**程序设计项目课程设计任务书**

**20202231014关竣佑**

**[一，系统需求分析 1](#_Toc14754)**

**[二，总体设计 1](#_Toc30339)**

**[三，界面各功能模块的具体实现 1](#_Toc2191)**

**[四，系统调试 3](#_Toc2191)**

**[五，使用说明及编程体会 3](#_Toc2191)**

**[六，附：源程序 1](#_Toc2191)**

**一，系统需求分析**

众所周知，获取到准确的天气状况是很多人出门或者规划当天的行程的重要依据，因此及时方便地获取天气情况成为了很多人的刚需。 本项目为天气预报安卓APP，目的在于在移动端为用户显示用户所在城市及其它城市三天的温度天气，空气质量等天气信息。便于用户快捷方便地获取到最新的天气资讯。本项目旨在为广大用户提供一个准确可靠的免费天气服务，为广大用户实现“手机在手，天气我知”。

本天气预报APP 具有切换城市显示天气，空气质量状况的功能，并自动获取识别用户所在的城市，为用户提供所在城市的天气预报。用户切换到菜单页，选择城市后跳转到该城市的天气预报，用户亦可以在菜单页等城市识别成功后查看所在城市的天气情况。

**二，总体设计**

本项目是一个无后端的简易天气预报APP，安装环境为安卓系统或鸿蒙系统。本APP设计有三个页面，分别为欢迎页面Home , 菜单页面Menu 以及 天气情况展示页面 weather 。在菜单页面，一共有三个城市选择按钮分别为北京广州和佛山。当用户选择其中一个城市时顶部显示当前选择的城市并跳转到该城市天气预报页面。菜单页面底部有一个面板显示识别用户所处的当前城市，识别成功后以后点击此按钮可跳转到该城市天气。在天气页面，顶部展示今天白天/夜晚的天气，气温。下面展示未来两天的天气和 温度信息。

本项目采用Java+XML+Android SDK等进行开发，使用IDEA作为开发平台。项目大致遵循MVC的设计模式。分为逻辑控制模块，样式模块和数据层模块。由于本项目不同于一般的前后端交互web项目，无需数据持久层，因此数据层模块被弱化。而本项目需要大量请求外界API ，故每个页面从逻辑控制层中剥离出单独的请求类，用于请求接口。在样式层中使用XML来控制每个页面的按钮，背景，图标等样式。在逻辑控制模块使用Java代码控制样式层的样式，文字的改变。

由于本APP一共有三页，而且并列，故使用IDEA 的Bottom Navigation Activity Android模板来进行创建。

**三，界面各功能模块的具体实现**

在菜单页面中有以下的关键类：Menu ，getLocationCity ，MyLocationListener ， OnClick

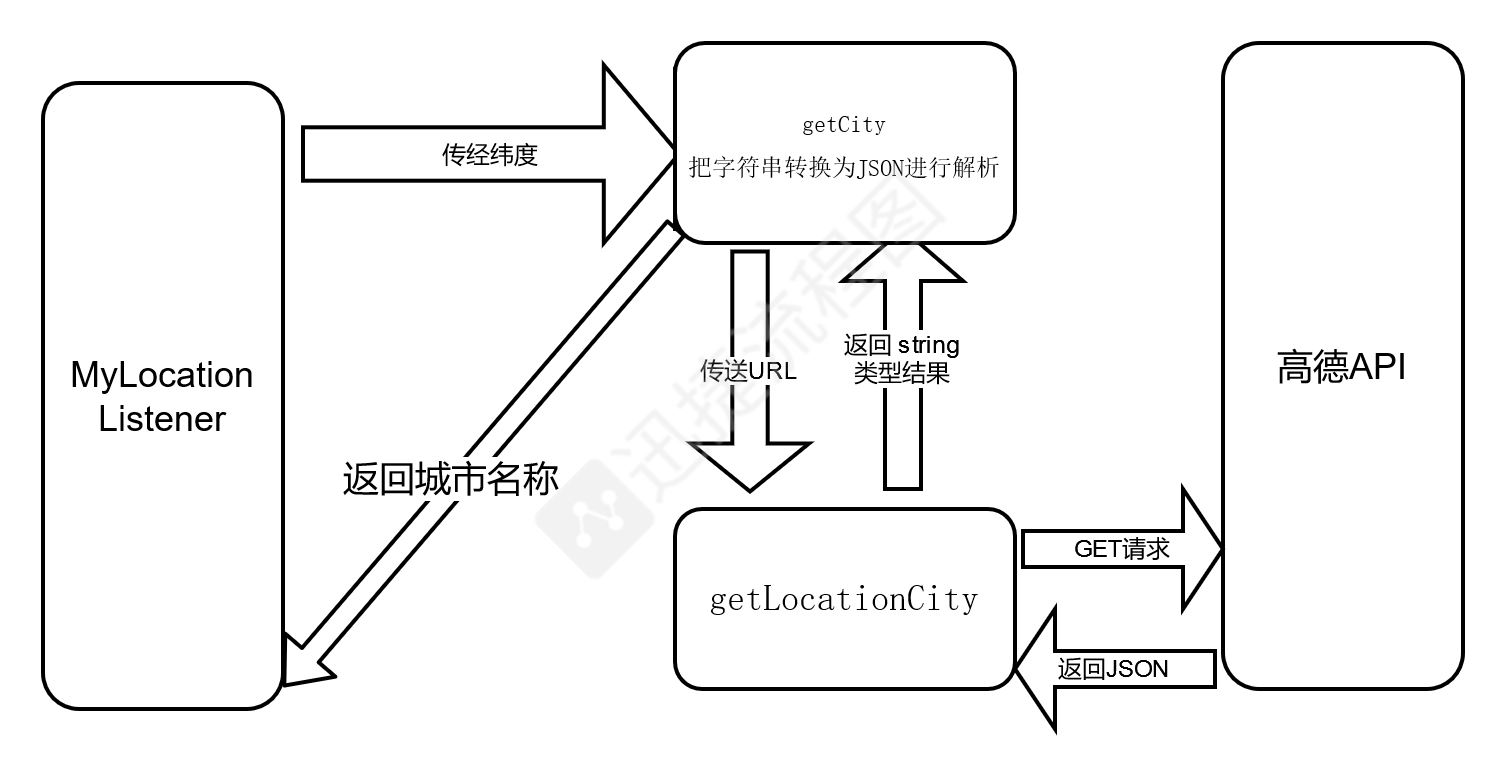
Menu类为主要控制类，MyLocationListener 和OnClick 类都是该类的成员类。由于本页面中用户选择了一个城市后要跳转到天气页面。故考虑编写OnClick类实现View.OnClickListener

接口(重写 onClick方法)来对按钮进行监听。同时由于在页面跳转的时候要把城市名称传过去，所以还要进行跨页面传参。考虑在mobile\_navigation.xml文件中设置页面间传参。在Menu类中把要穿走的city装载后通过controller.navigate 传走，在天气页面类NotificationsFragment中通过getArguments() 接收传来的城市名称。

在Menu类中还有一个关键的功能，识别用户所在城市。此功能需要用户的定位功能。而在Android上进行定位大致有三种方法：使用Android 自带的SDK进行GPS定位，Android 自带的SDK进行网络定位，导入高德/百度等第三方 SDK进行定位。在开发过程中，由于网络定位的精准度高且较稳定，原先考虑使用网络定位。但是测试程序过程中定位结果常常是null ，经查阅资料由于国内的手机厂商将Google的网络服务组件移除了。后又转为采用高德SDK定位，但是由于导入配环境未成功而不得不采用获取手机GPS定位。

采用GPS定位首先要获取用户的GPS权限。在MainActivity类里面写入了，判断若未给予本APP定位权限则需要进行请求。在Menu类中编写 MyLocationListener 类实现 LocationListener 接口，持续对获取用户GPS位置请求进行监听。一旦接收到用户所在的经纬度将交付给 getLocationCity 类处理。GetLocationCity 类中getCity方法接收经纬度并调用同类中的方法getReq 向高德地理编码API发送GET请求。高德API返回JSON数据后经过处理返回一个StringBuilder给getCity方法。该方法将其转换为JSON对象后进行解析，返回用户所在城市名称给Menu类。

流程图如下



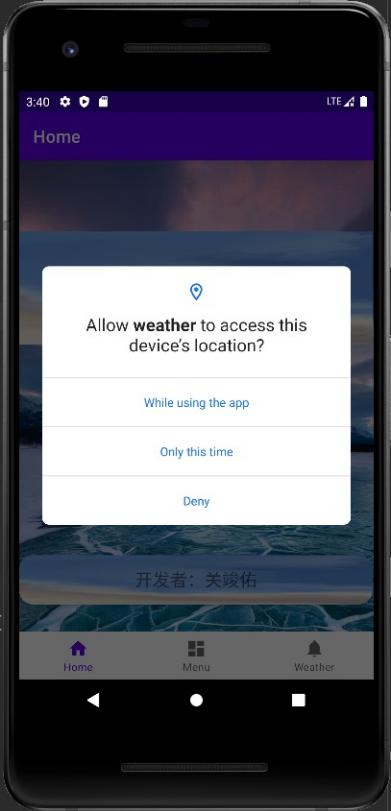
当用户进入了天气页面时有三个关键类：NotificationsFragment ， getRequest , getCityCode 。 getCityCode 类用于GET请求高德接口，用城市名称请求获取高德地理编码。当从上一个页面传来城市名称时，需要调用此类来获取该城市的地理编码。getRequest类用于使用城市地理编码请求高德天气API获取三天天气信息。当NotificationsFragment类获得了地理编码后调用getRequest类获取天气信息。并且获取用户手机的系统时间判断是夜间还是白天。最后通过操纵XML的样式显示天气信息。

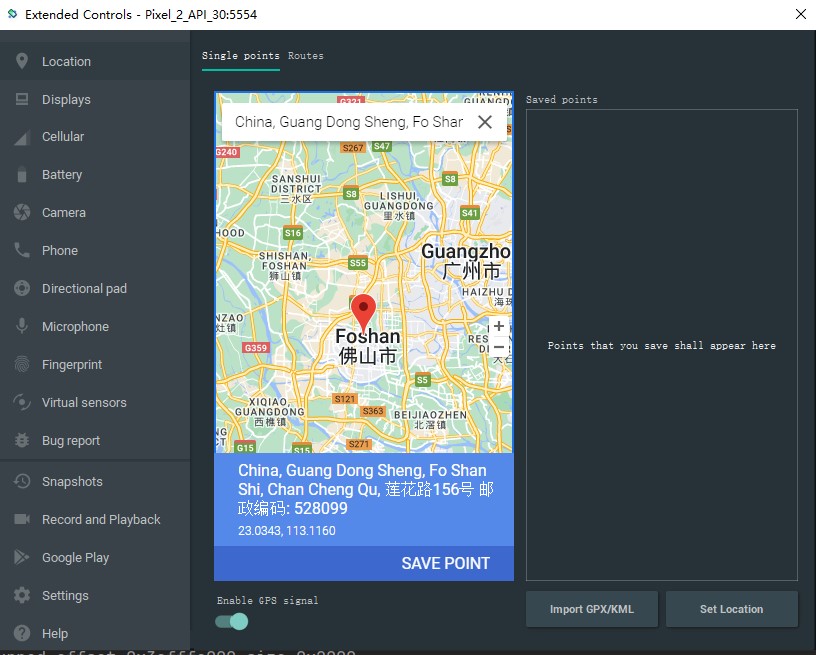
1. **系统调试**
2. **使用说明及编程体会**

使用说明：

1. 在安卓/鸿蒙手机上安装好apk后，点击apk启动。给予定位权限
2. 选择城市或等待定位成功后选择城市
3. 跳转到天气页面查看该城市天气

以下为在Android模拟器上运行的情况(模拟器的GPS位置被设定为佛山市内)

编程体会：

在编写该Android app的过程中经过许多困难及阻碍，都通过查阅资料以及不懈的尝试去解决。由于第一次接触Android开发，经验能力不足，因此仍有不少问题无法解决。

在请求高德API时，Android抛出了异常不允许在主线程中调用网络。意在防止网络阻塞主线程导致程序“假死”。考虑过开启子线程进行网络操作，但是由于知识不足难以解决线程间的通信，将请求的结果返回给主线程而放弃。最后采用折衷方法，使用Android严苛模式来在主线程强制执行网络操作。

在进行定位获取时，尝试了多种方法，因为自身能力问题环境没配成功以及客观上的阻碍导致被迫选择了手机GPS定位方法。

在测试过程中，发现onLocationChanged函数不断地获取手机GPS定位，每刷新一次就重新向高德API发送请求，导致不过多久API调用量便达到5000。考虑到天气预报APP不必频繁获取定位数据(因为用户不可能在打开APP期间频繁改变所在城市)，所以我在该类设置了一个静态的flag变量， 通过静态成员函数初始化为false. 当获取到城市时赋为true ,则APP前台使用期间只需要取出城市即可，且设置当flag 为 true 时不再执行onLocationChanged函数，避免了重复大量请求。