# Mobile Computing - iOS (or Android) Final Project Observations

## Team Members:

Rohith Reddy Avisakula,

Soujanya Janapatla

## Team Name:

Titans

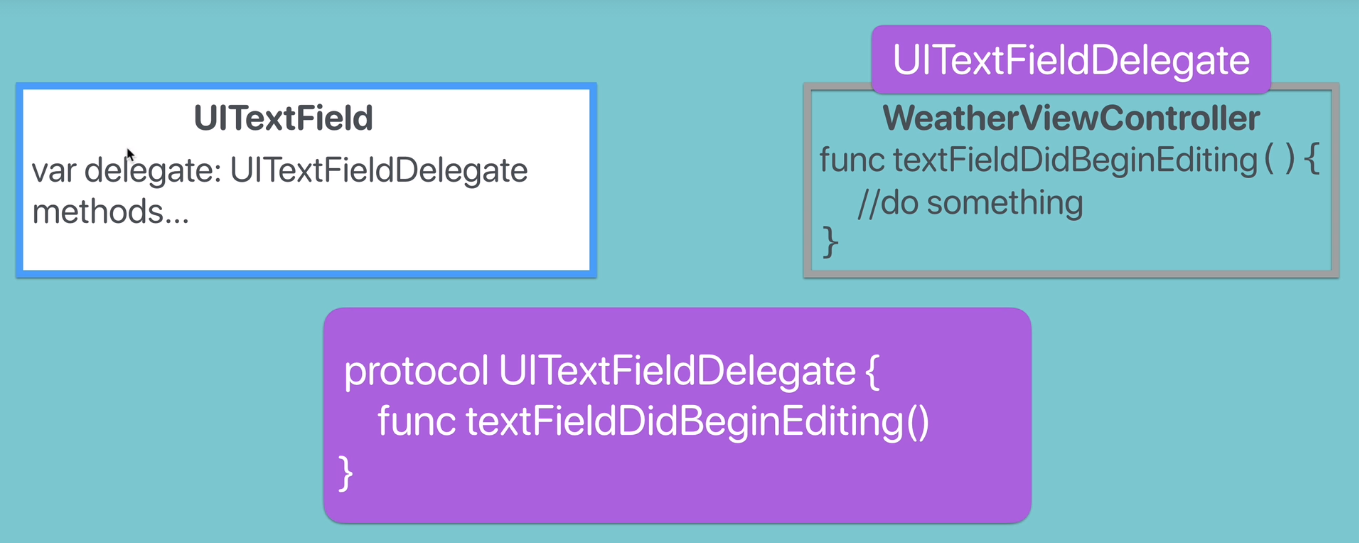
## Environment for Creating HTTP Request

XCode 12.2

Swift 13.01

## The Delegate Design Pattern

The definition for a design patterns is a simply just a proven solution to a common problem. We are trying to design a template code for UI Text field, they need not define all the properties right from scratch.



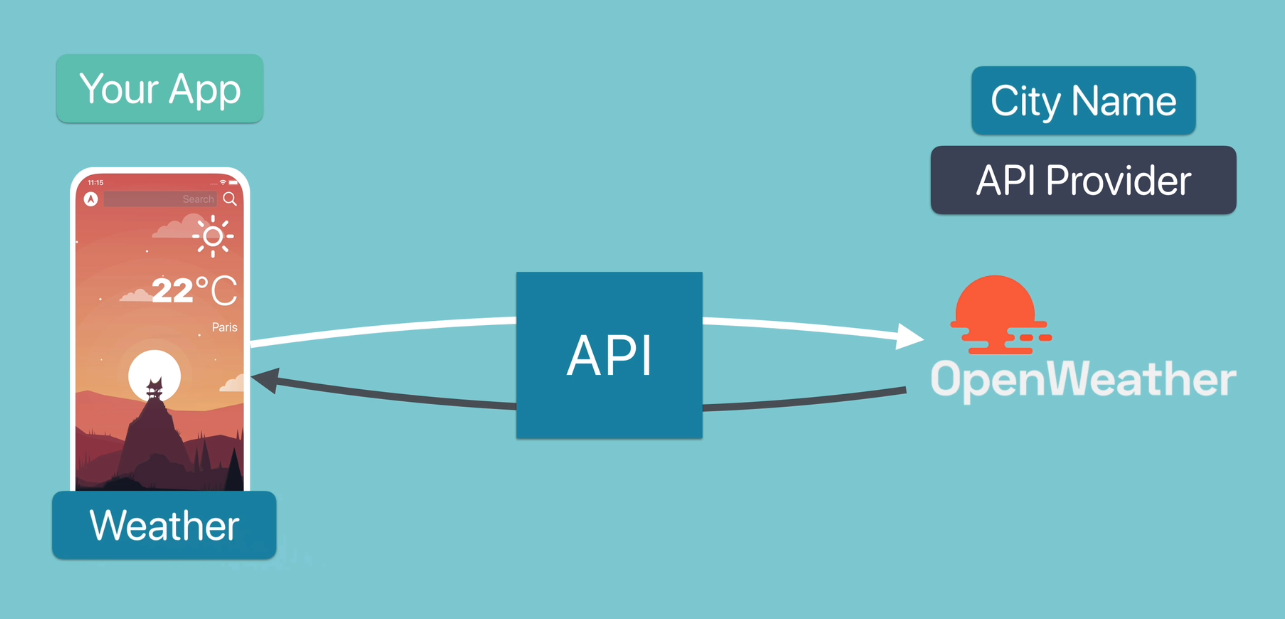
We decide to build a WeatherViewController to detect when the text field start being edited by the user. We need to define a WeatherViewController object and see the call text field begin to update and check if the methods sends data in respond.

A possible solution that we have using these protocols, Reusable UITextField by apple and WeatherViewController need to tell the text field when the user has stop typing in the text by adopting this protocol in the main Controller.

## Use the URL session for Networking

Work with API's to fetch live weather data across the internet. An Application Programming Interface (API) is a set of commands, functions, protocols and objects to create a software and interact with external systems.

If we head over to OpenWeatherMap we can look at the documentation at the API's that they provide, all we need to do is to signup for the API Key and use the Key to connect to the API.



The API Keys secion provides a key to allow yourself fo access to the service. The data that we need s Current Weather Data for a location by city name, city iD, Longitude and Latitude. Lets try to structire our API with the API Key:

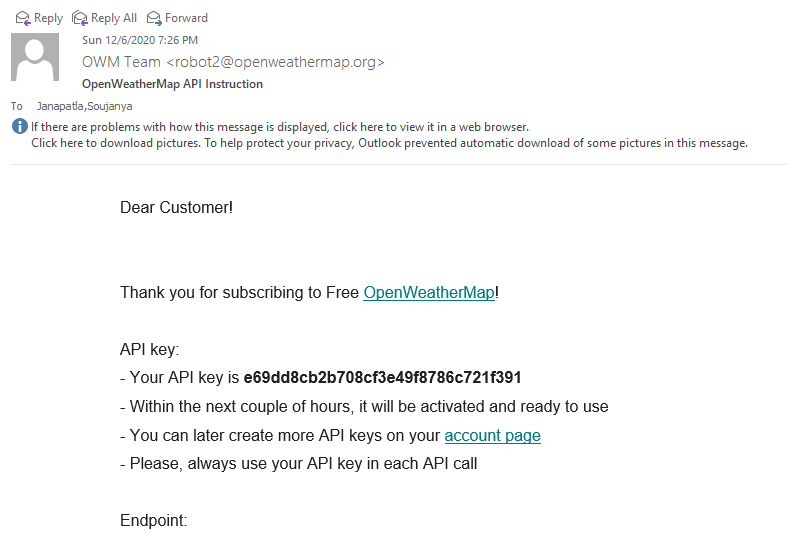


Parameters:

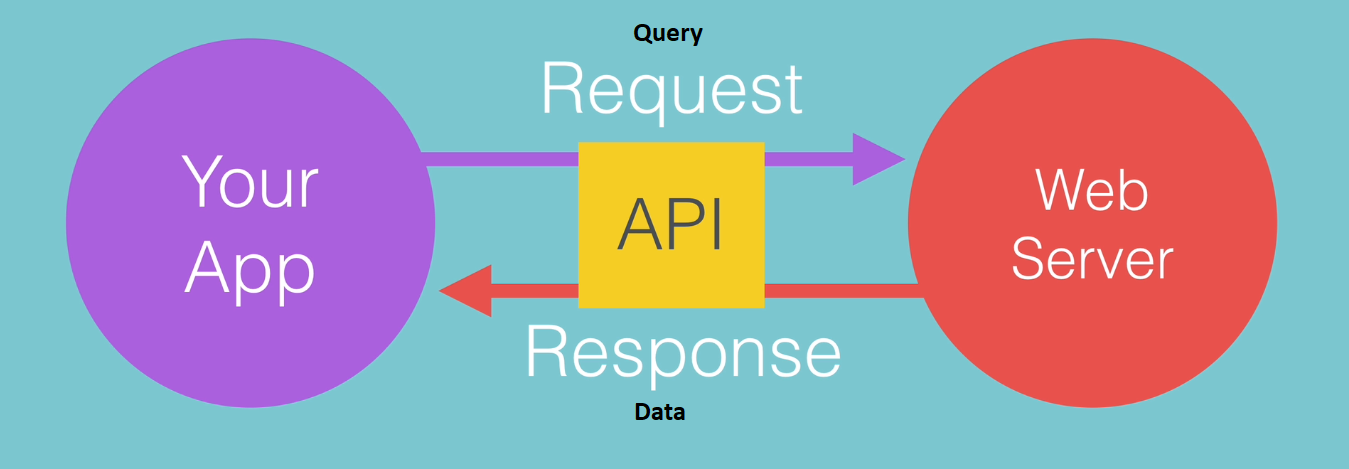
q city name and country code divided by comma, use ISO 3166 country codes

Response:

{"coord":{"lon":2.35,"lat":48.85},"weather":[{"id":701,"main":"Mist","description":"mist","icon":"50n"}],"base":"stations","main":{"temp":1.47,"feels\_like":-1.15,"temp\_min":0,"temp\_max":2.78,"pressure":999,"humidity":93},"visibility":3500,"wind":{"speed":1,"deg":0},"clouds":{"all":84},"dt":1607304959,"sys":{"type":1,"id":6550,"country":"FR","sunrise":1607326219,"sunset":1607356466},"timezone":3600,"id":2988507,"name":"Paris","cod":200}

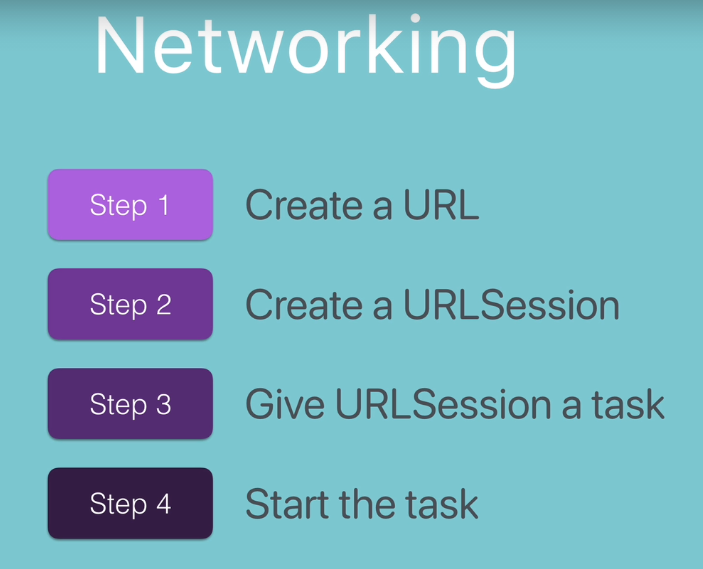


The Final URL that queries for the weather in Paris, we manage to get the hold of that by simply putting it on the browser and getting the browser to fetch the data back, this process in programming is called networking.



The App is talking to a web server, in other case is the OpenWeatherMap. A web server is an application in itself, and we are trying to talk to the web server of open weather map and make a request to the weather data, we pass a query for the data we are interested in, example name of the city: Paris. You are passing over an app ID, if it’s a valid app id, it responds back with the data.

There are couple of steps that we need to go through:



1) We have to create an actual URL object,

2) Create URL session which is going to be the object that’s going be doing the networking just like our browser.

3) We give this URl session a task, just like we put the URL in the browser URL bar, and gave our browser a task to do, which is fetching the data from that particular source.

The final step to actually start the app, and then when we hit the “Enter” that triggers the entire networking process and hopefully we get the data back.

## Work Distribution:

Rohith:

1. Writing the Delicate protocol functions
2. Adding the appropriate vector images for the weather conditions

Soujanya:

1. Worked on the OpenWeatherMap api integration
2. Defining the model and weather manager classes and Finally configuring the key wit the api’s.

## Github URL

<https://github.com/jscodebit/GeoWeather>