项目说明文档-10组

1.项目概述

本次实验我们实现了一个简单的天气预报小程序，实现了查看本地天气情况、查询某一地区的天气、收藏某些地区的天气情况等功能。演示视频：https://www.bilibili.com/video/BV1bC4y1r7aV/

2.代码结构

```
│  .eslintrc.js
│  project.config.json
│  project.private.config.json
│  README.md            # 项目说明文档
│
├─cloudfunctions        # 云函数
│  ├─append             # 向数组的末尾增加一个元素，用于用户增加收藏的城市
│  │
│  ├─delete             # 删除数组中的一个元素，用于帮助用户删除收藏的城市
│  │
│  ├─getUserData        # 获取用户对应的地理位置数组
│  │
│  ├─getUserID          # 获取用户ID，作为每一位用户数据的标识
│  │
│  └─initUserData       # 用于初始化用户在数据库中的存储(若已初始化则无影响，可放心使用)
│
└─miniprogram
    │  app.js
    │  app.json
    │  app.wxss
    │  envList.js
    │  sitemap.json
    │  util.js           # 封装了一些工具函数，如位置获取、天气获取、地点封装等
    │
    ├─components         # 组件
    │  ├─CityItem        # List页中的列表项，使用时点击可跳转到相应地点的Weather页
    │  │
    │  ├─DayPart         # Weather页中的24小时天气模块，使用时点击可跳转到该地DayWeather页
    │  │
    │  ├─NowPart         # Weather页中的实时天气模块，使用时点击可跳转到该地NowWeather页
    │  │
    │  ├─WeatherIndices # Weather页中的天气指数模块
    │  │
    │  └─WeekPart        # Weather页中的7日天气模块，使用时点击可跳转到该地WeekWeather页
    │
    ├─images             # 图片资源，如天气图标、tabBar图标等
    │
    └─pages              # 页面
        ├─DayWeather     # 24小时天气详情页
        │
        ├─List           # 列表页(tabBar页)，可查看收藏列表、进入收藏城市的Weather页、搜索页
        │
        ├─LogIn          # 登录页(首页)
```

```
│
├─NowWeather    # 实时天气详情页
│
├─Search        # 搜索页，可输入文字搜索、重定位到当前位置、选择热门城市
│
├─Weather       # 天气页(tabBar页)，可查看天气、进入天气详情页、搜索页、收藏/取消收藏
│
└─WeekWeather   # 7日天气详情页
```

3.项目内容

3.1 用户信息存储

我们需要一个数据库，存储每一位用户收藏的城市。用户登录小程序时，数据库会根据用户的唯一标识id查询到用户收藏的城市，供列表页面使用。同时，用户也可以在列表页面增加或删除收藏的城市。本次基于获取用户ID以及数据库的增删改查设计了五个云函数，包括获取用户ID、初始化用户数据库、查询数据库、添加一个元素和删除一个元素。云函数的源码存储在cloudfunctions文件夹中。

3.2 用户登录

用户打开小程序后，进入LogIn页面。我们在这里要获取用户ID、用户位置，用户允许授权后直接进入到当前位置的Weather页面中（若误点击拒绝，可在右上角的设置中手动开启定位权限）

3.3 weather页面的设计

用户将位置授权给我们后，我们就进入了weather页面，这是本次小程序比较重要的一个页面，用于查询某一位置的天气基础信息。这一功能可以分为一下几步。

1. 获取该地的天气信息
   我们使用了和风天气的API，并且注册了多个key（由于免费版有访问次数限制，测试时如果访问次数超过了一个key的最大限额，会导致无法获取天气信息，此时请在util.js文件中切换其他的备用key）。我们将位置信息和key一起上传后，和风天气会返回相应的天气信息（这里我们采用了await Promise.all()让各种天气查询请求并行、且保证后面的过程能使用到前面获取的数据）
2. 组件的运用
   我们将Weather页面拆分出NowPart、DayPart、WeekPart、WeatherIndices四个组件，分别展示了当前位置的实时天气、二十四小时天气、七日天气以及天气指数，每一部分都配备有相应的onclick函数，用户想要了解某一部分的详细信息后就可以点击相应部分，进入到对应的组件，组件内部展示了天气的详细信息
3. 其他微小功能
   收藏：绑定onClickToStar函数，如果按钮呈现红色表示该地区已收藏，呈现白色表示未收藏；用

户可以通过点击收藏按钮切换该地区的收藏状态。
城市：绑定onClickToSearch函数，显示Weather页的数据对应的城市名，点击可跳转到搜索页面。

4. 我们还实现了底部导航栏，选择Weather页和List页作为tabBar页面。

3.4 列表页面

通过底部的导航栏我们就可以进入到列表页面，也就是list页面，页面最上方是搜索按钮，通过这里我们可以跳转到搜索页面，然后下面是展示用户收藏列表（将列表项封装成了CityItem组件），展示用户收藏的城市，点击城市就可以跳转到该城市的详细天气信息，并且可以决定是否将其加入到列表页面里

3.5 搜索页面

我们的搜素页面采用调用和风天气的搜索功能，进行模糊搜索，如果用户输入了准确的地理位置，系统将提供给对应地区的天气；如果用户输入了一个模糊的地理位置，系统将会匹配到一个最相近的地理位置；如果该地区系统没有相关的天气信息，系统会返回“抱歉，暂时没有你查询的地区的天气信息“；如果找不到该地区，系统会返回“查询的地区不存在，请规范输入后重试”，其他错误情况系统也会有相应的返回语句。同时搜索界面可以重定位到当前地点或一些热门城市。