Weather App

CS426 - Mobile Device Application Development

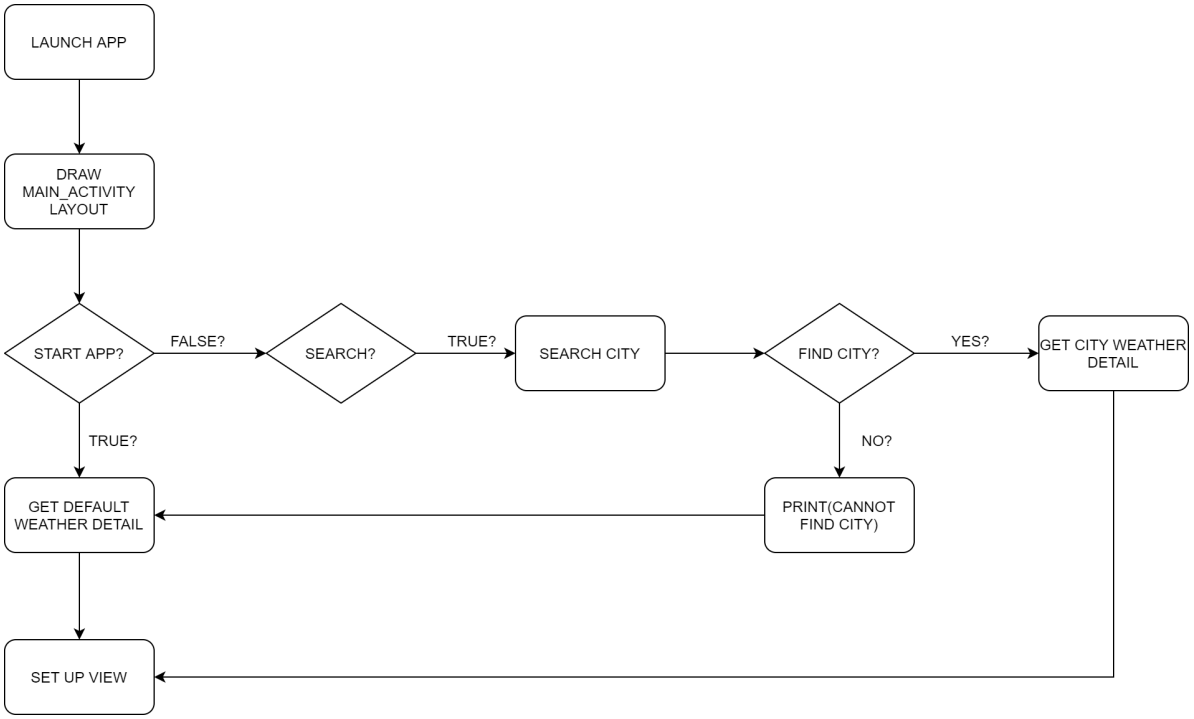
19125063 - Châu Hữu Phát - 19CTT2

19125119 - Trương Lăng Trường Sơn - 19CTT2

I. App structures: The application's structures consist of 2 activities: Main Activity and Search Activity, and each activity has a separate layout.

1. Main Activity:

- Below is the simplified structure of the Main Activity. This activity will draw the main layout and call the API to get the needed information.



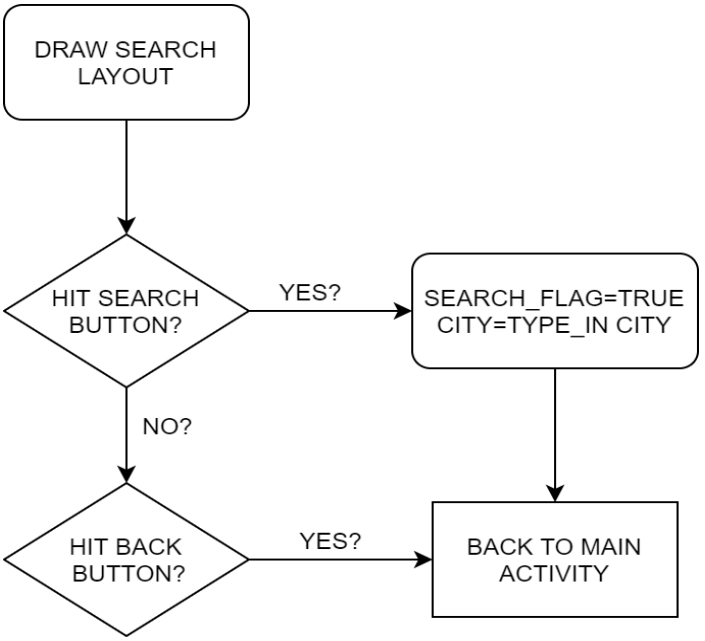
- In terms of API, we chose to use the API provided by Openweather because it is free of charge and one of the most common API for weather forecasting.
- The default city is Ho Chi Minh City and the reason for our choice will be presented in the next section.
- In brief, we will collect 5 blocks of weather information, one of which is for the current time, and the other 4 are used for 4 3-hour-period pieces.

Every block of information includes the temperature, the humidity, and a brief description of the weather condition used for determining the icon.

- There is also an intent built in the main/current block. This intent, when clicked, will move the users to the main website of OpenWeather for further detail.
- For the searching algorithm, we take advantage of the API as this API can look for the weather conditions based on the city. Therefore, the API is equipped with a decent search tool.

2. Search Activity:

- Below is the simplified structure of the Search Activity. This activity will simply collect the city name for the Main Activity.



- This activity will be triggered as an intent if the users clicked the search button from the Main Activity.
- The users can either click the Left Arrow Button to get back to the Main Activity or click the Search Button. When it is licked, the data in the Edit Text box will be passed to the city variable and the search_flag will be marked as true.
- The search algorithm is handled in the Main Activity.

3. Major Functions:

```
@Override
protected void onResume() {
    super.onResume();
    // if users open the app, it will search weather in the default location (HCMC)
    // then it will not search default location anymore
    if(start) {
        getSelectedWeatherDetail(defaultCity);
        start = false;
    }
    // after openning the app (after searching default location)
    else {
        Intent intent = getIntent();
        if (intent != null) {
            // the variable "searched": check if user is searching or not
            // if "searched" is false, the app will get the data from local storage and not search for new location
            if(!(intent.getBooleanExtra( name: "searched",  defaultValue: false))) {
                setUpViews(new Gson().fromJson(local.getString( s: "data",  s1: null), WeatherDetail.class));
            }
            // if "searched" is true and the input from user is not empty, then the app will search for new location
            else {
                if(intent.getStringExtra( name: "city") != null) {
                    getSelectedWeatherDetail(intent.getStringExtra( name: "city"));
                }
            }
        }
    }
}
```

```
private void getSelectedWeatherDetail(String city) {
    // calling API to get weather infomation
    ApiService.apiService.getWeather(city,  units: "metric",  cnt: 5,  appid: "8e8cf1deb446983e8a0bd9e743170fde").enqueue(new Callback<WeatherDetail>() {
        // response
        @Override
        public void onResponse(Call<WeatherDetail> call, Response<WeatherDetail> response) {
            WeatherDetail detail = response.body();
            if(detail == null) {
                Log.d(TAG,  msg: "onResponse: detail = null");
                Log.d(TAG,  msg: "onResponse: "+new Gson().fromJson(local.getString( s: "data",  s1: null), WeatherDetail.class).city.name);
                detail = new Gson().fromJson(local.getString( s: "data",  s1: null), WeatherDetail.class);
                // if the input from user can not be found by API, the app will display "Can not find {input}."
                AlertDialog.Builder builder = new AlertDialog.Builder( context: MainActivity.this);
                builder.setMessage("Can not find "+city+".")
                        .setPositiveButton( text: "Ok",  listener: null)
                        .show();
            } else {
                // store data to local storage
                String data = new Gson().toJson(detail);
                local.edit().putString( s: "data", data).commit();
            }
            setUpViews(detail); // setup the views
        }
        // failure
        @Override
        public void onFailure(Call<WeatherDetail> call, Throwable t) {
            Log.d( tag: "DEBUG", t.getMessage());
        }
    });
}
```

II. App functions:

- Our application provides 2 main functions: weather forecasting and searching weather conditions for a city. Due to the limitations of the free API provided by OpenWeatherMap, there are two approaches we could

consider for our application. The first approach uses the user's current location (longitude and latitude) to find the weather conditions while the second approach provides the users with 5 pieces of weather information (1 piece for the current time and 4 remaining ones for the 3-hour-period forecast). We believe that the users can search their cities with our search tool to get the information. Therefore, we chose the second approach.

- Weather forecast:
  + Initially, after launching the application, it will provide the users with 5 pieces of weather information, one of which is on top and based on the current time while the remaining ones are for every 3 hours time mark.
  + The default location is Ho Chi Minh City.
  + The weather information displayed is temperature, humidity, and condition description.

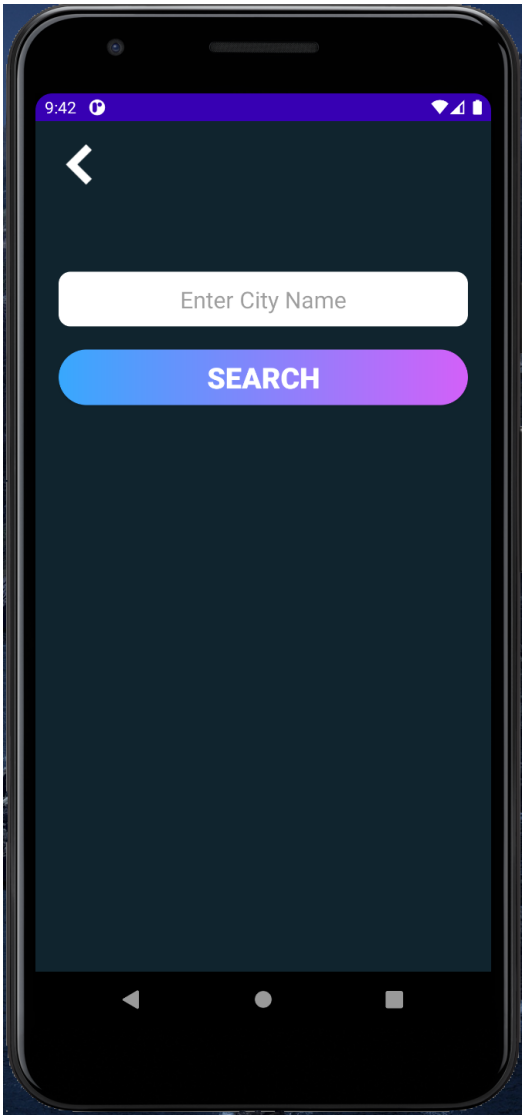
Default weather in HCMC

Search weather in New York City

Search weather in Holland

Search weather in Brazil

- Search weather conditions of a city:
  - + At the bottom of the screen, there is a search button that can be used for searching the weather conditions of a city.
  - + Clicking the Search button will get the users to the search page.
  - + From the search page, the users can type in the city they want to search weather information for.
  - + The search takes advantage of the API tools of searching weather conditions based on the city name.



Search Activity

Input “AbCxyZ”