

CS22120

Group 06 Test Specification

Configuration Reference:

E.g. : SE_06_PP

Authors:	Address:
Alexander Hardy (alh35)	Department of Computer Science
Michael Kitchen ()	Aberystwyth University
Forname Surname (UserID)	Aberystwyth
Forname Surname (UserID)	Ceredigion
Version: 1.0 (Draft)	SY23 3DB
Date: November 12, 2013	

Copyright © Aberystwyth University 2013

Contents

1	Introduction	3
1.1	Purpose of the Document	3
1.2	Scope	3
1.3	Objectives	3
2	Levels of testing and in what order	3
2.1	Unit testing	3
2.2	Integration testing	4
2.3	System testing	4
2.4	Beta Testing	5
2.5	Software To Be Tested	5
3	Time schedules for tests	5
4	What Realises of Hardware and Software Environments will be used	5
5	Where will the tests and the test results be stored	6
6	What tools maybe used	6
7	What Tests May Be Run	7
8	Who says that the tests have been carried out successfully	8
9	References	8
10	DOCUMENT HISTORY	8

1 Introduction

1.1 Purpose of the Document

This document displays the group's understanding of the purpose and requirements of testing and the aims of carrying out a testing process.

1.2 Scope

The document covers the process of testing our application and the various testing methods that will be used. It's contents are intended for use by the QA team and the development team.

1.3 Objectives

The aim of the testing processes is to eliminate all bugs from the code. To accomplish this testing will be done on multiple levels. This rigorous testing will hopefully catch as many bugs as possible. Unit testing with white box testing, integration testing using black box testing methods, validation testing, and high end system testing will all be carried out.

2 Levels of testing and in what order

There will be four main testing strategies:

- Unit testing
- Integration testing
- System testing
- Public testing ...

The unit tests will be done first and when these all pass the system will be integrated and all the system tests will be done. Once the testing team are satisfied that the whole program works the whole system will be tested by the public.

2.1 Unit testing

Unit testing will be done in white box fashion. All top level classes will have a set of unit tests that cover their functionality even if this is not covered in the requirements specification. Be done using Basis Path testing methods due to their simplicity and high effectiveness. Loop testing will also be conducted to compliment the basic path testing. The tests will be written by the programmer and should give a pass or fail result. The purpose of these tests is to ensure correct operation of all controls in all the pieces of software tested. The unit testing will be done throughout the building of the system and

should also be run frequently. All tests results shall be logged and any failures in these tests shall be addressed and solutions found. When any part of the system is changed all the unit tests will be run again to check that the alteration hasn't affected any parts of the system. The reliability is important so only the exact same tests will be performed so that we can be sure that we have not regressed. The pieces of software are expected to perform within design specifications.

2.2 Integration testing

The system will be integrated incrementally, to control the amount of bugs that need to be fixed at any given time. The system will be tested for errors in a black box fashion, after each component is integrated . The system is expected to integrate without major flaws.

2.3 System testing

System testing will be closely examined to make sure that the software meets the system requirements. These tests will be made to look for errors with a function requirement. The requirements specification will help us as it will give information on the tests that are required for the project. The features and functionality in the final system will be cross referenced with the design requirements specification to verify that the software is within the requirements. The system tests will test for:

- Inputs and outputs of the system.
- Places on the system where the data needs to be stored or computed.
- Places where the system interacts with the underlying operating system on the machine...

To test for bad inputs we will:

- Apply inputs that force all the error messages on erroneous input to occur, meaning that error handling code is executed.
- Try defaults – try to give no input and see if okay
- Try putting in characters that might be interpreted by the program
- Put illegal combinations of data
- Overflow input buffers
- Force the screen to refresh ...

To test for wrong data storage or computation again the data will have to be tested with bad input. The boundaries of the data storage will have to be tested as will data that is either too large or too small for the boundaries will have to be tested.

No Recovery testing will occur, as while system failures are undesirable, termination of the program in the event of a crash is acceptable. Security testing will be done to make sure that user can not just log on to other people walks that have been tracked. For stress testing the walk may be done at a faster walking pace to make sure that the tracking can keep up with the speed. It will also be tested with multiple people at one time to test the performance. Testing will be done on other errors such as disks being too full to handle data, permissions not being right on files or the network not responding.

2.4 Beta Testing

Beta testing will be semi-public. Select individuals outside of PA Testers will be expected to submit bug reports, as well as their opinions concerning performance and interface layout.

2.5 Software To Be Tested

The following pieces of software will need to be tested:

- Interface
- Android system
- PHP
- PostgreSQL
- Mapping system ...

3 Time schedules for tests

The unit testing will commence on the 1/10/14 and finish 1/23/14 Integration testing will commence on the 1/20/14 and finish 1/24/14 System Testing Beta testing

4 What Realises of Hardware and Software Environments will be used

The software used to test the application will have to run the program so below are the different software that the group can use to test the application and its features:

- Eclipse
- Netbeans
- Android Studio.

Because the Application will be made using Java which can be programmed on various different programming studios such as the ones above. This also makes it easier to test because the group will be able to carry out Junit tests on the various different features.

The Application will be able to be tested on the university machines as the above software are compatible with both windows and Unix machines alike so group members will be able to test both on the university campus and on their own personal computers if they wish to.

The Website will be written in XHTML, and one way in which this could be tested would be we could let friends/family members try out the website to see if its easy to get around/work. To see if the actual application works we could put the finished application on to one of the groups phones only if they have a android phone and try it out like an actual product. This as well could be tested on friends or family.

5 Where will the tests and the test results be stored

The tests are going to be analysed in Microsoft excel and then be stored in Microsoft Access in a database.

6 What tools maybe used

- Static analysers – These analyses the possible paths through a program so as to detect suspicious circumstances. These can include: Variable that declared but never used or code that is never reached. These checks are often carried out by compilers.
- Dynamic analysers – Keep record of the running of the programs and show how many times each statement has been executed. Useful for checking that each statement has been executed. Let's tester know where the program spends most of its time so performance can be enhanced.
- Test data generators – These accept a description of the format of the required data and then generate random test data according to this format.
- File comparators – detects the differences between two files so will be useful for checking large volumes of test output.
- Test running tools – We will be using Junit as this is an excellent java test running tool
- Interface testers – can be used to remember two things about the interface. One where the user has clicked and two what the user has clicked on. ...

7 What Tests May Be Run

Test No.	Test Name	Description
1	New Walking Tour	User creates a new walking tour
2	Check user input	User enters details
3	Initial GPS	Sets the initial GPS
4	Recording start options	Options should appear
5	Cancel recording button	This button should work
6	Saving tour button	This button should work
7	Adding locations button	This button should work
8	Single Name	Allows the user to enter a single word name when they have click
9	Title of the walk	User inputs a title for the walk
10	Short description for the walk	User enters a short description up to 100 characters
11	Long description for the walk	User enters a long description for the walk up to 1000 characters
12	Text Descriptions	Allows the user to add text descriptions to the walking tour
13	Map Co ordinates	In the text Description the map coordinates should be recorded
14	Name of Location	Adds a name to this text description
15	Description of location	Adds a description of the location to this text description
16	Time stamp of location	Adds the time
17	Add Photo	User is able to add a photo
18	Location of photo	Adds coordinates to the photo
19	Name of photo	Asks users and adds a name to the photo
20	Description of photo	Asks users and adds a description to the photo
21	Time stamp of photo	Adds the time the photo was taken
22	Cancel walk	Cancels the walk without saving
23	Edit	Any information recorded about the walk can be edited
24	Save walk	User saves the walk
25	Multi-purpose Internet Mail Extensions	Messaged is formatted as a Multi-purpose Internet Mail Extension
26	HTTP POST	Messaged is sent to the server via an HTTP POST to a predefined
27	Data information	Data that needs to be held is: name, title, long description, short
28	Android application	When the user switches application using the WTC system stores
29	Walk selected	Users sees all the places included in the walk with correct coordin
30	Choose Place	User selects a place and then the correct details appear including
31	Data sent	Data sent from the android should be in the correct format
31	SQL Database	Data for the new walk should be stored in an SQL Database
32	SQL Format	Data should be in the correct format as appendix a

33	Conformation message	Conformation message should be received by the user
34	List of walks	New walks should be added to the list of walks
35	Response of program	All inputs from user should appear within 1 second
36	Software	All software should run correct and be on the appropriate platform
37	Three walks	Data should be made for at least three walks

8 Who says that the tests have been carried out successfully

There will be no special resources required for this project. Alex Hardy will be the test leader, however other member of the team will also be involved in the testing process.

9 References

10 DOCUMENT HISTORY

Version	CCF No.	Date	Changes made	Changes made by
1.0	N/A	30/10/2013	First L ^A T _E X document created	LMC
1.1	N/A	03/11/2013	Added Sections 02, 06 & 09	LMC
1.2	N/A	04/11/2013	Added Sections 01, 04 & 05	LMC
1.3	N/A	05/11/2013	Edit to Use-Case Diagrams	LMC