Project Group 05

Final report

|  |  |
| --- | --- |
| Authors: bmo; sr11; hac22; wia2;  wjl3; njv1  Config Ref: SE\_05\_TS\_01  Date: 11/11/13 Version: 2.0 Status: Release | Department of Computer Science  Aberystwyth University  Aberystwyth, Ceredigion  SY23 3DB  Copyright © Group 05  Aberystwyth University 2013 |
|  |  |

Table of Contents

[1. Introduction 3](#_Toc380273506)

[1.1. Purpose of this document 3](#_Toc380273507)

[1.2. Scope 3](#_Toc380273508)

[1.3. Objectives 3](#_Toc380273509)

[2. Management summary 4](#_Toc380273510)

[3. Historical account 4](#_Toc380273511)

[4. Final state 4](#_Toc380273512)

[5. Performance of each team member 5](#_Toc380273513)

[5.1. William Lea 5](#_Toc380273514)

[5.2. Harvey Clark 5](#_Toc380273515)

[5.3. William Arslett 6](#_Toc380273516)

[5.4. Stephen Paul Clasby 6](#_Toc380273517)

[6. Critical evaluation of the team and the project 7](#_Toc380273518)

[6.1. How did the team perform as a whole, and how could that have been improved? 7](#_Toc380273519)

[6.2. How could the project have been improved? 7](#_Toc380273520)

[6.3. What were the most important lessons learned about software projects and about working in teams? 7](#_Toc380273521)

[7. Appendices 7](#_Toc380273522)

[A. Project Test Report 7](#_Toc380273523)

[B. Project Maintenance manual 7](#_Toc380273524)

[C. Personal Reflective Reports 7](#_Toc380273525)

[D. Revised project plan 7](#_Toc380273526)

[E. Revised test specification 7](#_Toc380273527)

[F. Revised design specification 7](#_Toc380273528)

[8. References 7](#_Toc380273529)

[9. Document history 7](#_Toc380273530)

# Introduction

## Purpose of this document

The purpose of this document is to present the final state of the group project and its manual, detail the process of delivering the system and describe all members’ contribution and performance.

## Scope

This document contains the maintenance manual for the system, all the revised documents delivered during development including project plan, test specification, design specification as well as a testing section where the tests performed on the system and their results are detailed. The group members’ self-reflective reports and critical evaluation are also present. This document should be read by anyone wishing to use the Walking tour system or to modify it.

## Objectives

The aim of this document is to present the maintenance manual of the walking tour system and a relevant FAQ section that can be used to aid anyone wishing to use or modify it. It presents the details of how the project was developed and in what state it is in currently.

# Management summary

The project as it was submitted, is in an operational state, but with some limitations and some of the functional requirements not completely implemented.

On the android side of the system, the application works well and is able to fulfill most of the requirements. A walk can be created and it can have a name, a description and a long description. The requirement, to have a limitation on the character number, is in place and working. When the user decides to create a tour, the GPS tracking is activated and works. An option we have added, is that the user can define how often their position is being recorded, depending on the speed they are going at. Locations are saved along the route successfully. When the user decides, along the walk, so called key locations can be added. These are special locations that can have a name and description as well as a timestamp, which is to be used further on to determine the distance and length of the walk. After they are added, they appear as pins on the map. The user can add multiple photos (up to 5) per location after tapping the location and selecting the option to add a new photo. A limitation here is that the user can only add photos from the camera app, but not from the gallery, meaning the user will have to take the pictures they want associated with a location. The user can remove photos added to a location by selecting it and then pressing the delete photos button. If the user so chooses, they can remove a key location form the map similarly to the photos, by selecting a location and tapping the remove location button.

In the settings menu the option to cancel the walk is present. That ends the current walk and discards any data on the walk without uploading it to the server. When the user decides to finish their walk, they can do so by selecting the finish walk option and wait for the walk to be uploaded. After they are prompted with success, they can exit the app or create another tour. The upload is achieved with the data from the tour formatted as a JSON string inside a multipart MIME message sent as a HTTP POST request.

On the server side, the database intercepts this message decodes it and stores the data into the appropriate relations and records. The website, having a connection to the database then presents the user with the option to view the walk. The website consists of 4 main pages. The home screen allows for quick access to the most recent walks. Using AJAX the user can simply click on a tour name and it will be displayed instantly. Should the user decide to look at all walks, they can navigate to the list of tours page, and make their selection again. That will take them to a separate dedicated map page, where the map is larger and the focus is only on that one tour. Tours appear on the map as red lines, which connect the key and non-key locations. The key locations are pins, when clicked on display the location’s name, description and photos if any are present.

The documentation submitted alongside the project has been revised and is of good quality.

The Project plan has been updated after the initial changes to design were made to more clearly reflect them (removed some features, not specified in the requirements). The test specification has been updated with more precise system tests and more room for fault description. The initially included unit tests have been removed, but still implemented in the actual code. The design specification has been improved and updated with the latest information on the project.

# Historical account

The Project Management Plan was the first main event over the lifetime of the project. As a team we met up in various locations to discuss the main sections that needed to be completed for the deadline. In order to produce the first plan for the Project Plan, we firstly decided on the roles of each team member considering personal experiences within the fields. The very first hurdle that we tackled was the proposed system designs, figuring out the platforms and high level architecture that we were proposing to use and also the Use case diagrams for the android side and the server. These were mainly the tasks of the web and android teams. As well as the general design ideas, the both teams also had to create wireframes of the proposed GUI giving a reason why they chose these screens/pages. After each member of the team created written documents and image files of the design, the Management team then created the first draft of the document. During the same day, the web team had to change the use case diagram because of a few spelling mistakes with the actions.

The second main event was the Test Specification. All of the individuals in the group worked together for this document, to produce multiple tables and detailed descriptions of the tables. The android team worked along with the testing team in order to create module tests and system tests for the android device. And the web team also worked alongside the testing team to create similar documents for the server side system tests. After all of the final documents were completed, the management team put the Test Specification PDF together ready for submission. After creating the first draft of the document the web and testing teams decided to change the server side test table because we had missed out a lot of important content.

After creating the first two documents the team sat down and decided on how we were going to create the Design Specification. This became the busiest part of the project due to the detail that was needed from each team member. Knowing this, we all split into two groups; web and android team. The first main task within the design specification was the classes and components of the system; we planned to show structure and descriptions of them. One member of the android team designed all of the diagrams such as the component, sequence and UML class, whilst the other filled out the class or interface code with JavaDoc and comments. The web team also had to create a component and sequence diagram for the server side design. The second main task for the Design Specification was the Algorithms and data structures that we were proposing to use within the server and android sides. The two main team members completed this part of the task whilst the rest of the team worked on the existing document feedback. [1] As well as completing the Algorithms and data structures the web and android teams put together sample data for the MIME strings and JSON code and the management team added them into the document as appendices.

[1] The team came together after the Design Specification was completed to work on the submitted documents. In order to efficiently get each document updated the management team handed out roles for each member, from looking through the files comparing them to the feedback to readjusting work to fit in with the new, improved design. The main changes made to the Project Management Plan were the user account details that we had included in the first design. To change this we had to remove the Log in action from the android use-case along with the create account included. We also had to remove a lot of actions within the web side use case diagram, because we had included user preferences within the site too. Not only did the use cases change, the QA manager had to change the wireframes for both of the programs because they had originally included the user preferences.

The next document that the team had to work on was the Test Specification, alongside the prototype demo. The testing team had to work together with the android team for this event in the project, in order to create the test log forms and improve the testing tables.

Whilst getting ready for the demonstration of the prototype, the team had a full week in the Orchard on campus. In order to successfully complete this main event in the project, the team worked together to make sure that we had a prototype to demo. Throughout the week, the team split up into the assigned groups to work on specified tasks that the Project Leader had given. The android team worked on the general code, making sure that it tied in with the functional requirements. The web team also did the same, as well as making sure that the two systems worked together in unison. As we neared the end of the week, the testing team managed to apply the system tests to the application and the website, deciding whether they had passed or failed.

After receiving the feedback to the Design Specification, this became the main focus for the android and management teams. The teams had to work together to change the decomposition description due to the layout of the application changing, from previously stated designs. Many of the class and interface names had changed as well as the UML diagram structure. To complete the task we assigned the main programmer from the android team to go through the code and change what we discussed. Not only did the layout of the application change from original design implications, the web layout also had to change. Whilst the two teams worked on this document, the rest of the group were concentrating on specified sections of the Final Report given to them by the Project Leader. This included personal reports as well as group tasks.

Throughout the whole project, there has been up to eighty hours on average, from each team member. These hours include meetings as well as personal work on the project. In order to complete the milestones that were set these were the kinds of hours that were deemed needed from each member.

# Final state

# Performance of each team member

## William Lea

William Lea took the initiative and started researching the android development side right away. He attended the android development seminar and showed his ability quickly. He also supplied the group with the testing device and devoted a lot of time keeping the rest of the group updated on the current state of the application. William attended all but a few of the meetings with our project manager and had legitimate reasons for missing the ones that he did. He attended all the group meetings dedicated to the android development and all other briefings we had across the first semester. William’s contribution towards the project’s completion was invaluable.

When addressing his teammates William was always respectful and helpful. Any questions and suggestions he got from the rest of the team was discussed and always considered during development. He proved to be able to work alone when needed and as part of a team. Most of the development was done with him and Harvey Clark working together on the android side. The two of them worked quickly and efficiently, proving they found a common language.

## Harvey Clark

Harvey Clark was passionate about android development and along with William Lea, he went to the induction seminar, did research and worked towards software completion. Harvey proved to be very reliable as he was able to follow all of the project guidelines the project manager, leader and QA gave towards the android application. He showed his ability to deliver high quality level code: well documented, well structured, reliable and robust. Harvey attended most of the meetings with the project manager and had legitimate excuses for missing the ones he did, he attended all the group meetings where the android app was in focus and all other group briefings throughout the first semester. Harvey’s contribution towards the project’s completion was invaluable.

When teamwork was required, Harvey proved to be a team player, he was honest and to the point with everyone and was ready to help anyone that needed him to. When working with William or alone, he worked quickly and efficiently and produced the results he was expected to.

## William Arslett

William Arslett was experienced in database design and administration, hence why he was in charge of that part of the project. He proved to be responsible as he volunteered to go to the GitHub seminar and was ready to explain to the whole group how to use it properly, as well as he volunteered to keep the meeting minutes. He demonstrated great teamwork and ability to communicate efficiently with the whole group and mainly with his partner Stephen with whom he worked very closely.

As his job was to deliver the database he had to actively communicate with the android team and the website team to ensure a proper connection between the two with the database and he managed to do that providing great results. He delivered code on time and with good functionality and was able to establish the link between both sides of the system. He attended all the group meetings where the main topic was website and database development and most meetings with the project manager. William’s contribution towards the project’s completion was invaluable.

## Stephen Paul Clasby

Stephen Clasby was given the task to develop the website and connect to the database. He had previous experience and showed passion for working with web-based applications. He worked closely with William Arslett and proved to be a good communicator. During the project’s development he was active and produced good results in a timely manner. He had to develop the website so that it would be usable on any device and produce the same desired effect. Implementation of the mapping API was also his responsibility.

Stephen attended most meetings with the project manager and group meetings dedicated to the website development. He was mostly absent from the group briefings, but was available online to communicate from a distance.

## Neil Vicker

Neil Vicker was given the task to develop and implement the whitebox and blackbox testing of the system. He worked closely with Benjamin O’Donovan, the group’s Quality Assurance manager to develop the tests and then to later run them on the system. He was to also research and produce a report about the multiple mapping APIs that we were considering to use. He presented his findings in front of the group and the group decided to choose the Leaflet mapping API. Neil was quick and efficient and completed all of his assigned tasks on time. He proved to be a good communicator by working with both the android and web side of the project to develop the appropriate system level tests.

Neil attended most meetings with the project manager and most meetings the group help do discuss the project in general.

## Benjamin O’Donovan

Benjamin was assigned as the group’s Quality Assurance manager and deputy leader. He showed passion and determination while working on the project. As QA he controlled the process of development making sure all the requirements are met and that the quality of code that was delivered was at the level expected. He proved to be a great communicator as his job involved constant contact with all group members. Benjamin also assisted Neil with the system testing. He developed tests alongside him, making sure they answer the standards. He performed and documented the testing done on the application and website.

When working with the rest of the group he addressed everyone with respect and was ready to help out anyone that asked for help. Benjamin attended all the meetings the group held with the project manager, as well as most of the meetings the android and web teams held on their own. He attended all project briefings.

## Sevastiyan Raychev

Sevastiyan was assigned to the task of project leader. His tasks included organizing meetings for the group, discussing any issues the group faced during development and answering any questions that the group had regarding the project. As group leader he had to keep in constant contact with all members of the group and to make sure every one of them had a task to work on and that that task was to be completed in a timely manner and that it would be what was expected. He was to develop a Gantt chart detailing the milestones the group was facing, who was to tackle them and how much time they had to finish. He also dealt with the majority of the documentation with help from the group QA.

Sevastiyan took into account everyone’s opinion and made an effort to hear everyone out and implement their ideas into the project. Sevastiyan attended all the meetings with the project manager and most of the meetings for the android and web sides and all of the project briefings.

# Critical evaluation of the team and the project

## How did the team perform as a whole, and how could that have been improved?

The team performed well as a whole. The biggest issue in the beginning was communication. That led to some uncertainties and miscommunication, but soon after we started working, we managed to solve the issue effectively. The biggest issue we faced really was the fact that out android developers were not able to work on campus, which meant that in order to compile and put the app on the device, someone would have to go down to town and back, which didn’t waste as much time as we thought it would, because we just worked form a distance, but it could have been avoided. The project faced the issue of requirement creep. In the beginning we thought we would have to add a user system and we planned with that in mind, but later we decided to drop the feature for now as it would have slowed us down. The task assignment could have been different, to include some members that were left with less work at some points while others were juggling multiple tasks, but sometimes just the skills and knowledge of the person decided what tasks they would have to take on. To improve this situation the group took on pair programming to make sure that there was not only one person that knew everything about that one task. There is still room to improve on this though.

## How could the project have been improved?

The project could have been improved by making sure all the features were delivered on time and by extension that all the functionality was in place at the deadline. From the management side, one thing that could be improved would be the milestones set. We had set milestones but they were large and not as clear as possible. If they were split into smaller increments it would have made development a little faster.

## What were the most important lessons learned about software projects and about working in teams?

The most important things learned were that communication is key in any group task, but also that there need to be measures taken so that in the situation where communication is impossible or bad, the results of that do not lead to catastrophic consequences.

The group learned what the consequences of requirements creep are and how important it is to deal with them as soon as possible.

The importance of time management was very important. When a deadline was close the group was ready to present when the time of the members was used appropriately and tasks were spread out throughout and were clear and exact.

# Appendices

## Project Test Report

## Project Maintenance manual

**WTC app maintenance guide**

**Program Description**

The Walking Tour Creator app, WTC for short, is used to create walking tours of towns, cities or smaller areas of interest. A user will enter the details of the area they are making a tour for. These details include a single word name for the area they are touring, a short description for the area they are touring and a long description for the area they are touring. Whilst making the tour, the user will be able to add key points of interest to the tour. As they do this, they will be prompted for the name of this point of interest and a description for the point of interest. The user is able to add photos to the points of interest and also remove those photos. The user is also able to remove key points of interest they have added. The user is able to cancel a walking tour they are in the process of making, this will erase all the locations they have added to the tour. Finally the user is able to save their walking tour to an online database and view the tour they made.

**Program Structure**

The app is separated into three activities:

* MainActivity - this is the launcher activity for the app, when the app first starts this is the screen the user is presented with.
* WalkActivity - this is the main part of the app, this where the user builds their tour. From here they can:
  + Add / Remove points of interest.
  + Add / Remove photos to / from a point of interest.
  + Cancel their walk.
  + Save their walk for later viewing.
  + They are also able to adjust the sample rate for non-key locations to save on how much data they transmit to the server.
* AboutActivity - this just a credits screen, it just says who is on the team, the app name and version number.

There are also some supporting UI classes for dialogs.

* NewWalkFragment - sets the details of the walk and starts WalkActivity with that information.
* EndWalkFragment - ends the walk if the user selects ‘OK’.
* FinishWalkFragment - does the same as EndWalkFragment but saves the walk to the server before quitting.
* NoNetworkFragment - alerts the user that there is no network available.
* LocationDetailsFragment - gets the details for a key location from the user.

**Main Data Areas**

The main data area of the app is under the package **com.wtc.grp5.model.**  This package contains the classes which contain the data for a tour.

**WTCTour -** This class represents the walking tour the user will create.

|  |  |  |
| --- | --- | --- |
| **Field type** | **Field name** | **Description** |
| String | tourName | The name of this tour. |
| String | shortDesc | The short description of this. |
| String | longDesc | The long description of this tour. |
| LinkedList<WTCLocation> | locations | The list of locations in the tour (includes key locations by inheritance). |

**WTCLocation -** This class represents non-key locations in the tour (used for tracing lines between points on the website map)

|  |  |  |
| --- | --- | --- |
| **Field type** | **Field name** | **Description** |
| double | longitude | The longitude of this location. |
| double | latitude | The latitude of the location. |
| Calendar | oldTime | The original time stamp in a date/time format. |
| long | timeStamp | The timestamp for this location as minutes since the start of the walk. |

**WTCKeyLocation -** This class represents key points of interest in the tour.

|  |  |  |
| --- | --- | --- |
| **Field type** | **Field name** | **Description** |
| List<String> | photos | The list of file paths for photos the user adds to this key location. |
| String | locName | The name of this key location. |
| String | locDesc | The description of this key location. |

**Algorithms**

**Saving to the server - main thread**

Check network connectivity

**If** connected

Set up async task to save to server with the server URL and Tour data.

Execute async task.

**Else** tell the user they have no network connectivity.

**Saving to the server - async task**

Adjust the timestamps of the locations in the tour minutes since the tour start.

Set up the HTTP Client.

Set up the POST message with the URL from WalKActivity.

Set up the Multi-part MIME.

Add Name/Value pair for the JSON data of the tour.

**For each** key location **in** the tour

**For each** photo file path **in** key location photo list.

Add photo file path to Name/Value pair list.

**For each** Name/Value pair

Add Name/Value pair to the Multi-part MIME.

Send the POST data to the server.

**Adding key locations to the tour**

User presses the ‘Add Key Location’ button.

User is prompted for the name and description of the key location they’re making.

User enters the details in presses ‘OK’.

A map marker is added to map at the user’s current location.

The marker title is given the name of the location the user entered.

The marker description is given the description of the location user entered.

The user’s current longitude and latitude are put in a key location object and added to the tour list.

**Removing key locations from the tour**

User selects the marker for the key location.

User presses the ‘Remove Key Location’ button.

**For each** location **in** thetour

**If** location’s longitude and latitude match marker’s longitude and latitude.

Remove location from tour.

**Adding a photo to a location**

User selects the marker for the location they want to add a photo to.

User presses ‘Add Photo’ button.

Image file is created.

Camera app is started.

User takes a photo.

Image saved to the image file.

**For each** Location **in** Tour

**If** currently selected marker’s longitude and latitude match Location’s longitude and latitude

add photo file path to Location.

**Files**

The app uses a serialised data file, which is made via the TourSave class, to store the tour when the user navigates away from WalkActivity or even the entire app. The app also produces JPEG files for the pictures the user takes. The facilities for saving the pictures are provided by the Android OS, however, the naming for the picture files is handled by the app and the names of the pictures is based on the date and time it was taken to ensure name uniqueness.

**Physical Limitations**

The physical limitations of this program are the need for network communication (mobile data Wi-Fi) and GPS signal. The GPS signal, obviously is for recording the user’s location into the tour and the network is for saving the tour to the server. Without either of these prerequisites the app can’t function properly.

**Suggestions for Improvements**

The first improvement to make would be to add the ability to save a walk temporarily when the user selects ‘Finish Walk’ and there is no network. Another improvement that is needed is the ability for the user to be able to see the photos they have taken for the locations in the walk. The user should also be able to add a photo to locations from their device’s photo gallery.

**Rebuilding and Testing**

To build/rebuild and test this app you will need:

* The Eclipse IDE with the ADT plugin.
* A copy of the Google Play Services Library project to put in the same workspace as the app project.
* The Google Play Services SDK.
* An Android device with at least Android 4 installed on it.

For actual system testing you need to install the app on your Android device. To do this just plug it into the computer you developing on and press “Build/Run” in Eclipse. You may be prompted for a device to build the app to. Just select your phone and press “OK”. The app should start up on its own.

## Personal Reflective Reports

## William Lea

Overall I would say that our group progressed and worked well together, in spite of some communication breakdowns. We were able to meet the deadlines for the deliverables, even if sometimes we did other tasks better than others. For example, the first prototype was severely lacking and we weren’t able to demonstrate any real capabilities of our system.

At the end of the project around coding week there were multiple slip ups with the Android app. They weren’t really massive issues but they were exacerbated by the project setup for the app. The app was only runnable on my home computer. Not thinking ahead, I elected to add a more involved UI using google maps. Since the university machines didn’t have the Google Play Services SDK, the app couldn’t be compiled and run/tested at the university with the rest of my team.

At the start of the project during the design phase we started off quite well with organising the meetings and planning things. However things slowed down a fair amount, there were was a week or two where nothing seemed to have progressed but we eventually got the ball rolling again.

The Android team (Harvey and myself) had regular Google Hangouts to work on the design of the app. We worked on the Class diagrams, Sequence diagrams, etc. We ended up redesigning it several times due to few issues with Android.

By the time we were meant to start prototyping things were moving slowly. I felt this was due to other assignments had to do around this time. Though I am unsure of what the web team were doing as far as prototyping goes, I spent around an hour every other night building the prototype.

When it came to finally showing the prototype, I felt the prototype was lacking. As far test info goes, the only thing that worked for testing was the server communication. The other buttons just output a message to the user about the action they performed. For example, if they pressed the add location button it would output “Location added”.

What I would do better if we did another group project.

As far as doing another group project goes, the first thing I would do is bear in mind my development environment. When building the app, we had the idea to put a map on the app so the user can get a preview of what their walk would look like. This meant we needed addition APIs which the university did not have! So, because of this I had to do the Android stuff on my home computer, which would have been fine if it was a laptop. So next time I think it would be better to first check what I’m working with first before picking features. Another thing that the group as a whole could do better is communication. We had several communication breakdowns that lead work not being done. However, as far as meetings go we did alright, we would often meet on Wednesdays to discuss what had been achieved that before we went into our meetings with our project manager. In addition to these meetings, our teams had extra meetings to discuss how they were approaching their tasks. I.e. the web team discussed the database, mapping API and the frontend of the site. The Android team would, for example, discuss things like how to represent the tour in Java, how to collect GPS locations and the UI of the app.

## Harvey Clark

## William Arslett

## Stephen Paul Clasby

The project to design and create a Tour creation app and website to complement; I enjoyed the project. The roles were chosen based on the strengths and weaknesses of each person, I chose to be on the web side of the project as I was more suited to that task rather than for the android. I created the design and layout based around the group’s feedback after each modification; I used PHP to generate the webpages based on what was in that pages PHP file after it generated the template. I did encounter a few troubles during this phase of development with the Leaflet API, which required the layout to be modified. I did have some problems while I was refactoring “connection.php” which I had to rewrite some of the methods, so they didn’t depend too much on what was above. The group as a whole worked well together, there was a bit of a delay between the two teams in the group which frustrated me, but was only a minor setback, and I used the time to tidy up the website and the code, while we waited for the android app to catch up. The code in the template and the src files could be tidier, commented and overall improved but due to the time constraint I had to fit everything into the time scale required.

The front page needed to have a map with a list that was generated from the database and displayed neatly; I created a few more areas on the webpage and added another CSS entry for the front page map to display correctly. The PHP code needed loop through the database and displayed the entries as hyperlinks which linked the ID into the map thus loading all the relevant information from the database. This code I feel could be better, but there wasn’t enough time, so I created a simple script to do what was required as fast as possible.

Living twenty miles away from the university hindered me somewhat as I couldn’t attend all the group meetings due to bus time constraints; so there was some delay on what was discussed and what tasks were handed to me. The group’s chemistry as a whole was well; there wasn’t any clashes of personality between members of the group or myself, the general attitude of the group was well and focused on the task at hand.

## Neil Vicker

## Benjamin O’Donovan

## Sevastiyan Raychev

## Revised project plan

Project Group 05

Project Management Plan

Authors: bmo; sr11; hac22; wia2;

wjl3; njv1

Config Ref: SE\_05\_TS\_01

Date: 11/11/13 Version: 2.0 Status: Release Department of Computer Science

Aberystwyth University

Aberystwyth, Ceredigion

SY23 3DB

Copyright © Group 05

Aberystwyth University 2013

Contents

[1. Introduction 3](#_Toc378680224)

[1.1. Purpose of this document 3](#_Toc378680225)

[1.2. Scope 3](#_Toc378680226)

[1.3. Objectives 3](#_Toc378680227)

[2. Overview 4](#_Toc378680228)

[2.1. Platforms and high level architecture 4](#_Toc378680229)

[2.2. Target user base 4](#_Toc378680230)

[3. Use case 5](#_Toc378680231)

[3.1. Android Use Case 5](#_Toc378680232)

[3.2. Descriptions of the Android Use Case 6](#_Toc378680233)

[3.2.2. Settings 6](#_Toc378680234)

[3.3. Web Side Use Case 7](#_Toc378680235)

[3.4. Description of the Web Side Use Case 8](#_Toc378680236)

[4. Android User Interface Design 9](#_Toc378680237)

[4.1. Main Screen 9](#_Toc378680238)

[4.2. About screen 9](#_Toc378680239)

[4.3. Map screen 9](#_Toc378680240)

[4.4. User Navigation 10](#_Toc378680241)

[5. Website User Interface Design 11](#_Toc378680242)

[5.1. Home Page 11](#_Toc378680243)

[5.2. Map page 11](#_Toc378680244)

[5.3. About page 11](#_Toc378680245)

[5.4. Overview 11](#_Toc378680246)

[6. Gant Chart 12](#_Toc378680247)

[7. Risk Assessment 13](#_Toc378680248)

[8. References 14](#_Toc378680249)

[9. Document History 14](#_Toc378680250)

1. **Introduction**
   1. Purpose of this document

The purpose of this document is to show how the project group has decided to carry out the client’s   requirement specification for a walking tour application as a set of objectives and milestones.

* 1. Scope

This document contains the details of the group project but does not go into detail about design, testing or maintenance. It contains the group’s choice of platforms and high level architecture as well as justification for the choices made. The Use-Case diagram is included showing how different users will interact with the different parts of the system as well as screenshots and descriptions of the GUI. The proposed Gantt chart is part of this document, detailing what the group will be working on in the process of development and in what timeframes. [1] This document was created after familiarisation with the Project Management Standards. [2]

* 1. **Objectives**

The objectives of this document are as follows:

* + 1. To describe the overview of the proposed system;
    2. To describe how the main components of the system will interact with each other;
    3. To present the base user interface and describe how the user will interact with it;
    4. To provide a list of the project milestones;
    5. To provide a list of all tasks that need to be completed on the project and their anticipated timeframe in the form of a Gantt chart;
    6. To list possible issues the team might encounter during development in the form of risk analysis.

1. **Overview**

The proposed system is a walking tour android application that allows people to “record” walks they make through GPS and add information and pictures at points they find something interesting. The walks will be available to view on a website for everyone.

* 1. **Platforms and high level architecture**
     1. **Android**

The platform has been specified by the client in the project guideline.

* + 1. **IDE**

We are using the Eclipse IDE with the ADT plugin because that is the IDE the team is most familiar with. We took into account Android Studio but reached the conclusion that Eclipse was more stable provided a better user interface.

* + 1. **Android Mapping API**

We are using Google Maps for the Android mapping API because it comes as part of the Android SKD and gives the user a full screen map to view, which the programmer can overlay with their own controls. This extends the FR8 requirement in the requirement specification document [1] to allow for the user to view the map on their android device.

* + 1. **PHP**

The research showed that PHP is capable of processing easily JSON files, which we will be using, also it is available on most servers and is currently taught in one of the second year modules.

* + 1. **JSON**

JSON would be the best data set to use for sending the information about the recorded walk to the server. This is largely due to it being significantly lighter in weight than XML and how easy it is to process in PHP.

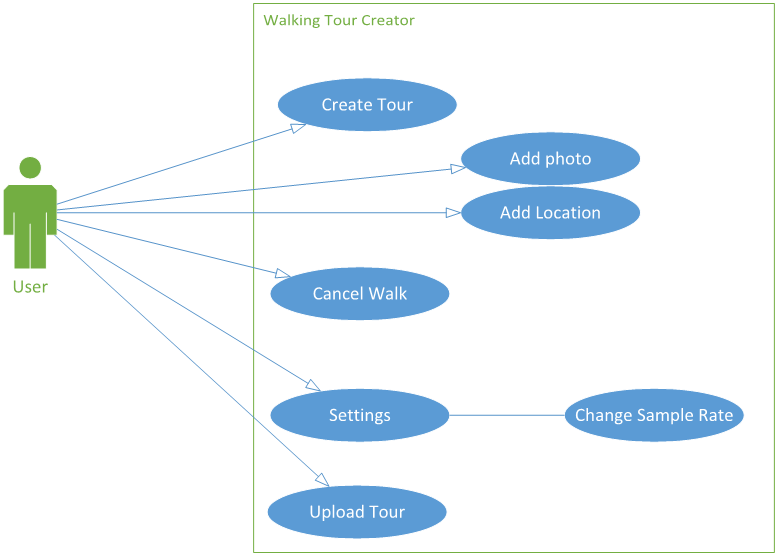
* + 1. **Web side mapping API**

Because of its wide variety of browser support, features and simplicity, the Leaflet API stood out as exactly what the project group would need to use.

* 1. **Target user base**

The client has suggested second year computer science students as target base, but the app is not targeted specifically at that user group. Its purpose can be different for different users thus making it usable in a variety of ways and the actual system has to be easy to use by most age/background groups

1. **Use case**
   1. **Android Use Case**



* 1. **Descriptions of the Android Use Case**
     1. **Create Tour:**

This will allow the user to create a new walking tour, regardless of logging in to our servers or not the user will be asked to select a title for their tour and a short description (as minimal) before starting the tour, during the tour they can attach photos of local scenery and the like with a short description of the photo.

* + - 1. **Add Photo:**

This will use the built-in camera app on the android device to take a photo for the user to add to their walk.

* + - 1. **Add Location:**

This will get called periodically to allow for an accurate walking tour to be created.

* + - 1. **End Walk:**

When the user presses to end the tour, they will be given a summary of the tour, and be asked to fill in the missing long description (if they did not do so before creating the tour), they will then be prompted to see if they want to have the walk saved locally or uploaded straight away.

* + 1. **Settings**:

This is where the user will give their preferences for different in app options, such as the upload option, if a user is concerned about their data limits they can choose to only upload over Wi-Fi.

* + - 1. **Upload Tour:**

If the user has chosen to save their tours locally on the device, this option will allow the user to select which tours they wish to upload to the website.

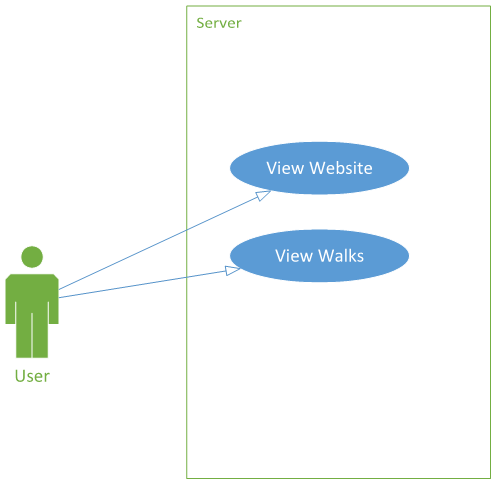
* + - 1. **Cancel Tour:**

This will allow the user the ability to cancel a tour at any time if they desire.

* + - 1. **Change sample rate:**

The sample rates that the user can change to are two, four and six seconds.

* 1. **Web Side Use Case**



Log i

* 1. **Description of the Web Side Use Case**
     1. **View Website**

Displays our walking tour viewer homepage.

* + 1. **Database Dependent Use Cases**
       1. **Manage Walks**

Allows the user the ability to edit/delete previous walks on their profile.

* + - 1. **View Tour:**

This will display a map for the user to view with "pins" in it that have pictures attached along with the notes associated with the walk.

* + - 1. **Search Walks:**

Searches based on keyword/location and delivers the top “x” amount of walks.

* + - 1. **Receive Tour:**

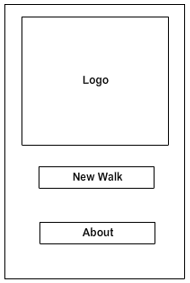
Takes a MIME file sent from the app, decodes it and stores the information in a

SQL database.

1. **Android User Interface Design**

The following wireframes are initial concept for the design and will be modified accordingly with time.

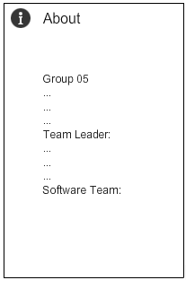
* 1. **Main Screen**



On the main screen the user can see whether he is logged in or not. The start button will take the user to the preparation screen for a recording where they can add descriptions and a title for the walk.

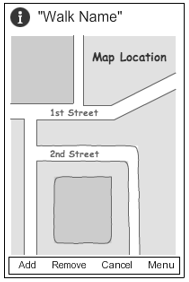
The settings and Info buttons take the user to the specified sub-screen.

* 1. **About screen**



This screen displays information on the app such as version, development team and so on. The user will be able to get contact details for the group from there.

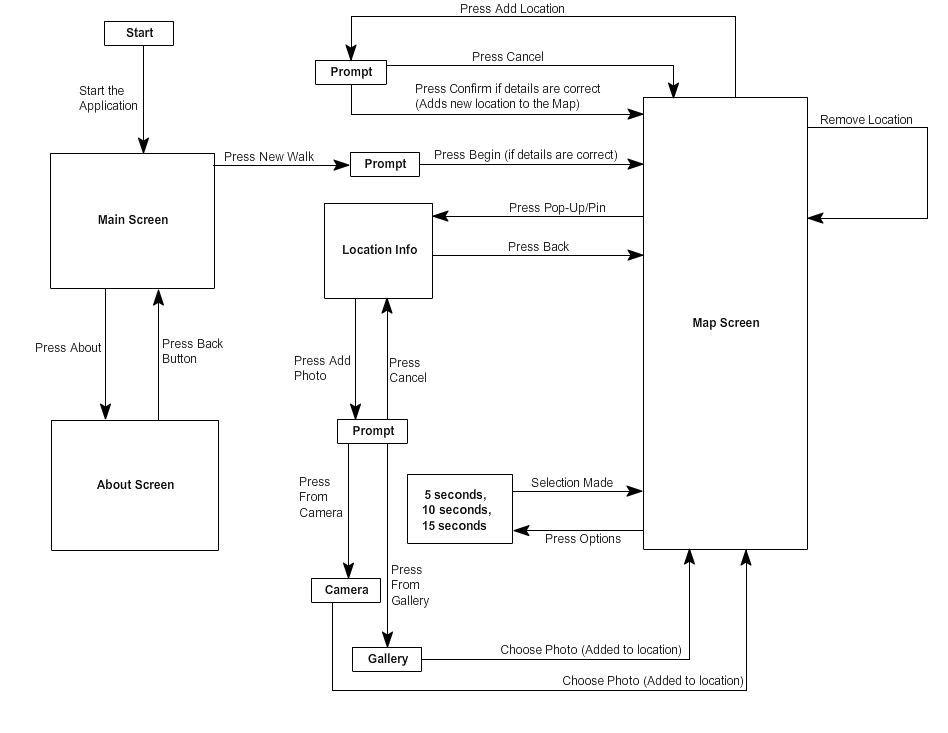
* 1. **Map screen**



On the map screen the user will see his position on the map and the path that he has already walked on. They can use the add photo button to select/take a photo and add a description to it. The cancel walk will exit the walk without saving it.

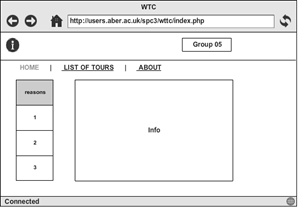
* 1. **User Navigation**

Below is a flow Diagram that describes the relationships between the screens in the Android App. The arrows show the direction of the link and what the user needs to do in order to follow that arrow.



1. **Website User Interface Design**

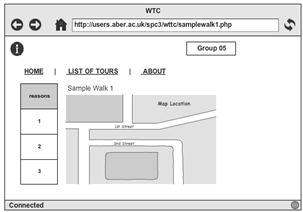
   2. **Home Page**



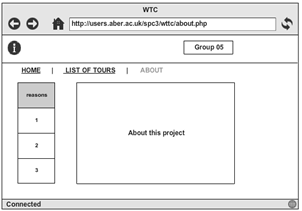
* 1. **List of Walks page**



* 1. **Map page**

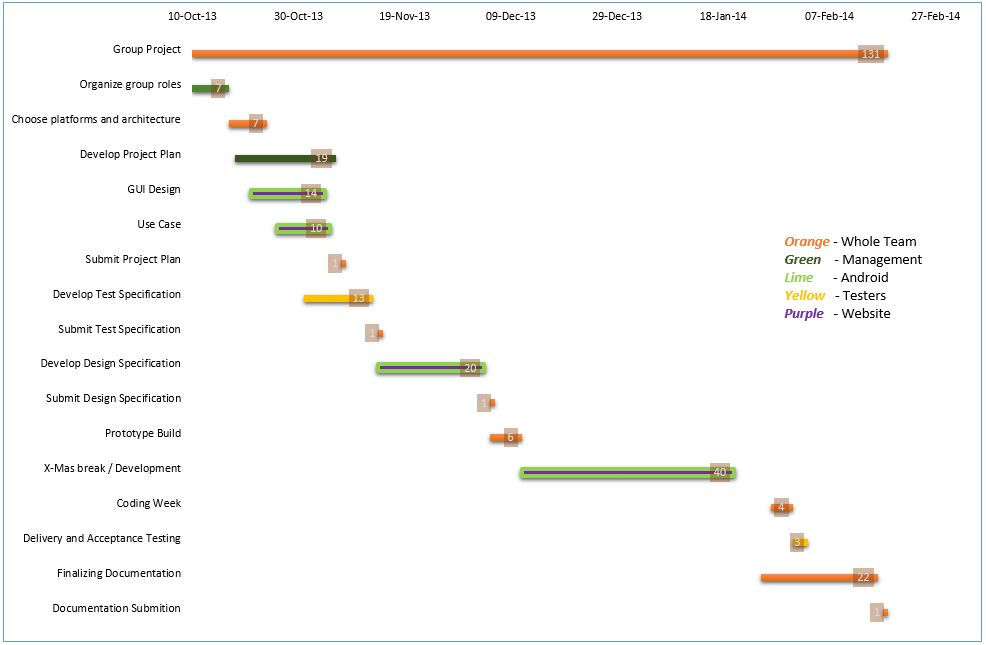


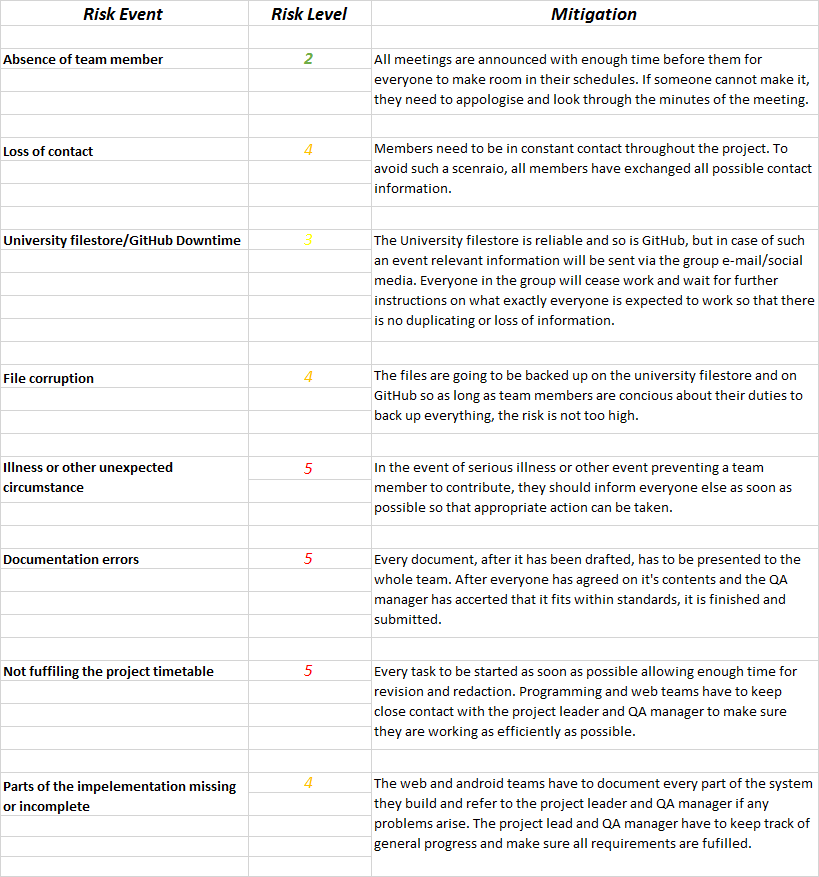
* 1. **About page**



* 1. **Overview**

The site will consist of 3 main pages. The home page, where the user can see information about the project, view a tutorial on how to use the website and navigate further. The map page, where a user can load and view a walk that is in the database. The about page where the project group’s contacts are listed so that users can give feedback to us about the site.

1. **Gant Chart**
2. **Risk Assessment**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Risk Level** | ***2*** | ***3*** | ***4*** | ***5*** | ***6*** |
| **Consequences** | ***Low*** | ***Low/Med*** | ***Med*** | ***Med/High*** | ***High*** |

1. **References**

[1] QA Document SE.QA.RS – Requirement Specification.

[2] QA Document SE.QA.02 – Project Management Standards.

1. **Document History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | CCF No. | Date | Changes Made to the Document | Changed by: |
| 1.0 | N/A | 06/11/2013 | N/A – First release of project plan | srr11 |
| 1.1 | N/A | 06/11/2013 | Changed the Web Side Use Case | bmo |
| 1.2 | N/A | 03/12/2013 | 2.1.3 - Referenced requirement specification document.  2.1.5 - Added where JSON is being used.  4.2 - Removed the log in and delete walk feature.  4.4 - Removed subsection entirely. | njv1 |
| 2.0 | #1 | 28/01/2013 | Updated information about the project and methodologies used in development, updated wireframes for the android app and the website, updated the UML Use-Case diagrams for both platforms. Updated Gantt chart. | srr11 |

## Revised test specification

Group Project 05

Test Specification

Authors: bmo; sr11; hac22; wia2; Department of Computer Science

wjl3; njv1 Aberystwyth University

Config Ref: SE\_05\_TS\_01 Aberystwyth, Ceredigion

Date: 14/11/13 SY23 3DB

Version: 1.2 Copyright © Group 05

Status: Draft Aberystwyth University 2013

Contents

[1. Introduction 3](#_Toc378673306)

[1.1. Purpose of this document 3](#_Toc378673307)

[1.2. Scope 3](#_Toc378673308)

[1.3. Objectives 3](#_Toc378673309)

[2. Android side testing 4](#_Toc378673310)

[2.1. System testing table 4](#_Toc378673311)

[3. Server side testing 5](#_Toc378673312)

[3.1. System testing table 5](#_Toc378673313)

[4. Appendix A: Test Log Form 6](#_Toc378673314)

[5. References 7](#_Toc378673315)

[6. Document change history 7](#_Toc378673316)

# Introduction

## Purpose of this document

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

## Scope

This document includes all of the system tests that the team will implement on the android and server side of the application.

This Document should be read after familiarisation with the Project Management Plan [1]. The specifications listed in Functional Requirements [2] and the Test Procedure Standards [3].

## Objectives

The objective of this document is to show what tests will be used to make sure the application, the website and database will meet the 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.

# Android side testing

## System testing table

# Server side testing

## System testing table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Ref | Req being  tested | Test content | Unexpected Input | Expected Input | Output | Pass Criteria |
| SE\_05\_TEST\_15 | FR8 | Test that the sample route has correct GPS locations. | N/A | 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\_16 | FR8 | Test that the points of interest are recorded along the sample tour. | N/A | 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\_17 | FR8 | Test that the pop-up shows up in the correct location and it holds the correct information. | N/A | Click on a point of  interest. | A CSS popup with correct information. | A css popup appears by the POI on the map. The title and short description are the same as the correct route. |
| SE\_05\_TEST\_18 | FR8 | Test image thumbnails in pop-ups and check that they are the correct images. | N/A | 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\_19 | FR6 | Check that the phone can send a HTTP post to the server. | Empty POST request. | A 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\_20 | FR6 | Check that the data is  formatted as a valid MIME message. | Invalid 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\_21 | FR6 | Check that the data is  formatted as valid JSON. | Invalid 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\_22 | FR6 | Check that the image files have been saved. | Image in the wrong format or missing an  expected image | 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. |

# Appendix A: Test Log Forms

|  |  |  |
| --- | --- | --- |
| Test Log No: 001 | Group: 05 | Testers(s): bmo, njv1 |
| Date: 28/01/2014 | Tagged version ID: 0.1 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Pass / Fail | Fail description | CCF / issue # |
| SE-F-001 | Pass |  |  |
| SE-F-001.1 | Pass |  |  |
| SE-F-002 | Pass |  |  |
| SE-F-003 | Pass |  |  |
| SE-F-004 | Pass |  |  |
| SE-F-005 | Pass |  |  |
| SE-F-006 | Pass |  |  |
| SE-F-007 | Fail | The camera works and takes an image but it doesn't add to the location pop-up. | #3 |
| SE-F-008 | Fail | The camera works and takes an image but it doesn't add to the location pop-up. | #3 |
| SE-F-009 | Fail | It is in the Model, but not in the App | #6 |
| SE-F-010 | Pass |  |  |
| SE-F-011 | Pass |  |  |
| SE-F-012 | Pass |  |  |
| SE-F-013 | Pass |  |  |
| SE-F-014 | Pass |  |  |
| SE-F-015 | Pass |  |  |
| SE-F-016 | Pass |  |  |
| SE-F-017 | Pass |  |  |
| SE-F-018 | Fail | Multiple thumbnails are showing with invalid images inside | #7 |
| SE-F-019 | Pass |  |  |
| SE-F-020 | Pass |  |  |
| SE-F-021 | Pass |  |  |
| SE-F-022 | Fail | The image files are being saved as Null | #5 |

|  |  |  |
| --- | --- | --- |
| Test Log No: 002 | Group: 05 | Testers(s): bmo, srr11 |
| Date: 28/01/2014 | Tagged version ID: 0.2 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Pass / Fail | Fail description | CCF / issue # |
| SE-F-001 | Pass |  |  |
| SE-F-001.1 | Pass |  |  |
| SE-F-002 | Pass |  |  |
| SE-F-003 | Pass |  |  |
| SE-F-004 | Pass |  |  |
| SE-F-005 | Pass |  |  |
| SE-F-006 | Fail | Not completed yet | #4 |
| SE-F-007 | Fail | Not completed yet | #4 |
| SE-F-008 | Fail | Not completed yet | #4 |
| SE-F-009 | Fail | Not completed yet | #4 |
| SE-F-010 | Pass |  |  |
| SE-F-011 | Pass |  |  |
| SE-F-012 | Pass |  |  |
| SE-F-013 | Pass |  |  |
| SE-F-014 | Pass |  |  |
| SE-F-015 | Pass |  |  |
| SE-F-016 | Pass |  |  |
| SE-F-017 | Pass |  |  |
| SE-F-018 | Fail | Not completed yet | #4 |
| SE-F-019 | Pass |  |  |
| SE-F-020 | Pass |  |  |
| SE-F-021 | Pass |  |  |
| SE-F-022 | Fail | The image files are being saved as Null | #5 |

# References

[1] Software Engineering Group 05. Project Plan. S. Raychev, B. O’Donovan, H. Clark, W. Arslett, W. Lea, N. Vicker and S. Clasby. 1.2 Release.

[2] Software Engineering Group 05. Requirements Specification. C. J. Price and B. P. Tiddeman. SE.QA.RS. . 1.4 Release.

[3] Software Engineering Group 05. Test Procedure Standards. C. J. Price, N. W. Hardy and B. P. Tiddeman. SE.QA.06. . 1.7 Release.

# 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 |
| 1.2 | N/A | 28/01/2014 | Fixed problems with document and added test log forms | bmo, njv1 |

## Revised design specification

# References

# Document history