****

**本科生毕业论文（设计）册**

**基于Android的天气预报软件的设计与实现**

学 院： 软件学院

专 业： 软件工程

年 级： 2017级

学生姓名： 王佳成

指导教师： 丁盟

完成日期： 2021 年 3 月 26 日

目 录

[1. 研究背景与开发意义 5](#_Toc9998)

[2. 需求分析 5](#_Toc1125)

[2.1 主要功能分析： 5](#_Toc23759)

[2.1.1 天气显示页面： 5](#_Toc3572)

[2.1.2 城市管理页面： 6](#_Toc26701)

[2.1.3 添加城市页面： 6](#_Toc30998)

[2.1.4 删除城市页面： 6](#_Toc27561)

[2.2 功能结构图： 7](#_Toc22489)

[2.3 其他问题 7](#_Toc28541)

[2.3.1 合理的存储数据 7](#_Toc23588)

[2.3.2 简洁友好的界面 8](#_Toc25319)

[3. 关键设计 8](#_Toc29226)

[3.1 开发与设计的总体思想 8](#_Toc31918)

[3.2 技术支持 9](#_Toc2925)

[3.3 数据库存储设计 10](#_Toc3771)

[3.4 具体功能设计 11](#_Toc1758)

[3.4.1 天气信息展示功能设计： 11](#_Toc9796)

[3.4.2 城市管理功能设计： 12](#_Toc18130)

[3.4.3 添加城市功能设计： 12](#_Toc26198)

[3.4.4 删除城市功能设计： 13](#_Toc19441)

[4. 系统实现 14](#_Toc14239)

[4.1 软件整体介绍 14](#_Toc9286)

[4.1 具体功能展示 16](#_Toc32685)

[5. 总结 22](#_Toc14356)

[5.1 软件的成果与收获： 22](#_Toc25339)

[5.1 软件的不足： 23](#_Toc2596)

**摘 要**

随着时代的发展和科技的进步，人们的出行变得多种多样，而天气情是况就愈发的离不开人们的生活。而随着智能手机的普及的网络时代的到来，人们也越来越希望能在手机上能实时接受天气情况。

本软件使用Android技术设计开发了一种在Android系统上的手机实时天气情况接收软件，通过用户搜索，来展示不同城市的天气情况，实时温度，最低气温和最高气温等基础天气信息帮助您了解当天天气情况，还有包括紫外线系数、降雨指数等在内的一系列生活系数帮助您在出行前做好充分准备，更包括了未来三天的天气情况助您未雨绸缪，极大的方便了您的出行和行程安排，节约了时间并减少了很多不必要的麻烦，有超强的实用性。

本文首先介绍了本项目的项目背景，需求分析，产品优点，接着分析了对Android系统的理解，并介绍了项目开发过程中遇到的技术难关及攻克时自己对其理解，然后论述了系统的详细设计（包括功能设计、页面跳转关系、数据库设计等），并附有自我测试用例分析，最后给出了产品未来的一些开发方向和展望。

**关键词：** 天气 Android 异步线程 数据库

**Abstract**

With the development of The era and the progress of science and technology, people's travel has become diversified, and the weather situation is increasingly inseparable from people's life.With the advent of the Internet era with the popularity of smart phones, people increasingly expect to be able to receive real-time weather conditions on their mobile phones.

This software USES the Android technology designed and developed a mobile phone on a Android system real-time weather conditions to receive software, through the users to search, to show the weather in different cities, real-time temperature, lowest temperature and highest temperature basic weather information to help you understand the weather that day, including ultraviolet radiation coefficient, coefficient of rainfall index, etc. A series of life to help you fully prepared before travel, more includes the next three days, the weather help you save for a rainy day, a great convenience to your travel and travel arrangements, and saves time and reduces a lot of unnecessary trouble, has a strong practicability.

This article first introduces the background of the objective of the project, requirement analysis, product advantages, and then analyzes the understanding of the Android system, and introduces the project encountered in the process of development of technical challenges and conquer their understanding to its, and then discusses the detailed design of the system, including functional design, page jump relations, database design, etc.), with self test case analysis, finally gives the product some of the future development direction and prospects.

**Key words**: weather Android Asynchronous thread The database

## 基于Android的天气预报软件的设计与实现

# 研究背景与开发意义

Android一词本义为“机器人”，同时Android是一种基于Linux内核的开放源代码的智能操作系统，主要应用于移动设备中作为操作系统出现，如平板电脑、智能手机以及当今流行的智能家电等，大多数都采用了Android作为其系统软件.Android平台作为Google开发的系统软件，由操作系统、中间件、用户界面和应用软件组成，其底层由C语言开发，提供基础功能；中间层由C++编写，包括虚拟机和数据库；应用软件则由安卓开发人员自行开发，以Java语言或者kotlin语言编写。

本软件就是基于此Android平台开发，可以在Android系统中运行，正好顺应了当今大批人使用Android手机的潮流。而当今时代，伴随着人们生活密不可分的另一因素就是天气情况，到了季节交替之际天气情况诡异多变，话转之间乌云密布，对人们出行规划有极大的影响。准确的了解此刻与未来的天气情况，几乎成了每个人每天必须了解的信息。而现有的大多数天气软件华而不实，花哨的界面和层出不穷的广告严重影响着用户的体验。

而相比之下，本产品旨在以简洁的界面和功能，来向大众用户提供最合适的天气信息，温度信息和未来天气预报，以便用户可以在任何时刻轻松的了解当前与未来的天气信息，方便大众出行和行程安排，节约了时间并减少了很多不必要的麻烦，有超强的实用性。

# 需求分析

## 主要功能分析

设计并完成一个实用且方便的天气预报软件系统，同时还要满足简洁的界面和完整的功能以满足用户对天气信息了解的需求，本项目需完成以下需求：

### 天气显示页面

天气显示页面需按照不同城市划分，显示至多五个城市的各种天气信息，每个城市中要包括最基础的实时温度、天气状况和当前显示的城市名称，还要具有预测今天、明天、后天三天的天气情况（包含天气状况、最低温度和最高温度），与此同时为了方便用户出行以及其他计划，还应显示包括穿衣指数、防晒指数、感冒指数等在内的六大生活指数。整体界面布局应秉持着简介、明了的主题，以最直接的方式显示出用户最希望关注的信息。

### 城市管理页面

考虑到大多数用户需求的不仅仅是单个城市的天气信息，由于工作，生活以及其他原因，用户需要了解多个城市的天气状况，所以此软件还需要能够对城市进行添加、删除等操作。在城市管理页面需要以列表的形式简洁的展示出当前已添加的所有城市，还应显示出每个城市中最关键重要的天气信息，即城市名、天气状况、当前温度以及最低气温和最高气温。在城市管理页面还应有跳转到“添加城市页面”的入口以及跳转到“删除城市页面”的入口。整体页面布局应秉持着简洁明了的主题。

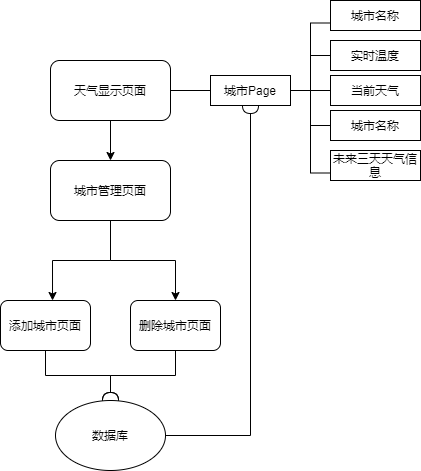
### 添加城市页面

由“城市管理页面”跳转到“添加城市页面”，在“添加城市页面”要含有搜索框，能够搜索全国范围内的城市，对搜索内容要能够进行筛选，对空内容、错误城市进行拦截并提示，防止因为添加错误信息导致的程序崩溃，信息错乱的现象。“添加城市页面”还应包括“热门城市”部分，方便用户直接点击对应城市添加。

### 删除城市页面

由“城市管理页面”跳转到“删除城市页面”，在“删除城市页面”中要按列表形式展示出所有已添加的城市，并含有删除按钮，当用户点击城市对应的删除按钮即删除城市，但为了防止用户误操作删除了不应删除的城市，点击删除按钮后城市虽然消失，但只是将城市暂存在删除列表中，待用户点击提交按钮并确认后才将所选城市删除。还应设有取消按钮，点击取消后，恢复删除列表中的城市。添加删除列表和多次确认的逻辑防止了用户误删掉了不应删除的城市。

## 功能结构图



## 其他问题

### 合理的存储数据

在该软件中，考虑到信息存储，需要按照城市区分来对每个城市的城市名称，城市代码，实时温度，天气状况，最低气温和最高气温以及未来三天天气状况进行存储。此软件选用了Android平台的轻量级数据库SQLite，SQLite与其他数据库相比，具有操作方便，管理简单，易于维护的优点。

在软件创建初始时即创建sqlite数据库，初始化好表名weather并设置合适的数据结构构建各字段名，其中城市id，城市名，和城市代码都是唯一字段不可重复。在添加城市时将城市名添加入数据库，从网络请求返回天气信息时将该城市的具体天气信息和未来三天信息存入数据库，在“展示天气信息页面”按照城市名称和城市代码从数据库中检索出其对应的城市信息并在界面展示给观众，这样可以避免每次查看天气时都访问网络数据，一方面节省了网络流量，另一方面也减少了等待时间，同时还保证了数据的全面性。

### 简洁友好的界面

用户界面（user Interface 简称UI）是一个软件与用户之间信息传递最直接的方式，也是普通用户对一个软件的第一印象，所以一个有好的界面对于一个软件产品来说至关重要。相比市面上大多数天气软件复杂繁琐且充斥着广告的界面，此软件界面最大优点就是简洁、直白，用最简单的方式将用户最希望了解的信息展示给用户，使用户可以第一时间知晓最有用的天气信息。

不单单是界面布局简约，颜色搭配也采用柔和的蓝色与白色相结合的主题，在天气状况旁还有合适的天气图标会根据阴晴雨雪不同的天气来改变，让用户能够更加直接的体会到天气的变化。

# 关键设计

## 开发与设计的总体思想

本软件采用平台为Android开发，基本功能为国内城市天气信息查询，故应遵循以下Android开发思想：

1）MVC思想：UI层需采用Android平台的MVC架构（Model-View-Controller）来接收用户动作、处理数据并显示UI界面等工作。Model代表数据模型，在这里负责对数据的操作、对网络服务等的操作；View代表视图层，用来显示用户界面及获取用户动作，是用户可以直观感受到的层级；Controller代表控制器层，这里负责接收View层获取到的用户动作（点击事件、双击事件、输入事件等），并分别处理事件的后续响应，操作Model进行数据更新处理，然后将结果交由View来展示给用户，可以说，是Controller将View和Model紧密联合起来，组成了一个完整的APP界面或功能。

2）异步思想：为了避免用户在操作界面时长时间进行无必要的等待，程序在进行网络请求及操作数据库这一类复杂耗时操作时，需采用多线程的方式来实现异步思想，以优化程序的流畅度，防止进程阻塞从而产生ANR（Application not Response）等崩溃现象，同时增加用户体验。在使用多线程时，还应注意避免因异步线程进行数据处理而导致的结果不同步问题。

## 技术支持

和风天气sdk：调用了和风天气sdk来进行城市搜索、天气信息实时查询、未来天气查询、生活指数查询等数据查询功能，具体回调接口如下：

1）城市查询接口：

QWeather.getGeoCityLookup(context, cityName, new QWeather.OnResultGeoListener());

此接口根据城市名称返回城市代码，在后续接口中需要上传城市代码来返回数据。

A.参数说明：第一个参数为应用上下文；第二个参数为要搜索的城市汉语名称；第三个参数为回调接口的监听。

B.回调监听中会根据返回结果重写两个方法：onError(Throwable e)和onSuccess(GeoBean geoBean)：当返回失败时，执行onError方法，返回带有错误信息的Throwable 对象；当返回成功时，执行onSuccess方法，返回带有城市信息的GeoBean对象。

GeoBean对象中含有两个参数，分别是表示返回结果成功与否的code和含有城市信息的locationBean对象的list容器，我们需要接收locationBean对象中的id参数来作为城市唯一标识-城市代码，后续将城市代码传入其他接口来获取其他信息。

2）实时天气查询：

QWeather.getWeatherNow(context, cityCode, new QWeather.OnResultWeatherNowListener());

此接口根据城市代码返回该城市当前实时温度及天气状况。

1. 参数说明：第一个参数为应用上下文；第二个参数为要搜索的城市代码，即上个接口得到的城市代码；第三个参数为回调接口的监听。
2. 回调监听中会根据返回结果重写两个方法：onError(Throwable e)和onSuccess(WeatherNowBean weatherBean)：当返回失败时，执行onError方法，返回带有错误信息的Throwable 对象；当返回成功时，执行onSuccess方法，返回带有城市信息的WeatherNowBean 对象。

WeatherNowBean 对象中含有四个参数，分别为表示返回结果成功与否的code，带有实况天气信息的NowBaseBean对象，表示数据来源和数据授权的Refer对象和表示Basic 基础信息的Basic对象。我们需要处理的是NowBaseBean对象，从中获取到当前实时温度和实时天气状况等信息，后续将其展示在界面显示。

3）未来天气查询：

QWeather.getWeather3D(context, cityCode, new QWeather.OnResultWeatherDailyListener())

此接口根据城市代码返回该城市未来三天内的天气信息

1. 参数说明：第一个参数为应用上下文；第二个参数为要搜索的城市代码，即第一个接口得到的城市代码；第三个参数为回调接口的监听。
2. B.回调监听中会根据返回结果重写两个方法：onError(Throwable e)和onSuccess(WeatherDailyBean weatherDailyBean)：当返回失败时，执行onError方法，返回带有错误信息的Throwable 对象；当返回成功时，执行onSuccess方法，返回带有城市信息的WeatherDailyBean 对象。

WeatherDailyBean 对象中含有四个参数，分别为表示返回结果成功与否的code，带有未来逐天天气信息的DailyBean对象数组，表示数据来源和数据授权的Refer对象和表示Basic 基础信息的Basic对象。我们需要处理的是带有未来逐天天气信息的DailyBean对象数组，数组中每个元素分别含有今天、明天、后天的天气状况、最低气温和最高气温等天气信息，我们需要得到这些信息，后续将其展示在界面显示。

## 数据库存储设计

在程序中我们需要按照城市区分来对每个城市的城市名称，城市代码，实时温度，天气状况，最低气温和最高气温以及未来三天天气状况进行存储，我们选择的数据库为SQLite。SQLite为Android平台自带的一种轻型数据库，它的设计目标是嵌入式的，而且目前已经有很多嵌入式产品使用了它。它占用资源非常的低，可能只需要几百K的内存就足够了，而且它比起一些流行的数据库操作速度是要更快的，而且较独立没有额外依赖。SQLite这样的特性使其很适合用在我们这个轻型的app中，我们只需要在程序创建之初，定义好数据库的结构并初始化就可以了。

我们需要在使用数据库时设计好我们的表结构，我们需要在表中定义的字段有：整形的id作为城市天气信息的唯一标识；字符串类型的城市名称cityName表示城市的汉语名称；字符串类型的城市代码cityCode用于在接口中传递搜索天气信息；字符串类型的condition表示该城市的实时天气状况（阴晴雨雪等）；字符串类型的curTemp表示该城市的实时气温；字符串类型的todayIcon表示该城市今天天气状况的图标，用于在天气展示界面未来三天预报处根据天气状况更换图标；字符串类型的todayMinTemp表示该城市今天的最低气温；字符串类型的todayMaxTemp表示该城市今天的最高气温；字符串类型的todayCondition表示该城市今天的总体天气状况；字符串类型的tomorrowIcon表示该城市明天天气状况的图标，用于在天气展示界面未来三天预报处根据天气状况更换图标；字符串类型的tomorrowMinTemp表示该城市明天的最低气温；字符串类型的tomorrowMaxTemp表示该城市明天的最高气温；字符串类型的tomorrowCondition表示该城市明天的总体天气状况；字符串类型的nextIcon表示该城市后天天气状况的图标，用于在天气展示界面未来三天预报处根据天气状况更换图标；字符串类型的nextMinTemp表示该城市后天的最低气温；字符串类型的nextMaxTemp表示该城市后天的最高气温；字符串类型的nextCondition表示该城市后天的总体天气状况。

## 具体功能设计

### 3.4.1 天气信息展示功能设计

天气信息展示界面是程序的主界面，在这里我们不仅要展示一个城市的所有天气信息，还要按照不同城市予以区分，所以我们在主界面最外层的相对布局下，引用了ViewPager。ViewPager是一种视图翻页工具，它提供了拥有多个界面时的页面切换效果。ViewPager类直接继承了ViewGroup类，所以它是一个容器类，还可以在其中添加其他的view类。ViewPager是Android 3.0后引入的一个UI控件，位于v4包中，低版本使用需要导入v4包，但是我们程序使用的是AndroidX，所以不用额外导包，是一种比较方便的方式，ViewPager在AndroidX中位于viewpager.widget下。ViewPager经常和fragment一起用，多个fragment组成ViewPager的子页面，所以我们需要在创建一个CityWeatherFragment类和一个fragment\_city\_weather的xml页面来作为实际展示城市天气信息的部分。

在fragment\_city\_weather页面中我们创建好一个城市天气展示的前端部分。因为要展示的页面信息较多，为了防止页面布局出现拥挤的情况，我们在整个fragment的最外层采用了ScrollView。ScrollView是Android的一种基本UI控件，它可以使页面出现竖直滚动条的效果，这样页面信息很多时我们只需要向下滑动屏幕就可以将信息展示全面，也不必担心页面拥挤的情况。但要注意的是ScrollView的子元素只能有一个，可以是View（如TextView或者ImageView），也可以是一个ViewGroup（如LinearLayout或者RelativeLayout），所以我们在ScrollView内加入一个相对布局RelativeLayout来存放内部信息。我们在整个页面的最上方用一个TextView显示城市名称，以区分不同的城市页面。因为当前实时温度是用户最关系的天气信息，所以在页面中央用较大字体显示当前城市的实时气温，并在实时气温下方显示当前城市的天气状况，阴晴雨雪等。在中央靠下方的位置我们需要展示该城市未来三天的简要天气信息，所以我们创建了一个相对布局展示今天的天气信息，宽度为填充父布局，高度自适应，最左侧一个ImageView用来显示今天的天气状况图标，图标右侧一个TextView显示该城市今天的总体天气状况，最右侧一个TextView展示该城市今天的最低气温和最高气温。然后将这个相对布局再次重复两遍以展示明天和后天的天气状况。在页面的最下方需要展示该城市当天的生活指数信息，包含穿衣指数、防晒指数、感冒指数、降雨指数、洗车指数、运动指数，我采用一个TextView显示具体信息，然后设置TextView的drawableBottom属性来显示指数图标，然后重复六次来显示所有的指数信息。这样一个城市天气信息分页就完成了，在主页面中，我们还需要在页面顶部增加一个LinearLayout在中央显示，让其存放页面指示器，即指示现在在哪个页面的小圆点。

ViewPager需要一个adapter来给他提供数据，并且提供了专门的FragmentPagerAdapter和FragmentStatePagerAdapter类使用。我们创建一个CityFragmentAdapter继承自FragmentStatePagerAdapter，在adapter里创建一个fragment泛型的List来存储城市fragment，并重写geiItem()方法返回list的当前位置和geiCount()方法返回list中数据的数量。

在MainActivity中我们需要初始化adapter，向adapter中提供数据，并将adapter设置到ViewPager中。数据源我采用fragment泛型的fragmentList来存储城市fragment和String泛型的cityList来存储所有的城市名称，在页面创建之初我们从数据库中检索出所有的城市名称放入citylist中，然后根据cityList的容量创建fragment，用Bundle传入城市名称，将创建好的fragment放入fragmentList，将fragmentList放入adapter中，最后获取到ViewPager调用setAdapter，将CityFragmentAdapter和ViewPager联系起来。

在CityWeatherFragment中我们需要调用三个接口，分别是获取城市代码的QWeather.getGeoCityLookup，获取实时天气的QWeather.getWeatherNow和获取未来三天天气的QWeather.getWeather3D。但要注意三个接口是有顺序性的，我们需要先获取城市代码，然后才能将城市代码传入剩余两个接口获得天气信息。还要注意三个接口返回成功数据的方法是回调机制，所以结果的同步性要处理好，在这里我们将三个接口放在一个串行流程里，让其依次执行，就可以保证在最后我们的结果是接收到了所有接口返回来的数据。接收到城市天气数据后，我们需要将城市天气信息存入数据库。重写onResume方法将信息展示在界面中。

### 3.4.2 城市管理功能设计

在主页面左上角的加号按钮会跳转到城市管理页面，在城市管理页面我们需要有跳转到添加城市界面的入口，和跳转到删除城市的入口，还应展示出当前已经存储的所有城市及其简要的天气信息。

在城市管理界面，我创建了一个ListView，用来展示所有城市及其天气信息，创建了item布局显示每个城市的具体细节。创建CityManagerAdapter为其提供数据，在adapter中创建ViewHolder类用来承载ListView中每个item，重写getView方法将item中的每个控件填充数据。

在CityManagerActivity中要将adapter绑定到ListView中，并从数据库中检索出所有城市的天气信息放入List中作为ListView的数据源，这样就可以显示出所有城市列表以及他们的粗略天气信息。最后再增加跳转到添加城市界面和删除城市界面的入口，此时还要判断当前已经添加的城市数量是否达到上限，这里我们设定城市上限为5个，达到五个后不能继续添加城市。

### 3.4.3 添加城市功能设计

在城市管理页面可以跳转到添加城市界面。在添加城市界面用一个EditText作为搜索框，但要注意当用户在EditText输入回车时，会出现换行，而我们想做的是输入回车即触发搜索，所以要为EditText设置setOnEditorActionListener，重写onEditorAction方法，onEditorAction方法有三个参数：TextView v，因为EditText是TextView的子类，所以第一个参数表示当前触发事件的EditText，相当于findViewById的功能；第二个参数：int actionId，表示动作标识，需要与EditorInfo中的参数对比可以判断出执行了什么动作；第三个参数：KeyEvent event，表示事件，需要跟KeyEvent中的参数比较值判断他的事件。在方法中我们判断如果cationId等于EditorInfo.IME\_ACTION\_SEARCH，就代表此时用户点击了软键盘中的回车，就执行搜索事件，在这里我们让其触发搜索按钮的点击事件，这样我们就不用写两次点击事件了，即submitIv.performClick()。

我们还需要在添加城市界面展示出热门城市，因为热门城市比较多，所以我们用一个网格视图GridView展示，为GridView添加adapter和数据源使其能够展示出热门城市，但我们还希望用户点击城市能够直接添加，所以还需要为其添加setOnItemClickListener事件，重写onItemClick方法，当用户点击网格中的任意一个item后会触发这个事件，而调用onItemClick方法，该方法有四个参数，第一个参数：AdapterView<?> parent，parent相当于GridView适配器的一个指针，我们可以通过parent来获取适配器中的数据；第二个参数： View view，view表示你点击的item，可以通过view来获取item中的控件；第三个参数：int position表示点击的item在GridView适配器的位置，我们可以通过位置来知晓点击的是哪个item；第四个参数：long id，id表示view的id，是与position功能相同的。在onItemClick方法中，我们通过position获取到点击的内容，然后将其传递到搜索方法中，执行搜索。搜索成功后将直接跳转到主界面（天气展示界面）展示搜索到的城市的天气情况，如果搜索内容为空或者输入内容为空的话，需要弹出Toast提示用户重新输入。

### 3.4.4 删除城市功能设计

在城市管理页面可以跳转到删除城市界面。在删除城市界面需要展示出所有城市列表，但不需要再显示出其天气信息，所以我们用一个ListView展示所有城市名称，创建item显示每个城市的具体细节，需要显示城市名称和一个删除按钮。在DeleteCityActivity中我们需要为ListView添加数据源，并添加点击事件，当用户点击城市旁边的删除按钮时，我们在list中删除该城市，并提示设配器刷新，但我们暂时不在数据库中删除，只是将他暂存在删除列表中，带用户点击提交按钮时，再将所有在删除列表中的城市从数据库中删除，当用户点击取消按钮时，我们需要将删除列表中的城市重新显示在list中，添加删除列表和多次确认的逻辑防止了用户误删掉了不应删除的城市，优化了用户体验。

# 系统实现

## 软件整体介绍



图4-1-1 天气信息展示界面 图4-1-2城市管理界面



图4-1-3 添加城市界面 图4-1-4删除城市界面

## 具体功能展示

天气信息与百度天气对比，相差极小：



图4-2-1 天气app中天气情况 图4-2-2 百度天气中的天气情况

添加城市功能展示，以添加上海为例：

①此时城市管理界面没有上海市：



图4-2-3 添加城市步骤1

②在添加城市界面搜索上海市：



图4-2-4 添加城市步骤2

③搜索成功，跳转到主界面展示上海市天气信息：



图4-2-5 添加城市步骤3

④此时城市管理界面也添加了上海市：



图4-2-6 添加城市步骤4

删除城市功能展示，以删除石家庄市和上海市为例：

①此时在城市管理界面有石家庄市：



图4-2-7 删除城市步骤1

②在城市管理界面点击右上角删除按钮进入删除城市界面：



图4-2-8 删除城市步骤2

③点击石家庄右边的❌删除石家庄：

④此时删除城市界面没有了石家庄：



图4-2-9 删除城市步骤3和步骤4

⑤点击右上角提交按钮，并点击确认：



图4-2-10 删除城市步骤5

⑥城市管理页面没有了石家庄：



图4-2-11 删除城市步骤6

⑦再次删除上海市：

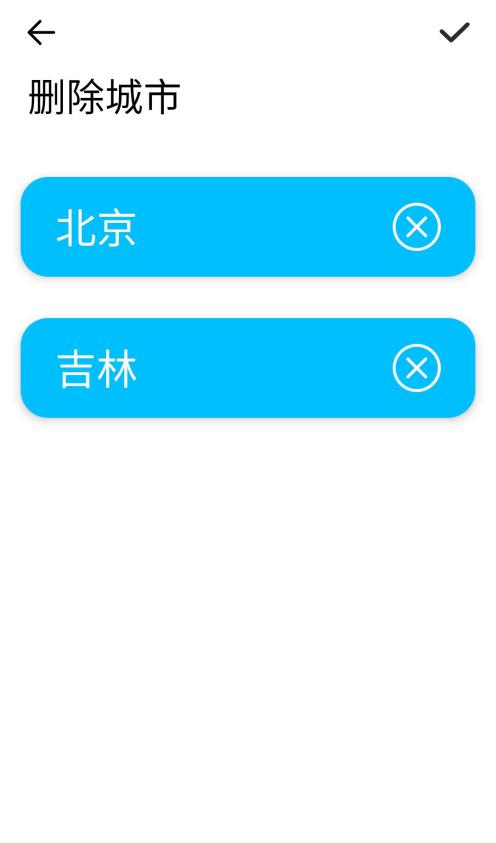


图4-2-12 删除城市步骤7

⑧点击左上角撤销按钮，并点击放弃更改：



图4-2-13 删除城市步骤8

⑨返回城市管理界面，上海市依然存在：



图4-2-14 删除城市步骤9

# 总结

## 软件的成果与收获

1）下面介绍软件的具体实现情况：

* 天气展示界面实现了根据城市分页展示天气信息

天气展示界面以ViewPager的形式展示不同城市，城市不同布局相同，节俭了代码的同时也方便了用户在不同城市之间的切换；界面中央显示实时温度和实时天气状况，使得用户能够最直接的了解此时天气；预测未来三天天气，方便用户行程规划；报告常用生活指数，帮助用户出行准备。

* 城市管理界面实现了对所有存储城市的管理

在城市管理界面用listView展示所有城市天气概况，有利于用户比较城市间的天气差别，也帮助用户概览关注的城市；搜索添加城市和删除城市入口帮助用户管理添加的城市。

* 添加城市界面实现了搜索城市添加功能

添加城市界面用GridView的形式展示热门城市，方便用户一键添加想要关注的城市；还可以搜索全国城市，用户可以随时查看全国各地城市状况。

* 删除城市界面实现了删除城市的功能

删除城市界面利用删除城市列表和多次确认的逻辑减少了用户误删了不想删除的城市的可能。

1. 在本次设计过程中我有了以下收获：

* 对Android系统有了更深入的了解和熟悉
* 对Android一些UI控件例如ScrollView、CardView、EditText等有了更深入的学习
* 对Android操作数据库更加熟悉
* 对Android多线程操作有了更多的实践

## 软件的不足

由于时间仓促原因和个人开发能力不足，现软件还包含诸多问题，列举如下：

* 由于开发能力受限，所以在处理联网接口时会存在少许延迟现象；
* 由于开发前没有充分考虑到内存处理问题和缓存问题，所以可能存在内存占用过多的现象；
* 由于接口原因，目前只能显示到市级城市，位置不够精确，城市采集也不够完整。

## 软件展望

待个人开发技术提升后，打算对软件做出如下优化：

* 对上述不足予以改进；
* 增加定位功能，用户打开后可以直接获取定位显示当前地址的天气信息；
* 增加动态背景，使背景能够根据天气情况进行调整；
* 增加未来七天天气折线图表示。

接下来，我会尽可能的提升个人水平，以努力完善该软件，使其做到最好。