

# SW Engineering CSC648/848 Spring 2020

## Gator Media

Team Number 5

Ahmad Moussalli (Team Lead) [amoussalli@mail.sfsu.edu](mailto:amoussalli@mail.sfsu.edu)

Felipe Carmona - Frontend Team Lead

Pak Lik Siu - Backend Team Lead

Viral Amin - Github Master

Olivia Wissig - Frontend Team Member

Nicole Bernardo - Frontend Team Member

Milestone 4

18 May 2020

History table: V 1.0

## 1) Product summary:

a) Product name: Gator Media

b) Product functions:

- Unregistered and registered users shall be able to Register and Sign in for the Gator Media Store.
- Unregistered and registered users shall be able to search the store content.
- Unregistered and registered users shall be able to view any listing individually.
- Unregistered and registered users shall be able to view the content's thumbnail from the main page.
- Registered users shall be able to download free media.
- Registered users shall be able to message other users directly about buying media content.
- Registered users shall be able to provide a product description when they post an item.
- Registered admin shall be able to approve or decline the pending posts from users.

c) What makes our product unique:

Our product offers a system specifically for San Francisco State University students and faculty, where they can purchase and or sell digital media for extra income. Our site ensures that digital media being sold and or shared won't get stolen and or plagiarized through admins who will approve each post. There are no advertisements and or web trackers, and students and faculty who create accounts on our site will be verified so buyers will know who they're purchasing from. Only San Francisco State University students and faculty will be able to access our site, however we are hoping to expand to other campuses as well.

d) LINK (We will provide this separately)

## 2) Usability test plan:

Usability test plan for search function:

- Test Objective: Test the ability of searching the content of the website for both registered and unregistered users while validating the accuracy of the search results.
- Test background and setup:
  - System setup:
    - Install docker for the host machine.
    - Clone the repository on the host machine.
    - Via a terminal, navigate the Application directory.
    - Run docker-compose up.
  - Starting Point:
    - After the application runs correctly.
    - Check the existence of the search bar and the search button.
    - Check the existence of a few demo content.
  - Intended users are the students and faculty of San Francisco State University.
  - URL
- Usability Task description:
  - Testing the correct functionality of the search bar
  - Tasks to do the test:
    - Test One
    - After the site has fully loaded
    - Enter a word (type) in the search bar for example, “ocean”
  - Questions about about the results:
    - Effectiveness:
      - Did the site only return contents that contain the word “ocean”?
    - Efficiency:
      - Did the site return the results in less than 30 seconds?
    - Lickert scale questions:
      - Please answer from 1 - 5 (5 is strongly Agree and 1 is strongly disagree)
      - As a user, you felt that accessing the site and finding the search bar was easy:
        - 1 Strongly disagree
        - 2 Disagree
        - 3 Neutral
        - 4 Agree
        - 5 Strongly agree

- As a user, you were satisfied with the search results relevance:
  - 1 Strongly disagree
  - 2 Disagree
  - 3 Neutral
  - 4 Agree
  - 5 Strongly agree
- As a user, you were satisfied with the time it took to get the results back:
  - 1 Strongly disagree
  - 2 Disagree
  - 3 Neutral
  - 4 Agree
  - 5 Strongly agree

### 3. QA test plan:

- Test objectives
  - Test the ability of searching the content of the website for both registered and unregistered users while validating the accuracy of the search results.
- HW and SW setup
  - Hardware:
    - MacOS running Docker
  - System setup:
    - Install docker for the host machine.
    - Clone the repository on the host machine.
    - Via a terminal, navigate the Application directory.
    - Run docker-compose up.
  - Starting Point:
    - After the application runs correctly.
    - Check the existence of the search bar and the search button.
    - Check the existence of a few demo content.
  - Intended users are the students and faculty of San Francisco State University.
  - URL
- Feature to be tested
- QA test plan (suggested format for QA test plan table)

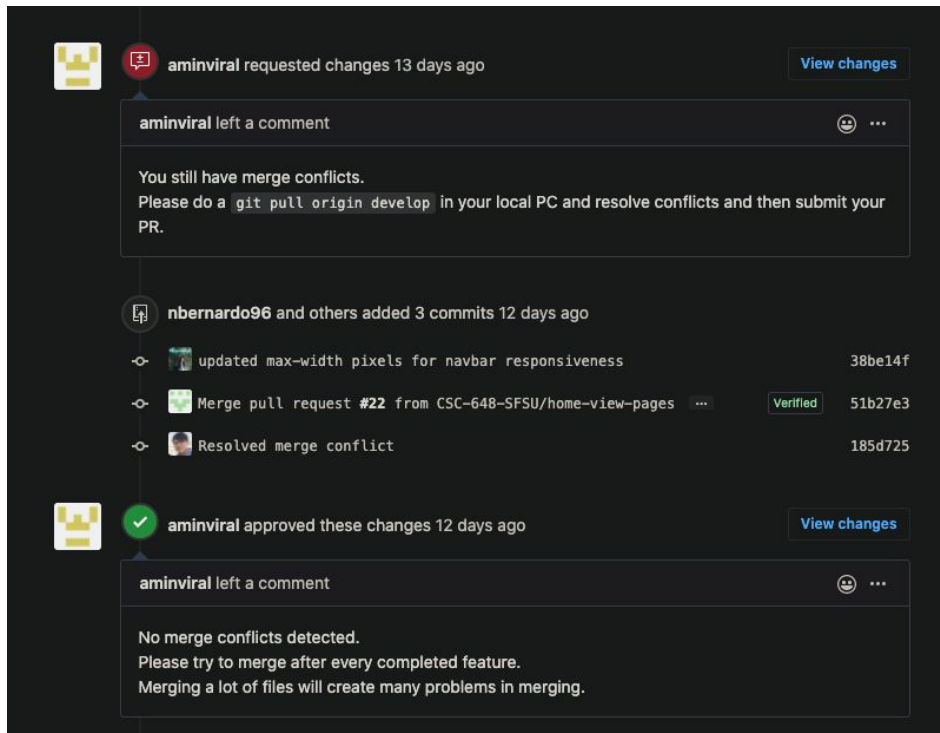
Test #	Test title	Test Description	Test input	Expected correct output	Test Results (PASS/FAIL )
01	Search bar exist	The search bar should be visible in the main page of the application at all time	N/A	Visible search bar	PASS
02	Search button works	The search button should update the main page content when pressed	Text “Cat” in search bar Search button press	Update in the main page	PASS
03	Search results are relevant	The main page results should be relevant to the	Text “Cat” in search bar Search button	Shows cat content	PASS

		searched term.	press		
--	--	----------------	-------	--	--

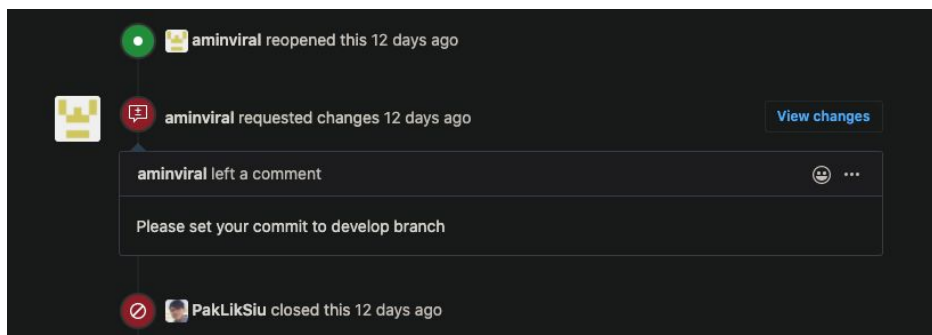
When performing the testing, we followed the table above and tested the feature on two different types of browsers.

## 4. Code review:

- Coding style used: Google Javascript coding style (<https://google.github.io/styleguide/jsguide.html>)
- Example of asking for feedback on the code:



### Example for the feedback:



Example of the code:

```
165 +
166 +
167 +
168 + exports.getIndex = (req, res, next) => {
169 +   db.query('SELECT * FROM sys.media_table', function (error, results, fields) {
170 +     items= results
171 +   });
172 +   db.query('SELECT DISTINCT category FROM sys.media_table', function (error, results, fields) {
173 +     categories= results
174 +   });
175 +   if (req.isAuthenticated()) {
176 +     res.render('index', {
177 +       user: req.user,
178 +       item: items,
179 +       category: categories,
180 +       searchCate: "All categories",
181 +       title: ""
182 +     })
183 +   } else {
184 +     res.render('index', {
185 +       user: "",
186 +       item: items,
187 +       category: categories,
188 +       searchCate: "All categories",
189 +       title: ""
190 +     })
191 +   }
192 + }
```



## 5. Self-check on best practices for security

- Major assets we are protecting
  - SFSU students - we require an SFSU account when registering for the website
  - Customer data - in order to download a product/listing, the user must be signed in with their registered SFSU email.
  - Copyright rights - there are a select few users who are administrators and have the responsibility to review and approve posts to ensure they are free of any copyright infringements.
  - User authentication - in addition to requiring SFSU emails, the login process requires completing a captcha process.
- Major threats for our assets
  - SFSU students - anyone that is not an SFSU student and faculty cannot sign up for our site, they must provide a valid sfsu email address upon registration - this filters out unknown and unassociated users.
  - Customer data - our process avoids the possibility of user data getting leaked or used without permission by the owner.
  - Copyright rights - before the posts are published to the website for all users to see, they are filtered through an approval process by administrators who review the uploaded content and ensure no rights are violated.
  - User authentication - the captcha implementation assures that a human is registering for our website, and not a bot or automated process that can be harmful or a threat to our website.
- We have confirmed that the passwords stored in the database are encrypted.
- Validation
  - Captcha
    - Google captcha
  - Email validation
    - Script written in code to validate email contains “@sfsu.edu”
  - Password protection
    - Encrypted in database
  - Search bar, not more than 50 characters
    - Custom lines of code that restrict input value to 50 characters or less
  - Upload only consists of images
    - The only type of files allowed in the upload are JPEGs (images)

## 6. Self-check: 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 two major browsers. **DONE**
3. Selected application functions must render well on mobile devices. **DONE**
4. Data shall be stored in the team's chosen database technology on the team's deployment server. **DONE**
5. Full resolution free media shall be downloadable directly, and full resolution media for selling shall be obtained after contacting the seller/owner **DONE**
6. No more than 50 concurrent users shall be accessing the application at any time. **DONE**
7. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users. **DONE**
8. The language used shall be English (no localization needed). **DONE**
9. Application shall be very easy to use and intuitive. **DONE**
10. Google analytics shall be used. **ON TRACK**
11. No email clients shall be allowed. **DONE**
12. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI. **DONE**
13. Site security: basic best practices shall be applied (as covered in the class) for main data items. **DONE**
14. Media formats shall be standard as used in the market today. **DONE**
15. Media material shall be either free or for sale, as determined by the media owner. **DONE**
16. Each media material shall have its license info as one of the following:
  - a) Free use and modification.
  - b) Free but only allowed for SFSU related projects.
  - c) For sale.**DONE**
17. Modern Software Engineering processes and practices shall be used as specified in the class, including collaborative and continuous Software Engineering development. **DONE**
18. The website shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Spring 2020. For Demonstration Only" at the top of the WWW page. (Important so as to not confuse this with a real application). **DONE**