Week 2 The Geo-location app

**Goal**: Allow a parent to set up a geo-fencing monitor for their child. The parent sets a point and a radius within which the child is supposed to be. If the child moves out of that, the parent is notified.

**App user interface:** There are two apps, parent and child.

The parent app does two things

• Create parent userid

• Set preferences for the parent

• Get status

The child app does only one thing – once activated with the parent’s userid, it constantly sends its location to a server.

**Web Version of the above app**

Visit http://protected-wildwood-8664.herokuapp.com/ and explore the various options. It is a fully working web version of both the parent and child apps. Our goal is to create iOS apps that communicate with the backend using JSON.

**URI’s for JSON Services required to build the two apps explained above**

• Create parent userid

• Assuming you have three properties userID, latitude and longitue

• Create JSON string from this Objective-C dictionary using NSDictionary to JSON conversion API.

• NSDictionary \*userDetails = @{@"utf8": @"✓", @"authenticity\_token":@"EvZva3cKnzo3Y0G5R3NktucCr99o/2UWOPVAmJYdBOc=", @"user":@{@"username":self.userID,@"latitude":self.latitude,@"longitude":self.longitude,@"radius":self.radius}, @"commit":@"Create User", @"action":@"update", @"controller":@"users"};

• Convert to JSON String using NSDictionary to JSON conversion API

• HTTP POST above JSON to  /users

• Set preferences for the parent

• Create JSON from this dictionary

• NSDictionary \*childDict = @{@"utf8": @"✓", @"authenticity\_token":@"EvZva3cKnzo3Y0G5R3NktucCr99o/2UWOPVAmJYdBOc=", @"user":@{@"username":self.userID,@"current\_lat":self.latitude,@"current\_longitude":self.longitude}, @"commit":@"Create User", @"action":@"update", @"controller":@"users"};

• HTTP PATCH request – send above JSON to /users/username

• Get status

• Send HTTP GET request to /users/username.json

• Convert JSON to NSDictionary and extract value for “**Is in zone”**

**Technologies we’ll explore**

• HTTP, JSON and some debugging tools

• HTTP calls using Objective-c

• Using the device GPS

• Synchronous vs. asynchronous calls. Pros and cons.

• Underlying design patterns for asynchronous

• XCode interface builder

• Lifecycle of a simple objective-c app

• User interface development for iOS

• JSON-Obj-C conversions

**Resources**

**HTTP related tools**

http://curl.haxx.se/

http://www.wireshark.org/

**Geolocation**

http://stackoverflow.com/questions/6894624/how-can-i-get-gps-location-in-iphone

Official doc https://developer.apple.com/library/mac/documentation/CoreLocation/Reference/CoreLocation\_Framework/CoreLocation\_Framework.pdf

**Networking**

Simple tutorial http://codewithchris.com/tutorial-how-to-use-ios-nsurlconnection-by-example/

Official docs

https://developer.apple.com/library/ios/documentation/Cocoa/Conceptual/URLLoadingSystem/URLLoadingSystem.pdf

**General App development**

https://developer.apple.com/library/ios/referencelibrary/GettingStarted/RoadMapiOS/RoadMapiOS.pdf