Group Project 05 Test Specification

Authors:	bmo; sr11; hac22; wia2; wjl3; njv1	Department of Computer Science Aberystwyth University
Config Ref:	SE_05_TS_01	Aberystwyth, Ceredigion
Date:	14/11/13	SY23 3DB
Version:	1.1	Copyright © Group 05
Status:	Release	Aberystwyth University 2013

Contents

1. Introduction..3
1.1. Purpose of this document..3
1.2. Scope...3
1.3. Objectives..3
2. Android side testing..4
2.1. Module testing table..4
2.2. Module testing details..4
2.3. System testing table..5
2.4. System testing detail...6
3. Server side testing..7
3.1. System testing table..7
3.2. System testing details..8
4. References..10
5. Document change history...10

1. Introduction

1.1. Purpose of this document

The purpose of this document is to display all of the system level and module level testing that the team will use to make sure the walking tour application meets the standards of the requirement specification, the client gave.

1.2. Scope

This document includes short descriptions in the form of tables and long descriptions of all the system tests and all of the module tests the team will implement on the android and server side of the application.

1.3. Objectives

The objective of this document is to show what tests will be used to make sure the application, the web site and database meet requirement specification given by the client. All tests will have description and implementation information. The client should make sure that the tests provided here test the application fully and if the application passes all the tests displayed here, the client is to accept the project as complete.

2. Android side testing

2.1. Module testing table

Tour

Test	Detail	Result
TestInitialise()	Test to see if a Tour object is initialised with default values.	
TestInitWithGivenName()	Test to see if a Tour object is initialised with a given name as an argument.	
TestInitWithGivenNameAndShortDesc()	Test to see if a Tour object is initialised with a given name and short description as arguments.	
TestInitWithGivenNameAndShortDescAndLongDesc()	Test to see if a Tour object is initialised with a given name, short description and long description as arguments.	
TestAddLocation()	Test to see if when addLocation() is called a Location object is added to the list of Location objects.	
TestRemoveLocation()	Test to if when removeLocation() is called, the selected Location object is removed from the list of Location objects.	

Location

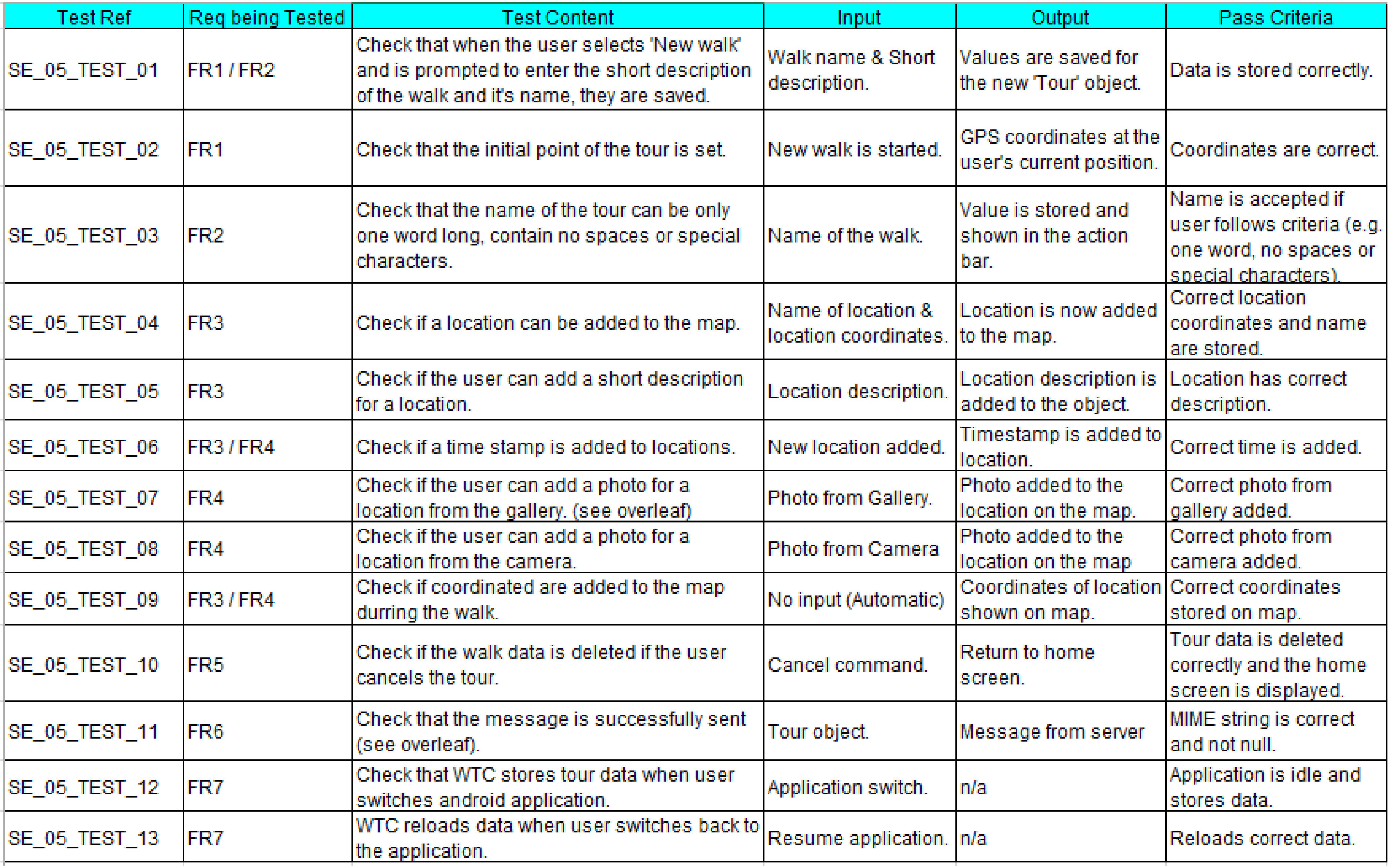
Test	Details	Result
TestInitialise()	Test to see if a Location object is initialised with default values.	
TestInitWithLongAndLat()	Test to see if a Location object is initialised with given longitude and latitude as arguments.	
TestAddPhoto()	Test to see if when addPhoto() is called the given Image is added to the list.	
TestRemovePhoto()	Test to see if when removePhoto() is called the given Image is removed from the list.	

2.2. Module testing details

The purpose of the JUnit test is to give us an assurance that when we release the app, it actually works at a core level. This means if there is an issue with the app, we know that it is more than likely a user issue or a UI interaction issue. For example, it might that an event isn't being processed correctly.

The JUnit tests will cover the constructors and functional methods of each class and to some extent, the getters and setters of the classes.

2.3. System testing table



Test Ref	Req being Tested	Test Content	Input	Output	Pass Criteria
SE_05_TEST_01	FR1 / FR2	Check that when the user selects 'New walk' and is prompted to enter the short description of the walk and it's name, they are saved.	Walk name & Short description.	Values are saved for the new 'Tour' object.	Data is stored correctly.
SE_05_TEST_02	FR1	Check that the initial point of the tour is set.	New walk is started.	GPS coordinates at the user's current position.	Coordinates are correct.
SE_05_TEST_03	FR2	Check that the name of the tour can be only one word long, contain no spaces or special characters.	Name of the walk.	Value is stored and shown in the action bar.	Name is accepted if user follows criteria (e.g. one word, no spaces or special characters).
SE_05_TEST_04	FR3	Check if a location can be added to the map.	Name of location & location coordinates.	Location is now added to the map.	Correct location coordinates and name are stored.
SE_05_TEST_05	FR3	Check if the user can add a short description for a location.	Location description.	Location description is added to the object.	Location has correct description.
SE_05_TEST_06	FR3 / FR4	Check if a time stamp is added to locations.	New location added.	Timestamp is added to location.	Correct time is added.
SE_05_TEST_07	FR4	Check if the user can add a photo for a location from the gallery. (see overleaf)	Photo from Gallery.	Photo added to the location on the map.	Correct photo from gallery added.
SE_05_TEST_08	FR4	Check if the user can add a photo for a location from the camera.	Photo from Camera	Photo added to the location on the map	Correct photo from camera added.
SE_05_TEST_09	FR3 / FR4	Check if coordinated are added to the map durring the walk.	No input (Automatic)	Coordinates of location shown on map.	Correct coordinates stored on map.
SE_05_TEST_10	FR5	Check if the walk data is deleted if the user cancels the tour.	Cancel command.	Return to home screen.	Tour data is deleted correctly and the home screen is displayed.
SE_05_TEST_11	FR6	Check that the message is successfully sent (see overleaf).	Tour object.	Message from server	MIME string is correct and not null.
SE_05_TEST_12	FR7	Check that WTC stores tour data when user switches android application.	Application switch.	n/a	Application is idle and stores data.
SE_05_TEST_13	FR7	WTC reloads data when user switches back to the application.	Resume application.	n/a	Reloads correct data.

2.4. System testing detail

SE_05_TEST_01:

This will test that the details (name and short description of tour), entered in by the user, are saved correctly when they creates a new tour Object.

SE_05_TEST_02:

When the activity starts the map should display an initial point (where the user is standing) using correct GPS coordinates. So this will test that the coordinates are the correct ones using the longitude and latitude in google maps app.

SE_05_TEST_03:

In the Specifications, the name of the walk has to be submitted as one word, so this test is to make sure that the user has only enters a single word for the tour name. There will be an error message that displays when the exception is called.

SE_05_TEST_04:

This test will check whether or not the correct location is added on the map. When the user enters the location that he/she wants, the coordinates of that location (google maps app) are used as a pinpoint to that location.

SE_05_TEST_05:

The short Description test will make sure that when the description of the location is added, the user cannot exceed more than 100 characters.

SE_05_TEST_06:

When a location is added, a time stamp is submitted alongside. We will test that the timestamp is in Time format (as described in Java Time class (*Time (int hour, int minute, int second))*) and that it is the correct time.

SE_05_TEST_07:

This will test that the photo uploaded into a location on the map is the correct photo from the user's gallery. We will also have to check that the location has no more than five images. If the location already has five images, an exception is thrown when adding the photo.

SE_05_TEST_08:

Similar to *SE_05_TEST_06* but the user adds an image from the camera.

SE_05_TEST_09:

This test will check that the coordinates added to the location on the map are in fact the correct coordinates. This will be checked using google maps app.

SE_05_TEST_10:

When the user cancels the walk by pressing the *cancel walk* button, we will have to test that the walk has been cancelled and the application returns to the home screen. This test will show that the walk has been removed from the phone.

SE_05_TEST_11:

The MIME message will have to be checked that it is formatted in the correct way before being sent over to the server. And also that the message is not null.

SE_05_TEST_12:

The test has to make sure that the Tour creator stores the tour data when the user switches android application.

SE_05_TEST_13:

When the user switches back to the Walking tour application, the correct data from the idle state must be loaded back into the application so the user can use the information that he/she added.

3. Server side testing

3.1. System testing table

Test Ref	Req being tested	Test content	Input	Output	Pass Criteria
SE-05-TEST-001	FR9	Check that the user can delete their tours.	A reference to a tour in the database associated with the user logged into the current session to be deleted.	A system message displayed on the site denoting success/failure.	The system displays a success message.
SE-05-TEST-002	FR9	Check that a basic user cannot delete another user's tours.	A reference to a tour in the database, which is not associated with the user logged into the current session to be deleted.	A system message displayed on the site denoting success/failure.	The system displays a failure message.
SE-05-TEST-003	FR9	Check that an Administrator can delete another user's tours.	A reference to a tour in the database, which is not associated with the administrator logged into the current session to be deleted.	A system message displayed on the site denoting success/failure.	The system displays a success message.
SE-05-TEST-004	FR9	Create new user account.	Enter "example@email.com" when prompted for an email. Enter "user1" when prompted for a username. Enter "password" when prompted for a password.	A new user record in the database.	A new user with the username "user1" with respective email address and password has been created.
SE-05-TEST-005	FR9	Search for a walk.	Search for "Aberystwyth".	A list of walks.	All walks relating to Aberystwyth are displayed.
SE-05-TEST-006	FR9	Log on to the website with a user who is on the database with the correct password.	Log on with username "user1" and the password "password".	System message.	System message confirms login success.
SE-05-TEST-007	FR9	Log on to the website with a user who is not on the database.	Log on with username "user2" and the password "password".	System message.	System message declares user does not exist.
SE-05-TEST-008	FR9	Log on to the website with a user who is on the database with the wrong password.	Log on with username "user1" and the password "password2".	System message.	System message declares password incorrect.
SE-05-TEST-009	FR8	Test that the sample route has correct GPS locations.	Co-ordinates of sample route.	Vector polygon plotted on the map.	The plotted line is the same as the co-ordinates of the sample route.
SE-05-TEST-010	FR8	Test that the points of interested are recorded along the sample tour.	Set of points of interest.	The points of interest plotted on the map.	The co-ordinates of the points of interest are plotted to the correct locations on the map.

SE-05-TEST-011	FR8	Test the pop-ups.	Click on a point of interest.	A CSS popup.	A CSS popup appears by the POI on the map. The title and short description are the same as sample route.
SE-05-TEST-012	FR8	Test image thumbnails in pop-ups.	Click on point of interest on the sample tour.	Strip of thumbnails.	The thumbnails represent all of the images associated with the point of interest.
SE-05-TEST-013	FR6	Check that the phone can send a HTTP post to the server.	A logged in user sending a request to the server, via the phone.	Site log file.	The log file is updated with the transaction associated with the user.
SE-05-TEST-014	FR6	Check that the data is formatted as a valid MIME message.	A String field in the post request.	The attachments and the JSON data from the MIME messages.	The PHP program is able to decode the MIME message and extract the data and the attachments.
SE-05-TEST-015	FR6	Check that the data is formatted as valid JSON.	String of JSON data.	Records of the tour and its relations in the database.	The record and its relations in the database contain the correct data in the correct fields.
SE-05-TEST-016	FR6	Check that the image files have been saved.	The image files associated with the sample route.	The file system on the server.	The images have been saved to the correct directory within the file system on the server.

3.2. System testing details

SE-05-TEST-001

- This test will check that a user can delete their own tours.

SE-05-TEST-002

- This test will check that a basic user (a user who doesn’t need have a log in) cannot delete another user’s tours.

SE-05-TEST-003

- This test is to check that an administrator can delete any tour.

SE-05-TEST-004

- This test is to create a new user, when creating a new user they will need to input an; email address, username and a password.

SE-05-TEST-005

- This test is checking the search function on the website, it does this by searching for a key-word i.e. “Aberystwyth”.

SE-05-TEST-006

- This test is to check that a user who has created an account can log on to the website.

SE-05-TEST-007

- This test is to check that a user who has not created an account can’t log on to the website.

SE-05-TEST-008

- This test is to check that a user who has created an account but enters the wrong password on the website can’t log on.

SE-05-TEST-009

- This is to test that the sample route (route used for testing only) has the correct GPS locations.

SE-05-TEST-010

- This test is to check that all of the points of interest are recorded along the sample tour.

SE-05-TEST-011

- The map will have popups at all of the points of interest, this test checks that the popups are showing the title and short description as a minimum, there may also be a picture of the POI.

SE-05-TEST-012

- This test will check that all of the thumbnails associated with the point of interest are displayed in the popup.

SE-05-TEST-013

- The phone needs to communicate with the server as to send the data associated with the walk. This test checks that the HTTP post is sent to the server and it has been recorded in the log file.

SE-05-TEST-014

- The data will be sent via HTTP post in a MIME message, this tests that it is a valid MIME message. This is achieved if the PHP program can decode the message and the relevant data is extracted.

SE-05-TEST-015

- When the MIME message arrives at the server it will need to be converted into files that will be stored in the database, we will be using JSON for this. This test checks that the data is formatted as valid JSON.

SE-05-TEST-016

- This test checks that the image files have been saved in the correct directory within the file system on the server

4. References

N/A

5. Document change history

Version	CCF No.	Date	Changes Made to the Document	Changed by
1.0	N/A	14/11/2013	N/A – First release of test specification.	srr11
1.1	N/A	15/11/2013	Corrected server side test table.	wia2