**CS 4720 - F17 - Final Project Documentation**

Device Name: Yanma Platform: Android

Name: Timothy Davison Computing ID: tfd2xq

Name: Seven Starosta Computing ID: sbs3bx

App Name: CacheMapper

**(Project proposal section)**

**Project Description:**

Our app, CachMapper, will provide a connecting service through which geocaching hobbyists can digitally connect and participate. Geocaching is a popular outdoor recreational activity involving the hiding of and visiting of cache sites around the world, where a logbook/ any items are stored in a waterproof container and hidden in an unmarked outdoors location. Our app will facilitate that hobby by providing local cache sites through GPS, allowing cachers to add their own sites (including pictures and descriptions of their caches).

What we propose to do is create an app that will do the following:

* The system shall allow a user to create a profile using a username and password.
* The system shall allow a user to access a GPS generated map of cache sites worldwide;
* The system shall allow a user to log their visits. This will be shown on a view cache page, where the visited state of each cache is stored and displayed.
* The system shall allow a user to create their own cache site, complete with a picture and a description, and register that cache site with the app. They shall be able to store caches they make in the cloud.
* The system will allow users to send emails to each other from the cache viewing page

We plan to incorporate the following features:

* GPS/ Location awareness: the application will give the user access to a google generated map indicating their position relative to registered caches. GPS will also be used when creating a cache at the current location.
* Camera: the application allows users to access their camera when registering a cache site for taking pictures to be associated with that cache’s page
* Build and consume your own web service using a third-party platform - cache information will be stored via the web service Firebase. **Note: if you consider firebase storage (for images) and firebase database (for strings and other more primitive objects) to be separate web services, please evaluate just our implementation of the database.**
* Data storage using key/value pair storage - We will store the user’s account information along with other basic information on the device.
* Open shared activity / features - We will provide the user with the ability to email the creators of other caches from the view cache screen.

**Wireframe Explanation:**

(see wireframe.pdf in the root of the project directory).

The first page is a login screen, where the user inputs an email and a password. If the username has not been used before, they are registered now, otherwise it is checked with the stored username/password combination.

The second page is the main screen with a map, which will show caches, and has an add button which leads to the add cache page. It also leads to individual caches by tapping.

The add cache page allows for a name, a username, and a picture.

The view page shows the name, username of the creator, and the picture.

**Platform Justification**

Mobile/android offers the clear advantage of being portable, which is essential to the task of creating geocaches dynamically out in the field. Mobile also has built in GPS which can function without internet (unlike a laptop), which furthers its suitability for this application. Using a phone/tablet allows the user to easily take their device off road, as well as to take pictures to include in their cache descriptions using a phone/tablet’s outward facing camera.

**Main Screens:**

**Login:** here the user can log in using a gmail account and a password chosen specifically for the app (i.e. not their actual gmail password). The username and password pairs are stored into system preferences, and the currently logged in username is passed onto the main activity screen if login is successful. We chose to use systemprefs for username/password because it could function without any internet access.

**Main Screen / Map Screen:** Here caches automatically downloaded from firebase are displayed as red X’s on the google map. The user’s current location is also shown by a blue marker. The map fragment is set to not be destroyed upon rotation to prevent the time and data cost of reloading the map on rotation. The geocaches are also cached (haha) by firebase locally here, meaning that if the user loses connection to the internet they can still access the caches on the view screen after entering from the main activity. Note that the images associated with each cache are **not** cached in order to prevent excessive space usage by the app. The main screen has a logout button which takes the user back to the login screen, which wipes the password from the text field. The main screen also has a blue add cache button which takes the user to the add cache screen.

**View Cache Screen:** this screen shows the information for the cache that the user has tapped on. This information is as such: the username of the creator of the cache, the name of the cache, the description of the cache, the number of users who have visited the cache, whether the current user has visited the cache displayed as a checkbox, and an optional image associated with the cache. The visited checkbox will be preset according to whether or not the user has been recorded as having visited the cache on firebase. If the user taps the checkbox, firebase and the number of visitors to the cache are appropriately updated. On the bottom of the screen there is an email button, which allows the user to email the creator of the cache in order to inquire about hints etc. The email activity is opened with the to: and subject: lines prepopulated specific to the geocache. The view cache also has a back button to the main screen. While this screen successfully rotates, it works best vertical in order to let the cache image display biggest.

**Add Cache Screen:** on this screen the user can give a name, multiline description, and choose to take a picture or choose an already taken one. A picture is optional and the cache can upload to firebase without a picture. The username will be automatically included, having been retrieved through systemprefs. The cache will also be automatically marked as visited by the creator, and will thus have an initial visitor count of 1. Upon creating the cache, a dialog prompts the user to return to the add cache screen.