Software Engineering Group Project 15
Project Plan Document

Authors: Samuel Sherar,
Cellen Williams
Date: 14/02/2013
Version: 1.6
Status: Final

Department of Computer Science
Aberystwyth University
Aberystwyth
Ceredigion
SY23 3DB

Table of Contents

Purpose of this Document

The purpose of this document is to clarify and define the project mandate. It will state the possible constraints that the group could face in regards to producing the Monster Mash game. The processes the group will have to commit will also be defined. This document will have to stay within the requirements and specification of the QA documents.

Scope

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

Objectives

This document's objective is to show the reader what materials we used to produce our Monster mash game.

- Use-case diagrams
- User interface design
- Gantt chart
- Risk Analysis

All of the above are material that will be explained in this document with the necessary screenshots and information stated.

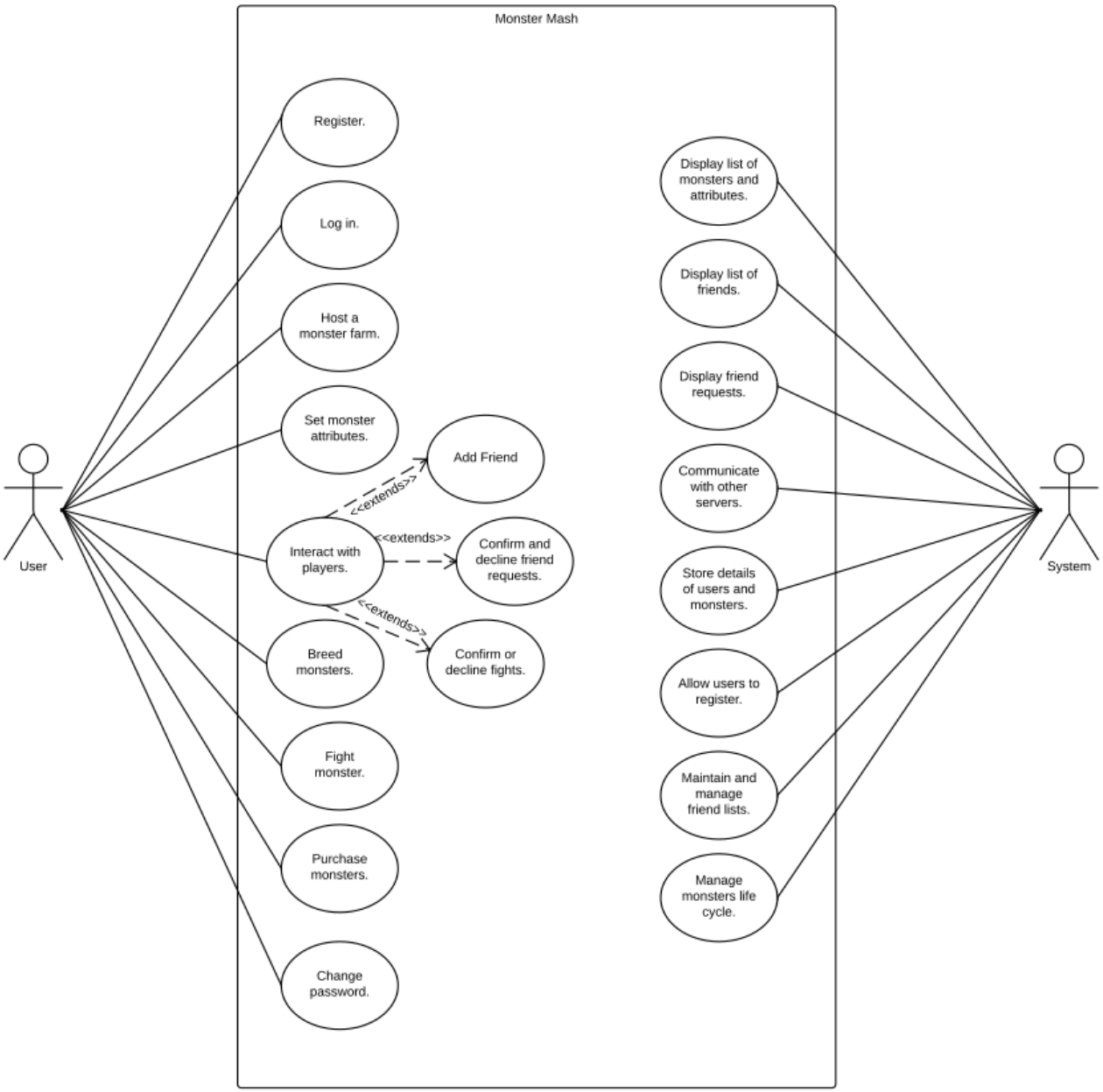
Overview

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 under a private account, so we can control who can access it.

We are using Microsoft Word for documentation and Microsoft Visio / Lucid Chart for diagrams.

Use Case Diagrams



Here is the Use-Case diagram, clearly showing what the user and system should be able to do. The user needs to be able to:

- Register or log in
- Host a monster farm

- Set monster attributes
- Interact with other players
- Breed, fight and purchase monsters
- Change their password

The system needs to be able to:

- Display:
  - A list of monsters and attributes
  - A list of friends
  - Friend requests
- Communicate with other servers
- Store details of users and monsters
- Allow users to register
- Maintain and manage friend lists
- Manage monsters lifecycle

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 begin with they only can interact with two of the links (Homepage and Register/login). When the user has been authenticated Register/login changes to logout, and two links appear to allow the user to fight monsters and view 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.

Name	Actions
Friend	Fight | View Profile | Delete | Send Breed Request
Friend	Fight | View Profile | Delete | Send Breed Request
Friend	Fight | View Profile | Delete | Send Breed Request
Friend	Fight | View Profile | Delete | Send Breed Request
Friend	Fight | View Profile | Delete | Send Breed Request
Friend	Fight | View Profile | Delete | Send Breed Request
Friend	Fight | View Profile | Delete | Send Breed Request
Friend	Fight | View Profile | Delete | Send Breed Request
Friend	Fight | View Profile | Delete | Send Breed Request
Friend	Fight | View Profile | Delete | Send Breed Request
New friend	Confirm | Delete

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.

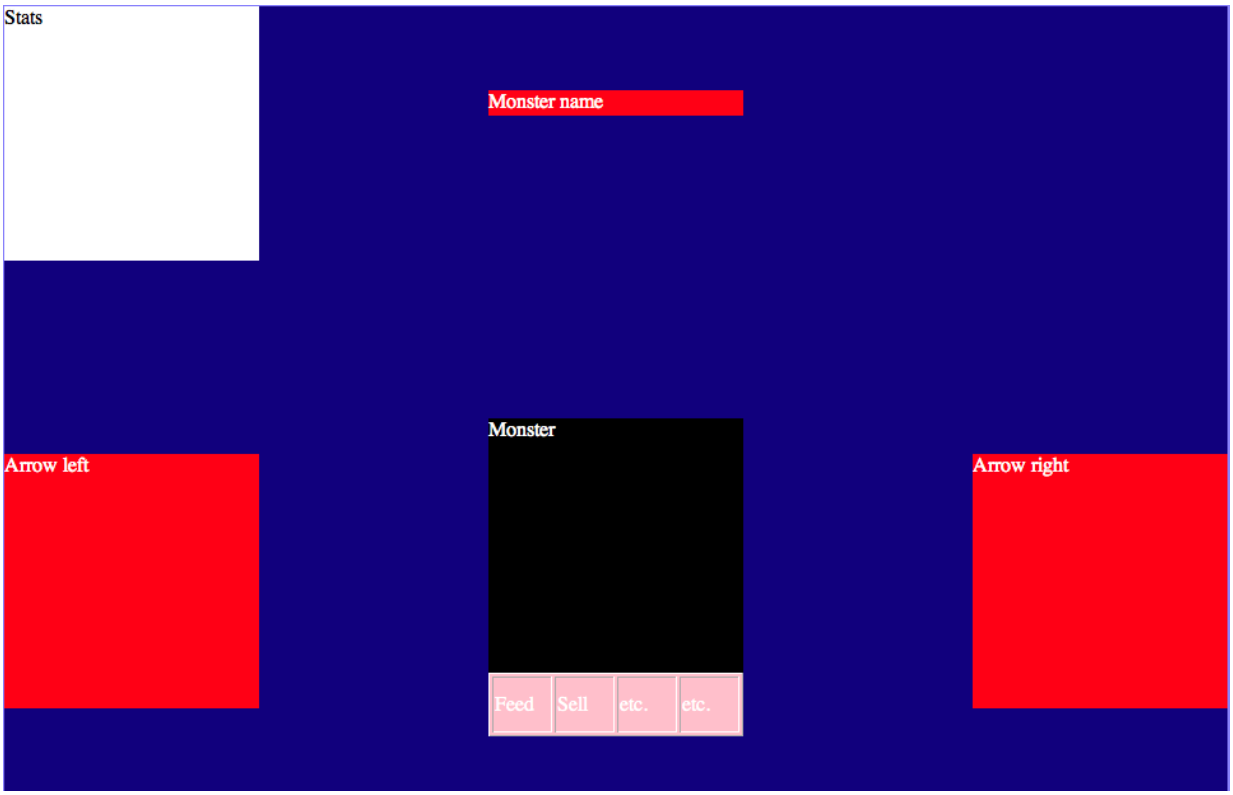


Fig 4 - Monster Information

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.

Monster1	View
Monster2	View
Monster3	View
Monster4	View
Monster5	View

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.

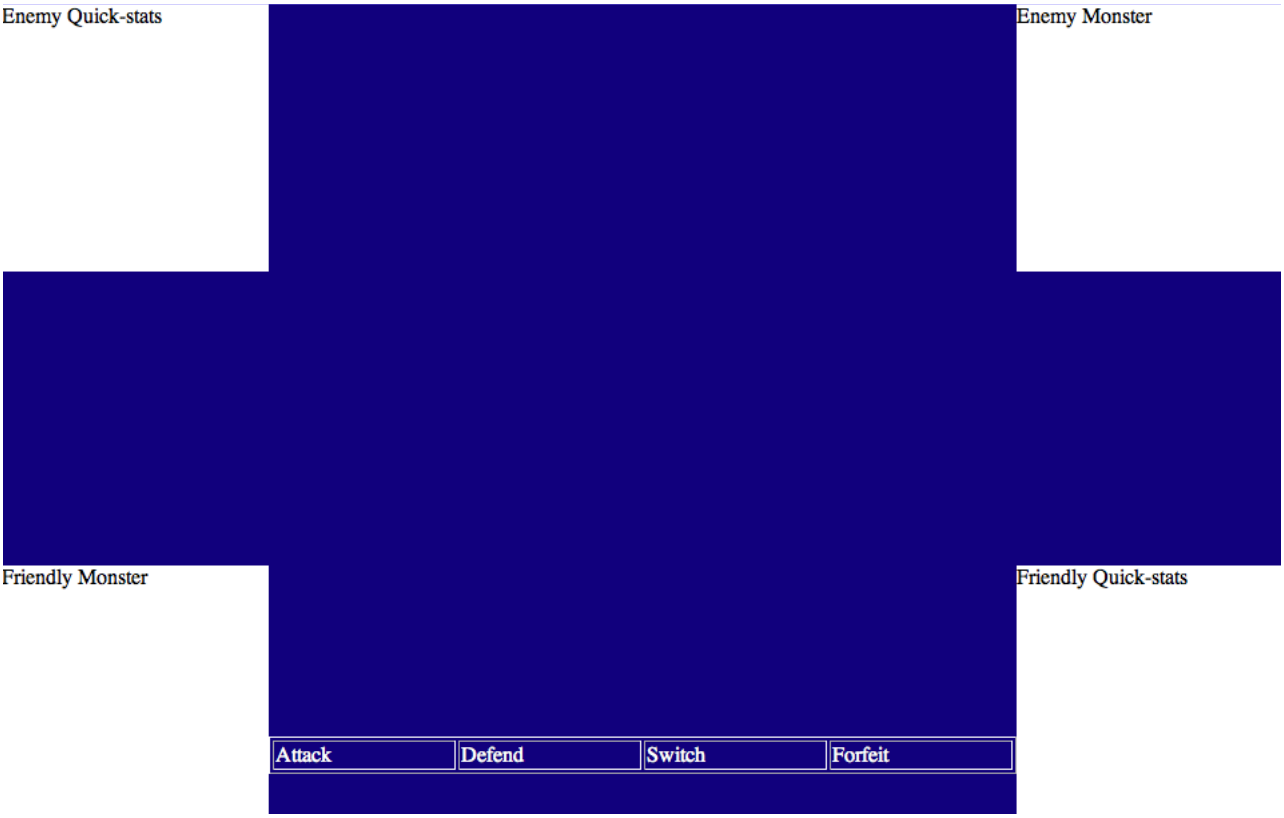


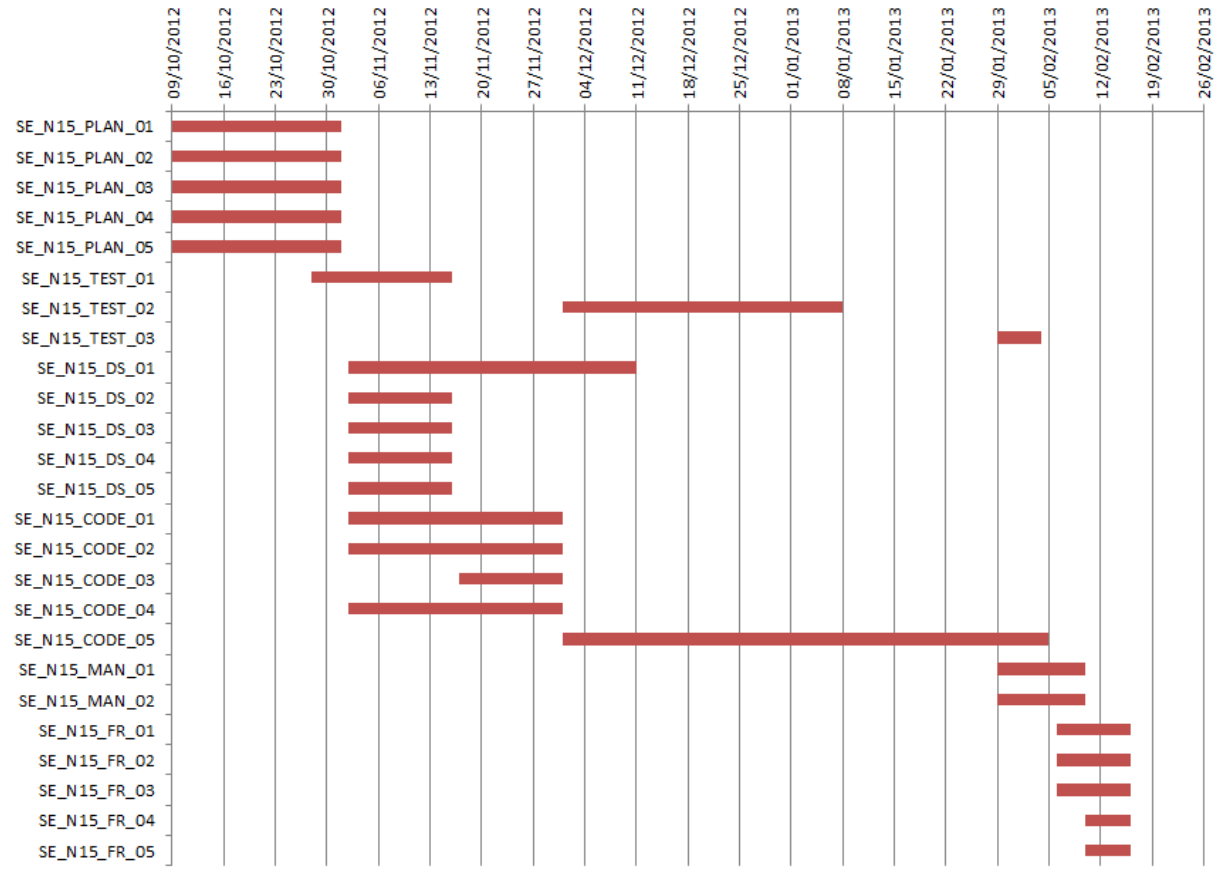
Fig 6 - Fight screen

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		Cellan 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	29/01/2013	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		Cellan 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_COD	CODE_02 - 04	Kamil Mrowic	02/11/2012	01/12/2012	Prototypes	

E_01						
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, Cellan 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	06/02/2013	16/02/2013	Writing Final Report	Everyone
SE_N15_FR_02	PLAN	Andy Watkins	06/02/2013	16/02/2013	Finalising Project Plan document	
SE_N15_FR_03	TEST	Grant David	06/02/2013	16/02/2013	Finalising Test Document	
SE_N15_FR_04	MAN	Fiona Samy	10/02/2013	16/02/2013	Finalising User manual	
SE_N15_FR_05	DS	Imran Mungul	10/02/2013	16/02/2013	Finalising Design Specification document	


09/10/2012
16/10/2012
23/10/2012
30/10/2012
06/11/2012
13/11/2012
20/11/2012
27/11/2012
04/12/2012
11/12/2012
18/12/2012
25/12/2012
01/01/2013
08/01/2013
15/01/2013
22/01/2013
29/01/2013
05/02/2013
12/02/2013
19/02/2013
26/02/2013
SE_N15_PLAN_01
SE_N15_PLAN_02
SE_N15_PLAN_03
SE_N15_PLAN_04
SE_N15_PLAN_05
SE_N15_TEST_01
SE_N15_TEST_02
SE_N15_TEST_03
SE_N15_DS_01
SE_N15_DS_02
SE_N15_DS_03
SE_N15_DS_04
SE_N15_DS_05
SE_N15_CODE_01
SE_N15_CODE_02
SE_N15_CODE_03
SE_N15_CODE_04
SE_N15_CODE_05
SE_N15_MAN_01
SE_N15_MAN_02
SE_N15_FR_01
SE_N15_FR_02
SE_N15_FR_03
SE_N15_FR_04
SE_N15_FR_05

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 caused by 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 to 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, more members of the group can be assigned to reduce time loss.
5. Authentication – should be kept separate from the other group 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.

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	cew10
V1.4	N/A	1/11/2012	Risk Analysis	cab27
V1.5	N/A	1/11/2012	Overview and introduction	sbs1
V1.6	N/A	19/01/2013	Added Use-Case diagram description Proofread and corrected mistakes.	cew10