



课程设计报告

题目: 天气预报 APP —— 天气の



专业班级: 软工 191

学 号: 2019212212168

姓 名: 陈亮亮

指导教师: 徐舒畅

报告日期: 2021/12/18

信息科学与技术学院



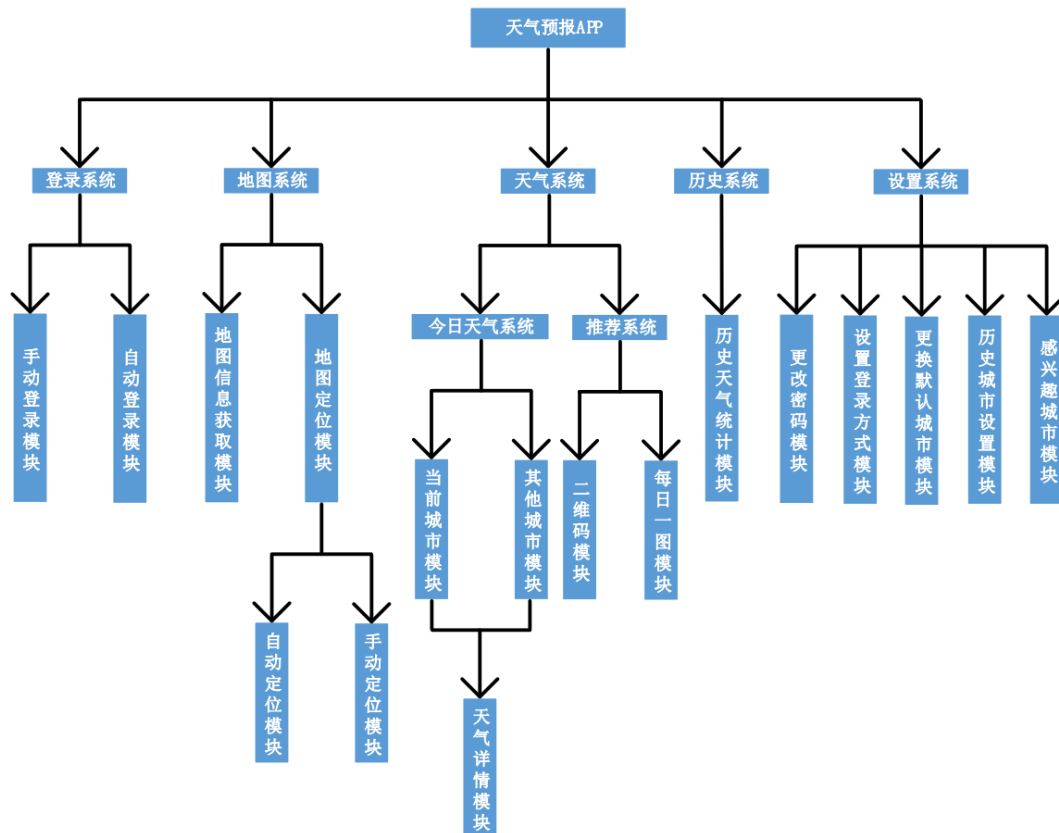
目 录

一、系统功能.....	3
1.1 系统功能模块图	3
1.2 系统功能模块介绍	3
1.2.1 登陆系统	3
1.2.2 地图系统	3
1.2.3 天气系统	3
1.2.4 历史系统	4
1.2.5 设置系统	4
二、系统设计.....	5
2.1 系统设计概述	5
2.2 组件和控件的引入	5
2.2.1 WebView 和 ImageView.....	5
2.2.2 Vue 和 Mintui 的引入	5
2.2.3 百度地图 SDK 的引入.....	6
2.2.4 QRcodejs2 的引入.....	6
2.2.5 Echarts 的引入.....	6
三、重点难点实现.....	7
3.1 Vue+WebView 模式的搭建.....	7
3.2 百度地图 SDK 的调用	7
3.3 天气详情的获取	8
3.4 二维码的生成	8
3.5 JavaScript 和 WebView 的通讯.....	9
3.6 每日一图拍照功能的实现	9
3.7 自动登录的实现	10
3.8 数据库 CURD 的实现.....	10
3.9 Echarts 的调用.....	10
四、系统实现.....	11
4.1 登录界面.....	11
4.2 地图 Tab 页	11
4.3 天气 Tab 页.....	12
4.3.1 今日 Tab 页.....	12
4.3.2 推荐 Tab 页.....	12
4.4 历史 Tab 页.....	13
4.5 设置 Tab 页.....	13
4.6 良好的代码管理	14
五、版本兼容问题.....	15
六、课程心得.....	16



一、系统功能

1.1 系统功能模块图



1.2 系统功能模块介绍

1.2.1 登陆系统

- **手动登录模块：**判断账号密码是否存在数据库中，若存在则进入应用。
- **自动登录模块：**若设置自动登录，存在则跳过登录进入应用。

1.2.2 地图系统

- **地图信息获取模块：**显示地图页面定位的城市情况、混合地图、地图实景。
- **地图定位模块：**获取当前定位信息、更改定位。

1.2.3 天气系统



- **当前城市模块：**以卡片的形式展示当前城市基本信息、背景为随机的图片。
- **其他城市模块：**显示感兴趣城市的简略天气信息。
- **天气详情模块：**点击当前城市卡片或者其他城市都将展示该城市的天气详情。
- **二维码模块：**输入内容或者网址，生成对应的二维码。
- **每日一图模块：**拍照记录图片，生成每日一图。

1.2.4 历史系统

- **历史天气统计模块：**显示历史查询城市天气的折线图。

1.2.5 设置系统

- **更改密码模块：**重新设置用户登录的密码。
- **设置登录方式模块：**设置用户登录方式，自动登录/手动登录。
- **更改默认城市模块：**更改当前定位城市。
- **历史城市设置模块：**设置历史页面城市折线图显示数量。
- **感兴趣城市模块：**添加自己感兴趣的的城市。

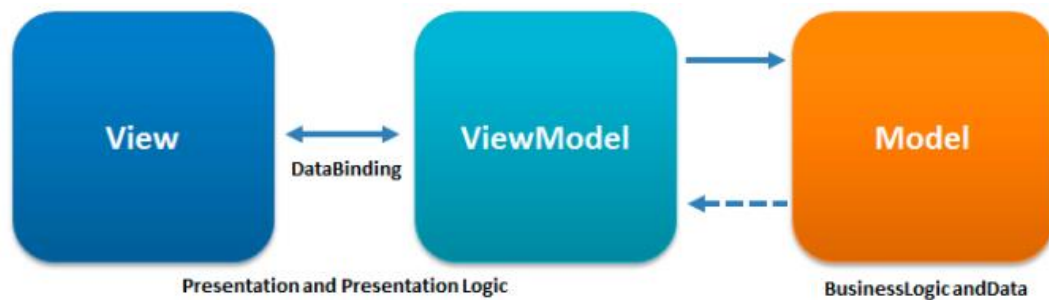


二、系统设计

2.1 系统设计概述

本学期我自学了当下比较热门的前端框架 Vue，并利用 Vue 完成了两个网页应用。感谢徐帅的支持，我借着做安卓大作业的这个机会再次利用 Vue 框架来完成一个手机端的应用程序，当然这也是我第一次尝试，因此很多地方设计、实现的并不是很满意。

整个 APP 主要使用 VUE+原生 Android 的 WebView 控件实现。Vue.js 是当下很火的一个 MVVM 开发模式框架，它是以数据驱动和组件化的思想构建的。相比于 Angular.js，Vue.js 提供了更加简洁、更易于理解的 API，使得我们能够快速地上手并使用 Vue.js。mint 是一个基于 vue 的前端 UI 框架，而它的样式比较类似于手机的样式，可以说是一个基于 vue 打包 app 的 UI 框架，使用 mint 框架可以给使用 vue 打包的 app 的用户更好的交互体验。mint 已支持 vue2.0。



前端部分主要采用 VUE + Mintui 搭建，另外引入了 axios、百度地图 SDK、qrcodejs2、echarts 等等模块和 package 来完成前端页面的构建；在原生安卓部分，我主要使用了 WebView 控件和 ImageView 控件；后端采用 room+sqlite 来维护数据库。

APP 主要包括 4 大功能模块（地图模块、天气模块、历史模块、设置模块）、分为五个子系统，包括登录系统、地图系统、天气系统、历史系统、设置系统，其中天气信息系统又包含今日天气系统和推荐系统。这四大功能模块，以类似微信的 Tab 页面方式，显示在页面的底部。

2.2 组件和控件的引入

2.2.1 WebView 和 ImageView

在原生安卓中引入 WebView 和 ImageView 控件，WebView 是一个可视化的组件，是一个嵌入式的浏览器，是嵌入在原生应用中的，用户可能都意识不到自己在用浏览器。ImageView 用来实现拍照功能。

2.2.2 Vue 和 Mintui 的引入



前端页面搭建的第一步就是创建 Vue 工程项目，并引入 Mintui 相关组件做好相关配置。在这里一次性导入 Vue 相关的 Vuex、vue-router、axios 等一些基础的必要组件，为后续的开发打好基础。

2.2.3 百度地图 SDK 的引入

地图系统所依赖的是百度地图的 SDK 包。通过调用百度地图的 SDK，我可以很轻松地获取到我想要的地图相关的信息。

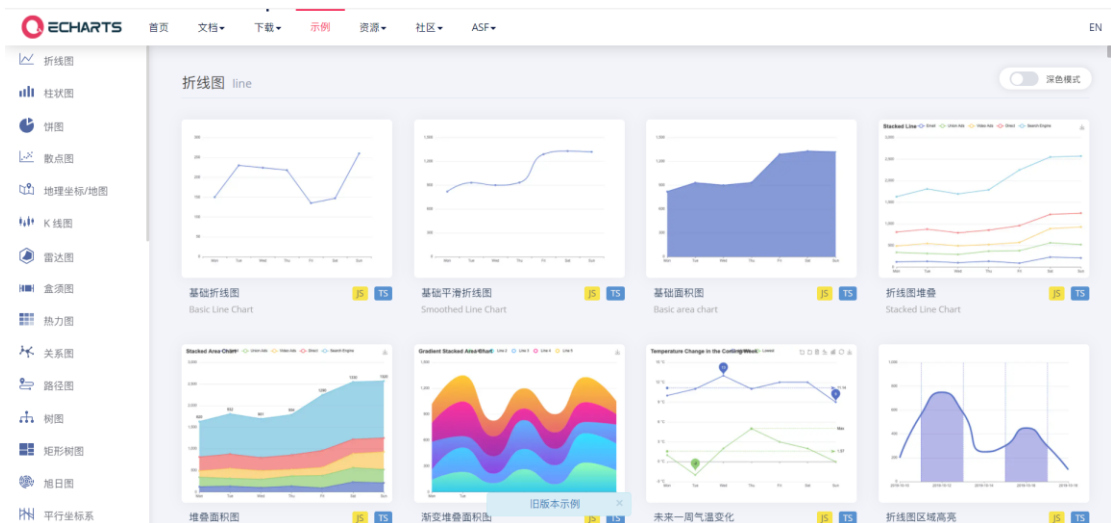


2.2.4 QRcodejs2 的引入

如果在 vue 项目中，想快速简单的根据后台返回的链接生成二维码，那么 qrcodejs2 是一个很好的选择。引入 qrcodejs2 组件后配合 canvas 元素即可快速的实现二维码的生成。

2.2.5 Echarts 的引入

Echarts 是一款可以配合 Vue2.0 使用的图表生成组件。





三、重点难点实现

3.1 Vue+WebView 模式的搭建

因为我采用的是浏览器式的前端应用程序内嵌在 WebView 组件中的开发模式，因此在一开始，我需要对 WebView 进行一系列的设置，其中包括了允许应用程序的网络访问权力、允许 JavaScript 执行、允许浏览器弹窗、访问 IP 地址设置等等的冗杂的配置。经过一番很长时间的试错和斗争，最终才完成各项配置。

```
WebView webView = findViewById(R.id.webView);
webView.getSettings().setJavaScriptEnabled(true);
webView.getSettings().setDomStorageEnabled(true);
// 防止所有后打开默认浏览器
webView.setWebViewClient(new MyWebViewClient());
webView.setWebChromeClient(new MyWebChromeClient());
//webView.loadUrl("https://www.baidu.com");
//webView.loadUrl("file:///android_asset/dist/index.html");
webView.loadUrl("http://192.168.31.242:8080/");
```

3.2 百度地图 SDK 的调用

前端通过 IP 地址进行定位可以定位到当前所在城市，但无法精准定位到很具体的位置。比如我只能自动定位到杭州市余杭区，但是无法自动定位到杭州市余杭区仓前街道杭州师范大学 B 区精思苑 2 号楼。经过探讨后，我发现使用原生安卓的同学可以直接获取到当前的精准定位，于是我开始了解决之路，我查阅了百度地图 SDK 的开发文档并使用了其中的各种函数、属性来获取当前的具体定位，但是无论如何都无法获得很具体的定位，每次都只能定位在杭州市市民中心，也不知道什么原因会定位在这个位置。

但是我发现如果我直接赋予精确的经纬度是可以控制当前位置的，于是我希望通过原生安卓获取当前定位的经纬度然后传给前端定位，然而最终还是失败了。无奈下，我选择放弃，比较查询天气只需要精确到余杭区这一级别也已经足够了！

```
<template>
  <div>
    <mt-header title="我的位置" style="height: 60px; font-size: 1.2em">
      <mt-button icon="more" slot="right"></mt-button>
    </mt-header>
    <baidu-map
      class="bm-view"
      id="allmap"
      :center="center"
      :zoom="zoom"
    >
      <bm-map-type
        :map-types="['BMAP_NORMAL_MAP', 'BMAP_HYBRID_MAP']"
        anchor="BMAP_ANCHOR_TOP_LEFT"
      ></bm-map-type>
      <bm-panorama></bm-panorama>
    </baidu-map>
  </div>
</template>
```

设置当前位置中心、大小范围

设置图例、地图类型、实景等等



3.3 天气详情的获取

这个部分还算比较顺利，利用 `axios` 组件来执行跨域请求即可得到对应城市的天气。唯一的问题就是并发问题。我一开始采取了暴力循环请求的方式来获取一组城市的天气，然而我发现如果一次性请求的城市过多会导致请求丢失覆盖乱序，甚至请求失败的问题。为此，我选择一次性只请求有限个城市的天气，不惜上锁来解决该问题（由于上锁太卡了，严重影响用户体验，我又删除了）。当然是用 `JavaScript` 写的很简陋的锁，在上一个请求完成之前将程序阻塞在一个 `while` 循环中，直到上一个请求完成。

```
searchCitiesWeather() {
  for (var index in this.citys) {
    this.citiesWeather = [];
    axios({
      url: "http://wthrcdn.etouch.cn/weather_mini",
      method: "get",
      params: { city: this.citys[index].county },
    })
      .then((res) => {
        this.citiesWeather.push(res.data.data);
      })
      .catch((err) => {
        console.log(err);
      });
  }
  console.log(this.citiesWeather);
},
```

3.4 二维码的生成

二维码的生成算是最容易实现的部分了，引入 `qrcodejs2` 组件后只需按照操作文档进行操作即可，唯一的问题就是该组件是配合 `canvas` 实现的，每次产生不会覆盖上一次的结果，需要自己清除上一次的结果再新增。

```
/**
 * @description 生成二维码
 * @param {number} qWidth 宽度
 * @param {number} qHeight 高度
 * @param {string} qText 二维码内容（跳转连接）
 * @param {string} qRender 渲染方式（有两种方式 table和canvas，默认是canvas）
 */
qrcode(qWidth, qHeight, qText, qRender) {
  this.$refs.qrcode.innerHTML = "";
  // eslint-disable-next-line no-unused-vars
  let qrcode = new QRCode("qrcode", {
    width: qWidth,
    height: qHeight,
    text: qText,
    render: qRender,
  });
},

/**
 * @description 点击显示二维码
 */
showQRcode() {
  // 二维码初始化 点击一次添加一个二维码
  this.$nextTick(function () {
    this.qrcode(200, 200, this.link, "canvas");
  });
},
```




3.5 JavaScript 和 WebView 的通讯

JavaScript 和 WebView 的通讯这一部分算是实现 Vue+WebView 模式的关键了。因为无论是数据库的 CRUD 或者是后面的拍照功能的实现都需要在 JavaScript 中调用原生安卓中的方法来实现。这一模块我在网上找到了两种方法，一是构造 bridge 类来封装请求，这样比较安全便于管理；二是直接调用。出于便捷快速完成功能的角度，我选择了第二种方法。但为了管理，我还是构建了几个类，稍微封装了一下。

```
webView.addJavascriptInterface(new ShowCamera(), name: "camera");
webView.addJavascriptInterface(new DBPerform(), name: "DBPerform");
webView.addJavascriptInterface(new AutoLogin(), name: "AutoLogin");
```

```
public final class DBPerform {
    @JavascriptInterface
    public void getAllUserName()
    {
        StringBuilder stringBuilder = new StringBuilder();
        List<User> userList = AppDatabase.getInstance().userDao().getAll();
        for (User user:userList) {
            stringBuilder.append(", "+user.getName());
        }
        Toast.makeText(context: MainActivity.this, text: "查询到所有的名字为:"+stringBuilder, Toast.LENGTH_SHORT).
    }
}
```

3.6 每日一图拍照功能的实现

拍照功能也是耗费了我漫长漫长漫长漫长无比（将近 10 小时）的时间，听说使用原生安卓实现不需要几行即可完成。但我既然选择了 Vue，那么我就想着一定有方法可以不使用原生安卓即可调用手机摄像头的方法。一番苦寻后发现确实有！使用 H5 语法即可很轻松的调用手机摄像头，但是前提是整个 App 应用程序是搭建在 HBuilder 编译环境下，直接打包即可实现。很遗憾，我使用的是 VSCode。于是我又开始了新的苦寻，最终我失败了，我妥协了。我选择使用原生安卓调用摄像头并用前端 JavaScript 调用 Java 方法，最终才得以实现！！

```
class ShowCamera {
    //Html调用此方法传递数据，注解一定要留着否则会出错
    @JavascriptInterface
    public void showCamera() {
        photoImage.setVisibility(View.VISIBLE);
        photoImage = findViewById(R.id.photoImage);
        Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        startActivityForResult(intent, requestCode: 10);
    }
    @JavascriptInterface
    public void closeImage() { photoImage.setVisibility(View.GONE); }
}

@Override
public void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    switch (requestCode) {
        case 10:
            if (resultCode == RESULT_OK) {
                Bitmap bitmap = (Bitmap) data.getExtras().get("data");
                photoImage.setImageBitmap(bitmap);
            }
            break;
    }
}
```



3.7 自动登录的实现

自动登录实现还算简单，只需查看尝试一下 `SharedPreferences` 的使用即可实现。`SharedPreferences` 是一个存于本地的文本文件，有点像 `xml`，用起来还算方便容易。

```
public final class AutoLogin {
    @JavascriptInterface
    public boolean getStatus()
    {
        // 获取自动登录字段的状态
        SharedPreferences preferences=getSharedPreferences( name: "user", Context.MODE_PRIVATE);
        return preferences.getBoolean( s: "autoLogin", b: false);
    }
    @JavascriptInterface
    public void setStatus(boolean status) // 设置自动登录字段的状态
    {
        SharedPreferences preferences=getSharedPreferences( name: "user", Context.MODE_PRIVATE);
        SharedPreferences.Editor editor=preferences.edit();
        editor.putBoolean( s: "autoLogin", status);
        editor.commit();
    }
}
```

3.8 数据库 CURD 的实现

连接 `SQLite` 数据库的时候一开始并没有其他同学、老师所说的到处是坑。直到直到我新增了一张表、我更新了表字段，我遇到了前所未有的问题，怎么寻找都找不到 `bug` 的 `bug`。在和同学讨论后才发现，我进坑了，`SQLite` 在更新表结构时是必须要更迭版本的，并且还要手写 `sql` 语句来新增表、新增字段。当然，作为懒人的我选择直接删库重新建库省去了这些不必要的麻烦。但是这个操作的前提是我原有的数据库中并没有很多数据，在实际生产中，这种方法是要不得的（这里偷个小懒~）。

3.9 Echarts 的调用

最后做的是历史天气的折线图展示，这里我选用了 `echarts` 组件，获取到数据库中所存的历史城市天气数据，格式化之后赋值给对应的 `echarts` 属性即可完成绘制。调用 `echarts` 期间遇到了一个问题：不能同时绘制多个图表，在一番搜索后发现是我导入包的语句有点问题，最新版的导入方法和之前老的方法有所不同。从这一点再次看出，教程发布时间的重要性！！尽管是近两年的教材，但由于版本的更迭，很多网上的资料还是 `OUT` 了。



四、系统实现

4.1 登录界面

一个很简单的登录界面。



4.2 地图 Tab 页

地图页面展示了当前的位置信息，并且可以切换地图类型以及查看全景图。





4.3 天气 Tab 页

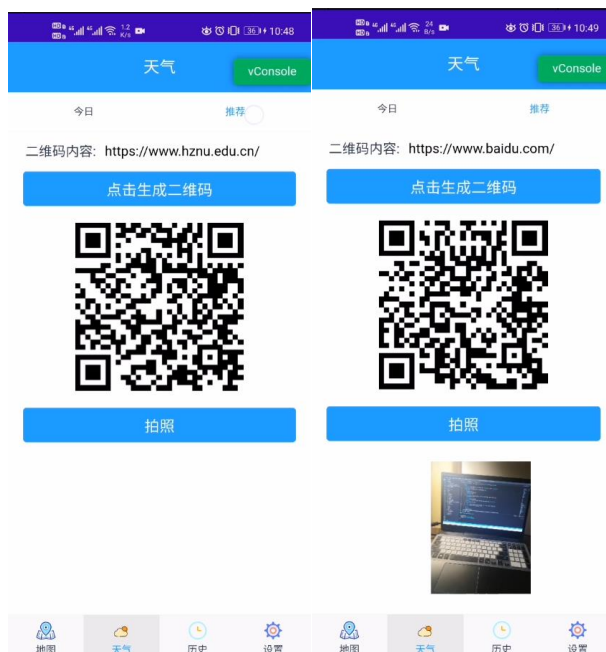
4.3.1 今日 Tab 页

今日页面展示的是当前城市和感兴趣城市的天气简介，点击后即可查看该城市的天气详情。



4.3.2 推荐 Tab 页

推荐页面包含二维码生成和每日一图拍照功能。输入二维码内容点击按钮即可字段生成对应二维码；点击拍照将调用摄像头完成拍照后照片将监视在页面中。





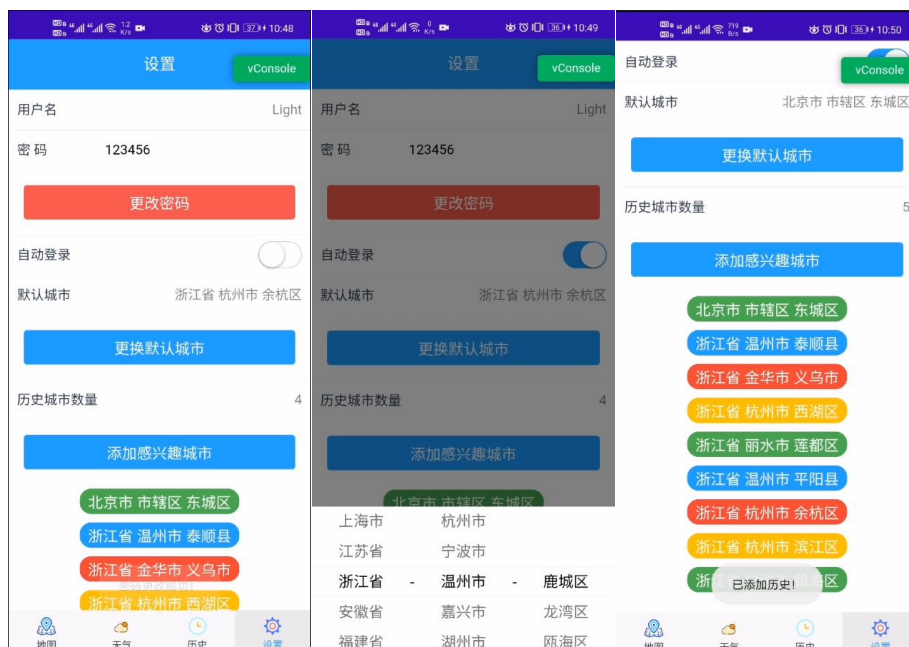
4.4 历史 Tab 页

历史页面展示的是历史城市的天气统计信息，展示的数量来自设置页面的历史城市数量设置。点击折线图即可查看对应 x 轴的 y 轴情况，也可以点击图例来设置该折线的显示与隐藏。



4.5 设置 Tab 页

设置页面包含用户名信息查看、密码更改、自动登录状态选择、默认城市选择、历史城市数量选择、感兴趣城市添加的功能。

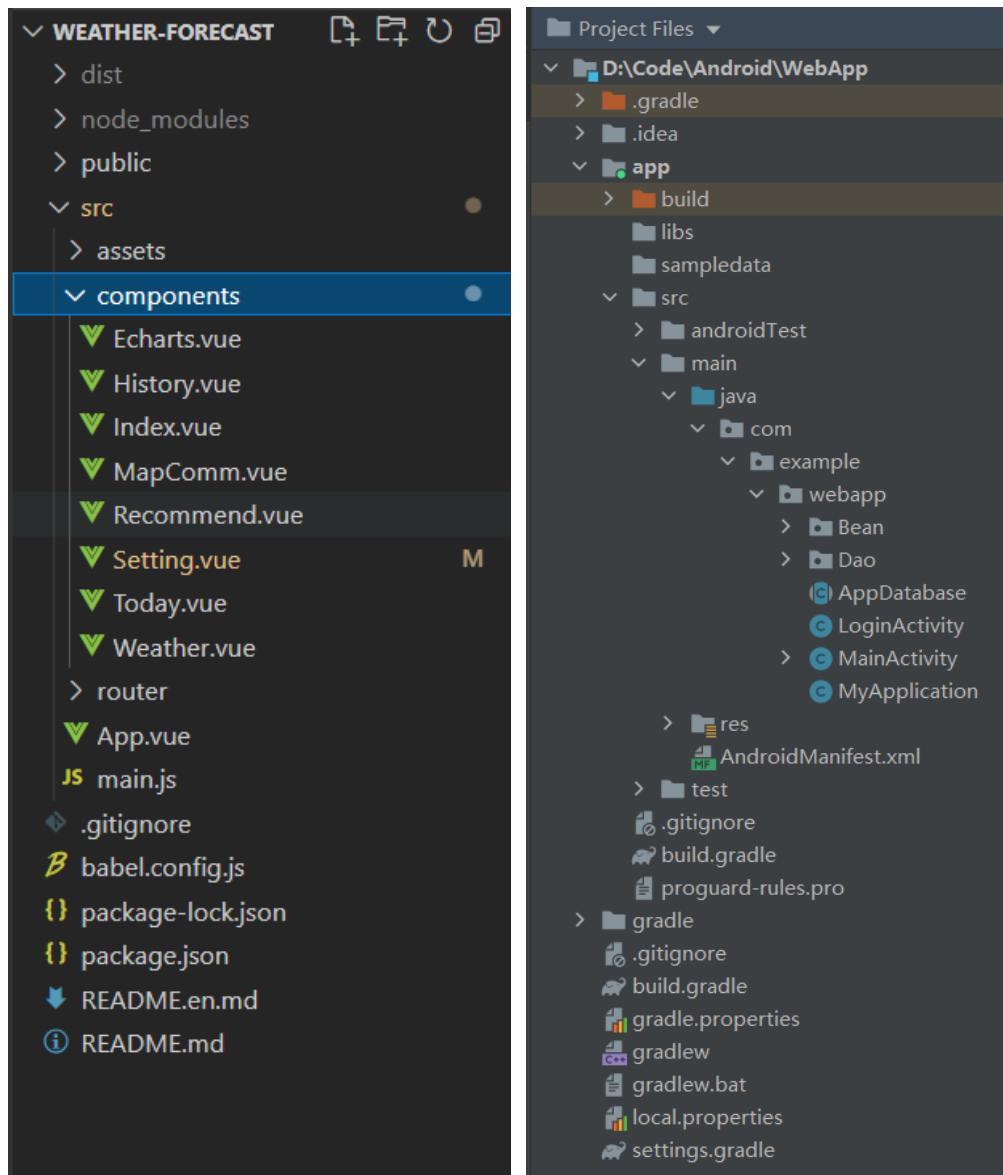




4.6 良好的代码管理

做一个项目时会涉及很多代码，如何更好的分类管理代码，是我们完成一个稍大项目的重要工作。本实验的代码量还不算大，我的前后端代码管理如下。

由于前端使用 nodejs 的方式编写，因此代码管理的层次结构还是比较清晰、有条理的。





五、版本兼容问题

经过测试，本 APP 可在 Android8、Android9、Android11、Android12 版本上顺利稳定地运行。由于我采用的是 Vue+WebView 的开发模式，因此本 APP 对安卓版本的宽容度较为广。只要是兼容 es6 语法的内嵌浏览器的安卓机基本都可以顺利且稳定地运行本 APP，但对于一些老旧的安卓机型和浏览器可能会有一些兼容问题，暂未做测试。

兼容问题还是比较头痛的，毕竟是实际生产应用中经常会碰到且几乎无法避免的问题，特别是移动应用开发，Android 的版本更迭太快以至于老的版本可能在某个时期就被时代抛弃了，例如本款 APP 的兼容度仅仅受限于是否支持 es6 语法、是否支持 vue.js、是否支持 mint-ui 以及其它的相关基础组件。



六、课程心得

作为一名软件工程专业的学生,我认为我有义务来学习《移动应用开发》这门课程。另外,我也为自己在学长学姐们的“**墙裂安利**”下有这样的勇气选择**徐帅**的这门课程。一学期下来,确实如学长学姐以及**徐帅**老师本人所说:这门课知识点多且杂,作业量庞大且难,**Android** 版本更迭的速度之快以至于网上陈列的资料仅有 30%是可以解决当下问题的。这个学期我们从美工设计、AS 的使用、UI 布局、UI 控件、UI 对话框及数据绑定、Fragment 类、Activity 类、Menu、Tab、ActionBar、广播、RecyclerView、服务等等等各类知识点、各个方面基本了解了安卓原生开发的经常用到的一些知识,算是带大家入门了。

本课程的大作业是做一个天气预报 APP,说实话,在第一次听到这个作业名字时我天真地认为,这个作业真水,应该很快就可以顺利搞定。然而在第一眼看到大作业功能清单的时候,我有点发懵,甚至是有无法承受的,这超出了我的心理预期!

在一番心理斗争后,我选择使用 **Vue** 来实现大作业的大部分内容。本学期我自学了当下比较热门的前端框架 **Vue**,并利用 **Vue** 完成了两个网页应用。在**徐帅**的支持,我借着做安卓大作业的这个机会再次利用 **Vue** 框架来完成一个手机端的应用程序,当然这也是我第一次尝试,因此很多地方设计、实现的并不是很完美。但好在,最终还是在合适的时间内完成了大作业的编写。

总的来说,经过一学期的学习,我基本入门了安卓,了解了安卓的基本写法,在此基础上,还通过自学,初步了解了当今一个比较流行的跨平台 APP 开发模式,提升了我对代码编写的理解,拓宽了我写代码时候的思路与方向,更是有了一丝时代前沿的流行气息。

此外,我想为大作业的功能提一些**小建议**:目前看来,功能清单中关于业务方面的功能已经比较全面了,但是登录模块着实有点过于草率,建议可以让下一届的学弟学妹们尝试使用邮箱或者手机号接受验证码注册并登录 APP,亦或调用腾讯 QQ、微信或者新浪微博的接口来进行创建用户并登录。想必调用接口的过程,也能学到不少知识,更是拓展了知识面!

最后,我不由地感叹,徐舒畅老师的《移动应用开发》课程真是一门值得选修的课程!作为学长,我一定**不忘初心、继承传统**,继续向学弟学妹极力安利**徐帅**的《移动应用开发》课程!