

Group 06 - Project Plan

23rd October 2012

Amy Rebecca James, James Slater, Dan McGuckin, Samuel Mills, Felix Farquharson,
Ben Brooks, Aiman Arafat, Chris Arom

Project Coordinator: Nigel Hardy

Version: 1.1

Status: Release

Computer Science Department, University of Wales, Aberystwyth

The following document is copyright of CS22120 Group 06 Aberystwyth University

Contents

1	Introduction	2
1.1	Purpose of the Document	2
1.2	Scope	2
1.3	Objectives	2
2	Overview of Proposed System	3
2.1	Choice of platform	3
2.1.1	Glassfish Open Source Server	3
2.1.2	Google App Engine	3
2.1.3	Apache Tomcat	3
2.2	High Level Architecture	3
2.2.1	Version Control	3
2.2.2	Integrated Development Environment	3
2.2.3	Documentation Tool	4
2.3	Description of Target User	4
3	Use-case Diagram	5
4	User Interface Design	6
4.1	Register/Login	6
4.2	Home	7
4.3	Shop	8
4.4	Battle	9
4.5	Post Battle	10
5	Gantt Chart	11
5.1	Introduction	11
6	Purpose of this document	11
6.1	Objectives	11
7	Gantt Chart	11
7.1	Key for Gantt Chart	11
7.2	Gantt Chart	12
8	Risks	13
9	Conclusion	14
10	References	15
11	Document History	15

1 Introduction

1.1 Purpose of the Document

This document describes the outline design for the Software Development Life Cycle Group Project 2012. It should be read in the context of the Group Project, taking into account the details of the group project assignment and the group project Quality Assurance (QA) plan [1].

1.2 Scope

The design specification splits the project into individual implementable parts and describes the interfaces and interaction between those components. The design specification refers to the Requirements Specification for the group project. It is important that this document is read by all members of the project group, especially the implementation team.

1.3 Objectives

The objectives of this document are:

- To identify and describe the main features of the Monster Mash game.
- To note the details of the criteria that the group project product must achieve.
- To administer interface details for each of the central classes in the Monster Mash game.

2 Overview of Proposed System

2.1 Choice of platform

We have decided to use the Glassfish Open Source version as our platform. We did not consider the Oracle Glassfish Server because it would cost to use it for the project. We considered the following:

- Glassfish Open Source Server
- Google App Engine
- Apache Tomcat

2.1.1 Glassfish Open Source Server

There are a number of benefits to this software above the other options. The main two reasons for using this server are because it is open source and because some members of our group have previous experience with it. Another reason is because we expect there will be support for this environment available. Both from the university and from the contributors to the glassfish project. Glassfish has many more features than Tomcat, the other open source option. Spike testing was carried out and it was found that this piece of software was easy to use and appropriate for the nature of our project.

2.1.2 Google App Engine

The main reason we didn't choose this software is because it proved unreliable in tests. This software is also closed source and using it would mean that you rely upon Google when the application is released.

2.1.3 Apache Tomcat

Tomcat was not as fully featured as Glassfish, and no one in the group has ever used it before, so there would be a steeper learning curve for them and there would be no "in-group" support for using it.

2.2 High Level Architecture

2.2.1 Version Control

For version control we are using Git. Git is a distributed version control system, which some members of the group already have experience with. Distributed version control systems give a slightly different development pattern which suited the qualities of a group better than SVN. Version control systems we considered:

- Git
- Bazaar
- Subversion

2.2.2 Integrated Development Environment

We have decided to use the NetBeans IDE, because it is available free and it is the preference of the majority of the group. Modules are available for NetBeans to help with Version Control (Git) and JUnit. IDEs considered:

- Eclipse
- NetBeans

2.2.3 Documentation Tool

We decided to use L^AT_EX because it is widely supported, there is a template provided, and because it was preferred by the majority of the group. Methods of documentation we considered:

- L^AT_EX
- Open Office/Libre Office
- Microsoft Word

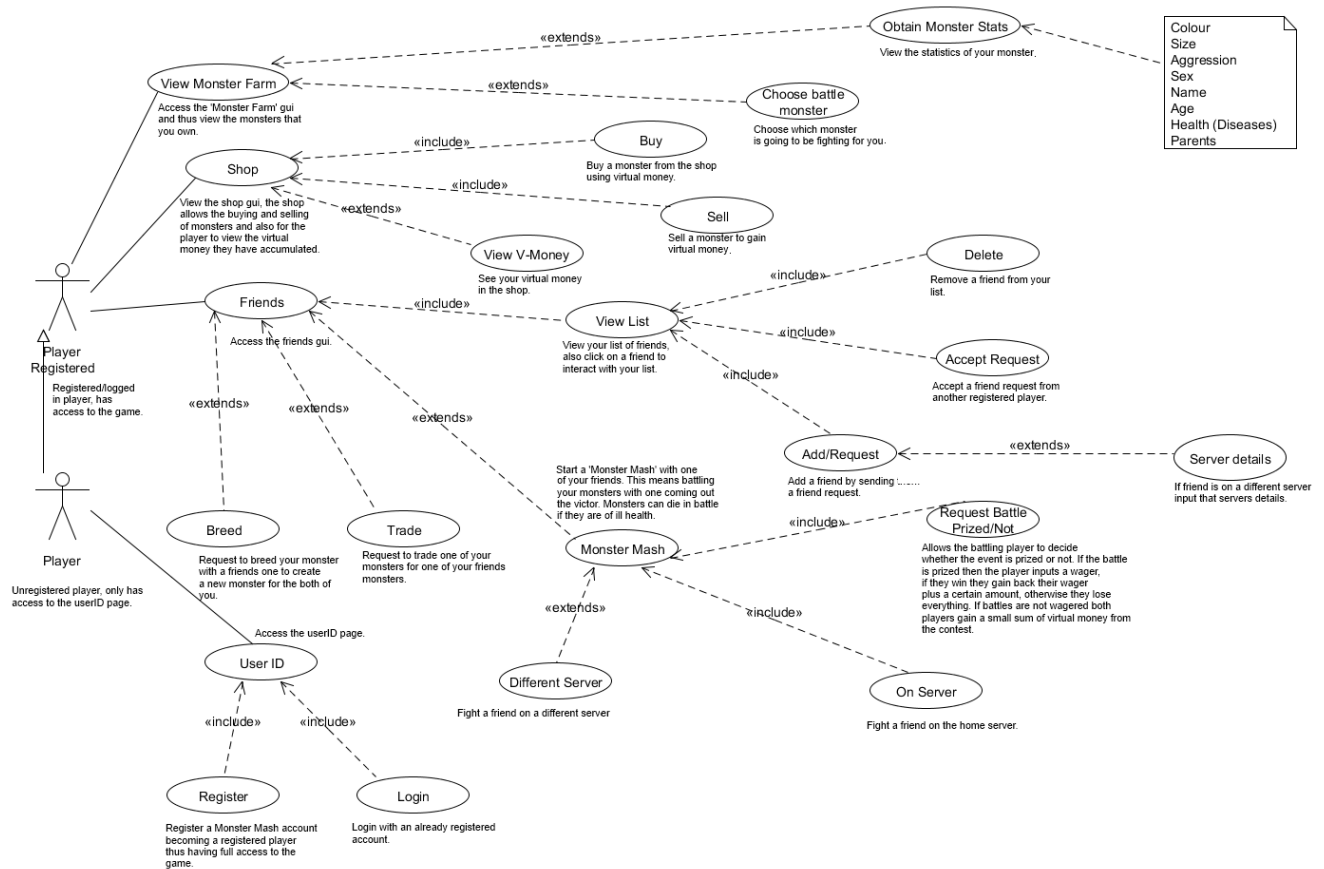
2.3 Description of Target User

The target user will be young people. Typically aged between 11 and 16. We will have to make sure that no complicated language is used without good reason and we will have to make sure that all content is appropriate. Other things to consider are:

- Make sure that there are no really lengthy tasks to do.
- Make sure that it will fit around the lifestyle of a young person of that age. ie. Around school, limited access to a computer.

3 Use-case Diagram

Year/Group Project/Group-P Use Case.pdf



4 User Interface Design

4.1 Register/Login

This screen is the first thing a user will see when visiting the Monster Mash website. The user will be provided with text boxes to input their username, password and optionally a server location to be connected to. Depending on which button a user presses will either Log them in if they already have an account, or register them if they do not. This makes the login and registration processes much easier than having two separate pages for each process.

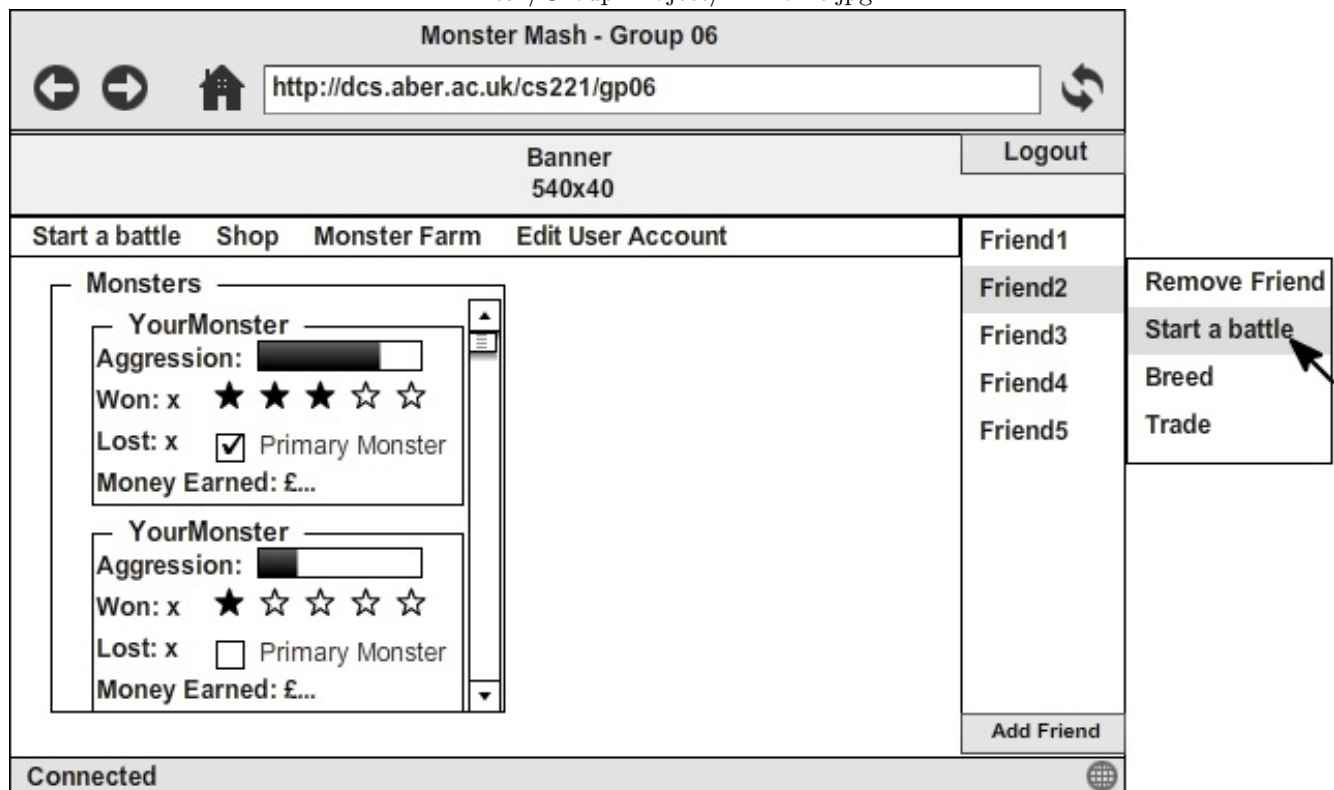
Year/Group Project/mmregisterlogin.jpg

The screenshot shows a web browser window titled "Monster Mash - Group 06". The address bar displays "http://dcs.aber.ac.uk/cs221/gp06". Below the address bar is a banner area labeled "Banner 540x40". The main content area features a "Login/Register" form with three input fields: "username", "password", and "server location". Below these fields are two buttons: "Login" and "Register". The browser's status bar at the bottom indicates "Connected" and shows a globe icon.

4.2 Home

On the home page, the user has a friends list down the right hand side. They can right click on any of the friend's names and a popup menu will appear to give them various options such as 'remove friend', 'start a battle', 'breed' or 'trade'. Along the top of the screen there is a banner which will be displayed on all of the pages. It will probably contain a graphical logo and text. Below that there is a menu which has links to other pages such as 'shop', 'farm' and 'edit user account'. Below the menu there will be a list of the users mosnters, containing their stats such as 'aggression', their win/loss count etc.

Year/Group Project/mmhome.jpg



4.3 Shop

In the shop the user has the option of buying or selling monsters (and possibly items). Down the left hand side there is a list of monsters available to buy. This lists the monster's stats, cost and rating. On the right hand side is similar, except it contains lists of the users monsters which they can sell for money.

Year/Group Project/mmshop.jpg

The screenshot shows a web browser window titled "Monster Mash - Group 06". The address bar displays "http://dcs.aber.ac.uk/cs221/gp06". The page has a navigation bar with links: "Start a battle", "Shop", "Monster Farm", and "Edit User Account". The "Shop" link is currently selected.

The main content area is divided into two columns. The left column is titled "Buy" and contains two identical monster listings. Each listing includes a "MonsterName" input field, an "Aggression" bar (partially filled), "Won: x" with a star rating (3 stars filled, 2 empty), "Lost: x", and a "Buy - £..." button. The right column is titled "Sell" and contains two identical monster listings. Each listing includes a "YourMonster" input field, an "Aggression" bar (partially filled), "Won: x" with a star rating (3 stars filled, 2 empty), "Lost: x", and a "Sell - £...." button.

On the right side of the page, there is a vertical list of friend names: "Friend1", "Friend2", "Friend3", "Friend4", and "Friend5". Below this list is an "Add Friend" button.

At the bottom of the page, there is a "Connected" status indicator and a globe icon.

4.4 Battle

During a battle, the user is displayed with stats of each monster partaking in the battle. This includes 'agression', 'rating', 'win/loss count' and 'total money earned'. When the user presses the 'Start the fight!' button, they are taken to a page with the results as seen below.

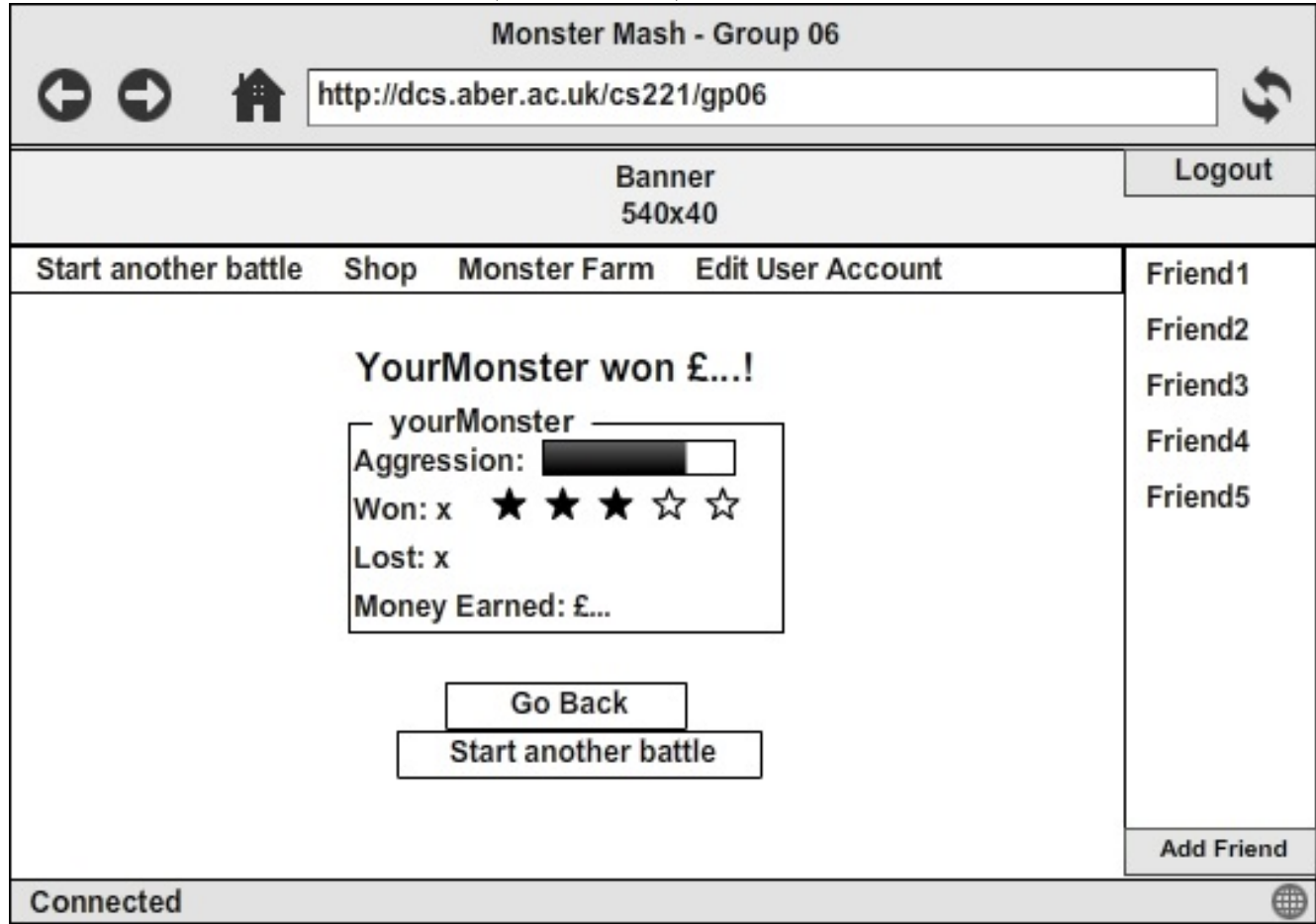
Year/Group Project/mmbattle.jpg

The screenshot shows a web browser window titled "Monster Mash - Group 06". The address bar displays "http://dcs.aber.ac.uk/cs221/gp06". The page has a navigation bar with links: "Start another battle", "Shop", "Monster Farm", and "Edit User Account". A "Logout" button is in the top right. The main content area is titled "Battle between yourMonster and theirMonster". It features two monster stat boxes. The "yourMonster" box shows "Aggression: [progress bar]", "Won: x [3 stars, 2 empty stars]", "Lost: x", and "Money Earned: £...". The "theirMonster" box shows "Aggression: [progress bar]", "Won: x [2 stars, 3 empty stars]", "Lost: x", and "Money Earned: £...". A "Start the fight!" button is centered below these boxes. On the right side, there is a list of friends: "Friend1", "Friend2", "Friend3", "Friend4", and "Friend5", with an "Add Friend" button at the bottom. The status bar at the bottom left says "Connected" and the bottom right has a globe icon.

4.5 Post Battle

This screen shows which monster won the battle. It displays how much money it earned for winning, and also the monster's stats. From here you can either click 'Go Back' to return to the homepage, or click 'Start another battle' to initiate another

Year/Group Project/mmpostbattle.jpg



5 Gantt Chart

5.1 Introduction

This Document is the plan this Group project, for each member to see what he/she has to do and when to finish by. Each member has their own colour in the Gantt Chart so its easy to follow.

6 Purpose of this document

6.1 Objectives

The main objective of a Gantt chart is for everyone to be able to follow what's being done by who and when.

7 Gantt Chart

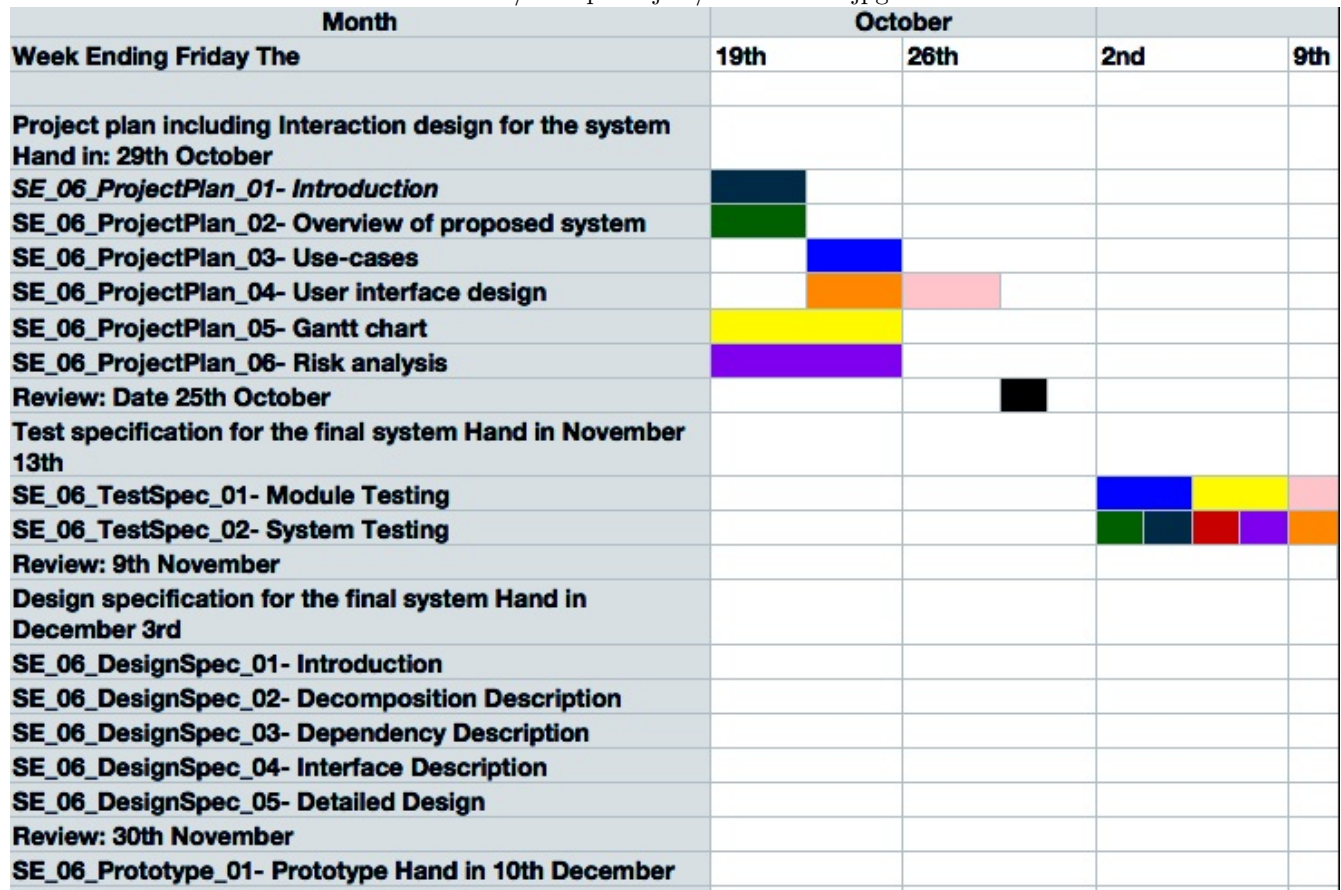
7.1 Key for Gantt Chart

Year/Group Project/GanntKey.jpg

Amy	
Dan	
Ben	
Aiman	
Slater	
Sam	
Felix	
Chris	

7.2 Gantt Chart

Year/Group Project/GanttChart.jpg



8 Risks

When planning and developing a piece of Software, there are many different aspects of the development process. This is why Software Engineers need to try and predict and possibility and minimise the chances of the problem having any serious effect on the process e.g. - losing time, unable to make a deadline or even to the extent of a particular part of the project being left incomplete.

Primarily it is the job of the Project Group to identify these risks and to remain committed so that they can be avoided or at least minimised.

When planning a project, all members would like the Software completed and available to the customer as soon as possible; this may result in time allocation becoming to hasty, and certain members being left unable to finish their task on time or to the best of their ability. This can lead to problems further down the development process.

The communication of all group members is key. All members need to have some idea of what other members are doing. Some tasks run closely alongside one another which make it even more vital that these group members communicate so that decisions remain consistent. Lack of communication can lead to the major problems as tasks may not be completed on time or at all. Also putting the different parts of the project together can prove extremely difficult without sufficient communication. For these reasons it is vital that Group Members attend all meetings and check their email so that they are up to date with all decisions made by the Project Manager.

Sudden illness of a key Group Member (Project Manager) can cause all sorts of problems. For this reason it is key that other Group Members (Deputy Project Manager) knows what the Manager's intended plan was so that the group can continue without any loss of time and achieve their goals.

Getting to grips with new programs etc. can be difficult and time consuming. The use of GitHub and Latex may cause some team members to fall behind with their tasks if they have not used them before. Documents may take longer to be produced in Latex, and it may cause confusion getting used to the push pull system with the GitHub repository. The use of books and advice from other group members will help them get used to them faster.

As there are a few different coders in the team this may cause difficulty when integrating all the code together as everybody codes differently, it may be hard to put all the code together and also it could be difficult to integrate with the GUI. Once again this is down to good planning and communication. The coders should work closely side-by-side and keep checking the repository for changes. It is a good idea to do small parts of the coding but often so that if that piece of code causes any issues with other people's work it is easier to correct.

As the Project Team may want to refer back to the Project Coordinator (Nigel Hardy) when unsure of what a task is, there may be a risk of set backs if he was absent for some reason, tasks then may not be completed correctly or may not even be able to be completed without his input. If this was to happen the group would have to make themselves aware of other members of staff that can help so that the situation doesn't hold them back too drastically.

It may also be the case that the group repository is inaccessible due to technical issues or due to the fact that it is under attack. This can cause setbacks that lead to missed deadlines. As these systems can be down for hours at a time the group should make sure that the work is completed well in advance so there is no last minute rushing. All members should back up their own work as well.

The use of complex algorithms in the projects can cause problems as they may cause bugs that the coders cannot correct. This can be prevented by completing small pieces of codes at a time so that bugs may be easier to correct. There may also be issues with memory and speed. There is a difference between an application working and it being usable so all these things must be verified.

9 Conclusion

These are all important issues that if taken into account at the beginning of the project can prevent loss of time. If taken into consideration then, if one of these risks were to happen, it would be of no major significance to the project and the group would be able to carry on without panicking.

10 References

[1] Software Engineering Group Project Plan Specification Standards. Bernard Tiddeman. SE.QA.05B. 1.1 Release

11 Document History

Table 1: Document History

Version	CCF No.	Date	Changes made to document	Changed by:
1.0	N/A	25/10/2012	Creating Project Plan	arj18
1.1	N/A	29/10/2012	Making changes from review	arj18