**CS22120 Software Development Life Cycle**

**Group 05 Test Specification**

**Author:** Krf

**Config Ref:** SE\_05\_TEST Group 5

**Date:** 8th December 2012

**Version:** 1.2

**Status:** Draft

Group 05

Aberystwyth University

Aberystwyth

Ceredigion

SY23 3DB

Copyright © Aberystwyth University 2012

Table of Contents

1. Introduction 3

1.1. Purpose of this Document 3

1.1.1. Scope 3

1.2. Objectives 3

2. General Approach to Testing 3

3. Test Table 4

4. References 12

5. Document Change History 12

# 1. Introduction

## 1.1. Purpose of this Document

The purpose of this document is to highlight any possible tests which will be needed in order to meet the Functional Requirements [1].

### 1.1.1. Scope

This Test Specification document will show what tests will be needed before the actual testing of the code, and how we plan to test these functions. Additional tests may be added under the Software Delivery for Acceptance Testing.

## 1.2. Objectives

The main objective of this document is to show all the tests which will cover each function in its functional requirement category. The tests will consist of a test reference, any inputs and outputs, what functional requirement category it belongs to, and the expected outcome. The test reference will be used in future testing.

# 2. General Approach to Testing

Our module/unit testing will be performed using JUnit tests for each class, designed and implemented by the programmer who is working on that class. This will be verified by the tester during the implementation phase as well as the integration testing phase of the project.

Our general system testing will take place in three different environments:

The first environment will be contained within our group and will use pre-generated test data, no real-time tests for time-based events and server interaction will take place between members of the group. The test data will be created based on the inputs required to fully test the functionality of the program.

The second environment will be contained within two to three groups and will use specific test data, some real-time tests for time-based events and server interaction will take place between members of the groups. The test data will be created based on the inputs required to fully test the functionality of the program.

The third environment will be contained within two to three groups, more if possible, will use specific test data and real-time tests and events will occur. Server interaction will take place between members of all groups. The test data will be created based on the inputs required to fully test the functionality of the program. This may occur during the Acceptance Testing phase.

# 3. Test Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Reference** | **Functional Requirement** | **Test Content** | **Input** | **Output** | **Pass Criteria** |
| SE-TS-001 | FR1/FR6 | Check that if valid data is entered, log-in is possible. | Valid email address and password. | User's Home Page should be displayed. | After log-in details are entered and log-in button is selected, user will be directed to their home page. |
| SE-TS-002 | FR1/FR6 | Check that if invalid data is entered, log-in is not possible and error message shown. | Invalid data – either email address and/or password not used or data that does not resemble an email address. | General Error Message should be displayed i.e. “Invalid Username and/or Password”. | After invalid details are entered, and log-in button pressed, user will see an error message and no forward to home page will occur. |
| SE-TS-003 | FR6 | Check that if sign-up button is selected, user is directed to sign up page. | User selects sign-up button. | Sign-up page displayed. | After clicking sign-up button, the user should be directed to the sign up page. |
| SE-TS-004 | FR1 | Check that sign-up details are valid at registration. | User enters valid sign-up details and then tries to log in with the new user details. | User should to be directed to log-in page, then to home page after log-in. **Design Specification:**  **Design Specification:** | If output is correct depending on what data has been entered, then test has passed. |
| SE-TS-005 | FR1/FR6 | Check that after SE-TS-004 has passed with valid data and valid sign up data has been added, that the user can now log-in. This tests whether the sign-up data has been processed properly. | Sign up data at registration. | User should be directed to their new home page. | After registration, user should be able to access the log-in page, log-in with their details and be re-directed to their home-page. |
| SE-TS-006 | FR2/FR8 | Check that a user's home page displays friends list with correct friends. | Initial test would use pre-generated test data, actual system test will use data obtained via interaction between users. | Friends list will be displayed on homepage with correct friends. | If the friends match up to the data within the server, then the friends list is being maintained correctly. |
| SE-TS-007 | FR3/FR8 | Check that user's home page displays a list of monsters with the correct monsters. | Initial test would use pre-generated test data, actual system test will use data obtained via interaction with the system and other users. | Monster list will be displayed on homepage with correct friends. | If the monsters match up with the data within the server, then the monster list is being maintained correctly. |
| SE-TS-008 | FR3 | Check that when user is first registered they are provided with a new basic monster and a small cash value. | User would sign up, then proceed to log-in. | At homepage, the monster list should display their new monster and their cash pile should display the amount given. | After log-in, the monster list and cash pile should be displayed with the required items. |
| SE-TS-009 | FR3 | Check that the server manages monster life-cycle, with regards to ageing then death of monsters. | The initial test will shorten the amount of time that ageing and death can occur in. The actual system test will take place in real-time using the algorithm contained within the program. | Monsters age, value and eventual death will be noted and displayed on web-page. | After a pre-determined amount of time, the monster will die. Between its creation and death the monster should also signs of ageing, such as change of age, value and characteristics. |
| SE-TS-010 | FR4 | Check that the monsters have a prize value assigned to them based on their characteristics. | Initial test would use pre-generated test data, actual system test will use data obtained via interaction with the system and other users. | Monster list will show monsters with their prize value. | The prize value of a monster should be equal to either the pre-generated values or determined correctly based on their characteristics. |
| SE-TS-011 | FR4/FR5/FR6 | Check that a user can select a monster and therefore issue a challenge with that monster against a friend. | Initial test would use pre-generated test data, actual system test will use data obtained via interaction with the system and other users. User should be able to select one of their own monsters, then select a friend and issue a challenge. | A challenge should appear in the friend's request list, with prize value of monster selected. | If the user is able to select a monster, issue a challenge to a friend on their friends list and that challenge appears on the selected friend's challenge list, this test has passed. |
| SE-TS-012 | FR4 | Check that after challenge is issued and accepted, the battle will take place and a winner will be decided. | Initial test would use pre-generated test data, actual system test will use data obtained via interaction with the system and other users. | A battle report page will be displayed showing the outcome of the battle. | A battle report page appears confirming a winner has been declared and it displays the chosen winner. If this occurs, the test is passed. |
| SE-TS-013 | FR4/FR10 | After battle, server should “pay” the winner the prize value of the losing monster. | Use the data of the monster used in  SE-TS-011 | Cash pile of winner should change. | If the cash pile increases by the prize value of the monster used, the test shall be passed. |
| SE-TS-014 | FR5/FR6/FR9 | Server should handle friend requests so users can see requests and accept or decline them. | User will enter email address of another user. Other user will be sent a friend request. | On accept, user's friend list should display the other user as a friend. | If the request is displayed in the other user's requests, it is accepted and the user is displayed, the test is passed. It is also passed if the user sends a request and upon reject the friend does not appear in the friend list. |
| SE-TS-015 | FR5/FR6 | User should be able to buy monsters from and sell monsters to their friends. | User will place a monster for sale and will attempt to buy a friend's monster which is displayed as being for sale. | Users will lose monster from list when sold, and gain monster from friend when bought. | If when the monster is bought it disappears from the seller's list and appears in buyer's list, then the test is passed. |
| SE-TS-016 | FR5/FR6 | User should be able to put a monster up for breeding and have another user choose that monster to buy for breeding. | User will place a monster up for breeding. One of the user's friends will try to buy that monster for breeding. | Users should keep both monsters. User who sold male would gain money equivalent to the sale. Other user would gain new baby monster. | If the users gain the appropriate items from the communication, the test is passed. |
| SE-TS-017 | FR6 | User should have the ability to unregister. | User will select the unregister button. Password confirmation could be required. | User should be removed from user data making them unable to log-in. | If user's details have been removed from the database and the user can no longer log-in, the test is passed. |
| SE-TS-018 | FR7 | Start-up page should give user the ability to register/ login. | The start-up page will be loaded. | The start-up page should be displayed. | If the start-up page contains area that would give the user the ability to register or log-in, the test is passed. |
| SE-TS-019 | FR7 | Throughout the site, the user should be able to logout and be returned to the start-up page. | On each page there should be an area which the user can select to logout. | User should be returned to the start-up page. | If the user selects the log-out link, the user should be logged out and the user should be returned to the start page. |
| SE-TS-020 | FR8 | Whilst using the game, the user should be able to see a friends list, a monster list and any requests and challenges they may have. | User will log-in to game. | The website. | If the website contains items that match with the requirements, then the test is passed. |
| SE-TS-021 | FR10 | After a battle, the loser's monster should be removed from their monster list. The winner's monster should be updated with any injuries that may have occurred. | After SE-TS-011, user and friend will check their monster lists. | Each of their monster lists should be updated. | If the lists have updated as required, the test is passed. |
| SE-TS-022 | FR11 | Friends list should be ordered by wealth and there should be a space that states where the user is within the list. | Initial test would use pre-generated test data, actual system test will use data obtained via interaction with the system and other users. | Friends list should be displayed in order. | If the order of the friends list is correct to the order the test data requires, the test is passed. |
| SE-TS-023 | PR1 | User should not be able to detect any noticeable lag while using the website. | User will browse website and interact with the website. | Outcomes of actions should be displayed as required. | If there is no noticeable lag between the user interaction and the required outcome, the test is passed. |
| SE-TS-024 | PR2 | Website should run on all standard browsers. | Website will be run on all standard browsers i.e. Browsers which are found on all departmental machines. | Website. | If the website displays correctly and runs as expected, the test is passed. |
| SE-TS-025 | FR1 | Check that sign-up details are valid at registration. | User enters invalid sign-up details. | Invalid data is added, an error message should be displayed. This error message should state the problem that has occurred. | If the error message appears correctly, then the test has passed. |
| SE-TS-026 | FR5/FR6 | Check that if an invalid cash value (negative figure, ‘0’, or non-numerical value) is entered for a monster when ‘apply’ is hit to put the monster up for selling; an error message occurs alerting the user and the selling application is rejected. | User enters in ‘0’, ‘-27’ and ‘45afs5’ over 3 separate tests into the selling value field for a monster when putting them up for sale. | Error message should occur when ‘apply’ is hit to put the monster up for sale alerting the user that an invalid selling value has been entered and the monster will not be put up for sale. | If error message appears correctly, then tests have passed. |
| SE-TS-027 | FR5/FR6 | Check that only monsters that have been put up for selling are available to buy for other users. | User selects a friend who has both a monster that is available for buying and one not available for buying and is taken to their page with the list of monsters. | A list of monsters should appear on the friend’s page with one monster which has a ‘buy’ button next to it and one monster without the ‘buy’ button next to it. | If the right monsters that are set to ‘buy’ are available to buy and the others are correctly not, then test has passed. |
| SE-TS-028 | FR5/FR6 | Check that if an invalid cash value (negative figure, ‘0’, or non-numerical value) is entered for a monster when ‘apply’ is hit to put the monster up for breeding; an error message occurs alerting the user and the breeding application is rejected. | User enters in ‘0’, ‘-2’ and ‘7ha1’ over 3 separate tests into the breeding value field for a monster when putting them up for breeding. | Error message should occur when ‘apply’ is hit to put the monster up for breeding alerting the user that an invalid breeding value has been entered and the monster will not be put up for breeding. | If error message appears correctly, then tests have passed. |
| SE-TS-029 | FR5/FR6 | Check that only monsters that have been put up for breeding are available to ‘rent’ for other users. | User selects a friend who has both a monster that is available for breeding and one not available for breeding and is taken to their page with the list of monsters. | A list of monsters should appear on the friend’s page with one monster which has a ‘breed’ button next to it and one monster without the ‘breed’ button next to it. | If the right monsters that are set to ‘breed’ are available to breed and the others are correctly not, then test has passed. |

# 4. References

[1] *Software Engineering Group Projects. Requirement Specification. B.P. Tiddeman. SE.CS.RS 1.1. Second Draft, 3.1 Functional Requirements.*

# 5. Document Change History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | CCF No. | Date | Changes made to document | Changed By |
| 1 | N/A | 08/11/12 | Original | Sjm16 |
| 1.1 | N/A | 15/11/12 | Changes to the test table | Olr1 |
| 1.2 | N/A | 08/11/12 | Changes to the test table and adding of 'General Approach to Testing' | Olr1, Krf |
| 1.3 | N/A | 28/01/13 | Changes to the test table | Krf |
| 1.4 | N/A | 29/01/13 | Ensured that there were enough tests to cover all bases. | Chs17 |