

# SW Engineering CSC648 / 848 Spring 2022

## BusyGator

<http://3.23.79.193/>

Team 04		
Role	Name	Email
Team Lead	Samantha Saxton-Getty	ssaxtongetty@mail.sfsu.edu
Github Lead	Vishal Ramanand Sharma	vsharma5@mail.sfsu.edu
Front End Lead	Elyssa Mari Tapawan	etapawan1@mail.sfsu.edu
Back End Lead	Aaron Carlson	acarlson8@mail.sfsu.edu
Front End	Abdullah Sharaf	fabdullah1@mail.sfsu.edu
Front End	Siqi Guo	sguo4@mail.sfsu.edu
Back End	Janvi Patel	jpatel6@mail.sfsu.edu

May 21, 2022

Milestone 05

Date	Version
05.21.22	Finished Version 01 submitted

# Table of Contents

<b>Product Summary</b>	4
<b>Milestone Documents - M1 - M4</b>	5
<b><i>Milestone 01</i></b>	5
Executive Summary	6
Personae and Main Use Cases	7
Main Data Items and Entities - Glossary / Definitions	10
List of Functional Requirements	12
List of Non-Functional Requirements	14
Competitive Analysis	15
High-level System Architecture and Technologies Used	16
Team and Roles	16
Checklist	17
<b><i>Milestone 02</i></b>	18
Executive Summary	19
Main Data Items and Entities - Glossary / Definitions	20
List of Functional Requirements - Prioritized	21
UI Storyboards for Each Main Use Case	23
High Level Architecture and Database Summary	27
Key Risks	29
Project Management	30
<b><i>Milestone 03</i></b>	31
Summary of Milestone 3 ZOOM meeting review with Prof. Petkovic and plans for further development	32
<b><i>Milestone 04</i></b>	36
Product Summary	37
Usability Test Plan	38
QA Test Plan	40
Code Review	42
Self-Check on Best Practices for Security	45
Self-Check of the Adherence to Original Non-Functional Specs	46

<b>Product Screenshots</b>	48
<b>Database Organization</b>	55
<b>GitHub Organization</b>	57
<b>Google Analytics Statistic Plot</b>	59
<b>Project Management</b>	60
<b>Team Member Self Assessment and Contributions</b>	61

## **1. Product Summary**

**Application Name:** BusyGator

Our application, BusyGator, provides various materials to help SFSU students, staff, and faculty find everything they are looking for on campus all online. BusyGator is an e-commerce application that allows students, staff, and faculty to buy different items, from school materials to gym equipment, as well as list items for others to buy. The BusyGator has numerous functionalities and services. The main service the application will provide is allowing SFSU students, staff and faculty to post and / or buy textbooks, sports gear, research devices, and much more all online. The application has search options and categories to look for more specific items, and a sign up / sign in option to confirm users are SFSU students or faculty. BusyGator gives SFSU an opportunity to help their community by managing their resources within the application. The unique aspect of BusyGator is that all of the features and services in buying and selling material will be in one place and in one press of a button, where anyone can easily follow.

### **Priority 1 Feature List:**

#### Unregistered, Registered, and Administrator Users:

1. View Listings and Marketplace
2. Search Listings and Marketplace
3. Filter Listings and Marketplace

#### Unregistered Users:

4. Register Account

#### Registered Users:

5. Login Account
6. Post Listing on Marketplace
7. Message Seller to receive Approval or Denial of interest

#### Administrator Users:

8. Remove User
9. Remove Listing
10. Approve or Deny Listings

**URL:** <http://3.23.79.193/>

## 2. Milestone Documents - M1 - M4

SW Engineering CSC648 / 848 Spring 2022

### BusyGator

Team 04		
Role	Name	Email
Team Lead	Samantha Saxton-Getty	ssaxtongetty@mail.sfsu.edu
Github Lead	Vishal Ramanand Sharma	vsharma5@mail.sfsu.edu
Front End Lead	Elyssa Mari Tapawan	etapawan1@mail.sfsu.edu
Back End Lead	Aaron Carlson	acarlson8@mail.sfsu.edu
Front End	Abdullah Sharaf	fabdullah1@mail.sfsu.edu
Front End	Siqi Guo	sguo4@mail.sfsu.edu
Back End	Janvi Patel	jpatel6@mail.sfsu.edu

March 31, 2022

### Milestone 01

Date	Version
03.05.22	First Version 01 submitted
03.31.22	Revision completed per instructor feedback Version 02

## **Executive Summary**

There are many students who cannot find the resources they need to properly study, work out, or research. Our application, BusyGator, provides various materials to help SFSU students, staff, and faculty find everything they are looking for on campus all online. BusyGator is an e-commerce application that allows students, staff, and faculty to buy different items, from school materials to gym equipment, as well as list items for others to buy. This will motivate the SFSU community to use these products offered by the application to better their education and health.

The BusyGator will have numerous functionalities and services. The main service the application will provide is allowing SFSU students, staff and faculty to post and /or buy textbooks, sports gear, research devices, and much more all online. Each product that can be bought will show how much is in stock during the time of browsing, the price of the material, and the user selling the item. The BusyGator will provide a map pin-pointing various safe areas to pick up the product all around the SFSU campus. Other important functionalities the application will include are a search option and categories to look for more specific items, and a sign up / sign in option to confirm users are SFSU students or faculty. These services will provide users with an easy and accessible way to find various resources all in one place. BusyGator gives SFSU an opportunity to help their community by managing their resources within the application. Instead of going to different locations on campus to see if they can buy materials in person, the BusyGator will make this task a lot easier. It saves students, staff, and faculties the time and hassle in their already busy schedules. Also, this allows alumnis to give textbooks they no longer need to newer students who will utilize them more for classes. The unique aspect of BusyGator is that all of the features and services in buying and selling material will be in one place and in one press of a button, where anyone can easily follow.

We are Team 4 from the Software Engineering class of Spring 2022 consisting of Aaron Carlson, Siqi Guo, Janvi Patel, Samantha Saxton-Getty, Abdullah Sharaf, Vishal Ramanand Sharma, and Elyssa Mari Tapawan, with Samantha being the team lead. Other roles consist of Aaron being the back-end lead, Elyssa being the front-end lead, and Vishal being the GitHub lead. The back-end operates the parts of the application that aren't accessed by a user such as data organization. The front-end works on the parts of the websites that allow users to interact with the web features, such as the application design. The GitHub lead ensures that there are no errors and that the application is functional. We are a group of aspiring students learning the aspects of team software development by making an application ourselves. Our goal is to work together to create the BusyGator to help us experience and prepare ourselves for our future careers.

## Personae and Main Use Cases

### Personas:



John is a SFSU student and a part time worker (buyer).

#### **About John:**

- On a budget
- Doesn't have a car
- Busy
- Basic WWW skills
- Wants to cut down on spending too much money shopping online
- Doesn't want to spend a lot of money on school materials

#### **Goals and Scenario:**

John needs to buy a textbook for a class. He wants to check if he can afford to buy the book. If he finds the book that he is looking for and the price is good, he can contact the seller.



Jennifer is a full time SFSU student (seller).

#### **About Jennifer:**

- Needs money
- Is a senior
- Has a lot of textbooks from previous semesters
- Very busy
- Basic WWW skills
- Doesn't have a car

### **Goals and Scenario:**

Jennifer decides to list a textbook for sale. She wants to quickly list the book on the website. If she finds a buyer for the book, she can close the textbook listing from the website.



Mike is an administrator.

### **About Mike:**

- Likes problem solving
- Has basic database management skills
- Doesn't live on campus
- Doesn't like working on multiple problems at once

### **Goals and Scenario:**

Mike needs to disable a user from the website. He wants to check if he can find the user on the database. If he finds the user on the database, he would like to remove the user but he also wants the items they have listed deleted when the user is deleted.

### **Use Cases:**

#### **1. Search Item**

John is looking for a textbook for one of his classes. He goes on to the site and enters what he is looking for in the search bar. The results for his search show up on the page. He scrolls through the search results and picks the items that he is looking for.

#### **2. Create Account (buyer)**

John is not able to log in because he doesn't have an account. To create an account, he goes into the signup page on the website. He is prompted to fill out the signup form and he gets his new account right after he fills out his information. Now when he tries to login, he is successful.

### **3. Add Item to Cart**

John finds the book that he is looking for, and when he tries to add it to the cart, he is asked to login/signup. John logs in to the website and tries adding the item to the cart for the second time, this time he is able to add the item to his cart because he is logged in.

### **4. Buy an Item**

John wants to buy an item that he found on the website. He wants to contact the seller by sending them a message. When he tries to send the seller a message, he is prompted to log in. When he logs in, he is able to send the message.

### **5. List Item for Sale**

Jennifer wants to list an item for sale. She goes to our website to list the item. When trying to list the item, she is asked to log in / sign up. After logging in, she is able to list the item and its details. However, the item wouldn't be visible on the website right away because it will take some time for the approval process.

### **6. Create Account (seller)**

Jennifer is not able to list an item for sale on the website because she doesn't have an account. She wants to create an account on the website. Jennifer goes into the signup page on the website. She fills out the signup form and gets his new account. Now when she tries listing an item for sale, she is allowed to do so.

### **7. Update Price for Listed Item**

Jennifer wants to update the price for an item that she has already listed. First, shs logs into her account. After that, she searches for the item that she wants to update. When Jennifer finds the item, she proceeds with the update process and updates the item.

### **8. Close Item Listing**

Jennifer doesn't want to sell one of the items that she had listed previously. She wants to go on to her account to find the item, she is asked to login. After logging in, she searches for the item, and proceeds with the closing process.

### **9. Access Database**

Mike wants to access the database system because he wants to review the data. To access the database, he needs to log into the database system application with his admin credentials. When he is logged in, he is able to see that database..

## 10. Disable User

Mike wants to disable a user by removing them from the database system. He logs into the database system, and searches the user. When he finds the user that he wants to remove, he proceeds to delete it from the database.

## 11. Remove Products

Mike wants to remove a listed item from the database. He logs into the database system and searches for the product that he wants to remove. When he finds the item, he proceeds with the closing process and removes it.

## 12. Update User Information

Mike wants to update a user's information in the database. He logs into the database system, and searches the user. When he finds the user, he proceeds to update it on the database.

## 13. Approve Items

Mike wants to approve items to be published on the website. To approve the items, he needs to be logged into the database system application. When he logs in, he can start approving the items, based on the terms and conditions.

## Main Data Items and Entities - Glossary / Definitions

1. **User:** Individuals utilizing the product and application.

a. **Administrator:**

This user has access to all privileges given to registered and unregistered users (as listed below). In addition, they are able to modify permissions for all users as well as edit existing posts or remove them regardless of it being their own post. This user also has access to modify the database for the application. Administrators are required to approve products before they go live.

b. **Registered User:**

This user is allowed to search, filter, and view existing items listed, message the sellers, and post, remove, and edit items of their own. They are able to buy items. In addition, this User can report items to Administrators for review. This user will be required to have a username and password to login or register. This user is required to have an SFSU email (faculty or student).

c. **Unregistered User:**

This user is allowed to search, filter, and view existing items. They are not required to login or register unless they want to post, buy, or message the seller.

2. **Data Items:**

a. **Categories:**

All items are categorized into five categories. These individual data items will have a key indicating which category they fall into.

- Electronics
- Pets
- Home
- Recreational
- Books

b. **Items:**

This is a list of data items available for buy and sale by the User.

● **Electronics:**

- Batteries
- Calculator
- Headphones
- Kindle
- Laptop
- Laptop Charger
- Phone Charger
- Television

● **Pets:**

- Pet Bowl
- Collar
- Pet Bed
- Food
- Leash
- Toy
- Bone
- Potty Pads

● **Home:**

- Mattress
- Shower Curtain
- Pillow
- Lamp
- Sofa
- Chair

- **Recreational:**
  - Tent
  - Sleeping Bag
  - Fishing Pole
  - Bike
  - Badminton Racket
  - Tennis Racket
  - Basketball
- **Books:**
  - Dictionary
  - Encyclopedia
  - Atlas
  - SFSU Textbook
  - Literature Book
  - E-Book Code

c. **Transaction:**

This item logs transaction details including, but not limited to, the time it was bought, the user(s) involved, the cost, etc.

d. **User Registration:**

This item holds all of the registration information of the users including their username, hashed password, and permission level.

## **List of Functional Requirements**

### **1. Unregistered User**

1.1 Unregistered Users will be able to search posts.

1.2 Unregistered Users shall be able to filter search results based on categories. This function will help to narrow down posts that fit what the user is looking for.

1.3 Unregistered Users shall be able to register for an account. The registration email can only be a SFSU email.

1.4 Unregistered Users shall not contact the seller. They will not be able message the seller unless they have a registered account.

## **2. Registered User**

2.1 Registered Users can perform all functions that Unregistered User can along with some additional functionalities.

2.2 Registered Users can create posts. This function is required to help Users sell their products.

2.3 Registered Users can contact the seller of the post via in-site messaging after buy. This function is required for transactions between users.

2.4 Registered Users shall be able to edit the post after it has been submitted.

2.5 If the product is not available anymore then the User can delete the post.

2.6 The product can be reported to the Administrator if the product isn't accurate to the description provided / images displayed. In this case the seller shall be reported to the Administrator.

2.7 Registered Users can log in to their accounts to buy / sell items.

2.8 Registered Users can log out of their accounts. This is to enhance security in case they are accessing the account from a public place.

2.9 Registered Users can request to reset their password in case they forget their password.

2.10 Registered Users can propose the meeting location and time to complete the transaction.

2.11 Registered Users can look at their transaction history in case they want to revisit a post.

## **3. Administrator**

3.1 Administrators can perform all functions that a Registered User can.

3.2 Administrators shall be able to delete or edit posts made by the User, if they do not follow the terms of service or they are offending in any capacity.

3.3 Administrators will be able to approve a post before it goes live, so that none of the posts go against the terms of service.

3.4 Administrators will be able to delete / ban a User if they consistently fail to comply with the terms of service.

## **List of Non-Functional Requirements**

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0.
2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers.
3. All or selected application functions must render well on mobile devices.
4. Data shall be stored in the database on the team's deployment server.
5. No more than 50 concurrent users shall be accessing the application at any time.
6. Privacy of users shall be protected.
7. The language used shall be English (no localization needed).
8. Application shall be very easy to use and intuitive.
9. Application should follow established architecture patterns.
10. Application code and its repository shall be easy to inspect and maintain.
11. Google analytics shall be used.
12. No email 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.
13. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.
14. Site security: basic best practices shall be applied (as covered in the class) for main data items.
15. Media formats shall be standard as used in the market today.
16. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development.
17. The application UI (WWW and mobile) shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only" at the top of the WWW page nav bar. (Important so as to not confuse this with a real application).

## Competitive Analysis

Feature	SFSU Library	Public Library	Amazon	BusyGator
Store Pickup	7 working days	15 working days	3 - 5 working days	Same day
Safe Locations	YES	NO	YES	YES
Marketing Strategies	Email / Website	Email	Website / Social Media	Website / Social Media / Email
Student Satisfaction	★	★★	★★★	★★★★
Map of Locations	NO	NO	YES	YES
Diverse Products	NO	NO	YES	YES

Our company BusyGator provides same day pickup services as compared to other companies, where users need to book in advance for the equipment. We also provide safe locations for our company to make it more convenient for them and ensure their safety. The application will provide a map of the safe locations for customers to know the pickup location of the items ahead of time. We have many ways to show our products, such as our website, social media, posters, email, etc. In addition, we predict a 4 star student satisfaction as the services we provide will be very convenient for students. In comparison to other companies, we have a diverse range of products. We have a variety of products available including, but not limited to, laptops, textbooks, calculators, lab coats, badminton rackets, basketballs, etc.

## High-level System Architecture and Technologies Used

- **Server Host:** Amazon AWS 1vCPU 2 GB RAM
- **Operating System:** Ubuntu 20.04.3 LTS Server
- **Database:** MySQL v 8.0.28
- **Web Server:** NGINX 1.21.6
- **Server-Side Language:** Javascript
- **Additional Technologies:**
- **Web Framework:** Express
- **IDE:** VS Code
- **Web Analytics:** Google Analytics
- **SSL Cert:** Lets Encrypt (Cert Bot)
- **SASS:** 1.49.7

## Team and Roles



## **Checklist**

- So far all team members are engaged and attending ZOOM sessions when required.

### **ISSUE**

The current issue is team member(s) not attending meetings or being engaged. This is being handled with the Professor through the Team Lead.

- Team found a time slot to meet outside of the class.

### **DONE / OK**

- Back end, Front end leads and Github master chosen.

### **DONE / OK**

- Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing.

### **ON TRACK**

- Team lead ensured that all team members read the final M1 and agree/understand it before submission.

### **DONE / OK**

- Github is organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.).

### **DONE / OK**

# SW Engineering CSC648 / 848 Spring 2022

## BusyGator

Team 04		
Role	Name	Email
Team Lead	Samantha Saxton-Getty	ssaxtongetty@mail.sfsu.edu
Github Lead	Vishal Ramanand Sharma	vsharma5@mail.sfsu.edu
Front End Lead	Elyssa Mari Tapawan	etapawan1@mail.sfsu.edu
Back End Lead	Aaron Carlson	acarlson8@mail.sfsu.edu
Front End	Abdullah Sharaf	fabdullah1@mail.sfsu.edu
Front End	Siqi Guo	sguo4@mail.sfsu.edu
Back End	Janvi Patel	jpatel6@mail.sfsu.edu

March 19, 2022

## Milestone 02

Date	Version
03.19.22	Finished Version 01 submitted
04.08.22	Revision completed per instructor feedback Version 02

## **Executive Summary**

There are many students who cannot find the resources they need to properly study, work out, or research. Our application, BusyGator, provides various materials to help SFSU students, staff, and faculty find everything they are looking for on campus all online. BusyGator is an e-commerce application that allows students, staff, and faculty to buy different items, from school materials to gym equipment, as well as list items for others to buy. This will motivate the SFSU community to use these products offered by the application to better their education and health.

The BusyGator will have numerous functionalities and services. The main service the application will provide is allowing SFSU students, staff and faculty to post and /or buy textbooks, sports gear, research devices, and much more all online. Each product that can be bought will show how much is in stock during the time of browsing, the price of the material, and the user selling the item. The BusyGator will provide a map pin-pointing various safe areas to pick up the product all around the SFSU campus. Other important functionalities the application will include are a search option and categories to look for more specific items, and a sign up / sign in option to confirm users are SFSU students or faculty. These services will provide users with an easy and accessible way to find various resources all in one place. BusyGator gives SFSU an opportunity to help their community by managing their resources within the application. Instead of going to different locations on campus to see if they can buy materials in person, the BusyGator will make this task a lot easier. It saves students, staff, and faculties the time and hassle in their already busy schedules. Also, this allows alumnis to give textbooks they no longer need to newer students who will utilize them more for classes. The unique aspect of BusyGator is that all of the features and services in buying and selling material will be in one place and in one press of a button, where anyone can easily follow.

We are Team 4 from the Software Engineering class of Spring 2022 consisting of Aaron Carlson, Siqi Guo, Janvi Patel, Samantha Saxton-Getty, Abdullah Sharaf, Vishal Ramanand Sharma, and Elyssa Mari Tapawan, with Samantha being the team lead. Other roles consist of Aaron being the back-end lead, Elyssa being the front-end lead, and Vishal being the GitHub lead. The back-end operates the parts of the application that aren't accessed by a user such as data organization. The front-end works on the parts of the websites that allow users to interact with the web features, such as the application design. The GitHub lead ensures that there are no errors and that the application is functional. We are a group of aspiring students learning the aspects of team software development by making an application ourselves. Our goal is to work together to create the BusyGator to help us experience and prepare ourselves for our future careers.

## Main Data Items and Entities - Glossary / Definitions

### 1. user:

**Definition:** stores the login and personal information of user accounts

**Sub-Items:**

- 2.1 user\_id: primary key used for database identification
- 2.2 first\_name: first name of the user
- 2.3 last\_name: last name of the user
- 2.4 email: email address of the user used for login and confirmation of official SFSU status
- 2.5 password: protected string created by the user for login
- 2.6 administrator: flag reporting whether the user is an administrator
- 2.7 date\_created: date that the user account was created

### 2. product:

**Definition:** stores descriptive information regarding the application's products

**Sub-Items:**

- 5.1 product\_id: primary key used for database identification
- 5.2 seller\_id: foreign key identifier of the user selling the product
- 5.3 category: foreign key identifier of the category the product falls under
- 5.4 location: foreign key identifier of the location the product is to be exchanged at
- 5.5 title: name of the product
- 5.6 description: description given to the product
- 5.7 image: image showcasing the product
- 5.8 image\_thumbnail: image showcasing thumbnail version of product
- 5.9 price: price value attached to the product
- 5.10 approved: flag reporting approval status
- 5.11 date\_created: date that the product post was created

### 3. location:

**Definition:** stores information regarding the locations used for exchanges

**Sub-Items:**

- 6.1 location\_id: primary key used for database identification
- 6.2 name: name of the exchange location

### 4. category:

**Definition:** stores information regarding the product categories

**Sub-Items:**

- 7.1 category\_id: primary key used for database identification
- 7.2 name: name of the product category

**7.3** description: description given to the product category

**5. message:**

**Definition:** stores information regarding the sending of in-site messages

**Sub-Items:**

**9.1** message\_id: primary key used for database identification

**9.2** creator\_id: foreign key identifier of the message's sender

**9.3** receiver\_id: foreign key identifier of the message's receiver

**9.4** subject: subject of the message being sent

**9.5** message\_body: body of the message being sent

**9.6** date\_created: date that the message was created

## **List of Functional Requirements - Prioritized**

### **Priority 1:**

#### ***Unregistered User:***

1.1 Unregistered Users will be able to search posts.

1.2 Unregistered Users shall be able to filter search results based on categories. This function will help to narrow down posts that fit what the user is looking for.

1.3 Unregistered Users shall be able to register for an account. The registration email can only be a SFSU email.

1.4 Unregistered Users shall not contact the seller. They will not be able message the seller unless they have a registered account.

#### ***Registered User:***

2.1 Registered Users can perform all functions that Unregistered User can along with some additional functionalities.

2.2 Registered Users can log in to their accounts to buy / sell items.

2.3 Registered Users can request to reset their password in case they forget their password.

2.4 Registered Users can create posts. This function is required to help Users sell their products.

2.5 Registered Users shall be able to edit the post after it has been submitted.

2.6 If the product is not available anymore then the User can delete the post.

2.7 Registered Users can log out of their accounts. This is to enhance security in case they are accessing the account from a public place.

2.8 Registered Users can contact the seller of the post via in-site messaging after buy. This function is required for transactions between users.

***Administrator:***

3.1 Administrators will be able to delete / ban a User if they consistently fail to comply with the terms of service.

3.2 Administrators shall be able to delete or edit posts made by the User, if they do not follow the terms of service or they are offending in any capacity.

3.3 Administrators will be able to approve a post before it goes live, so that none of the posts go against the terms of service.

**Priority 2:**

***Registered User:***

2.1 The product can be reported to the Administrator if the product isn't accurate to the description provided / images displayed. In this case the seller shall be reported to the Administrator.

***Administrator:***

3.1 Administrators can perform all functions that a Registered User can.

**Priority 3:**

***Registered User:***

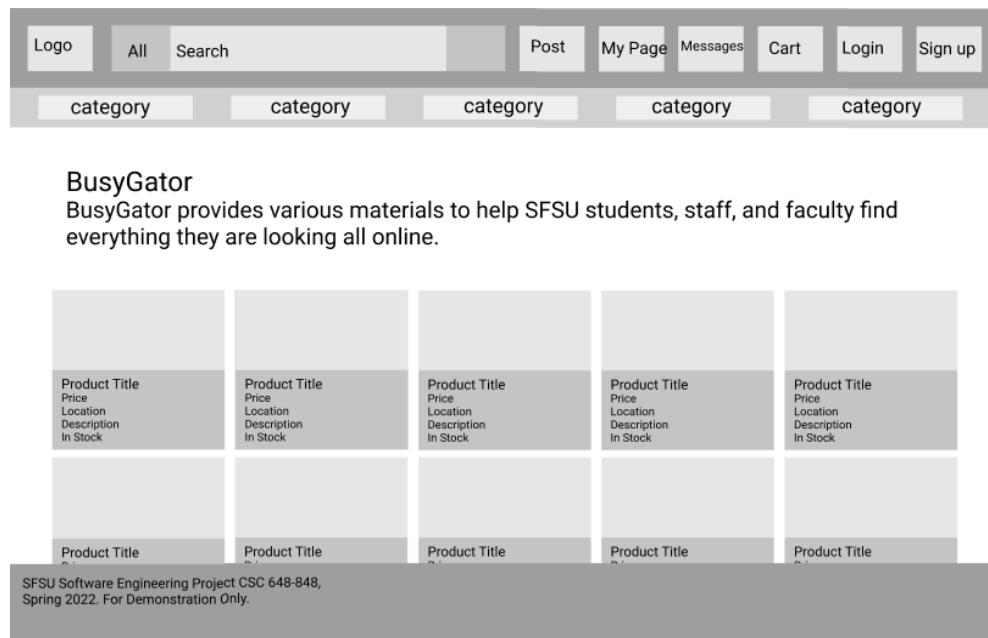
2.1 Registered Users can propose the meeting location and time to complete the transaction.

2.2 Registered Users can look at their transaction history in case they want to revisit a post.

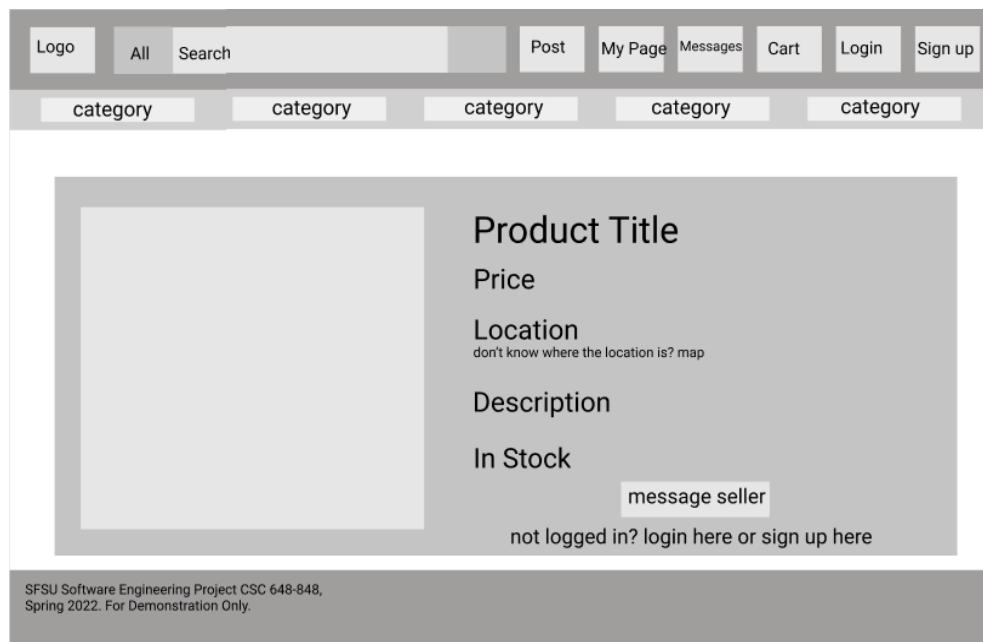
## UI Storyboards for Each Main Use Case

### Use Case 1: SFSU Student/Faculty (Buyer)

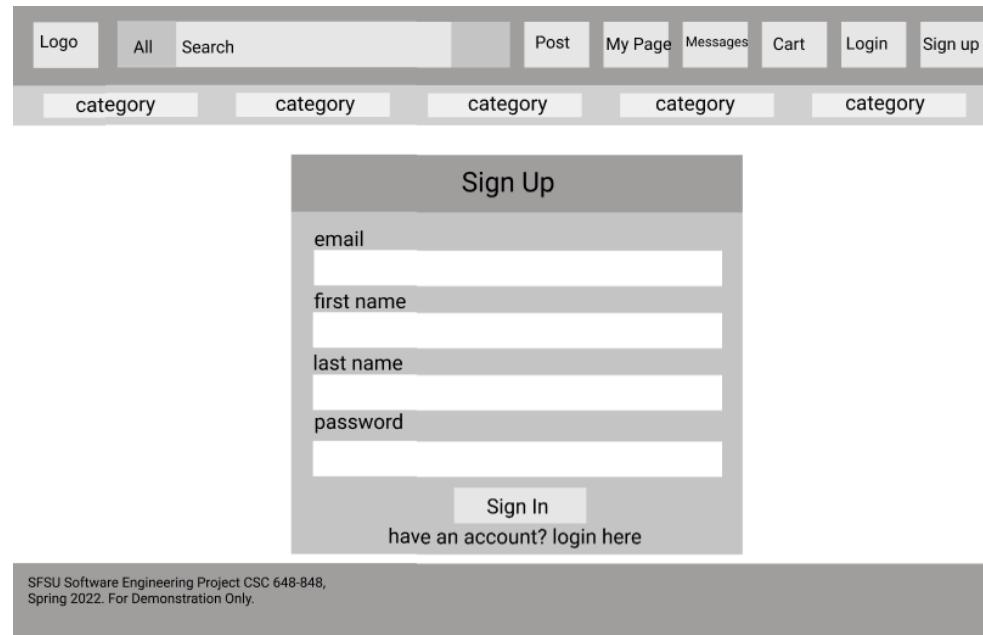
John is a student and a part-time worker at SFSU. He is looking for a textbook for his class on BusyGator. He enters what he is looking for in the search bar. The results show up on the “BusyGator” page.



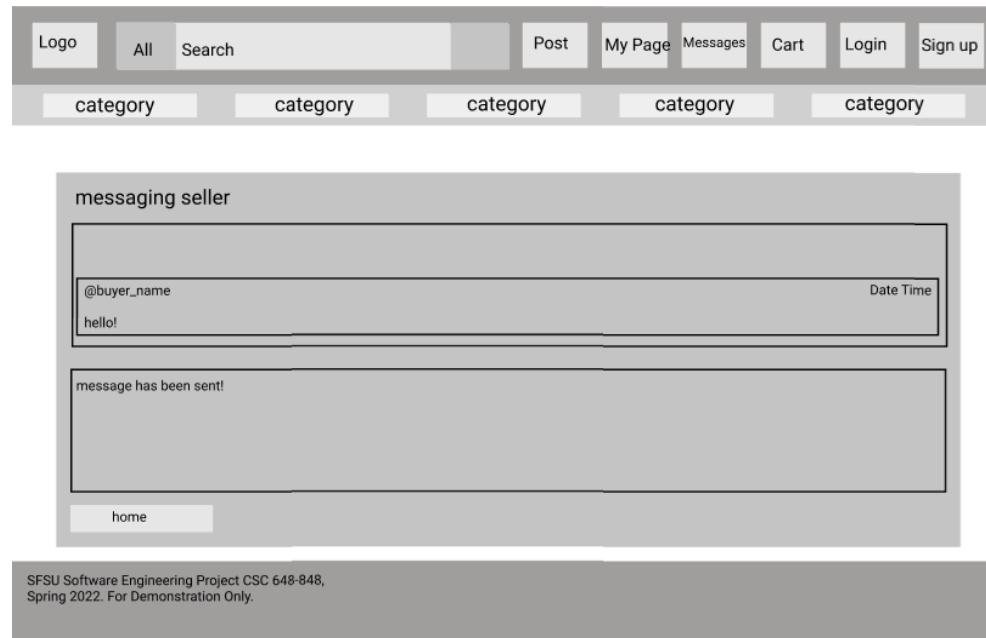
John looks on the page of all the results. He clicks on the product and sees the details of the textbook. He tries to message the seller, but it prompted to log in or sign up for an account before he can message the seller about the product.



John doesn't have an account yet, so he clicks on the signup button and creates an account on the website by inputting his SFSU email, first name, last name, and password.

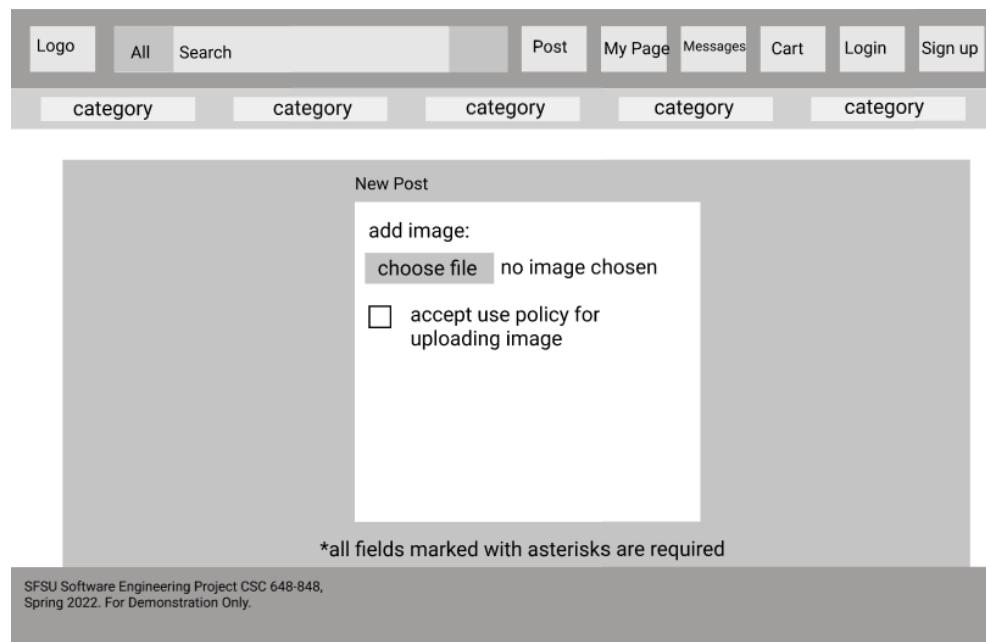


After the account has been made, John logs in and can now message the seller of the product to obtain the product. We will implement the cart feedback in the future.



### Use Case 2: SFSU Student (Seller)

Jennifer is a full-time SFSU student looking to list a textbook to sell. To list an item for sale, Jennifer needs to have an account on the BusyGator. Jennifer repeats the same process as use case 1 to log in, where she can now list a product. To list items, Jennifer needs to fill out the product's description when she is making a new post. After she finishes, she clicks the post button to have it listed.



Logo All Search Post My Page Messages Cart Login Sign up

category category category category category

\* all fields marked with asterisks are required

Product Title\*

Price\*

Location\*

Description\*

Cancel Post

may take up to 24 hours to be approved

SFSU Software Engineering Project CSC 648-848,  
Spring 2022. For Demonstration Only.

Jennifer wants to update the price for an item she has already listed. She presses the my page button on the navigation bar to see her listings and messages and clicks the one she wants to edit. To edit the listing, she presses the edit button to update the prices for the listed item. She updates the price and saves the listing.

Logo All Search Post My Page Messages Cart Login Sign up

category category category category category

**Listings:**

Product Title  
Price  
Location  
Description  
In Stock

edit

**Messages:**

SFSU Software Engineering Project CSC 648-848,  
Spring 2022. For Demonstration Only.

Jennifer finds out that she lost the textbook and wants to close the listed item. She clicks the listings button again on the top. She presses the edit button and presses the delete button where it will prompt “Are you sure you want to delete this listing?” She presses yes and deletes the listing.

The screenshot shows a user interface for editing a product listing. At the top, there is a navigation bar with links for Logo, All, Search, Post, My Page, Messages, Cart, Login, and Sign up. Below the navigation bar, there are five categories labeled category. The main content area contains fields for Product Title, Price, Location, and Description, each with a corresponding input field. At the bottom of the form are three buttons: Cancel, Editing Done, and Delete. A small note at the bottom left of the page states: "SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only."

## High Level Architecture and Database Summary

### DB Organization:

BusyGators database organization is centered around two account types, user and administrator. Both account type tables hold information regarding the login while the user account also has more in-depth details regarding the users personal information.

The user interacts with products through transactions and posting, while the administrator interacts with products through approving user posts. Products are items with detailed information, including references to tables of location and category. In addition, the user interacts with other users through messages with the message recipient.

The naming convention utilized in this database is keyword based. If there is more than one keyword in the name, it is separated by an underscore. This keyword describes the data variable. For example, *keyword1\_keyword2\_keyword3*, *keyword1\_keyword2* or *keyword*.

## **DB Tables:**

### **1. user**

- 1.1.** user\_id (PK)
- 1.2.** first\_name
- 1.3.** last\_name
- 1.4.** email
- 1.5.** password
- 1.6.** administrator
- 1.7.** date\_created

### **2. product**

- 2.1.** product\_id (PK)
- 2.2.** seller\_id (FK)
- 2.3.** category (FK)
- 2.4.** location (FK)
- 2.5.** title
- 2.6.** description
- 2.7.** image
- 2.8.** image\_thumbnail
- 2.9.** price
- 2.10.** approved
- 2.11.** date\_created

### **3. location**

- 3.1.** location\_id (PK)
- 3.2.** name

### **4. category**

- 4.1.** category\_id (PK)
- 4.2.** name
- 4.3.** description

### **5. message**

- 5.1.** message\_id (PK)
- 5.2.** creator\_id (FK)
- 5.3.** receiver\_id (FK)
- 5.4.** subject
- 5.5.** message\_body
- 5.6.** date\_created

### **Media Storage:**

BusyGators media will be stored in a file system. The relative paths for each file will be stored in the image attribute within the product table. There will be a limit of maximum file size. User passwords will be encrypted and the file system will be protected using bcrypt to scramble file names.

### **Search / Filter Architecture and Implementation:**

BusyGator will have a search bar that allows the user to search by item title and description. Users are able to then further filter by price and location. These filters will be created using the location and price attributes of the product table. If the user selects a category from the drop down or the sub navbar, that category is included with the search item. We will utilize SQL queries with %like% to match category and search terms together, or match the search term through all items if there is no category selected.

### **Key Risks**

#### **Skills Risk:**

We will be working on different platforms and using different programs. Some of these programs are new to several of our group members. Our team members are guiding and helping each other knowing how it works.

#### ***How to Resolve:***

For the members doing it for the first time, it will be difficult but our team shares resources, mini tutorials and guiding through Discord and Zoom meetings. We also do more research while using it.

## **Project Management**

As soon as milestone 02 was released and reviewed in class, we immediately reviewed the document as a team to ensure everyone understood the milestone in its entirety. We focused heavily on revising milestone 01 for a smooth transition into milestone 02 since it would be a backbone for a lot of the content in the second milestone. Following this, I made a copy of our milestone 01 Google document and tailored it to the current milestone and its tasks. I added comments to each portion of the document assigning which member would complete each task, some including a brief note on expected work. During our meeting, we reviewed each assigned task and cleared up any questions before each member began their work. I utilized Discord to keep a list of “to-do” items that included meetings, deadlines for deliverables, and notes so that each team member could reference what would be going on during the given week. However, in the future I plan to use Trello as suggested by the professor for a more clean and concise area to divide work and display “to-do” items. Another tool we relied heavily on during meeting times was When2Meet. As a team we utilized this tool to quickly schedule follow up meetings that all team members were able to make in order to check-in on work progress, give feedback, and answer questions that weren’t covered over text on Discord. This milestone proved challenging in coordinating the front-end and back-end teams on their deliverables and concepts. Going forward, we will be utilizing Trello for project management and scheduling more frequent meeting times to ensure both teams are on the same page for fluid development.

# SW Engineering CSC648 / 848 Spring 2022

## BusyGator

Team 04		
Role	Name	Email
Team Lead	Samantha Saxton-Getty	ssaxtongetty@mail.sfsu.edu
Github Lead	Vishal Ramanand Sharma	vsharma5@mail.sfsu.edu
Front End Lead	Elyssa Mari Tapawan	etapawan1@mail.sfsu.edu
Back End Lead	Aaron Carlson	acarlson8@mail.sfsu.edu
Front End	Abdullah Sharaf	fabdullah1@mail.sfsu.edu
Front End	Siqi Guo	sguo4@mail.sfsu.edu
Back End	Janvi Patel	jpatel6@mail.sfsu.edu

April 13, 2022

## Milestone 03

Date	Version
04.13.22	Finished Version 01 submitted

# **Summary of Milestone 3 ZOOM meeting review with Prof. Petkovic and plans for further development**

**Team Number:** 04

**Meeting Date:** 04/13/2022

## **Summary of Feedback on UI:**

- Remove sub-navbar.
- Remove messages button as messages should live within “My Page”.
- Merge “Log In” and “Sign Up” buttons.
- Make the pull down category menu gray or more visible.
- Change the price filter to buttons that allow high-to-low or low-to-high and remove the “Go” and “Clear” buttons.
- Cancel buttons need to be gray, while post / confirm / go should be colored.
- Align the “Check It Out” buttons on the product cards so that they are the same height for each card.
- Remove Cart for now, this is P2.
- Change the “Home” button on messages to “Cancel” (make gray).
- When clicking on “My Page”, force the user to log in or register.
- Add “Confirm Password” to the Sign Up page.
- Update page titles to be a bigger font size and bold, easier to see.
- Move the asterisk (\*) for mandatory fields to the top right of the UI.
- For new posts, increase the title size and make it bold.
- For new posts, add its category, price, location drop-down, description, and file upload.
- On “My Page”, have two tabs, one for “My Messages” and another for “My Posts”.
- Posts should have the date, the user involved, a message button, and the link to the individual product.
- The Sign Up page should have a terms and conditions checkbox. This can be a dummy link.
- Remove the previous message on the message field.
- Update the search bar so that it should work across all pages.
- Add an item count for the results upon searching for example (1/1).
- Pre-fill all titles of user input to be descriptive.
- Add a drop-down to pick the preset location for pick-up (this should not be a required field).

**Summary of feedback on Code and Architecture:**

- Update header comments to proper information (Purpose, I/O, and the owner name, not SFSU).

**Summary of feedback on GitHub Usage:**

- Edit all GitHub comments to be more descriptive.

**Summary of feedback on DB:**

- Add message content in SQL.
- Add a product\_id FK to the message table.
- Encrypt passwords.

**Summary of feedback on Teamwork:**

Not Applicable

**Any other comments and issues:**

Not Applicable

**Check Point (CP) if given, DUE:**

Not Applicable

**List below agreed upon P1 list of features for final delivery which constitute product plan. NOTE: after this meeting the team focuses solely on this P1 list of features, e.g. the development is in “feature freeze mode”. All listed P1 features (no more no less) MUST be delivered in a usable way, free of bugs.**

**Priority 1:**

***Unregistered User:***

- 1.1 Unregistered Users will be able to search posts.
- 1.2 Unregistered Users shall be able to filter search results based on categories. This

function will help to narrow down posts that fit what the user is looking for.

1.3 Unregistered Users shall be able to register for an account. The registration email can only be a SFSU email.

1.4 Unregistered Users shall not contact the seller. They will not be able message the seller unless they have a registered account.

***Registered User:***

2.1 Registered Users can perform all functions that Unregistered User can along with some additional functionalities.

2.2 Registered Users can log in to their accounts to buy / sell items.

2.3 Registered Users can request to reset their password in case they forget their password.

2.4 Registered Users can create posts. This function is required to help Users sell their products.

2.5 Registered Users shall be able to edit the post after it has been submitted.

2.6 If the product is not available anymore then the User can delete the post.

2.7 Registered Users can log out of their accounts. This is to enhance security in case they are accessing the account from a public place.

2.8 Registered Users can contact the seller of the post via in-site messaging after buy. This function is required for transactions between users.

***Administrator:***

3.1 Administrators will be able to delete / ban a User if they consistently fail to comply with the terms of service.

3.2 Administrators shall be able to delete or edit posts made by the User, if they do not follow the terms of service or they are offending in any capacity.

3.3 Administrators will be able to approve a post before it goes live, so that none of the posts go against the terms of service.

**Plan of action: Summary of plans for final several weeks e.g. what specific things will be done to address the above feedback in order to deliver on P1 features (note: seldom you can do everything that comes up in feedback so need to prioritize). This is IMPORTANT: analyze what needs to be done, prioritize based on two factors: a) importance for the product/user and b) cost/ability to deliver it in a given schedule. Based on this come up with the plan then execute it. Focus is on P1 features only.**

Our plan of action going forward is to focus strictly on priority one features that are not fully implemented as of yet, or not at all. These tasks will be laid out in Trello with due dates and individual teammate assignments.

In further detail, we will be focusing on the following condensed priority one list:

Unregistered, Registered, and Administrator Users:

1. View Listings and Marketplace
2. Search Listings and Marketplace
3. Filter Listings and Marketplace

Unregistered Users:

4. Register Account

Registered Users:

5. Login Account
6. Post Listing on Marketplace
7. Message Seller to receive Approval or Denial of interest

Administrator Users:

8. Remove User
9. Remove Listing
10. Approve or Deny Listings

These are the prioritized features in priority one that we will be executing within our plan of action first. As stated, these tasks will be provided in Trello and we will have three meetings a week to ensure everything is staying on time and the priority one list is being completed.

# SW Engineering CSC648 / 848 Spring 2022

## BusyGator

Team 04		
Role	Name	Email
Team Lead	Samantha Saxton-Getty	ssaxtongetty@mail.sfsu.edu
Github Lead	Vishal Ramanand Sharma	vsharma5@mail.sfsu.edu
Front End Lead	Elyssa Mari Tapawan	etapawan1@mail.sfsu.edu
Back End Lead	Aaron Carlson	acarlson8@mail.sfsu.edu
Front End	Abdullah Sharaf	fabdullah1@mail.sfsu.edu
Front End	Siqi Guo	sguo4@mail.sfsu.edu
Back End	Janvi Patel	jpatel6@mail.sfsu.edu

May 16, 2022

### Milestone 04

Date	Version
05.16.22	Finished Version 01 submitted

## **Product Summary**

### **Application Name:** BusyGator

Our application, BusyGator, provides various materials to help SFSU students, staff, and faculty find everything they are looking for on campus all online. BusyGator is an e-commerce application that allows students, staff, and faculty to buy different items, from school materials to gym equipment, as well as list items for others to buy. The BusyGator has numerous functionalities and services. The main service the application will provide is allowing SFSU students, staff and faculty to post and / or buy textbooks, sports gear, research devices, and much more all online. The application has search options and categories to look for more specific items, and a sign up / sign in option to confirm users are SFSU students or faculty. BusyGator gives SFSU an opportunity to help their community by managing their resources within the application. The unique aspect of BusyGator is that all of the features and services in buying and selling material will be in one place and in one press of a button, where anyone can easily follow.

### **Priority 1 Feature List:**

#### Unregistered, Registered, and Administrator Users:

1. View Listings and Marketplace
2. Search Listings and Marketplace
3. Filter Listings and Marketplace

#### Unregistered Users:

4. Register Account

#### Registered Users:

5. Login Account
6. Post Listing on Marketplace
7. Message Seller to receive Approval or Denial of interest

#### Administrator Users:

8. Remove User
9. Remove Listing
10. Approve or Deny Listings

**URL:** <http://3.23.79.193/>

# Usability Test Plan

## Test Objectives:

Our usability test plan's objective is to test the users satisfaction when posting data to our website, BusyGator. Since BusyGator is an e-commerce marketplace that allows users to post many items, we can expect this test to help determine which areas of posting need to be optimized. This test will ensure we can implement areas of improvement currently and in the future.

## Test Background and Setup:

### System Setup:

Any browser on a mobile device, laptop, or PC with access to the internet.

### Starting Point:

Any user that is already registered within BusyGator, but has yet to login.

### Intended Users:

Any San Francisco State University faculty, staff, and students who have basic knowledge and understanding of how to operate a browser on a mobile device, laptop, or PC.

URL: <http://3.23.79.193/>

## Usability Task Description:

Please open a browser and navigate to BusyGator. Once there, log into your BusyGator account. After logging in, navigate to the post page and fill out the prompted information on the screen. Once this is complete, please submit your post.

## Evaluation of Effectiveness:

In order to evaluate the effectiveness of posting an item, we need to measure the success rate of users that complete the following tasks successfully (see below). In addition to completing and recording these tasks, we need to record any errors that happened during the process.

Test (Use) Case	Percent Completed	Errors Encountered	Comments
<b>Navigate to BusyGator Website</b>	100	N/A	N/A
<b>Login</b>	50	Unable to Login.	This feature is not fully complete.
<b>Navigate to the</b>	100	N/A	N/A

<b>Post Page</b>			
<b>Fill out the Form</b>	100	N/A	N/A
<b>Upload a Picture</b>	100	N/A	N/A
<b>Submit the Form</b>	100	N/A	N/A

#### **Evaluation of Efficiency:**

In order to evaluate the efficiency of posting an item, we need to measure the time it takes for the user to complete each task, the number of errors encountered, as well as the number of clicks the user performs. These will be measured through the following tasks (see below).

<b>Test (Use) Case</b>	<b>Completion Time (Seconds)</b>	<b>Number of Errors Encountered</b>	<b>Number of Clicks Performed</b>
<b>Navigate to BusyGator Website</b>	3	N/A	3
<b>Login</b>	5	Unable to Login.	4
<b>Navigate to the Post Page</b>	1	N/A	1
<b>Fill out the Form</b>	15	N/A	8
<b>Upload a Picture</b>	4	N/A	3
<b>Submit the Form</b>	1	N/A	1

#### **Evaluation of User Satisfaction:**

In order to evaluate the user satisfaction of posting an item, we need to direct the focus group to this survey below.

<b>Task</b>	<b>Strongly Agree</b>	<b>Agree</b>	<b>Neutral</b>	<b>Disagree</b>	<b>Strongly Disagree</b>
<b>Navigating to the Post Page was easy</b>	✓				
<b>The mandatory sections were marked clearly</b>	✓				

I was able to complete the task in a reasonable amount of time	✓					
--	---	--	--	--	--	--

## QA Test Plan

### Test Objectives:

The objective of the test is to verify the functionality of making posts works according to the specifications.

- When the user wants to make a new post, the user can fill out the required information needed to make a new post and press the post button to post their product. When submitting a post, the post will show up on the home page along with the information the seller filled out.
- The information the user filled out in the new post should be recorded in the database.
- All required information within the new post should be filled, if not then it should produce an error.

The final outcome of the test would be a new post of the product with the correct details and data of the new post being recorded into the database.

### HW and SW Setup (including URL):

HW Setup: A mobile device, laptop, and/or computer that is connected to the internet.

SW Setup: Google Chrome or Firefox browser.

URL: <http://3.23.79.193/>

### Feature to be Tested:

- Making a new post.
  - The user must input all the required data within the new post, if left empty then errors will display under the mandatory fields.
  - When all the fields are filled correctly, the user can press the post button and it will be posted onto the home page where it displays the product and its details.
- The post data will be stored in the database.
  - All the data the user put in should be stored correctly into the database.
  - The product title, price, location, description, seller, and date listed should be recorded in the database.

### QA Test plan:

Test #	Test Title	Test Description	Test Input	Expected Correct Output	Test Results (Firefox)	Test Results (Google Chrome)
1	Inputting no fields in the New Post form	When the user inputs no fields and attempts to post by clicking the post button.	1. Keep the fields empty. 2. Press the Post button.	Error messages should appear under the mandatory fields labeled by asterisks reminding the user to fill out the required fields.	Pass	Pass
2	Filling all inputs to the New Post form	When the user completes all the fields on the New Post form and post by clicking the post button.	1. Fill in all the fields on the New Post form correctly. 2. Press the Post button.	The user should be redirected to the home page while the fields that were filled are waiting to be approved by the admin to be posted.	Pass	Pass
3	Canceling the New Post fields	When the user fills out one or more fields on the New Post form, they will be able to cancel the post.	1. Fill in one more field on the New Post form correctly. 2. Press the cancel button.	The forms that were filled out should be cleared.	Fail	Fail
4	Data recorded in the database	When the user fills out the forms on the New Post correctly and presses the Post button, the data is stored within the database.	1. Fill in all the fields on the New Post form correctly. 2. Press the post button. 3. Check the database to see all the post data.	All the post data is stored correctly into the database.	Pass	Pass

## Code Review

Conversation 3    Commits 2    Checks 0    Files changed 24

 ajccarlson commented yesterday

Changes:

- Implemented React-Bootstrap input validation for Signup.js
- Replaced userInfo DataContext with useState()
- Cleaned up code

-o  Added the display of validation messages to Signup.js 490a121

  ajccarlson requested a review from vishals9711 yesterday

  vishals9711 requested changes 7 hours ago View changes

vishals9711 left a comment

Changes Requested

application/frontend/src/Pages/Signup.js Outdated Hide resolved

```
35 +   isValid: true,
36 +   errorMessage: ""
37 + });
38 + const [cPasswordFormObj, setCPasswordFormObj] = useState({
```

 vishals9711 7 hours ago ...

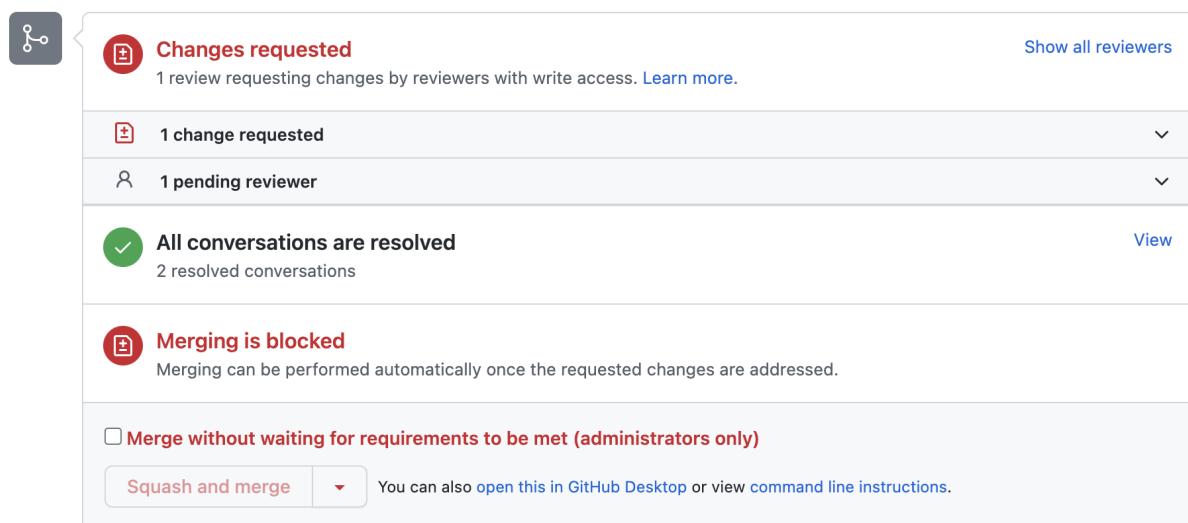
rename variable to confirmPasswordFormObj

 Reply...

application/frontend/src/Pages/Signup.js Outdated

Hide resolved

```
169 +     else if (key === 'cPassword') {  
170 +         if (!value) {  
171 +             setCPasswordFormObj({...cPasswordFormObj,  
172 +                 isValid: false,  
173 +                 errorMessage: 'Please confirm password'})  
174 +         }  
175 +     else if (value != passwordFormObj.value) {  
176 +         setCPasswordFormObj({...cPasswordFormObj,  
177 +             isValid: false,  
178 +             errorMessage: 'Passwords must match'})  
179 +     }  
180 +     else{  
181 +         setCPasswordFormObj({...cPasswordFormObj,  
182 +             value: value,  
183 +             isValid: true,  
184 +             errorMessage: ''})  
185 +     }  
vishals9711 7 hours ago ...  
Change occurrence of cPassword to confirmPassword
```



vishals9711 approved these changes now

[View changes](#)

vishals9711 left a comment

Changes look good to me

Add more commits by pushing to the `M4_Signup_Display_Validation` branch on **CSC-648-SFSU/csc648-03-sp22-team04**.

**Changes approved**  
1 approving review by reviewers with write access. [Learn more](#).

**1 approval**

**All conversations are resolved**  
[View](#)  
2 resolved conversations

**This branch has no conflicts with the base branch**  
Merging can be performed automatically.

**Squash and merge** ▾ You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

ajccarlson requested a review from vishals9711 7 hours ago

vishals9711 approved these changes 1 minute ago

[View changes](#)

vishals9711 left a comment

Changes look good to me

vishals9711 merged commit `46a8ea9` into `develop` now

[Revert](#)

**Pull request successfully merged and closed**  
You're all set—the `M4_Signup_Displa...` branch can be safely deleted.

[Delete branch](#)

## **Self-Check on Best Practices for Security**

<b>Asset to be Protected</b>	<b>Types of possible / expected Attacks</b>	<b>Your Strategy to mitigate / protect the Asset</b>
Database	SQL Injection	All inputs have validation, queries are structured carefully prepared statements, and the database passwords are encrypted
Search	SQL Injection or XSS	The search bar input is limited to 40 alphanumeric characters
Password	Vulnerable account information	Database password encryption
Account	Robot, Scam, or Spam Users	User accounts are required to have an SFSU email to successfully create an account
Post	Inappropriate or dangerous posts, along with threats, or potential scammers	Administrators must approve and validate each new post within a 24 hour window

## Self-Check of the Adherence to Original Non-Functional Specs

<b>Non-Functional Specification</b>	<b>Status</b>
Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0.	<b>ON TRACK</b>
Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers.	<b>DONE</b>
All or selected application functions must render well on mobile devices.	<b>DONE</b>
Data shall be stored in the database on the team's deployment server.	<b>DONE</b>
No more than 50 concurrent users shall be accessing the application at any time.	<b>DONE</b>
Privacy of users shall be protected.	<b>DONE</b>
The language used shall be English (no localization needed).	<b>DONE</b>
Application shall be very easy to use and intuitive.	<b>DONE</b>
Application should follow established architecture patterns.	<b>DONE</b>
Application code and its repository shall be easy to inspect and maintain.	<b>DONE</b>
Google analytics shall be used.	<b>DONE</b>
No email 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.	<b>DONE</b>
Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.	<b>DONE</b>

Site security: basic best practices shall be applied (as covered in the class) for main data items.	<b>DONE</b>
Media formats shall be standard as used in the market today.	<b>DONE</b>
Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development.	<b>DONE</b>
The application UI (WWW and mobile) shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only" at the top of the WWW page nav bar. (Important so as to not confuse this with a real application).	<b>DONE</b>

### 3. Product Screenshots

SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only.

**Busy Gator** Electronics Search Post My Page About Us Login / Signup Default ▾

Items: 5 / 19

Product Title	Price	Location	Description	Seller	Date Listed	Status	Check it out!	Message Seller
Laptop	\$200	Parking Garage	Laptop	Shauna Jones	Fri Apr 29 2022	In Stock	<a href="#">Check it out!</a>	<a href="#">Message Seller</a>
Calculator	\$7	University Park South	Calculator	Shauna Jones	Fri Apr 29 2022	In Stock	<a href="#">Check it out!</a>	<a href="#">Message Seller</a>
Headphones	\$40	Manzanita Square	Headphones	Steve Russel	Fri Apr 29 2022	In Stock	<a href="#">Check it out!</a>	<a href="#">Message Seller</a>
Kindle	\$130	Gymnasium	Kindle	Bryan Matthew	Fri Apr 29 2022	In Stock	<a href="#">Check it out!</a>	<a href="#">Message Seller</a>
Television	\$618	Mashouf Wellness Center	Television	Shauna Jones	Fri Apr 29 2022	In Stock	<a href="#">Check it out!</a>	<a href="#">Message Seller</a>

SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only.  
Software Engineering Class SFSU Spring 2022 Section 3 - Team 4

**Busy Gator** All Search Post My Page About Us Login / Signup

**Sign Up**

All fields with an asterisk (\*) are mandatory

Email \*

First Name \*

Last Name \*

Password \*

Confirm Password \*

I have read and agree to the Terms and Conditions and Privacy Policy

[Sign Up](#)

Already have an account? [Login Here](#)

SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only.  
Software Engineering Class SFSU Spring 2022 Section 3 - Team 4

SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only.



All ▾

Search

Search

Post

My Page

About Us

Login / Signup



**Product Title:** Laptop

Price: \$200

Location: Parking Garage

Don't know where the location is? [Map](#)

Description: Laptop

Seller: Shauna Jones

Date Listed: Fri Apr 29 2022

**In Stock**

[Message Seller](#)

SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only.

Software Engineering Class SFSU Spring 2022 Section 3 - Team 4

SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only.



All ▾

Search

Search

Post

My Page

About Us

Login / Signup

## Login

All fields with an asterisk (\*) are mandatory

Email \*

Email

Password \*

Password

[Forgot Password?](#)

[Login](#)

Don't have an account? [Sign Up Here](#)

SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only.

Software Engineering Class SFSU Spring 2022 Section 3 - Team 4

SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only.



All

Search

Search

Post

My Page

About Us

Login / Signup

## About the Team 4:

**Aaron Carlson**

Back-End Lead

**Abdullah Sharaf**

Front-End

**Elyssa Mari Tapawan**

Front-End Lead

**Janvi Patel**

Back-End

**Samantha Saxton-Getty**

Team Lead

**Siqi Guo**

Front-End

**Vishal Ramanand Sharma**

Github Lead

SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only.

Software Engineering Class SFSU Spring 2022 Section 3 - Team 4

SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only.



All

Search

Search

Post

My Page

About Us

Logout

**Welcome**

My Messages    My Posts

No Messages

SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only.

Software Engineering Class SFSU Spring 2022 Section 3 - Team 4

SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only.



All

Search

Search

Post

My Page

About Us

Logout

## Welcome

[My Messages](#) [My Posts](#)

No Listings

SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only.

Software Engineering Class SFSU Spring 2022 Section 3 - Team 4

SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only.



All

Search

Search

Post

My Page

About Us

Logout

## New Post

All fields with an asterisk (\*) are mandatory

Add Product Image \*

No file chosen

Accept User Policy for Uploading Image

Product Title \*

e.g: Foundations of Computer Science Textbook

Product Category \*

Product Price \*

e.g: 200

Product Pickup Location

Brief Product Description \*

e.g: Textbooks require Author and Edition

May take up to 24 hours to be approved

SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only.

Software Engineering Class SFSU Spring 2022 Section 3 - Team 4

SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only.



All ▾

Search

Search

Post

My Page

About Us

Login / Signup

## Forgot Password

All fields with an asterick (\*) are mandatory

Please enter your email address that is associated with your account.

Email \*

Email

Submit

SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only.

Software Engineering Class SFSU Spring 2022 Section 3 - Team 4

SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only.



All ▾

Search

Search

Post

My Page

About Us

Logout

## Messages

Click here to write a message

Cancel

Send Message

SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only.

Software Engineering Class SFSU Spring 2022 Section 3 - Team 4



All ▾ Search

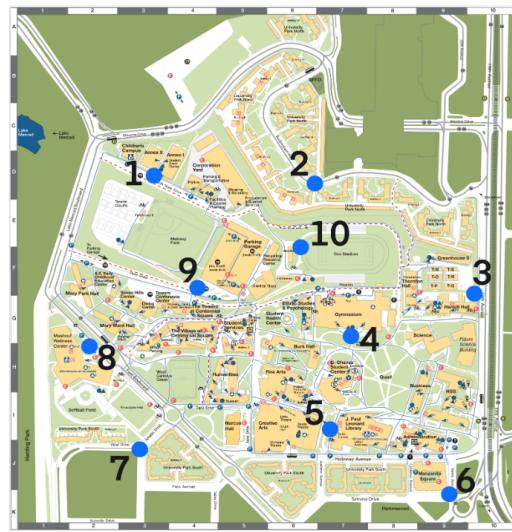
Search

Post

My Page

About Us

Login / Signup



Locations:

1. Annex I & Annex II
2. University Park North
3. Hensill Hall
4. Gymnasium
5. J. Paul Leonard Library
6. Manzanita Square
7. University Park South
8. Masshouf Wellness Center
9. Parking Garage
10. Cox Stadium



All ▾ Search

Search

Post

My Page

About Us

Logout

## Hello, I am Aaron Carlson



Hi, my name is Aaron Carlson. I was originally pursuing a career in Actuarial Science until I was introduced to computer programming at my local community college, Mission College in Santa Clara. And since then, I've transferred to San Francisco State University last spring where I'm now working toward my bachelor's degree in Computer Science. Outside of Computer Science, some things I enjoy are biking, listening to music, and gaming; with my current favorite game series being the Yakuza series and my favorite band being Thank You Scientist.

[Go Back](#)

***All other individual pages look identical to this one above and are not included.***

## 4. Database Organization

**Category:**

	category_id	name	description
▶	1	Electronics	euanfaofa
	2	Pets	awdwada
	3	Home	adwwad
	4	Recreational	igoker
	5	Books	uindwajkm
*	NULL	NULL	NULL

**Location:**

	location_id	name
▶	1	Annex I & Annex II
	2	University Park North
	3	Hensill Hall
	4	Gymnasium
	5	J. Paul Leonard Library
	6	Manzanita Square
	7	University Park South
	8	Masshouf Wellness Center
	9	Parking Garage
	10	Cox Stadium
*	NULL	NULL

**Message:**

	message_id	creator_id	receiver_id	product	subject	message_body	date_created
▶	1	12	2	6	Calculator	Hi! I was wondering when you'd be available to ...	2022-04-29 18:07:36
	2	15	4	11	Collar	hey, what size dog is the collar you listed meant...	2022-04-29 18:07:36
	3	6	3	13	Pet Food	when can you meet up for the pet food??? thx	2022-04-29 18:07:36
	4	11	5	21	Fishing Pole	I am interested in purchasing the fishing pole yo...	2022-04-29 18:07:36
	5	8	1	19	Tent	Hello! I'm thinking about buying your tent. How ...	2022-04-29 18:07:36
	6	7	2	23	Dictionary	would oyu be willing to sell the dictionary for \$1...	2022-04-29 18:07:36
	7	9	3	18	Sofa	Hi. Do you have any equipment for moving the ...	2022-04-29 18:07:36
	8	13	1	20	Sleeping Bag	hey! are you available to meet up tomorrow to ...	2022-04-29 18:07:36
	9	10	5	14	Leash	can you meet up tomorrow to sell the leash? an...	2022-04-29 18:07:36
	10	14	3	8	Kindle	Hello. Do you have any other pictures of your ki...	2022-04-29 18:07:36
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## Product:

	product_id	seller_id	category	location	title	description	image	image_thumbnail	price	approved	date_created
▶	1	2	1	9	Laptop	Laptop	1_laptop.jpeg	1_laptop_thumb.jpg	200	1	2022-04-29 18:07:28
	2	1	2	3	Bone	Bone	2_bone.jpeg	2_bone_thumb.jpg	10	1	2022-04-29 18:07:28
	3	3	3	2	Pillow	Pillow	3_pillow.jpeg	3_pillow_thumb.jpg	25	1	2022-04-29 18:07:28
	4	1	4	6	Bike	Bike	4_bike.jpeg	4_bike_thumb.jpg	450	1	2022-04-29 18:07:28
	5	4	5	10	Textbook	Textbook	5_textbook.jpeg	5_textbook_thumb.jpg	15	1	2022-04-29 18:07:28
	6	2	1	7	Calculator	Calculator	6_calculator.jpeg	6_calculator_thumb.jpg	7	1	2022-04-29 18:07:28
	7	1	1	6	Headphones	Headphones	7_headphones.jpeg	7_headphones_thumb.jpg	40	1	2022-04-29 18:07:28
	8	3	1	4	Kindle	Kindle	8_kindle.jpeg	8_kindle_thumb.jpg	130	1	2022-04-29 18:07:28
	9	2	1	8	Television	Television	9_television.jpeg	9_television_thumb.jpg	618	1	2022-04-29 18:07:28
	10	4	2	9	Pet Bowl	Pet Bowl	10_pet_bowl.jpeg	10_pet_bowl_thumb.jpg	6	1	2022-04-29 18:07:28
	11	4	2	7	Collar	Collar	11_collar.jpeg	11_collar_thumb.jpg	9	1	2022-04-29 18:07:28
	12	2	2	5	Pet Bed	Pet Bed	12_pet_bed.jpeg	12_pet_bed_thumb.jpg	20	1	2022-04-29 18:07:28
	13	3	2	2	Pet Food	Pet Food	13_pet_food.jpeg	13_pet_food_thumb.jpg	21	1	2022-04-29 18:07:28
	14	5	2	1	Leash	Leash	14_leash.jpeg	14_leash_thumb.jpg	15	1	2022-04-29 18:07:28
	15	3	3	7	Mattress	Mattress	15_mattress.jpeg	15_mattress_thumb.jpg	619	1	2022-04-29 18:07:28
	16	5	3	6	Shower Cur...	Shower Cur...	16_shower_curtai...	16_shower_curtain_thum...	11	1	2022-04-29 18:07:28
	17	4	3	10	Lamp	Lamp	17_lamp.jpeg	17_lamp_thumb.jpg	17	1	2022-04-29 18:07:28
	18	3	3	4	Sofa	Sofa	18_sofa.jpeg	18_sofa_thumb.jpg	630	1	2022-04-29 18:07:28
	19	1	4	2	Tent	Tent	19_tent.jpeg	19_tent_thumb.jpg	125	1	2022-04-29 18:07:28
	20	1	4	10	Sleeping Bag	Sleeping Bag	20_sleeping_bag....	20_sleeping_bag_thumb....	40	0	2022-04-29 18:07:28
	21	5	4	9	Fishing Pole	Fishing Pole	21_fishing_pole.jpeg	21_fishing_pole_thumb.jpg	48	0	2022-04-29 18:07:28
	22	3	4	3	Basketball	Basketball	22_basketball.jpeg	22_basketball_thumb.jpg	19	0	2022-04-29 18:07:28
	23	2	5	5	Dictionary	Dictionary	23_dictionary.jpeg	23_dictionary_thumb.jpg	21	0	2022-04-29 18:07:28
	24	2	5	5	Encyclopedia	Encyclopedia	24_encyclopedia.jpeg	24_encyclopedia_thumb.jpg	18	0	2022-04-29 18:07:28
*	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

## User:

	user_id	first_name	last_name	email	password	administrator	date_created
▶	1	Steve	Russel	steve@gmail.com	\$2a\$12\$Sxe.5U7k1yIB9/crrXICOd4TXpiQO3yI...	0	2022-04-29 18:05:32
	2	Shauna	Jones	shauna@gmail.com	\$2a\$12\$Q6sz0sWLH0rP2aIAYpZtm.loxLicuCWw...	0	2022-04-29 18:05:32
	3	Bryan	Matthew	bryan@gmail.com	\$2a\$12\$KX2REVZXZv5gBhM9jhnpbuP6OatWdt...	0	2022-04-29 18:05:32
	4	George	Martin	george@gmail.com	\$2a\$12\$b83FoZN5xmKmQOFw61pPF0lh1Gq3R...	0	2022-04-29 18:05:32
	5	Carly	Brown	carly@gmail.com	\$2a\$12\$QzAImgESEq7SCMZzikfwqOBNx0.xy4g...	0	2022-04-29 18:05:32
	6	Noah	Williams	noah@gmail.com	\$2a\$12\$yY9/4QtfyXEbjbfQ6RD1u.GBDQLN1ZI3...	0	2022-04-29 18:05:32
	7	Olivia	Johnson	olivia@gmail.com	\$2a\$12\$9SLjzSCTkBRzXrR.wzWU4.jw9WTisx51...	0	2022-04-29 18:05:32
	8	Emma	Jones	emma@gmail.com	\$2a\$12\$w5DHYmZgPqrKpuXP4.oe/N5g3nOw...	0	2022-04-29 18:05:32
	9	Elijah	Garcia	elijah@gmail.com	\$2a\$12\$LOfcmJcZcuZoJk12QMCOF/Tzk2AbE...	0	2022-04-29 18:05:32
	10	Charlotte	Miller	charlotte@gmail.com	\$2a\$12\$IPcZ/xoufithSggafEnTgOLPvM7ZZJ8B6J...	0	2022-04-29 18:05:32
	11	William	Davis	william@gmail.com	\$2a\$12\$ElLdszQostobR1KIYctY5uiuVPQnjkvAq...	0	2022-04-29 18:05:32
	12	James	Rodriguez	james@gmail.com	\$2a\$12\$b6k/0/rKs4nsg.qQj5UzEOpJkDvtEuaPu...	0	2022-04-29 18:05:32
	13	Amelia	Wilson	amelia@gmail.com	\$2a\$12\$tAP8HkJQn8ri5TrMzJ1jQe5pbc27LaHH...	0	2022-04-29 18:05:32
	14	Lucas	Anderson	lucas@gmail.com	\$2a\$12\$xhcs/VdyCaOjmXVjvIqs0eP.7U/J/RXON...	0	2022-04-29 18:05:32
	15	Evelyn	Thomas	evelyn@gmail.com	\$2a\$12\$.yCISdVGDNmL1pe9v/vpb.JlrBLGm10...	0	2022-04-29 18:05:32
	16	Vishal	Sharma	vsharma5@mail.sfs...	\$2b\$12\$nYQK0ytd3/VKVMadIfI1J.dkkRig2E/Sv0...	0	2022-05-16 18:25:04
*	HULL	HULL	HULL	HULL	HULL	HULL	HULL

## 5. GitHub Organization

### Main Branches

1) ***develop*** - Used to test everyone's changes on local environment before deploying on production environment. Only once all the changes were tested then these changes were merged on the main branch.

2) ***main*** - Used to deploy on production environment.

Both of these branches were protected only the Github Lead and Team Lead had access to write to these branches.

### File Structure

1) ***Milestones*** - This folder contains all the documents related to each milestone.

2) ***Application*** - This folder contains all the code related to the application, both backend and frontend.

3) ***Credentials*** - This folder contains all the code related to SSH key for EC2 access and MySQL DB credentials.

## Github Team Home Page

The screenshot shows a GitHub repository page for 'CSC-648-SFSU / csc648-03-sp22-team04'. The repository is private and has 29 branches. The 'Code' tab is selected. The main content area displays a table of files and their last commits:

File	Last Commit	Author
Milestones	Develop (#113)	4 days ago
application	Develop (#117)	4 days ago
credentials	Updated instructions for DB	2 months ago
.gitignore	Merging Develop to Main (#27)	2 months ago
LICENSE	Initial commit	3 months ago
README.md	Updated member table	3 months ago

The 'README.md' file content is as follows:

## csc648 Repository

Please when ready add your teams application URL or IP to the repository description. This will help with grading. Teams are expected to keep this value up to date.

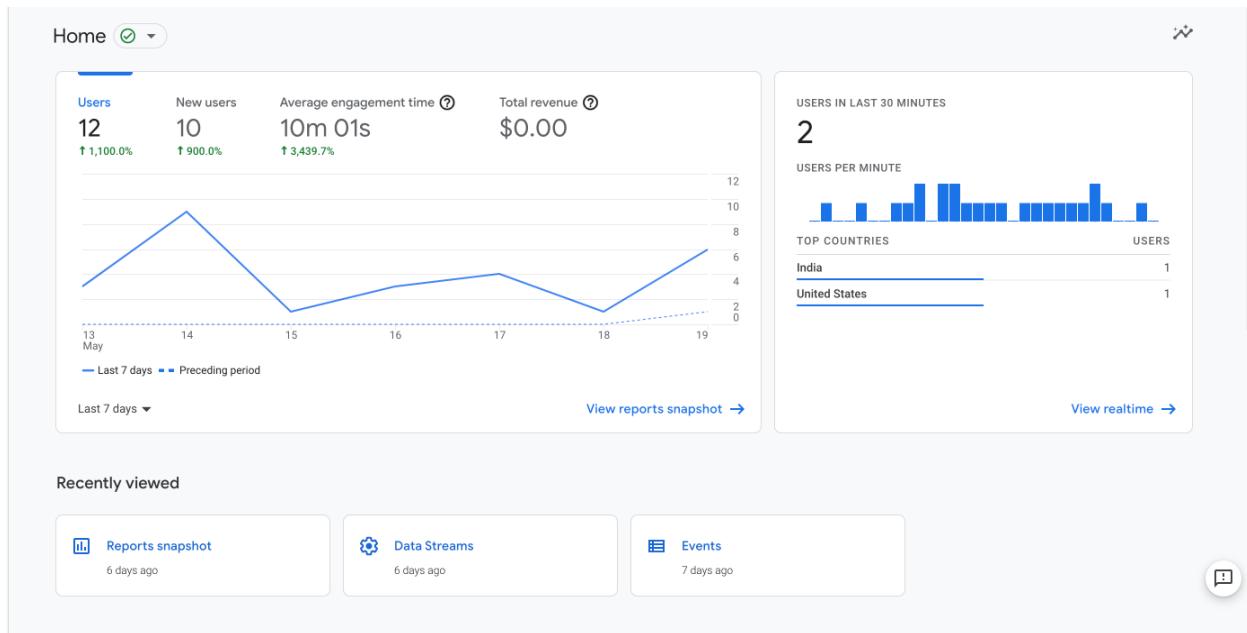
Please do the following steps before completing Milestone 0.

1. Change the name of the repository. All that needs to change is the NN to your respective team number. Team numbers whose value is less than 10, please pad with a 0. Ex team 1 is Team01 team 11 is Team11. Please make sure to also remove the username from the repository as well. Teams with incorrectly name repository will have points deducted from their milestone 0 grades.
  - Please follow the naming convention assigned by your instructor.
2. PLEASE REMOVE THE USERNAME FROM THE REPOSITORY NAME!!!
3. Add ALL members of your team to this repository. For it to count, they must ACCEPT the invite.
4. Fill out the table below

Student Name	Student Email	GitHub Username
Aaron Carlson	acarlson8@mail.sfsu.edu	ajccarlson
Siqi Guo	sguo4@mail.sfsu.edu	Guo-1999
Janvi Patel	jpatel6@mail.sfsu.edu	pjanvi22
Samantha Saxton-Getty	ssaxtongetty@mail.sfsu.edu	saxgetty
Abdullah Sharaf	fabdullah1@mail.sfsu.edu	abdullahsharafsfu
Vishal Ramanand Sharma	vsharma5@mail.sfsu.edu	vishals9711
Elyssa Mari Tapawan	etapawan1@mail.sfsu.edu	ElyssaMari

NO code needs to be stored in the root of your repository. You may rename the application folder if you like to your team's application name. But all source code related to your team's application should be stored inside the application folder.

## 6. Google Analytics Statistic Plot



## 7. Project Management

The tools we utilized throughout this project process were Discord, Trello, When2Meet, and Zoom. We utilized Discord and Zoom for communication purposes, while Discord served its purpose as a quick and easy way to text and communicate and we met three times a week in Zoom for voice communication. Trello and When2Meet were used to help organize, assign, and maintain tasks. Presented below is a snippet of our BusyGator Trello board, with organized columns depending on the task at hand. I've also included a snippet of our Discord server to demonstrate our communication when we were offline on Zoom.

The image displays two screenshots of our project management tools. The top half shows a Trello board titled 'BusyGator' with several lists: 'MEETINGS', 'MILESTONES', 'GITHUB', 'TO DO', and 'DONE'. The 'TO DO' list contains various tasks related to backend connections, page edits, product card uniqueness, and UI/UX improvements. The bottom half shows a Discord server interface for 'CSC 640 - Team 4'. The left channel, '#m3', has a message from Sami about React-Bootstrap. The right sidebar shows team roles and member status.

## 8. Team Member Self Assessment and Contributions

### Aaron Carlson

CSC 848 Team 4 Self Evaluation - Aaron Carlson

 Aaron James Carlson  
To: Siqi Guo; Janvi Manishbhai Patel; Samantha Troi Saxton-Getty; Abdullah Sharaf; Vishal Ramanand Sharma; Elyssa Mari Tapawan

Sat 5/21/2022 1:04 PM

As the backend lead for Team 4, my contributions mainly involved connecting frontend features to our database. Some such contributions include:

- Providing functionality to and connecting pages such as the individual product, message, and "My Page" to the database
- Handling all instances of input validation and validation messages
- Creating middleware to streamline backend tasks
- Creating dummy data for the database and updating its schema to better suit tasks
- Fixing more complicated bugs and annoyances found during meetings
- Helping team members with issues involving the backend or the database specifically

In my work on the project, I have made 24 commits to our Github repository's "develop" branch. I would try to make separate commits and branches off develop for each feature and create pull requests to merge back with develop once each feature was complete.

The main challenge I faced with this project was learning and getting accustomed to React-Bootstrap, as I had little to no experience with it. Though the process took quite a while, once I became comfortable with it, I was able to complete tasks with much greater ease and speed due to its conveniences. Outside of coding, another major challenge I faced lay with using Github in a team setting. In the beginning, I wasn't very familiar with the correct procedures for creating branches and pull requests and handling merge conflicts, leading to some headaches and errors early on.

Given everything I've learned from this project, one thing I would do better is provide more frequent and more descriptive inline comments in my code. This would give team members an easier time in understanding what my code does so they can build off and around it with minimal issues.

### Siqi Guo

CSC 648 Team 4 Self Evaluation

 Sigi Guo  
To: Vishal Ramanand Sharma; Samantha Troi Saxton-Getty; Aaron James Carlson; Janvi Manishbhai Patel; Abdullah Sharaf; Elyssa Mari Tapawan

Sat 5/21/2022 1:10 PM

Sigi Guo -Front end

Self Assessment and contributions

Contribution to team project and teamwork -

I am the front-end member of the team. Worked on the task given by the team leader. Worked on the part of the post, the App.css, UI changes and changed/fixed our website, and made all work better. Finish given task on milestone documentation. join and communicate with the team in every zoom meeting or discord.

- Github file headers
- Signup and message change buttons
- UI changes for Login, signup, and My posts
- Item count aesthetic
- change category/location on the My post page
- work with my groupmate to figure out the problem

Number of Contributions to Dev-Branch - 3 total

On the Dev-branch, I have made 3 commits. I made every change in other branches, which will make it easier to let my teammates know what I made changes. Then my Github leader will merge to the development branch when my task is complete.

Main challenges encountered in team project -

I don't have much knowledge about the front-end, only related to class 317. In the 317 courses, I only know some basic stuff like login/register/Error fix and some Javascript. I enjoy learning about the resume or cover letter knowledge without the project, which is a big help when I start to find the work. I also enjoy and let me add some new experiences throughout the whole project.

What would I do better next time based on what was learned in the class about SE management and processes -

I am very thankful to my group. They let me know a lot of new stuff. Based on what was learned in the class, I feel that this project is the first time we have completed it together, and it also made me know the importance of the team. I will better use the experience I have learned during this time in every project in the future. For example, I will listen carefully to customers' needs, improve work efficiency, and formulate strict rules. Overall, I learned a lot of methods that will better help me complete my future tasks.

Best  
Sigi Guo

## **Janvi Patel**

CSC 648- Self assessment and contributions

JP Janvi Manishbhai Patel  
To: Samantha Troi Saxton-Getty; Abdullah Sharaf; Siqi Guo; Vishal Ramanand Sharma; Aaron James Carlson; Elyssa Mari Tapawan  
Sat 5/21/2022 1:48 PM

Hi all,  
as a backend member of my team I mainly worked on the connections of the login, signup and homepage to the database. some of the tasks include:

- merging login and signup buttons
- creating dummy password link
- resetting search results
- moving buttons inside the page
- forcing user to sign up if not logged in

On the GitHub develop branch I have made 9 commits. I made changes in other branches and merged them to develop branch.

The first challenge I faced was working with React-Bootstrap as I never worked with react-bootstrap before, and once I learned it through the process I became more comfortable with it. The other challenge I faced was with MySQL Workbench as I didn't work with it before too, and I had many troubles connecting with the database and server. My team members were very supportive and helped me work with MySQL workbench all along. This was the first time I worked in a GitHub team setting so I did have merge conflicts and the GitHub lead always made it easier to fix those errors and I learned a lot after that.

There are many things I could do better, like learn more about the database or the server I will be using like in this case MySQL as I was going to work on the backend team. Other things like learn more about how to make commits in GitHub where it is a team-based project to avoid conflicts.

From,  
Janvi Patel

## **Samantha Saxton-Getty**

Team Member Self-Assessment and Contributions | Team 04

SS Samantha Troi Saxton-Getty  
To: Aaron James Carlson; Siqi Guo; Janvi Manishbhai Patel; Abdullah Sharaf; Vishal Ramanand Sharma; Elyssa Mari Tapawan  
Sat 5/21/2022 1:43 PM

Samantha Saxton-Getty

**My Contributions:**

- Designed and upheld SQL database
- Creation of partial database sample data
- Implemented all comments throughout code
- Team Lead management and task delegation
- Formatted and prepared all documentation
- Designed website logo and map
- Implemented placeholder data for several user inputs
- Implemented a large portion of each milestone documentation
- Bridged communication between team questions and professor responses
- Organized and assigned deadlines for workflow through Trello
- Held 3 weekly meetings to maintain schedule and progress of workflow

**Submissions to development branch: 1**

**Challenges:**

From a general standpoint, the biggest challenge I encountered in this team project was the lack of understanding with backend besides creating the database and filling it with temporary data. The most experience I have had with backend is the database course offered at SFSU and it didn't cover details in utilizing a database with a website. I think I could have contributed more help to the backend having been more educated in it beforehand.

From a team lead standpoint, the biggest challenge I encountered was leading a team successfully through having solid communication on GitHub, in meetings, and offline through text. Our team ran into issues with task delegation where some individuals would work on their tasks and not finish in the allotted time. Due to this, other team member who would need to start their work would then be working on an out-dated repository and when both members would commit their work, there would be many merge conflicts to be resolved.

**Improvements:**

As a team member, I have trouble with public speaking and having high anxiety due to being diagnosed with an anxiety disorder. Because of this, I had difficulty preparing for each meeting and participating in vocal interactions often. Going forward I need to work on my delivery when presenting and / or speaking to members of my team and management.

As a team lead, as stated in the challenges section, I struggled quite a bit with efficient task delegation, causing errors when members would merge their code due to working on the same repository at the same time. Going forward in any team lead position, I will work harder to have more cohesive GitHub pull requests for merging to avoid so many errors. While everything was completed on time, I think it would cause less stress and time to have a lower amount of major merge conflicts and allow us to have a much more successful workflow.

--  
Best,

Samantha Saxton-Getty  
SFSU ID: 916198535

## **Abdullah Sharaf**

### CSC 648- Self assessment and contributions | Team 4



**Abdullah Sharaf**

To: Samantha Troi Saxton-Getty; Vishal Ramanand Sharma; Elyssa Mari Tapawan; Aaron James Carlson; Siqi Guo; Janvi Manishbhai Patel  
Hi All,



Sat 5/21/2022 1:30 PM

As a Front End of Team 4, I made many responsibilities and contributions to the team project and teamwork. The tasks I worked on consists of:

#### My Page

- Create My Page with tabs for user data (posts/ messages)
- UI for My Messages and My Posts
- Redirecting user depending on their state

#### Home

- Create items count
- Allowing the user to search for products from any pages
- Align product details
- Implement required text for header

#### Messages

- Create Message Seller page
- UI and design of the Message Seller page

#### About

- Cooperate with design and responsiveness of about page

#### Other

- Help team member fix issues with setting up MySQL on their PC, creating schema, inserting data, and connecting to the DB.
- Make the footer responsive and align correctly for pages with short data

I have made a total of 33 commits from which 16 are to the development branch, and the rest are commits to feature branches which were merged to the development branch by Vishal.

The first challenge that I faced was working with react bootstrap because I've never used react up until now, but it was a great learning experience. The second challenge I faced was with GitHub, I've never used GitHub when working with a big team and a big project, so initially it was challenging for me to commit properly on the correct branches and raise pull requests the right way, but I was able to figure it out quickly and everything became much smoother.

There are many things that I could do better next time based on what was learned in the class about SE management and processes. For example, I could design the UI for My Page in a way that it doesn't look crowded and looks simpler when it's on mobile view.

Thanks,

**Abdullah Sharaf**

Student | SFSU | Computer Science

## **Vishal Ramanand Sharma**



**Vishal Ramanand Sharma**

To: Samantha Troi Saxton-Getty; Aaron James Carlson; Siqi Guo; Janvi Manishbhai Patel; Abdullah Sharaf; Elyssa Mari Tapawan



Sat 5/21/2022 1:02 PM

Hi All,

As a GitHub lead of Team 4, I made many responsibilities and contributions to the team project and teamwork. From setting up the folder structure for the application and creating the initial code for the team to follow, I did numerous tasks:

- Setting up AWS EC2 for deployment
- Setting up MySQL and Nginx in AWS
- Setting up higher level code relating to DB connection from backend and frontend context for data.
- I reviewed each PR and checked if code was upto the mark.
- Merging code onto the develop branch, having it tested and then pushing everything onto main branch.
- Deploying the latest changes on AWS after everything was merged onto main branch
- Helped team members with their issues in frontend and backend.

On the GitHub team development branch, I have made 44 commits. On other branches, I would try and commit each change I do and then merge it to the development branch whenever my tasks were complete. While I made 44 commits, a lot of these were a build version to be deployed onto the main branch to be used by AWS.

The main challenges I encountered in the team project were keeping the codebase intact so that the code on main branch was always runnable and not breaking in any form. I also faced a lot of issue setting up EC2 instance for the 1<sup>st</sup> time, before each deployment if there were any changes in SQL structure, I had to change the DB in AWS as well. I had to password protect MySQL and then add code to access it. Sometimes the Nginx would be down due to a faulty piece of code, so we had to revert to previous build and see what went wrong.

There are many things I can do better next time based on what was learned in the class. For example, I would streamline the process of deployment, and raising the PR. I would set up auto-deploy as that would cause less interference from user. I would set up strict rules for raising PR and how a PR message would be structured based on what I learnt in class. Overall, I learned a lot of methods and processes that I could have implemented if I have known them early on, which would have better improved the GitHub process.

Thanks & Regards

Vishal Ramanand Sharma

MS Computer Science

Email : vsharma5@mail.sfsu.edu

## *Elyssa Mari Tapawan*

### CSC 648-848 Group 4: Team Member Self Assessment and Contributions

ET

Elyssa Mari Tapawan

To: Samantha Troi Saxton-Getty; Vishal Ramanand Sharma; Aaron James Carlson; Abdullah Sharaf; Siqi Guo; Janvi Manishbhai Patel  
Elyssa Tapawan - Front End Lead



Sat 5/21/2022 1:08 PM

As a Front-End lead of Team 4, I made many responsibilities and contributions to the team project and within our team. From UI design to writing up documents, I did numerous tasks:

- Executive Summary
- QA Test Plan
- UI Storyboards for Each Main Use Case
- Set up most of the pages and components: Navbar, Footer, About Us, Home page, Product cards, Login and Sign Up, and Individual Product Page
- Oversaw UI Changes from other team members
- Updated pages and components based on UI feedback and back-end implementations
- Helped team members with react-bootstrap
- Actively communicate with the team on Discord and Trello on any help that was needed by providing some solutions and/or opinions

On the Github team development branch, I have made 7 commits. On other branches, I would try and commit each change I do and then merge it to the development branch whenever my tasks were complete, which lessened my commits. While I made 7 commits, I made many contributions by means of additions and deletions of code on GitHub. (additions: 29,636 ++, deletions: 57,384 --)

The main challenges I encountered in the team project were learning to react and bootstrap. Before this project, I had little to no knowledge of how to use react and bootstrap so I had a very hard time learning and teaching these tools at the beginning of the project. Also, I would have a hard time communicating with other members when I was building each page, so I didn't really know how to split the workload or know who is able to do what. As we progressed, we did improve the way our team communicated and was upfront on tasks that needed to be done.

There are many things I can do better next time. For example, the process of building our UI design was difficult. The storyboards were a great base to build our UI design, but what I would like to have done differently is making a high-fidelity storyboard just to get a better idea of how we would like to have the site's design be from the start. Overall, I learned a lot of methods and processes that I could have implemented if I have known them early on, which would have better improved the website.

From,

Elyssa Tapawan