|  |
| --- |
| Northern Arizona University |
| Technology Feasibility |
| CS 476 – Localization |

|  |
| --- |
| Blayne Kennedy, Kimi Oyama, Daren Rodhouse, Chihiro Sasaki |

Table of Contents

[Introduction 2](#_Toc342819201)

[Technology Overview 2](#_Toc342819202)

[Apache Tomcat Server Framework 2](#_Toc342819203)

[RESTful Web Service 2](#_Toc342819204)

[HTTP 2](#_Toc342819205)

[iOS, X Code, Objective C 3](#_Toc342819206)

[Android 3](#_Toc342819207)

[JAVA 4](#_Toc342819208)

[Git and GitHub 4](#_Toc342819209)

[Google Maps API for Android 4](#_Toc342819210)

[Google Maps API for iPhone 5](#_Toc342819211)

[Android Location 5](#_Toc342819212)

[RestKit 6](#_Toc342819213)

[MySQL 6](#_Toc342819214)

[Technology Integration 6](#_Toc342819215)

[Proof of Feasibility 8](#_Toc342819216)

[Sources 9](#_Toc342819217)

[Appendix A: 10](#_Toc342819218)

[Appendix B: 10](#_Toc342819219)

# Introduction

This report is an overview of the technologies that are going to be utilized in this project.

# Technology Overview

The back end technologies involved in this project are the Apache Tomcat Server Framework, a RESTful web service, and HTTP. The front end technologies include Java for programming an Android application, X Code for programming the iPhone application, and PHP for the website. The technology most likely to cause trouble is the Android application because that is a new aspect of the system; the other technologies are already implemented and work together, we are just adding to them.

Apache Tomcat Server Framework

Apache Tomcat is an open source implementation of a web server, mainly the Java Servlet and JavaServer Pages technologies. It is developed by the Apache Software Foundation (ASF), and it can also be downloaded from their main website [#].

The database of MCM is hosted on the Apache Tomcat server framework (need to double check with Josh).

The MCM network is based on the Apache Tomcat server framework, and thus the localization features of MCM will also utilize Tomcat.  By using Tomcat as the server framework, it will fulfill the requirement that the new feature will be easily integrated into the existing infrastructure.

RESTful Web Service

A RESTful (Representational State Transfer) web service is a style of architecture that can handle clients using different programming languages by focusing on the system resources and how they are transferred over HTTP. The web service does this by explicitly implementing HTTP methods, being state independent, using directory structure-like URIs, and using XML or JSON to transfer data. These attributes make the web service platform, client, and middleware independent. Using explicit HTTP methods reduces the likelihood of side effects. A stateless web service utilizes complete, independent requests that do not rely on context or state, which allows the server(s) to handle the different components of the request more efficiently than waiting for other, smaller, requests to complete. Using a directory structure like URI makes the web service generic and predictable. This means it is more user friendly. Using a single standard language to transfer data allows all clients to use the web service effectively, regardless of the language they are using on the user end [%].

HTTP

HTTP is a common web protocol used to transfer data between clients, websites, and servers, databases. HTTP utilizes multiple methods to query or send data such as PUT, GET, and POST[$$$].

## iOS, X Code, Objective C

iOS is the mobile operating system for Apple. Currently the official ios is 6.0.1 running on iPhone 3GS and up, we will be developing for iOS 6 for the iOS part of our project. ios allows for an array of new functions that will allow us to apply the needed functions into our application such as notifications and better integration with social networks besides better security and other functions. Our involvement with iOS will be using the IDE xCode which’s mine API is Cocoa using the primary language Objective-C.  
  
xCode is the IDE apple provides for all development in both OS X and iOS. It supports an array of languages such as C, C++, and Java; However the main graphics is written in Objective-C but can be done in Java as well. It provides an emulator and benchmark tool as well as a suite of other tools that allows a developer efficient and clean work. The main reasons we are going to be using this is for developing our iOS application and any other apple related software.   
  
Cocoa is the object oriented API for Apple’s OS and iOS. It’s environment is used mainly using Python, Perl or some other scripting language. It is the main API used for setting up application’s user-interface and mapping the overall functions of the application to the interface. Our use with this will be using it in xCode’s IDE for the user-interface of our mobile application and all the work in the background of the program. We will also have to use this in any instance for writing code for Apple’s products.  
  
Objective-C is a high level, object oriented programming language that is the main language used by Apple in all their development. The structure of the language is just a layer over C with object-oriented features implemented from Simpletalk-style messaging. Our need for this is to interlay all the functions of our IDE (xCode) and our API (Cocoa) to have a working program.

Android

Android is a mobile operating system (OS) utilized by numerous different smartphones. The OS is based off of the open Linux Kernel. Android is open source so it is possible for developers to customize it the way they choose. Due to this, it is possible for two phones both using Android to have differing graphical user interfaces (GUI).   
  
Android phones usually come with a few pre-installed applications but they also support third-party programs. These third-party applications are created using a free Android software developer kit (SDK) and are written in the Java programming language. The applications can then conveniently be distributed using the online Android Market [^].

## JAVA

Java is a high-level programming language created by Sun Microsystem in the early 1990’s. It has a syntax that is similar to C and C++, however, unlike C or C++, Java is strictly object-oriented. It is a general purpose, class based language and was designed to be simple so that it could be utilized by a large number of programmers.   
  
There are no untested features included in Java as it was designed to be a production language rather than a research language. The Java language is both strongly and statically typed. This makes it possible to determine whether an error must be caught at compile-time or at run-time.  
  
Java has automatic storage management and has a garbage collector to help manage it. Java’s garbage collector helps avoid safety problems seen in C and C++ that stemmed from explicit deallocation. Java also increased safety by disallowing unsafe constructs such as array access without index checking [\*].

## Git and GitHub

Git is a version control service that allows the user to maintain any file structure they want. It is a free service that the user can install and run locally. It provides a bash shell and a GUI that allows the user to manage their local repository[%%%].

GitHub is a free repository manager for Git version control systems. It allows the user to maintain local repositories and it provides server space for public repositories. A user would need to pay for server space in order to store a private repository[^^^].

## Google Maps API for Android

The Google maps Javascript API can be used for both mobile and desktop browser applications.  The API allows the developers to embed interactive or optionally static Google Maps into these applications.  The API Version 3, the newest version of the Google Maps API, is free to developers offering free applications and service to consumers [111].

The API uses JavaScript and object-oriented schemes.  All applications using the Google Maps API must load the API using an API key.  The API key enables the user to monitor API usage by the consumer, ensuring that the usage does not exceed the limits set by the API.  The usage limit set by the API is 25,000 map loads per day, where one load is counted when a map is first initialized on the page, independent of user usage after the initial load.  Additional usage must be purchased through Google [222].

The Google Maps API allows the developer to enter latitude and longitude information to display specific points on the map.  Addresses are also able to be transformed into latitude/longitude information, called geocoding.  Version 3 of the API supports geocoding [111].  Geocoding requests are rate-limited.  Because geocoding is computationally intense, it is advised to store location information of geocoded static addresses in advance in a cache, using the Gecoding API provided by Google [333].

The Google API offers Directions Service to calculate directions from one location to another [444].  The directions feature would be useful for providing customers with directions to a merchant providing a coupon or special offer.  In the API, directions are able to be specified using text strings, such as “Flagstaff, AZ,” or as specific latitude/longitude values.

For Android devices, Google provides a Maps external library.  This includes the com.google.androids.maps packets, which provides built-in downloading, rendering, and caching of Maps tiles.  In this Maps package, map data obtained from Google Maps is able to be displayed.  In essence, the MapView class is a wrapper around the Google Maps API that allows applications to use Google Maps data using class methods.  The Maps external library is not a part of the standard Android library, and thus may not be present on some devices.  The Maps external library can be obtained through the Google API add-on for the Android SDK [555].

## Google Maps API for iPhone

Blayne

## Android Location

Obtaining specific user location through their device is a fundamental part of localization.  Developers are able to utilize GPS and Android’s Network Location Provider information to acquire user location.  The advantages of GPS is that it is the most accurate.  However, GPS only works outdoors, consumes battery power very fast, and it does not return the location very fast.  Android’s Network Location Provider utilizes cell towers and Wi-Fi signals to determine the device location.  It works both indoors and outdoors, has a fast response, and uses less power than GPS.  User location can be obtained using both GPS and the Network Location Provider, or just one depending on the application [666].

Reliable user location may be difficult to achieve through several factors.  A combination of GPS, Cell-ID, and Wi-Fi information may be used to provide the user’s location.  Each of these methods provide varying levels of accuracy, and thus location may be inaccurate depending on which one of these methods are used.  Also, location determined while users are moving from location to location can degrade the location accuracy [666].

The Android SDK provides a callback to obtain user location from an Android device.  Before obtaining any form of user location, however, the application must request user permissions to obtain their location [666].  The optimal flow for obtaining user location while minimizing battery power and user data is the following:

1. Start the application
2. Start listening for updates from location providers.
3. Maintain a “current best estimate” of location through filtering.
4. Stop listening for location updates.
5. Use the last best location estimate.

## RestKit

RESTKit the iOS framework for interfacing Objective-C with a RESTful web service. It provides user friendly request and response functionality and the capability to easily switch between servers or environments. RESTKit also maps payloads to objects that the user can use without parsing the data.  This abstraction allows RESTKit to integrate the payloads with Apple’s Data framework [&&&].

## MySQL

MySQL is an open source database. The query structure is simple and easy to learn and use. It runs on multiple platforms to reach the largest possible user base. MySQL is also very stable, well documented, and well supported. These features make it ideal for our use because it is very unlikely that it will be incompatible with any of our systems [\*\*\*].

# Technology Integration

The GitHub repository holds and keeps track of the source code for both applications, for the website, and for the statistical analysis. This development tool logs the changes made to the system and allows the team to revert changes if need be. The source code is not actually run from the repository.

There are five main pieces of our system:

1. The website,
2. The Android application,
3. The iPhone application,
4. The MySQL database, and
5. The statistical analysis program

The database and the statistical analysis program are hosted on the server. The server has the Apache Tomcat framework on it.

There is also a separate server, currently NAU’s host server, which hosts the website. The website is written in a mixture of HTML, JavaScript, and PHP. The HTML and JavaScript are the front end languages and the PHP handles the server side functionality. The Google Maps API is also in JavaScript.

The iPhone runs the Objective C application on the iOS. The Google Maps API is used in the application as well and is written in Objective C for better integration. RestKit is the technology that allows the iPhone application to connect with our server in a RESTful way.

The Android phone runs the Java application on the Android operating system. The Google Maps API is integrated with the Java code as a part of the application. The application uses HTTP requests, similar to the website, to connect with the server and query the database.

The statistical analysis program will be written in PHP for seamless integration with the database. It will use the values in the database when it is updated and it updates the advertisement table.

Figure 1: An Overview of the Localization System

Website

iPhone App

RESTful HTTP Commands

Server

RestKit

iPhone

Android

Android App

Git Pull/Push Commands

Website Source Code

Android App Source Code

iPhone App Source Code

Statistical Analysis Source Code

Documentation

Website Source Code

Android App Source Code

iPhone App Source Code

Statistical Analysis Source Code

Documentation

Local GitHub Repository

(working copy)

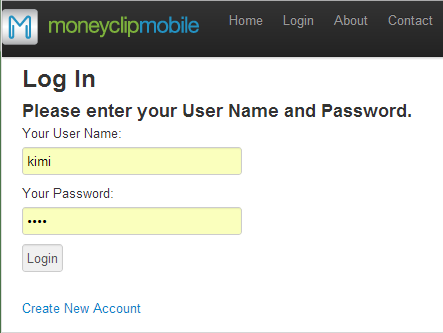
Public GitHub Repository (available to all collaborators)

# Proof of Feasibility

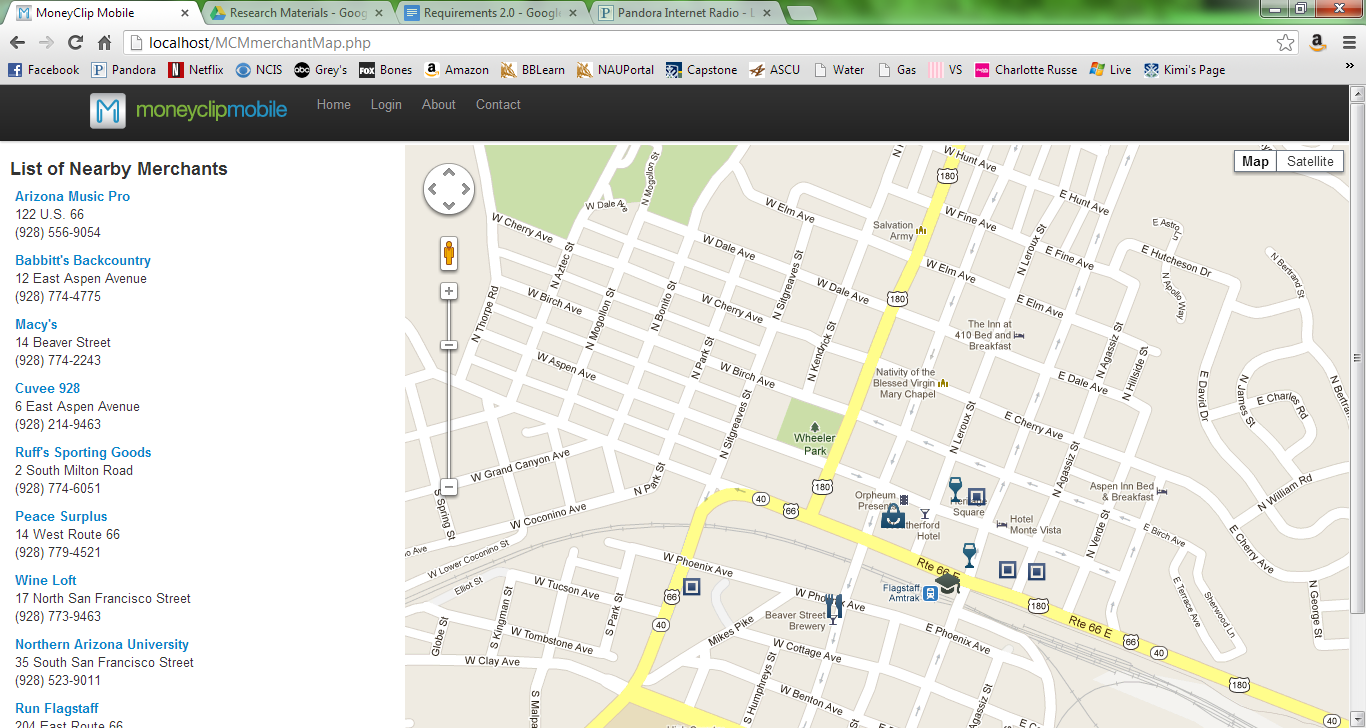
This section is a walkthrough of the localization system. It will demonstrate the functionality of the website, the android application, and the iPhone application.

## Website

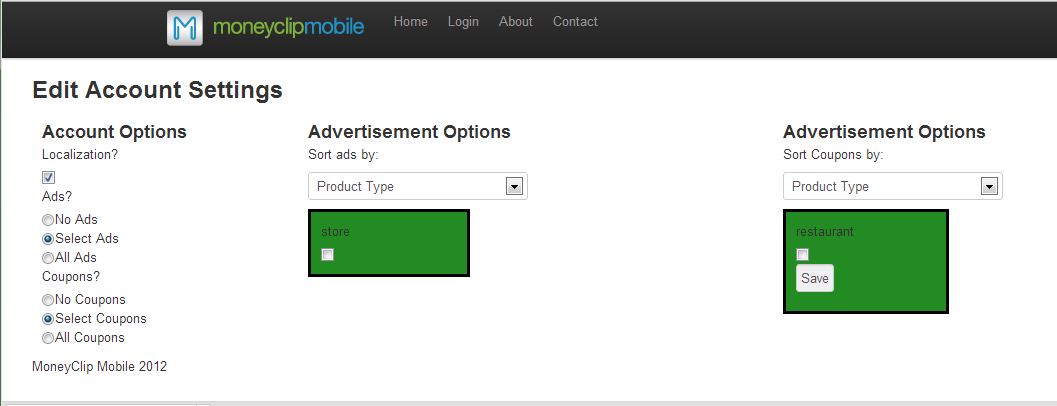
User Login



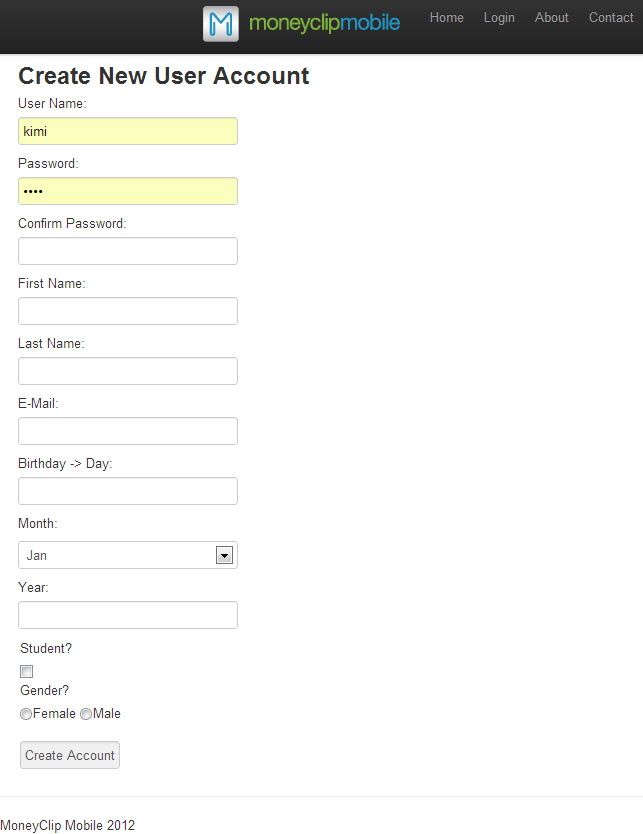
Merchant Map Page



Account Settings page

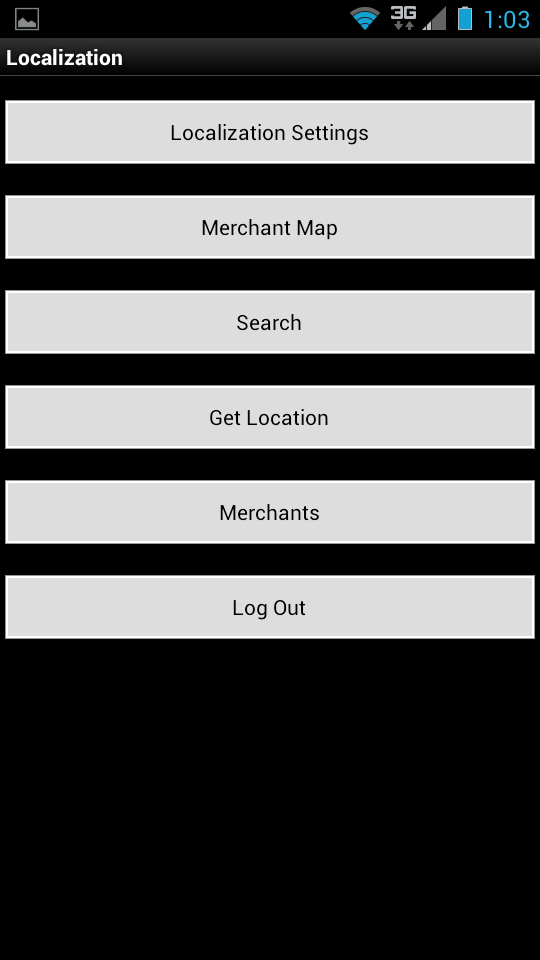


Create Account Page

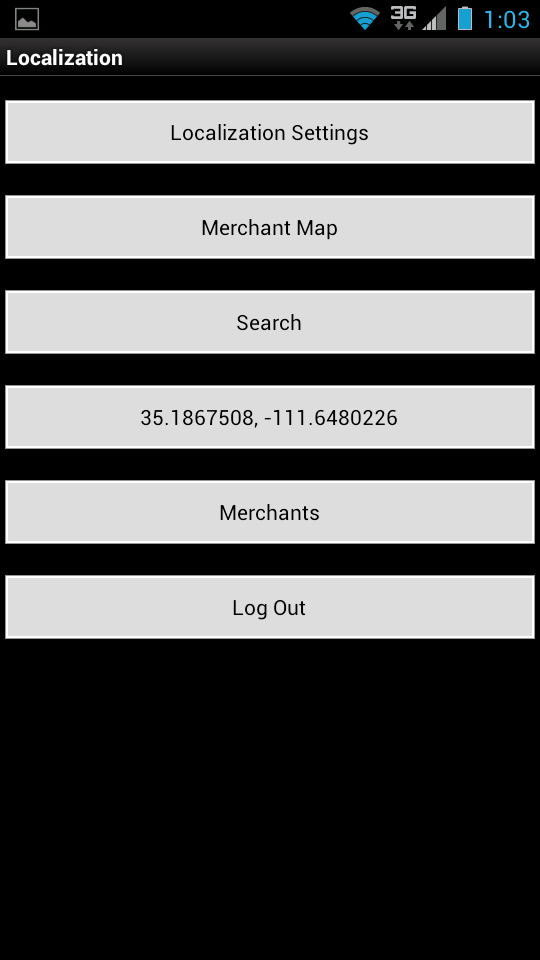


## Android App

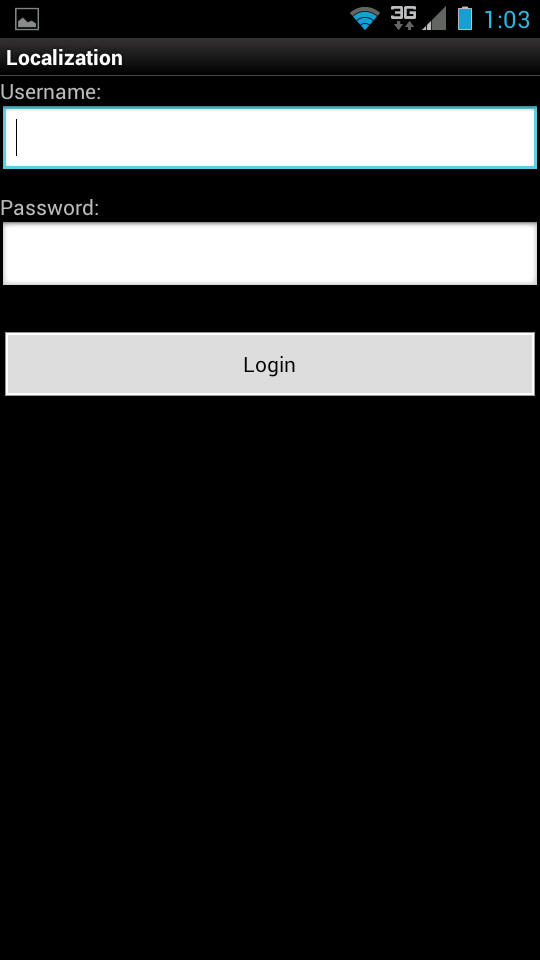
Home page before “Get Location” is clicked



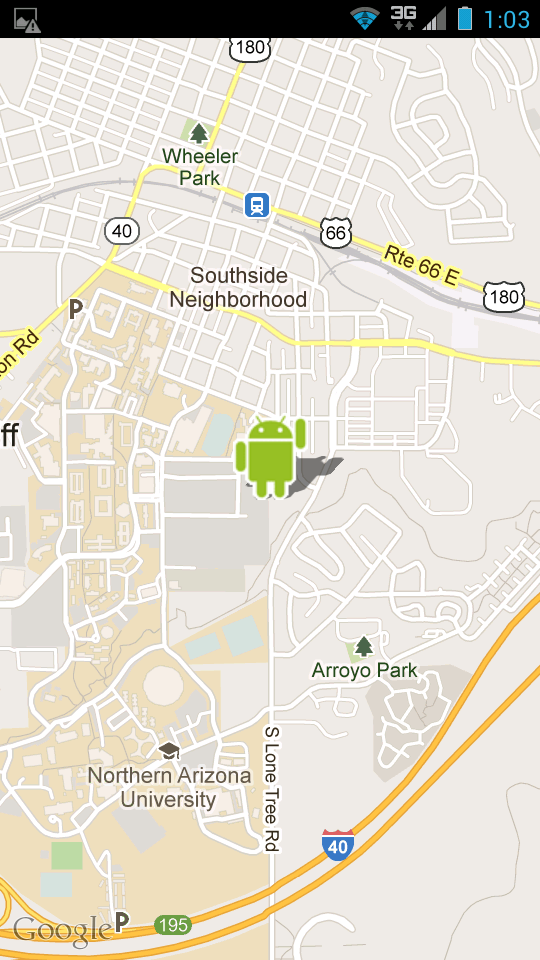
Home page after “Get Location” is clicked



Login

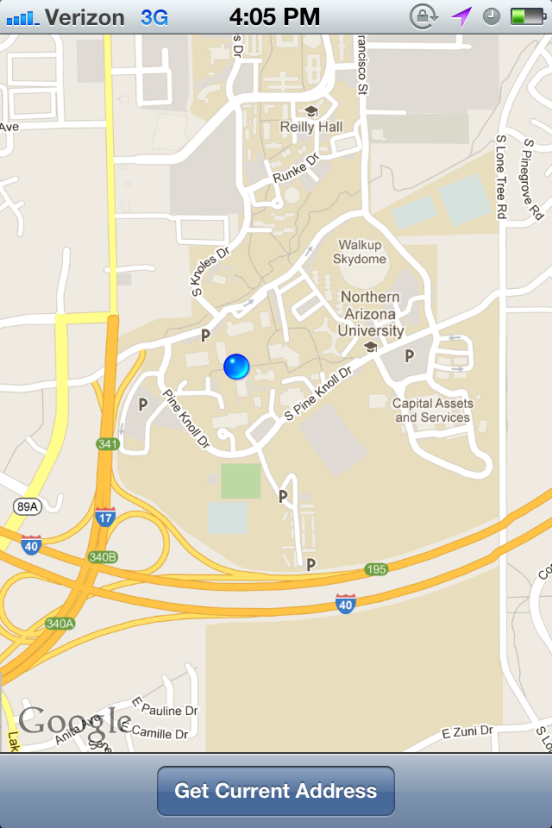


Map page

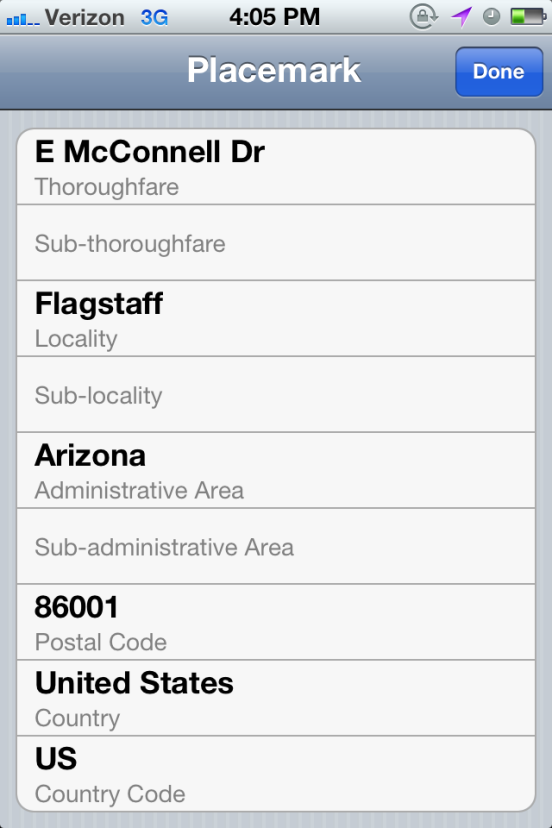


# iPhone

Map Page



Location Information



# Sources

The following sources were used as part of our research in completing the technology overview section.

[1] Apache Tomcat. <http://tomcat.apache.org/getinvolved.html>  
[\*] Servlets and JSP: An Overview. <http://www.apl.jhu.edu/~hall/java/Servlet-Tutorial/Servlet-Tutorial-Overview.html>  
[2] Rodrigues, Alex. 06 Nov 2008. RESTful Web services: The basics.   
<http://www.ibm.com/developerworks/webservices/library/ws-restful/>  
[3]  
  
[111] Google Maps JavaScript API v3: Getting started. https://developers.google.com/maps/documentation/javascript/tutorial  
[222] Usage limits and billing. https://developers.google.com/maps/documentation/javascript/usage#usage\_limits  
[333] Geocoding service. <https://developers.google.com/maps/documentation/javascript/geocoding>  
[444] Directions service. <https://developers.google.com/maps/documentation/javascript/directions>  
[555] Locations and maps. Android Developers. <http://developer.android.com/guide/topics/location/index.html>  
[666] Location strategies. Android Developers. http://developer.android.com/guide/topics/location/strategies.html

[&&&] RestKit. <http://restkit.org/>

[\*\*\*]<http://www.mysql.com/why-mysql/>

[^^^]<https://github.com/about>

[%%%]<http://git-scm.com/about>

[$$$]http://www.w3.org/Protocols/rfc2616/rfc2616.html

Appendix A:

Appendix B: