SW Engineering CSC648/848 Fall 2022

GatorExchange

(WWW site for Buy/Sell/Share of Digital media exclusively for SFSU students and faculty)

Team 4

Mahisha Patel: mpatel17@mail.sfsu.edu (Team Lead)
Sudhanshu Kulkarni (GitHub Master)
Sophia Chu (Front End Lead)
Jerry Liu (Back End Lead)
Ekarat Buddharuksa
Ruben Ponce

Milestone 4

December 9 , 2022

Date submitted for review	Date revised
December 9, 2022	Created as the first version

Table Of Contents

1. Product Summary

- Name of the Product: **GatorExchange**

- Description: GatorExchange provides an easy-to-share, sell, and buy website for exchanging digital media among SFSU students and staff members. The users of the website can search digital materials related to their interests without any login and sign-up process. It enables the students and staff members of the SFSU to contact to the owner of the media for buying or using it for free.

- Major Functions:
    1. Login: One can login in GatorExchange Website to be able to message seller or upload any media for sharing/selling.
    2. Registration: One can register to become a GatorExchange member to be able to post media and contact seller.
    3. Browsing: One can browse through different pages from navigation bar.
    4. Homepage: One can view recently approved media postings from the home page.
    5. Search: One can search the list of digital media posted by SFSU students and staff based on different categories and search query.
    6. Post: One can post the digital media for sharing or selling.
    7. Post Detail: One can view the post details of any digital media.
    8. Contact Seller: One can contact the seller by sending a message along with his contact details.
    9. Dashboard: One can view his/her own posted media on dashboard.
    10. Inbox: One can see the list of messages received by other users.
    11. Admin Approval: The admin approves the posts using SQL Workbench to be seen by the other users.
    12. Google Analytics: The admin can view the analytics of the GatorExchange Website.
    13. About: One can see the brief details of the developers of GatorExchange.

- Uniqueness:
    1. GatorExchange is exclusive to SFSU, it allows searching from limited and most relevant materials posted by someone from the same community.
    2. The buyer can send a message to the seller for any digital media that he/she is interested in.
    3. The Homepage will show the most recent 10 posts for latest updates.

- URL to the product: http://54.200.101.218/

2. Usability Test Plan

Usability Test: **Search Functionality**

- Test Objectives:
    1. The main objective is to test search functionality whether it returns results relevant to the search query.
    2. The test objective is to check if the user interface for this functionality is easy to access and use.
    3. The time for search query execution is to be tested.

- Test background and setup:

  System Setup:

  The users are required to have a device i.e., (Computer / Laptop) with stable internet connection and go to GatorExchange URL from Google Chrome or Mozilla Firefox browser.

  Starting Point:

  The user has to go to GatorExchange URL, enter the search query in the search bar and select a category from dropdown and the relevant list of digital media should appear. The length of the search query should not exceed 40 characters.

  Who are the intended users?

  The intended users for the website are SFSU students and staff and they can be a guest, registered or admin users. The guest users can browse and search but have to get registered if they wish to contact seller or post digital media via lazy registration.

  URL of the system to be tested:

  http://54.200.101.218/

- Usability Task Description:

  The user has to search for a digital media related to some specific keyword or interest and choose the category. The search results should display relevant search results based on the search query and category field.

  The following 4 cases should be tested before filling out the above survey:

1. User can directly click on search button without typing or selecting anything.

2. User can select the category and click on search.

3. User can simply type some search query and click on search.

4. User can select the category from the dropdown and type the search query and then click on search.

- Evaluation of Effectiveness:

  The effectiveness will be measured as follows:

Test/Use Cases	% Completed	Errors	Comments
Search Query			
Search Category			
Contact Seller			

- Evaluation of Efficiency:

  The evaluation of efficiency can be measured by the time taken for executing the search query, Number of clicks, Number of screens and Number of pages of instructions.

- Evaluation of User Satisfaction:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The search results were useful					
The category dropdown was helpful					
The time taken for search was less					
The search process was smooth and easy					
The UI was convenient to use					

3. QA Test Plan

- Test Objectives:

  The objective is to test search, post and contact seller as a guest/registered user.

- HW and SW setup:

  HW Setup: A computer or laptop device with stable internet connection.

  SW Setup: The below software setup is required while testing on local machine, otherwise a browser with internet supportability (Chrome/Firefox) is sufficient.

  Server Host: AWS EC2
  Operating System: Ubuntu 18.04
  Database: MySQL 8.0
  Web-Server: Nginx 1.22.0
  Server-Side Language: Python
  Browser: Chrome/ Firefox
  Additional Technologies:
  Web Framework: Flask, React, Node
  IDE: VS Code/PyCharm
  Web Analytics: Google Analytics

- Feature to be tested:

  Test Sequence

  Search:

  Go to GatorExchange via http://54.200.101.218/, locate the search bar, type the search query and select the category from category dropdown beside the search bar and finally click on “search” button.

  Upload Post:

  Go to GatorExchange via http://54.200.101.218/, Click on Upload in the navigation bar, fill the upload post form (title, description, price, media type, media file) and click on “post” button.

Contact Seller:

First, perform the steps of Search as written above, then from the search results, click on any post detail, and locate “Contact Seller” button at the end of the media detail page and click on it. A pop up will appear with pre-filled text message, edit and click “Send”.

- QA Test Plan:

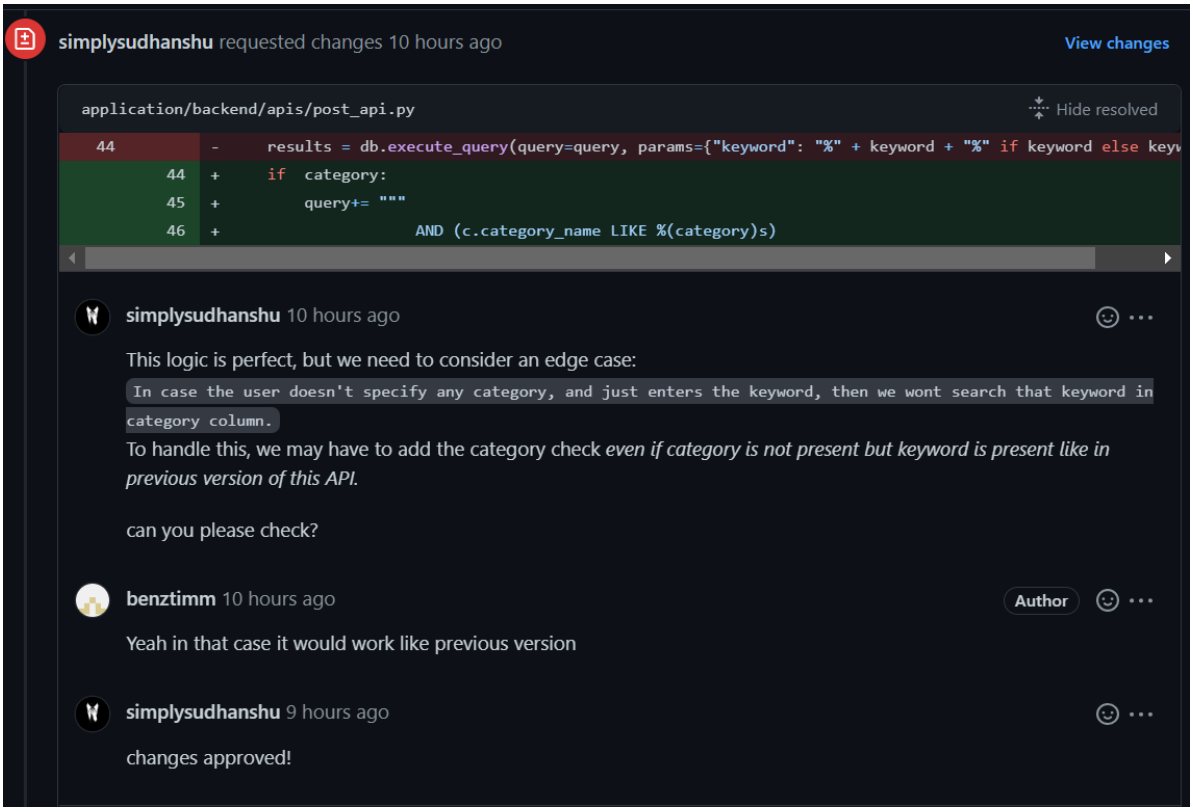
Test #	Test Title	Test Description	Test Input	Expected Correct Output	Test Result (Pass / Fail)
1.	Search result	Test search results (%like logic shall be tested)	Type “CS” in the search field and click search	List of 3 media that has “CS” either in description or title	Pass in Chrome/ Pass in Firefox
2.	Search Input Validation	Test the search input field validation, i.e., no special characters should be allowed	Type any special character ($, %, ^, &, *, etc.) in search field	An alert message pops saying “Search input must be alphanumeric”	Pass in Chrome/ Pass in Firefox
3.	Search query length validation	Test the search query length, i.e., the search query should not exceed 40 characters	Type any search query that exceeds 40 characters in total	More than 40 characters are not allowed and one can not type more than 40. A message pops saying maximum allowed characters are 40.	Pass in Chrome/ Pass in Firefox
4.	Upload Post	Registered and logged in User is uploading the media for selling	Submit the media file, title, description, category, and price for posting it on the website.	An alert message pops saying “Post will be approved within 24 hours.”	Pass in Chrome/ Pass in Firefox
5.	Contact Seller	Guest User tries to contact seller but lazy	Opens the post details and clicks on	Lazy Registration (Registration form pops us)	Pass in Chrome/ Pass in Firefox

		registration happens.	contact seller button		
6.	Test Post Data	Admin will test the Media post data on the workbench.	The admin opens the workbench and have login credentials to access the database.	Checks all the data (media file, category, title, price and description) from the Post Table in Workbench.	Pass in Workbench

4. Code Review

Code review was done in the following way:

1. The code was submitted by a backend team member for review to the github master and team lead.
2. Github Master asked for some clarification which was followed by the reply from the backend team member.
3. Github Master reviewed and approved.
4. Team Lead reviewed the entire code and made some comments and approved.



Review by Github Master

Review by Team Lead

5. Self-check on best practices for security

Asset to be protected	**Types of possible/expected attacks**	**Strategy to mitigate/protect the asset**
The user's email/password in database (High value)	Unauthorized user gains access to the database (High probability)	The password of the users will be encrypted in the database. The database is accessible by the admin only.
The user's email/password in database (High value)	SQL Injection through login/registration forms (Medium probability)	Input Validation on the field to have only alphanumeric characters. The email of the user should be the SFSU school email, i.e., sfsu.edu at the end.
The media/category database (Low value)	SQL Injection through search bar (Medium probability)	Input Validation on the search bar to have only alphanumeric characters with a maximum total length of 40 characters.

		The backend server has connected to the database via 'pymysql' library and can handle SQL injection vulnerabilities.
Application's hosting service credentials	Unauthorized user makes application unavailable (Low probability)	Strict management to credentials and keys to web hosting server.
An individual user's account (Low value)	Unauthorized access to the user's account (Medium probability)	The admin can use the database stored e-mail for verification in order to restore or reset a user's login credentials.

6. Self-check of the 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. **DONE**

2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers. **DONE**

3. All or selected application functions must render well on mobile devices. **ON TRACK**

4. Data shall be stored in the database on the team's deployment server. **DONE**

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

6. Privacy of users shall be protected. **DONE**

7. The language used shall be English (no localization needed). **DONE**

8. Application shall be very easy to use and intuitive. **DONE**

9. Application should follow established architecture patterns. **DONE**

10. Application code and its repository shall be easy to inspect and maintain. **DONE**

11. Google analytics shall be used. **ON TRACK**

12. No e-mail clients shall be allowed. Interested users can only message to sellers via in-site messaging. One round of messaging (from user to seller) is enough for this application. **DONE**

13. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI. **DONE**

14. Site security: basic best practices shall be applied (as covered in the class) for main data items. **DONE**

15. Media formats shall be standard as used in the market today. **DONE**

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

17. The application UI (WWW and mobile) shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Fall 2022. For Demonstration Only" at the top of the WWW page nav bar. **DONE**