*[打印时忽略本页]*

**变更履历**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 修订类别 | 简要说明 | 日期 | 变更人 |
| 1.0 | C | 1.0版本 | 2019-10-12 | 龙小南 |

修订类别：C = 创立，A = 增加，M = 修改，D = 删除

【云租宝】

**概要设计说明书**

广东省智瑞科技有限公司

**目 录**

[阅读对象 0](#_Toc430936835)

[版本历史 0](#_Toc430936836)

[目 录 0](#_Toc430936837)

[1 引言 0](#_Toc430936838)

[1.1 编写目的 0](#_Toc430936839)

[1.2 术语定义 0](#_Toc430936840)

[1.3 引用标准 0](#_Toc430936841)

[2 总体设计 0](#_Toc430936842)

[1.1 运行环境 0](#_Toc430936843)

[2.1.1 开发环境 0](#_Toc430936844)

[2.1.1.1 硬件环境 0](#_Toc430936845)

[2.1.1.2 软件环境 0](#_Toc430936846)

[2.1.2 生产环境 0](#_Toc430936847)

[2.1.2.1 硬件环境 0](#_Toc430936848)

[2.1.2.2 软件环境 0](#_Toc430936849)

[2.2 系统部署架构 0](#_Toc430936850)

[2.3 技术体系结构 0](#_Toc430936851)

[2.4 系统架构 0](#_Toc430936852)

[4 功能模块设计 0](#_Toc430936853)

[4.1 通讯录信息维护 0](#_Toc430936854)

[4.1.1 个人照片上传 0](#_Toc430936855)

[4.1.2 设计思路 0](#_Toc430936856)

[4.1.2.1 程序描述 0](#_Toc430936857)

[4.1.2.2 权限分配 0](#_Toc430936858)

[4.1.2.3 数据结构 0](#_Toc430936859)

[4.1.2.4 详细功能 0](#_Toc430936860)

[4.1.2.5 实现与接口 0](#_Toc430936861)

[5 外部系统接口 0](#_Toc430936862)

[5.1 接口分类 0](#_Toc430936863)

[5.2 接口性能要求 0](#_Toc430936864)

[5.3 接口功能要求 0](#_Toc430936865)

[接口关系概况 0](#_Toc430936866)

[5.4 接口列表 0](#_Toc430936867)

[6 数据库设计 0](#_Toc430936868)

[6.1 数据库规划 0](#_Toc430936869)

[6.2 数据库产品 0](#_Toc430936870)

[6.3 数据库对象命名规则 0](#_Toc430936871)

[6.4 数据表描述 0](#_Toc430936872)

[7 设计约定与要求 0](#_Toc430936873)

[7.1 开发规范 0](#_Toc430936874)

[7.1.1 开发工具 0](#_Toc430936875)

[7.1.2 命名规范 0](#_Toc430936876)

[7.1.2.1 包（Package） 0](#_Toc430936877)

[7.1.2.2 类（Class） 0](#_Toc430936878)

[7.1.2.3 接口（Interface） 0](#_Toc430936879)

[7.1.2.4 方法（Method） 0](#_Toc430936880)

[7.1.2.5 变量（Variables） 0](#_Toc430936881)

[7.1.2.6 常量 0](#_Toc430936882)

[7.1.3 注释 0](#_Toc430936883)

[7.2 程序结构 0](#_Toc430936884)

[7.3 配置管理 0](#_Toc430936885)

[7.4 日志输出log4j 0](#_Toc430936886)

[7.5 版本要求 0](#_Toc430936887)

[7.6 其他共性要求 0](#_Toc430936888)

[8 附录Ⅰ-参考文档 0](#_Toc430936889)

# 引言

## 编写目的

## 术语定义

|  |  |
| --- | --- |
| **术语** | **术语解释** |
| OPENAPI | 平台所提供标准开放API |

## 引用标准

|  |  |  |
| --- | --- | --- |
| **标准名称** | **制订者** | **备注** |
|  |  |  |

# 总体设计

## 运行环境

### 开发环境

#### 硬件环境

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **产品描述** | **数量** | **备注** |
| 1 | **负载均衡服务器** | 1台负载均衡服务器，Nginx应用；虚拟机，划分资源(4CPU、8G) | 1 | 虚拟机 |
| 2 | **数据库服务器** | 1台数据库服务器部署Mysql数据库；虚拟机，划分资源(4CPU、8G、500G硬盘) | 1 | 虚拟机 |
| 3 | **应用服务器** | 2台应用服务器，分别WEB平台、APP服务，虚拟机 ，划分资源(4CPU、8G) | 2 | 虚拟机 |
| 4 | **缓存服务器** | 1台缓存服务器，Redis应用；虚拟机，划分资源(4CPU、8G) | 1 | 虚拟机 |

#### 软件环境

|  |  |
| --- | --- |
| **软件环境** | **说明** |
| 操作系统 | CentOS7.2 |
| 应用服务器 | Tomcat8 |
| JDK | JDK1.8 |
| 数据库 | Mysql |
| 缓存服务 | Redis |

### 生产环境

#### 硬件环境

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **产品描述** | **数量** | **备注** |
| 1 | **WEB应用服务器** | 2台应用服务器，WEB服务、同步服务；划分资源(8核/16G/500G)；负载模式 | 2 | 虚拟机 |
| 2 | **APP应用服务器** | 2台应用服务器，APP服务；划分资源(8核/16G/500G)；负载模式 | 2 | 虚拟机 |
| 3 | **数据库服务器** | 2台应用服务器，Mysql服务；划分资源(8核/16G/1T)；主从模式 | 2 | 虚拟机 |
| 4 | **缓存服务** | 1台应用服务器，Redis服务；划分资源(4核/8G/500G)；主从模式 | 1 | 虚拟机 |
| 5 | **负载均衡服务** | 2台应用服务器，WEB服务、同步服务；划分资源(4核/8G/500G)；VIP模式 | 2 | 虚拟机 |

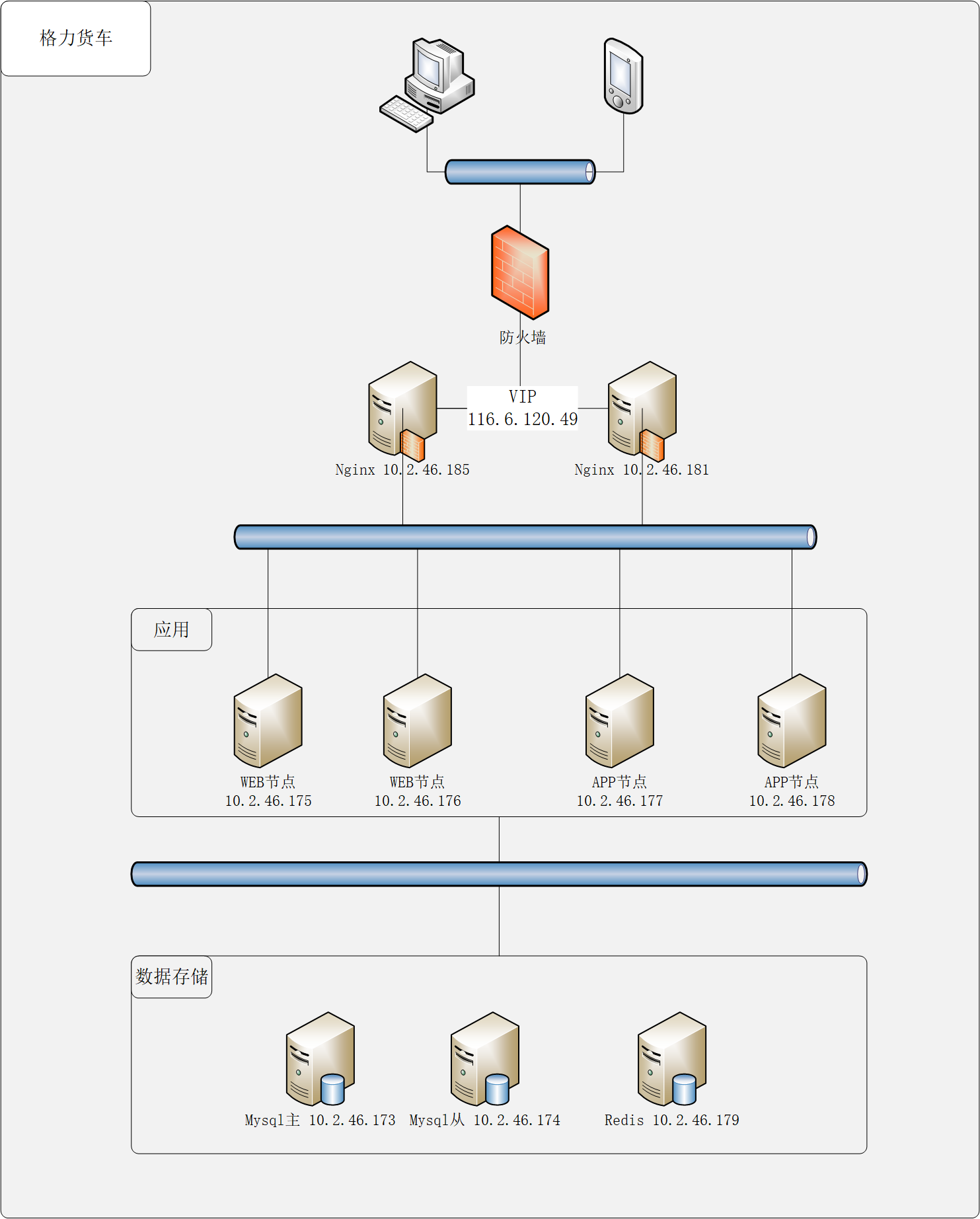
#### 软件环境

|  |  |
| --- | --- |
| **软件环境** | **说明** |
| 操作系统 | CentOS7.2 |
| 应用服务器 | Tomcat |
| JDK | JDK1.8 |
| 数据库 | Mysql5.7 |
| 缓存服务 | Redis |

## 系统部署架构

应用软件采用B/S的应用架构，根据本项目的业务需求，系统应满足12000用户的业务容量。本项目的业务需求及软件架构，建议配置四台应用服务器和两台数据库服务器。

系统网络拓扑结构如下：



由于网站的访问量会不断加大，负载越来越高，需要实现在web前端放置nginx负载均衡,同时结合keepalived对前端nginx实现高可用。利用两台两台的Nginx负载均衡器，用来分发接收到客户端的请求，在上图已经配置好了Nginx185，Nginx181也是一样的配置。现在要在两个Nginx负载均衡器上做高可用配置，Nginx185作为主节点，Nginx181作为备节点；keepalived可以认为是VRRP协议在Linux上的实现，主要有三个模块，分别是core、check和vrrp。

core模块为keepalived的核心，负责主进程的启动、维护以及全局配置文件的加载和解析。

check负责健康检查，包括常见的各种检查方式。

vrrp模块是来实现VRRP协议的。

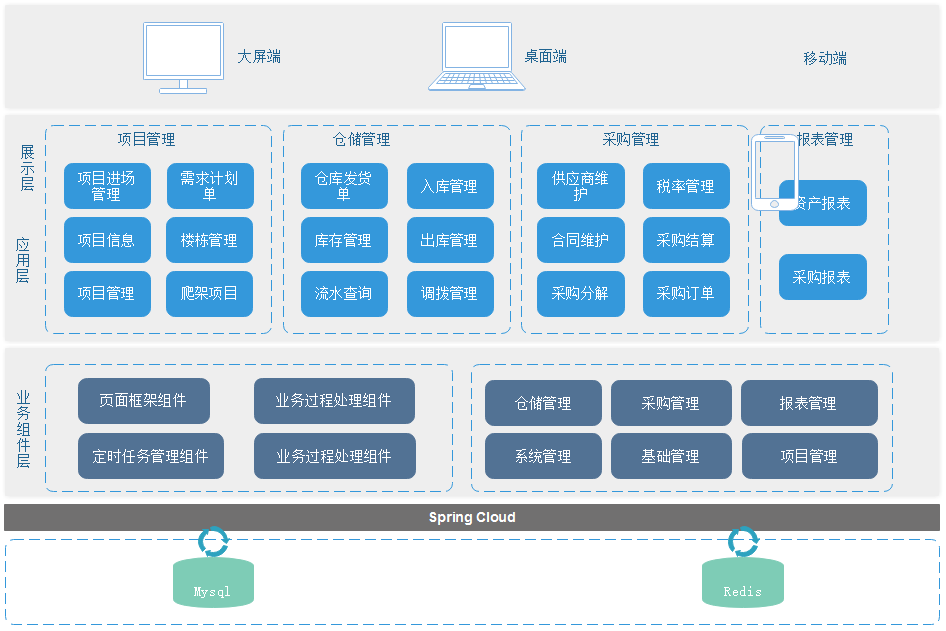
即前端使用两台服务器，一台主服务器和一台热备服务器，正常情况下，主服务器绑定一个公网虚拟IP，提供负载均衡服务，热备服务器处于空闲状态；当主服务器发生故障时，热备服务器接管主服务器的公网虚拟IP，从而实现应用处理的负载均衡。

四台应用服务器需连接文件存储服务，用以存取静态文件、静态工程、附件、图片等非结构化文件（可以采用NFS，也可以采用分布式文件存储服务）；两台mysql数据库服务器实现主从同步，当master(主)库的数据发生变化的时候，变化会实时的同步到slave(从)库，解决数据库的负载能力，高可用以及数据备份。还有一台Redis缓存数据库服务器，非关系型数据库， 数据结构简单，使用与数据高并发的读写，减少io的读操作，解决荫公园内给服务器的cpu和内存压力，再结合mysql主从，减少数据库压力，提交效率。

## 技术体系结构

系统采用J2EE B/S MVC三层架构，结合spring-boot技术组件。

## 系统架构



# 功能模块设计

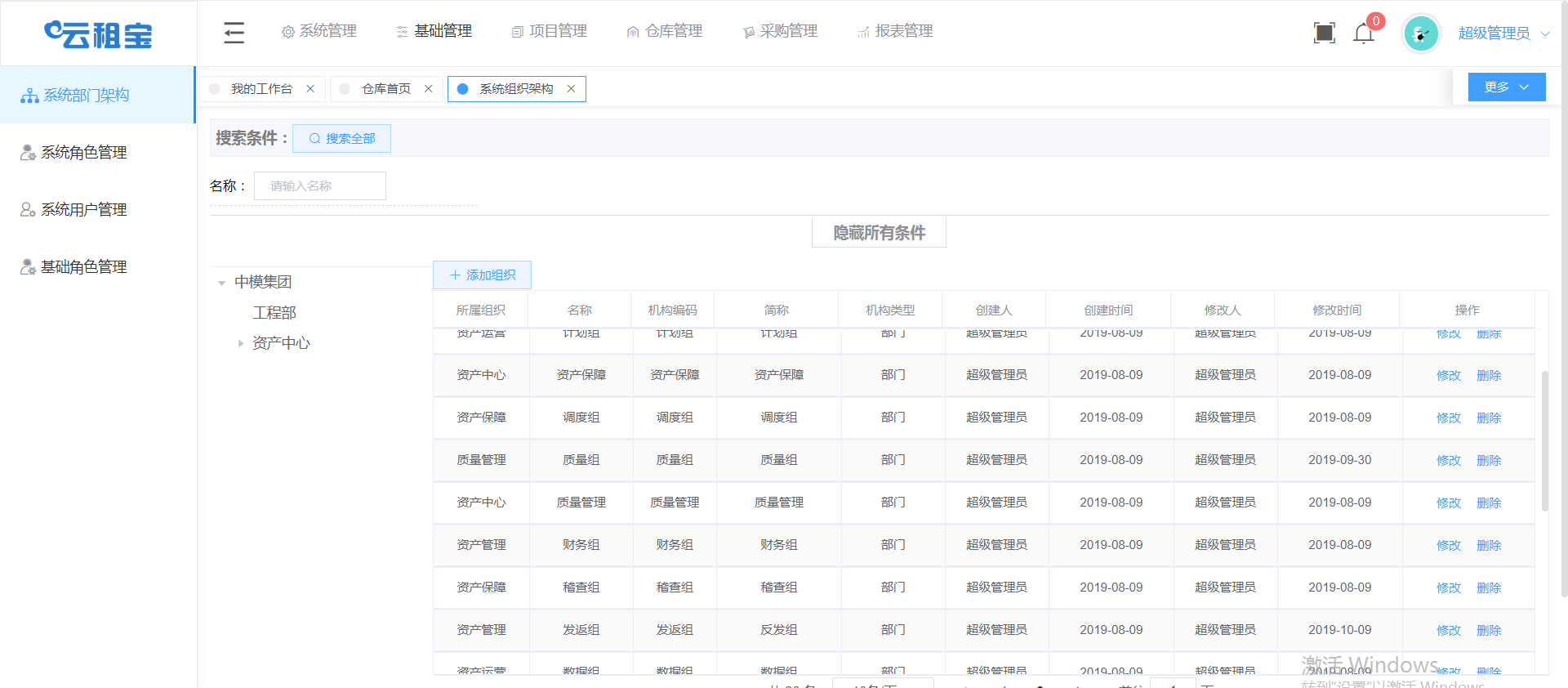
## 系统管理

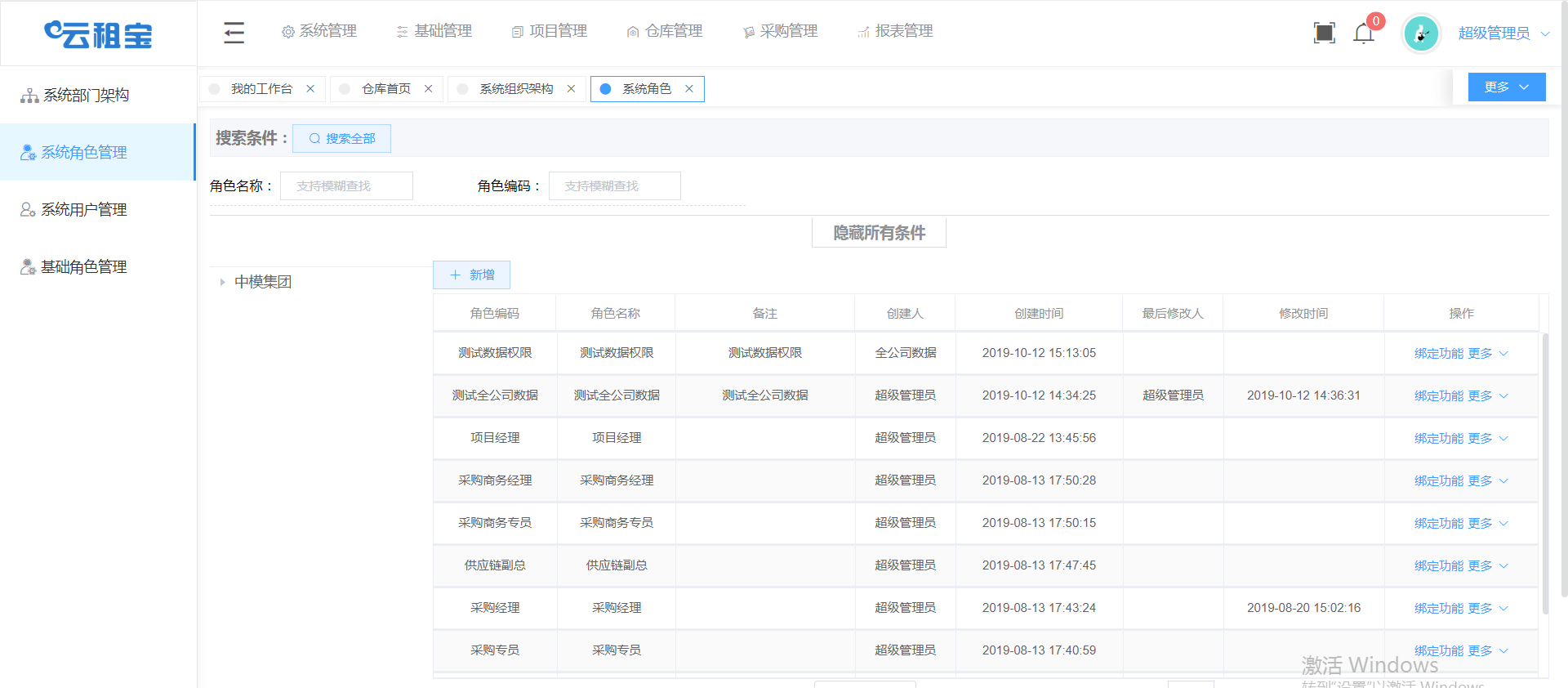
对系统的的部门架构，角色，用户等信息进行管理维护。

### 设计思路

#### 程序描述

创建系统组织架构，创建相应的角色和用户





#### 权限分配

管理员可以使用。

#### 数据结构

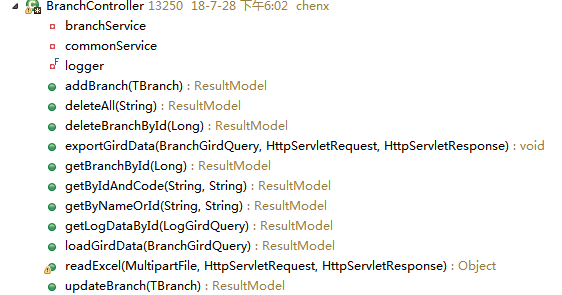
表t\_user，t\_org，t\_ore。还有其中的中间表

#### 详细功能

1. 添加：点击“添加”按钮会出现添加界面，点击界面上的地图选择地点然后填写其他信息再点击“保存”按钮即可成功增加应用。
2. 删除：选中一条信息点击列表上行的“删除”或者点击列表中的删除按钮会弹出确认提示对话框点击“确定”即可删除，点击“返回”即取消删除。
3. 列表：查看应用信息列表，按每页10条分页显示。
4. 查询：通过站点名称、站点编号即可进行查询。
5. 修改：选中一条应用信息点击“修改”按钮打开修改应用界面，正确修改应用信息后点击“保存”即可成功修改应用信息。
6. 导入：点击导入进入导入窗口，下载出导入模板，按照导入须知填写每一列的信息，然后导入窗口选择填好的Excel文件进行信息导入。
7. 导出：点击导出将把列表查出来的所有的数据导出到一个Excel表上。

#### 实现与接口

应用管理业务处理类定义如下：



主要应用接口定义如下：

方法定义：addBranch(Tbranch)

功能说明：添加应用信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：deleteAll(String)

功能说明：批量删除应用信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：deleteBranchById(Long)

功能说明：根据id删除应用信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：exportGridData(BranchGripQuery, HttpServletRequest, HttpServletResponse)

功能说明：导出应用信息

方法定义：getBranchById(Long)

功能说明：根据id查询应用信息，返回查询到的对象

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：getByIdAndCode(String, String)

功能说明：根据ID和编号查询应用信息，返回查询到的对象

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：getByNameOrId(String, String)

功能说明：根据ID或者名称查询应用信息，用于应用的名称唯一性校验，返回查询到的对象

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：getLogDataById(LogGripQuery)

功能说明：查询应用信息的操作日记

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：loadGripData(BranchGripQuery)

功能说明：查询应用信息列表，分页查询

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：readExcel(MultipartFile, HttpServletRequest, HttpServletResponse)

功能说明：导入应用信息，将导入成功或导入失败的问题记录成一个excel表并输出。

返回值： 返回导入文件无效的信息或返回excel表

方法定义：updateBranch(Branch)

功能说明：修改应用信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

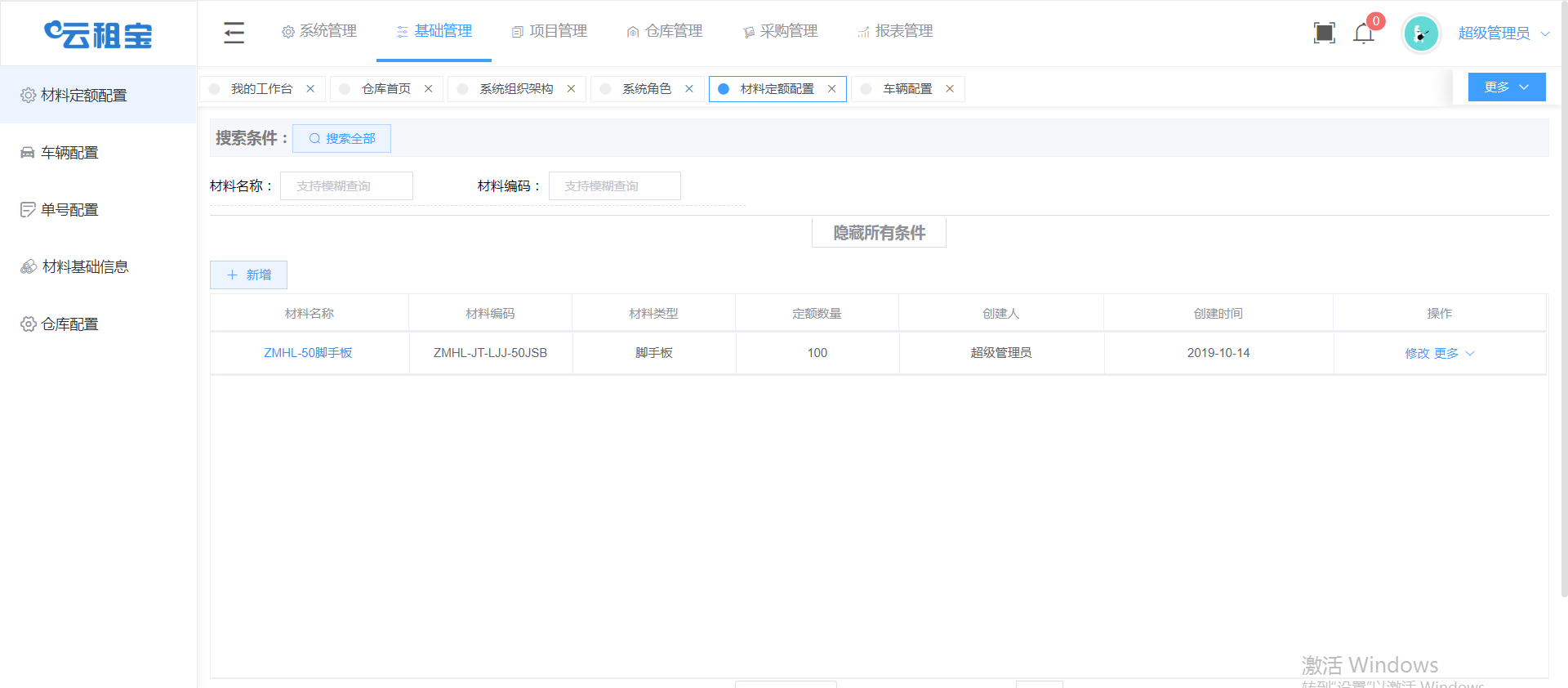
## 材料定额配置

站点耗时信息记录两站点间的耗时时长，对应车型版位数等信息。

### 设计思路

#### 程序描述

指定材料的类型和名称，再指定额度。



#### 权限分配

全体人员可以使用。

#### 数据结构

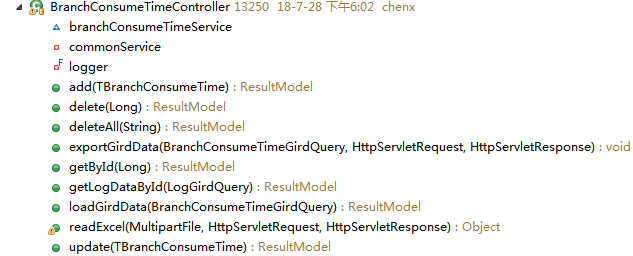
表zm\_material\_setting

#### 详细功能

1. 添加：点击“添加”按钮会出现添加界面，选择出发站点、到达站点等信息再点击“确定”按钮即可成功增加应用。
2. 删除：选中一条信息点击列表上行的“删除”或者点击列表中的删除按钮会弹出确认提示对话框点击“确定”即可删除，点击“返回”即取消删除。
3. 列表：查看应用信息列表，按每页10条分页显示。
4. 查询：通过出发站点、到达站点进行查询。
5. 修改：选中一条应用信息点击“修改”按钮打开修改应用界面，正确修改应用信息后点击“保存”即可成功修改应用信息。
6. 导入：点击导入进入导入窗口，下载出导入模板，按照导入须知填写每一列的信息，然后导入窗口选择填好的Excel文件进行信息导入。
7. 导出：点击导出将把列表查出来的所有的数据导出到一个Excel表上。

#### 实现与接口

应用管理业务处理类定义如下：



主要应用接口定义如下：

方法定义：add (TbranchConsumeTime)

功能说明：添加站点耗时信息，包含出发站点、到达站点、耗时时长、车型版位数四个信息

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：delete(Long)

功能说明：根据id删除站点耗时信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：deleteAll(String)

功能说明：批量删除应用信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：exportGridData(BranchGripQuery, HttpServletRequest, HttpServletResponse)

功能说明：导出应用信息

方法定义：getById(Long)

功能说明：根据id查询站点耗时信息，返回查询到的对象

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1。

使用result作为KEY接收一个对象给前端处理

方法定义：getLogDataById(LogGripQuery)

功能说明：查询应用信息的操作日记

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：loadGripData(BranchConsumeTimeGripQuery)

功能说明：查询站点耗时信息列表，分页查询

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：readExcel(MultipartFile, HttpServletRequest, HttpServletResponse)

功能说明：导入应用信息，将导入成功或导入失败的问题记录成一个excel表并输出。

返回值： 返回导入文件无效的信息或返回excel表

方法定义：update (TBranchConsumeTime)

功能说明：修改站点耗时信息， 只能修改耗时时长

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

## 配置

添加的基础信息，包括名称，编号和吨位。

### 设计思路

#### 程序描述

可以通过名称和编码进行查询

信息列表查询界面如下：



#### 权限分配

全体人员可以使用。

#### 数据结构

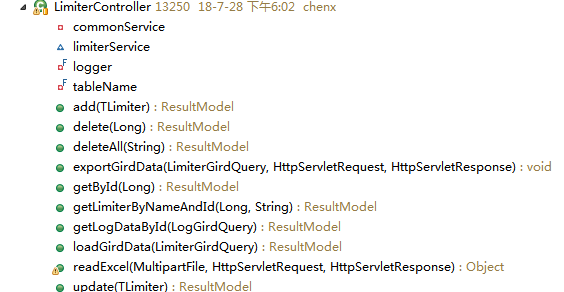
表zm\_vehicle\_manager

#### 详细功能

1. 添加：点击“添加”按钮会出现添加界面，填入信息再点击“确定”按钮即可成功增加道路限行信息。
2. 删除：选中一条信息点击列表上行的“删除”或者点击列表中的删除按钮会弹出确认提示对话框点击“确定”即可删除，点击“返回”即取消删除。
3. 列表：查看信息列表，按每页10条分页显示。
4. 查询：通过限行信息名称进行查询。
5. 修改：选中一条应用信息点击“修改”按钮打开修改应用界面，正确修改应用信息后点击“确定”即可成功修改信息。
6. 导入：点击导入进入导入窗口，下载出导入模板，按照导入须知填写每一列的信息，然后导入窗口选择填好的Excel文件进行信息导入。
7. 导出：点击导出将把列表查出来的所有的数据导出到一个Excel表上。

#### 实现与接口

应用管理业务处理类定义如下：



主要应用接口定义如下：

方法定义：add (TLimiter)

功能说明：添加信息

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：delete(Long)

功能说明：根据id删除信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：deleteAll(String)

功能说明：批量删除信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：exportGridData(BranchGripQuery, HttpServletRequest, HttpServletResponse)

功能说明：导出应用信息

方法定义：getById(Long)

功能说明：根据id查询信息，返回查询到的对象

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1。

使用result作为KEY接收一个对象给前端处理

方法定义：getLimiterByNameAndId(Long,String)

功能说明：根据id和名称查询信息，返回查询到的对象

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1。

使用result作为KEY接收一个对象给前端处理

方法定义：getLogDataById(LogGripQuery)

功能说明：查询应用信息的操作日记

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：loadGripData(LimiterGripQuery)

功能说明：查询站点耗时信息列表，分页查询

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：readExcel(MultipartFile, HttpServletRequest, HttpServletResponse)

功能说明：导入应用信息，将导入成功或导入失败的问题记录成一个excel表并输出。

返回值： 返回导入文件无效的信息或返回excel表

方法定义：update (TLimiter)

功能说明：修改信息

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

## 单号设置

系统中涉及的各种单号规则设置。

### 设计思路

#### 程序描述

提供单号的设置功能，可以启用禁用单号规则，也可以修改。

信息列表查询界面如下：



#### 权限分配

全体人员可以使用。

#### 数据结构

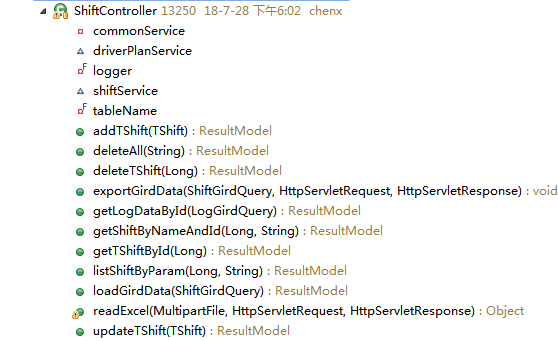
表zm\_billnumber

#### 详细功能

1. 添加：点击“添加”按钮会出现添加界面，填入信息再点击“确定”按钮即可成功增加班次类型信息。
2. 删除：选中一条信息点击列表上行的“删除”或者点击列表中的删除按钮会弹出确认提示对话框点击“确定”即可删除，点击“返回”即取消删除。
3. 列表：查看信息列表，按每页10条分页显示。
4. 查询：通过班次类型进行查询。
5. 修改：选中一条应用信息点击“修改”按钮打开修改应用界面，正确修改应用信息后点击“确定”即可成功修改信息。
6. 导入：点击导入进入导入窗口，下载出导入模板，按照导入须知填写每一列的信息，然后导入窗口选择填好的Excel文件进行信息导入。
7. 导出：点击导出将把列表查出来的所有的数据导出到一个Excel表上。

#### 实现与接口

应用管理业务处理类定义如下：



主要应用接口定义如下：

方法定义：addTshift (TShift)

功能说明：添加信息

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：deleteTshift(Long)

功能说明：根据id删除信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：deleteAll(String)

功能说明：批量删除信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：exportGridData(BranchGripQuery, HttpServletRequest, HttpServletResponse)

功能说明：导出应用信息

方法定义：getTShiftById(Long)

功能说明：根据id查询信息，返回查询到的对象

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1。

使用result作为KEY接收一个对象给前端处理

方法定义：getShiftByNameAndId(Long,String)

功能说明：根据id和名称查询信息，返回查询到的对象

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1。

使用result作为KEY接收一个对象给前端处理

方法定义：listShiftByParam(Long,String)

功能说明：根据id和名称查询信息，返回查询到的对象

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1。

使用result作为KEY接收一个数组给前端处理

方法定义：getLogDataById(LogGripQuery)

功能说明：查询信息的操作日记

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：loadGripData(ShiftGripQuery)

功能说明：查询信息列表，分页查询

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：readExcel(MultipartFile, HttpServletRequest, HttpServletResponse)

功能说明：导入应用信息，将导入成功或导入失败的问题记录成一个excel表并输出。

返回值： 返回导入文件无效的信息或返回excel表

方法定义：updateTShift(TShift)

功能说明：修改信息

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

## 材料基础信息

新增材料，指定所属分类，添加图号、单位、重量等基本信息。

### 设计思路

#### 程序描述

新增、修改和删除材料的基本信息。

信息列表查询界面如下：



#### 权限分配

全体人员可以使用。

#### 数据结构

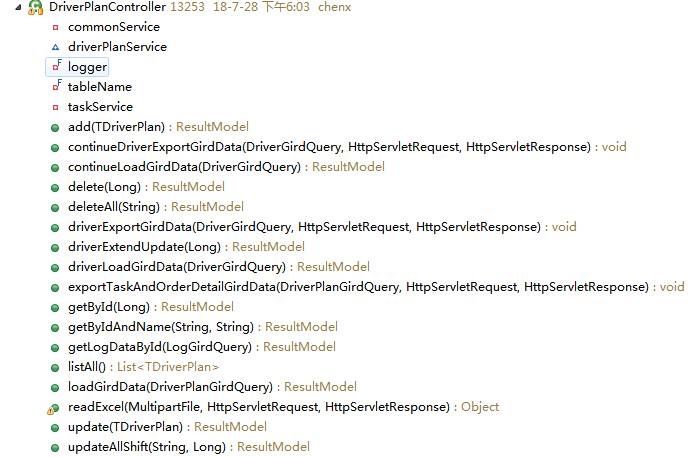
表zm\_goods\_base

#### 详细功能

1. 添加：点击“添加”按钮会出现添加界面，填入信息再点击“确定”按钮即可成功增加司机排班信息。
2. 删除：选中一条信息点击列表上行的“删除”或者点击列表中的删除按钮会弹出确认提示对话框点击“确定”即可删除，点击“返回”即取消删除。
3. 列表：查看信息列表，按每页10条分页显示。
4. 查询：通过司机名称、车牌号、班次类型进行查询。
5. 修改：选中一条应用信息点击“修改”按钮打开修改应用界面，正确修改应用信息后点击“确定”即可成功修改信息。
6. 导入：点击导入进入导入窗口，下载出导入模板，按照导入须知填写每一列的信息，然后导入窗口选择填好的Excel文件进行信息导入。
7. 导出：点击导出将把列表查出来的所有的数据导出到一个Excel表上。
8. 班次变更：用一个司机只能有一个排班信息，同一辆车一个班次类型只能有一个排班信息

#### 实现与接口

应用管理业务处理类定义如下：



主要应用接口定义如下：

方法定义：add (TDriverPlan)

功能说明：添加信息

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：continueDriverExportData (DriverGripQuery, HttpServletReques, HttpServletResponse)

功能说明：司机连续上班信息导出接口

方法定义：continueLoadGirdData (DriverGripQuery,)

功能说明：司机连续上班信息导出接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1。

使用result作为KEY接收一个对象给前端处理

方法定义：delete (Long)

功能说明：根据id删除信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：deleteAll(String)

功能说明：批量删除信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：driverExportGirdData (BranchGripQuery, HttpServletRequest, HttpServletResponse)

功能说明：司机上班统计信息导出接口

方法定义：driverExtendUpdate (Long)

功能说明：司机连续上班信息消除接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1。

方法定义：driverLoadGirdData (Long)

功能说明：司机连续上班信息消除接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1。

使用result作为KEY接收一个数组给前端处理

方法定义：exportTaskAndOrderDetailGirdData (DriverPlanGirdQuery,HttpServletRequest, HttpServletResponse)

功能说明：司机排班管理导出接口

方法定义：getById(Long)

功能说明：根据id查询信息，返回查询到的对象

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1。

使用result作为KEY接收一个对象给前端处理

方法定义：getByIdAndName(String,String)

功能说明：通过司机id查询对象司机排班信息,可用来校验司机唯一性

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1。

使用result作为KEY接收一个对象给前端处理

方法定义：getLogDataById(LogGripQuery)

功能说明：查询信息的操作日记

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：listAll()

功能说明：查询所有排班数据接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：loadGripData(ShiftGripQuery)

功能说明：查询信息列表，分页查询

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：readExcel(MultipartFile, HttpServletRequest, HttpServletResponse)

功能说明：导入应用信息，将导入成功或导入失败的问题记录成一个excel表并输出。

返回值： 返回导入文件无效的信息或返回excel表

方法定义：update(TDriverPlan)

功能说明：修改信息

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：updateAllShift (TShift)

功能说明：批量修改班次接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

## 仓库配置

指定仓库的所在地，名称，编码，类型等信息。

### 设计思路

#### 程序描述

列表提供名称、编码等条件查询功能。

任务流程信息列表查询界面如下：



#### 权限分配

全体人员可以使用。

#### 数据结构

表zm\_warehouse\_base、表zm\_warehouse\_detail、表zm\_warehouse\_region、表zm\_warehouse\_region\_mid

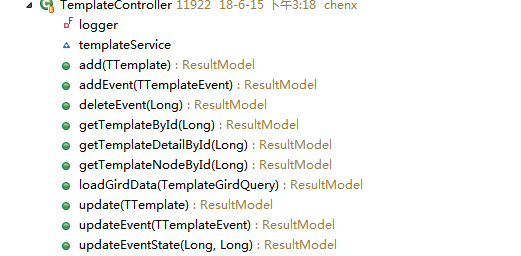
#### 详细功能

1. 查询：通过司机名称、车牌号、班次类型进行查询。
2. 详情：选中一条流程点击操作列中的详情按钮可查看对应的流程节点和异常事件信息



#### 实现与接口

应用管理业务处理类定义如下：



主要应用接口定义如下：

方法定义：add (TTemplate)

功能说明：添加信息

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：addEvent(TTemplateEvent)

功能说明：根据流程模板id新增异常事件接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1。

方法定义：deleteExent (Long)

功能说明：异常事件删除接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：getTemplateById(Long)

功能说明：根据ID查询流程模板接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1。

使用result作为KEY接收一个对象给前端处理

方法定义：getTemplateDetailById (Long)

功能说明：根据ID查看流程模板明细接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1。

使用result作为KEY接收一个对象给前端处理

方法定义：getTemplateNodeById (Long)

功能说明：根据ID查询流程节点接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1。

使用result作为KEY接收一个对象给前端处理

方法定义：loadGripData(ShiftGripQuery)

功能说明：查询信息列表，分页查询

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：update (TTemplate)

功能说明：流程模板修改接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：updateEvent (TTemplateEvent)

功能说明：根据异常id和流程模板id修改异常事件接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：updateEventState (Long,Long)

功能说明：根据异常id修改异常事件状态接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

## 项目信息

自定义信息是对信进行扩展维护，加上状态、板位数、所属站点、工作时间、可用时间以及可用站点等信息。

### 设计思路

#### 程序描述

新增和维护项目信息。包括项目的名称，编码，地址，备案等等信息。

信息列表查询界面如下：



#### 权限分配

全体人员可以使用。

#### 数据结构

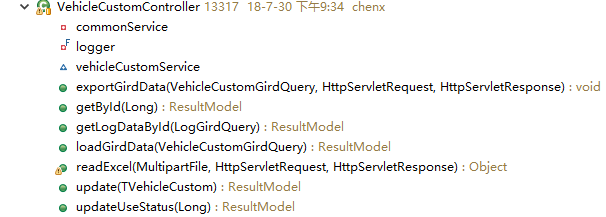
表zm\_project、表zm\_project\_build、表zm\_marketing\_project、表zm\_supplement\_order、表zm\_supplement\_order\_detail

#### 详细功能

1. 列表：查看信息列表，按每页10条分页显示。
2. 查询：通过司机名称、车牌号、班次类型进行查询。
3. 修改：选中一条应用信息点击“修改”按钮打开修改应用界面，正确修改应用信息后点击“确定”即可成功修改信息。
4. 导入：点击导入进入导入窗口，下载出导入模板，按照导入须知填写每一列的信息，然后导入窗口选择填好的Excel文件进行信息导入。
5. 导出：点击导出将把列表查出来的所有的数据导出到一个Excel表上。
6. 状态变更：列表上每个都有状态按钮修改，点击即可切换修改的可用状态。

#### 实现与接口

应用管理业务处理类定义如下：



主要应用接口定义如下：

方法定义：add (TDriverPlan)

功能说明：添加信息

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：exportGridData(VehicleCustomGripQuery, HttpServletRequest, HttpServletResponse)

功能说明：导出列表信息

方法定义：delete (Long)

功能说明：根据id删除信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：deleteAll(String)

功能说明：批量删除信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：driverExportGirdData (BranchGripQuery, HttpServletRequest, HttpServletResponse)

功能说明：司机上班统计信息导出接口

方法定义：getById(Long)

功能说明：根据id查询信息，返回查询到的对象

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1。

使用result作为KEY接收一个对象给前端处理

方法定义：getLogDataById(LogGripQuery)

功能说明：查询信息的操作日记

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：loadGripData(VehicleCustomGripQuery)

功能说明：查询信息列表，分页查询

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：readExcel(MultipartFile, HttpServletRequest, HttpServletResponse)

功能说明：导入应用信息，将导入成功或导入失败的问题记录成一个excel表并输出。

返回值： 返回导入文件无效的信息或返回excel表

方法定义：update(TVehicleCustom)

功能说明：自定义修改接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：updateUseStatus (TShift)

功能说明：根据ID启用禁用

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

## 项目管理、楼栋管理

创建项目，指定项目的合同起始时间，业务类型是否发货等等。

### 设计思路

#### 程序描述

列表提供项目编号，项目名称等条件查询功能

信息列表查询界面如下：





#### 权限分配

全体人员可以使用。

#### 数据结构

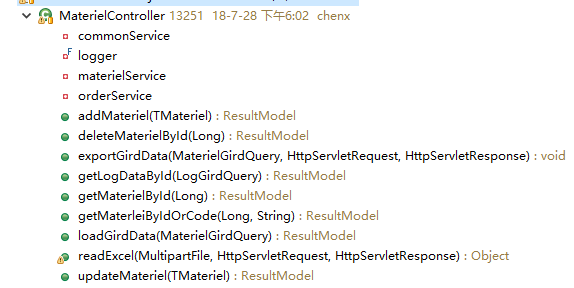
表zm\_project、表zm\_project\_build、表zm\_marketing\_project、表zm\_supplement\_order、表zm\_supplement\_order\_detail

#### 详细功能

1. 添加：点击“添加”按钮会出现添加界面，填入信息再点击“确定”按钮即可成功增加司机排班信息。
2. 删除：选中一条信息点击列表上行的“删除”或者点击列表中的删除按钮会弹出确认提示对话框点击“确定”即可删除，点击“返回”即取消删除。
3. 列表：查看信息列表，按每页10条分页显示。
4. 查询：通过司机名称、车牌号、班次类型进行查询。
5. 修改：选中一条应用信息点击“修改”按钮打开修改应用界面，正确修改应用信息后点击“确定”即可成功修改信息。
6. 导入：点击导入进入导入窗口，下载出导入模板，按照导入须知填写每一列的信息，然后导入窗口选择填好的Excel文件进行信息导入。
7. 导出：点击导出将把列表查出来的所有的数据导出到一个Excel表上。

#### 实现与接口

应用管理业务处理类定义如下：



主要应用接口定义如下：

方法定义：addMateriel(TMateriel)

功能说明：添加信息

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：deleteMaterielById (Long)

功能说明：根据id删除信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：exportGridData(VehicleCustomGripQuery, HttpServletRequest, HttpServletResponse)

功能说明：导出列表信息

方法定义：getLogDataById(LogGripQuery)

功能说明：查询信息的操作日记

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：getMaterielById(Long)

功能说明：根据id查询信息，返回查询到的对象

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1。

使用result作为KEY接收一个对象给前端处理

方法定义：getMaterielByIdOrCode(Long)

功能说明：通过材料编号查询对象,可用来校验编号唯一性

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1。

方法定义：loadGripData(VehicleCustomGripQuery)

功能说明：查询信息列表，分页查询

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：readExcel(MultipartFile, HttpServletRequest, HttpServletResponse)

功能说明：导入应用信息，将导入成功或导入失败的问题记录成一个excel表并输出。

返回值： 返回导入文件无效的信息或返回excel表

方法定义：updateMatereiel (TMateriel)

功能说明：修改物料信息

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

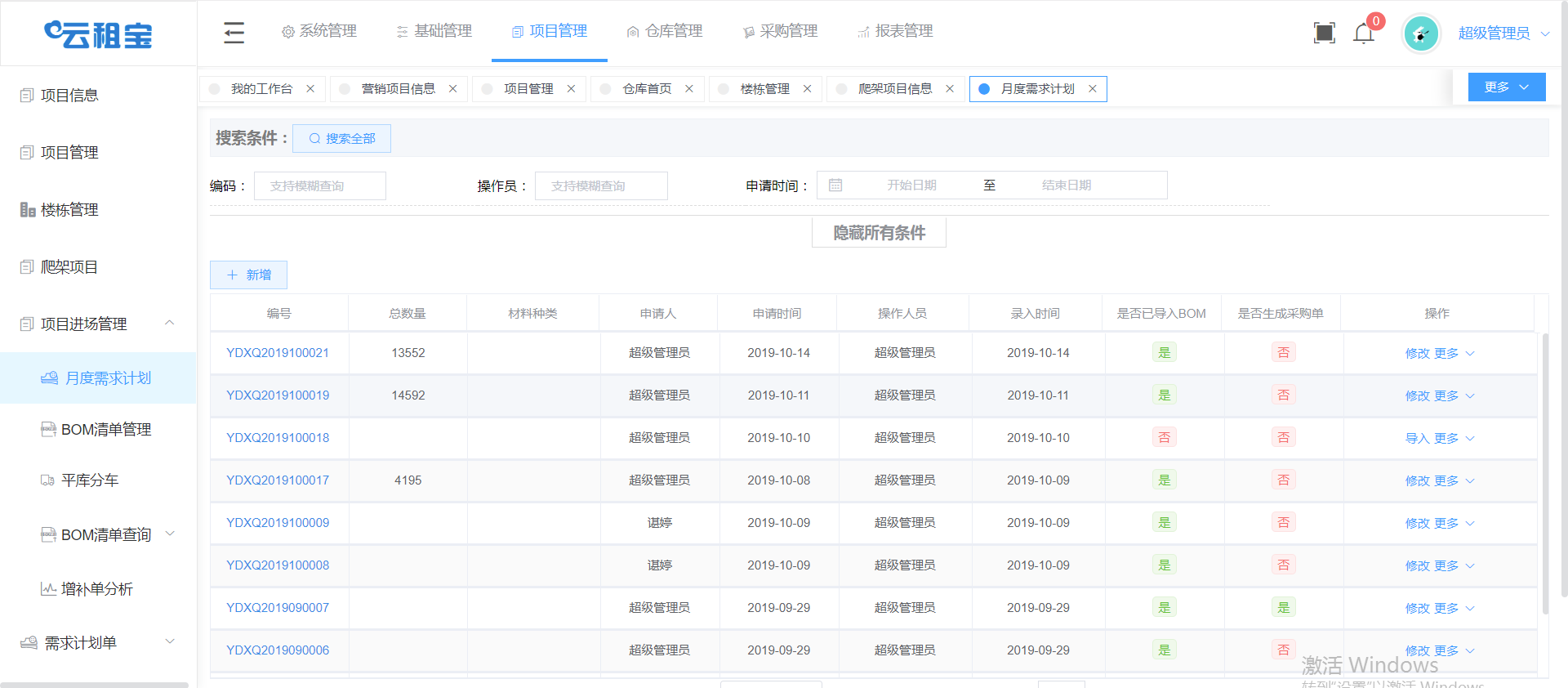
## 项目进场管理

项目进场管理对项目的入场流程提供管理，包括月度需求计划，BOM清单管理，平库分车等等。

### 设计思路

#### 程序描述

月的



#### 权限分配

全体人员可以使用。

#### 数据结构

表zm\_month\_plan、表zm\_month\_plan\_detail、表zm\_month\_plan\_detail\_temp、表zm\_import\_record

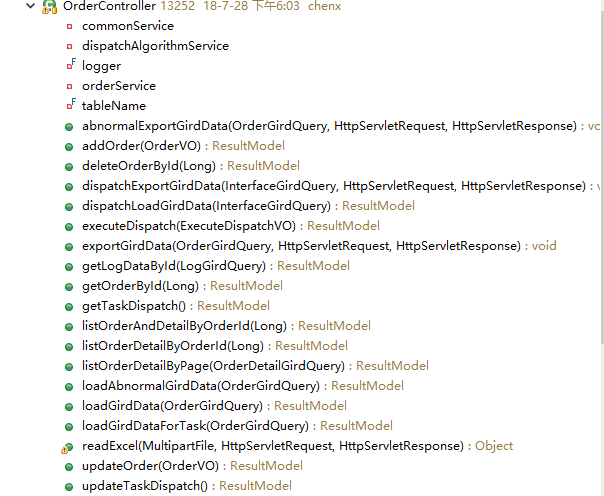
#### 详细功能

（主要描述物料订单列表的功能）

1. 列表：查看信息列表，按每页10条分页显示。
2. 查询：通过订单日期、订单号、子订单号、任务编号、装货站点、卸货站点、订单调度状态、订单执行状态、申请部门、订单来源进行查询。
3. 添加：点击“添加”按钮会出现添加界面，填写订单的基本信息和添加物料明细，再点击“确定”按钮即可成功添加。
4. 删除：订单调度状态为未调度的时候才可以删除，选中一条信息点击列表上行的“删除”或者点击列表中的删除按钮会弹出确认提示对话框点击“确定”即可删除，点击“返回”即取消删除。
5. 修改：订单调度状态为未调度的时候才可以修改，选中一条应用信息点击“修改”按钮打开修改界面，修改物料明细，正确修改信息后点击“确定”即可成功修改信息。
6. 任务调度：点击任务电镀调度，提交订单时间范围即可执行任务调度，系统通过算法给任务分配订单。
7. 导入：点击导入进入导入窗口，下载出导入模板，按照导入须知填写每一列的信息，然后导入窗口选择填好的Excel文件进行信息导入。
8. 导出：点击导出将把列表查出来的所有的数据导出到一个Excel表上。
9. 查看明细：点击每行的操作栏的查看按钮，可以查看该条订单的物料明细信息。

#### 实现与接口

应用管理业务处理类定义如下：



主要应用接口定义如下：

方法定义：abnormalExportGirdData (OrderGirdQuery,HttpServletRequest, HttpServletResponse)

功能说明：异常物料订单信息管理导出接口

方法定义：addOrder (OrderVO)

功能说明：新增物料订单信息

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：deleteOrderById (Long)

功能说明：删除订单信息

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：dispatchExportGirdData (InterfaceGirdQuery, HttpServletRequest, HttpServletResponse)

功能说明：调度信息管理导出接口

方法定义：dispatchLoadGirdData (InterfaceGirdQuery)

功能说明：调度信息列表查询接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：executeDispatch (ExecuteDispatchVO)

功能说明：执行订单任务调度

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：exportGirdData (OrderGirdQuery, HttpServletRequest, HttpServletResponse)

功能说明：物料订单信息管理导出接口

方法定义：getLogDataById(LogGripQuery)

功能说明：查询信息的操作日记

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：getOrderById (Long)

功能说明：根据ID查询单个物料订单信息

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1。

使用result作为KEY接收一个对象给前端处理

方法定义：getTaskDispatch ()

功能说明：查询调度情况

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1。

使用result作为KEY接收一个对象给前端处理

方法定义：listOrderAndDetailByOrderId (Long)

功能说明：通过订单id查询订单和订单明细接口（用于编辑的回显）

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个对象给前端处理

方法定义：listOrderDetailByOrderId (Long)

功能说明：通过订单id查询物料订单明细接口（用于物料明细查看）

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个对象给前端处理

方法定义：listOrderDetailByPage (OrderDetailGirdQuery)

功能说明：通过订单id分页查询物料明细，需要传订单id

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：loadAbnormalGirdData (OrderDetailGirdQuery)

功能说明：物料订单信息异常列表查询接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：loadGirdData (OrderGirdQuery)

功能说明：物料订单信息列表查询接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：loadGirdDataForTask (OrderGirdQuery)

功能说明：任务订单的物料订单信息

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：readExcel(MultipartFile, HttpServletRequest, HttpServletResponse)

功能说明：导入应用信息，将导入成功或导入失败的问题记录成一个excel表并输出。

返回值： 返回导入文件无效的信息或返回excel表

方法定义：updateOrder (OrderVO)

功能说明：修改物料订单信息

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：updateTaskDispatch ()

功能说明：修改调度情况

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

## 入库管理

针对各种产品或者材料的入库进行管理，根据类型进行处理。

### 设计思路

#### 程序描述

成品入库：



外购入库：



喷涂入库：



其他入库：



改制入库：



#### 权限分配

全体人员可以使用。

#### 数据结构

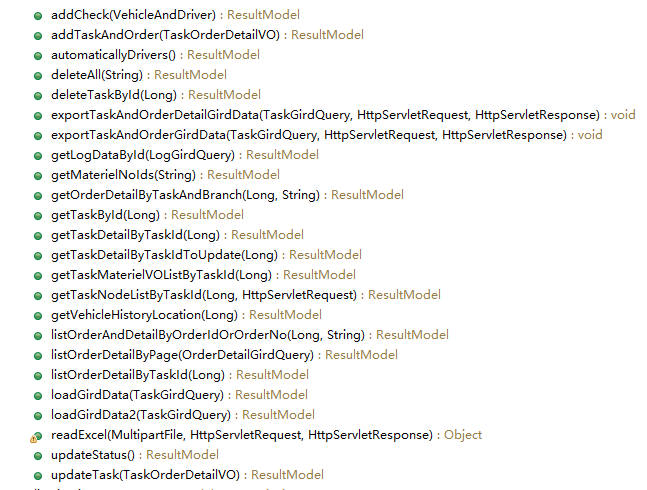
表zm\_in\_warehouse\_order、表zm\_in\_warehouse\_detail、表zm\_in\_warehouse\_detail\_ext、表zm\_in\_warehouse\_inventory、

#### 详细功能

1. 列表：查看信息列表，按每页10条分页显示。
2. 查询：通过计划时间、任务编号、司机名称、车牌号、装货站点、下发状态、业务类型进行查询。
3. 添加：点击“人工派车”按钮会出现添加界面，填写任务基本信息，再点击“提交”按钮即可成功添加返回列表页面，或者点击“提交并新增”进行添加并停留在添加页面。
4. 修改：未下发的任务才可以进行修改，选中一条应用信息点击“修改”按钮打开修改应用界面，正确修改应用信息后点击“确定”即可成功修改信息。
5. 删除：订单调度状态为未调度的时候才可以删除，选中一条信息点击列表上行的“删除”或者点击列表中的删除按钮会弹出确认提示对话框点击“确定”即可删除，点击“返回”即取消删除。
6. 取消任务：已下发但未执行的任务才可以进行取消任务操作，选中一条应用信息点击“取消任务”按钮图标弹出确认框界面，点击“确定”即可成功取消未执行的任务。
7. 修改：未下发的任务才可以进行修改，选中一条应用信息点击“修改”按钮打开修改应用界面，正确修改应用信息后点击“确定”即可成功修改信息。
8. 补发任务：点击“补发任务”按钮会出现补发界面，填写任务基本信息，与添加不同的就是补发任务就想当是补充任务记录，需填入实际开始和完成时间、实装板位数，然后点击“提交”按钮即可成功添加返回列表页面，或者点击“提交并新增”进行添加并停留在补发页面。
9. 分配司机：点击分配司机，根据任务的计划开始时间给任务分配该时间点可用的司机。
10. 导入：点击导入进入导入窗口，下载出导入模板，按照导入须知填写每一列的信息，然后导入窗口选择填好的Excel文件进行信息导入。
11. 导出：点击导出将把列表查出来的所有的数据导出到一个Excel表上。

#### 实现与接口

应用管理业务处理类定义如下：



主要应用接口定义如下：

方法定义：addCheck(Obj)

功能说明：添加前验证信息（验证此时间段该或司机有任务）

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：addTaskAndOrder(Obj)

功能说明：添加任务信息

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：automaticallyDrivers()

功能说明：自动分配驾驶

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：deleteAll(String)

功能说明：批量删除信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：deleteTaskById(String)

功能说明：删除单个运输任务接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：exportTaskAndOrderDetailGirdData (TaskGirdQuery,)

功能说明：备货指引管理导出接口

方法定义：exportTaskAndOrderGirdData (TaskGirdQuery,)

功能说明：运输任务管理导出接口

方法定义：getLogDataById(LogGripQuery)

功能说明：查询信息的操作日记

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：getMaterielNoIds(String)

功能说明：去添加任务页面的运输货物添加接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个对象给前端处理

方法定义：getTaskById(Long)

功能说明：根据ID查询单个运输任务信息

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个对象给前端处理

方法定义：getTaskDetailByTaskId(Long)

功能说明：根据任务ID查询任务明细详情(用于任务明细详情的数据回显)

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个对象给前端处理

方法定义：getTaskDetailByTaskIdToUpdate(Long)

功能说明：根据任务ID查询运输任务详情（打开编辑运输任务页面）

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个对象给前端处理

方法定义：getTaskNodeListByTaskId(Long)

功能说明：根据任务ID查询当前任务节点

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个对象给前端处理

方法定义：getVehicleHistoryLocation(Long)

功能说明：任务的历史轨迹

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个对象给前端处理

方法定义：listOrderAndDetailByOrderIdOrOrderNo (Long,String)

功能说明 通过订单id或者订单号查询订单和订单明细接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个对象给前端处理

方法定义：listOrderDetailByPage(OrderDetailGirdQuery)

功能说明：通过任务id分页查询物料明细，需要传任务id

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个对象给前端处理

方法定义：listOrderDetailByTaskId(Long)

功能说明：通过任务id查询物料订单明细接口（用于货物明细查看）

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个对象给前端处理

方法定义：loadGirdData(TaskGirdQuery)

功能说明：运输任务信息列表查询接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个对象给前端处理

方法定义：loadGirdData2(TaskGirdQuery)

功能说明：备货指引管理列表查询接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个对象给前端处理

方法定义：readExcel(MultipartFile, HttpServletRequest, HttpServletResponse)

功能说明：任务批量导入

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：updateStatus()

功能说明：确认所有未确认的任务

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：updateTask(TaskOrderDetailVO)

功能说明：修改运输任务订单信息

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个对象给前端处理

## 出库管理

根据业务场景进行出库。

### 设计思路

#### 程序描述

改制出库：



销售出库：



项目出库：



其他出库



喷涂出库



#### 权限分配

全体人员可以使用。

#### 数据结构

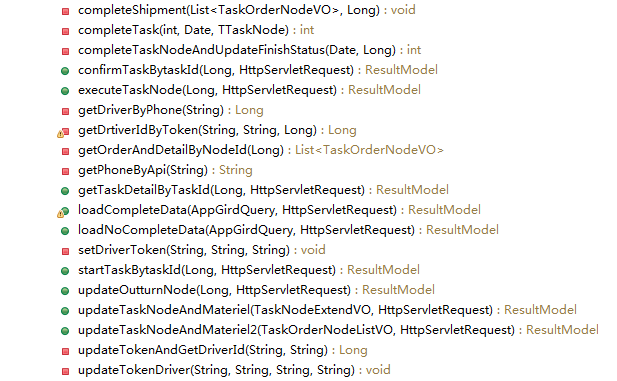
表zm\_out\_warehouse\_order、表zm\_out\_warehouse\_order\_detail、表zm\_in\_warehouse\_inventory

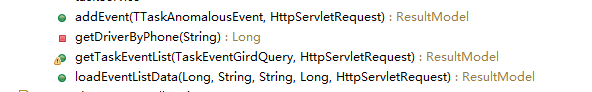
#### 详细功能

1. 开始任务：点击“开始任务”按钮会出现确认界面，确定后开始执行此任务，进入此任务的节点列表，包含任务开始节点、出发或到达类型的节点以及装卸货节点。
2. 执行节点：选中节点“点击开始”执行节点，执行完成后显示完成时间并跳到下一个节点。（装货节点需要多执行一步填写实际装货版位数的步骤）
3. 待完成/已完成列表：待完成列表可以执行和查看任务详情，已完成列表可以查看任务的执行详情以及浏览取消任务。
4. 我的上报：查看司机对任务的异常上报信息。
5. 异常上报：在每个任务节点列表上都可以对当前任务进行异常上报，选择异常类型和异常原因提交即可。

#### 实现与接口

业务处理类定义如下：





主要应用接口定义如下：

方法定义：confirmTaskBytaskId(Long, HttpServletRequest)

功能说明：任务确认

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：executeTaskNode(Long, HttpServletRequest)

功能说明：执行节点（出发、到达、装货、卸货等动作状态更变）

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：getTaskDetailByTaskId(Long, HttpServletRequest)

功能说明：根据任务ID查询当前任务节点

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY返回节点列表给前端处理

方法定义：loadCompleteData(AppGirdQuery, HttpServletRequest)

功能说明：运输任务信息列表查询接口(完成的任务)

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY返回节点列表给前端处理

方法定义：loadNoCompleteData(AppGirdQuery, HttpServletRequest)

功能说明：运输任务信息列表查询接口(未完成的任务)

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY返回节点列表给前端处理

方法定义：startTaskBytaskId(Long, HttpServletRequest)

功能说明：开始任务(将任务改为执行中状态)

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：updateOutturnNode(Long, HttpServletRequest)

功能说明：确认卸货接口（将卸货节点状态改为已完成）

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：updateTaskNodeAndMateriel(TaskNodeExtendVO, HttpServletRequest)

功能说明：确认装货接口（将对应的货物明细的实装数量修改确认，并将装货节点状态改为已完成）

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义 updateTaskNodeAndMateriel2(TaskOrderNodeListVO, HttpServletRequest)

功能说明：确认装货接口（将对应的货物明细的实装数量修改确认，并将装货节点状态改为已完成）

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义 addEvent(TTaskAnomalousEvent, HttpServletRequest)

功能说明：异常上报提交接口(异常事件保存)

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义 getTaskEventList(TaskEventGirdQuery, HttpServletRequest)

功能说明：异常事件列表

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1；使用result作为KEY返回异常事件列表给前端处理

方法定义 loadEventListData(Long, String, String, Long, HttpServletRequest)

功能说明：去异常上报页面需要查询的异常类型接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY返回异常事件类型列表给前端处理

## 调拨管理

对仓库直接的库存进行调拨管理操作。

### 设计思路

#### 程序描述

调拨单体现出调出仓库，调入仓库等信息。

信息列表查询界面如下：



#### 权限分配

全体人员可以使用。

#### 数据结构

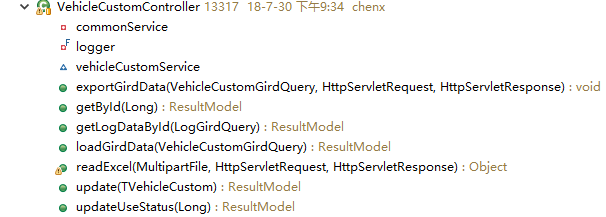
表zm\_allot\_warehouse\_order、表zm\_allot\_warehouse\_order\_detail、表zm\_in\_warehouse\_inventory

#### 详细功能

1. 列表：查看信息列表，按每页10条分页显示。
2. 查询：通过司机名称、车牌号、班次类型进行查询。
3. 修改：选中一条应用信息点击“修改”按钮打开修改应用界面，正确修改应用信息后点击“确定”即可成功修改信息。
4. 导入：点击导入进入导入窗口，下载出导入模板，按照导入须知填写每一列的信息，然后导入窗口选择填好的Excel文件进行信息导入。
5. 导出：点击导出将把列表查出来的所有的数据导出到一个Excel表上。
6. 状态变更：列表上每个都有状态按钮修改，点击即可切换修改的可用状态。

#### 实现与接口

应用管理业务处理类定义如下：



主要应用接口定义如下：

方法定义：add (TDriverPlan)

功能说明：添加信息

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：exportGridData(VehicleCustomGripQuery, HttpServletRequest, HttpServletResponse)

功能说明：导出列表信息

方法定义：delete (Long)

功能说明：根据id删除信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：deleteAll(String)

功能说明：批量删除信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：driverExportGirdData (BranchGripQuery, HttpServletRequest, HttpServletResponse)

功能说明：司机上班统计信息导出接口

方法定义：getById(Long)

功能说明：根据id查询信息，返回查询到的对象

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1。

使用result作为KEY接收一个对象给前端处理

方法定义：getLogDataById(LogGripQuery)

功能说明：查询信息的操作日记

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：loadGripData(VehicleCustomGripQuery)

功能说明：查询信息列表，分页查询

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：readExcel(MultipartFile, HttpServletRequest, HttpServletResponse)

功能说明：导入应用信息，将导入成功或导入失败的问题记录成一个excel表并输出。

返回值： 返回导入文件无效的信息或返回excel表

方法定义：update(TVehicleCustom)

功能说明：自定义修改接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：updateUseStatus (TShift)

功能说明：根据ID启用禁用

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

## 库存管理

对仓库的货物进行库存和库龄查询统计。

### 设计思路

#### 程序描述

可以通过仓库名称，编码，类型，物料编码等进查询。还可以选择展示成图表和列表。

信息列表查询界面如下：





#### 权限分配

全体人员可以使用。

#### 数据结构

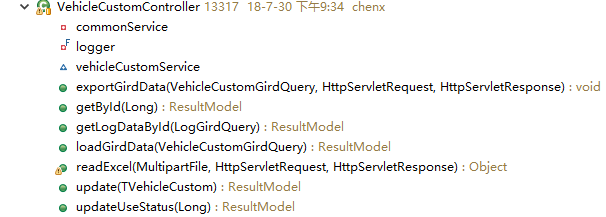
表zm\_project、表zm\_project\_build、表zm\_marketing\_project、表zm\_supplement\_order、表zm\_supplement\_order\_detail

#### 详细功能

1. 列表：查看信息列表，按每页10条分页显示。
2. 查询：通过司机名称、车牌号、班次类型进行查询。
3. 修改：选中一条应用信息点击“修改”按钮打开修改应用界面，正确修改应用信息后点击“确定”即可成功修改信息。
4. 导入：点击导入进入导入窗口，下载出导入模板，按照导入须知填写每一列的信息，然后导入窗口选择填好的Excel文件进行信息导入。
5. 导出：点击导出将把列表查出来的所有的数据导出到一个Excel表上。
6. 状态变更：列表上每个都有状态按钮修改，点击即可切换修改的可用状态。

#### 实现与接口

应用管理业务处理类定义如下：



主要应用接口定义如下：

方法定义：add (TDriverPlan)

功能说明：添加信息

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：exportGridData(VehicleCustomGripQuery, HttpServletRequest, HttpServletResponse)

功能说明：导出列表信息

方法定义：delete (Long)

功能说明：根据id删除信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：deleteAll(String)

功能说明：批量删除信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：driverExportGirdData (BranchGripQuery, HttpServletRequest, HttpServletResponse)

功能说明：司机上班统计信息导出接口

方法定义：getById(Long)

功能说明：根据id查询信息，返回查询到的对象

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1。

使用result作为KEY接收一个对象给前端处理

方法定义：getLogDataById(LogGripQuery)

功能说明：查询信息的操作日记

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：loadGripData(VehicleCustomGripQuery)

功能说明：查询信息列表，分页查询

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：readExcel(MultipartFile, HttpServletRequest, HttpServletResponse)

功能说明：导入应用信息，将导入成功或导入失败的问题记录成一个excel表并输出。

返回值： 返回导入文件无效的信息或返回excel表

方法定义：update(TVehicleCustom)

功能说明：自定义修改接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：updateUseStatus (TShift)

功能说明：根据ID启用禁用

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

## 供应商维护

维护供应商的基本信息，如联系方式，地址，分类等等。

### 设计思路

#### 程序描述

可以通过供应商的名称编号，联系人等信息进行查询

供应商列表如下。



#### 权限分配

全体人员可以使用。

#### 数据结构

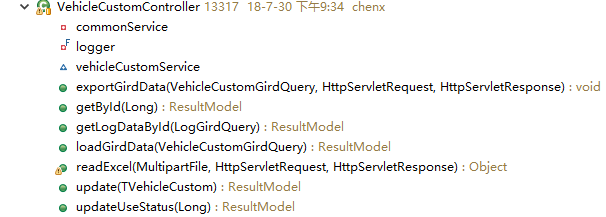
表zm\_supplier\_manager

#### 详细功能

1. 列表：查看信息列表，按每页10条分页显示。
2. 查询：通过司机名称、车牌号、班次类型进行查询。
3. 修改：选中一条应用信息点击“修改”按钮打开修改应用界面，正确修改应用信息后点击“确定”即可成功修改信息。
4. 导入：点击导入进入导入窗口，下载出导入模板，按照导入须知填写每一列的信息，然后导入窗口选择填好的Excel文件进行信息导入。
5. 导出：点击导出将把列表查出来的所有的数据导出到一个Excel表上。
6. 状态变更：列表上每个都有状态按钮修改，点击即可切换修改的可用状态。

#### 实现与接口

应用管理业务处理类定义如下：



主要应用接口定义如下：

方法定义：add (TDriverPlan)

功能说明：添加信息

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：exportGridData(VehicleCustomGripQuery, HttpServletRequest, HttpServletResponse)

功能说明：导出列表信息

方法定义：delete (Long)

功能说明：根据id删除信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：deleteAll(String)

功能说明：批量删除信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：driverExportGirdData (BranchGripQuery, HttpServletRequest, HttpServletResponse)

功能说明：司机上班统计信息导出接口

方法定义：getById(Long)

功能说明：根据id查询信息，返回查询到的对象

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1。

使用result作为KEY接收一个对象给前端处理

方法定义：getLogDataById(LogGripQuery)

功能说明：查询信息的操作日记

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：loadGripData(VehicleCustomGripQuery)

功能说明：查询信息列表，分页查询

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：readExcel(MultipartFile, HttpServletRequest, HttpServletResponse)

功能说明：导入应用信息，将导入成功或导入失败的问题记录成一个excel表并输出。

返回值： 返回导入文件无效的信息或返回excel表

方法定义：update(TVehicleCustom)

功能说明：自定义修改接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：updateUseStatus (TShift)

功能说明：根据ID启用禁用

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

## 采购订单

采购以采购订单的方式进行体现流程和采购的情况，到货付款情况等等。

### 设计思路

#### 程序描述

可以通过编号，供应商名称，合同编号，采购申请单等进行查询。

列表查询界面如下：



#### 权限分配

全体人员可以使用。

#### 数据结构

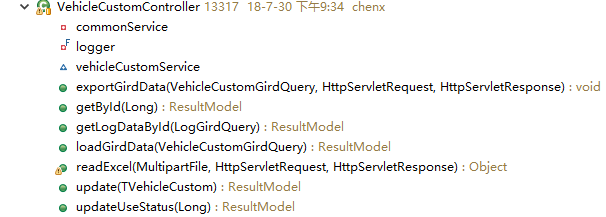
表zm\_purchasing\_order、表zm\_purchasing\_order\_detail

#### 详细功能

1. 列表：查看信息列表，按每页10条分页显示。
2. 查询：通过司机名称、车牌号、班次类型进行查询。
3. 修改：选中一条应用信息点击“修改”按钮打开修改应用界面，正确修改应用信息后点击“确定”即可成功修改信息。
4. 导入：点击导入进入导入窗口，下载出导入模板，按照导入须知填写每一列的信息，然后导入窗口选择填好的Excel文件进行信息导入。
5. 导出：点击导出将把列表查出来的所有的数据导出到一个Excel表上。
6. 状态变更：列表上每个都有状态按钮修改，点击即可切换修改的可用状态。

#### 实现与接口

应用管理业务处理类定义如下：



主要应用接口定义如下：

方法定义：add (TDriverPlan)

功能说明：添加信息

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：exportGridData(VehicleCustomGripQuery, HttpServletRequest, HttpServletResponse)

功能说明：导出列表信息

方法定义：delete (Long)

功能说明：根据id删除信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：deleteAll(String)

功能说明：批量删除信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：driverExportGirdData (BranchGripQuery, HttpServletRequest, HttpServletResponse)

功能说明：司机上班统计信息导出接口

方法定义：getById(Long)

功能说明：根据id查询信息，返回查询到的对象

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1。

使用result作为KEY接收一个对象给前端处理

方法定义：getLogDataById(LogGripQuery)

功能说明：查询信息的操作日记

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：loadGripData(VehicleCustomGripQuery)

功能说明：查询信息列表，分页查询

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：readExcel(MultipartFile, HttpServletRequest, HttpServletResponse)

功能说明：导入应用信息，将导入成功或导入失败的问题记录成一个excel表并输出。

返回值： 返回导入文件无效的信息或返回excel表

方法定义：update(TVehicleCustom)

功能说明：自定义修改接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：updateUseStatus (TShift)

功能说明：根据ID启用禁用

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

## 资产报表

资产的一些报表，如日报的汇总报表，按照材料类别的报表，资产的周/月报表。

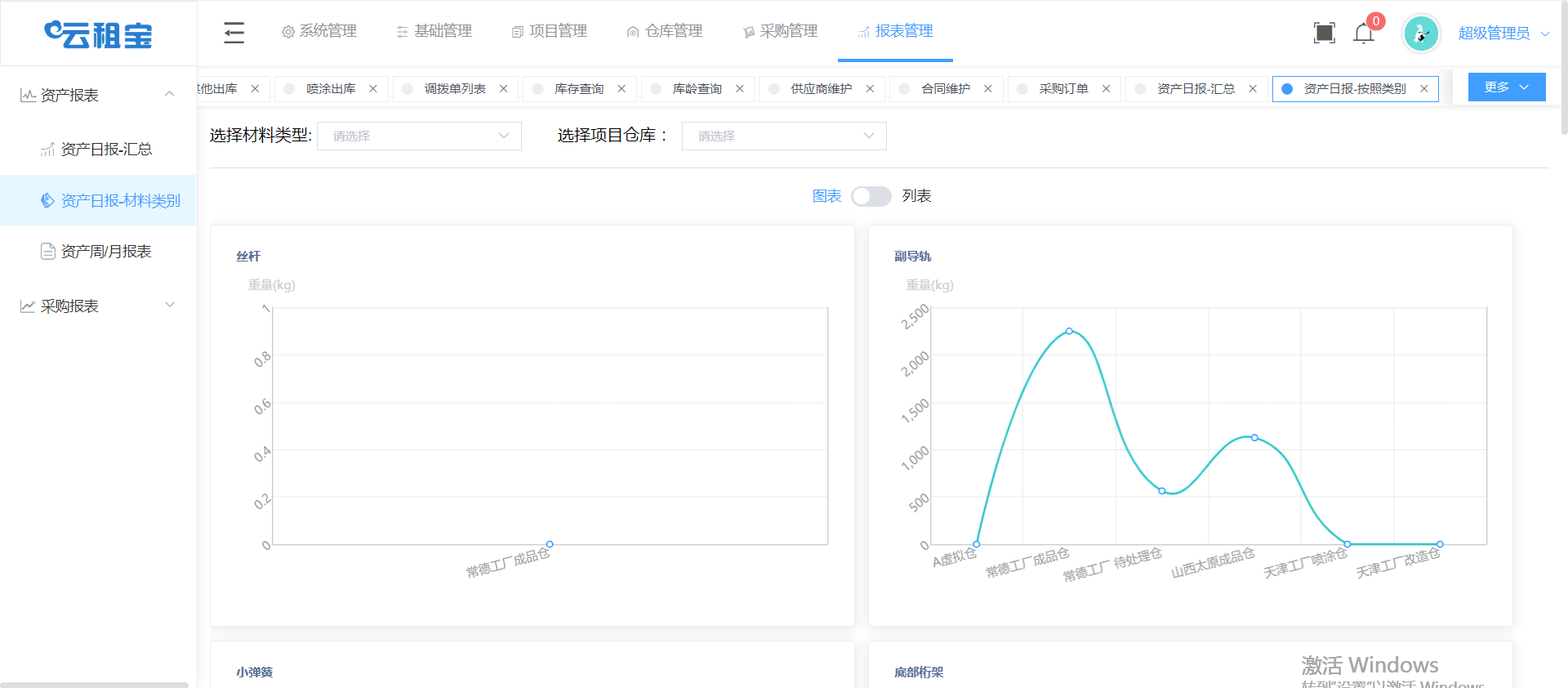
### 设计思路

#### 程序描述

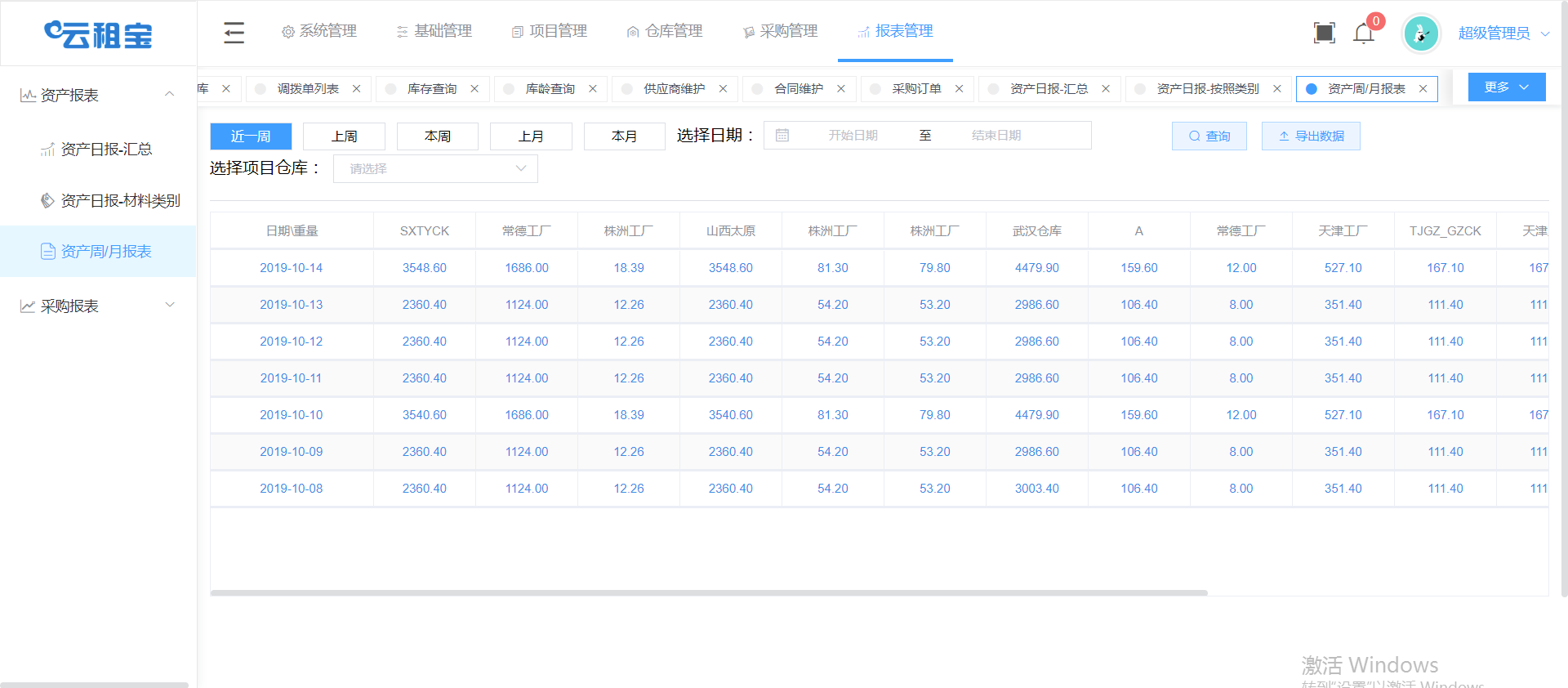
统计近几日的数据生成报表。



按照材料类别显示报表



按周和月统计报表



#### 权限分配

全体人员可以使用。

#### 数据结构

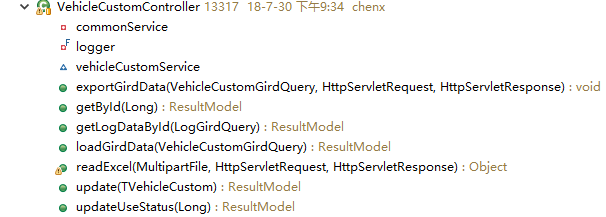
表zm\_asset\_management

#### 详细功能

1. 列表：查看信息列表，按每页10条分页显示。
2. 查询：通过司机名称、车牌号、班次类型进行查询。
3. 修改：选中一条应用信息点击“修改”按钮打开修改应用界面，正确修改应用信息后点击“确定”即可成功修改信息。
4. 导入：点击导入进入导入窗口，下载出导入模板，按照导入须知填写每一列的信息，然后导入窗口选择填好的Excel文件进行信息导入。
5. 导出：点击导出将把列表查出来的所有的数据导出到一个Excel表上。
6. 状态变更：列表上每个都有状态按钮修改，点击即可切换修改的可用状态。

#### 实现与接口

应用管理业务处理类定义如下：



主要应用接口定义如下：

方法定义：add (TDriverPlan)

功能说明：添加信息

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：exportGridData(VehicleCustomGripQuery, HttpServletRequest, HttpServletResponse)

功能说明：导出列表信息

方法定义：delete (Long)

功能说明：根据id删除信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：deleteAll(String)

功能说明：批量删除信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：driverExportGirdData (BranchGripQuery, HttpServletRequest, HttpServletResponse)

功能说明：司机上班统计信息导出接口

方法定义：getById(Long)

功能说明：根据id查询信息，返回查询到的对象

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1。

使用result作为KEY接收一个对象给前端处理

方法定义：getLogDataById(LogGripQuery)

功能说明：查询信息的操作日记

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：loadGripData(VehicleCustomGripQuery)

功能说明：查询信息列表，分页查询

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：readExcel(MultipartFile, HttpServletRequest, HttpServletResponse)

功能说明：导入应用信息，将导入成功或导入失败的问题记录成一个excel表并输出。

返回值： 返回导入文件无效的信息或返回excel表

方法定义：update(TVehicleCustom)

功能说明：自定义修改接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：updateUseStatus (TShift)

功能说明：根据ID启用禁用

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

## 采购报表

按照月份统计采购的情况。

### 设计思路

#### 程序描述

可以通过按周，月份，获取日期等统计报表，且可以进行导出。

列表查询界面如下：





#### 权限分配

全体人员可以使用。

#### 数据结构

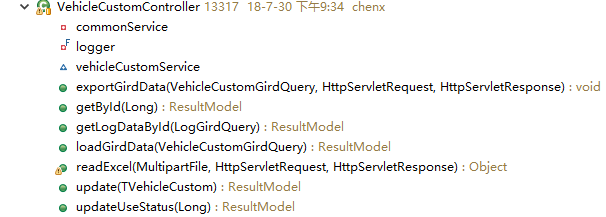
表zm\_asset\_management

#### 详细功能

1. 列表：查看信息列表，按每页10条分页显示。
2. 查询：通过司机名称、车牌号、班次类型进行查询。
3. 修改：选中一条应用信息点击“修改”按钮打开修改应用界面，正确修改应用信息后点击“确定”即可成功修改信息。
4. 导入：点击导入进入导入窗口，下载出导入模板，按照导入须知填写每一列的信息，然后导入窗口选择填好的Excel文件进行信息导入。
5. 导出：点击导出将把列表查出来的所有的数据导出到一个Excel表上。
6. 状态变更：列表上每个都有状态按钮修改，点击即可切换修改的可用状态。

#### 实现与接口

应用管理业务处理类定义如下：



主要应用接口定义如下：

方法定义：add (TDriverPlan)

功能说明：添加信息

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：exportGridData(VehicleCustomGripQuery, HttpServletRequest, HttpServletResponse)

功能说明：导出列表信息

方法定义：delete (Long)

功能说明：根据id删除信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：deleteAll(String)

功能说明：批量删除信息，

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

方法定义：driverExportGirdData (BranchGripQuery, HttpServletRequest, HttpServletResponse)

功能说明：司机上班统计信息导出接口

方法定义：getById(Long)

功能说明：根据id查询信息，返回查询到的对象

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1。

使用result作为KEY接收一个对象给前端处理

方法定义：getLogDataById(LogGripQuery)

功能说明：查询信息的操作日记

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：loadGripData(VehicleCustomGripQuery)

功能说明：查询信息列表，分页查询

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

使用result作为KEY接收一个数组给前端处理

方法定义：readExcel(MultipartFile, HttpServletRequest, HttpServletResponse)

功能说明：导入应用信息，将导入成功或导入失败的问题记录成一个excel表并输出。

返回值： 返回导入文件无效的信息或返回excel表

方法定义：update(TVehicleCustom)

功能说明：自定义修改接口

返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

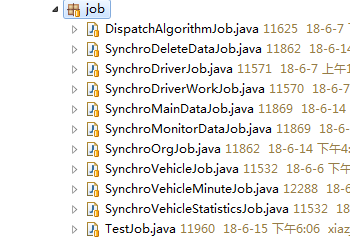
方法定义：updateUseStatus (TShift)

功能说明：根据ID启用禁用

# 返回值： ResultModel(JSON格式，前台统一处理),成功返回sub\_code=0失败返回sub\_code=1

# 系统接口

## 定时同步接口



### 获取任务调度计算法执行结果

#### 业务功能

通过此定时器接口可以获取任务调度结果信息

#### 业务流程

### 定时删除日志等数据

#### 业务功能

通过此定时器接口操作数据库进行定时删除历史数据

#### 业务流程

### 同步用户数据同步

#### 业务功能

通过此定时器条用api接口同步用户的数据

#### 业务流程

#### 接口方法

##### 方法名称明

// updatetime开始时间、orgcode\_like组织编号

public ListUserDTO> getUserList(String updatetime, String orgcode\_like) throws Exception {

List<DriverDTO> result = null;

//请求path

final String path = "/v1/base/driver/driver\_list";

JSONObject json = new JSONObject();

if(StringUtils.isNotBlank(updatetime)){

json.put(OpenApiConstants.QUERY\_PARAM\_UPDATETIME, updatetime);

}

if(StringUtils.isNotBlank(orgcode\_like)){

json.put(OpenApiConstants.QUERY\_PARAM\_ORGCODE\_LIKE, orgcode\_like);

}

//Body内容

String body = json.toJSONString();

Request request = this.getPostRequest(path, body);

Response response = openApiClient.execute(request);

String responseBody = response.getBody();

if(StringUtils.isNotBlank(responseBody)){

JSONObject responseResult = JSON.parseObject(responseBody);

Integer code = responseResult.getInteger(OpenApiConstants.RESULT\_CODE);

Integer subCode = responseResult.getInteger(OpenApiConstants.RESULT\_SUB\_CODE);

if(code == 0 && subCode == 0){

JSONArray data = responseResult.getJSONArray(OpenApiConstants.RESULT\_DATA);

if(data != null && !data.isEmpty())

result = JSON.parseArray(data.toJSONString(), DriverDTO.class);

}else{

throw new RuntimeException("调用查询司机列表接口返回异常"+responseBody);

}

}

return result;

}





### 仓储数据同步

#### 业务功能

通过此定时器接口操作数据库进行定时同步司机工作的数据

#### 业务流程

### 系统首页数据同步

#### 业务功能

通过此定时器接口操作数据库进行首页需要显示的数据进行定时同步

#### 业务流程

### 监控大屏数据同步

#### 业务功能

通过此定时器接口操作数据库进行缓存任务信息进行监控大屏数据同步

#### 业务流程

### 定时同步机构数据

#### 业务功能

通过此定时器接口操作Openapi进行同步机构数据

#### 业务流程

#### 接口方法

##### 方法名称明

@Override

@Transactional

public void synchroOrgData() throws Exception {

List<OrgDTO> dataList = openApiManager.getOrgList(InterfaceConfig.ORG\_CODE);

List<String> ids = new ArrayList<>();

if(dataList != null && !dataList.isEmpty()) {

//删除

for (OrgDTO orgDTO : dataList) {

System.out.println(orgDTO.getId()+"---------------");

ids.add(orgDTO.getId()+"");

}

taskProducerMapper.deleteExistOrg(ids);

List<OrgTreeNode> orgTreeNodes = this.listToTree(dataList);

List<TOrg> existOrgList = tOrgMapper.selectByExample(new TOrgExample());

Map<String, TOrg> existOrgMap = MapUtils.listToMap(existOrgList, "g7sid");

this.storingData(orgTreeNodes, null, existOrgMap);

}

}





### 定时同步项目数据

#### 业务功能

通过此定时器接口操作openapi进行定时同步数据

#### 业务流程

#### 接口方法

##### 方法名称明

public List<TruckDTO> getTrunckList(String orgcode\_like, String carnum\_like,

Integer page\_no, Integer page\_size, String updatetime\_ge) throws Exception {

List<TruckDTO> result = null;

//请求path

final String path = "/v1/base/truck/truck\_list";

JSONObject json = new JSONObject();

if(StringUtils.isNotBlank(orgcode\_like)){

json.put(OpenApiConstants.QUERY\_PARAM\_ORGCODE\_LIKE, orgcode\_like);

}

if(StringUtils.isNotBlank(carnum\_like)){

json.put(OpenApiConstants.QUERY\_PARAM\_CARNUM\_LIKE, carnum\_like);

}

if(page\_no != null && page\_size != null && page\_size > 0){

json.put(OpenApiConstants.QUERY\_PARAM\_PAGE\_NO, page\_no);

json.put(OpenApiConstants.QUERY\_PARAM\_PAGE\_SIZE, page\_size);

}

if(StringUtils.isNotBlank(updatetime\_ge)){

json.put(OpenApiConstants.QUERY\_PARAM\_UPDATETIME\_GE, updatetime\_ge);

}

//Body内容

String body = json.toJSONString();

Request request = this.getPostRequest(path, body);

Response response = openApiClient.execute(request);

String responseBody = response.getBody();

if(StringUtils.isNotBlank(responseBody)){

JSONObject responseResult = JSON.parseObject(responseBody);

Integer code = responseResult.getInteger(OpenApiConstants.RESULT\_CODE);

Integer subCode = responseResult.getInteger(OpenApiConstants.RESULT\_SUB\_CODE);

if(code == 0 && subCode == 0){

JSONArray data = responseResult.getJSONArray(OpenApiConstants.RESULT\_DATA);

if(data != null && !data.isEmpty())

result = JSON.parseArray(data.toJSONString(), TruckDTO.class);

}else{

throw new RuntimeException("调用查询查询列表接口返回异常"+responseBody);

}

}

return result;

}





### 定时同步报表数据

#### 业务功能

通过此定时器接口操作数据库进行定时维护的任务数，里程等数据

#### 业务流程

### 定时同步基础数据

#### 业务功能

通过此定时器接口操作数据库进行定时油耗等数据

#### 业务流程

# 数据库设计

## 数据库产品

Mysql 5.2

## 数据库对象命名规则

第1位：对象标识

T 表

L 逻辑表

I 索引

V 视图

S 存储过程

F 函数

P 包

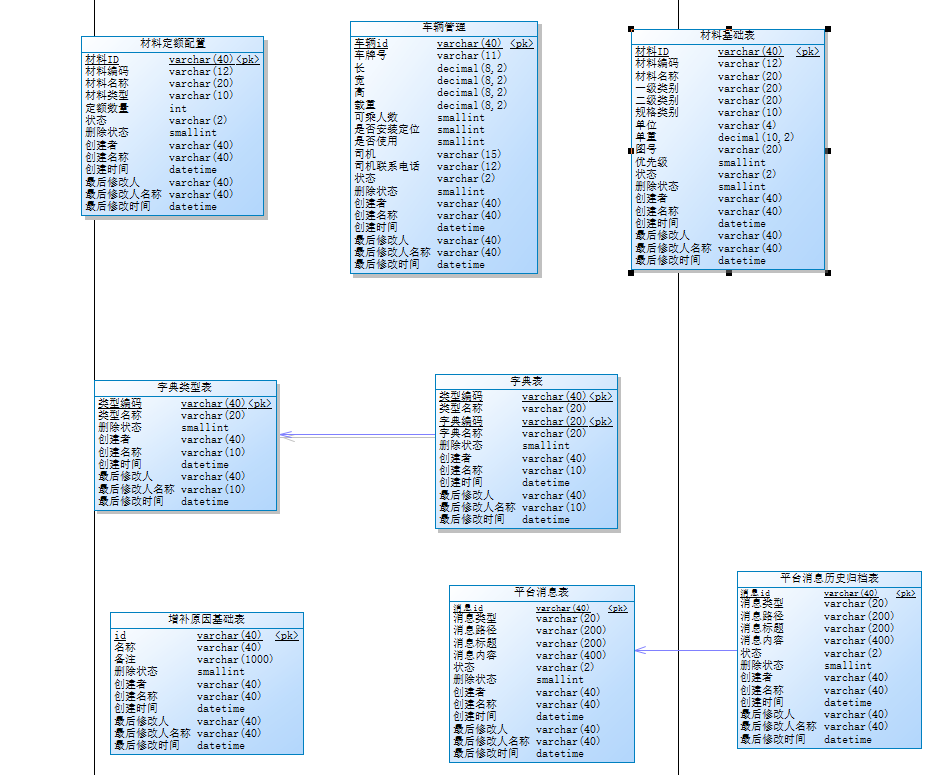
Q 序列号

第2位：\_

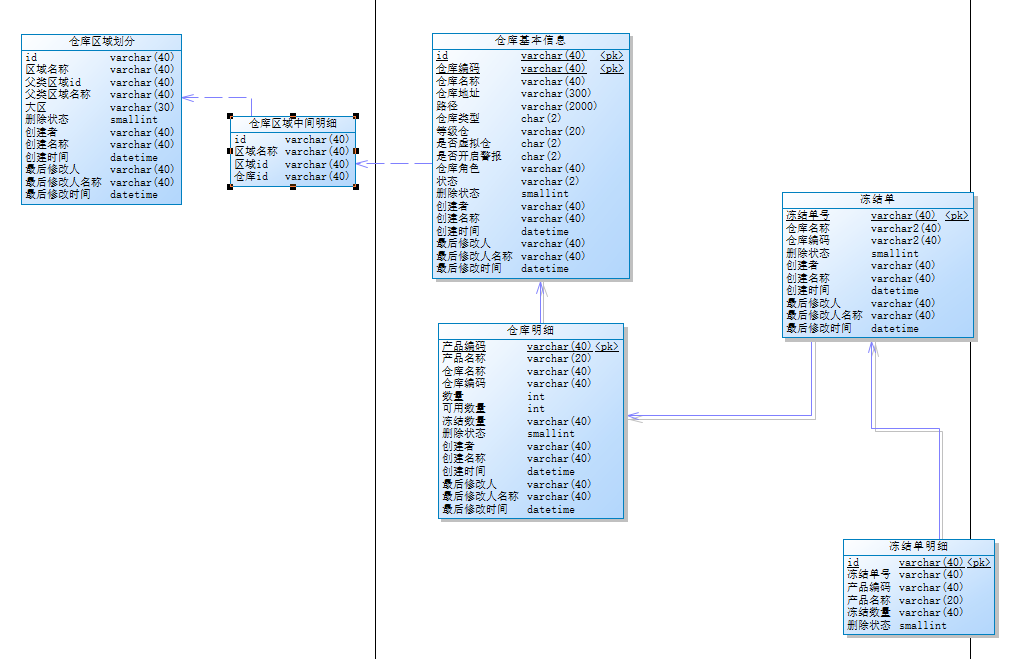
第3位开始对象标志名称（一个单词、两个单词全部或部分内容，多个单词使用“\_”分隔）

## 数据表关联关系图

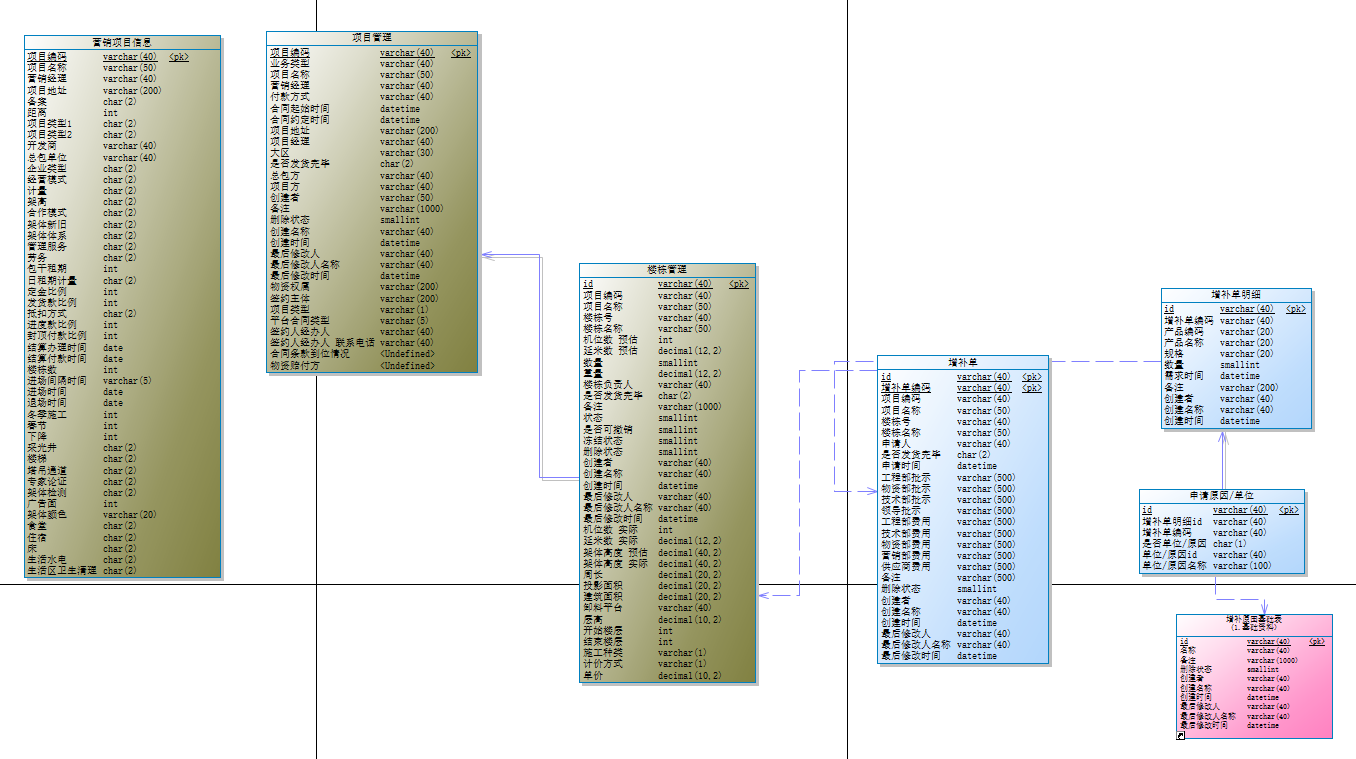
### 基础数据



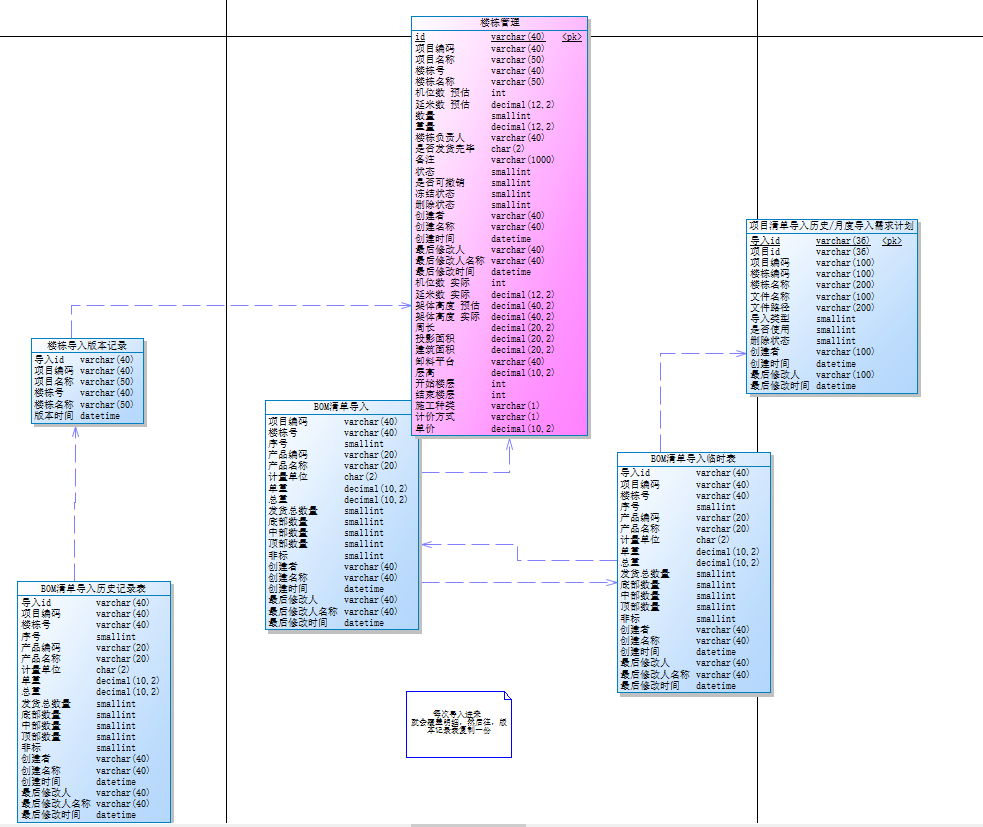
### 仓库管理



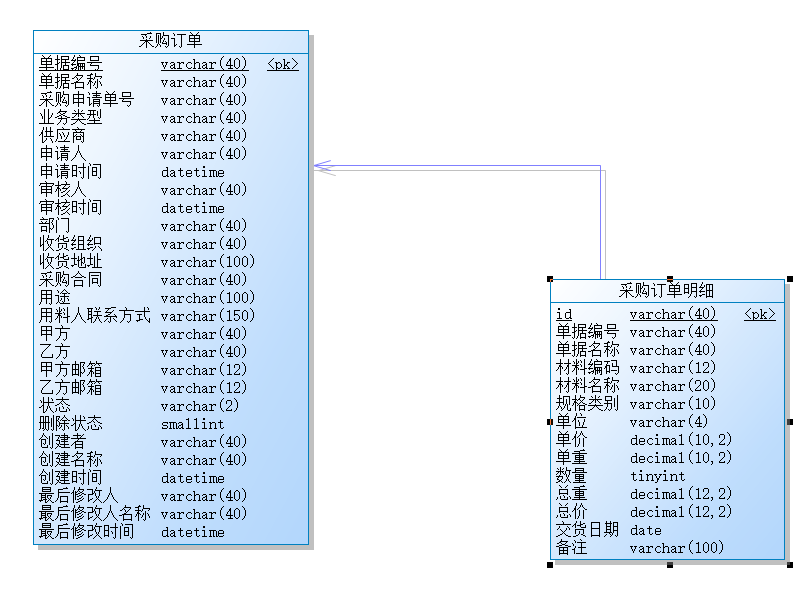
### 项目管理

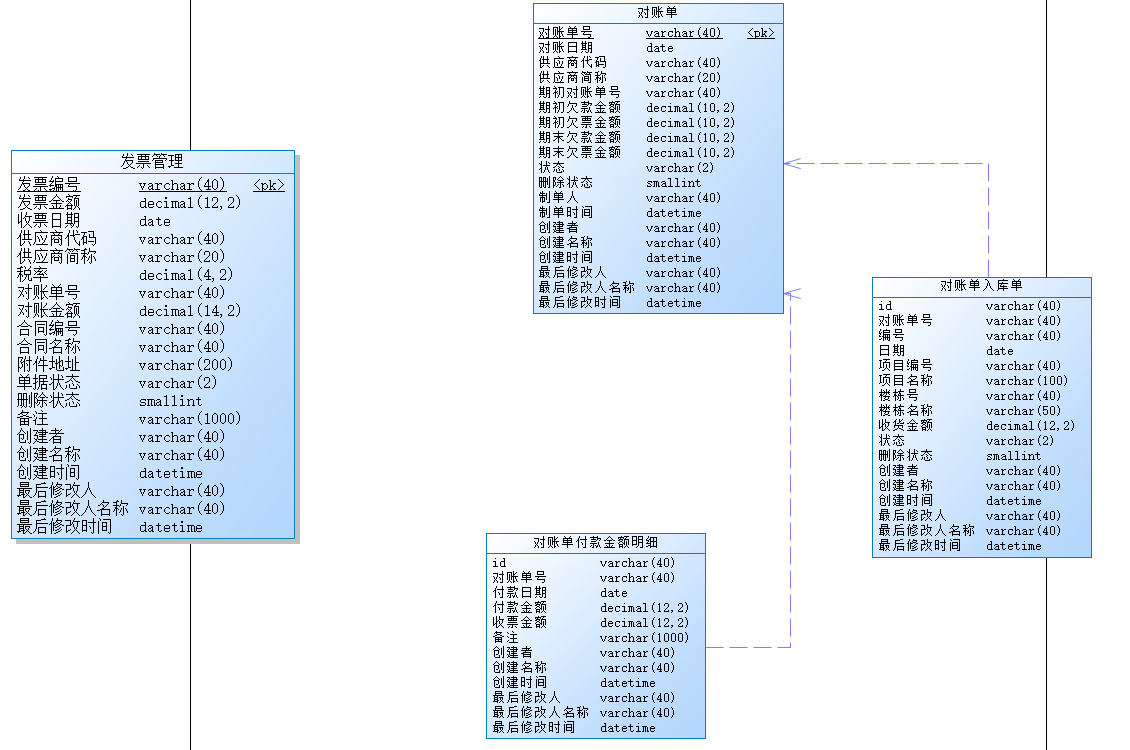


### BOOM管理

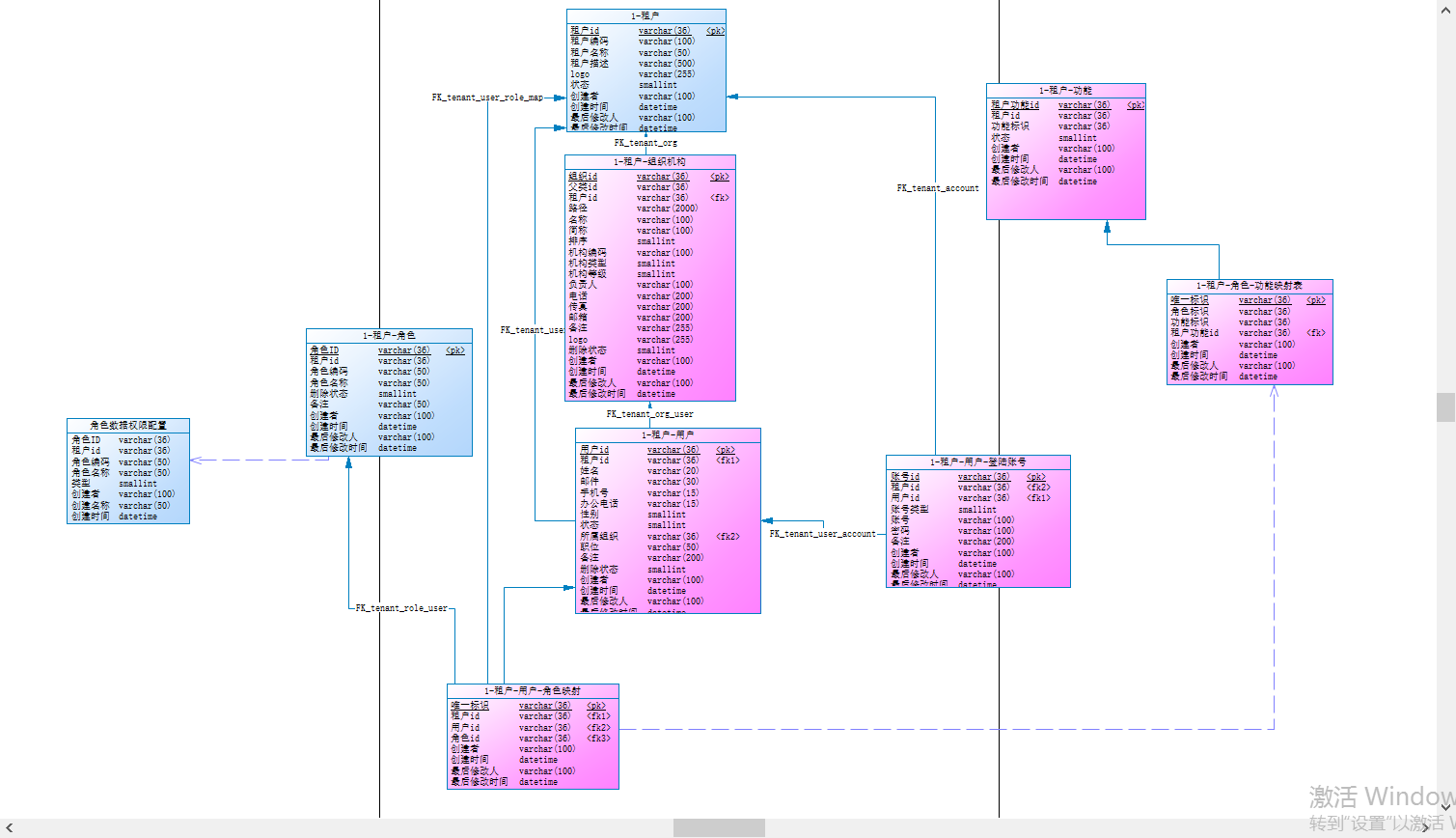


### 采购管理





### 系统权限管理



### 定时任务



## 数据表清单

列出所有数据表的名称和基本功能

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | itr\_work\_order | 工单表 | | itr\_work\_order\_file | 工单表附件表/或者评论附件上传 | | itr\_work\_order\_question\_lib | 工单问题库 | | itr\_work\_order\_question\_lib\_answer | 问题库部分答案 | | itr\_work\_order\_review | 工单评论 | | itr\_work\_order\_type\_config | 工单类型配置 | | p\_function | 平台支持的功能,console管理的mes功能菜单 | | p\_menu | 平台支持的菜单 | | p\_menu\_function | 平台支持的菜单与功能映射 | | sec\_function | console后台管理系统的功能 | | sec\_menu | 后台管理系统的菜单 | | sec\_menu\_function | 后台管理系统的菜单与功能映射 | | sec\_org | 组织机构表 | | sec\_role | console后台管理系统的角色表 | | sec\_role\_function | 后台管理系统的角色与功能映射 | | sec\_user | 后台管理-用户表 | | sec\_user\_account | 用户登陆账号表 | | sys\_dict | 字典表 | | sys\_dict\_type | 字典类型 | | t\_basics\_role | 基础角色（操作权限、数据权限配置） | | t\_basics\_role\_function | 后台管理系统的角色与功能映射 | | t\_org | 租户的组织机构表 | | t\_role | wms租户创建的角色 | | t\_role\_basics\_mid | 租户-角色-基础角色映射表 | | t\_role\_function | mes后台管理系统的角色与租户拥有的功能映射 | | t\_tenant | 租户的基本信息表 | | t\_tenant\_function | p\_function分配给租户的功能表 | | t\_user | 租户的用户表 | | t\_user\_account | 租户用户登陆账号表，支持多账号登录 | | t\_user\_dep\_role | 租户-角色-部门映射表 | | t\_user\_role | 租户的用户角色 | | zm\_add\_reasons | 增补原因基础表 | | zm\_allot\_warehouse\_order | 调拨单 | | zm\_allot\_warehouse\_order\_detail | 调拨单明细 | | zm\_asset\_management | 资产管理报表，每日统计插入 | | zm\_bom\_detail | BOM清单导入 | | zm\_bom\_detail\_history | BOM清单导入历史记录表 | | zm\_bom\_detail\_temp | BOM清单导入临时表 | | zm\_bom\_implement | BOM表执行查询 | | zm\_build\_record | 楼栋导入版本记录 | | zm\_climbing\_project | 爬架项目 | | zm\_deliver\_order | 发货通知单 | | zm\_deliver\_order\_agreement | 发货通知单通知协议 | | zm\_deliver\_order\_agreement2 | 发货通知单通知协议 | | zm\_deliver\_order\_detail | 发货通知单明细 | | zm\_dict | 字典表 | | zm\_dict\_type | 字典类型表 | | zm\_freeze\_order | 冻结单 | | zm\_freeze\_order\_detail | 冻结单明细 | | zm\_goods\_base | 材料基础表 | | zm\_goods\_base\_temp | 材料基础表零时 | | zm\_import\_record | 项目清单导入历史/月度导入需求计划 | | zm\_in\_warehouse\_detail | 入库单明细 | | zm\_in\_warehouse\_detail\_ext | 改制入库单 | | zm\_in\_warehouse\_inventory | 出入库明细清点人员 | | zm\_in\_warehouse\_order | 入库单 | | zm\_invoice\_manager | 发票管理 | | zm\_library\_age | 库龄查询 每天进行统计一次 异步跑数据 | | zm\_marketing\_project | 营销项目信息 | | zm\_material\_setting | 材料定额配置 | | zm\_month\_plan | 月度需求计划 | | zm\_month\_plan\_detail | 月度需求计划明细 | | zm\_month\_plan\_detail\_temp | 月度需求计划明细导入临时表 | | zm\_out\_warehouse\_order | 项目出库单 | | zm\_out\_warehouse\_order\_detail | 出库单明细 | | zm\_payment\_record | 对账单付款记录 | | zm\_payment\_request | 付款申请 | | zm\_pingku\_detail\_warehouse | 平库仓库表 | | zm\_project | 项目管理 | | zm\_project\_build | 楼栋管理 | | zm\_purchase\_contract | 供应商管理 | | zm\_purchase\_contract\_other | 采购合同条款 | | zm\_purchasing\_decomposition | 采购申请 | | zm\_purchasing\_decomposition\_detail | 采购申请明细表 | | zm\_purchasing\_order | 采购订单 | | zm\_purchasing\_order\_detail | 采购订单明细 | | zm\_requirement\_order | 需求计划单 | | zm\_requirement\_order\_detail | 需求计划单明细 | | zm\_requirement\_order\_detail\_ext | 需求计划明细扩展表 | | zm\_statement\_account | 对账单 | | zm\_statement\_account\_detail | 对账单付款金额明细 | | zm\_statement\_account\_in | 对账单入库单 | | zm\_supplement\_order | 增补单 | | zm\_supplement\_order\_detail | 增补单明细 | | zm\_supplement\_order\_impl | 增补单表执行查询 | | zm\_supplement\_order\_reasons | 申请原因/单位 | | zm\_supplier\_manager | 供应商管理 | | zm\_unit\_price | 采购单价信息 | | zm\_vehicle\_manager | 车辆管理 | | zm\_warehouse\_base | 仓库基本信息 | | zm\_warehouse\_detail | 仓库明细 | | zm\_warehouse\_hisprice | 仓库明细调价历史 | | zm\_warehouse\_region | 仓库区域划分 | | zm\_warehouse\_region\_mid | 仓库区域中间明细 | |

# 设计约定与要求

## 开发规范

J2EE标准开发规范。

Apache OpenApi开发规范。

阿里巴巴开发规约。

## 程序结构

\

|\_bin（编译生成的文件）

|\_lib（使用的库）

|\_src（源代码）

|\_com

|\_xxx

|\_...

|\_docs（文档）

## 版本要求

填写说明：运行该系统时对软件、硬件环境和支持环境的要求。

操作系统：CentOS 7

数据库应用：mysql 5.2

内存数据库：redis 4.1

负载均衡：keepalived + Nginx

JDK： SUN JDK 1.8

## 其他共性要求

填写说明：

服务器时钟同步策略

# 附录Ⅰ-参考文档

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **文档名称** | **版本** | **作者** | **备注** | **附件** |
|  |  |  |  |  |
|  |  |  |  |  |