基于智能移动端的智慧校园应用系统



易修哥

学校：山东大学

团队：猛犸团队

队长：赵衍琛

队员：崔玉峰、张翰林

邵 睿、陈怡凡

指导老师：何 萌

目录

[一、需求分析 3](#_Toc527399309)

[1.1 开发背景 3](#_Toc527399310)

[1.2使用人群 4](#_Toc527399311)

[1.3策划方案 5](#_Toc527399312)

[二、系统设计 6](#_Toc527399313)

[2.1前后端整体架构 6](#_Toc527399314)

[2.2数据库主体构建 6](#_Toc527399315)

[2.3数据库系统构架 7](#_Toc527399316)

[2.3数据库模型图 8](#_Toc527399317)

[三、主要开发流程 8](#_Toc527399318)

[四、界面及功能展示 9](#_Toc527399319)

[4.1Android（师生端）实现情况及使用说明图文介绍 9](#_Toc527399320)

[4.2Android（管理员维修员端）实现情况及使用说明图文介绍 15](#_Toc527399321)

[4.3 WEB端实现情况及使用说明图文介绍 21](#_Toc527399322)

[五、工作总结 23](#_Toc527399323)

[六：未来计划 27](#_Toc527399324)

[6.1未来的软件功能迭代计划 27](#_Toc527399325)

[6.2未来的市场推广方案及计划 27](#_Toc527399326)

[七、指导教师评语 28](#_Toc527399327)

[附1：数据库建表语句 28](#_Toc527399328)

[附2：前后端接口文档 29](#_Toc527399329)

[附3：安装部署流程 36](#_Toc527399330)

# 一、需求分析

## 1.1 开发背景

随着移动设备的快速普及，移动应用时代已经到来，4G的普及与5G的

即将推广，手机作为最便携的网络终端设备，使得高校信息化这一话题又有了新的生命。在数据中心不断增强的服务性能之外最显著的特征就是在用户移动端的精彩表现。单纯用PC的时代将一去不复返。以手机、平板电脑介质为代表的移动终端应用将为高校信息化带来巨大变革。移动应用不只是在手机上运行软件那么简单，它涉及到高校信息化应用场景的完善、扩展，让数据无所不在，通过广泛的产业联合作为用户提供低成本整体解决方案。

在高校及科研院所，一些公共设施经常需要定期维护和及时维修，不及时的检修和一些突发的事件可能会造成重大财产损害，甚至危害学生和职工的生命。如何防治和及时应对，成为大家普遍关心的热点问题。根据调查，比如在高校校园内，经常有公共设施:房屋、电力设备等的损坏，经常会出现教学用品：电脑、投影仪等的故障，这些情况大大影响了正常的教学工作活动，及时地维修和定期的维护是必要的。如果仅仅等待维修人员的定期检修和维护，一些重大突发的事故隐患可能无法及时发现。作为学校的学生和教工群体遇到这类事情，往往会发现不知如何处理，不知道向那个部门汇报的情况，传统的电话报修系统也不便于管理和维护，无法支持先进的数据分析和智能故障预测。并且，对于目前高校，如果大量增加维修保养人员无疑会增加额外的支出。

在这样的现实需求中，学生和教工普遍使用的智能手机中开发手机应用程序，构造一个智慧校园应用系统将可以很好地解决这一问题。

选择智能手机充当报修设备并不是偶然的，因为：智能手机具有一定的运算能力，基于主流智能手机系统的开发环境和应用的底层接口较为成熟；智能手机自身集成了摄像头、无线网络、各种传感器、2G\3G\4G网络协议，语音录音通话等等模块，这些都为我们的校园数据展示与分析以及报修智能软件提供了硬件上的保障；另外，智能手机目前普及率高，基本目前人手一部，只需要安装智能移动应用软件及可以实现系统，无需其他的投资。

## 1.2使用人群

学生：可以使用自己携带的智能手机，启动报修系统，选择相应的故障类型、具体所在地，并选择拍照提交给报修系统数据中心。

教师：可以使用自己携带的智能手机，启动报修系统，选择相应的故障类型、具体所在地，并可以选择拍照的形式提交给报修系统数据中心。并且可以选择是否对后勤的维修提出建议或评价。

维修工：可以查看主管派下来的维修任务，并进行及时的修理并且在手机端进行完成确认。

管理员：可以通过手机APP的管理员端查看空闲维修员及任务完成情况，可以随时指派维修员去对某故障进行及时修理，并且可以过智能移动端，查看通过后台数据中心分析得出的分析报告，内容包括统计最近的问题类型，可能的问题预测等。

**新增功能：**班级二维码点名系统，老师可以通过主页的课堂点名按钮生成二维码，学生扫码签到后可以再次点击查看名单查看签到同学名单。

## 1.3策划方案

**1.3.1 用户及市场分析**

1.3.1.1产品及价值定位

①产品定位——主要面向广大学校师生，以报修功能与课堂点名起步积累平台用户，逐渐添加社区论坛等功能，最后可以添加旧物市场、跑腿办事等功能实现平台盈利。

②价值定位——为广大师生提供一个及时解决校园问题，并且在将来成为一个能够互帮互助的的平台，包含代取快递，有偿跑腿，传授知识等。

1.3.1.2人群习惯分析

（1）行为习惯

越来越多的年轻人较早的拥有手机等电子设备，并且对手机的依赖度也越来越高，用时习惯于通过app来获取信息、反馈交流信息，并且在线语音、支付功能愈发强大。

（2）消费习惯

随着网络的快速发展，让年轻人尤其是大学生群体早已习惯于网络消费，愿意在app中投资消费。本产品目前以及将来所提供的服务皆与师生日常生活的衣食住行有关，需求量大，市场容量大，可行性强。

**1.3.2 可能的竞争对手分析**

①目前市场上现有的校园跑腿APP如：俺来也等

②各高校学生自研内部使用的APP

# 二、系统设计

## 2.1前后端整体架构

**2.1.1硬件环境**

操作系统：Windows、Linux均可

手机系统：Android4.0+版本以及目前主流屏幕分辨率均可完美适配

**2.1.2软件环境**

前端：Android手机客户端、管理端

WEB端：目前主流浏览器包括IE均可使用

服务器：腾讯云+tomcat容器

后台：Springboot+MyBaits主流后端框架，系统易扩展可移植性高

数据库：MySQL8.0+

## 2.2数据库主体构建

根据题目需求，主要是师生与主管、维修工人之间的维修任务问题，设计到权限管理、任务多状态变换，另外还有我们新增加的点名系统，所以设计了用户表、职工表、任务表、登录状态表、签到表。

还有一个图片表（image），主要存放用户头像与故障的反馈图片，为后台hibernate自动生成，无需自主配置。

## 2.3数据库系统构架

数据库：MySQL8.0+

位置位于服务器本地数据库，非额外购买数据库

数据库管理系统：Navicat For MySQL

## 2.3数据库模型图

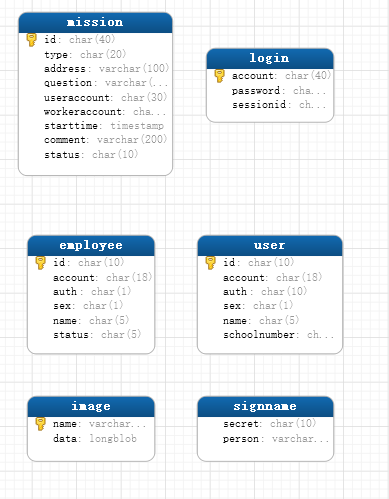


图2.1 数据库模型图

# 三、主要开发流程

8月11日-8月15日：完成需求分析、数据库设计等前期准备工作建立github项目进行统一版本管理协作开发

8月15日-25日：完成APP前端页面编写

9月1日-9月5日：完成WEB端页面编写及主要数据库设计与后台框架搭建

9月6日-9月16日：完成后端代码编写并同APP与WEB端一起联调接口

9月17日：配置服务器、测试功能

9月18日-19日：录制演示视频、撰写相关文档

# 四、界面及功能展示

## 4.1Android（师生端）实现情况及使用说明图文介绍

4.1.1登录注册引导界面：

学生用户可以自行注册和登录

老师应使用分配的账号直接登录

4.1.2 登录界面：

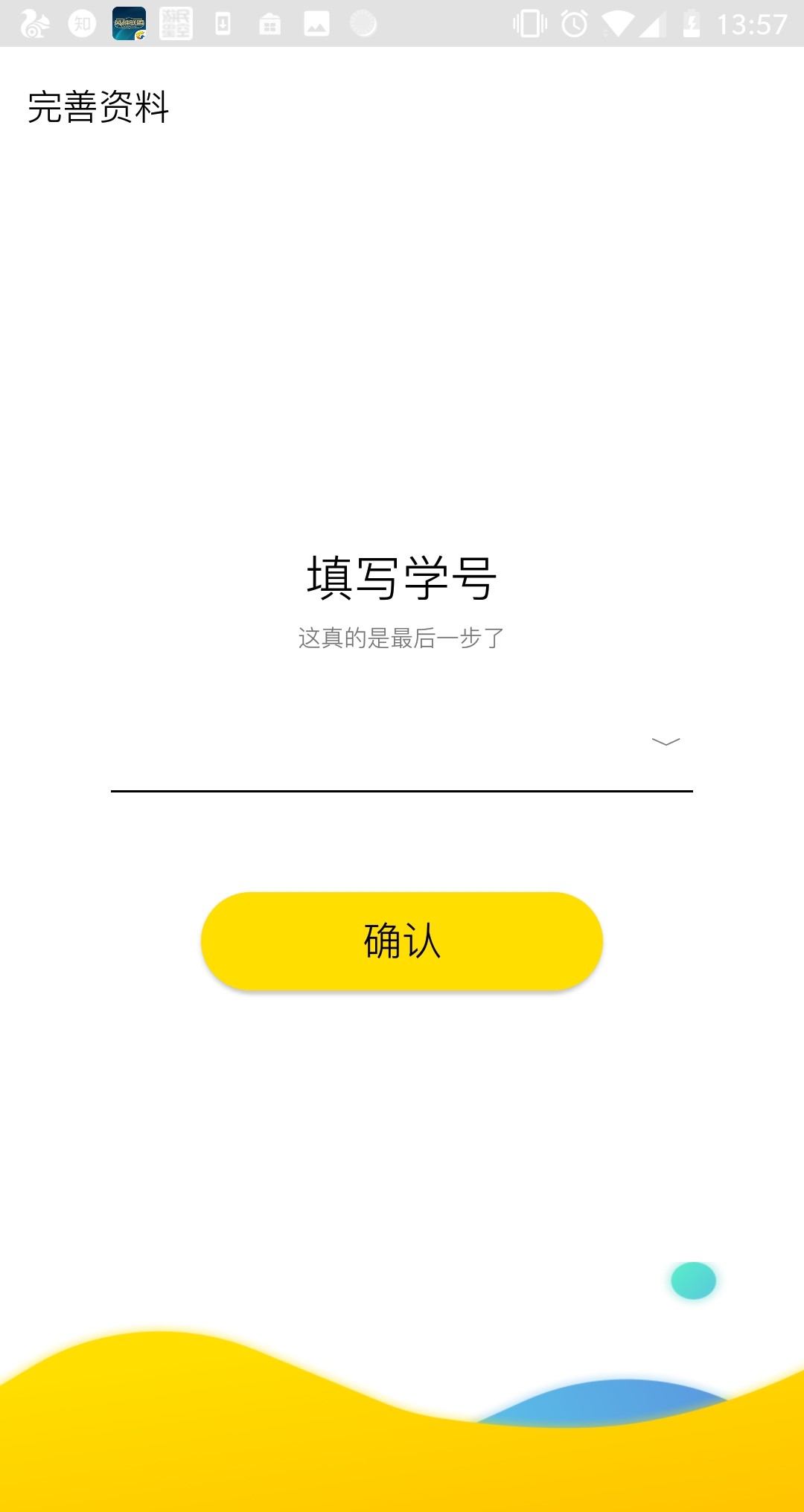
用户根据提示输入用户名和密码登录入系统。

4.1.3 注册界面：

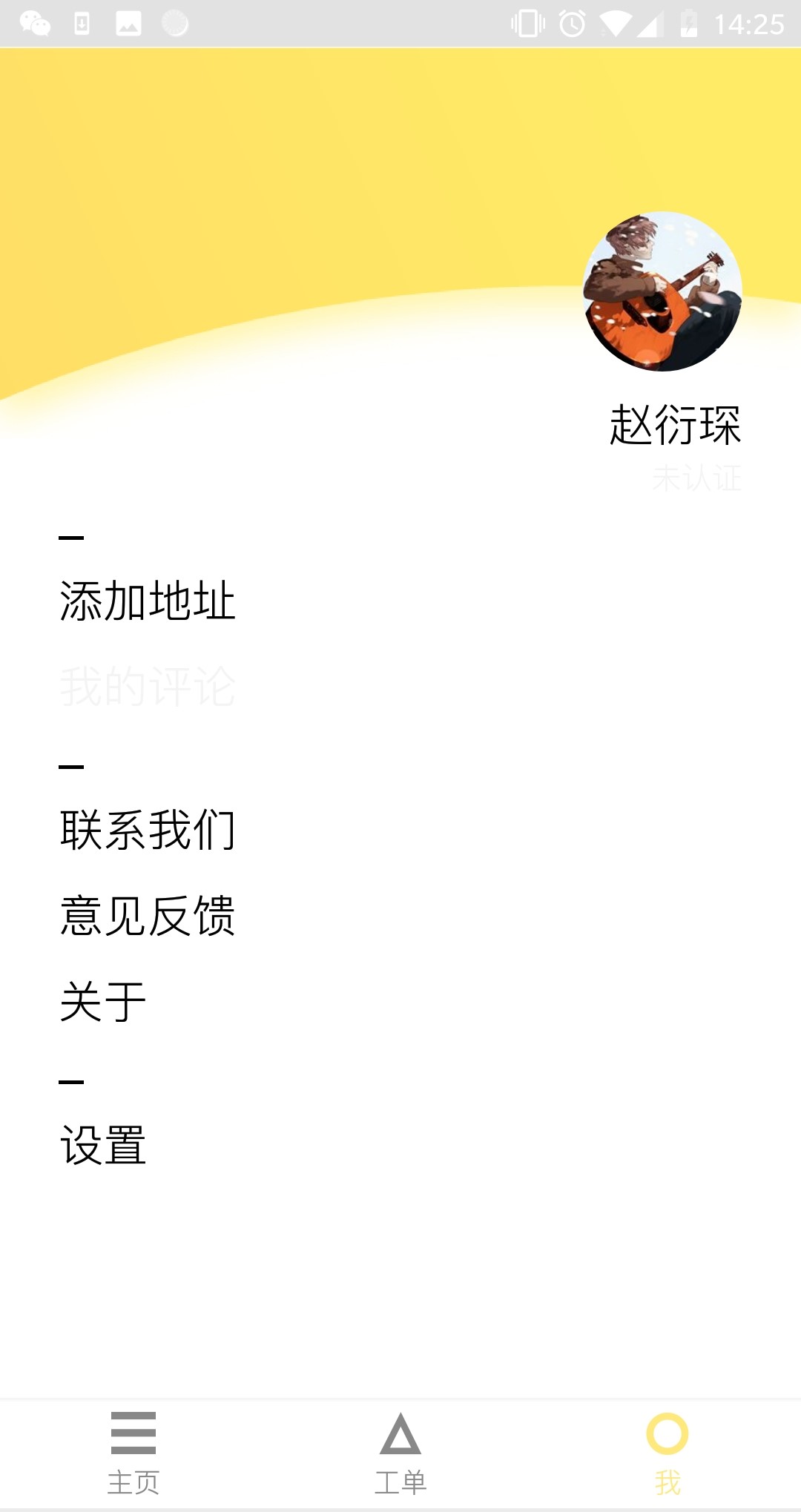
学生用户可通过系统自行完成注册，根据提示依次输入信息，账号为个人手机号，方便维修人员进行联系。

4.1.4 主界面：

App主界面预览，主要功能可完成故障报修，课堂点名等功能，并且可以对报修状态进行跟踪，实时查看故障状态，与报修人员沟通。

**** **** ****

4.1.5 故障报修界面：

填写故障地址，故障类型，并且填写故障详细情况，提交故障图片，点击提交可以将故障信息上传到后台数据中心。

**** ****

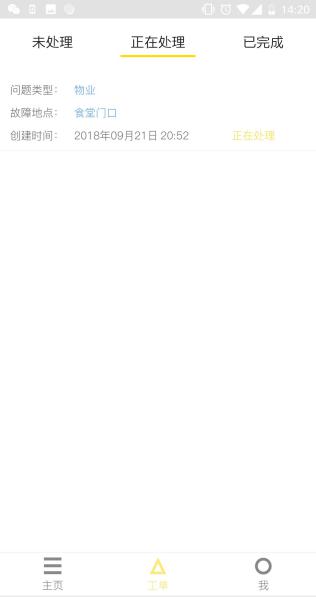
4.1.6 查看报修处理情况界面：

处理情况分为未处理，正在处理，已完成三种状态，点击进入查看故障详情。

未处理：报修系统接到报修请求，但还未有维修人员进行处理

正在处理：已分配维修人员进行处理的问题

已完成：维修提出者或维修人员确认已经解决的问题

**** **** ****

4.1.7 报修详情界面：

对于未处理的报修问题，还未分配工人，仅显示报修的信息。

****

对于正在处理的的报修问题，可以显示维修人员姓名和电话号码方便与其联系。也可以在确认故障解决后点击“确认解决”按钮，可以确认完成该次报修。

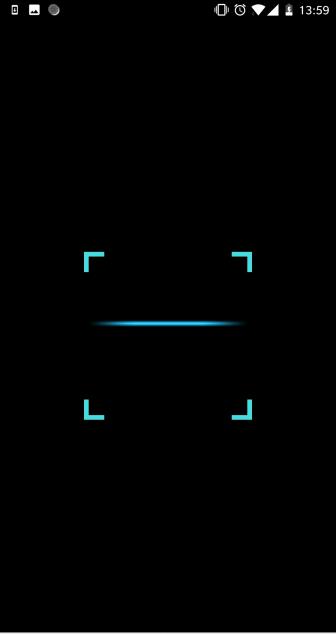
****

已成功处理的故障，用户可以填写评论，对故障处理结果进行评价。

**** ****

4.1.8 课堂点名界面：

点击获取二维码，可进行二维码的获取，其他用户扫描二维码可以被记录，点击查看人数可获得详细的名单。

**** **** ****

## 4.2Android（管理员维修员端）实现情况及使用说明图文介绍

4.2.1登录界面：

维修人员和维修主管可根据后台分配的的账号和密码进行登录，登入系统后会跟据不同身份显示不同的界面和数据。

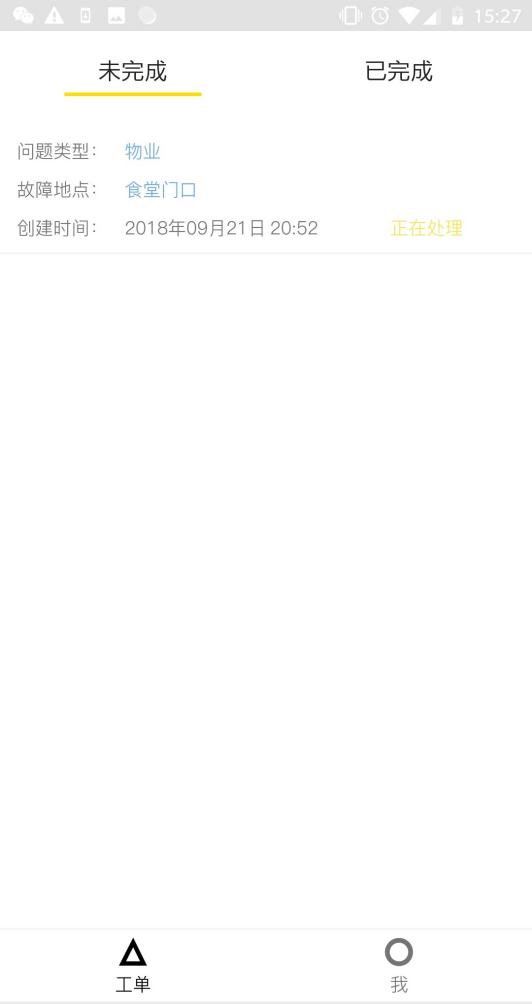
**** ****

维修人员：学校中负责处理故障事故的员工，通过WEB端后台权限管理系统进行注册，可通过系统查看其要完成的维修任务情况等信息。

维修主管：学校维修部门主管，通过后台权限管理系统进行注册，允许查看所有状态下的维修任务，并且可以为未处理的任务分配维修工人，可以查看工人状态以及近期维修情况数据图表分析统计。

4.2.2 维修人员界面：

维修人员仅可以查看，他被分配的维修任务，以及他已完成的维修任务

**** ****

对于他正在处理的报修任务可显示故障详情，以及提出人的信息，并且可以点击确认解决按钮，告知系统已解决该问题。

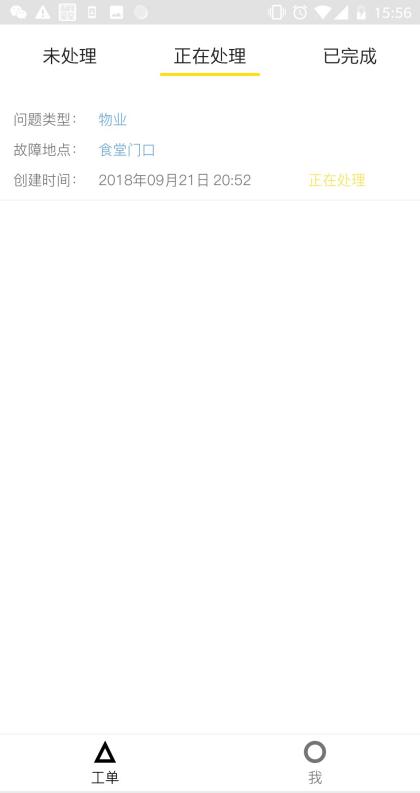
****

已完成报修任务可以显示提出人信息，问题详情，以及提出人写的评价。

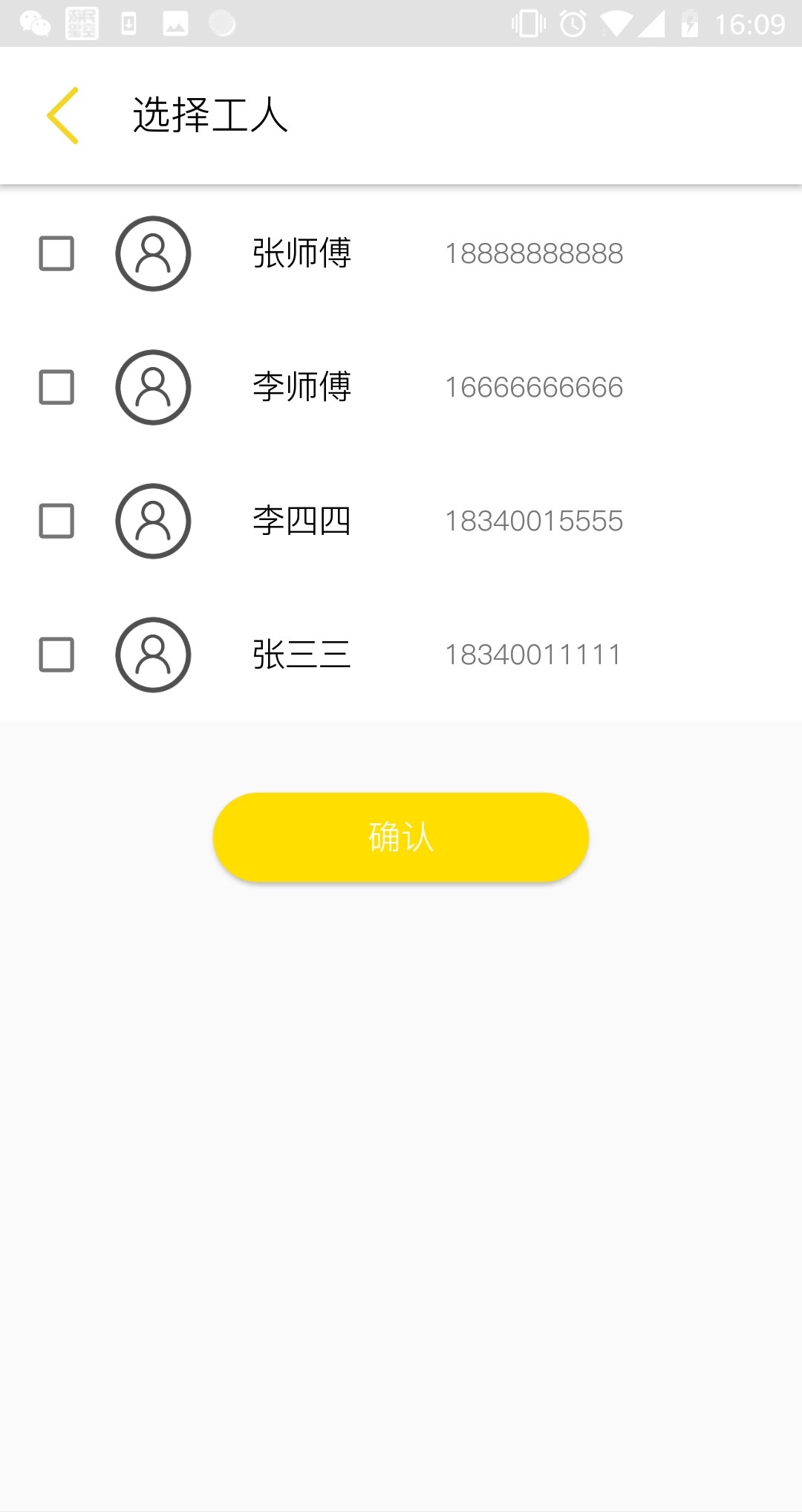
****

4.2.3 主管人员界面：

维修主管可以查看全部的报修信息，未处理，正在处理和已完成全部报修工单

**** **** ****

未处理故障，显示提出人和问题说明信息，并且可为此问题分配工人处理。

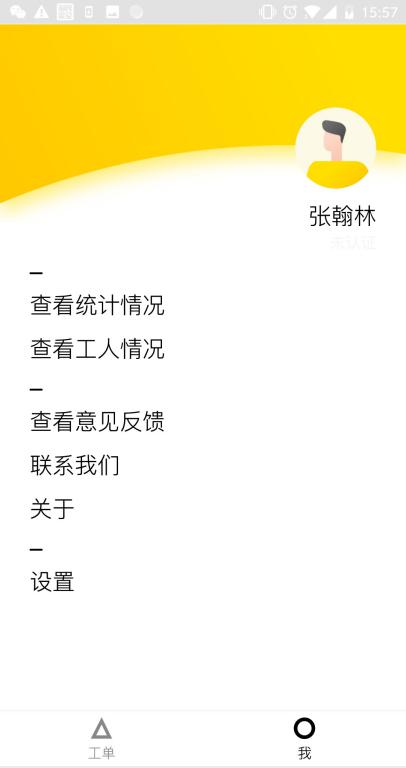
**** ****

正在处理的报修任务显示故障提出人，故障详情，故障的处理者

已完成的报修任务显示提出人、问题详情，处理者以及提出者给出的评价。

****

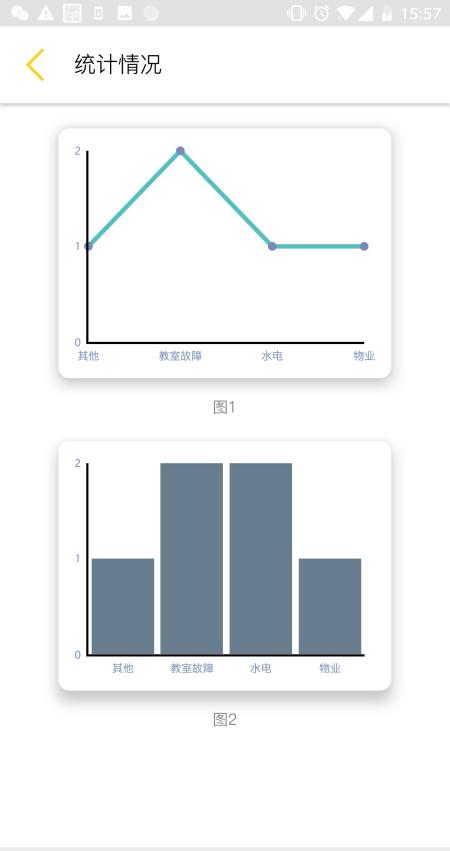
4.2.4 主管的个人界面：

****

4.2.5 统计情况界面：

可显示最近十天各类型故障发生的数量

以及全部时间内各故障发生数量的统计情况

****

## 4.3 WEB端实现情况及使用说明图文介绍

4.3.1 登录界面

权限管理系统的登录界面，使用后台管理员账号登录



4.3.2 主界面

登录后的权限管理系统，菜单第一栏为账号注册，输入主管或维修人员的工号、账号、密码、姓名、性别即可进行注册



菜单栏第二栏权限列表栏，在此表中可以知道已注册人员的工号、账号、权限、性别、姓名与当前状态



第三栏为授权系统，可以给新注册的用户进行授权，也可以对已授权的用户更改权限



# 五、工作总结

**Android及WEB前端：**

**1.总体概况：**

开发语言：Java

开发周期：一个月

开发环境：Android Studio、windows

适配手机: oppoR9 一加五T 等

Android 最低版本要求: android 5.0+

**2.主要技术架构详解：**

**2.1整体设计架构：**

整体设计架构MVC,（ Model View Controller )

Model 层负责数据操作获取包括从网络上获取数据，获取本地存储的数据

View 层负责界面加载，对于组件的绑定，并且显示数据，相应用户交互

Controller 层负责Model 层和View 层的交互实现层次之间的通信

模块化编程 根据不同的需求模块，用户管理模块，任务模块，设置模块，即时通讯IM模块，进行不同层次的实现，相互之间耦合性较低，代码可复用性高 。对于部分常用的需求，也同样编写了多种的实用工具类，如图片裁剪，数据本地化持久储存，用户数据的管理，及一些实用的View切换辅助类。

整体设计思想符合面向对象的开发思想，代码精简，观看性强。

整体结构采用单Activity+多fragment架构整体APP运行流畅，易于开发

**2.2 UI设计：**

整体的UI设计，动画效果设计，均有组内成员设计并实现。UI风格整体简洁美观，符合当代年轻人的审美需求，并且已经适配多款不同分辨率的机型。

**2.3网络传输层：OKHttp+ Retrofit2+RxJava+Gson**

底层网络框架采用OKhttp ：这是一个开源项目,是安卓端最火热的轻量级框架，用于实现底层的网络链接。对于Cookie的获取和储存，我采用了的方式实现对Cookie的管理sharepreference。

网络框架封装层 Retrofit2 :一个针对Android和Java类型安全的http客户端，他封装了很多注解，可以很方便的编写与后台交互的网络接口，其底层框架为Okhttp

线程处理框架Rxjava：Rxjava现今最火最安全的线程处理调度框架之一，用于处理异步网络请求，回调到主线程的问题，与Retrofit2 有极高的互动性。

封装：根据程序需要，我将以上四种网络传输用到的框架进行了封装，可以方便的进行网络通信，避免了反复的代码，大大提高了开发效率，并且提高了运行速度

**2.4图片加载框架：Frseco**

Fresco:是Facebook开发的一款开源的图片加载框架，用到了三级缓存（两层内存，一层磁盘）可以解决网络加载图片时内存占用过大的问题。Fresco实现简洁，并且效果很好，并且可以很好的适应滑动组件中图片的加载

**3.后端概况：**

Spring Boot：全新后台框架快速应用开发领域的领导者

Mybaits：优秀的持久层框架，它支持定制化 SQL、存储过程以及高级映射。便于SQL的统一管理和优化。将业务逻辑和数据访问逻辑分离，效率更高，更易维护。

MySQL：最流行的关系型数据库管理系统之一。

运用Springboot+Mybaits+MySQL的后台框架以及规范的接口API使得仅仅用了半个月的时间就实现了后台的快速高效开发。并且能够保证高效性及稳定性。

**感悟总结：**

通过本次齐鲁软件设计大赛大赛，培养了我们的创新精神，竞争意识，克服困难、坚持不懈的毅力以及团队合作精神。开发的这款软件，从设计到开发都经过了细致摸索和推敲和实地考察，做到了作品的原创性。这是一款独立研发且具有成品性质的软件，是我们大家共同努力的结果。开发中，大家的能力，诸如大家的合作，个人的协作能力，策划能力，以及时间观念都有一定的提高。希望软件的设计能给大家耳目一新的感觉，希望成为广受用户欢迎的软件。一个成功的设计应该是以用户为出发点，始终在考虑“用户需要什么”，而不是自娱自乐用的。一味从自我考虑，只做符合自己的软件，你会发现它的需求是如此的不足，功能有很大的缺失，最后会发现做出来的软件连你自己的愿望都没能真正实现。软件一定要有自己的亮点，不要落入平庸。设计上一定要有重点，突出自己的特色和主要的功能。细节决定一切。

# 六：未来计划

## 6.1未来的软件功能迭代计划

待与学校沟通推广，使用人数稳定后，可同我们已经开发成型的“小学弟”APP合并使用，加入小学弟中的校园跑腿模块，用户可以发布任务，其他用户可以接取任务赚取赏金，该功能可涉及帮拿快递，帮买东西，帮做PPT，二手置换等等方面，扩展成为一个供需撮合平台，每单中初步进行抽成逐步实现平台运营维护费用的收支平衡。

图6.1 二次迭代初步雏形设计图

## 6.2未来的市场推广方案及计划

产品目标——三年内不断完善APP功能，使产品不断趋于成熟。

市场目标——在三年内，尽可能提升市场份额。

研发目标——在提供优质服务的同时，也要不断完善APP的内容，走在市场最前沿，提供更好的服务。

财务目标——在一到两年内收回成本，在五年内实现大额的盈利。

1、网络推广

建立供需群、公众号、垂直需求群并发展成员容量，培养APP前期用户习惯，到时机成熟直接引流入APP。

2、线下推广

先在本校进行推广，有一定使用用户之后再在各大学校内通过张贴海报，举办小型的线下活动等，利用少量的资金，来达到宣传推广的作用。

3、广告推广

通过和一些媒体，或是大学生创业或是大学生文化，建立合作关系，帮助推广APP。

4、分享有奖

在APP里设立分享有奖的专页，用户通过分享可以获得一定的积分奖励，实现公司宣传与用户受益的双赢关系。

# 七、指导教师评语

猛犸团队的五位同学在参加本次比赛的过程中团结一心，齐心协力，分工合理，经常能够看到他们在学校实验室内一起讨论问题解决问题的身影，不会不懂的问题他们自己都会主动上网搜索解决，自立自主、努力上进，只有当遇到自己实在解决不了的问题时才会寻求老师的帮助，十分令老师欣慰。

软件开发过程中，五位同学合理运用所学知识，即学即用，能够将数据库系统概念、软件工程设计测试的开发流程、面向对象的思想规范的，完备的运用到比赛程序的开发之中，能够做到对学校知识活学活用，甚至能够自学Android、后台Spring、网页WEB端开发技术。没有成为为学习而学习的死读书的书呆子，真正能够把概念、技术同实际、实践相结合，开发出来这么一款实用、高效、界面友好的智慧校园APP。十分令老师满意。

2018年10月14日

指导教师：何 萌

# 附1：数据库建表语句

CREATE TABLE user (

id CHAR(10) NOT NULL PRIMARY KEY,

account CHAR(18) NOT NULL,

auth CHAR(10) NOT NULL,

sex CHAR(1) NOT NULL,

name CHAR(5) NOT NULL,

schoolnumber CHAR(15) NOT NULL

);

CREATE TABLE employee(

id CHAR(10) NOT NULL PRIMARY KEY,

account CHAR(18) NOT NULL,

auth CHAR(18) NOT NULL,

sex CHAR(1) NOT NULL,

name CHAR(5) NOT NULL,

status CHAR(5) NOT NULL

);

CREATE TABLE mission(

id CHAR(40) PRIMARY KEY,

type CHAR(20),

address VARCHAR(100),

question VARCHAR(255),

useraccount CHAR(30),

workeraccount CHAR(30),

starttime TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

comment VARCHAR(200),

status CHAR(10)

);

CREATE TABLE login(

account CHAR(40) PRIMARY KEY,

password CHAR(40),

sessionid CHAR(40)

);

CREATE TABLE signname(

secret CHAR(10),

person VARCHAR(255)

);

# 附2：前后端接口文档

**WEB页面端**

**分配账号即注册（POST）**

api/account/adminregis

String id; //工号

String account; //手机号

String password;

String sex;

String name;

返回值：字符串："admin regis success"

**获取全部职工列表（GET）**

api/account/admingetallworkers

返回值：

List<AllWorkersOutput>

String id; //工号

String account; //手机号

String auth; //1为职工 2为经理主管 3为你自己超级用户不用显示 “”空为新账号没有权限

String sex; //1为男 0为女

String name;

String status; //0为空闲 1为工作中 “”空为新账号

**权限修改（POST）**

api/account/adminauthconfrim

String id;

String auth;

返回值："success"

----------------------------------------------------------------------

**APP端**

**点名：**

**获取secret（GET）**

api/statistic/getsigninsecret

{

"code": 0,

"codeInfo": "OK",

"data": "y4UDY7uH"

}

**扫码记录（POST）**

api/statistic/signin

String secret;

{

"code": 0,

"codeInfo": "OK",

"data": "cyf"

}

**统计名单（POST）**

api/statistic/getSigninNames

String secret;

返回值：

List<String>

**获取统计数据（GET）**

http://localhost:8080/api/statistic/getData

返回值：

{

"code": 0,

"codeInfo": "OK",

"data": {

"values1": [

{

"type": "其他",

"count": "1"

},

{

"type": "教室故障",

"count": "3"

},

{

"type": "水电",

"count": "1"

},

{

"type": "物业",

"count": "2"

}

],

"values2": [

{

"type": "其他",

"count": "1"

},

{

"type": "教室故障",

"count": "3"

},

{

"type": "水电",

"count": "2"

},

{

"type": "物业",

"count": "2"

}

]

}

}

**获取工人列表（GET）**

http://localhost:8080/api/mission/getWorkers

返回值：

String account;

String name;

**指派工人（POST）**

http://localhost:8080/api/mission/assign

String questionId;

String workerid;

{

"code": 0,

"codeInfo": "OK",

"data": "cyf"

}

**任务确认完成（POST）**

http://localhost:8080/api/mission/missionComplete

String questionId;

{

"code": 0,

"codeInfo": "OK",

"data": "cyf"

}

**提交评价（POST）**

http://localhost:8080/api/mission/setcomment

String id;

String comment;

{

"code": 0,

"codeInfo": "OK",

"data": "cyf"

}

**获取任务详情（POST）**

http://localhost:8080/api/mission/getMyMissionDescription

String questionId;

{

"code": 0,

"codeInfo": "OK",

"data": {

"QuestionPerson": {

"name": "赵衍琛",

"phoneNumber": "18340018831"

},

"SolvePerson": {

"name": null,

"phoneNumber": null

},

"QuestionType": "教室故障",

"QuestionAddress": "当饭吃穿多大的",

"QuestionText": "地方呀任覅咕咕咕换个词广发银行重复的复古风地方呀观后感好幸福",

"Comment": null,

"comment": null,

"questionType": "教室故障",

"questionText": "地方呀任覅咕咕咕换个词广发银行重复的复古风地方呀观后感好幸福",

"solvePerson": {

"name": null,

"phoneNumber": null

},

"questionAddress": "当饭吃穿多大的",

"questionPerson": {

"name": "赵衍琛",

"phoneNumber": "18340018831"

}

}

}

**获取任务列表（POST）**

http://localhost:8080/api/mission/getMyMissions

String status; (待处理，处理中，已处理)

{

"code": 0,

"codeInfo": "OK",

"data": [

{

"id": "9wGswNnK",

"type": "教室故障",

"address": "山东大学",

"starttime": "2018-09-21",

"status": "待处理"

}

]

}

**发布任务(POST)**

http://localhost:8080/api/mission/upload

String type;

String address;

String question;

MultipartFile file;

{

"code": 0,

"codeInfo": "OK",

"data": "cyf"

}

注：任务图片名为 18340018831mission.jpg 即账号加mission.jpg 接口可以用下面的获取用户头像

**获取用户头像（GET）**

http://localhost:8080/api/image/getuserheadphoto?name=18945678923head.jpg

byte[] data

**设置用户头像**

http://localhost:8080/api/image/setuserheadphoto

MultipartFile file

{

"code": 0,

"codeInfo": "OK",

"data": "cyf"

}

注：头像名为 18340018831head.jpg 即账号加head.jpg

**注册（仅为学生注册）**

http://localhost:8080/api/account/register

String account;

String password;

String sex;

String name;

String schoolnumber;

{

"code": 0,

"codeInfo": "OK",

"data": {

"id": "YghbcEpw",

“account”:”18340018831”

"name": "赵衍琛",

"sex": "1",

"schoolnumber": "201600301315"

“auth”:”0” (0是学生 1是工人 2是管理)

}

}

**登录**

http://localhost:8080/api/account/login

String account;

String password;

注：

① 普通学生用户的Bean

{

"code": 0,

"codeInfo": "OK",

"data": {

"id": "YghbcEpw",

“account”:”18340018831”

"name": "赵衍琛",

"sex": "1",

"schoolnumber": "201600301315"

“auth”:”0” (0是学生 1是工人 2是管理)

}

}

② 职工用户的bean

String id;

String account; （account是电话号码 id是后台自己生成的）

String auth; (0是学生 1是工人 2是管理)

String sex;

String name;

错误信息：

{

"code": 100,

"codeInfo": "AccountNotExists",

"data": ""

}

{

"code": 102,

"codeInfo": "PasswordError",

"data": ""

}

# 附3：安装部署流程

1、手机APP端直接安装用户端与管理端APP即可



2、后台服务器部署：

1. 首先购买服务器之后运行以下init.sh 文件

#!/usr/bin/env bash

mysql\_passwd=v9jOBbSSWNQzMZEY

apt update

apt upgrade -y

apt dist-upgrade -y

apt install -y apt-transport-https

wget https://get.docker.com -O get.sh

chmod +x get.sh

./get.sh --mirror Aliyun

mkdir -p /etc/docker

tee /etc/docker/daemon.json <<-'EOF'

{

"registry-mirrors": ["https://dpzpmu1h.mirror.aliyuncs.com"]

}

EOF

systemctl daemon-reload

systemctl restart docker

tee /home/data/nginx/conf.d/nginx.conf <<-'EOF'

server\_tokens off;

server{

listen 80 default\_server;

keepalive\_timeout 60;

location / {

proxy\_pass http://tomcat:8080/;

proxy\_redirect off;

proxy\_set\_header Host $host:$server\_port;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

}

}

EOF

docker run -dp 3306:3306 --restart=always -e MYSQL\_ROOT\_PASSWORD=$mysql\_passwd -v /home/data/mysql:/var/lib/mysql --name mysql mysql

docker run -dp 8080:8080 --restart=always --link mysql:mysql -v /home/data/tomcat/webapps:/usr/local/tomcat/webapps --name tomcat tomcat:alpine

docker run -dp 80:80 -p 443:443 --restart=always --link tomcat:tomcat -v /home/data/nginx/conf.d:/etc/nginx/conf.d -v /home/data/nginx/home:/home --name nginx nginx:alpine

1. 通过电脑上Navicat for mysql连接云端服务器数据库

账号 root 密码 v9jOBbSSWNQzMZEY

然后以下运行SQL语句

CREATE TABLE user (

id CHAR(10) NOT NULL PRIMARY KEY,

account CHAR(18) NOT NULL,

auth CHAR(10) NOT NULL,

sex CHAR(1) NOT NULL,

name CHAR(5) NOT NULL,

schoolnumber CHAR(15) NOT NULL

);

CREATE TABLE employee(

id CHAR(10) NOT NULL PRIMARY KEY,

account CHAR(18) NOT NULL,

auth CHAR(18) NOT NULL,

sex CHAR(1) NOT NULL,

name CHAR(5) NOT NULL,

status CHAR(5) NOT NULL

);

CREATE TABLE mission(

id CHAR(40) PRIMARY KEY,

type CHAR(20),

address VARCHAR(100),

question VARCHAR(255),

useraccount CHAR(30),

workeraccount CHAR(30),

starttime TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

comment VARCHAR(200),

status CHAR(10)

);

CREATE TABLE login(

account CHAR(40) PRIMARY KEY,

password CHAR(40),

sessionid CHAR(40)

);

CREATE TABLE signname(

secret CHAR(10),

person VARCHAR(255)

);

1. 将后台SpringBoot打包成War包 放入服务器/home/data/tomcat/webapps之中稍等
2. 可以通过浏览器访问<http://XXXXXXX:8080/admin> 获取后台管理页面，手机APP可直接使用