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
12/03/2018	Revised according to CEO's feedback

I. Product Summary

i). Name of the Product
Aquirium

ii). List of ALL Major Committed Functions (Final P1 Functions)
P1 function(s) for unregistered user:
1). Unregistered user shall be able to browse items, with pagination available.

2). Unregistered user shall be able to search items by text search bar together with selection group of categories.

3). Unregistered user shall be asked for registration when posting an item.

P1 function(s) for registered user:
4). Registered user shall be able to post items on the website.

5). Registered user shall be able to log in by using username and password.

6). Registered user shall be able to contact seller through in-site messages within the website.

7). Registered user shall be able to access the seller dashboard to view the items posted by him/her and messages sent to him/her.

P1 function(s) for admin user:
8). Admin user shall be able to access the admin dashboard to approve or reject item postings.

iii). What is Unique in Aquirium
Since Aquirium is a web application designed for students at SFSU, 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
Deployment Server: Amazon AWS
URL: 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 will be for 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 #	Task for Usability Test
1	Find a chair to buy.
2	Find a math book to buy.
3	Find a book for CSC510 to buy.

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 **Strongly Agree** to **Strongly Disagree**) scaling user's response.

	Strongly Agree	**Agree**	**Neutral**	**Disagree**	**Strongly Disagree**
Category browsing is intuitive.					
Search function is convenient to use.					
The user interface for finding books is pleasant to use.					

Your comments are especially appreciated.

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

Here are the features of searching function, and the table of QA test plan is available below.

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	Test Description	Test Input	Expected Output	Test Results PASS / FAIL
1	Test the approximate text search (case 1).	Select "**All**" category and type "book" in the text search bar (under the top nav bar).	14 items will be retrieved and listed in 2 pages, which are all books.	PASS on Chrome PASS on Firefox
2	Test the approximate text search (case 2).	Select "**All**" category and type "csc510" in the text search bar.	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 (case 3).	Select "**All**" category and type nothing (empty string "") in the text search bar.	An empty string means all items. Under "**All**" category, there will be 52 items listed in 6 pages.	PASS on Chrome PASS on Firefox

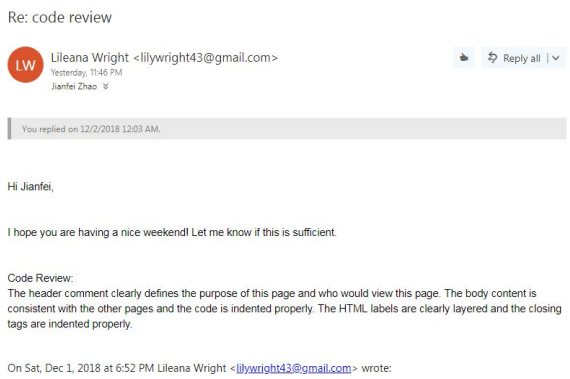
4	Test the input validation, for the content of search text.	Select any category and type special characters (“*”, “@”, etc.) in the text search bar.	An alert message “**Your input has to be alphabetic or numeric characters.**” will notify the user that the input text contains invalid special characters.	PASS on Chrome PASS on Firefox
5	Test the selection group of categories.	Switch to different categories in the selection group (next to the text search bar).	Specifically, there are 9 items in **Clothes**, 8 in **Entertainment**, 14 in **Books**, 8 in **Electronic Devices**, 4 in **Fishes**, 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


Re: code review

LW Lileana Wright <lilywright43@gmail.com>
Yesterday, 11:46 PM
Jianfei Zhao

Reply all

You replied on 12/2/2018 12:03 AM.

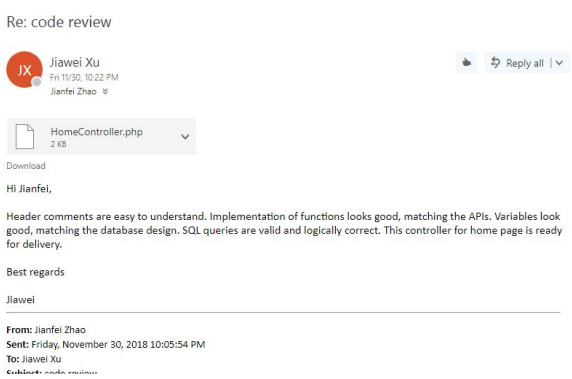
Hi Jianfei,

I hope you are having a nice weekend! Let me know if this is sufficient.

Code Review:
The header comment clearly defines the purpose of this page and who would view this page. The body content is consistent with the other pages and the code is indented properly. The HTML labels are clearly layered and the closing tags are indented properly.

On Sat, Dec 1, 2018 at 6:52 PM Lileana Wright <lilywright43@gmail.com> wrote:

Figure 1. Code review by Lileana


Re: code review

JX Jiawei Xu
Fri 11/30, 10:22 PM
Jianfei Zhao

Reply all

HomeController.php
2 KB

Download

Hi Jianfei,

Header comments are easy to understand. Implementation of functions looks good, matching the APIs. Variables look good, matching the database design. SQL queries are valid and logically correct. This controller for home page is ready for delivery.

Best regards

Jiawei

From: Jianfei Zhao
Sent: Friday, November 30, 2018 10:05:54 PM
To: Jiawei Xu

Figure 2. Code review by Jiawei

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. The pictures above are the screenshots of the examples of their code review, with the source code updated on GitHub.

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.
- Posted sales items are being protected. For example, images are protected from being messed up. This is achieved by storing all the sales items in MySQL database, with protected db access.

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 4 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**