Swansea University
Prifysgol Abertawe

Swansea University

MEng Computing

Group Project CS-M04

Team Structure, Methodology, Requirements and Specifications

Authors:
Adam Barrell *(632975)*
Thomas Milner *(xxxxxx)*
Lewis Hancock *(xxxxxx)*
Christopher Lewis *(xxxxxx)*

Supervisor:
Parisa Eslambolchilar

January 20, 2014

Contents

1 Introduction

1.1 Client Background

1.2 Project Overview

1.3 Term Definitions

Term	Definition
Some Term	Some definition.

2 Team Structure

2.1 Roles and Responsibilities

2.2 Communication Strategy

2.3 Development Strategy

3 General Description

3.1 Product Perspective

3.2 Product Functions

3.3 User Characteristics

3.4 Constraints

3.5 Assumptions and Dependencies

4 Requirements

4.1 Functional

4.1.1 Web Portal

Code	Requirement
WEBREQ1	The web portal should allow users to register a new account.
WEBREQ2	The web portal should allow registered users to log in.
WEBREQ3	The web portal should allow registered users to log out.
WEBREQ4	The web portal should allow registered users to modify their account details.
WEBREQ5	The web portal interface should feature distinctive White Rock branding.
WEBREQ6	The web portal should allow registered users to add sub-branding to created walks.
WEBREQ7	The web portal should display a registered user?s own walks and contributions when logged in.
WEBREQ8	The web portal should allow registered users to create, edit and delete their own walks.
WEBREQ9	The web portal should allow registered users to create, edit and delete their own waypoints.
WEBREQ10	The web portal should allow registered users to add GPS waypoints to a walk.
WEBREQ11	The web portal should allow registered users to modify the order each waypoint should be visited.
WEBREQ12	The web portal should allow registered users to upload, edit and delete waypoint data such as images, text, audio and video.
WEBREQ13	The web portal should allow registered users to provide English and Welsh translation for text.
WEBREQ14	The web portal should display download statistics for each of the user?s walks.
WEBREQ15	The web portal should display reviews for each of the user?s walks.

WEBREQ16	The web portal should display an average rating for each of the user?s walks.
WEBREQ17	The web portal should notify registered users of waypoint addition requests.
WEBREQ18	The web portal should allow registered users to accept or reject requested waypoint additions.
WEBREQ19	The web portal should allow registered users to report bugs and errors.
WEBREQ20	The web portal should contain an FAQ and user guide.

4.1.2 Android Application

Code	**Requirement**
APPREQ1	The application should allow users to register user accounts.
APPREQ2	The application should allow registered users to log in.
APPREQ3	The application should allow registered users to log out.
APPREQ4	The application should allow registered users to modify their account details.
APPREQ5	The application should allow users to download walks created by registered users.
APPREQ6	The application should group walks by local area.
APPREQ7	The application should sort grouped walks by current proximity to the user.
APPREQ8	The application should allow users to use Google earth and street maps interchangeably while viewing a walk.
APPREQ9	The application should allow registered users to choose a default map view for a created walk.
APPREQ10	The application should download data from a remote database.
APPREQ11	The application should synchronise data whilst the host device is connected to WiFi.
APPREQ12	The application should give users the option to synchronise data over cellular networks.
APPREQ13	The application should give users the option to store data on an SD card if available.
APPREQ14	The application should detect a user?s GPS location consistently between devices.
APPREQ15	The application should display the prefered order to visit waypoints in a walk.
APPREQ16	The application should display waypoint information even if it has already been visited.
APPREQ17	The application should allow registered users to create, edit and delete their own walks.
APPREQ18	The application should allow registered users to tag GPS waypoints in their own walks.

APPREQ19	The application should track the order GPS waypoints were tagged in a created walk.
APPREQ20	The application should allow registered users to upload audio, images and video to a waypoint.
APPREQ21	The application interface should separate audio, images, text and video waypoint data.
APPREQ22	The application interface should feature distinctive White Rock branding.
APPREQ23	The application should allow registered users to contribute waypoints to other user?s walks by request.
APPREQ24	The application should allow users to leave reviews on walks.
APPREQ25	The application should allow users to rate walks they have completed.
APPREQ26	The application should allow users to report bugs and errors.
APPREQ27	The application should provide links to the app?s Twitter and Facebook channels.
APPREQ28	The application should provide English and Welsh translations for text.
APPREQ29	The application should allow users to report inappropriate content to an administrator.

4.2 Non-Functional

Code	**Requirement**
NFREQ1	The web portal must share the same database as the second development group.
NFREQ2	The application should be intuitive to users with little technology experience.
NFREQ3	The application should be usable by colour blind and deaf users.
NFREQ4	The application code should be maintainable for future developers.
NFREQ5	The application code should be fully documented.
NFREQ6	The application should run efficiently on lower end GPS-enabled android tablets.
NFREQ7	The application should run efficiently on android smartphones.
NFREQ8	The application should be compatible with typical tablet screen sizes.
NFREQ9	The application should be compatible with typical smartphone screen sizes.
NFREQ10	The application should be battery efficient by sampling GPS locations based on the user?s mode of transport.
NFREQ11	The application should be bandwidth efficient.

4.3 Nice To Have

Code	**Requirement**

NTHREQ1	A Facebook channel to enable feedback and contributions from users.
NTHREQ2	A Twitter channel to enable feedback and contributions from users.
NTHREQ3	The application should display the latest tweets from the Twitter channel.
NTHREQ4	The application should allow users to advertise their current walk on Facebook and Twitter.

5 Specifications

5.1 Functional

5.1.1 Web Portal

Code	Requirement
WEBSPEC1	The web portal will feature an ?Edit Account? button.
WEBSPEC2	Clicking the ?Edit Account? will display a view that allows the user to edit their ?full name, email and password?, a ?cancel? button and a ?save? button.
WEBSPEC3	Clicking the ?Save? button will validate the full name, email and password text boxes. If successful, a notification will inform the user that changes were made to their details.
WEBSPEC4	Clicking the ?Cancel? button will cancel current changes made to the user?s details.
WEBSPEC5	All web portal views will feature the company logo and use a house colour scheme.
WEBSPEC6	The ?user account settings? view will feature a company branding tab, allowing the user to attach their company logo to a walk.
WEBSPEC7	The company branding tab will feature a ?Browse? button, allowing the user to browse their computer for a particular image and will feature a ?Save? button, allowing the user to upload their image.
WEBSPEC8	Successfully logging in will take the user to a ?Homepage? view featuring their own walks and contributions.
WEBSPEC9	The web portal will feature a ?Walk? button on its side bar.
WEBSPEC10	Clicking the ?Walk? button will navigate the user to a view featuring a list of walks and an ?Add walk? button.
WEBSPEC11	Clicking the ?Add walk? button will display a view featuring ?Walk name, description? textboxes, ?Cancel? and ?Create? button.
WEBSPEC12	The ?Walk name? and ?Description? text boxes will include a profanity filter, preventing inappropriate words being used.
WEBSPEC13	Clicking the ?Cancel? button will navigate the user back to the ?Homepage? view.
WEBSPEC15	Clicking the ?Create? button will create a new walk.
WEBSPEC16	The walk list view will feature an ?Edit? button and ?Delete? button adjacent to each walk.

WEBSPEC17	Clicking the ?Edit? button will navigate to a view where the user can edit the ?Walk name, description? text boxes.
WEBSPEC18	Clicking the ?Delete? button will display a view, asking the user to confirm they wish to delete the walk.
WEBSPEC19	When in user-walk list view, clicking on a walk will display a view featuring a list of walk waypoints and an ?Add waypoint? button.
WEBSPEC20	Clicking the ?Add waypoint? button will display a view featuring ?Waypoint name?, ?Waypoint description? text boxes, ?Location Map?, ?Add image?, ?Add video? and ?Add audio?.
WEBSPEC21	Clicking the ?Add image? button will display a ?Browse?, ?Add? and ?Cancel? button.
WEBSPEC22	Clicking the ?Browse? button will allow the user to browse their computer for a suitable image of the waypoint.
WEBSPEC23	Clicking the ?Save? button will upload the image to the waypoint.
WEBSPEC24	Clicking the ?Cancel? button will navigate the user back to the waypoint view.
WEBSPEC25	Clicking the ?Add video? button will display a ?Browse?, ?Add? and ?Cancel? button.
WEBSPEC26	Clicking the ?Browse? button will allow the user to browse their computer for a suitable video of the waypoint.
WEBSPEC27	Clicking the ?Save? button will upload the video to the waypoint.
WEBSPEC28	Clicking a specific point on the ?Location map? of the walk will place a waypoint based on its geological location.
WEBSPEC29	The longitude and latitude co-ordinates must be accepted in the following formats: ?British Grid?, ?Decimal degrees?, ?Degrees minutes and seconds? and ?Degrees and decimal minutes?.
WEBSPEC30	The waypoint list view will feature an ?Edit? button and a ?Delete? button adjacent to each waypoint.
WEBSPEC31	Clicking the ?Edit waypoint? button will navigate to a view where the user can edit the ?Waypoint name, description? textboxes, waypoint image, waypoint video, waypoint audio and waypoint geological location.
WEBSPEC32	Each feature of a waypoint shall include a delete button adjacent to the feature.
WEBSPEC33	Clicking the ?Delete? button of a waypoint feature will display a view asking the user if they wish to delete it. For example, clicking the delete button adjacent to the waypoint video will ask the user if they wish to delete the video.
WEBSPEC34	Clicking the ?Delete waypoint? will display a view, asking the user to confirm they wish to delete the waypoint.
WEBSPEC35	The waypoints view will feature a ?Change order? button.
WEBSPEC36	Clicking the ?Change order? button will display a list view of the waypoints, an ?Up? button and a ?Down? button.

WEBSPEC37	When a waypoint is selected, clicking the ?Up? button will move the waypoint up the list order, clicking the ?Down? button will move the waypoint down the list order.
WEBSPEC38	The web portal side bar will feature a ?Translate? button that will display a view featuring ?Welsh? and ?English? buttons.
WEBSPEC39	Clicking the ?Welsh? button will translate the text into English.
WEBSPEC40	Clicking the ?English? button will translate the text into Welsh.
WEBSPEC41	The web portal side bar will feature a ?Statistics? button, a ?Ratings? button, a ?Review? button, a ?Requests? and a ?Report? button.
WEBSPEC42	Clicking the ?Statistics? button navigates the user to a statistics view, displaying each of the user?s walks and their download history.
WEBSPEC43	Clicking the ?Ratings? button navigates the user to a ratings view, displaying each of the user?s walks and average ratings.
WEBSPEC44	Clicking the ?Reviews? button navigates the user to a review view, displaying each of the user?s walks a list of reviews.
WEBSPEC45	Clicking on the ?Request? button navigates the user to a request view, displaying pending waypoints to be added to one of their walks.
WEBSPEC46	Clicking on a pending waypoint will display a view featuring an ?Accept? or ?Reject? button.
WEBSPEC47	Clicking the ?Accept? button will add the waypoint to the walk.
WEBSPEC48	Clicking the ?Reject? button will delete the waypoint request.
WEBSPEC49	Clicking the ?Report? button will display a report view, featuring a report/bug ?Name?, ?Description? textboxes, ?Send? button and ?Cancel? button.
WEBSPEC50	Clicking the ?Send? button will file a report to the White Rock maintenance team.
WEBSPEC51	Clicking the ?Cancel? button will navigate the user to the homepage.
WEBSPEC52	Clicking the ?FAQ? button will navigate to the FAQ.
WEBSPEC53	The FAQ page shall contain two panels, one for links to question, on the left, and one for answers, on the right.
WEBSPEC54	The web portal shall have a ?Guide? button which links to the User Guide.
WEBSPEC55	The User Guide page shall contain two panels, one for the contents page, on the left, and one for the guide, on the right.
WEBSPEC56	The User Guide page will update the guide when the user clicks a link in the contents panel.
WEBSPEC57	The User Guide guide panel will contain a back and next button to change the content in the panel.

5.1.2 Android Application

Code	**Requirement**

APPSPEC1	The app?s login view will feature a ‘Register? button.
APPSPEC2	Clicking the ‘Register? button will display a view featuring full name, email and password text boxes, ‘Register? and ‘Cancel? buttons.
APPSPEC3	Clicking the ‘Register? button will validate the full name, email and password text boxes. If successful, the account will be created, user logged in and navigate the user to a personal home view.
APPSPEC4	Clicking the ‘Cancel? button will cancel the registration process and navigate the user back to the launch view.
APPSPEC5	The app?s launch view will feature a ‘Login? button.
APPSPEC6	Clicking the ‘Login? button will display a login view featuring email and password text boxes, ‘Sign In? and ‘Cancel? buttons.
APPSPEC7	Clicking the ‘Sign In? button will validate the email and password text boxes, log the user in and navigate to a personal home view if successful.
APPSPEC8	Clicking the ‘Cancel? button will cancel the sign in process and navigate the user back to the launch view.
APPSPEC9	The personal home view and options menu will feature a ‘Log Out? button.
APPSPEC10	Clicking the ‘Log Out? button will log the user out of the app and navigate to the launch view.
APPSPEC11	The personal home view and options menu will feature an ‘Account? button.
APPSPEC12	Clicking the ‘Account? button will navigate the user to a view featuring full name, email, password text boxes and ‘Save?, ‘Cancel? buttons.
APPSPEC13	Clicking the ‘Save? button will validate the full name, email, password text boxes and save any changes to the user?s account if successful. A notification will be displayed to inform the user of success or failure.
APPSPEC14	Clicking the ‘Cancel? button will cancel the process of account modification and navigate the user to the personal home view.
APPSPEC15	The launch view will feature a ‘Search? button which is enabled only when the app detects a WiFi connection.
APPSPEC16	Clicking the ‘Search? button will navigate the user to a view containing a search text box, ‘Go? button and list of walks.
APPSPEC17	Clicking the ‘Go? button will filter the walk list to display only the walk names with a full or partial match to the search text.
APPSPEC18	Walk list items will display the walk name, author, partial description, user rating, downloads and ‘Download? button.
APPSPEC19	Clicking the ‘Download? button will download the walk to the user?s device and replace the button with a ‘tick? icon.
APPSPEC20	The app will group walk list items which are close in proximity together in a sublist with a heading name of the common local area.
APPSPEC21	Walk list items grouped by local area will be displayed in descending order based on closest proximity to the app user.

APPSPEC22	An options menu will feature ‘Google Earth? and ‘Google Street? buttons which will be accessible whilst viewing a walk.
APPSPEC23	Clicking the ‘Google Earth? button will change the current walk?s map to the Google Earth view.
APPSPEC24	Clicking the ‘Google Street? button will change the current walk?s map to the Google Street view.
APPSPEC25	Walk edit and add views will feature a default map drop down box containing the options ‘Google Earth? and ‘Google Street?.
APPSPEC26	App data will be downloaded through a web API interfacing a remote database which will be hosted on a Linux server.
APPSPEC27	App data will be synchronised with a remote database when the app is launched and the device is connected to WiFi.
APPSPEC28	The app will display a loading view whilst synchronising app data.
APPSPEC29	The app will feature a settings view containing a toggle button to enable data download over the cellular data network.
APPSPEC30	Clicking the toggle button will change its state to on or off accordingly and the setting will be saved.
APPSPEC31	The settings view will feature a drop down box containing the storage options ‘Internal? and ‘SD Card?.
APPSPEC32	Selecting an option from the storage drop down box will save the setting and move data to the chosen location if it has changed.
APPSPEC33	The app will display a loading view whilst moving app data to a different location.
APPSPEC34	A location strategy will be re-implemented using the Android location services API.
APPSPEC35	The app?s GPS accuracy will be tested on the target Hudle tablet and a low end Android 2.2 mobile device to ensure consistency.
APPSPEC36	The walk view will place a graphic over each waypoint on the map containing a number to represent it?s preferred visit order.
APPSPEC37	The walk view will highlight the next waypoint to visit using a different graphic colour.
APPSPEC38	Waypoint information will be displayed to the user even if the waypoint has previously been visited since the walk view was opened.

5.2 Non-Functional

Code	**Requirement**
NFSPEC1	Database schema shall be agreed with the second development group
NFSPEC2	The final database will be hosted on a central server.
NFSPEC3	The user interface shall be well labelled.

NFSPEC4	The user will follow traditional mobile interface design decisions to reduce the learning curve of the application.
NFSPEC5	Use the interface of default Android applications where possible. e.g. The Holo-light theme
NFSPEC6	Create an optional step by step tutorial for first time users.
NFSPEC7	Create button to access the tutorial in the settings screen.
NFSPEC8	In the settings view there will be an option to enable colour blind mode.
NFSPEC9	Colour blind mode will modify the user interface to use a colour pallette usable by the majority of colorblind users.
NFSPEC10	In the settings view there will be an option to enable a speech to text feature.
NFSPEC11	The speech to texture feature will turn user submitted audio into on-screen text.
NFSPEC12	The settings view will contain an option to allow the device to vibrate when approaching a destination along the route.
NFSPEC13	The settings view will contain an option to allow the device to emit an audible beep when approaching a destination along the route.
NFSPEC14	Coding conventions shall be enforced.
NFSPEC15	Unit tests shall be ran and results documented during development.
NFSPEC16	All code shall be commented, according to conventions.
NFSPEC17	Code documentation shall be generated by Doxygen.
NFSPEC18	Compile with Android 2.2 as a minimum Operating System.
NFSPEC19	Use compatibility libraries to allow for compatibility for both low and high end devices.
NFSPEC20	Create a dynamic user interface using Fragments.
NFSPEC21	Sample GPS based on mode of transport the user has input. (update when better idea).

5.3 Nice To Have

Code	**Requirement**
NTHSPEC1	A Facebook page for the digital trails application will be created to promote the app and enable community feedback.
NTHSPEC2	A Twitter page for the digital trails application will be created to promote the app and enable community feedback.
NTHSPEC3	The Twitter API will be used to access tweets from the app's Twitter page.
NTHSPEC4	The Facebook API will be used to allow sharing to user Facebook accounts.
NTHSPEC5	Allow users to link their accounts to their Facebook accounts.

6 Cross References

6.1 Functional

6.1.1 Web Portal

Requirement Code	Specification Code
RALFREQ1	RALSPEC1, RALSPEC2, RALSPEC3

6.1.2 Android Application

Requirement Code	Specification Code
AFAFREQ1	AFASPEC1, AFASPEC2, AFASPEC3, AFASPEC4, AFASPEC5

6.2 Non-Functional

Requirement Code	Specification Code
NFREQ1	NFSPEC1

7 Methodology

7.1 Development Methodology

7.2 Alternative Methodologies

7.3 Software Testing

8 Risk Analysis

8.1 Risk Matrix

8.2 Technical Risks

Code: TECRSK3 **Likelihood:** 1 **Impact:** 7 **Rating:** 7

Risk: The target tablet device, the Tesco Hudle fails and is irrecoverable to its factory state. This will prevent software testing on the target device and demonstrations to the client.

Mitigation: Non target devices owned by members of the development team will be used to test the software and demonstrate it to the client should the target device fail.

Code: TECRSK4 **Likelihood:** 2 **Impact:** 3 **Rating:** 6

Risk: The personal computers used for software development by members of the team fail and are irrecoverable to an operational state.

Mitigation: University lab computers with pre-installed IDE's (Integrated Development Environments) will be used to develop the software should team member's personal computers fail.

Code: TECRSK5 **Likelihood:** 2 **Impact:** 10 **Rating:** 20

Risk: Data associated with project documents and source code becomes corrupt or lost and cannot be recovered. Extra work will be required to rewrite source code and documents which will severely delay the final delivery.

Mitigation: GIT (see terms) will be used as a version control system which will synchronise and back up source code to a remote GitHub (see terms) repository. This will ensure code is recoverable to a previous state if it becomes corrupted or lost on a single development computer.

Code: TECRSK6 **Likelihood:** 7 **Impact:** 9 **Rating:** 63

Risk: The GPS hardware on the target tablet device is not strong enough to maintain a consistent satellite signal whilst moving. This will limit the accuracy of a user's current location whilst using a walk on the app.

Mitigation: The GPS hardware of the target device, the Tesco Hudle will be researched to ensure its capabilities are fully known. A GPS polling strategy will also be devised to ensure the app locates the user accurately, regardless of transportation.

8.3 Personal Risks

Code: PERRSK1 **Likelihood:** 4 **Impact:** 7 **Rating:** 28

Risk: A member of the team is absent for a substantial amount of time and is therefore not able to contribute to project work. This will cause the work rate to slow and may impact the delivery dates set out in the project timetable.

Mitigation: Extra time will be allocated to complete project tasks so delivery dates are not severely impacted if one team member is absent.

Code: PERRSK2 **Likelihood:** 8 **Impact:** 5 **Rating:** 40

Risk: New software features are continually requested by the client after the initial requirements have been agreed. Unforeseen features will need to be allocated development time thus delaying subsequent project tasks.

Mitigation: An initial set of requirements will be agreed with the client and serve as a contract. Extra feature requests will be developed if they do not impact subsequent tasks.

Code: PERRSK3 **Likelihood:** 5 **Impact:** 9 **Rating:** 45

Risk: The client requests the change of a feature which has already been developed. If this is a large feature such as the maps engine, a large amount of time will be required to change it, thus delaying the project timetable.

Mitigation: Features will be prototyped and shown to the client prior to implementation. This will ensure the client agrees with the features at a user interface level.

Code: PERRSK4 **Likelihood:** 3 **Impact:** 3 **Rating:** 9

Risk: The user interface design of the web portal or app is poor causing users to struggle when using basic functions.

Mitigation: The best practices of mobile interaction design will be adopted to ensure the app's user interface is intuitive. Neilson's design heuristics will also be applied to the web portal user interface to ensure it is also intuitive.

Code: PERRSK5 **Likelihood:** 7 **Impact:** 3 **Rating:** 21

Risk: Major or minor software bugs are present in the app or web portal which inhibit the desired functionality. This will have a negative impact on the user experience and require extra development time to fix.

Mitigation: Blackbox testing methods such as unit and regression testing will be used at the end of each sprint to ensure modules are free from bugs. Acceptance and compatibility tests will also ensure the software is free from user interface and hardware specific bugs.

Code: PERRSK5 **Likelihood:** 8 **Impact:** 2 **Rating:** 16

Risk: The work completed by the previous development team is substandard and has left the app in an unmaintainable state. A large amount of time will be required to rewrite source code to a maintainable state.

Mitigation: .

9 Project Plan

9.1 Milestones

Task	**Start Date**	**End Date**
Milestone 1	08 Nov 2013	13 Dec 2013
General Description	08 Nov 2013	13 Nov 2013
Software Requirements	13 Nov 2013	18 Nov 2013
Software Specifications	18 Nov 2013	22 Nov 2013
Development Methodology	22 Nov 2013	27 Nov 2013
Risk Analysis	27 Nov 2013	02 Dec 2013
Initial Work	02 Dec 2013	06 Dec 2013
Draft Review	06 Dec 2013	13 Dec 2013
Milestone 2	14 Feb 2014	28 Feb 2014
Interim Report	14 Feb 2014	21 Feb 2014
Draft Review	21 Feb 2014	28 Feb 2014
Milestone 3	21 Mar 2014	09 May 2014
Project Poster	21 Mar 2014	28 Mar 2014
User Manual	28 Mar 2014	04 Apr 2014
Final Report	04 Apr 2014	02 May 2014
Software Design	04 Apr 2014	11 Apr 2014
Software Testing	11 Apr 2014	18 Apr 2014
Reflective Account	18 Apr 2014	25 Apr 2014
Draft Review	25 Apr 2014	09 May 2014

Figure 1: Milestone time table.

9.2 Software Development

Task	Start Date	End Date
Initial Requirements	01 Oct 2013	11 Oct 2013
Background Research	01 Oct 2013	04 Oct 2013
User Interviews	04 Oct 2013	11 Oct 2013
Paper Prototypes	11 Oct 2013	31 Oct 2013
Prototype Development	11 Oct 2013	21 Oct 2013
Wizard of Oz Experiments	21 Oct 2013	23 Oct 2013
Revision & Enhancement	23 Oct 2013	31 Oct 2013
Static HTML Pages	31 Oct 2013	22 Nov 2013
Prototype Development	31 Oct 2013	14 Nov 2013
Cognitive Walkthrough Experiments	14 Nov 2013	17 Nov 2013
Revision & Enhancement	17 Nov 2013	22 Nov 2013
Service Simulation	13 Dec 2013	29 Jan 2014
Prototype Development	13 Dec 2013	10 Jan 2014
Prototype Feedback	10 Jan 2014	15 Jan 2014
Revision & Enhancement	15 Jan 2014	22 Jan 2014
Software Testing	22 Jan 2014	29 Jan 2014
Service Implementation	29 Jan 2014	14 Mar 2014
Prototype Development	29 Jan 2014	19 Feb 2014
Prototype Feedback	19 Feb 2014	21 Feb 2014
Revision & Enhancement	21 Feb 2014	28 Feb 2014
Software Testing	28 Feb 2014	14 Mar 2014

Figure 2: Software development time table.

10 Initial Work

10.1 Minutes of Meetings

10.2 User Interface Wire Frames

10.3 Future Work