**Team 1: Precision Beacon Navigation**

**Testing Document**

**Tahir Aziz**

**Adeel Khan**

**Sabur Khan**

**Alejandro Guzman**

**Casey Boyle**

**04/07/17**

**ABSTRACT**

One important aspect of the software process is testing the software in question so the team is sure it actually works as it is supposed to. This means that the requirements and use cases of the software should be tracked to the software testing, so that way the software can be tested effectively. This also means that the requirements should be measurable and testable, so that during the software testing step, the software can be evaluated to make sure it’s doing what it’s supposed to. In this document we will introduce the scope of this document, talk about the test cases, the traceability to the use cases, and the techniques used.

**TABLE OF CONTENTS**

**INTRODUCTION**

**REQUIREMENTS/SPECIFICATIONS-BASED SYSTEM LEVEL TEST CASES TRACEABILITY OF TEST CASES TO USE CASES**

**TECHNIQUES FOR TEST GENERATION**

**RESEARCH TESTING**

**EVIDENCE THE TEST CASES, DOCUMENT HAVE BEEN PLACED UNDER CONFIGURATION MANAGEMENT**

**REFERENCES**

**INTRODUCTION**

This document will outline the plan that the team has for testing the application that we are building. The purpose of a testing plan is to plan out how we want to test the software will be tested. Testing is important in the software development process because it is a way to check the code that has been written, to help identify flaws and errors in the code or underlying logic, and in test-driven development, allow a way to make the code testable and modular. In our case we will be testing the application to ensure that the use cases and functionality dictated by the requirements are accurate and correct. The scope of this document will cover the system level test cases that we plan on having, the traceability of these system test cases to the use cases themselves, the details of the test generation and techniques and the rationale of the testing method we are using, and outlining additional research testing we want to perform to gather more data. The goal of this document is to have a plan that will discuss how we will be testing the application and why.

**REQUIREMENTS/SPECIFICATIONS-BASED SYSTEM LEVEL TEST CASES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case ID | Test Scenario | Test Steps | Test Data | Expected Results |
| TestUC1 | The user wants to Navigate from point A to point B. | 1. User opens App  2. User selects Start Point  3. User selects destination  4. User follows directions  5. User reaches destination | User chooses the start point.  User chooses the destination | User should reach destination based on application Map |
| TestUC2 | Navigation feedback is given to the user | 1. User navigates  2. Haptic or audio feedback is given while user is on path |  | User should feel haptic feedback |
| TestUC3 | User adds new map to the application | 1. User opens App  2. User adds map to the application | User chooses new map to add to the application | User should be able to see the new map in the application. |
| TestUC4 | User gives the application access to location | 1. User opens app  2. User grants device location access |  | The device should display the user’s current location |
| TestUC5 | User adds beacons to the application | 1.User sees no beacons connected  2.User selects a beacon to add | User selects beacon to add | The device should display added beacon. |

**TRACEABILITY OF TEST CASES TO USE CASES**

The test cases that we are including in this document relate directly to the use cases in the requirements document. To be specific the Test Case ID’s correlate with the Use cases with them being: TestUC1 goes with use case 1, TestUC2 goes with use case 2, TestUC3 goes with use case 3, TestUC4 goes with use case 4, TestUC5 goes with use case 5

**TECHNIQUES FOR TEST GENERATION**

In order to generate the test cases, we plan to use JUnit for helping us to write the unit tests. It will allow us to test these basic functions and return the pass/fail values from the tests, while allowing us to configure various settings such as creating test objects to run the tests on and customizing what we want to do with the test using the various built in JUnit fixtures. We will also need to map the steps of our use cases to the appropriate test so that the tests are as realistic as possible.

Since we are doing unit testing to determine the outcomes of various steps per the use cases, we will be using black box testing. Black box testing looks at testing the input and output of the test rathe the inner workings of the code, which is what white box testing is. Since our applications is built off existing code and various native Android classes, it isn’t feasible to test the inner workings of the code. We will just look at the results of our test cases to ensure the application is functioning.

In order to measure the quality of our tests, we need to determine criteria that measures the success of those tests. Since our tests are supposed to be traceable to our use cases, the expected outcome of the use case is going to be a huge factor that determines the quality of our tests. Additionally, in order to have the inner workings of the code do what they need to, we need to provide proper inputs to the test cases. Also, as mentioned earlier, the correct mapping of the use case steps to the tests is another aspect that is important to having good quality and realistic test cases.

Summarized, the criteria for measuring the quality of the test cases are:

* Correct mapping of use case steps to the test cases
* Proper and realistic inputs for expected success, bad/fake input for expected failures
* Correct expected outcome given the input

**RESEARCH TESTING**

In addition to conducting system level and unit testing, we also wanted to do research on various configurations of the application setup itself. By doing this, this will allow the team to gather more data so we can determine the ideal conditions and hardware/software initial configuration that will work best given the requirements.

The areas we want to do additional testing on include:

* Showing the beacon RSSI signal emissions: Currently, we believe the beacons are emitting at a very slow rate so to test this, we will be posting the RSSI values so we can see where the RSSI values project our location versus where we actually are.
* Testing the application on different devices: Since different devices have different settings and interactions with external signals, we want to test the application on different devices. We want to find out if the device we are using is a reason for the innaccurate RSSI readings, or if it is the beacon emission rate, or if it is simply RSSI bouncing.
* Dave from HP was telling Dr. Hill that the beacons seemed to work better at waist level, so we also want to see if doing that will help increase the accuracy of the location precision and trilaterization of the beacons.

**EVIDENCE THE TEST CASES, DOCUMENT HAVE BEEN PLACED UNDER CONFIGURATION MANAGEMENT**

The document template from the class website was copy/pasted into a Google document so that members of the team could work collaboratively on filling in the necessary content for the Testing Plan.

**REFERENCES**

“How to Write Test Cases: Step by Step Guide with Examples,” *Guru99*. [Online]. Available: http://www.guru99.com/test-case.html. [Accessed: 29-Mar-2017].