

Follow Me

an Android Application

Requirements Analysis Document

CIS 450: Fall 2014

September 15, 2014



Table of Contents

Introduction	3
Proposed System.....	3
Functional Requirements.....	3
Non-Functional Requirements.....	4
Functional and Non-Functional Requirements of the Website	4
Use Case Model.....	5
Use Cases.....	6
Current System	15
User Interface	15
.....	16
Glossary	17

Introduction

This documentation contains a description of the functional and nonfunctional requirements that our team is able to commit to our client in the Android application called FollowMe and its supporting web site. We also present a potential design of the software product through a Use Case model and discuss our current progress developing the application and website.

Proposed System

The scenario presented by the client consists of the following. The user creates a route starting at point A and then drives to point B where the user stops and records specific notes about wildlife for future users to view. The user then drives or walks from point B to point C where the process is repeated. The process of driving from point to point and recording data continues until the user decides that he or she has reached the end of the tour.

The application we propose will fulfill all of the requirements of the scenario. It will show the route with all the stops made and all the observations created on a map displayed on the screen. It then stores the excursion on the local device and if it is made a public excursion, it will be uploaded to the application server for others to use. A user also has the ability to download a premade excursion. If the user chooses to use a premade excursion, the application downloads the excursion information from the application server and then stores it on the phone, allowing the user to follow the route and view the notes that the previous user had made about the given route.

The system that we propose consists of developing an application with the intent to “get the public out into nature”. To accomplish this goal, the application will allow a user to create and store trail and road excursions to later be used as a guided tour. An excursion consists of the route taken and observations made by a user. The tour application will allow for the user, the trail maker or explorer, to record GPS locations along the route and various text descriptions of nature items that they run across. For example: a user will be able to record that they saw a beaver at some given location on the trail they are traversing. In addition, this application will connect to an application server, to store the routes recorded as well as storing the observations of wildlife and nature that were taken.

Functional Requirements

The functional requirements we promise to our clients are listed below.

1. The application allows users to create accounts
2. The application can display a map with a user's position.
3. The user can create or load publically available excursions.
4. The application can record and display the route the user is traversing.
5. The user may add observations at various locations on the route and others may view these observations.
6. The user may set excursion to private or public

Non-Functional Requirements

The non-functional requirements we promise to our clients are listed below.

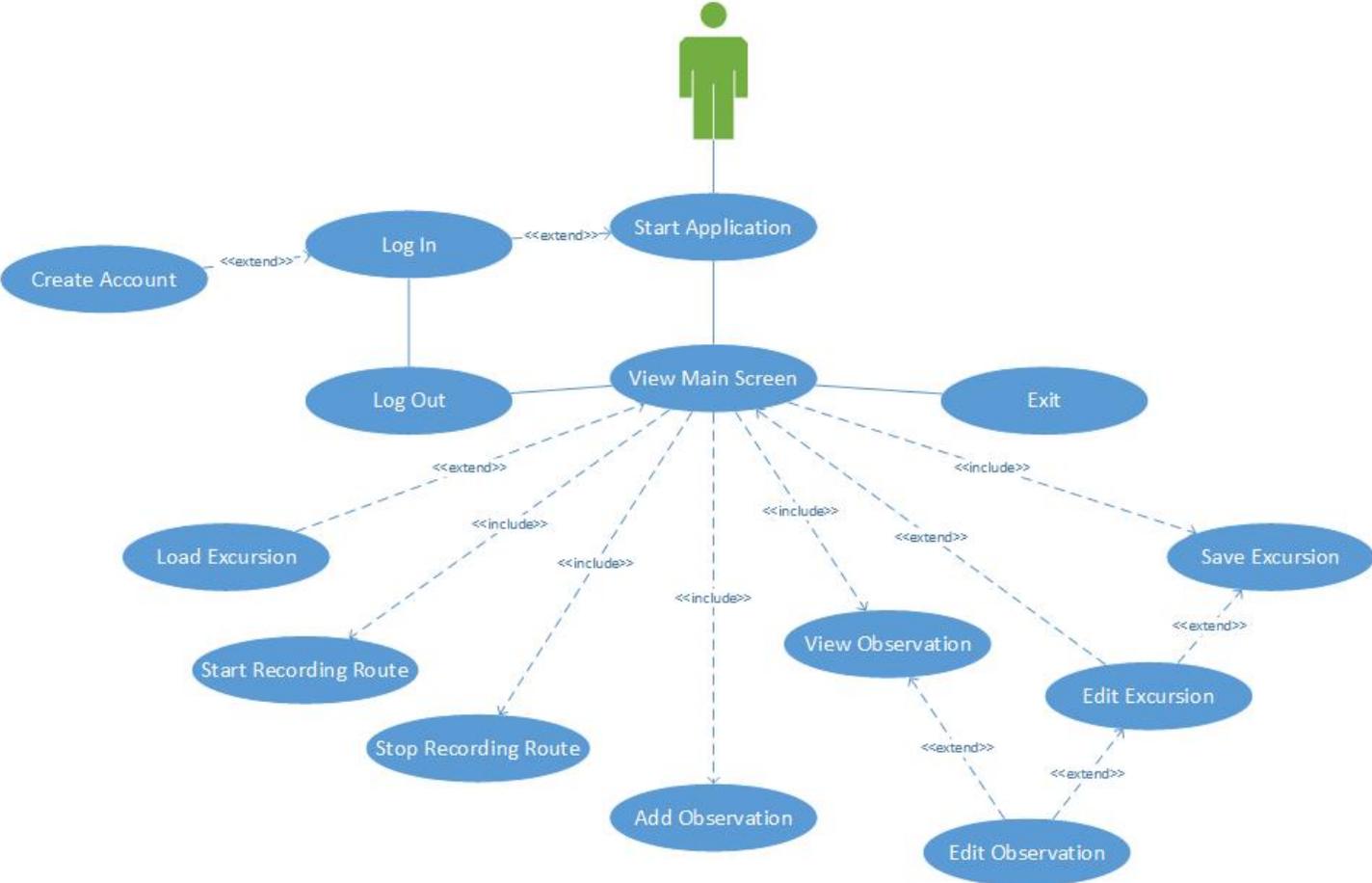
1. The system is restricted to registered users. Registration occurs in the application.
2. The application stores and retrieves data from the local and application server databases as needed.
3. The user may download excursions prior to losing cell signal and use the excursion (in a limited capacity) without cell service.
4. Data collected by the application about the excursion, if made public, is saved in the cloud database for other users to download and use.
5. The user may create and modify excursions (in a limited capacity) without cell signal and once cell signal is gained again the changes made will be uploaded to the cloud database.

Functional and Non-Functional Requirements of the Website

The functional and non-functional requirements of the web page that we promise to our clients are listed below.

1. The website allows the user to download the Android application.
2. The website provides an instructional manual about the application.
3. The website will look professional and will have a similar look and feel to the application.

Use Case Model



Use Cases

- I. *Use Case Name: Start Application**
- II. Participating Actors: User
- III. Entry Conditions: User has downloaded and installed the application on an Android device. The user presses the application launch icon.
- IV. Flow of Events:
 - a. A splash screen, that includes the name of the application and the application logo, displays briefly on the screen.
 - b. If the user is not already logged in, the application invokes the *Login* use case. Otherwise the *View Main Screen* use case is invoked.
- V. Exit Conditions: The user is logged in and the main screen is loaded.
- VI. Exceptions: The user was not able to log in.
- VII. Quality Requirements:

- I. *Use Case Name: Login**
- II. Participating Actors: User
- III. Entry Conditions: The user is not currently logged in.
- IV. Flow of Events:
 - a. The login screen is displayed that includes the following controls.
 - Login name and password fields
 - A button labeled Login
 - A button labeled Create Account
 - b. If the user clicks the Create Account button they invoke the *Create Account* use case.
 - c. Otherwise the user enters their login name and password and clicks the Login button.
 - d. The application compares the credentials with the credentials stored on the application server.
 - i. If the user name is invalid, or the password is not authenticated, the fields are cleared, an error message is displayed, and the flow of events continues at step a.
 - ii. Otherwise the application registers the user as being logged in and invokes the *View Main Screen* use case.
- V. Exit Condition:
 - a. User is logged in to the application and the main application screen is displayed.
- VI. Exceptions:
 - a. The user doesn't supply valid credentials.
 - b. The local database server is corrupted, preventing the credentials from being validated.

VII. Quality Requirements: The local database server works properly.

I. *Use Case Name: Create Account**

II. Participating Actors: User

III. Entry Condition: The user is not be logged into the application.

IV. Flow of Events:

- a. A screen is displayed with the following controls.
 - Login name field
 - Email address field
 - First and last name fields
 - A password field and a confirm password field
 - A button labeled Create Account.
- b. The user enters their credentials.
- a. If the two passwords don't match an error message appears and both password fields are cleared.
- c. The user presses the Create Account button.
- b. The credentials are sent to the application server.
- c. The application server checks that the email address and user name are not already registered. If either is already registered the field is cleared and an error message is displayed.
- d. The application server stores the user information.
- e. The application also stores the user information locally and logs the user in.

V. Exit conditions: The user has a valid account and is logged into the application.

VI. Exceptions:

- a. The user doesn't supply valid credentials.
- b. The connection to the application server can not be established or is severed before the account is created.

VII. Quality Requirements:

- a. The user's email address must be a properly formatted email address.
- b. The login name can only contain upper and lower case letters and numerals. It must also be at least 3 characters in length.
- c. The password can only contain upper and lower case alphabet characters and numerals. It must also be at least 8 characters in length, start with an alphabet character and include at least one number.
- d. The application must maintain a stable connection to the application server.

- I. *Use Case Name: View Main Screen**
- II. Participating Actors: User
- III. Entry Condition: The user is logged into the application.
- IV. Flow of Events:
 - a. The users GPS location is regularly computed.
 - b. If a current excursion is not loaded, the current excursion is set to a default excursion with no route points and no observations. A map of the area, containing the user's GPS location, is downloaded and displayed on the screen. The user's GPS location is also plotted on the map.
 - c. If the current excursion is a saved excursion, its route and observations are displayed on the map. If the users location is within the viewing area, the location is marked on the map.
 - d. A menu bar is displayed at the top of the screen with the following buttons.
 - Load Excursion
 - Start Recording Route
 - Add Observation
 - Stop Recording Route
 - Edit Excursion
 - Save Excursion
 - Log Out
 - Exit
 - e. The buttons that are active depends on the included use cases that are being invoked.
 - i. The Add Observation, Edit Excursion, Save Excursion, Log Out, and Exit buttons and are always active.
 - ii. The Load Excursion button and the Start Recording Route button is active if the application is not recording the user's location.
 - iii. The Stop Recording Route button is active if the application is recording the user's location.
 - f. When any button is pressed a corresponding use case is invoked.
 - g. If an observation mark is displayed on the map and the user presses the observation the View Observation use case is invoked.
- V. Exit Conditions: The user presses the Log out or Exit buttons.
- VI. Exceptions:
- VII. Quality Requirements: The device must have GPS capabilities.

- I. *Use Case Name: Load Excursion**
- II. Participating Actors: User
- III. Entry Condition: User has the main screen loaded.
- IV. Flow of Events:
 - a. If the current excursion has changes that have not been saved, the application asks the user if he wants to save the changes.
 - b. A Load Excursion Screen is displayed.
 - c. If the device has an Internet connection, the application updates a list of available (on the device and on the Internet) excursion.
 - d. The application displays a scrollable list of available excursions. The application also displays a button labeled Cancel.
 - e. The user clicks an excursion that he wants to view.
 - f. If the excursion data is not on the device, it is downloaded onto the device.
 - g. The application's current excursion is set to the new excursion.
 - h. The route of the excursion is plotted on the map.
 - i. The locations of the observations are marked on the map.
- V. Exit conditions: The application's current excursion is set to a new excursion and the route and observations points of the excursion are plotted on the screen or the user presses cancel and the previous excursion is still loaded.
- VI. Exceptions:
 - a. The user selects a route requiring downloading, but the application fails to download it and a message is displayed on the screen.
- VII. Quality Requirements:
 - a. Device must have sufficient permanent storage to store the downloaded excursions.
 - b. An Internet connection is needed to download excursions.

- I. *Use Case Name: Start Recording Route**
- II. Participating Actors: User
- III. Entry Condition: The application is displaying the main screen and the application is not recording the GPS location of the user.
- IV. Flow of Events:
 - a. At a regular interval the application calculates the user's GPS location, stores the location and plots the location on the map.
- V. Exit Conditions: The user's location is continually updated in the local database and on the map.
- VI. Exceptions:
- VII. Quality Requirements: The device must have a device that can calculate the GPS location of the device.

- I. *Use Case Name: Add Observation**
- II. Participating Actors: User
- III. Entry Condition: The application is displaying the main screen.
- IV. Flow of Events:
 - a. An Add Observation Screen is displayed with the following controls.
 - Title field
 - Description field
 - GPS location field
 - A button labeled Cancel
 - A button labeled Save
 - b. The application calculates the current GPS location and sets the GPS location field.
 - c. The user may edit all fields and when complete hits the Save button if he wants the observation saved in memory.
 - d. If saved, the date and time is also saved.
- V. Exit Conditions: A new observation is saved in memory and a mark for the observation is added on the map.
- VI. Exceptions: The user presses the Cancel button.
- VII. Quality Requirements:

- I. *Use Case Name: View Observation**
- II. Participating Actors: User
- III. Entry Condition: The application is displaying the main screen and the user clicked on an observation mark on the map.
- IV. Flow of Events:
 - a. The View Observation screen appears.
 - b. A scrollable box is displayed on the screen that contains the title, description and GPS location of the observation along with a button labeled Back.
 - c. If the user created the excursion, an Edit button also appears on the screen. If the user presses edit, the Edit Observation use case is invoked.
 - d. When the user is finished reading about the observation, he presses the Back button which causes the View Main Screen use case to be invoked.
- V. Exit Conditions: The user viewed the observation.
- VI. Exceptions:
- VII. Quality Requirements:

VIII. *Use Case Name: Edit Observation**

IX. Participating Actors: User

X. Entry Condition: The user chose to edit an observation either from the View Observation use case or the Edit Excursion use case.

XI. Flow of Events:

a. The Edit Observation Screen is displayed with the following controls.

- Title field
- Description field
- GPS location field
- A button labeled Save
- A button labeled Cancel
- A button labeled Delete

b. The title, description and GPS location fields are populated with data from an existing observation.

c. The user may edit all fields and when complete hits the Save button if he wants the observation saved in memory.

XII. Exit Conditions: The user viewed the observation.

XIII. Exceptions: The user pressed Cancel.

XIV. Quality Requirements:

I. *Use Case Name: Stop Recording Route**

II. Participating Actors: User

III. Entry Condition: The application is displaying the main screen and the application is recording the user's GPS location.

IV. Flow of Events:

a. The application stops recording the user's GPS location.

V. Exit Conditions: The application is not recording the user's GPS location.

VI. Exceptions:

VII. Quality Requirements:

- I. *Use Case Name: Edit Excursion**
- II. Participating Actors: User
- III. Entry Condition: The application is displaying the main screen.
- IV. Flow of Events:
 - a. An Edit Excursion screen is displayed with the following preference controls.
 - Excursion Name field
 - Excursion Description field
 - Toggle controls to determine if the route is for walking or driving
 - Toggle controls to determine if the route is private or public
 - A list of observations titles along with an edit button next to each observation.
 - A button labeled Cancel
 - A button labeled Save
 - b. If the name of the excursion is the same as the name of the default excursion an asterisk appears to the right of the name field.
 - c. The user may edit the name and description and select the toggles.
 - d. If the user presses the edit button next to an observation the Edit Observation use case is invoked.
 - e. If the user presses cancel, the changes are not saved and the View Main Screen use case is invoked.
 - f. If the user presses Save, the application checks to see if the excursion name is different from the default excursion name. If not, a message appears to the right of the excursion name and the changes are not made. Otherwise, the changes are saved and the View Main Screen use case is invoked.
- V. Exit conditions: Modifications to the current excursion's properties are saved or the user presses cancel. The View Main Screen use case is invoked.
- VI. Exceptions:
- VII. Quality Requirements:

- I. *Use Case Name: Save Excursion**
- II. Participating Actors: User
- III. Entry Condition: The application is displaying the main screen and some element of data for the excursion has changed.
- IV. Flow of Events:
 - a. If the name of the current excursion is the default excursion name the Edit Excursions use case is invoked with the Cancel button on the Edit screen made inactive.
 - b. The excursion data is saved to the local database.
 - c. If an Internet connection exists and the excursion is public, the excursion is uploaded to the application server and a message indicating the excursion was saved and uploaded is briefly displayed on the screen. Otherwise a message indicating that the excursion was saved but not uploaded is displayed.
- V. Exit Conditions: The excursion is saved to the local database.
- VI. Exceptions:
- VII. Quality Requirements: An Internet connection is required in order to upload the excursion to the application server

- I. *Use Case Name: Log Out**
- II. Participating Actors: User
- III. Entry Condition: The application is displaying the main screen.
- IV. Flow of Events:
 - a. The application displays a dialog box and asks the user if they are sure that they want to log out.
 - b. If the user selects *Yes*, the user is logged off and the Login use case is invoked.
- V. Exit Conditions: The user is logged off and the log in screen is displayed.
- VI. Exceptions: The user decided he does not want to log out and selects *No* when the dialog box is displayed.
- VII. Quality Requirements:

- I. *Use Case Name: Exit**
- II. Participating Actors: User
- III. Entry Condition: The application is displaying the main screen.
- IV. Flow of Events:
 - a. The application checks if an excursion is being uploaded to the application server. If so, the application displays a message to inform the user that an excursion is being loaded and waits for the upload to complete.
 - b. When uploading is complete, the application terminates.
- V. Exit Conditions: The application terminates.
- VI. Exceptions:
- VII. Quality Requirements:

Current System

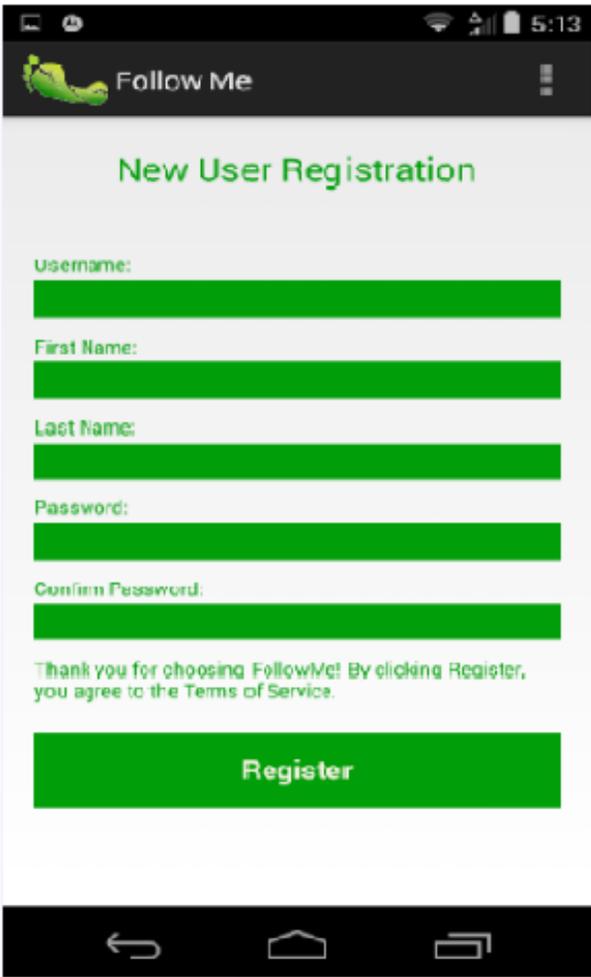
To date, our team has created a splash page that is presented to the user when they open the application and graphical layouts for the login page, an account registration page, and a basic main task page.

When the application starts, the user is greeted with the splash screen that includes a picture of a trail accompanied by the FollowMe title and logo. From there, the user is taken to the login screen where they can choose to enter their credentials or click a link to register for a new account. If they choose to enter and submit their credentials they will be taken to the main task screen that tells them they have successfully logged in. If the user chooses to click on *Register Me*, they will be taken to the register screen where they will be able to fill out various data fields including username, first name, last name, password and also confirm password.

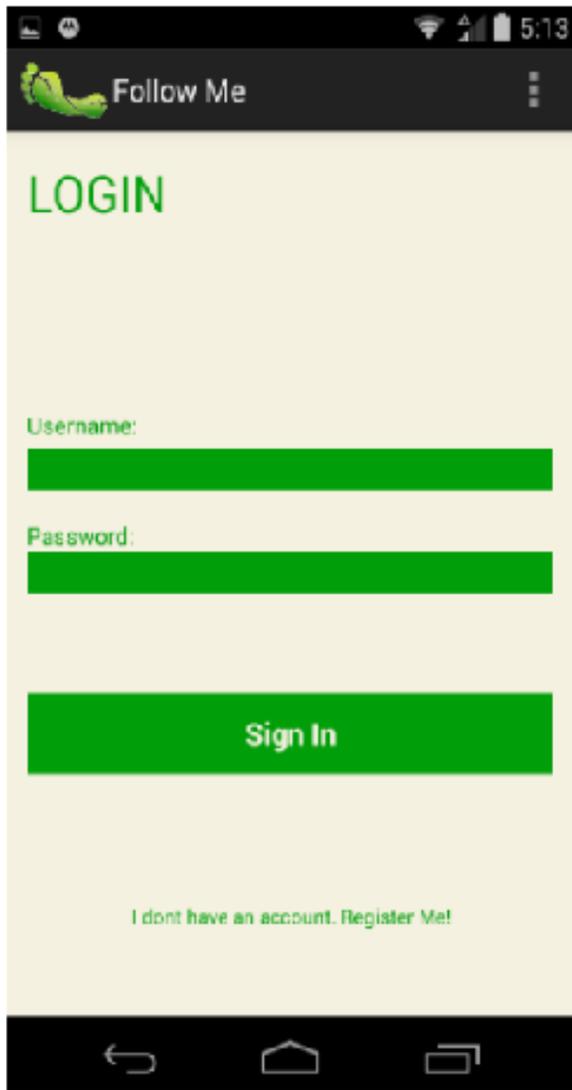
User Interface



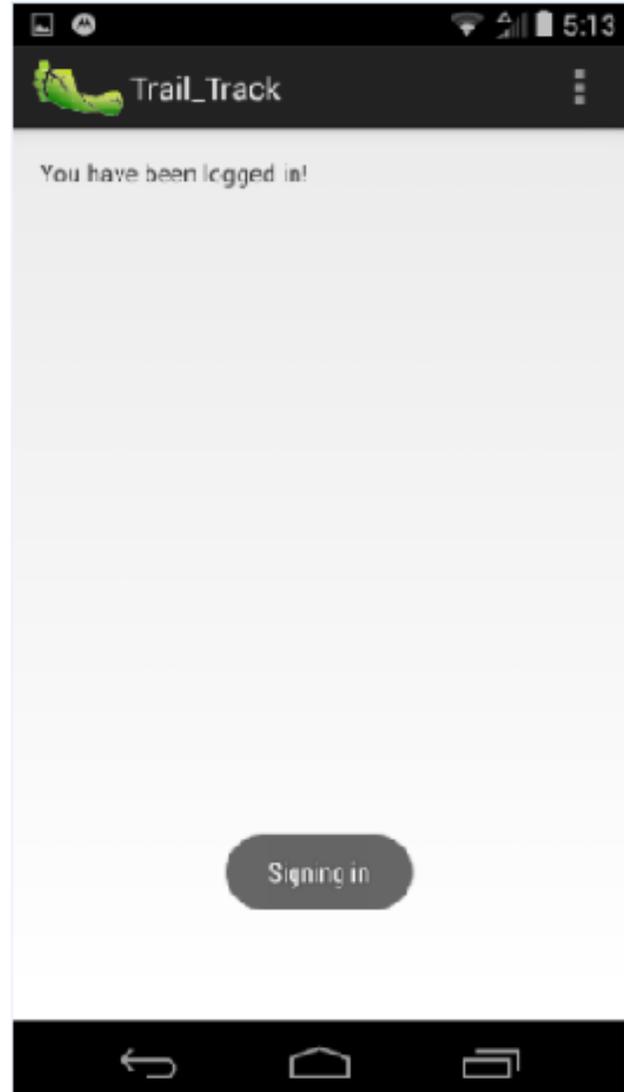
Splash Screen



Registration Page



Login Page



Main Screen

Glossary

Account: a description counting user information, used to distinguish among users.

Credentials: The user name, email address and password associated with the user's account.

Excursion: A route a user takes and the observations that are recorded.

Observation: notes made by user while on excursion

Password: user authentication method, consists of a certain amount of characters of a certain length, paired with a username.

Route: the path created by a user

Splash Screen: A temporary screen the user sees that displays the logo and application name when the application opens.

User: person who utilizes the app to go on an excursion

User Interface: The display that the user will interact with when they are using the application.

User Name: name or emailed used to identify who is using the application, also the username matches an account with information about the user.