SW Engineering
CSC648/848 Spring 2020
Gator Trade

By:
Team 03
Samuel Bahlibi (Team Lead) samuel.bahlibi@gmail.com
Qian Hu (Front-end Lead)
Tingfeng Wan (Back-end Lead)
John Joshua Gutierrez (Github Master)
Eric Ngo (Full-Stack Developer)
Marvin Nguyen (Front-end Team)
Local Team
Milestone 4
05/17/2020

May 18, 2020	Milestone Version 1
May 19, 2020	Finalized Version

Product Summary

Product Name: Gator Trade

Our Product: Gator Trade offers San Francisco State students and faculty a convenient way to post and access media pertaining to their classes and other activities through a web client service. We aim to build a local e-commerce market that will exclusively allow students and faculty to post, search, browse, buy, sell and trade items for San Francisco State University student projects and/or class material for faculty members. Gator Trade's content is exclusive only to San Francisco State's class materials, therefore eliminating the need to browse other sites for content related to the user's class. Gator Trade is unique in which users are able to filter searches through categories pertaining to SFSU specific departments. Users are also able to select a meetup location on the SFSU campus when choosing to buy a physical object. Various major locations are available to select when a user is uploading media to the website such as the Malcolm X plaza or the library.

Major Functions:

1. Users shall be browse for media on the website
2. Users shall be register an account
3. Users shall be filter media by SFSU media classes
4. Users shall be search for media
5. Users shall be able to log in
6. Registered users shall also do 1-5
7. Registered users shall download free media
8. Registered users shall be able to message another user
9. Registered users shall be able to upload digital media to the website
10. Registered users shall be able to view a list of media they posted and messages

Website URL: http://3.22.78.154:3000/home/home.html

Usability Test Plan

Test Objectives: Post an item and if successful be directed to the home page with the new post added.

Feature to be Tested: Posting an Item.

Test Background and Setup:

System Setup: A desktop running Windows 10, with all major update

Starting Point: http://3.22.78.154:3000/home/home.html, after logged in using:

Username: test@sfsu.edu

Password: password

Intended Users: Registered non-admin user

Metric: User-Friendly Interface, Pleasing Visual Scheme, Ease of Navigation, Ease to Perform Task

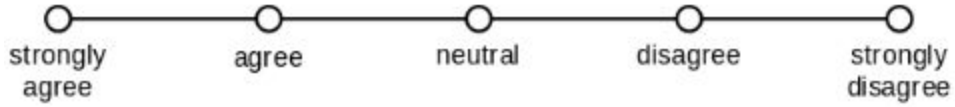
Usability Task description: After logged into the test account, attempt to create a posting that includes a file and a preview picture.

Effectiveness: Effectiveness can be measured by whether or not the user was able to create an entry, and if all media and contents are present.

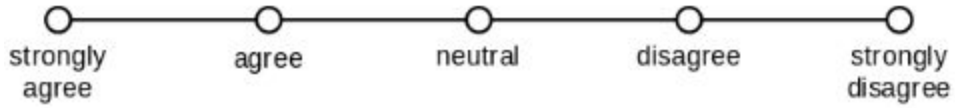
Efficiency: Efficiency can be measured by measuring the time elapsed to perform the task, with less time spent representing a more efficient interface, and measured by measuring the amount of click operations, with less clicks representing a more efficient interface.

Likert Scale Questions:

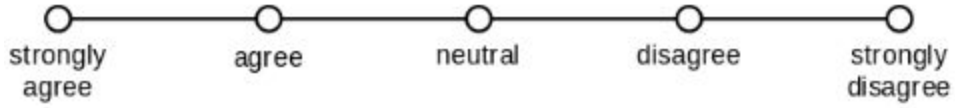
The website has an intuitive interface.



The website is very easy to navigate.



It was easy to create new postings to the website.



QA Test Plan

Test Objectives: Post an item and if successful be directed to the home page with the new post added.

Feature to be Tested: Posting an Item.

HW Setup: A desktop with the current Windows 10 update.

SW Setup:

- Go to http://3.22.78.154:3000/.
- Login with the following credentials:
    - test@sfsu.edu
    - password
- Click on the “Create Post” which is the first button located at the top right of the navbar.
- Fill out the input fields and click create post when finished
- The user should be redirected to the home page.
- Wait for the item to be approved by an administrator.
- Once the administrator approves of the item, click the posted item on the “Recent Posting” area and check whether the entered data persists.

Test Plan:

Test #	Test Title	Test Description	Test Input	Test Output	PASS/FAIL
1	Google Chrome Test	Test that making a post stores the information submitted on the current Chrome build.	**Look at **SW Setup above** for directions on how to create a post to make a test input** **Title:** “Milestone 4” **Category:** “Science” **License**: “Free but only allowed for” **Price:** “0.00” **Physical Object:** False **Description**: “A document	Once, the item is approved by an administrator, an item card in the Recent Posts in the home page with the price and date information along with a default preview image. When the item is clicked, the description and licensing should be displayed.	PASS

			guideline of how to pass CSC 648” **Upload File:** “CSC648848Spring 2020 Milestone4” **Image Preview:** Leave empty		
2	Firefox Test	Test that making a post stores the information submitted on the current Firefox build.	**Look at **SW Setup above** for directions on how to make a post for a test input** **Title:** “Carpe Diem” **Category:** “English” **License:** “For sale” **Price:** “$10” **Physical Object:** true **Meetup Location:** “Library” **Description**: “A bucket list journal” **Image Preview:** “book.png”	Once, the item is approved by an administrator, an item card in the Recent Posts in the home page with the price and date information along with a book preview image. When the item is clicked, meet up location, description and licensing should be displayed.	PASS
3	Safari Test	Test that making a post stores the information submitted on the current Safari Test build.	**Look at **SW Setup above** for directions on how to make a post for a test input** **Title:** “Denim Jacket” **Category:** “Other” **License:** “For sale” **Price:** “$70” **Physical Object:** true **Meetup Location:** “Malcolm X Plaza” **Description:** “Denim jacket with lilo and stitch painting” **Image Preview:** from an iphone	Once, the item is approved by an administrator, an item card in the Recent Posts in the home page with the price and date information along with a denim jacket preview image. When the item is clicked, meet up location, description and licensing should be displayed.	PASS

Code Review

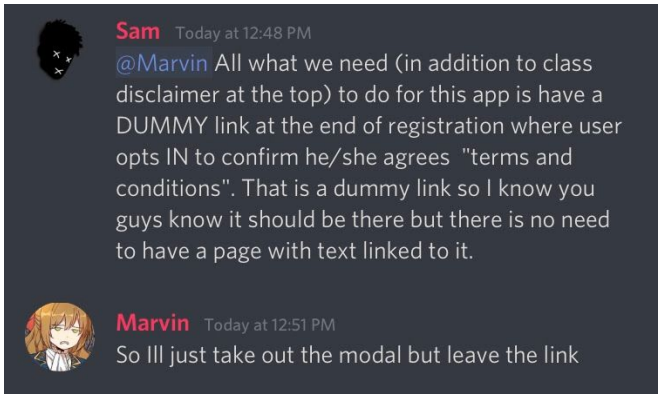
a) Code Style

- Use a semicolon after every declaration
- Use double quote ("") around attribute values
- Use single quotes (')
- Use camelCase for variable and function names
- Declare variable with let ,var and const
- Separate rules by new lines and put a blank line between rules.
- Use a new line for every element

b)Code Review and Comment

```
/**
 * checks if text fields and recaptcha is valid
 * if valid, send to createuser() to create user
 */
function validateForm() {
  event.preventDefault();

  let captcha = document.getElementById('captcha').value;
  console.log(captcha);

  if (validateFields()) {

    //sends captcha response to google to verify that it's correct
    axios.post("http://3.22.78.154:3000/user/authenticate", {
      captcha
    })
      .then((res) => {
        console.log(res.data);
        if (res.data.success) {
          document.getElementById("recaptcha-error").innerHTML = "";
          createUser();
        }
        else {
          document.getElementById("recaptcha-error").innerHTML = "Incorrect recaptcha";
        }
      })
      .catch((err) => {
        console.log(err);
        document.getElementById("recaptcha-error").innerHTML = "Please verify that you are a human";
      });
  }
}
```

```
/**
 * Takes the inputs from the form and puts them into the
 * database on the server using axios.
 */
function createUser() {

  let email = document.forms['register']['email'].value;
  let hashedPassword = md5(document.forms['register']['password'].value);
  let firstName = document.forms['register']['first-name'].value;
  let lastName = document.forms['register']['last-name'].value;
  let phoneNumber = document.forms['register']['phone-number'].value;
  let faculty = false;

  if (document.forms['register']['school-role'] == "Faculty") {
    faculty = true;
  }

  axios.post('http://3.22.78.154:3000/user/signup', {
    email: email,
    hashed_password: hashedPassword,
    first_name: firstName,
    last_name: lastName,
    is_faculty: faculty,
    phone_number: phoneNumber
  })
    .then((res) => {
      console.log(res.data);

      if (res.data.status) {
        window.location = "register-success.html";
      }
    })
    .catch((err) => {
      document.getElementById("email-error").innerHTML = "Email already exists. Please enter another email.";
      document.getElementById("email").style.border = "1px solid red";
      console.log(err)
    });

}
```



Security Best Practices

Major Assets Protected:

- Registered User Records
    - Threats: User's credentials may be sniffed if not protected properly. Users may be able to alter the database records.
    - Measures: User passwords are hashed before being sent over the internet. No plaintext passwords are stored on the database. DB access controls are password protected.
- Digital Media
    - Threats: Users shall not be able to download the contents of non-free digital media. Users could potentially corrupt the media.
    - Measures: Conditions are checked on the server side to only show the contents of free digital media, preventing the downloading of non-free digital media. Filesystems have access controls that prevent writes from users.
- Input Data / Search Filters:
    - Threats: User supplied input data may lead to Server Side Injection attacks.
    - Measures: User inputs are sanitized before being used in queries.

Confirm Database Passwords:

email	hashed_password
admin_faculty_test@sfsu.edu	0b14d501a594442a01c6859541bcb3e8164d183d32937b851835442f69d5c94e

Confirm Input Validation:

```
sanitizer: (str) => {
    if ( !str || str === '')
        return '';
    const map = {
        '&': '&amp;',
        '<': '&lt;',
        '>': '&gt;',
        '"': '&quot;',
        "'": '&#x27;',
        "/": '&#x2F;',
    };
    const reg = /[&<>"'/]/ig;
    return str.replace(reg, (match)=>(map[match]));
},
```

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 In Progress
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 In Progress
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 **Done**
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 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 SE processes and practices shall be used as specified in the class, including collaborative and continuous SW 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**