

Software Engineering Group Project

Final Report

Author: Group 14
Config. Ref.: SE-14-FINAL
Date: 2014-02-16
Version: 1.0
Status: Release

Department of Computer Science,
Aberystwyth University,
Aberystwyth,
Ceredigion, SY23 3DB,
U.K.

©Aberystwyth University 2014

CONTENTS

1	INTRODUCTION	3
1.1	Purpose of this Document	3
1.2	Scope	3
1.3	Objectives	3
2	MANAGEMENT SUMMARY	4
3	HISTORICAL ACCOUNT OF THE PROJECT	5
4	FINAL STATE OF THE PROJECT	7
4.1	Android App	7
4.1.1	Features	7
4.1.2	User Interface	7
4.1.3	GPS	7
4.1.4	Points Of Interest	7
4.1.5	Server Interaction	7
4.2	Website	8
4.2.1	Features	8
4.2.2	Walk List	8
4.2.3	Walk Viewer	8
4.3	Documentation	8
5	TEAM MEMBER PERFORMANCE	9
5.1	Josh Tumath (jmt14) - Project Leader	9
5.2	Theo Taylor (tht5) - Deputy Project Leader	9
5.3	Lars Lunde (lah25) - QA Manager	10
5.4	Jake Maguire (jam64) - Deputy QA Manager	10
5.5	James Berry (jab73)	11
5.6	Rob Bolton (rab26)	11
5.7	James Mellors (jam66)	12
5.8	Michael Oddie (meo9)	12
5.9	Dan Wakefield (daw46)	13
6	CRITICAL EVALUATION	14
7	APPENDIX A: PROJECT TEST REPORT	16
8	APPENDIX B: MAINTENANCE MANUAL	17
8.1	Walking Tour Creator	17
8.1.1	Program Description	17
8.1.2	Program Structure	17
8.1.3	Algorithms	17
8.1.4	Main data areas	17

8.1.5	Files	17
8.1.6	Interfaces	17
8.1.7	Suggestions for improvements	18
8.1.8	Things to watch for when making changes	18
8.1.9	Physical limitations of the program	18
8.1.10	Rebuilding and Testing	19
8.2	Walking Tour Displayer	20
8.2.1	Program description	20
8.2.2	Program structure	20
8.2.3	Algorithms	20
8.2.4	Main data areas	20
8.2.5	Files	20
8.2.6	Interfaces	20
8.2.7	Suggestions for improvements	21
8.2.8	Things to watch for when making changes	21
8.2.9	Physical limitations	22
8.2.10	Rebuilding and Testing	22
9	APPENDIX C: REVISED PROJECT PLAN	23
10	APPENDIX D: REVISED DESIGN SPECIFICATION	24
	REFERENCES	25
	DOCUMENT HISTORY	25

1 INTRODUCTION

1.1 Purpose of this Document

1.2 Scope

1.3 Objectives

2 MANAGEMENT SUMMARY

The group achieved a working version of the Walking Tour Creator (WTC) that is capable of providing the user the means to record a walk and add a description about the walk, add points of interests at anypoint of the walk, also to be able to take pictures and have them saves and upload it to a database for viewing, in other words a fully functional personal Walking Tour Creator; that the user can use to create there own walks.

The group was able to varify that documents were in good state after the drafts had been reviewed. The majority of our documents got a grade of B and only one with a grade of C for the draft. The document with the C grade was in a bad state so what the group did was go and fix the issues with it to get the grade up. This doesn't mean that we left the other document they all had some minor issues with them so we delegated the fixes though out the team to get all of the documents into a good state.

The group had minor issues with collabouration and version control. This was mainly due to group members struggling to use GitHub at the beginning of the project, so it was difficult to get all of the documents together. The way the group overcome these problems is that Dan Wakefield wrote a tutorial and put it on the repository. We also switched to using the free online L^AT_EX editor "writeL^AT_EX" for producing documentation, as it allowed for easy collabouration (but consiquently discouraged frequent commits to the repository).

When developing the Android app, there was a major bug where - if the user changed the orientation of their phone during an opperation, the app would lose the asynchronous task, causing the task to attempt to open popups on an Activity that no longer existed. To overcome this, the Android sub-team passed the AsyncTask to a newly opened Activity each time this happened and updated the AsyncTask for the new Activity.

When developing the Web site, one of the problems that the Web sub-team had was deciding how to store in a database whether a coordinate was a point-of-interest or just an ordinary coordinate along the route. They overcame this problem by meeting together to work on it. They decided to add another column in the database which was a simple boolean value to represent whether a coordinate was a point-of-interest.

On a whole, the entire team performed with a hard-working attitude towards the goals of the project. While there were some issues with some group members not pulling their weight initially, but - by the end of the project - each member displayed a passion and enthusiasm towards their work. Communication between the group was good; particularly during integration and testing week, where everyone showed a team spirit.

3 HISTORICAL ACCOUNT OF THE PROJECT

At the beginning of the group work, the group all had to decide on major roles - namely Project Leader (Josh Tumath), Deputy Project Leader (Theo Taylor), QA Manager (Lars Lunde), Deputy QA Manager (Jake Maguire) and Chief Architect (Rob Bolton). Later during the prototyping phase of the project, the group was split into two development teams: the Android team and the Web team. These sub-teams were led by Rob Bolton and Jake Maguire, respectively.

The group met during Fridays for general meetings with the Project Manager. Additionally, there would be mid-week meetings - typically on Wednesdays - where the group would normally discuss the progress of work to be completed by the following Friday meeting. Facebook was the preferred platform for communication.

The first task the group had was to write the Project Plan document. This involved outlining what is to be produced and the architecture of both the app and website. This gave the group a first attempt at working together to produce something. Each section would be given to different group members to work on. If a section required more work compared to others, more group members would be assigned to it.

Following this, the group was tasked with two documents to produce simultaneously. These were the Test Specification and the Design Specification. During this time, many members of the group had other assignments. Fortunately, The Test Specification did not require too much time to write. Therefore, only Dan Wakeeld was asked to write it. The Deputy Project Leader led the creation of the Design specification with James Mellors and Michel Oddie. They delegated sections of the Design specification to each other. James Mellors was given the task to do the dependency description; Theo Taylor had the tasks to do the introduction, decomposition description and the interface description; and Michel Oddie had the task to do the detailed design.

Before the submission of each of these documents, a formal review meeting would be held by the Project Leader and QA Manager. Following this, issues would be created on GitHub listing the necessary changes.

During the creation of the Design Specification, the members of the group who were not working on it were simultaneously producing the prototypes for Android application, the front-end of the Web site and the database. This ensured every member of the group had work while also allowing for efficiency in the rate at which work was completed. By Christmas, both the Android sub-team and Web site sub-team were able to give a good demonstration of the front-ends of their software to the Project Manager.

Following the Christmas break, integration and testing week itself was a very busy week because of the great amount of work that needed to be completed in time for the Acceptance Test with the Project Manager at Friday afternoon. The way the

group tackled it was to begin with a meeting together to talk about how far along we are and what tasks need to be done. The group made a plan of what needed to be done for each day. After the meeting, the group looked for a suitable workspace to allow them to work together.

Once the group was set up, they separated into the two sub-teams; as was the case during the prototyping stage. The sub-team leaders then delegated all the tasks that needed to be completed for the day. This process of meetings and task delegation was repeated each day of integration and testing week and the group found it to be a very efficient way of handling the work load.

Towards the end of the week, the app and Web site were falling together. This meant that some group members had no tasks assigned to them and therefore worked on revisions of previously submitted documentation to be efficient with their time.

During the Acceptance Test, the Project Manager was pleased that the majority of the produced software was fully functional.

4 FINAL STATE OF THE PROJECT

4.1 Android App

4.1.1 Features

The features of the app were mostly correct. The only feature that was missing was the ability to edit previous Points Of Interest. Most features were implemented correctly, except that the app is meant to tell you if you try to take a Point Of Interest without any GPS signal, and prevent a Point Of Interest from being created. Instead, our app uses the user's last known location for the new Point Of Interest. GPS, the UI, and server interaction all work correctly.

4.1.2 User Interface

The UI works correctly. The user cannot enter any invalid data and are told what is wrong when they attempt to do so. The user cannot navigate to displays which would not make sense e.g. returning to the "Create a new walk" Activity whilst a current walk is being created. The UI can properly handle the user pressing the home button or rotating the screen on certain windows. These are any which require input before closing, and cannot be dismissed. Any windows which can be dismissed will not return upon the user opening the app again.

4.1.3 GPS

The GPS collects data correctly and filters any points which are not accurate enough. The only issue with the GPS is that it will automatically use the last known position if no new data is available, after warning the user once that they need GPS data.

4.1.4 Points Of Interest

The only issue with the Points Of Interest is that they were meant to have "one or more" images, but we allowed the user to have no images as well. The user can take as many pictures as they want, and when adding pictures to a Point Of Interest, they can look through thumbnails of pictures they've already taken. Tapping on a picture deletes it. This was our interpretation of editing a Point Of Interest.

4.1.5 Server Interaction

Aside from a memory issue, the server interaction works well. It attempts to upload the image to the server and if anything other than an http "200 OK" is received

then the user is told that their upload failed. The user can attempt to upload again if they want. This means that if anything goes wrong with sending the image then the user will be told - this includes website issues and connection issues.

4.2 Website

4.2.1 Features

All features work correctly. The only potential issue would be if two separately uploaded images had the same hash. If this were to occur then the old image would be overridden with the new one. Parsing of the uploaded JSON and saving this data in the database works correctly.

4.2.2 Walk List

The page which displays the list of all recorded walks works correctly, and they are shown in a large grid with the walk's name and a picture from it. If a walk has no image then the app/webiste logo is displayed instead.

4.2.3 Walk Viewer

The walk viewer works correctly. On the left of the page it gives the user the walk's name, short & long descriptions, the total distance covered, how long the walk took. On the right the walk is displayed using Google Maps. Walks are correctly displayed in this area, with each GPS point and Point Of Interest being linked.

4.3 Documentation

5 TEAM MEMBER PERFORMANCE

5.1 Josh Tumath (jmt14) - Project Leader

Josh was given the role of Project Leader, where he led and directed the team in preparation for the final implementation of the software. During integration and testing week, he was also a Web Developer in the Web site sub-team, due to his previous experience in Web programming and design.

It was remarked by other project members that Josh was very well-organised throughout the duration of the project, planning the team's actions well. He was generally very helpful towards others; being easy to work with and not forceful in regards to pushing for completion of work.

However, Josh could be very unclear about deadlines, failing to set targets for when particular sections of work should be completed. Additionally, he did not delegate work evenly, leaving some members to be completing more work than others.

The above report was agreed to by all members of the group.

5.2 Theo Taylor (tth5) - Deputy Project Leader

Theo was given the role of Deputy Project Leader, where he led group meetings in times when the Project Leader was unavailable. He also led ad-hoc sub-teams that worked on a specific task for the project, such as the Design Specification. He was also an Android Developer in the Android sub-team.

During integration and testing week, Theo worked on certain features in the Android app, such as disabling the back button and creating Bundles for communication between Activities. He struggled in this area due to a lack of experience with Android development.

However, overall Theo could be quite unreliable at times. Additionally, when leading members of the team, he failed to push for high quality of work - being too lenient towards slacking and meeting deadlines. For example, the Design Specification required many last-minute changes before it was handed in.

The above report was agreed to by the group member.

5.3 Lars Lunde (lah25) - QA Manager

Lars was given the role of QA Manager, where he took minutes in meetings, held formal reviews and aided the group in meeting the QA standards.

As Larss native language is not English, he was concerned this may affect some of the duties of his role. However, this was not an issue. For the most part, he followed through with his role effectively, and was willing to put in the hours for the sake of the group. During integration and testing week, he was very focused and worked closely with Rob on the development of the Android app, while always ensuring he had work to do.

The main issue with Lars was that he could be very slow at publishing meeting minutes - usually taking more than 24 hours after the meeting. Additionally, as the weeks progressed, his minutes became less detailed, which affected some members understanding of their tasks following the meetings, or reminding themselves of the items discussed during the meetings. Furthermore, he did not enforce the Java coding standards during integration and testing week.

The above report was agreed to by the group member.

5.4 Jake Maguire (jam64) - Deputy QA Manager

Jake was given the role of Deputy QA Manager, where he mainly took minutes in meetings when Lars was no able to, and prepared formal reviews when the QA Manager was too busy. During the prototyping and implementation stages of the project, he was also tasked as Lead Developer of the Web site sub-team, due to his previous experience in server-side programming. Additionally, he allowed us to use his Web server for hosting the Web site.

From the beginning of the project, Jake had been very helpful and involved. When taking the minutes for meetings, he would usually be very prompt in uploading them. He also worked hard on developing the user interface design of both the Android app and the Web site. Jake had very good attendance of meetings and in the computer rooms during implementation and testing week.

Jake had proven himself to be very efficient in leading the Web site sub-team, as the majority of their prototyping and final implementation has been completed by Christmas. However, it is possible that he took on more work himself and not enough for group member James Berry.

The above report was agreed to by the group member.

5.5 James Berry (jab73)

James was given the role of Web Developer in the Web site sub-team, where he aided in the early prototyping and implementation of both the client-side and server-side systems. This suited James's skillset, having studied Web development in previous university modules.

James was reliable in both attendance and completion of work. At the beginning of the project, he wrote the risk analysis and a schema for the Tour database, which were done to a high standard. Working with Jake Maguire, they worked efficiently to complete a prototype of the Web site before Christmas.

During integration and testing week, James put a lot of work into researching how to solve certain problems, such as decoding base64 encoded images on the server-side. Additionally, he aided with the Google Maps APIs to ensure the front-end was feature-complete. Finally, James was very helpful in completing the PHPdoc.

The above report was agreed to by the group member.

5.6 Rob Bolton (rab26)

Rob was given the role of Chief Architect, where he was expected to handle the overall design of the system from initial planning to implementation. In early meetings during the project, he expressed an interest and ability in planning class structures and imagining relationships between parts of the system. During the prototyping and implementation stages of the project, he was also tasked as Lead Developer of the Android sub-team. This required him to gain a detailed understanding in the Android SDK - building on his experience in using the Java programming language.

Rob displayed a great passion for the work of the project and a strong desire for work to be completed to a high standard. He explained that he has enjoyed the experience; particularly during integration and testing week, when he stated that he thrived from the pressures of working towards a deadline and being heavily relied upon by other members of the team. He was also very helpful and patient with other Android developers in the team when allocating and expecting work from them.

However, he was perhaps too willing to take on too much work for himself. Much of the work on prototyping for the Android app was completed by him - as was much of the coding for the completed app. One night, he and Dan Wakefield stayed awake for over 24 hours to work on the final features for the app.

The above report was agreed to by the group member.

5.7 James Mellors (jam66)

James was given the role of Android Developer in the Android sub-team, where he helped with smaller changes in the development of the Android app. However, James's more significant contributions were towards documentation efforts, where he worked on various sections of the Project Plan and Design Specification. In addition, he designed the logo of the software.

James very quickly proved himself to be reliable and prompt when the project began. Work he was given - such as gathering information on the Google Maps APIs or writing about the high-level architecture of the Android app - was completed promptly. He was always seeking or work to do.

During the former half of integration and testing week, James did some peer programming on adding certain features to the Android app, such as checking if a device supports location services. In the latter half, he worked on the UI and updating the design specification.

The above report was agreed to by the group member.

5.8 Michael Oddie (meo9)

Oddie was given the role of Android Developer in the Android sub-team, where he aided in the implementation of the Android app. However, he was initially tasked to create the initial wireframe design for the Web site and certain sections of the Design Specification. During integration and testing week, he worked mainly on testing and documentation.

Initially, Oddie's meeting attendance and contribution effort could be quite poor. Part of this was due to society events clashing with the group's unofficial weekly meetings. Other times, however, work was simply not being completed or completed late - namely specific sections of the Project Plan he was tasked on completing. This improved over time, and, by the latter half of Semester 1, he was contributing much more.

Oddie performed greatly during integration and testing week, when he was very enthusiastic about working on the Android app and testing it very extensively. This showed he thrives much better in a constant working environment. He was a great value to the team, working with other members of the group to help them and ensure that they were producing a good quality of work.

The above report was agreed to by the group member.

5.9 Dan Wakefield (daw46)

Dan was given the role of Version Control Manager, where he acted as a consultant for the group when using the project repository on GitHub. This required him to attend a lecture on the subject and maintain the health of the tree during development. He also aided in the development of both the Android app and the Web site.

Though Dan lacked access to an internet connection off-campus during Semester 1, he still put in a great deal of work and contribution towards the project. Over the course of the project, Dans experience with Git improved greatly, and his help was invaluable by integration and testing week.

He worked very hard on the project. Most notably, he devoted a lot of time to helping Jake Maguire and James Berry to fix a critical bug in the communication between the Android app and Web server. At one point, he stayed awake overnight with Rob Bolton trying to refactor the software when the completion of some features was behind schedule.

The above report was agreed to by the group member.

6 CRITICAL EVALUATION

In terms of final product, the whole team performed efficiently and competently. The project was finished on time with nearly all the functionality that was specified at the start. Our initial designs were thorough to the point where the implementation of the project did not encounter errors or big changes from what we decided early on.

Each member played their part in the project with everyone coming together at the end to get everything done. The Project Leader made sure that progress was ticking along at an appropriate place, that all team members had a job to do and that attainable deadlines were set to ensure the project would be finished on time. Everyone pulled their weight and were committed to getting their part done for the rest of the team.

There were a few issues with delegation with those in various leadership roles preferring to tackle most things on their own rather than delegating tasks to other member on their team. With more organised delegating the project may have been completed quicker and a rush at the end to get everything completed would have been avoided.

Communication could have been better in terms of responding promptly to questions and tasks on online platforms such as GitHub and Facebook. It could be hard getting a response from all members quickly when a question was asked or tasks assigned on GitHub would be left open even if completed.

Improvements could have been made during our project lifecycle. Git was an under-utilised tool by some members, especially by the Web team. Due to a lack of knowledge, complications with merges and branches were apparent during implementation. This led to Git being side-lined and meant features such as rolling back to previous versions and allowing multiple members to work on the same file could not be used effectively.

Our project also had some missing functionality. The ability to edit locations on a walk was not present, which was reflected in our acceptance testing. This was more a oversight from the team rather than a technical issue, but meant that our Android app did not do everything that was asked by the Requirements Specification.

The team learned about the importance of communication and how critical it is to keep in contact to track progress and know what to do. The minutes had to be taken with detail so that members who were unable to attend the meeting still knew what task they would have to complete that week. Both the Web sub-team and Android sub-team leaders had to communicate to make sure that things were set up at their own ends to allow tests to take place between the two platforms, to check they were interacting with each other correctly. The Project Leader had to communicate tasks to each member giving them specific targets of deadlines and the content of their

work. Feedback from members would also have to be given to the Project Leader in terms of whether or not these targets were attainable or if they had disagreements with what had been set.

Another lesson learned was the equal importance of all stages of the project life cycle. Our emphasis on a proficient design meant that our implementation ran into little problem and allowed us to finish on time. The analysis we carried out early on meant that most functionality was in place as we knew what was required from our end programs, making our project a viable solution to the specification given to us at the start.

7 APPENDIX A: PROJECT TEST REPORT

8 APPENDIX B: MAINTENANCE MANUAL

8.1 Walking Tour Creator

8.1.1 Program Description

This application allows a user to record a walk, tracking their GPS coordinates and allowing them to add named points of interest with pictures. It can upload this walk to a server once it is finished.

8.1.2 Program Structure

The component diagram can be found in *Design Specification 3.1.1* The sequence diagram can be found in *Design Specification 5.1* A list and definition of the classes & interfaces and their methods used can be found in the Design Specification:

- **MainAppActivity** 4.1

8.1.3 Algorithms

The significant algorithms can be found in *Design Specification 5.2*

8.1.4 Main data areas

The main data areas can be found in *Design Specification 5.3*

8.1.5 Files

Our application does not use any specific files. The only files it relies on being there are the photos taken for the Points Of Interest. The path returned by the camera intent after taking the photo is used for these. If the application cannot find the file, then the photo will not be added to the JSON before sending it to the server.

8.1.6 Interfaces

The only requirements for the application to successfully function are that the user has a camera, internet connection, and GPS running. If the user does not have a camera then they cannot install the application. If the user does not have GPS running then the application will not let them create a new walk. Turning this off whilst the application is running will prevent the application from gaining any

more GPS data, and it will continue to use the last known location for new Points Of Interest. If the user does not have an internet connection then saving and uploading the walk to the server will fail.

8.1.7 Suggestions for improvements

Several improvements could be made to the application. At the moment there is no way to edit points of interest after they have been created. In order to fix this, I would recommend having a new activity using a list view and an adapter, which takes the Walk in its bundle to see which points of interest we have.

When the methods converting the walk to JSON were created they were made as a quick solution to upload the walk. As a result of this they attempt to compile the whole walk in one go. This means that the entire JSON object of the walk is in memory. Two solutions which would help are downscaling the image before converting it to base 64, and finishing the code which converts and sends the JSON one image at a time in a stream. Realistically, both of these should be completed for good practice, as we do not need high resolution images on the website, and storing the whole JSON object in memory can cause phones with small amounts of memory to crash.

8.1.8 Things to watch for when making changes

There are a couple of places in which you need to take care when changing. Anything dealing with an AsyncTask needs to ensure that if it displays any dialogs that the context it originally had is still available for use. An example of doing this correctly is where we call "setDialogsAndNotify" on our walkUploader in order to give it the new dialogs for the current Activity. Another area in which it is important to take care when changing is the JsonPackager. Any changes to the format of the JSON in here need to be reflected in the website's code which deals with the received JSON.

8.1.9 Physical limitations of the program

There are a few physical limitations of the program. In order to be able to install it, the user's device needs to have a camera. If the user does not have GPS when attempting to create a new walk then they will be not be able to create one. If the user does not have enough memory to store all of the images they took, then the application will crash upon compiling the JSON representation of the walk. Another physical limitation is that the application will not accept any GPS data which is too inaccurate. This uses the Location.getAccuracy() method and filters out any results which are greater than 15 meters. This method returns *accuracy* in meters, which is a circle of radius *accuracy* which has a 68% probability that the device

was within the bounds of this circle. Due to this, if the device never receives any Location objects which are at least within 15 meters accurate then no GPS data will be recorded. The last limitation is that the user needs an internet connection to upload the walk. The application will keep the walk until the user successfully uploads it or cancels it.

8.1.10 Rebuilding and Testing

All of the documentation is either in LaTeX or Javadoc. Any pdftoLaTeX compiler should work for rebuilding any of the LaTeX documentation. The Javadoc can be rebuilt using Eclipse by selecting *File > Export... > Java > Javadoc* and following through with the wizard. Another alternative is to use the "javadoc" command.

8.2 Walking Tour Displayer

8.2.1 Program description

This Web site receives tours from the Walking Tour Creator and stores the data within a MySQL database. Web site visitors can view a list of the tours and view each of them on a detailed map that displays the route with points-of-interest along it.

8.2.2 Program structure

See Design Specification.[2]

TODO: which sections of the design spec?

8.2.3 Algorithms

See Design Specification.[2]

TODO: which sections of the design spec?

8.2.4 Main data areas

TODO: This specifies the data structures, including arrays, objects etc. where important information is stored for a substantial part of the main program. For example, in a program that adds a students marks together and calculates a grade, there might be data structures used to store a students project and examination marks for each course. Again, if this information is contained in the design specification, it can just be referenced here.

8.2.5 Files

TODO: It may be that the program accesses certain fixed files or needs files of a certain type to be available. Give such information here. For example, The program creates the file XYZ.test as workspace and later deletes it. If such a file exists already then its contents will be lost. Another example is The program assumes that the current directory contains a file of integers at three per line, separated by spaces.

8.2.6 Interfaces

TODO: Many programs control or read devices such as measuring instruments. Usually there will be certain protocols to be observed, requirements that a terminal

is set up in a particular way, etc. For example, The terminal should be set to read and transmit at a baud rate of at most 1200. The possibilities here are endless, but each application is likely to have a few simple rules that must be observed, and such information should be given in this section.

8.2.7 Suggestions for improvements

You could entertain the idea of adding a search bar to the website. This would be particularly useful when the website is populated with a large amount of walks. You could add a search bar where the user can search for a particular name of a walk that they wish to view. An SQL statement would then be used to search through the database and return the results of the search. Ajax could be used to create a more dynamic webpage, where the results for the search get returned while the user is typing in their search. Other search features could be to search walks by the distance or by the time they take to complete.

Another feature that could be added is to print out the directions for the walk next to the map. I imagine that there is probably a Google API which can be used to do this. Which will check all of the coordinates and guide the user through their walk, giving them directions for the turnings that they need to take.

A pagination should also be considered where large amounts of data are being used so that the user would not have to endlessly scroll through all of the walks, but rather go through pages with a certain amount of walks per page. This could be implemented using a PHP script which takes in all of the walks from the database and then divides them all by a certain number, say 9, and splits all of the walks into several pages of 9 walks.

A final feature that should be added to the website is to have a clear start and a finish of the walk. At the moment the map only displays all of the coordinates and locations of a walk, but it is not clear where the walks start and end. The only clue is that the centre of the map is set to the first coordinate. However this can cause confusion, particularly if the walk is a giant loop which starts and finishes in the same place. To overcome this a marker, similar to those used for a location but different in some way, should be used to show where the first coordinate is and the last coordinate is.

8.2.8 Things to watch for when making changes

It should be noted that if you wish to change any of the SQL querying, we have used a PDO abstraction layer to implement these, rather than using the MySQLi APIs.

8.2.9 Physical limitations

As our Website has been written on a personal server, it should be noted that if it is intended for this website to handle a greater amount of traffic then it would be necessary to move to a new host server, as the current server only allows for a certain amount. There is also a limit to the disk space available, so if there is expected to be a large amount of data stored on the server, particularly with image data, then extra disk space will most likely be needed to stop this from becoming a problem.

8.2.10 Rebuilding and Testing

As this has been written on a personal server, the files cannot be easily viewed. However they are all available at "www.jakemaguire.co.uk/projects/wtc".

In terms of testing the website, the best thing to do would be to run the app and put some dummy data in, including some locations with pictures to see that they write correctly to the database and then check the website homepage to see that it then displayed correctly.

9 APPENDIX C: REVISED PROJECT PLAN

10 APPENDIX D: REVISED DESIGN SPECIFICATION

REFERENCES

- [1] *Software Engineering Group Projects* General Documentation Standards. C. J. Price, N. W. Hardy, SE.QA.03. 1.5 Release.
- [2] *Software Engineering Group Projects* Design Specification. Group 14, SE.14.DESIGN. 1.2 Release.

DOCUMENT HISTORY

Version	CCF No.	Date	Changes made to Document	Changed by
0.1	N/A	2014-02-09	Initial creation with performance reports	jmt14
0.2	N/A	2014-02-14	Management summary draft	tht5
0.3	N/A	2014-02-14	Critical evaluation	jam64
0.4	N/A	2014-02-14	Final state of the project	rab26
0.5	N/A	2014-02-14	Historical account of the project	jam66
0.6	N/A	2014-02-14	Maintenance manual initial draft	rab26, jam66
0.7	N/A	2014-02-14	Minor changes to language to meet QA	jmt14