Project Plan

Monster Mash

Author: Samuel Sherar <sbs1>

Config Ref: SE\_N15\_PLAN\_01

Date: 23/10/2012

Version: V1.4

Status: Draft

Department of Computer Science

Aberystwyth University

Aberystwyth

Ceredigion

SY23 3DB

Copyright © Aberystwyth University 2012

# Table of Contents

[Table of Contents](#h.4ofzo5umpqfu)

[Introduction](#h.yd5r5colta19)

[User Interface Design](#h.7ce9lfwv7cvp)

[Gantt Chart](#h.t9m40x2wgmqi)

[Risk Analysis](#h.lc7rt17s0kp6)

[References](#h.r9caw6xnn9t4)

[Change log](#h.8xcujg4vfdze)

# Introduction

In this document, we will go over the the plan for for the project, on both a time and user perspective. We have included basic user interaction design, including use case diagrams, and also the decisions we have made in regards to the project

# Overview

For the project we have decided to use Glassfish for the server interaction as it is more flexible compared to Google App Engine (GAE) in regards to data persistence and it means that the end users don’t have to register for a google account when authenticating with our application.

For the Version Control System, we have gone with Git over SVN, as one of our developers know how to use it and is able to teach the rest of the group how to use. Our repository is hosted on [www.github.com](http://www.github.com) under a private account, so we can control who can access.

We are also using Microsoft Word for documentation and Microsoft Visio for diagrams.

**Use Case Diagrams**

# 

# User Interface Design

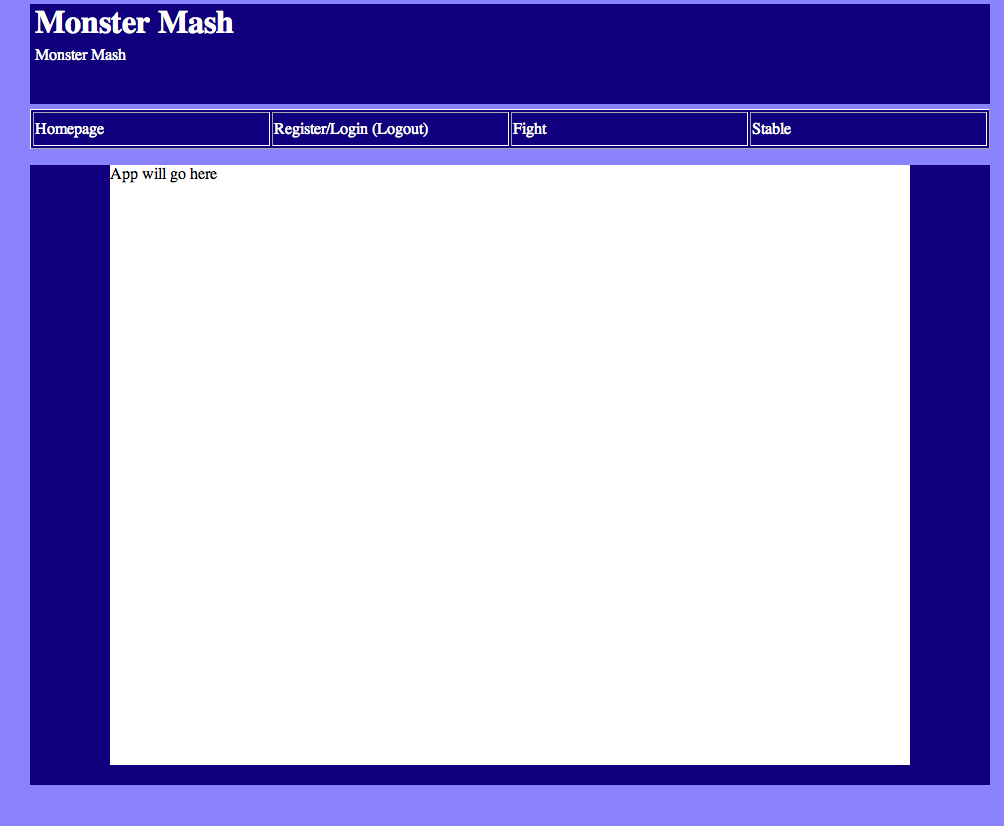


Fig.1 - Main page

This is our main page for the project. Here we have 4 links: Homepage, Login/Register/Logout, Fight and Stable. When the user comes to the main page to start with, they only can interact with two of the links (Homeage and Register/login). When the user has been authenticated Register/login chanes to logout, and two links appear for fighting monsters and viewing the stable

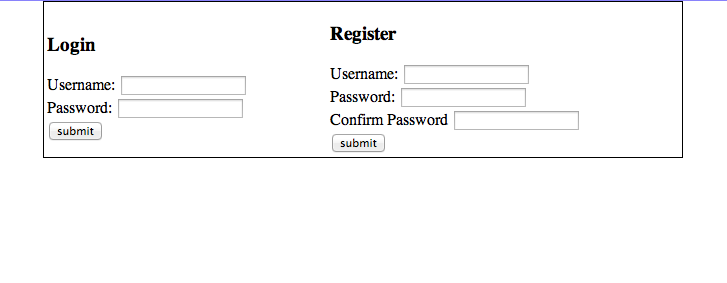


Fig 2 - Registering and Logging in

Here is the form for the user to both login and register. It is clearly marked out so the user can distinguish between both forms.

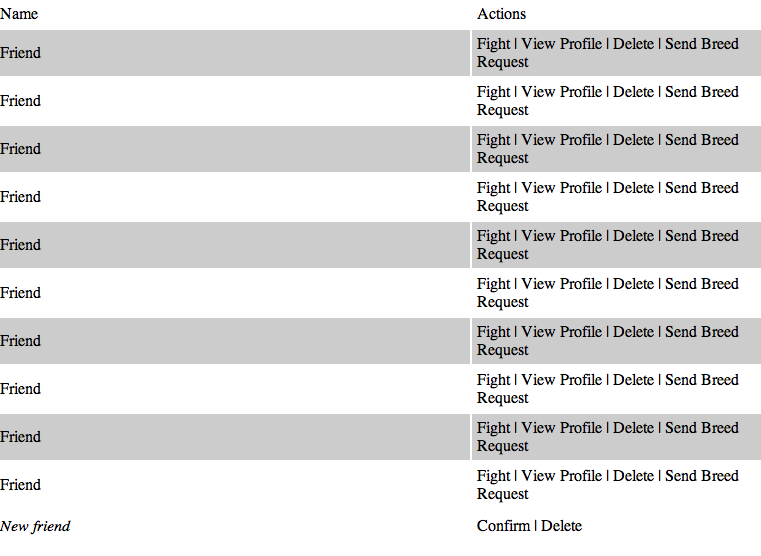


Fig 3 - Friends list

Here is the list of friends which the user may have - both confirmed and requests. With confirmed friends, there are more interactions available, such as fighting and sending breed requests. Unconfirmed friends are in italics at the bottom with only an option to confirm or delete.

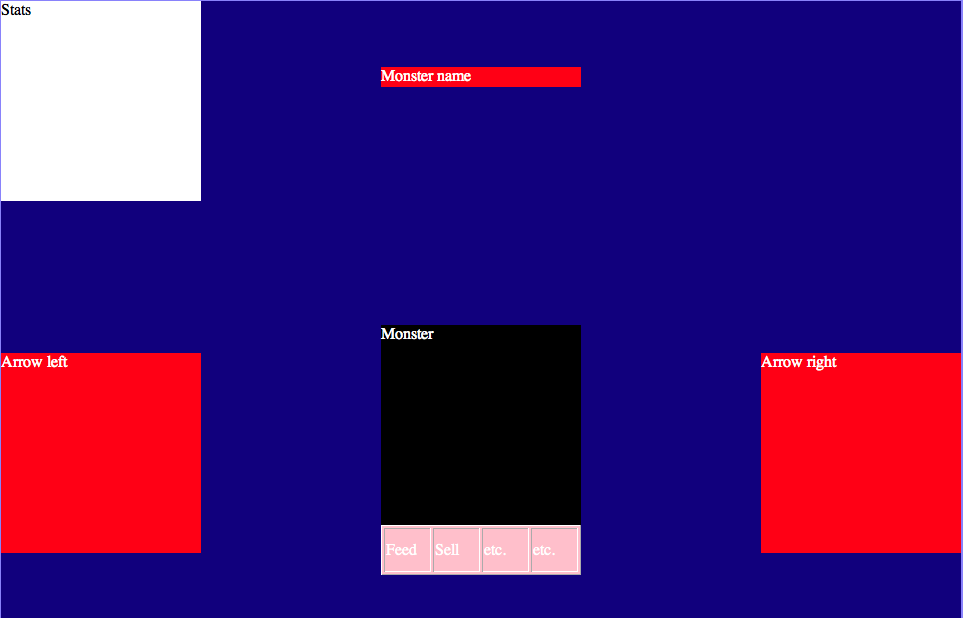


Fig 4 - Monster Information

We will be designing the stable and monster information into more of a console game, so we have the ability to scroll 1 by 1 back and forward between all of our monster. We will also give access to a list. There will be certain actions the user can do with the monster, such as feed and sell.

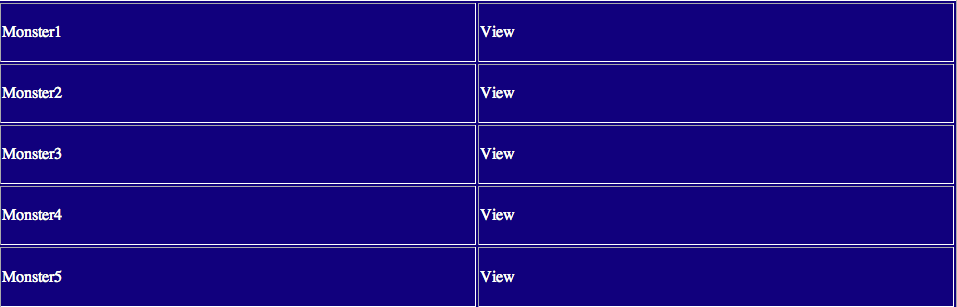


Fig 5 - Stable list

As well as the monster information screen, we have a list with certain quick actions on there, such as view the monster.

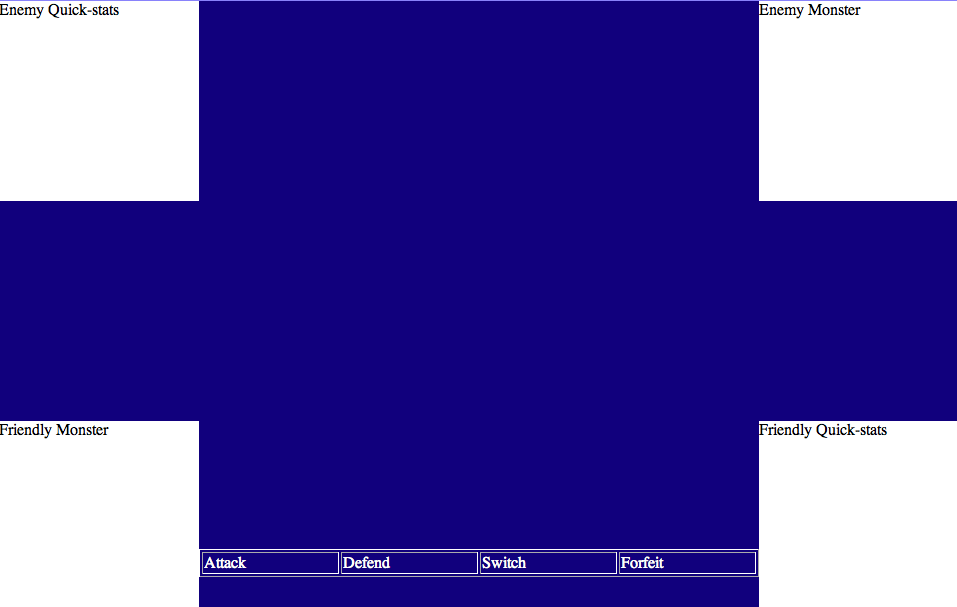


Fig 6 - Fight screen

We have created the interface as simply as we can, with only a certain amount of actions the user can perform during the fight.

# Gantt Chart

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Predecessors | Task Owner | Start Date | End Date | Description | Additional members |
| SE\_N15\_PLAN\_01 | PLAN\_02 - 05 | Sam Sherar | 09/10/2012 | 01/11/2012 | Writing the Overall Document |  |
| SE\_N15\_PLAN\_02 |  | Cellen Williams | 09/10/2012 | 01/11/2012 | Creating Use Case diagrams |  |
| SE\_N15\_PLAN\_03 |  | Andy Watkins | 09/10/2012 | 01/11/2012 | User Interaction design |  |
| SE\_N15\_PLAN\_04 |  | Charlie Bird | 09/10/2012 | 01/11/2012 | Risk Analysis |  |
| SE\_N15\_PLAN\_05 |  | Sam Sherar | 09/10/2012 | 01/11/2012 | Gantt Chart |  |
| SE\_N15\_TEST\_01 |  | Grant David | 28/10/2012 | 16/11/2012 | Writing the Test Document |  |
| SE\_N15\_TEST\_02 | TEST\_01 | Grant David | 01/12/2012 | 08/01/2013 | Testing Prototypes | Imran Mungul |
| SE\_N15\_TEST\_03 | TEST\_02 | Grant David | 29/01/2012 | 04/02/2013 | Testing Final Product | Imran Mungul |
| SE\_N15\_DS\_01 | DS\_02 - 05 | Charlie Bird | 02/11/2012 | 11/12/2012 | Writing the Overall Document |  |
| SE\_N15\_DS\_02 |  | Imran Mungul | 02/11/2012 | 16/11/2012 | Decomposition Description |  |
| SE\_N15\_DS\_03 |  | Andy Watkins | 02/11/2012 | 16/11/2012 | Dependancy Description |  |
| SE\_N15\_DS\_04 |  | Cellen Williams | 02/11/2012 | 16/11/2012 | Interface Description |  |
| SE\_N15\_DS\_05 |  | Fiona Samy | 02/11/2012 | 16/11/2012 | Detailed Design |  |
| SE\_N15\_CODE\_01 | CODE\_02 - 04 | Kamil Mrowic | 02/11/2012 | 01/12/2012 | Prototypes |  |
| SE\_N15\_CODE\_02 |  | Kamil Mrowic | 02/11/2012 | 01/12/2012 | Server-Server Interaction |  |
| SE\_N15\_CODE\_03 |  | Andy Watkins | 17/11/2012 | 01/12/2012 | User Interface |  |
| SE\_N15\_CODE\_04 |  | Sam Sherar | 02/11/2012 | 01/12/2012 | Data Persistance and Modelling |  |
| SE\_N15\_CODE\_05 | CODE\_01 | Kamil Mrowic | 01/12/2012 | 05/02/2013 | Developing Final Product | Sam Sherar, Andy Watkins, Cellen Williams |
| SE\_N15\_MAN\_01 |  | Fiona Samy | 29/01/2013 | 10/02/2013 | Writing User Manual |  |
| SE\_N15\_MAN\_02 |  | Imran Mungul | 29/01/2013 | 10/02/2013 | Writing Technical Manual |  |
| SE\_N15\_FR\_01 |  | Sam Sherar | 06/02/2012 | 16/02/2012 | Writing Final Report | Everyone |
| SE\_N15\_FR\_02 | PLAN | Andy Watkins | 06/02/2012 | 16/02/2012 | Finalising Project Plan document |  |
| SE\_N15\_FR\_03 | TEST | Grant David | 06/02/2012 | 16/02/2012 | Finalising Test Document |  |
| SE\_N15\_FR\_04 | MAN | Fiona Samy | 10/02/2012 | 16/02/2012 | Finalising User manual |  |
| SE\_N15\_FR\_05 | DS | Imran Mungul | 10/02/2012 | 16/02/2012 | Finalising Design Specification document |  |

# Risk Analysis

1. Group-to-Group coordination – if not organised and handled correctly this could easily

generate a lot of wasted time. If groups don't agree on common protocols it could become

very awkward to program the server-to-server part of the project. This should be allowed for

when time is being allocated.

2. Slippage – If slippage due to certain parts of the project occurs then other members should

be drafted in to help speed up that part of the project.

3. Illness – If a group member is ill, they should notify the project leader and have their current

assignment reassigned if possible, if not then they should be included in other assignments

once they are well again in order to make up for the lost time.

4. Complicated algorithms – if the individual assigned to create an algorithm is struggling with

it the more members of the group can be assigned to reduce time loss.

5. Authentication – should be kept seperate from the other groups projects in order to maintain

security.

6. User interface – needs to be suitable for a primary/secondary school audience, meaning it

must be simple to understand and use. If the audience can't understand how to use it, it's

unlikely to be popular.

7. Git – Whilst git is a very useful program, it is very important that space is managed as

everyone having a copy of the entire repository could take up quite a lot of space.

8. Data Protection – as the application will is aimed at a school audience it is especially

important that data is kept secure, especially if they are to input personal details.

# References

# Change log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | CCF Number | Date | Changes | User |
| V1.0 | N/A | 23/10/12 | First draft of the document | sbs1 |
| V1.1 | N/A | 25/10/2012 | User Interface | ajw14 |
| V1.2 | N/A | 26/10/2012 | Gantt Chart | sbs1 |
| V1.3 | N/A | 26/10/2012 | Use Case Diagrams | cew1 |
| V1.4 | N/A | 1/11/2012 | Risk Analysis | cab27 |
| V1.5 | N/A | 1/11/2012 | Overview and introduction | sbs1 |
|  |  |  |  |  |