

郑州大学毕业设计（论文）

题 目： 重载铁路智能安全系统
 移动客户端的设计与实现

指导教师： 李晓宇 职称： 副教授

指导教师(校外)： 黄鹏 职称：

学生姓名： 王博文 学号： 20142480131

专 业： 软件工程

院（系）： 信息工程学院

完成时间： 2018 年 6 月 5 日

2018 年 6 月 5 日

重载铁路智能安全系统移动客户端的设计与实现

摘要 铁路行车安全是铁路行业的一个重要关注点，在行车安全中需要考虑包括自然因素与人为因素等多种信息的影响。我国的重载铁路虽然发展较晚，但发展迅速，高速发展的背后离不开稳定的基础，将信息化管控与风险管理机制相结合，能够加强安全风险抵御能力，从而提高重载铁路安全管理水平。通过移动设备实现信息化管控是一项必然的趋势，我们根据近些年来获得的数据，整合已有的网页端功能，将部分必要的功能移植到移动客户端中，并加入如天气、定位等方便客户端使用的功能，让用户能够随时随地根据当前情况及时记录遇到的突发状况或处理推送的各种问题。本重载铁路安全系统基于 Android 操作系统，采用 Retrofit + RxJava + OkHttp 网络架构与 MVP 代码结构，调用微服务服务端进行数据传输与展示，数据库使用 SQLite，数据传输内容使用专业插件进行加密与解密，实现了整个系统的高安全性与高用户体验度的优点。整个系统共实现了对危险源、隐患、联控信息、监察通知、灾害预警、监控日报六大模块的监管，涵盖了重载铁路运行过程中可能遇到的绝大部分情况，使用户在处理问题时做到高效、准确。

关键字 重载铁路，Android，安全

Abstract The safety of the heavy-haul railways is an important concern of the railway industry, and the impact of various information including natural factors and human factors needs to be taken into consideration in traffic safety. Although the development of China's heavy-haul railways is relatively late, it is developing rapidly. The stable foundation behind the high-speed development cannot be separated. Combining information management and risk management mechanisms can strengthen the security risk resilience and thus improve the safety management of heavy-haul railways. Information management and control through mobile devices is an inevitable trend. Based on the data obtained in recent years, we have integrated existing web-side functions and ported some necessary functions to mobile clients, adding weather, positioning, etc. The function that is convenient for the client, allows the user to record the sudden situation encountered or handle various problems in time according to the current situation at any time. This heavy-haul railways safety system is based on the Android operating system. It uses the Retrofit + RxJava + OkHttp network architecture and MVP code structure to call the micro-service server for data transmission and display. The database uses SQLite, and the data transmission content is encrypted and decrypted using professional plug-ins. The advantages of high security and user experience are achieved in the entire system. The entire system has realized the supervision of the six major modules of hazard sources, hidden dangers, joint control information, monitoring notices, disaster warning, and monitoring daily reports, covering most of the situations that may be encountered during the operation of heavy-haul railways, enabling users to handle The problem is efficient and accurate.

Key words the heavy-haul railways, Android, security

目录

1	引言.....	1
1.1	研究的背景和意义.....	1
1.2	国内外研究现状及趋势.....	1
1.3	本文的主要内容及安排.....	2
2	相关技术基础.....	4
2.1	ANDROID 语言	4
2.2	网络框架.....	4
2.2.1	Retrofit.....	5
2.2.2	RxJava.....	6
2.2.3	OkHttp.....	6
2.3	MVP 代码架构	7
2.4	SQLITE 数据库	10
2.5	百度定位 API	11
2.6	极光推送 API	11
2.7	MPANDROIDCHART.....	11
3	需求分析.....	12
3.1	功能需求.....	12
3.2	性能需求.....	14
3.3	技术可行性分析.....	14
4	系统设计.....	15
4.1	设计原则.....	15
4.2	编程环境.....	15
4.3	功能设计.....	16
4.3.1	基础功能模块.....	18
4.3.2	录入功能模块.....	18
4.3.3	展示功能模块.....	21
4.4	界面设计.....	23
4.5	安全设计.....	23
5	系统实现.....	26

5.1	模块功能实现.....	26
5.1.1	版本验证功能.....	26
5.1.2	登陆功能.....	28
5.1.3	记录功能.....	29
5.1.4	整改功能.....	30
5.1.5	验收功能.....	31
5.1.6	档案功能.....	33
5.1.7	统计分析功能.....	34
5.1.8	日报查看功能.....	37
5.2	插件功能实现.....	38
6	系统性能评析.....	40
6.1	系统性能优化.....	40
6.2	性能评析.....	40
7	总结与展望.....	42
7.1	总结.....	42
7.2	展望.....	42
	参考文献.....	43
	致谢.....	44

1 引言

1.1 研究的背景和意义

信息化是提高铁路安全管理的重要手段之一，重载铁路本质安全管理信息系统于 2012 年建设，通过信息化手段，实现了对危险源信息处置、专业设备监测报警信息处置、安全监督检查信息处置、综合信息分析处置的整体管理。系统在保障铁路运输安全方面发挥了显著作用。

随着系统业务的发展，安全管理方面的变化和要求进一步的提高，为保障未来的持续发展，深入推进本质安全体系建设，需要通过移动信息化手段进一步提高管理效率，即通过智能手机、平板电脑等移动设备，实时接收消息、实时查询信息和移动处理业务，实现移动办公、动态管理、远程监控、快速反应，同时解决现场突发事件的报告及解决能力，保障铁路运输安全。重载铁路智能安全系统移动客户端（以下简称重载铁路安全系统）的建设将使重载铁路安全管理工作向更加智能化、科学化发展，使重载铁路安全管理再上一个新的台阶。

1.2 国内外研究现状及趋势

铁路行车安全是复杂、庞大的系统工程，需要综合考虑线路、桥梁、隧道、牵引供电、信号制式、车辆动力学特性、列车运行和运输组织等因素，才能做出全面、系统的科学决策。随着我国铁路运输技术复杂度不断增加，既有的安全管理模式面临严峻挑战。为此，有必要在铁路运营中引入风险管理机制的同时，借鉴国内外先进安全管理理念和方法，参考国际铁路相关标准，总结和发展我国铁路安全管理成功经验，对影响运输安全的各方面进行系统整合，运用信息化手段提升铁路安全风险抵御能力，从而全面提升铁路安全综合管理水平。

国内以神华集团及朔黄铁路公司的研究为主导^[1]，朔黄铁路公司的铁路本质安全管理信息系统以本质安全为目标，以风险预控为中心，以检查检测为基础，利用拓扑系统结构实现构建全员参与，全过程管控的安全风险控制平台。

对比发达国家在行车安全方面有很多值得我国借鉴的经验，其中建立完善的安全监控体系是发达国家安全管理的重要内容。而科技的进步让实时化信息处理变得更加方便。

发达国家都设立了完善的安全管理监督部门。这些监管部门在强调保障公众

利益、强化激励与约束机制、不断调整健全法律和法规、建立对监管行为的监督制衡机制、推动安全保障技术的研发方面发挥了极其重要的作用。

发达国家的政府制定了比较完善的法律、法规来对事故进行预防，确保铁路安全。比如德国关于铁路的法律、法规比较齐全、严谨，主要有《铁路法》《事故处罚规则》《危险品运输管理规定》等，日本政府也有《铁路事业法》《铁路营业法》等。完善的法律、法规是铁路安全及健康发展的有力保证。

按照铁路跨越式发展战略的要求，我国铁路将进一步研制开发和引进先进的安全技术装备，逐步建立防止列车冲突安全监控防护系统、提速旅客列车安全检测保障系统、灾害预警防护系统、货物装载加固和危险品运输安全监测系统、事故救援快速抢险系统和行车安全信息网络及综合管理系统，使我国铁路的安全技术装备达到发达国家先进水平，实现铁路运输安全有序可控^[2]。

1.3 本文的主要内容及安排

本文的主要研究内容主要围绕基于微服务设计模式的重载铁路安全系统展开，包括对其的研究背景及意义、国内外研究、设计思路、实现过程、性能优化等展开，详细介绍整个重载铁路安全系统。本论文主要内容安排如图 1.1 所示。

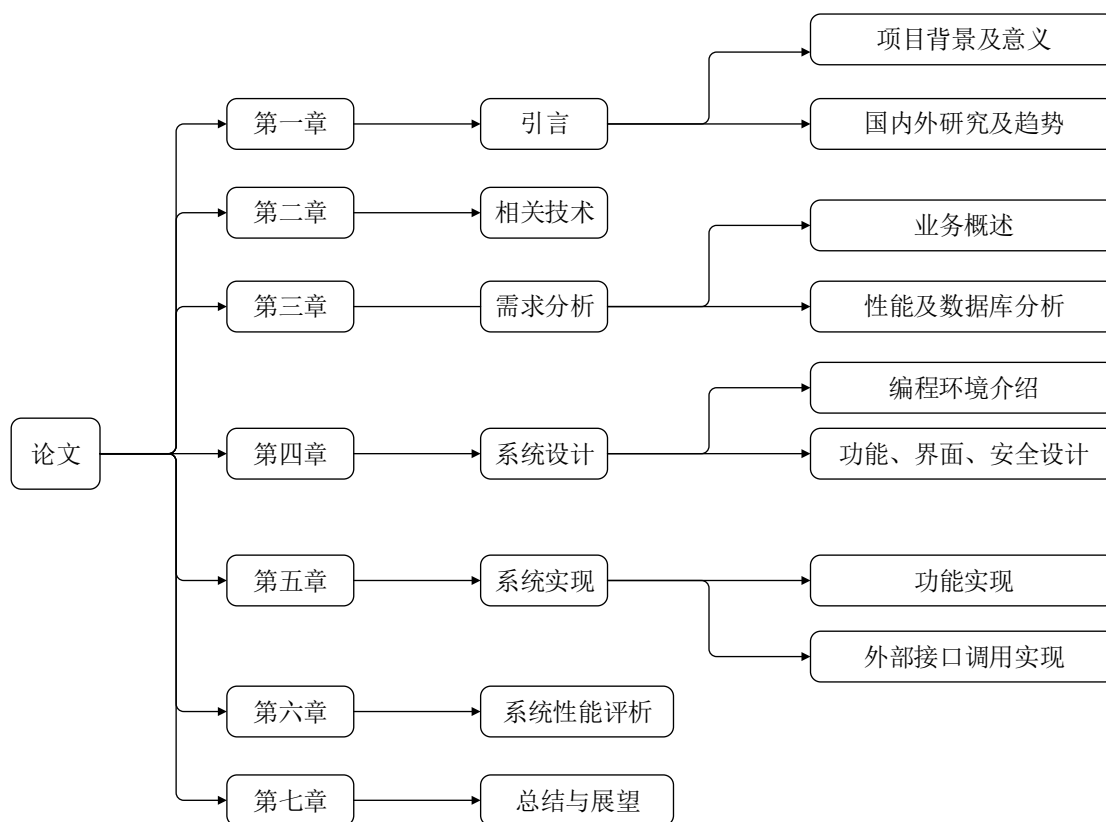


图 1.1 论文主要内容安排

第一章，引言。介绍重载铁路安全系统的研究背景及意义，简单概括重载铁路相关安全系统的国内外研究情况。

第二章，相关技术基础。对本系统中所用到的主要编程语言、开发技术进行介绍与分析。

第三章，需求分析。对重载铁路安全系统的需求进行系统化分析，确定系统中主要功能以及使用的关键技术，分析系统的性能需求以及数据库需求。

第四章，系统设计。对重载铁路安全系统的整体设计进行阐述。首先介绍开发本系统所使用的编程环境 **Android Studio**。然后介绍系统的总体设计，包括每个功能模块的界面设计和功能设计

第五章，系统的实现。主要阐述重载铁路安全系统移动端实现的功能，以及使用到的一些外部接口。

第六章，系统性能评析。对系统的性能提出优化方案，并对优化的效果进行评析。

第七章，总结与展望，对所做的工作进行总结与展望。

2 相关技术基础

2.1 Android 语言

Android 由 JAVA 语言发展而来^[3]，是一种基于 Linux 的自由及开放源代码的操作系统，主要使用于移动设备，如智能手机和平板电脑，由 Google 公司和开放手机联盟领导及开发。Android 的系统架构和其操作系统一样，采用了分层的架构^[4]。

Android 分为四个层，从高层到低层分别是应用程序层、应用程序框架层、系统运行库（C/C++库以及 Android 运行库）层和 Linux 内核层^[5]。

应用程序层：应用是用 java 语言编写的运行在虚拟机上的程序，比如 Email 客户端，SMS 短消息程序，日历等。

应用程序框架层：这一层是编写 Google 发布的核心应用时所使用的 API 框架，开发人员同样可以使用这些框架来开发自己的应用，这样便简化了程序开发的结构设计，但是必须要遵守其框架的开发原则。

系统运行库层：当使用 Android 应用框架时，Android 系统会通过一些 C/C++ 库来支持我们使用的各个组件，使其更好的为我们服务，比如其中的 SQLite（关系数据库），Webkit（Web 浏览器引擎）。

Linux 内核层^[6]：Android 的核心系统服务给予 Linux2.6 内核，如安全性、内存管理、进程管理、网络协议栈和驱动模型等都依赖于该内核，比如 Binder IPC(Internet Process Connection 进程间通信)驱动，android 的一个特殊驱动程序，具有单独的设备节点，提供进程间通信的功能。

2.2 网络框架

系统最初采用原生 HTTP 工具类进行数据请求，但由于考虑到图片加载等异步大规模数据传输以及系统模块划分精细，最终采用 Retrofit + RxJava + OkHttp 网络框架^[7]。

OkHttp 是 Square 公司开发的一套网络请求库，已经得到 Google 官方的认可。Retrofit 是 Square 公司开发的一款针对 Android 网络请求的框架，底层基于 OkHttp 实现。RxJava 是一个在 Java 虚拟机^[8]上使用可观测的序列来组成异步的、基于事件的程序的库，简单的说就是让异步变得更加简洁方便。

2.2.1 Retrofit

(1) 引入

```
compile 'com.squareup.retrofit2:retrofit:2.1.0' //retrofit
compile 'com.google.code.gson:gson:2.6.2' //Gson 库
```

(2) 创建 Retrofit 实例

```
public static final String BASE_URL = "https://www.example.com/";
Retrofit retrofit = new Retrofit.Builder()
    .baseUrl(BASE_URL)
    .addConverterFactory(GsonConverterFactory.create())
    .build();
```

(3) 创建接口

```
public interface DataService {
    @GET("data")
    Call<DataSubject> getData(@Query("id")String id);
}
```

(4) 创建接口实例，如图 2.1 所示

```
//获取接口实例
DataService dataService = retrofit.create(DataService.class);
//调用方法得到一个Call
Call<DataSubject> call = dataService.getData("id");
//进行网络请求
call.enqueue(new Callback<DataService>() {
    @Override
    public void onResponse(Call<DataSubject> call, Response<DataSubject> response) {
        mDataAdapter.setDatas(response.body().subjects);
        mDataAdapter.notifyDataSetChanged();
    }
    @Override
    public void onFailure(Call<DataSubject> call, Throwable t) {
        t.printStackTrace();
    }
});
```

图 2.1 Retrofit 使用方式

(5) 请求

```
Response<DataSubject> response = call.execute();
```

2.2.2 RxJava

在 2.2.1 节中所提供的架构上，加入 RxJava 的使用。如图 2.2 所示。

```
Subscription subscription = dataService.getData("id")
    .subscribeOn(Schedulers.io())
    .observeOn(AndroidSchedulers.mainThread())
    .subscribe(new Subscriber<DataSubject>() {

        @Override
        public void onCompleted() {

        }

        @Override
        public void onError(Throwable e) {

        }

        @Override
        public void onNext(MovieSubject movieSubject) {
            mDataAdapter.setDatas(movieSubject.subjects);
            mDataAdapter.notifyDataSetChanged();
        }
    });
```

图 2.2 RxJava 使用方式

2.2.3 OkHttp

在 2.2.2 节中所提供的架构上，加入 OkHttp 配置，设置链接超时时间、缓存、拦截器等。如图 2.3 所示。

```
// 创建 OkHttpClient
OkHttpClient.Builder builder = new OkHttpClient.Builder();
builder.connectTimeout(DEFAULT_TIME_OUT, TimeUnit.SECONDS);//连接超时时间
builder.writeTimeout(DEFAULT_TIME_OUT, TimeUnit.SECONDS);//写操作超时时间
builder.readTimeout(DEFAULT_TIME_OUT, TimeUnit.SECONDS);//读操作超时时间

// 添加公共参数拦截器
BasicParamsInterceptor basicParamsInterceptor = new BasicParamsInterceptor.Builder()
    .addHeaderParam("userName", "")//添加公共参数
    .addHeaderParam("device", "")
    .build();

builder.addInterceptor(basicParamsInterceptor);

// 创建 Retrofit
mRetrofit = new Retrofit.Builder()
    .client(builder.build())
    .addCallAdapterFactory(RxJavaCallAdapterFactory.create())
    .addConverterFactory(GsonConverterFactory.create())
    .baseUrl(ApiConfig.BASE_URL)
    .build();
```

图 2.3 OkHttpClient 使用方式

2.3 MVP 代码架构

本系统采用 MVP 框架，MVP 框架是 MVC 框架的衍生版本，在 Android 中广泛使用^[9]，同样分为三层：Model 层、View 层、Presenter 层。其中 Model 层负责获取数据，数据的来源可以是网络或本地数据库等；View 层负责界面数据的展示，与用户进行交互；Presenter 层是 Model 层与 View 层之间通信的桥梁，将 Model 层与 View 层分离开来^[10]。MVC 结构框架与 MVP 结构框架的主要区别如图 2.4 所示。

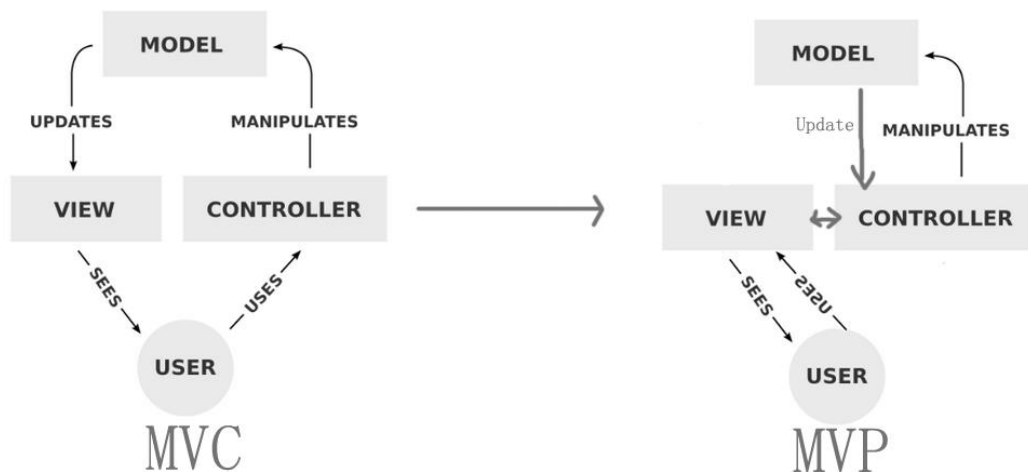


图 2.4 MVC 与 MVP 区别

在 Android 传统开发中，使用的依然是 MVC 结构，但在这其中存在许多问题，代表 View 的 layout 资源不能完全解决如何渲染的问题，还需要 Activity 的帮助，所以在 Activity 中也有 View 的内容，但是同时，Activity 主要代表了生命周期，拥有生命周期回调，并且常常设置所关联的 View 的监听，进而接收来自用户的输入，所以在一个完整的 Android 项目中 Activity 必然包含 Control 的功能。因此，使用 MVC 并不能完全将 View 与 Control 分离开。这种模式虽然在 web 端开发中有着高效使用，但由于 Android 开发与 web 开发的区别，较大的项目为了降低耦合性，更应该选择 MVP 结构。

与 MVC 结构不同的是，MVP 中的 View 在接收用户输入的同时，能够解决如何渲染的问题，进而减少 Activity 的压力。View 和 Model 的直接关联完全消失，不再直接操纵 Model，而是通过 Presenter 在 Model 中操作数据^[11]。

因此，最终本系统选择了 MVP 框架，以期望更好的实现“高内聚，低耦合”理念。本小节中展示的代码为登陆功能。

Contract 层的作用是将关联的 Model、View、Presenter 关联在一起。如 2.5 所示。

```

public interface LoginContract {

    interface Model extends BaseModel {
        //密码登录接口
        Observable<InformationBean> getInformation(String token);
    }

    interface View extends BaseView {
        //调用密码登录返回结果
        void returnInformation(InformationBean loginDoctorBean);
    }

    abstract static class Presenter extends BasePresenter<View, Model> {
        //发起密码请求
        public abstract void getInformation(String token);
    }
}

```

图 2.5 Contract 使用方式

Model 层在本系统中的使用方式如图 2.6 所示。

```

public class LoginModel implements LoginContract.Model {

    //获取 个人信息
    @Override
    public Observable<InformationBean> getInformation(String token) {
        return Api.getDefault()
            .getInformation(token)
            .map(new Func1<InformationBean, InformationBean>() {
                @Override
                public InformationBean call(InformationBean informationBean) {
                    return informationBean;
                }
            })
            .compose(RxSchedulers.io_main());
    }
}

```

图 2.6 Model 层使用方式

View 层在本系统中的使用方式为 Activity 进行接口声明，并实现所需内容。如图 2.7 所示。

```

public class LoginActivity extends BaseActivity<LoginPresenter, LoginModel>
    implements LoginContract.View {

    //获取个人信息
    @Override
    public void returnInformation(LoginDoctorBean loginDoctorBean) {
        //保存个人信息
        SaveUtil.putInfo(mContext, loginDoctorBean);
    }

}
    
```

图 2.7 View 层使用方式

Presenter 层在本系统中的使用方式如图 2.8 所示。

```

public class LoginPresenter extends LoginContract.Presenter {
    //获取个人信息
    @Override
    public void getInformation(String token) {
        mRxManage.add(mModel.getInformation(token)
            .subscribe(new RxSubscriber<InformationBean>(mContext, showDialog: false) {
                @Override
                protected void _onNext(InformationBean informationBean) {
                    mView.returnInformation(informationBean);
                }

                @Override
                protected void _onError(String message) {

                }
            }
        ));
    }
}
    
```

图 2.8 Presenter 层使用方式

2.4 SQLite 数据库

由于本系统中绝大部分数据都是从服务端获取，本地只需要存储部分静态数据以及储备非频繁更新的数据，并且由于 Android 设备的存储空间小，重载铁路安全系统采用 SQLite 数据库。

SQLite 是一款轻量级关系型数据库^[12]，具有传统关系型数据库 ACID 的操作规范，主要面向嵌入式设备，被广泛使用在嵌入式应用开发中。占用资源少，不需要额外的数据库驱动来进行装载。由于不存在界面化操作，因此所有数据库操作都依托于 JAVA 代码来读取并处理数据库文件。

2.5 百度定位 API

LBS (Location Based Service, 基于位置的服务) 的三大目标是^[13]: 你在哪里; 你和谁在一起; 附近有什么资源。其中“你在哪里”是 LBS 服务的核心。

百度智能定位服务, 是为了帮助广大开发者更好解决『你在哪里』这个难题而开放的服务。支持 GPS、WiFi、基站融合定位, 完美支持各类应用开发者对位置获取的诉求。

百度地图开放平台定位服务每日定位请求超过 300 亿次, 开发者市场占有率超过 70%。

2.6 极光推送 API

JPush 是经过考验的大规模 APP 推送平台^[14], 每天推送消息数超过 5 亿条。开发者集成 SDK 后, 可以通过调用 API 推送消息。同时, JPush 提供可视化的 web 端控制台发送通知, 统计分析推送效果。JPush 全面支持 Android, iOS, Winphone 三大手机平台。

2.7 MPAndroidChart

MPAndroidChart 是一款专门用于 Android 绘制图表的库, 在 Github 上开源发布。能够轻松实现折线图、条形图、条形折线图、圆饼图、雷达图、K 线图、泡泡图、网状图等常用图的绘制, 具有支持轴上的缩放、可拖动、可平移、可定制轴、可突出选中的值 (瞄准点及弹出 pop)、定制图例、动画显示等特点, 方便 Android 开发人员对图表进行制作。

3 需求分析

3.1 功能需求

（1）客户端版本验证。

本系统实现确保用户使用的是最新客户端软件。如果客户端是最新版本则进入系统；如果不是最新版本则进行以下判断：如果新的客户端必须更新，则必须等待更新并安装完成后才允许进入系统，如果不是必须更新，则可以直接进入系统。

（2）用户登录验证。

本系统实现拒绝非法用户的登录。本系统需要实现根据用户使用时提供的用户名及密码与服务端进行交互判定是否为合法用户，如果不是合法用户则提示用户名或密码错误，如果是合法用户则进入系统，并且当用户勾选记住密码后，能够保存当前用户名和密码，在下次登录时默认使用上一次登录成功的用户名和密码进行登录。

（3）危险源信息的记录、整改、验收、预警、档案查看、统计分析。

本系统要实现危险源信息的记录以及自动上报，待整改危险源信息的处理和查看，整改完成后验收销号。对于危险源等级升级、12 小时到期未整改、超期未整改的危险源，系统将通知手机用户尽快处理，对所有危险源内容进行图表式统计分析。对于记录并上报成功的危险源进行推送，相关部门的用户接收危险源推送并及时整改，在整改危险源完成后推送整改完成消息，记录的部门用户进行验收操作。

本模块需要实现危险源录入时所需的基础数据，提前保存到移动设备，提高危险源录入速度。实现待整改危险源超期预警提示功能。危险源整改期限当天，给整改人移动设备自动发送提示信息。实现危险源超期未整改提示功能。危险源超期限 1 天未整改，给整改人移动设备自动发送提示信息。危险源超期限 1 天以上未整改，给整改单位上级部门管理人员自动发送提示信息。实现危险源整改功能。实现危险源的验收销号功能。实现统计分析相关功能。

（4）隐患信息的记录、整改、验收、预警、档案查看、统计分析。

本系统要实现隐患信息的记录以及自动上报，待整改隐患信息的处理和查看，整改完成后验收销号。对于 12 小时到期未整改、超期未整改的隐患，系统通知手机用户尽快处理，对所有隐患内容进行图表式统计分析。对于记录并上报

成功的隐患信息进行推送，相关部门的用户接收隐患信息推送并及时整改，在整改隐患信息完成后推送整改完成消息，记录的部门用户进行验收操作。本模块实现已记录隐患的信息查看功能。每次展示 20 条信息，通过向下滚屏实现获取下一页的信息列表，通过向上滚屏实现获取上一页的信息列表。实现待整改隐患超期预警提示功能。隐患整改期限当天，给整改人移动设备自动发送提示信息。实现隐患超期未整改提示功能。隐患超期限 1 天未整改，给整改人移动设备自动发送提示信息。隐患超期限 1 天以上未整改，给整改单位上级部门管理人员自动发送提示信息。实现隐患整改功能。实现隐患验收销号功能。实现统计分析相关功能。

（5）监察通知的记录、整改、验收、预警、档案查看。

本系统要实现安监通知书的录入上报功能。安监通知书录入时所需的基础数据，提前保存到移动设备，提高录入速度。实现待整改安监通知书超期预警提示功能。安监通知书整改期限当天，给整改人移动设备自动发送提示信息。实现安监通知书超期未整改提示功能。安监通知书超期限 1 天未整改，给整改人移动设备自动发送提示信息。安监通知书超期限 1 天以上未整改，给整改单位上级部门管理人员自动发送提示信息。实现安监通知书整改功能。实现安监通知书验收销号功能。通过档案查看功能，用户可以看到已经记录过的安监通知书。每次展示 20 条信息，通过向下滚屏实现获取下一页的信息列表，通过向上滚屏实现获取上一页的信息列表。

（6）实现雨量、天气预报信息、轨温的查看以及预警信息的及时通知。

由于铁路安全与天气等信息有着十分密切的关系，因此在系统中要实现对天气、雨量、轨温以及灾害报警四项信息的信息推送，其中天气、雨量和轨温三项信息要做到每天进行更新，并且可以看到未来三天的信息。灾害预警要做到实时更新，页面上显示的信息必须是最新的信息。

（7）实现联控信息的移动录入和处理。

在铁路上发生需要多个不同部门进行配合的事件，要进行记录并调配，在联控信息中需要实现的功能有停送电管理（对停电和送电进行记录和推送）、调监系统（对机车调配进行信息记录和推送）、配合作业（对于某一项目的多部门配合进行记录和推送）、综合信息（对于其他需要多部门配合的事件进行记录和推送）。

（8）实现监控日报和事件汇总的查看。

服务端会每日生成对整个系统覆盖区域的所有相关铁路信息进行汇总和分析，在客户端要实现每次从服务端获取已经制作好的 HTML 界面，不经过任何处理直接展示，并且可以对日期进行搜索，获取过去某一天的事件汇总和监控日报。

3.2 性能需求

一款受用户欢迎的手机软件要有良好的使用体验，本系统主要通过以下几个方面来实现：

（1）界面友好。用户使用界面要简洁直观，功能布局合理，方便用户使用，界面切换流畅。

（2）响应迅速。对于所有的操作以及请求，要在短时间内给出处理结果，如果页面或数据内容出现不符合预期的内容，要提供人性化的错误界面或等待界面，让用户确认自己已经操作并且系统正在处理。

（3）节省系统资源。系统中所使用的 **activity**、**service** 要做到合理利用，避免冗余，隐藏的后台进程要及时进行无用释放。

（4）数据加密传输。在现在的互联网时代，信息已经变得越来越容易被泄露，因此系统中所涉及的所有重要信息项都要经过加密后再进行传输。为了抵抗可能发生的信息窃听、冒充发送、数据篡改、数据抵赖等危险，合理地重要数据进行加密便成为了一种有效的手段。

3.3 技术可行性分析

重载铁路安全系统使用原生 **Android** 语言编写，这是现在最为主流的移动端开发语言，并且不需考虑使用框架带来的框架自身问题以及引发的其他不可预知错误。系统通过 **Retrofit + RxJava + OkHttp** 网络框架组合调用 **Springboot** 的微服务后台，获取 **JSON** 数据，这是当前十分流行的搭配方式，移动端与服务端只需约定好数据传输格式和内容格式便可进行分离开发。系统采用的 **MVP** 代码结构更加方便开发人员进行代码编写和维护。**SQLite** 数据库简单有效的存储系统中的必要数据。数据传输安全加密和解密功能使用已经成熟的加密插件。除基础技术外，系统中使用到百度地图 **API** 实现位置定位功能，使用极光推送 **API** 实现消息的实时推送与接收，使用 **MPAndroidChart** 实现图表绘制。这三项插件在各自官网或主页上有着详细的文档可供参考。

综上所述，重载铁路智能安全系统移动客户端使用流行的开发方式与技术，调用成熟的外部插件，使用简洁的数据库，让整个系统满足客户开发要求。因此它在技术上可行。

4 系统设计

4.1 设计原则

（1）可行性原则

重载铁路安全系统采用 Retrofit + RxJava + OkHttp 网络框架组合调用 Springboot 的微服务后台进行数据传输的方式，调用如百度地图 API 等成熟的插件，保证了技术可行性。

（2）安全性原则

系统采用已经成熟的加密安全插件，对于所有数据进行加密传输。对于如用户密码等重要信息采用加密存储技术。

（3）易用性原则

由于本系统基于 Android 操作系统，界面美观大方，页面切换流畅，对于错误的提示精简合理。

（4）单一职责原则

系统中编写的所有实体类具有聚合性，即同一模块下功能集成在同一个类中，类的接口都具有单一性，每个接口只负责一项职责，并且每个接口的职责没有交集。

4.2 编程环境

本系统的编程环境是 Android Studio^[15]，这是 Google 公司研发的一款专门针对 Android 开发的编译器，它从 IntelliJ IDEA 延伸而来，因此包含了 IDEA 拥有的所有优点，包括智能代码助手、代码自动提示、重构、方便的版本工具操作界面。并且由于它专门为 Android 开发而生，Android Studio 提供基于 Gradle 的构建支持以及 Android 专属的重构和快速修复，它强大的 UI 式布局编辑器，适合绝大多数的 Android 开发人员。界面如图 4.1 所示。

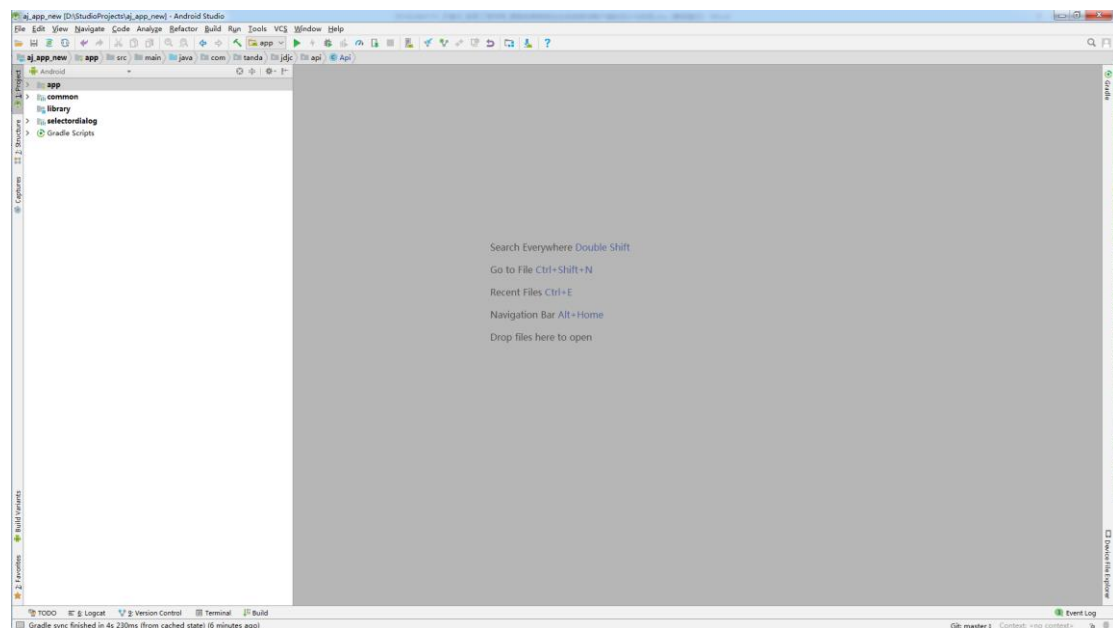


图 4.1 Android Studio 界面

4.3 功能设计

整个移动端划分为 8 大模块，分别为版本验证模块、登陆模块、危险源模块、隐患模块、监察通知模块、联控信息模块、灾害预警模块、监控日报模块。其中，登陆模块和版本验证模块为基础功能模块，危险源模块、隐患模块、监察通知模块和联控信息模块为录入功能模块，灾害预警模块和监控日报模块为展示功能模块。系统模块结构图如图 4.2 所示。

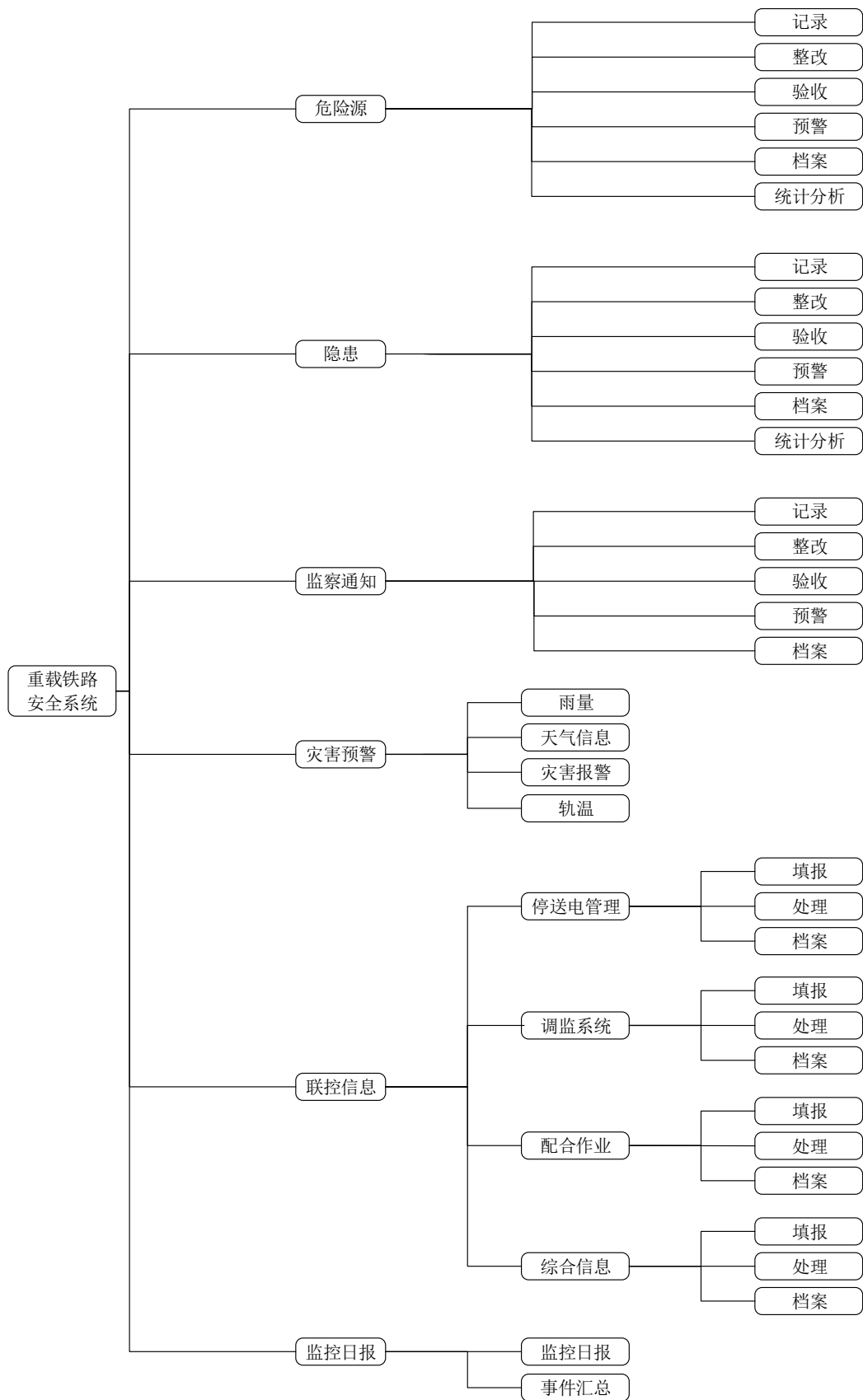


图 4.2 系统模块结构图

4.3.1 基础功能模块

(1) 版本验证模块

由于本系统的部分基础数据或录入项需要不定期更改，因此客户端的版本可能需要保持最新，进而保证客户端需要在最新版本下进行使用。

(2) 登陆模块

由于本系统为非公开 APP，因此不提供注册功能，所有用户的用户名、密码等信息都事先存储在服务端数据库中，并且客户端系统不提供用户管理功能，因此在本系统中登陆模块在用户输入用户名密码后，提交服务器验证是否通过，如果通过则进入主目录页面，否则提示“用户名或密码错误”。当用户再次开启 APP 时自动使用上次保存的用户名和密码进行登陆。

4.3.2 录入功能模块

(1) 危险源模块

危险源模块包含子模块，分别为记录、整改、验收、预警、档案、风险排序、趋势分析、现状分析、现状统计。

记录、整改、验收的具体信息项如表 4.1 所示。

表 4.1 危险源模块信息项

功能名称	结构划分	说明	输入项	约束条件
记录	记录	实现危险源信息的录入、存本地和上报功能。	专业、线别、地点、管理对象、危险源定位、危险源等	
	已记录	实现已记录危险源的信息查看功能。		
整改	待整改	实现危险源的整改功能。	整改结果、整改措施、现场图片	
	已整改	实现已整改危险源的信息查看功能。		
验收	待验收	实现危险源的验收功能。	验收情况	
	已验收	实现已验收危险源的信息查看功能。		

(2) 隐患模块

隐患模块包含子模块，分别为记录、整改、验收、预警、档案、隐患侦测几项功能。

记录、整改、验收的具体信息项如表 4.2 所示。

表 4.2 隐患模块信息项

功能名称	结构划分	说明	输入项	约束条件
记录	记录	实现隐患信息的录入 上报功能。	专业、线别、地点、 类别、隐患定位、隐 患等	
	已记录	实现已记录隐患的信 息查看功能。		
整改	待整改	实现隐患的整改功 能。	整改结果、整改措施、 现场图片	
	已整改	实现已整改隐患的信 息查看功能。		
验收情况	待验收	实现隐患的验收功 能。		
	已验收	实现已验收隐患的信 息查看功能。		

(3) 监察通知模块

监察通知模块包含子模块，分别为记录、整改、验收、预警、档案几项功能，其中填报、整改、验收的相关信息项如表 4.3 所示。

表 4.3 监察通知模块信息项

功能名称	结构划分	说明	输入项	约束条件
填报	填报	实现安监通知书的录入上报功能。	责任单位、发现问题、整改建议、查看范围等	
	已填报	实现已填报安监通知书的查看功能。		
整改	待整改	实现安监通知书整改功能。	整改情况、整改部门	
	已整改	实现已整改安监通知书的查看功能。		
验收	待验收	实现安监通知书验收销号功能。	验收意见、验收结果	
	已验收	实现已验收安监通知书的查看功能。		

（4）联控信息模块

联控信息模块包含子模块，分别为停送电管理，调监系统，配合作业，综合信息。这四个子模块又各自包含记录、处理、档案三个模块。

联控信息模块的主要功能是方便各个部门之间进行配合作业。在四个子模块中，停送电管理用来处理停送电信息并推送到各个配合部门。调监系统处理重载铁路上各个机车发生的事故导致的部门人员调配信息。配合作业处理突发事件，组织相关部门的相关人员临时组成处理小组赶赴案发现场进行事故处理，综合信息用来处理使用系统时遇到的需要多部门配合且不属于其余三个子模块的信息。

本模块中的每个子模块实现联控信息的录入上报功能。联控信息录入时所需的基础数据，提前保存到移动设备，提高录入速度。实现联控信息的处理功能。实现待处理联控提醒功能。在本安系统中每录入一条联控信息，自动给处理人推送危联控信息待处理信息。实现待查看联控提醒功能。当联控信息的处理人有回复信息时，自动给联控信息所涉及的人推送危联控信息待查看信息。相关信息项如表 4.4 所示。

表 4.4 联控信息模块信息项

功能名称	结构划分	说明	输入项	约束条件
停送电管理	记录	记录停电或送电时需要通知的各个部门功能。	接收部门、信息内容、现场拍照、附件	
	处理	对于已经记录的停电或送电通知进行处理。		被请求配合的部门必须配合
调监系统	记录	记录各个机车调配信息并推送到需要处理事件的接收部门。	设备位置、设备信息、故障现象、机车类型、机车号。	
	处理	对于已经记录的调监信息进行处理。		被请求配合的部门必须配合
配合作业	记录	记录重载铁路上发生的需要多个部门配合的突发状况。	开始与结束时间、项目名称、地点、内容、影响范围、作业方案。	
	处理	对于已经记录的突发状况配合作业信息进行配合处理，如果选择不配合则需要推送部门进行再次处理。		被请求配合的部门可以选择不配合。
综合信息	记录	记录重载铁路中发生的不属于以上三个模块的问题。	接收部门、信息内容、现场拍照、附件	
	处理	处理已经记录的综合信息。		被请求配合的部门必须配合

4.3.3 展示功能模块

本部分模块与录入功能模块的区别为所有的信息项皆不需要进行录入，全部信息从服务器端进行获取并直接展示或经过图表化处理展示。

（1）灾害预警模块

灾害预警模块包含四个子模块，分别为天气预报，雨量，灾害报警，轨温。这四项皆仅仅涉及系统所服务的两座城市及这两所城市的车站信息，其中灾害报警模块会显示最近一次的灾害报警信息，而其它模块则每日更新最新一天的相关信息。相关信息如表 4.5 所示。

表 4.5 灾害预警模块信息项

功能名称	结构划分	说明	输入项	约束条件
天气预报	列表	展示所有车站当天的天气情况。		根据登录用户所属单位默认显示该单位下的车站
	详情	展示所选车站未来 3 天天气预报的详细信息。		
雨量	列表	展示沿线各站 24 小时雨量信息。		根据登录用户所属单位默认显示该单位下的车站
	详情	查看某个车站 24 小时详细雨量信息。		
灾害报警	查看	显示最近一次的灾害报警信息，并且当有新的灾害报警推送到时立即进行声音报警。		只会显示最近一次信息。
轨温	列表	对于所有车站某一日轨温信息进行显示，可以通过状态标题栏查询功能进行日期选择。		根据登录用户所属单位默认显示该单位下的车站

（2）监控日报模块

监控日报模块是从服务端直接获取一个 HTML 文件页面代码，通过 textView 标签的 setText(Html.fromHtml(html))方法直接展示所有内容。

4.4 界面设计

界面通过业务类型划分为以下几个部分：初始界面、目录界面、列表界面、查询界面、详情查看界面、图表展示性界面。

所有界面包含顶部状态标题栏以及页面主题。状态标题栏分为三部分，左侧为返回上一页按钮，中间为标题，右侧为一个按钮，用于进入查询页面或完成查询操作。

（1）初始界面

初始界面为版本检测页面以及登陆页面，并且在此页面上会根据是否连接到网络判断目录页面所提供的功能，如果没有连接到网络，则对于危险源模块、隐患模块、监察通知模块、联控信息模块四项带有记录功能的模块仅提供记录功能其它功能直接隐藏，并且登陆用户信息采用上次保存密码设置自动登陆的用户信息，并且不可使用上报提交的功能。其它模块则直接不在目录中进行展示。

（2）目录界面

目录界面分为一级目录和二级目录，一级目录包含 3.3 节中所提到的四项录入功能模块以及两项展示功能，二级目录为各模块中细分的功能列表（如果有），对于没有细分的模块则不包含二级目录。二级目录与一级目录布局相同。

（3）列表界面

列表界面用于展示所查询项的简要信息集合，默认初始最多放置 20 条数据，在最顶部下拉可以更新，在最底部上划可以继续加载，当数据项不足时不再继续显示。

（4）查询界面与详情查看界面

这两种界面都是将所需要元素放置在布局中，并且进行元素值传入或传出。

（5）图表展示性界面

图表展示性界面用来展示危险源或隐患的统计分析图表，灾害预警模块中天气预报、雨量、轨温等信息的图表，监控日报的 HTML 页面等需要经过图表制作或页面转换等方法处理的信息。

4.5 安全设计

信息作为一种资源，它的普遍性、共享性、增值性、可处理性和多效用性，使其对于人类具有特别重要的意义。信息安全的实质就是要保护信息系统或信息网络中的信息资源免受各种类型的威胁、干扰和破坏，即保证信息的安全性。根

据国际标准化组织的定义，信息安全性的含义主要是指信息的完整性、可用性、保密性和可靠性。信息安全在任何国家、政府、部门、行业都必须十分重视的问题，是一个不容忽视的国家安全战略。但是，对于不同的部门和行业来说，其对信息安全的要求和重点却是有区别的。

本系统中，APP 与 Server 基于无线公网进行信息交互，APP 端 OS 基于 Android 4.2 开发，数据传输系统部署在 APP 与 Server 之间，实现之前的全部交互信息的加解密功能，保证信息在无线公网中传输安全，系统网络结构如图 4.3 所示。

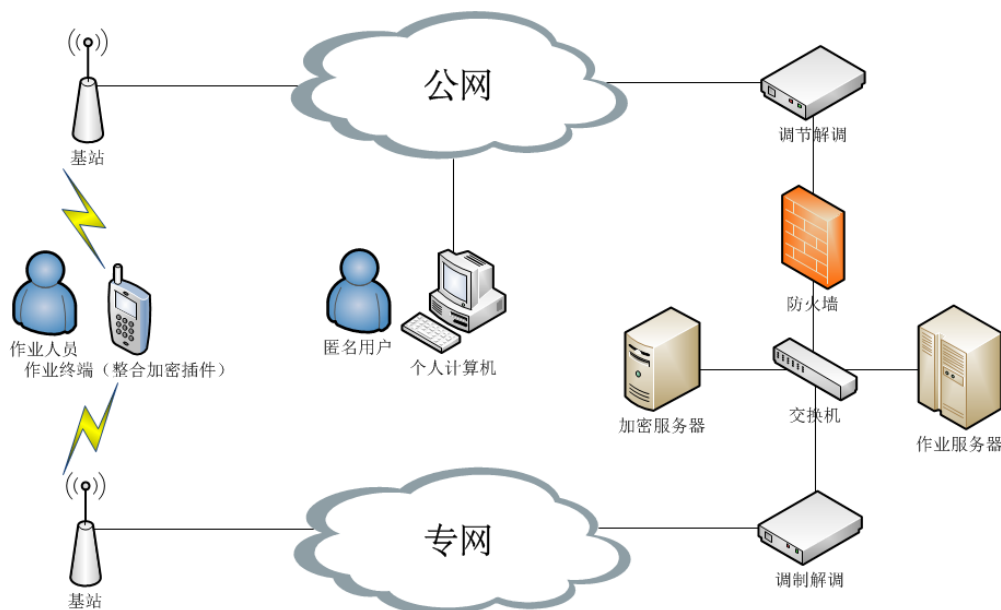


图 4.3 系统网络结构

APP 应用客户端实现业务功能，加解密功能插件绑定到 APP 应用客户端上，统一安装运行，加解密功能插件兼容 Android4.2 版本，加解密插件与加密服务器进行安全信息交互。

Server 端由加密服务器与业务服务器组成，其中业务服务器为与 APP 交互的业务功能 Server，加密服务器为信息安全传输 Server，实现 APP 与业务服务器间的交互信息加解密，同时实现密钥的生成、存储、分发、更新等功能，保证数据传输与密钥交互环节的信息安全。

加密解密流程如图 4.4 所示。

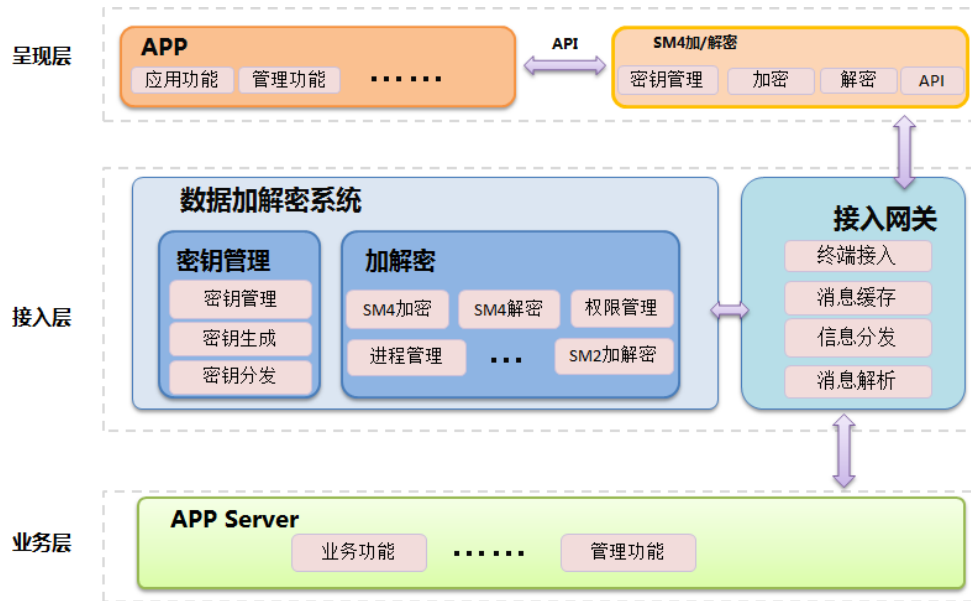


图 4.4 加密解密流程

APP 加解密插件包含密钥管理、SM4 数据加密、SM4 数据解密以及 APP 调用的功能 API 功能。数据加解密密钥信息，密钥管理模块负责与密钥管理 Server 进行密钥交互，实现密钥远程获取以及动态更新等功能。SM4 加/解密即采用国密 SM4 规范实现对 APP 与 Server 间交互信息加解密功能。

APP 与 Server 间为双向通信，信息交互方式复杂，接入层处理具备数据加解密功能外，还需要承担 APP 与 Server 间的信息中转，实现信息缓存、信息解析、信息分发等功能。

数据加解密针对两类数据进行加解密：业务信息数据采用 SM4 对称算法进行加解密，保证业务数据在网络中传输安全；密钥数据采用 SM2 非对称进行加解密，保证密钥在网络中传输的安全。数据加解密依赖于密钥数据，APP 与 Server 间存在多种端信息交互，对称加密需要大量的密钥支撑，密钥管理负责密钥信息的维护、更新、回收等功能。

5 系统实现

5.1 模块功能实现

第 4 节中所描述的系统中所有模块可拆分为以下几项主要功能：版本验证功能，登陆功能，记录功能，整改功能，验收功能，档案功能，统计分析功能，灾害查看功能，日报查看功能。由于模块中的部分功能为相同的功能，因此在本小节中以功能进行描述。所有功能的讲解适用于所有包含该功能的模块。

5.1.1 版本验证功能

每次开启 APP 时，系统自动向服务器端提供客户端的软件版本号，服务器端验证所使用的客户端版本是否为最新版本，若为最新则允许用户进入主界面。若为否，需要进行如下判断：

①判断当前最新版本是否为必须更新的版本。若为是，则强制用户更新最新的客户端版本才能使用。否则，进行判断②。

②判断用户是否使用 wifi 联网登陆系统。若为是，则强制用户更新最新的客户端版本才能使用。否则，进行判断③。

③提供用户选择界面，是否进行版本更新。若为是，则更新最新的客户端版本。若为否，则不更新最新的客户端版本，直接进入主界面。

本功能具体流程如图 5.1 所示。

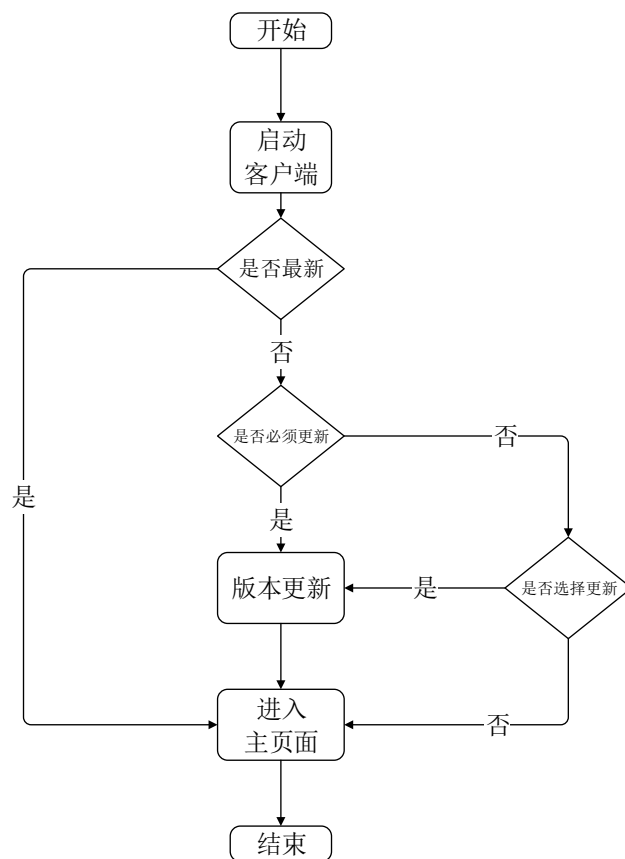


图 5.1 版本验证流程图

本功能的界面如图 5.2 所示。



图 5.2 版本验证界面

5.1.2 登陆功能

在用户输入用户名密码后，客户端提交用户名和加密后的密码到服务器，服务器验证是否为合法用户（数据库中存在用户名并且密码匹配），如果为合法用户则客户端通过并进入主目录页面，否则提示“用户名或密码错误”。登陆页面提供记住密码并自动登陆功能，如果勾选记住密码，则会在登陆成功后，在手机中通过密文保存当前登陆用户名以及相对应的密码。当用户再次开启 APP 时自动使用上次保存的用户名和密码进行登陆。

本功能的流程图如图 5.3 所示。

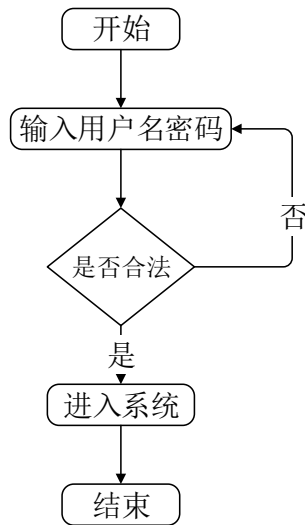


图 5.3 登陆流程图

本系统的界面如图 5.4 所示。



图 5.4 登录页面

5.1.3 记录功能

在危险源模块、隐患模块、监察通知模块、联控信息模块四个模块中有记录的功能。本功能实现了记录下对应问题，并自动上报到服务器，并且可以查看已经记录的信息列表和查看已经记录信息的具体内容。

本功能的流程图如图 5.5 所示。

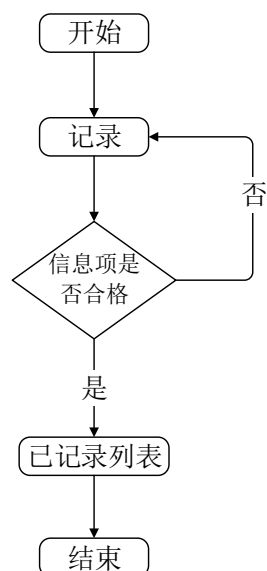


图 5.5 记录功能流程图

本功能的界面如图 5.6 所示。



图 5.6 记录界面

本功能的部分主要代码如下：

```

FragmentTransaction transaction = fragmentManager.beginTransaction();
mDangerWaitInfoFragment = new DangerWaitInfoFragment();
Bundle bundle = new Bundle();
bundle.putString("type", type);    //类型
bundle.putString("source", source);    //信息来源
bundle.putString("ID", list.get(0).get("mid") + "");
mWxyWzgInfoFragment.setArguments(bundle);
transaction.add(R.id.fl_container, mWxyWzgInfoFragment);
transaction.commit();
    
```

5.1.4 整改功能

在危险源模块、隐患模块、监察通知模块、联控信息模块四个模块中有整改的功能。本功能实现将记录下的问题进行整改，并在整改后自动上报到服务器，并且进行消息推送，在整改列表中可以看到需要整改的信息。在已整改列表中可以看到已经整改的信息。

本功能的流程图如图 5.7 所示。

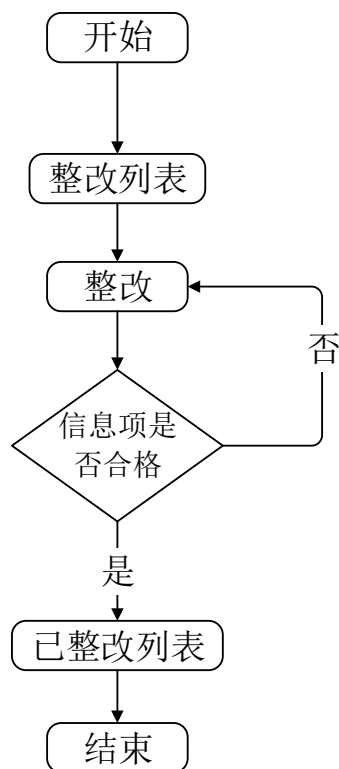


图 5.7 整改功能流程图

本功能的界面如图 5.8 所示。



图 5.8 整改界面

本功能的部分主要代码如下：

```
List<Map<String, Object>> list = (List<Map<String, Object>>) msg.obj;
if (list != null) {
    if (list.size() > 0) {
        act.updateUI(list);
    } else if (list.size() == 0) {
        showText(act.getInstance(), R.string.no_data);
    }
} else {
    showText(act.getInstance(), R.string.no_data);
}
```

5.1.5 验收功能

在危险源模块、隐患模块、监察通知模块三个模块中有验收功能。本功能实现对记录下问题的部门人员对整改后的问题进行验收。在验收列表可以查看需要验收的信息，在已验收的列表查看已经验收的信息。

本功能的流程图如图 5.9 所示。

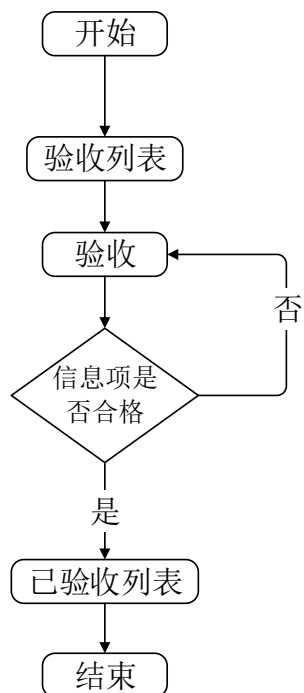


图 5.9 验收功能流程图

本功能的界面如图 5.10 所示。

图 5.10 验收页面

本功能的部分主要代码如下：

```
switch (which) {
    case 1:
        transaction.setCustomAnimations(R.anim.push_left_in,
            R.anim.push_left_out);
        break;
    case 2:
        transaction.setCustomAnimations(R.anim.push_right_in,
            R.anim.push_right_out);
        break;

    default:
        break;
}
```

5.1.6 档案功能

在危险源模块、隐患模块、监察通知模块、联控信息模块四个模块中有档案功能。本功能实现查询并列举出已有的对应模块问题。点击可以查看详细信息。

本功能的流程图如图 5.11 所示。

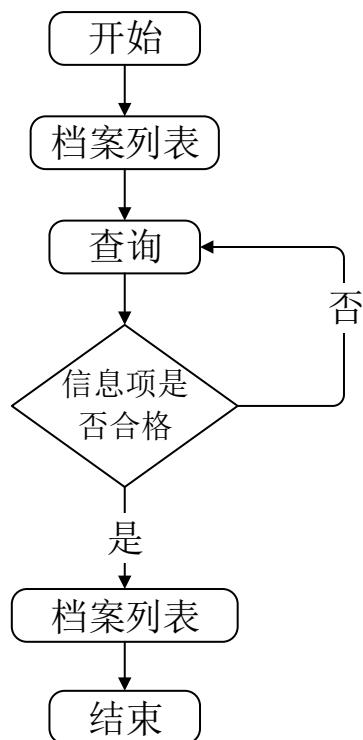


图 5.11 档案功能流程图

本功能的界面如图 5.12 所示。



图 5.12 档案页面

本功能的部分主要代码如下：

```
if (map.get("type") != null) {
    if (map.get("type ").equals("ZT02")) { //待整改--红色
        mColor = text_color[1];
    } else if (map.get("type ").equals("ZT03")) { //验收中-黄色
        mColor = text_color[2];
    } else if (map.get("type ").equals("ZT04")) { //超期待整改--橘黄色
        mColor = text_color[3];
    }
}
```

5.1.7 统计分析功能

在危险源模块、隐患模块两个模块中有统计分析功能。本功能实现将过往数据通过图表的方式展示。

本功能的流程图如图 5.13 所示。

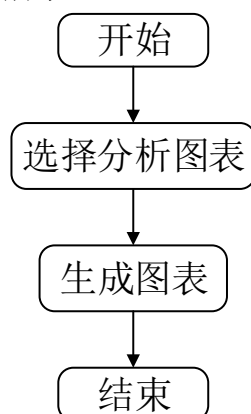


图 5.13 统计分析功能流程图

本功能的界面如图 5.14 所示。

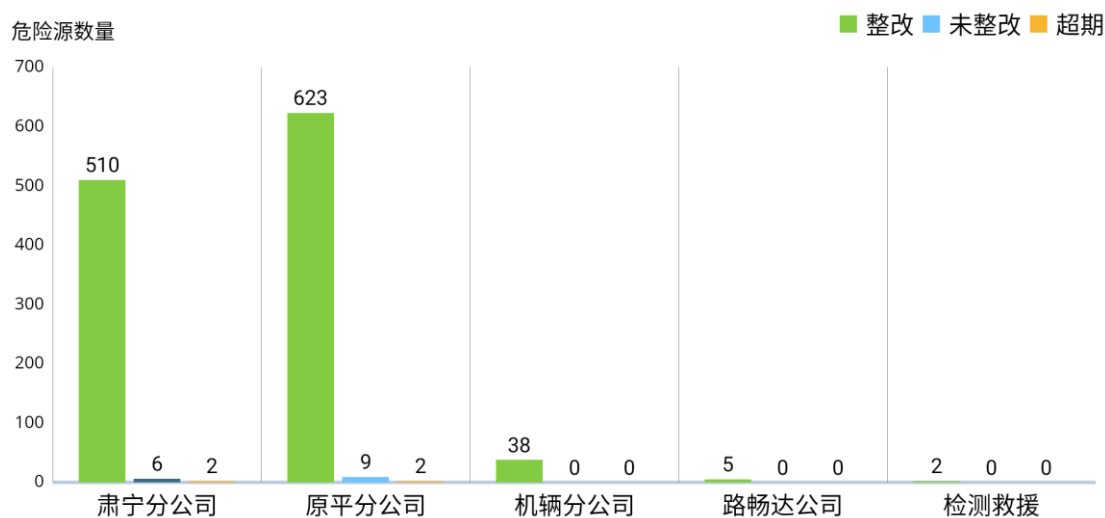


图 5.14 统计分析页面

本功能的部分主要代码如下：

```

<com.github.mikephil.charting.charts.BarChart
    android:id="@+id/mDanger"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>
    
```

(6) 灾害查看功能

在灾害预警模块中，可以查看天气、雨量、灾害报警、轨温四项信息。本功能实现在列表中查看当天对应信息，并且可以通过点击某一条数据在详情页面查看未来三天的信息。

本功能的流程图如图 5.15 所示。

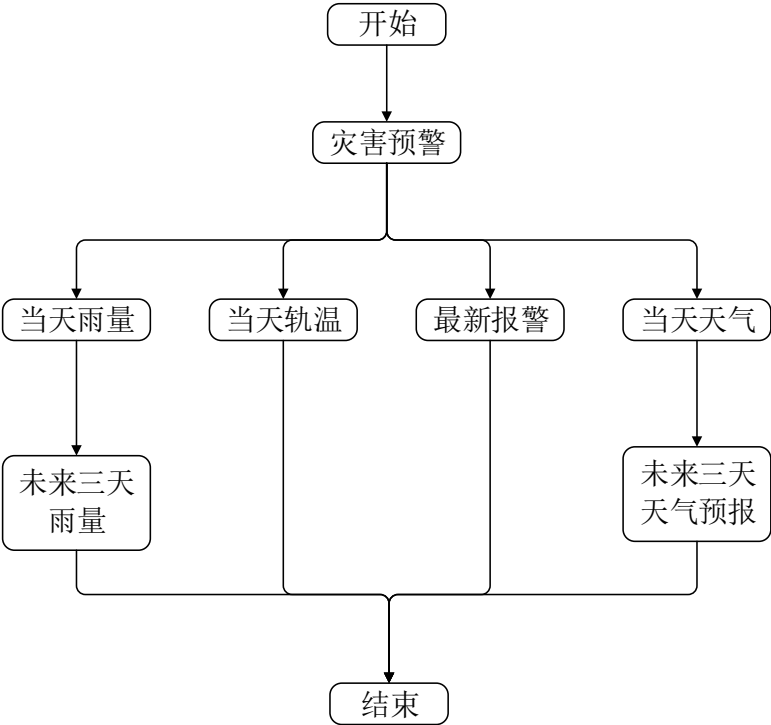


图 5.15 灾害预警功能流程图

本功能的界面如图 5.16 所示。

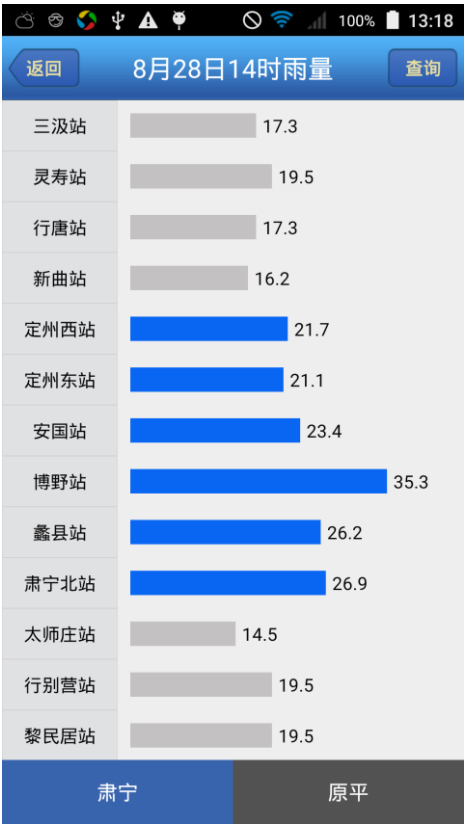


图 5.16 雨量界面

本功能的部分主要代码如下：

```

if (rainfallValue < 20) {
    mHolde.tv_rainfall.setBackgroundResource(R.color.rainfall_color_20);
} else if (rainfallValue >= 20 && rainfallValue < 40) {
    mHolde.tv_rainfall.setBackgroundResource(R.color.rainfall_color_40);
} else if (rainfallValue >= 40 && rainfallValue < 60) {
    mHolde.tv_rainfall.setBackgroundResource(R.color.rainfall_color_60);
} else if (rainfallValue >= 60 && rainfallValue < 90) {
    mHolde.tv_rainfall.setBackgroundResource(R.color.rainfall_color_90);
} else if (rainfallValue >= 90) {
    mHolde.tv_rainfall.setBackgroundResource(R.color.rainfall_color_91up);
}
    
```

5.1.8 日报查看功能

在监控日报模块中，可以查看某一日的整体监控日报和事件汇总。本功能实现获取服务端已经生成的 HTML 页面进行展示。通过查询条件可以查询过去某一日的监控日报和事件汇总。

本功能的流程图如图 5.17 所示。

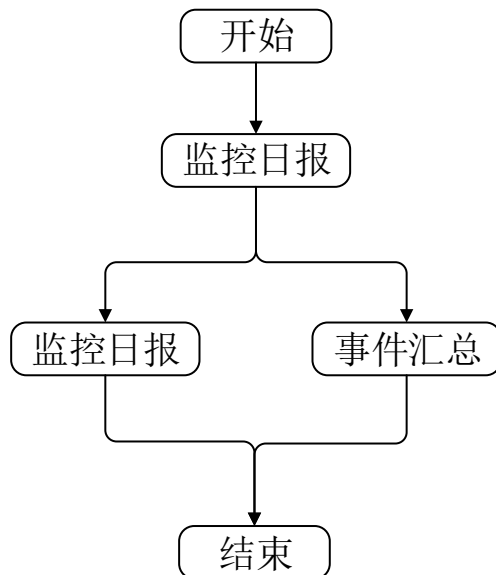


图 5.17 日报查看功能模块

本功能的界面如图 5.18 所示。

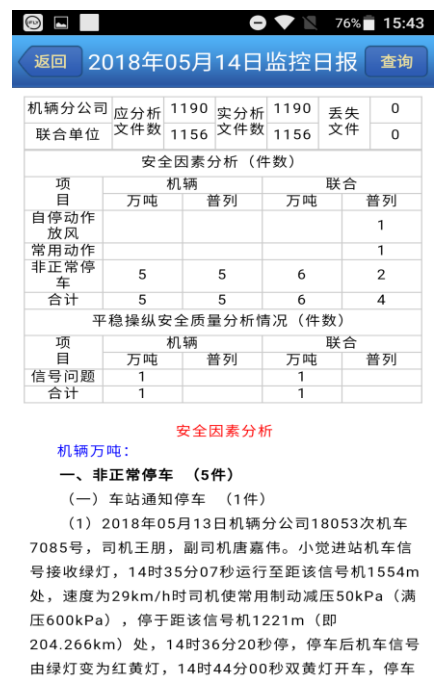


图 5.18 监控日报页面

本功能的主要代码部分如下:

```
content.setText(Html.fromHtml(origin));
```

5.2 插件功能实现

本系统中除基本业务功能外, 由于需要用到位置定位、信息推送、图表制作功能, 因此通过百度定位 API 实现定位功能, 通过极光推送 API 实现实时消息推送与接收, 通过 MPAndroidChart 控件实现图标制作功能。

(1) 定位

导入百度提供的 BaiduLBS_Android.jar 包及相关文件, 在 AndroidManifest.xml 文件中声明百度定位 API 的 service, 并在“meta-data”标签中标明 API_KEY 及其参数。声明相关权限。如图 5.19 所示。

```
<service
    android:name="com.baidu.location.f"
    android:enabled="true"
    android:process=":remote" >
</service>
<meta-data
    android:name="com.baidu.lbsapi.API_KEY"
    android:value="3yoD80PlKwMoP9yHHvmhS77k" />
```

图 5.19 百度定位 API 调用部分代码

在代码中调用百度定位 API 提供的方法进行当前定位获取。

(2) 消息推送

导入极光推送提供的相关 jar 包,并在 AndroidManifest.xml 中声明相关权限、service、receiver,并声明 key 值。部分声明如图 5.20 所示。

```
<!-- com.ssmis.jcxh.permission.JPUSH_MESSAGE -->
<!-- Required 极光推送权限 -->
<permission
    android:name="com.ssmis.jcxh.permission.JPUSH_MESSAGE"
    android:protectionLevel="signature" />
```

图 5.20 极光推送部分代码

在代码中调用极光推送 API 提供的方法,并保持推送服务活跃,实现消息实时接收功能。

(3) 图表制作

MPAndroidChart 是 Android 一个强大且容易使用的图表库,它支持线状图、柱状图、散点图、饼状图和蜘蛛网状图等等常见的图表,并且支持缩放、拖动(平移)、选择和动画。在本系统中主要使用到表格制作以及柱状图、散点图。

由于在 Android Studio 中使用 Gradle 方式十分简便,因此在 build.gradle 文件中进行关联声明:

```
repositories {
    maven { url "https://jitpack.io" }
}
dependencies {
    compile 'com.github.PhilJay:MPAndroidChart:v2.1.6'
}
```

在布局文件中根据所需图表,放入 MPAndroidChart 提供的元素标签,在代码中设置数据值以及操作其它属性,进行图表绘制。

6 系统性能评析

6.1 系统性能优化

(1) 代码结构优化。从 MVC 结构框架更换为 MVP 结构框架，虽然多了许多不必要的代码，但将代码重合度降低，尽量将每个功能下的代码耦合度降到最低。

(2) 界面优化。对于界面背景、图标背景实现颜色过度，而不是整个版面一个颜色；对图表显示的衔接处进行显示锐化处理，让连接更加平滑；对于界面切换使用动画出现或消失，让界面切换不显得突兀；对于部分界面除了可以使用状态标题栏的返回按钮返回上一级界面或点击目标元素项进入下一级界面外，还可以通过左右滑动进行页面切换。

(3) 图片显示优化。对于第一次加载的图片不进行直接加载，而是通过 RxJava 的异步方法进行加载，并且顺序单张图片加载，使页面其它部分的加载不受网络状态的影响。

6.2 性能评析

本系统由于采用 Retrofit + RxJava + OkHttp 的 Android 端网络框架连接微服务的服务器端进行数据交互，因此在数据加载方面十分迅速，并且由于所搭配的额外 service 较少，对于宿主机的资源消耗是较低的。

在代码架构方面，由于减少了不同模块之间的耦合性，大大降低了以后进行维护或修改漏洞时对于维护人员的工作量，并且由于良好的代码习惯，代码中每一块相关功能都有着详细的注释，方便后续人员跟进维护。

根据前几年的统计数据，预计本系统用户在 2000 人左右，每天发现的危险源或隐患一共为人均 1 处，每处隐患平均需要上传 1 张图片，每张图片的大小平均 200KB。

本系统相关服务器存储信息参数如表 6.1 所示。

表 6.1 存数信息参数表

参数	数值	说明
图片文件保存月数	永久	图片文件作为处理过程数据永久保留
文件系统膨胀系统	1.1	文件存入文件系统中的管理消耗
系统冗余系数	30%	考虑到用户和业务的增长，及随着业务的推广，访问人数的增加，考虑 30% 的冗余

按照处理规模估算，每天上传的图片约为 2000 张，约为 400M 大小，在运行 10 年的情况下，需要消耗的存储空间为：

$$400M * 1.1 * 365 * 10 / 0.7 = 2.5 \text{ TB}$$

本系统的运行环境要求：基于 4.2 以上版本的 Android 手机操作系统，可用存储空间 8G 以上，运行内存 2G 以上。

7 总结与展望

7.1 总结

移动终端和计算机技术的发展日新月异。在设计中，依据先进性与实用性并重的原则；技术实现上，采用先进、成熟、可靠性高、稳定性好的软件产品；业务实施中，以需求为导向，依托先进的移动通讯技术，建设重载铁路智能安全系统移动客户端。系统的层次结构清晰、明确，每一层之间采用标准的协议接口，协议之间保证与国际、国家标准相一致。整个网络传输采用标准的 TCP/IP 协议。重载铁路的研究一直是我国重点研究项目，本系统的开发工作将为重载铁路的安全增加一份保障。具体的工作有：

（1）介绍了重载铁路安全系统的研究意义，简述了重载铁路在国内外发展的状况及趋势。

（2）对系统的整体需求进行描述，包括业务的需求、系统性能的需求、数据库需求。

（3）对系统的设计进行阐述，包括系统的编程环境、功能设计、界面设计、安全设计。其中重点描述了功能设计，对整个系统的 8 大功能模块进行详细介绍。

（4）对于系统各项内容的实现进行描述。其中网络框架的实现使用了目前非常受欢迎的 Retrofit + RxJava + OkHttp 组合，代码结构使用 MVP 结构，系统额外调用了百度定位 API，极光推送 API，使用 MPAndroidChart 系列控件进行图表制作。

（5）最后对系统的性能进行优化与评析，并且描述系统所需设备硬件环境。

7.2 展望

本系统根据当前重载铁路的安全处理系统进行了移动端开发，但由于我国铁路行业发展迅速，许多功能在未来可能会不断进行更新，当前部分功能虽然适用于目前的行业规范，但当铁路安全处理规范变更时，当前系统的一些模块可能被废弃。所以在以后可以将系统中的部分模块或接口更改成适配器模式适用的格式，这样可以在修改或更新模块时快速完成工作。

参考文献

- [1] 赵彦星. 铁路本质安全管理信息系统[J]. 中国铁路. 2014 (4) :19-23
- [2] 《人民铁道》报铁路跨越式发展整体框架研究课题组. 利锥方凿为我用 鉴往开来大业兴——从世界铁路发展看中国铁路[J]. 铁道经济研究. 2005(04)
- [3] 靳岩, 姚尚朗. Android 入门开发与实践[M]. 北京: 人民邮电出版社, 2012
- [4] J.F.DiMarzio. Android a programmer's Guide[M]. McGraw.Hill, 2012
- [5] 汪源, 刘芳. Android 的进程与单线程模型浅析[J]. 仪器仪表用户, 2011,(5):30-31
- [6] 李佐彬. Android 开发入门与实战体验[M]. 北京: 机械工业出版社, 2010
- [7] 简书. Retrofit + RxJava + OkHttp 让网络请求变的简单 - 基础篇 [EB/OL]. <https://www.jianshu.com/p/5bc866b9cbb9>
- [8] 霍斯特曼科内尔. Java 核心技术[M]. 北京: 机械工业出版社, 2014
- [9] 曾露. MVP 模式在 Android 中的应用研究[J]. 软件, 2016, 37(06): 75-78
- [10] 杨靖, 洪蕾. MVP 设计模式研究及在 Android 中的设计[J]. 科技创新导报, 2016, 13(34): 97-98
- [11] PAU Valentin Corneliu, MIHAILESCU Marius Iulian, STANESCU Octavian. Model View Presenter Design Pattern[J]. Journal of Computer Science and Control Systems, 2010, 3(1)
- [12] 张恒喜, 史争军. 基于 SQLite 的 Android 数据库编程[J]. 电脑编程技巧与维护, 2011(21): 30-31
- [13] 杜传明. 百度地图 API 在小型地理信息系统中的应用[J]. 测绘与空间地理信息, 2011.(2)
- [14] 黄英伟. 整合极光推送和地图定位的智能手机打车应用研究[D]. 湖南大学, 2016
- [15] 周明韬. 基于 Android Studio 的天气预报 APP 设计与实现[J]. 电子制作, 2017(21): 40-42

致谢

时光飞逝，我的大学生活即将结束，想在此感谢李晓宇老师和黄鹏导师，他们在我实习期间热情的帮助我解决生活和学习中的问题，对于我完成毕业论文给予了许多帮助。在论文撰写过程中，李晓宇老师认真的指点出我论文中存在的问题并热情的解答我的疑惑、帮助我完善整篇论文的结构及内容。黄鹏导师在我实习期间带领我完成许多困难的工作，帮助我确定了论文题目的大方向，提供给我许多资料来帮助我解决毕业设计中遇到的难点。

另外十分感谢北京东方坦达科技有限公司，因为其包容、热情的氛围，让我在第一次踏上社会时没有感受到被冷落，公司对待我们就像对待正式员工，不但在工作上，在生活中也提供很多帮助。各位同事也耐心的为我提供问题的解决方法，让我在大四一年中有了更多的进步。

谨以此致谢所有帮助我的导师、同学、同事以及抽时间对本文进行审阅的各位老师。