**实 验 报 告**

**课程名称 移动应用系统**

**实验项目 通过网络发送和获取数据**

**实验仪器 计算机一台、Android手机一部**

**系 别 计算机学院**

**专 业 计算机科学与技术**

**班级/学号**  计科1606 / 2016010311

**学生姓名**  耿瑞

**实验日期**  2018年12月21日

**成 绩** \_\_\_\_\_\_\_\_\_\_

**指导教师** \_\_\_\_ **郝保水**\_\_\_\_ \_\_

1 实验目的

（1）掌握Android中线程概念，掌握Handler用法；

（2）掌握Android网络访问等相关知识和技术;

（3）了解应用中前端和后端相关概念和技术等；

2 实验要求

（1）课前预习实验内容，并查找相关资料。

（2）按照实验步骤完成各个相关内容。

（3）撰写实验报告。

A） 实验报告格式必须符合学校要求（例如必须采用学校规定的实验封面）；

B） 写出实验详细步骤，包括主要采用的技术方案、相关分析和核心代码。注意：不要简单地近包括截屏和代码，完整代码可以作为附录放在实验报告结尾；

C） 总结实验中遇到的问题、分析和解决方法。

D） 写出心得体会与收获等。

3 实验步骤

主要分为几个大步骤进行实现。

（1）需求分析，调研单词类应用具有哪些功能，确定界面。

（2）系统设计，对系统进行模块划分，确定技术方案。

（3）编码，编写代码实现各项功能。

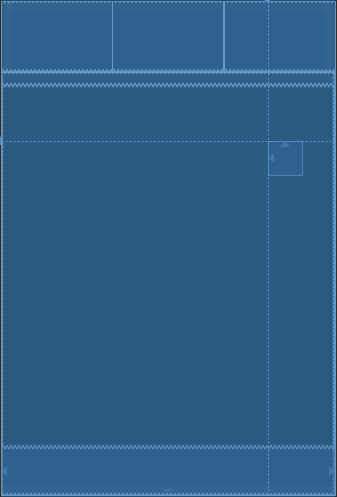
（4）测试。

3.1 需求分析

（1）主要功能

* 获取实时天气
* 获取未来天气预报
* 获取空气质量指数信息
* 遍历全国省市县
* 一键切换到定位城市

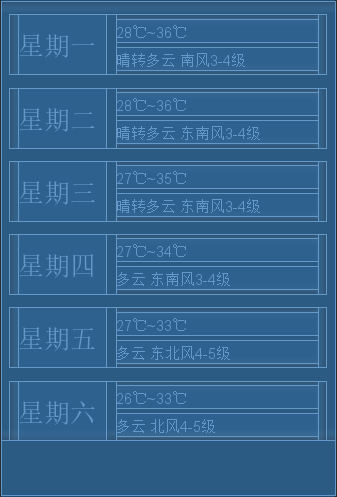
（2）界面设计



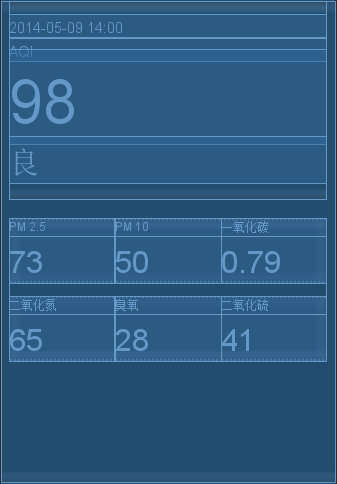
主界面



实时天气界面



天气预报页面



空气质量指数页面

3.2 系统设计

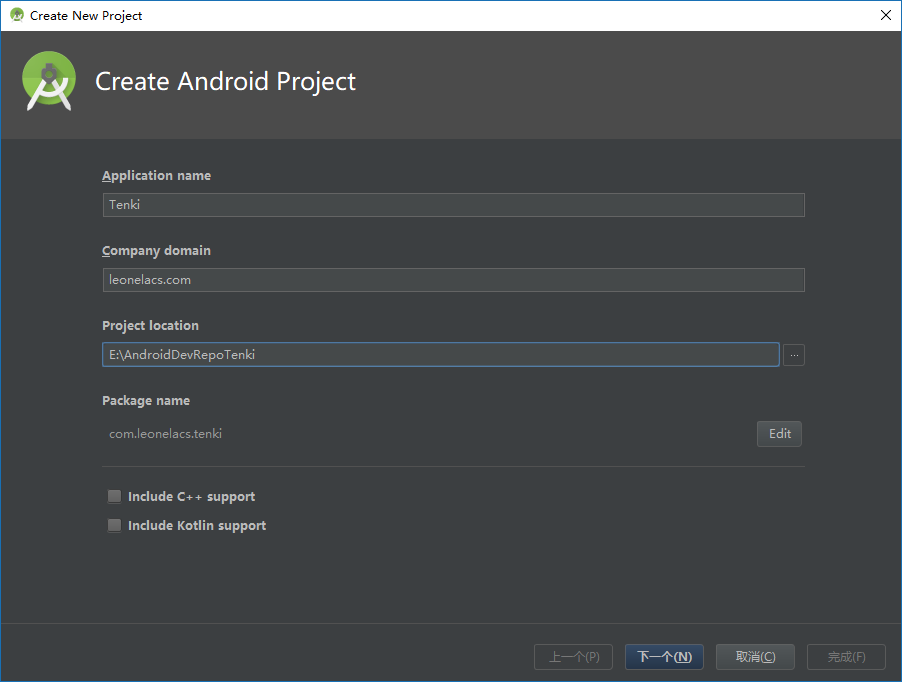
天气类应用包括以下模块：

* 省市县下拉选单模块
* 实时天气模块
* 天气预报模块
* 空气质量指数模块
* 省市县信息解析（本地）模块
* 天气信息解析（在线）模块

3.3 项目实施

主要步骤：

（1）创建工程



在Android Studio中创建Android项目。

应用名称：Tenki

域名：leonelacs.com

包名：com.leonelacs.tenki

最低SDK版本API 24: Android 7.0

（2）编写各界面布局、菜单等资源文件

主界面，使用LinearLayout实现，包含：

* 用于选择省市县的三个Spinner：**ProvinceSelection**、**CitySelection**、**DistrictSelection**；
* 一个用于显示其它页面的FrameView；
* 底部导航栏BottomNavigationView **navigation**
* 用于实现下拉刷新的SwipeRefreshLayout **SwipeRefresh**
* 用于获取地理位置信息刷新天气的FloatingActionButton：**fab**

ProvinceSelection响应点击事件，当点击该控件时，弹出下拉列表供选择。当点击选择列表中项目后，查询该选项省份的所有城市，添加进入CitySelection。

CitySelection响应点击事件，当点击该控件时，弹出下拉列表供选择。当点击选择列表中项目后，查询该选项城市的所有区县，添加进入DistrictSelection。

DistrictSelection响应点击事件，当点击该控件时，弹出下拉列表供选择。当点击选择列表中项目后，查询API获取该地区的天气信息，解析后于各个页面中显示。

Navigation响应点击事件，当点击底部导航栏按钮时显示所选页面

页面响应下拉刷新事件，当下拉页面时可以获取新的天气信息刷新

fab响应点击事件，当点击时，获取设备地理位置的经纬度值，并以此查询API获得该位置的城市名及天气信息，解析后于各个页面中显示。

主界面的背景颜色会根据实时气温的数值范围而改变。

实时天气界面，使用LinearLayout实现，包含：

* 一个用于显示更新时间的TextView：**TenkiUpdateTime**
* 一个用于显示实时天气图标（Emoji文字）的TextView：**MokuzenTenkiEmoji**
* 一个用于显示实时风向、风力和湿度的TextView：**MokuzenWindAndHumidity**
* 一个用于显示今日温度的TextView：**TodayTemperature**
* 一个用于显示今日天气的TextView：**TodayTenki**
* 一个用于显示今日着装建议的TextView：**TodayClothes**

天气图标由Emoji Unicode文字显示，预先保存于Emojis类中

天气编号 天气名称 天气图标（Emoji）

00 晴 ☀

01 多云 ⛅

02 阴 ☁

03 阵雨 🌦

04 雷阵雨 🌦⚡

05 雷阵雨伴有冰雹 🌦⚡⚪

06 雨夹雪 ❄

07 小雨 🌧

08 中雨 🌧🌧

09 大雨 🌧🌧🌧

10 暴雨 🌧❗

11 大暴雨 🌧❗❗

12 特大暴雨 🌧🈲

13 阵雪 ❄

14 小雪 ❄

15 中雪 ❄❄

16 大雪 ❄❄❄

17 暴雪 ❄❗

18 雾 🌫

19 冻雨 〰

20 沙尘暴 💲

21 小雨-中雨 🌧🌧

22 中雨-大雨 🌧🌧🌧

23 大雨-暴雨 🌧❗

24 暴雨-大暴雨 🌧❗❗

25 大暴雨-特大暴雨 🌧🈲

26 小雪-中雪 ❄❄

27 中雪-大雪 ❄❄❄

28 大雪-暴雪 ❄❗

29 浮尘 💲➖

30 扬沙 💲 🔼

31 强沙尘暴 💲❗

53 霾 ♾



天气预报界面，使用LinearLayout实现，包含六组用于显示未来六天天气情况的LinearLayout，其中分别包含：

* 一个用于显示当天星期的TextView：**Youbi*n***
* 一个用于显示当天温度的TextView：**Youbi*n*Temp**
* 一个用于显示当天天气的TextView：**Youbi*n*Tenki**



天气预报界面，使用LinearLayout实现，包含：

* 一个用于显示更新时间的TextView：**AQIUpdateTime**
* 一个用于显示AQI数值的TextView：**MokuzenAQIValue**
* 一个用于显示AQI指数文字的TextView：**MokuzenAQIWord**
* 一个用于显示污染物PM 2.5浓度的TextView：**PM25**
* 一个用于显示污染物PM 10浓度的TextView：**PM10**
* 一个用于显示污染物一氧化碳浓度的TextView：**CO**
* 一个用于显示污染物二氧化氮浓度的TextView：**NO2**
* 一个用于显示污染物臭氧浓度的TextView：**O3**
* 一个用于显示污染物二氧化硫浓度的TextView：**SO2**

AQIUpdateTime、MokuzenAQIValue、MokuzenAQIWord所在的LinearLayout的背景颜色会根据当前AQI数值所在的范围发生改变。



获取全国省市县信息：

全国省市县信息由查询聚合数据全国天气预报-支持城市列表API（<http://v.juhe.cn/weather/citys?key=>申请的key）获得JSON数据。由于此项数据基本不会发生变动，故将查询结果保存于本地，而不是每次启动应用时查询，以节省API调用次数。

根据城市名或经纬度查询天气信息：

天气信息由查询聚合数据的以下API获得获得JSON数据：

* 全国天气预报-根据城市名查询天气API

<http://v.juhe.cn/weather/index?format=2&cityname=>城市名&key=申请的key

* 全国天气预报-根据GPS坐标查询天气API

<http://v.juhe.cn/weather/geo?format=2&key=>申请的key&lon=经度值&lat=纬度值

* 空气质量-城市空气PM2.5指数API

<http://web.juhe.cn:8080/environment/air/pm?city=>城市名&key=申请的key

查询API的操作执行于子线程中

new Thread(new Runnable() {  
 @Override  
 public void run() {  
 String districtSelected = DistrictSpinner.getSelectedItem().toString().trim();  
 String bigCity = CitySpinner.getSelectedItem().toString().trim();  
 newInfo = questAPI.refreshByCityName(districtSelected, bigCity);  
 handler.post(runnable);  
 }  
}).start();

查询全国省市县信息和天气信息所获得的JSON字符串，均由Okhttp访问API URL地址获取，并使用阿里巴巴的开源JSON解析库fastjson解析，分别得到地区（class Chiiki）对象的列表和天气信息（TenkiInfo）对象。

ChiikiLocal.java包含以下功能

解析全国省市县信息JSON

JSONObject obj = JSONObject.*parseObject*(localJSON);  
String resultCode = obj.getString("resultcode");  
if (resultCode != null && resultCode.equals("200")) {  
 String result = obj.getString("result");  
 JSONArray arr = JSONArray.*parseArray*(result);  
 for (Object o: arr) {  
 JSONObject jsonChiiki = JSONObject.*parseObject*(o.toString());  
 //以下省略  
 }  
}

由省份名获得城市列表 public List<String> getShiByShou(String shou)

由城市名获得区县列表 public List<String> getKuByShi(String shi)

由城市名查询所在省份 public String getShouByShi (String shi)

TenkiAPIO.java包含以下功能

解析天气预报信息JSON获得天气信息对象

if (receive != null) {  
 JSONObject obj = JSONObject.*parseObject*(receive);  
 String resultCode = obj.getString("resultcode");  
 if (resultCode != null && resultCode.equals("200")) {  
 String result = obj.getString("result");  
 JSONObject objResult = JSONObject.*parseObject*(result);  
  
 String strSk = objResult.getString("sk");  
 JSONObject objSk = JSONObject.*parseObject*(strSk);  
 String temp = objSk.getString("temp");  
 //以下省略  
 JSONObject objWeatherId = JSONObject.*parseObject*(strWeatherId);  
 //以下省略  
  
 String strFuture = objResult.getString("future");  
 JSONArray arrFuture = JSONArray.*parseArray*(strFuture);  
 JSONObject obj1 = arrFuture.getJSONObject(0);  
 JSONObject obj2 = arrFuture.getJSONObject(1);  
 JSONObject obj3 = arrFuture.getJSONObject(2);  
 JSONObject obj4 = arrFuture.getJSONObject(3);  
 JSONObject obj5 = arrFuture.getJSONObject(4);  
 JSONObject obj6 = arrFuture.getJSONObject(5);  
 //以下省略  
 }  
}

解析空气质量信息JSON

if (receive\_a != null) {  
 JSONObject obj = JSONObject.*parseObject*(receive\_a);  
 String resultCode = obj.getString("resultcode");  
 if (resultCode != null && resultCode.equals("200")) {  
 String result = obj.getString("result");  
 JSONArray arrResult = JSONArray.*parseArray*(result);  
 JSONObject objInResult = arrResult.getJSONObject(0);  
 //以下省略  
 }

获取经纬度值：

Location location;  
 try {  
 location = locationManager.getLastKnownLocation(locationProvider);  
 final String lon = String.*valueOf*(location.getLongitude());  
 final String lat = String.*valueOf*(location.getLatitude());  
 new Thread(new Runnable() {  
 @Override  
 public void run() {  
 newInfo = questAPI.refreshByCoordinate(lon, lat);  
 handler.post(runnable);  
 }  
 }).start();  
 }  
 catch (SecurityException e) {  
 Snackbar.*make*(view, "未获得定位权限", Snackbar.*LENGTH\_LONG*).setAction("Action", null).show();  
 }  
 catch (NullPointerException e) {}  
 }  
});

下拉刷新

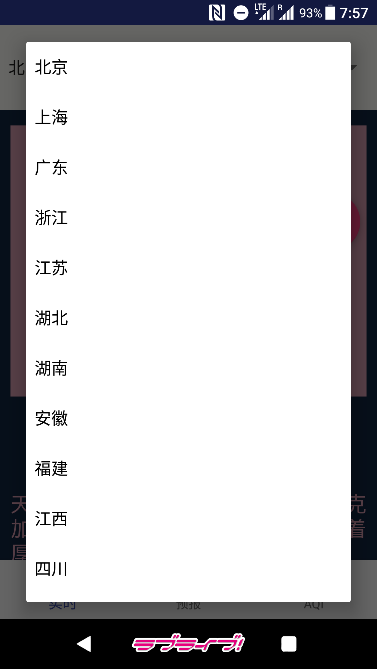
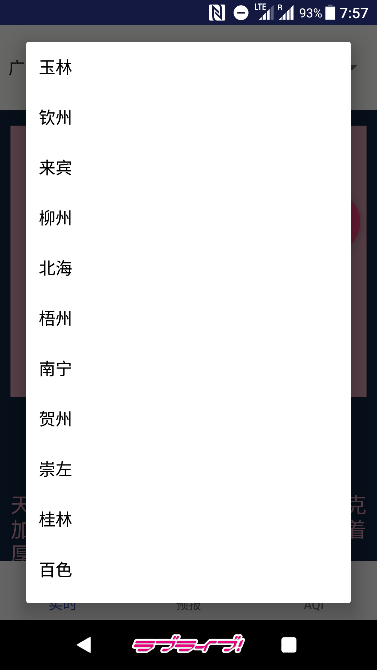
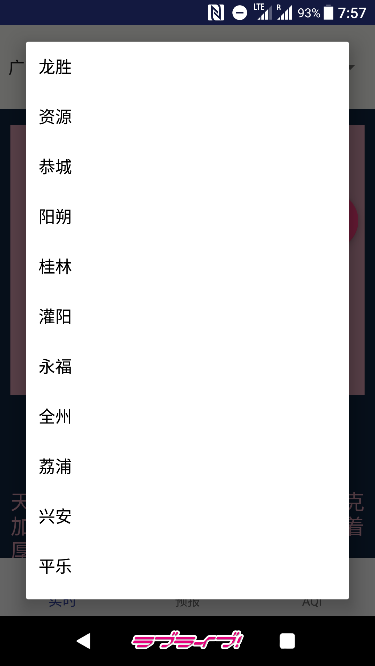
swipeRefresh.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener() {  
 @Override  
 public void onRefresh() {  
 new Thread(new Runnable() {  
 @Override  
 public void run() {  
 String districtSelected = DistrictSpinner.getSelectedItem().toString().trim();  
 String bigCity = CitySpinner.getSelectedItem().toString().trim();  
 newInfo = questAPI.refreshByCityName(districtSelected, bigCity);  
 handler.post(runnable);  
 }  
 }).start();  
 swipeRefresh.setRefreshing(false);  
 }  
});

3.4 测试

（1）根据城市名查询天气信息

点击ProvinceSelection选择省份，然后点击CitySelection选择城市，再点击DistrictSelection选择区县，应用依据区县字段查询天气信息，解析后于各个页面中显示。

例：依次点击三个Spinner，选择广西-桂林-桂林，应用显示桂林的天气信息

点击底部导航栏按钮切换显示的页面

（2）根据经纬度查询天气信息

点击定位按钮，应用获取设备所在位置的经纬度，并以此查询天气信息，解析后于各个页面中显示。

例：点击定位按钮，Spinner显示北京-北京-朝阳，应用显示朝阳区的天气信息



4 实验中遇到的问题、分析和解决方法

（1）显示天气图标

本来计划使用图片作为天气图标，但是发现使用Emoji Unicode字符，将其置于TextView中显示更加方便。

（2）点击选择其它省份后，区县列表不会自动改变

在ProvinceSelection的点击事件中增加区县的查询操作。

5 心得体会与收获

通过这次实验，我完成了一个天气预报Android应用的开发。掌握了Android中的线程概念、Handler用法、Android网络访问等相关知识和技术；了解了应用中前端和后端相关概念和技术等。

在这次实验中，我通过自己选择API，选择了聚合数据的全国天气预报API和全国空气质量API。使用Spinner控件来显示和选择省份、城市和区县的值，通过设定Spinner的选择事件触发查询天气与更新界面的操作。除此之外，我还实现了通过系统获取设备所在位置的经纬度并以此查询天气的功能。

6 附录

代码仓库：<https://github.com/leonelacs/Tenki>

程序完整代码：

**com.leonelacs.tenki**

Chiiki.java

package com.leonelacs.tenki;  
  
public class Chiiki {  
 String id;  
 String shou;  
 String shi;  
 String ku;  
  
 public Chiiki() {}  
  
 public Chiiki(String id, String shou, String shi, String ku) {  
 this.id = id;  
 this.shou = shou;  
 this.shi = shi;  
 this.ku = ku;  
 }  
  
 public String getId() {  
 return id;  
 }  
  
 public void setId(String id) {  
 this.id = id;  
 }  
  
 public String getShou() {  
 return shou;  
 }  
  
 public void setShou(String shou) {  
 this.shou = shou;  
 }  
  
 public String getShi() {  
 return shi;  
 }  
  
 public void setShi(String shi) {  
 this.shi = shi;  
 }  
  
 public String getKu() {  
 return ku;  
 }  
  
 public void setKu(String ku) {  
 this.ku = ku;  
 }  
}

TenkiEmoji.java

package com.leonelacs.tenki;  
  
public class TenkiEmoji {  
 public String code;  
 public String name;  
 public String emoji;  
  
 public TenkiEmoji(String code, String name, String emoji) {  
 this.code = code;  
 this.name = name;  
 this.emoji = emoji;  
 }  
}

TenkiInfo.java

package com.leonelacs.tenki;  
  
public class TenkiInfo {  
 public String tenkiUpdateTime = "-";  
 public String mokuzenTenkiEmoji = "🈚";  
 public String mokuzenTemperature = "🌡-℃　";  
 public String mokuzenWindAndHumidity = "🌬 -　|　💧 -%";  
 public String todayTemperature = "-";  
 public String todayTenki = "-";  
 public String todayClothes = "-";  
  
 public String youbi1Youbi = "-";  
 public String youbi1Temp = "-";  
 public String youbi1Tenki = "-";  
 public String youbi2Youbi = "-";  
 public String youbi2Temp = "-";  
 public String youbi2Tenki = "-";  
 public String youbi3Youbi = "-";  
 public String youbi3Temp = "-";  
 public String youbi3Tenki = "-";  
 public String youbi4Youbi = "-";  
 public String youbi4Temp = "-";  
 public String youbi4Tenki = "-";  
 public String youbi5Youbi = "-";  
 public String youbi5Temp = "-";  
 public String youbi5Tenki = "-";  
 public String youbi6Youbi = "-";  
 public String youbi6Temp = "-";  
 public String youbi6Tenki = "-";  
  
 public String aqiUpdateTime = "";  
 public String mokuzenAQIValue = "";  
 public String mokuzenAQIWord = "";  
 public String pm25Value = "";  
 public String pm10Value = "";  
 public String coValue = "";  
 public String no2Value = "";  
 public String o3Value = "";  
 public String so2Value = "";  
  
 public int imaOndo;  
  
 public TenkiInfo() {  
 }  
}

Emojis.java

package com.leonelacs.tenki;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class Emojis {  
 List<TenkiEmoji> emojis = new ArrayList<>();  
 public Emojis() {  
 emojis.add(new TenkiEmoji("00", "晴", "☀"));  
 emojis.add(new TenkiEmoji("01", "多云", "⛅"));  
 emojis.add(new TenkiEmoji("02", "阴", "☁"));  
 emojis.add(new TenkiEmoji("03", "阵雨", "🌦"));  
 emojis.add(new TenkiEmoji("04", "雷阵雨", "🌦⚡"));  
 emojis.add(new TenkiEmoji("05", "雷阵雨伴有冰雹", "🌦⚡⚪"));  
 emojis.add(new TenkiEmoji("06", "雨夹雪", "🌧❄"));  
 emojis.add(new TenkiEmoji("07", "小雨", "🌧"));  
 emojis.add(new TenkiEmoji("08", "中雨", "🌧🌧"));  
 emojis.add(new TenkiEmoji("09", "大雨", "🌧🌧🌧"));  
 emojis.add(new TenkiEmoji("10", "暴雨", "🌧❗"));  
 emojis.add(new TenkiEmoji("11", "大暴雨", "🌧❗❗"));  
 emojis.add(new TenkiEmoji("12", "特大暴雨", "🌧🈲"));  
 emojis.add(new TenkiEmoji("13", "阵雪", "❄"));  
 emojis.add(new TenkiEmoji("14", "小雪", "❄"));  
 emojis.add(new TenkiEmoji("15", "中雪", "❄❄"));  
 emojis.add(new TenkiEmoji("16", "大雪", "❄❄❄"));  
 emojis.add(new TenkiEmoji("17", "暴雪", "❄❗"));  
 emojis.add(new TenkiEmoji("18", "雾", "🌫"));  
 emojis.add(new TenkiEmoji("19", "冻雨", "〰"));  
 emojis.add(new TenkiEmoji("20", "沙尘暴", "💲"));  
 emojis.add(new TenkiEmoji("21", "小雨-中雨", "🌧🌧"));  
 emojis.add(new TenkiEmoji("22", "中雨-大雨", "🌧🌧🌧"));  
 emojis.add(new TenkiEmoji("23", "大雨-暴雨", "🌧❗"));  
 emojis.add(new TenkiEmoji("24", "暴雨-大暴雨", "🌧❗❗"));  
 emojis.add(new TenkiEmoji("25", "大暴雨-特大暴雨", "🌧🈲"));  
 emojis.add(new TenkiEmoji("26", "小雪-中雪", "❄❄"));  
 emojis.add(new TenkiEmoji("27", "中雪-大雪", "❄❄❄"));  
 emojis.add(new TenkiEmoji("28", "大雪-暴雪", "❄❗"));  
 emojis.add(new TenkiEmoji("29", "浮尘", "💲➖"));  
 emojis.add(new TenkiEmoji("30", "扬沙", "💲🔼"));  
 emojis.add(new TenkiEmoji("31", "强沙尘暴", "💲❗"));  
 emojis.add(new TenkiEmoji("53", "霾", "♾"));  
 }  
  
 String getEmojiByCode(String code) {  
 for (TenkiEmoji e: emojis) {  
 if (e.code.equals(code)) {  
 return e.emoji;  
 }  
 }  
 return "";  
 }  
}

ChiikiLocal.java

package com.leonelacs.tenki;  
  
import com.alibaba.fastjson.JSONArray;  
import com.alibaba.fastjson.JSONObject;  
  
import java.util.ArrayList;  
import java.util.HashSet;  
import java.util.List;  
  
public class ChiikiLocal {  
  
 StringBuilder sb = new StringBuilder();  
 public String localJSON;  
 public List<Chiiki> chiikiIchiran = new ArrayList<>();  
  
 public ChiikiLocal() {

localJSON = sb.toString();  
 JSONObject obj = JSONObject.*parseObject*(localJSON);  
 String resultCode = obj.getString("resultcode");  
 if (resultCode != null && resultCode.equals("200")) {  
 String result = obj.getString("result");  
 JSONArray arr = JSONArray.*parseArray*(result);  
 for (Object o: arr) {  
 JSONObject jsonChiiki = JSONObject.*parseObject*(o.toString());  
 String id = jsonChiiki.getString("id");  
 String province = jsonChiiki.getString("province");  
 String city = jsonChiiki.getString("city");  
 String district = jsonChiiki.getString("district");  
 Chiiki tmp = new Chiiki(id, province, city, district);  
 chiikiIchiran.add(tmp);  
 }  
 }  
 }  
  
 public List<String> getShiByShou(String shou) {  
 List<String> shiList = new ArrayList<>();  
 for (Chiiki c: chiikiIchiran) {  
 if (c.shou.equals(shou)) {  
 shiList.add(" " + c.shi + " ");  
 }  
 }  
 HashSet<String> h = new HashSet<>(shiList);  
 shiList.clear();  
 shiList.addAll(h);  
 return shiList;  
 }  
  
 public List<String> getKuByShi(String shi) {  
 List<String> kuList = new ArrayList<>();  
 for (Chiiki c: chiikiIchiran) {  
 if (c.shi.equals(shi)) {  
 kuList.add(" " + c.ku + " ");  
 }  
 }  
 HashSet<String> h = new HashSet<>(kuList);  
 kuList.clear();  
 kuList.addAll(h);  
 return kuList;  
 }  
  
 public String getShouByShi (String shi) {  
 for (Chiiki c: chiikiIchiran) {  
 if (c.shi.equals(shi)) {  
 return c.shou;  
 }  
 }  
 return "";  
 }  
}

TenkiAPIO.java

package com.leonelacs.tenki;  
  
import com.alibaba.fastjson.JSONArray;  
import com.alibaba.fastjson.JSONObject;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import okhttp3.OkHttpClient;  
import okhttp3.Request;  
import okhttp3.Response;  
  
public class TenkiAPIO {  
 private static final String *URL\_BY\_CITYNAME* = "http://v.juhe.cn/weather/index?key=6690cb86309398c0f2e54670d4725605&format=2&cityname=";  
 private static final String *URL\_BY\_COORDINATE* = "http://v.juhe.cn/weather/geo?key=6690cb86309398c0f2e54670d4725605&format=2&";  
 private static final String *URL\_AQI* = "http://web.juhe.cn:8080/environment/air/pm?key=c779b3e9680227e637793e104bd33c26&city=";  
 public List<Chiiki> chiikiIchiran = new ArrayList<>();  
  
 public TenkiAPIO() {  
 }  
  
 public TenkiInfo refreshByCityName(String cityName, String bigCity) {  
 TenkiInfo tenki = new TenkiInfo();  
 String url = *URL\_BY\_CITYNAME* + cityName;  
 String receive = null;  
 String receive\_a = null;  
  
 OkHttpClient clientTenki = new OkHttpClient();  
 Request requestTenki = new Request.Builder().url(url).build();  
 try {  
 Response responseTenki = clientTenki.newCall(requestTenki).execute();  
 receive = responseTenki.body().string();  
 }  
 catch (Exception e) {  
 e.printStackTrace();  
 }  
  
// String receive = NIO.get(url);  
 if (receive != null) {  
 JSONObject obj = JSONObject.*parseObject*(receive);  
 String resultCode = obj.getString("resultcode");  
 if (resultCode != null && resultCode.equals("200")) {  
 String result = obj.getString("result");  
 JSONObject objResult = JSONObject.*parseObject*(result);  
  
 String strSk = objResult.getString("sk");  
 JSONObject objSk = JSONObject.*parseObject*(strSk);  
 String temp = objSk.getString("temp");  
 tenki.imaOndo = Integer.*parseInt*(temp);  
 tenki.mokuzenTemperature = "🌡"+temp+"℃　";  
 String wind\_direction = objSk.getString("wind\_direction");  
 String wind\_strength = objSk.getString("wind\_strength");  
 String humidity = objSk.getString("humidity");  
 String wind\_and\_humidity = "🌬 " + wind\_direction + " " + wind\_strength + "　|　💧 " + humidity;  
 tenki.mokuzenWindAndHumidity = wind\_and\_humidity;  
 String time = objSk.getString("time");  
 tenki.tenkiUpdateTime = "更新时间　" + time;  
 String strToday = objResult.getString("today");  
 JSONObject objToday = JSONObject.*parseObject*(strToday);  
 String temperature = objToday.getString("temperature");  
 tenki.todayTemperature = "今日温度　" + temperature;  
 String weather = objToday.getString("weather");  
 String strWeatherId = objToday.getString("weather\_id");  
 JSONObject objWeatherId = JSONObject.*parseObject*(strWeatherId);  
 String fa = objWeatherId.getString("fa");  
 String fb = objWeatherId.getString("fb");  
 String wind = objToday.getString("wind");  
 String dressing\_index = objToday.getString("dressing\_index");  
 String today\_tenki = weather + "　" + wind + "　" + dressing\_index;  
 tenki.todayTenki = today\_tenki;  
 Emojis emojis = new Emojis();  
 if (fa.equals(fb)) {  
 tenki.mokuzenTenkiEmoji = emojis.getEmojiByCode(fa);  
 }  
 else {  
 tenki.mokuzenTenkiEmoji = emojis.getEmojiByCode(fa) + "/" + emojis.getEmojiByCode(fb);  
 }  
 String dressing\_advice = objToday.getString("dressing\_advice");  
 tenki.todayClothes = dressing\_advice;  
  
 String strFuture = objResult.getString("future");  
 JSONArray arrFuture = JSONArray.*parseArray*(strFuture);  
 JSONObject obj1 = arrFuture.getJSONObject(0);  
 JSONObject obj2 = arrFuture.getJSONObject(1);  
 JSONObject obj3 = arrFuture.getJSONObject(2);  
 JSONObject obj4 = arrFuture.getJSONObject(3);  
 JSONObject obj5 = arrFuture.getJSONObject(4);  
 JSONObject obj6 = arrFuture.getJSONObject(5);  
 String y1Temp = obj1.getString("temperature");  
 String y2Temp = obj2.getString("temperature");  
 String y3Temp = obj3.getString("temperature");  
 String y4Temp = obj4.getString("temperature");  
 String y5Temp = obj5.getString("temperature");  
 String y6Temp = obj6.getString("temperature");  
 tenki.youbi1Temp = y1Temp;  
 tenki.youbi2Temp = y2Temp;  
 tenki.youbi3Temp = y3Temp;  
 tenki.youbi4Temp = y4Temp;  
 tenki.youbi5Temp = y5Temp;  
 tenki.youbi6Temp = y6Temp;  
 String y1Weather = obj1.getString("weather");  
 String y2Weather = obj2.getString("weather");  
 String y3Weather = obj3.getString("weather");  
 String y4Weather = obj4.getString("weather");  
 String y5Weather = obj5.getString("weather");  
 String y6Weather = obj6.getString("weather");  
 String y1Wind = obj1.getString("wind");  
 String y2Wind = obj2.getString("wind");  
 String y3Wind = obj3.getString("wind");  
 String y4Wind = obj4.getString("wind");  
 String y5Wind = obj5.getString("wind");  
 String y6Wind = obj6.getString("wind");  
 tenki.youbi1Tenki = y1Weather + " " + y1Wind;  
 tenki.youbi2Tenki = y2Weather + " " + y2Wind;  
 tenki.youbi3Tenki = y3Weather + " " + y3Wind;  
 tenki.youbi4Tenki = y4Weather + " " + y4Wind;  
 tenki.youbi5Tenki = y5Weather + " " + y5Wind;  
 tenki.youbi6Tenki = y6Weather + " " + y6Wind;  
 String y1Week = obj1.getString("week");  
 String y2Week = obj2.getString("week");  
 String y3Week = obj3.getString("week");  
 String y4Week = obj4.getString("week");  
 String y5Week = obj5.getString("week");  
 String y6Week = obj6.getString("week");  
 tenki.youbi1Youbi = y1Week;  
 tenki.youbi2Youbi = y2Week;  
 tenki.youbi3Youbi = y3Week;  
 tenki.youbi4Youbi = y4Week;  
 tenki.youbi5Youbi = y5Week;  
 tenki.youbi6Youbi = y6Week;  
 }  
 }  
  
 String url\_a = *URL\_AQI* + bigCity;  
  
 OkHttpClient clientAQI = new OkHttpClient();  
 Request requestAQI = new Request.Builder().url(url\_a).build();  
 try {  
 Response responseAQI = clientAQI.newCall(requestAQI).execute();  
 receive\_a = responseAQI.body().string();  
 }  
 catch (Exception e) {}  
  
// String receive\_a = NIO.get(url\_a);  
 if (receive\_a != null) {  
 JSONObject obj = JSONObject.*parseObject*(receive\_a);  
 String resultCode = obj.getString("resultcode");  
 if (resultCode != null && resultCode.equals("200")) {  
 String result = obj.getString("result");  
 JSONArray arrResult = JSONArray.*parseArray*(result);  
 JSONObject objInResult = arrResult.getJSONObject(0);  
// JSONObject objInResult = JSONObject.parseObject(result);  
 String aqi = objInResult.getString("AQI");  
 tenki.mokuzenAQIValue = aqi;  
 String quality = objInResult.getString("quality");  
 tenki.mokuzenAQIWord = quality;  
 String PM2\_5 = objInResult.getString("PM2.5");  
 tenki.pm25Value = PM2\_5;  
 String PM10 = objInResult.getString("PM10");  
 tenki.pm10Value = PM10;  
 String CO = objInResult.getString("CO");  
 tenki.coValue = CO;  
 String NO2 = objInResult.getString("NO2");  
 tenki.no2Value = NO2;  
 String O3 = objInResult.getString("O3");  
 tenki.o3Value = O3;  
 String SO2 = objInResult.getString("SO2");  
 tenki.so2Value = SO2;  
 String time = objInResult.getString("time");  
 tenki.aqiUpdateTime = time;  
 }  
 else {  
 tenki.mokuzenAQIValue = "-";  
 tenki.mokuzenAQIWord = "-";  
 tenki.pm25Value = "-";  
 tenki.pm10Value = "-";  
 tenki.coValue = "-";  
 tenki.no2Value = "-";  
 tenki.o3Value = "-";  
 tenki.so2Value = "-";  
 tenki.aqiUpdateTime = "-";  
 }  
 }  
 return tenki;  
 }  
  
 public TenkiInfo refreshByCoordinate(String lon, String lat) {  
 TenkiInfo tenki = new TenkiInfo();  
 String url = *URL\_BY\_COORDINATE* + "lon=" + lon + "&lat=" + lat;  
 String receive = null;  
 String receive\_a = null;  
  
 OkHttpClient clientTenki = new OkHttpClient();  
 Request requestTenki = new Request.Builder().url(url).build();  
 try {  
 Response responseTenki = clientTenki.newCall(requestTenki).execute();  
 receive = responseTenki.body().string();  
 }  
 catch (Exception e) {}  
 String city = "notfound";  
 if (receive != null) {  
 JSONObject obj = JSONObject.*parseObject*(receive);  
 String resultCode = obj.getString("resultcode");  
 if (resultCode != null && resultCode.equals("200")) {  
 String result = obj.getString("result");  
 JSONObject objResult = JSONObject.*parseObject*(result);  
  
 String strSk = objResult.getString("sk");  
 JSONObject objSk = JSONObject.*parseObject*(strSk);  
 String temp = objSk.getString("temp");  
 tenki.mokuzenTemperature = "🌡"+temp+"℃　";  
 String wind\_direction = objSk.getString("wind\_direction");  
 String wind\_strength = objSk.getString("wind\_strength");  
 String humidity = objSk.getString("humidity");  
 String wind\_and\_humidity = "🌬 " + wind\_direction + " " + wind\_strength + "　|　💧 " + humidity;  
 tenki.mokuzenWindAndHumidity = wind\_and\_humidity;  
 String time = objSk.getString("time");  
 tenki.tenkiUpdateTime = "更新时间　" + time;  
 String strToday = objResult.getString("today");  
 JSONObject objToday = JSONObject.*parseObject*(strToday);  
 String temperature = objToday.getString("temperature");  
 tenki.todayTemperature = "今日温度　" + temperature;  
 String weather = objToday.getString("weather");  
 String strWeatherId = objToday.getString("weather\_id");  
 JSONObject objWeatherId = JSONObject.*parseObject*(strWeatherId);  
 String fa = objWeatherId.getString("fa");  
 String fb = objWeatherId.getString("fb");  
 String wind = objToday.getString("wind");  
 String dressing\_index = objToday.getString("dressing\_index");  
 String today\_tenki = weather + "　" + wind + "　" + dressing\_index;  
 tenki.todayTenki = today\_tenki;  
 Emojis emojis = new Emojis();  
 if (fa.equals(fb)) {  
 tenki.mokuzenTenkiEmoji = emojis.getEmojiByCode(fa);  
 }  
 else {  
 tenki.mokuzenTenkiEmoji = emojis.getEmojiByCode(fa) + "　/　" + emojis.getEmojiByCode(fb);  
 }  
 String dressing\_advice = objToday.getString("dressing\_advice");  
 tenki.todayClothes = dressing\_advice;  
 city = objToday.getString("city");  
  
 String strFuture = objResult.getString("future");  
 JSONArray arrFuture = JSONArray.*parseArray*(strFuture);  
 JSONObject obj1 = arrFuture.getJSONObject(0);  
 JSONObject obj2 = arrFuture.getJSONObject(1);  
 JSONObject obj3 = arrFuture.getJSONObject(2);  
 JSONObject obj4 = arrFuture.getJSONObject(3);  
 JSONObject obj5 = arrFuture.getJSONObject(4);  
 JSONObject obj6 = arrFuture.getJSONObject(5);  
 String y1Temp = obj1.getString("temperature");  
 String y2Temp = obj2.getString("temperature");  
 String y3Temp = obj3.getString("temperature");  
 String y4Temp = obj4.getString("temperature");  
 String y5Temp = obj5.getString("temperature");  
 String y6Temp = obj6.getString("temperature");  
 tenki.youbi1Temp = y1Temp;  
 tenki.youbi2Temp = y2Temp;  
 tenki.youbi3Temp = y3Temp;  
 tenki.youbi4Temp = y4Temp;  
 tenki.youbi5Temp = y5Temp;  
 tenki.youbi6Temp = y6Temp;  
 String y1Weather = obj1.getString("weather");  
 String y2Weather = obj2.getString("weather");  
 String y3Weather = obj3.getString("weather");  
 String y4Weather = obj4.getString("weather");  
 String y5Weather = obj5.getString("weather");  
 String y6Weather = obj6.getString("weather");  
 String y1Wind = obj1.getString("wind");  
 String y2Wind = obj2.getString("wind");  
 String y3Wind = obj3.getString("wind");  
 String y4Wind = obj4.getString("wind");  
 String y5Wind = obj5.getString("wind");  
 String y6Wind = obj6.getString("wind");  
 tenki.youbi1Tenki = y1Weather + " " + y1Wind;  
 tenki.youbi2Tenki = y2Weather + " " + y2Wind;  
 tenki.youbi3Tenki = y3Weather + " " + y3Wind;  
 tenki.youbi4Tenki = y4Weather + " " + y4Wind;  
 tenki.youbi5Tenki = y5Weather + " " + y5Wind;  
 tenki.youbi6Tenki = y6Weather + " " + y6Wind;  
 String y1Week = obj1.getString("week");  
 String y2Week = obj2.getString("week");  
 String y3Week = obj3.getString("week");  
 String y4Week = obj4.getString("week");  
 String y5Week = obj5.getString("week");  
 String y6Week = obj6.getString("week");  
 tenki.youbi1Youbi = y1Week;  
 tenki.youbi2Youbi = y2Week;  
 tenki.youbi3Youbi = y3Week;  
 tenki.youbi4Youbi = y4Week;  
 tenki.youbi5Youbi = y5Week;  
 tenki.youbi6Youbi = y6Week;  
 }  
 }  
  
 String url\_a = *URL\_AQI* + city;  
  
 OkHttpClient clientAQI = new OkHttpClient();  
 Request requestAQI = new Request.Builder().url(url\_a).build();  
 try {  
 Response responseAQI = clientAQI.newCall(requestAQI).execute();  
 receive\_a = responseAQI.body().string();  
 }  
 catch (Exception e) {}  
 if (receive\_a != null) {  
 JSONObject obj = JSONObject.*parseObject*(receive);  
 String resultCode = obj.getString("resultcode");  
 if (resultCode != null && resultCode.equals("200")) {  
 String result = obj.getString("result");  
 JSONArray arrResult = JSONArray.*parseArray*(result);  
 JSONObject objInResult = arrResult.getJSONObject(0);  
 String aqi = objInResult.getString("aqi");  
 tenki.mokuzenAQIValue = aqi;  
 String quality = objInResult.getString("quality");  
 tenki.mokuzenAQIWord = quality;  
 String PM2\_5 = objInResult.getString("PM2.5");  
 tenki.pm25Value = PM2\_5;  
 String PM10 = objInResult.getString("PM10");  
 tenki.pm10Value = PM10;  
 String CO = objInResult.getString("CO");  
 tenki.coValue = CO;  
 String NO2 = objInResult.getString("NO2");  
 tenki.no2Value = NO2;  
 String O3 = objInResult.getString("O3");  
 tenki.o3Value = O3;  
 String SO2 = objInResult.getString("SO2");  
 tenki.so2Value = SO2;  
 String time = objInResult.getString("time");  
 tenki.aqiUpdateTime = time;  
 }  
 else {  
 tenki.mokuzenAQIValue = "-";  
 tenki.mokuzenAQIWord = "-";  
 tenki.pm25Value = "-";  
 tenki.pm10Value = "-";  
 tenki.coValue = "-";  
 tenki.no2Value = "-";  
 tenki.o3Value = "-";  
 tenki.so2Value = "-";  
 tenki.aqiUpdateTime = "-";  
 }  
 }  
 return tenki;  
 }  
}

MainActivity.java

package com.leonelacs.tenki;  
  
import android.app.Activity;  
import android.content.Context;  
import android.content.Intent;  
import android.graphics.Color;  
import android.location.Location;  
import android.location.LocationManager;  
import android.os.Bundle;  
import android.os.Handler;  
import android.provider.Settings;  
import android.support.annotation.NonNull;  
import android.support.design.widget.BottomNavigationView;  
import android.support.design.widget.FloatingActionButton;  
import android.support.design.widget.Snackbar;  
import android.support.v4.widget.SwipeRefreshLayout;  
import android.support.v7.app.AppCompatActivity;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.LinearLayout;  
import android.widget.Spinner;  
import android.widget.TextView;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class MainActivity extends AppCompatActivity {  
  
// List<String> cityItems = new ArrayList<>();  
// List<String> districtItems = new ArrayList<>();  
// Spinner ProvinceSpinner;  
// Spinner CitySpinner;  
// Spinner DistrictSpinner;  
  
 LinearLayout mainMiddle;  
 LinearLayout aqiSquare;  
  
 ChiikiLocal chiikiLocal = new ChiikiLocal();  
// ArrayAdapter<String> CityAdapter; // = new ArrayAdapter<String>(this, android.R.layout.simple\_spinner\_item, cityItems);  
// ArrayAdapter<String> DistrictAdapter; // = new ArrayAdapter<String>(this, android.R.layout.simple\_spinner\_item, districtItems);  
 TenkiAPIO questAPI = new TenkiAPIO();  
  
 TextView tenkiUpdateTime;  
 TextView mokuzenTenkiEmoji;  
 TextView mokuzenTemperature;  
 TextView mokuzenWindAndHumidity;  
 TextView todayTemperature;  
 TextView todayTenki;  
 TextView todayClothes;  
  
 TextView youbi\_1;  
 TextView youbi\_2;  
 TextView youbi\_3;  
 TextView youbi\_4;  
 TextView youbi\_5;  
 TextView youbi\_6;  
  
 TextView youbi\_1Temp;  
 TextView youbi\_2Temp;  
 TextView youbi\_3Temp;  
 TextView youbi\_4Temp;  
 TextView youbi\_5Temp;  
 TextView youbi\_6Temp;  
  
 TextView youbi\_1Tenki;  
 TextView youbi\_2Tenki;  
 TextView youbi\_3Tenki;  
 TextView youbi\_4Tenki;  
 TextView youbi\_5Tenki;  
 TextView youbi\_6Tenki;  
  
 TextView aqiUpdateTime;  
 TextView mokuzenAQIValue;  
 TextView mokuzenAQIWord;  
 TextView pm\_25;  
 TextView pm\_10;  
 TextView coGas;  
 TextView no2Gas;  
 TextView o3Gas;  
 TextView so2Gas;  
  
 SwipeRefreshLayout swipeRefresh;  
  
 TenkiInfo newInfo;  
 private Handler handler;  
  
 LocationManager locationManager; // = (LocationManager)this.getSystemService(Context.LOCATION\_SERVICE);  
 List<String> providers; // = locationManager.getProviders(true);  
 String locationProvider; // = null  
  
 private BottomNavigationView.OnNavigationItemSelectedListener mOnNavigationItemSelectedListener  
 = new BottomNavigationView.OnNavigationItemSelectedListener() {  
  
 @Override  
 public boolean onNavigationItemSelected(@NonNull MenuItem item) {  
 switch (item.getItemId()) {  
 case R.id.*navigation\_home*:  
 //Left  
 findViewById(R.id.*MokuzenPage*).setVisibility(View.*VISIBLE*);  
 findViewById(R.id.*YohouPage*).setVisibility(View.*INVISIBLE*);  
 findViewById(R.id.*OsenPage*).setVisibility(View.*INVISIBLE*);  
 return true;  
 case R.id.*navigation\_dashboard*:  
 //Middle  
 findViewById(R.id.*MokuzenPage*).setVisibility(View.*INVISIBLE*);  
 findViewById(R.id.*YohouPage*).setVisibility(View.*VISIBLE*);  
 findViewById(R.id.*OsenPage*).setVisibility(View.*INVISIBLE*);  
 return true;  
 case R.id.*navigation\_notifications*:  
 //Right  
 findViewById(R.id.*MokuzenPage*).setVisibility(View.*INVISIBLE*);  
 findViewById(R.id.*YohouPage*).setVisibility(View.*INVISIBLE*);  
 findViewById(R.id.*OsenPage*).setVisibility(View.*VISIBLE*);  
 return true;  
 }  
 return false;  
 }  
 };  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 BottomNavigationView navigation = (BottomNavigationView) findViewById(R.id.*navigation*);  
 navigation.setOnNavigationItemSelectedListener(mOnNavigationItemSelectedListener);  
  
 final List<String> cityItems = new ArrayList<>();  
 final List<String> districtItems = new ArrayList<>();  
  
 mainMiddle = findViewById(R.id.*MainMiddle*);  
 aqiSquare = findViewById(R.id.*AQISquare*);  
  
 final Spinner ProvinceSpinner = (Spinner)findViewById(R.id.*ProvinceSelection*);  
 final Spinner CitySpinner = (Spinner)findViewById(R.id.*CitySelection*);  
 final Spinner DistrictSpinner = (Spinner)findViewById(R.id.*DistrictSelection*);  
  
 final ArrayAdapter<String> CityAdapter = new ArrayAdapter<String>(this, android.R.layout.*simple\_spinner\_item*, cityItems);  
 final ArrayAdapter<String> DistrictAdapter = new ArrayAdapter<String>(this, android.R.layout.*simple\_spinner\_item*, districtItems);  
 CityAdapter.setDropDownViewResource(android.R.layout.*simple\_spinner\_dropdown\_item*);  
 DistrictAdapter.setDropDownViewResource(android.R.layout.*simple\_spinner\_dropdown\_item*);  
  
 CitySpinner.setAdapter(CityAdapter);  
 DistrictSpinner.setAdapter(DistrictAdapter);  
  
 handler = new Handler();  
  
 tenkiUpdateTime = findViewById(R.id.*TenkiUpdateTime*);  
 mokuzenTenkiEmoji = findViewById(R.id.*MokuzenTenkiEmoji*);  
 mokuzenTemperature = findViewById(R.id.*MokuzenTemperature*);  
 mokuzenWindAndHumidity = findViewById(R.id.*MokuzenWindAndHumidity*);  
 todayTemperature = findViewById(R.id.*TodayTemperature*);  
 todayTenki = findViewById(R.id.*TodayTenki*);  
 todayClothes = findViewById(R.id.*TodayClothes*);  
  
 youbi\_1 = findViewById(R.id.*Youbi1*);  
 youbi\_2 = findViewById(R.id.*Youbi2*);  
 youbi\_3 = findViewById(R.id.*Youbi3*);  
 youbi\_4 = findViewById(R.id.*Youbi4*);  
 youbi\_5 = findViewById(R.id.*Youbi5*);  
 youbi\_6 = findViewById(R.id.*Youbi6*);  
  
 youbi\_1Temp = findViewById(R.id.*Youbi1Temp*);  
 youbi\_2Temp = findViewById(R.id.*Youbi2Temp*);  
 youbi\_3Temp = findViewById(R.id.*Youbi3Temp*);  
 youbi\_4Temp = findViewById(R.id.*Youbi4Temp*);  
 youbi\_5Temp = findViewById(R.id.*Youbi5Temp*);  
 youbi\_6Temp = findViewById(R.id.*Youbi6Temp*);  
  
 youbi\_1Tenki = findViewById(R.id.*Youbi1Tenki*);  
 youbi\_2Tenki = findViewById(R.id.*Youbi2Tenki*);  
 youbi\_3Tenki = findViewById(R.id.*Youbi3Tenki*);  
 youbi\_4Tenki = findViewById(R.id.*Youbi4Tenki*);  
 youbi\_5Tenki = findViewById(R.id.*Youbi5Tenki*);  
 youbi\_6Tenki = findViewById(R.id.*Youbi6Tenki*);  
  
 aqiUpdateTime = findViewById(R.id.*AQIUpdateTime*);  
 mokuzenAQIValue = findViewById(R.id.*MokuzenAQIValue*);  
 mokuzenAQIWord = findViewById(R.id.*MokuzenAQIWord*);  
 pm\_25 = findViewById(R.id.*PM25*);  
 pm\_10 = findViewById(R.id.*PM10*);  
 coGas = findViewById(R.id.*CO*);  
 no2Gas = findViewById(R.id.*NO2*);  
 o3Gas = findViewById(R.id.*O3*);  
 so2Gas = findViewById(R.id.*SO2*);  
  
 locationManager = (LocationManager)this.getSystemService(Context.*LOCATION\_SERVICE*);  
 providers = locationManager.getProviders(true);  
 if (providers.contains(LocationManager.*GPS\_PROVIDER*)) {  
 //如果是GPS  
 locationProvider = LocationManager.*GPS\_PROVIDER*;  
 } else if (providers.contains(LocationManager.*NETWORK\_PROVIDER*)) {  
 //如果是Network  
 locationProvider = LocationManager.*NETWORK\_PROVIDER*;  
 } else {  
 Intent i = new Intent();  
 i.setAction(Settings.*ACTION\_LOCATION\_SOURCE\_SETTINGS*);  
 this.startActivity(i);  
 }  
  
 ProvinceSpinner.setOnItemSelectedListener(new Spinner.OnItemSelectedListener() {  
 @Override  
 public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {  
 String provinceSelected = ProvinceSpinner.getSelectedItem().toString().trim();  
 //cityItems = chiikiLocal.getShiByShou(provinceSelected);  
 List<String> items = chiikiLocal.getShiByShou(provinceSelected);  
 cityItems.clear();  
 cityItems.addAll(items);  
 CityAdapter.notifyDataSetChanged();  
 try {  
 String citySelected = CitySpinner.getSelectedItem().toString().trim();  
 items = chiikiLocal.getKuByShi(citySelected);  
 districtItems.clear();  
 districtItems.addAll(items);  
 DistrictAdapter.notifyDataSetChanged();  
 }  
 catch (Exception e) {  
 districtItems.clear();  
 DistrictAdapter.notifyDataSetChanged();  
 }  
  
 }  
  
 @Override  
 public void onNothingSelected(AdapterView<?> adapterView) {  
  
 }  
 });  
  
 CitySpinner.setOnItemSelectedListener(new Spinner.OnItemSelectedListener() {  
 @Override  
 public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {  
 String citySelected = CitySpinner.getSelectedItem().toString().trim();  
 //districtItems = chiikiLocal.getKuByShi(citySelected);  
 List<String> items = chiikiLocal.getKuByShi(citySelected);  
 districtItems.clear();  
 districtItems.addAll(items);  
 DistrictAdapter.notifyDataSetChanged();  
 }  
  
 @Override  
 public void onNothingSelected(AdapterView<?> adapterView) {  
 districtItems.clear();  
 DistrictAdapter.notifyDataSetChanged();  
 }  
 });  
  
 DistrictSpinner.setOnItemSelectedListener(new Spinner.OnItemSelectedListener() {  
 @Override  
 public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {  
// String districtSelected = DistrictSpinner.getSelectedItem().toString().trim();  
// String bigCity = CitySpinner.getSelectedItem().toString().trim();  
  
// TenkiInfo newInfo = questAPI.refreshByCityName(districtSelected, bigCity);  
// tenkiInfoChanged(newInfo);  
 new Thread(new Runnable() {  
 @Override  
 public void run() {  
 String districtSelected = DistrictSpinner.getSelectedItem().toString().trim();  
 String bigCity = CitySpinner.getSelectedItem().toString().trim();  
 newInfo = questAPI.refreshByCityName(districtSelected, bigCity);  
 handler.post(runnable);  
 }  
 }).start();  
 }  
  
 @Override  
 public void onNothingSelected(AdapterView<?> adapterView) {  
  
 }  
 });  
  
 FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.*fab*);  
 fab.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 Location location;  
 try {  
 location = locationManager.getLastKnownLocation(locationProvider);  
 final String lon = String.*valueOf*(location.getLongitude());  
 final String lat = String.*valueOf*(location.getLatitude());  
 new Thread(new Runnable() {  
 @Override  
 public void run() {  
 newInfo = questAPI.refreshByCoordinate(lon, lat);  
 handler.post(runnable);  
 }  
 }).start();  
 }  
 catch (SecurityException e) {  
 Snackbar.*make*(view, "未获得定位权限", Snackbar.*LENGTH\_LONG*).setAction("Action", null).show();  
 }  
 catch (NullPointerException e) {}  
 }  
 });  
  
 swipeRefresh = findViewById(R.id.*SwipeRefresh*);  
 swipeRefresh.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener() {  
 @Override  
 public void onRefresh() {  
 new Thread(new Runnable() {  
 @Override  
 public void run() {  
 String districtSelected = DistrictSpinner.getSelectedItem().toString().trim();  
 String bigCity = CitySpinner.getSelectedItem().toString().trim();  
 newInfo = questAPI.refreshByCityName(districtSelected, bigCity);  
 handler.post(runnable);  
 }  
 }).start();  
 swipeRefresh.setRefreshing(false);  
 }  
 });  
  
 }  
  
 void tenkiInfoChanged(TenkiInfo info) {  
 tenkiUpdateTime.setText(info.tenkiUpdateTime);  
 mokuzenTenkiEmoji.setText(info.mokuzenTenkiEmoji);  
 mokuzenTemperature.setText(info.mokuzenTemperature);  
 mokuzenWindAndHumidity.setText(info.mokuzenWindAndHumidity);  
 todayTemperature.setText(info.todayTemperature);  
 todayTenki.setText(info.todayTenki);  
 todayClothes.setText(info.todayClothes);  
  
 youbi\_1.setText(info.youbi1Youbi);  
 youbi\_2.setText(info.youbi2Youbi);  
 youbi\_3.setText(info.youbi3Youbi);  
 youbi\_4.setText(info.youbi4Youbi);  
 youbi\_5.setText(info.youbi5Youbi);  
 youbi\_6.setText(info.youbi6Youbi);  
  
 youbi\_1Temp.setText(info.youbi1Temp);  
 youbi\_2Temp.setText(info.youbi2Temp);  
 youbi\_3Temp.setText(info.youbi3Temp);  
 youbi\_4Temp.setText(info.youbi4Temp);  
 youbi\_5Temp.setText(info.youbi5Temp);  
 youbi\_6Temp.setText(info.youbi6Temp);  
  
 youbi\_1Tenki.setText(info.youbi1Tenki);  
 youbi\_2Tenki.setText(info.youbi2Tenki);  
 youbi\_3Tenki.setText(info.youbi3Tenki);  
 youbi\_4Tenki.setText(info.youbi4Tenki);  
 youbi\_5Tenki.setText(info.youbi5Tenki);  
 youbi\_6Tenki.setText(info.youbi6Tenki);  
  
 aqiUpdateTime.setText(info.aqiUpdateTime);  
 mokuzenAQIValue.setText(info.mokuzenAQIValue);  
 mokuzenAQIWord.setText(info.mokuzenAQIWord);  
 pm\_25.setText(info.pm25Value);  
 pm\_10.setText(info.pm10Value);  
 coGas.setText(info.coValue);  
 no2Gas.setText(info.no2Value);  
 o3Gas.setText(info.o3Value);  
 so2Gas.setText(info.so2Value);  
  
 int ima = info.imaOndo;  
 if (ima >= 28) { mainMiddle.setBackgroundColor(Color.*parseColor*("#F05E1C")); }  
 else if (ima <= 0) { mainMiddle.setBackgroundColor(Color.*parseColor*("#0F2540")); }  
 else { mainMiddle.setBackgroundColor(Color.*parseColor*("#26453D")); }  
  
 String aqi = info.mokuzenAQIWord;  
 if (aqi.equals("优")) { aqiSquare.setBackgroundColor(Color.*parseColor*("#1B813E")); }  
 else if (aqi.equals("良")) { aqiSquare.setBackgroundColor(Color.*parseColor*("#F7D94C")); }  
 else if (aqi.equals("轻度污染")) { aqiSquare.setBackgroundColor(Color.*parseColor*("#F05E1C")); }  
 else if (aqi.equals("中度污染")) { aqiSquare.setBackgroundColor(Color.*parseColor*("#C73E3A")); }  
 else if (aqi.equals("重度污染")) { aqiSquare.setBackgroundColor(Color.*parseColor*("#4A225D")); }  
 else if (aqi.equals("严重污染")) { aqiSquare.setBackgroundColor(Color.*parseColor*("#562E37")); }  
 else { aqiSquare.setBackgroundColor(Color.*parseColor*("#1C1C1C")); }  
  
 }  
  
 Runnable runnable = new Runnable() {  
 @Override  
 public void run() {  
 tenkiInfoChanged(newInfo);  
 }  
 };  
  
}