SE Testing Document

By: Corbin Robinson, Ethan Smith, Zaniken Gurule, Parker Rosenberg

Revision History

Version	Date	Participants	What was changed
v1.0	11/13/2020	Corbin Robinson, Ethan Smith, Zaniken Gurule, Parker Rosenberg	Created document
v2.0	12/11/2020	Corbin Robinson, Ethan Smith, Zaniken Gurule, Parker Rosenberg	Completed the tests

Table of Contents

1. Project description………………………………………………………………………….3

2. Overall testing plan………………………………………………………………………….3

3. Unit Tests……………………………………………………………………………………….3

4. Top-Down Integration Tests…………………………………………………………………..5

5. Validation Tests……………………………………………………………………………….6

6. System Tests…………………………………………………………………………………..8

Project Description

Our project is a website that will recommend board games. Users will be able to add games that they own/like to an account library and get recommendations similar to those games based on mechanics and theme.

Overall testing plan

Ethan Smith will do unit testing. Corbin Robinson will do integration testing. Zaniken Gurule will do validation testing. Parker Rosenberg will do system testing. Bias shouldn't be an issue because our testing results will be objective and essentially binary (does it work, does it not).

Unit Tests

Test ID: U1
Date: 12/11/20
Requirements ID: 1
Tester: Ethan Smith
Environment: python Venv running program on (ubuntu LTS20.04) using python 3.6
Goal: to test the board game selector function for errors, and to confirm that it returns the expected results in all cases.
Testing procedure: input various values and combinations of values for possible board game seeds (including ones that are incorrect/should cause errors) to test that the system correctly handles each input.
Expected results: the program should output correct suggestions based on the inputs tested, or catch errors
Actual results: TODO
Status: COMPLETE
Comments: This is probably the most important unit test for the program.

Test ID: U2
Date: 12/11/20
Requirements ID: 2
Tester: Ethan Smith
Environment: python Venv running program on (ubuntu LTS20.04) using python 3.6

Goal: To test the login functionality, allowing users to successfully login to the program.

Testing procedure: Send many different values to the login function, including valid and invalid users, empty strings, or situations where the password is in the database, but not related to the user it is tested with

Expected results: The program should correctly handle all different input, correctly authenticating a correct and matching username/password combo, and denying any other combinations.

Actual results: TODO

Status: COMPLETE

Comments: Tests to user features are important for security.

Test ID: U3

Date: TODO

Requirements ID: 3

Tester: Ethan Smith

Environment: python Venv running program on (ubuntu LTS20.04) using python 3.6

Goal: to test the board game list, which allows users to add games that they own to their profile.

Testing procedure: test adding and removing games(same function handles both) from a user profile, including cases such as adding a game that already exists, or trying to add a game that doesn’t exist, as well as the generic adding and removing of valid games.

Expected results: the program should correctly add and remove games from user profiles, if it receives a duplicate it should ignore the request, and if it receives a request for an invalid/nonexistent game it should also ignore the request.

Actual results:

Status: TODO

Comments: This tests a tertiary functionality, and is less important than the other two tests, although this test should still pass.

Top-Down Integration Tests

Test ID: I1

Date: 12/11/2020

Requirements ID 4

Tester: Corbin Robinson

Environment: Python Venv running program on (ubuntu LTS20.04) using python 3.6

Goal: test user library permanence.

Testing procedure: Create users, create different libraries, sign out then back to test permanence. Also close/refresh the webpage to test library permanence.

Expected results: Libraries should maintain contents based on user

Actual results: Libraries are maintained between accounts and logouts

Status: Complete

Comments:

Test ID: I2

Date: 12/11/2020

Requirements ID 5

Tester: Corbin Robinson

Environment: Python Venv running program on (ubuntu LTS20.04) using python 3.6

Goal: make sure the website can recommend based off of different users libraries

Testing procedure: Create users, create different libraries, see if it recommends the correct things

Expected results: Website will recommend games based on users library

Actual results: Website correctly recommends games based on users library

Status: Complete

Comments:

Test ID: I3

Date: 12/11/2020

Requirements ID 6

Tester: Corbin Robinson

Environment: Python Venv running program on (ubuntu LTS20.04) using python 3.6

Goal: Test page navigation

Testing procedure: Click on all the page buttons

Expected results: Each button will bring user to correct page and display correct information

Actual results: Each buttons works correctly

Status: Complete

Comments:

Validation Tests

Test ID: V1

Date: 12/11/2020

Requirements ID: 7

Tester: Zaniken Gurule

Environment: Running the website on a browser like Google Chrome, accessing the website as a regular user.

Goal: Test if the website gives recommendations as intended.

Testing Procedure: Create and set up an account and try to get recommendations from the website.

Expected results: The website should provide a list of board game recommendations based on input during account creation.

Actual results: Was successfully able to create an account and get recommendations based on the games I selected.

Status: COMPLETE

Comments:

Test ID: V2

Date: 12/11/2020

Requirements ID: 8

Tester: Zaniken Gurule

Environment: Running the website on a browser like Google Chrome, accessing the website as a regular user.

Goal: Test if user accounts behave as expected with account creation and logging in.

Testing Procedure: Sign up for an account on the website, log off, log back in.

Expected results: Tester should be able to create an account and log back into the website.

Actual results: Was able to create a user account, log out of account, and log back into the account.

Status: COMPLETE

Comments:

Test ID: V3

Date: 12/11/2020

Requirements ID: 9

Tester: Zaniken Gurule

Environment: Running the website on a browser like Google Chrome, accessing the website as a regular user.

Goal: Test the drag and drop features of the website.

Testing Procedure: Tester will login into an account and try to interact with the drag and drop functionality of the game selection menu.

Expected results: Tester should be able to make selections of games, using a drag and drop menu system.

Actual results: Was able to select games, however no drag and drop system was in place.

Status: COMPLETE

Comments: A drop down menu was implemented instead.

System Tests

Test ID: S1

Date: 12/11/20

Requirements ID: 10

Tester: Parker Rosenberg

Environment: Computer

Goal: Test the complete system on a PC. Check every input for desired outputs. Test user experience.

Testing procedure: Use the website as a user, test all functionality, try to input bad data.

Expected results: Runs as a standard website, no errors, all the components work nicely together..

Actual results: Runs as a standard website, no errors, everything works nice together, doesn't let bad data in.

Status: COMPLETE

Comments: flawless

Test ID: S2

Date: 12/11/20

Requirements ID: 11

Tester: Parker Rosenberg

Environment: Phone

Goal: Test the complete system on an Iphone. Check every input for desired outputs. Test user experience.

Testing procedure: Use the website as a user, test all functionality, try to input bad data.

Expected results: Runs as normal, formatting might be wonky.

Actual results: Runs as a standard website, no errors, everything works nice together, doesn't let bad data in, and the formatting is actually really good on the Iphone, very similar to the PC.

Status: COMPLETE

Comments: it's even better than the pc formatting

Test ID: S3

Date:12/11/20

Requirements ID: 12

Tester: Parker Rosenberg

Environment: Tablet

Goal: Test the complete system on an Ipad. Check every input for desired outputs. Test user experience.

Testing procedure: Use the website as a user, test all functionality, try to input bad data.

Expected results: Runs basically the same as IPhone, formatting might be less wonky, because larger screen.

Actual results: Runs as a standard website, no errors, everything works nice together, doesn't let bad data in, the formatting was also really good on the Ipad, very similar to that on the Iphone and PC.

Status: COMPLETE

Comments: incredible