Final Report: GROUP 12

Authors: bea16, dpb1, alh32, fik, anp28, srp11, lps1, lot15, thw10
Config Ref: SE_12_FR_01
Date: 17/02/2014
Version: 1.2
Status: Final Release

Department of Computer Science
Aberystwyth University
Aberystwyth
Ceredigion
SY23 3DB

Table of Contents

1. Introduction

1.1 Purpose of this document

This document is here to demonstrate exactly what the project has accomplished and how well it accomplished these things. It will show how the project progressed from the start and what parts of the project work as intended. As well as highlighting any issues which the group came across throughout the project.

1.2 Scope

This document includes a management summary of how the project went and what works and what doesn't, a historical account of what has happened since the group first met and what steps we've taken along the way, including all major events. Along with what the final state of the project is and the parts of the project which are correct and which parts are not quite right. This document will also include the performance of each team member, including duties undertaken by group members and how the team worked together, and what could have been improved.

This document should be read with consideration to the project test report and should be read by the client.

1.3 Objectives

The objectives this document has are:

* To show the achievements that the project has accomplished

* To display how well the project has progressed from the start

* To demonstrate how well the group worked together for the duration of the project

* To show what role each member was tasked with and how their performance was

* To provide the client with an evaluation of the final project

2. End-of-Project Report

2.1 A management summary

The project managed to achieve the major elements of the requirement specification and ended up with a working Android application, combined with a functional database, server and website. The project managed to record the users co-ordinates at a suitable rate and send the data to the server which is then saved in the database and displayed on an interactive map on the website. The location name and description are shown when a location is clicked and each location is joined by a line to show the route. The map did however, fail to point out which end the walk was started at and which end it finished.

Each walk that's uploaded can be viewed by selecting it from a drop down menu system just above the map on the website. The pictures taken/chosen are uploaded to the server and displayed underneath the map. However they do not link to a location, so there is nothing to say which picture is for which walk/location.

When a walk is started on the phone, the user is taken to a screen where they can enter the walk name, short description and long description before starting the walk. This is the name which will be listed in the drop down menu box on the website once it's been uploaded. It is however, not sanitised, so spaces and other characters are allowed to be entered.

All of the documents created throughout the project are in a good state, have been checked along quality assurance guidelines. As the project has progressed, documents like the project plan and design specification have been updated and improved as necessary and are now in their final state. The test specification has been followed up by a test report, so this has stayed fairly unchanged since its final completion.

All minutes recorded in meetings since the start of the project are in their final state and have been quality assurance tested. They have a specified filename for easy sorting, and include a minute list stating the date of the minutes and their purpose.

The team worked well together to complete much of the project, and everyone contributed to sections of the documents as required. Despite a few minor hiccups along the way, all documents were completed on time. The project prototype was also completed on time, although due to the absence of a team member in the weeks before the prototype was due to be finished, reallocation of team members to certain parts was necessary as a temporary solution. This was remedied soon after the prototype was finished. The final application progressed well although some issues took longer than expected to resolve.

In the end the project had the major functionality that was required and was overall successful. The Android application, website, database and server all work together and communication between them is successful. As well as this, the documents have been finalised, checked and released.

2.2 A historical account of the project

At the start of the project we began by assigning roles to each member of the group, based on the skills each member had, such as programming, testing and web development. This helped to ensure that all members were assigned appropriate tasks which they were capable of completing to a suitable standard. Some larger pieces of work were shared between roles and necessary steps were taken to guarantee the completion on time.

Two group members (Quality assurance manager and group leader) attended a two hour session to learn how to use GitHub and setup a Git repository, as well as how to push, commit and pull from this repository. This information was then relayed to the rest of the group to ensure that everyone knew how to operate Git so that it could be used by everyone, making it a communal resource for the group to upload documents/code.

Two of the lead programmers in the group attended a several hour session to learn how to operate the Android development software. This session was very important to make sure that they fully understood how to use the software and create an application to run on the Android operating system.

The next task our group had was to create a project plan which outlined each of the areas we needed to cover, and gave a timeline for things to be completed in the form of a gantt chart. The document was broken down into the sections which needed to be included, and each group member contributed to one of these sections to get the document completed on time.

Once the project plan was completed, we started work on the first sections decided in the plan. The next document we worked on was the test specification, which outlined the necessary test which would have to be conducted to produce a quality piece of software.

We then started work on the design specification, again assigning parts of the document to a group member. This gave us an idea of how parts of the project should look and what we should be aiming for. This document included screen views of relevant pages that would need to be included in the project. Again each group member took a section of the document to complete.

Once each document was completed, time was spent by the quality analysis manager to check that that the document conformed to quality analysis standards and if necessary, documents were edited to fit standards. This was carried out by checking documents against the quality assurance standards. This also helped guarantee that all documents were in the correct final format for submission and that all required sections were included.

To keep our project on track and make sure that everyone knew what they were supposed to be doing, we had a weekly meeting with our project manager every Thursday. We also had a group meeting every Tuesday, for which we pre-booked a room for every week up to Christmas to ensure that we would always have somewhere to meet, discuss and work on the project. It also gave the group a chance to air, discuss and resolve any worries that they had about how the project was progressing or a specific task relating to the project. These meetings were also very important to set/review work for the week and ensure that each group member were doing what they should and that time was being distributed suitably.

The project prototype was created to show what the final program would look like and how parts of the design would interact, without including all of the functionality necessary for the final version. This prototype was then presented and reviewed by our project manager in one of the Thursday meetings, to make sure that our group was on track to build a quality piece of software and to highlight any areas which needed revising.

A final report was one of the last documents which needed to be completed. This document had to show what has been accomplished throughout the project and had to include a management summary, a historical account of the project and a summary of the final state of the project.

The project test report was another document which had to be constructed at the end of the project. This document required a list of the name and number of each test planned for the project, including any failed tests, along with an explanation of why the test failed.

The last document required for the project was a project maintenance manual. The purpose of this document was to answer all of the likely questions which a program maintainer may ask. At the very least this document should show the maintainer which part of the program source is likely to provide the answer to their question.

Integration and testing week was the last week in which the project could be completed. At the end of this week, the final version of software that our group had built was to be submitted. This week was when all the different parts of the project had to come together and work as one. This is also the week that any unchecked documents were QA checked and necessary documents were completed and finalised. Once this all documents were completed and submitted, the project was complete.

2.3 Final state of the project

The final state of the project was successful with the application meeting all 9 functional requirements fully with no erroneous features.

Passed Tests

FR1 Startup of software on Android device – The application was observed to install and execute correctly.

FR2 Providing info about the whole walking tour – The application was observed to provide correct data on both Android and web side.

FR3 Adding locations to the walking tour – A location can be successfully added to a walking tour.
FR4 Adding photos to the walks – Photos can be added to a walk from both camera and photos stored locally.
FR5 Cancelling walks - A walk can be cancelled.
FR6 Sending the walk to the server – The walk can be sent successfully to the server via a Base64 string.
FR7 Switching from the WTC – Multitasking successful.
FR8 Trying out a created walking tour – Tours display correctly however thumbnails are not visible on server.
FR9 Saving data on server – All data is stored correctly on server.

Failed Tests
A total of 1.89% (3 of 63) tests failed without an improvement being made. The tests that failed were minor in regards of functional requirements as can be seen from the table below.

Test ID	Requirement Tested	Pass Criteria	Outcome.
SE-F-055	FR8	Locations set by user should have thumbnails.	Thumbnails do not have pictures.
SE-F-057	FR9	Android server upload upload confirmation.	No message returned.
SE-F-058	FR9	Confirming data is correct when sending data from client to server.	No message returned.

Figure 1: Table showing the failed tests from the test report

Known Problems with Documentation

The documentation is perceived to be correct following all QA documentation however it is noted that the documentation does not reference the corresponding QA documentation as much is as needed. Is it also noted that the CCF No was not used throughout the project with team members misunderstanding its intended use throughout the project it was however correctly used during integration and testing week.

2.4 Performance of each team member

Filip Kunkiewicz, [fik]

My role within the project was being the team's leader. At the very begging of the project, I familiarised myself with every document on BlackBoard in order to have a good idea on the projects requirements. Throughout the project, I arranged meetings, monitored progress, gave out tasks, produced documentation as well as tested the system. Whenever an issue was arising, I made sure that this issue was taken care of and the project was not delayed. This allowed for a functional product to be delivered. I made sure that each time a document was to be delivered, all sections of it were completed by the team members which were most suitable to submit it. I kept an eye on work hours, this is to make sure that the work which needed to be produced was spread out fairly. Meetings were arranged, documents were reviewed with changes being made where required. I made sure that the right people were present at the reviews and that the correct changes were made. In case of absences, I made sure that the people who weren't there, were updated on the progress and knew the tasks which they were set to complete. When team mates had trouble with completing any of the work they were set to, I offered my help and made sure that the work was done as required. When the group wasn't sure about any of the work, I made sure to contact the client and/or the tutor in order to make sure that what had to be done was understood by the team.

Louis Taylor, [lot15]

Louis has been an active part of the group with his duties being in the area of Android application development. He has been given various tasks which were performed to a high standard. He has been working alongside Andy Poll, Declan Beynon and Sion Purnell during the integration and testing week in order to produce a working Android application. Louis has also worked alongside of Tom Wise in order to allow correct data transfer between the Android application and the database. Furthermore Louis had input when producing some of the documentation and prototype files for the project. Amongst other things, this included mock up screens of the Android application and Json schemas. Louis has attended most meetings and was present every day during integration and testing week, not complaining when having to stay for longer than required in order to work on completion of the project. Louis was not afraid to help the team to resolve any issues related to programming as well as GitHub. He always did so even if it meant delaying the completion of his own work. Although absent for a period of time at the end of the first semester, Louis more made up for this with his consistent and high quality efforts during integration and testing week.

Sion Purnell, [srp11]

Sion has been a very active member of the group right from the beginning of the project. Throughout, he has turned up to a vast majority of the meetings, only missing one during the first

semester. Sion's original role was to work on anything which needed but he soon acquired the role of an Android programmer due to loss of contact with another group member. This is exactly what he did during integration and testing week. Sion has worked on the Android application alongside of Andy Poll, Declan Beynon and Louis Taylor, communicating and getting his ideas across in a professional manner. Sion worked on many aspects of the Android application itself. Amongst many others this included the ability of taking pictures and editing the locations within a tour. Not only working on the Android application, Sion also produced some of the documentation which was done during the first semester. This included parts of the Test Specification document as well as the Project Plan and the Design Specification. Sion also helped with the review of documents just before submissions therefore eliminating errors and allowing the group to get a better grade. Throughout the project, Sion was reliable team mate, consistently involved in its progress and development.

Andrew Poll, [anp28]

Andy's original role in the group was to be a tester. This mean that he worked on the test specification document alongside of Sion Purnell. Due to an absence and loss of contact with one of the team members at the end of the first semester, the team was re-organised and Andy was assigned the Android application developer role. Alongside of Sion Purnell, Declan Beynon and Louis Taylor, Andy worked on developing the Android application during integration and testing week. While doing the work he was assigned, he also managed to find time to help his team mates, especially Declan when he needed it. With a consistent presence during meetings in the first term as well as integration and testing week, Andy's attendance was very good. He was never late while at times finding himself staying late as well as coming in early in order to ensure that the project was completed on time. When faced with difficulties with the Android programming, Andy thrived to complete the task he was assigned in order to ensure the group produced a functional end product. Andy was not afraid to take on any aspect of the project and had great attendance throughout which made him a reliable team mate.

Declan Beynon, [dpb1]

Declan's role from the beginning of the project was to be part of the team responsible of programming the Android application. This also meant that he was highly involved in production of the documents which were delivered over the first semester. These included the Project Plan, Design Specification and the prototype which was then demoed to the group's tutor. He has worked alongside of Sion Purnell, Andy Poll and Louis Taylor when developing the Android application. Declan always performed to his best ability even when facing difficulties along the way. While the development of the Android application wasn't going to plan, Declan looked for advice from his team mates as well as offering his own help when others needed it. During integration and testing week Declan turned up every day and helped the team to produce a functional end product. Declan had very good attendance during the first semester which was then carried onto the integration and testing week. When faced with a temporary loss of team mate, he expressed this issue which helped the team to re-organise and get back on track quickly. Declan was a consistent team mate, he always expressed his problems or made notice of any project related issues, whether this was to fellow team mates or the project leader. This allowed for tasks to be completed in accordance with the schedule and a functional product to be developed.

Thomas Wise, [thw10]

Tom has been highly involved in the development of the project. His role was the back end web developer. Tom's attendance in meetings over the first semester was very good and this was reflected by consistency when delivering assigned work. In the early stages of the project, when not working on programming itself, Tom was involved in producing some of the documentation which was delivered throughout the first term. This included aspects of the Project Plan as well as the Design Specification document. Tom also produced a web prototype, which along with other components, the group was able to demonstrate to the tutor. During integration and testing week, Tom worked alongside of Ben Aeyrite when working on the web side of the project as well as Louis Taylor when working on transferring the data between the Android application and the database itself. Tom communicated with team-mates well, expressing his ideas which allowed the team to proceed with tasks in order to produce a functional piece of software. During the last week of development, Tom's work escalated into the front-end programming as well as back-end programming. This was not ideal but Tom did not complain and altogether had great input when developing the final product. In the final stages of the project, Tom was not afraid to put extra hours into the project. By doing so, he helped the group abundantly to make sure that a functional product was delivered.

Benjamin Erefagha Ayerite, [bea16]

Ben's role in the project was to work on the front-end web design. This meant that he was to work with Tom Wise in order to ensure a functional web interface capable of receiving data as sent by the Android client. Over the first semester, Ben produced the work he was required. Ben's attendance in meetings was good. This meant that he could express any problems or make notice of any issues with the documentation or programming aspects of the project which could then require change. Amongst other features of the software, Ben worked on decoding the image data when sent by the Android client. This turned out to be a much bigger task than originally expected but due to Ben's spike work, the group was able to integrate this feature much quicker. Ben also worked on some of the documentation as delivered over the first semester. Over the first semester, Ben did not fail to deliver the work he was asked to produce. This included a prototype of the front end web interface as well as parts of the Project Plan and the Design Specification documents. Unfortunately, in the very final stages of the integration, Ben was not able to implement some features which could have made the front-end website look more eye-catching, therefore leaving it looking slightly unprofessional. Nevertheless, group 12 produced a fully functional product which met the client's requirements as demonstrated during acceptance testing.

Alexander Heaven, [alh32]

As decided by the group's first meeting, Alex was appointed the QA manager role. During that meeting, Alex pointed out his attention to detail. This was reflected at some points during the project. Alex attended the two hour GitHub session as required although struggled with its use at first. Soon enough, Alex got used to GitHub and was able to use it as required by the group. As a QA Manager, Alex did carry out some QA checking over the first semester although the frequency of this could have been improved. Alex's attendance during the meetings over the first semester was good although not great. He failed to make an appearance to some meetings without letting the rest of the group know of such absence. This meant that some QA checking was not carried out on time. While lacking effort in some areas over the first semester, Alex did work as required during integration and testing week. While attendance still wasn't great, he

carried out a vast amount of QA checking which benefited the group and the final state of the project. Furthermore, Alex made a contribution to the documents which were to be delivered. He made improvements to the documents which were done over the first semester as well as completing sections of other documents. He also made sure that the documents which were on GitHub meet all the QA requirements, something which perhaps should have been done as the documents were produced.

Luke Suggett, [lps1]

Luke's responsibilities within the project was producing documentations as well as testing to go with the software being developed. Throughout the task, Luke raised project related issues which then allowed the group to address these and make positive progress. Luke's attendance during the first term was very good. If absent, he let the group know beforehand which meant that the the project schedule could be maintained. Luke's good attendance over the first term was carried into integration and testing week. As documentation was his main area within the project, this is exactly what he worked on during integration and testing week. Luke worked on improving the documentation which was handed in over the first semester as well as producing parts of documents which were to be handed in after finalising the code. He also did some sub-system testing which allowed to identify aspects of the software which were not working or not yet implemented. As well as producing a vast amount of the documentation, Luke was not afraid to do research into some code. Once finished with a task, he offered his help to the team which allowed for the project to move on quicker. Furthermore, Luke did not complain when having to do work which was to be done by someone else, but due to absence or lateness it simply wasn't. Overall, during the project, Luke proved to be a reliable and consistent team mate.

2.5 Critical evaluation of the team and the project

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

From the begging of the project the team performed well. During the second meeting, roles have been assigned based on the abilities of each group member. Team members who did not feel comfortable doing a large amount of programming were assigned roles such as documentation as well as testing. People who felt more comfortable programming were assigned roles according to the area of programming they felt was most convenient. Meetings were arranged all throughout the first term. During these, the team discussed the current progress as well as reviewing any documents which were to be delivered. This allowed for mistakes to be corrected and the documents submitted to be of higher quality. Team communicated very well, with emails being sent from the group leader to the rest of the group. This included work to be completed as well as information about upcoming meetings. Performance of the group could have been improved by more frequent QA checks over the first semester. This would have increased the quality of the documents submitted which could then result in better grades.

2. How could the project have been improved?

The project could have been improved by more spike work being done before integration and testing week. This could have improved the quality of final software. Furthermore, it would have allowed for more features to be added. More QA checking over the first term could have perhaps provided better grades for the documents which were delivered. In order to do more QA

checking, better planning would have also been required. Ideally, this would provide enough time in between hand ins and reviews to change what was required, have it reviewed and ready for delivery.

3. What were most important lessons learned about software projects and working in teams?

One of the most important lessons learned in this project was that good communication between team members is absolutely key when building software. When communication is lost, tasks are not completed on time which therefore delays the completion of the whole project. This then has a negative effect on the quality of the final software. Another important lesson to take from this project is assigning roles to people based on their abilities. Some team members might not be comfortable doing certain aspects of the project but more confident doing others. These strengths and weaknesses need to be spotted early on in the project to ensure the best possible outcome. While working in a team, one has to ensure that the tasks which are to be completed are understood correctly by the person which is to complete them. This is especially key when building software which is composed of sub systems such as an Android application and a web application. Team members working on both of these need to be aware of each others progress in order to make sure that the progress being made is worth while. This leads to another important lesson learned which is that a lot of the work which needs to be done is best done in parallel with other parts. This was very key when building the Android application as programmers had to be aware of each others progress and the way in which that progress is being made. While working in a team it is very important to point out any issues which might be arising. This will allow the group to address these issues and proceed with the project with a positive effect. Lastly, understanding the requirements is a major part of a software project. By making sure exactly what the client wants, team can deliver software which matches the requirements precisely. Furthermore it allows for feature creep to be eliminated, therefore focusing efforts on aspects which have been asked for in the requirements specification.

3. Appendices

3.1 Project test report

3.1.1 Test table

Test ID	Requirement tested	Test context	Input	Output	Pass criteria
SE-F-001	FR1	Check that the application can be installed and is runnable	Install and start of the application.	Application starts and the home screen should show.	Correct presentation of the home screen.
SE-F-002	FR1	The	Press the quit	Closed	Application

		application can be closed by pressing the quit button.	button.	application.	closes correctly and is no longer in the running tasks of the phone.
SE-F-003	FR1 & FR2	Check to see if walk can be created	Clicking Start new walk	Asks for name of walk.	Screen changes to a new screen that prompts user for the name of the walk
SE-F-004	FR1	Checking if the application launches.	Clicking on the application logo.	Home screen of the application shows up.	Home screen of the application should show up.
SE-F-005	FR1	Recording of a walk can be started.	Once starting a walk the coordinates are being recorded every stated period of time.	Coordinates from the walk stored in the SQLite database on the phone.	Correct recording of the coordinates from the walk.
SE-F-006	FR1 & FR5	Recording can be canceled.	Pressing the cancel walk button.	Screen goes back to main.	Recording of a walk is stopped and the main screen is displayed.
SE-F-007	FR1	Tour can be uploaded to the server.	Pressing the upload/finish button.	Tour is uploaded to the server.	Tour should now be on the server.
SE-F-008	FR1	Checking whether the uploaded tour is displayed on the website itself.	Going onto the tour website.	Tour is showed on the “Tours” page within the website.	Tour should be displayed on the website.
SE-F-008	FR1	Location can be added during the recording of the tour.	Pressing the add location during the currently recording tour.	Location added and screen goes back to the main tour recording	Location should be added to the list of locations within the

				screen.	tour.
SE-F-009	FR2	Check to see if a name can be entered in for the walk.	Enter "Aberystwyth Pub Crawl"	User can move on to the short description of the walk.	Name accepted and prompts user for a short description.
SE-F-010	FR2	Check to see if the name of the walk has to be typed in before being able to proceed with the walk.	Try to start the tour with the name of the tour field being blank.	Message requiring for the user to fill in the field before proceeding pops up.	User is not able to proceed without a value typed in for the name of the walk.
SE-F-011	FR2	Check to see if user can enter a short description	Enter "This short length walk from Constitution Hill to Aberystwyth castle."	Screen changes to map/walk.	Description is accepted
SE-F-012	FR2	Check to see if the short description of the walk has to be typed in before being able to proceed with the walk.	Enter ""	Prompts the user asking to enter a valid description.	Description is rejected and prompted to retry.
SE-F-013	FR2	Check if the character limit of the short description is 100 characters	Try to enter a short description that's over 100 characters long.	User should not be able to input a short description which is that long.	User not able to enter any more than 100 characters for the short description.
SE-F-014	FR2	Check to see if user can enter a long description	Enter (Long description)	User is able to proceed to the next screen.	Description accepted.
SE-F-015	FR2	Check to see if user has entered a long description	Enter ""	Prompts the user asking to fill in the field.	Description is rejected and prompted to retry.
SE-F-016	FR2	Check if the	Try to enter a	User should	User not able

		user can't enter a description that is over 1000 characters in length.	description over 1000 characters long.	not be able to enter more than 1000 characters.	to enter any more than 1000 characters for the short description.
SE-F-017	FR2	Check to see if the user can edit a walk name.	Edit walk name	New walk name is typed in.	User allowed to edit the current walk name.
SE-F-018	FR2	Check to see if the user can edit the short walk description.	Edit walk's short description.	A new short description is typed in.	User allowed to edit the short description of a walk.
SE-F-019	FR2	Check to see if the user can edit the long walk description.	Edit walk's long description.	A new long description is typed in.	User allowed to edit the long description of a walk.
SE-F-020	FR3	Checking whether the users location can be identified correctly.	Going onto the Android application and pressing add location.	Users GPS location.	The location is checked against a correct location on Google maps and a correct match is produced.
SE-F-021	FR3	Checking whether an error message is shown when GPS signal is lost.	GPS turned off.	A message saying GPS signal has been lost.	When GPS is turned off a message saying letting the user know of the loss is displayed.
SE-F-022	FR2 & FR3	Check to see if user can add a location	Clicking on add a location.	Location is saved and user is able to continue on with the walk.	Current coordinates are saved.
SE-F-023	FR3 &FR4	Checking whether the latitude and longitude for a location have	Clicking on add location button.	Latitude and longitude are now saved and can be used to plot the line of	Latitude and longitude of the location are saved correctly and

		been saved.		the walking tour.	they are they are the correct lat and long values. This is checked against Google maps on a computer.
SE-F-024	FR3 &FR4	Check to see if user can enter a location name.	Enter “Sea front”	Screen changes to accept a description.	Name is accepted.
SE-F-025	FR3 & FR 4	Check to see if user has to enter a location name.	Enter “”	Prompts the user asking to enter a valid name.	User must re-enter a name.
SE-F-026	FR3 & FR4	Check to see if user can enter a location description.	Enter “A great place to be at sunset or on a hot day.”	Returns user back to the current walk's screen and the user can add more locations.	Description is accepted and user can continue on with the walk and add more locations.
SE-F-027	FR3	Check to see if user has to enter a location description	Enter “”	Message prompting the user to enter a valid location description is displayed.	Description is rejected and asks for another.
SE-F-028	FR3 & FR4	Check to see if user can't enter a description that's over 1000 characters in length.	Trying to type in a description which is more than 1000 characters long.	Prompts the user asking to enter a valid description.	Description is rejected and prompted to retry.
SE-F-029	FR3 & FR4	Check to see if time stamp is taken.	Save a location	Returns user back to map/walks	A correct location and time stamp are saved locally.
SE-F-030	FR4	Checking whether the camera application is launched correctly after	Pressing add photo after pressing add location.	Camera application is launched.	Camera application is launched and after taking the picture, user is

		adding a location.			directed back to the location screen.
SE-F-0031	FR4	Check to see if the photo taken is saved.	Take photo.	User can go back to the add location screen.	Photo is accepted and saved locally.
SE-F-032	FR4	Checking to see if the user can take a different photo after one has been taken.	After taking a photo, pressing dismiss and taking another photo.	User can go back to the add location screen.	The new photo is accepted and the add location screen is displayed.
SE-F-033	FR4	Check to see if coordinates are taken when image is taken.	Take a photo/ get a photo from library.	Returns user back to map/walks.	Image is saved internally with the coordinates of the location.
SE-F-034	FR5	Check to see if user can cancel a walk	Start a walk, store some information, and then cancel.	Returned to main menu of the application.	All locally stored data on the current walk is deleted.
SE-F-035	FR5	Check to see if a message letting the user know that once canceled all walk data will be lost shows up.	Press cancel walk.	Returned to the main menu and message produced.	Once the user pressed cancel, a message letting the user know that data will be lost should show up.
SE-F-036	FR5	After canceling a walk, checking whether user can then start a new one.	Pressing cancel walk and then starting a new walk from the main screen.	Data about the old walk is dismissed and new data is stored locally.	After canceling and starting a new walk, the current data is dismissed and the new data about the walk is stored locally.
SE-F-037	FR6	Checking whether the name is sent correctly to	Finish walk and press upload.	Message saying the data is uploaded or will be	Name of the walk is sent to the sever correctly.

		the server.		uploaded soon in case of wi-fi unavailability is displayed.	
SE-F-038	FR6	Checking whether the short description is sent correctly to the server.	Finish walk and press upload.	Message saying the data is uploaded or will be uploaded soon in case of wi-fi unavailability is displayed.	Short description the walk is sent to the sever correctly.
SE-F-039	FR6	Checking whether the long description is sent correctly to the server.	Finish walk and press upload.	Message saying the data is uploaded or will be uploaded soon in case of wi-fi unavailability is displayed.	Long description of the walk is sent to the sever correctly.
SE-F-040	FR6	Checking whether the GPS coordinates are sent correctly to the server.	Finish walk and press upload.	Message saying the data is uploaded or will be uploaded soon in case of wi-fi unavailability is displayed.	GPS coordinates of the walk are sent to the sever correctly.
SE-F-041	FR6	Checking whether the time stamps for each location are sent to the server correctly.	Finish walk and press upload.	Message saying the data is uploaded or will be uploaded soon in case of wi-fi unavailability is displayed.	Time stamps of the locations are sent to the sever correctly.
SE-F-042	FR6	Checking whether the locations in a tour are sent correctly to the server.	Finish walk and press upload.	Message saying the data is uploaded or will be uploaded soon in case of wi-fi unavailability is displayed.	Locations within the walk are sent to the sever correctly.
SE-F-043	FR6	Checking whether the pictures in a	Finish walk and press upload.	Message saying the data is uploaded or	Pictures are converted to base64 and are

		tour are sent correctly to the server.		will be uploaded soon in case of wi-fi unavailability is displayed.	sent to the sever correctly.
SE-F-044	FR6 & FR8	Checking whether the name, title, long description and short description can now be displayed on the website itself.	Going onto the website and clicking on the tours page.	Specified data which was sent to the server from the android application.	Name, title, long description and short description which have been sent to the server should now show up on the website correctly.
SE-F-045	FR6 & FR8	Checking whether the list of GPS coordinates and the time stamps can now be displayed on the website itself.	Going onto the website and clicking on the tours page.	Specified data which was sent to the server from the android application.	GPS coordinates and time stamps which been sent to the server should now show up on the website correctly.
SE-F-046	FR6 & FR8	Checking whether the list of locations can now be displayed on the website itself.	Going onto the website and clicking on a specific tour.	Locations which have been sent from the android and to the server.	Locations sent from the android application should now show up on the map within the web application.
SE-F-047	FR6 & FR8	Checking whether the pictures sent are displayed on the website correctly.	Going onto the website and clicking on a specific tour.	Pictures which have been sent from the android and to the server.	Pictures sent from the android application should now show up on the map within the web application.
SE-F-048	FR7	Closing mid way through a	User pressing the home	Data which the user has	The data which is

		walk.	button.	already inputted is stored locally.	stored locally and once the application is reopened, the data is still present.
SE-F-049	FR7	Reopening the app halfway through a walk.	Open the WTC app.	Returns the user back to the screen they left it on.	Previous screen loads and data on the walk is retrieved from local storage and no data is lost. User can continue on with the walk.
SE-F-050	FR8	Checking whether when the user goes onto the WTD it shows the list of uploaded tours.	Going onto the Tours page within the website.	All of the tours which have been uploaded.	The tours which have been uploaded by the user show up correctly on the web client.
SE-F-051	FR8	Checking whether when the user clicks on a certain tour, all the details on it display correctly as sent by the Android client.	Clicking on a set tour.	Tour details.	All of the details about the tours which have been originally uploaded by the client are displayed correctly on the server.
SE-F-052	FR8	Checking whether the way points from the tour plot a correct line on the Google map embedded in the website.	Clicking on a specific tour.	Line drawn on the Google map embedded in the website.	Correct displaying of the tour way points and locations. Correct order required for this test to pass.
SE-F-053	FR8	Checking whether the locations	Clicking on tour details at the Google	Locations on the Google map as	Correct locations display on the

		added by the user are displayed correctly on the Google map embedded in the website.	map of each tour.	selected by the user.	Google map.
SE-F-054	FR8	Checking whether the pictures which the user has taken at each way point are displayed correctly and are assigned to a correct way point.	Clicking on details about each tour.	Pictures assigned to locations.	Correct pictures are assigned to locations.
SE-F-055	FR8	Checking whether the locations set by the user have thumbnails which when hovered on have the picture taken at each location.	Hovering onto a location.	Picture of a location.	When hovering onto a location a correct location picture is displayed.
SE-F-056	FR9	Checking whether the server database has a correct format.	Look at the table fields in the database on the server.	Correct number and format of fields.	The fields match the fields specified in SE.QA.RS.
SE-F-057	FR9	Checking whether the server returns a success or error message back to the client after a successful or failed upload.	Input of correct or incorrect data to the server.	A message saying the upload has been correct or incorrect.	The android client sending a correct message as to whether the upload of the data has been successful or not.
SE-F-058	FR9	Checking whether the	The Android user sending	Message of whether the	Correct message sent

		format of the message is correct.	data from the Android client and to the web server.	data has been uploaded received by the Android client.	from the server and received by the Android client.
SE-F-059	FR9 & FR8	Checking whether the uploaded walk shows up in the walks on the website.	Sending of the data from the Android client to the server as well as the retrieval of the data from the server and into the website.	Json data sent by the android client.	Correct data sent by the Android client and received by the
SE-F-060	EIR1	Checking whether the Android application is intuitive to other users.	Clicking on different buttons within the applications.	User is taken to the correct screens in the application.	The correct screens within the app
SE-F-061	PR1	Checking the responsiveness of the Android interface.	Accessing different screens within the application.	User is taken around the application.	The app is responsive and the screen change within one second of choosing a certain option.
SE-F-062	PR2	Checking whether the Android application runs on a phone with Android 2.2 and above.	Installing the application on a phone.	Application is installed on a phone.	The application is installed correctly on the Android 2.2 and up to Android 4.2
SE-F-063	PR2	Checking whether the web application runs on the latest versions of Google Chrome, Mozilla Firefox and IE.	Accessing the website on latest versions of all 3 browsers.	Website displays on all browsers.	Website displays correctly on all 3 browsers.
SE-F-064	DC3	Checking	Looking at the	Database	The structure

		whether the database has a structure as defined in the requirement specification document SE.QA.RS	specified structure and the structure of the database.	structure.	of the database matches the structure as specified in the SE.QA.RS document.

Figure 2: Table showing the test table.

3.1.2 Test Log

Test ID	Pass/Fail	Fail Description	CCF/issue#
SE-F-001	PASS		
SE-F-002	FAIL	App crashes when pressing the quit button.	#13
SE-F-003	PASS		
SE-F-004	PASS		
SE-F-005	PASS		
SE-F-006	PASS		
SE-F-007	PASS		
SE-F-008	FAIL	Tour is uploaded to the server correctly and processed by the database. The website is not showing the tour yet.	#14
SE-F-009	PASS		
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	PASS		
SE-F-019	PASS		
SE-F-020	PASS		
SE-F-021	PASS		
SE-F-022	PASS		
SE-F-023	PASS		

SE-F-024	PASS		
SE-F-025	PASS		
SE-F-026	PASS		
SE-F-027	PASS		
SE-F-028	PASS		
SE-F-029	FAIL	Current time stamp collected is not in a correct format.	#15
SE-F-030	PASS		
SE-F-031	PASS		
SE-F-032	PASS		
SE-F-033	PASS		
SE-F-034	PASS		
SE-F-035	PASS		
SE-F-036	PASS		
SE-F-037	PASS		
SE-F-038	PASS		
SE-F-039	PASS		
SE-F-040	PASS		
SE-F-041	PASS		
SE-F-042	PASS		
SE-F-043	PASS		
SE-F-044	FAIL	Pictures not being sent to the server.	#16
SE-F-045	PASS		
SE-F-046	PASS		
SE-F-047	PASS		
SE-F-048	PASS		
SE-F-049	PASS		
SE-F-050	PASS		
SE-F-051	PASS		
SE-F-052	PASS		
SE-F-053	PASS		
SE-F-054	PASS		
SE-F-055	FAIL	Thumbnails do not have pictures.	N/A
SE-F-056	PASS		
SE-F-057	FAIL	No message returned	N/A
SE-F-058	FAIL	No message returned	N/A

SE-F-059	PASS		
SE-F-060	PASS		
SE-F-061	PASS		
SE-F-062	PASS		
SE-F-063	PASS		
SE-F-064	PASS		

Figure 3: Table showing the tests log.

3.1.2 Failed Tests

Test ID	Reason for Failure	Solved	How Solved
SE-F-002	App crashes when pressing the quit button.	YES	Research was done on whether an Android app requires a quit button. The quit button was removed.
SE-F-008	Tour is uploaded to the server correctly and processed by the database. The website is not showing the tour yet.	YES	The feature was added and the website application is able to show the tours as uploaded by the Android client.
SE-F-029	Current time stamp collected is not in a correct format.	YES	More research was done on the format of a time stamp which meant that it could be changed allowing for the issue to be resolved.
SE-F-044	Pictures not being sent to the server.	YES	This issue was resolved by adding the functionality to the Android application code as well as the web application in order to process the pictures sent.
SE-F-055	Thumbnails do not have pictures.	NO	This issue was not resolved due to the lack of time.
SE-F-057	No message returned	NO	The group did not include this feature in the project. The data transfer between

			Android and web functional anyway.
SE-F-058	No message returned	NO	The group did not include this feature in the project. The data transfer between Android and web functional anyway.

Figure 4: Table describing the failed tests

3.2 Project maintenance manual

3.2.1 Program description

As described in the design specification this will consist of three main parts:

1. Android application (WTC)

   The Walk program provides a user interface on a android platform to enable a user to make a walk, way-points, amend walk, way-point, take pictures of way-points and put in information about the way-points.

2. The server (WTD)

   The Server provides a storage space for the database and all the information about walks so that they can view them. The Server will also use a Test Data file. The server saves the information it receives inside of a database.

3. The website (WTD)

   The Website provides a user interface for a person to view a walk online. It also allows a user to search for a specific walk. The Website will query information from the Server and display it on to the Google maps interface so the user can take the tour. The locations within the tour are be shown as pins. The line showing the tour is drawn based on the way poins recorded during the walk.

3.2.2 Program structure

For a list of methods, as well as modules please refer to the Interface Description page of the Design Specification on page 63.

WTC

This section will describe the design of the program using pseudo code.

```
HomeActivity {
      onStartTourCreator() {
            tour = new tour();
            start(tour);
       }
      public void onStartUpload(View view) {
```

```
            temp = stored tour;
            post = new post(post address);
            post asynchronously;
      }
      public void onStartQuit(View view) {
            quit the application;
      }
}

TourActivity{
      onCreate(){
            tour = WalkingTourApplication tour
            locationManager = location service;
            creiteria = new criteria;
            provider = locationManager getProvider;
            Location location = get last location(provider);
            if the location = null
            {
                  onLocationChanged(location);
            }
}
      createOptionsMenu(){
            create an option menu;
      }
      startLocationCreator(){
            create a locationCreator();
      }
      deleteTour(){
            close window;
      }
      edit location(){
            create a location editor;
      }
      upload(){
            url;
            if debug is true
            {
                  url = "debug url";
            } else {
                  url = "literal url";
```

```
            }

            post = new post;
            post asynchronously;
      }
      onResume(){
            resume;
      }
      onPause(){
            pause;
      }
}

LocationCreatorActivity
{
      protected void onCreate
      {
            tour = GetCurrentTour();
            BestProvider = Find_Best_Provider; (This returns either Wifi or
Data)
            location = Get LocationGPSInfo(BestProvider);

      if location != null
      {
      onLocationChanged(location);
      }

      }

      public void onLocationChanged(location)
      {
            setLongitude;
            setLatitude;

      }

TourCreatorActivity
{

      public void onStartTour
      {

      setFields =
      [
            SetName;
            setShortDescription;
            setLongDescription;
      ]

      if Name, or the descriptions are left blank
      {
```

```
            DisplayMessage("Please fill in all fields");
            return;

      }

      tour = new tour(setfields);
      Start TourActivity;
      }
}
```

WTD

The index page layout is simple, despite having PHP mixed with javascript and MySQL queries. The tasks performed by the page are organised in a chronological order and performed as such. The first task of the page is to connect to the storage area, under the heading of "Creating Database Connection…". And subsequently, performing the task of selecting the correct database to use containing all necessary tables and data layout.

The page then initialises the map, with all necessary styling and positioning involved, and begins setting up the walk data to be used. This involves creating a drop-down menu to select all existing walks in the database so that upon request of the data, it can print out all associated information with the selected walk upon submission of a request.

After the first request is made, the map will now display all associated data with a walk, by name. The selected name is resubmitted to the page which updates the map by fetching all locations associated with a walk, first by whether they have a timestamp or not. If there is a timestamp, this represents a location, locations have names and descriptions and a marker, but are not associated with the waypoint system for mapping a route between locations. Points with a timestamp of 0 are waypoints and are used in the mapping of routes as the app records lat/long continuously during a walk. All information is associated with ID numbers belonging to a certain walk.

Finally, after all map information is printed, the photos within the database are printed out to the page for the client to view the places that someone has been to.

3.2.3 Algorithms

Please refer to the design specification, page 76.

3.2.4 Main data areas

WTC

Please refer to the design specification, page 77.

WTD

The flow of data in the application is all generated and fed from the mobile device. It provides user data as well as automatic acquisition of co-ordinates as and when requested. This is all compressed into a format easily sent via http in the JSON format to the upload landing page. This page parses out the JSON post, and separates it into the necessary areas of the database that each item belongs to, by table and field. The database is the head storage location and contains all long term storage solutions to ensure the continual use of the application. Lastly, the only other data area is the main page which only retrieves data for use by the user. It does not update, modify or create data in any way.

3.2.5 Files

WTD

The web application contains only two files, excluding database structure which may be considered a third. The database structure as an .sql extension file. It contains the necessary organisation of the sql database to contain and organise all data for the application. The upload page processes and saves data sent to it in the JSON format, and can also be used to create a text file containing input data sent to it if desired, which is not contained with the final build, only as a debug tool. The index page is a PHP, Javascript and HTML styling heavy file which displays stored data to the user.

3.2.6 Interfaces

WTC

For the Home screen the user must be asked to create a walk and when clicked must be prompted for a Title of the tour, a short description which cannot exceed 100 characters and a long description which cannot exceed 1000 characters. When a tour is created the app will record the user's coordinates every 20 seconds and store locally on a device.

The user will also be able to record a location during a tour, when this happens the coordinates of the user's current position are taken and the user is prompted again. A name, description and time stamp must then be received.

The user will be able to add photos that are .jpg format to a walk from the phones photo library at which point the current coordinates and a time stamp are taken and again the user is prompted for a name of the location, a description.

The tour should be able to be cancelled. When a walk is cancelled all data currently stored about that particular tour are removed from the device

The tour should also be editable by the user, such as locations, names, etc. and overwrite the current information stored when saved.

When a tour is saved it is uploaded to a database. The information is formatted as a MIME and sent via a HTTP post. Information in the post are the Title, short description and long description, a list of the coordinates saved during that tour and the time stamps as well as a list of locations and the images which are converted into base64 strings

When the application is minimised on the device the current information about a tour is saved in the devices cache. When the app is re-opened the user will be returned back to the last page with the information saved.

For the web side the user is presented with a drop down box of all the tours the server has received, when a tour is selected from the box a map beneath is updated with the tours way points and with the locations marked showing the name and a description. Below this is a list of photos uploaded to the site.

3.2.7 Suggestions for improvements

As in most cases programs are a compromise between what one liked to do and what one has enough time to do, this was exactly the case in our group project. Most features were implemented although there was not enough time to include some functionality.

One of the most desirable improvements which were not included due to lack of time was the look of the website. Due to a last minute delivery, the website was left looking slightly unprofessional with only a map and a simple drop down list being present. We wished to have a more detailed and professional website with each tour having a dedicated web page with its details and pictures being displayed on there. This could have been solved by completing the functionality of the whole project earlier, therefore leaving more time to improve the aesthetics of the software.

Furthermore, we would like to have added the ability to add more than one picture to each location. This means that more work would have to be done on Android as well as the website. This could have been solved by doing more spike work before integration and testing week.

Features which were not included due to lack of time included thumbnails which do not have pictures next to them. Ideally we wanted each location on the map to show a picture/pictures from that location on the location thumbnail itself. Furthermore, a message regarding whether the tour has been uploaded is not returned from the server. Therefore the user doesn't know if the tour has been uploaded before actually checking on the website itself. Once again this could have been implemented by doing more spike work before coding week.

3.2.8 Things to watch out for when making changes

WTC

The main thing to watch out for when making changes to the program is that when you want to add additional features. To do this you will need to either create separate class' or if it's similar to certain features the programme currently has then create a new method within that class and refactor the code accordingly. To make sure that the new changes have not broken the application every change will need to have the full testing on the application to prove that it still works.

If the programmer that is editing the code wants to use the tour object then you will need to make sure that you use the appropriate code to pull the tour object into that class/activity .

WTD

If wanting to install the database locally, one has to watch out for the file permissions as well as the server permissions. These are based on the server configuration and settings. One has to change files to act accordingly.

If wished to change any database headings, programmer has to ensure that this is done all throughout the system.

The map, this should be put in the right div class and the right height and width should be set in order for the map to display in the web application.

Changing CSS will also disrupt the appearance and location of the map and image gallery.

3.2.9 Physical limitations

WTC

One of the limitations is the amount of locations a user can add. This depends on how much RAM the phone has so that limits how much information can be stored before the memory gets full. So that the users should know how many locations they can add to the tour.

Another limitation is the size of storage the phone has. With this application the user will need sufficient data storage space, this is so that the information of the tour will can be saved on the device if it is unable to send to the server. There also will need to be sufficient storage space for all of the photo's taking into account that these files can be very large.

Lastly, a limitation is that of a user finishing their tour and then to uploading it. Depending on how many locations the user has used and the hardware of the phone depends on how long it will take the phone to process the information stored. Then depends on how long it will take to send the information through the user's mobile internet.

Another limitation is that for the application to constantly record your coordinates the phone has to be kept on and not on standby mode. This is a bad limitation because to constantly have you phone on throughout the whole tour this would drain lots of power from the phone battery.

WTD

You can only store the amount of data which the server allows you to. Things such as the computer's graphics power are also something to consider.

User will need the latest version of Firefox, Chrome or Internet Explorer in order to be able to view the website correctly.

3.2.10 Rebuilding and testing

WTC

Importing

Rebuilding the Android client is relatively easy. The code placed in SRC/Controlled/WalkingTour should be imported into an IDE with the Android toolchain set up for compilation. The supported IDE is Intellij with the Android plugins, which is distributed under the name 'Android Studio' from developer.android.com.

Building

When it is imported correctly, the project can then be built and transferred onto a device. To install on a handheld device, enable USB debugging on the device, then change the run configuration to use this device. This can be done by performing the following steps:

1. Click on the Create Run Configuration tab, then from that tab press Target Device
2. Click Show chooser dialog
3. Click USB Devices
4. Select your USB device

Now press SHIFT + F10 (Win) or CTRL + R (Mac) to build the project and run on your device.

Important files

There are a few important files used for the maintenance of this system. These include:

SRC/Controlled/WalkingTour/AndroidManifest.xml

This contains all of the metadata the Android SDK needs to know to put together the main activities and build the project with the correct permissions.

Data models and non-android specific Java classes are stored in:

SRC/Controlled/WalkingTour/src/uk/ac/aber/group12/walkingtour/data

These include JSON encoding and decoding, Tour class and TourLocation class.

The layouts of the activities are stored in:

SRC/Controlled/WalkingTour/res/layout

For changing the look and feel of the application, these are the files which should be edited.

3.3 Personal reflective reports

Filip Kunkiewicz, [fik]

From the begging of the project, my role was being the group's leader. I believe that the project as a whole went well, with some obstacles along the way which were addressed and allowed for a functional product to be delivered. As with any team based project, this was to be expected.

Due to my role as the group's leader, I made sure that any issues arising were addressed as quickly as possible and the project could continue in accordance to the schedule. This was the case at the end of the first semester, where contact with one of the team members was lost. Due to little time being available before the Design Specification hand in as well as the prototype demo, team had to be reorganised in order to meet both of these deadlines. More people were assigned to work on the Android demo with others being reassigned to different roles such as testing. This way, the team was able to meet both deadlines on time.

Throughout the first semester, the team had two meetings a week. One of these was with the tutor, Nigel Hardy. This was on Thursday every week. The other was organised by myself on Tuesday mornings and took place in room 319 of the Physics building. I was able to book this room for one hour a week for the whole of the first semester. This proved to be very beneficial as it allowed the team to discuss any problems or review documents before hand in to the tutor on Thursday. During these meetings, minutes were recorded and QA checked by the QA manager. Minutes were recorded by various team members throughout the first semester with no particular person being selected to record minutes in every meeting.

Prior to deliveries of documentation, work was divided by myself and and email was sent to the team including a description or a document with attached instructions to the team. Work was divided by myself, based on each team members abilities and duties within the project. In meetings, the progress has been discussed and any problems regarding the completion of assigned work were discussed and resolved as quickly as possible. At times, emergency meetings were arranged by myself. These were as a result of either team members struggling to complete work or deadlines approaching. These then allowed to review documents and get them ready for submission.

Throughout the project, the group communicated well. Although the team agreed to use Facebook as the primary communication medium. This was eventually changed to email as due to some group members not being a member of the social networking site, I found it difficult to communicate with them. By sending work out and replying via email, I was able to ensure that all issues were addressed and all team members were able to view the content on the agenda. Good communication during the first semester was then carried into integration and testing week. By dividing the team into three sub-teams: Android programming, web programming and documentation/testing team, I was able to ensure that all work is completed. This also allowed me to make sure than an even amount of work hours is produced by every team member during this week. Along with two or sometimes three team members, during integration and testing week, meetings with the tutor were attended and progress was reviewed. We discussed what has been done up until that point in the day and what were the goals until the meeting next day. The team was able to stick to these goals, although at times it required myself and others to work outside of the designated work hours.

In order to ensure the best functionality of the software, I redesigned the original tests which included a lot of requirements creep testing. This allowed for the tests to focus primary on what was required by the client. Furthermore, as the integrated software was not ready until Friday, I carried out sub system testing which allowed for identification of minor errors which were then addressed by the team members. I also worked on improving the documents which were delivered over the first semester. This included improving their quality as well as removing the feature creep which was pointed out by the client. As the group leader, I kept an eye on the whole team, making sure that the work I set out is done on time and in case any problems, I helped or assigned other group members to help people that needed it.

To conclude, the project went well. The team was able to produce good quality documents as well as software which met the client's requirements. Some minor requirements were not met which was indicated in acceptance testing as well as the test report documents.

Benjamin Erefagha Ayerite, [bea16]

This report puts a non-biased personal reflection on the contribution planned and carried out by Group 12 members. It sets out the basis of my role specifically.

Throughout the focus, the report reflects on incidents leading up to the submission of our project, and through a series of meetings with our group coordinator and a series of emergency meeting set aside so we could discuss, and point out possible improvement, the emergency meetings refers to the aspect of designs and review of documents, give out priorities and put up documents in place to present to our project coordinator.

A group of 9 students was put together to plan, organize and develop an Android application, as part of assessing the module “Software Development Life Cycle” (CS22120).
Group members in development of the application: dpb1, alh32, fik, anp28, srp11, lps1, lot15, thw10

My role was on the web design, which is the front end of the website and Tom Wise deals with the backend and mapping of the website. During the whole planning of the project I had difficulties understanding the data I will be working with because I have no mockup data to work with and the team also work and make adjustment as they move on. The application was finally reached with all the data format working as they wish. I find it difficult to cope with the data type that was stored in the database as I have no experience with decoding Blob data, this makes it

difficult for me to research and reorganize the layout of the website as there was little to no time available to make the adjustment.

I was really pleased with the effort of the team, every one showed devotion to their duties and Filip was a great leader, making sure everyone is coordinated in the whole project. I learnt a lot of new things and it gave me a good insight on how working with a team of programmers is like.

Declan Beynon, [dpb1]

During the group project I was tasked with being a programmer for the android program.

Overall, I thought that the project was a success and that as a team we worked together well.

I feel that there were some team mates that let the group down during some stages but overall I think that people pulled their weight.

During the beginning of the Project, we had issues with our Project Advisor. This meant that some decisions, (Such as setting up the Group Blog) were delayed until we knew for definite what we were supposed to be doing, this was very frustrating as we were then berated for not keeping a log of how many hours we had done up until November 13th

The group project gave me many new experiences with things I had not used before.

I have, over the course of the project, learned to use GIT, used an Apple Mac for the first time, learned my way around Android Studio, learned about Json posts and attempted to get to grips with Android Activities and Services.

I also feel that the Group Project has massively improved my ability to do documentation, (This was reflected in a mark I received for documentation in my AI module at the time – It was much better than Documentation marks I had received previously.) I found this to be massively useful especially as I feel I had not been taught how to do this correctly to a good standard in any lectures I have taken during my time at Aberystwyth University.

I enjoyed the Android workshop that I was told to go with Andy, even though there were issues with the Macs at the time, I found the class very beneficial and the slides explaining how to use Activities were crucial in my understanding of how the Android program could be created.

The team, in general, were hard-working and worked together well. I felt I was with a friendly group of people who I got along with quite easily.

The only issues that arose that I feel are worth mentioning are that Alex did not do as much QA checking as he was required and also turned up to meetings late frequently, and that during the construction of the prototype, Louis stopped contacting the group due to struggles with the pressure of the work.

During Work week, however, Louis more than made up for this and was an integral part of the team, helping fix issues that I had come across and being a Wizard at using GIT. Helping me set up a branch for bits of code that I could not get working in time.

Alex was late to even some days in work week due to 'oversleeping' and we were lucky that his deliverables were handled by Filip and Luke, getting it (and their own work) done in time.

Towards the end of the Work week though, Alex did turn up and became very useful in researching things to be coded and also did the 'About page' for the website.

During work week, I had struggled with setting up the waypoint tracking system due to some error with a thread. I quickly alerted the group leader of this issue and he set up an alert on the Git page.

After some help from an advisor for almost the whole advisory (Who also couldn't understand why my code didn't work) the code was fixed by Louis who got the code working.

I also wrote the Javadoc and comments for the entirety of the android program and the team were always happy to explain code that they had written that I didn't understand from looking at it.

Finally, I spent one day working with Andy in Pair Programming in making the local storage for tours on the android phone. I was upset to find that this code was not implemented in the final version due to some issues with time.

Overall I thoroughly enjoyed working with the team and they were good at keeping my moral up when bugs in my code left me very frustrated.

I would happily work with the group again but disliked using Android studio. If I could redo the project I would like to be able to use Eclipse, which did not seem to work on the mac computers.

Sion Purnell, [srp11]

At the start of the project my role was to be a part of the testing team this was the position alongside Andy. This role was to create the test specification and to be testing the program when it was to be completed.

At the start of the project I was put on to write some parts of each document that we was to produced and the team worked well on separating the documents for everybody to complete a different part make sure each team member had work to do.

The only problem is that the team was producing addition features for the project that wasn’t needed and to put them in was to put more work load on to certain members but when we found out that we have put a lot of feature creep in the project we went through the documents and took them out.

At the stage of creating the test specification I and Andy did a good job of producing this document on time but at the stage we still had our feature creep in the project which lowered our mark for this document.

At the stage of producing a prototype and creating the design document we lost communication with a team member which had a major impact as he was the lead programmer on the android team. This meant that out prototype for our android app was far off what we wanted to show, as no features were working at this stage and all the user could do was to switch from 1 activity screen to another. Due to the loss to communication my role got changed from the testing team to the android team in which I created the home screen for the prototype, which was the only thing I was capable of doing at the time because I didn’t have any knowledge of programming in java.

At the stage of implementation I had done lots of work for the application. The things I had done on this part was editing certain screen that had already been created, create additional screens that was needed due to the fact when we removed our feature creep from our documents we accidently took out a feature that was already needed. I refactored all the pages to make sure that each class had the appropriate name for them. I adjusted the layout for the screens and applied a colour scheme for the application. I created the screen to which all locations would be displayed which works really well. I created the edit location screen which changing the details within the tourLocations arraylist. I created constructors for the tourlocations and for the tour. To which

these constructors was mergered with another team member constructors. I created the part of the application to allow users to take a picture from the camera or from the library on the mobile device, which this part was hard due to the fact of mobile phones with o.s. 4.4 kitkat would break at this part because we was not aiming for 4.4 and I could not find a solution for it. Also later on I had to fix this part so that if the users selected the camera and didn't want to take a picture then they can press the back button and the application wouldn't break. I also created a simple for of validation on all the input fields to check to make sure that a user must enter in information before they can proceed further.

The implementation stage went well and the android application we produced was what the specification require but the application as it is has major security flaws and is missing certain checks for the user in case they accidentally do something they didn't want to like deleting their tour.

Andrew Poll, [anp28]

At the start of the group project I felt cautious about whether other group members would do their fair share of work or if someone would try to avoid tasks. After our first group meeting I felt happy with our team, we all seemed to get along without any conflicts or arguments.

For this project I was first on the testing team, I felt confident about my first task which was to create a testing specification document for the Android device. During the creation of the document I made some syntax errors which were corrected by Sion This document was then reviewed by the majority of the team. Comments were made about some missing tests, I reviewed the document and the changes were made. At the second review of the document the team and I felt confident enough that it was at a suitable point to submit.

I was then moved to programming and helped to create the prototype as Louis shirked his job. We were unsure if we could get a working prototype in by the deadline, but Louis returned to help and the prototype was completed. I felt comfortable being on the programming team as I had taken part in an Android training lesson and I am capable of programming well in Java. I remained on the programming team for the remainder of the project.

At the start of integration and testing week I felt confident that we could create a program which would be capable of the requirements set by the client's documentation, others in the group were less confident. My first task was to create a database on the Android device. I had never created a database before and was unsure how to and how long it would take. After two days of attempting to create a database, with no success, we decided to save the information as a serialized document. Struggling with creating a database and changing to writing as a serialized document felt better as I know how to do it. We wrote a test class for it and it worked fine this took half the day, but took a further two days to implement into the existing code. I was not too happy in the fact that it had taken four days to save data to the device. On the last day as I had not updated my version of the code, it was thought too unsafe to merge and was scrapped twenty minutes before the deadline. I found this annoying but not unexpected as I had been told there maybe problems with trying to merge.

I am feeling more positive about how to work in a group and what to expect in future collaborations. From this however I have learned how to program java for Android and work better in a group. Improvements I would make would be to update my code more.

Louis Taylor, [lot15]

During this project I feel I was a very important team member, carrying out a lot of the work needed for the project to progress.

During the initial phases of the project I carried out a lot of the design of the application. These tasks included designing and creating mockups of the UI design, leading the design of the protocol between the server and the Android client and drafting the JSON schema once the protocol had been discussed and decided upon. During this phase I was a very active participant in group discussions, particularly about technical decisions.

During Integration and Testing week, I coordinated the effort to create a working product with the other technical team members. I assigned tasks to the other members of the android team and took over if they were unable to complete them. I made sure the web team was on track for completing the functionality needed for project completion in order of priority. I helped debug the PHP server side software, in particular the JSON string decoding. I refactored and improved the robustness of the code written by other team members when necessary. I was also in charge of maintaining a clean git repository and fixing issues team members had with version control.

The server used for hosting the database and server side components were hosted on my personal VPS (virtual private server), so I was also responsible for setting up and configuring the necessary parts for the system to run: Apache, mySQL and PHP. Also administrating it during the week.

Luke Suggett, [lps1]

I believe that this project under the circumstances has gone as well as could have been expected. Any group project at university level is going to be demanding, and in our project we were trying to emulate the software industry but in reality without truly analogous employment. For example in our allotted groups, we aren't constantly working as a team, neither the hierarchy nor the procedures are truly alike. This could make things very challenging when unforeseen circumstances occurred, including the disappearance of a group member.

Although everybody contributed to the project there were several small problems that would have been inevitable in any group work. Communication issues and differing standards stood out as the biggest problems, especially at the beginning. From my end (Documentation/Testing) this was manifested as document presentation/software tool issues along with lack of QA checking. For instance, somebody would produce an incorrect piece of documentation using a tool that would give a different aesthetic look from the previous diagram. This would be QA checked and accepted, although incorrect. As relations were still in their infancy it could be difficult to bring up the issue, which would mean that at a later stage this would have to be remedied, adding to the workload. There was also a potentially serious incident where one of the Android coders stopped answering any communications or producing work. With deadlines looming (Design Specification & Prototype Demo) the group leader re-structured the group to have a larger Android team and shifted people to different responsibilities. This restructuring worked and we were able to meet deadlines.

However this occurred when the group project was in it's infancy. As the project progressed we began to communicate much more efficiently, realising that there would be much more overlap of responsibilities, and began to produce much better quality work, more efficiently. This was especially true of coding week which compared with the chaos that I witnessed last year went very smoothly: everybody knew what they were doing, how to achieve their objectives, and were communicating with each other. I was also given time to correct issues with the documentation,

including formatting and presentation. This is I believe important as the documentation is worth 75%. It was also good to see everybody utilizing the repository. This made progress much smoother as it was possible to keep track of what was happening, who was working on a particular task, and see what issues needed to be resolved via the GitHub issue system.

Although there were niggling issues at the start, I believe the group project went very well once they were resolved. The project then began to gain momentum and function efficiently. I have learnt a lot from this project, especially the value of team work with clearly defined roles, and the need to resolve issue at the earliest opportunity.

Thomas Wise, [thw10]

This is the personal report for Thomas Wise on the CS22120 group project for 2013/2014. From the onset I knew that it wouldn't be impossible, but I also knew that it would rely on the specialisation of a few team members. The capacity of leadership was already present so it was just a case of assigning members to their respective fields, of which there was a shortage. Our group did have some great specialists, in fact there was every skill the group needed in order to build a functioning application to be distributed.

Given the information within the design specification I was quickly able to manufacture a working preview of what we could expect from the full platform. My own work was able to accelerate ahead of schedule at times but eventually would have to depend upon the mobile application to be complete. Quite often at times in development I did wonder if there was any front-end progress for the web or any android development happening whatsoever. And eventually I found myself unable to continue due to gaps in technical knowledge that relied on the methods being incorporated by the android application… methods that had not been decided upon yet.

My role as back end developer seemed to strike me as the person who would set up the database table structure, and ensure that data was being retrieved correctly to the page which would be provided by the front end developer, with the API already present on the page. Sadly, my expectations were very much shattered. The website itself appeared several times during development, every time with suspicion that it was a template website. This website only actually managed to appear within the last week of development and was not worth hiding months of "hard development" for, a single page with barely the tools I had mentioned would be required, most of which had to be written over to ensure a working build. Despite my role as the backend developer, I had to make decisions and integrate most of the front end myself also.

Sadly due to the missing components so close to the deadline, at least one of the listed features of the website could not become reality. It was only late on that I discovered the use of Base64 strings would be the representation of images on the website, before that there were no other suggestions to be implemented. As such, individual points on the map API would not show images of their respective location and images were not queried by walk, instead they were simply fetched in their entirety as part of the database table and all were echoed forth onto the page. There was no structure as to how these images were organised, yet there could have been given an appropriate timeframe to work off as long as the other components of the system were up to speed with my own.

Should I embark on another project I wish my colleagues would remain present at all times and for their contributions to show immediately.

Alexander Heaven, [alh32]

Each week since the project began, our group has had two meetings a week to check progress, discuss what needs to be done and issue tasks to group members. Assigning roles to each group member went pretty smoothly and everyone seemed happy with the role assigned to them. I didn't notice any real personality clashes within our group, although at a few points communication could probably have been better between the group.

A few weeks into the project, the group leader and I attended a two hour session on GitHub. This showed us how to get GitHub set up for use with our group and creating a group repository, so that all members can see what is uploaded and upload documents/code as necessary. This session helped me learn how to use GitHub, although at first it did seem over-complicated and awkward to use, so it did take some getting used to.

Throughout the project I have conducted numerous quality assurance checks on all final documents to ensure that the deliverables meet the standards and that they are in an appropriate format to be submitted. This required me to become familiar with the quality assurance documents that the group were provided with, in order to know how the documents should look and what sections/information had to be included.

I also kept an eye on the version of each document, to make sure that the versions were appended correctly when edited and that the right version was submitted when complete. The version configuration file on GitHub shows the documents name and the configuration ID associated with it, and the latest version available.

I have contributed the sections assigned to me for many of the necessary deliverables to be submitted. This includes sections for documents like the project plan, design specification, test specification and the end of project report. I've created sections like introduction, sequence diagram and historical account, as well as a few skeleton documents to be filled in. I also had to set review dates for the design specification.

I quality checked every set of minutes which have been produced for every meeting the group has had, ensuring that each set were in the correct format and included the necessary information, such as author, members in attendance and date of meeting. This also meant changing the filename to a pre-specified format, for easy sorting. Along with this, I compiled a list containing the filename and date for all of the minutes that have been recorded.

3.4 Revised project plan

PROJECT PLAN: GROUP 12

Authors: bea16, dpb1, alh32, fik, anp28, srp11, lps1, lot15, thw10
Config Ref: SE_12_PP_01
Date: 29/01/2014
Version: 1.1
Status: Final

Department of Computer Science
Aberystwyth University
Aberystwyth
Ceredigion
SY23 3DB

3.4.1 Introduction

3.4.1.1 Purpose of this Document

This document is here to ensure that the application we intend to build, a walking tour creator application for android, fully meets the requirement specification, as given by the client. It will help make sure that we cover all necessary features and that a quality product in produced.

This document will also outline how parts of the application will interact, along with how the user interface will actually look. As well as this, the document will highlight the project schedule, showing the major tasks to be completed, along with when each task needs to be achieved by. This will be displayed in the form of a Gantt chart.

3.4.1.2 Scope

This document should be read by the client, with reference to the requirement specification to ensure that what we intend to build meets what they expect us to build.

This document contains the following sections:

Overview of the proposed system: This section will show how both the web and phone applications will run and what will include descriptions on the functionalities of the system.

Use Case diagrams: This will show how every part of the program will interact and the links that must be in place for successful operation.

User interface design: This section will show how the finished application will look, as well as giving an idea of how it will feel to use. It will help ensure that the navigation and use of the app is simple, but suitable.

Gantt chart: In this section the major tasks will be made clear, along with when each task needs to be started and completed.

Risk analysis: This section stands to outline the possible problems we may encounter throughout this project, and how these problems can be avoided or solved.

3.4.1.3 Objectives

This document is here to show our preliminary plan for the proposed system and show that we have interpreted and understood the clients requirement specification.

The objectives this document has are:

- To give an summary of the project and the procedures, technologies and languages we will be using.
- To show the how each main part of the project with join together and interact.
- To display what we aim for the user interface to include, as well as how it will look when finished.
- To give a plan, with a schedule, of the tasks and milestones that need to be accomplished throughout the project.
- To highlight and understand possible issues with the project, and avoid them where feasible.

3.4.2 Overview of the proposed system

3.4.2.1 Overview of the web system

PHP - Phone will upload all data to a HTML file for a PHP file to parse from. PHP files will be able to create a connection to our database at db.dcs.aber.ac.uk or any other server capable of PHP and MySQL, and pass all relevant items of information into the tables according to the needs of the data. The phone can upload data using a combination of POST data retrieved from a HTML file and used through a PHP page. During use of the system the information posted to the HTML page from the phone will be used to INSERT INTO the tables for paths and points.

MySQL – Connect to dcs.db.aber.ac.uk or other server, allows information to be stored and retrieved for later use. A dummy database will be used to store false data for retrieval and allow us to commit tests to train our real database before use. Series of queries will be required to make use of the database. Must include fields for point descriptions, names, latitude, longitude, images associated with the point of interest, and path associated with the point, and in what order the point occurs. This will allow for a correct line to be plotted on the map, based on where the person walked.

HMTL – HTML code will be used to embed PHP code which can then be used to add functionality to the website itself.

CSS – CSS code will be used to make the website look more professional and eye-catching.

Google map – the website will have a Google map which will allow the user to see the whole walk. The map will be displayed as soon as the user accesses the website and each tour will be shown after selecting in from the drop down menu, as described below.

Drop down menu – User will be able to select a tour from the drop down list. User will then have to press select in order for the tour to load onto the Google map.

Connectivity – Phone must automatically acknowledge a loss in connection to either Mobile or WiFi services. If lost during an attempted retrieval or transmission of information. If unsuccessful then prompt the user a failed attempt at transfer, or return confirmation upon success. Must also store transmission information for return of connectivity, so a post can be made on the user's request.

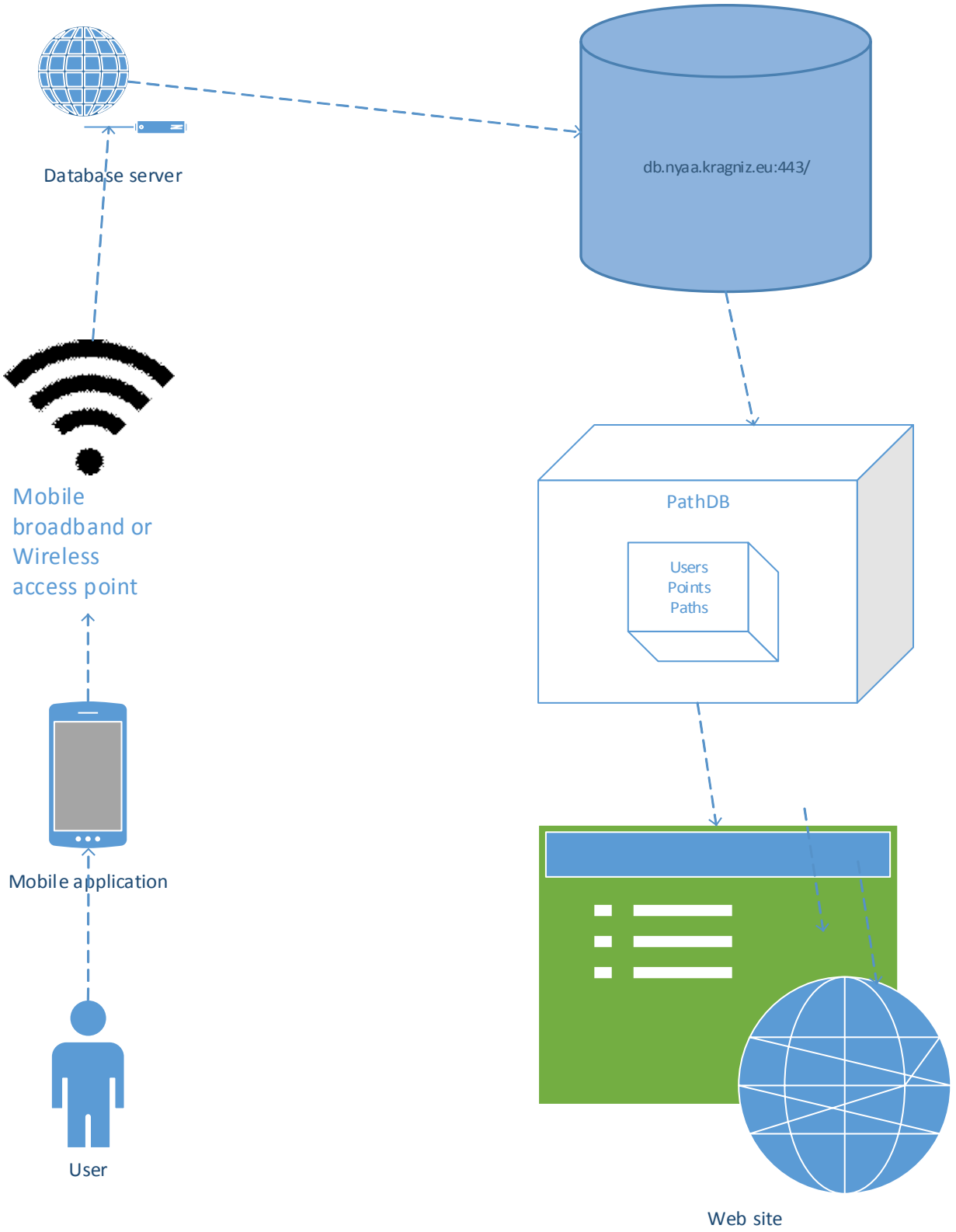

Figure 5: Deployment diagram

3.4.2.2 Overview of the Android system

The android application is expected to have the following functions:

Uploading Walks

Users may select a walk to upload to the server. The entire data contained within the walk is synchronised with the database on the server.

Recording Walks

Users, while recording their walks will store information as local data on the hand-held device. This prevents any information from being lost due to a connectivity error. Users will also be able to record walks offline due to this feature. Users will be able to take photos, choose significant landmarks and write descriptions about each walk here.

Local Walk Data

This is where walk information will be stored before the walk information such as Walk name, Journey time and the Descriptions of the walk has been sent to the database. This is important in case of a connectivity failure to prevent.

Editing Walks

Users may change details about their walks before finally sending the information to the server. Here they can add a large description to their walk and can delete any walks they are unhappy with. The locations within a tour will be shown as a list. The user will be able to click on each one of these which will then bring up the details of that location which can be edited.

These will be explained further in the user interface design section.

3.4.3 Use Case diagrams

3.4.3.1 Android use case

Figure below shows the use case diagram for the android client. It describes the functions that the user will be allowed to use on the android app and a description of what each function does.

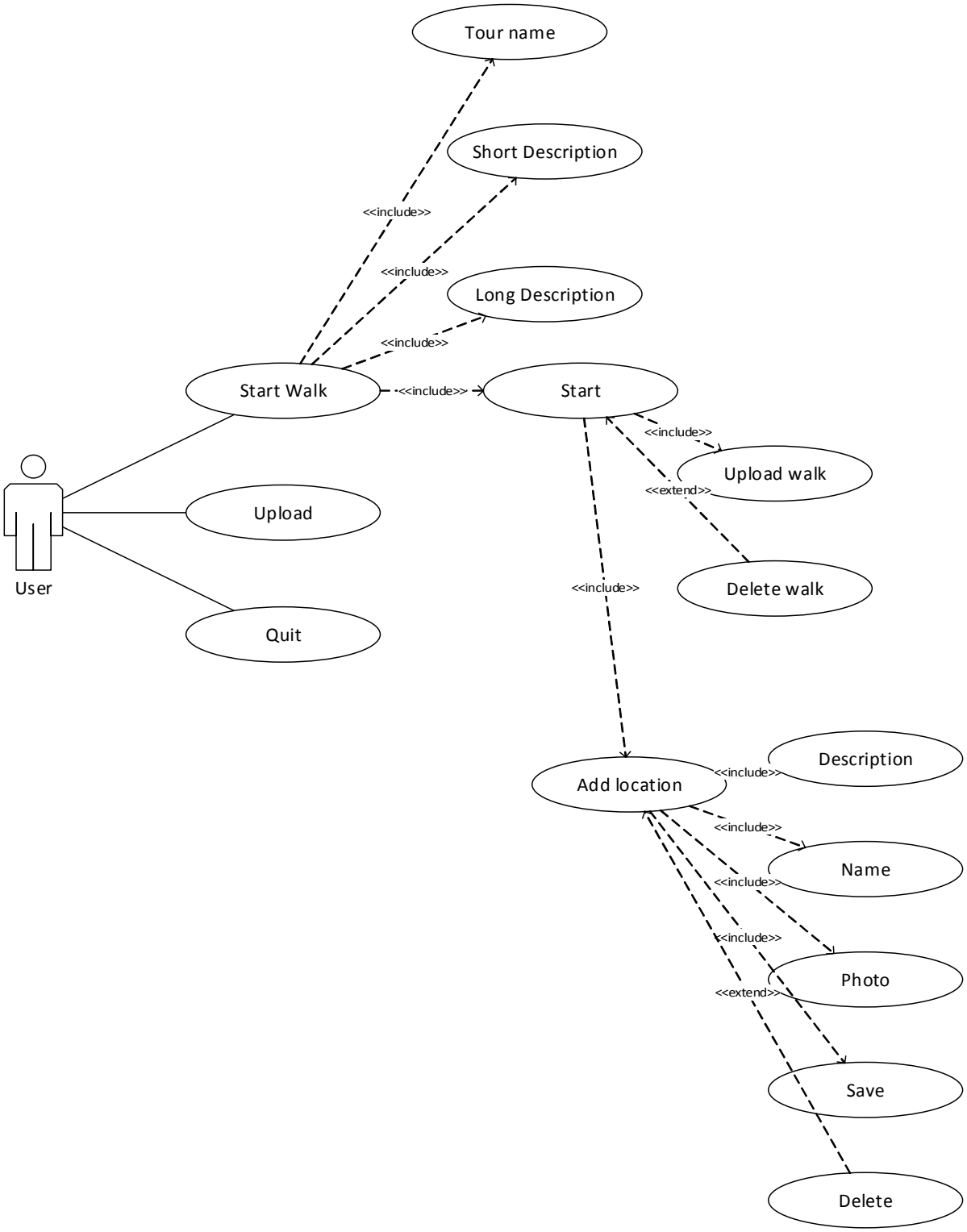

Figure 6: Use case diagram for the Android application.

Record new walk

When a new walk is selected the user will be asked to enter information about the walk such as the walk name and a short description. When the user confirms the information inputted a snapshot of the time and GPS location will be taken and a request will be stored locally on the device.

Add point of interest (location)

When the user adds a point of interest a location description, name will be asked for. The lat and long co-ordinates as well as a time stamp of that location will be recorded and stored. The user will then be given a choice to take a photo for that location. This will be stored locally before uploading to the server.

Save/Upload walk

When the walk is complete and the user saves the walk then they will be asked for a long description of the walk and the user has checked and confirmed the information all the data will be sent to the data base. If no connection can be made then a message will appear warning the user that it could not connect to the server. If a successful connection is made, a request will be sent to the server to add the local information to the database.

The server will then return a token of 1 or 0. If Successful the local data will be removed otherwise a warning will tell the user that there was an error with the server and to try later and keep the local data stored to the device.

Edit walk

The user will be given the option to review a walk. A list of locations within a walk will be shown. User will then be able to click on each location edit its details.

Cancel walk

If the recording is cancelled the data stored locally on the device will be removed, a message will notify the user that the walk has been cancelled. User will be able to cancel the current walk and start a new one.

3.4.3.2 Web Use Case diagram

Below describes the interactions the client program has with the server. It does not describe the administrative functions, since these will probably be performed by a special, direct client. Note: Although there is a little figure of a person on the diagram, this is actually two programs talking to each other.

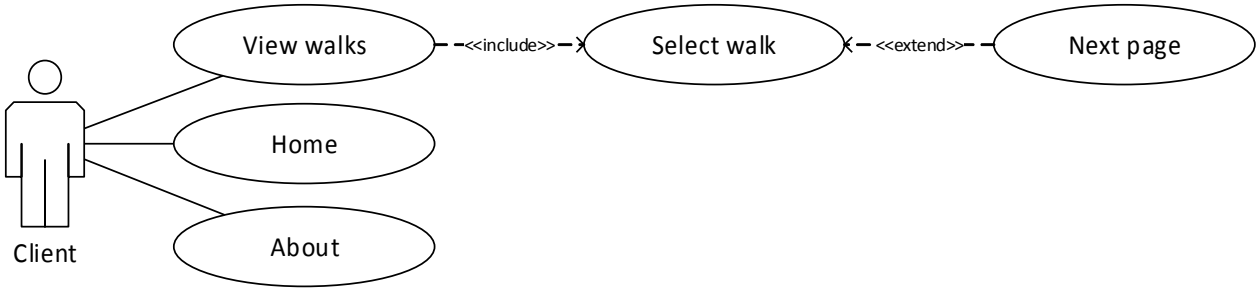

Figure 7: Use Case diagram for the web side of the system

Drop down list

A drop down list will allow the user to search through the walks created. Once selected a walk from the list, the user will press select and a given walk will be loaded onto the Google map. A number of locations will then show on the Google map.

Google map

A google map will be provided which will show the locations and way-points within each tour. These will be joined by a line, based on where the user walked. The locations will be joined in the order of how the user accessed them.

3.4.4 User Interface Design

Figures below provide mock-ups of the final user interface screens within the web application as well as the Android app.

3.4.4.1 Web User Interface

Figure 8: Mock up screen for the Web Application

3.4.4.2 Android User Interface

Figure 9: Home Screen

Figure 10: Create Tour Screen

Figure 11:Start Tour Screen

Figure 12: Add Location Screen

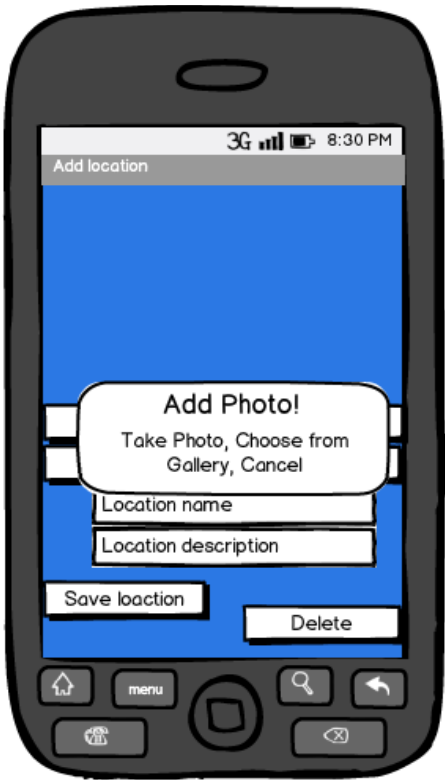

Figure 13: Add picture screen

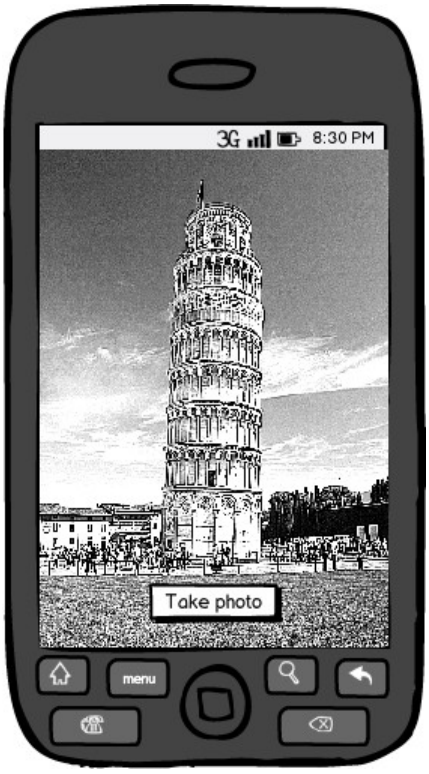

Figure 14: Taking pictures

Figure 15: Edit locations

3.4.5 Gantt Chart

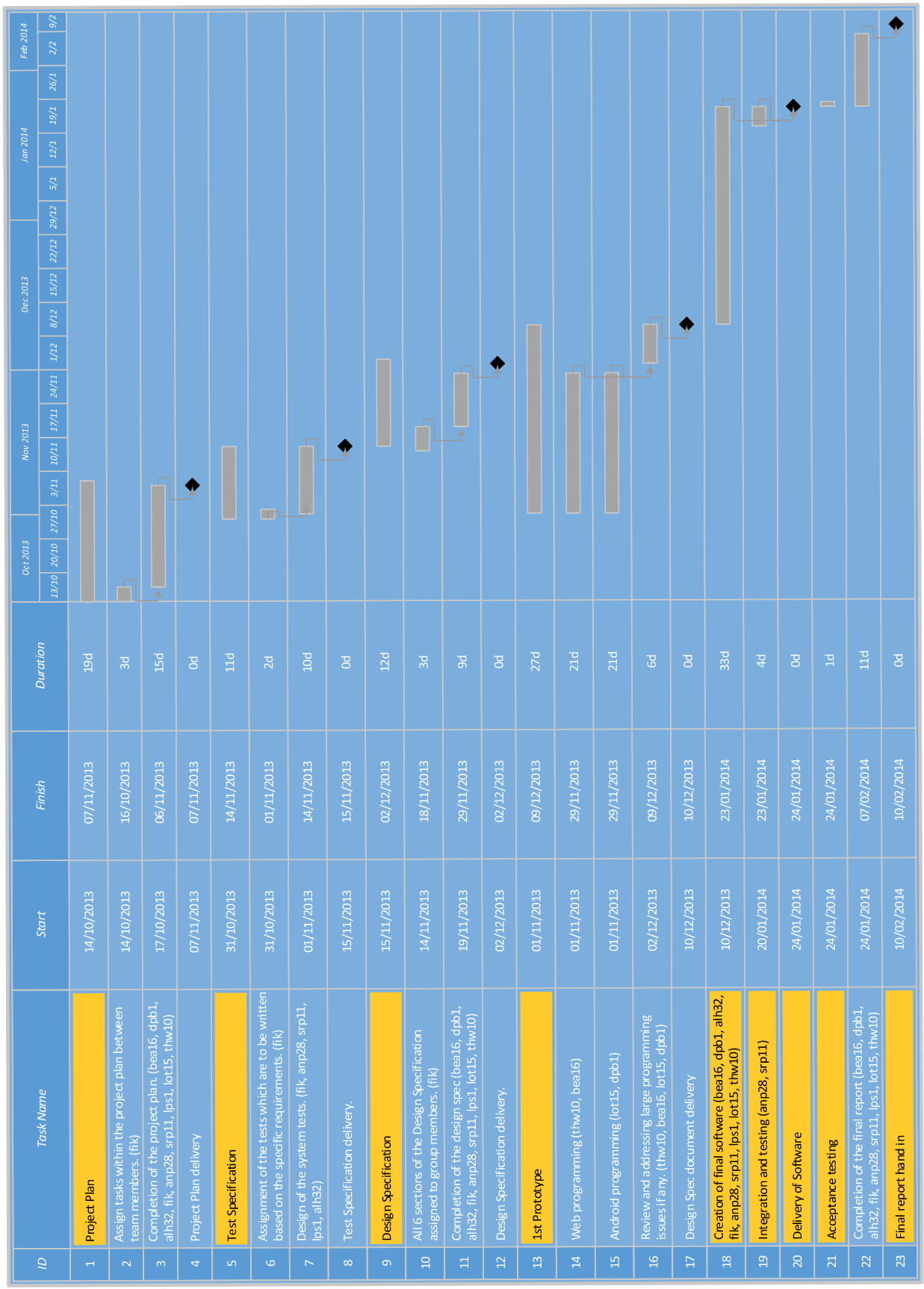

Figure 16: Gantt Chart

3.4.6 Risk Analysis

3.4.6.1 Preliminary risk analysis

Risk	Probability	Effects	Strategy
Loss of team member	Low	Serious	Reorganise team with more overlap of duty of each member.
Illness of team member	Moderate	Serious	Reorganise team with more overlap of duty of each member for the period of member illness.
Documentation incomplete/late	Low	Series	Aim to complete parts of documentation well in advance of deadline. If problems occurs contact team members so that they can help to complete / fix errors.
Scale of Project underestimated	Low	Catastrophic	Produce project plan detailing the scale and scope of proposed system.
Requirement change	Low	Tolerable	Communicate with client. Communicate with group / project leader. Re-arrange project to incorporate feature making sure that it does not interfere with project deadline.
Loss of Project Leader	Low	Serious	Restructure team & reorganise duties with more overlap.

Figure 17: Table showing the preliminary risk analysis

3.4.6.2 Life-cycle risk analysis

Risk	Probability	Effects	Strategy
Feature Creep	Moderate	Tolerable	Adhere to design documents. If problems occur the development team should communicate with the project leader.
Version control errors due to naivety	High	Tolerable	Make sure that the version control expert fully trains each member of the team & that they understand proper Git use.
Loss of source code	Low	Catastrophic	Use a repository, have multiple backups. Contact project manager if does occurs.
Android Cross Platform Development Issues	Moderate	Tolerable	Agree on specific Android version that the development team will use.
Parts of system late or incomplete	Moderate	Serious	Adhere to Gantt chart and try to complete well in advance of deadline. Communicate with project leader / team.
Parts of system defective	Moderate	Serious	Use proper testing strategies, e.g. Black-box / whitebox. Communicate with team

			to daft members in to help with fault finding.

Figure 18: Table showing the Life-cycle risk analysis

3.4.6.3 Operational risk analysis

Risk	Probability	Effects	Strategy
Database performance	Low	Serious	Investigate database performance, specifically MySQL vs. PostgreSQL.
Unresponsive UI - outlined in performance requirements (PR1).	Moderate	Serious	Investigate design patterns for responsive Android design.
Unable to send walk via HTTP POST or formatted to MIME specification outlined in functional requirements (FR6).	Low	Serious	Investigate the Multipurpose Internet Mail Extension standard in conjunction with PHP's $_POST superglobal.
Unable to obtain GPS coordinates.	Low	Serious	Investigate android.location API, specifically the LocationManager class.
GPS consumes too much battery.	Moderate	Tolerable	Investigate Location Strategies especially Network Location Provider.
UI may not be intuitive to regular computer users as outlined in External Interface Requirements (*EIR1).*	Low	Serious	Test UI, gather & incorporate feedback from test users.

Figure 19: Table showing the operational risk analysis

3.4.7 References

[1] Software Engineering Group Projects. Design Specification Standards. C. J. Price, N.W.Hardy and B.P.Tiddeman. SE.QA.05A. 1.7. Release.

3.4.8 Project Plan Document history

Version	CCF No.	Date	Changes made to the document	Changed by
0.1	N/A	31/10/13	N/A Original draft.	fik

0.2	N/A	07/11/13	Changes/updates to each section of the document.	fik
1.0	#7	27/01/14	Corrected major formatting issues.	LPS1
1.1	#9	29/09/14	Updated and inserted use case for both Android and web.	LPS1

3.5 Revised design specification

Design Specification: GROUP 12

Authors: bea16, dpb1, alh32, fik, anp28, srp11, lps1, lot15, thw10
Config Ref: SE_12_DS_01
Date: 28/01/14
Version: 1.1
Status: Final

Department of Computer Science
Aberystwyth University
Aberystwyth
Ceredigion
SY23 3DB

3.5.1 Introduction

3.5.1.1 Purpose of this Document

This document is here to ensure that the format and content of the information which will be a part of the final design specification document in the the software engineering group project. It will help to make sure that all the major details and functions of the system are described. This will include a description of the programs which are a part of the overall system with a further description of significant classes in each one of them. Component diagram will be shown. The system will be described in detail using a variety of diagrams, algorithms and data structures.

3.5.1.2 Scope

This document specifies the standards for writing software design specifications. It describes the necessary layout and content of a design specification. This document will be produced in accordance to the design specification standards document **[1]** SE.QA.05.A.

The main sections to be covered are:

- Introduction: to include the purpose of the document, scope and objectives.
- Decomposition Description: this section will include a description of the programs in system as well as significant classes in each program. A table showing Specific Requirements and how they are mapped into classes will also be provided.
- Dependency Description: this section will include a component diagram. What is more, inheritance relationships will be described.
- Interface Description: this section will include interface specification for each of the classes.
- Detailed Design: the final section will include a sequence diagram, significant algorithms, significant data structures as well as an entity relationship diagram.
- References: will include references to the external sources used when producing the document.

3.5.1.3 Objectives

The main objective is to aid the production of a design specification which is a complete and accurate translation of the client's requirements into a description of the design elements necessary for the implementation phase. These will include the software structure, components, interfaces, and data. In a complete design specification, each requirement must be traceable to one or more design entities. **[1]**

Objectives are:

- To describe the programs in system and how they will interact together
- To show the components and inheritance within the system using component diagrams and inheritance relationships.
- To specify interfaces for each program.
- To describe significant algorithms and data structures.

3.5.2 Decomposition Description

3.5.2.1 Programs in system

Firstly we decompose the software into 3 main subsystems which are:

- The android application.
- The server.
- The web application.

The Android Application (WTC)

The Walk program provides a user interface on a android platform to enable a user to make a walk, way-points, amend walk, way-point, take pictures of way-points and put in information about the way-points
Implements requirements (FR1),(FR2),(FR3),(FR4),(FR5),(FR6),(FR7),(PR1),(PR2) and (DC1) . This program will be form-based so it will comply with (EIR1). The walk program will send the information it generates from the user as described in(FR6)

The Server (WTD)

The Server provides a storage space for the database and all the information about walks so that they can view them. This should implement (FR9),(PR1) and (DC3). The Server will also use a Test Data file which is described in (DC2). The server will save the information it receives inside of a database.

The Website (WTD)

The Website provides a professional user interface for a person to view a walk online. It also allows a user to search for a specific walk. Details on each walk will then be displayed. Implementing requirements (FR8),(EIR1) and (PR1). The Website will query information from the Server and display it on to the google maps interface so the user can take the tour. The locations within the tour will be shown as pins. The line showing the tour will be drawn based on the way poins recorded during the walk.

3.5.2.2. Significant classes in each program

Significant classes within the Web application

As the PHP is procedural each function is treated as a class for the purpose of description.

index.php

Description of classes for the file index.php:

src="https://maps.googleapis.com/maps/api/js?key=&sensor=false">** This piece of javascript is part of the google maps API, it gathers a user key and allows access to the functionality of the google map.

function calcRoute() The calcRoute() function takes the parameters of an array, from there it will plot the path between all points in the array and direct the user to follow, drawing the path on the map.

function initialize() The initialize() function is the basis for booting the map. Within contains the while loop that accesses our MySQL database and returns all points for a given path. During this time it will display all points with their appropriate marker location and give the descriptions stored of each and every point.

$con Establishes connection to host based on server permissions.

$res Returns appropriate data from selected database for use with google map.

var LatLng, var ContentString These are variables concerning the retrieval of data from the database, and as such are unique per point. LatLng creates a new marker posted at the given points within our tables, the contentstring updates the information window with the stored point description.

google.maps.event.addListener This creates the information window given at every marker on the map. Creates a listener that waits for a mouse click to open the information window.

var marker = new google.maps.Marker Constructor for the marker, reads the position it should be set at, the animation accompanying it upon opening of the map, the drag mode, the icon and the map it should be assigned to.

google.maps.event.addDomListener(window, 'load', initialize) Creates map upon load of the window, runs the initialize function to ensure all points and paths are plotted upon creation. Opens the map within the specified window space.

upload.php: This page will receive the information as sent by the android client

Significant classes within the Android application

Basic descriptions of each significant class within the phone application. A more detailed description provided in the Interface description section.

Tour.java – stores information about each tour. This will include a string for the name, an array of locations, a string for the description, another string for the longer description.

HomeActivity.java – holds methods that take user input, iit is the home page of the application.

TourCreatorActivity.java – Creates the screen and GUI which takes user input to intern create a tour based on that information.

TourActivity.java – links going to adding and editing locations as well as stopping the tour and uploading it to the server.

LocationCreatorActivity.java – allows to put in the values for each location as well as the picture. This can then be save in the array of location or delete that location.

EditLocationActivity.java – Same activity as the LocationCreatorActivity except for instead of creating the location, it finds the location in the array list and changes its values.

ViewWalksActivity.java – shows all the walks stored in the array list and allows the user to edit it.

3.5.2.3 Modules shared between programs

The only modules that will be shared between the web application and the android side application is the JSON script that would pass information between the phone and the web storage page via http. The android would pack data into JSON regarding points and this would be sent to the web application to be parsed out and inserted into the relevant tables. The android user may specify as to whether a new path is being generated, and if this is true a new row must be added to the paths table to signify that this path and all points associated are separate entities. Similarly all points made under this new path would be associated on the database side via a pathID foreign key within the rows for each point. JSON must also be able to pack images into a string to be sent, which would allow for easy transmission and ensure that the data can be received and stored. Decoding of such data need only happen upon arrival at the database or on the page as and when requested.

3.5.2.4 Mapping from requirements to classes

This table shows each functional requirement that must be met, and which class provides its operation. This will ensure that all requirements are covered, and show the impact that changes to the design may have.

Functional requirement	**Class providing requirement**
FR1- Startup of software on Android device	HomeActivity.java
FR2 - Providing info about the whole walking tour	TourCreatorActivity.java
FR3 - Adding locations to the walking tour	TourActivity.java
FR4 - Adding photos to the walks	LocationCreatorActivity.java
FR5 - Cancelling walks	TourActivity.java
FR6 - Sending the walk to the server	TourActivity.java
FR7 - Switching from the WTC	All activity classes
FR8 - Trying out a created walking tour	index.php
FR9 - Saving data on server	database

Figure 20: Table showing the classes which provide functional requirements

3.5.3 Dependency Description

3.5.3.1 Component Diagram

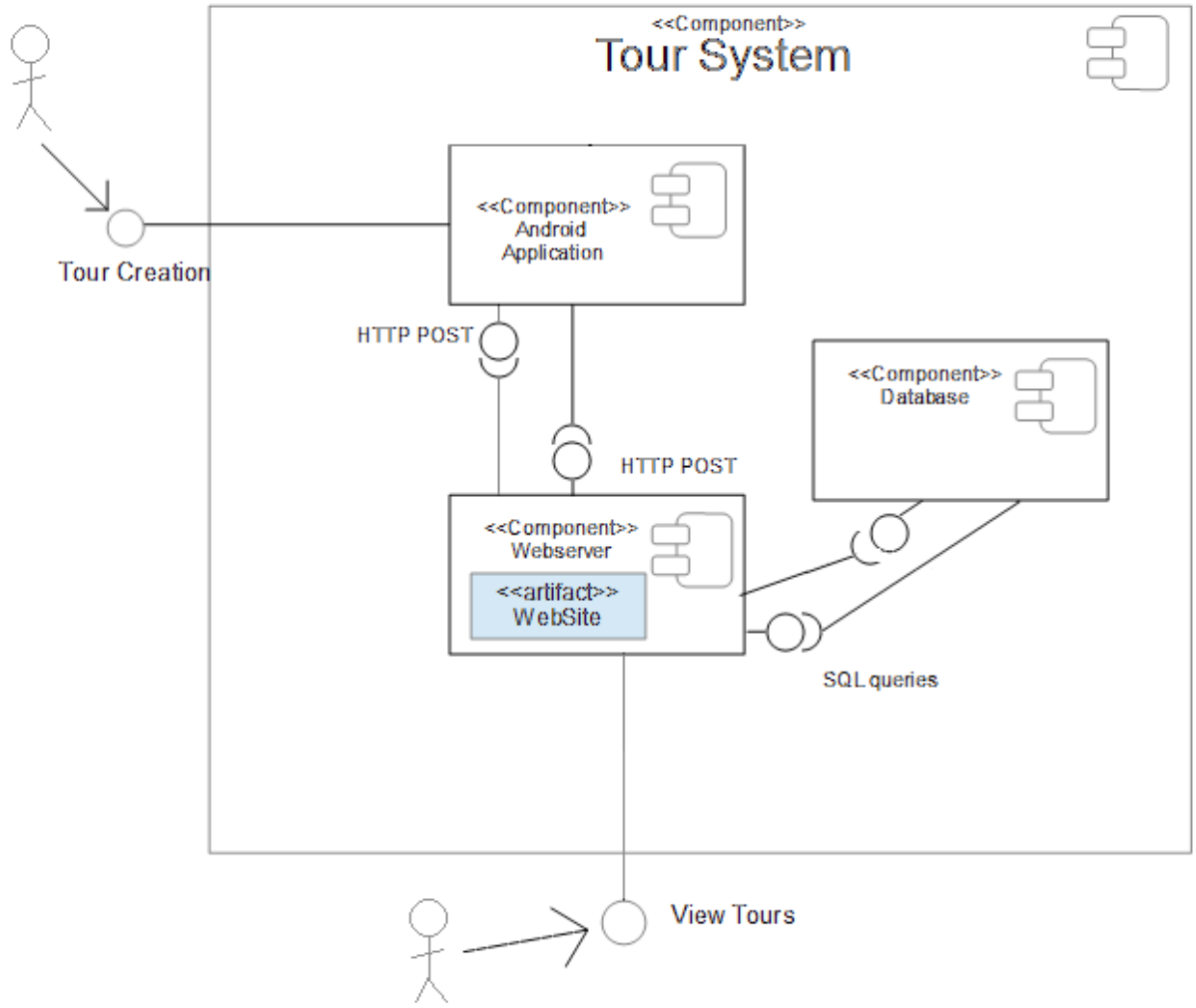

Figure 21:Component diagram of the system

3.5.3.2 Inheritance Relationships

Activity classes will inherit from Android API packages.

3.5.4 Interface Description

3.5.4.1 Web Class Interface

```
<DOCTYPE html>

<html>

/* Using HTML 5 Doctype to support our project. This ensures that the google
maps API will be fully supported on the latest web standard, that the phone
viewport can be resized and viewed for and there will be no implementation
issues concerning any new web tools we may wish to access or incorporate. */
<head>

<meta name="viewport" content="initial-scale=1.0, user-scalable=no" />  <style
type="text/css">

html { height: 100% }

body { height: 100%; margin: 0; padding: 0 }

#map-canvas { height: 100% }

</style>

/* A temporary inline CSS for use with this test page. Only contains code
necessary for display of the map.*/

<script type="text/javascript"  src="https://maps.googleapis.com/maps/api/js?
key=**************************************&sensor=false">

</script>

/*Fetches a google maps API key to be able to use the google maps

functionality and tools across the site.*/

<script type="text/javascript">

var image = 'icon2.png';

var waypts[];

/*Variable name "image" accesses the location of the saved marker image, can
be removed to revert to default marker or file name can be changed to
allocate new marker should future developments occur. Variable name "waypts"
is an array that contains the lat and long values of points along the path to
be taken for the lines of direction to be drawn for the user. This can only
hold a size of 8 or the path will not draw due to API limits.*/

function calcRoute()

{

var request=

{

origin: new google.maps.LatLng(<?php=$start['lat']?>,<?php=$start['long']?>),

destination: new google.maps.LatLng(<?php=$end['lat']?>,<?php=$end['long']?>),

waypoints: waypts;

travelMode: google.maps.TravelMode.WALKING

};

<?php=$pointpath = mysql_query("SELECT * FROM pathlines WHERE `pathID` =
$_POST["path"] && `pointID` = $_POST["points"];

while($b=mysql_fetch_array($pointpath)

{>?
```

```
for(i=0;i<8;i++)
{
waypts.push[{<?php=$b['lat']?>,<?php=$b['long']?>}];
}
<?php=}?>
directionsService.route(request, function(response, status)
{
if (status == google.maps.DirectionsStatus.OK)
{
directionsDisplay.setDirections(response);
directionsDisplay.setOptions({suppressMarkers: true});
}
});
/*Calculates routes between two points. Uses an origin point to find start of
path, and destination point to use as end of path. Travel mode dictates what
type of route and paths are allowed. Walking indicates any available path
over land, driving may only use roads however. Walking is appropriate for
prototype path. The tool behind this to be used is google map's very own
"polyline" type, which will run through all of the points specified between
two points selected. Suppressed markers is a tool used to prevent the
appearance of additional markers dictating start/end points of a path and
cluttering the GUI.*/
}
function initialize()
{
var mapOptions = {center: new google.maps.LatLng(52.413571,-4.073489), zoom:
14};
var map = new google.maps.Map(document.getElementById("map-
canvas"),mapOptions);
var infowindow = new google.maps.InfoWindow();
window.directionsService = new google.maps.DirectionsService();
window.directionsDisplay = new google.maps.DirectionsRenderer();
directionsDisplay.setMap(map);
/*Initializes map for later, starts by creating variables and generates paths
within the window. Adds path to the map and sets the position at center when
the map is first displayed. Also sets the rate of zoom so the entirety of the
path is covered.*/
<?php
$con = mysql_connect("localhost","root","");
if (!$con)
{
die('Could not connect: ' . mysql_error());
}
mysql_select_db("pathdb", $con);
```

```
$res = mysql_query("SELECT * FROM points");
while($a = mysql_fetch_array($res))
{
?>
/*Creates a connection to my local hosting server, selects database to use
and then chooses the data from which tables I am currently interested in.
Enters a while loop to assign each marker and create it as such.*/
var LatLng = new google.maps.LatLng(<?=$a['lat']?>,<?=$a['long']?>);
var ContentString = "<b><?=$a['shortDesc']?></b></br><?=$a['longDesc']?>";
var marker = new google.maps.Marker(
{
map:map,
draggable:false,
animation: google.maps.Animation.DROP,
position: LatLng,
icon: image
});
marker.content = ContentString;
google.maps.event.addListener(marker, 'click', function(){
infowindow.setContent(this.content);
infowindow.open(this.getMap(),this);
});
/*Uses value (Lat,Long) to create point marker on map, assigns variable
"ContentString" which creates a bolded up title as the short description,
followed by a new line for the long description per point. Marker is
generated for our map, with drag disabled, a drop animation on generation,
marker appears at the given (Lat,Long) position and marker at point is
assigned  to image variable given earlier. Listener sets up the information
to go into our information window and sets the function to open the window
upon the click of the marker happening.*/
}
?>
calcRoute();
/*Closes the while loop and calls the path finding function between points.*/
}
google.maps.event.addDomListener(window, 'load', initialize);
/*Upon loading of page, the map is generated with it's appropriate window
space, and the function initialize is called and therein all paths are
brought up and all points are created.*/
</script>
</head>
<body>
<div id="map-canvas"/>
</body>
```

```
/*Creates the map on the website, closes all HTML concerned with page.*/
</html>
```

3.5.4.2 Android Interface Description

This section will provide an outline specification for each significant class within the Android system.

3.5.4.2.1 Tour.java

```
public class Tour implements Serializable {

private String name;
private String shortDescription;
private String longDescription;
private ArrayList<TourLocation> locations;
private ArrayList<Double> latitudes;
private ArrayList<Double> longitudes;

/**
* The constructor for a Tour.
* It has a name, a short Description and a Long Description.
* @param name The name of the Tour.
* @param shortDescription A short description that describes the route taken.
* @param longDescription A description that can be used to describe the route
* taken in more detail.
*/

public Tour(String name, String shortDescription, String longDescription) {
}


public void addLocation(TourLocation location) {
}

public void addWaypoint(double lat, double _long){
}

/**
* Attempts to convert a Tour into a Json string, so that it may be sent to
the database via HTTP Post.
* @return a Json String that holds the information of a single tour.
*/
public String toJSON() {
}

public String getName() {
}

public String getShortDescription() {
}

public String getLongDescription() {
}

public ArrayList<TourLocation> getLocations() {
}
```

3.5.4.2.2 LocationCreatorActivity.java

```
public class LocationCreatorActivity extends Activity implements
LocationListener {

private LocationManager locationManager;

private String provider;

private double latitude = 0;

private double longitude = 0;

private String imageFilePath;

private ImageView imageView;

private Tour tour;

private Bitmap savedImage;


protected void onCreate(Bundle savedInstanceState) {

}

public void onPhotoClick(View view) {

}

private void onSelectImage() {

}

 public void onCoordinateClick(View view) {

}

protected void onActivityResult(int requestCode, int resultCode, Intent data)
{

}

public void onLocationChanged(Location location) {

}

private void setupActionBar() {

}

public boolean onCreateOptionsMenu(Menu menu) {

}

public boolean onOptionsItemSelected(MenuItem item) {

}

public void onStartAddLocation(View view) {

}

public void onStartDeleteLocation(View view) {

}

protected void onResume() {

}

protected void onPause() {

}

public void onProviderDisabled(String arg0) {
```

```
}
 public void onProviderEnabled(String arg0) {
}
public void onStatusChanged(String arg0, int arg1, Bundle arg2) {
}
```

3.5.4.2.3 HomeActivity.java

```
public class HomeActivity extends Activity {

protected void onCreate(Bundle savedInstanceState) {
}
public boolean onCreateOptionsMenu(Menu menu) {
}
public void onStartTourCreator(View view) {
}
public void onStartUpload(View view) {
}

public void onStartQuit(View view) {
}
}
```

3.5.4.2.4 TourCreatorActivity.java

```
public class TourCreatorActivity extends Activity {

protected void onCreate(Bundle savedInstanceState) {
}
private void setupActionBar() {
}
public boolean onCreateOptionsMenu(Menu menu) {
}
public boolean onOptionsItemSelected(MenuItem item) {
}

/**
* When Tour Creation begins, it stores the name, Short Description and Long
Description
* by creating a new instance of tour, passing in these parameters.
* If any field is left blank, a pop-up message is displayed, and the Tour *
is not saved.
* @param view the button that was clicked to call this method.
```

```
*/

public void onStartTour(View view) {
}
public void onStartTour(View view) {
}
}
```

3.5.4.2.5 TourActivity.java

```
public class TourActivity extends Activity implements LocationListener{
private static double TIME_BETWEEN_WAYPOINTS = 10000;
    private static boolean DEBUG = true;
    private LocationManager locationManager;
    private String provider;
    private double latitude = 0;
    private double longitude = 0;
    private double time = 0;
    private Tour tour;
protected void onCreate(Bundle savedInstanceState) {
}
public void onLocationChanged(Location location) {
}
public boolean onCreateOptionsMenu(Menu menu) {
}
public void onStartLocationCreator(View view) {
}
public void onStartDeleteTour(View view) {
}
public void onStartEditLocations(View view) {
}
public void onUpload(View view) {
}
protected void onResume() {
}
protected void onPause() {
}
public void onProviderDisabled(String arg0) {
}
public void onProviderEnabled(String arg0) {
}
```

```
public void onStatusChanged(String arg0, int arg1, Bundle arg2) {
}
}
```

3.5.4.2.6 EditLocationActivity.java

```
public class EditLocationActivity extends Activity{
private ImageView imageView;
    private Tour tour =new Tour(null,null,null);
    private String imageFilePath;
    private double latitude = 0;
    private double longitude = 0;
    private ArrayList<TourLocation> locations;
    private int locationNumber;
protected void onCreate(Bundle savedInstanceState) {
}
public void onCoordinateClick(View view) {
}
public void onPhotoClickEdit(View view) {
}
private void onChangeImage() {
}
protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
}
public void onStartEditLocation(View view){
}
}
```

3.5.4.2.7 ViewWalksActivity.java

```
public class ViewWalksActivity extends Activity {
    public int editLoc;
    ListView listView;
    private Tour tour =new Tour(null,null,null);
    private ArrayList<TourLocation>locations;
protected void onCreate(Bundle savedInstanceState) {
}
private void setupActionBar() {
}
public boolean onCreateOptionsMenu(Menu menu) {
}
public boolean onOptionsItemSelected(MenuItem item) {
```

```
}
}
```

3.5.4.2.8 JSON.java

```
public class JSON {
/**
*
* Prevents illegal characters from being sent in the Json Post.
* it adds a \ in front of illegal characters so that they do not
* get misinterpreted.
* @param string to be escaped for illegal characters.
* @return A corrected Json String, ready to be Posted.
*/
public static String quote(String string) {
}
```

3.5.4.2.9 Post.java

```
public class Post {
    private String urlString;
    private String postBody;
public Post(String urlString, String postBody) {
}
/**
* Attempts to Post a Json string via HTTP.
*
* @return a boolean value, to confirm whether the Post was
* successful
*/
public boolean send() {
}
/**
* Runs the send method, and if it fails,
* it attempts to save the tour locally on the device.
*/
public void sendAsync() {
}
```

3.5.4.2.10 Image.java

```
public class Image {
/**
```

```
* This method converts the Bitmap image taken from the camera for a location
* and converts it into a base64 String. This String will be send to the
database for storage
* via Json Post.
* @param image The image to be converted
* @return The base64 Converted image String
*/
public static String base64(Bitmap image){
}
```

3.5.4.2.11 FileManager.java

```
/**
* This class manages the local saving and loading of tours to the Android
* device.
* This is done so that in the event of a connection failure, the tours can be
* uploaded at a later
* date.
*/


public class FileManager {
/**
* This method serializes an array of tours to a .ser file. The file is stored
locally on the
* device running the application.
* If the file already exists on the device, then the file is ammended and the
new array of tours
* are saved with the old array of tours.
*
* @param tour The array of tours we wish to save locally.
*/
public void writeToFile(Tour[] tour) {
}
/**
* This method deserializes the tours stored in the file and turns it back
into an array of tours.
* It then removes the written file. The array of Tours is returned so that it
can then be converted
* into the JSON String to send to the Web-Server Database.
*
* @return The array of tours that has been deserialized from local storage.
*/
```

```
public Tour[] readFromFile() {

}

public void remove() {

}

/**

* This is the method used if an attempt to write on a file that already
exists takes place.

* In order to prevent the file from being overwritten, it is read back into
an array,

* and is combined with with the new array we are trying to save.

* The new, updated array is then written back into the file.

*

* @param tour This is the new array that must be combined with the original
array already stored in file, before being saved.

*/

public void append(Tour[] tour) {

}
```

3.5.4.2.12 Tour.java

```
public class Tour implements Serializable {

      private static String TOUR_JSON;

      private String name;

      private String shortDescription;

      private String longDescription;

      private ArrayList<TourLocation> locations;

      private ArrayList<Double> latitudes;

      private ArrayList<Double> longitudes;


/**

* The constructor for a Tour.

* It has a name, a short Description and a Long Description.

*

* @param name The name of the Tour.

* @param shortDescription A short description that describes the route taken.

* @param longDescription A description that can be used to describe the route
taken in more detail.

*/

public Tour(String name, String shortDescription, String longDescription) {

}

public void addLocation(TourLocation location) {

}
```

```
public void addWaypoint(double lat, double _long) {

}

/**

*Attempts to convert a Tour into a Json string, so that it may be sent to the
database via HTTP Post.

* @return a Json String that holds the information of a single tour.

*/

public String toJSON() {

}

public String getName() {

}

public String getShortDescription() {

}

public String getLongDescription() {

}

public ArrayList<TourLocation> getLocations() {

}

}
```

3.5.4.2.13 TourLocation.java

```
public class TourLocation {

      private static String LOCATION_JSON;

      private String name;

      private String description;

      private String image;

      private String base64Bitmap;

      private double latitude;

      private double longitude;

      private double time;

/**

* This is the constructor for a location in a Tour.

* @param name The name of a location. (E.g. "The Pug's tail")

* @param description The Descripion of a location. (E.g. "The local watering
whole for the crimnal puppy masterminds.")

* @param imagePath The image filepath of a picture taken at the location.
(E.g. " C:\Users\Benson\My Pictures\woof.bark")

* @param latitude The latitude coordinate of a location.

* @param longitude The longitude coordinate of a location.

* @param time The date/Time of when a location was recorded into the phone.

*/
```

```
public TourLocation(String name, String description, String imagePath, String
base64Bitmap, double latitude, double longitude, double time) {
}


/**
*Attempts to convert a location into a Json string, so that it may be sent to
the database via HTTP Post.
* @return a Json String that holds the information of a single location.
*/
public String toJSON() {
}
public String getName() {
}
public String getDescription() {
}
public String getImage() {
}
public double getLatitude() {
}
public double getLongitude() {
}
public double getTime() {
}
public void setName(String name) {
}
 public void setDescription(String description) {
}
public void setImage(String image) {
}
}
```

3.5.5 Detailed Design

3.5.5.1 Sequence diagram

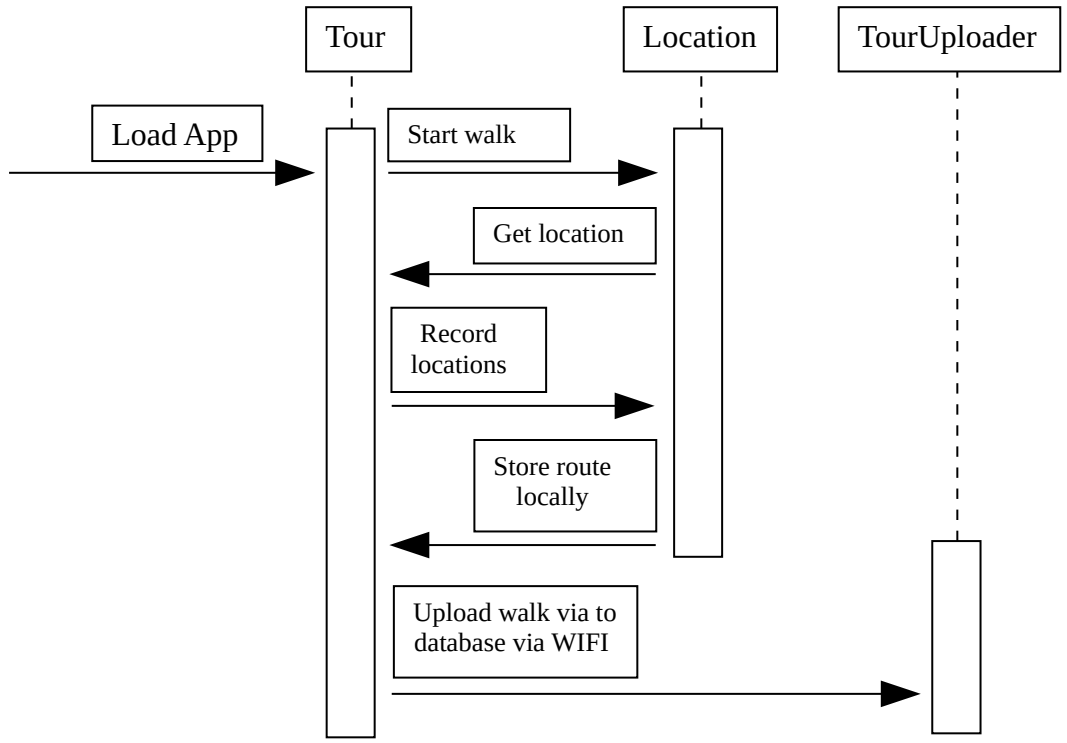

Figure 22: Sequence diagram

3.5.5.2 Significant algorithms

The first significant algorithm to be concerned at for the backend of the map would be that of the marker positions, marker positions are projected using stored lat/long positions for points based on the path selected by the user.

$res = mysql_query("SELECT * FROM points where `pathID` = $path"); while($a = mysql_fetch_array($res)) {var marker = new google.maps.Marker}

This snippet of code outlines the selection process for markers on the map. At the top the markers are selected based upon the path they belong to, and then for every marker meeting this criteria, it is based onto the map based on it's latitude/longitude values along with the description of the point stored within the information window.

var flightPath = new google.maps.Polyline({path: flightPlanCoordinates,strokeColor: '#FF0000',strokeOpacity: 1.0,strokeWeight: 2});

The google maps API comes packed with a tool called "Polyline", this will allow the web page to sketch it's own paths around the map, in which the path of the Polyline will be drawn from an array of lat/long values under the "path" method of the Polyline class.

The methods for obtaining the array to give to the polyline involves a for loop that counts from 0 to 7 (8 values) and for each loop, will push the new lat/long value to the array.

Present on the page for the map will now be buttons that will cast POST calls to the same header page, altering the values used and will show different routes between points and different paths as and when called.

This is however the extent of the significant algorithms to be used in the maps functionality.

3.5.5.3 Significant data structures

Significant data structure used is a schema made in JSON.

```
{
    "type": "object",
    "properies": {
        "name": {
            "type": "string",
            "description": "The name of the tour"
        },
        "short-description": {
            "type": "string",
            "description": "The short description of the tour"
        }
        "long-description": {
            "type": "string"
            "description": "The longer description of the tour"
        },
        "locations": {
            "type": "array",
            "description": "An array of all the locations in the tour",
            "items": {
                "type": "object",
                "description": "A single location",
                "properies": {
                    "name": {
                        "type": "string",
                        "description": "The name of the location to be displayed to the user"
                    },
                    "latitude": {
                        "type": "number",
                        "description": "Latitude in degrees"
                    },
                    "longitude": {
                        "type": "number",
                        "description": "Longitude in degrees"
                    },
                    "time": {
                        "type" "number",
                        "description": "Unix timestamp of the time the location was recorded"
                    },
                    "image": {
                        "type": "string",
                        "description": "Base64 encoded JPEG image"
                    }
                },
                "required": [
                    "name",
                    "latitude",
                    "longitude",
                    "time",
                    "image"
                 ]
            }
        }
    },
    "required": ["name", "short-description", "locations"]
}
```

This is the schema that we will be using. Essentially, Tours that we create in the android application are made in Java, which is converted into a Json post. This is done so the application must adhere to provide the correct data each time it sends information to the database for storage.

If the application tries to send a Post to the database, and the data sent does not conform to the JSON schema, it will fail to validate, and an error will be displayed.

The JSON Post can be interpreted in PHP due to the fact that it is language-independent. This means that we do not have to make any difficult conversions to the data that is sent back and forth between the Database and the Java application.

json_decode() can be used to convert JSON strings into PHP variables very easily, and this can then be stored in the database without much fuss. The function **json_encode()** can then be used to send information back to the Handheld device.

3.5.5.4 Entity Relationship Diagram

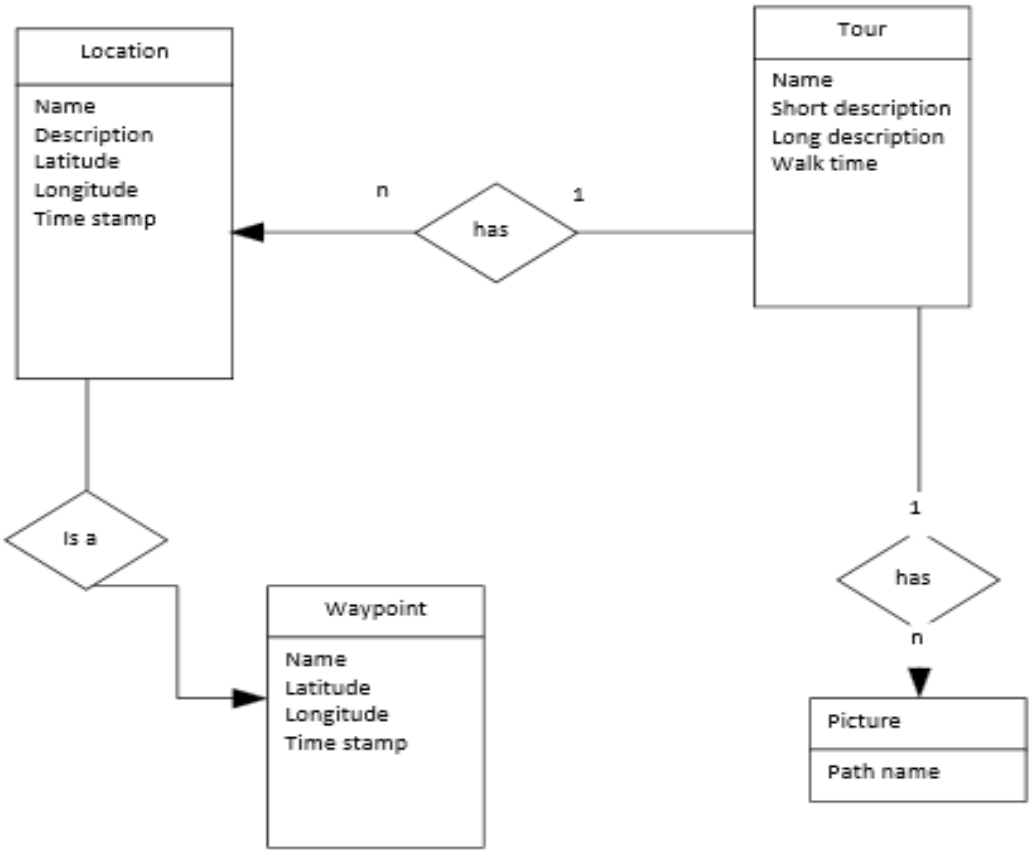

Figure 23: Entity Relationship diagram

3.5.5.5 Class Diagram

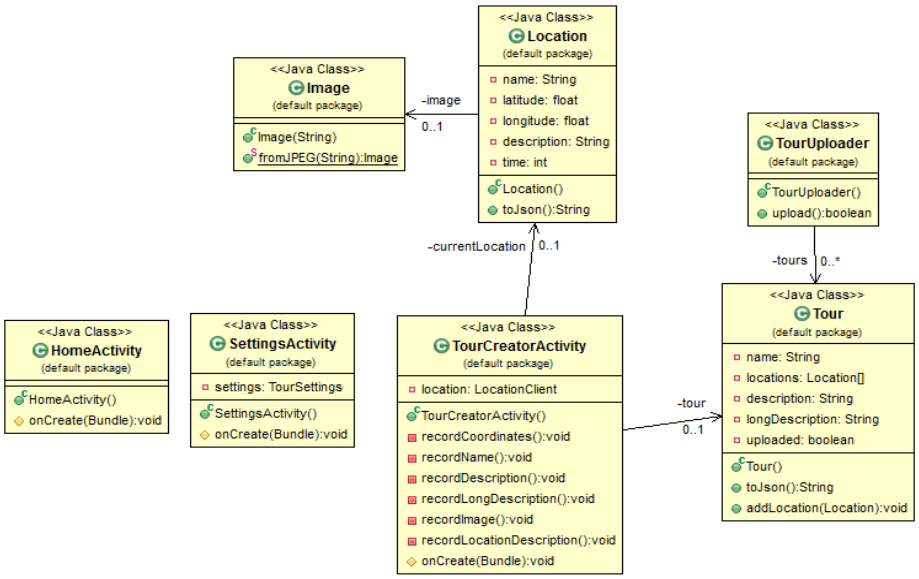

Figure 24: Class diagram

The above class diagram has been updated since finishing the whole of the Android code but due to a problem with Eclipse it could not be generated.

3.5.6 References

[1] Software Engineering Group Projects. Design Specification Standards. C. J. Price, N.W.Hardy and B.P.Tiddeman. SE.QA.05A. 1.7. Release.

3.5.6 Appendices

3.5.7 Revised Design Specification Document History

Version	CCF No.	Date	Changes made to the document	Changed by
0.1	N/A	25/11/13	Initial creation.	LPS1
0.2	N/A	01/12/13	Introduction added	LPS1
0.3	N/A	01/12/13	Mapping requirements to classes section added	LPS1
0.4	N/A	04/12/13	Added significant classes for web, programs in system and decomposition description	LPS1

0.5	N/A	04/12/13	Added Significant classes for Android	LPS1
0.6	N/A	05/12/13	Interface descriptions for Android added, squence diagram added, significant data structures added, added algorithms for web.	LPS1
0.7	N/A	06/12/13	Document revised & vetted.	LPS1
1	N/A	28/01/14	Major formatting issues corrected.	LPS1.
1.1	N/A	17/02/14	Revised and updated the document. Each section updated based on the requirements.	FIK

4. References

[1] Software Engineering Group Projects – Walking Tour Creator Requirements Specification 1.6 (Release), C. J. Price and B.P.Tiddeman SE.QA.RS (29th January 2014)

[2] Software Engineering Group Projects – QA Plan / 1.8 (Release), C. J. Price and B.P.Tiddeman SE.QA.01 (18th September 2013)

[3] Software Engineering Group Projects – Project Management / 1.8 (Release), C. J. Price SE.QA.02 (19th September 2013)

[4] Software Engineering Group Projects – General Document Standards / 1.6 (Release), C. J. Price, N.W.Hardy & B.P.Tiddeman SE.QA.03 (22nd September 2013)

[5] Software Engineering Group Projects – Design Specification Standards for SE&Comp Sci projects/1.7 (Release), C. J. Price, N.W.Hardy and B.P.Tiddeman SE.QA.05A (29th September 2013)

[6] Software Engineering Group Projects – Project Plan Specification Standards v1.2 (Release), B. P. Tiddeman SE.QA.05b (23rd September 2013)

[7] Software Engineering Group Projects – Test Procedure Standards/1.7 (Release), C. J. Price, N.W.Hardy and B.P.Tiddeman SE.QA.06 (24th September 2013)

[8] Software Engineering Group Projects – Review Standards/1.6 (Release), C. J. Price, N.W.Hardy and B.P.Tiddeman SE.QA.07 (26th September 2013)

[9] Software Engineering Group Projects – Operating Procedures and Configuration Standards/1.81 (Release), C. J. Price SE.QA.08 (31st October 2013)

[10] Software Engineering Group Projects – Java Coding Standards/1.7 (Release), C. J. Price, A. McManus SE.QA.09 (29th September 2013)

[11] Software Engineering Group Projects – Producing A Final Report/1.7 (Release), C. J. Price, N.W. Hardy and B.P.Tiddeman SE.QA.11 (11th February 2013)

Document History

Version	**CCF No.**	**Date**	**Changes made to the document**	**Changed by**
0.1	N/A	16/02/14	N/A Original draft.	lps1
1	N/A	17/02/14	Inserted relative sections as produced by the group.	fik
1.1	N/A	17/02/14	Inserted revised project plan and design specification.	fik
1.2	N/A	17/02/14	Added references, updated section numbers, updated the design, cosmetic changes, numbered figures and other other minor improvements.	fik