

SW Engineering CSC 648-848 Spring 2025

Project Name: GatorTrade

Team 08

Team Lead: Jonathan Carbajal - Jcarbajal@sfsu.edu

Backend Lead: Christian Kurt Balais - cbalais@sfsu.edu

Frontend Lead: Tung-Ying Lee - nlee@sfsu.edu

Frontend Lead 2: Chia-Tung Liu - cliu26@sfsu.edu

Github Master: Luis Moreno - lmoreno10@sfsu.edu

Milestone 2 Part 1

Date: 03/25/2025

History Table:

Version:	Date Submitted:	Date Revised:
M1	03-12-2025	03-21-2025
M2	03-25-2025	04-03-2025

Table of Contents

I. Executive Summary.....	2
II. List of Main Data Items and Entities.....	3
III. Functional Requirements.....	5
IV. UI Storyboards.....	8
V. High-Level Architecture.....	23
VI. Key Risks.....	26
VII. Project Management.....	26
VIII. Use of GenAI.....	27
IX. Team Lead Checklist.....	28

I. Executive Summary

While popular, platforms like Facebook Marketplace, Craigslist, OfferUp, and eBay have challenges that impact SFSU students daily. Buying and selling essentials like electronics, furniture, and textbooks can be frustrating due to scams, scalpers, and location-based limitations. Additionally, these platforms prioritize corporate interests, leading to issues like data privacy concerns, intrusive ads, and irrelevant listings. Motivated by these challenges, my team and I created GatorTrade, a dedicated, secure, and student-exclusive platform designed to ensure every buyer and seller is a member of the SFSU community.

Since GatorTrade is a secure marketplace where only SFSU students and faculty can participate, all users must have university email accounts. This eliminates the risks of anonymous buyers and sellers to ensure a safe and secure environment. While it is possible for individuals within the SFSU community to engage in misconduct, such as scams, harassment, or theft, our platform prioritizes security by adding an admin system that monitors and moderates listings and interactions to maintain a safe and trustworthy marketplace. To further streamline transactions, the platform includes additional features such as **integrated chat for direct communication**, suggested **on-campus meetup locations** for safe exchanges, **item tracking** to monitor availability and status, and **a profile rating system** that helps users build trust within the community. These tools not only make transactions more efficient but also reinforce the security and reliability of the marketplace. With its unique student-centric design, GatorTrade will focus on those students in need, unlike other marketplaces that cater to a broad audience. Different dedicated sections for textbooks, school supplies, and dorm essentials will be boosted to make navigating for these student materials easier.

Our team is dedicated to creating this website for our community with focus and passion. As senior-year computer science majors, we want to showcase the skills we have acquired throughout our studies and apply them to a project that makes a meaningful impact.

II. List of Main Data Items and Entities

1. User (Buyer/Seller/Admin)

Represents a registered individual on the platform.

- **User_id** – Unique identifier for the user (**mandatory**)
- **First_name** – User's first name (**mandatory**)
- **Last_name** – User's last name (**mandatory**)
- **Email** – User's unique email address (**mandatory**)
- **Password_hash** – Encrypted password for account login (**mandatory**)
- **Role** – The user's role being Admin or non-admin (e.g., Admin or Regular User) (**mandatory**)
- **Created_at** – Date the account was created (**mandatory**)

2. Listing

Represents an item posted by a seller.

- **Listing_id** – Unique identifier for the listing (**mandatory**)
- **user_id** (Seller) – Reference to the seller's user ID (**mandatory**)
- **Title** – Title of the item (**mandatory**)
- **Description** – Full description of the item (**mandatory**)
- **Price** – Price set by the seller (**mandatory**)
- **Category** – Category the item belongs to (e.g., Furniture, Textbooks) (**mandatory**)
- **image_url(s)** – One or more image links (**mandatory**)
- **Thumbnail** - Lower resolution of the item's first image (**mandatory**)
- **Status** – Listing status (e.g., Pending Approval, Active, Sold) (**mandatory**)
- **Created_at** – When the item was posted (**mandatory**)
- **Location** (proximity data or campus area) – Campus area or proximity-based info (**mandatory**)

3. Message

Represents a message sent between users.

- **Message_id** – Unique identifier for the message (**mandatory**)
- **Receiver_id** – Sender's user ID (**mandatory**)
- **Sender_id** – Receiver's user ID (**mandatory**)
- **Listing_id** – Reference to associated listing (**mandatory**)
- **Content** – Text of the message (**mandatory**)
- **Timestamp** – When the message was sent (**mandatory**)

4. Admin Review Log

Captures moderation actions performed by Admins on listings.

- **Admin_review_log_id** – Unique review log ID (**mandatory**)
- **Admin_id** – Admin who reviewed the listing (**mandatory**)
- **Listing_id** – Listing under review (**mandatory**)
- **Decision** – Outcome of the review (Approved / Rejected) (**mandatory**)

- **Moderator_Notes** – Notes added by the moderator (**optional**)
- **Reviewed_at** – When the review was completed (**mandatory**)

5. Review

Represents buyer feedback on a seller after a transaction.

- **Review_id** – Unique identifier for the review (**Mandatory**)
- **Reviewer_id** – User ID of the reviewer (Buyer) (**Mandatory**)
- **Reviewee_id** – User ID of the reviewee (Seller) (**Mandatory**)
- **Listing_id** – Related listing being reviewed (**Mandatory**)
- **Rating** – Numeric rating score (e.g., 1–5 stars) (**Mandatory**)
- **Comment** – Textual feedback from buyer (**Optional**)
- **Created_at** – When the review was submitted (**Mandatory**)

6. Favorite/Watchlist

Tracks listings saved or favorited by users.

- **Favorite_id** – Unique identifier for the saved item (**Mandatory**)
- **User_id** – Reference to the saving user's ID (**Mandatory**)
- **Listing_id** – Listing that was saved/favorited (**Mandatory**)
- **Timestamp** – Timestamp of when it was added to watchlist (**Mandatory**)

7. Reports

Used by users to report inappropriate or fraudulent listings.

- **Report_id** – Unique identifier for the report (**Mandatory**)
- **Reporter_id** – User ID of the person reporting (**Mandatory**)
- **Listing_id** – Listing being reported (**Mandatory**)
- **Report_type** – Type/category of report (e.g., Fraud, Inappropriate Content) (**Mandatory**)
- **Report_description** – User explanation of the issue (**Optional**)
- **Timestamp** – When the report was submitted (**Mandatory**)
- **Review_status** – Status of report review (Pending, Reviewed, Action Taken) (**Mandatory**)

8. System Announcement

Admin-generated announcements visible to users.

- **Announcement_id** – Unique identifier for the announcement (**Mandatory**)
- **Admin_id** – Admin user who posted the announcement (**Mandatory**)
- **Title** – Title of the announcement (**Mandatory**)
- **Content** – Full message content (**Mandatory**)
- **Timestamp** – When the announcement was posted (**Mandatory**)
- **Status** – Whether the announcement is still active (Active/Expired) (**Mandatory**)

9. Lost and Found

Platform section for posting lost or found items.

- **Post_id** – Unique identifier for the post (**Mandatory**)

- **Poster_id** – User who posted the lost/found item (**Mandatory**)
- **Title** – Brief title of the post (**Mandatory**)
- **Description** – Description of the lost/found item (**Mandatory**)
- **Category** – Whether the item is lost or found (Lost or Found) (**Mandatory**)
- **Location_description** – Description of where it was found/lost (**Mandatory**)
- **Contact_method** – How to contact the poster (e.g., email, phone) (**Mandatory**)
- **Image_url** – Photo of the item (**Optional**)
- **Timestamp** – When the item was posted (**Mandatory**)

10. Event Promotion

For promoting campus events or sales by students or orgs.

- **Event_id** – Unique identifier for the event post (**Mandatory**)
- **Poster_id** – User or organization who posted the event (**Mandatory**)
- **Title** – Title of the event (**Mandatory**)
- **Description** – Description and details of the event (**Mandatory**)
- **Event_date** – Date the event is happening (**Mandatory**)
- **Category** – Type of event (e.g., Graduation Sale) (**Mandatory**)
- **Image_url** – Image or flyer for the event (**Optional**)
- **Timestamp** – When the event post was created (**Mandatory**)

III. Functional Requirements

Priority 1 – Must Have

Unregistered Users

- **1 Browsing items** – Unregistered users shall view all listed items, including categories, prices, images, and descriptions.
- **2 Searching and filtering items** – Unregistered users shall be able to access keyword search, category filtering, and price range filtering.
- **3 Viewing item details** – Shall allow users to click on an item to view the full description, images, price, and time of posting.
- **4 User registration** – Unregistered users shall sign up using a valid email/university email.
- **15 University email verification only** – Users shall be required to register and participate in transactions using a valid university email.
- **21 User Login** - Users shall be able to log in and access features previously locked to unregistered users.

Registered Users

- **5 Posting items for sale** – Users shall be able to fill in item details, upload images, and submit items for admin approval before they go live.
- **6 Managing posted items** – Sellers shall be able to view, edit, or delete their listings from the “My Listings” section.
- **7 In-site messaging system** – Buyers shall be able to contact sellers using an in-platform messaging system, while sellers shall be able to view received messages from the dashboard.

Admin

- **11 Approving/rejecting items** – Admins shall review all newly posted items before they go live to ensure compliance with platform policies.

Priority 2 – Desired

Registered Users

- **8. Favoriting/tracking items** – Users shall be able to add items to the “Favorites” list for easy access later.
- **9. Managing user profile** – Users shall be able to update their information in the future, except for their email.
- **10. Reporting inappropriate items** – Users shall be able to report listings for violations such as fraudulent activity or inappropriate content.
- **13 Banning users** – Admins shall have the authority to suspend or permanently ban users who repeatedly violate marketplace policies.
- **20. On-campus meetup locations** – The platform shall suggest safe meetup locations on university campuses (e.g., library, student center).

Admin

- **12. Removing inappropriate listings** – Admins shall be able to delete listings that violate marketplace rules and notify the seller.
- **14. Managing system announcements** – Admins shall be able to post announcements about maintenance schedules or platform policy updates.

Priority 3 – Opportunistic

Registered Users

- **16. Miscellaneous Listing**– There shall be a dedicated category for random items that may not be school-related.
- **18. Verified review system for students** – Buyers and sellers shall be able to leave reviews only after completing a transaction, and these reviews shall affect the user's credibility.
- **19. Lost and found section for university students** – Users shall be able to post lost items or list found belongings to help reunite owners with their possessions.

Other Users (Clubs/Organizations)

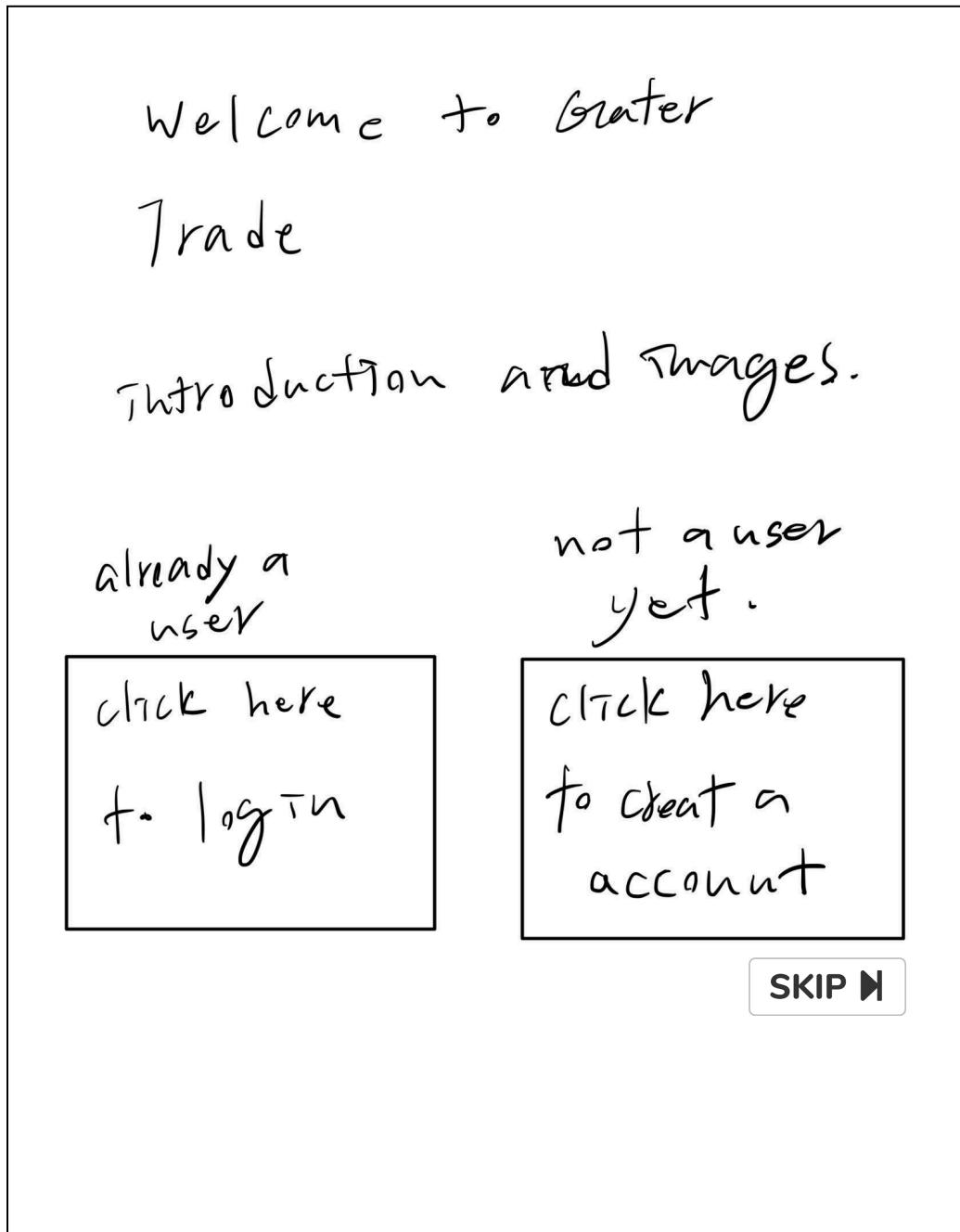
- **17. University event promotions** – The platform shall allow student clubs and organizations to post event-related listings (e.g., graduation furniture sales).

IV. UI Storyboards

Storyboard 1 - Use Case A: User Registration and Login

Emma registering with her SFSU email and logging in to access the website.

Screen 1: Home Page (Unregistered)



Emma goes onto the website and is presented with three options: logging in, registering, or skipping to view the site's listings. She decides to create an account.

Screen 2: Registration Page

(signup page)

Log in

Create an account

fill out information

First Name Last Name

Username

Email

School email only

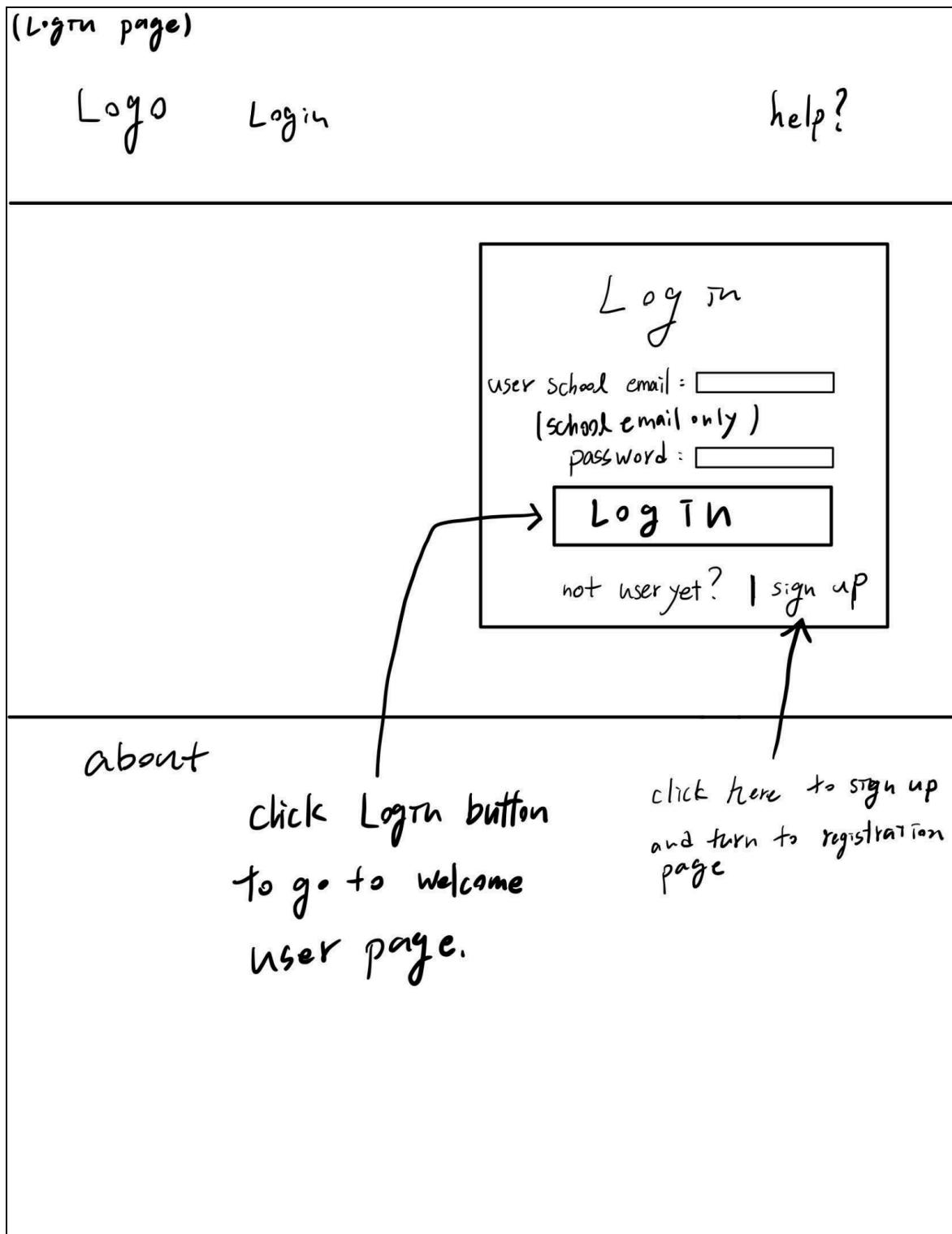
Password

Create account

click create account
and go back to login
page.

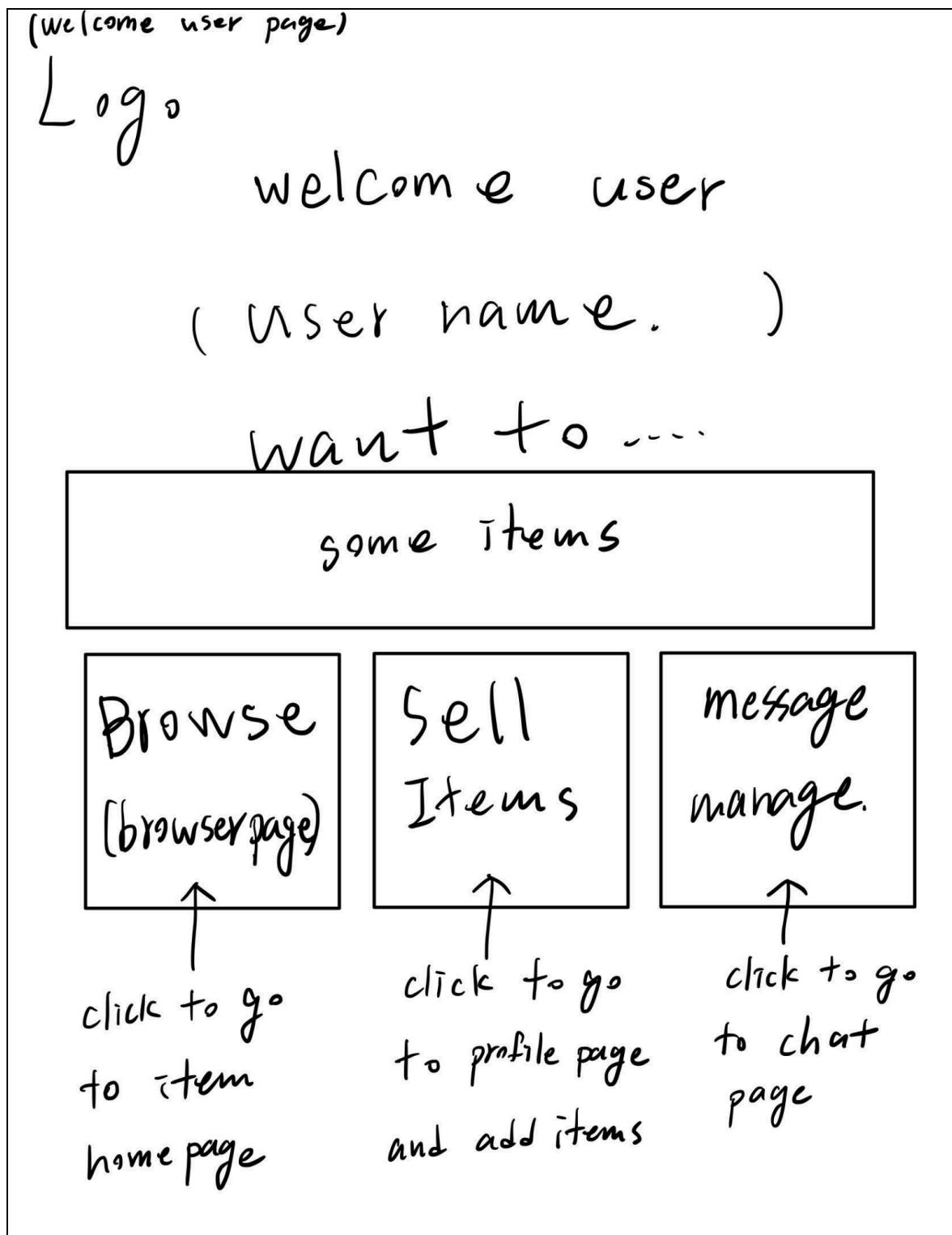
Emma is presented with fields for her name, username, SFSU email, and password. There is also a "Create Account" button and a note saying "School emails only" (e.g., @mail.sfsu.edu).

Screen 3: Login Page



Emma's account creation went through, allowing for the page to automatically redirect to the log-in page where she inserted her username and password to log in.

Screen 4: Home Page (Logged-in)

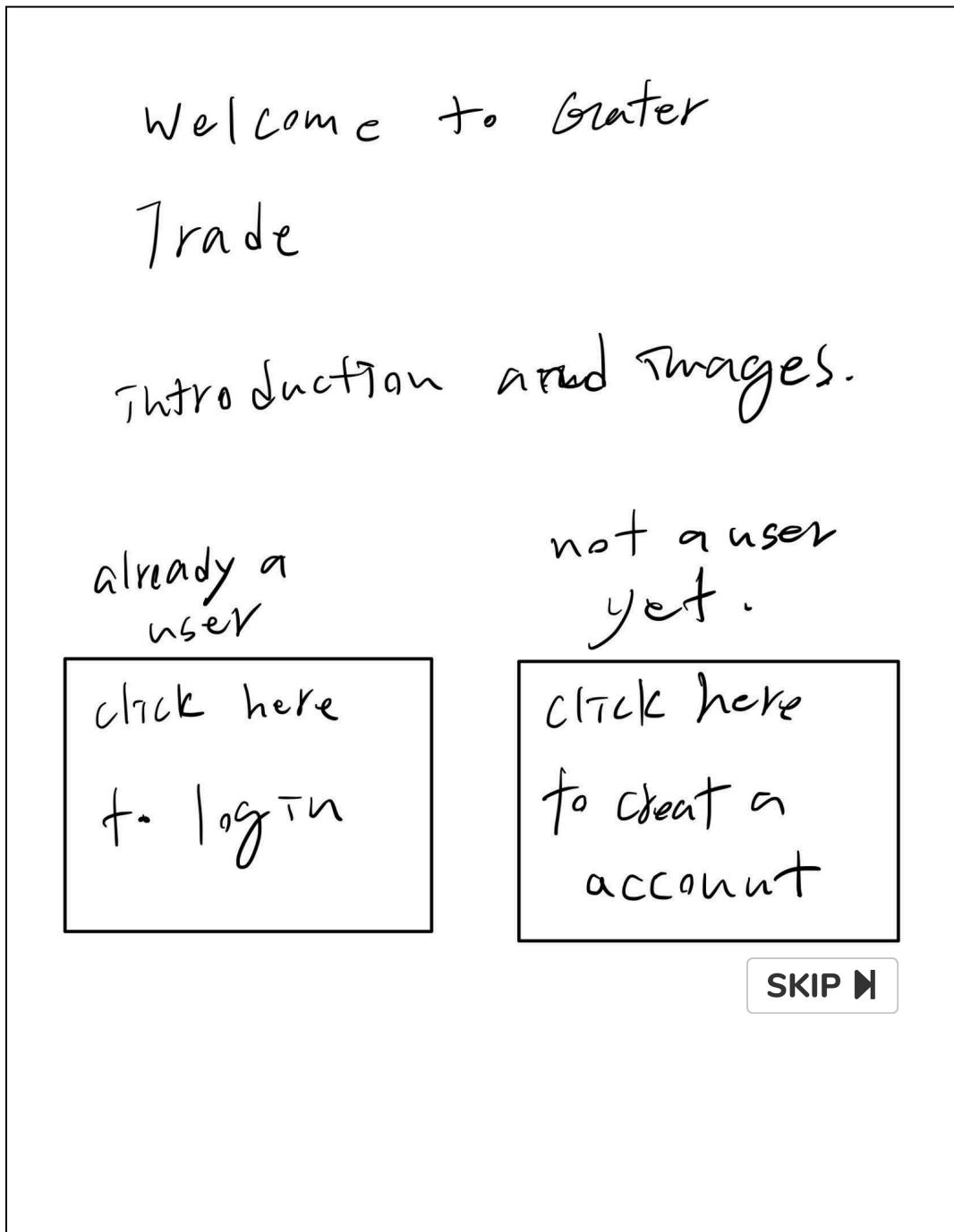


Upon successful login, Emma lands on the dashboard with key options such as "Browse Items," "Sell Now," and "Manage Messages". A featured item is at the top.

Storyboard 2 - Use Case B: Listing An Item

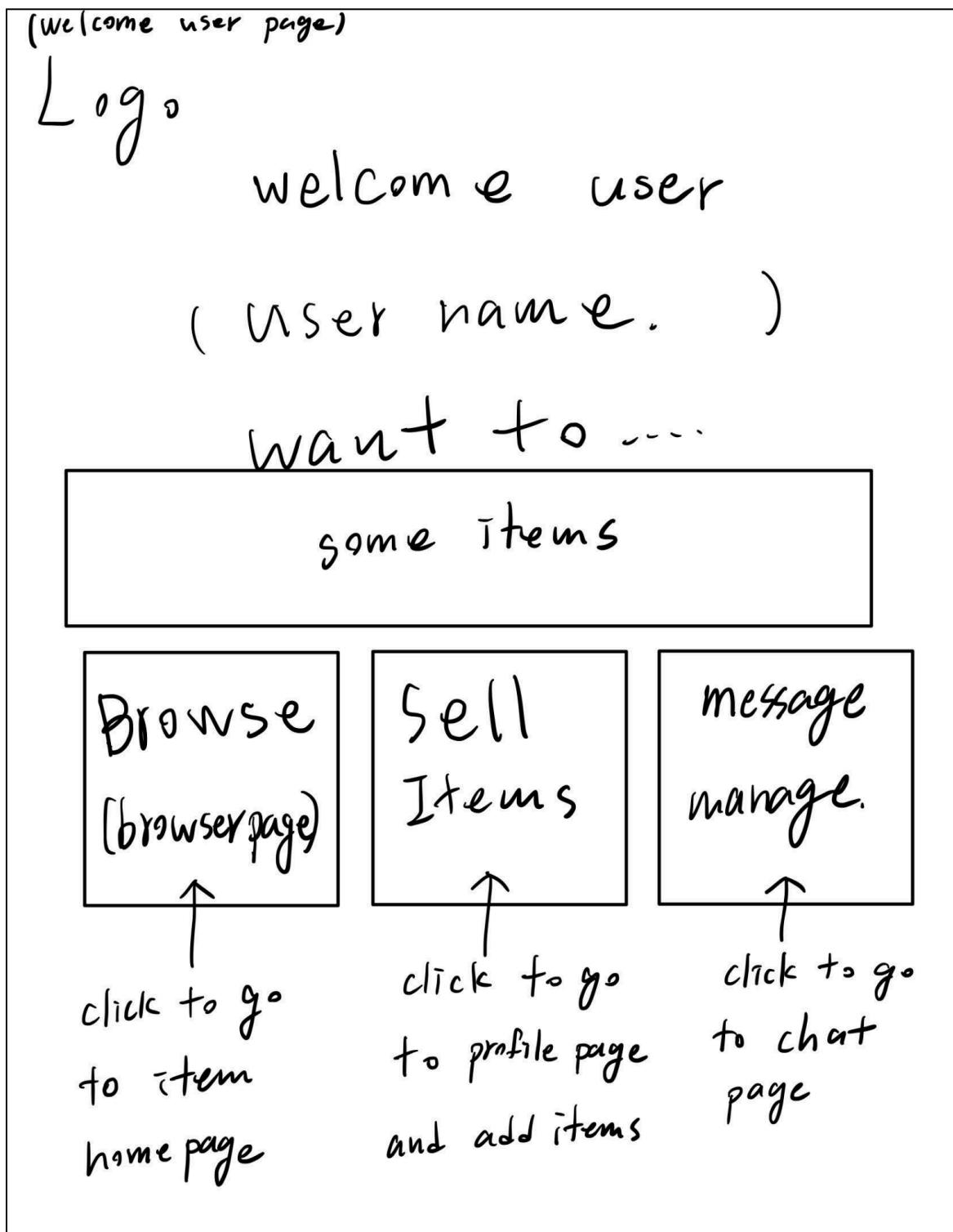
James is listing his gaming chair for sale.

Screen Page 1: Home Page (Unregistered)



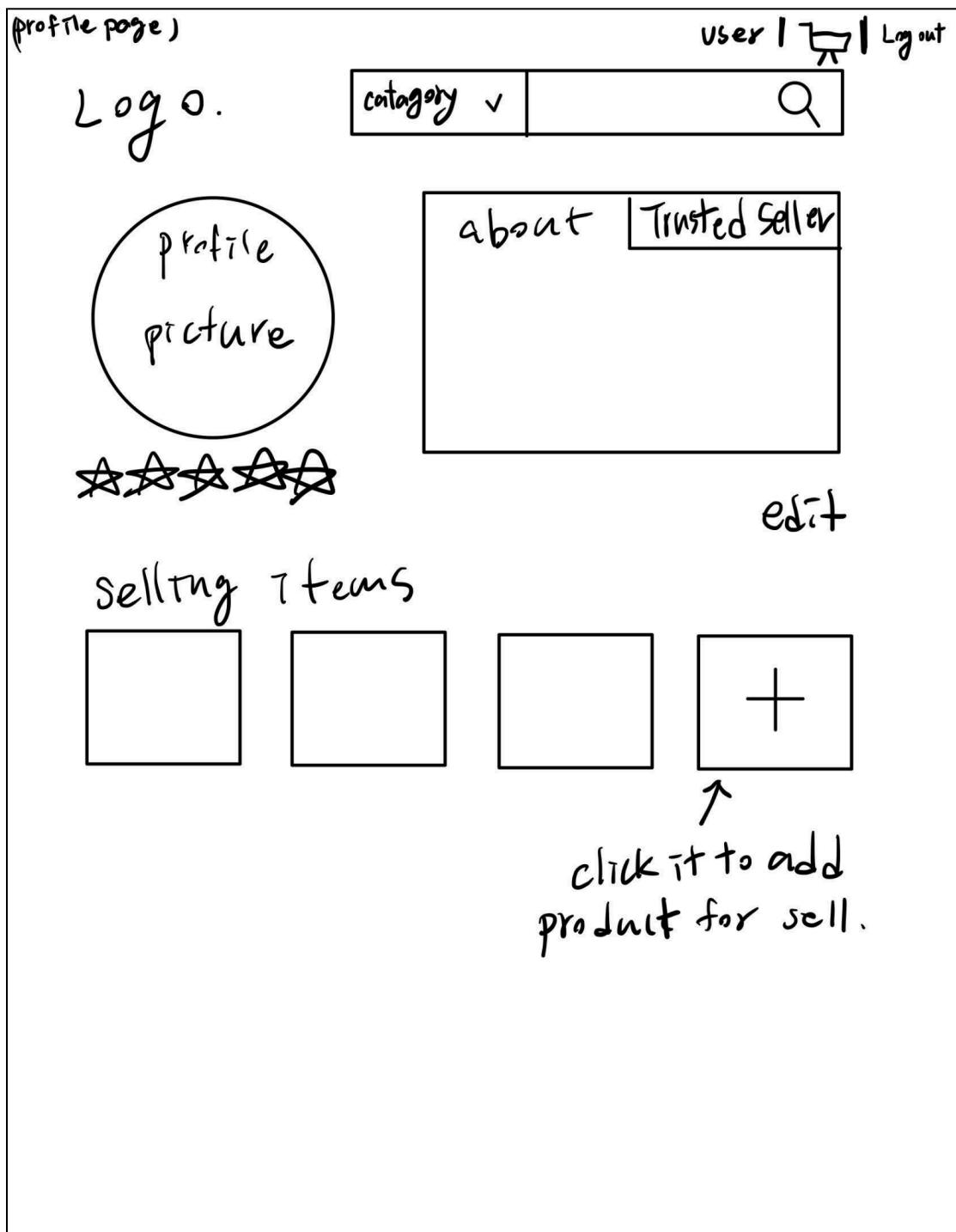
After a break from using the site, James decides to sell his gaming chair. Since his session had expired, he logs in with his existing account.

Screen Page 2: Home Page (Registered)



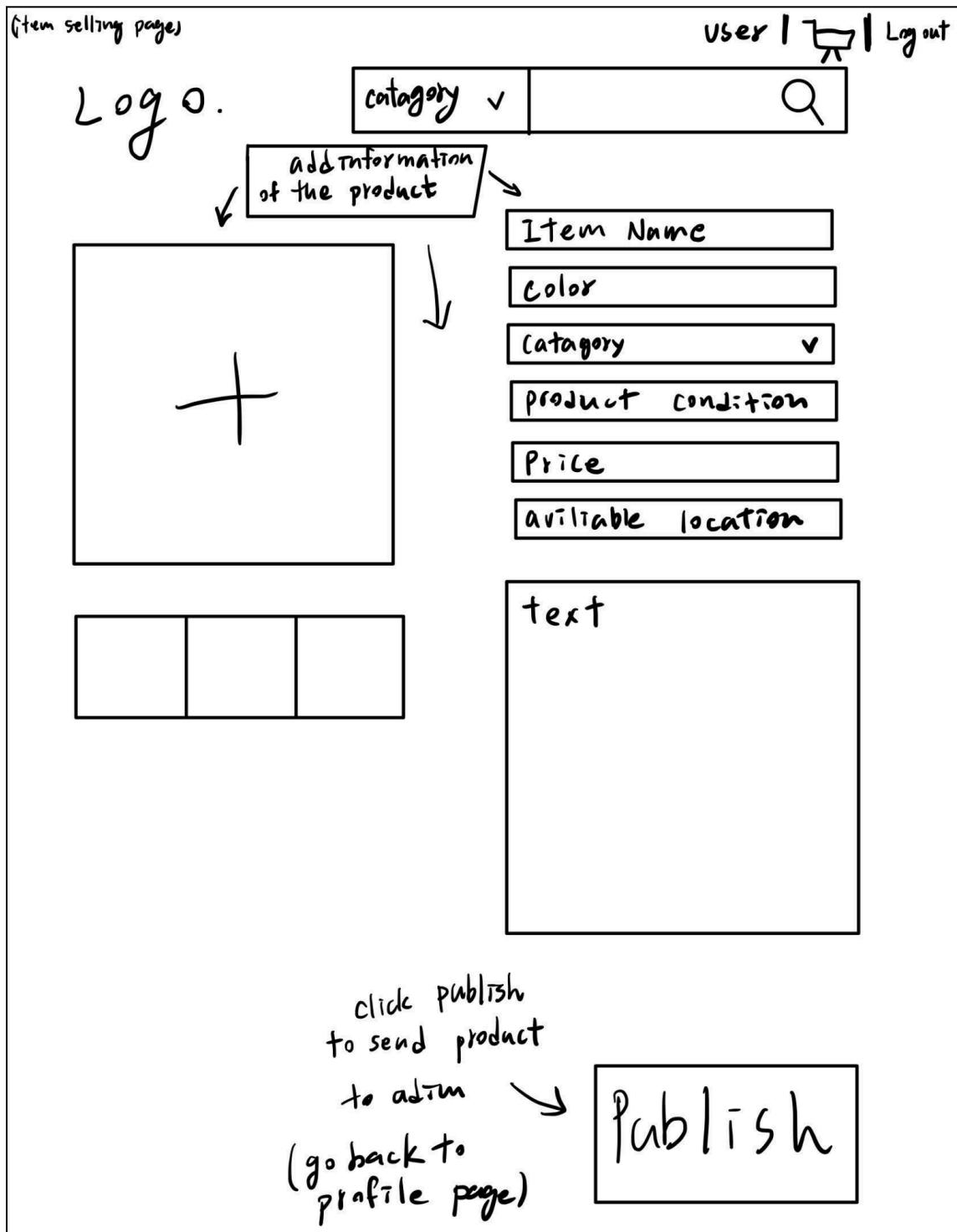
After logging in, James clicks on "Sell Items," which redirects him to his profile page where all his listings are displayed.

Screen Page 3: Profile Page



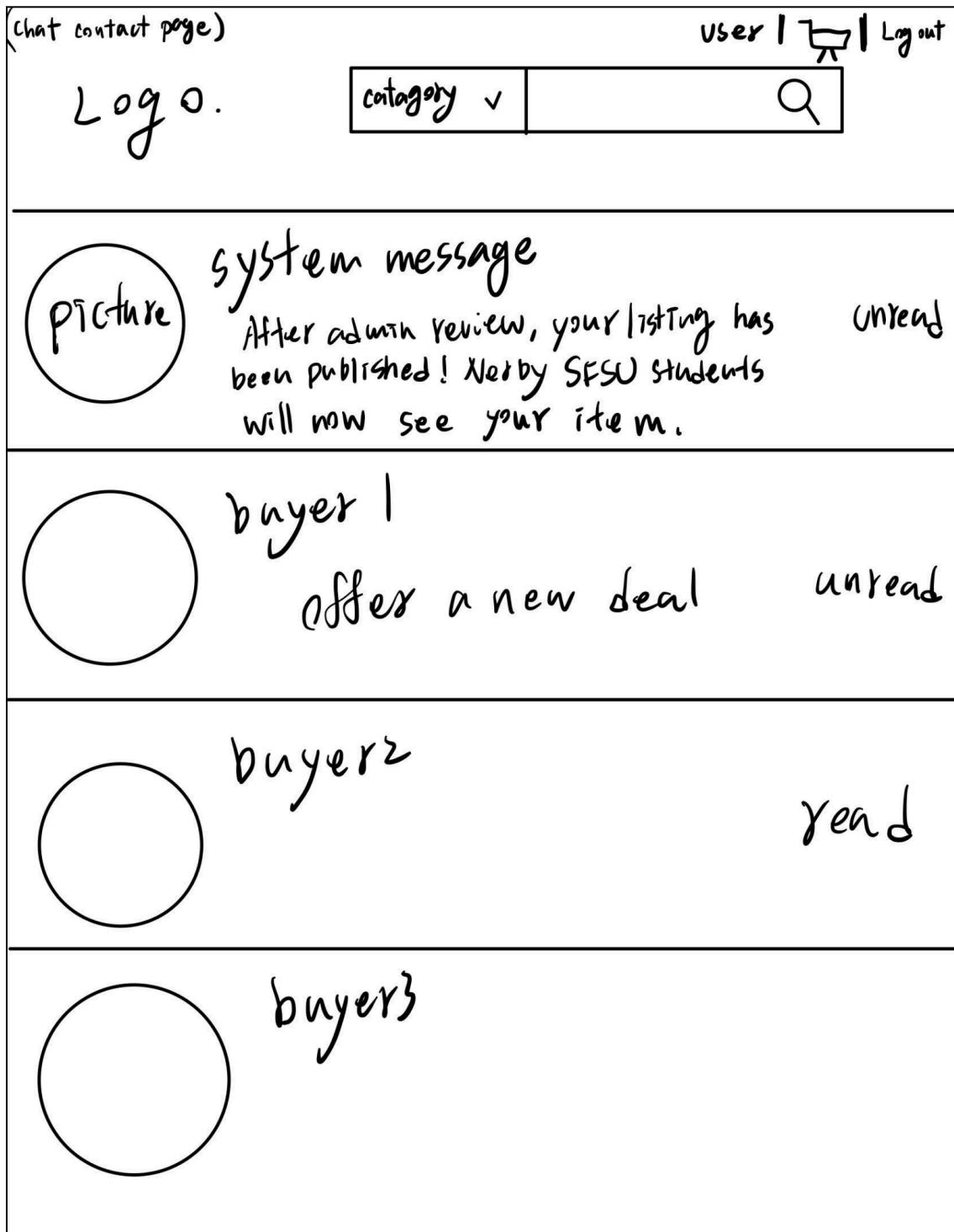
On his profile page, James sees his account details and current listings. He then clicks the "+" button to create a new listing for his gaming chair.

Screen Page 4: Listing Creation Page



On the listing creation page, James enters the details of his gaming chair, adding a title, description, price, category, and photos before selecting a nearby available location.

Screen Page 5: Message Hub Page

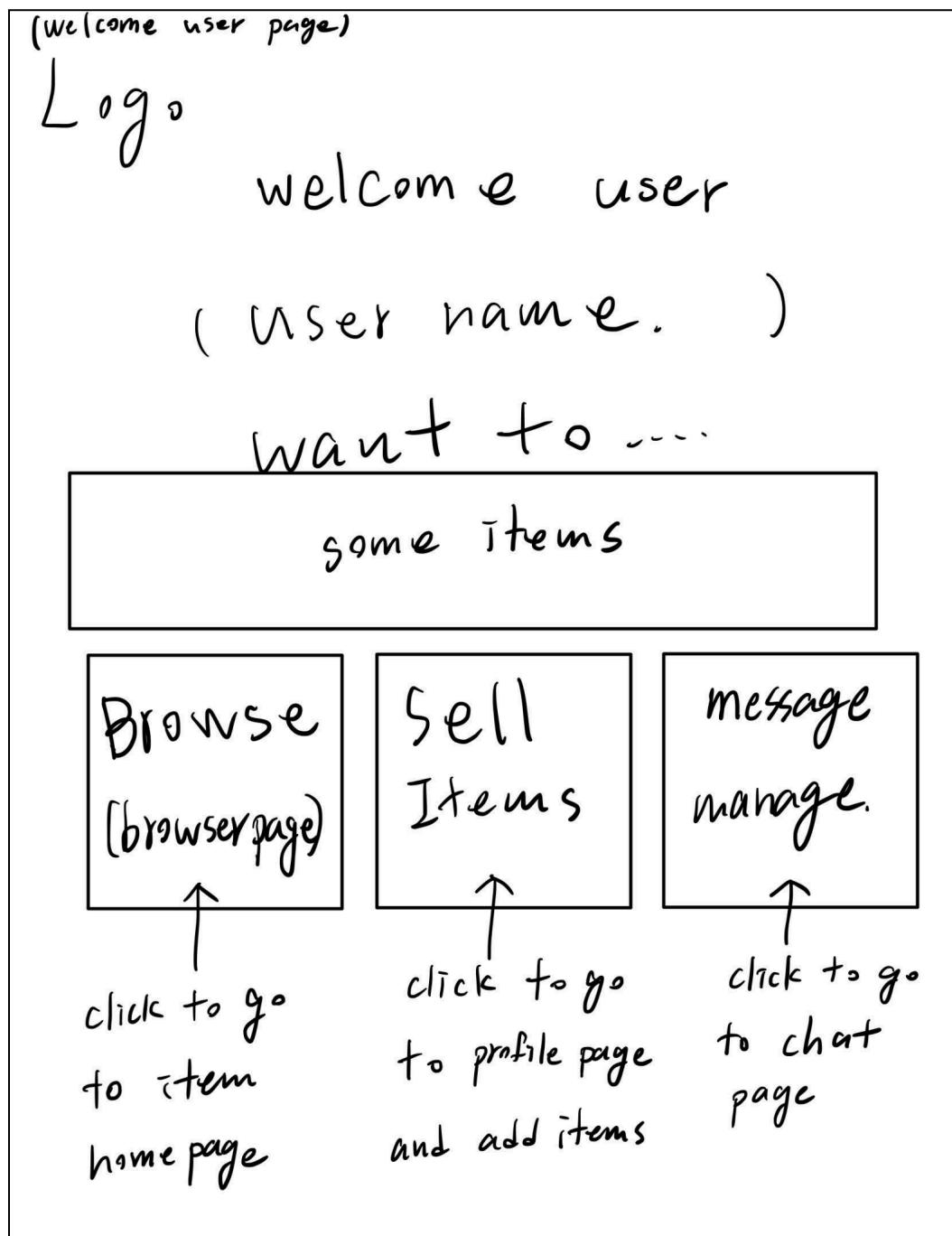


After publishing the listing, James waits for an admin to review it. Once approved, he receives an automatic message confirming that his item is now live on the site and visible to potential buyers.

Storyboard 3 - Use Case C: Browsing For Items

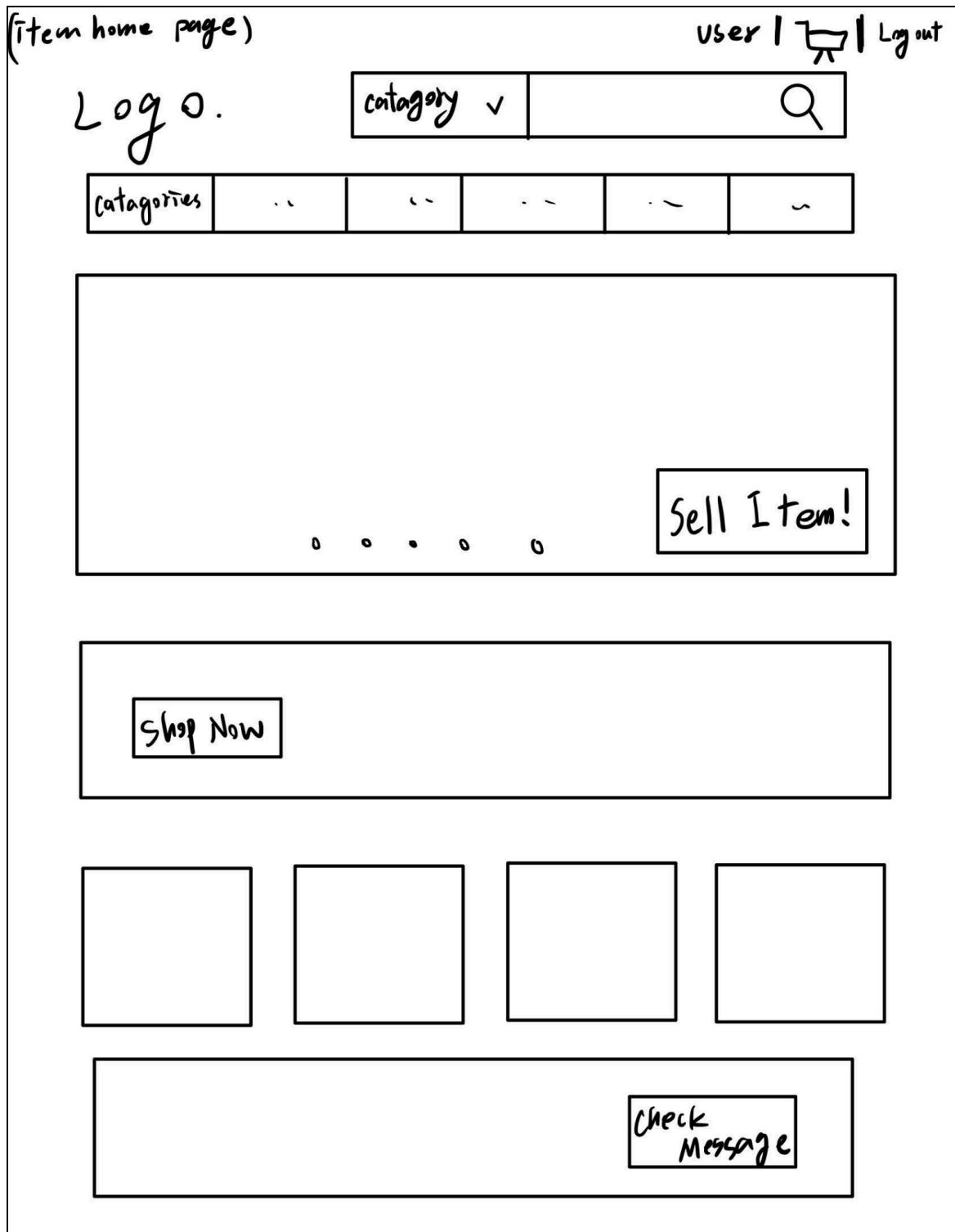
Emma uses the site to search for a used bike and messages a seller to learn more.

Screen 1: Home Page



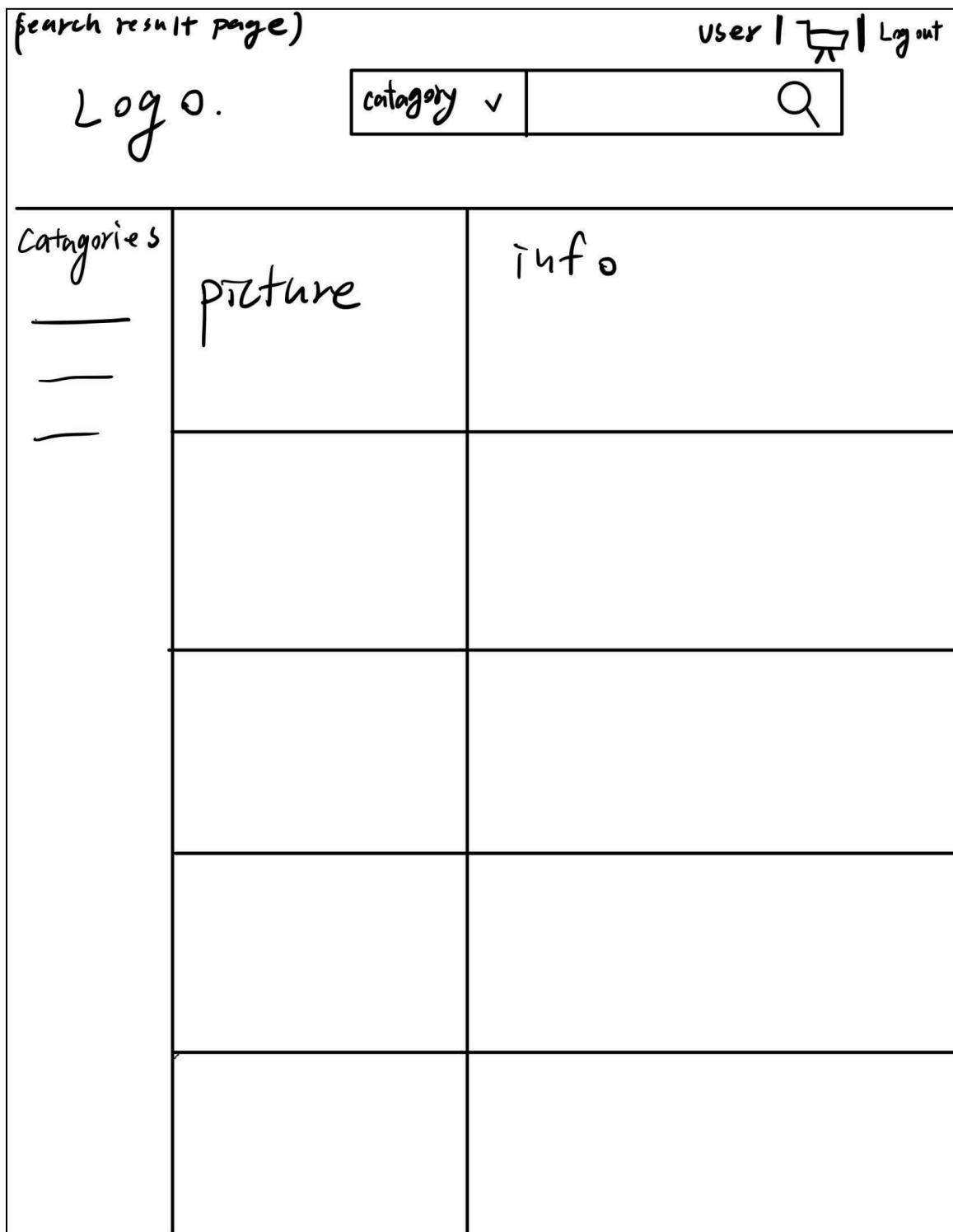
With Emma being a daily user, going onto the site automatically redirected her to the Home page and logged in through sessions. From here, she clicks Browse items.

Screen page 2: Browsing Page



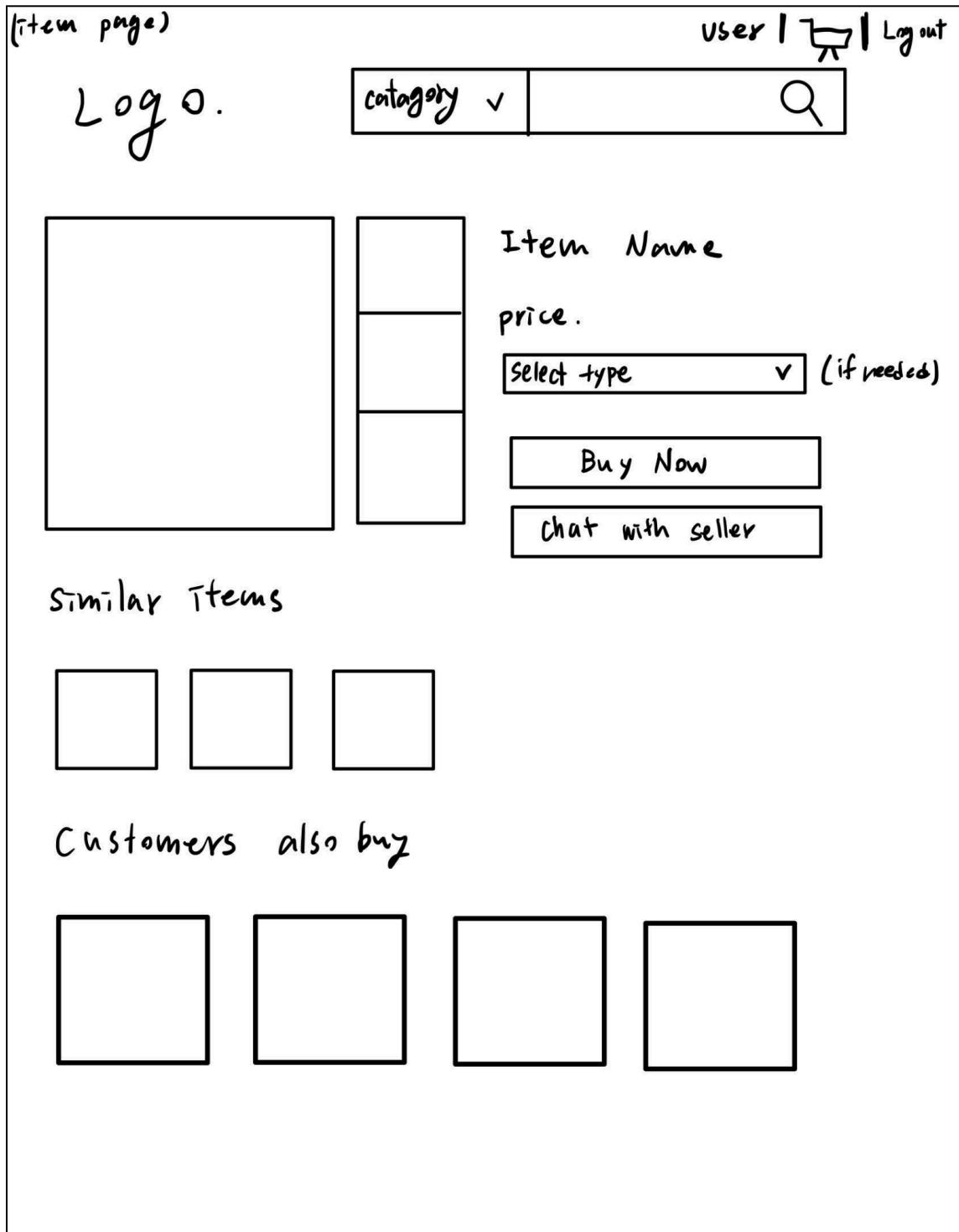
After being redirected to the browsing page, she used the search bar to type in "bikes" and then refined the results by selecting relevant categories.

Screen 3: Listing/Results Page



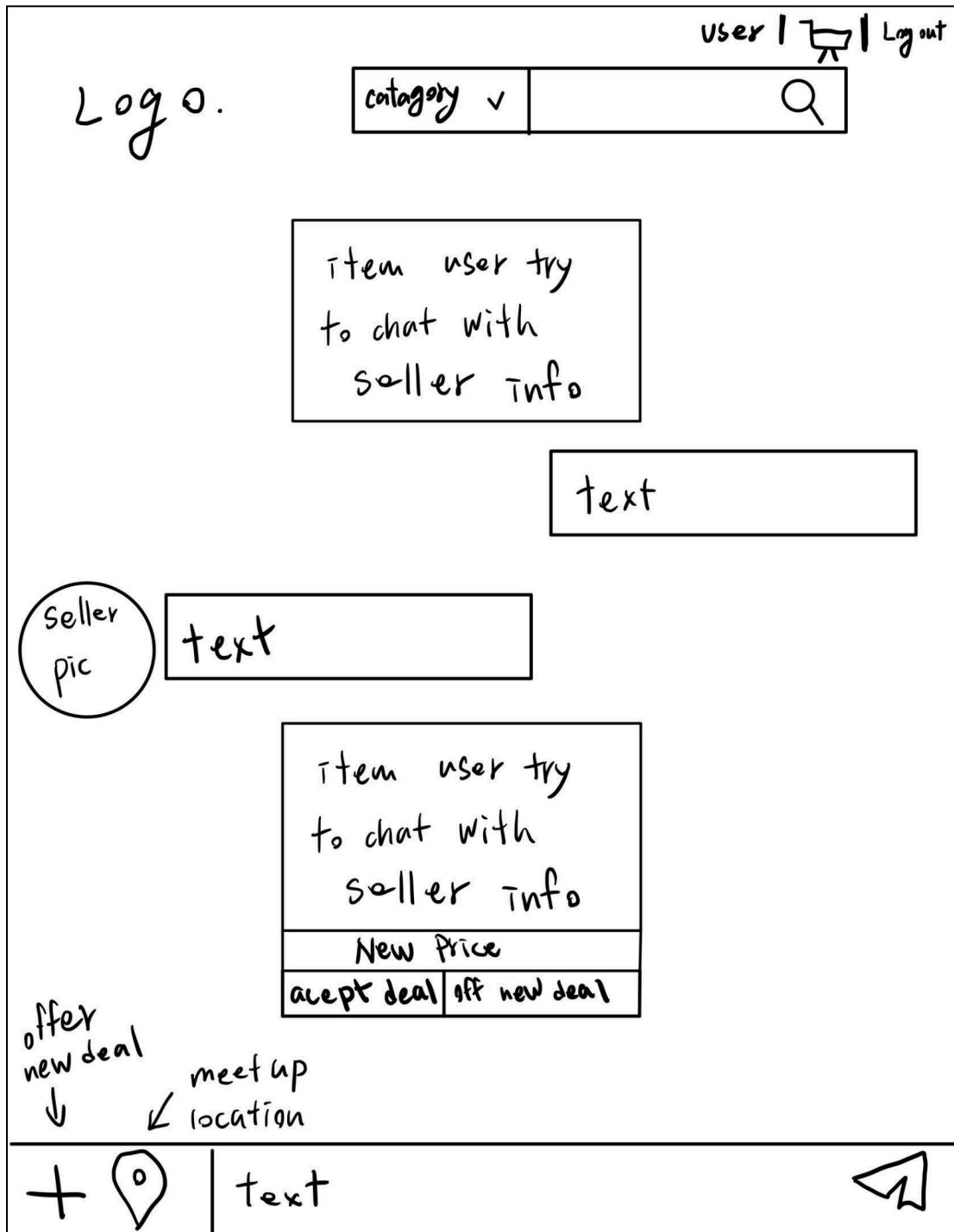
On the listings page, Emma browses several options and clicks on one that catches her eye to view more details.

Screen Page 4: Item Page



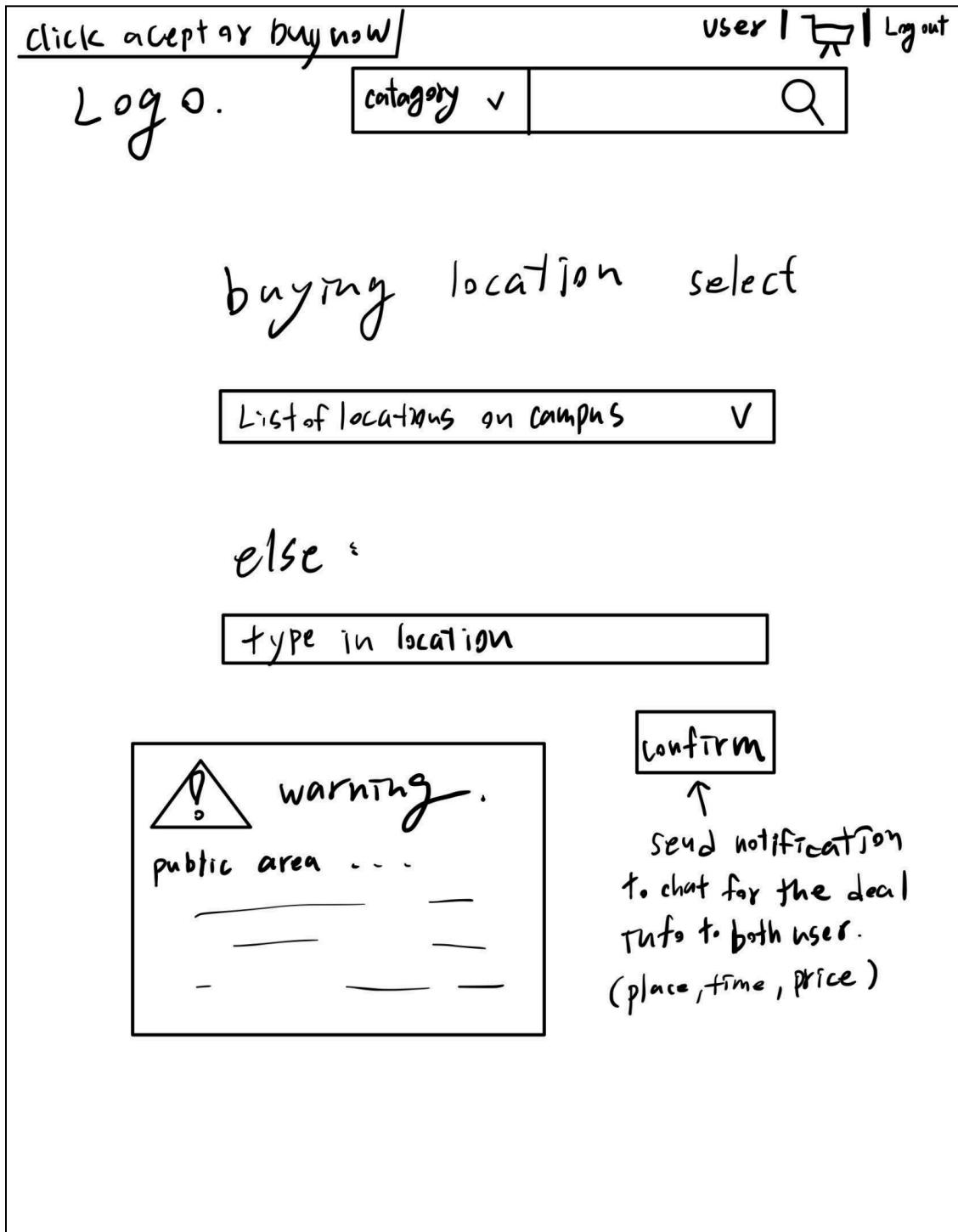
On the item page, Emma views detailed information and sees similar listings. To make a purchase, she clicks "Buy Now," which opens a chat with the seller.

Screen Page 5: Messaging Page



Emma messages the seller. Here, they can negotiate the deal and agree on a meeting spot near campus. If they agree on a deal, both can click "Accept" to proceed to the confirmation page.

Screen Page 6: Confirmation Page



The seller selects a meeting spot—limited to locations near campus, and once the buyer clicks "Confirm," the system sends a message with the item details, including price, location, and time.

V. High-Level Architecture

A. Database Organization

1. Users

- id (Primary Key)
- first_name
- last_name
- email
- password_hash
- role
- created_at

2. Listings

- id
- user_id (Foreign Key to users.id)
- title
- description
- price
- category
- Image_url
- thumbnails
- status
- created_at

3. Messages

- id
- sender_id (Foreign Key to users.id)
- receiver_id (Foreign Key to users.id)
- listing_id
- content
- timestamp

4. Admin_Review_Logs

- id
- admin_id (Foreign Key to users.id)
- listing_id (Foreign Key to listings.id)
- decision (ENUM: ‘approved’, ‘rejected’, ‘flagged’)
- moderator_notes
- reviewed_at

5. Review

- review_id
- reviewer_id
- reviewee_id
- listing_id

- rating
 - comment
 - created_at
6. Favorite
- favorite_id
 - user_id
 - listing_id
 - timestamp
7. Reports
- report_id
 - reporter_id
 - listing_id
 - report_type
 - report_description
 - timestamp
 - review_status
8. System_Announcement
- announcement_id
 - admin_id
 - title
 - content
 - timestamp
 - status
9. Lost_and_Found
- post_id
 - poster_id
 - title
 - description
 - category
 - location_description
 - contact_method
 - image_url
 - timestamp
10. Event_Promotion
- event_id
 - poster_id
 - title
 - description
 - event_date
 - category

- image_url
- timestamp

B. Media Storage Strategy

1. Image storage will be handled via file system or AWS S3
2. Only the URL/path to the image is stored in the image_url field in the Listings table.
3. The team is not using BLOBS for media, keeping the database lightweight and performant.
4. Currently, we do not plan to store or handle video data due to limitations with free AWS

C. Search & Filter Architecture

1. We will implement search functionality using SQL Queries with the LIKE operator

- Searchable Fields:
 - (1) title
 - (2) description
 - (3) Category
- Search Logic

```
SELECT * FROM listings
WHERE title LIKE '%keyword%'
    OR description LIKE '%keyword%'
    OR category LIKE '%keyword%';
```

- Filtering and Sorting
 - (1) Handled via API query parameters

```
GET /listings?category=books&min_price=10&max_price=50&sort=newest
```

- (2) Filtering is done using WHERE clauses, and sorting via ORDER BY in the SQL query

D. Special Logic or Non-Trivial Algorithms

1. We will be implementing ranking and rating algorithms.

VI. Key Risks

Skill Risk:

Challenges relating to skills are one of the primary risks for the GatorTrade project. Certain team members may lack full-stack deployment skills, which include front-end deployment, database administration, and backend hosting. Additionally, people who are not familiar with these technologies may find it challenging to set up secure authentication, such as OAuth and SFSU email verification. We will solve this through cooperation between the team. More experienced group members will help the less experienced team members, and backend developers will investigate authentication alternatives and prototype implementations beforehand.

Teamwork risk:

Teamwork risks include potential communication breakdowns between front-end and backend teams, which could lead to integration issues, as well as unclear task distribution, which could result in redundant work or missing features. To avoid these issues, we will keep extensive API documentation using tools like Swagger or Postman and hold weekly team sync meetings via Slack or Discord. Team members will be given specific tasks and responsibilities to guarantee accountability and efficiency.

Technical risk:

Technical risks are database performance and security vulnerabilities. We can mitigate these risks by first indexing and caching correctly to ensure the utmost speed in the database. To prevent SQL injection, we will carefully add input validation and careful parameters to prevent foreign database changes.

Legal Risk:

Some Legal risks associated with managing user data, such as emails and messages, which need strict privacy protection, and with adhering to SFSU standards. We will examine SFSU's policies on student-based marketplaces and put strong encryption.

VII. Project Management

I plan to approach milestone 2 the same way I've handled all our previous assignments: by breaking the project into smaller, manageable tasks and assigning each team member a clear plan with a due date. Since we're all seniors, I have confidence in my teammates to meet their deadlines, and so far, they've consistently delivered.

For milestone 2 specifically, the back-end and front-end teams will each focus on their respective responsibilities, which will eventually integrate as we progress toward project

completion. The back-end team will handle the database and develop the JavaScript search function, while the front-end team will focus on designing and building the web pages.

As the team lead, I act as the communication bridge between both teams, ensuring that everything runs smoothly and that we stay on schedule. In every meeting, I assign tasks, document them in our Discord server, and provide short but reasonable deadlines. This gives everyone a clear reference point if they need to revisit their responsibilities.

Once all the tasks are completed, either I or another teammate will thoroughly review and edit the document to correct any formatting or grammatical issues. Once we're confident in the final product, we submit it.

VIII. Use of GenAI

Brainstorming Ideas:

ChatGPT was used extensively during the early planning phase to brainstorm potential project risks, feature ideas, and UI/UX flow suggestions. It helped propose several use-case scenarios and alternative flows that shaped the initial direction of the app. Additionally, it offered input for early-stage storyboard suggestions, such as how users would navigate between listing pages, messaging, and admin moderation.

Coding:

ChatGPT provided support in debugging and troubleshooting frontend and backend code. While it currently isn't used as much right now for milestone 2 part 1, we predict the use of it will go up as we program more in part 2.

Writing and structure:

ChatGPT was used to improve the clarity, grammar, and professional tone of milestone documents, presentation slides, and report content. It assisted in rewording long or unclear sentences, ensuring smoother flow between sections, and standardizing terminology across documents.

List of Main Data Items & Functional Requirements:

ChatGPT assisted in organizing and naming core data entities such as *User*, *Listing*, *Message*, and *Review*. It helped in defining the structure of each data item, including value examples and required/optional status.

Rating (out of 5):

- **Brainstorming Ideas** – 2 (Low)
- **Code Assistance** – 2 (Low)
- **Main data items and functional requirements** – 4 (Medium)
- **Writing & Formatting Help** – 5 (High)

Additional Information:

- ChatGPT Used: 3.5/4.0
- Pretty good to expand upon ideas already written down.

Example Prompts:

- “Can you format this so it follows what the question is asking for?”
- “Simplify the formatting.”
- “Should I add more things to priorities 2 and 3?”

IX. Team Lead Checklist

- a. So far, all team members are fully engaged and attending team sessions when required - DONE
- b. 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
- c. Team reviewed suggested resources before drafting Milestone 2 - DONE
- d. Team lead checked Milestone 2 document for quality, completeness, formatting and compliance with instructions before the submission - DONE
- e. Team lead ensured that all team members read the final Milestone 2 document and agree/understand it before submission - DONE
- f. Team shared and discussed experience with GenAI tools among themselves - DONE