-

**智能终端软件开发**

**（天气预报）**

**课程论文报告**

|  |  |
| --- | --- |
| 课程名称： | 智能终端软件开发 |
| 课程题目： | 基于Android的天气预报APP |
| 姓 名： | 程聪 成海斌 |
| 指导教师： | 徐 战 亚 |

2020年 11 月21日

**《智能终端软件开发》课程**

**课程论文结果评定**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 评定内容 | 评定指标 | | 评分权值 | 评定成绩 |
| 工作态度 | 认真努力，讨论充分，命题合理；问题分析和反馈及时；按期完成规定的任务 | | 0.1 |  |
| 论文格式 | 格式规范、结构合理、内容完整 | | 0.1 |  |
| 论文质量 | 选题正确；分析全面；设计思路清晰；原型图设计美观、实用，表述清晰。技术方案合理，文字通顺，技术用语准确，符号统一，编号齐全，书写工整规范，图表完备、整洁、正确。 | | 0.6 |  |
| 汇报评价 | 按要求组织汇报内容，充实详细，讲解全面系统，PPT和对应素材准备充分； | | 0.1 |  |
| 工作量与工作难度 | 工作量饱满，工作难度大 | | 0.1 |  |
| 成绩： | | | | |
| 评定时间 | | 2020年 月 日 | | |

**目 录**

[摘要 1](#_Toc56285299)

[关键字 1](#_Toc56285300)

[1、 天气预报软件领域现状分析 2](#_Toc56285301)

[1.1 天气预报软件的现状分析 2](#_Toc56285302)

[1.2 Android的发展趋势 2](#_Toc56285303)

[1.3 开发的目的与意义 2](#_Toc56285304)

[2、 目标群体分析 2](#_Toc56285305)

[2.1当前使用天气预报的用户的基本状况 2](#_Toc56285306)

[2.2当前使用天气预报的用户的行为特点 2](#_Toc56285307)

[2.3当前使用天气预报的用户的需求 3](#_Toc56285308)

[2.4 分析总结 3](#_Toc56285309)

[3、 产品构思 3](#_Toc56285310)

[4、 功能设计 3](#_Toc56285311)

[5、 技术路线与关键点 3](#_Toc56285312)

[5.1 技术路线 3](#_Toc56285313)

[5.2 系统聚类分析 5](#_Toc56285314)

[5.3模型的检验与分析 5](#_Toc56285315)

[6、 总体UI设计 6](#_Toc56285316)

[7、 功能模块划分 6](#_Toc56285317)

[8、 具体实现 6](#_Toc56285318)

[9、 课程总结 6](#_Toc56285319)

[参考文献 6](#_Toc56285320)

[附 录 7](#_Toc56285321)

**报告简介**

摘要**：**

本设计使用Android设计技术开发了一种运用在Android系统上的手机天气预报软件系统，本系统通过选择城市来获得天气，风向，风向，温度等信息。基于手机的天气预报系统软件可以使用户对于各地的天气情况的实时掌握，极大的方便了用户的出行和行程安排，避免了不必要的麻烦，具有很强的实用性。

本课题研究的是基于Android平台下的天气软件，目的是使用户能随时随地的查询天气信息及天气变化情况，让生活更加舒适方便。本文讲述Android平台下天气软件的界面设计以及数据的获取和解析，主要包括界面的实现，控件的使用，界面的布局和数据处理等内容，结合中国气象网的天气数据，将准确的天气情况以友好的方式呈现给用户，使用户更加赏心悦目的体验本软件的服务。通过对界面的布局及控件的应用等相关内容的学习，使我们对Android平台下界面开发工作有一个初步的了解，并且经过这次开发过程，能够基本掌握这些内容。

关键字**：** Android ；天气软件 ；智能手机平台

# 天气预报软件领域现状分析

## 1.1 天气预报app的现状分析

在科技日益发达的今天，社会竞争越来越激烈，这要求人们不断的摄取信息，同时手机发展到现在已经成为广大人民群众的日常使用的通读工具，手机的功能也日趋完善。我国的通读移动通讯功能同样得到了很大的提高。手机软件可以安装在手机上的软件，完善原始系统的不足与个性化。随着科技的发展，现在手机的功能也越来越多，越来越强大。经济发展和社会进步，人们的出行更加频繁，对于天气的关注更加强烈，基于这种原因我们开发了一种运用在Android系统上的手机天气预报软件系统，本系统通过选择城市来获得天气，风向，风向，温度等信息。基于手机的天气预报系统软件可以使用户对于各地的天气情况的实时掌握，极大的方便了用户的出行和行程安排，避免了不必要的麻烦。本次设计在Android手机系统可以准确预测包括：时间，温度，风向等，并显示在手机界面上，方便准确地了解天气信息。

## 1.2 Android系统的发展趋势

Android最吸引人的特点是开源特性，因其开源，允许所有人对其代码进行开发和修改，同时又由于其代码的简单易学性，Android在全世界拥有众多的粉丝。Android在中国的前景十分广阔，首先是有成熟的消费者，在国内，Android社区十分红火，这些社区为Android在中国的普及做了很好的推广作用。国内厂商和运营商也纷纷加入了Android阵营，包括中国移动，中国联通，中兴通讯，华为通讯，联想等大企业，同时不仅仅局限于手机，国内厂家也陆续推出了采用Android系统的产品，我们可以预见Android将会被广泛应用在国产智能上网设备上，将进一步扩大Android系统的应用范围。

## 1.3 目的和意义

天气软件是一种非常实用的信息服务，随着智能手机的不断普及，各种智能手机平台下都有各种各样的天气软件。Android作为现在主流的智能手机之一，自然也需要这方面的软件。天气关系到人们的日常生活，如告知温暖，方便出行等。一款简单实用的天气软件会很吸引用户。

基于Android平台的天气软件已经有很多，经过分析，用户第一眼往往是一个软件的外观，一个好看的界面才能让用户以一种爱美之心去尝试，才能进一步了解一个软件的好坏，所以吸引眼球的界面是争取用户的第一步。因此，界面设计工作对应用程序来说非常重要，在界面上吸引了用户，才能进一步让用户体验应用程序中的功能，使用这个软件。

# 目标群体分析

## 2.1 当前使用天气预报的用户的基本状况

以中年及以上人群为主，高达90%，其中30~50岁的约占75%，50岁以上的约占15%，男女比例3:2，以男性为主，中年人对天气有更多的需求。在地域分布上，超越五层的用户来自华东和华中，主要集中在人口多，流动大的省份和地区，其中山东、河南、江苏的最多。

## 2.2 当前使用天气预报的用户的行为特点

30岁以后，大多数人生活趋于稳定，关注点慢慢集中到家庭和工作上，他们开始计划工作和生活，关注能够影响计划的任何事件，中年男性更加注重规避风险，对生活的规划性的要求也较高。墨迹天气的用户画像是：男性为主，年龄30至50岁，人口多流动大地区，追求规划生活和生活品质的人群。

## 2.3 当前使用天气预报的用户的需求

用户进入APP的场景有以下几种：早晚：确定天气，第二天着装，雨具，出行方式；外出：查看特定地点未来特定时间段的天气，随时关注；异常天气的关注：对自己的影响（担心亲朋好友，或者自己做好预防准备），或者好奇关注国家大事。

需要查询当地未来一天或几天的天气情况；需要查询异地天气，并有渠道获得当地的实时情况；需要能够了解全国甚至世界各地的极端天气情况；需要一个能够记录，并分享好天气好心情的平台。

用户集中在30岁以上，这群人既要为事业，也要兼顾家庭，时间碎片化，没有整体的时间去关注外界。然而，这类人却更加渴望和外界的沟通，了解外界的信息。因此，有价值的信息也是用户的需求。

## 2.4 分析总结

不同用户查看天气预报可能会有各不相同的原因，但是总结起来其本质都是一样的，即通过天气预报信息来指导自己的行为。由于不同用户的生活习惯不同，所以可能还会有很多其他使用场景。但是这些基础的使用场景基本上可以涵盖了大多数用户的使用情况，从这些场景中可以抽象出用户一些基本需求，进一步得到一款天气预报APP需要具有的一些基本功能。

总而言之，天气预报APP的任务就是让用户用最直观的方式获取最有效的信息，并且在这个过程中有很好的体验，产生愉悦的心情。

# 产品构思

实时天气界面：显示城市当前的实时天气状况，包括城市名称、城市名片、天气情况、温度高低、星期等信息。界面左边有一列城市列表，显示中国各个城市名称，当用户想知道哪个城市的天气情况时，点击这个城市名就可以获取天气状况，考虑到使软件占用尽量小的内存，软件去掉了不必要的花哨，显示城市名称，城市名片，温度和天气情况，同时对未来6天的天气情况简要描述，简要显示星期、天气描述、温度高低等数据，使用户一目了然，使人们及时了解天气变化，对日常生活出行、穿衣做好准备。

# 功能设计

## 4.1产品功能：

1）、定位城市天气显示；  
2）、城市编辑功能（增、删、改、查）以及对应天气显示信息的改变；  
3）、天气信息的Widget窗口显示（城市的编辑功能可以远程的更新Widget窗口信息的显示）  
4）、下拉刷新、天气显示界面左右滑动、城市拖拽等小模块

## 4.2功能实现：

首先背景图片是每天会更新的，是从必应网上获取到的背景图片。  
2）下拉刷新功能。  
3）天气显示信息左右活动切换已选择要显示的城市。  
4）通过点击右上角的编辑按钮进入城市管理功能。  
5）导航组件功能。

## 4.3界面功能具体显示：

1）最上部分是城市名的显示和编辑按钮。  
2）然后是导航原点显示。  
3）其次是当前温度，当天天气和当天最低最高温的显示。  
4）接着显示的是接下来几天的天气的大体介绍，这里显示的数据同样受限于获取到的数据。  
5）再接着是一些生活指数的显示，例如风条件、湿度等等。   
6）最后就是一些生活建议的显示了。

已选择城市信息界面显示设计：



添加城市信息界面显示设计：  


说明；使用ListView去显示省、市、县三个级别的城市，根据选择的城市去网络或者本地加载数据，然后显示。

显示上的设计  
1、背景图片：widget的背景图片同样是网络上下载并且每天会自动更换的，不同的是为了保证用户滑动界面时的流畅性，这里做了图片缩放处理之后再显示。  
2、中间固定文字“当前天气”  
3、下面是一个ListView用来显示简略的已选择城市的信息

功能上的设计  
1、服务listView改变的server进程  
2、contentProvider提供跨进程间的数据通信  
3、图片下载的异步线程和图片缩放实现  
4、异步线程与UI线程通过handler实现通信

5、界面转换设计

有界面转换实现的：  
1、点开app进入到城市天气信息显示界面  
2、点击编辑按钮进入到城市管理界面  
3、城市管理界面中点击添加按钮进入到城市添加界面  
4、城市管理界面中点击返回按钮回到城市天气信息显示界面  
5、添加城市界面中添加完成或者点击返回按钮回到城市天气信息显示界面

# 技术路线与关键点

## 5.1 技术路线

项目总体流程思路：

1）、天气app最重要的是获取城市列表和天气信息，所以首先要解决的问题是在网络上找到合适的api接口，并根据服务商提供的数据转换成自己需要显示的数据。  
2）、有了需要的显示信息之后，我们需要自己去设计怎么显示，怎么让用户去有一个好的体验。我的设计是在使用三个Activity去和用户交互，参照我的项目截图，其中WeatherActivity作为启动活动，用于显示天气信息，提供的是多页带导航栏可左右滑动的效果。ChooseAreaActivity是管理城市的活动，用于添加、删除、改变要显示天气信息的城市列表。AddCountyActivity是用于添加城市的活动。  
3）、实现了这些基本的城市管理和天气显示之后，接下来就是进阶功能了，首先我们实现百度定位功能，根据定位结果加载当前城市天气。  
4）、实现widget功能。

## 5.1关键点

1）、城市信息获取的api

城市信息这里我们直接使用的是Android前辈搭建的一个服务器获取的，数据链接是http://guolin.tech/api/china，访问返回的是JSON数据类型的省份信息（JSON数据类型的解析后面会再详细说明），需要返回城市时只需要在本链接后加上“/对应省份id”即可获取到相应的城市信息，县市信息也是一样的，原链接加上“/对应省份id/对应城市id”即可。  
这里其实也可以从其它天气服务商提供的api接口获取城市信息。

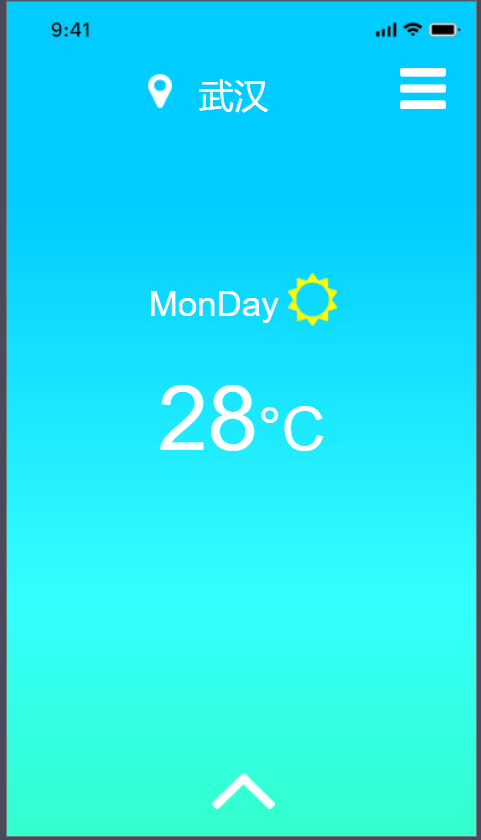
2）、天气信息获取的api

天气信息的获取我使用的是[和风天气](http://console.heweather.com/login?next_action=/my/service)提供的免费的api，和风天气每天有提供4000次免费的基础天气查询，用来做开发测试是足够用的了。而且和风天气api接口返回的JSON数据类型也比较简单，作为Android初学者做项目是比较好的。想使用该接口只需要简单注册一个账号就可以了（对返回数据的处理我后面再详细说明）。老手的话可以在网上搜索别的服务商提供的免费接口，现在网上的免费接口少了很多，不过有还是有的。

3）、定位信息获取的api

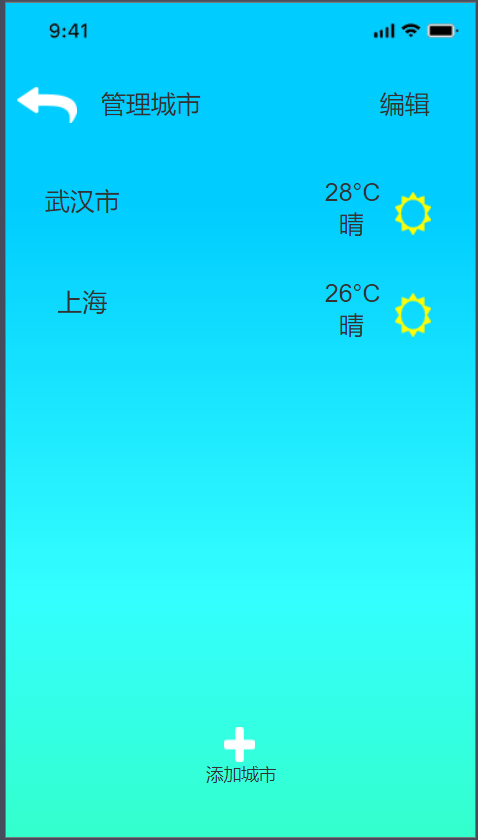
我们这里使用的是百度提供的免费api接口http://lbsyun.baidu.com/apiconsole/key，因为Android原生定位API在国产手机中一般被阉割了，或者国内网络限制的原因，使用Android原生定位API一般是很难获取到定位信息的，跟手机厂商和网络环境都有关系。

# 总体UI设计

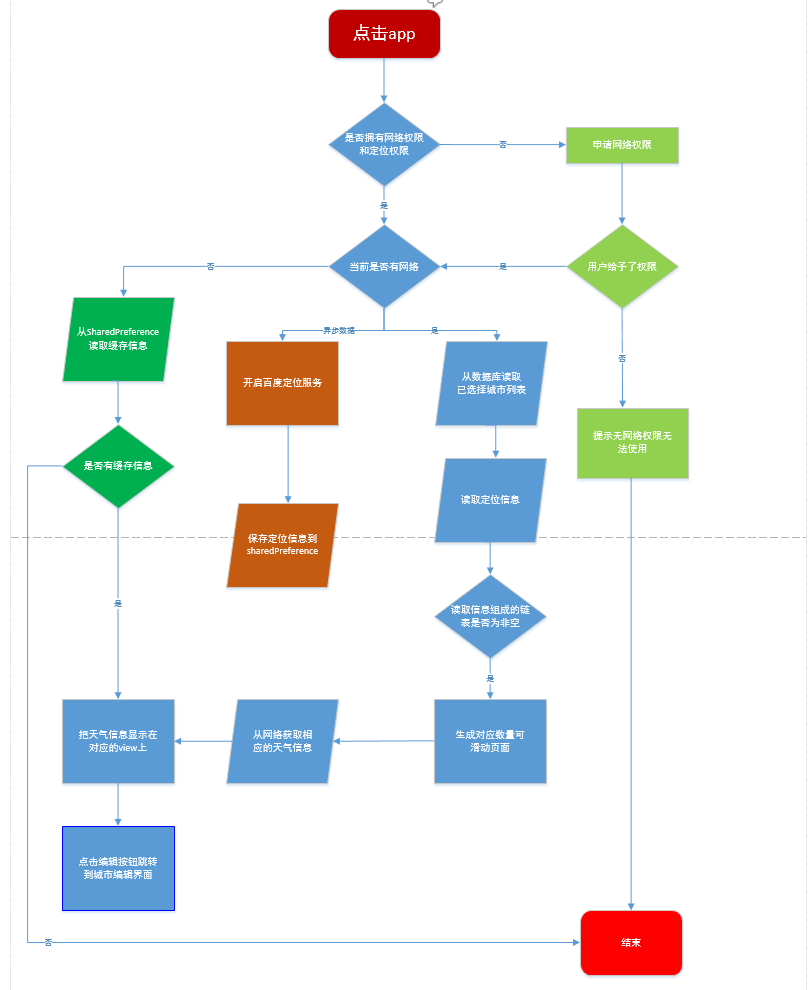
 

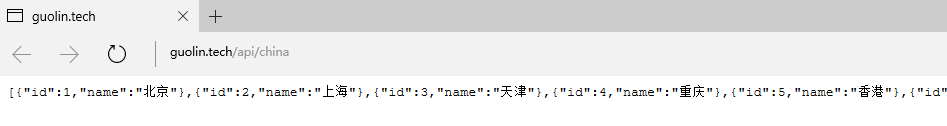
# 功能模块划分

系统总体和局部流程设计（流程图）：



# 具体实现

## 8.1 获取城市信息

数据链接http://guolin.tech/api/china，访问返回的是JSON数据类型的省份信息，需要返回城市时只需要在本链接后加上“/对应省份id”即可获取到相应的城市信息，县市信息也是一样的，原链接加上“/对应省份id/对应城市id”即可。  
  
这里得到的是一个JSON数据，以下是对它的解析代码:

**JSONArray jsonArray** =newJSONArray(**response**);

for(int **i** =0; **i** < **jsonArray**.length();++**i**){

**JSONObject jsonObject** = **jsonArray**.getJSONObject(**i**);

**Province province** =newProvince();

**province**.setProvinceCode(**jsonObject**.getInt("id"));

**province**.setProvinceName(**jsonObject**.getString("name"));

**province**.save();

}

publicstaticvoidsendRequestOkHttpForGet(final **String adress**,final **MyCallBack myCallBack**){

newThread(newRunnable(){

@Override

publicvoidrun(){

**HttpURLConnection connection** = **null**;

try{

**URL url** =newURL(**adress**);

**connection** =(**HttpURLConnection**) **url**.openConnection();

**connection**.setRequestMethod("GET");

**connection**.setConnectTimeout(8000);

**connection**.setReadTimeout(8000);

**InputStream in** = **connection**.getInputStream();

**String response** =convertStreamToString(**in**);

**myCallBack**.onResponse(**response**);

}catch(MalformedURLException **e**){

**e**.printStackTrace();

//失败

**myCallBack**.onFailure(**e**);

}catch(ProtocolException **e**){

**e**.printStackTrace();

**myCallBack**.onFailure(**e**);

}catch(IOException **e**){

**e**.printStackTrace();

**myCallBack**.onFailure(**e**);

}finally{

if(**null** != **connection**){

**connection**.disconnect();

}

}

}

}).start();

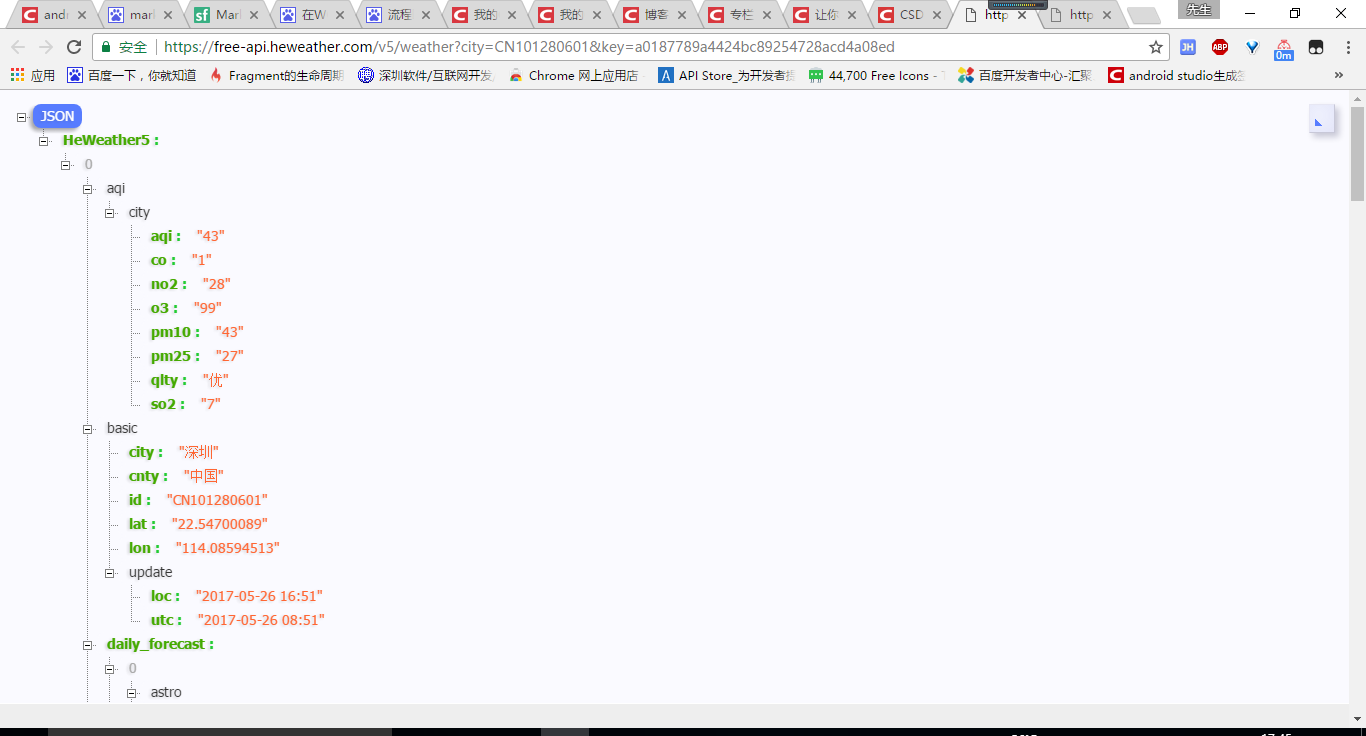
publicinterfaceMyCallBack{

voidonFailure(**IOException e**);

voidonResponse(**String response**)throws **IOException**;

}

## 8.2 获取天气信息

获取天气信息的网络操作是和获取城市信息的操作是一样的，使用上面那个网络异步函数即可，如果觉得不好，也可以使用网络开源项目包装的网络访问接口，比如说OKHttp。不同的是天气信息的JSON数据要比城市信息的JSON数据复杂得多。  
这里是深圳的[天气](https://free-api.heweather.com/v5/weather?city=CN101280601&key=a0187789a4424bc89254728acd4a08ed)显示：  
  
这里是用Chrome的JSON-handle解析之后的结果。这里我们采用GSON方式来解析JSON，方便我们后面对数据的操作。  
GSON方式是把JSON数据解析成相应的对象的一种方式，主要步骤如下：  
1）、根据JSON数据建立不同的类，JSON数据的每一个结点对应一个类，并且根据不同的结点的复杂程度选择是否还要使用内部类。  
2）、@SerializedName(“JSON中的结点名”)需要转换成的节点名；  
使用关键字把一些JSON数据中意义晦涩的名词转换成类中名字可以见名知意的属性。  
3）、JSON数据转换成对象实例

**JSONObject jsonObject** =newJSONObject(**response**);

**JSONArray jsonArray** = **jsonObject**.getJSONArray("HeWeather5");

**String weateherContent** = **jsonArray**.getJSONObject(0).toString();

returnnewGson().fromJson(**weateherContent**, **类名**.class);

最后，我们把得到的对象的数据对应的添加到要显示的活动的布局当中就可以了。

## 8.3 天气信息的显示：

获取背景图片和图片的更新

这里我使用的是必应主页提供的背景图片作为天气信息显示的背景图片http://guolin.tech/api/bing\_pic这个链接是获取必应每日背景图片下载链接的，可以通过该链接获取图片下载地址，然后再去下载。  
由于下载图片是耗时的网络操作，所以我们这里需要使用一个异步线程去下载图片，然后在下载好之后再通知UI线程去加载。  
具体代码：

publicvoidupdateBingPic(){

**String requestBingPic** ="http://guolin.tech/api/bing\_pic";

**OkHttp**.sendRequestOkHttpForGet(**requestBingPic**,newCallback(){

@Override

publicvoidonFailure(**Call call**, **IOException e**){

**e**.printStackTrace();

}

@Override

publicvoidonResponse(**Call call**, **Response response**)throws **IOException** {

**String bingPic** = **response**.body().string();

**SharedPreferences**.**Editor editor** = **PreferenceManager**.getDefaultSharedPreferences(**WeatherActivity**.this).edit();

**editor**.putString("bing\_pic", **bingPic**);

**editor**.apply();

}

});

}

这个是获取图片下载地址的代码,变量bingPic的内容就是下载链接  
if (bingPic != null) {  
Glide.with(WeatherActivity.this).load(bingPic).into(bingPicIv);  
}  
当它不为空时，我们使用Glide去下载并加载图片到天气显示背景。Glide 是 Google 员工的开源项目， Google I/O 上被推荐使用Glide具有获取、解码和展示视频剧照、图片、动画等功能，它还有灵活的API，这些API使开发者能够将Glide应用在几乎任何网络协议栈里。创建Glide的主要目的有两个，一个是实现平滑的图片列表滚动效果，另一个是支持远程图片的获取、大小调整和展示。

ViewPager是android扩展包v4包中的类，主要功能是实现view页面的左右切换。在本项目中，就是一个view包含一个城市的天气信息，然后view又加入到ViewPager中。  
这里说一下ViewPager的使用步骤，ViewPager的实现与ListView有很多相似之处，主要步骤如下：  
1）、创建或设置数据源。  
2）、根据数据源创建或配置好相应的适配器。  
3)、在布局文件中加入ViewPager控件，并在程序给控件设置步骤2中的适配器。  
4)、给控件添加监听器。

下拉刷新功能

本项目中的下拉刷新功能是使用SwipeRefreshLayout控件实现的，实现的步骤很简单：  
1)、在布局文件中实现下拉刷新功能的地方添加android.support.v4.widget.SwipeRefreshLayout控件，这里推荐使用v4包，因为能够支持低版本的Android手机。  
2)、在程序中定义并设置相应属性（样式等等）和监听器。  
3)、设置事件的相应响应和启动下拉刷新和结束下拉刷新。

通过点击右上角的编辑按钮进入城市管理功能

这里的实现就很基础了，简单讲一下步骤：  
1)、在布局文件定义按钮  
2)、在程序中找到按钮并设置监听器  
3)、在响应事件中做进入城市功能活动的逻辑

导航组件功能

本项目的导航栏功能是用Selector实现，Selector主要是用来改变各种view控件的默认背景的。实现步骤如下：  
1)、xml文件定义  
?xml version=“1.0” encoding=“utf-8” ?>  
selector xmlns:android=“http://schemas.android.com/apk/res/android”>  
!-- 默认时的背景图片–>  
item android:drawable="@drawable/pic1" />  
!-- 没有焦点时的背景图片 -->  
item android:state\_window\_focused=“false”  
android:drawable="@drawable/pic1" />  
!-- 非触摸模式下获得焦点并单击时的背景图片 -->  
item android:state\_focused=“true” android:state\_pressed=“true” android:drawable= “@drawable/pic2” />  
!-- 触摸模式下单击时的背景图片–>  
item android:state\_focused=“false” android:state\_pressed=“true” android:drawable="@drawable/pic3" />  
!–选中时的图片背景–>  
item android:state\_selected=“true” android:drawable="@drawable/pic4" />  
!–获得焦点时的图片背景–>  
item android:state\_focused=“true” android:drawable="@drawable/pic5" />  
/selector>

2)、使用  
LinearLayout layout = (LinearLayout)findViewById(R.id.vp\_guide\_layout);  
LinearLayout.LayoutParams mParams = new LinearLayout.LayoutParams(20, 20);  
mParams.setMargins(0, 0, 0, 0);//设置小圆点左右之间的间隔

**guideShapeViewArrayList.clear();**

**layout.removeAllViews();**

**ImageView imageView = new ImageView(this);**

**imageView.setLayoutParams(mParams);**

**imageView.setImageResource(R.drawable.guide\_shape\_select);**

小时天气小时超出屏幕宽度时的当前页面左右滑动（RecycleListView）

RecycleListView是Android官方出品的一个可以代替甚至超越ListView的东西。RecycleListView的实现比不优化的ListView麻烦一些，但是功能上比ListView要更强大，因为他的显示不仅可以竖屏，还可以横屏。  
实现步骤：  
1)、准备数据源  
2)、根据数据源设置适配器  
static class ViewHolder extends RecyclerView.ViewHolder {  
TextView hourlyTimeTV;  
ImageView hourlyWeatherImageV;  
TextView hourlyTemperatureTV;

**public ViewHolder(View view){**

**super(view);**

**hourlyTimeTV = (TextView) view.findViewById(R.id.hourly\_time\_tv);**

**hourlyWeatherImageV = (ImageView) view.findViewById(R.id.hourly\_weather\_iv);**

**hourlyTemperatureTV = (TextView) view.findViewById(R.id.hourly\_temperature\_tv);**

**}**

**}**

**@Override**

**public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {**

**View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.hourly\_forecast\_item, parent, false);**

**ViewHolder holder = new ViewHolder(view);**

**return holder;**

**}**

**@Override**

**public void onBindViewHolder(ViewHolder holder, int position) {**

**HourlyWeather hourlyWeather = hourlyWeatherList.get(position);**

**holder.hourlyTimeTV.setText(hourlyWeather.hourlyTime + "时");**

**holder.hourlyWeatherImageV.setImageBitmap(hourlyWeather.hourlyImageBit);**

**holder.hourlyTemperatureTV.setText(hourlyWeather.hourlyTemperature + "º");**

**}**

**@Override**

**public int getItemCount() {**

**return hourlyWeatherList.size();**

**}**

要实现RecyclerView.Adapter主要是要实现三个函数  
onCreateViewHolder()  
onBindViewHolder()  
getItemCount()

3)、在布局文件定义RecycleView控件，并在代码中为控件设置以上适配器。  
4)、选择是否要设置监听器。

布局圆角功能

布局圆角主要是为了让布局中的控件看起来美观一些。  
实现很简单  
1)、在drawable中定义xml文件

**<?xml version="1.0" encoding="utf-8"?>**

**<shape xmlns:android="http://schemas.android.com/apk/res/android">**

**<solid android:color="#8000" />**

**<corners**

**<android:topLeftRadius="10dp"**

**<android:topRightRadius="10dp"**

**<android:bottomRightRadius="10dp"**

**<android:bottomLeftRadius="10dp"/>**

**</shape>**

2)、在需要引入圆角的布局文件中引入本配置作为背景

**android:background="@drawable/corners\_bg"**

## 8.4 城市信息的显示：

1)、活动切换按钮，这里就不再重复说明了。  
2)、添加城市。  
3)、可拖拽的ListView（DragListView）的城市信息实现这里实现的主要功能有：删除城市、城市排序切换。  
实现步骤：  
1)、准备数据源  
2)、设置适配器

publicclassCountiesAdapterextendsBaseAdapter{

private **Context context**;

//适配器的数据源 selectedCityList

private **List**<SelectedCounty> **items**;

publicCountiesAdapter(**Context context**,**List**<SelectedCounty> **selectedCityList**){

this.**context** = **context**;

this.**items** = **selectedCityList**;

**LogUtil**.d(**TAG**,"CountiesAdapter: selectedCityList size:"+ **selectedCityList**.size());

**LogUtil**.d(**TAG**,"CountiesAdapter: selectedCityList items size:"+ **items**.size());

}

@Override

publicintgetCount(){

return **items**.size();

}

@Override

public **Object** getItem(int **arg0**){

return **items**.get(**arg0**);

}

@Override

publiclonggetItemId(int **arg0**){

return **arg0**;

}

publicvoidremove(int **arg0**){//删除指定位置的item

**items**.remove(**arg0**);

this.notifyDataSetChanged();

}

publicvoidinsert(**SelectedCounty item**,int **arg0**){

**items**.add(**arg0**, **item**);

this.notifyDataSetChanged();

}

publicvoidchange(**List**<SelectedCounty> **selectedCityList**){

**items** = **selectedCityList**;

this.notifyDataSetChanged();

}

@Override

public **View** getView(int **position**, **View convertView**, **ViewGroup parent**){

**SelectedCounty item** =(**SelectedCounty**)getItem(**position**);

**ViewHolder viewHolder**;

if(**null** == **convertView**){

**viewHolder** =newViewHolder();

**convertView** = **LayoutInflater**.from(**context**).inflate(**R**.**layout**.**drag\_listview\_item**, **null**);

**viewHolder**.**dragMoveIv** =(**ImageView**) **convertView**.findViewById(**R**.**id**.**drag\_move\_iv**);

**viewHolder**.**dragCountyNameTv** =(**TextView**) **convertView**.findViewById(**R**.**id**.**drag\_county\_name\_tv**);

**viewHolder**.**drag\_click\_remove** =(**ImageView**) **convertView**.findViewById(**R**.**id**.**drag\_click\_remove**);

**moveImageViewList**.add(**viewHolder**.**dragMoveIv**);

**deleteImageViewList**.add(**viewHolder**.**drag\_click\_remove**);

**convertView**.setTag(**viewHolder**);

}else{

**viewHolder** =(**ViewHolder**) **convertView**.getTag();

}

//是否点击了edit按钮，

if(**isEditClick**){

if(!**WeatherActivity**.**isLocationCountyRemove** && **null** != **WeatherActivity**.**locationCountyWeatherId** &&0== **position** ){

**viewHolder**.**drag\_click\_remove**.setVisibility(**View**.**GONE**);

}else{

**viewHolder**.**dragMoveIv**.setVisibility(**View**.**VISIBLE**);

**viewHolder**.**drag\_click\_remove**.setVisibility(**View**.**VISIBLE**);

}

}else{

**viewHolder**.**dragMoveIv**.setVisibility(**View**.**GONE**);

**viewHolder**.**drag\_click\_remove**.setVisibility(**View**.**GONE**);

}

**viewHolder**.**dragCountyNameTv**.setText(**item**.getCountyName());

return **convertView**;

}

classViewHolder{

**ImageView dragMoveIv**;

**TextView dragCountyNameTv**;

**ImageView drag\_click\_remove**;

}

}

别的地方和ListView是一样的，不同的是多了一个remove和insert函数

publicvoidremove(int **arg0**){//删除指定位置的item

**items**.remove(**arg0**);

this.notifyDataSetChanged();//不要忘记更改适配器对象的数据源

}

publicvoidinsert(**SelectedCounty item**,int **arg0**){

**items**.add(**arg0**, **item**);

this.notifyDataSetChanged();

}

3)、控件绑定适配器  
这里也有区别：首先要定义两个函数：

//监听器在手机拖动停下的时候触发

private **DragSortListView**.**DropListener onDrop** =

newDragSortListView.DropListener(){

@Override

publicvoiddrop(int **from**,int **to**){//from to 分别表示 被拖动控件原位置 和目标位置

//如果定位城市存在，则去除定位城市的操作

if(!**WeatherActivity**.**isLocationCountyRemove** && **null** != **WeatherActivity**.**locationCountyWeatherId**){

if(0== **from** ||0== **to**){

return;

}

}

if(**from** != **to**){

**SelectedCounty item** =(**SelectedCounty**)**countiesAdapter**.getItem(**from**);//得到listview的适配器

**countiesAdapter**.remove(**from**);//在适配器中”原位置“的数据。

**countiesAdapter**.insert(**item**, **to**);//在目标位置中插入被拖动的控件。

**isSwapCounty** =true;

}

}

};

//删除监听器，点击左边差号就触发。删除item操作。

private **DragSortListView**.**RemoveListener onRemove** =

newDragSortListView.RemoveListener(){

@Override

publicvoidremove(int **which**){

**delCountyId**.add(**selectedCityList**.get(**which**).getId());

**delCountyIndex**.add(**which**);

**countiesAdapter**.remove(**which**);

**Log**.d(**TAG**,"onClick: remove position:"+ **which**);

}

};

然后绑定适配器时这两个函数也一起绑定

**hourlyRecycler** =(**RecyclerView**) **currentView**.findViewById(**R**.**id**.**hourly\_recycler**);

**layoutManager** =newLinearLayoutManager(**currentView**.getContext());

**layoutManager**.setOrientation(**LinearLayoutManager**.**HORIZONTAL**);

**hourlyRecycler**.setLayoutManager(**layoutManager**);

**hourlyWeatherAdapter** =newHourlyWeatherAdapter(**hourlyWeatherList**);

**hourlyRecycler**.setAdapter(**hourlyWeatherAdapter**);

# 课程总结

在本次Android天气预报软件开发过程中，我们通过学校的图书馆资料和网上查询，了解了Android的搭建、Intent、Activity、Service等知识。Android环境搭建的基本工作完成后，对界面的控件、布局和美观方面进行设计，对数据获取和解析做了分析，通过模拟器显示城市名片，实时天气情况和未来5天天气情况，这个软件还可以进一步发展，比如“短信预报”、“新闻模块”等功能，如果这些功能都能实现将会对用户有更多的帮助，使他们的生活更加愉悦和方便。Android 智能手机应用程序的开发涉及了它的整个体系结构，是一项非常复杂的工程。

# 参考文献

[1] 靳岩，姚尚朗. Google Android 开发入门与实战.人民邮电出版社[J]

[2] Michael J. Young.轻松搞定XML林嘉胜译[M] 2001年01 月08日

[3] 王向辉，张国印，沈洁. Android应用程序开发[M].清华大学出版社，2010年3月

[4] 李宁。Android/iPhone 开发完全讲义[M].悲剧:中国水利水电出版社，2010. (6)

[5]李刚.疯狂Android讲义[M].北京:电子工业出版社，2011. (6)

[6] 靳岩，姚尚朗. Google Android开发入门与实战[M].北京:中国邮电出版社, 2009. (7)