|  |
| --- |
|  |
| Test design Specification |
|  |
|  |
|  |
|  |

|  |
| --- |
| [Type the abstract of the document here. The abstract is typically a short summary of the contents of the document. Type the abstract of the document here. The abstract is typically a short summary of the contents of the document.] |

**Table of Contents**

[1 Introduction 2](#_Toc326083786)

[2 Test Objective 2](#_Toc326083787)

[3 Detailed Testing Strategy 2](#_Toc326083788)

[3.1 Unit Testing 2](#_Toc326083789)

[3.1.1 White Box Testing 2](#_Toc326083790)

[3.1.1.1 Basis Path Testing – Tree Repository Module 2](#_Toc326083791)

[3.1.2 Test Cases 3](#_Toc326083792)

[3.1.3 [TC001] Install mobile application 3](#_Toc326083793)

[3.1.4 [TC002] MA User login 4](#_Toc326083794)

[3.1.5 [TC003] MA User logout 5](#_Toc326083795)

[3.1.6 [TC004] Mobile Application Settings 5](#_Toc326083796)

[3.1.7 [TC005] Start Transmitting 6](#_Toc326083797)

[3.1.8 [TC006] Start Transmitting 7](#_Toc326083798)

[3.1.9 [TC007] Resume Transmitting 7](#_Toc326083799)

[3.1.10 [UC008] Stop Transmitting 8](#_Toc326083800)

[3.1.11 [TC009] WA User Login 8](#_Toc326083801)

[3.1.12 [TC010] WA User Logout 9](#_Toc326083802)

[3.1.13 [TC011] WA User Registration 9](#_Toc326083803)

[3.1.14 [TC012] WA View Route 10](#_Toc326083804)

[3.1.15 [TC013] WA Rename Route 11](#_Toc326083805)

[3.1.16 Black Box Testing 12](#_Toc326083806)

[3.2 Integration Testing 12](#_Toc326083807)

[3.2.1 Incremental Testing 12](#_Toc326083808)

[3.3 System Testing 13](#_Toc326083809)

[3.3.1 Function Validation Testing 14](#_Toc326083810)

[3.3.2 Performance testing 14](#_Toc326083811)

[4 Pass/Fail Criteria 14](#_Toc326083812)

[4.1 Test Log 14](#_Toc326083813)

[4.2 Shipping or Live Release 14](#_Toc326083814)

[4.2.1 Shipping/Live Release Entry Criteria 14](#_Toc326083815)

[4.2.2 Shipping/Live Release Exit Criteria 15](#_Toc326083816)

[5 References 16](#_Toc326083817)

# Introduction

This document provides the test documentation that will facilitate the technical tasks of testing including the detailed test cases for both white box and black box testing. Each test case specifies who will be performing the test, the preconditions required to execute each test case, the specific item to be tested, the input, expected output or results, and procedural steps where applicable.

# Test Objective

The objective of this document is to expand on the test plan and provide specific information needed to actually perform the necessary tests. By providing detailed test information, we hope to reduce the probability of overlooking items and improve test coverage. Testers will be able to use each test case provided in this document to move forward and begin testing. Test results will be logged in a database and a complete bug report generated for each test failure.

# Detailed Testing Strategy

## Unit Testing

Unit Testing is done at the source or code level for language-specific programming errors such as bad syntax, logic errors, or to test particular functions or code modules. The unit test cases shall be designed to test the validity of the programs correctness.

### White Box Testing

In white box testing, the UI is bypassed. Inputs and outputs are tested directly at the code level and the results are compared against specifications. This form of testing ignores the function of the program under test and will focus only on its code and the structure of that code. The test cases that have been generated shall cause each condition to be executed at least once. To ensure this happens, we are applying Basis Path Testing. Because the functionality of the program is relatively simple, this method will be feasible to apply.

//Todo

Each function of the backend is executed independently ; therefore, a program flow for each function has been derived from the code. The development team will be performing all white box testing.

#### Basis Path Testing – Tree Repository Module

Using the program flow graph for each function in our tree repository module, we were be able to determine all of the paths that will need to tested and have developed the corresponding test cases. In order to test the success of each path, return values were added to verify successful completion. Any preconditions needed to exercise a path have been included in the test case.If the expected result/output is not achieved, the test will be considered a failure and a bug report filed.

### Test Cases

### [TC001] Install mobile application

|  |  |
| --- | --- |
| [ID] Name | [TC001] Install mobile application |
| Summary | The MA is installed on a device |
| Items to be tested | Module X : Login //ToDO what module is being tested |
| Users | All users |
| Pre-conditions | The device has GPS capabilities and the minimum RAM and storage requirements (TBD) |
| Basic Course of Events | 1. The user access the app market and downloads the MA  2. The MA is installed on the device |
| Input | N/A |
| Expected output | The app is ready to be used |

### [TC002] MA User login

|  |  |
| --- | --- |
| [ID] Name | [TC002] MA User login |
| Summary | The user logs in to the MA |
| Items to be tested | Module X : Login //ToDO what module is being tested |
| Users | All users |
| Pre-conditions | The user has opened an account using the WA |
| Basic Course of Events | 1. The user launches the MA 2. The first time the user logs in the MA presents the user with a Settings screen with Login, Password and URL fields. Additionally the application settings fields (Sampling Interval and Recording Interval) will also be shown. Subsequent login attempts will show a simplified version of the Login screen with Login, Password and URL fields only. Additionally the password field will show asterisk characters only. 3. The user enters his username and password. Additionally the user enters the URL ans settings fields when required. 4. The user authenticates and access the MA 5. The MA session remains open until the user explicitly logs out. The user logs in once and gains access to the MA without being prompted to log in again. |
| Alternative Paths | 1. In Step 3 the username the user enters does not exist. In this case the MA shows an error message and information on how to recover the account information using the WA  2. In Step 3 the password for the username doesn’t match the system records.In this case the user is shown a message and gets two more attempts for login in. If the attempts are unsuccessful the account is locked and an email is sent to the user to inform of the lock. |
| Input |  |
| Expected Output | The user is authenticated and has access to the MA functions, the MA is an idle state |

### [TC003] MA User logout

|  |  |
| --- | --- |
| [ID] Name | [TC003] MA User logout |
| Summary | The user logs out of the MA |
| Items to be tested | Module X : Login //ToDO what module is being tested |
| Users | All users |
| Pre-conditions | The user is currently logged in to the MA |
| Basic Course of Events | 1. The taps the logout button 2. The MA closes the user session and returns to the login screen |
| Alternative Paths | 1. In Step 1 the MA is currently either on a transmitting or paused state. In this case:   1. The MA sends a message to the WA that it is about to terminate its session 2. The MA closes the user session and returns to the login screen |
| Input |  |
| Expected Output | The session is terminated and the MA returns to the login screen |

### [TC004] Mobile Application Settings

|  |  |
| --- | --- |
| [ID] Name | [TC004] Mobile Application Settings |
| Summary | The user accesses the MA settings for viewing or modification |
| Items to be tested | Module X : Login //ToDO what module is being tested |
| Users | All users |
| Pre-conditions | The user is logged in to the MA |
| Basic Course of Events | 1. The user taps the Settings icon and app Settings screen is shown. 2. The user modifies the following values:  * Location sampling interval: Defined by an integer value and unit {seconds}. 1 < t < 3600. * Location recording interval: Defined by a type {distance, time}, integer value and unit {{m},{seconds}}. 1< d < 100. 1 < t < 3600. * Web server to report the location points to {URL}.  1. After a change is made the Save and Cancel buttons are enabled 2. The user clicks the Save or Cancel buttons and the Settings screen grays out both buttons. 3. The user clicks the Close button and goes back to the Main Screen |
| Alternative Paths | 1. In Step 2 the user does not make any changes to the settings and interacts with the Close button only. In this case the system does not enable the Save and Cancel buttons. |
| Input |  |
| Expected Output | The application settings are changed (settings) |

### [TC005] Start Transmitting

|  |  |
| --- | --- |
| [ID] Name | [TC005] Start Transmitting |
| Summary | The user turns the MA transmitting function on |
| Items to be tested | Module X : Login //ToDO what module is being tested |
| Users | All users |
| Pre-conditions | The user is logged in to the MA |
| Basic Course of Events | 1. The user taps the Start button 2. The MA verifies the GPS is enabled on the device, if it is disabled it asks the user for his authorization for enabling it 3. The MA starts transmitting location points to the web server 4. The MA receives acknowledgement that the web server is receiving information 5. The MA shows a UI indicator that it is currently transmitting 6. The Start button changes its text from ‘Start’ to ‘Pause’ |
| Alternative Paths | 1. In Step 2 the user decides not to authorize enabling the GPS. In this case the application goes back tot he main screen. |
| Post-conditions | The MA starts transmitting location points and it is set to a transmitting state |

### [TC006] Start Transmitting

|  |  |
| --- | --- |
| [ID] Name | [TC006] Pause Transmitting |
| Summary | The user pauses the transmission of location points |
| Items to be tested | Module X : Login //ToDO what module is being tested |
| Users | All users |
| Pre-conditions | The MA was transmitting to the WA |
| Basic Course of Events | 1. The user taps the ‘Pause’ button. 2. The MA pauses transmitting location 3. The ‘Pause’ button changes its text to ‘Resume’ 4. The MA shows a UI indicator that it is currently on an paused state |
| Alternative Paths | None |
| Input |  |
| Expected Output | The MA is in a paused state |

### [TC007] Resume Transmitting

|  |  |
| --- | --- |
| [ID] Name | [TC007] Resume Transmitting |
| Summary | The user resumes the transmission of location points |
| Items to be tested | Module X : Login //ToDO what module is being tested |
| Users | All users |
| Pre-conditions | The MA is in a paused state |
| Basic Course of Events | 1. The user taps the ‘Resume’ button 2. The MA resumes transmitting 3. The ‘Resume’ button changes its text to ‘Pause’ 4. The MA shows a UI indicator that it is currently on an transmitting state |
| Alternative Paths | NA |
| Input |  |
| Post-conditions | The MA resumes transmitting location points and it is in a transmitting state |

### [UC008] Stop Transmitting

|  |  |
| --- | --- |
| [ID] Name | [UC008] Stop Transmitting |
| Summary | The user turns the transmitting function off |
| Items to be tested | Module X : Login //ToDO what module is being tested |
| Users | All users |
| Pre-conditions | The mobile application is initially transmitting information to the web application |
| Basic Course of Events | 1. The user taps the Stop icon. 2. The MA sends a message to the WA that it is about to stop transmitting 3. The MA stops transmitting 4. The application shows a UI indicator that it is currently on an idle state |
| Alternative Paths | None |
| Post-conditions | The application gets set to an idle state |

### [TC009] WA User Login

|  |  |
| --- | --- |
| [ID] Name | [TC009] WA User Login |
| Summary | The user logs in to the WA |
| Items to be tested | Module X : Login //ToDO what module is being tested |
| Users | All users |
| Pre-conditions | The user has opened an account using the WA |
| Basic Course of Events | 1. The user browses to the WA URL 2. The WA presents the user with a login screen with a login and a password fields. The password field will show asterisk characters only. 3. The user enters his username and password 4. The user authenticates and access the WA 5. The WA navigates to a ‘Home’ screen that shows a list of the user routes previously saved |
| Alternative Paths | 1. In Step 3 the password for the username doesn’t match the system records.In this case the user is shown a message and gets 4 more attempts for login in. If the attempts are unsuccessful the account is locked and an email is sent to the user to inform of the account locking |
| Post-conditions | The user is authenticated and has access to the WA functions |

### [TC010] WA User Logout

|  |  |
| --- | --- |
| [ID] Name | [TC010] WA User Logout |
| Summary | The user logs out of the WA |
| Items to be tested | Module X : Login //ToDO what module is being tested |
| Users | All users |
| Pre-conditions | The user is currently logged in to the WA |
| Basic Course of Events | 1. The user clicks the log out function 2. The WA closes the user session and returns to the login screen |
| Alternative Paths | None |
| Post-conditions | The session is terminated and the WA returns to the login screen |

### [TC011] WA User Registration

|  |  |
| --- | --- |
| [ID] Name | [TC011] WA User Registration |
| Summary | The user creates a new account in the WA |
| Items to be tested | Module X : Login //ToDO what module is being tested |
| Users | All users |
| Pre-conditions | None |
| Basic Course of Events | 1. The user browses to the WA url 2. The WA shows the login screen 3. The user clicks the ‘Open new account’ link 4. The user enters his/her personal information including: name, email and password 5. The WA validates this info, including verifying that the username does not exist already and validating the email and password formats 6. The WA creates the new account and sends a confirmation email to the user 7. The WA browses to the login screen |
| Alternative Paths | 1. In Step 5 the user information returns errors after the validation. In this case:   * The WA indicates in the UI which fields contain errors so the user can fix the problems * Once the detected issues are fixed the WA continues with step 6 |
| Post-conditions | The WA creates a new account |

### [TC012] WA View Route

|  |  |
| --- | --- |
| [ID] Name | [TC012] WA View Route |
| Summary | The user views a previously saved route in the WA |
| Items to be tested | Module X : Login //ToDO what module is being tested |
| Users | All users |
| Pre-conditions | The user is logged in to the WA |
| Basic Course of Events | 1. The ‘Home’ screen shows a list of the routes previously saved by the user 2. The user clicks on a route from the list 3. The WA retrieves the saved route data points on a list. Additionally it may or may not display these points on a map service provided by an external entity (e.g. Google Maps, Bing Maps). 4. The user clicks on the ‘Home’ link 5. The ‘Home’ screen shows a list of the saved routes |
| Alternative Paths | 1. On step 4 the user executes the the browser Back command. In this case the WA will execute step 5. |
| Post-conditions | The selected route data points are shown on a map |

### [TC013] WA Rename Route

|  |  |
| --- | --- |
| [ID] Name | [TC013] WA Rename Route |
| Summary | The user deletes a previously saved route in the WA |
| Items to be tested | Module X : Login //ToDO what module is being tested |
| Users | All users |
| Pre-conditions | The is currently logged in to the WA and in the ‘Home’ screen |
| Basic Course of Events | 1. The ‘Home’ screen shows a list of the routes previously saved by the user 2. The user selects one or more routes from the list 3. The user clicks the ‘Delete’ button/link 4. The saved routes are removed from the WA back end storage area 5. The ‘Home’ screen shows an updated list of the saved routes |
| Alternative Paths | None |
| Post-conditions | The WA shows an updated ‘Home’ screen |

|  |  |
| --- | --- |
| [ID] Name | [UC014] WA Delete Route |
| Summary | The user deletes a previously saved route in the WA |
| Items to be tested | Module X : Login //ToDO what module is being tested |
| Users | All users |
| Pre-conditions | The is currently logged in to the WA and in the ‘Home’ screen |
| Basic Course of Events | 1. The ‘Home’ screen shows a list of the routes previously saved by the user 2. The user selects one or more routes from the list 3. The user clicks the ‘Delete’ button/link 4. The saved routes are removed from the WA back end storage area 5. The ‘Home’ screen shows an updated list of the saved routes |
| Alternative Paths | None |
| Post-conditions | The WA shows an updated ‘Home’ screen |

### Black Box Testing

Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would. We have decided to perform Equivalence Partitioning and Boundary Value Analysis testing on our application. The Equivalent Partitioning will be performed at both the unit test level and the system test level. Boundary Value analysis will only be done at the system test level. In considering the inputs for our equivalence testing, the following types will be used:

* Legal input values – Test values within boundaries of the specification equivalence classes. This shall be input data the program expects and is programmed to transform into usable values.
* Illegal input values – Test equivalence classes outside the boundaries of the specification. This shall be input data the program may be presented, but that will not produce any meaningful output.

The equivalence partitioning technique is a test case selection technique in which the test designer examines the input space defined for the unit under test and seeks to find sets of input that are, or should be, processed identically. Black box testing will be performed by the test team. All procedural step have been included to assist the team in executing the various tests.

## Integration Testing

### Incremental Testing

// ToDo

There are two primary modules that will need to be integrated: the MA Interface and the backend . The two components, once integrated, will form the complete Application testing . The following describes these modules as well as the steps that will need to be taken to achieve complete integration. We will be employing an incremental testing strategy to complete the integration. The integration testing will be performed by the development team.

**Module 1 - Graphic User Interface (GUI) Module**

This module provides a simple GUI where the user can perform the different actions (functions). This module will be tested separate from the backend to check if each interface (e.g. insert button) is functioning properly, and in general, to test if the mouse-event actions are working properly. The testing will be performed by writing a stub for each element in the interface.

**Module 2 – Backend Module**

The DB interface provides the storage for the data elements and implements the algorithms and associated functionality of the. This module will be tested separate from the GUI by printing out the results to the Console????. In testing this module we will follow the incremental testing method i.e. testing one function first and then keep adding additional function and test it again until all the required functions are tested.

**Application**

|  |  |
| --- | --- |
| **GUI Function** | **DB Interface Function(s)** |
| Load Button | Read  Search  Insert |
| Store Button | Store  Write |
|  |  |
|  |  |
|  |  |
| Insert Button | Insert  Search |
| Delete Button | Delete  Search |
| Search Button | Search |

## System Testing

The goals of system testing are to detect faults that can only be exposed by testing the entire integrated system or some major part of it. Generally, system testing is mainly concerned with areas such as performance, security, validation, load/stress, and configuration sensitivity. But in our case well focus only on function validation and performance. And in both cases we will use the black-box method of testing.

### Function Validation Testing

The integrated Application will be tested based on the requirements to ensure that we built the right application. In doing this test, we will try to find the errors in the inputs and outputs, that is, we will test each function to ensure that it properly implements the

| **Function** | **Expected Behavior** |
| --- | --- |
| Load | see Software Program Specification for detailed explanation |
| Store | see Software Program Specification for detailed explanation |
| Insert | see Software Program Specification for detailed explanation |
| Delete | see Software Program Specification for detailed explanation |
| Search | see Software Program Specification for detailed explanation |
|  |  |
|  |  |
|  |  |

### Performance testing

This test will be conducted to evaluate the fulfillment of a system with specified performance requirements. It will be done using black-box testing method. And this will be performed by:

* Storing the maximum data in the file and trying to insert, and observe how the application will perform when it is out of boundary.
* Deleting data and check if it follows the right sorting algorithm to sort the resulting data or output.
* Trying to store new data and check if it over writes the existing once.
* Trying to load the data while they are already loaded

# Pass/Fail Criteria

This section will include the master list of both white box and black box tests which will be used to track the progress of the testing. A test will be considered a failure if the expected result or output is not achieved. A bug report will be filled out for each failure and will be submitted to the development team for correction. After the bug has been fixed, the test case will be repeated.

## Test Log

## Shipping or Live Release

### Shipping/Live Release Entry Criteria

### Shipping/Live Release Exit Criteria

The Shipping/Live Release stage is when the product is ready for general availability to the public and the user documentation is final. The product must fully satisfy its release specifications and the user documentation must adequately describe the product’s functionality. Both should be ready for use by the end user.

* QA tests the final product version to verify that the product to be released to the general public is of the utmost quality and satisfies original design specifications.
* The product must receive approval from the product team.
* QA and Development must prepare Release Notes.

The product is now ready to ship or published to production environment.

# References