

1、任务分析

本次任务要求设计一款手机端雾霾app探测系统, 需要满足以下三个功能:

- 1.定位功能: 获取当前定位城市信息, 并显示在客户端。
- 2.界面设计: 包含显示天气详情和空气质量指数的动态显示。
- 3.天气详情和空气质量指数: 通过和风天气、墨迹天气、我的天气等均可获取, 百度地图、高德地图等可以辅助位置信息。

2、系统方案

2.1 app实现方案

app实现方案大致有三种:

1. 使用html进行网页开发, 手机通过浏览器的方式进行访问。虽然这能够解决不同机型和操作系统的适配问题, 但不是真正的app。
 2. 使用uniapp将网页开发成果在手机端进行部署, 和网页开发方式类似, 跨平台能力强, 开源社区活跃, 组件丰富, 用户体验佳, 开发潜力大。
 3. 使用Android原生开发, 选择Android Studio作为编辑器, 这样能够在Android操作系统上获得最佳的性能表现。不足之处在于无法适用于Apple的操作系统。
- 综合上述考虑, 我们考虑到用户体验和跨平台使用, 因此, 我们选择了uniapp开发。

2.2 app定位方案

uniapp定位实现方案大致有两种:

1. GPS定位。该方案需要用户打开GPS定位模块, 该方案准确度较高。
2. 网络定位。该方案通过用户的网络ip进行定位, 但准确率不高。

综合上述考虑, 我们选择使用H5中的函数获取手机GPS定位, 再通过获得的经纬度调用高德地图提供的地理逆编码API获取详细地理位置

2.3 天气查询实现方案

天气数据可以通过各种天气网站进行获取。我们选择了和风天气的数据接口。

3、系统实现

3.1 获取GPS经纬度信息

为了获取用户的定位信息, 我们需要用户在首次启动时, 对GPS调用权限进行授权。具体实现如下。

```
1 getLocation() {  
2   let _this = this
```

```

3      plus.geolocation.getCurrentPosition(onSucess,function(error){
4          switch (error.code) {
5              case error.PERMISSION_DENIED:
6                  x.innerHTML = "用户拒绝对获取地理位置的请求。"
7                  break;
8              case error.POSITION_UNAVAILABLE:
9                  x.innerHTML = "位置信息是不可用的。"
10                 break;
11              case error.TIMEOUT:
12                  x.innerHTML = "请求用户地理位置超时。"
13                  break;
14              case error.UNKNOWN_ERROR:
15                  x.innerHTML = "未知错误。"
16                  break;
17          }
18      });
19      function onSucess(position){
20          _this.location = String(position.coords.longitude) + ',' +
String(position.coords.latitude)
21      }
22  }

```

3.2 获取详细地理位置

我们采用高德地图的API来实现定位功能，主要步骤如下：

1. 在百度地图开放平台申请账号并新建应用。
2. 创建项目获取key
3. 通过调用高德地图地理逆转换API获得详细地址：<https://restapi.amap.com/v3/geocode/regeo>

具体实现如下

```

1  getAddress(){
2      let data = {}
3      data.key = this.gd_key
4      data.location = this.location
5      let _this = this
6      if (this.debug) console.log({
7          name: 'uni.request',
8          data: data
9      })
10     uni.request({
11         url: this.gd_xy_api_url,
12         data,
13         success: (e) => {
14             if (this.debug) console.log(e.data)
15             if (!this.check('obj', e.data)) {
16                 this.showToast('请检查url接口是否正确')
17                 return
18             }
19             if (this.check('state', e.data)) {
20                 console.log(e)
21                 let address = e.data.regeocode.addressComponent
22                 _this.province = address.province
23                 _this.city = address.city

```

```

24         _this.district = address.district
25         _this.city_adcode = address.adcode
26     },
27     },
28     fail: (e) => {
29         this.showToast('网络问题')
30         console.log(e)
31     }
32 })
33 }

```

3.3 天气查询功能实现

我们采用和风天气的接口来查询天气数据，空气质量数据以及一周内的温湿度数据，我们首先在和风天气App注册账号，然后获取网站提供的免费key值，最后就能调用接口。

接口地址如下：

- 实时天气数据接口：
<https://devapi.qweather.com/v7/weather/now>
- 天气预报接口
<https://devapi.qweather.com/v7/weather/7d>
- 空气质量数据接口：
<https://devapi.qweather.com/v7/air/now>

有了接口之后，我们需要在App内发起网络请求。

核心代码如下：

```

1  update() {
2      if (this.debug) console.log({
3          key: this.he_key,
4          location: this.location,
5      })
6      if (!this.check('key')) return
7      // 天气
8      uni.request({
9          url: this.he_weather_api_url,
10         data: {
11             key: this.he_key,
12             location: this.location,
13         },
14         success: (e) => {
15             console.log(e)
16             if (this.debug) console.log(e.data)
17             if (!this.check('obj', e.data)) {
18                 this.showToast('请检查url接口是否正确')
19                 return
20             }
21             if (!this.check('state', e.data)) return
22             this.weather.temp = e.data.now.temp
23             this.weather.text = e.data.now.text
24             this.weather.windDir = e.data.now.windDir
25             this.weather.windScale = e.data.now.windScale
26             this.weather.humidity = e.data.now.humidity

```

```

27         this.weather.vis = e.data.now.vis
28         this.updateTime = e.data.updateTime
29     },
30     fail: (e) => {
31         this.showToast('网络问题')
32         console.log(e)
33     }
34 })
35 // 天气预测
36 uni.request({
37     url: this.he_weather7d_api_url,
38     data: {
39         key: this.he_key,
40         location: this.location,
41     },
42     success: (e) => {
43         // 1
44         console.log(e)
45         if (!this.check('obj', e.data)) {
46             this.showToast('请检查url接口是否正确')
47             return
48         }
49         if (!this.check('state', e.data)) return
50         // 2
51         let daily = e.data.daily;
52         let res = {
53             categories:
54             [daily[0].fxDate.substr(5),daily[1].fxDate.substr(5),daily[2].fxDate.substr(
55             5),daily[3].fxDate.substr(5),daily[4].fxDate.substr(5),daily[5].fxDate.subst
56             r(5),daily[6].fxDate.substr(5)],
57             series: [{
58                 name: "最高温",
59                 data:
60                 [daily[0].tempMax,daily[1].tempMax,daily[2].tempMax,daily[3].tempMax,daily[4]
61                 ].tempMax,daily[5].tempMax,daily[6].tempMax]
62             },
63             {
64                 name: "最低温",
65                 data:
66                 [daily[0].tempMin,daily[1].tempMin,daily[2].tempMin,daily[3].tempMin,daily[4]
67                 ].tempMin,daily[5].tempMin,daily[6].tempMin]
68             }
69         ]
70         };
71         this.chartData = res;
72     }
73 })
74 // 空气质量
75 uni.request({
76     url: this.he_air_api_url,
77     data: {
78         key: this.he_key,
79         location: this.location,
80     },
81     success: (e) => {
82         console.log(e)

```

```
75         if (this.debug) console.log(e.data)
76         if (!this.check('obj', e.data)) {
77             this.showToast('请检查url接口是否正确')
78             return
79         }
80         if (!this.check('state', e.data)) return
81         this.air.category = e.data.now.category
82         this.air.aqi = e.data.now.aqi
83         this.air.pm10 = e.data.now.pm10
84         this.air.pm2p5 = e.data.now.pm2p5
85         this.air.no2 = e.data.now.no2
86         this.air.so2 = e.data.now.so2
87         this.air.co = e.data.now.co
88         this.air.o3 = e.data.now.o3
89     },
90     fail: (e) => {
91         this.showToast('网络问题')
92         console.log(e)
93     }
94 })
95 }
```

3.5 系统界面设计

系统主界面我们主要参考了手机自带的天气App的设计布局，使用了Vue框架和开源UI组件，大大提高了开发的效率，也提高了用户的体验效果。

4、结果分析

在实机上，我们的界面展示如下：



雾霾探测系统



参考资料

[Android 天气APP（一）开发准备android天气初学者-Study的博客-CSDN博客](#)

[HTML5 简介\(w3school.com.cn\)](#)

[uni-app官网\(dcloud.net.cn\)](#)

[地理/逆地理编码-API文档-开发指南-Web服务 API|高德地图API\(amap.com\)](#)

[实时天气 for API | 和风天气开发服务\(qweather.com\)](#)

[每日天气预报 for API | 和风天气开发服务\(qweather.com\)](#)

[实时空气质量 for API | 和风天气开发服务\(qweather.com\)](#)

[演示 - uCharts跨平台图表库](#)