

This assessment contains materials that may be subject to copyright and other intellectual property rights. Modification, distribution or reposting of this document is strictly prohibited. Learners found reposting this document or its solution anywhere will be subject to the college's Academic Integrity policy.

Assignment 3

Testing Environment

Your application will be tested on an **iPhone 11** simulator
Before submitting, please ensure your application displays correctly on this device.

Submission Checklist

In the course dropbox, submit a zip file containing your complete project code.

- The zip file should be named: **a3-studentname-username.zip**.
- Replace **studentname** and **username** with your name and username
- **No .7z or .rar files accepted.**

Academic Integrity

- This is an individual assessment.
- Permitted activities: Usage of Internet to search for syntax only; usage of course materials
- Not permitted:
 - Communication with others (both inside and outside the class)
 - Discussion of solution or approaches with others
 - Searching the internet for full or partial solutions
 - Sharing of resources, including links, computers, accounts

App Description

You are to create a **two screen** IOS application that allows the user to search for a weather report for their current city.

1. **Screen 1 - Current Weather Report:** This screen displays weather information at the device's current location.
2. **Screen 2 - Weather History:** This screen displays a list of previous weather information.
3. Screen 1 and Screen 2 must be organized using a **Navigation Controller**. Screens must be properly labeled.
4. The user can access Screen 2 by clicking on a button on Screen 1's Navigation Bar, as follows:



Screen 1 Implementation Details

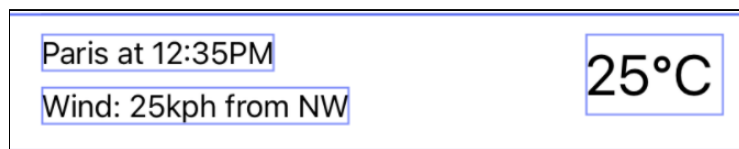
1. When the screen loads, the app must use location services to fetch the current location of the device.
2. Each time the device's location updates, the app should perform reverse geocoding to obtain the city name for the new location
3. Using the obtained city name, the screen must display:
 - a. The name of the city the device is currently located in
 - b. The weather information for the current city, specifically:
 - Current temperature in Celsius (the actual temperature, not the "feels like" temp)
 - Wind speed in kph,
 - Direction of the wind

This assessment contains materials that may be subject to copyright and other intellectual property rights. Modification, distribution or reposting of this document is strictly prohibited. Learners found reposting this document or its solution anywhere will be subject to the college's Academic Integrity policy.

4. The screen must contain a button called SAVE REPORT. If the user taps this button, the current weather report must be displayed in the list on Screen #2.

Screen 2 Implementation Details

1. When Screen 2 loads, display any “saved” weather reports in a TableView
2. Each row of the tableview must display
 - a. The city where the weather report was taken
 - b. The time the report was saved (ie: the time the user pressed the SAVE REPORT button on Screen #1)
 - c. The temperature in C
 - d. Windspeed and direction
3. Each row should have the following user interface:



5. The user must be able to navigate “back” to the previous screen using the < **Back** button in the menu bar.

Data Modeling Requirements

1. Weather related data must be represented in code as a custom **Weather** class
2. There is no data persistence requirement for this application. When the app closes, it should forget any previous weather data.
3. Data should be passed between screens using an appropriate data type. Do not use any persistence layers, such as files or SharedPreferences.

This assessment contains materials that may be subject to copyright and other intellectual property rights. Modification, distribution or reposting of this document is strictly prohibited. Learners found reposting this document or its solution anywhere will be subject to the college's Academic Integrity policy.

User Interface Requirements

Your user interface must be reasonably pretty.

Weather Data Requirements

For this application, you must use the WeatherAPI.com API endpoints. Specifically:

- Signup for a free account on WeatherAPI.com
- Use the following endpoint in your application:

```
https://api.weatherapi.com/v1/current.json?key=YOUR_API_KEY&q=CITY&aqi=no
```

- Replace YOUR_API_KEY with the API key associated with your WeatherAPI.com account
- Replace CITY with the city you are searching for weather in.

You may use the API Explorer provided by WeatherAPI to learn about possible response formats.

You are **not required** to use the Singleton or Observable patterns.

END OF ASSESSMENT