

# **Final Project Report**

**Joshua Boyd**

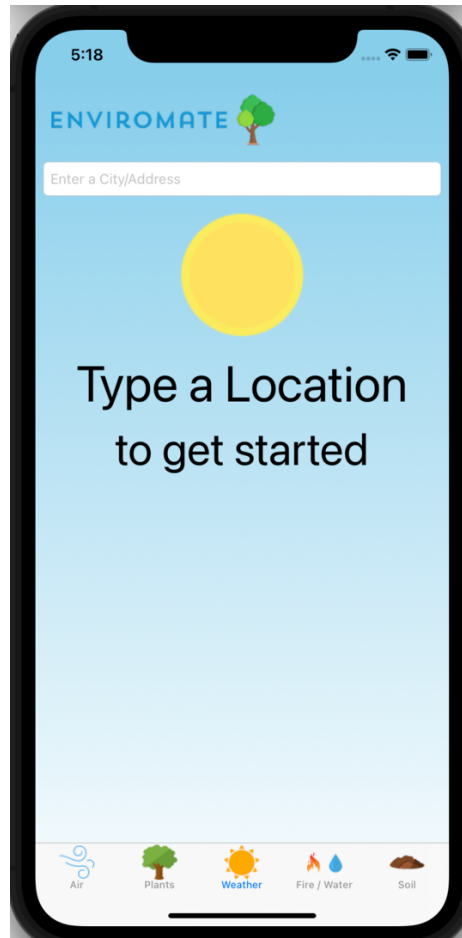
**COMP 5970**

# Purpose

The purpose of my application is to display environmental data for a given city, state, or address. The name of the application is EnviroMate and it uses two application programming interfaces (API) to provide it's services. Environmentalists could find the app useful when they need to know certain specifics about the location they are in, such as the air quality, weather, or soil temperature. The app provides data for both the continental United States, as well is international locations.

# How to use

When the app first starts up, the user will be met with the weather view (Figure 1).



**Figure 1: Initial App View**

The user will need to type in a location into the text-field to get the app to work. There are two triggers to alert the app to retrieve environment data. The user can either type in a location and press the return key on the phone keyboard, or after typing in a location, touch any location on

the screen. The app color gradient will change colors and darken if the system time is 5pm or later. Additionally, the app will also display a moon instead of a sun if it is past 5pm.



**Figure 2: Weather View After Data Population**

After the user has entered a location and data has been retrieved, the user interface is updated and populated with the data. As shown in figure 2, some of the data includes the latitude and longitude of the location, weather summary, and temperature. Now that the data has been retrieved, the other tabs along the bottom of the screen have also been populated.

# How the app works

The app is designed to only be ran in portrait mode. However, layout constraints have been added so that the app is compatible with the following phones:

- iPhone 13 Pro Max
- iPhone 11 Pro Max
- iPhone 11
- iPhone 13 Pro
- iPhone 13 Mini
- iPhone 8 Plus
- iPhone SE (3<sup>rd</sup> Generation)

The app works by accessing two APIs. The first API is Geoapify geocoding API at <https://apidocs.geoapify.com/playground/geocoding>. Once the user enters an address or location, it is sent to the geocoding API and the latitude and longitude coordinates are retrieved for that area. This latitude and longitude data is passed into the environmental API Ambee, located at <https://docs.ambeedata.com/>. From this API, the app has access to environmental information such as air quality, pollen information, weather, fires, greenhouse gas emissions and more.

For security, the API keys are stored in Config.xcconfig and should not be included in version control. Each view is stored as a tab bar item within the tab bar controller. Most of the

logic for the program is in the TabBarController file. This file contains the code that reaches out to both APIs, as well as holding the data and updating the labels for each view. If a user enters a location that is not found by the Geoapify API, then an error message will be displayed for the user (Figure 3).



**Figure 3: Error Displayed for Location Not Found**

Since the labels on each view do not technically exist until they are first displayed, a major issue during development was how to update all the data when the user enters a location. The work around for this was to only update the labels for the view that the APIs were called

from. However, since all the data for each view is retrieved, it is stored inside of a dictionary.

When a new tab is selected, the labels for that tab are populated on the fly when the view switches. This is accomplished in the `didSelect()` function for the `TabBarController`.