

# SW Engineering CSC648/848 Spring 2022

## GatorTrade

### Team05

### Milestone 2

20 March 2022

Members:

**Team Lead:** Kishan Patel - kpatel0@sfsu.edu

**Front End Lead:** Michael Cheung

**Front End Member:** Joanne Wong

**Back End Lead:** Faisal Zaheer

**Back End Member:** Juan Hernandez

**Github Master:** Ze Lei

History Table:

Date Submitted	Date Revised
20 March 2022	23 March 2022

	2
<b>Executive Summary:</b>	<b>3</b>
<b>List of Data Items and Entities - Expanded As Necessary</b>	<b>4</b>
<b>Functional Requirements - Prioritized</b>	<b>6</b>
Priority 1:	6
Priority 2:	6
Priority 3:	7
<b>UI Storyboards - Low-Fidelity and B/W Diagrams</b>	<b>8</b>
Use Case: Buying a textbook from GatorTrade	8
Use Case: Reviewing the seller after a transaction	11
Use Case: Steve receives a message and completes a transaction	15
Use Case: Managing the GatorTrade application	18
Use Case: Listing a job opening	20
<b>High-Level Architecture and Database Organization Summary</b>	<b>23</b>
<b>Key Risks</b>	<b>24</b>
<b>Project Management</b>	<b>25</b>

## **1. Executive Summary:**

Currently, there is a lack of a dedicated platform for the students, faculty, and staff of San Francisco State University (SFSU) to quickly and securely exchange goods. Students often try to find the cheapest and most efficient way to get the items they need for their classes or for their stay at SFSU. If a student wanted to sell a used textbook for a particular class, it is hard to find a suitable platform since they would not be certain what type of platform other students would use to find used textbooks for sale. Although there are some undedicated places for SFSU members to sell their belongings like Amazon, SFSU Reddit, and SFSU Discord, they have less traffic and exposure since not every SFSU member would use or know about that platform. Students and employees that have hobbies in arts and crafts can also list their handicrafts on this platform to be sold and exchanged with students and fellow coworkers. Having a dedicated platform can also provide a sense of safety for the users, especially since having a place to have secure transactions can allow for accountability to occur.

Our application will be tailored to the students, faculty, and staff at SFSU. Belongings they may not want anymore or think others can benefit from can be listed here. The platform will allow for SFSU students, faculty, and staff to buy, sell, or exchange items with one another. This means that the used textbooks, electronics, and other used or handcrafted goods can find new owners amongst the SFSU community. Users of this application shall be able to search and filter through tags for the goods they are looking for such as technology, textbooks, or crafts. Also, since this application is exclusively for the SF State community, users will get some unique features like being able to search for class materials by class name. It can also be an outlet for artistic students to showcase their skills and sell handmade goods to fellow students and SFSU employees.

Our team is a student start-up composed of students at SFSU. Each member is well-versed in the problems and concerns we address with our application since we have all experienced it firsthand at least once. Therefore, it is safe to say that we know the target audience well because we are in fact part of that same target audience the application strives to appeal to.

## 2. List of Data Items and Entities - Expanded As Necessary

- a. Users: stores user information such as user password, email, and name.
  - 1. `idUsers`: primary id used in database identification of the user
  - 2. `password`: user-created key used for login
  - 3. `email`: user's SFSU email account; used for login
  - 4. `firstname`: user's first name
  - 5. `lastname`: user's last name
  
- b. Items: stores item information such as title, price, and description
  - 1. `idItems`: primary id used in database identification of the item
  - 2. `category`: used to reference the `Categories` table
  - 3. `seller`: id that is linked to the user's id
  - 4. `photopath`: path of the photo of the item
  - 5. `description`: description of the item
  - 6. `title`: title of the item
  - 7. `price`: price set for the item by seller
  - 8. `approved`: boolean to check if the post has been approved by admin or not
  
- c. Reviews: stores review information such as reviewee, rating, and reviewer
  - 1. `idReviews`: primary id used in database identification for review
  - 2. `reviewer`: used the reference the `Users` table for the person making the review
  - 3. `reviewee`: used to reference the `Users` table for the person being reviewed
  - 4. `rating`: value given to the reviewer gave the reviewee
  - 5. `header`: title of the review
  - 6. `body`: context of the review
  
- d. Messages: stores information of the messages being sent [one way only]

1. `idMessages` : primary id used in database identification of messages
  2. `item`: used to reference the `Items` table for the product the message is about
  3. `sender` : used to reference the `Users` table for the person sending the message
  4. `receiver`: used to reference the `Users` table for the person receiving the message
  5. `body`: content of the message
- e. Categories: It will contain the following fields for title: Electronic, Text, Handicraft, Educational\_supply, and Food.
1. `idCategories`: primary id used in database identification of categories
  2. `title`: name of the category

### **3. Functional Requirements - Prioritized**

#### **Priority 1:**

##### **1. Unregistered Users:**

- 1.1** Unregistered users shall be able to register a new account with a valid SFSU email address.
- 1.2** Unregistered users shall be able to access the homepage of the website.
- 1.3** Unregistered users shall be able to browse items on the website.
- 1.4** Unregistered users shall be able to view details of an item, including price, image, item details, etc.
- 1.5** Unregistered users shall be able to search for items on the website.
- 1.6** Unregistered users shall be able to create an account (lazy registration).
- 1.7** Unregistered users shall be able to sort search results by price and title.

##### **2. Registered Users:**

- 2.1** Registered users inherit all functional requirements from Unregistered users.
- 2.2** Registered users shall be able to login
- 2.3** Registered users shall be able to change their password.
- 2.4** Registered users shall be able to post items to sell on the website.
- 2.5** Registered users shall be able to send messages to other registered users.  
[one way only]
- 2.6** Registered users shall be able to logout from their account.

##### **3. Admin:**

- 3.1** Admin inherits all functional requirements from Registered users.
- 3.2** Admin shall be required to authorize items posted by registered users.
- 3.3** Admins shall be able to delete any item from the website.

#### **Priority 2:**

##### **2. Registered Users:**

**2.7** Registered users shall be able to exchange items with other registered users.

**2.8** Registered users shall be able to delete items posted by them only.

**2.9:** Registered users shall be able to write reviews for other registered users only.

3. Admin:

**3.4** Admin shall be able to delete user accounts permanently from the website.

**Priority 3:**

1. Unregistered Users:

**1.7** Unregistered users shall be able to filter search results with specific tags and labels.

2. Registered Users:

**2.10** Registered users shall be able to edit their user profile picture.

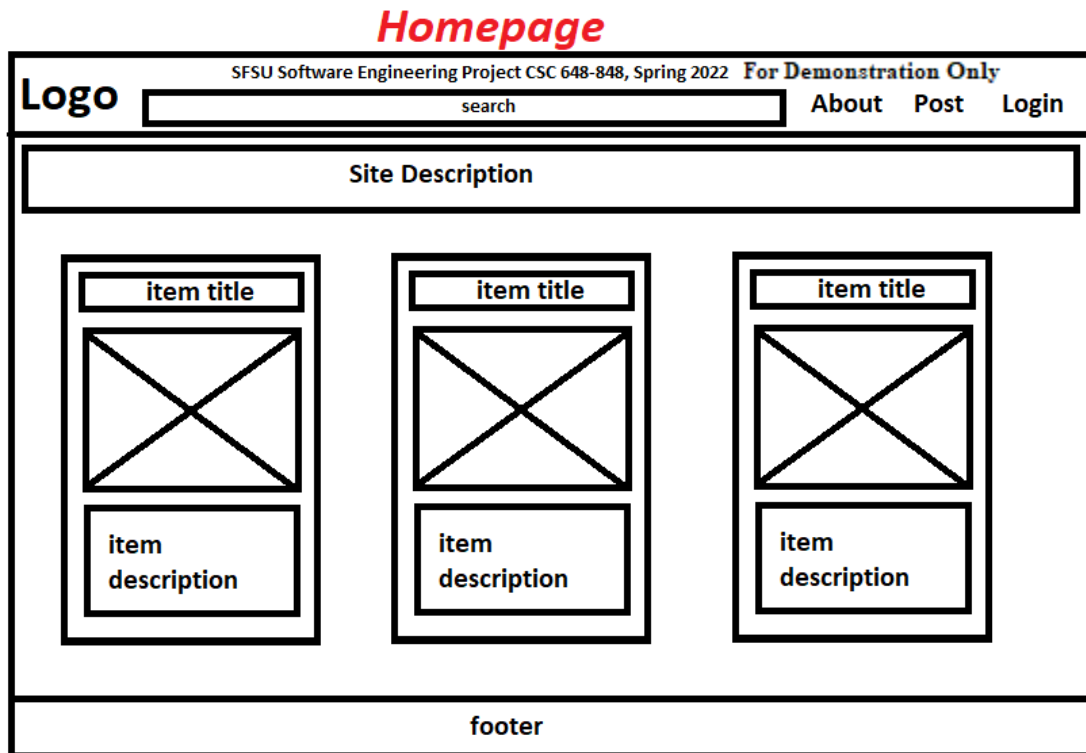
3. Admin:

**3.5** Admin shall be able to add or delete categories of items from the website.

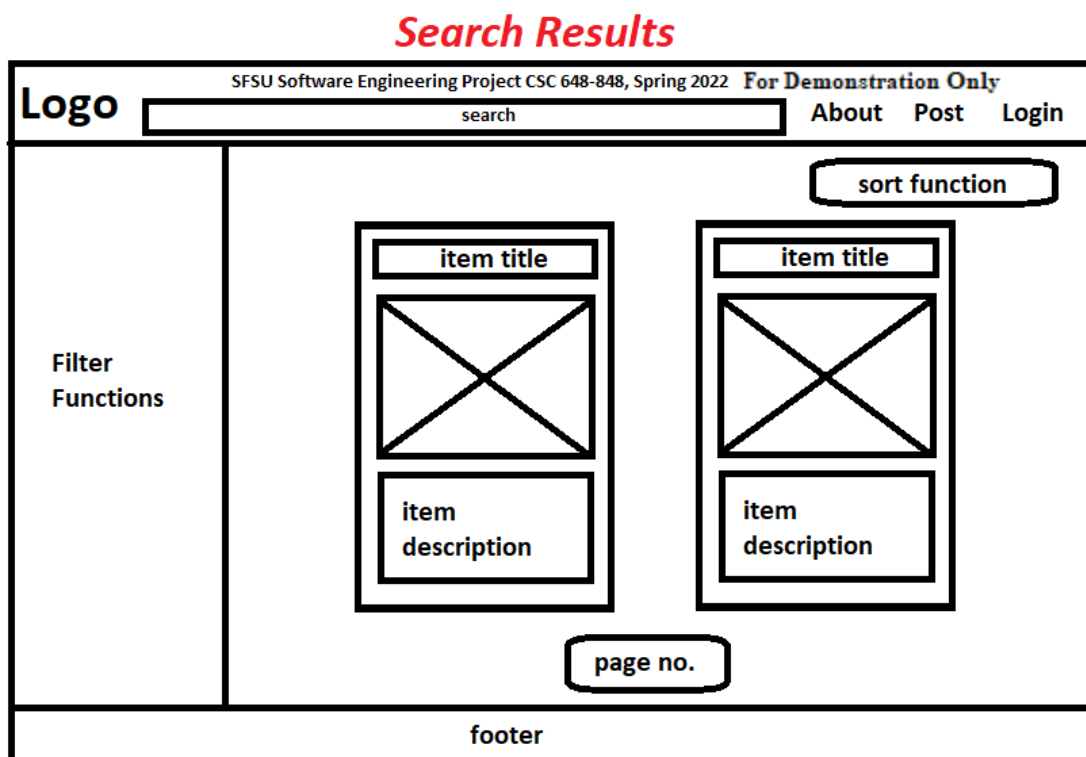
#### 4. UI Storyboards - Low-Fidelity and B/W Diagrams

##### Use Case: Buying a textbook from GatorTrade

A. Jennifer lands on the homepage and searches for a textbook in the search bar.



B. Jennifer gets taken to the search results page and clicks on a post







E. But Jennifer doesn't have an account, so she registers one

### Register Page

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

Logo

\* First Name

\* Last Name

\* Email

\* Password

\* Re-enter Password

\* ☐ terms and conditions

Register

footer

F. After successfully registering for an account, Jennifer gets taken to the message page.

### Send Message Page

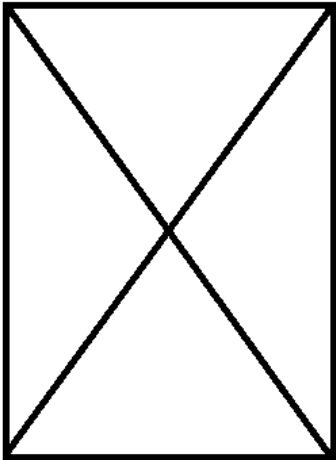
SFSU Software Engineering Project CSC 648-848, Spring 2022 For Demonstration Only

Logo

search

About Post Login

Item Title

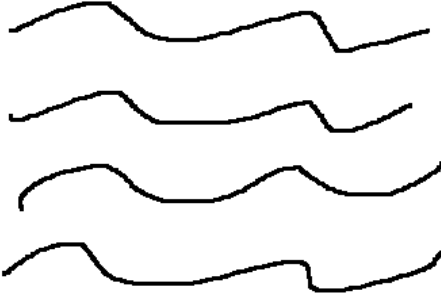


Send

footer

G. Jennifer gets a confirmation that her message was sent successfully

### Confirmation Page

Logo	SFSU Software Engineering Project CSC 648-848, Spring 2022 For Demonstration Only	
	<input type="text" value="search"/>	<a href="#">About</a> <a href="#">Post</a> <a href="#">Login</a>
<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; width: 150px; height: 150px; position: relative;"> <div style="position: absolute; top: 0; bottom: 0; left: 0; right: 0;">X</div> </div> <div style="text-align: center;">  </div> </div>		
footer		

### Use Case: Reviewing the seller after a transaction

A. Jerry just finished a transaction and the seller asked him to do a review.

### Homepage

Logo	SFSU Software Engineering Project CSC 648-848, Spring 2022 For Demonstration Only	
	<input type="text" value="search"/>	<a href="#">About</a> <a href="#">Post</a> <a href="#">Login</a>
Site Description		
<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 5px; text-align: center;"> <div style="border: 1px solid black; width: 100px; height: 20px; margin-bottom: 5px;">item title</div> <div style="border: 1px solid black; width: 100px; height: 100px; position: relative; margin: 0 auto;"> <div style="position: absolute; top: 0; bottom: 0; left: 0; right: 0;">X</div> </div> <div style="border: 1px solid black; width: 100px; height: 40px; margin-top: 5px;">item description</div> </div> <div style="border: 1px solid black; padding: 5px; text-align: center;"> <div style="border: 1px solid black; width: 100px; height: 20px; margin-bottom: 5px;">item title</div> <div style="border: 1px solid black; width: 100px; height: 100px; position: relative; margin: 0 auto;"> <div style="position: absolute; top: 0; bottom: 0; left: 0; right: 0;">X</div> </div> <div style="border: 1px solid black; width: 100px; height: 40px; margin-top: 5px;">item description</div> </div> <div style="border: 1px solid black; padding: 5px; text-align: center;"> <div style="border: 1px solid black; width: 100px; height: 20px; margin-bottom: 5px;">item title</div> <div style="border: 1px solid black; width: 100px; height: 100px; position: relative; margin: 0 auto;"> <div style="position: absolute; top: 0; bottom: 0; left: 0; right: 0;">X</div> </div> <div style="border: 1px solid black; width: 100px; height: 40px; margin-top: 5px;">item description</div> </div> </div>		
footer		

B. Jerry goes to the login page from the homepage.

### Login Page

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

**Logo**

Email

Password

Forgot Password?

Log In

New user? Register Here!

footer

C. After successfully logging in, Jerry gets taken to the homepage.


### Homepage

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

**Logo**  [About](#) [Post](#) [Login](#)


Site Description

item title




item description

item title



item description

item title




item description

footer

























D. Jerry goes to his account to find his recent messages.

### Registered User's Account Page

Logo	SFSU Software Engineering Project CSC 648-848, Spring 2022 For Demonstration Only	
	search	About Post Login
		
Profile	Your Posts	Settings
Recent Messages		
footer		

E. Jerry sees the message he sent to the seller he just completed a transaction with.

### Messages Page

Logo	SFSU Software Engineering Project CSC 648-848, Spring 2022 For Demonstration Only									
	search	About Post Login								
<table border="1"> <thead> <tr> <th>Sent</th> <th>Incoming</th> </tr> </thead> <tbody> <tr> <td>   </td> <td>   </td> </tr> <tr> <td></td> <td>   </td> </tr> <tr> <td></td> <td>   </td> </tr> </tbody> </table>			Sent	Incoming	 	 		 		 
Sent	Incoming									
 	 									
	 									
	 									
footer										

F. Jerry writes his review and clicks submit

**Review Page**

Logo

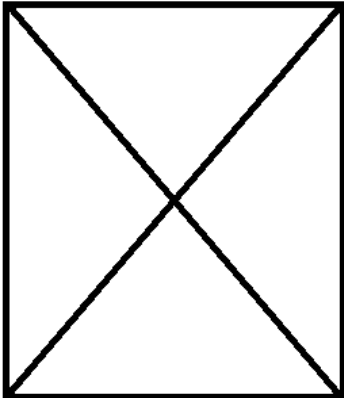
SFSU Software Engineering Project CSC 648-848, Spring 2022 For Demonstration Only

About

Post

Login

Site Description



submit

cancel

footer

G. Jerry gets a thank you page for submitting a review

**Thank You Page**

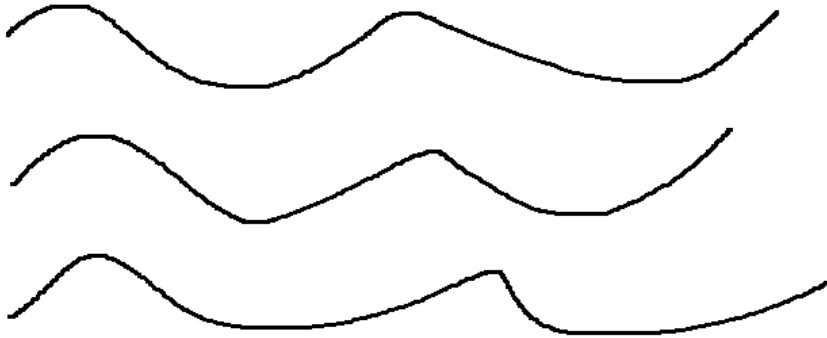
Logo

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

About

Post

Login

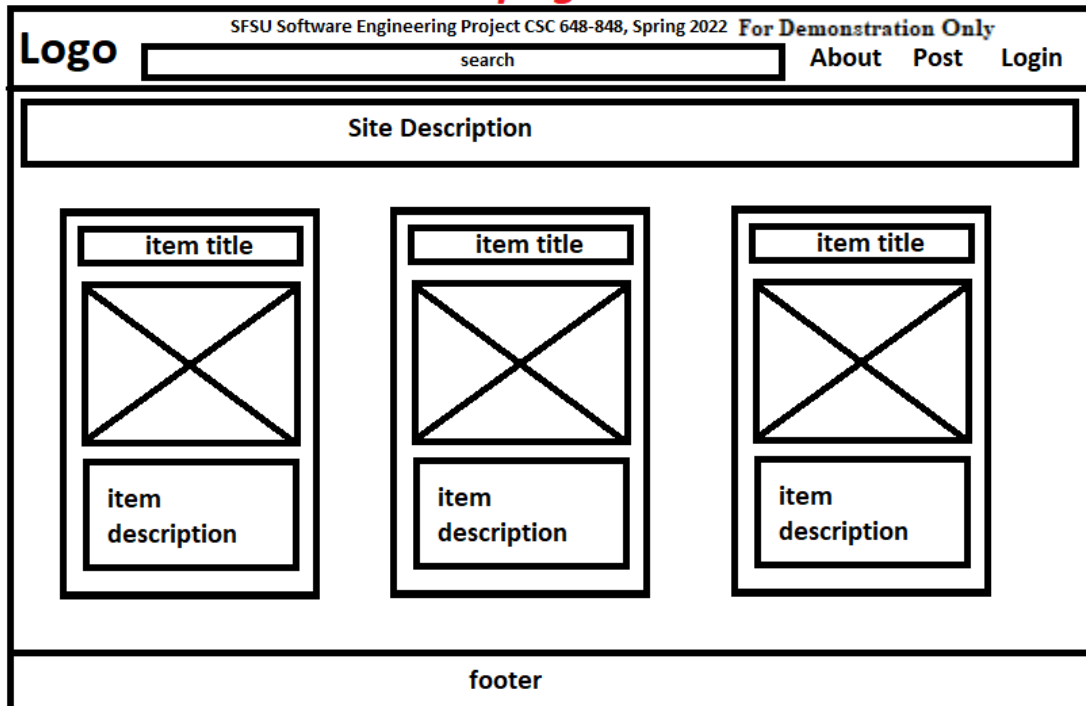


footer

**Use Case: Steve receives a message and completes a transaction**

A. *Steves goes to the homepage to login..*

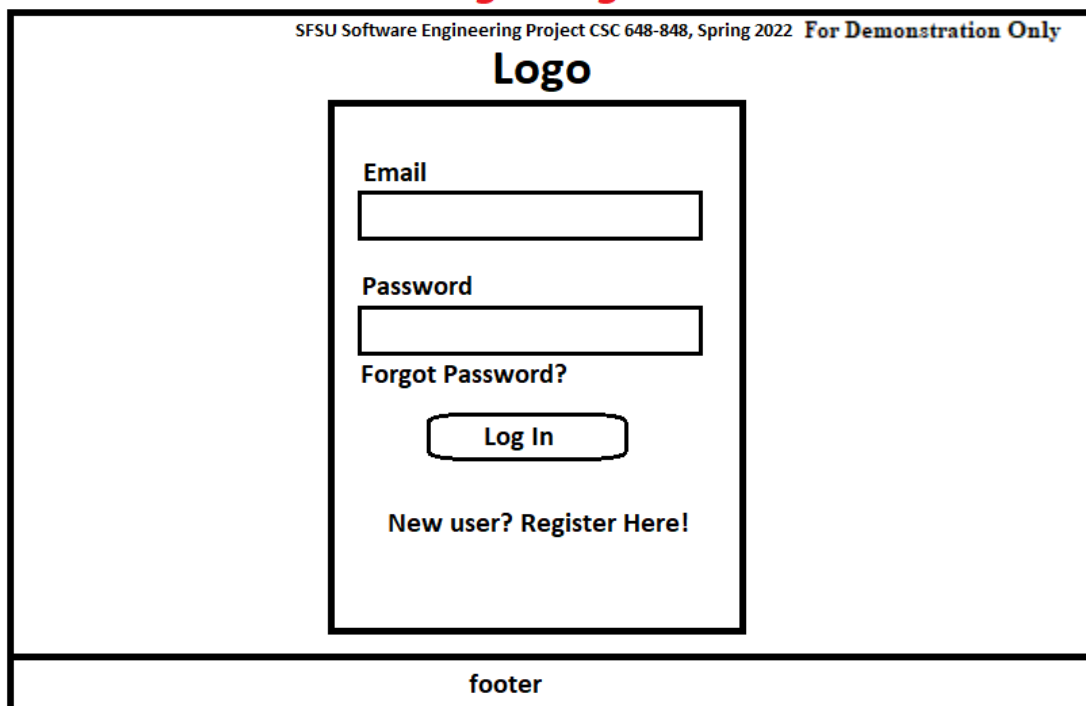
**Homepage**



The diagram shows a web page layout for a homepage. At the top is a header bar containing a 'Logo' on the left, a search bar in the center, and links for 'About', 'Post', and 'Login' on the right. Below the header is a 'Site Description' section. The main content area features three identical item cards arranged horizontally. Each card has a title box labeled 'item title', a placeholder image box with a large 'X', and a description box labeled 'item description'. At the bottom is a 'footer' section.

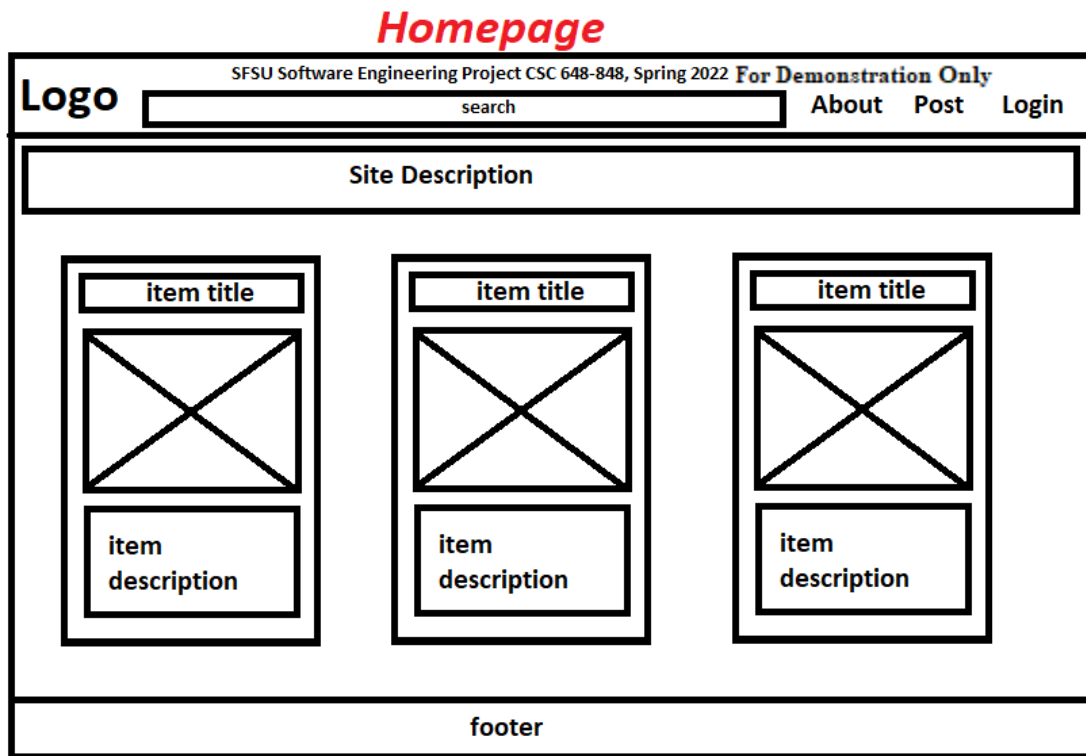
B. *Steve is taken to the homepage after logging in.*

**Login Page**

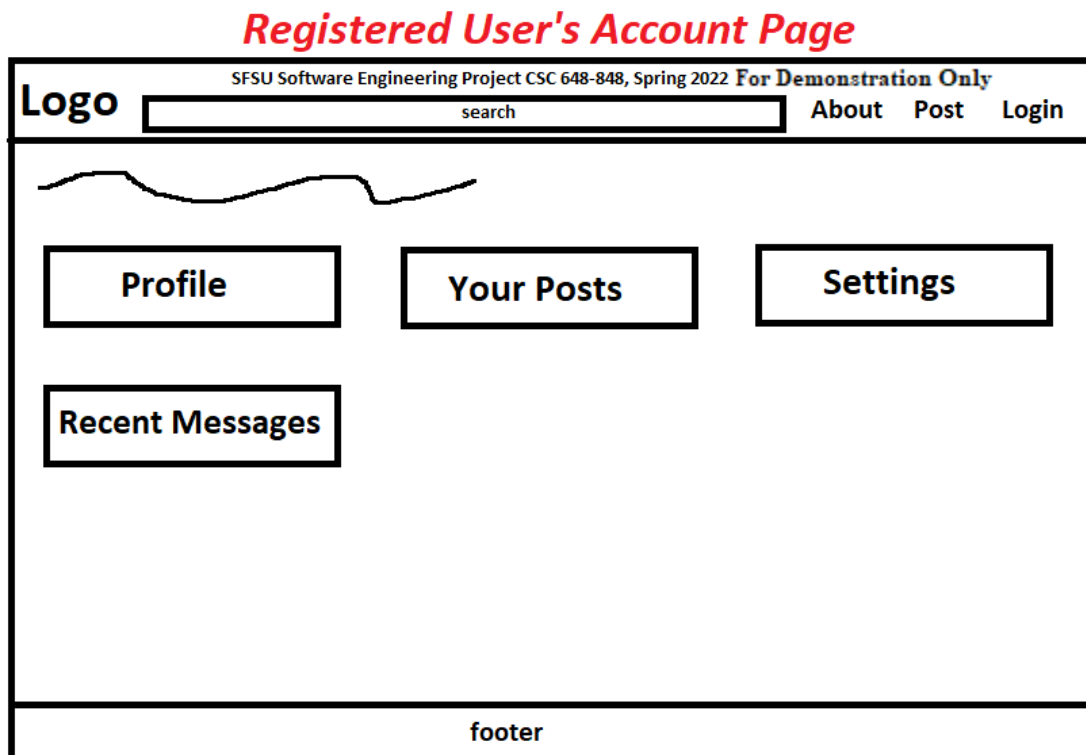


The diagram shows a web page layout for a login page. At the top is a header bar containing the text 'SFSU Software Engineering Project CSC 648-848, Spring 2022' and 'For Demonstration Only'. Below the header is a 'Logo'. The main content area is a login form with fields for 'Email' and 'Password', a link for 'Forgot Password?', a 'Log In' button, and a link for 'New user? Register Here!'. At the bottom is a 'footer' section.

C. Steve goes to his account page from the homepage.



D. Steve is taken to the recent message page after clicking on recent messages





E. Steve checks the most recent incoming message and reaches out to sender

**Messages Page**

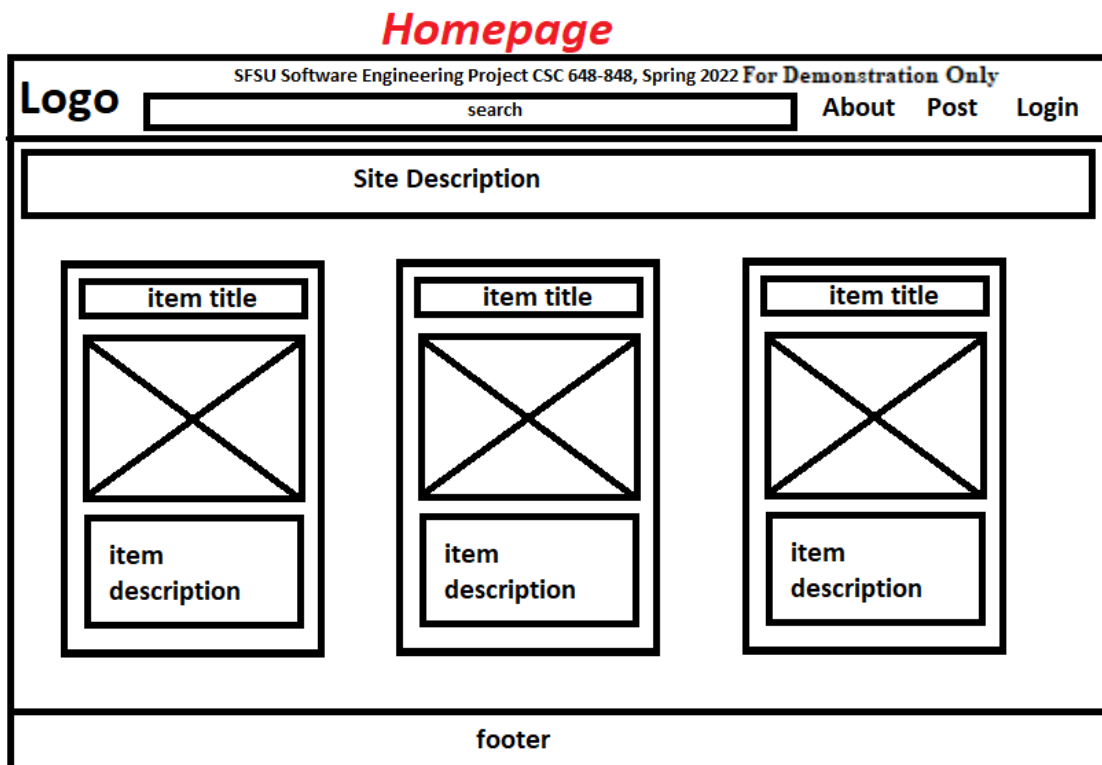
<div style="display: flex; justify-content: space-between; align-items: center;"><div><b>Logo</b></div><div>SFSU Software Engineering Project CSC 648-848, Spring 2022 <b>For Demonstration Only</b></div><div><a href="#">About</a> <a href="#">Post</a> <a href="#">Login</a></div></div> <div style="display: flex; align-items: center; margin-top: 5px;"><input style="width: 350px; border: 1px solid black;" type="text" value="search"/></div>	
<div style="text-align: center; border-bottom: 1px solid black; margin-bottom: 10px;"><b>Sent</b></div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"><div style="display: flex; align-items: center;"><div style="border: 1px solid black; width: 40px; height: 40px; margin-right: 10px; text-align: center; line-height: 40px;">X</div><div style="border-bottom: 1px solid black; width: 100%;"></div></div></div>	<div style="text-align: center; border-bottom: 1px solid black; margin-bottom: 10px;"><b>Incoming</b></div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"><div style="display: flex; align-items: center;"><div style="border: 1px solid black; width: 40px; height: 40px; margin-right: 10px; text-align: center; line-height: 40px;">X</div><div style="border-bottom: 1px solid black; width: 100%;"></div></div></div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"><div style="display: flex; align-items: center;"><div style="border: 1px solid black; width: 40px; height: 40px; margin-right: 10px; text-align: center; line-height: 40px;">X</div><div style="border-bottom: 1px solid black; width: 100%;"></div></div></div> <div style="border: 1px solid black; padding: 5px;"><div style="display: flex; align-items: center;"><div style="border: 1px solid black; width: 40px; height: 40px; margin-right: 10px; text-align: center; line-height: 40px;">X</div><div style="border-bottom: 1px solid black; width: 100%;"></div></div></div>
<b>footer</b>	

### Use Case: Managing the GatorTrade application

Robert is a Faculty member. He was approached by a SFSU student to help manage the GatorTrade application as an admin. Using an admin account given to Robert, he can manage all buy, sell, and exchange postings on the application. One of his major responsibilities include authorizing all posts before they are live on the website. He can browse through the posts on the home page or on any specific pages. From there, as an admin, Robert can delete the posting should it contain any inappropriate information or content. He can perform these actions using Workbench.

### Use Case: Listing a job opening

A. Franklin, an instructor, goes to the Homepage and tries to make a new post.



B. Since he is not logged in, he is taken to the login page.

### Login Page

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

Logo

Email

Password

Forgot Password?

Log In

New user? Register Here!

footer

C. Since Franklin is not a registered user, he registers on the register page

### Register Page

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

Logo

\* First Name

\* Last Name

\* Email

\* Password

\* Re-enter Password

\* ☐ terms and conditions

Register

footer

D. After registration, Franklin goes to add a new post and fills out all the information required.

**Create Post Page**

Logo

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

search

About

Post

Login

Creating new post

\* Title:

\* Price:

\$

\* Tag:

\* Description:

\* Photo(s):

Photo function

Submit

Cancel

footer

E. Franklin gets an alert after submitting the form that tells him an admin will review his posting first.

**Alert Page**

Logo

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

search

About

Post

Login

alert message

okay

footer

## 5. High-Level Architecture and Database Organization Summary

### A. Database Organization:

- a. Users Table:
  - i. idUsers: INT (Primary Key)
  - ii. email: VARCHAR(45)
  - iii. password: VARCHAR(45)
  - iv. firstname: VARCHAR(45)
  - v. lastname: VARCHAR(45)
- b. Items Table:
  - i. idItems: INT (Primary Key)
  - ii. category: INT (Foreign Key)
  - iii. seller: INT (Foreign Key)
  - iv. photopath: VARCHAR(45)
  - v. description: VARCHAR(128)
  - vi. title: VARCHAR(45)
  - vii. price: DOUBLE
  - viii. approved: TINYINT
- c. Categories Table:
  - i. idCategories: INT (Primary Key)
  - ii. title: VARCHAR(45)
- d. Reviews Table:
  - i. idReviews: INT (Primary Key)
  - ii. reviewer: INT (Foreign Key)
  - iii. reviewee: INT (Foreign Key)
  - iv. rating: TINYINT(5)
  - v. header: VARCHAR(45)
  - vi. body: VARCHAR(128)
- e. Messages Table:
  - i. idMessages: INT (Primary Key)
  - ii. item: INT (Foreign Key)
  - iii. sender: INT (Foreign Key)
  - iv. receiver: INT (Foreign Key)
  - v. body: VARCHAR(128)

### B. Media Storage:

Images are stored on the file system. The database will store pointers to the images, relative to the application directory. The webpages have a link to their corresponding images. The images will be in a designated folder. We will

downscale all images to save space. We do not have plans to support other forms of media (e.g. video/audio/GPS)

C. Search/filter architecture and implementation:

Search is implemented using the MySQL % Like operator. String comparison will be used to find corresponding items. Search items will be organized in FIFO order, so that old listings can be resolved. Item titles (and potentially descriptions) will be searched. Items can also be filtered using their assigned categories. Categories are selected from a drop down menu.

D. Own API(s): [if any]

None

E. Describe any significant non-trivial algorithm or process if any (like rating, ranking, automatic prioritizing of items etc.):

None

F. Changed SW Tools or Frameworks:

None

## 6. Key Risks

A. Skill Risks:

- Team needs to learn MVC Architecture.
- Unfamiliarity with AWS

**How we will address:**

- Team is learning and following the resources and tutorials of MVC and AWS.

B. Schedule Risks: N/A

C. Technical Risks:

- Remote hosting might be an issue temporarily as we have very few free hours left on our AWS account due to an extra instance kept running on the server.

**How we will address:**

- The issue shall be resolved from the month of April as we have shut down the extra instance.

D. Teamwork Risks:

- Sometimes, there is a communication gap between the frontend and backend teams.

**How we will address:**

- Team lead has proposed that the backend lead and the frontend lead schedule a weekly meeting where they each discuss the plans and progress of their respective teams.

E. Legal/Content Risks: N/A

## 7. Project Management

Team members coordinate with each other over Discord and Zoom meetings. The agenda, the work assigned to individual team members and the deadlines are all discussed in the meeting and posted on Discord to keep everyone updated about the same. There are several channels created on the Discord server group which makes it easier to access the particular information for everyone in the team. Once a team member is done with the individual assigned work, he/she updates it on discord to let everyone else know and then also help others if something is left to do before the deadline.

Except for the mandatory zoom meeting after the class, we have our own individual team meeting to discuss the plans for the next few days, assign deliverables and deadlines to all individuals and communicate about any doubts or difficulties. Apart from this general meeting, the backend and frontend teams schedule a short meeting of their own every week. The team lead distributes tasks amongst the frontend and backend team in the general meeting. The frontend and backend leads then distribute tasks to each individual of their respective team in their respective meeting. Also, if the work is interlinked between team members, they schedule internal team meetings on their own to discuss the same.

Team members make sure to complete the tasks assigned and upload the final edited version of it at least a day or two before the deadline so that the document master can just merge all of it, properly format it and submit it on/before time. Once finalized, it is then committed on Github.

As of now, we haven't set up Trello as communication has been pretty smooth over zoom and discord and we didn't want to make things complex. However, we plan to set up Trello after finishing up Milestone 2 as there would be a lot of coding moving forward and hence Trello will simplify communication and project management.