浙 江 工 商 大 学

计算机与信息工程学院

**《移动计算方向专业实践》**



项目名称： 智能信号灯(移动端)

组 员 1： 叶镓瑜 1912190222

组 员 2： 杨曦 1912190234

班 级： 计科1901

指导教师： 韩建伟

日 期： 2022.6.6

# 1. 项目介绍【叶镓瑜】

## 1.1 背景

我国城市交通现状特点——城市交通基础设施建设速度跟不上迅速增长的交通需求，常规公共交通萎缩、出租车和私家车迅速增加、轨道交通开始起步、交通管理技术水平低。具体表现为：

**一、车辆保有量剧增，车型种类复杂、混合交通严重：**据公安部统计，截至2021年3月，全国机动车保有量达3.78亿辆，其中汽车2.87亿辆；机动车驾驶人4.63亿人，其中汽车驾驶人4.25亿人。中国是一个发展中的国家，贫富差距现状没有被完全解决，因而适应不同人群、不同消费需求的各种车辆混杂在道路交通中，各类交通混行导致交通管理压力剧增，交通形势日趋严峻，每年因交通事故引起的直接损失折款多达几百亿。

**二、信号灯人工配时，无法满足日趋复杂的城市交通：**当前我国城市的信号灯时长全部采用人工设置的方式。固定的信号灯时长和配时方式无法满足变化的城市交通，拥堵程度不同的路段采用统一的信号灯时长，同时人工配时的方式效率低下，无法及时地缓解路口的交通压力，从而造成了城市郊区和市区通畅程度的两极分化，增加了城市在交通管理上的开支，也造成了城市居民生活成本的提高。

**三、城市信号灯管控体系不全，交通决策存在误差：**城市的信号灯管控需要建立在完善的体系之上，信号灯的配时不仅仅取决于路口当前的拥堵状况，还涉及到天气、周边路口交通状况、路口车辆类型以及路口历史的交通状况等因素，这些数据决定了交通决策的正确性，同时也是信号灯配时的根据。但我国城市目前尚未有一套合理完善的信号灯管控体系，片面的交通信息导致了交管部门在交通决策上的失误，使城市交通拥堵现状无法缓解，交通管理效率低下。

**四、交通指挥缺乏实时性，无法及时做出指挥和人员调度，无法应对交通紧急状况：**城市交通的管理由指挥部门和现场警员组成，为了合理及时且高效的分配警力维护交通秩序，在交通指挥上就要做到实时联调，在交通信息上就需要做到全面同步。但目前我国缺少在信息上的同步，交通信息的时差性导致指挥部门无法做出及时的人力分配，无力应对交通紧急状况；同时现场警员无法同步获取到现场的交通信息，无法正确的完成交通疏导，从而导致交通管理效率低下，交通管理开支增加。

## 1.2项目需求分析

### 1.2.1 业务需求

随着经济发展，城市交通越来越复杂，为了方便道路通畅和交通有序进行，需要打造一款**功能强大、效率极高**的智能信号灯管理系统来维持良好的交通秩序。

本系统希望能够统计监控道路数据，方便管理人员掌握更多道路信息，远距离便能观测操控信号灯配时，同时还能尽量减少人力劳动，实现自动化操控配时。道路数据对道路管控十分重要，因此本系统还希望能对道路历史记录进行系统管理，并且进一步积累整个城市的交通网络相关数据，汇总成以年为形式统计数据，形成相应算法在季节（暴风、暴雨、浓雾等）变化中进行预测功能。

除基本的信号灯操控功能之外，本系统还希望能对道路中的违法车辆和特殊车辆进行监控记录，紧急情况下能为特殊车辆开辟绿波带。

道路交通变幻莫测，因此本系统希望能打造一个能适应绝大多数路口的智能信号灯系统，希望它能推广到更多城市中，并且能对交通管控起到有效的帮助，为城市建设提供力量。

**1.2.2 角色分析**

本系统主要的使用人员有两种，分别为：**指挥调度人员**和**交通控制人员**。

指挥调度人员主要使用中控监测（大屏）端工作。其使用小屏操控大屏显示内容，可以在大屏上浏览城市的交通总体情况，查看各地交通监控，检查车辆违规情况，起到决策作用。同时指挥调度人员作为管理员还有管理交通控制岗账号的权力，大屏的功能涵盖小屏所有的功能。

道路现场的交通控制工作人员则需要使用移动管理（小屏）端进行工作。当现场工作人员在道路指挥交通时，可以通过小屏端来了解当前道路的一些情况，方便做出对道路疏通的决策。当需要通过信号灯疏通道路时，指挥调度人员可以在小屏端人工控制信号灯时长。当遇到特殊车辆有紧急情况时，工作人员会收到来自系统的警报，及时查看情况并为之设置绿波带。

**1.2.3 角色分析**

|  |
| --- |
|  |
| 图1-1 交通控制人员用例分析 |

**1.2.4 用户故事**

|  |  |
| --- | --- |
| **用户故事** | **用户故事描述** |
| 登录系统 | 作为一名交通控制人员，我希望能够登录系统进行工作 |
| 忘记密码找回 | 作为一名交通控制人员，我希望能找回自己忘记的密码 |
| 查看城市总览情况 | 作为一名交通控制人员，我希望我能查看城市的统计数据 |
| 查看城市热力图 | 作为一名交通控制人员，我希望能查看城市车辆热力图 |
| 查看点位详情 | 作为一名交通控制人员，我希望查看点位的详细信息 |
| 查看违规车辆详情 | 作为一名交通控制人员，我希望能查看违规车辆的详细信息 |
| 查看信号灯详情 | 作为一名交通控制人员，我希望能查看每个信号灯的详细信息 |
| 信号灯智能配时 | 作为一名交通控制人员，我希望信号灯能根据路口车流量和天气、周边路口状况等因素智能配置时长 |
| 查看信号灯配时详情 | 作为一名交通控制人员，我希望能查看信号灯配时详情 |
| 设置信号灯配时情况 | 作为一名交通控制人员，我希望我能人工设置信号灯配时 |
| 查看绿波带 | 作为一名交通控制人员，我希望能查看某一路段的绿波带情况 |
| 设置绿波带 | 作为一名交通控制人员，我希望在遇到特殊车辆紧急情况时为其设置绿波带 |
| 查看历史点位流量 | 作为一名交通控制人员，我希望能查看某一点位指定历史时间的点位流量信息和变化 |
| 查看车辆类型统计 | 作为一名交通控制人员，我希望查看所有车辆类型统计和特殊车辆类型统计 |
| 查看违规类型统计 | 作为一名交通控制人员，我希望能查看违规车辆类型的统计情况 |
| 查看历史拥堵指数 | 作为一名交通控制人员，我希望能查看某一点位历史拥堵指数信息和变化 |
| 修改个人信息 | 作为一名交通控制人员，我希望能够修改个人信息 |
| 修改密码 | 作为一名交通控制人员，我希望能够修改登录密码 |

**1.2.5 任务分工**

|  |  |
| --- | --- |
| **姓名** | **主要工作** |
| 叶镓瑜 | 负责产品规划，完成产品功能的概念设计和原型展示，实现项目前端的开发，负责文档编写和PPT视频制作 |
| 杨曦 | 负责项目源代码的管理，git上传下载，后端的部署，负责进行功能测试，负责文档编写和PPT视频制作 |

# 2.界面原型设计【叶镓瑜】

**2.1 总览**

总览页由车辆总数、车辆类型、交通点位信息和告警信息四块组成。图表和文字的结合使城市的交通信息更加直观易懂。

|  |
| --- |
|  |
| 图2-1总览 |

**2.2 交通监测**

交通监测由城市热力图、点位监测和违规监测三部分构成。

**城市热力图**

城市热力图显示当前城市的交通车流量分布状况

|  |
| --- |
|  |
| 图2-2城市热力图 |

**点位监测**

点位监测显示城市某个点位的具体信息。控制岗人员可以切换点位位置、查看点位实时监控视频、了解点位基本信息和点位的车流量数据

|  |
| --- |
|  |
| 图2-3点位监测 |

**违规监测**

违规监测显示当前城市车辆违规信息列表，并对违规的信息进行分类统计

|  |
| --- |
|  |
| 图2-4违规监测 |

控制岗人员可以点击相应的违规信息来查看违规详情

|  |
| --- |
|  |
| 图2-5违规监测（详情） |

**2.3 信号灯管理**

信号灯管理由信号灯详情、配时管理和绿波带配置三块内容组成

**信号灯详情**

信号灯详情展示指定点位的信号灯基本信息、点位的实时画面、信号灯配时变化、信号灯当前时长和状态，以及该点位的告警信息。控制岗人员可以输入切换信号灯

|  |
| --- |
|  |
| 图2-6信号灯详情 |

**配时管理**

配时管理是对信号灯时长和运行状态进行管理。控制岗人员可以根据当前信号灯的基本状态信息和系统推荐的配时方案进行配时。信号灯的运行状态分为人工配时和系统托管两种，在系统托管下，系统将根据路口的车流量并结合疏导算法来动态调整信号灯的配时。控制岗通过点击修改按钮，以弹窗的形式配置信号灯的时长信息。

|  |
| --- |
|  |
| 图2-7配时管理 |

|  |
| --- |
|  |
| 图2-8配时管理（弹窗） |

**绿波带配置**

控制岗人员可以输入点位和时长，设置一条城市交通绿波带

|  |
| --- |
|  |
| 图2-9绿波带设置 |

**2.4 数据统计**

数据统计由历史点位流量、历史热力图、历史拥堵指数和历史失衡指数四块组成

**历史点位流量**

历史点位流量展示指定路口一段时间内的点位车流量变化。采用折线图和表格的两种形式展示。

|  |
| --- |
|  |
| 图2-10历史点位流量 |

|  |
| --- |
|  |
| 图2-11历史点位流量（表格） |

**历史热力图**

控制岗人员可以查看指定路口一段时间内城市热力图的信息

|  |
| --- |
|  |
| 图2-12历史热力图 |

**历史拥堵指数**

控制岗人员可以查看指定点位的历史拥堵指数。采用折线图和表格的形式展示

|  |
| --- |
|  |
| 图2-13历史拥堵指数 |

|  |
| --- |
|  |
| 图2-14历史拥堵指数（表格） |

**历史失衡指数**

控制岗人员可以查看指定点位的历史失衡指数。采用折线图和表格的形式展示

|  |
| --- |
|  |
| 图2-15历史失衡指数 |

|  |
| --- |
|  |
| 图2-16历史失衡指数（表格） |

**2.5 设置**

设置由个人信息、修改密码和退出系统三部分组成

**个人信息**

控制岗人员可以修改个人的基本信息

|  |
| --- |
|  |
| 图2-17个人信息 |

**修改密码**

控制岗人员可以修改个人的密码

|  |
| --- |
|  |
| 图2-18修改密码 |

# 3.系统架构设计【杨曦】

**3.1 开发架构设计**

|  |
| --- |
| Generated |
| 图1-3 开发架构设计图 |

上图是系统的开发架构设计图，依次包含了四个层：底层硬件、数据层、技术支撑层、业务层、通讯层、技术组件层以及展示层。

**1.底层硬件**

底层硬件主要由视频监控设备、无人机等组成，负责采集实时路口交通流、监控车辆、人员行为等基础数据，并利用无线通信，由主流4G/5G无线网络为后台提供现场实时监控。

**2.数据层**

数据层基于Mysql与redis为业务层提供数据操作以及数据的存储

**3.技术支撑层**

技术支撑层负责结合数据层对整个智能信号灯管理系统提供技术支撑。主要包括基于YOLOv5视觉分析的车辆相关信息识别技术，基于GAT的智能交通流量预测和交通疏导技术和基于SUMO的交通疏导效果评估设计。

**4.业务层**

业务层是基于Springboot + Mysql + Mybatis-plus的后端开发框架处理与业务相关的部分，包括本系统所需要的所用功能上的算法和计算过程，根据业务逻辑将系统功能划分为上图所示模块，例如交通监测模块、多屏互动模块等等。

**5.通讯层**

前后端通信依靠http协议和webscoket协议，http协议负责普通的前后端通信请求，例如查询交通数据，webscoket协议负责两个客户端的互动，它允许服务端主动向客户端推送数据。

**6.技术组件层**

技术组件层主要包括前端技术的选型，web端基于vue框架，移动端基于uniapp框架，两者均在vue语法上建立。

**7.展示层**

展示层包括移动管理端、中控监测端和信号灯，其中移动管理端由交通控制岗人员使用，使用者可以通过它监测点位交通情况、调整信号灯配时以及控制中控监测端屏幕内容显示，中控监测端由指挥调度人员使用，作为交通指挥中心大屏幕显示系统实现远程监控、指挥城市交通状况。信号灯由支持层控制，接受来自支持层发送的配时方案信号，以此来调整红绿灯时长。

**3.2 前端核心框架与技术**

本项目中前端采用Vue + Vue-router + Vuex + Axios以及构建工具Vue-cil作为系统前端开发的核心框架

Vue.js 是一个JavaScript MVVM库，是一套构建用户界面的渐进式框架。它是以数据驱动和组件化的思想构建的，采用自底向上增量开发的设计。相比于Angular.js，Vue.js提供了更加简洁、更易于理解的API，使得我们能够快速地上手并使用Vue.js；同时比起 React + Redux 相对复杂的架构，Vue.js 更加轻量级也更加容易上手，是初创项目的首选前端框架。

Vue 的核心库只关注视图层，它不仅易于上手，还便于与第三方库或既有项目整合。另外，Vue.js的模块化和组件化开发，能大大增加代码的复用性，允许我们使用小型、独立和通常可复用的组件构建大型应用。

本项目中使用Vue-router来实现领域实体之间的交互关系。添加Vue-router完成后，将组件映射到路由然后指定渲染路径。通过注入路由器，可以在任何组件内通过this.\$router 访问路由器，也可以通过 this.\$route 访问当前路由。

本项目中使用Vuex进行了集中式存储管理应用的所有组件的状态，Vuex 是专门为 Vue.js 设计的状态管理库，以利用 Vue.js 的细粒度数据响应机制来进行高效的状态更新。主要是为了组件之间的交流，同时把组件的耦合性降到最低。对于前后端实时通讯数据保存以及组件交流提供了有效的管理。

本项目中使用Axios，它是基于 Promise 的 HTTP 请求客户端，可同时在浏览器和 Node.js 中使用。axois客户端支持保护安全免受 CSRF/XSRF 攻击,这一特点保障了数据高安全性、高保密性。

**3.3 数据库和后端，后端与前端交互**

Mybatis-plus是持久层框架Mybatis的一个插件，在Mybatis的基础上做了增强，用于快速开发数据库接口。

通过Restful API的规范以websocket建立稳定连接进行交互，Restful API相关文档见链接。<http://www.ruanyifeng.com/blog/2011/09/restful.html>

**3.4 后端核心框架与技术**

本项目中后端采用SSM作为系统后端开发的核心框架。

SSM 是 Spring + SpringMVC + Mybatis集成的框架。

|  |
| --- |
| Generated |
| 图2-2 后端核心框架图 |

本项目中后端采用了Springboot + Mysql + Mybatis-plus作为系统后端开发的核心框架。

SpringBoot是一个开发基于Spring框架的应用的快速开发框架，相对于普通的Spring应用，使用SpringBoot构建的Spring应用可以直接打包为一个独立可执行的jar或war包，使用java -jar命令即可运行，不需要管理依赖的第三方库，也不需要依赖外部容器来启动应用。且SpringBoot内置了web容器，不需要依赖外部的Web容器部署。SpringBoot启动应用时，会推断应用类型，并检测引入的Spring模块或第三方库，当条件满足时自动加载Spring模块或第三方库的组件到容器中，以提供给应用使用。SpringBoot提供了应用部署产品环境运行所必须的日志、指标、健康检查、外部化配置等特性，为部署后的运维提供工具支持，支撑应用尽可能快的部署到产品环境。

**3.5 数据库选择**

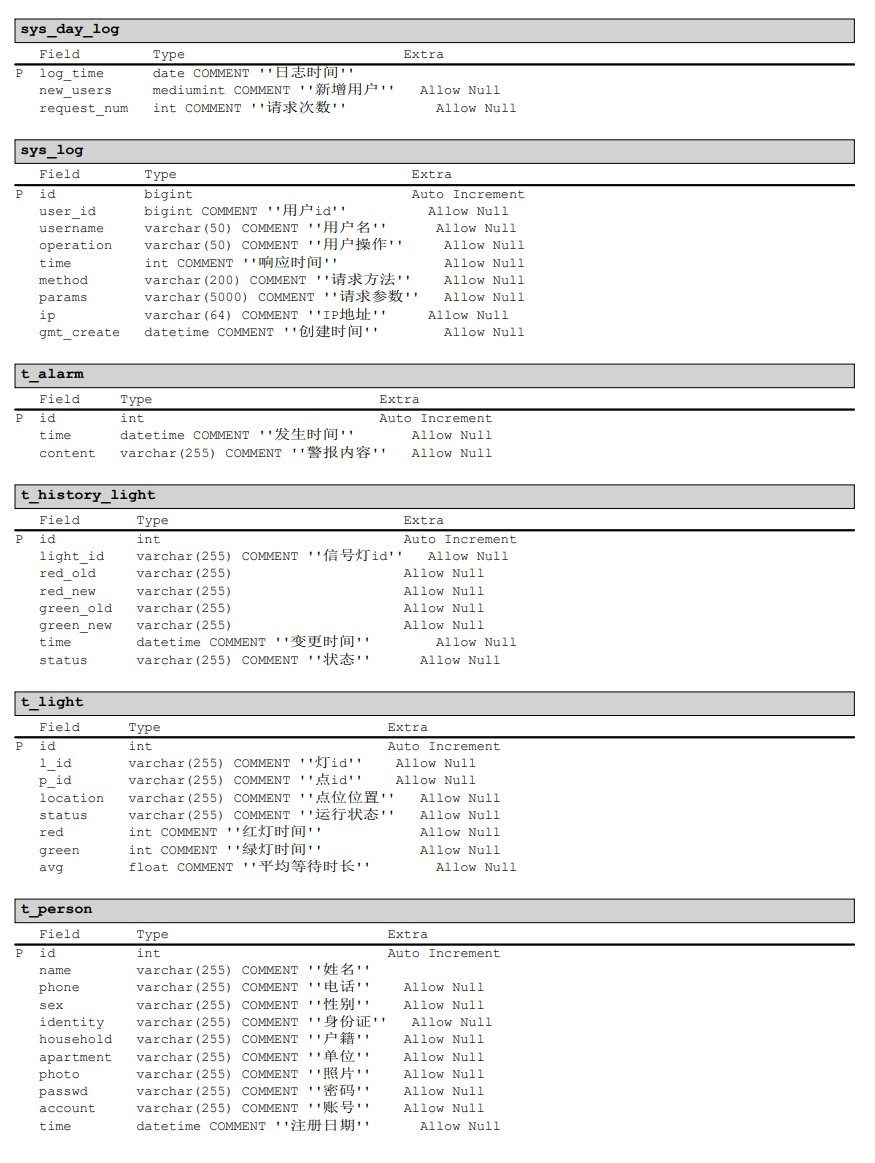
本项目使用mysql数据库，Mysql是一个 MySQL是一个多用户、多线程的关系型数据库，对于商业和个人用户是免费的，且跨平台，易部署。

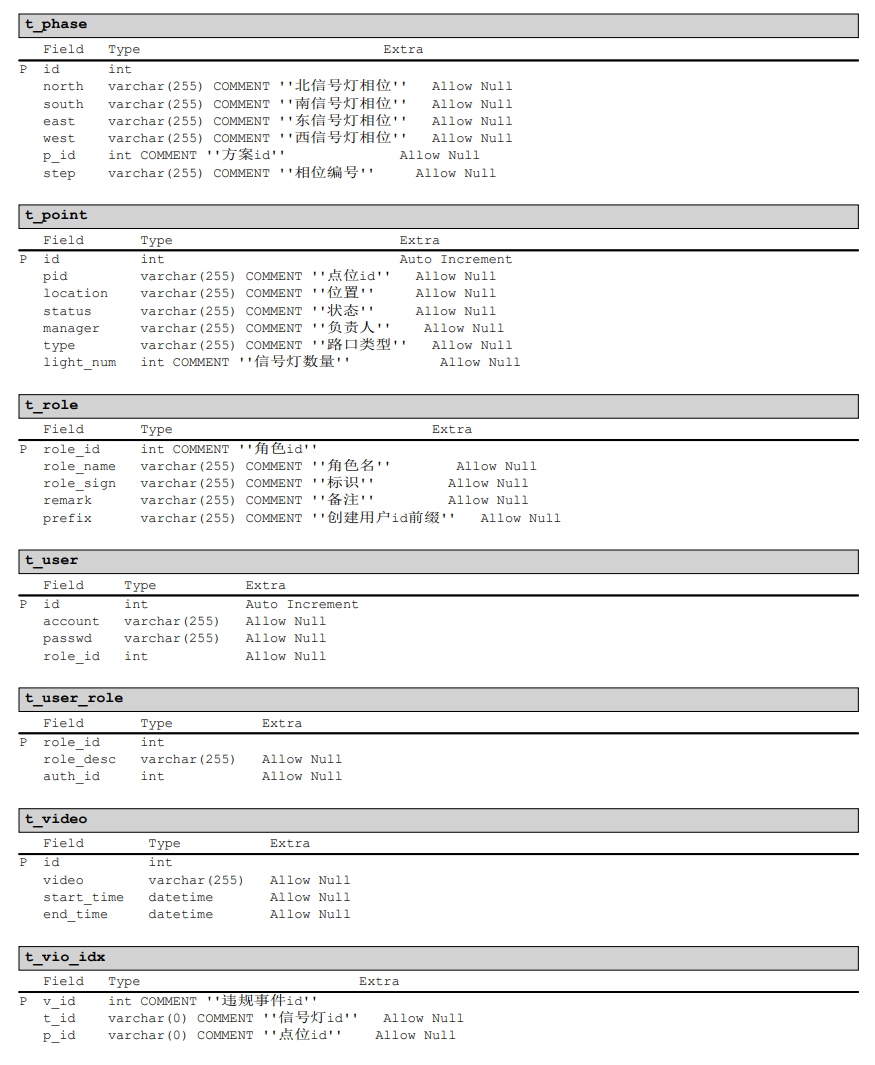
# API设计【杨曦】

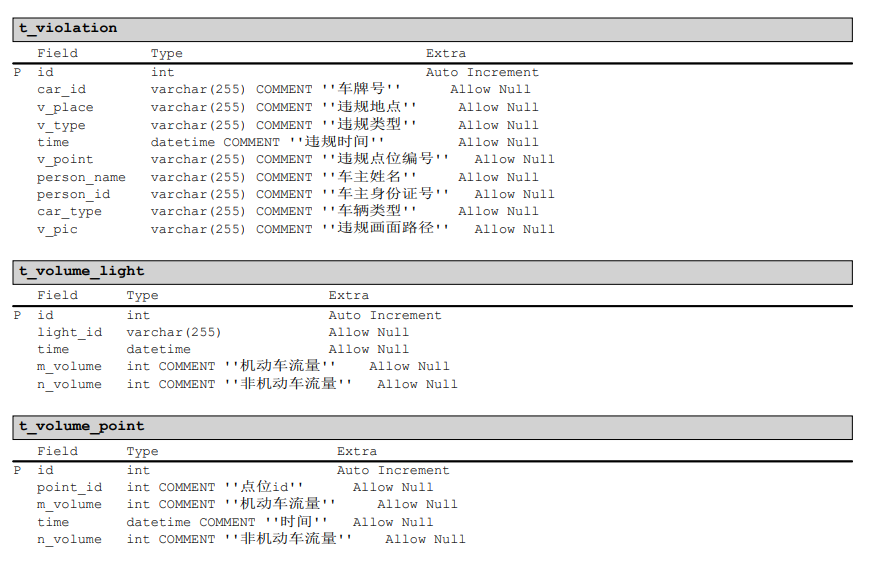
|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **接口名称** | **路径** | **请求方式** |
| 1 | 检索信号灯基本信息 | /light/index | POST |
| 2 | 设置绿波带 | /light/green\_belt | GET |
| 3 | 配置信号灯配时 | /light/duration | GET |
| 4 | 获取信号灯流量信息 | /light/volume | GET |
| 5 | 获取信号灯列表 | /light/list | GET |
| 6 | 获取信号灯变更历史 | /light/history | GET |
| 7 | 获取信号灯状态信息 | /light/index\_info | GET |
| 8 | 交通概况 | /msg/overview | GET |
| 9 | 获取点位通行状态信息 | /msg/passinfo | GET |
| 10 | 获取城市车辆 | /msg/motorNum | GET |
| 11 | 概况统计 | /msg/stt | GET |
| 12 | 交通指数 | /msg/figure | GET |
| 13 | 交通警报信息 | /msg/alarm | GET |
| 14 | 获取监控视频 | /monitor | GET |
| 15 | 获取点位详情 | /point/detail | GET |
| 16 | 获取违规列表子项 | /violation/{id} | GET |
| 17 | 获取违规信息列表 | /violation/list | POST |
| 18 | 添加人员 | /user/person | POST |
| 19 | 获取人员信息 | /user/index | GET |
| 20 | 登录接口 | /user/login | POST |
| 21 | 更改密码 | /user/passwd | POST |
| 22 | 获取人员列表 | /user/list | GET |
| 23 | 统计信号灯各段时长 | /statistic/light\_duration | GET |
| 24 | 统计拥堵指数 | /statistic/congestion | GET |
| 25 | 统计城市车流量 | /statistic/city\_volume | GET |
| 26 | 统计违规项 | /statistic/violation | GET |
| 27 | 获取城市热力数据 | /statistic/heat | GET |

详细请见[http://altairxixi.top:8081/swagger-ui.html#/](http://altairxixi.top:8081/swagger-ui.html" \l "/)

# 数据库设计【杨曦】







# 6.移动端的实现【叶镓瑜】

**6.1 页面布局**

移动端采用顶部导航栏的形式，呈现上下1：6的优质视觉比例。在项目开发中单独编写导航栏代码，在后续的各个界面中将其作为组件导入。从而提高了前端开发的效率。

导航栏按照功能模块分成总览、交通监测、信号灯管理、数据统计和设置五个模块。效果如下：

|  |
| --- |
|  |
| 图6-1页面布局 |

**6.2 总览**

总览页由车辆总数、车辆类型、交通点位信息和告警信息四块组成。图表和文字的结合使城市的交通信息更加直观易懂。

总览界面的在开发过程中要注意图表的引入。在此次项目开发中，引入的是Apache ECharts提供的图表。引入的过程是按照官方文档的教程进行。引入echarts包后，则可以在界面中引入相关的图表样式。

例如在总览界面中车辆类型引用了饼状图，示例代码如下：

1. **<view** class="titlebox"**>**
2. **<view** class="charttitle"**>**车辆类型**</view>**
3. **</view>**
4. **<view** class="charts-box"**>**
5. **<qiun-data-charts**
6. type="pie"
7. :chartData="chartData\_pie"
8. background="#041439"
9. **/>**
10. **</view>**

其中*type=“pie”* 则代表了图表的类型为饼图，并通过*:chartData="chartData\_pie"*绑定图表数据到charData\_pie中。

需要注意的是图表的数据形式需要十分规范。当时在处理图表数据时，因为格式的错误导致了图表无法正常的显示。规范的数据格式如下：

1. chartData\_pie:{
2. series: [
3. {
4. "data": [
5. {
6. "name": "小轿车",
7. "value": 50
8. },
9. {
10. "name": "特殊车辆",
11. "value": 30
12. },
13. {
14. "name": "自行车",
15. "value": 20
16. },
17. {
18. "name": "摩托车",
19. "value": 18
20. },
21. {
22. "name": "电瓶车",
23. "value": 8
24. },{
25. "name": "大客车",
26. "value": 18
27. },{
28. "name": "大货车",
29. "value": 18
30. }
31. ]
32. }
33. ]
34. }

最终呈现的效果如下：

|  |
| --- |
|  |
| 图6-2 图表呈现效果 |

同样，车辆数量和交通点位信息也是采用引入图表的形式呈现。

告警信息是采用列表的形式展现，即通过遍历服务端传送的告警信息进行呈现。

最终，总览界面的整体效果如下：

|  |
| --- |
|  |
| 图6-3 总览 |

**6.3 交通监测**

交通监测由城市热力图、点位监测和违规监测三部分构成。

**城市热力图**

城市热力图显示当前城市的交通车流量分布状况

热力图的引进是此次开发中最难的部分之一。项目中采用了百度地图提供的API，引入流程则是借鉴了百度地图的官方文档。项目中我将地图和热力点位数据挂载到页面中，这样就提高了渲染效率。此外因为地图数据较大，在项目中我使用了动态引入相关库文件，避免页面的展示效果低下。

1. mounted(){
2. // if (typeof window.echarts === 'function') {
3. //  this.initEcharts()
4. // } else {
5. // //   // 动态引入较大类库避免影响页面展示
6. //  const script3 = document.createElement('script')
7. //  // view 层的页面运行在 www 根目录，其相对路径相对于 www 计算
8. //  script3.src = 'static/echarts.js'
9. //  script3.onload = this.initEcharts.bind(this)
10. //  document.head.appendChild(script3)
11. //  }

14. **var** script=document.createElement("script")
15. script.type='text/javascript'
16. script.src="http://api.map.baidu.com/api?v=2.0&type=webgl&ak=AcLhbaEjGo3At21EZmFK5SfNb2IkT4sp&callback=init"
17. document.head.appendChild(script)
19. **var** script2=document.createElement("script")
20. //script2.src="https://unpkg.com/mapvgl/dist/mapvgl.min.js"
21. script2.src="https://code.bdstatic.com/npm/mapvgl@1.0.0-beta.140/dist/mapvgl.min.js"
22. document.head.appendChild(script2)
24. **var** script4=document.createElement("script")
25. script4.src="https://mapv.baidu.com/build/mapv.min.js"
26. //script2.src="https://code.bdstatic.com/npm/mapvgl@1.0.0-beta.140/dist/mapvgl.min.js"
27. document.head.appendChild(script4)
29. **var** script5=document.createElement("script")
30. script5.src="https://mapv.baidu.com/gl/examples/static/common.js"
31. //script2.src="https://code.bdstatic.com/npm/mapvgl@1.0.0-beta.140/dist/mapvgl.min.js"
32. document.head.appendChild(script5)

35. window.init=**this**.init
37. },

具体的热力数据则保存于*hangzhou-tracks.json*文件中，展示的是杭州钱塘区的热力信息。最终效果如下：

|  |
| --- |
|  |
| 图6-4 城市热力图 |

**点位监测**

点位监测显示城市某个点位的具体信息。控制岗人员可以切换点位位置、查看点位实时监控视频、了解点位基本信息和点位的车流量数据

此模块的要点是引入监控视频，这里采用了video组件进行视频的导入。要注意的是视频格式的问题，这里采用了mp4格式的视频进行导入，部分格式的视频不支持。

示例代码如下：

1. **<video** id="myVideo" :src="'/static/video/video.mp4'" loop="true" autoplay="true" style="margin-top: 5.55rpx; width: 240.11rpx;height: 200rpx;"
2. @error="videoErrorCallback" controls**></video>**

最终界面呈现效果如下：

|  |
| --- |
|  |
| 图6-5 点位监测 |

**违规监测**

违规监测显示当前城市车辆违规信息列表，并对违规的信息进行分类统计

|  |
| --- |
|  |
| 图6-6 交通监测（总） |

这里每一条数据根据id号进行页面跳转获取数据，控制岗人员可以查看违规详情

|  |
| --- |
|  |
| 图6-7 交通监测详情 |

**6.4 信号灯管理**

信号灯管理由信号灯详情、配时管理和绿波带配置三块内容组成

**信号灯详情**

信号灯详情展示指定点位的信号灯基本信息、点位的实时画面、信号灯配时变化、信号灯当前时长和状态，以及该点位的告警信息。控制岗人员可以输入切换信号灯

|  |
| --- |
|  |
| 图6-8 信号灯详情 |

**配时管理**

配时管理是对信号灯时长和运行状态进行管理。控制岗人员可以根据当前信号灯的基本状态信息和系统推荐的配时方案进行配时。信号灯的运行状态分为人工配时和系统托管两种，在系统托管下，系统将根据路口的车流量并结合疏导算法来动态调整信号灯的配时。控制岗通过点击修改按钮，以弹窗的形式配置信号灯的时长信息。

在开发过程中，这块最难处理的是界面布局的问题。元素的尺寸无法按照比例调节，于是在查阅网上资料后，在*APP.vue*文件中按照系统自动读取页面尺寸，从而保证了页面比例的正常化。

1. onLaunch: function() {
2. plus.key.hideSoftKeybord()
3. plus.screen.lockOrientation('landscape-primary')
4. uni.getSystemInfo({      //页面尺寸调整
5. success: function(e) {
6. // #ifndef MP
7. Vue.prototype.StatusBar = e.statusBarHeight;
8. if (e.platform == 'android') {
9. Vue.prototype.CustomBar = e.statusBarHeight + 50;
10. } else {
11. Vue.prototype.CustomBar = e.statusBarHeight + 45;
12. };
13. // #endif
14. // #ifdef MP-WEIXIN
15. Vue.prototype.StatusBar = e.statusBarHeight;
16. let custom = wx.getMenuButtonBoundingClientRect();
17. Vue.prototype.Custom = custom;
18. Vue.prototype.CustomBar = custom.bottom + custom.top - e.statusBarHeight;
19. // #endif
20. // #ifdef MP-ALIPAY
21. Vue.prototype.StatusBar = e.statusBarHeight;
22. Vue.prototype.CustomBar = e.statusBarHeight + e.titleBarHeight;
23. // #endif
24. }
25. })
26. },

|  |
| --- |
|  |
| 图6-9 配时管理 |

项目中采用弹窗的形式，实现配时修改和绿波带设置

弹窗运用了组件*uni-popup-dialog*实现，示例代码如下：

1. <uni-popup-dialog mode="" style="width: 333.33rpx;font-size:10rpx;" title="配时控制" message="成功消息"
2. :duration="2000" :before-close="true" @close="close" @confirm="confirm">
3. <view>
4. 其他代码...
5. </view>
7. </uni-popup-dialog>

|  |
| --- |
|  |
| 图6-10 配时修改 |

|  |
| --- |
|  |
| 图6-11 绿波带设置 |

大屏显示是采用websocket实时通信来实现大屏小屏互动。项目中在处理服务协议时遇到了较大的难处，最终发现是广播协议等错误的代码导致。示例代码如下：

1. **var** stompClient = **null**;
2. //加载完浏览器后  调用connect（），打开双通道
3. // $(function(){
4. // //打开双通道
5. // connect()
6. // })
7. //强制关闭浏览器  调用websocket.close（）,进行正常关闭
8. window.onunload = **function**() {
9. disconnect()
10. }
11. **function** connect(){
12. **var** userId=1;
13. **var** socket = **new** SockJS('http://127.0.0.1:9091/sbjm-cheng/endpointOyzc'); //连接SockJS的endpoint名称为"endpointOyzc"
14. stompClient = Stomp.over(socket);//使用STMOP子协议的WebSocket客户端
15. stompClient.connect({},**function**(frame){//连接WebSocket服务端
16. console.log('Connected:' + frame);
17. //通过stompClient.subscribe订阅/topic/getResponse 目标(destination)发送的消息
18. stompClient.subscribe('/user/' + userId + '/queue/getResponse',**function**(response){
19. **var** code=JSON.parse(response.body);
20. showResponse(code)
21. });
22. });
23. }
24. //关闭双通道
25. **function** disconnect(){
26. **if**(stompClient != **null**) {
27. stompClient.disconnect();
28. }
29. console.log("Disconnected");
30. }
31. **function** showResponse(message){
32. **var** response = $("#response");
33. response.append("<p>只有userID为"+message.userId+"的人才能收到</p>");
34. }

**6.5 数据统计**

数据统计由历史点位流量、历史热力图、历史拥堵指数和历史失衡指数四块组成

数据统计模块是调用了图表进行展示。图表数据皆来自于服务端的传输，下面是前后端连接的示例代码：

1. //获取城市车辆
2. **function** get\_msg\_motorNum(){
3. **var** promise = **new** Promise((resolve,reject) => {
4. **var** postData = '';
5. uni.request({
6. url:commoneUrl + 'msg/motorNum',
7. data:postData,
8. method:"GET",
9. dataType:'json',
10. header:{
11. 'content-type': 'application/json','token':uni.getStorageSync('token')},
12. success:**function**(res){
13. resolve(res.data);
15. },
16. error:**function**(e)
17. {
18. reject('网络出错');
19. }
20. });
21. });
22. **return** promise;
23. }

调用结果：

|  |
| --- |
|  |
| 图6-12 接口调用结果 |

界面效果展示：

|  |
| --- |
|  |
| 图6-13 点位车辆流量（折线图） |

|  |
| --- |
|  |
| 图6-14 点位车辆流量（表格） |

|  |
| --- |
|  |
| 图6-15 拥堵指数 |

|  |
| --- |
|  |
| 图6-16 车辆类型统计 |

|  |
| --- |
|  |
| 图6-17 违规类型统计 |

**6.6 设置**

设置由个人信息、修改密码和退出系统三部分组成

**个人信息**

控制岗人员可以修改个人的基本信息

|  |
| --- |
|  |
| 图6-18 个人信息 |

**修改密码**

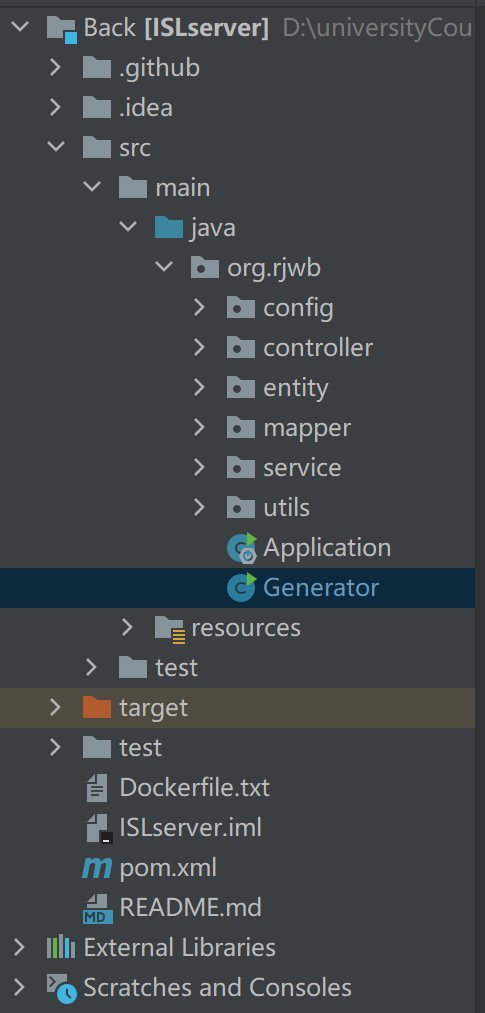
控制岗人员可以修改个人的密码

|  |
| --- |
|  |
| 图6-19 修改密码 |

# 7.后端的实现【杨曦】

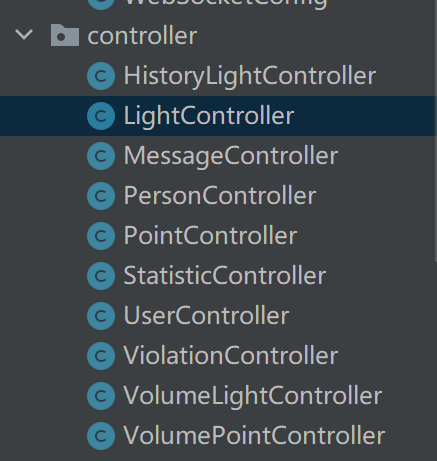
主要依靠的是springboot框架。首先项目是使用IDEA编译器开发的，因为其对springboot项目的支持性比较高。然后使用maven管理和构建项目。在当前的系统架构中，系统自上而下被分为控制器层、业务逻辑层、数据访问层、数据持久层。

**7.1 项目结构总览**



首先这个pom.xml是使用maven构建项目时需要依赖的一些包，在项目构建之初会自动下载，test文件夹下是一些测试文件。Target目录用于放置生成的目标文件，包括最后项目打包成的jar包。application是整个项目的启动类，类似于android的first application。Generator代码生成器可以帮我们生成entity,service,serviceImpl,mapper,mapper.xml。省去了建立一大堆实体类的麻烦。.github文件下放着workflows文件夹，文件夹下放着github action持续集成的yml文件。

**7.2 控制层**

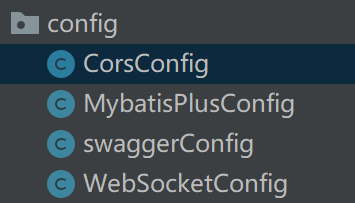


其中有信号灯控制，信息获取，点位等控制器。



以light为例，该模块有/list和/index两个功能。

**7.3 工具类**



spring boot有一些好用的工具，corsconfig可以解决一些跨域问题，MybatisPlus则可以方便的拼接数据库的分页。Swagger则可以方便地生成在线的API文档，方便测试。WebSocketConfig方便地实现前后端交互，因其采用了相较https，更为稳定的连接，节省了不少头部的开销。

# 8.系统测试【杨曦】

**8.1.用例执行**

用例一共63个，一共执行了三次，执行时间分别为2021年3月25日，共通过了41个用例，有22个未通过；2021年4月9日，通过了56个用例，9个未通过，2021年4月13日，通过了63个用例，0个未通过。详情请见表1：

注：OK表示用例执行通过，NG表示执行失败，即测试用例。由于新增一类的用例无法重复执行，每一次用例全部执行完毕后会将数据库还原，因此第二次第三次执行注册、新增的用例时不会因为已存在相同记录而失败。

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 2021年3月25日 | 2021年4月9日 | 2021年4月13日 |
| TA-0001 | OK | OK | OK |
| TA-0002 | OK | OK | OK |
| TA-0003 | OK | OK | OK |
| TA-0004 | OK | OK | OK |
| TA-0005 | NG | OK | OK |
| TA-0006 | NG | NG | OK |
| TA-0007 | OK | OK | OK |
| TA-0008 | OK | OK | OK |
| TA-0009 | OK | OK | OK |
| TA-0010 | OK | OK | OK |
| TA-0011 | OK | OK | OK |
| TA-0012 | NG | OK | OK |
| TA-0013 | NG | NG | OK |
| TA-0014 | NG | OK | OK |
| TA-0015 | NG | OK | OK |
| TA-0016 | NG | OK | OK |
| TA-0017 | OK | OK | OK |
| TA-0018 | OK | OK | OK |
| TA-0019 | OK | OK | OK |
| TA-0020 | OK | OK | OK |
| TA-0021 | OK | OK | OK |
| TA-0022 | OK | OK | OK |
| TA-0023 | NG | OK | OK |
| TA-0024 | OK | OK | OK |
| TA-0025 | OK | OK | OK |
| TA-0026 | NG | NG | OK |
| TA-0027 | OK | OK | OK |
| TA-0028 | OK | OK | OK |
| TA-0029 | OK | OK | OK |
| TA-0030 | OK | OK | OK |
| TA-0031 | OK | OK | OK |
| TA-0032 | NG | OK | OK |
| TA-0033 | OK | OK | OK |
| TA-0034 | OK | OK | OK |
| TA-0035 | OK | OK | OK |
| TA-0036 | NG | OK | OK |
| TA-0037 | OK | OK | OK |
| TP-0038 | NG | OK | OK |
| TP-0039 | OK | OK | OK |
| TP-0040 | OK | OK | OK |
| TP-0041 | NG | NG | OK |
| TP-0042 | OK | OK | OK |
| TP-0043 | NG | NG | OK |
| TP-0044 | OK | OK | OK |
| TP-0045 | OK | OK | OK |
| TP-0046 | OK | OK | OK |
| TP-0047 | NG | OK | OK |
| TP-0048 | OK | OK | OK |
| TP-0049 | OK | OK | OK |
| TP-0050 | OK | OK | OK |
| TP-0051 | NG | OK | OK |
| TP-0052 | NG | NG | OK |
| TP-0053 | NG | OK | OK |
| TP-0054 | OK | OK | OK |
| TP-0055 | OK | OK | OK |
| TP-0056 | OK | OK | OK |
| TP-0057 | NG | OK | OK |
| TP-0058 | OK | OK | OK |
| TP-0059 | OK | OK | OK |
| TP-0060 | OK | OK | OK |
| TI-0061 | NG | NG | OK |
| TI-0062 | NG | NG | OK |
| TI-0063 | NG | NG | OK |

表1 用例执行

**8.2.缺陷管理**

本次测试共发现了22个缺陷，直到发布已经全部修复，由于篇幅太多，这里就不全部列举了，仅仅展示部分缺陷。如下表2所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 用例编号 | 缺陷描述 | 打开时间 | 修复时间 | 关闭时间 |
| B-01 | TA-0005  TA-0006 | 无法通过手指触摸页面中央热力图调整大小 | 4月25日 | 5月2日 | 5月10日 |
| B-02 | TA-0012 | 交通控制人员无法手动控制信号灯配时 | 4月25日 | 5月2日 | 5月2日 |
| B-03 | TA-0023 | 交通控制人员无法查看交通点位历史流量折线图 | 4月25日 | 5月5日 | 5月5日 |
| B-04 | TP-0041 | 中控监测端无法从其他页面跳传到“交通预警页面” | 4月25日 | 5月10日 | 5月10日 |
| B-04 | TI-0061  TI-0062  TI-0063 | 移动管理端无法控制中控监测端 | 4月25日 | 5月10日 | 5月12日 |

**8.3.测试范围**

本次测试主要对系统的核心功能进行了测试，共覆盖了29个功能点，如表3所示：

|  |  |
| --- | --- |
| 功能点编号 | 功能点名称 |
| 1 | 交通控制人员查看城市统计数据 |
| 2 | 交通控制人员查看城市车辆热力图 |
| 3 | 交通控制人员查看点位详细信息 |
| 4 | 交通控制人员查看违规车辆列表 |
| 5 | 交通控制人员查看违规车辆详细信息 |
| 6 | 交通控制人员查看信号灯详情 |
| 7 | 交通控制人员查看信号灯配时信息 |
| 8 | 交通控制人员手动修改信号灯配时 |
| 9 | 交通控制人员智能控制信号灯配时 |
| 10 | 交通控制人员查看绿波带 |
| 11 | 交通控制人员设置绿波带 |
| 12 | 交通控制人员管理绿波带 |
| 13 | 交通控制人员查看点位历史流量 |
| 14 | 交通控制人员查看车辆类型统计 |
| 15 | 交通控制人员查看违规事件统计 |
| 16 | 交通控制人员查看历史拥堵指数 |
| 17 | 指挥调度人员查看城市总体情况信息 |
| 18 | 指挥调度人员查看城市监控 |
| 19 | 指挥调度人员查看信号灯详情 |
| 20 | 指挥调度人员修改信号灯配时 |
| 21 | 指挥调度人员设置信号灯智能配时 |
| 22 | 指挥调度人员设置绿波带 |
| 23 | 指挥调度人员查看交通预警信息 |
| 24 | 指挥调度人员查看交通数据统计情况 |
| 25 | 指挥调度人员查看交通控制人员列表 |
| 26 | 指挥调度人员查看交通控制人员详细信息 |
| 27 | 指挥调度人员添加新的交通控制人员详细信息 |
| 28 | 页面进入 |
| 29 | 移动管理端控制中控监测端 |

# 9.系统部署【杨曦】

**9.1数据库部署**

1、首先在服务器端下载docker,即使用apt-get install docker命令。

2、 搜索docker镜像仓库中提供的mysql

docker search mysql

3、下载mysql镜像

docker pull mysql

4、创建并运行mysql容器

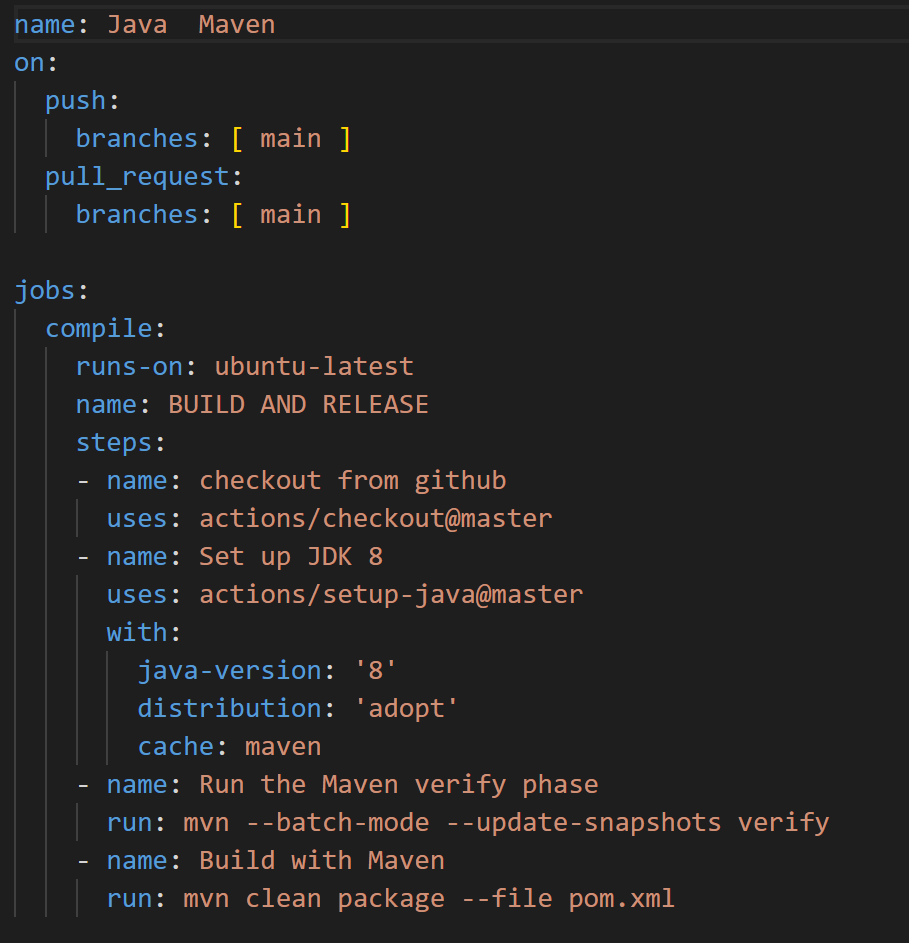
docker run -d --name myMysql -v /data/mysql:/var/lib/mysql -e MYSQL\_ROOT\_PASSWORD=admin -p 3306:3306 mysql

5、启动已创建的mysql容器

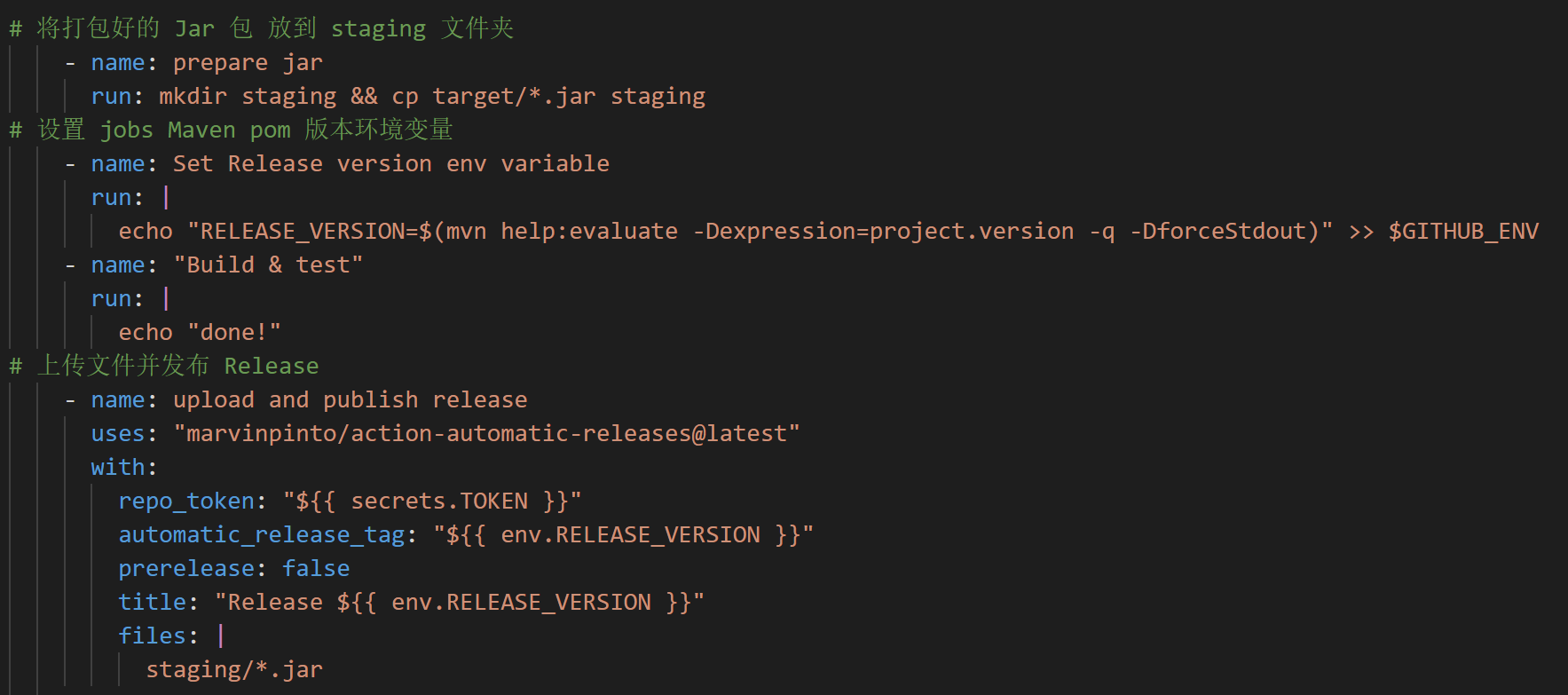
docker start myMysql

1. 用navicat连接数据库，最后运行sql文件。

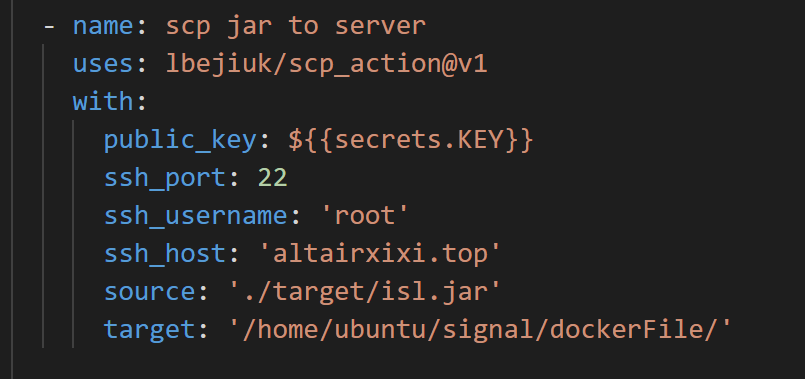
**9.2 后端部署**



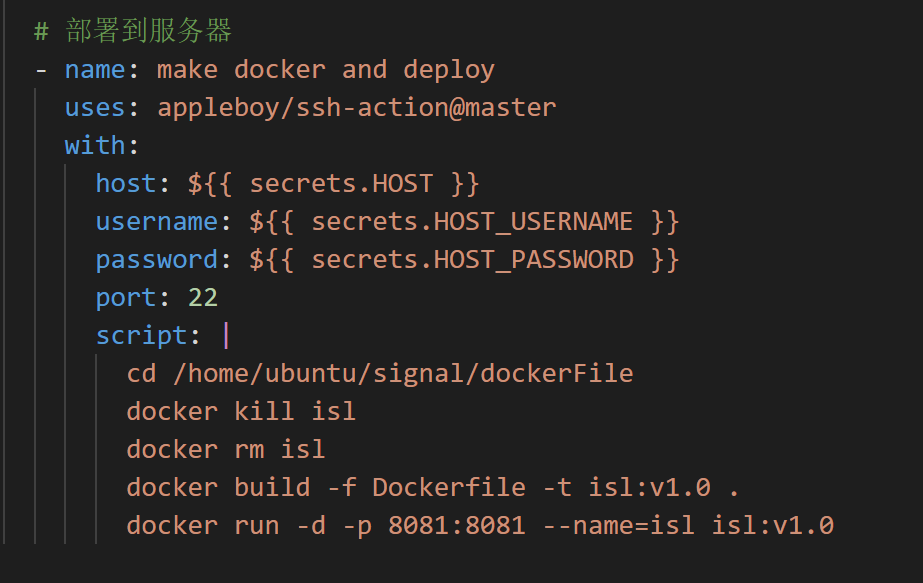
首先是mvn -verify和 mvn clean package命令，测试打包代码。



将打包好的jar包发布。



随后是将打包好的项目通过scp发送到服务器。



最后将docker镜像结束并移除后，重制镜像，最后运行即可。

# 10.功能展示【叶镓瑜、杨曦】

这部分见演示视频

# 11.项目清单【叶镓瑜、杨曦】

# 12.项目总结【叶镓瑜、杨曦】

# 13.参考文献【叶镓瑜、杨曦】

无参考文献