Software Development Life cycle
Project Plan

Author:	Tom Reed, Matt Whitmore, Dave Clark, Silhab Csoma, Mike Steel, Chris 'Tux' Lloyd, Aleksandra Badyda, Samuel Jackson, Chris Marriott

Config. Ref.:	SE.17.DS.01
Date:	2012-12-5
Version:	1.3
Status:	draft

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

CONTENTS

1 Introduction

1.1 Purpose of this Document

This is the user interface specification. This document is intended to show how the system will look and feel. It will also show the risk assessment and the Gantt chart.

1.2 Scope

This document will show how the user will be able to interact with the system through use case diagrams. How the system will look through user interface design. It will also list the risks that could possibly arise and what people in the groups primary and secondary roles are. This will be shown in the risk assessment. There will also be a example Gantt chart that will list people in the group, their task and time frame for which it is intended to be completed.

1.3 Objectives

The objective of this document is to design the interface and usability of the system for the monster mash game. The areas covered by this plan are:
Start bullet points

- Overview of system
- Use Cases
- User Interface mock ups
- Gantt Chart
- Risk Analysis

2 Overview

This project requires the group to be able to design and create a web based application that allows the users to socially interact with one another. Monster Mash (The name of the social web application) will allow the user to create an on-line profile and once this is completed, the application will give each new user their own monster, of which they can use to challenge other users of the application to a battle, and depending on the outcome (win or lose) will award you with points and will alter the statistics of your monster. To commence a battle with another user, its required that they are both "Friends" with one other on the web application, like

many other social networking sites, this will be done by one user sending a friend request, and the other either accepting or rejecting. Aside from battling each others monsters, the game should be able to include other features such as breeding, so that users can create stronger monsters, and also a high scores page, on which friends can check to see who has the best overall score out of each of them.

3 Deployment

3.1 Overview

We have decided to use a small enterprise style system, which will be hosted on Chris Lloyd's VPS (virtual private server). While this would normally not be hosted on a single server (due to security), it does allow almost all of the benefits such as it being live(non-local), 99.9 percent uptime and self-managed hosting. It also gives us some experience with how to deal with real world solutions that we may come across in our future lines of work.

3.2 Hosting

The hosting has been sorted by Chris Lloyd; he has allowed the group to use his VPS which he has installed Apache http server, MySQL database and a tomcat. Chris Lloyd has also purchased the domain monstermashgame.co.uk to which he has added an A-record to the VPSs IP.

3.3 Apache Tomcat HTTP

Apache Tomcat is used to run the Java environment in a public accessible zone. The servlets will handle the HTTP requests from the Java environment, which sit on Apache tomcat. This will mainly involve sending HTML pages.

3.4 MySQL

MySQL, while not being used at the higher end of the corporate market (held by Oracle), it does allow us to use an enterprise like database system, as it is quite commonly used for small to medium sized business and is often used on small scale web hosting.

3.5 TomCat

Tomcat as a project, is developed by the Apache foundation. Tomcat is widely deployed as a Java application server, all though through the various configurations

it can be used to make a mass distributed Java cluster. This is to say that it is possible to run a single Java application across a distributed processing cluster.

3.6 Server Side Conclusion

While there is other software out there that may do a better job, we found that by sticking to the widely used software, implementation and application, we would find a good, well rounded base knowledge in the area of server side web applications.

4 Methodology

Although the structure of the hand in documents leads us to waterfall, we shall aim to do incremental within this. (See fig1)
The advantages are:

- Easier management
- Can get user feedback earlier
- Can avoid crises by be alerted to problems earlier
- Can respond easier to changing user requirements
- Earlier exchange of functionality

The disadvantages are:

- Can be harder to manage, with more steps and crucial decencies and overall progress monitoring
- Version and configuration control is crucial and can be complex

Incremental Development

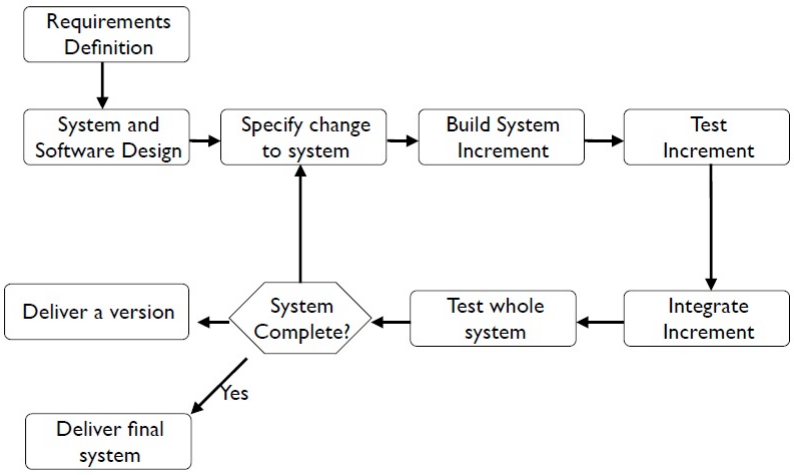
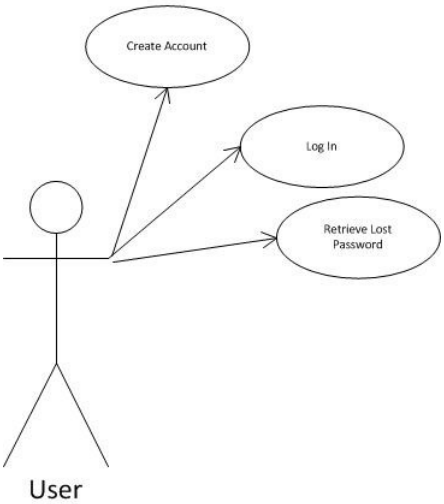


Fig1

5 Use Case

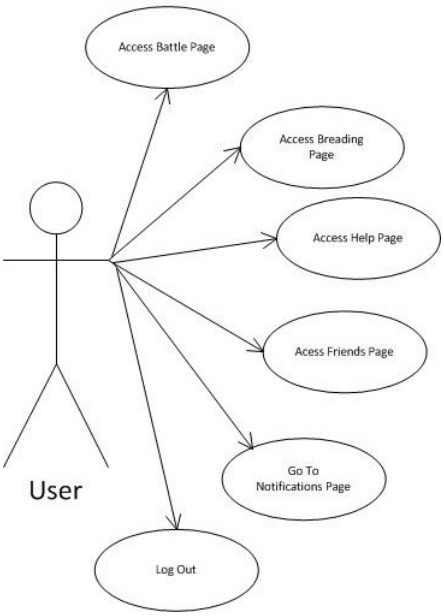
This section will highlight the functional requirements that were defined in the requirement pdf.

5.1 Welcome Page



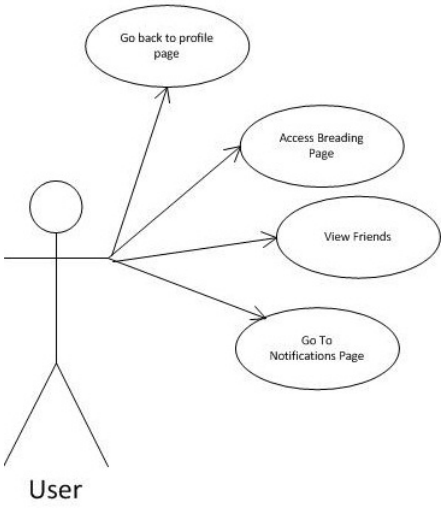
Functional Requirements- FR1 - Server-based Authentication FR6 - Client Options FR7 - Start-up of software in browser

5.2 Main profile page



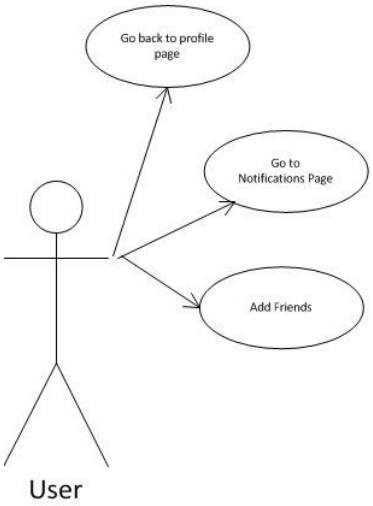
Functional Requirements- FR8 - Game Display In Browser FR10 - Fight Notifications

5.3 Friends Page



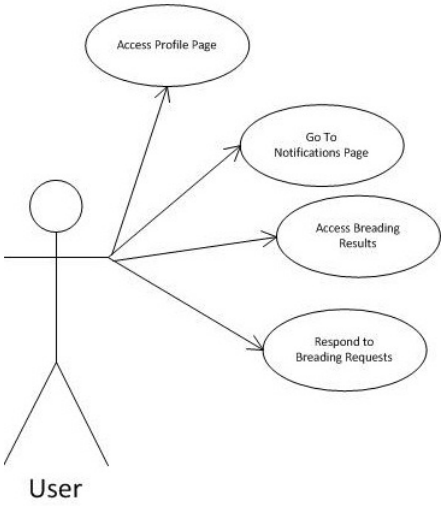
Functional Requirements- FR2 - Server Friends List FR5 - Server-server communication FR6 - Client Options FR9 - Friend Matching FR11- Friends Rich List

5.4 Add Friends



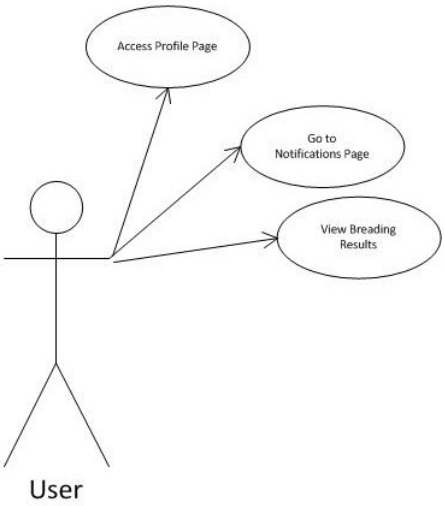
Functional Requirements- FR5 - Server-server communication FR6 - Client Options FR9 - Friend Matching

5.5 Breeding



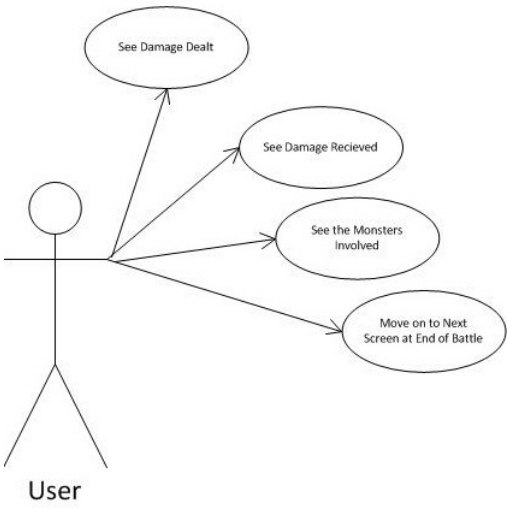
Functional Requirements- FR3 - Server Monster List FR5 - Server-server communication FR6 - Client Options FR8 - Game Display In Browser

5.6 Breeding Results



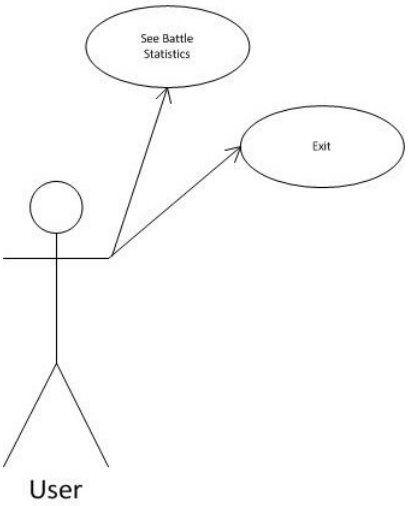
Functional Requirements- FR5 - Server-server communication FR6 - Client Options FR8 - Game Display In Browser

5.7 Battle Screen



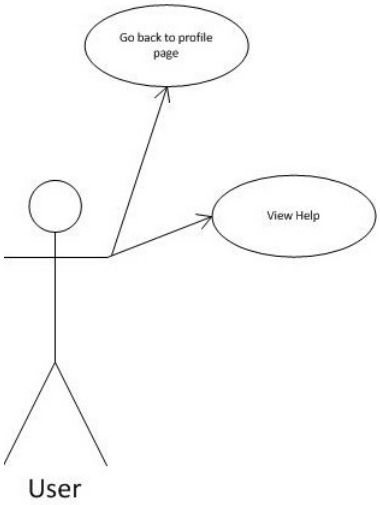
Functional Requirements- FR4 - Server Monster mash management FR5 - Server-server communication FR6 - Client Options FR8 - Game Display In Browser

5.8 Battle Results Page

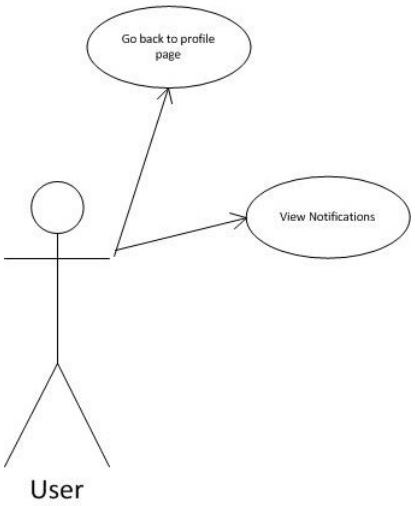


Functional Requirements- FR3 - Server Monster List FR8 - Game display in browser FR10 - Fight Notifications

5.9 Help Page

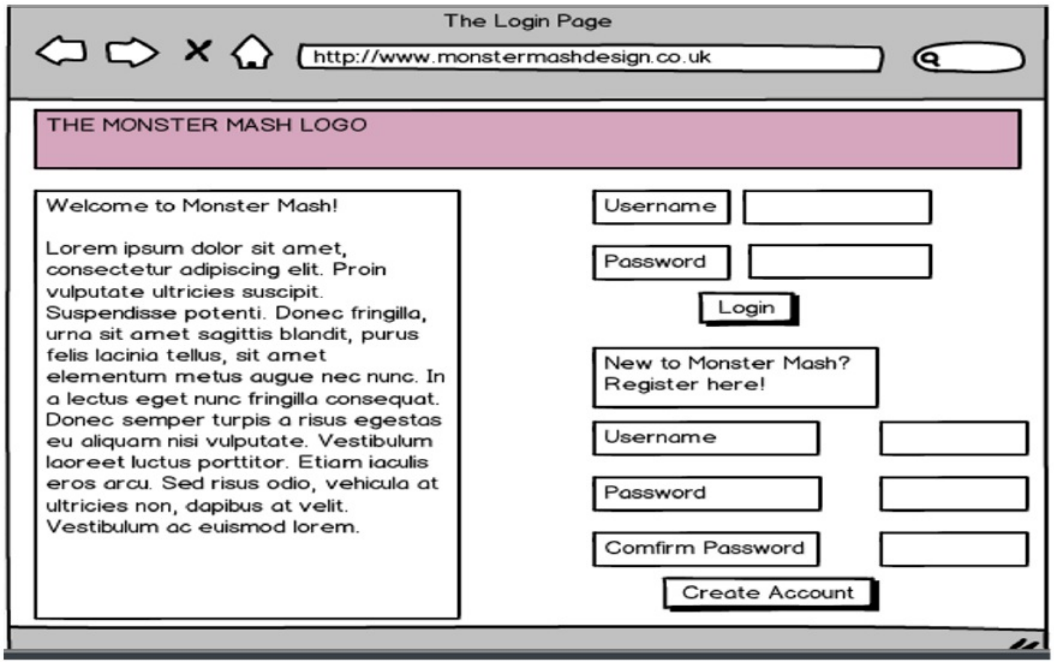


5.10 Notifications Page



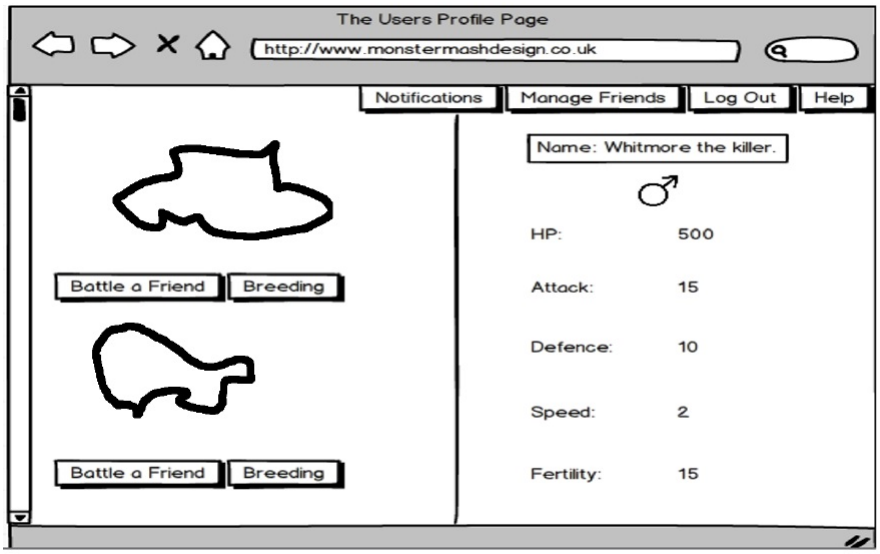
6 User Interface Design

6.1 Login Screen(fig2)



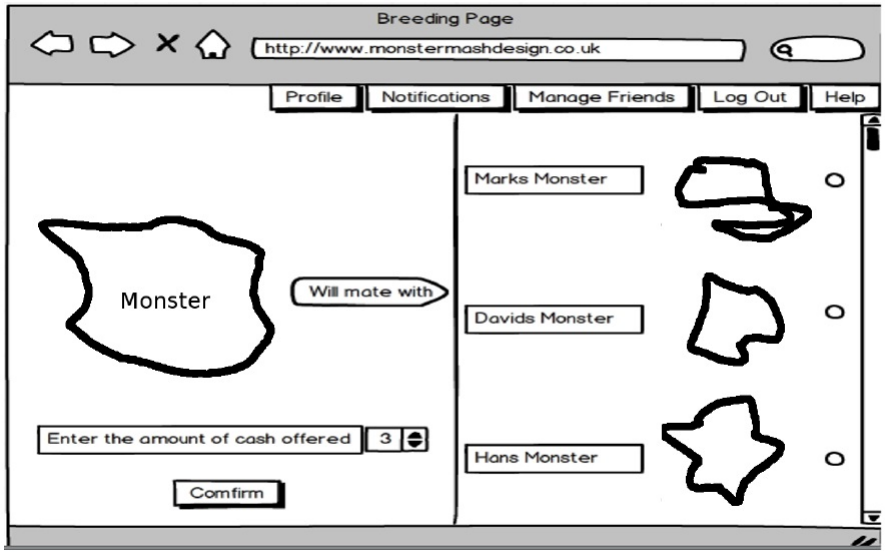
This is the basic design of the page the user will first encounter when they want to either create an account or log in if they already have an account. Both the Login button and the Create Account button (if filled in correctly) will take the user to their profile page. This is the only page in which the user is not signed into their account. Once logged in, the user will be directed to Figure 3.

6.2 Profile page(fig3)



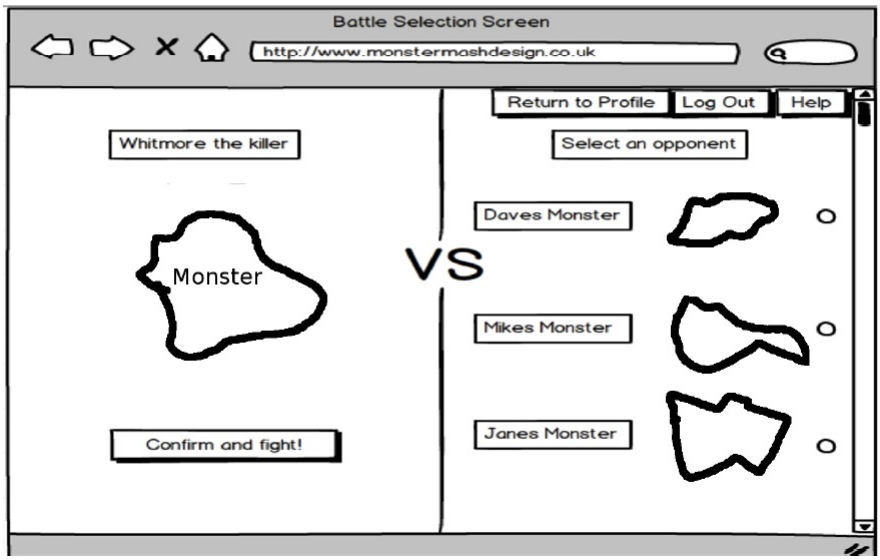
This is the basic design for the profile page of our project, this will be the page the login screen takes you each time you log in. It is from here, you (the user) will be able to select each of your monsters to either battle or breed with a friends monster. There are other features that are accessible from this page, such as the Help page and the Manage Friends page, which can be seen in the top right corner of the screen, however these are also accessible from other pages aside from the Profile page, so we will just focus on the Breeding and Battle a Friend page. On the right hand side of the page, the stats for the monster you have selected will be shown, this will be done by clicking or possibly hovering over the monster with the mouse. You select which monster you want to breed by clicking on the corresponding breeding button. Located underneath the monster you wish to select. Once you have selected your monster you wish to breed. You are taken to figure 4.

6.3 Breeding Page(fig 4)



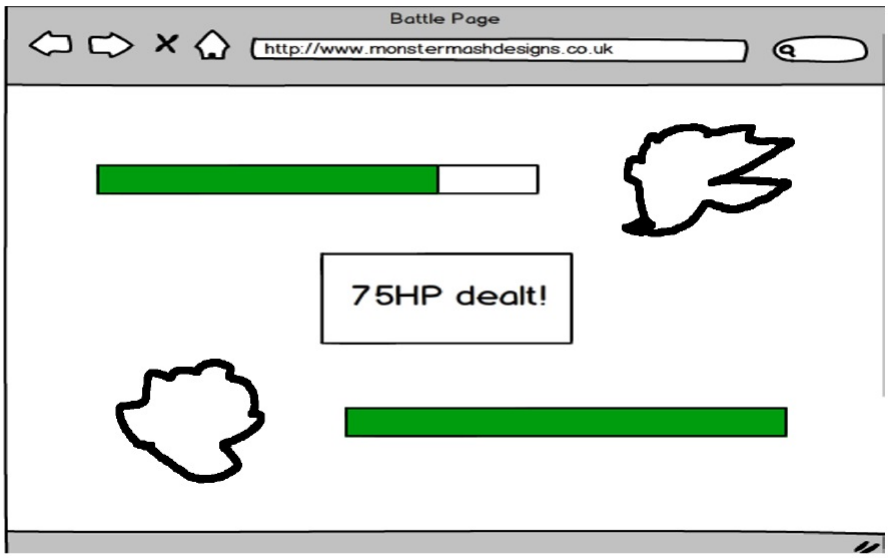
The monster that you selected for breeding from the previous page will be shown on the left hand side of the screen, on the right hand side of the screen are all the monsters that your friends own, the idea here being that you have to select one of your friends monsters to breed with the monster you have, seeing that you are the one who benefits from the breeding process (the one being requested to be bred with does not gain a monster) the user must enter an amount of points, or cash into the box located on the left hand side of the page (the one with the number 3 in it). It is then up to the other player to either accept or reject this offer, if it is accepted, the breeding takes place and the points are transacted from one users account into the others. The player who initiated the breeding process then gains a monster. Returning to Figure 3, if you want to select a monster to battle another friends monster, as with breeding, simply select the corresponding button underneath the monster of your choice, you will then be taken to Figure 5.

6.4 Battle Selection Screen(fig 5)



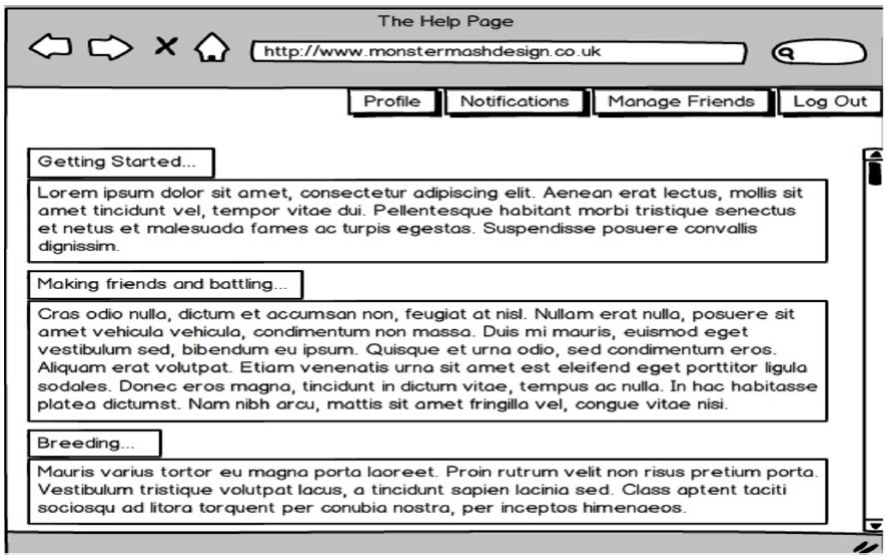
Similar to the breeding page, this requires the user to select one of their friends monsters to challenge, once they have selected, the user simply presses the confirm and fight button, and the request is sent. Both pending battle and breeding requests can be viewed in the Manage Friends page. Incoming requests will appear on your notifications.

6.5 Battle Page(fig 6)



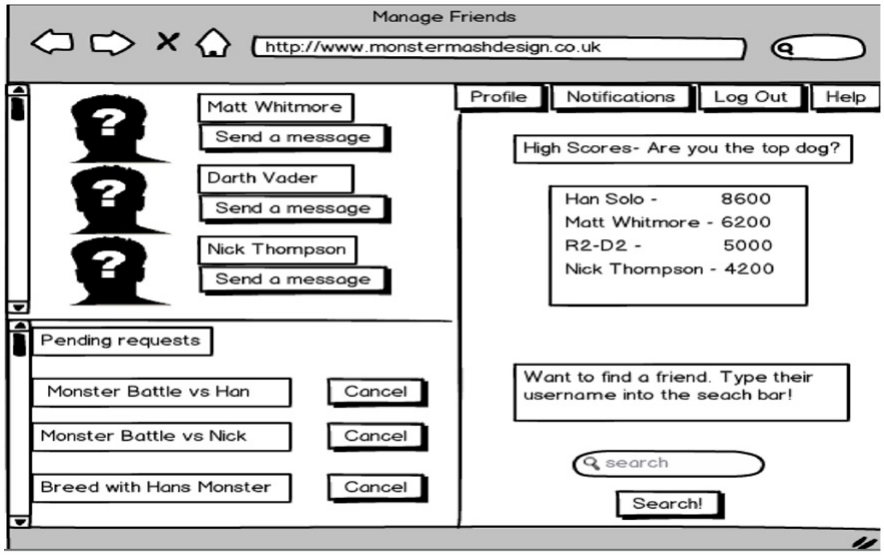
It should be noted that this page will only be viewed by the user accepting the battle request. Once the user has accepted a battle request off their friend, they will be redirected to this page. This is an example of a battle taking place between two monsters. The user (so the one who accepted the battle request) can watch the battle unfold. The text in the middle constantly updates, telling the user how much damage each monster deals to each other. The green bar next to each of the monsters is their health bar, this gradually decreases as the battle progresses. The first monster to lose all of its life it declared the loser. After the battle, the user is directed back to the Profile page (Figure 3), the other user that wasn't present for the fight receives a message in their notifications telling them how the fight unfolded. Moving back to Figure 3, if the Help button, located in the top right hand corner is selected, Figure 7 should appear.

6.6 The Help Page(fig 7)



A simple page in design and implementation. This page will tell the user all they need to know about how to user the application, including things such as how to battle, how to breed and how to add friends.

6.7 Friend Management Page(fig 8)



The Manage Friends page can be accessed from most of the web pages, the button that directs to it is located in the top right hand corner of the screen. The Manage Friends page holds many functions. From here you are able to cancel pending requests to either battle or breed with another friends monster, send personal messages to your friends, and new friends to your list and also check who has the top score overall out of all your friends. To log out of Monster Mash, simple click the Log Out button which is found on most of the pages to the top right hand corner. This will take you to back to Figure 1.

7 Gantt chart

GANTT CHART PICTURE HERE This is the Gantt chart that is used for assigning tasks and a time period that those tasks should be finished by. The image below is merely an example as the actual Gantt chart is constantly changing. Can be seen on-line at http://users.aber.ac.uk/cpm4/groupProject/gantt.html

8 Risk Assessment

Hazard	Who/What is at Risk	Preventative Procedures Present	Necessary Further Action	Whom is to Action This	When to Action this by	Date Actioned:
Unfamiliarity with coding language being used.	Project files/Ability to meet deadlines	• Anyone with problems may contact the group leader to highlight and discuss anything that is wrong. • People selected for programing roles have all had experience with the particular code that they are each using.	• Secondary 'Coders' to be found for each language, who have knowledge of the given language being used, and can offer help to anyone who needs it.	Mike	1/11/12	24/10/12
Illness	Group members/Ability to meet deadlines	• Rooms people work in are cleaned regularly. • When working at the university, the Union Shop provides adequate balanced meals to improve health. • Group Members are not penalised for having days off for illness.	• Days of illness should be officially recorded and used to make sure that no one is regularly faking to get out of work. • Group members should be encouraged to eat and live healthily.	Chris	1/11/12	
Quitting the Course or Group	Ability to meet deadlines	• Anyone with problems may contact the group leader to highlight and discuss anything that is wrong.	• Secondary people should be found for each role, who can take over should the original role holder be unavailable.	Mike	1/11/12	24/10/12
Printer Malfunction	Ability to meet deadlines	• A number of printers are available in different parts of the University. • Technicians are constantly on hand to repair broken University printers.	• Items that need to be printed should be printed as soon as they are agreed as finished by the group, and not before the deadline.	Chris	1/11/12	24/10/12
Computer Malfunction	Project files/Ability to meet deadlines	• The project files are regularly backed up online, on both the Facebook page and on Github. • Backups are regularly made on memory sticks. • Many computers are available at the University if any personal computers break.	• A set time for when backups should be made should be set out, making sure that each group member has everything backed up; either daily or weekly or as seen fit by the group leader.	Chris	1/11/12	24/10/12
Online Backup Malfunction	Project Files/Ability to meet deadlines	• Two backup sites are used; the main collaborative store on Github, as well as the files kept on the Facebook group page. • What is backed up online is also backed up on hardware owned by group members, in case both online backups are unavailable.	• Reiterate to group members the importance to keep a number of types of regular backups of all work.	Chris	1/11/12	24/10/12
File Insecurity	Project Files	• All file stores have passwords. • All memory sticks and similar storage items are kept securely.	• Work out a group policy on passwords and keeping passwords secure.	Chris	1/11/12	24/10/12

		• All computers or computer accounts used by group members have password protection on them.	• Store items kept on memory sticks in password protected folders to make sure that files are safe even if the memory stick is lost or stolen.			
Eye Strain from Computer Over-usage	Group members	• It is group policy to take regular breaks from computers. • Computer rooms are well lit and screens are facing away from windows to reduce glare.	• Backup roles to be assigned in case anyone is injured and cannot work.	Chris	1/11/12	24/10/12
RSI from Computer Over-usage	Group members	• It is group policy to take regular breaks from computers.	• Backup roles to be assigned in case anyone is injured and cannot work.	Chris	1/11/12	24/10/12
Back Problems from Computer Over-usage	Group members	• It is group policy to take regular breaks from computers. • All computer seats are suitable and kept at a good height and are adjustable.	• Backup roles to be assigned in case anyone is injured and cannot work. • Backup roles to be assigned in case anyone is injured and cannot work.	Chris	1/11/12	24/10/12
Slips and Trips	Group members	• Wet floor signs are used on University property when necessary. • Suitable shoes for walking up the hill are advised by group policy. • Computer rooms are cleaned regularly and floors kept clear.	• Advise group members to clean up any objects they find that may be a hazard. • Backup roles to be assigned in case anyone is injured and cannot work.	Chris	1/11/12	24/10/12
		• All areas on campus are well lit.				
Stress	Group members (Particularly group leader)	• None	• Backup roles to be assigned in case the stress is too much for one person.	Chris	1/11/12	24/10/12

As part of the above assessment, the following secondary roles have been assigned.

Name	**Primary Role**	**Secondary Role**
Christopher Marriott	Group Leader	Q/A
Aleksandra Badyda	HTML/CSS	Group Leader
Sam Jackson	Java Script/Server to User Engineer	HTML/CSS
Tux Lloyd	Server side	Java Script/Server to User Engineer
Silhab Csoma	Java	Server side
Dave Clark	Server side Documenter	Java
Tom Reed	Q/A	Java Documenter
Matt Whitmore	HTML Documenter	Server side Documenter
Mike Steel	Java Documenter	HTML Documenter

REFERENCES

[1] *N/A*

DOCUMENT HISTORY

Version	CCF No.	Date	Changes made to Document	Changed by
1.0	N/A	2012-10-31	Initial creation	CPM4
1.1	N/A	2012-11-2	Added information from Mike	CPM4
1.2	N/A	2012-12-5	Updated config ref and added other documents	CPM4
1.3	N/A	2012-12-6	Added missing data and fixed few mistakes	CPM4