**CS22120 Software Development Life Cycle**

**Group 05 Final Report**

**Author:** Chris Savill – chs17

**Config Ref:** SE.05.DS Group 5

**Date:** 28th January 2013

**Version:** 1.0

**Status:** Draft

Group 05

Aberystwyth University

Aberystwyth

Ceredigion

SY23 3DB

Copyright © Aberystwyth University 2012

Table of Contents

**Project Plan:**

**Test Specification:**

**Design Specification:**

**End-Of-Project Report:**

[Management Summary 8](#_Toc347222418)

[Historical Account of Project 8](#_Toc347222419)

[Performance of Team 8](#_Toc347222420)

[Critical Evaluation of Team and Project 8](#_Toc347222421)

**Appendices:**

[Requirements 9](#_Toc347222422)

[Test Report 10](#_Toc347222423)

[Test table 10](#_Toc347222424)

[Failed tests 11](#_Toc347222425)

[Maintenance Manual 11](#_Toc347222426)

[Program description 11](#_Toc347222427)

[Program structure 11](#_Toc347222428)

[Algorithms 11](#_Toc347222429)

[Main data areas 11](#_Toc347222430)

[Files 11](#_Toc347222431)

[Interfaces 11](#_Toc347222432)

[Suggestions for improvements 11](#_Toc347222433)

[Things to watch when making changes 11](#_Toc347222434)

[Physical limitations of program 11](#_Toc347222435)

[Rebuilding and testing 11](#_Toc347222436)

[Personal Reflective Reports 12](#_Toc347222437)

[Chris Savill – chs17 12](#_Toc347222438)

[Richard Gray – rig6 13](#_Toc347222439)

[Edward Davies – edd14 14](#_Toc347222440)

[Sam Morrison – sjm16 15](#_Toc347222441)

[Jacob Smith – jas32 16](#_Toc347222442)

[Ivan Cholakov – ivc 17](#_Toc347222443)

[Katherine Rose Farmer – krf 18](#_Toc347222444)

[Oliver Roe – olr1 19](#_Toc347222445)

Project Plan

Test Specification

Design Specification

End-Of-Project Report

# Management Summary

# Historical Account of Project

# Performance of Team

# Critical Evaluation of Team and Project

Appendices

# Requirements

**Functional Requirements**

**FR1 Server-based authentication**

The server will be used to authenticate a user, allowing them to log-in or register from their browser.

**FR2 Server friends list**

The server will maintain a list of friends for each user. Users will only be able to interact directly with their friends. Friends will be identified by their email address and added by a request-confirm mechanism.

**FR3 Server monster list**

The server will maintain a list of the monsters owned by each player and their attributes. These include genetic attributes and phenotypic attributes (such as age, health etc). The server will manage the monster lifecycle i.e. mating, birth, ageing, illness, injury and death. New users should be allocated a basic (random) monster and a small pot of virtual money.

**FR4 Server monster fights**

The server will handle monster fights with a (virtual) cash prize available. The system will provide a fixed value prize to the winner. Users can select one of their monsters and challenge one of their friend’s monsters to a match. The friend can accept or decline the challenge. If they accept, the server will decide the winner based on the characteristics of the monsters along with an element of random chance (see Appendix A for an outline suggested algorithm). The server “pays” the winner the prize value and the loser’s monster should die.

**FR5 Server-server communication**

The server should be able to communicate with other servers using a standard protocol (agreed between groups) in order to play the game (add friends, buy/sell monsters, arrange monster breeding, manage fights, etc).

**FR6 Client options**

The client will allow users to interact with the system i.e. register/unregister, add/remove friends, offer for sale/buy monsters, offer for breeding / purchase breeding, etc. The sale and breeding of monsters will be managed in a similar way. If a user wishes to offer a monster for sale or breeding they can assign a value to the monster.

Any of their friends can view the monster’s price and purchase it or hire it for breeding. When purchasing the monster is transferred to the purchaser, when breeding the offspring are transferred to the purchaser. In both cases the sale price is transferred from the buyer to the seller. If the buyer does not have sufficient funds the transaction should not take place.

**FR7 Startup of software in browser**

When the software first starts, it will display a set of choices for the user as follows:

• Log in

• Create new account

Once logged in the system should provide an option to log-out. This will take the user back to the initial log-in/register screen.

**FR8 Game display in browser**

When the player has logged in they should be able to see a list of their monsters (with status info), their friends (with offers of monsters for sale and for breeding), challenge requests (with prize money etc) and have options to interact with these options as described in FR6.

**FR9 Friend matching**

The system should allow users to send a friend request to other users of the system (identified e.g. by their email) and to accept or reject requests sent to them. On accept the friend would be added to the friend list.

**FR10 Fight notifications**

Following a fight that the user has entered, the monster lists off all competitors should be updated. Loser’s monsters should be removed from their list, the winner will have the prize money added to his account and the monster’s status will be updated (accounting for injuries etc).

**FR11 Friends rich list**

A user should be able to see a list of his friends (including himself) and the wealth of each, ordered by wealth.

**External Interface Requirements**

**EIR1 Appearance of Interface**

The program should conform to usual look and feel guidelines for web-based applications.

**Performance Requirements**

**PR1 Response of program to user input**

The user should feel like the system is responding to them at all times during game play. There should not be any perceptible lag between attempting a game action and the system responding.

**PR2 Target computer for system**

The client software produced should run correctly on standard browsers (i.e. one of the browsers installed on the IS desktop). The servers should also run either on the Department’s or University’s systems or a third-party system, but should be accessible from the department for testing.

# Test Report

## Test table

## Failed tests

# Maintenance Manual

## Program description

## Program structure

## Algorithms

## Main data areas

## Files

## Interfaces

## Suggestions for improvements

## Things to watch when making changes

## Physical limitations of program

## Rebuilding and testing

# Personal Reflective Reports

## Chris Savill – chs17

## Richard Gray – rig6

## Edward Davies – edd14

## Sam Morrison – sjm16

## Jacob Smith – jas32

## Ivan Cholakov – ivc

## Katherine Rose Farmer – krf

## Oliver Roe – olr1