Monster Mash – Team Awesome-er

Test Specification

|  |  |
| --- | --- |
| Author: | Joshua Bird, Phil Wilkinson, Tom Hull, Dave Haenze, Chris Morgan, Kamarus Alimin, Szymon Stec, Lewis Waldron |
| Config Ref: | SE\_02\_DS\_01 |
| Date: | 14-11-2012 |
| Version: | 1.0 |
| Status: | Draft |

Department of Computer Science

Aberystwyth University

Aberystwyth

Ceredigion

SY23 3DB

Copyright © Aberystwyth University 2012

CONTENTS

CONTENTS 2

1. Introduction 3

1.1 Purpose of this Document 3

1.2 Scope 3

1.3 Objectives 3

2. Approach to testing 3

3. Test Specification 4

REFERENCES 8

DOCUMENT HISTORY 9

# Introduction

## Purpose of this Document

The purpose of this document is to outline how the design specification/program will be tested against the tests in the test specification table.

## Scope

The test plan aims to set standardisation for the testing of the program, and parameters in which the program will be tested to. All program testers are required to read this document to understand the guidelines they will follow when testing the program.

## Objectives

The objectives of the test plan are to see whether the features in the program work, given different variables. The testing will be done in the following way:-

* the first column will contain the test reference;
* the second column will contain the requisite being tested;
* the third column is test content which will give a short description of what is being tested;
* the fourth column will contain the input, i.e. what action will be done;
* the fifth column will contain the output after the action has taken place;
* the sixth column will contain the pass criteria i.e. how the test will be passed.
* the seventh column will contain the Test criteria i.e. to determine if that each test cases is a success or failure

# Approach to testing

Testing is a fundamental component of the software quality assurance and represented a review to meet the requirement as specified in the document QA Document SE.QA.01. The technique use to test is on black box testing techniques where we perform functional testing to ensure that the functionality meet with the requirement. In this approach, we focus on determining whether or not a program does what it is supposed to do base on its functional requirements. Black box testing attempts to find errors in the external behavior of the code which is following categories ; incorrect or missing functionality; interface errors; errors in data structures used by interfaces; behavior or performance errors; and initialization and termination errors. Through this testing, we can determine if the functions appear to work according to specifications. Black box testing tools are mainly record and playback tools. The tool that is use for testing is Netbean, JavaDB and other relevant IDE. Black-box testing focus on the inputs and outputs of the software without knowing their internal code implementation .The following is the guideline that should we follow during the testing phase.

1. Follow the requirement and specification that is shown in the QA documents
2. Conduct number of test case by choosing valid input to check the positive test scenario and invalid input to check the negative test scenario of the system
3. Tester should determine the expected output from the input given.
4. Test cases is should be construct to be review during the testing with the number of selected input
5. Chosen test cases are executed and documented for the outcome
6. Comparing the actual output with the expected outputs
7. If detect any errors, it should be fixed if possible
8. Test table is drawn up to show the number of test conducted

# Test Specification

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Ref** | **Req. being tested** | **Test content** | **Input** | **Output** | **Pass Criteria** | **Test pass/fail?** |
| SE-N02-001 | FR1 + FR6(register) | Test whether the “Register button” on the login page will navigate the user to the register page | Click the register button first - the button on the left on the login screen | A page which consists of forms and buttons will be shown on the user’s browser | The page should be load correctly | Pass |
| SE-N02-002 | FR1 + FR6(register) | Check if the system does not allow adding already existing user | Enter an user email that already exists in database | Account is not created, user gets error message | System displays  "User already exists"  error message | Pass |
| SE-N02-003 | FR1 + FR6 (Register) | Check if the system does not allow adding new user if all required fields are not filled | Leave all the fields blank | Account is not created, user gets error message | System displays  "You have to fill all required fields"  error message | Pass |
| SE-N02-004 | FR1 + FR6 (Register) | Check if the system does not allow using restricted characters | Enter special characters (^%#) in email section | Account is not created, user gets error message | System displays  "Incorrect input"  error message | Pass |
| SE-N02-005 | FR1 + FR6 (Register) | Check if system creates new user account | Enter user email and all required data correctly | List of stored users should now include this user. | Data should be stored  correctly | Pass |
| SE-N02-006 | FR1 + FR6 (Unregister) | Check if system deletes user data from database | Logged user clicks “Unregister” Option | List of stored  users should not include this user. User is redirected to login page | Data should be remove correctly | Fail |
| SE-N02-007 | FR6 (Unregister) | Check if system deletes user data from database | Logged user clicks “Unregister”  Option | Could not remove the user | System displays  "Some problems occurred, please try again or contact administrator " error message | Fail |
| SE-N02-008 | FR2 + FR6 | Check if the “friends button” on the Home page navigates the user to the ‘friends.html’ page | Click the friend button on the home page | The ‘friends.html’ page will be shown on the user’s browser | The page should be load correctly | Pass |
| SE-N02-009 | FR2 + FR6 (Add Friend) | Check if system does not allow to add non existing user | Enter user email that does not exist in database  or leave it blank | Friend should not be added | System displays “Sorry, user does not exist” error message | Pass |
| SE-N02-010 | FR2 + FR6 (Add Friend) | Check if user is added correctly if request accepted | Enter friends email correctly and person accepts friend request | Friend should be added correctly | Friend is added to user’s friends list | Pass |
| SE-N02-011 | FR2 + FR6 (Add Friend) | From the test SE-N02-010, We test out from the other user which is about to be accepted if his account has been notify with a friend request | Log-on to the other account which is about to receive the friend request | There should be a notification in the friend list page with the detailed of the email of the user. | The friend request should perform well | Pass |
| SE-N02-012 | FR2 + FR6 (Add Friend) | From the test From the test SE-N02-010, We check if the user able to add two successful friend request. | Add the two friend request by email in the friend request page. After that, Login to that chosen email account | Both account should receive the friend request | Adding more than one friend request in the friend page is successful | Pass |
| SE-N02-013 | FR2 + FR6 (Decline friend request) | To test the decline button of the friend request session | From the test SE-N02-010 ,we test it out with the decline button | If the person is declined, that user will not appear on the friend request box. | Friend invitation should be performing well | Pass |
| SE-N02-014 | FR2 + FR6 (Remove Friend) | Check if system allows removing friend without selecting any from the list | No friend is selected, use delete button | Error message should be displayed | System displays “You have to select a friend first.” error message | Fail |
| SE-N02-015 | FR2 + FR6 (Remove Friend) | Check if friend is deleted correctly | Select friend from “friend list”, use delete button | Friend should be deleted correctly | Friend is deleted from user’s friends list | Fail    Delete Friend function is unavailable |
| SE-N02-016 | FR3 + FR6 | Check if the “monster button” on the homepage will navigate the user to the monster menu page | Click the “monster button” shown in the homepage | The browser loads myMonster.html onto the screen | The navigation button should direct the user to the selected page | Pass |
| SE-N02-017 | FR3 (New account) | To verify if a Newbie package such as virtual money, is given to new users after their registration | Create an account on the login screen | After creating an account, the system will automatically allocate virtual money to the new account | Newbie package system should works | Pass |
| SE-N02-018 | FR4 | Test whether the user monster can challenge another monster | We test out by using an admin account: search another valid account or search from the list of friends | The chosen user will appear from the drop-down menu | The monster challenge system should works | Pass |
| SE-N02-019 | FR4 | Check if the user can accept a challenge from another user | Use an account to invite another member to a challenge | There will be a notification from the homepage that another user wishes to challenge | Challenge system should works | Pass |
| SE-N02-020 | FR4 | Check if the user can decline a challenge from another user | Use an account to invite another member to a challenge and have that other member decline the challenge | There will be a notification from the homepage that other user declined the challenge | Declining challenge from the user should works | Pass |
| SE-N02-021 | FR4 | To confirm that the winner of the monster challenge receives their reward | From test SE-N02-017, which is another account accepts the challenge from the current account to test which has a more powerful monster | There will be a short video play on screen of monsters fighting while the system calculates the winner | The monster challenge should works | Fail  No media is loaded |
| SE-N02-022 | FR6 (Offer for sale) | Check if system allows selling a monster without selecting any from the list | No monster is selected, use sell monster button | Error message should be displayed | System displays “You have to select a monster first.” error message | Pass |
| SE-N02-023 | FR6 (Offer for sale) | Check if system allows monsters to be sold | Select monster from “monster list”, use sell monster button | Monster should be added to the market list | Monster is added correctly to the market list | Pass |
| SE-N02-024 | FR6 (Offer for sale) | Check if system allows monsters with no price to be sold | Set monster price to 0 or leave it blank | Monster is not added to the market list, error message should be displayed | System displays “You have to set a monster price first” error message | Fail |
| SE-N02-025 | FR6 (Buy monster) | Check if system allows user with not enough funds to buy a monster | Select monster in market, use buy button | Monster is not added to users monsters list | System displays “You don’t have enough money to buy this monster” error message | Fail |
| SE-N02-026 | FR6 (Buy monster) | Check if system allows buying a monster without selecting any from the market list | Select no monster in market, use buy it button | No monster should be added to users monsters list | System displays “You have to select a monster first” error message | Fail |
| SE-N02-027 | FR6 (Buy monster) | Check if system allow to buy a monster from other user | Select monster, use buy button on the user | Monster should be added to users monsters list and removed from market | Monster is correctly added to users monsters list and removed from market | Pass |
| SE-N02-028 | FR6 (Offer for breeding) | Check if system allows breed a monster without selecting any from the list | No monster is selected, use breed monster button | Error message should be displayed | System displays “You have to select a monster first.” error message | Fail |
| SE-N02-029 | FR6 (Offer for breeding) | Check if system allows monsters with no price to be bred | Set breed price to 0 or leave it blank | Monster is not added to the market list, error message should be displayed | System displays “You have to set a breed price first” error message | Pass |
| SE-N02-030 | FR6 (Offer for breeding) | Check if system allows monsters to be bred | Select monster from “monster list”, use breed monster button | Monster should be added to the market list | Monster is added correctly to the market list | Pass |
| SE-N02-031 | FR6 (Purchase breeding) | Check if system allows user with not enough funds to buy a monster | Select monster in market, use breed button | Monster is not added to users monsters list | System displays “You don’t have enough money to buy this monster” error message | Pass |
| SE-N02-032 | FR6 (Purchase breeding) | Check if system allows buying a monster without selecting any from the market list | Select no monster in market, use breed button | No monster should be added to users monsters list | System displays “You have to select a monster first” error message | Fail |
| SE-N02-033 | FR6 (Purchase breeding) | Check if system allow to buy a monster for breeding | Select monster, use breed button | New monster should be added to users monsters list | New monster is correctly added to users monsters list | Fail |
| SE-N02-034 | FR1 + FR7(Log-in) | Test whether the ”login button” on the login page works – this will navigate the user to the home page | Entering a valid username with the password and clicking the login button | After the login process, the home page will be loaded onto the user’s browser | The page should have been load correctly | Pass |
| SE-N02-035 | FR7(Log-in) | Check if the system allows to log-in not providing any details | Leave blank user log-in and password | Error message should be displayed | System displays “Please, enter your username and password.” error message | Pass |
| SE-N02-036 | FR7(Log-in) | Check if the system allows not existing users to log-in | Enter new user email and password | User is not logged-in, error message should be displayed | System displays “Wrong username or/and password” error message | Pass |
| SE-N02-037 | FR7(Log-in) | Check if the system allows user to log-in no password | Enter only username, leave password blank | User is not logged-in, error message should be displayed | System displays “Wrong username or/and password” error message | Pass |
| SE-N02-038 | FR7(Log-in) | Check if the system allows user to log-in with incorrect password | Enter username and wrong password | User is not logged-in, error message should be displayed | System displays “Wrong username or/and password” error message | Pass |
| SE-N02-039 | FR7(Log-in) | Check if the system allows user to log-in | Enter username and password | User should be logged-in | User is logged-in and redirected to the home page | Pass |
| SE-N02-040 | FR7(Log-out) | Check if the systems allows user to log-out | Use log-out button | User should be logged-out | User is logged out and redirected to the log-in page | Pass |
| SE-N02-041 | FR8(monsters list) | Check if the system loads the list of user’s monsters correctly even though it is empty | Use “my monsters“ button | System should show an empty list | System should be loaded an empty list correctly | Pass |
| SE-N02-042 | FR8(monsters list) | Check if the system loads the list of user’s monsters | Use “my monsters“ button | System should show a list of monsters | System should be loading the list correctly | Pass |
| SE-N02-043 | FR8(friends list) | Check if the system loads the list of user’s friends correctly even though it is empty | Use “friends“ button | System should show an empty list | System should be loading an empty list correctly | Pass |

REFERENCES

[1] Software Engineering Group Projects: General Documentation Standards. C. J. Price, N. W. Hardy. SE.QA.03. 1.5 Release

[2] Software Engineering Group Projects: Test Procedure Standards. C. J. Price, N. W. Hardy. SE.QA.06. 1.6 Release

DOCUMENT HISTORY

| *Version* | *CCF No.* | *Date* | *Changes made to document* | *Changed by* |
| --- | --- | --- | --- | --- |
| 1.0 | N/A | 14/11/2012 | N/A - original version | PW |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |