ID: 180917

Course: Application Development for iOS

Date: May 20, 2019

Class Summary

In the class on the date specified above we delved further into IOS’s location ecosystem, specifically the Apple MapKit framework which enables developers to display satellite or map imagery directly within the application’s interface across a wide array of systems. Annotations and overlays such as pins or route highlighting can be implemented on the map to enable additional functionality. Maps can be implemented on IOS devices using the MKMapView. This immediately loads Apple Maps within the view without any additional coding.

We were also taught about making HTTP connections using the general API that are provided by the IOS Sdk. Using these API’s additional functionalities can be implemented into an app such as retrieving data from a server or saving files to disk. The build in library that is used to make HTTP requests as of Swift4 is URLSession which simplifies the prior method of making requests into easy to use methods. There are also third-party libraries for making HTTP requests using swift such as popular Alamofire which provides a simplified interface for making requests with options such as payloads.

The combination of MKMapView and HTTP requests allows for the logging of users coordinates on a remote server which aids in optimizing the in-app experience by analyzing the data gathered and making suggestions to users.

For the class exercise we were required to:

1. Compile, run, and understand the Maps service iOS app

The app was compiled and run. A picture of the running application is provided in this project’s directory.

1. Research and compare how maps service work in iOS/Android

The map services in both Android and iOS use a combination of GPS coordinates and proximity from cell towers located within range in order to provide a very accurate approximation of where the user is located.

1. Add an option to choose standard/satellite/hybrid map layers

This was implemented with the help of a few classmates

1. Try different map/location services in iOS, e.g. Tencent LBS
2. Compile, run, and understand the HTTP/JSON iOS app
3. Research and compare how HTTP/JSON work in iOS/Android

In android JSON string that is received from a server should be serialized using a Class. This can then be carried out quickly using the GSON

1. Try Alamofire for HTTP connection in iOS

Almofire was included into the app using pods

1. Try different HTTP connection methods

URLSessions were used to make requests to the server

1. Try decode more JSON fields, including your current web service

A method was created to make requests to the server. It is provided in the TaskFetcher.swift file.