

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: Finalised

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

©Aberystwyth University 2013

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:

- 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

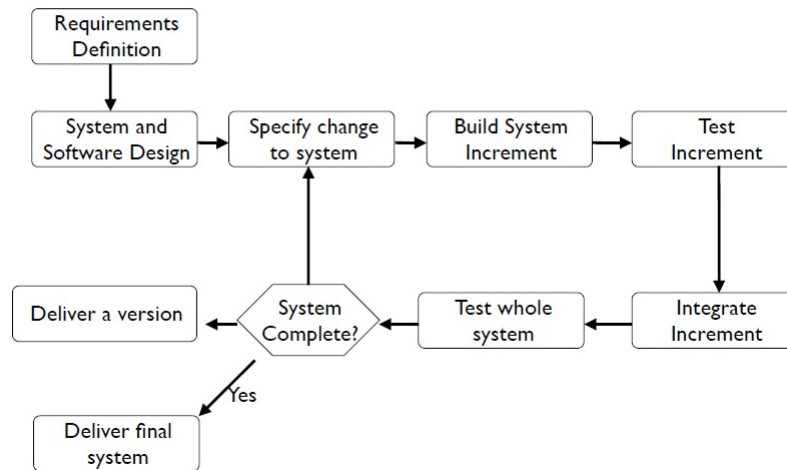
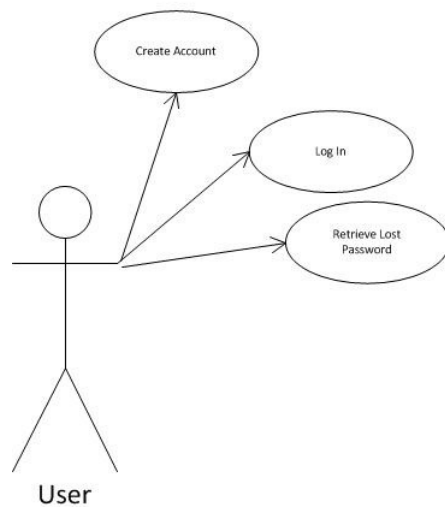


Fig1

5 Use Case

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

5.1 Welcome Page

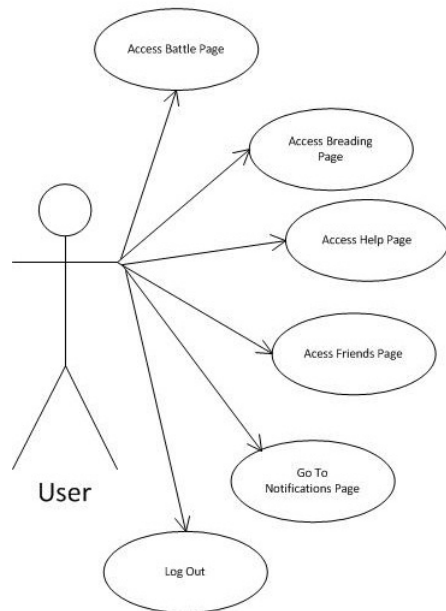


Functional Requirements- FR1 - Server-based Authentication FR6 - Client Options
FR7 - Start-up of software in browser

This is the screen you first see when accessing the website

Example of usage: You create an account, you then log in using your details.

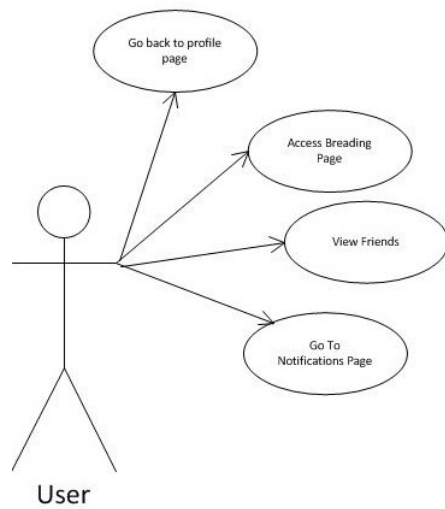
5.2 Main profile page



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

From this view you can access all the different pages
Example usage: You navigate to your friends page

5.3 Friends Page

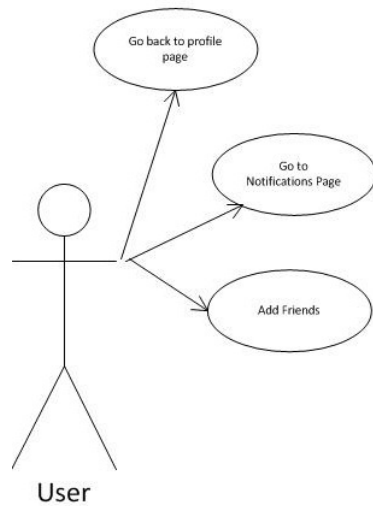


Functional Requirements- FR2 - Server Friends List FR5 - Server-server communication FR6 - Client Options FR9 - Friend Matching FR11- Friends Rich List

From here you can view your friends or go back navigate to other pages

Example usage: You view your friends you then decide to navigate to your notifications page

5.4 Add Friends

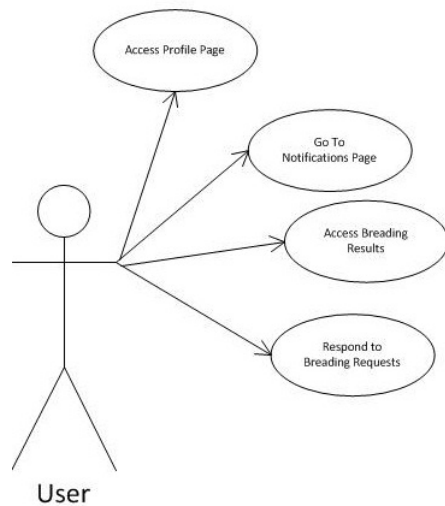


Functional Requirements- FR5 - Server-server communication FR6 - Client Options
FR9 - Friend Matching

From here you can add friends

Example usage: You add a friend by entering their email address they use with their account.

5.5 Breeding

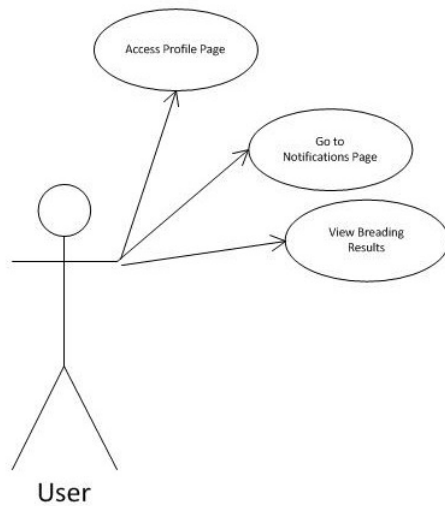


Functional Requirements- FR3 - Server Monster List FR5 - Server-server communication FR6 - Client Options FR8 - Game Display In Browser

From here you can see navigate to the results of any breeding you have done or respond to any breeding requests you might have.

Example request: Yo see a friend has requested a monster to breed with you, you accept.

5.6 Breeding Results

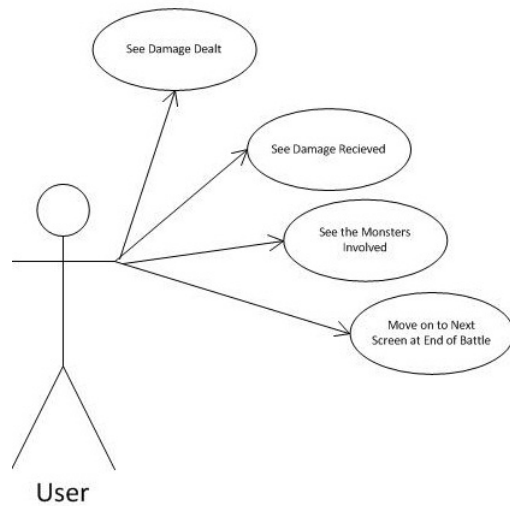


Functional Requirements- FR5 - Server-server communication FR6 - Client Options
FR8 - Game Display In Browser

From here you view breeding results

Example usage: you have recently responded to an request you check back to see the results

5.7 Battle Screen

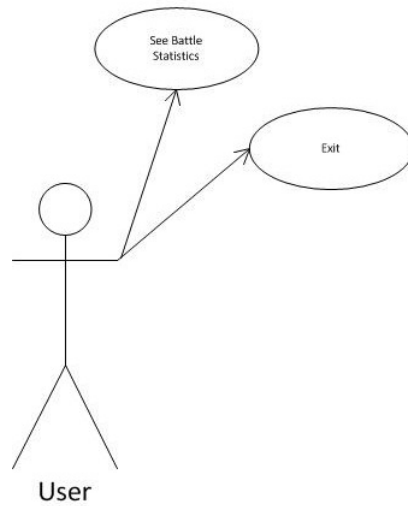


Functional Requirements- FR4 - Server Monster mash management FR5 - Server-server communication FR6 - Client Options FR8 - Game Display In Browser

From here you view information about a battle taking place

Example usage: you have accepted a battle and now want to see how much damage was dealt.

5.8 Battle Results Page



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

From here you can see all the stats from the battle

Example usage: you have a battle and wants to see the stats