Software Engineering Group Project
Test Specification

Author:	Group 12
Config Ref:	SE_G12_TS_00
Date:	2012-11-15
Version:	1.0
Status:	release

Department of Computer Science
Aberystwyth University
Aberystwyth
Ceredigion
SY23 3DB

CONTENTS

1. INTRODUCTION

2. PURPOSE OF THIS DOCUMENT

This document shows the test table for our group's monster mash game, for use when we begin designing our code, taking these into account. It is in the context of being viewable to anyone who needs to test the game, and especially the coders of the game. It should be taken into account when testing the game, and when designing the game, since it contains all of the tests for the Functional Requirements.

2.1 Scope

This test table document describes the Functional Requirements that the game will be required to meet, both server and client side, these are the tests that will be used to determine the functionality and the extent to which our project conforms to the function requirements as set out in the requirements specification [1].

2.2 Objectives

- To allow the group to understand the functionality of the game, through the FRs
- To provide information to possible testers of the prototypes
- To give the coders more information on what should be included.

3. TEST TABLES

Test Ref	Requirement being tested	Test Content	Input	Output	Pass Criteria
SE_TT_001	FR1	Check that a user can register.	Valid data for the registration form, email, user, and password, such as; "sis13@aber.ac.uk" as email, "secret" as passwords, and "sis13" as username.	User should be redirected to the main page, and the new user should be created in the database.	New user is created in DB, and user is redirected to main page. The information in the database is the same as that entered in the form
SE_TT_002	FR1	If the password given in the confirmation is different do not create a new user.	"sis13@aber.ac.uk" as email, "foo" as password, and "bar" as confirmation, and "sis13" as username.	The user should be shown an error saying that the two passwords are not the same.	The user is shown an error and no new user is created in the DB.

SE_TT_003	FR1	See if email is registered already	“sis13@aber.ac.uk” as email, “secret” as passwords, and “sis13” as username.	The user should be shown an error saying that email is in use, and link to recover a password.	Error is shown, and no new user is created in the DB.
SE_TT_004	FR1	See if user can login with valid login information	“sis13@aber.ac.uk” as email, “secret” as password	User should get a session and be redirected to mainpage.	The user is given a valid session and is redirected to the mainpage.
SE_TT_005	FR1	Invalid login information should be rejected.	“sis13@aber.ac.uk” as user and “wrongpassw” as password.	The user should see an error saying that his login information is incorrect.	The user is shown an error and no valid session is created for the user
SE_TT_006	FR1	Check that a user can register.	Valid data for the registration form, email, user, and password	User should be redirected to the main page, and the new user should be created in the DB.	New user is created in DB, and user is redirected to main page.
SE_TT_007	FR2	User should see list of their friends upon login.	Log in details. lwv@aber.ac.uk as the username. 1234pass as the password. Also accounts for the friends are entered into the DB.	List of users friends: John Paul.	Shows List of friends.
SE_TT_008	FR2	User should be able to send a friend request by entering an email address.	Enter email: yap@aber.ac.uk, where there is a an account registered for this email.	Tell the User that a friend request was sent successfully.	Message saying request sent.
SE_TT_009	FR2	Check that user had entered a vald email address.	Enter a invalid email: lwv@@dg.cffriu.	Error message warns the user they must enter a valid email.	Warning of invalid email.
SE_TT_010	FR2	Check the user has entered a	Enter a email not presently used by a user:	Error message saying that no such user	Warning that no such user exists.

		users email that exists on a Monster Mash Server.	llion@me.com.	exists.	
SE_TT_011	FR3	Check that new window will appear when user hovers mouse over monsters in the monster list.	Hover mouse over single monster from the monster list.	Small window with all the monster's attributes appears.	All the monster attributes are visible to the user in new window.
SE_TT_012	FR3	When user creates new account, it receives a random monster and small amount of money.	User creates new account.	New account with basic monster and small amount of money is created.	New user has basic monster and small pot of virtual money attached to their account.
SE_TT_013	FR3	Check that there is some probability that monster can be injured or ill.	Leave server running with some random account for x time.	After x time it is possible that monster is ill or injured. This will be reflected in the monsters attributes.	Monster being ill is added to the attributes and shown to a user.
SE_TT_014	FR3	Check that monster is getting older while server is running.	Leave server running with some random account for x time.	After x time monster is older.	Monster is older and the age attributes is increased and shown in the attributes box.
SE_TT_015	FR3	Check that very old or ill monster can die.	Leave server running for a long time with random account.	After x time monster will die, because it was very old or ill.	Monster dies and is removed from the list, an update message is sent to the user.
SE_TT_016	FR3	Check that there is some probability that monster gets better	Leave server running with some random account for x time.	After x time monster gets better.	Monster gets better, illness is no-longer displayed as part of its

		after illness.			attributes.
SE_TT_017	FR4	To test that a user has the ability to "monster mash" (battle other users with their monsters.	Clicking on a friend and then choosing to battle them.	Some data about the battle should be given out.	That correct data about the battle is sent to both players
SE_TT_018	FR4	To test whether the system accurately assigns the monetary value to monsters and that this is distributed correctly after battle.	Getting two monsters of known value to fight.	The distribution of money to the winning player.	The correct amount of money is added to the winners total.
SE_TT_019	FR4	To test whether a user can send battle requests.	Clicking on another player and choosing to send them a request.	The request received by the other player.	The second player receives the notification and correct data about being challenged.
SE_TT_020	FR4	To test that our 'battle algorithm' is working and produces the expected results.	Conducting multiple battles with known attributes, so that the likely outcome is known.	The battle logs.	That although there is an element of chance involved the algorithm is carried out.
SE_TT_021	FR4	To test whether an element of chance is included in the algorithm.	Conducting many battles with monsters of known attributes.	The battle logs.	That in one of the battles a monster with lower attributes that should have lost does in fact win.
SE_TT_022	FR5	User enters an email not registered on	An email that is not registered such as	A message saying the user could not	A message saying that the user is not

		any servers an error should be shown.	"friend@server.com" as email.	be found.	found and no request is stored.
SE_TT_023	FR5	Whether a user can buy a monster listed on another server.	User buys a monster stored on another server.	Monster has been bought message.	Message that monster has been bought and monster is now stored under the user locally, and users founds are changed.
SE_TT_024	FR5	User cannot buy monsters without the right amount of money.	User buys a monster stored on another server	Error message saying the user has insufficient funds is shown	The buy request is not sent and user is shown an error message.
SE_TT_025	FR5	User can sell monster to friends on other servers.	Friends on a different server buys a monster from our test user	A notification that the monster is sold, to whom and for what.	A notification is shown to the user and the monster is transferred to the other server and removing the local entry.
SE_TT_026	FR5	User's monster is not sold when the request has insufficient funds.	A friend on a different server buys a monster from our test user while having insufficient funds.	Response to the server sending the request with an exception.	The users monster is not sold and exception is sent to the server sending the request.
SE_TT_027	FR5	Users from other servers can request user data from us by passing a valid email.	Remote server requests the data for a user with a valid email.	Response containing the user data is sent.	Response containing the user data is sent.
SE_TT_028	FR5	A user can breed their monster with users on other servers.	User views the mating list of friends and accepts a mating offer with a user on another server.	The local user should receive some funds and a message saying the "breeding" was successful.	Users receives funds, is shown a notification and the results are stored.

SE_TT_029	FR5	User can list their monsters for breeding so users on other servers can breed with them	User lists a monster for breeding, and a user on another server accepts the offer.	Local user should get a notification with the attributes of the child	User receives notification and the child is stored in users monster list.
SE_TT_030	FR5	User can send fight request to friends on other servers.	User selects a friend and sends a fight request.	The user gets a notification that the request is sent.	Request is sent to remote server, and the user gets a notification.
SE_TT_031	FR5	Remote request is accepted and the monsters fight.	A request has been sent to a remote server.	The output should be a message saying if the user has won/lost with new attributes for the user's monster.	User should be shown notification, and the result should be stored correctly in the DB.
SE_TT_032	FR5	User remote fight request is canceled.	A fight request is sent to another user, and then the pending request is canceled.	No battle occurs, and notification is sent to the user.	A notification detailing the challenge and the subsequent withdrawal is sent to both users
SE_TT_033	FR5	User can accept remote fight requests	User clicks the accept button of a remote fight request.	The output should be a message saying if the user has won/lost with new attributes for the user's monster.	User should be shown notification, and the result should be stored correctly in the DB.
SE_TT_034	FR5	User can decline remote fight requests.	User clicks the decline button.	Notification should be removed.	No battle is held and the notifications of the users updated.
SE_TT_035	FR6	Successfully register with correct credentials	Username, e-mail address, password	No error, the user is registered and logged in to the Mainpage.	Data passed correctly between server and database

SE_TT_036	FR6	Reject registration with already registered username	Existing username, e-mail address, password	Error about username already taken	Server rejects adding record to a database after comparing username with existing records
SE_TT_037	FR6	Reject registration with already registered e-mail address	Username, existing e-mail address, password	Error about e-mail address already registered	Server rejects adding record to a database after comparing e-mail with existing records
SE_TT_038	FR6	Reject registration with password not meeting criteria (e.g. too short)	Username, e-mail address, weak password	Error about too weak password	Server rejects adding record to a database after unsuccessful validation of password
SE_TT_039	FR6	Login attempt with correct username and password	Existing username, correct password	No error	Data passed correctly between server and database
SE_TT_040	FR6	Login attempt with incorrect username	Not existing username, any password	Error about incorrect username or password	Data passed correctly between server and database
SE_TT_041	FR6	Login attempt with incorrect password	Existing username, incorrect password	Error about incorrect username and password	Data passed correctly between server and database
SE_TT_042	FR6	Check that a friend request can be sent to existing user	Send a friend request to existing user	User gets the request	Server passes the request
SE_TT_043	FR6	Check that a friend request cannot be sent to not existing user	Send a friend request to not existing user	Error message sent to the user notifying them that the user does not exist.	Server returns an error after not finding the user in the database

SE_TT_044	FR6	Remove friend from friends list	Clicking on a friend and select to remove that friend.	Friend list updated without selected friend.	Server removes the connection
SE_TT_045	FR6	Remove user who has removed you from their friends list is also removed from your friends list.	By having two test users that are friends were one removes the other from their friends list.	The second user's friend list is also updated with the first user removed from their friends list as well.	Server returns an error after not finding the user in the friends list
SE_TT_046	FR6	Accept someone’s request.	Accept request.	User added to friends.	Server adds users to each others friends lists.
SE_TT_047	FR6	Deny someone’s friend request request.	Deny request.	The user is not added to the friend list and the request is removed.	Server cancels the request and returns an appropriate message to user who sent the request
SE_TT_048	FR6	Check that a monster can be offered for sale	Offer monster for sale	Other users see the monster on the market and are able to buy it	Another user is able to see and purchase the monsters the test user has put up for sale.
SE_TT_049	FR6	Check that a monster can be bought	Buy a monster	Monster is transferred to the user who bought it	The test user is able to purchase a monster put up for sale by another test user.
SE_TT_050	FR6	Check that a monster can be offered for breeding	Offer monster for breeding	Other users see the monster on the breeding page and are able to purchase breeding	Another test user is able to see and accept the offer for breeding
SE_TT_051	FR6	Check that a monster can be purchased	Purchase monster for breeding	Clicking and selecting monsters that	The user is able to accept the offers of

		for breeding		are being offered for breeding on the mating screen	other users for breeding
SE_TT_052	FR7	Check that user with correct credentials can be logged in	Username and password of a registered user	User is logged in and redirected to the main game page	Data passed correctly between client and server
SE_TT_053	FR7	Check that user with incorrect credentials cannot be logged in	Incorrect username and / or password	User is redirected to a page displaying "Incorrect username / password" message	Data passed correctly between client and server
SE_TT_054	FR7	Check that a logged in user can log out	Click the "logout" button	User redirected back to login screen	Data passed correctly between client and server
SE_TT_055	FR7	Check that user can register	Fill in the registration form and press "register button"	User is now registered and can log in	Data passed correctly between client and server
SE_TT_056	FR8	Whether the main screen is visible after a user has logged.	Their log in details.	The main screen is displayed.	The main screen is correct displayed with the update panel and side tabs.
SE_TT_057	FR8	Whether the update panel is correctly displayed	Looking at the main page and the update panel	The update panel is displayed	The current updates for that is user is displayed in the center of the screen.
SE_TT_058	FR8	Whether the friends list is displayed	Looking at the main screen	The friends list is displayed to a user	The list of friends is correctly shown at the side of the screen to the user
SE_TT_059	FR8	Whether the monster list is displayed	Looking at the main screen	The monster list is displayed to	The list of the users monsters is displayed at

				the user	the side of the screen.
SE_TT_060	FR8	Whether challenge requests are shown to a user	Looking at the update panel, after a second test user issues a challenge to the user.	The updates on the update panel	The challenge is displayed in the update panel correctly.
SE_TT_061	FR8	Whether a user can interact with the displayed elements of the Mainpage.	Clicking on various aspects of the page and interacting with them such as the monsters or friends.	The update for chosen action occurs such as a battle or friend request	The action chosen by the user occurs and the update panel is updated with this information.
SE_TT_062	FR9	When a user's friend request is accepted the friend should be added to the user's friend list.	Friend request will be sent and needs to be accepted by the friend.	Friend should appear on the users friend list.	Friend appears on users friend list.
SE_TT_063	FR9	User should be able to decline a friend request and the request should be removed.	The user clicks decline on the friend request.	Friend request should disappear and the friend should not appear on the list.	Request disappears and friend is added.
SE_TT_064	FR9	When a users friend request is declined the rejected request should disappear.	Friend request will be sent and needs to be declined by the friend.	Friend shouldn't appear on the users friend list and the request should disappear.	Request disappears and no friend is added.
SE_TT_065	FR10	Whether a friend's monster that was killed, has been removed from their Monster list.	Win a battle against a friend and kill their monster.	Monster should now be removed from the list following the battle.	The monster list for that user is updated in the DB and no longer appears there or is shown to the user.

SE_TT_066	FR10	That a user's monster was killed and has been removed from their monster list.	Lose a battle, and have the monster killed.	The user's monster should now be removed from the user's monster list.	The monster is no longer displayed
SE_TT_067	FR10	Winning a battle will cause the user to gain prize money, and have it added to their account	Win a battle, with prize money.	The user's money should have increased by the prize money amount. Adding this to their previous total.	The user has increased wealth and the display is updated
SE_TT_068	FR10	Update the user's monster after a battle.	Win a battle and receive an injury	Monster should be damaged in some way, following the battle with another monster.	The monsters attributes should be changed to show that it has an injury
SE_TT_069	FR11	User should be able to see a list of their friends (Including themselves) with the wealth of each, ordered by wealth.	User should click on the leaderboard.	List of friends and their wealth should appear in order of wealth.	List of friends in order of wealth.

REFERENCES

[1] **Software Engineering Group Projects Monster Mash Game Requirements Specification**
Config Ref: SE.CS.RS

DOCUMENT HISTORY

Version	*CCF No.*	*Date*	*Changes made to document*	*Changed by*
1.0		16/11/12	Release version of document creates	Jau1