

SW Engineering CSC648/848 Spring 2022

GatorBay

Team 6

Team Lead: Bobby Roots broots@mail.sfsu.edu

Backend Lead: Michael Davichick

Backend Member: William Rattan

Frontend Lead: Michael Wang

GitHub Master: Michael Almeda

Milestone 4

Date of submission: 05/14/2022

1) Product summary

a) Name:

- i) GatorBay

b) Describe your Product:

- i) All of us from Team 06 acknowledge that Students, Faculty, and Staff could use a website that caters more to them. That is why Team 6 has decided to build GatorBay. GatorBay will offer the exchange of goods such as Furniture, Clothes, Books, and electronics. What makes GatorBay so unique is that it is exclusively for SFSU Students, Faculty, and Staff. That along with our 24/7 operating hours at safe locations throughout SFSU we help to assist in the ease of making transactions in a non-harmful environment. Team 06 is dedicated to making this dream a reality.

c) Itemized list of ALL major committed functions:

- i) System allows unregistered users to browse posts
- ii) System allows unregistered users create an account
- iii) System allows unregistered users to log in
- iv) System allows registered users to have all the rights of unregistered users
- v) Registered users are able to buy displayed products
- vi) Registered users are able to post products
- vii) Registered users are able to sell products
- viii) Registered users are able to see available messages
- ix) Admin users have all the abilities as registered users.
- x) Admin users can approve a post request
- xi) Admin users can disapprove a post request.
- xii) Admin users can remove a post.

d) Unique:

- i) Search has the ability to filter results from four main categories.

e) URL to your product accessible to instructors:

- i) <https://inlaid-crane-341607.uc.r.appspot.com/>

2) Usability test plan

- a) Major function to be tested: Search
- b) Test objectives: Testing usability of search is to obtain the website's search abilities. When the intended user inputs data via search the correct results will be displayed within the results.
- c) Test background and setup:
 - i) Start system desktop
 - ii) Go to this URL : <https://inlaid-crane-341607.uc.r.appspot.com/>
 - iii) Load application
 - iv) Intended users are SFSU Faculty, Staff, and Students with low level (basic) knowledge of computer technology
- d) Usability Task description:
 - i) Task display only books from the URL
 - ii) Computer state search bar is blank
 - iii) Successful compilation When book results are only displayed
 - iv) Benchmark Completed within 1 minute
- e) Evaluation of Effectiveness:
 - i) Percentage of people who could find the search bar
 - ii) Percentage of people who could find the search button
 - iii) Percentage of people who found the correct results
- f) Evaluation of Efficiency
 - i) Average number of clicks user did to complete task
 - ii) Average total time it took to complete task
 - iii) Average number of screens user saw before task was complete
- g) Evaluation of user satisfaction:

Question	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
Easy to locate the search bar.					
Results were correct.					
Overall experience was pleasant.					

3) QA test plan

a) Test objectives:

i) Testing that the search is working as per specs

b) HW and SW setup (include URL):

i) Go to your laptop

ii) Open your firefox, edge, or safari browser

iii) Go to URL <https://inlaid-crane-341607.uc.r.appspot.com/>

c) Feature to be tested

i) Searching for an item

d) QA Test plan in table format:

Test #	Test title	Test description	Test input	Expected correct output	Test result Pass/Fail
1	Search Test	Testing for correction of search function	Input book into the field of text search	Results should show two separate books stored in the DB.	Pass
2	Search Test	Testing for correction of search function	Input chair into the field of text search	Results should show no chairs because there are none stored in the DB.	Pass
3	Search Test	Testing for correction of search function	Input jeans into the field of text search	Results should show one pair of jeans stored in the DB.	Pass

4) Code Review

Screenshots of email messages:



Michael Andrew Davichick

To: Michael Anthony Almeda

Cc: Bobby Roots

Hi Michael Almeda,

When you have a minute, please do a QA review / usability test for the following features as per Milestone 4, section 4:

- Register User
- Sign In

You can find the following code segments as follows:

- Register User - application/my-app/client/src/CreateUser.js
 - Password Validation: line 48
 - CheckBox Validation: line 60
 - Form Field Validation: line 99 +
- Sign In - application/my-app/client/src/SignIn.js
 - Verify User: line 53
 - Form Field Validation: line 93+

Please submit your findings as per protocol of Milestone 4.

Thank you in advance for your assistance.

Regards,
Michael Davichick



Michael Anthony Almeda

To: Michael Andrew Davichick

Cc: Bobby Roots

Hi Michael Davichick,

Thank you for providing a thorough outline to perform a QA review/usability test.

Below is a summary of my review on the code segments you provided:

- Register User
 - Password Validation:
 - The field validation function used for password validation contains good use of in-line comments.
 - CheckBox Validation:
 - No in-line comments are used in the implementation of the checkFunction().
 - Contains consistent naming conventions among variables.
 - Form Field Validation:
 - Great use of in-line comments.
 - The formatting of every onChange event stays consistent with every field the user needs to enter.
- Sign In
 - Verify User:
 - Spacing for the setTimeout function is inconsistent and should be properly formatted.
 - Good use of in-line comments and consistent variable naming.
 - Form Field Validation:
 - Clean formatted code.
 - In-line comments well place.
 - setEmail, setPassword, etc... all were consistent with one another.

Overall, the code segments I reviewed contained useful in-line comments, header comments, and consistent naming conventions. A few more comments could be used in some areas. The header comments provide sufficient information about the contents of the file. Class methods and variable naming conventions stay true to previous naming conventions stated in Milestone 2.

In addition to the reviewed code segments, Github commits did provide detail on what was being committed to the repository, however including the name of the file being committed should be included in the commit message.

Great work!

Best regards,
Michael Almeda



Sat 5/14/2022 8:30 PM

Screenshots of commented reviewed code:

```
1  /*****
2  * Project: SFSU Software Engineering Project CSC648-848, Spring 2022
3  * Author(s): Michael Davichick
4  * Team: 06
5  *
6  * File: CreateUser.js
7  *
8  * Description: This file allows the user to create an account with our
9  * web app by having them provide their information such as name, phone
10 * number, email, and password.
11 *
12 *
13 *****/
14
15 /*          --- CODE REVIEW ---
16 *
17 * @Header Comment
18 * Great use of a basic header comment. It provides enough information
19 * to get a good understanding of this file's purpose and it's author.
20 *
21 */
22
23 // import './App.css';
24 import './Forms.css';
25 import React from "react";
26 import { useState } from "react";
27 import Axios from "axios";
28 import { useNavigate, useLocation, Link } from "react-router-dom";
29 import Footer from "./Footer";
30 import Header from "./Header";
31 import { Form } from "react-bootstrap";
```

```
68 /*          --- CODE REVIEW ---
69 *
70 * @function checkFunction()
71 * A comment to describe what this function's purpose is would be ideal
72 * for keeping the in-line comment consistency.
73 * Overall, good consistency with in-line comments.
74 *
75 */
76 function checkFunction() {
77     if ((document.getElementById("check")).checked) {
78         alert("The checkbox is checked");
79     }
80     else {
81         alert("The checkbox is not checked")
82     }
83 }
```

```
1  ✓ /*****
2   * Project: SFSU Software Engineering Project CSC648-848, Spring 2022
3   * Author(s): Michael Davichick
4   * Team: 06
5   *
6   * File: SignIn.js
7   *
8   * Description: This file allows the user to sign into their account
9   * by providing their email and password. If they don't have an account
10  * they can choose the link to create an account.
11  *
12  *
13  *****/
14
15  ✓ /*          --- CODE REVIEW ---
16  *
17  * @Header Comment
18  * Great use of a basic header comment. It provides enough information
19  * to get a good understanding of this file's purpose and it's author.
20  *
21  */
22
23  ✓ import './Forms.css';
24  import axios from "axios";
25  import React from "react";
26  import { useState } from "react";
27  import { Link } from "react-router-dom";
28  import { useNavigate, useLocation } from "react-router-dom";
29  import Footer from "../Footer";
30  import Header from "../Header";
```

```

60  /*          --- CODE REVIEW ---
61  *
62  * @function setTimeout()
63  * Spacing among teh if statements is not consistent.
64  * Consider reformatting.
65  *
66  */
67
68  const verifyUser = () => {
69      // Send our User Parameters to the backend for retrieval
70      SignIn();
71
72      // If our user List is not null, then there was a successful match between user login and backend
73      setTimeout(function(){
74          if(userID>=0){
75              console.log("success");
76              console.log("User name: " + userName);
77              console.log("User ID " + userID);
78              // Once in this scope, our user has been logged in and we can add them to the session
79              sessionStorage.setItem("id", userID); // User ID is a unique key in Users Table to identify individual users
80              sessionStorage.setItem("fname", userName);
81              sessionStorage.setItem("loggedIn","true"); // loggedIn is a boolean value used to change CSS properties when logged in
82
83
84              if (location && location.state && location.state.referrer) {
85                  navigate(location.state.referrer);
86              } else {
87                  navigate('/');
88              }
89
90          }
91          if (userID < 0){ // Else invalid login, prompt user to try again
92              // console.log("passwords do not match!");
93              alert("Email or Password was invalid, Please try again.");
94          }
95      }, 1000);
96  };

```


5) Self-check on best practices for security

a) Make a table

Asset to be protected	Types of possible/expected attacks	Strategy to mitigate/protect the asset
User information	Using non college email	Verify sfsu.edu during registration
Product information	Malicious code injected	Admin approves prior to going live
Search	SQL attack	Validation Process

6) Self-check of the adherence to original Non-functional specs:

- System requirements: describe the system requirements (architecture, system services, networks, platforms etc.)
 - Server Host: Google Done
 - Operating System: Linux Done
 - Server Database: MySQL Done
 - Web Server Node JS Done
 - Server-Side Language: JavaScript Done
 - Additional Technologies: Done
- Web Framework: Bootstrap Done
- IDE: Visual Studio, VS code Done
- Usability requirements: describe specific usability and UI issues, users who will use the system, delivery clients
 - Anyone on the website can search for posts without logging in. Done
 - Registered users can create posts and see the user's contact information from already approved posts. Done
 - Admins are required to approve posts prior to being visible to the Public. Done
- Performance requirements:
 - Average latency will be 150 ms for refreshing pages and after posting or creating a new account. Done
 - Accuracy will be 97 percent so every one hundred requests sent to the server, a maximum of 3 errors shall be thrown. Done

- Bandwidth is a maximum of 7 gigabytes of flow from up to 50 external IP addresses and a maximum of 3 gigabytes of flow from one IP address. On track

- Storage, security, environmental requirements:

- MySQL database provides secure connections, authentication services, fine-grained authorization and controls as well as data Encryption. On track

- All passwords will have a separate encryption process prior to being stored on the back end for verification. On track

- Marketing, legal requirements (logos, branding, licensing):

- No logos currently exist or are present for GatorBay