ITIS/CS 4180 – Mobile App Development
Homework 03

Basic Instructions:

1. In every file submitted you MUST place the following comments:
   a. Assignment #.
   b. File Name.
   c. Student Full Name.
2. This is an individual assignment, each student is expected to work alone and submit their own work.
3. Your assignment will be graded for functional requirements and efficiency of your submitted solution. You will loose points if your code is not efficient, does unnecessary processing or blocks the UI thread.
4. Please download the support files provided with this assignment and use them when implementing your project.
5. Create a zip file which includes all the project folder, any required libraries, and your presentation material.
6. Submission details:
   a. You should submit the assignment through canvas: Submit the zip file.
7. **Failure to follow the above instructions will result in point deductions.**

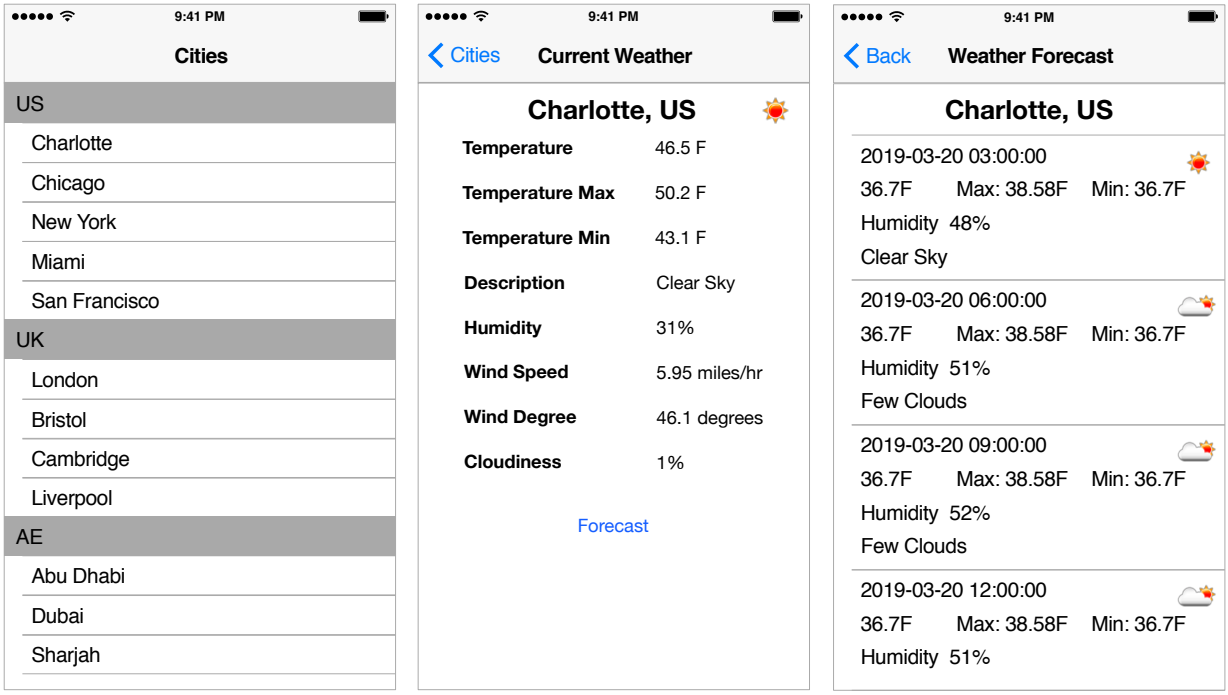
Homework 03 (100 Points)

In this assignment you build a weather application using the OpenWeatherMap api.

API Details:

The current weather api should be used by this app:

1. Create an account at https://openweathermap.org/ and get an api key, which will be used to make the api calls.
2. You should use the Weather API, check the documentation for that is provided by the OpenWeatherMap APIs
   a. For the city and country information you should use the data provided in the support files.
   b. All the retrieved information should be in the imperial units. Check the api documentation to retrieve the data in this unit. All data should be retrieved in JSON format.
   c. The Current Weather API, check the documentation provided at https://openweathermap.org/current
   d. The 5 Day Weather Forecast API, check the documentation provided at https://openweathermap.org/forecast5
   e. To load the Weather Icon check the documentation provided at https://openweathermap.org/weather-conditions



(a) Cities List (b) Current Weather (c) Weather Forecast

Figure 1, Application Wireframe

Part 1 : Cities List View Controller (0 Points)

The interface should be created to match the UI presented in Figure 1(a). The requirements are as follows:

1. This ViewController should be embedded in a navigation controller to provide the transition to the application ViewController.
2. This ViewController should use the cities dictionary in the provided AppsData class.
    a. You should use the country code as the tableview section.
    b. Under each section list the cities under this country code.
3. Clicking on a row item should segue to the Current Weather ViewController and should pass the required information to it.

Part 2 : Current Weather Controller (10 Points)

The interface should be created to match the UI presented in Figure 1(b). The requirements are as follows:

1. This ViewController should retrieve the current weather information for the selected city. You should use the Current Weather API.
    a. You should integrate AlamoFire library into your project and use it to make the API requests.
    b. Create a Weather class that should hold the data returned in the JSON API response.
2. This View Controller should display the Weather Icon based on the retrieved weather information. You should load the weather icon based on the documentation provided at https://openweathermap.org/weather-conditions
    a. You should integrate the AlamofireImage library into your project to load the weather image icon from the derived icon url. https://github.com/Alamofire/AlamofireImage
3. Display the weather information and shown in Figure 1(b).
4. Clicking the “Forecast” button should segue to the Weather Forecast View Controller, and should pass the required information to it.

Part 3 : Weather Forecast Controller (90 Points)

The interface should be created to match the UI presented in Figure 1(c). The requirements are as follows:

1. This ViewController should retrieve the weather forecast for the selected city. You should use the 5 Day Weather Forecast API.
    a. You should integrate AlamoFire library into your project and use it to make the API requests.
    b. Create a Forecast class that should hold the data returned in the JSON API response. You should create an array of the retrieved Forecast items.
2. This View Controller should display the retrieved forecast in a TableView as shown in Figure 1(c). Note that each cell should display, the date/time, weather icon, temperature, max temp, min temp, humidity, and the description.
    a. Use the AlamofireImage library to load the weather icon for each of the tableview cells.