的安全性，所有的接口均需要进行权限认证，权限认证方式采用OAuth2.0标准。

第三方平台接入我们的服务时，需先申请到平台认证，之后即可接入我们的后台服务，而根据其服务对象的不同，取得的数据也不一样。

# 公车管理服务平台角色定义

根据公车管理过程中出现的各类人员，我们将角色分为以下7种：平台管理员、机构管理员、机构领导用户、稽查管理员、公车管理员、公车使用者、公车驾驶员等用户角色，不同用户角色根据各自的业务需求，系统赋予不同业务定义和角色权限。同时各类用户角色内部，还可以根据实际业务需要，进一步划分次级权限。

系统将可以为同一个账号分配多个角色以便方便管理（见附件1）。对于不同的角色，所呈现的功能视图不同的，能修改和获取的数据也不一样。

在功能分配列表的基础上，根据各功能所需的数据库表格做进一步的权限管理，对不同的角色控制对数据表的读取或写入权限，从而保证系统数据的安全性，完整性。



## 运营平台管理员

核心管理层，对下面各层级拥有全部管理功能，定义和管理系统运营的关键参数，汇总和挖掘运营分析数据。同时负责各机构、用户申请信息的审批工作，信息录入等职能。

## 机构管理员

机构管理员是各机构的平台管理员，担负着本机构或授权机构内所有相关的权限操作。机构管理员对系统的运行管理有着较深的理解，能解决平台使用过程中产生的各种问题，或直接与运营平台管理员对接。

## 稽查管理员

各机构监管公车使用各项业务的用户。拥有更多的数据报表读取权限。各政府机构公车监管管理员可查询公车使用过程中的行程监管、费用监管和统计分析等功能，并可根据各机构自身的业务需求，定制相关的监管项目。

## 公车管理员

各部门负责公车管理的人员。公车管理员的管理内容涉及公车管理的各个方面，包括调度，日常维护，报修等多项管理职责。

## 公车驾驶员

各机构的公车驾驶员。公车驾驶员可查看自身的驾驶任务，去往地点、驾驶时间、加班记录等相关功能。同时，关于车辆使用过程中产生的费用，如油费、路桥费等需由驾驶员负责录入。

## 公车使用者

各类需求使用公车的用户。该用户可以申请该部门下的公车，管理个人信息和查看相关公车使用历史记录。

## 机构领导用户

政府机构分管公车业务的领导用户，拥有对本部门公车审批的权限。

# 慧管车平台详细设计

## 用户管理详细设计

### 用户权限认证详细设计



用户登录流程如上图所示，根据接口定义（见接口文档），客户端提供机构号、用户名和密码三个参数。密码在客户端进行md5加密后传输至后台。后台通过机构号和用户名查找对应的用户，并比对数据库中存储的md5密码值与前台传输的md5密码值，若对比成功，则将该用户的登录记录存储至授权数据库中，并返回用户的令牌。授权令牌的有效期被设为30天，以满足客户端维持登录的要求。

登录和获取用户信息被分为两个接口，在获取登录令牌后可通过令牌获取当前登录的用户信息。在以后的接口调用中需要授权时必须附带该令牌以便服务器获取用户信息。

#### 输入

该设计的输入是用户的机构号和登录名，登录名和机构号以@符分割，形式为“登录名@机构号”。为保证安全性，在传输前密码需经过md5加密。

#### 输出

该逻辑输出的是用户的授权令牌，由32位uuid标示。在以后的接口访问中该令牌可标识用户身份。使用方法是在http请求头中添加Authorization字段，值为Bearer <token>的形式。

#### 对应接口

见接口文档0.1

### 用户session保持与令牌管理



用户的session存储在redis缓存中。多个网页服务器连接到一台redis缓存可以达到共享session的目的。当网页集群中某台机器时效时，其他机器仍然可以对该用户进行服务。

授权服务是保证系统数据安全的服务器，授权服务根据用户名密码生成相应的令牌并存储在数据库中。

Session时效时间和令牌失效时间均为30天。

#### 输入

该过程的输入是用户的授权信息，包括用户名和密码。用户名为“用户的登录id@机构号”的形式。该形式在前台进行拼装。

#### 输出

该接口的输出是一个令牌，在以后的接口访问中在HTML头部带上该令牌即可知道用户身份和相应的权限。

### 组织架构详细设计



组织架构由平台管理员和机构管理员管理，平台需测试登录用户是否有操作权限。若有操作权限，用户可以添加或修改部门信息。

机构是由平台管理员分配的特殊部门，该部门的管理员即该机构的机构管理员，拥有该机构内最高的用户权限。

#### 输入

部门基本信息，名称、所属部门、部门排序等。平台管理员有权限操作所有的组织机构，机构管理员可以操作本机构内的组织架构。

#### 输出

操作结果，成功或失败。

#### 对应接口

见接口文档0.2；0.3；0.4及第5章节。

### 用户管理详细设计



机构管理员的用户管理分为四种角色，根据角色的不同，用户具体信息有所区分。所有的用户都存储在用户表中，根据角色不同，他们的具体信息将存储在职工表、驾驶员表中。

#### 输入

用户的基础信息，根据进一步的提示，输入对应角色的必选信息和可选信息。

#### 输出

操作结果，成功或失败。

#### 对应接口

见接口文档第1章节

### 驾驶员信息管理详细设计



驾驶员是比较特殊的一类用户，除了基本的用户信息、职工信息、驾驶员还包含了驾驶技能的相关信息。如驾龄、领证时间等，也有关于驾驶员工作的信息，如年审时间等。通过对这些信息的管理从而更好地管理驾驶员。

#### 输入

驾驶员管理的输入是在用户管理的基础上添加了驾驶员特有的相关信息。

#### 输出

操作结果，成功或失败。

#### 对应接口

见接口文档第4章节。

## 车辆管理详细设计

### 实时位置详细设计



车辆实时信息的上报涉及到obd与车辆的绑定信息、车辆实时信息的存储和区域管理等过程。具体的信息上报流程如上图所示。信息上报完成后Redis数据缓存中即包含了最新的车辆状态和位置数据。

#### 输入

实时位置输入为obd上报的实时位置信息，包含静态信息、动态信息和相关报警信息。这些信息存储在kafka队列中，由服务程序从kafka队列中读取出来并解包处理。

#### 输出

该过程的输出包含多种类型。会将相关信息存入数据库，obd的消息会推送至移动端。

#### 对应接口

见接口文档3.4节

### 车辆日常信息管理详细设计



车辆日常信息管理包含车辆日常使用过程中的各项处理，如维修、保养、年审、保险、事故、封存启用、车辆处置等信息。车辆的这些基本信息的管理在系统的详细设计上采用类似的方式。由车管员对相关的信息进行维护、修改。

#### 输入

车管员首先选择需要管理的车辆信息类型，在对应界面中输入录入相关信息并保存。后台将相关数据存入数据库备查。

#### 输出

操作结果，成功或者错误类型。

### 人车匹配



进行人车匹配的绑定，将车辆和司机绑定在一起，形成对应关系，并可查看匹配历史记录。人车匹配后，该车和驾驶员将不再可以通过订单调用，对该车的实时监控消息也将不再推送。但是车辆的信息仍将记录。

#### 输入

该流程的输入是车辆的唯一标识ID和驾驶员的唯一标识ID。车辆与驾驶员的绑定记录在数据库中。

#### 输出

输出为该操作的结果，成功或失败。

### 保险管理



车辆保险管理由系统自动进行检查提醒，当保险到期时，系统会提示车管员进行保险信息的更新。填写保险信息和下次保险到期时间。

#### 输入

该流程的输入为保险的相关信息，如保险费用，保险到期时间等。

#### 输出

输出为消息提醒，当到期时会产生消息。

### 封存启用



车辆封存后即无法继续使用，系统也不再对其状态进行记录和判断。只有启用车辆后车辆才恢复使用。

#### 输入

本过程的输入是车辆的启用或封存状态。

#### 输出

输出为操作结果，成功或失败。

## 调度管理详细设计

### 订单录入详细设计



用户申请订单时，根据提示逐步输入订单所需信息，包括目的地，上车地点，预计用车时间，用车数量等必须信息。同时可以输入随车人员、用车需求等。

#### 输入

输入为订单的信息，根据不同的信息类型，输入方式有所区别，具体请见上。

#### 输出

操作成功则输出新建的订单ID，操作失败则输出失败信息。

### 工作流详细设计



对于用车订单的工作流，我们采用了部分可配置的设计方案。整个工作流分为四个大流程：申请、审批、分配车辆、执行。其中，审批流程可指定多个机构领导进行层层审批。公车管理员可以根据用户期望的车辆数和实际情况进行配车。订单的流程根据各部门的实际情况设定，每个部门有一个单独的流程配置。

#### 输入

输入由订单当前的状态决定。当订单建立后，机构领导可以输入订单的审批意见，并确定是否允许订单。当订单被允许后，根据各部门的流程设定可进行下一层领导审批或到达公车管理员等待派车。

公车管理员的输入是分配的车辆信息，当司机被分配任务后可在自己的订单中列表中看到信息。

司机的输入是订单的执行情况，确认上车、行车记录等。最终由司机结束该订单。

#### 输出

该流程的每一步输出均为表示操作成功与否的信息。

### 行车记录详细设计



车行的行车记录是指车辆使用过程中产生的各项支出或事件的记录。由当前使用该车的驾驶员负责记录。驾驶员点击开始订单后，可以查看、添加、编辑行车日志，在完成前，驾驶员可以随时修改已经记录的行车记录。此时车管员看不到驾驶员记录的行车记录。

订单完成时，提醒驾驶员是否确认完成此订单。若完成，驾驶员不再可以修改查看相关订单，车管员此时可以查看驾驶员提交的订单。

#### 输入

根据驾驶员的操作，可以是新订单信息或原有订单的修改信息。

#### 输出

操作结果，成功或失败。

#### 对应接口

见接口文档3.4，3.5。

### 区域管理详细设计



本系统提供对车辆位置的实时判定，以确定车辆是否违规或越界行驶。设置区域的流程图如上图所示。用户可根据需要设置区域的生效时间和触发方式。触发方式包括驶入触发、驶出触发、驶入驶出触发三种。

#### 输入

输入为区域的定义，包括生效时间，触发方式，名称，监控车辆等信息。

#### 输出

后台收到车辆的GPS信息后即快速对GPS信息进行检查。若符合触发方式，则向消息系统推送一条消息告知相关用户车辆越界消息。

## 地图管理详细设计

#### 地图数据处理

地图数据主要是位置数据，我们采用百度地图作为我们的地图提供商。百度地图采用自主设计的一套地理位置坐标，我们的系统采用统一的全球GPS坐标存储。为保持数据的一致性，需要对坐标进行相应的转换。



#### 输入

坐标值，由obd上报的数据为GPS全球坐标。从前台传输的数据是百度坐标。根据不同的数据来源，对坐标做不同的处理后入库。

#### 输出

根据不同的接口输出不同。

#### 相应接口

见接口文档第3章。

## 事件提醒详细设计



本系统的消息系统分为两大部分，消息的检查和消息推送。其中消息检查又分为实时消息检查和定时消息检查两种。实时消息来自于obd、第三方等数据源，当数据获得时即对数据进行处理并产生相应的消息，产生消息后推送至消息推送系统，由消息推送系统负责具体的消息存储、处理、推送等业务。

#### 输入

该系统的输入由多个来源生成，如obd的上报，第三方接口的上报，用户输入等。根据不同的来源，系统采用不同的处理方式。具体的实现请见以下两节。

#### 输出

该系统是一个消息系统，所产生的输出均为消息。这些消息会存入数据库备查，同时根据用户的设置推送或不推送。

### 实时消息检查



实时消息是有车载obd设备产生的，包括GPS数据、车辆工况以及提醒消息。其中，对于提醒信息，消息系统直接将信息放入推送系统作进一步处理，而对于未知信息，系统会获取对应的区域检测信息，并比对当前的车辆状态，从而产生区域报警消息。

#### 输入

Obd产生的提醒消息或实时GPS数据。

#### 输出

对应到车辆的提醒消息。该消息包括车辆信息，提醒的具体内容，所产生的数据等信息。

### 定时消息检查



定时消息检查常驻后台，以半小时为周期进行检查。对于不同的提醒类型，如保险、年审等等信息分散到不同的时间段检查。

#### 输入

车辆的年审、保险、维护等定时信息。并从系统中获取当前的实际时间。

#### 输出

提醒消息，并将该提醒消息推送到消息处理系统进行处理。

### 消息系统详细设计



消息系统对不同来源的消息进行统一处理。主要的作用是查找消息的责任人，整理消息为方便阅读的形式，决定是否推送实时消息。

#### 输入

不同的消息类型都可以发送到消息系统内，使用的接口采用webapi形式。

#### 输出

将消息整理后推送到移动客户端。网页端可通过查询数据库的方式获取。

### 应急救援详细设计



应急救援是为驾驶员提供的紧急救援接口。驾驶员通过两次点击方便地将紧急信息推送到系统中，并将消息推送给公车管理员。由公车管理员进行处理。

#### 输入

该接口的输入均为可选输入，驾驶员在紧急情况下不需要任何输入。客户端可自动获取用户的实时位置并传输至后台。

#### 输出

驾驶员的紧急信息将推送到驾驶员对应部门的车管员移动客户端。车管员处理后消息反馈给驾驶员。

## 设备管理详细设计

### OBD设备管理详细设计



Obd设备管理主要包含obd的注册和与车辆的绑定。如上图所示，首先进行obd设备的注册，若已经注册，则可以直接选择需要绑定的车辆。当车辆与obd绑定后，即可根据obd的信息查找到对应的车辆信息。

#### 输入

当需要注册obd时，首先需要输入obd的详细信息。完成后即可选择车辆进行绑定。

#### 输出

操作结果，成功或失败。

#### 相应接口

见接口文档第7章节。

### OBD数据分表详细设计



obd分表策略是为防止obd数据表中单表数据过大设计的分表策略，以帮助提高数据存储和查询的速度。数据库分表程序每天的2:00执行，此时是服务器负载最轻的时间。程序启动后即将后几天的数据表进行转移，以为第二天的数据入库准备。

#### 输入

这个系统没有输入

#### 输出

系统启动后会将obd数据表进行拷贝转移，会生成相应的数据表。

## 统计分析详细设计

### 统计表详细设计



统计图和统计表均采用统一的查询方式。由于统计图表的统计需求灵活性较高，因此需根据具体情况生成相应的查询语句。

#### 输入

不同的查询条件

#### 输出

使用json格式生成的查询结果。

### 统计图详细设计

统计图采用与统计表相同的查询接口。当统计返回时，我们采用百度echart库生成相应的图表，包括直方图、折线图、饼图等。

## 第三方接口详细设计

### 信息交换接口

平台应具备与其他系统的信息交换功能，包括车辆动态信息、静态信息、监管信息、统计分析信息等的交换。

为提升公车监管与服务平台用户体验，平台可为用户提供及时、准确和详实的其他信息服务，可以整合跨行业跨部门的信息化系统数据。

1. 交调、卡口数据。
2. 运输管理数据。
3. 气象信息。
4. 其他。

### 数据共享接口

平台设计即以开放性为前提，使用oauth授权方式授权后，根据不同的授权等级，第三方程序可以访问不同级别的接口。

## 稽查管理详细设计

稽查管理功能主要为各政府机构公车稽查管理员提供公车使用过程中的行程稽查、费用稽查和专题稽查等功能，并可根据各机构自身的业务需求，定制相关的稽查项目。

### 行程稽查



行程稽查主要是检查车辆的行驶过程与系统中的记录是否符合，该稽查主要是找出违规用车的记录从而杜绝违规用车的情况。

### 费用稽查



### 专题稽查

（待补充）

## 运营管理需求

运营管理需求主要是指系统在运行过程中所需的各项管理功能需求。拥有运营平台管理员的角色才能对改系统做相应操作。

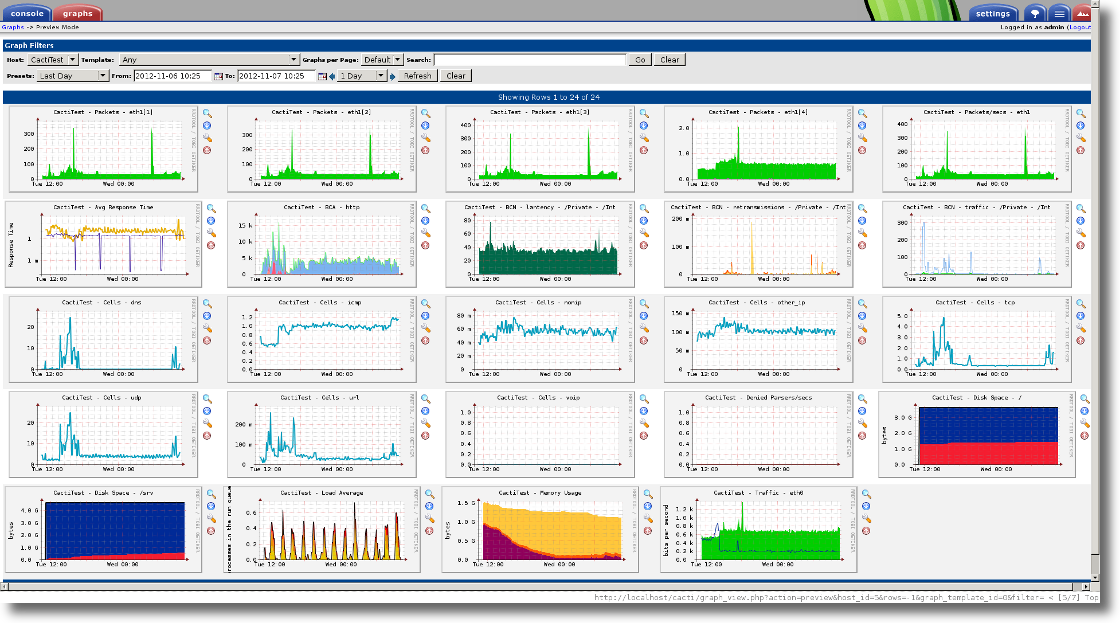
### 系统运行监控

信息中心对各系统的运行情况进行监控，当系统运行出现异常时及时报警，方便工作人员的及时处理，保证系统的正常运行。

信息中心也可以展示系统运行的各项参数，包括当前服务器负荷、登陆用户总量、网络数据总量。并能同时监控系统运行中各项服务的状态，如数据库、百度地图服务等等，从而及时发现系统问题，避免经济损失。

数据库监控包括数据库安全性监测、数据库容量监测、数据库的备份与恢复等。数据库安全性监测指系统对数据库的安全性进行监控，如数据库sa空密码报警、数据库密码上次修改的时间、数据库用户及权限等；数据库容量监测指系统对数据库的容量、大小进行监测；数据库的备份与恢复指系统数据库需要定期进行维护、备份，系统将对这些数据库操作进行监控，另外在数据库故障时能够主动恢复。

实时监测系统内各节点的网络通讯情况，主要有应用服务器之间的通讯、应用服务器与存储设备的通讯、应用服务器与各前置机间的通讯、前置机与车载终端间的通讯等。网络通讯的监测包括以下内容：各个节点的网络地址设置情况、各个节点网络通讯中断情况、各个节点网络通讯成功率统计等。





### 组织架构管理

维护全国性的组织架构树，根据国家的部门设定、车管部门级别以及监管层级设置组织架构。采用国家统计局规定的全国各部门组织的唯一编码，根据组织架构，建立树形组织菜单。也可根据需要添加或删除相应的部门以及为相应部门分配账号。同时，在相关部门注册时即对部门内的车辆予以登记注册。

组织架构管理采用树形管理，有唯一的全国根节点，根据实际情况下面添加多层根节点。从而适应更复杂的应用情况。

1）机构基础信息管理：进行静态机构数据维护，包括机构名称、编号、用户容量、服务内容设置等内容。当机构信息新增或变更时，机构管理员核实相应信息无误后，上报给公车监管与服务平台管理员，通过平台审核，完成相应的数据操作。

2）机构用户信息管理：进行机构用户关联操作，包括机构所辖用户规模规划、申请审批、管理员设置等内容。机构用户信息变更，主要包括两类情况：一是单一用户信息新增和变更，由单一用户向机构管理员用户发出用户申请或变更请求，机构管理员核实后上报平台管理员；二是批量用户新增、变更，主要发生于新机构加入平台时，由机构管理员统一汇总上传至公车监管与服务平台，由平台管理员负责审核相关信息，并完成后台数据操作。

机构管理的界面如下图所示：



机构管理员身份登陆，在“组织机构”中 通过点击“新增部门”或者右击机构名称来增加下级部门，并可以编辑本机构的基本信息。

可以在组织机构中对下级部门进行编辑、删除和详情查询，并可以按照部门名称、部门联系人、联系电话和所属部门等条件进行查询。

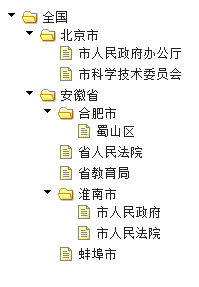


图.1树形菜单预览

在以地域划分的基础上，还提供根据交管部门和监管部门分配的垂直管理结构。这两种结构独立于按地域划分的组织架构，拥有其下辖相关部门公车的管理和监管的权限。

### 日志管理

日志管理记录系统的各种操作日志信息，主要包含平台日志和系统日志两种类型的日志。对于平台日志，主要记录平台运行过程中产生的信息和问题，如用户登录信息、参数修改信息等。日志管理包括日志查询、日志导出、日志清除等方面功能。而对于系统日志，主要记录系统运行过程中的各项参数、问题。如系统启动、关闭时间，各服务运行状态和日志等。

对于日志系统，需要满足日志存储的安全性和可查性，即使在系统遭受破坏的情况下依然能读取日志，从而定位问题。

系统日志统一记录在服务器中的某个目录中，根据具体的项目名记录不同的日志。

### 系统版本更新与管理

由于系统发布后需要进行不断的升级更新，需要对系统的更新版本进行管理，采用自动或者人工的方式对系统进行更新，保障系统的正常运行。

版本需采用三段命名法，即[版本号.功能版本号.补丁版本号]的方式发布系统版本。其中，版本号为系统整体版本号，功能版本号指示不同版本的不同功能级别，当有新功能添加或旧功能删除、升级时应当升级。补丁版本号为系统补丁发布的版本，对没有功能该表仍应该发布的版本应当升级该版本号。

## 移动应用（APP）功能设计

移动APP端包含Android和iOS两个平台的移动应用，两个应用均采用HTML5架构，并在用户登录时检测用户的角色，从而根据用户角色跳转至相应的角色页面。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 角色 | 模块 | 子模块 | 功能点 | 功能描述 |
| 使用者APP | 用车申请 | 用车申请 | 用车申请 | 用车申请 |
| 我的申请 | 待办任务 | 待办任务 |  |
| 在办任务 | 在办任务 | 驾驶员未确认前提下可申请取消 查看车辆位置 |
| 已办任务 | 已办任务 |  |
| 系统设置 | 系统设置 | 系统设置 | 帐号的设置 |
| 司机APP | 上/下车 | 上下车 | 上下车 | 当司机未绑定车辆时，显示上车，绑定时显示下车 |
| 我的任务 | 我的任务 | 待办任务 |  |
| 在办任务 | 在办任务 |  |
| 已办任务 | 已办任务 | 查看流水和车辆轨迹 |
| 我的车 | 车况检测 | 车况检测 | 如采用人车绑定，司机绑定后可检测当前车况和查看历史车况查询记录，如未采用人车绑定，司机随时可进行检测 |
| 维修保养 | 维修申请 | 向平台发送维修申请单 |
| 零件领用 | 发送轮胎、电池等领用申请 |
| 保养申请 | 向平台发送保养申请 |
| 我的车 | 我的车 | 司机绑定车辆后查看车辆位置等信息 |
| 信息提交 | 信息提交 | 信息提交 | 司机下车后对用车过程中产生的各项税费或信息予以确认提交。 |
| 消息提醒 | 消息提醒 | 消息提醒 | 当前任务的推送提醒 |
| 系统设置 | 系统设置 | 系统设置 | 帐号的设置 |
| 管理员/部门管理员 | 企业车辆 | 车辆定位 | 在地图上显示车辆的位置 | 先通过部门树选择车辆。当某个区域的车辆数较多，无法区分显示时，可聚合显示，只显示总车辆数即可。 |
| 车辆跟踪 | 选择车辆进行实时跟踪 | 对指定车辆进行实时跟踪 |
| 轨迹查询 | 查询车辆轨迹 | 查询车辆轨迹 |
| 统计报表 | 统计报表 | 统计报表 | 显示各部门及车辆的详细报表 |
| 车务管理 | 派单管理 | 派单管理 | 用车申请的审批，查看任务状态，查看任务轨迹等 |
| 事件提醒 | 事件提醒 | 事件提醒 | 车队各类报警事件的提醒 |
| 设置 | 设置 | 设置 | 包括提醒信息的设置，分享，帮助等内容 |



### 公车使用者

#### 订单申请

订单申请的具体流程参考3.3.2节。用户首页即是订单申请页面，可以快速申请公车使用。

#### 订单管理

用户的历史订单，当前订单等通过订单管理进行查看。

#### 个人信息

用户的个人信息管理，包括手机号，头像等。

### 部门领导

部门领导的功能在公车使用者的基础上增加了订单审批功能。界面上两个角色的设计是一致的。

部门领导在订单管理页面除了可以看到自己的订单外，还能管理本部门的申请订单。可以对订单进行快速的批复，也可以查看订单详情并进行批复。

### 公车管理员

#### 车辆派遣



车管员可对已经经过审批的订单进行车辆派遣。车辆的派遣流程如上图所示，车管员对某个订单总是先选择一辆车，再分派一个司机。根据一个订单申请车辆数的不同，可以多次选择车辆和驾驶员。

#### 车辆管理

移动端对车辆的管理不能做到如PC端那么精细，移动端的车辆管理主要涉及到车辆的位置，车辆的状态等信息。

移动端车辆管理主要包含以下几个方面



#### 驾驶员管理

移动端主要是查看驾驶员当前的状态和相关信息。

#### 统计分析

移动端可查看当前系统的统计分析图，包括：

* 行驶里程
* 总油耗
* 平均油耗
* 总油费
* 用车时间
* 车机拆除
* 怠速超长
* 越界
* 非调度用车
* 车辆违章
* 车辆超速

## 车载终端设备

### OBD设备

#### 外部接口

* OBD标准接头

用于连接车载标准16 Pin 诊断接口。

车载OBD系统通过此接口可以和遵循同一通信协议的外部设备通信。

* Mini USB接口

通过USB设置线与电脑连接。

* SIM卡座

此接口用于安装SIM卡。

#### 技术参数

|  |  |
| --- | --- |
| 存储 | ≥2MB FLASH, 最少可存储20000条GPS数据 |
| 数据传输方式 | GPRS/SMS |
| 定位方式 | GPS/A-GPS |
| 诊断协议 | SAE J1850 PWM  SAE J1850 VPW  ISO 9141-2  ISO 14230-4  ISO 15765-4  SAE J1939（商用车）  SAE J1587/J1708（商用车） |
| Power工作电压 | 9-36VDC |
| 工作电流 | 平均工作电流: <150mA@13.8/27.6VDC  最大工作电流: <200mA@13.8/27.6VDC  待机工作电流: <10mA@12/24VDC |
| 备用电池 | 3.7V/160mA 锂电池 |
| 三轴加速传感器 | +/-2g、+/-4g、+/-8g、+/-16g |
| GPS | 通道: 50  灵敏度: -160dBm  定位精度: 5m CEP  冷启动时间: <32s  暖启动时间: <32s  热启动时间: <1s |
| GSM | 工作频率: 850/900/1800/1900MHz  通讯协议: TCP/IP  灵敏度: -107dBm@850/900MHz |

### ETC设备

ETC设备包括OBU和用户卡，车辆需安装符合《全国高速公路电子不停车收费联网工作总体技术方案》的设备。

## 其他

### 注释设计

一份完整的代码应至少包含以下几部分注释：

1. 头部的版权申明，包括文件创建时间，工程名等。
2. 该类的作者
3. 其他javadoc约定的注释

注释范例如下所示：



除以上必要注释外，公共方法、公共变量都应该加上该变量的具体功能描述，以便代码的后期查阅，修改。

### 测试用例设计

对于关键的接口、方法都需要编写相应的测试用例。测试用例的编写应当不影响实际系统的运行、能充分测试各种可能出现的情况。

### 限制条件

开发语言的限制：本系统的开发基于Java语言，使用spring开发框架。对于系统的后期扩展与修改，只能在该语言的基础上进行修改扩充，无法使用其他的开发语言进行修改。

运行平台限制：项目开发基于Linux系统进行，对于Windows下的系统运行状态未做充分测试，可能会出现兼容性问题。

# 用户权限分配表

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 功能与权限 | | | | 管理员 | | | | 用户 | | |
| 功能分类 | 序号 | 权限列表 | 具体权限 | 运营平台  管理员 | 机构  管理员 | 稽查  管理员 | 公车  管理员 | 公车  驾驶员 | 公车  使用者 | 机构  领导用户 |
| 运营管理 | 1 | 设置系统参数 | 读取 | √ |  |  |  |  |  |  |
| 设置 | √ |  |  |  |  |  |  |
| 2 | 部门管理 | 读取 | √ | √ |  | √ |  |  |  |
| 写入 | √ |  |  |  |  |  |  |
| 3 | 公车管理 | 读取 | √ | √ | √ | √ |  |  |  |
| 写入 | √ |  |  |  |  |  |  |
| 4 | 终端设备管理 | 读取 | √ | √ | √ | √ |  |  |  |
| 写入 | √ |  |  |  |  |  |  |
| 调度管理 | 5 | 申请用车 | 读取 | √ |  |  | √ |  | √ |  |
| 设置 | √ |  |  | √ |  | √ |  |
| 6 | 审批用车 | 读取 | √ | √ |  |  |  |  | √ |
| 设置 | √ | √ |  |  |  |  | √ |
| 7 | 派单管理 | 读取 | √ | √ | √ | √ | √ | √\* |  |
| 设置 | √ | √ |  | √ |  |  |  |
| 8 | 轨迹回放 | 读取 | √ | √ | √ | √ |  |  |  |
| 写入 | √ | √ |  |  |  |  |  |
| 设置 | √ | √ |  | √ |  |  |  |
| 9 | 车辆定位 | 读取 | √ | √ | √ | √ |  |  |  |
| 写入 | √ | √ |  |  |  |  |  |
| 设置 | √ | √ |  | √ |  |  |  |
| 监管 | 10 | 数据报表 | 读取 | √ | √ | √ | √\* |  |  |  |
| 设置 | √ | √ | √ | √\* |  |  |  |
| 11 | 区域设定 | 读取 | √ | √ | √ | √ |  |  |  |
| 设置 | √ | √ | √ | √ |  |  |  |
| 12 | 绕道提醒 | 读取 | √ | √ | √ | √ |  |  |  |
| 设置 | √ | √ | √ | √ |  |  |  |
| 13 | 其他相关设置 | 读取 | √ | √ | √ | √ |  |  |  |
| 写入 | √ | √ |  | √ |  |  |  |
| 事件提醒 | 14 | 事件提醒 | 读取 | √ | √ | √ | √ |  |  |  |
| 设置 | √ | √ | √ | √ |  |  |  |
| 费用管理 | 15 | 油耗管理 | 读取 | √ | √ | √ | √ | √ |  |  |
| 写入 | √ | √ |  | √ | √ |  |  |
| 设置 | √ | √ |  | √ |  |  |  |
| 16 | 通行费管理 | 读取 | √ | √ | √ | √ | √ |  |  |
| 写入 | √ | √ |  | √ | √ |  |  |
| 设置 | √ | √ |  | √ |  |  |  |
| 17 | 交通罚款管理 | 读取 | √ | √ | √ | √ | √ |  |  |
| 写入 | √ | √ |  | √ | √ |  |  |
| 设置 | √ | √ |  | √ |  |  |  |
| 18 | 年审管理 | 读取 | √ | √ | √ | √ |  |  |  |
| 写入 | √ | √ |  | √ |  |  |  |
| 设置 | √ | √ |  | √ |  |  |  |
| 19 | 保险管理 | 读取 | √ | √ | √ | √ |  |  |  |
| 写入 | √ | √ |  | √ |  |  |  |
| 设置 | √ | √ |  | √ |  |  |  |
| 人车管理 | 20 | 机构角色管理 | 添加 | √ | √ |  |  |  |  |  |
| 设置 | √ | √ |  |  |  |  |  |
| 21 | 驾驶员信息 | 读取 | √ | √ | √ | √ | √ |  |  |
| 写入 | √ | √ |  | √ | √ |  |  |
| 注册 | √ | √ |  | √ |  |  |  |
| 22 | 人车匹配 | 读取 | √ | √ | √ | √ | √ | √ |  |
| 写入 | √ | √ |  | √ |  |  |  |
| 23 | 车辆基本信息 | 读取 | √ | √ | √ | √ | √ | √ |  |
| 写入 | √ | √ |  | √ |  |  |  |
| 24 | 车辆处置 | 读取 | √ | √ | √ | √ |  |  |  |
| 写入 | √ | √ |  | √ |  |  |  |
| 25 | 维修保养 | 读取 | √ | √ | √ | √ |  |  |  |
| 修改 | √ | √ |  | √ |  |  |  |
| 添加 | √ | √ |  | √ | √ |  |  |
| 26 | 驾驶员考评 | 读取 | √ | √ | √ | √ | √ |  |  |
| 写入 | √ | √ |  | √ |  |  |  |
| 27 | 驾驶员行为 | 读取 | √ |  | √ |  |  |  |  |
| 写入 | √ |  | √ |  |  |  |  |
| 地图管理 | 28 | 地图管理 | 读取 | √ | √ |  |  |  |  |  |
| 写入 | √ | √ |  |  |  |  |  |