SW Engineering CSC648/848 Fall 2018

Project Title: Aquirium

Team Number: 08

Names of Students (Local Team):

Jianfei Zhao (jzhao11@mail.sfsu.edu) (Team Lead)
Feras Alazzeh (Back-end Lead)
Tommy Lik (Front-end Lead)
Lileana Wright
Edward Barajas
Alex Li
Jiawei Xu

Milestone: 4

Date: 12/01/2018

History Table:

Date	Revision
12/01/2018	Created as the 1st version

I. Product Summary

i). Name of the Product
Aquirium

ii). List of ALL Major Committed Functions (Final p1 Functions)
1). Browsing
- 1.1 By clicking title image in home page, its detail page shall be popped up in a new tab.
- 1.2 The number of items (or search results) shall be shown to the user.
- 1.3 If more than one page, pagination shall be available.

2). Searching
- 2.1 The default category shall be "**All**", with all the items shown to the user.
- 2.2 By switching to a specific category, the items under that category shall be shown.
- 2.3 Only alphabetic or numeric characters shall be accepted for search (input validation).

3). Posting
- 3.1 By clicking "POST" in **top nav bar**, the detail page shall be popped up in a new tab.
- 3.2 The detail page can be either submitted or cancelled by clicking the bottom button.
- 3.3 User shall be prompted that it may take 24 hrs for the item to be shown on website.

4). Login
- 4.1 If username does not exist, prompt message shall notify the user.
- 4.2 If password does not match the valid username, prompt message shall notify the user.

5). Registration
- 5.1 If username already exists, prompt message shall notify the user.
- 5.2 All the required fields shall pass the input validation before submission.

6). Contact Seller
- 6.1 By clicking "CONTACT", the contact page shall be popped up in a new tab.
- 6.2 In contact page, the image, title, price, as well as username shall be prefilled.
- 6.3 The contact page can be either submitted or cancelled by clicking the bottom button.

7). Seller Dashboard
- 7.1 Registered user shall be able to edit the items posted by him/her.
- 7.2 Registered user who **sells** the item shall be able to view the messages sent to him/her.

8). Admin Dashboard
- 8.1 Admin user shall be able to approve or delete inappropriate items.

- 8.2 Admin user shall be able to approve or delete inappropriate users.

iii). What is Unique in Aquirium

Since Aquirium is a web application designed for students at SFSU, it mainly contains items that are needed for students, excluding luxury goods. For example, the categories are books, furniture, electronic devices, etc.

Moreover, it provides convenience for students to search for books. For example, students can utilize text search bar and input “csc” or “CSC510”, to help them narrow down the search results of books into the computer science subject.

iv). URL to Aquirium (on Server)

http://54.183.220.150

II. Usability Test Plan

Selected Function for Testing Usability: Searching

i). Test Objectives

Searching is one of the most basic functions within this web application, since it offers the customers a helpful tool to narrow down the results. Whether the searching function provides the user with convenience and whether it correctly responds to the user’s input will make the first as well as the most important impression.

Moreover, the validation for input is also necessary and should be tested before the final delivery. On one hand, we should restrict the length and content of user’s input to avoid any inappropriate code injection. On the other hand, if the input is valid, we should display the items that approximately match user’s search text.

All of the test objectives mentioned above are actually trying to collect feedback from users, in order to help our development team bring better user experience for both existing and potential customers.

ii). Test Plan

1). System Setup

The system setup process is fast and simple. From the user’s perspective, there would be no installation required. For the testing of search function, a user just needs to launch a web browser on his/her computer or laptop, and make sure that the Internet is connected. Currently, we support the latest two versions of Google Chrome and Mozilla Firefox as the web browsers. By

typing the URL in web browser, the user will be able to visit our Aquirium website.

2). Starting Point

The starting point of search function is exactly the home page, with URL available below. The search bar used for searching function is just under the top navigation bar. If the character(s) or the length of user's input is valid, corresponding search results will be shown on the same page. Otherwise, a message will be prompted for the invalid input.

3). Intended User

If the user is just trying to do some search on our website, s/he does not have to register or log in. Otherwise, if the user is asked or redirected to the registration or login page, s/he may become annoyed. Therefore, for the searching function, the intended user would be the users of all types, including unregistered, registered, as well as admin users. Basically, anyone can search the items on Aquirium without limit.

4). Task

Task to be Accomplished	**Criteria of Completion**
Input validation	Only alphabetic and numeric values are allowed, with the length of characters ≤ 40.
Search based on category	When switching to a different category, the search results belong to that category.
Search based on text input	When executing text search, the items at least partially match the input text.
Responsiveness	By expanding or shrinking the size of web browser, the format changes accordingly.

5). URL to be Tested

http://54.183.220.150/

Technically, this is the URL not only for the home page (starting page) of our web application, but also for the item list of search results.

iii). Questionnaire

Below is a Likert-Scale Questionnaire, with questions inquiring user's satisfaction after performing the above tasks. Each "question" is a statement, with five choices (from **STRONG AGREE** to **STRONG DISAGREE**) scaling user's response.

	STRONG AGREE	**AGREE**	**NEUTRAL**	**DISAGREE**	**STRONG DISAGREE**
The search bar is at the same position, easy to find.					

The search results belong to the chosen category.					
The search results match the input text approximately.					
User is notified with invalid input (>40 chars or not alphabetic or not numeric).					
If more than one page, pagination is available.					

III. QA Test Plan

i). Test Objectives

Apart from user satisfaction, the QA test for searching function mainly pays attention to the quality, to be more precise, the correctness of results and deployment.

To test the correctness of search function, each of the search results will be viewed to check whether the item title approximately matches the input text. Moreover, the number of results will also be counted to check whether all the matching records in database are successfully retrieved.

ii). HW and SW Setup

1). Server Host

- The server host has been installed on the operating system of Microsoft Windows Server 2012 R2 Base.
- The Amazon Web Services (AWS) has been installed on this Windows System, with the free-tier services selected for our web application.
- The AWS Server has been allocated with 1vCPU and 1GB-RAM.

2). WAMP

- Apache 2.4.33 has been successfully set up, which is required for running PHP.
- MySQL 5.7.21 has been successfully set up, which is required for running the database.
- PHP 5.6 has been successfully set up, which is the back-end programming language.

3). Web Browser

- The latest 2 versions of Chrome and Firefox browsers will be used for QA test.

- Most of the computers have these browsers either installed or automatically upgraded. If not available, the QA tester may have to download them and install.
- The URL for QA test is http://54.183.220.150

iii). Features to be Tested

1). Selection group of categories

- By switching to the "**All**" category, all the items on website should be shown to the user. This is also the default category when the user visits our home page.
- By switching to other categories, the search results should correspond to each specified category.

2). Validation of input text

- For text search, the text length has to be within 40 characters, where only alphabets and numbers are valid input.
- If violating any rules, a notification of invalid input will be prompted.

3). Approximate matching

- For text search, the results do not have to perfectly match user's input. Instead, approximate matching will be executed.

4). Responsiveness

- By expanding or shrinking the browser, the list page of search results is responsive to the change of browser size.

iv). Test Plan

To make the format of test plan easy to read, a tabular form is provided below. It is convenient for the user to understand the details that are included in this QA test.

The QA table columns are designed based on the standard as well as the recommended format, which contains: test #, test description, test input, expected output, and test results (either PASS or FAIL) on Chrome and Firefox browsers (two latest versions for each).

Test Number	Description	Test Input	Expected Output	Test Results PASS / FAIL
1	Test the approximate text search - 1st case.	"book"	14 items will be retrieved and listed in 3 pages, which are all books.	PASS on Chrome PASS on Firefox

2	Test the approximate text search - 2nd case.	"csc510"	Only 1 item will be retrieved, which is a book with "CSC510" as part of the title.	PASS on Chrome PASS on Firefox
3	Test the approximate text search - 3rd case.	""	An empty string means "**All**", and thus all the items will be displayed.	PASS on Chrome PASS on Firefox
4	Test the input validation, for the content and length of search text.	Any input with special characters (*, @, -, etc.), or with length over 40.	An alert message will notify the user that the text content is invalid or the length is out of bound.	PASS on Chrome PASS on Firefox
5	Test the selection group of categories.	No input needed. Just change the category.	Under "**All**" category, there are 48 items in 8 pages. Specifically, there are 9 in "**Clothes**", 14 in "**Books**", 8 in "**Entertainment**", 8 in "**Electronic Devices**", and 9 in "**Furniture**".	PASS on Chrome PASS on Firefox

IV. Code Review

i). Coding Style

Out team uses tab-indentations as the coding style. For each implementation file, such as front-end "**.html**" and back-end "**.php**" files, we put header comments. There are a lot of ".php" files, in which we only comment on the ones implemented by ourselves, not on the existing framework PHP files. Such coding style has been approved by all members and by CTO (in Milestone 3).

ii). Peer Review and Comments

Instead of team lead or sub-team leads, **Lileana** and **Jiawei (Norman)** are responsible for this peer code review. Both of them are members in the protection rule of master branch, helping the leads with the maintenance of our team repository on GitHub. Below are the screenshots of the examples of their code review, with the source code updated on GitHub.

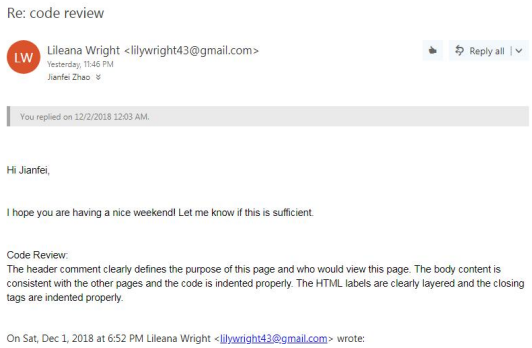

Figure 1. Code review by Lileana

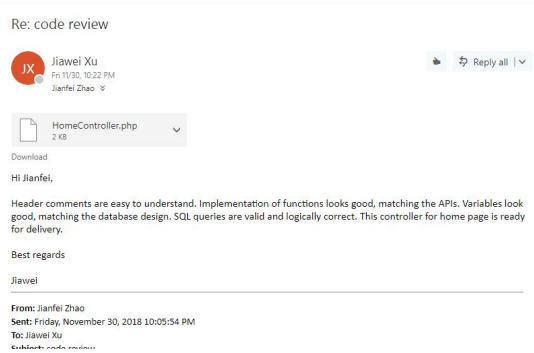

Figure 2. Code review by Jiawei

V. Self-Check on Practices for Security

i). Major Assets under Protection

- User's password is being protected through the encryption of md5() hash-function. More details are available in the section below (**V-ii**).
- User's email is being protected through the prohibition of email contact within this web application. Instead, user contacts each seller by using in-site messaging.

ii). Confirmation of PW Encryption in the DB

- When an unregistered user submits the form of registration (through the registration page), the field of password will be encrypted. This is done by using the md5() function in PHP, which is an advanced hash-function to transfer the original string into a hashed string with 32 characters. As a result, the password will always be saved as an encrypted string and can only be matched but not decrypted.

iii). Confirmation of Input Data Validation

- During text search, only alphabetic or numeric characters are allowed, and the length of input has to be within 40 characters.
- During registration, all the required fields are examined. For instance, the username has to be 6 characters at minimum, without colliding with any other usernames. The password has to be 6 characters at minimum, and the password confirmation has to match the password.
- The input validation is implemented by Bootstrap combined with JavaScript (jQuery) code.

VI. Self-Check on Adherence to Original Non-Functional Specs

1). Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0 (some may be provided in the class, some may be chosen by the student team but all tools and servers have to be approved by class CTO).

- **DONE**

2). Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of all major browsers: Mozilla, Safari, Chrome.

- **ON TRACK**

3). Selected application functions must render well on mobile devices.

- **ON TRACK**

4). Data shall be stored in the team's chosen database technology on the team's deployment server.

- **DONE**

5). No more than 50 concurrent users shall be accessing the application at any time.

- **ON TRACK**

6). Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.

- **DONE**

7). The language used shall be English.

- **DONE**

8). Application shall be very easy to use and intuitive.

- **DONE**

9). Google analytics shall be added.

- **DONE**

10). No e-mail clients shall be allowed.

- **DONE**

11). Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated.

- **DONE**

12). Site security: basic best practices shall be applied (as covered in the class).

- **DONE**

13). Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development.

- **DONE**

14). The website shall prominently display the following exact text on all pages "SFSU-Fulda Software Engineering Project CSC 648-848, Fall 2018. For Demonstration Only" at the top of the WWW page. (Important so as to not confuse this with a real application).

- **DONE**