

Final Project for SW Engineering Class

CSC648-848 Spring 2025

Team 15

ThriftAtSFSU

<https://linux-vm-1.tailce85ff.ts.net/>

Hilary Lui (hlui@sfsu.edu)	Team Lead
Annison Van	Frontend Lead
Sid Padmanabhuni	Backend Lead & GitHub master
Joseph Alhambra	Team Member Frontend
Joseph Shur	Team Member Backend

Date Submitted	5/19/2025
Date Revised	5/20/2025

1. Product summary:

Product Name: ThriftAtSFSU

What is ThriftAtSFSU?

ThriftAtSFSU is a student-built, campus-exclusive online marketplace designed specifically for SFSU students and faculty. It allows users to buy, sell, and trade items within the university community—no shipping, no strangers, and no scams. By keeping all transactions local and account-verified, the platform promotes safety, trust, and convenience for students looking to declutter, save money, or find secondhand deals.

What makes it unique and SFSU-specific?

ThriftAtSFSU is only accessible to users with an @sfsu.edu email address. It's tailor-made for the SFSU campus community, with features like local pickup, student-to-student messaging, and listings exclusive to classmates and faculty. It's a marketplace built by SFSU, for SFSU.

Final List of Committed Priority 1 (P1) Functions:

- Users can create an account and log in securely
- Users can browse all available listings on the homepage
- Users can post a new item for sale with a title, price, category, description, and photo
- Users can view detailed pages for each listing
- Users can delete or edit their own listings
- Users can view their posted listings in a dashboard
- Users can search for items by keywords
- Users can filter listings by item category
- Users can add items to a wishlist (favorites)
- Users can send and receive messages about a listing

Product URL:

<https://linux-vm-1.tailce85ff.ts.net/>

2. Milestone documents – M1-M4

SW Engineering CSC648-848 Spring 2025

ThriftAtSFSU

Milestone 1

Team 15

Hilary Lui (hlui@sfsu.edu)	Team Lead
Annison Van	Frontend Lead
Sid Padmanabhuni	Backend Lead & Github master
Joseph Alhambra	Team Member Frontend
Joseph Shur	Team Member Backend

Date Submitted	3/12/2025
Date Revised	3/25/2025

1. Executive Summary

In today's fast-paced college environment, affordability and convenience are key student concerns. ThriftAtSFSU is a dedicated resale platform designed exclusively for San Francisco State University (SFSU) students, providing a secure and efficient way to buy and sell second-hand goods within the campus community. Unlike generic resale platforms, ThriftAtSFSU is tailored to meet the specific needs of SFSU students, ensuring a trusted marketplace that fosters sustainability and affordability.

To create a seamless user experience, ThriftAtSFSU offers a range of features designed with students in mind. Users can easily browse and search for items using advanced filters for academic-specific needs, such as textbooks and dorm supplies. A secure in-site messaging system ensures safe communication between buyers and sellers, requiring SFSU email verification. Listing items is streamlined for students, featuring predefined academic categories for quicker uploads. Sellers benefit from a listing-based chatting system to keep track of their items and also the ability to put their schedule on their profile to make coordinating pick-up times easier. To maintain a safe and trustworthy marketplace, the platform includes strict item and user moderation managed by administrators. Additionally, ThriftAtSFSU is fully mobile-responsive, allowing students to access the marketplace conveniently from any device.

What sets ThriftAtSFSU apart is its exclusive focus on the SFSU community, ensuring a localized and student-friendly marketplace that prioritizes security, convenience, and affordability. This project directly addresses the need for a reliable, campus-specific resale network, fostering a culture of sustainability while making essential goods more accessible to students. Our team comprises passionate SFSU students committed to enhancing campus life through innovative solutions. With firsthand experience of students' challenges in accessing affordable goods, we are dedicated to making ThriftAtSFSU the go-to platform for peer-to-peer resale within our university. By supporting this project, you invest in a practical and impactful solution that benefits the entire SFSU community.

2. Personae

Persona: Dr. James Carter

Attitude:

Dr. Carter is a seasoned professor in the English department at SFSU. He prefers convenience when shopping online and values clear descriptions and easy navigation. He isn't tech-averse but prefers platforms that are simple and efficient. He appreciates sustainability and enjoys the idea of second-hand shopping to reduce waste.

Skills:

- Proficient in using computers but not a fan of overly complex or cluttered interfaces.
- Familiar with online shopping platforms like eBay and Amazon but not a frequent user of online marketplaces.
- Prefers using search filters to find what he needs quickly.

Limitations:

- Has a busy schedule, making it difficult to spend time browsing.
- Not interested in negotiating or haggling, prefers clear pricing.
- May struggle with messaging sellers if responses are delayed.

Pain Points:

- Finding clothes in his size can be difficult in a marketplace setting.
- Unreliable or unclear product descriptions can be frustrating.
- Lack of clear filtering options could make searching inefficient.

Goals:

- Quickly find professional attire in his size without sifting through irrelevant listings.
- Avoid lengthy negotiations and unclear pricing.
- Ensure that sellers are legitimate and trustworthy.

Persona: Maria Sanchez

Attitude:

Maria is a dedicated student and a mother of a 3-year-old. She is cautious when buying online and prioritizes safety, reliability, and affordability. She prefers to shop from trusted sellers and checks reviews before making a purchase. Convenience is essential for her, as she balances schoolwork and parenting.

Skills:

- Comfortable using technology but prefers mobile-friendly and easy-to-navigate websites.
- Experienced in online shopping, familiar with platforms like Facebook Marketplace and Craigslist.
- Relies on reviews and ratings to assess seller credibility.

Limitations:

- Limited time to browse listings due to her busy schedule.
- Prefers in-person meet-ups that fit within her class schedule.
- Avoids high-priced listings due to budget constraints.

Pain Points:

- Concerns about scams or unreliable sellers.
- Difficulty finding listings that include detailed descriptions and good photos.
- Hard to schedule meet-ups that align with her limited free time.

Goals:

- Find affordable school supplies, children's items, and household essentials.
- Ensure that transactions are safe and efficient.
- Minimize the time spent searching for trustworthy sellers.

Persona: Alex Nguyen

Attitude:

Alex is a first-year student living in an SFSU dorm. He is excited about his new independence but also mindful of his budget. He prefers to buy second-hand items to save money and make his dorm feel more like home. He is comfortable with technology and prefers quick, efficient transactions.

Skills:

- Tech-savvy and familiar with online marketplaces like Facebook Marketplace and OfferUp.
- Comfortable with messaging sellers and negotiating prices.
- Prefers a simple and intuitive platform for easy browsing.

Limitations:

- Limited space in his dorm, so he has to be selective with purchases.
- No car, so he needs sellers to meet on campus or nearby.
- Relies on financial aid, so affordability is key.

Pain Points:

- Finding affordable dorm essentials that are in good condition.
- Coordinating meet-ups without reliable transportation.
- Dealing with listings that lack detailed information or clear photos.

Goals:

- Buy second-hand dorm essentials like a mini-fridge, microwave, or desk chair at a low cost.
- Ensure sellers are trustworthy and products are as described.
- Arrange meet-ups that work with his class schedule and lack of transportation.

Persona: Jasmine Lee**Attitude:**

Jasmine is a junior at SFSU who commutes daily from Daly City. She prefers efficient and straightforward transactions because she has limited time on campus. She values clear communication with sellers and quick responses. She enjoys finding good deals but dislikes last-minute changes or unreliable sellers.

Skills:

- Proficient in using online marketplaces like eBay and Facebook Marketplace.
- Organized and plans her meet-ups ahead of time.
- Comfortable with digital payments and messaging apps for quick coordination.

Limitations:

- Limited time on campus, so she needs to arrange meet-ups that fit her schedule.
- Prefers not to carry large or heavy items on public transportation.
- Avoids complicated negotiations due to time constraints.

Pain Points:

- Sellers who don't respond quickly can make transactions difficult.
- Unclear listings that lack information or proper photos.
- Arranging meet-ups that align with her class and commute schedule.

Goals:

- Find good deals on electronics, books, and commuter-friendly items.
- Ensure smooth and reliable transactions with sellers.
- Minimize time spent coordinating purchases.

3. High-level Use cases

Use Case: Selling Items

Dr. Carter needs a new blazer for an upcoming conference but prefers to shop second-hand to support sustainability. He logs into the marketplace and uses the size filter to avoid sifting through irrelevant listings. Within minutes, he finds a well-maintained blazer in his size and at an affordable price. Since he dislikes lengthy negotiations, he appreciates that the price is clearly stated. He messages the seller to confirm availability and arranges a pickup on campus, making his shopping experience quick, efficient, and stress-free.

Use Case: Checking Reviews

Maria needs to buy a used stroller for her three-year-old but is cautious about online purchases. Before committing to a meet-up, she carefully checks the seller's profile and reads past reviews from other buyers. Seeing that the seller has multiple positive reviews for being reliable and honest gives her confidence in the purchase. She messages the seller, confirms the stroller's condition with additional photos, and arranges a safe, public meet-up near campus. Thanks to the review system, she feels reassured that she is dealing with a trustworthy seller.

Use Case: Needs to buy items

Alex has just moved into his dorm at SFSU and quickly realizes he needs a mini-fridge, a desk lamp, and some storage bins. Rather than spending a fortune on new items, he visits the market site for second-hand deals from students who are moving out. Using the search filters, he quickly finds listings that match his needs and budget. He messages a senior who is selling a mini-fridge at half the retail price and arranges a meet-up on campus. Within a day, he has everything he needs for his dorm without breaking the bank.

Use Case: Communicate/Plan with seller

Jasmine commutes from Daly City and has limited time between classes. She finds a great deal on a used laptop from a student seller but needs to make sure she can pick it up at a convenient time. Using the messaging feature, she coordinates a meet-up during her lunch break and confirms the seller's availability before making the trip. The quick response from the seller and the ability to arrange a hassle-free pickup help her complete the transaction without disrupting her busy schedule.

4. List of main data items and entities – data glossary/description

Entities

Unregistered User: Unregistered users are the first role anyone on the app has access to, and have limited access to most features. Unregistered users can view listings and reviews, but can't access their own profile page, chat logs with other users, or buy and sell items that are listed.

Registered User: Registered users are verified accounts that can create, delete, and edit listings, add their own photos, leave reviews under other listings, and have their own profile where they can share more about themselves and what they are looking to buy or sell on the app. Registered users can edit their own profile, share chat logs with other users, and buy and sell items - but can only edit and make changes to their own created photos, listings, reviews, and profile.

Admin: Administrators, or admins, have overarching access to the entirety of the app, and can make changes to other registered users' profiles, listings, photos, and reviews in order to moderate the app more efficiently. Admins share the permissions of registered and unregistered users as well, and are able to use the app as a normal registered user would.

Items

Photos: Photos belong to listings, and are used to better visually depict an item being sold or offered.

Listings: Every Registered user can create listings, which are how items are viewed online for other users to buy or leave reviews under. Listings typically have photos that belong to them, and the listings themselves belong to each user that created it.

Profile: Registered users can create a profile for their account, which may include a profile picture, an about section or bio, and other details regarding their intentions to buy or sell on the app.

Reviews: Reviews are attached to specific item listings, and better communicate the quality of the item as well as the seller's efficiency. Registered users can leave reviews under listings.

Chat Log: Chat logs are shared between users, and improve the flow of conversation between buyer and seller. Chat logs belong to users and are sorted by the second user being communicated with.

5. List high-level functional requirements.

1. Users of all levels of authority and privileges shall be able to access the app and see listings and reviews of associated listings.
2. Unregistered users shall be able to search based on text search and filter by category.
3. Unregistered users shall have access to a settings tab, where they can change the app's appearance to become a vision-friendly dark mode and set other settings.
4. Unregistered users shall be given the opportunity to create an account, which will allow them to become registered users and gain access to all associated privileges and benefits.
5. Registered users shall agree to the terms and conditions upon account creation.
6. Registered users shall be able to edit and change their user profile.
7. Registered users can create listings and reviews for other users.
8. Registered users shall be able to search based on various forms of input text search, filter by category, and user search.
9. Registered users shall have access to a more robust settings tab, complete with account owner features that enable more control over their personal user experience.
10. Registered users shall be able to create chat logs with other registered users.
11. Registered users shall be able to block or report other users and listings, automatically notifying the admin to enact moderation.
12. Admin shall be able to create, delete, and edit all listings, regardless of ownership, to moderate the app.
13. Admin shall be able to review reported listings and users.
14. Admin shall be able to ban registered users.
15. Admin shall be able to create, delete, and edit all categories for searching on the website.

6. List of non-functional requirements

1. Application shall be developed, tested and deployed using tools and cloud 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 shall render well on mobile devices (no native app to be developed)
4. Posting of sales information and messaging to sellers shall be limited only to SFSU students
5. Critical data shall be stored in the database on the team's deployment server.
6. No more than 50 concurrent users shall be accessing the application at any time
7. Privacy of users shall be protected
8. The language used shall be English (no localization needed)
9. Application shall be very easy to use and intuitive
10. Application shall follow established architecture patterns
11. Application code and its repository shall be easy to inspect and maintain
12. Google analytics shall be used
13. No e-mail clients or chat services 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
14. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.
15. Site security: basic best practices shall be applied (as covered in the class) for main data items
16. Media formats shall be standard as used in the market today
17. Modern SE processes and tools shall be used as specified in the class, including collaborative and continuous SW development and GenAI tools
18. The application UI (WWW and mobile) shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Spring 2025. For Demonstration Only" at the top of the WWW page Nav bar. (Important so as to not confuse this with a real application). You have to use this exact text without any editing.

7. Competitive analysis (functions/features only, not business or marketing)

<u>Feature</u>	<i>ThriftAtSFSU</i>	<i>Craigslist</i>	<i>Facebook Marketplace</i>	<i>eBay</i>
Browsing and Searching	Advanced filtering specific to academic needs (books, dorm supplies)	Generic with broad categories	Advanced with various filters but not student-specific	Highly sophisticated with detailed categories
Contacting the Seller	Secure in-site messaging with SFSU email verification	Direct, less secure	Integrated with Facebook Messenger, more public	In-site messaging, more formal and structured
Uploading Information	Streamlined for student items, includes academic categories	Simple, less structured	Detailed with photo uploads, general categories	Detailed with options for professional listings
Dashboard for Sellers	Customized for student sellers	Basic functionality	More developed with social insights	Advanced with analytics and sales tracking
Admin Functions	Strict item and user moderation to ensure academic integrity	Minimal, mostly automated	Some oversight, not academic-focused	Comprehensive oversight with robust user and listing management
Mobile Responsiveness	Fully responsive design optimized for student use on campus	Varies significantly	Highly responsive and well-designed	Fully responsive with a strong mobile app
Security	High, with SFSU-only access and strict data handling	Low, no specific student security measures	Moderate, with general security measures	High with advanced user verification and

<u>Feature</u>	<i>ThriftAtSFSU</i>	<i>Craigslist</i>	<i>Facebook Marketplace</i>	<i>eBay</i>
				transaction security measures
Exclusivity	Exclusive to SFSU students for safety and relevance	Open to the general public	Open to the general public, although community groups can be exclusive	Open to the general public with a global reach

8. High-level system architecture and technologies used

Server Host: Amazon AWS

Operating System: Ubuntu v.20.04 Server

Database: MySQL v.9.1

Web Server: NGINX 1.12.2

Server-Side Language: Python

Browser Support: Chrome (Version 133.0.6943.143), Firefox (136.0)

Additional Technologies: Web Framework: Flask

IDE: PyCharm, VScode

Web Analytics: Google Analytics

SSL Cert: Lets Encrypt (Cert Bot)

SASS: 3.5.5

9. Use of GenAI tools like ChatGPT and Copilot for Milestone 1

- What GenAI tool and version you used/tried.
 - GPT-4o
- List tasks for which you used or tried GenAI tools and for each rate how useful it was, use LOW, MEDIUM, HIGH.
 - Help generate an executive summary (HIGH)
 - Revise executive summary (HIGH)
 - Generate Persona personalities (MEDIUM)
- For each task above briefly explain how you used the tool and what benefit it offered. If the tool was not beneficial, do not use it for M1, but explain why.
 - Used the tool to generate an executive summary for this Milestone 1 document.
 - Then, to help modify the summary after CTO/CEO comments
 - Used the tool to help generate personas and high-level use cases with the description of what we needed
- Provide key examples and prompts.
 - Prompt:"Help me write an executive summary for my website: Short description of the final product/application and its key advantages, novelty, value (up to 1 page). Make it as an executive summary to be readable to broad and not just technical audience – think also of answering the question of why we should fund this project. We suggest you assign a name to your project for easier reference and good “marketing”. This summary should be readable to a general manager/executive that is not a CS specialist and is used to explain and also to advertise/promote your project. Typical outline is: one paragraph on the motivation and importance of the application you are developing, followed by a paragraph on what functions and services your application will be providing and how it helps the users (high level only, no jargon). You also must say what is unique and custom for SFSU in your design. At the end say in one paragraph something about your team (e.g. about your student startup team...). (BTW ChatGPT may help here BUT be careful not to use meaningless fluff written in perfect English, a common error in blindly using ChatGPT or likes) The name: ThriftAtSFSU Advantages: Local for SFSU students, seller ratings and schedule for meet up time convenience Novelty: SFSU specific resale site, Payment is done off platform Value: For ease of community access to affordable second-hand goods Only registered users with a valid sfsu email can access the platform and look at listings and post listings. There is search function with filters. “
 - Revising the Executive summary: Gave it the original and told it to change the 2nd paragraph to include the features we will support and gave them the table of features we had in the competitive analysis
 - The output is in [Executive Summary](#)
- Comment on anything else you found useful.
 -

10. Team and roles

Name	Email	Role
Hilary Lui	hlui@sfsu.edu	Team Lead
Annison Van	avan@sfsu.edu	Frontend Lead
Sid Padmanabhuni	spadmanabhuni@sfsu.edu	Backend Lead & GitHub maintainer
Joseph Alhambra	jalhambra@sfsu.edu	Team Member Frontend
Joseph Shur	jshur1@sfsu.edu	Team Member Backend

11. Team Lead Checklist to be completed by team lead

- So far, all team members are fully engaged and attending team sessions when required
 - DONE
- The team found a time slot to meet outside of the class
 - DONE
- 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
 - DONE
- The team reviewed class slides on requirements and use cases before drafting Milestone 1
 - DONE
- The team reviewed non-functional requirements from “How to start...” document and developed Milestone 1 consistently.
 - DONE
- The team lead checked Milestone 1 document for quality, completeness, formatting and compliance with instructions before the submission.
 - DONE
- The team lead ensured that all team members read the final M1 and agree/understand it before submission.
 - DONE
- The team shared and discussed experience with GenAI tools among themselves
 - DONE
- GitHub organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.)
 - DONE

SW Engineering CSC648-848 Spring 2025

ThriftAtSFSU

Milestone 2 Part 1

Team 15

Hilary Lui (hlui@sfsu.edu)	Team Lead
Annison Van	Frontend Lead
Sid Padmanabhuni	Backend Lead & GitHub master
Joseph Alhambra	Team Member Frontend
Joseph Shur	Team Member Backend

Date Submitted	3/25/2025
Date Revised	5/19/2025

1. Executive Summary

In today's fast-paced college environment, affordability and convenience are key student concerns. ThriftAtSFSU is a dedicated resale platform designed exclusively for San Francisco State University (SFSU) students, providing a secure and efficient way to buy and sell second-hand goods within the campus community.

Unlike generic resale platforms, ThriftAtSFSU is tailored to meet the specific needs of SFSU students, ensuring a trusted marketplace that fosters sustainability and affordability.

To create a seamless user experience, ThriftAtSFSU offers a range of features designed with students in mind. Users can easily browse and search for items using advanced filters for academic-specific needs, such as textbooks and dorm supplies. A secure in-site messaging system ensures safe communication between buyers and sellers, requiring SFSU email verification. Listing items is streamlined for students, featuring predefined academic categories for quicker uploads. Sellers benefit from a listing-based chatting system to keep track of their items and also the ability to put their schedule on their profile to make coordinating pick-up times easier. To maintain a safe and trustworthy marketplace, the platform includes strict item and user moderation managed by administrators. Additionally, ThriftAtSFSU is fully mobile-responsive, allowing students to access the marketplace conveniently from any device.

What sets ThriftAtSFSU apart is its exclusive focus on the SFSU community, ensuring a localized and student-friendly marketplace that prioritizes security, convenience, and affordability. This project directly addresses the need for a reliable, campus-specific resale network, fostering a culture of sustainability while making essential goods more accessible to students. Our team comprises passionate SFSU students committed to enhancing campus life through innovative solutions. With firsthand experience of students' challenges in accessing affordable goods, we are dedicated to making ThriftAtSFSU the go-to platform for peer-to-peer resale within our university. By supporting this project, you invest in a practical and impactful solution that benefits the entire SFSU community.

2. List of main data items and entities

1. User
 - a. Description: Represents a registered individual on the platform (student or professor with an SFSU email). Users can act as buyers, sellers, or both.
 - b. Key Attributes:
 - i. user_id (PK)
 - ii. first_name
 - iii. last_name
 - iv. email (SFSU only, unique)
 - v. password
 - vi. phone_number
 - vii. profile_picture
 - viii. bio
 - ix. created_at
 - x. is_admin
2. Product
 - a. Description: Items listed for sale by users. Includes key details such as title, price, description, and condition.
 - b. Key Attributes:
 - i. product_id (PK)
 - ii. seller_id (FK to User)
 - iii. title
 - iv. description
 - v. price
 - vi. category_id (FK to Category)
 - vii. condition
 - viii. status (e.g. available, sold)
 - ix. created_at
3. Category
 - a. Description: Organizes products into logical categories (e.g. Electronics, Furniture, Books) for easier browsing and searching.
 - b. Key Attributes:
 - i. category_id (PK)
 - ii. name
 - iii. description
4. ProductImages
 - a. Description: Stores one or more images associated with a product listing.
 - b. Key Attributes:
 - i. image_id (PK)

- ii. product_id (FK to Product)
- iii. image_data
- iv. uploaded_at
- v. image_order

5. Message

a. Description: Stores private messages exchanged between users regarding specific products. Useful for inquiries, negotiations, and meetups.

b. Key Attributes:

- i. message_id (PK)
- ii. sender_id (FK to User)
- iii. receiver_id (FK to User)
- iv. product_id (FK to Product)
- v. content
- vi. timestamp

6. Review

a. Description: Ratings and feedback left by users after interacting with a buyer or seller to build reputation and trust.

b. Key Attributes:

- i. review_id (PK)
- ii. reviewer_id (FK to User)
- iii. reviewed_user_id (FK to User)
- iv. rating (e.g. 1-5)
- v. comment
- vi. timestamp

7. Reports

a. Description: Tracks reports made by users against other users or products for moderation purposes.

b. Key Attributes:

- i. report_id (PK)
- ii. reporter_id (FK to User)
- iii. reported_user_id (FK to User)
- iv. reported_product_id (FK to Product)
- v. reason
- vi. status (e.g. pending, resolved, dismissed)
- vii. created_at
- viii. resolved_at

8. Wishlist

a. Description: Allows users to save items they're interested in for future consideration.

b. Key Attributes:

- i. wishlist_id (PK)

- ii. user_id (FK to User)
- iii. product_id (FK to Product)
- iv. date_added

9. UserAvailability

a. Description: Stores a user's available times for potential meetups or communication windows.

b. Key Attributes:

- i. availability_id (PK)
- ii. user_id (FK to User)
- iii. day_of_week (e.g. Monday)
- iv. time_slot (e.g. 6:00, 6:30)
- v. is_available
- vi. last_updated

3. Functional Requirements

Priority 1 – must-have:

1. Users of all levels of authority and privileges shall be able to access the app and see listings and reviews of associated listings.
2. Unregistered users shall be given the opportunity to create an account, which will allow them to become registered users and gain access to all associated privileges and benefits.
3. Unregistered users shall be able to search based on text search and filter by category.
4. Registered users shall agree to the terms and conditions upon account creation.
5. Registered users can create listings and reviews for other users.
6. Registered users shall be able to create chat logs with other registered users.
7. Admin shall be able to create listings, regardless of ownership.

Priority 2 – desired:

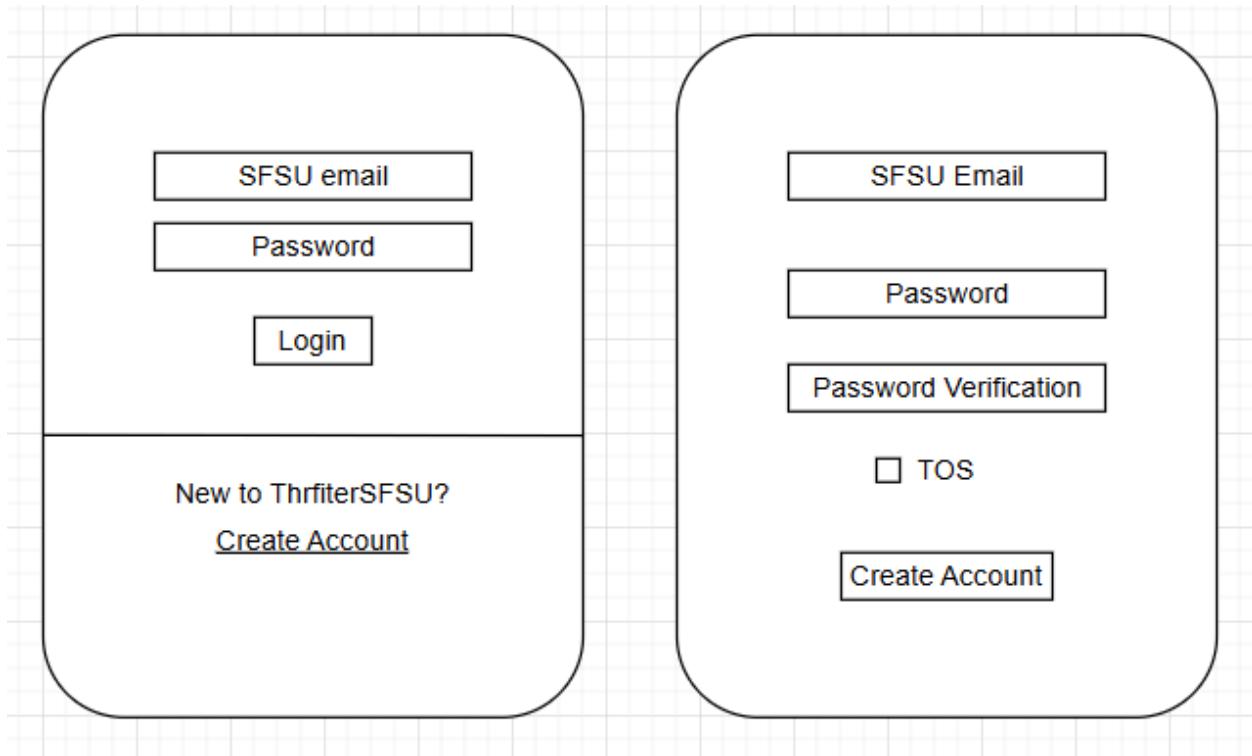
8. Registered users shall be able to edit and change their user profile.
9. Registered users shall be able to block or report other users and listings, automatically notifying the admin to enact moderation.
10. Admin shall be able to review reported listings and users.
11. Admin shall be able to ban registered users.
12. Admin shall be able to delete and edit listings, regardless of ownership, to moderate the app.

Priority 3 – opportunistic:

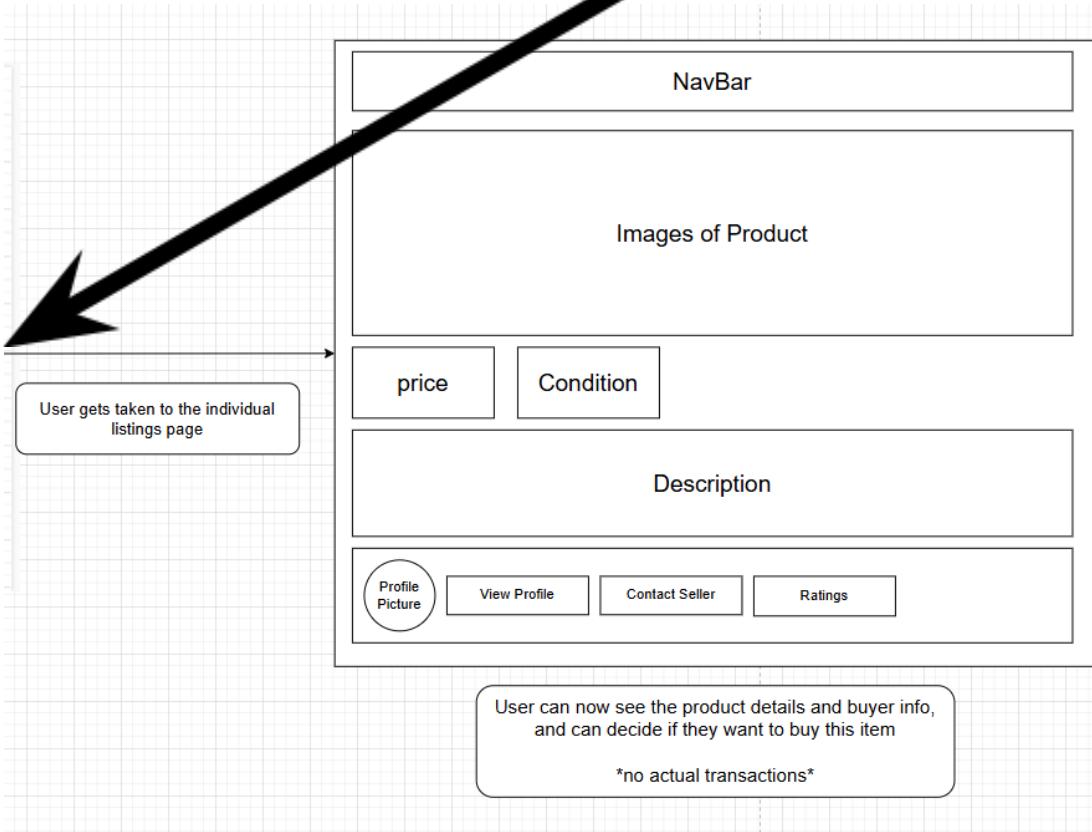
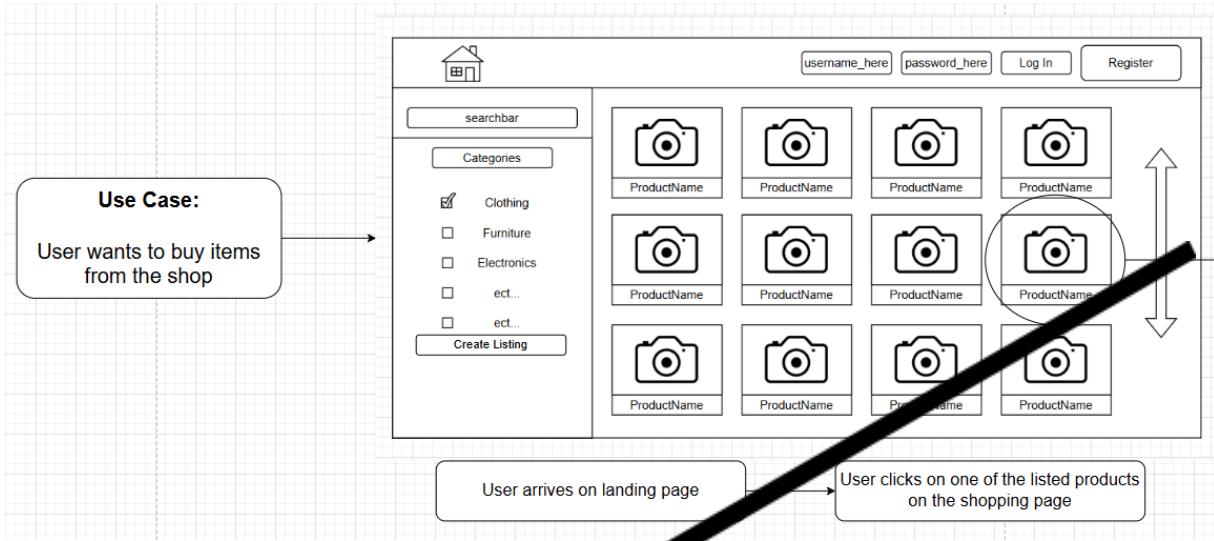
13. Unregistered users shall have access to a settings tab, where they can change the app's appearance to become a vision-friendly dark mode and set other settings.
14. Registered users shall have access to a more robust settings tab, complete with account owner features that enable more control over their personal user experience.
15. Admin shall be able to create, delete, and edit all categories for searching on the website.

4. UI Storyboards for each main use case

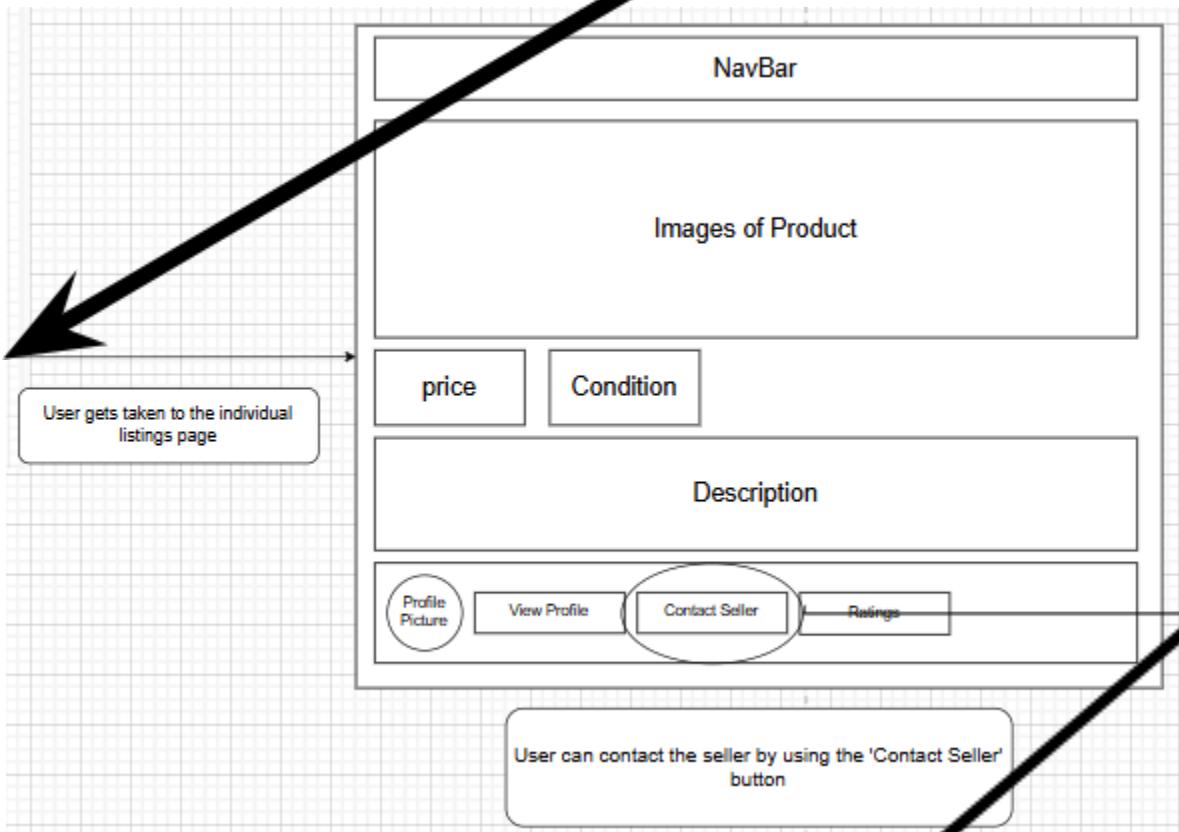
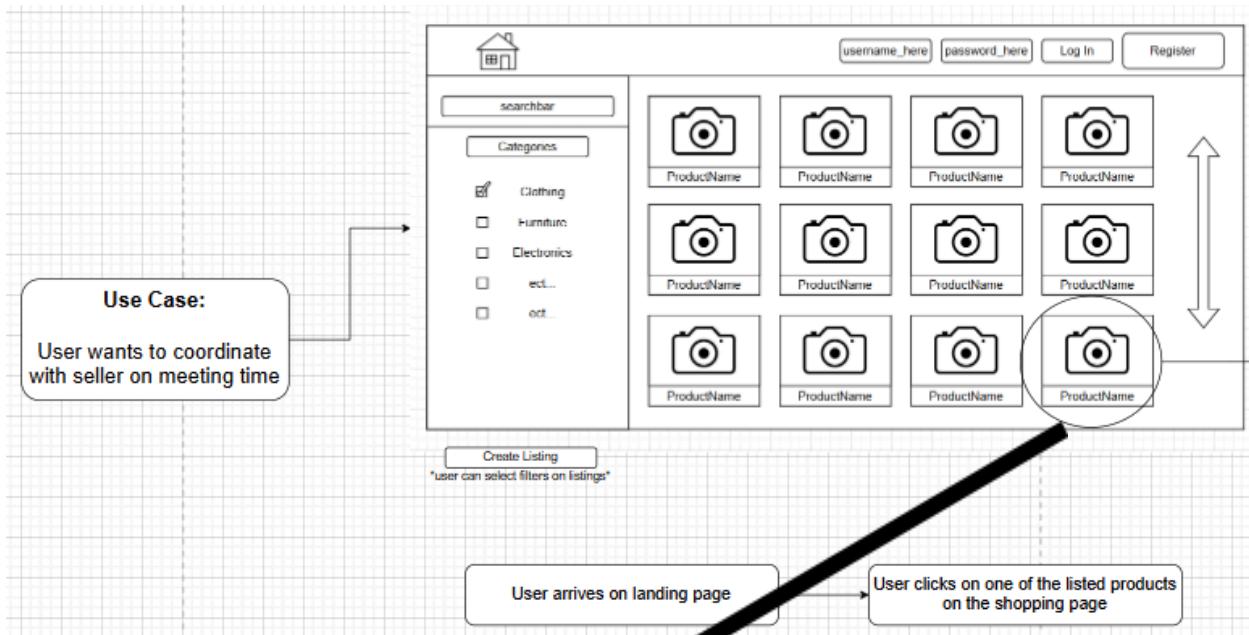
Login and registration mockups

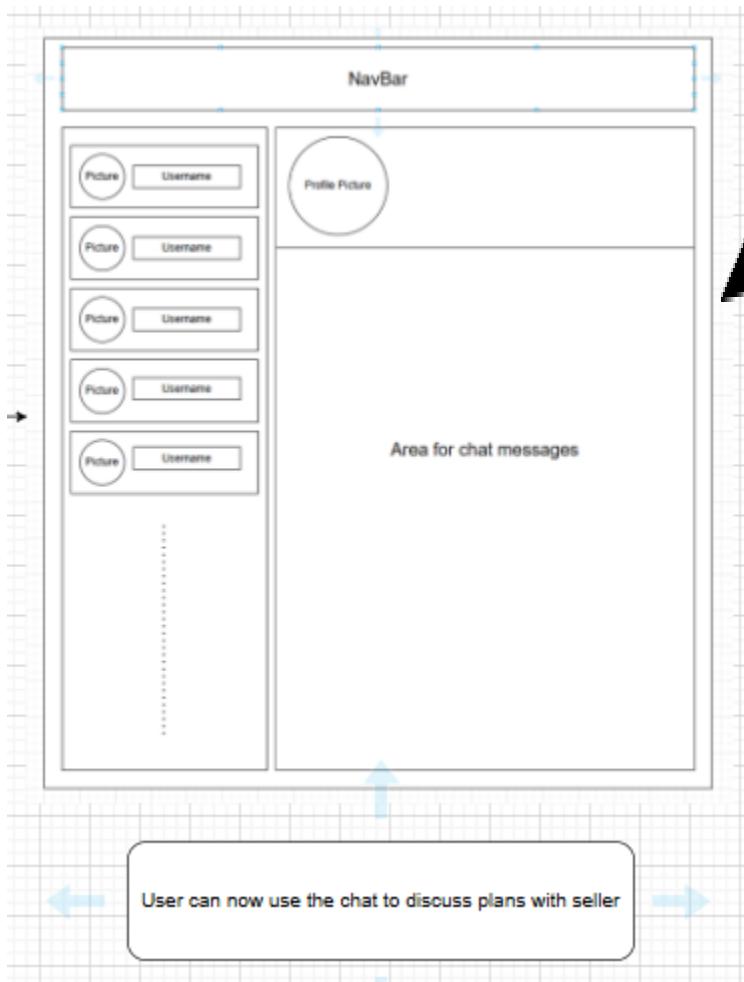
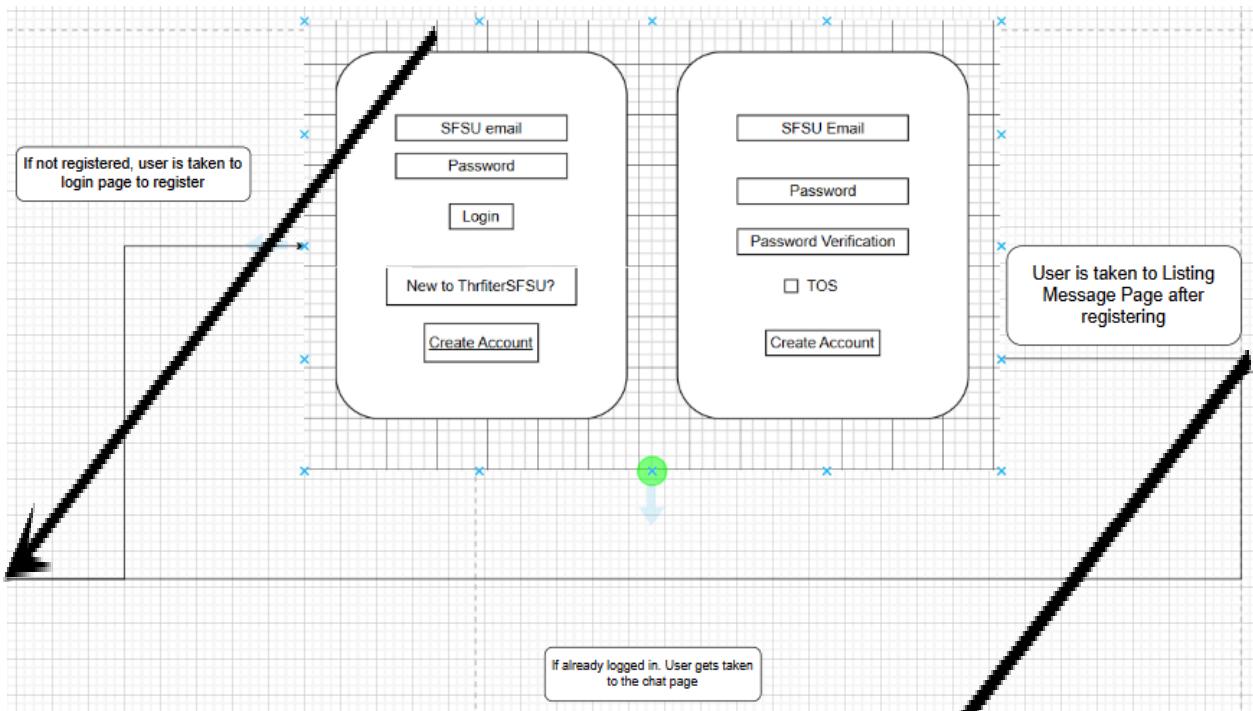


1.

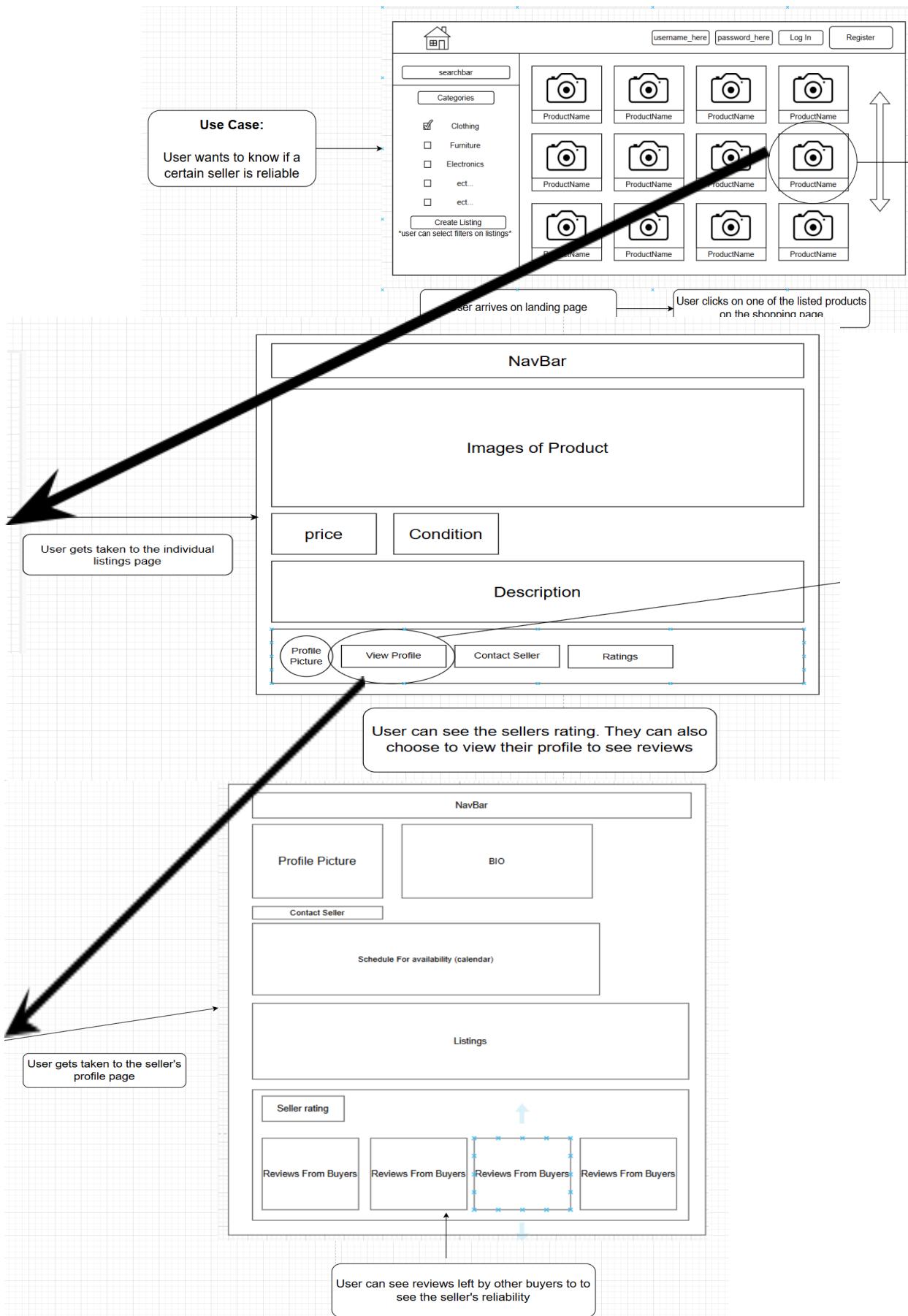


2.

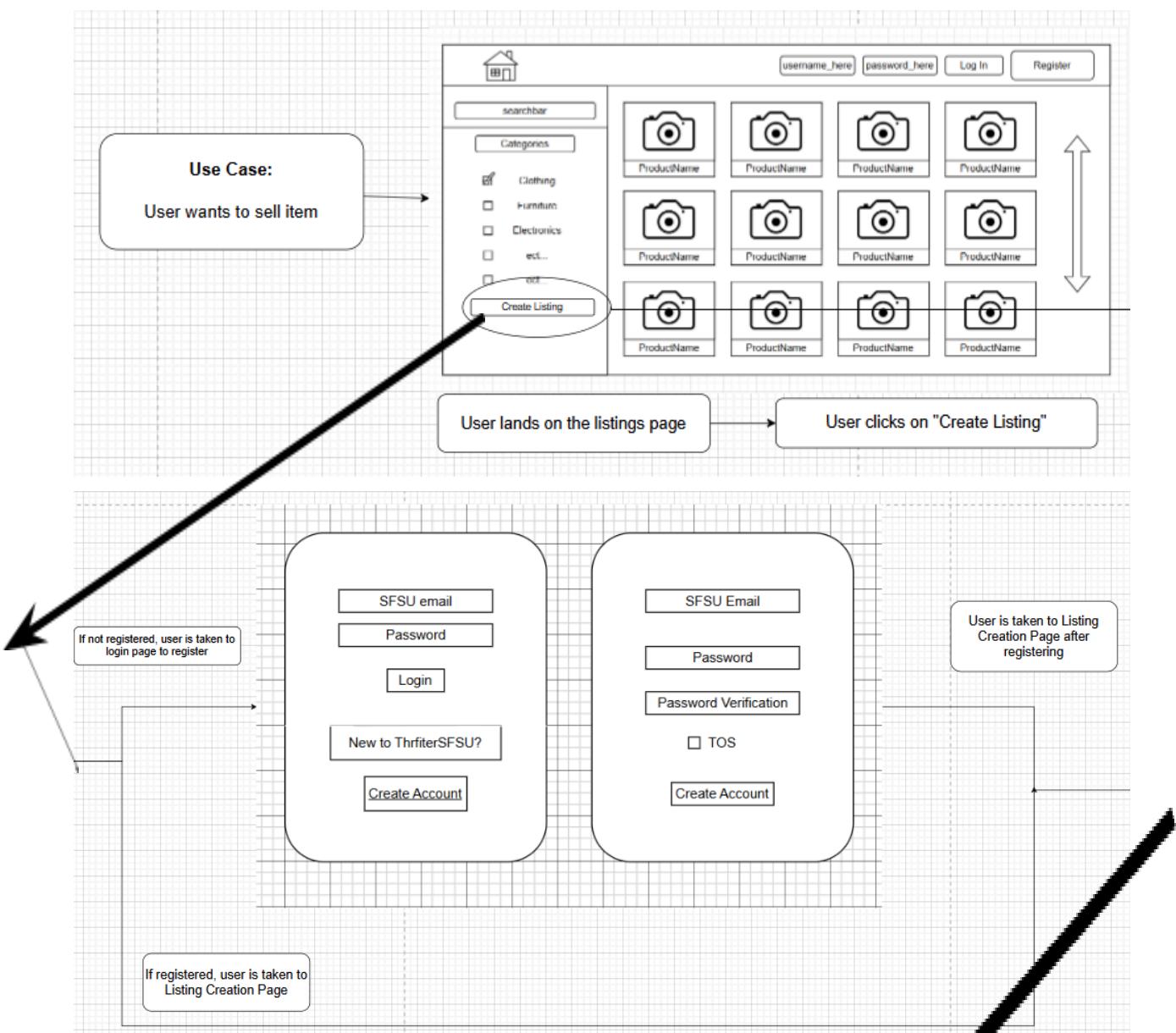


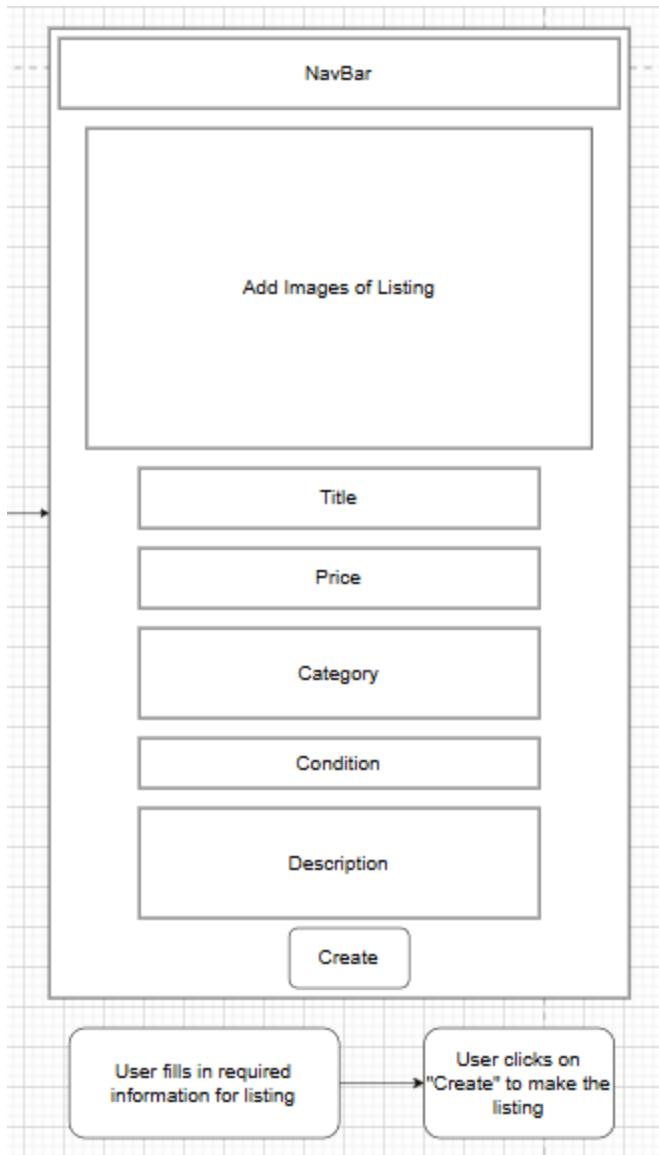


3.



4.





5. High level Architecture, Database Organization summary only

Table and column names follow the naming from Section 2 of our doc:

1. Users

- `user_id` (Primary Key)
- `first_name`
- `last_name`
- `email` (SFSU only)
- `password`
- `phone_number`
- `profile_picture`
- `created_at`
- `is_admin`

2. Products

- `product_id` (Primary Key)
- `seller_id` (Foreign Key to Users)
- `title`
- `description`
- `price`
- `category_id` (Foreign Key to Category)
- `status` (e.g., "available", "sold")
- `Created_at`
- `product_image`
- `thumbnail`

3. Categories

- `category_id` (Primary Key)
- `name`
- `description`

4. Messages

- `message_id` (Primary Key)
- `sender_id` (Foreign Key to Users)
- `receiver_id` (Foreign Key to Users)
- `product_id` (Foreign Key to Products)

- `content`
- `timestamp`

5. Reviews

- `review_id` (Primary Key)
- `reviewer_id` (Foreign Key to Users)
- `reviewed_user_id` (Foreign Key to Users)
- `rating` (e.g., 1–5)
- `comment`
- `timestamp`

6. Transactions

- `transaction_id` (Primary Key)
- `buyer_id` (Foreign Key to Users)
- `seller_id` (Foreign Key to Users)
- `product_id` (Foreign Key to Products)
- `agreed_price`
- `status` (e.g., "completed", "canceled")
- `timestamp`

7. Wishlist

- `wishlist_id` (Primary Key)
- `user_id` (Foreign Key to Users)
- `product_id` (Foreign Key to Products)
- `Date_added`

Media Storage

- Media Type: Images (e.g., profile pictures and product photos)
- Storage Method: Stored directly as BLOBs in the database.
- Justification: Since the project is not operating at production scale, and media files are relatively small, storing them in BLOBs keeps the architecture simpler and easier to manage.
- Fields that will store BLOBs:
 - `Users.profile_picture`
 - `Products` table for product images (e.g., `product_image` as BLOB)
- Other Media:
 - No video/audio is currently supported.
 - No GPS data planned at this stage

Search / Filter Architecture and Implementation

- Search Algorithm/Software: Standard SQL with `%LIKE%` for substring matching (as recommended in architecture slides).
- Searchable Data Fields:
 - Products
 - `title` (text search with `%LIKE%`)
 - `description` (text search with `%LIKE%`)
 - `price` (numeric filter using `BETWEEN`)
 - `status` (dropdown filter: available, sold)
 - `created_at` (used for sorting by recency)
 - Categories
 - `name` (text search with `%LIKE%`)
 - Users
 - `first_name` and `last_name` (text search with `%LIKE%`)
 - `email` (searchable for admin tools only)
- Filtering and Sorting Options:
 - Category-based filtering (via `category_id`)
 - Price range filtering (`price BETWEEN x AND y`)
 - Status filtering (`status = 'available'`)
 - Sorting by `created_at` to show the most recent listings first
- Implementation:
 - SQL queries using `WHERE ... LIKE '%query%'` for relevant fields.
 - Combined with `JOIN` clauses to pull category or user info as needed.
 - We won't be using things like ElasticSearch since this is a small-scale project

Special Algorithms or Processes

- Rating & Review System:
 - Users can leave ratings and comments for others post transaction.
 - A simple average rating can be shown on profiles.

6. Identify actual key risks for your project at this time

- Schedule risks (can you make it, given what you committed and the resources)
 - The front-end lead is getting more work on the side, so he might have less time to work on the website for the next few weeks. It's just a heads-up, but he should still be able to function normally within the team.
 - The meeting time conflicts with dinner for most people, so we may move the meeting to an earlier time.

7. Project management

We met on Friday to assign tasks to everyone. Independent work was done from there, and we used Discord to ask for feedback on work and mini-meetings situationally. We used GitHub projects to assign the wireframe task to the frontend team.

Wireframe #17

The screenshot shows a GitHub project issue titled "Wireframe #17". The issue was opened by Hluui 2 weeks ago and is currently private. It has three assigned users: JosephCVA, anvan882, and Hluui. There are no labels applied. The type is set to "Task". The project is CSC648-0104-team15, and the status is "In Progress". The issue details are as follows:

Hluui opened 2 weeks ago · edited by Hluui

Assignees: JosephCVA, anvan882

Labels: No labels

Type: Task

Projects: CSC648-0104-team15

Status: In Progress

Issue Details:

- Landing page (Not signed-in)
- Login
- Sign up (email w/ SFSU "verification", password, password verification, terms and conditions checkbox)

Every Page Beyond this point should include the navbar

- Listing home page (Includes search bar, filters, and listings)
 - [x] Listings Should Include a picture, Name of seller, and price
 - [x] Filters: Category(Sports, Books, Clothing, Food, Travel), Condition (new, like new, used)
- Listing Creation Page (Fields for adding image, title, price, category, and description)
- Individual Listing Page (image, title, price, category, and description)
 - [x] Button to contact seller and look at seller profile
- Profile Page (Has the seller schedule, bio, and area for reviews)
- Chat Screen (Multiple private chats, chat messages between seller and buyer, a method to connect back to listing, dropdown for each listing-related chat)
- Wishlist with listings in it
- Navbar (logo/Home, messages, wishlist, profile)

Create sub-issue

Looking back, I should have set more concrete deadlines at the end of the initial meeting so we didn't rush to finish everything on the due date. In the future, we will use more of the GitHub project's issues for the milestones and the roadmap on GitHub as a Gantt chart for managing deadlines.

8. Use of GenAI tools like ChatGPT and Copilot for Milestone 2

- What GenAI tool and version did you use/try?
 - GPT-4o
 - Copilot
- List tasks for which you used or tried GenAI tools, and for each, rate how useful it was, on a scale of LOW, MEDIUM, HIGH.
 - List of main data items and entities - Medium
 - High Level Architecture - Medium
- For each task above, briefly explain how you used the tool and what benefit it offered. If the tool was not beneficial, do not use it for M1, but explain why.
 - Gave a list of main data items, and a list of entities after explaining our project. Asked it to help explain them in a short description
 - Asked it the best way to store media for our project is, as well as the best way to search/filter. Gave it a list of fields that can be filtered/searched, and it gave a structured output.
- Provide key examples and prompts.
 - We gave it a list of main entities like User, Product, Category, Message, Review, Transaction, Wishlist, and asked it to describe and give us some attributes
 - A full list of searchable fields across all tables, organized clearly, and describe how we'll implement search and filters using SQL?
- Comment on anything else you found useful.
 - It gave us what we already knew, just extremely quickly saving us on time.

9. Team Lead Checklist to be completed by the team lead

- So far, all team members are fully engaged and attending team sessions when required
 - DONE
- Team is ready and able to use the chosen back and front-end frameworks, and those who need to learn are working on learning and practicing
 - DONE
- The team reviewed suggested resources before drafting Milestone 2
 - DONE
- The team lead checked the Milestone 2 document for quality, completeness, formatting, and compliance with instructions before the submission
 - DONE
- The team lead ensured that all team members read the final Milestone 2 document and agreed/understood it before submission.
 - DONE
- The team shared and discussed the experience with GenAI tools among themselves
 - DONE

SW Engineering CSC648-848 Spring 2025

ThriftAtSFSU

Milestone 2 Part 1

Team 15

Hilary Lui (hlui@sfsu.edu)	Team Lead
Annison Van	Frontend Lead
Sid Padmanabhuni	Backend Lead & GitHub master
Joseph Alhambra	Team Member Frontend
Joseph Shur	Team Member Backend

Date Submitted	3/25/2025
Date Revised	5/19/2025

1. Executive Summary

In today's fast-paced college environment, affordability and convenience are key student concerns. ThriftAtSFSU is a dedicated resale platform designed exclusively for San Francisco State University (SFSU) students, providing a secure and efficient way to buy and sell second-hand goods within the campus community.

Unlike generic resale platforms, ThriftAtSFSU is tailored to meet the specific needs of SFSU students, ensuring a trusted marketplace that fosters sustainability and affordability.

To create a seamless user experience, ThriftAtSFSU offers a range of features designed with students in mind. Users can easily browse and search for items using advanced filters for academic-specific needs, such as textbooks and dorm supplies. A secure in-site messaging system ensures safe communication between buyers and sellers, requiring SFSU email verification. Listing items is streamlined for students, featuring predefined academic categories for quicker uploads. Sellers benefit from a listing-based chatting system to keep track of their items and also the ability to put their schedule on their profile to make coordinating pick-up times easier. To maintain a safe and trustworthy marketplace, the platform includes strict item and user moderation managed by administrators. Additionally, ThriftAtSFSU is fully mobile-responsive, allowing students to access the marketplace conveniently from any device.

What sets ThriftAtSFSU apart is its exclusive focus on the SFSU community, ensuring a localized and student-friendly marketplace that prioritizes security, convenience, and affordability. This project directly addresses the need for a reliable, campus-specific resale network, fostering a culture of sustainability while making essential goods more accessible to students. Our team comprises passionate SFSU students committed to enhancing campus life through innovative solutions. With firsthand experience of students' challenges in accessing affordable goods, we are dedicated to making ThriftAtSFSU the go-to platform for peer-to-peer resale within our university. By supporting this project, you invest in a practical and impactful solution that benefits the entire SFSU community.

2. List of main data items and entities

1. User
 - a. Description: Represents a registered individual on the platform (student or professor with an SFSU email). Users can act as buyers, sellers, or both.
 - b. Key Attributes:
 - i. user_id (PK)
 - ii. first_name
 - iii. last_name
 - iv. email (SFSU only, unique)
 - v. password
 - vi. phone_number
 - vii. profile_picture
 - viii. bio
 - ix. created_at
 - x. is_admin
2. Product
 - a. Description: Items listed for sale by users. Includes key details such as title, price, description, and condition.
 - b. Key Attributes:
 - i. product_id (PK)
 - ii. seller_id (FK to User)
 - iii. title
 - iv. description
 - v. price
 - vi. category_id (FK to Category)
 - vii. condition
 - viii. status (e.g. available, sold)
 - ix. created_at
3. Category
 - a. Description: Organizes products into logical categories (e.g. Electronics, Furniture, Books) for easier browsing and searching.
 - b. Key Attributes:
 - i. category_id (PK)
 - ii. name
 - iii. description
4. ProductImages
 - a. Description: Stores one or more images associated with a product listing.
 - b. Key Attributes:
 - i. image_id (PK)

- ii. product_id (FK to Product)
- iii. image_data
- iv. uploaded_at
- v. image_order

5. Message

a. Description: Stores private messages exchanged between users regarding specific products. Useful for inquiries, negotiations, and meetups.

b. Key Attributes:

- i. message_id (PK)
- ii. sender_id (FK to User)
- iii. receiver_id (FK to User)
- iv. product_id (FK to Product)
- v. content
- vi. timestamp

6. Review

a. Description: Ratings and feedback left by users after interacting with a buyer or seller to build reputation and trust.

b. Key Attributes:

- i. review_id (PK)
- ii. reviewer_id (FK to User)
- iii. reviewed_user_id (FK to User)
- iv. rating (e.g. 1-5)
- v. comment
- vi. timestamp

7. Reports

a. Description: Tracks reports made by users against other users or products for moderation purposes.

b. Key Attributes:

- i. report_id (PK)
- ii. reporter_id (FK to User)
- iii. reported_user_id (FK to User)
- iv. reported_product_id (FK to Product)
- v. reason
- vi. status (e.g. pending, resolved, dismissed)
- vii. created_at
- viii. resolved_at

8. Wishlist

a. Description: Allows users to save items they're interested in for future consideration.

b. Key Attributes:

- i. wishlist_id (PK)

- ii. user_id (FK to User)
- iii. product_id (FK to Product)
- iv. date_added

9. UserAvailability

a. Description: Stores a user's available times for potential meetups or communication windows.

b. Key Attributes:

- i. availability_id (PK)
- ii. user_id (FK to User)
- iii. day_of_week (e.g. Monday)
- iv. time_slot (e.g. 6:00, 6:30)
- v. is_available
- vi. last_updated

3. Functional Requirements

Priority 1 – must-have:

1. Users of all levels of authority and privileges shall be able to access the app and see listings and reviews of associated listings.
2. Unregistered users shall be given the opportunity to create an account, which will allow them to become registered users and gain access to all associated privileges and benefits.
3. Unregistered users shall be able to search based on text search and filter by category.
4. Registered users shall agree to the terms and conditions upon account creation.
5. Registered users can create listings and reviews for other users.
6. Registered users shall be able to create chat logs with other registered users.
7. Admin shall be able to create listings, regardless of ownership.

Priority 2 – desired:

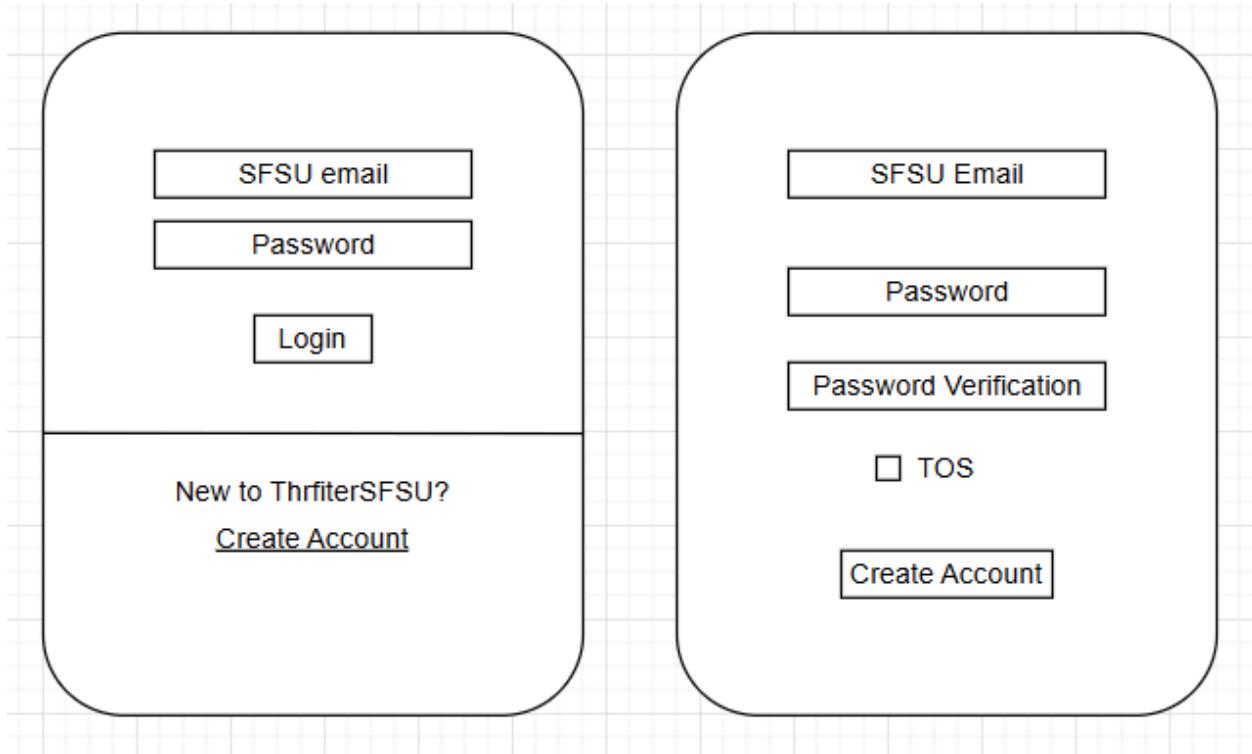
8. Registered users shall be able to edit and change their user profile.
9. Registered users shall be able to block or report other users and listings, automatically notifying the admin to enact moderation.
10. Admin shall be able to review reported listings and users.
11. Admin shall be able to ban registered users.
12. Admin shall be able to delete and edit listings, regardless of ownership, to moderate the app.

Priority 3 – opportunistic:

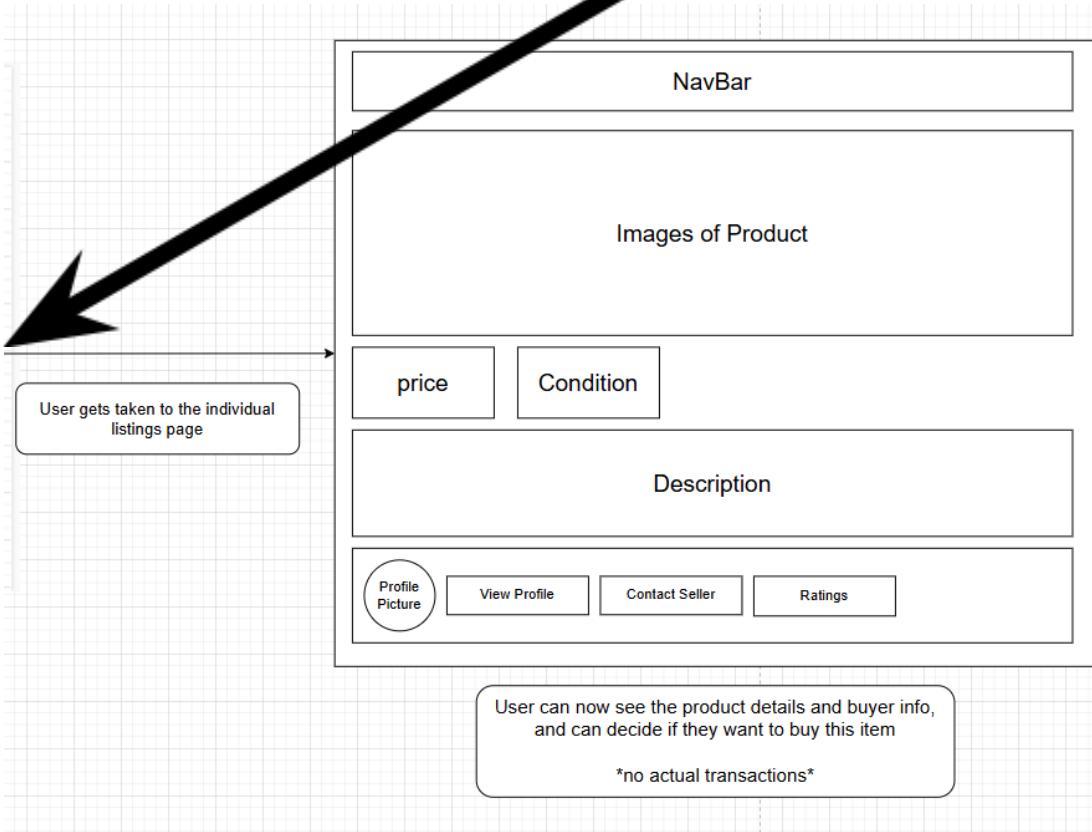
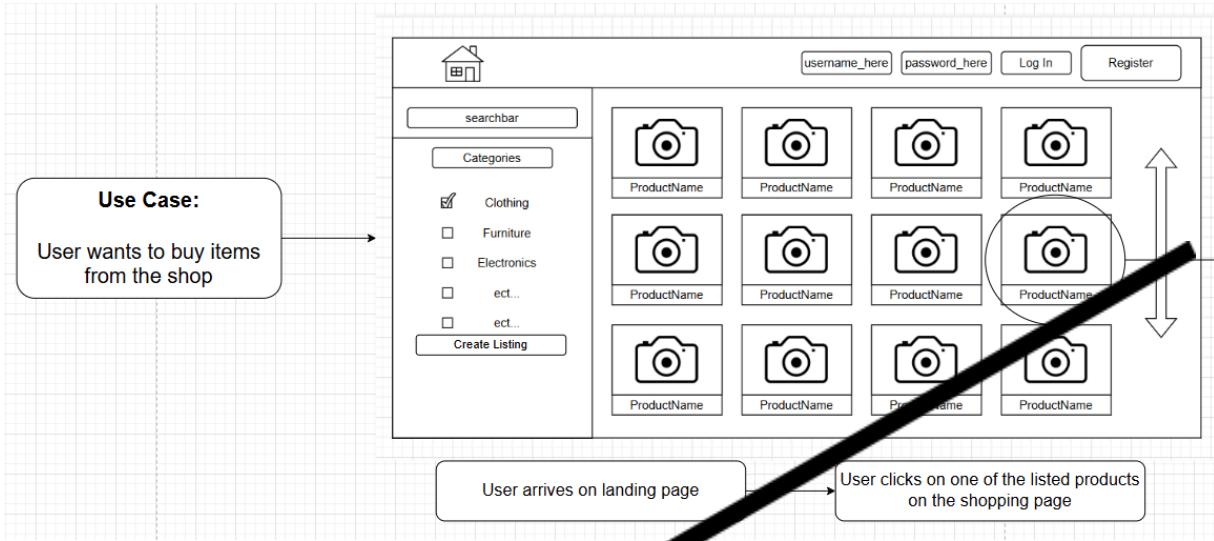
13. Unregistered users shall have access to a settings tab, where they can change the app's appearance to become a vision-friendly dark mode and set other settings.
14. Registered users shall have access to a more robust settings tab, complete with account owner features that enable more control over their personal user experience.
15. Admin shall be able to create, delete, and edit all categories for searching on the website.

4. UI Storyboards for each main use case

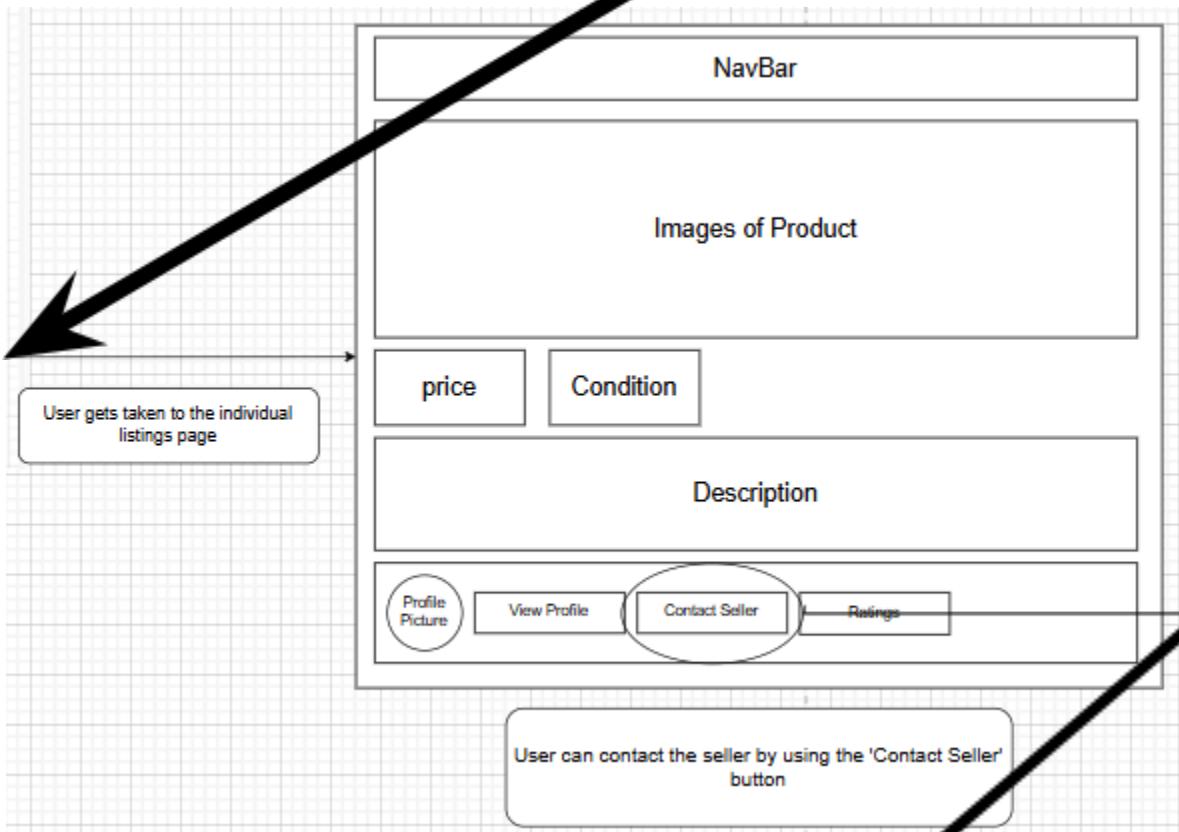
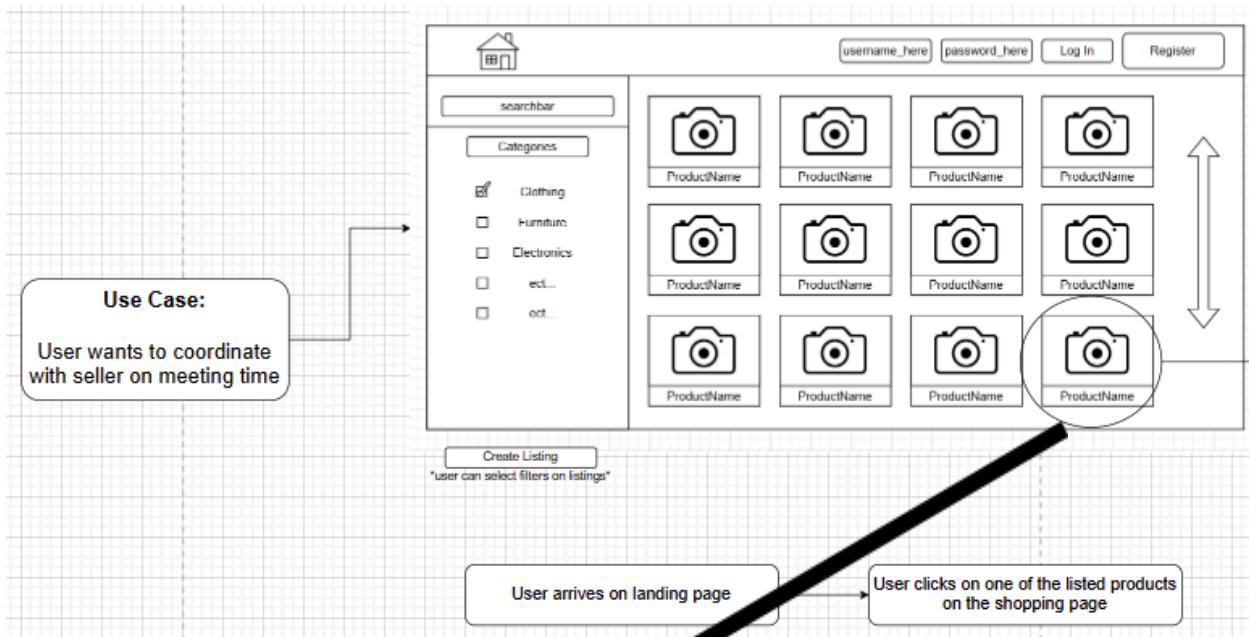
Login and registration mockups

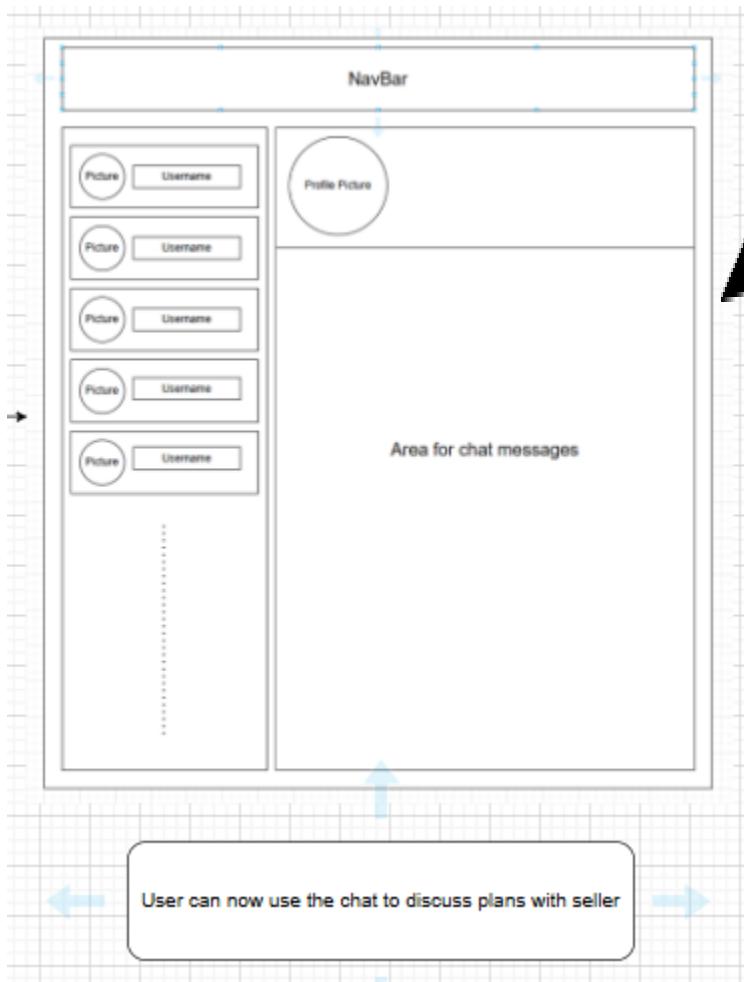
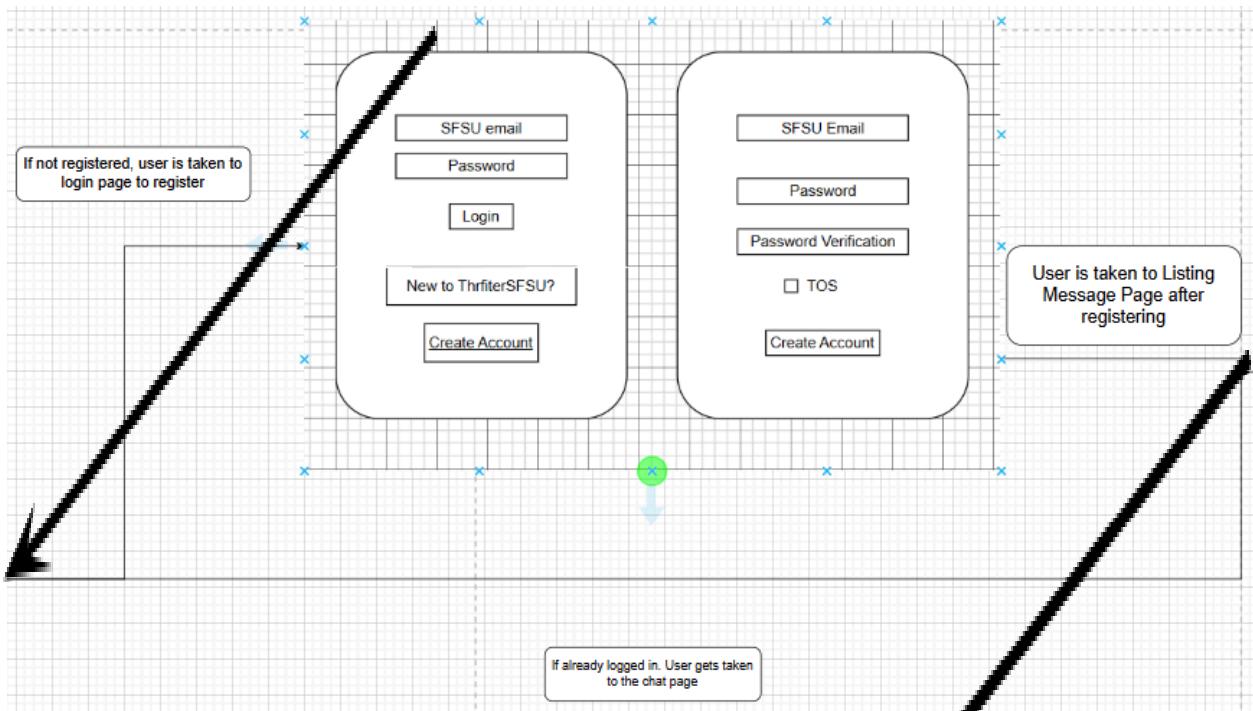


1.

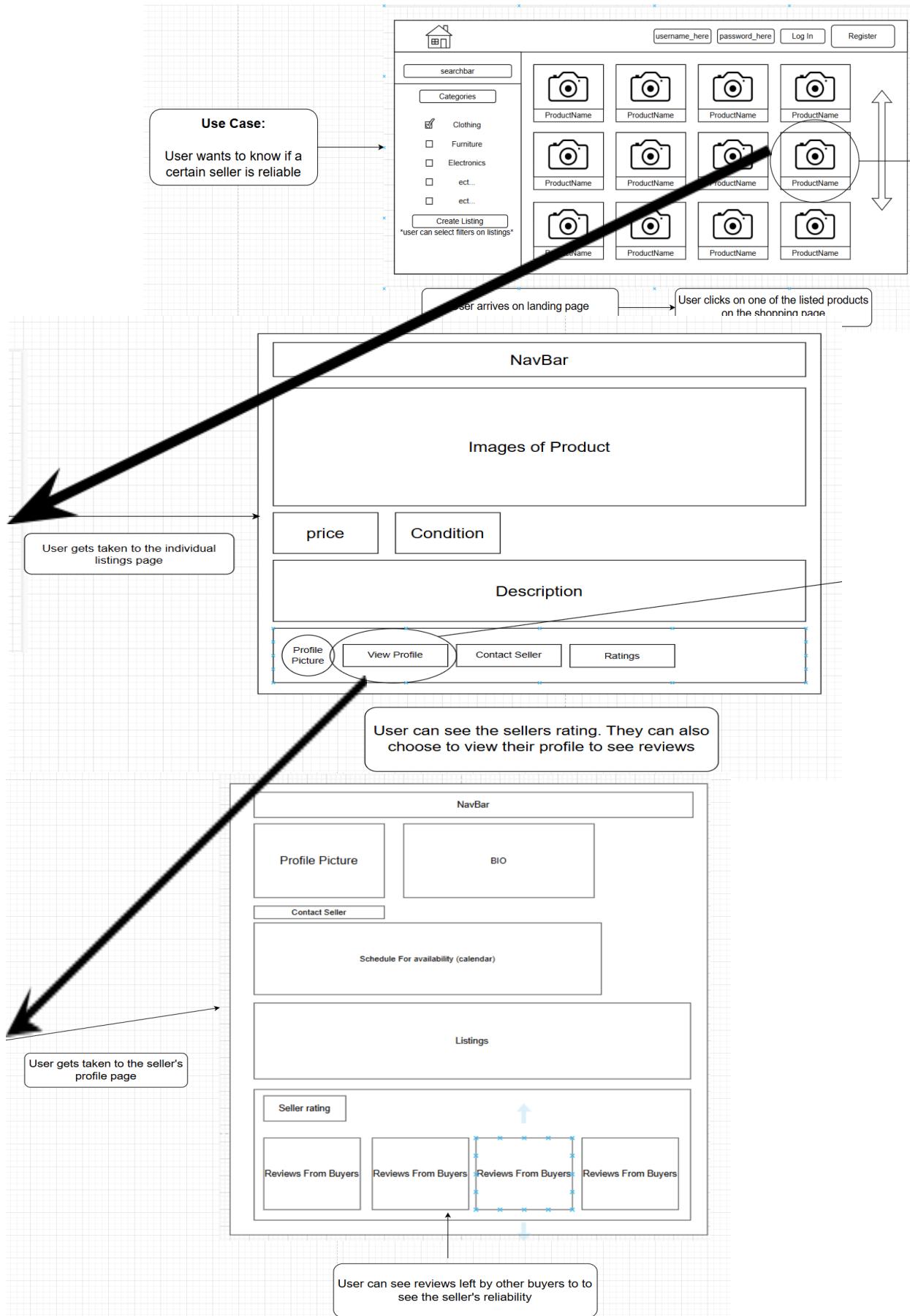


2.

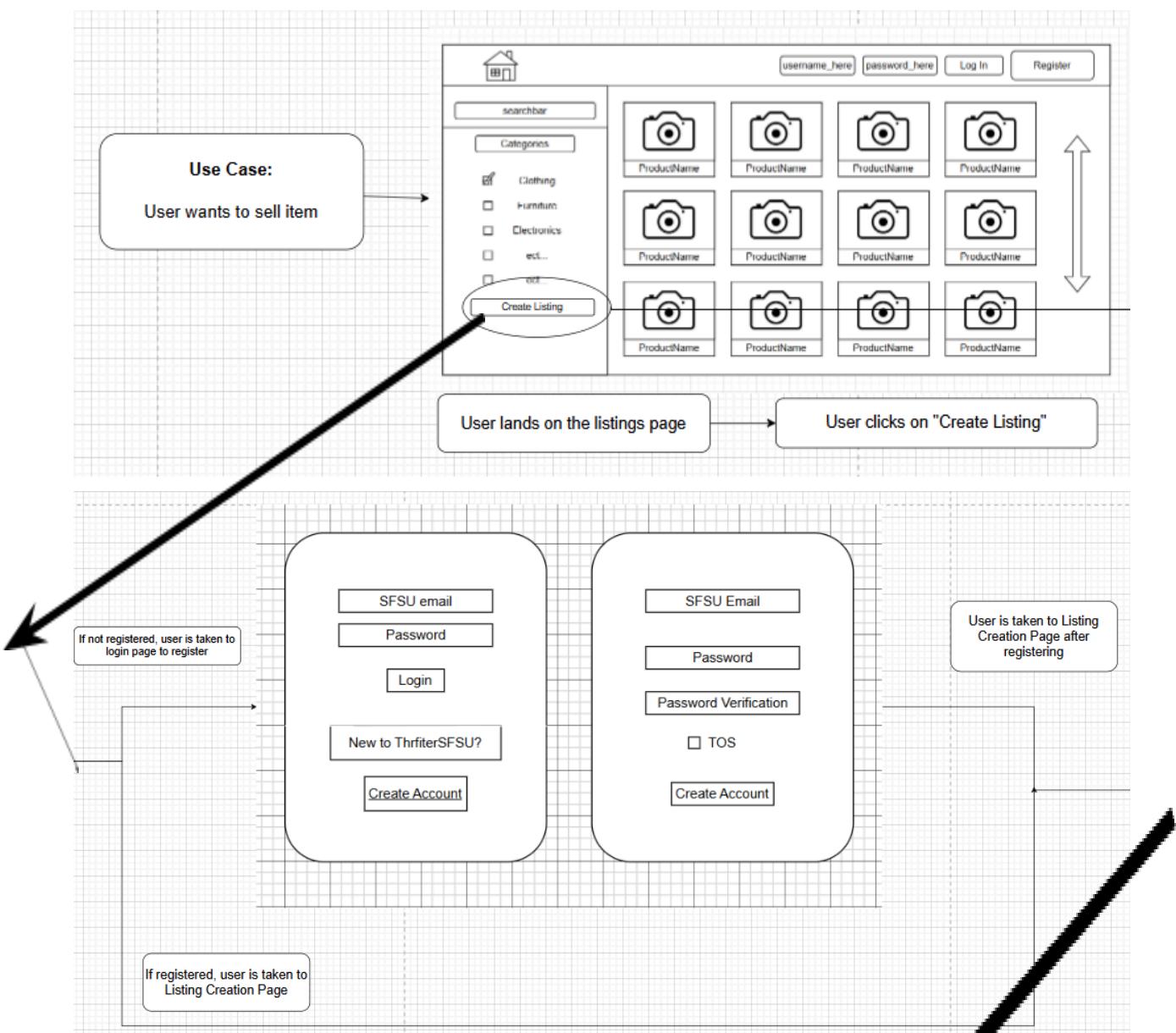


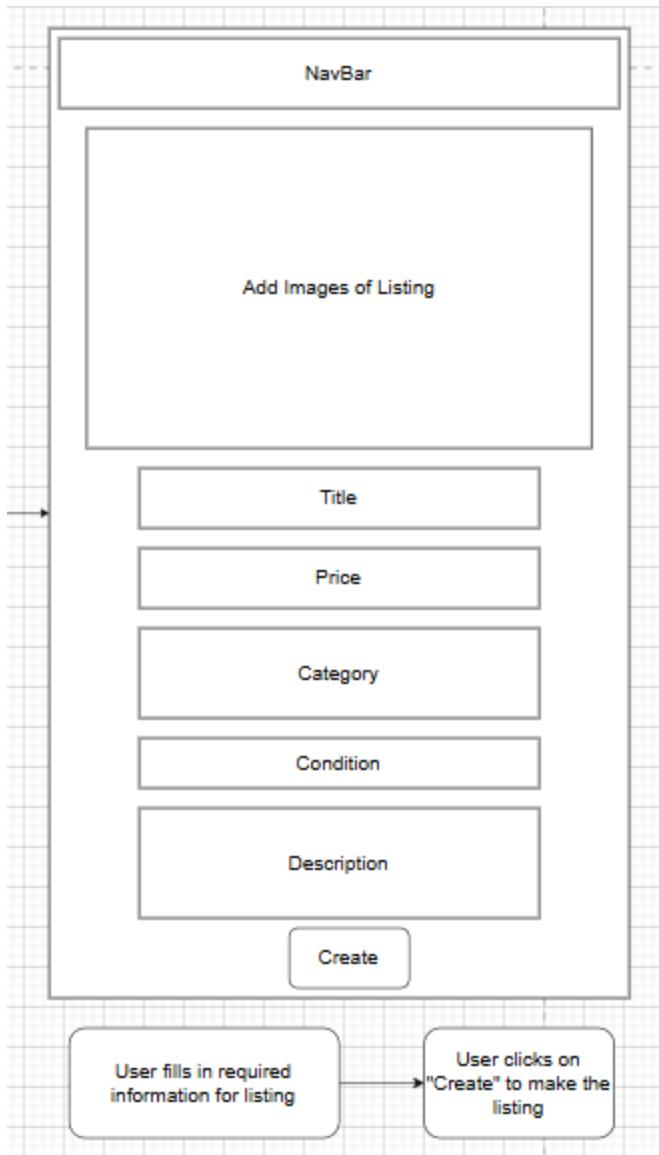


3.



4.





5. High level Architecture, Database Organization summary only

Table and column names follow the naming from Section 2 of our doc:

1. Users

- `user_id` (Primary Key)
- `first_name`
- `last_name`
- `email` (SFSU only)
- `password`
- `phone_number`
- `profile_picture`
- `created_at`
- `is_admin`

2. Products

- `product_id` (Primary Key)
- `seller_id` (Foreign Key to Users)
- `title`
- `description`
- `price`
- `category_id` (Foreign Key to Category)
- `status` (e.g., "available", "sold")
- `Created_at`
- `product_image`
- `thumbnail`

3. Categories

- `category_id` (Primary Key)
- `name`
- `description`

4. Messages

- `message_id` (Primary Key)
- `sender_id` (Foreign Key to Users)
- `receiver_id` (Foreign Key to Users)
- `product_id` (Foreign Key to Products)

- `content`
- `timestamp`

5. Reviews

- `review_id` (Primary Key)
- `reviewer_id` (Foreign Key to Users)
- `reviewed_user_id` (Foreign Key to Users)
- `rating` (e.g., 1–5)
- `comment`
- `timestamp`

6. Transactions

- `transaction_id` (Primary Key)
- `buyer_id` (Foreign Key to Users)
- `seller_id` (Foreign Key to Users)
- `product_id` (Foreign Key to Products)
- `agreed_price`
- `status` (e.g., "completed", "canceled")
- `timestamp`

7. Wishlist

- `wishlist_id` (Primary Key)
- `user_id` (Foreign Key to Users)
- `product_id` (Foreign Key to Products)
- `Date_added`

Media Storage

- Media Type: Images (e.g., profile pictures and product photos)
- Storage Method: Stored directly as BLOBs in the database.
- Justification: Since the project is not operating at production scale, and media files are relatively small, storing them in BLOBs keeps the architecture simpler and easier to manage.
- Fields that will store BLOBs:
 - `Users.profile_picture`
 - `Products` table for product images (e.g., `product_image` as BLOB)
- Other Media:
 - No video/audio is currently supported.
 - No GPS data planned at this stage

Search / Filter Architecture and Implementation

- Search Algorithm/Software: Standard SQL with `%LIKE%` for substring matching (as recommended in architecture slides).
- Searchable Data Fields:
 - Products
 - `title` (text search with `%LIKE%`)
 - `description` (text search with `%LIKE%`)
 - `price` (numeric filter using `BETWEEN`)
 - `status` (dropdown filter: available, sold)
 - `created_at` (used for sorting by recency)
 - Categories
 - `name` (text search with `%LIKE%`)
 - Users
 - `first_name` and `last_name` (text search with `%LIKE%`)
 - `email` (searchable for admin tools only)
- Filtering and Sorting Options:
 - Category-based filtering (via `category_id`)
 - Price range filtering (`price BETWEEN x AND y`)
 - Status filtering (`status = 'available'`)
 - Sorting by `created_at` to show the most recent listings first
- Implementation:
 - SQL queries using `WHERE ... LIKE '%query%'` for relevant fields.
 - Combined with `JOIN` clauses to pull category or user info as needed.
 - We won't be using things like ElasticSearch since this is a small-scale project

Special Algorithms or Processes

- Rating & Review System:
 - Users can leave ratings and comments for others post transaction.
 - A simple average rating can be shown on profiles.

6. Identify actual key risks for your project at this time

- Schedule risks (can you make it, given what you committed and the resources)
 - The front-end lead is getting more work on the side, so he might have less time to work on the website for the next few weeks. It's just a heads-up, but he should still be able to function normally within the team.
 - The meeting time conflicts with dinner for most people, so we may move the meeting to an earlier time.

7. Project management

We met on Friday to assign tasks to everyone. Independent work was done from there, and we used Discord to ask for feedback on work and mini-meetings situationally. We used GitHub projects to assign the wireframe task to the frontend team.

Wireframe #17

The screenshot shows a GitHub project issue titled "Wireframe #17". The issue was opened by Hluui 2 weeks ago and is currently private. The issue body contains a bulleted list of tasks:

- Landing page (Not signed-in)
- Login
- Sign up (email w/ SFSU "verification", password, password verification, terms and conditions checkbox)

Below the list, a note says "Every Page Beyond this point should include the navbar". Another list follows:

- Listing home page (Includes search bar, filters, and listings)
 - [x] Listings Should Include a picture, Name of seller, and price
 - [x] Filters: Category(Sports, Books, Clothing, Food, Travel), Condition (new, like new, used)
- Listing Creation Page (Fields for adding image, title, price, category, and description)
- Individual Listing Page (image, title, price, category, and description)
 - [x] Button to contact seller and look at seller profile
- Profile Page (Has the seller schedule, bio, and area for reviews)
- Chat Screen (Multiple private chats, chat messages between seller and buyer, a method to connect back to listing, dropdown for each listing-related chat)
- Wishlist with listings in it
- Navbar (logo/Home, messages, wishlist, profile)

On the right side of the issue view, there are sections for Assignees, Labels, Type, and Projects. The Assignees section lists "JosephCVA" and "anvan882". The Labels section shows "No labels". The Type section is set to "Task". The Projects section shows a single project named "CSC648-0104-team15" with the status "In Progress".

Looking back, I should have set more concrete deadlines at the end of the initial meeting so we didn't rush to finish everything on the due date. In the future, we will use more of the GitHub project's issues for the milestones and the roadmap on GitHub as a Gantt chart for managing deadlines.

8. Use of GenAI tools like ChatGPT and Copilot for Milestone 2

- What GenAI tool and version did you use/try?
 - GPT-4o
 - Copilot
- List tasks for which you used or tried GenAI tools, and for each, rate how useful it was, on a scale of LOW, MEDIUM, HIGH.
 - List of main data items and entities - Medium
 - High Level Architecture - Medium
- For each task above, briefly explain how you used the tool and what benefit it offered. If the tool was not beneficial, do not use it for M1, but explain why.
 - Gave a list of main data items, and a list of entities after explaining our project. Asked it to help explain them in a short description
 - Asked it the best way to store media for our project is, as well as the best way to search/filter. Gave it a list of fields that can be filtered/searched, and it gave a structured output.
- Provide key examples and prompts.
 - We gave it a list of main entities like User, Product, Category, Message, Review, Transaction, Wishlist, and asked it to describe and give us some attributes
 - A full list of searchable fields across all tables, organized clearly, and describe how we'll implement search and filters using SQL?
- Comment on anything else you found useful.
 - It gave us what we already knew, just extremely quickly saving us on time.

9. Team Lead Checklist to be completed by the team lead

- So far, all team members are fully engaged and attending team sessions when required
 - DONE
- Team is ready and able to use the chosen back and front-end frameworks, and those who need to learn are working on learning and practicing
 - DONE
- The team reviewed suggested resources before drafting Milestone 2
 - DONE
- The team lead checked the Milestone 2 document for quality, completeness, formatting, and compliance with instructions before the submission
 - DONE
- The team lead ensured that all team members read the final Milestone 2 document and agreed/understood it before submission.
 - DONE
- The team shared and discussed the experience with GenAI tools among themselves
 - DONE

SW Engineering CSC648-848 Spring 2025

ThriftAtSFSU

Milestone 2 Part 1

Team 15

Hilary Lui (hlui@sfsu.edu)	Team Lead
Annison Van	Frontend Lead
Sid Padmanabhuni	Backend Lead & GitHub master
Joseph Alhambra	Team Member Frontend
Joseph Shur	Team Member Backend

Date Submitted	3/25/2025
Date Revised	5/19/2025

1. Executive Summary

In today's fast-paced college environment, affordability and convenience are key student concerns. ThriftAtSFSU is a dedicated resale platform designed exclusively for San Francisco State University (SFSU) students, providing a secure and efficient way to buy and sell second-hand goods within the campus community.

Unlike generic resale platforms, ThriftAtSFSU is tailored to meet the specific needs of SFSU students, ensuring a trusted marketplace that fosters sustainability and affordability.

To create a seamless user experience, ThriftAtSFSU offers a range of features designed with students in mind. Users can easily browse and search for items using advanced filters for academic-specific needs, such as textbooks and dorm supplies. A secure in-site messaging system ensures safe communication between buyers and sellers, requiring SFSU email verification. Listing items is streamlined for students, featuring predefined academic categories for quicker uploads. Sellers benefit from a listing-based chatting system to keep track of their items and also the ability to put their schedule on their profile to make coordinating pick-up times easier. To maintain a safe and trustworthy marketplace, the platform includes strict item and user moderation managed by administrators. Additionally, ThriftAtSFSU is fully mobile-responsive, allowing students to access the marketplace conveniently from any device.

What sets ThriftAtSFSU apart is its exclusive focus on the SFSU community, ensuring a localized and student-friendly marketplace that prioritizes security, convenience, and affordability. This project directly addresses the need for a reliable, campus-specific resale network, fostering a culture of sustainability while making essential goods more accessible to students. Our team comprises passionate SFSU students committed to enhancing campus life through innovative solutions. With firsthand experience of students' challenges in accessing affordable goods, we are dedicated to making ThriftAtSFSU the go-to platform for peer-to-peer resale within our university. By supporting this project, you invest in a practical and impactful solution that benefits the entire SFSU community.

2. List of main data items and entities

1. User
 - a. Description: Represents a registered individual on the platform (student or professor with an SFSU email). Users can act as buyers, sellers, or both.
 - b. Key Attributes:
 - i. user_id (PK)
 - ii. first_name
 - iii. last_name
 - iv. email (SFSU only, unique)
 - v. password
 - vi. phone_number
 - vii. profile_picture
 - viii. bio
 - ix. created_at
 - x. is_admin
2. Product
 - a. Description: Items listed for sale by users. Includes key details such as title, price, description, and condition.
 - b. Key Attributes:
 - i. product_id (PK)
 - ii. seller_id (FK to User)
 - iii. title
 - iv. description
 - v. price
 - vi. category_id (FK to Category)
 - vii. condition
 - viii. status (e.g. available, sold)
 - ix. created_at
3. Category
 - a. Description: Organizes products into logical categories (e.g. Electronics, Furniture, Books) for easier browsing and searching.
 - b. Key Attributes:
 - i. category_id (PK)
 - ii. name
 - iii. description
4. ProductImages
 - a. Description: Stores one or more images associated with a product listing.
 - b. Key Attributes:
 - i. image_id (PK)

- ii. product_id (FK to Product)
- iii. image_data
- iv. uploaded_at
- v. image_order

5. Message

a. Description: Stores private messages exchanged between users regarding specific products. Useful for inquiries, negotiations, and meetups.

b. Key Attributes:

- i. message_id (PK)
- ii. sender_id (FK to User)
- iii. receiver_id (FK to User)
- iv. product_id (FK to Product)
- v. content
- vi. timestamp

6. Review

a. Description: Ratings and feedback left by users after interacting with a buyer or seller to build reputation and trust.

b. Key Attributes:

- i. review_id (PK)
- ii. reviewer_id (FK to User)
- iii. reviewed_user_id (FK to User)
- iv. rating (e.g. 1-5)
- v. comment
- vi. timestamp

7. Reports

a. Description: Tracks reports made by users against other users or products for moderation purposes.

b. Key Attributes:

- i. report_id (PK)
- ii. reporter_id (FK to User)
- iii. reported_user_id (FK to User)
- iv. reported_product_id (FK to Product)
- v. reason
- vi. status (e.g. pending, resolved, dismissed)
- vii. created_at
- viii. resolved_at

8. Wishlist

a. Description: Allows users to save items they're interested in for future consideration.

b. Key Attributes:

- i. wishlist_id (PK)

- ii. user_id (FK to User)
- iii. product_id (FK to Product)
- iv. date_added

9. UserAvailability

- a. Description: Stores a user's available times for potential meetups or communication windows.
- b. Key Attributes:
 - i. availability_id (PK)
 - ii. user_id (FK to User)
 - iii. day_of_week (e.g. Monday)
 - iv. time_slot (e.g. 6:00, 6:30)
 - v. is_available
 - vi. last_updated

3. Functional Requirements

Priority 1 – must-have:

1. Users of all levels of authority and privileges shall be able to access the app and see listings and reviews of associated listings.
2. Unregistered users shall be given the opportunity to create an account, which will allow them to become registered users and gain access to all associated privileges and benefits.
3. Unregistered users shall be able to search based on text search and filter by category.
4. Registered users shall agree to the terms and conditions upon account creation.
5. Registered users can create listings and reviews for other users.
6. Registered users shall be able to create chat logs with other registered users.
7. Admin shall be able to create listings, regardless of ownership.

Priority 2 – desired:

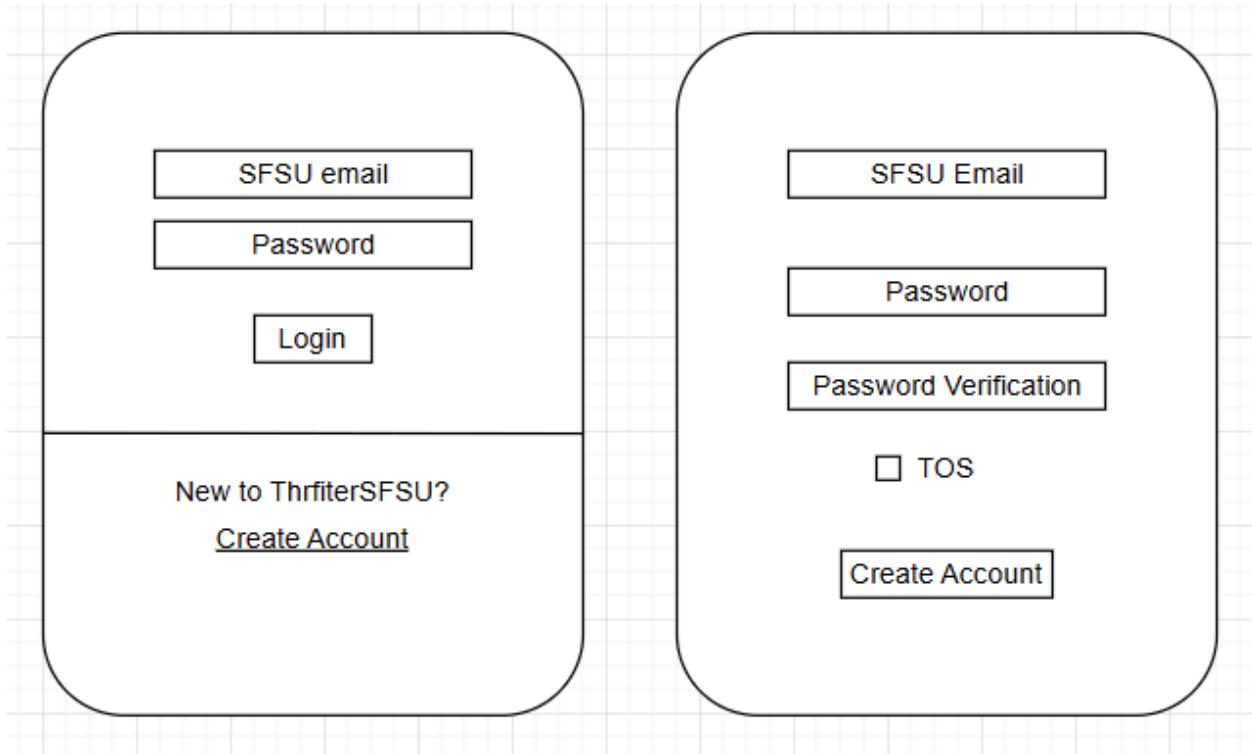
8. Registered users shall be able to edit and change their user profile.
9. Registered users shall be able to block or report other users and listings, automatically notifying the admin to enact moderation.
10. Admin shall be able to review reported listings and users.
11. Admin shall be able to ban registered users.
12. Admin shall be able to delete and edit listings, regardless of ownership, to moderate the app.

Priority 3 – opportunistic:

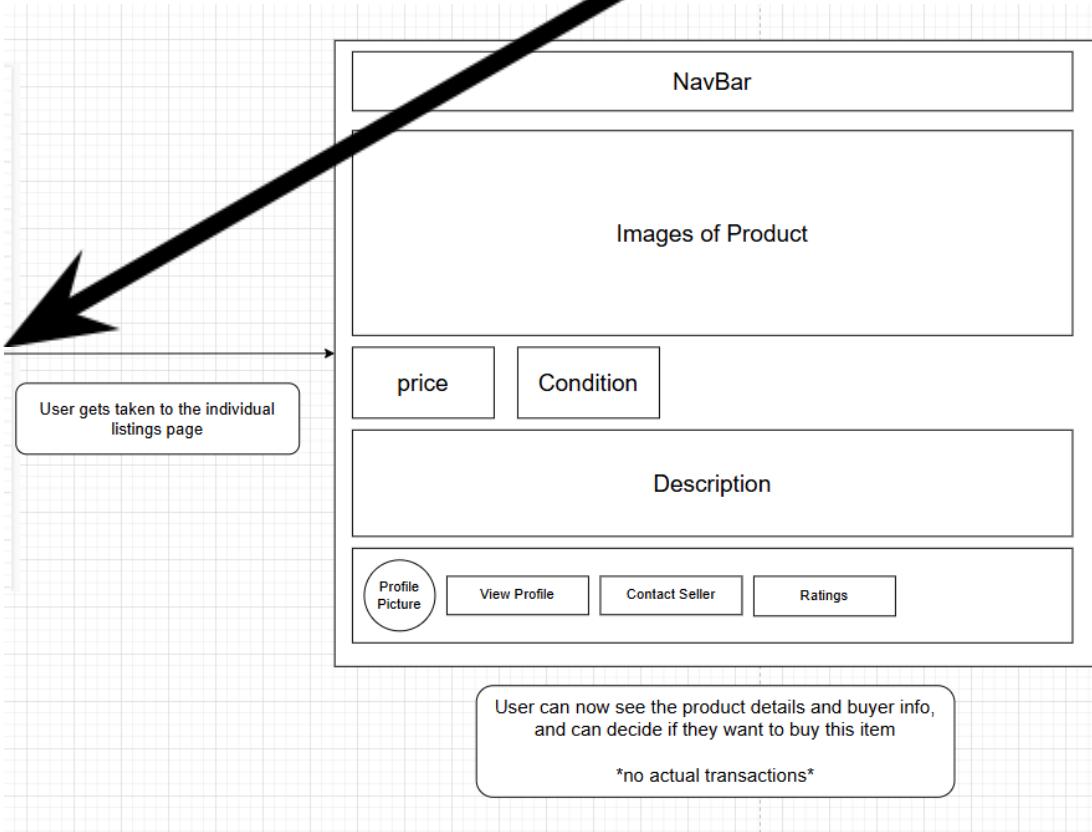
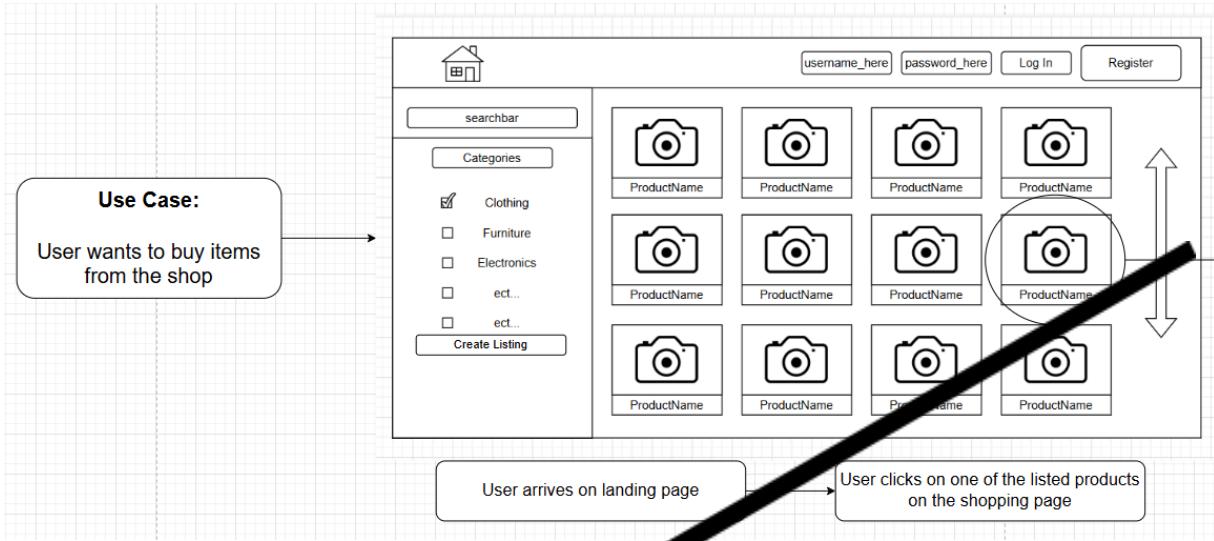
13. Unregistered users shall have access to a settings tab, where they can change the app's appearance to become a vision-friendly dark mode and set other settings.
14. Registered users shall have access to a more robust settings tab, complete with account owner features that enable more control over their personal user experience.
15. Admin shall be able to create, delete, and edit all categories for searching on the website.

4. UI Storyboards for each main use case

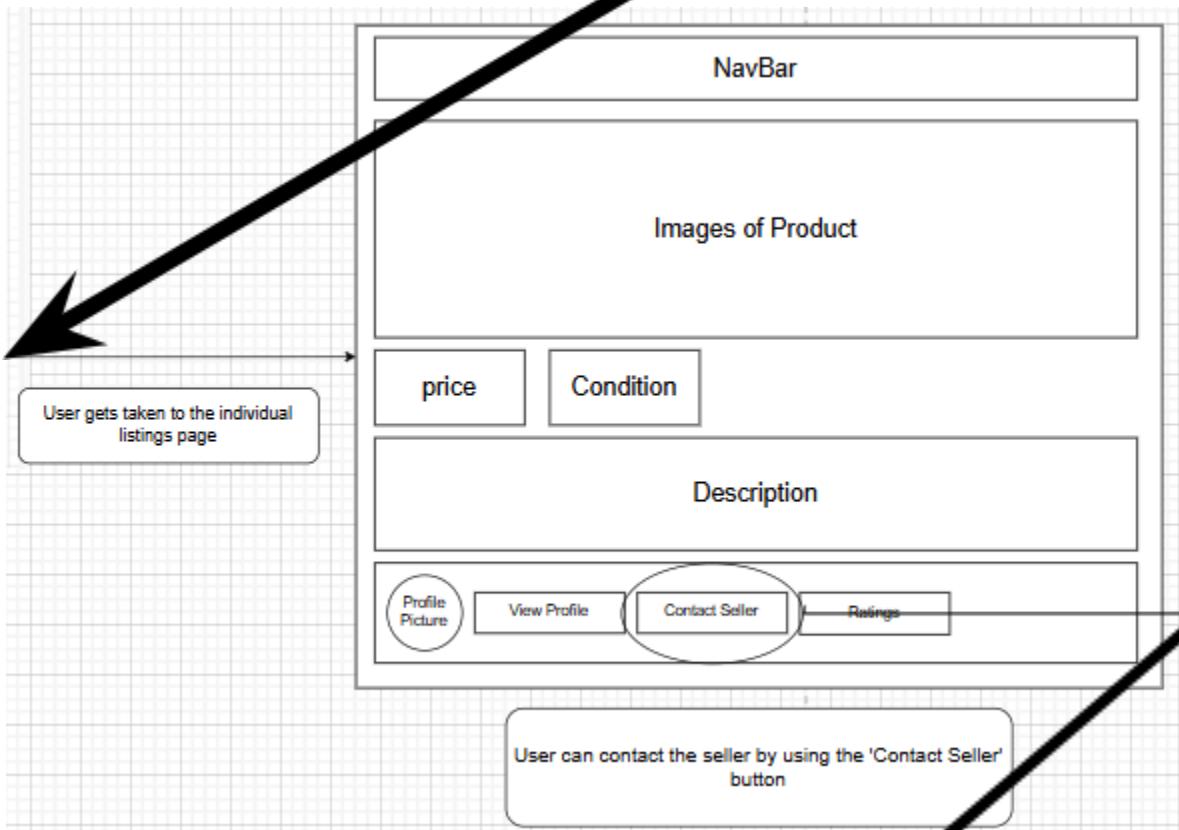
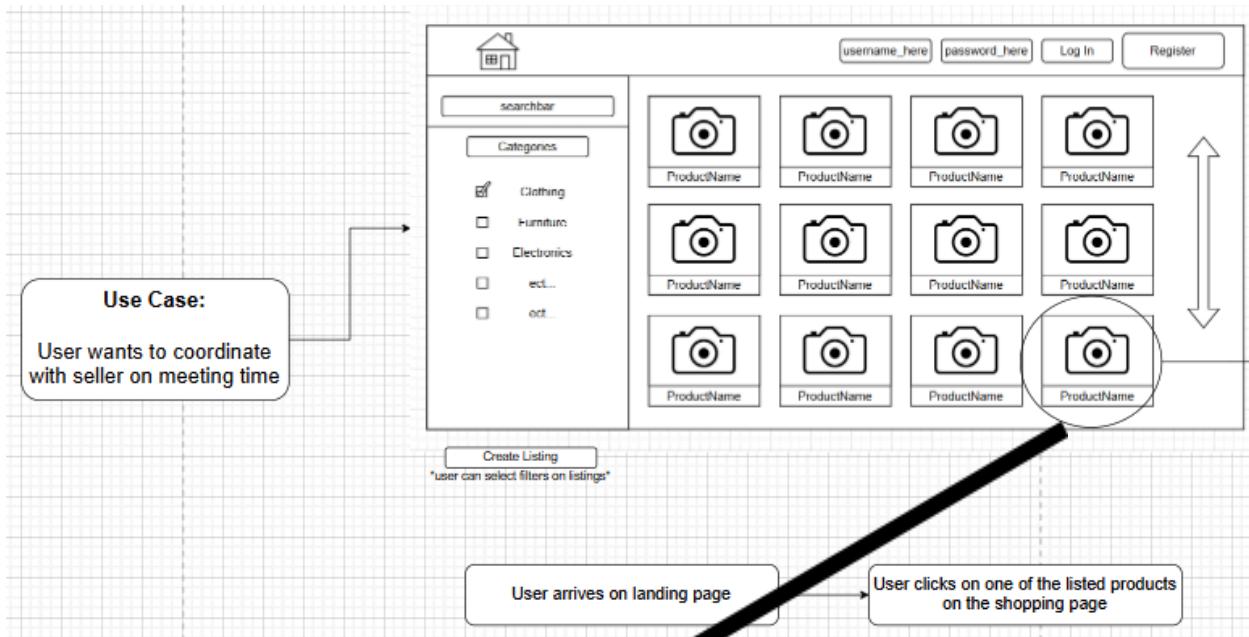
Login and registration mockups

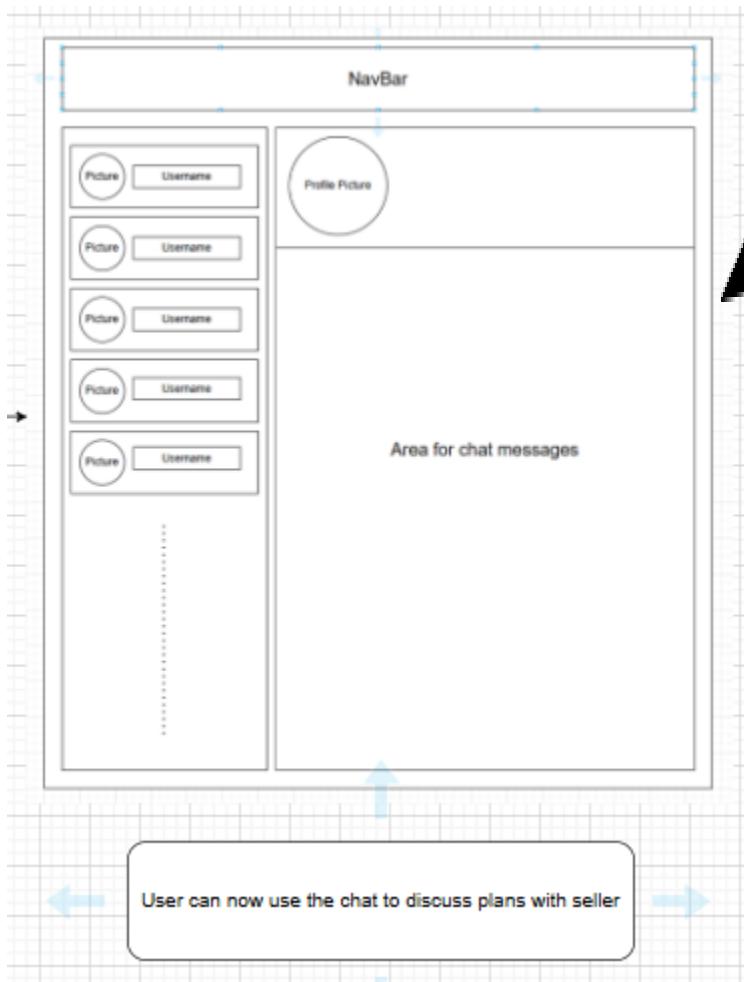
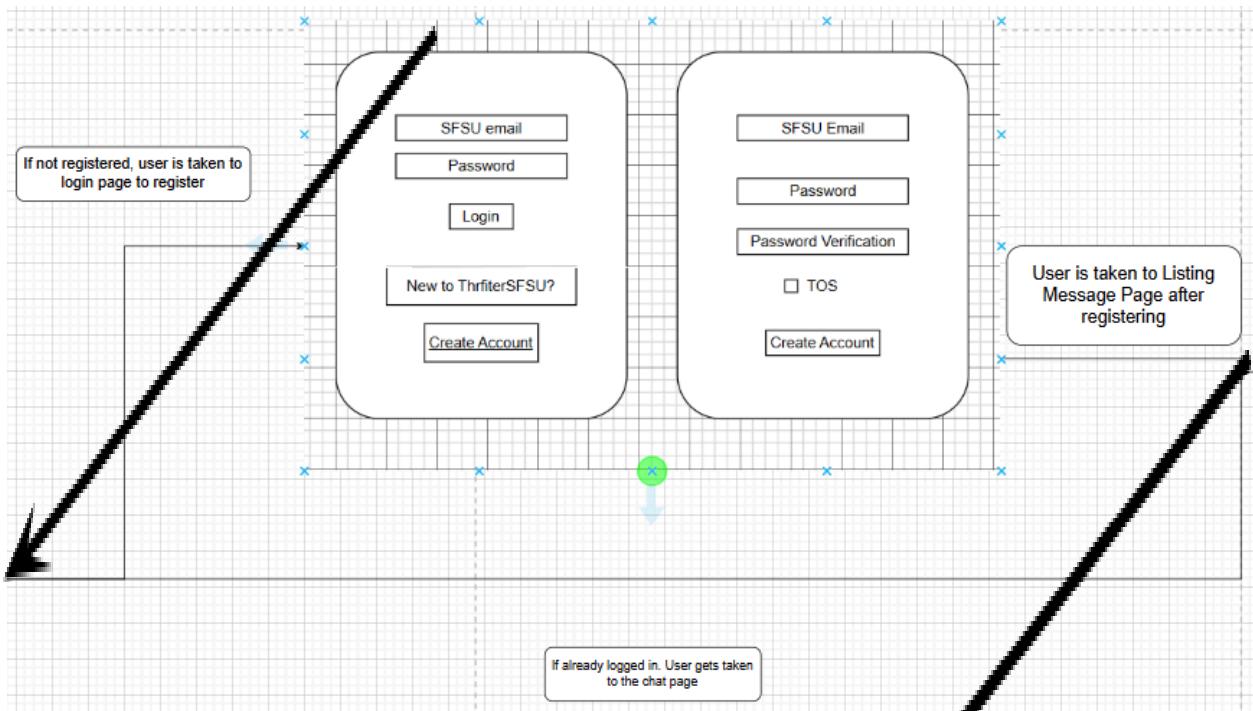


1.

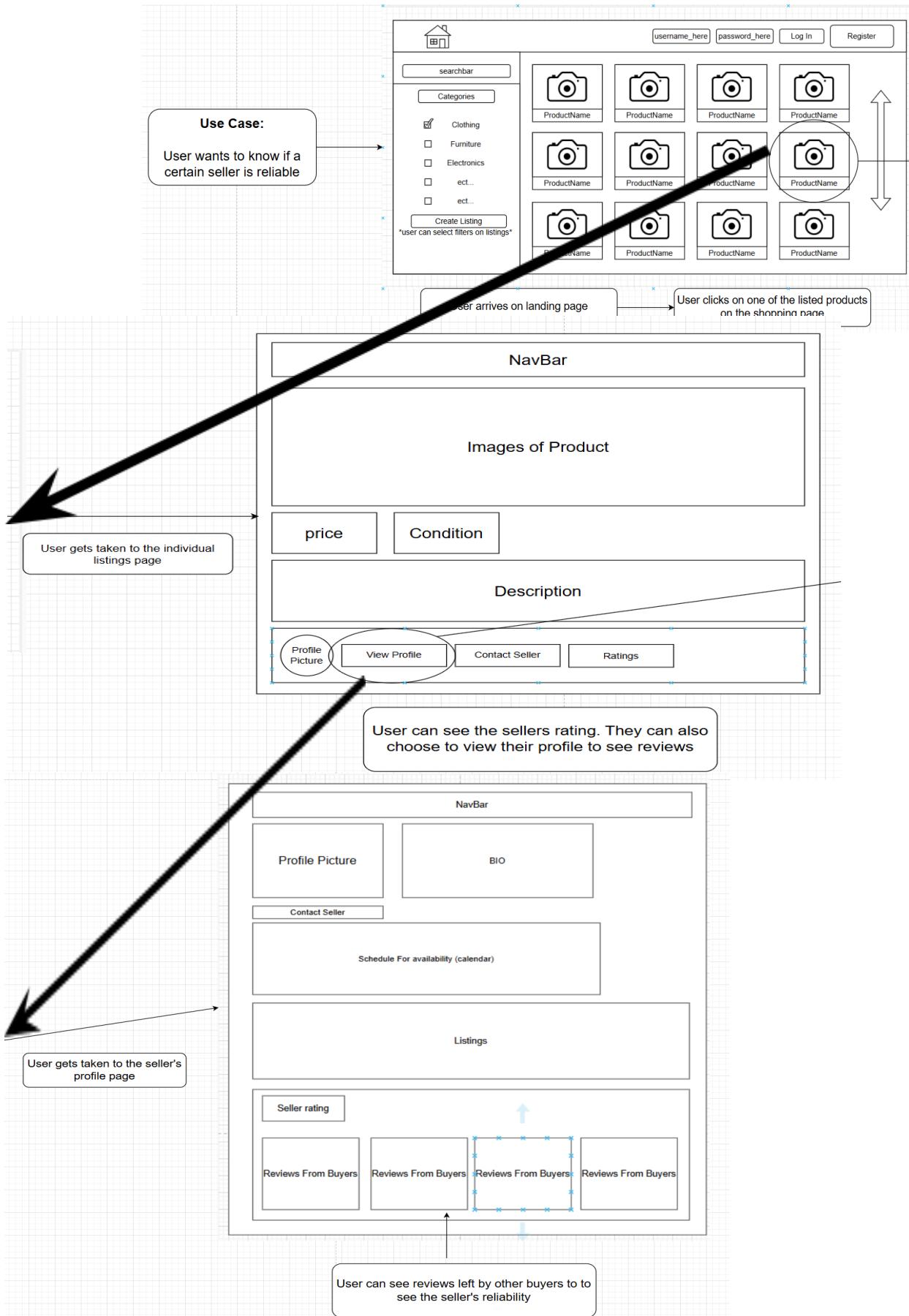


2.

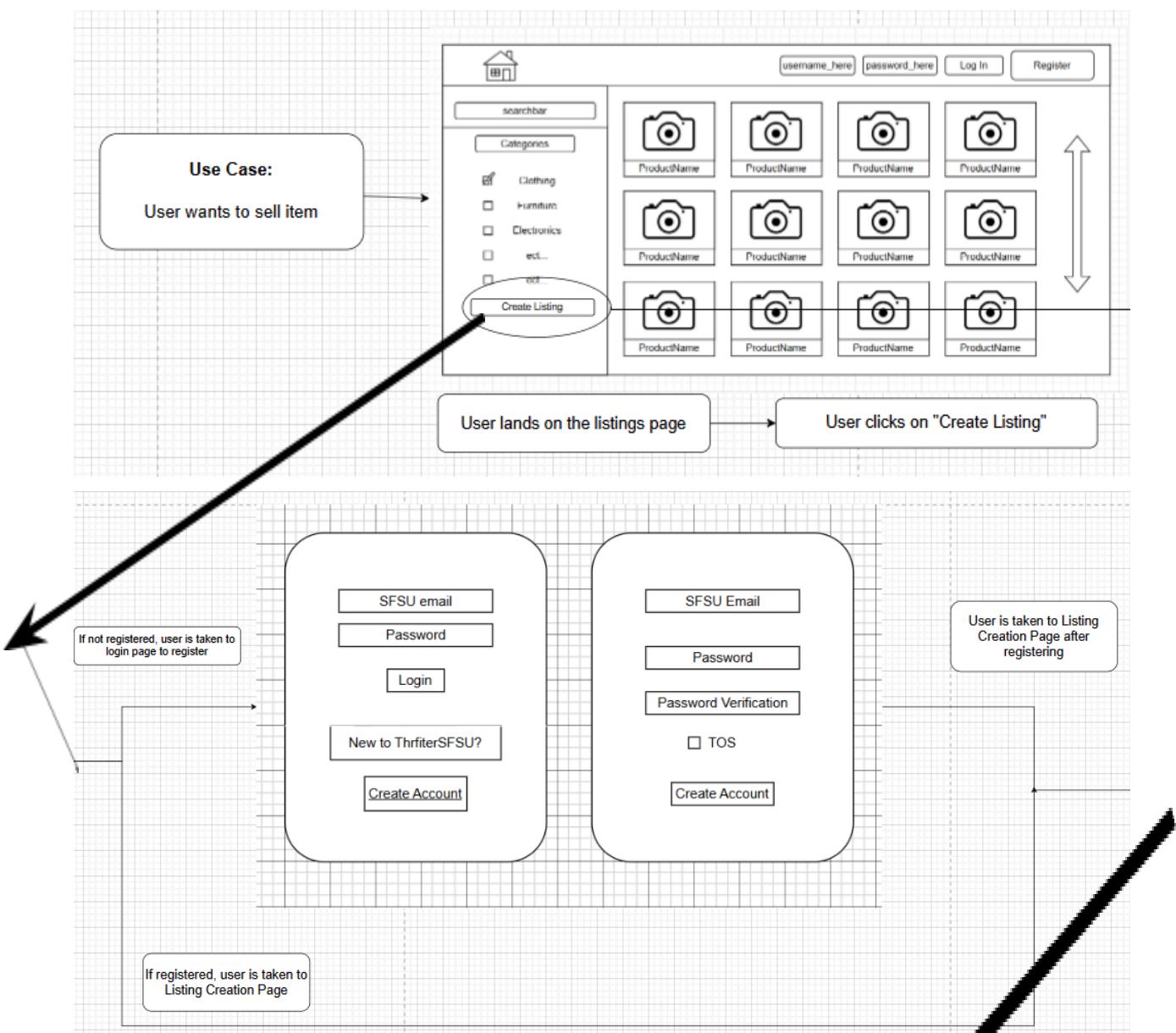


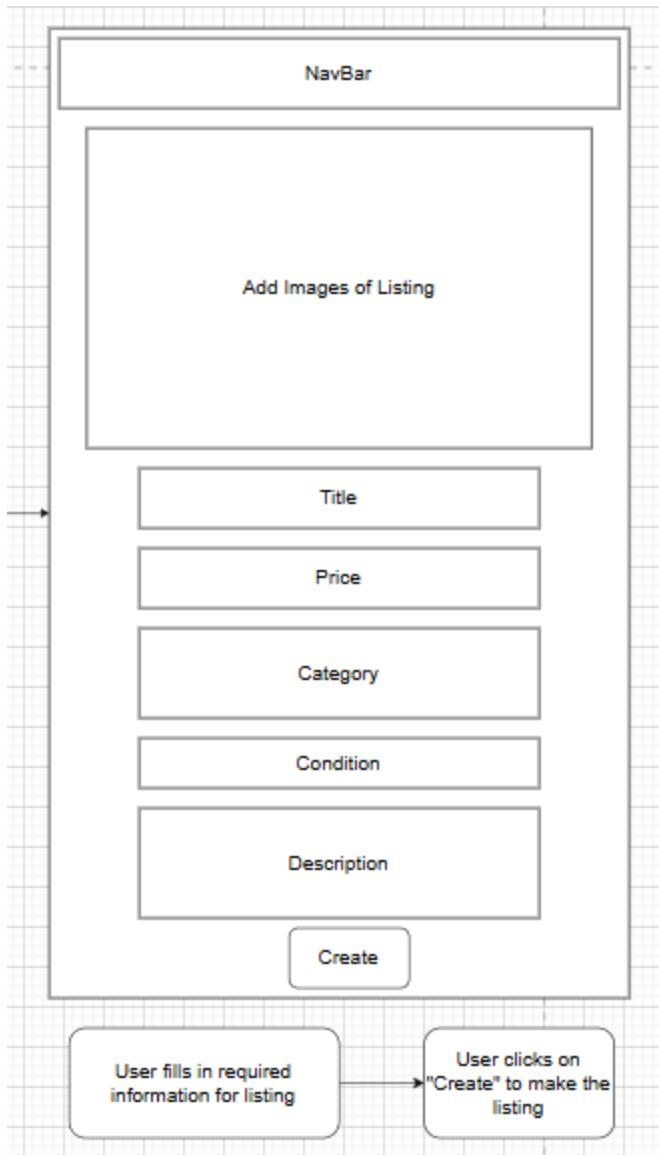


3.



4.





5. High level Architecture, Database Organization summary only

Table and column names follow the naming from Section 2 of our doc:

1. Users

- `user_id` (Primary Key)
- `first_name`
- `last_name`
- `email` (SFSU only)
- `password`
- `phone_number`
- `profile_picture`
- `created_at`
- `is_admin`

2. Products

- `product_id` (Primary Key)
- `seller_id` (Foreign Key to Users)
- `title`
- `description`
- `price`
- `category_id` (Foreign Key to Category)
- `status` (e.g., "available", "sold")
- `Created_at`
- `product_image`
- `thumbnail`

3. Categories

- `category_id` (Primary Key)
- `name`
- `description`

4. Messages

- `message_id` (Primary Key)
- `sender_id` (Foreign Key to Users)
- `receiver_id` (Foreign Key to Users)
- `product_id` (Foreign Key to Products)

- `content`
- `timestamp`

5. Reviews

- `review_id` (Primary Key)
- `reviewer_id` (Foreign Key to Users)
- `reviewed_user_id` (Foreign Key to Users)
- `rating` (e.g., 1–5)
- `comment`
- `timestamp`

6. Transactions

- `transaction_id` (Primary Key)
- `buyer_id` (Foreign Key to Users)
- `seller_id` (Foreign Key to Users)
- `product_id` (Foreign Key to Products)
- `agreed_price`
- `status` (e.g., "completed", "canceled")
- `timestamp`

7. Wishlist

- `wishlist_id` (Primary Key)
- `user_id` (Foreign Key to Users)
- `product_id` (Foreign Key to Products)
- `Date_added`

Media Storage

- Media Type: Images (e.g., profile pictures and product photos)
- Storage Method: Stored directly as BLOBs in the database.
- Justification: Since the project is not operating at production scale, and media files are relatively small, storing them in BLOBs keeps the architecture simpler and easier to manage.
- Fields that will store BLOBs:
 - `Users.profile_picture`
 - `Products` table for product images (e.g., `product_image` as BLOB)
- Other Media:
 - No video/audio is currently supported.
 - No GPS data planned at this stage

Search / Filter Architecture and Implementation

- Search Algorithm/Software: Standard SQL with `%LIKE%` for substring matching (as recommended in architecture slides).
- Searchable Data Fields:
 - Products
 - `title` (text search with `%LIKE%`)
 - `description` (text search with `%LIKE%`)
 - `price` (numeric filter using `BETWEEN`)
 - `status` (dropdown filter: available, sold)
 - `created_at` (used for sorting by recency)
 - Categories
 - `name` (text search with `%LIKE%`)
 - Users
 - `first_name` and `last_name` (text search with `%LIKE%`)
 - `email` (searchable for admin tools only)
- Filtering and Sorting Options:
 - Category-based filtering (via `category_id`)
 - Price range filtering (`price BETWEEN x AND y`)
 - Status filtering (`status = 'available'`)
 - Sorting by `created_at` to show the most recent listings first
- Implementation:
 - SQL queries using `WHERE ... LIKE '%query%'` for relevant fields.
 - Combined with `JOIN` clauses to pull category or user info as needed.
 - We won't be using things like ElasticSearch since this is a small-scale project

Special Algorithms or Processes

- Rating & Review System:
 - Users can leave ratings and comments for others post transaction.
 - A simple average rating can be shown on profiles.

6. Identify actual key risks for your project at this time

- Schedule risks (can you make it, given what you committed and the resources)
 - The front-end lead is getting more work on the side, so he might have less time to work on the website for the next few weeks. It's just a heads-up, but he should still be able to function normally within the team.
 - The meeting time conflicts with dinner for most people, so we may move the meeting to an earlier time.

7. Project management

We met on Friday to assign tasks to everyone. Independent work was done from there, and we used Discord to ask for feedback on work and mini-meetings situationally. We used GitHub projects to assign the wireframe task to the frontend team.

Wireframe #17

The screenshot shows a GitHub project issue titled "Wireframe #17". The issue was opened by Hluui 2 weeks ago and is currently private. The issue details are as follows:

- Assignees:** JosephCVA, anvan882
- Labels:** No labels
- Type:** Task
- Projects:** CSC648-0104-team15 (Status: In Progress)
- Priority:** Choose an option
- Size:** Choose an option
- Estimate:** Enter number...
- Iteration:** Choose an iteration
- Start date:** No date
- End date:** Mar 25, 2025

The issue body contains two sections of tasks:

- Landing page (Not signed-in)**: Checked
- Login**: Checked
- Sign up (email w/ SFSU "verification", password, password verification, terms and conditions checkbox)**: Checked

Every Page Beyond this point should include the navbar

- Listing home page (Includes search bar, filters, and listings)**: Checked
 - [x] Listings Should Include a picture, Name of seller, and price
 - [x] Filters: Category(Sports, Books, Clothing, Food, Travel), Condition (new, like new, used)
- Listing Creation Page (Fields for adding image, title, price, category, and description)**: Checked
- Individual Listing Page (image, title, price, category, and description)**:
 - [x] Button to contact seller and look at seller profile
- Profile Page (Has the seller schedule, bio, and area for reviews)**: Checked
- Chat Screen (Multiple private chats, chat messages between seller and buyer, a method to connect back to listing, dropdown for each listing-related chat)**:
 - [x] Chat Screen (Multiple private chats, chat messages between seller and buyer, a method to connect back to listing, dropdown for each listing-related chat)
- Wishlist with listings in it**:
 - [x] Wishlist with listings in it
- Navbar (logo/Home, messages, wishlist, profile)**:
 - [x] Navbar (logo/Home, messages, wishlist, profile)

At the bottom of the issue body are buttons for "Create sub-issue" and a smiley face icon.

Looking back, I should have set more concrete deadlines at the end of the initial meeting so we didn't rush to finish everything on the due date. In the future, we will use more of the GitHub project's issues for the milestones and the roadmap on GitHub as a Gantt chart for managing deadlines.

8. Use of GenAI tools like ChatGPT and Copilot for Milestone 2

- What GenAI tool and version did you use/try?
 - GPT-4o
 - Copilot
- List tasks for which you used or tried GenAI tools, and for each, rate how useful it was, on a scale of LOW, MEDIUM, HIGH.
 - List of main data items and entities - Medium
 - High Level Architecture - Medium
- For each task above, briefly explain how you used the tool and what benefit it offered. If the tool was not beneficial, do not use it for M1, but explain why.
 - Gave a list of main data items, and a list of entities after explaining our project. Asked it to help explain them in a short description
 - Asked it the best way to store media for our project is, as well as the best way to search/filter. Gave it a list of fields that can be filtered/searched, and it gave a structured output.
- Provide key examples and prompts.
 - We gave it a list of main entities like User, Product, Category, Message, Review, Transaction, Wishlist, and asked it to describe and give us some attributes
 - A full list of searchable fields across all tables, organized clearly, and describe how we'll implement search and filters using SQL?
- Comment on anything else you found useful.
 - It gave us what we already knew, just extremely quickly saving us on time.

9. Team Lead Checklist to be completed by the team lead

- So far, all team members are fully engaged and attending team sessions when required
 - DONE
- Team is ready and able to use the chosen back and front-end frameworks, and those who need to learn are working on learning and practicing
 - DONE
- The team reviewed suggested resources before drafting Milestone 2
 - DONE
- The team lead checked the Milestone 2 document for quality, completeness, formatting, and compliance with instructions before the submission
 - DONE
- The team lead ensured that all team members read the final Milestone 2 document and agreed/understood it before submission.
 - DONE
- The team shared and discussed the experience with GenAI tools among themselves
 - DONE

3. Product Screenshots:

Thrift@SFSU

All Search products...

About the Team •

Price Range \$ 10 - \$ 1399 **FILTERS**

Showing 20 results (All Products)

Condition Very Worn Used Fairly Used Good Condition Great Condition

Seller Rating

Posted After mm / dd / yyyy

Apply

Large Flat Screen TV \$400.00
 Butterfly Training Practic... \$15.00
 Jeans \$40.00
 Sneakers \$60.00
 Hoodie \$35.00
 Notebook Set \$12.50

Office Mouse \$29.99
 Monitor 27" \$279.99
 JBL Speaker \$45.99
 Winter Coat \$85.00
 iPhone 12 \$599.99
 White Shirt \$15.00

Python Programming \$25.00
 The Great Gatsby \$10.00
 Kindle Paperwhite \$89.99
 AirPods Pro \$159.00
 MacBook Pro 2020 \$1399.00
 Samsung Galaxy S21 \$499.99

<http://192.168.17.89:5000> © 2025 SFSU Thrift Market - THIS WEBSITE IS A PROJECT FOR CSC648

Thrift@SFSU

All Search products...

About the Team •

Upload Images JPG or PNG

No files chosen

Autofill Form Using LLM

Title

Price (\$)

Category --Select a category--

Condition --Select condition--

Description 0/500 characters

Create

© 2025 SFSU Thrift Market - THIS WEBSITE IS A PROJECT FOR CSC648

Thrift@SFSU

All Search products... 

About the Team • 

My Wishlist

Showing 5 items in your wishlist



Large Flat Screen TV — \$400.00



Butterfly Training Practice Knife — \$15.00



JBL Speaker — \$45.99



AirPods Pro — \$159.00



MacBook Pro 2020 — \$1399.00

© 2025 SFSU Thrift Market - THIS WEBSITE IS A PROJECT FOR CSC848

Thrift@SFSU

All Search products... 

About the Team • 

Messages  Hillary Lui

Hillary Lui I'm interested in your listing: But...

I'm interested in your listing: Butterfly Training Practice Knife (\$15.00) 

Type a message... 

Debug

Thrift@SFSU

All Search products... 

About the Team 

 **Sid Padmanabhu**

Joined: May 21 2025
No reviews yet

[Bio !\[\]\(7182297cfd6b1d0d173443cd3bd6c8b0_img.jpg\)](#)

No bio provided.

Availability

	Su	Mo	Tu	We	Th	Fr	Sa
7:00 AM							
8:00 AM							
9:00 AM							
10:00 AM							
11:00 AM							
12:00 PM							
1:00 PM							
2:00 PM							
3:00 PM							
4:00 PM							
5:00 PM							
6:00 PM							
7:00 PM							
8:00 PM							
9:00 PM							
10:00 PM							
11:00 PM							

SFSU Campus Map

[View larger map](#)



© 2025 SFSU Thrift Market - THIS WEBSITE IS A PROJECT FOR CSC648

Save Availability

Thrift@SFSU

All Search products... 

About the Team 

Sid Padmanabhu

Joined: May 21 2025
No reviews yet

[Bio !\[\]\(f87eae752ebe9764e6fed26bbdfbdd75_img.jpg\)](#)

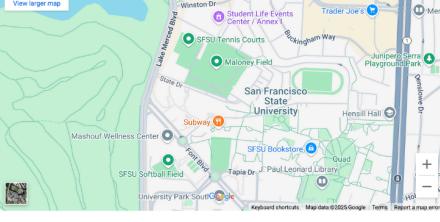
No bio provided.

Availability

	9:00 PM	10:00 PM	11:00 PM
9:00 PM			
10:00 PM			
11:00 PM			

SFSU Campus Map

[View larger map](#)



© 2025 SFSU Thrift Market - THIS WEBSITE IS A PROJECT FOR CSC648

Listings by Sid

No listings available

Reviews for Sid

No reviews yet.

Thrift@SFSU

All Search products... 

About the Team • 

Admin Dashboard

Users Management Post Management Reports

ID	Reporter	Reported User/Listing	Reason	Date	Status	Actions
1	Hillary Lui	White Shirt	This is a scam.	2025-05-21	Pending	 Dismiss  Delete

© 2025 SFSU Thrift Market - THIS WEBSITE IS A PROJECT FOR CSC648

Thrift@SFSU

All Search products... 

About the Team • 

Admin Dashboard

Users Management Post Management Reports

ID	Title	Seller	Price	Date Posted	Status	Actions
21	PNY GeForce RTX 5090 Graphics Card 32GB	Hilary Lui	\$1500.00	2025-05-21	Pending approval	 View  Delete

© 2025 SFSU Thrift Market - THIS WEBSITE IS A PROJECT FOR CSC648

Thrift@SFSU

All Search products... 

About the Team • 

Admin Dashboard

Users Management Post Management Reports

ID	Name	Email	Phone	Joined Date	Admin Status	Actions
1	Alice Smith	alice@example.com	1234567890	2025-05-21	User	 View Profile  Delete
2	Bob Johnson	bob@example.com	0987654321	2025-05-21	User	 View Profile  Delete
3	Hilary Lui	hlui@sfsu.edu	(123) 456-7890	2025-05-21	User	 View Profile  Delete
4	Sid Padmanabhu	spadmanabhu@sfsu.edu	(408) 819-6180	2025-05-21	Admin	 View Profile

© 2025 SFSU Thrift Market - THIS WEBSITE IS A PROJECT FOR CSC648

4. Database Organization:

The screenshot shows the MySQL Workbench interface with the following details:

- Left Panel (Schema Tree):** Shows the schema tree for the 'local' database, including Tables (Categories, Messages, ProductImages, Products, Reports, Reviews, UserAvailability, Users, Wishlist), Views, Stored Procedures, Functions, and sys.
- Central Panel (Query Editor):** Displays the query `SELECT * FROM local.Categories;` and its results. The results show 19 rows of data from the Categories table.
- Right Panel (Toolbars and Help):** Includes a toolbar with various icons, a context help message about automatic context help being disabled, and a sidebar with links to Result Grid, Form Editor, Field Types, Query Stats, and Execution Plan.
- Bottom Panel (Action Output):** Shows the execution history with three entries.

The screenshot shows the MySQL Workbench interface with the following details:

- Left Panel (Schema Tree):** Shows the schema tree for the 'local' database, including Tables (Categories, Messages, ProductImages, Products, Reports, Reviews, UserAvailability, Users, Wishlist), Views, Stored Procedures, Functions, and sys.
- Central Panel (Query Editor):** Displays the query `SELECT * FROM local.Messages;` and its results. The results show 1 row of data from the Messages table.
- Right Panel (Toolbars and Help):** Includes a toolbar with various icons, a context help message about automatic context help being disabled, and a sidebar with links to Result Grid, Form Editor, Field Types, Query Stats, and Execution Plan.
- Bottom Panel (Action Output):** Shows the execution history with four entries.

linux-vm1

Administration Schemas Categories Messages ProductImages Products Reports Reviews UserAvailability Users Wishlist Views Stored Procedures Functions sys

SCHEMAS Filter objects

local Tables Categories Messages ProductImages Products Reports Reviews UserAvailability Users Wishlist Views Stored Procedures Functions sys

Object Info Session No object selected

1 SELECT * FROM local.ProductImages;

Result Grid Filter Rows: Search Edit: Export/Import: Result Grid Form Editor Field Types Query Stats Execution Plan

image_id	product_id	image_data	uploaded_at	image_order
1	1	BLOB	2025-05-21 02:07:15	0
3	3	BLOB	2025-05-21 02:07:15	0
4	4	BLOB	2025-05-21 02:07:15	0
5	5	BLOB	2025-05-21 02:07:15	0
6	6	BLOB	2025-05-21 02:07:15	0
7	7	BLOB	2025-05-21 02:07:15	0
8	8	BLOB	2025-05-21 02:07:15	0
9	9	BLOB	2025-05-21 02:07:15	0
10	10	BLOB	2025-05-21 02:07:15	0
11	11	BLOB	2025-05-21 02:07:15	0
12	12	BLOB	2025-05-21 02:07:15	0
13	13	BLOB	2025-05-21 02:07:15	0
14	14	BLOB	2025-05-21 02:07:15	0
15	15	BLOB	2025-05-21 02:07:15	0
16	16	BLOB	2025-05-21 02:07:15	0
17	17	BLOB	2025-05-21 02:07:15	0
18	18	BLOB	2025-05-21 02:07:15	0
19	19	BLOB	2025-05-21 02:09:06	0
20	20	BLOB	2025-05-21 02:25:41	0
21	21	BLOB	2025-05-21 06:34:29	0
HULL	HULL	HULL	HULL	HULL

ProductImages 1 Apply Revert

Action Output

Time	Action	Response	Duration / Fetch Time
4	23:39:25 SELECT * FROM local.Categories LIMIT 0, 1000	19 row(s) returned	0.0045 sec / 0.00000...
5	23:39:35 SELECT * FROM local.Messages LIMIT 0, 1000	1 row(s) returned	0.0045 sec / 0.00000...
6	23:39:39 SELECT * FROM local.ProductImages LIMIT 0, 1000	21 row(s) returned	0.011 sec / 0.045 sec

Query Completed

linux-vm1

Administration Schemas Categories Messages ProductImages Products Reports Reviews UserAvailability Users Wishlist Views Stored Procedures Functions sys

SCHEMAS Filter objects

local Tables Categories Messages ProductImages Products Reports Reviews UserAvailability Users Wishlist Views Stored Procedures Functions sys

Object Info Session No object selected

1 SELECT * FROM local.Products;

Result Grid Filter Rows: Search Edit: Export/Import: Result Grid Form Editor Field Types Query Stats Execution Plan

product_id	seller_id	title	description	price	category_id	condition	status	created_at
1	1	Harry Potter book Set	All 7 books, hardcover.	15.00	1	Great Condition	available	2025-05-21 02:07:15
3	1	Leather Jacket	Black leather jacket, barely used.	120.00	3	Good Condition	available	2025-05-21 02:07:15
4	2	Samsung Galaxy S21	Brand new, works perfectly.	699.99	1	Great Condition	available	2025-05-21 02:07:15
5	1	MacBook Pro 2020	16-inch, excellent condition.	1399.00	1	Great Condition	available	2025-05-21 02:07:15
6	2	AirPods Pro 2020	With wireless charging case.	159.00	1	Good Condition	available	2025-05-21 02:07:15
7	1	Kindle Paperwhite	8GB version, almost new.	89.99	2	Great Condition	available	2025-05-21 02:07:15
8	2	The Great Gatsby	Vintage edition with notes.	10.00	2	Fairly Used	available	2025-05-21 02:07:15
9	1	Python Programming	Beginner to Advanced, 3rd edition.	25.00	2	Good Condition	available	2025-05-21 02:07:15
10	1	White Shirt	White cotton shirt, size M.	15.00	3	Good Condition	available	2025-05-21 02:07:15
11	2	Jeans	Blue Levi's jeans, size 32.	40.00	3	Good Condition	available	2025-05-21 02:07:15
12	1	Winter Coat	Hooded coat, size L.	65.00	3	Fairly Used	available	2025-05-21 02:07:15
13	2	JBL Speaker	Portable and waterproof.	45.99	1	Good Condition	available	2025-05-21 02:07:15
14	1	Monitor 27"	4K UHD monitor with HDMI input.	279.99	1	Great Condition	available	2025-05-21 02:07:15
15	2	Office Mouse	Wireless ergonomic mouse.	29.99	1	Fairly Used	available	2025-05-21 02:07:15
16	2	Notebook Set	Set of 5 college-ruled notebooks.	12.50	2	New	available	2025-05-21 02:07:15
17	1	Hoodie	Gray hoodie, fleece-lined.	35.00	3	Good Condition	available	2025-05-21 02:07:15
18	2	Sneakers	Nike running shoes, size 10.	60.00	3	Used	available	2025-05-21 02:07:15
19	3	Butterfly Training Pr...	Based on the image, this appears to be a butterfly.	15.00	8	Great Condition	available	2025-05-21 02:09:06
20	3	Large Flat Screen TV	Based on the image, this appears to be a flat screen TV.	400.00	1	Great Condition	available	2025-05-21 06:25:41
21	3	PNY GeForce RTX...	Based on the image, this appears to be a graphics card.	1500.00	1	Great Condition	pending...	2025-05-21 06:34:28
HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

Products 1 Apply Revert

Action Output

Time	Action	Response	Duration / Fetch Time
5	23:39:35 SELECT * FROM local.Messages LIMIT 0, 1000	1 row(s) returned	0.0045 sec / 0.00000...
6	23:39:39 SELECT * FROM local.ProductImages LIMIT 0, 1000	21 row(s) returned	0.011 sec / 0.045 sec
7	23:39:43 SELECT * FROM local.Products LIMIT 0, 1000	21 row(s) returned	0.0037 sec / 0.00001...

Query Completed

The screenshot shows the MySQL Workbench application window. The top menu bar includes 'File', 'Edit', 'View', 'Tools', 'Help', and 'Context Help'. The left sidebar is titled 'Administration' and contains sections for 'Schemas', 'Tables', 'Views', 'Stored Procedures', 'Functions', and 'sys'. The main area features a query editor with the SQL command 'SELECT * FROM local.Reports;'. Below it is a 'Result Grid' showing one row of data:

report_id	reporter_id	reported_user...	reported_product...	reason	status	created_at	resolved_at	
1	3	NULL	NULL	10	This is a scam.	pending	2025-05-21 02:16:28	NULL

On the right side, there are several toolbars: 'Result Grid', 'Form Editor', 'Field Types', 'Query Stats', and 'Execution Plan'. The bottom section shows the 'Action Output' tab with three recent actions listed:

Time	Action	Response	Duration / Fetch Time
23:39:39	SELECT * FROM local.ProductImages LIMIT 0, 1000	21 row(s) returned	0.011 sec / 0.045 sec
23:39:43	SELECT * FROM local.Products LIMIT 0, 1000	21 row(s) returned	0.0037 sec / 0.0001...
23:39:46	SELECT * FROM local.Reports LIMIT 0, 1000	1 row(s) returned	0.0042 sec / 0.0000...

A status bar at the bottom indicates 'Query Completed'.

The screenshot shows the MySQL Workbench application running on a Linux VM. The main window displays a query editor with the following SQL statement:

```
SELECT * FROM local.Reviews;
```

The results grid shows the following columns: review_id, reviewer_id, reviewed_user_id, rating, comment, and timestamp. All rows are currently empty.

On the right side of the interface, there is a vertical toolbar with several icons:

- Context Help
- Snippets
- Result Grid
- Form Editor
- Field Types
- Query Stats
- Execution Plan

At the bottom of the interface, there is a table titled "Action Output" showing the history of executed queries:

Time	Action	Response	Duration / Fetch Time
23:39:43	SELECT * FROM local.Products LIMIT 0, 1000	21 row(s) returned	0.0037 sec / 0.00001...
23:39:46	SELECT * FROM local.Reports LIMIT 0, 1000	1 row(s) returned	0.0042 sec / 0.00000...
23:39:49	SELECT * FROM local.Reviews LIMIT 0, 1000	0 row(s) returned	0.0035 sec / 0.00000...

A status bar at the bottom left indicates "Query Completed".

linux-vm1

Administration Schemas Categories Messages ProductImages Products Reports Reviews UserAvailability

SCHEMAS Filter objects local Tables Categories Messages ProductImages Products Reports Reviews UserAvailability Users Wishlist Views Stored Procedures Functions sys

1 • SELECT * FROM local.UserAvailability;

Result Grid Filter Rows: Search Edit: Export/Import:

availability_id	user_id	day_of_week	time_slot	is_available	last_updated
1	3	Monday	8:00	1	2025-05-21 02:15:38
2	3	Monday	9:30	1	2025-05-21 02:15:38
3	3	Monday	10:00	1	2025-05-21 02:15:38
4	3	Monday	10:30	1	2025-05-21 02:15:38
5	3	Monday	11:00	1	2025-05-21 02:15:38
6	3	Monday	11:30	1	2025-05-21 02:15:38
7	3	Monday	12:00	1	2025-05-21 02:15:38
8	3	Monday	12:30	1	2025-05-21 02:15:38
9	3	Monday	13:00	1	2025-05-21 02:15:38
10	3	Monday	13:30	1	2025-05-21 02:15:38
11	3	Monday	14:00	1	2025-05-21 02:15:38
12	3	Monday	14:30	1	2025-05-21 02:15:38
13	3	Monday	15:00	1	2025-05-21 02:15:38
14	3	Monday	15:30	1	2025-05-21 02:15:38
15	3	Monday	16:00	1	2025-05-21 02:15:38
16	3	Monday	16:30	1	2025-05-21 02:15:38
17	3	Monday	17:00	1	2025-05-21 02:15:38
18	3	Monday	17:30	1	2025-05-21 02:15:38
HULL	HULL	HULL	HULL	HULL	HULL

UserAvailability 1 Apply Revert

Action Output

Time	Action	Response	Duration / Fetch Time
23:39:46	SELECT * FROM local.Reports LIMIT 0, 1000	1 row(s) returned	0.0042 sec / 0.00000...
23:39:49	SELECT * FROM local.Reviews LIMIT 0, 1000	0 row(s) returned	0.0035 sec / 0.00000...
23:39:52	SELECT * FROM local.UserAvailability LIMIT 0, 1000	18 row(s) returned	0.0035 sec / 0.00000...

Query Completed

linux-vm1

Administration Schemas Categories Messages ProductImages Products Reports Reviews UserAvailability Users

SCHEMAS Filter objects local Tables Categories Messages ProductImages Products Reports Reviews UserAvailability Users Wishlist Views Stored Procedures Functions sys

1 • SELECT * FROM local.Users;

Result Grid Filter Rows: Search Edit: Export/Import:

user_id	first_name	last_name	email	password	phone_number	profile_picture	bio
1	Alice	Smith	alice@example.com	hashed_password1	1234567890	HULL	SFSU student selling items I no longer need
2	Bob	Johnson	bob@example.com	hashed_password2	0987654321	HULL	Grad student at SFSU, I buy and sell tech items
3	Hilary	Lui	hil@sfu.edu	scrypt:32768:8:1\$160wATgU9XqKbx\$40e2B... (123) 456-7890	HULL	HULL	
4	Sid	Padmanabhu	spadmanabhu@sfu.edu	scrypt:32768:8:1\$NNg7BQsB1QSD1j\$1eb08... (408) 819-6180	HULL	HULL	
HULL	HULL	HULL	HULL	HULL	HULL	HULL	

Users 1 Apply Revert

Action Output

Time	Action	Response	Duration / Fetch Time
23:39:49	SELECT * FROM local.Reviews LIMIT 0, 1000	0 row(s) returned	0.0035 sec / 0.00000...
23:39:52	SELECT * FROM local.UserAvailability LIMIT 0, 1000	18 row(s) returned	0.0035 sec / 0.00000...
23:39:55	SELECT * FROM local.Users LIMIT 0, 1000	4 row(s) returned	0.013 sec / 0.083 sec

Query Completed

Schemas

Administration

SCHEMAS

local

Tables

Categories

Messages

ProductImages

Products

Reports

Reviews

UserAvailability

Users

Wishlist

Views

Stored Procedures

Functions

sys

Object Info

No object selected

Session

1 • SELECT * FROM local.Wishlist;

Result Grid

Filter Rows: Search

Edit: Export/Import:

wishlist_id user_id product_id date_added

wishlist_id	user_id	product_id	date_added
3	4	20	2025-05-21 06:32:18
4	4	19	2025-05-21 06:32:19
5	4	13	2025-05-21 06:32:20
6	4	6	2025-05-21 06:32:22
7	4	5	2025-05-21 06:32:23
NULL	NULL	NULL	NULL

Wishlist 1

Apply Revert

Action Output

Time	Action	Response	Duration / Fetch Time
10	23:39:52 SELECT * FROM local.UserAvailability LIMIT 0, 1000	18 row(s) returned	0.0035 sec / 0.00000...
11	23:39:55 SELECT * FROM local.Users LIMIT 0, 1000	4 row(s) returned	0.013 sec / 0.083 sec
12	23:39:58 SELECT * FROM local.Wishlist LIMIT 0, 1000	5 row(s) returned	0.0042 sec / 0.00000...

Query Completed

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid

Form Editor

Field Types

Query Stats

Execution Plan

A screenshot of the MySQL Workbench interface. The left sidebar shows the schema 'local' with various tables like Categories, Messages, and Wishlist. The main area displays a query result grid for the 'Wishlist' table with columns: wishlist_id, user_id, product_id, and date_added. Below the grid is an 'Action Output' table showing the history of queries run. A tooltip on the right explains how to enable context help. The bottom right corner shows a small preview of the execution plan.

5. GitHub organization:

Sid has access to master branch

This branch is 112 commits ahead of [main](#).

Commits

- [SidPad03](#) Merge pull request #56 from CSC-648-SFSU/staging 5 hours ago 113 Commits
- [.github/workflows](#) Merge remote-tracking branch 'origin/main' into about_me... 3 months ago
- [Milestones](#) Delete Milestones/M2/CSC648-848 Spring 2025 Section 01 ... 5 hours ago
- [application](#) user management 5 hours ago
- [credentials](#) search functionality last month
- [db](#) user management 5 hours ago
- [.gitignore](#) landing page redesign last month
- [LICENSE](#) Initial commit 3 months ago
- [README.md](#) Update README.md 3 months ago
- [requirements.txt](#) Merge pull request #40 from CSC-648-SFSU/sid_m4_0422 last month

About

csc-648-sfsu-2025-fall-souza-01-csc648-base-repo created by GitHub Classroom

[thriftatsfsu.com/](#)

[Readme](#) [MIT license](#) [Activity](#) [Custom properties](#) [0 stars](#) [0 watching](#) [0 forks](#)

Releases

No releases published [Create a new release](#)

Packages

No packages published [Publish your first package](#)

Languages

HTML 50.3% Python 35.4% CSS 14.3%

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.

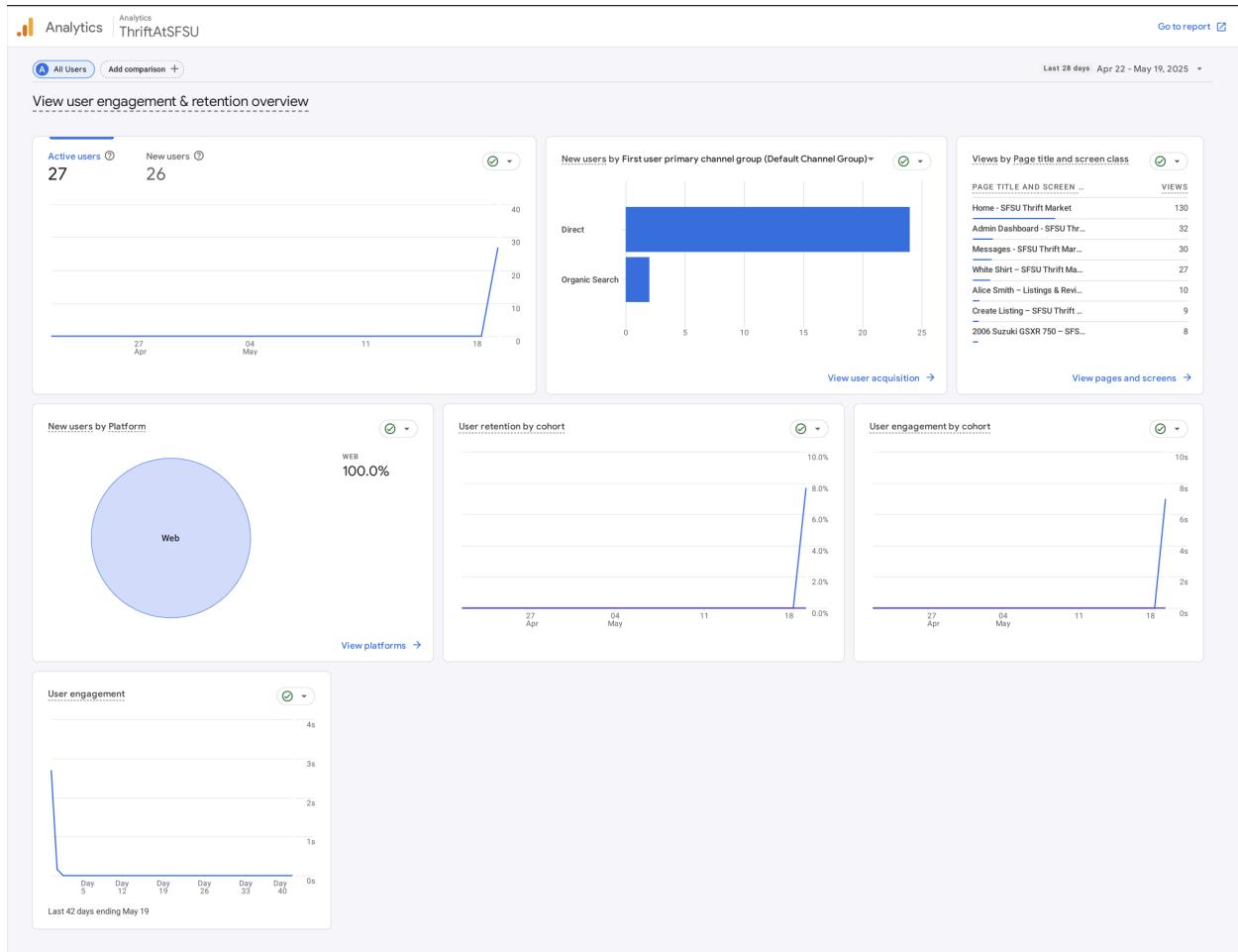
Please do the following steps before completing Milestone 0.

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.
 - o 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	School Email	GitHub Username
Sid Padmanabhu	spadmanabhuni@sfsu.edu	SidPad03
Joseph Shur	jshur1@sfsu.edu	joseph-shur
Hilary Lui	hlu1@sfsu.edu	Hluui
Joseph Alhambra	jahambra@sfsu.edu	JosephCVA
Annison Van	avan@sfsu.edu	avan882

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 stats plot for your WWW site



7. Project management

For project management, we used GitHub Projects, which is integrated directly into our repository. This helped us keep track of tasks, organize issues, and stay aligned as a team.

We used the basic GitHub project board to list out our main development tasks and features. Each task was represented as a GitHub issue, and we used the project board to track progress and check off completed work. This made it easy to assign tasks, track priorities, and keep everyone updated in one central place.

Hluii opened on Mar 11 · edited by anvan882

Edits ⋮ ⋮

Assignees JosephCVA
anvan882

Labels No labels

Type Task

Projects CSC648-0104-team15

Status Done

Priority Choose an option

Size Choose an option

Estimate Enter number...

Iteration Choose an iteration

Start date No date

End date Mar 25, 2025

Milestone No milestone

Relationships None yet

Development Create a branch for this issue or link a pull request.

Notifications Customize

Unsubscribe You're receiving notifications because you're subscribed to this thread.

Participants

Transfer issue

Lock conversation

Pin issue

Every Page Beyond this point should include the navbar

- Landing page (Not signed-in)
- Login
- Sign up (email w/ SFSU "verification", password, password verification, terms and conditions checkbox)

- Listing home page (Includes search bar, filters, and listings)
 - [x] Listings Should Include a picture, Name of seller, and price
 - [x] Filters: Category(Sports, Books, Clothing, Food, Travel), Condition (new, like new, used)
- Listing Creation Page (Fields for adding image, title, price, category, and description)
- Individual Listing Page (image, title, price, category, and description)
 - [] Button to contact seller and look at seller profile
- Profile Page (Has the seller schedule, bio, and area for reviews)
- Chat Screen (Multiple private chats, chat messages between seller and buyer, a method to connect back to listing, dropdown for each listing-related chat)
- Wishlist with listings in it
- Navbar (logo/Home, messages, wishlist, profile)

Create sub-issue ⋮ ⋮

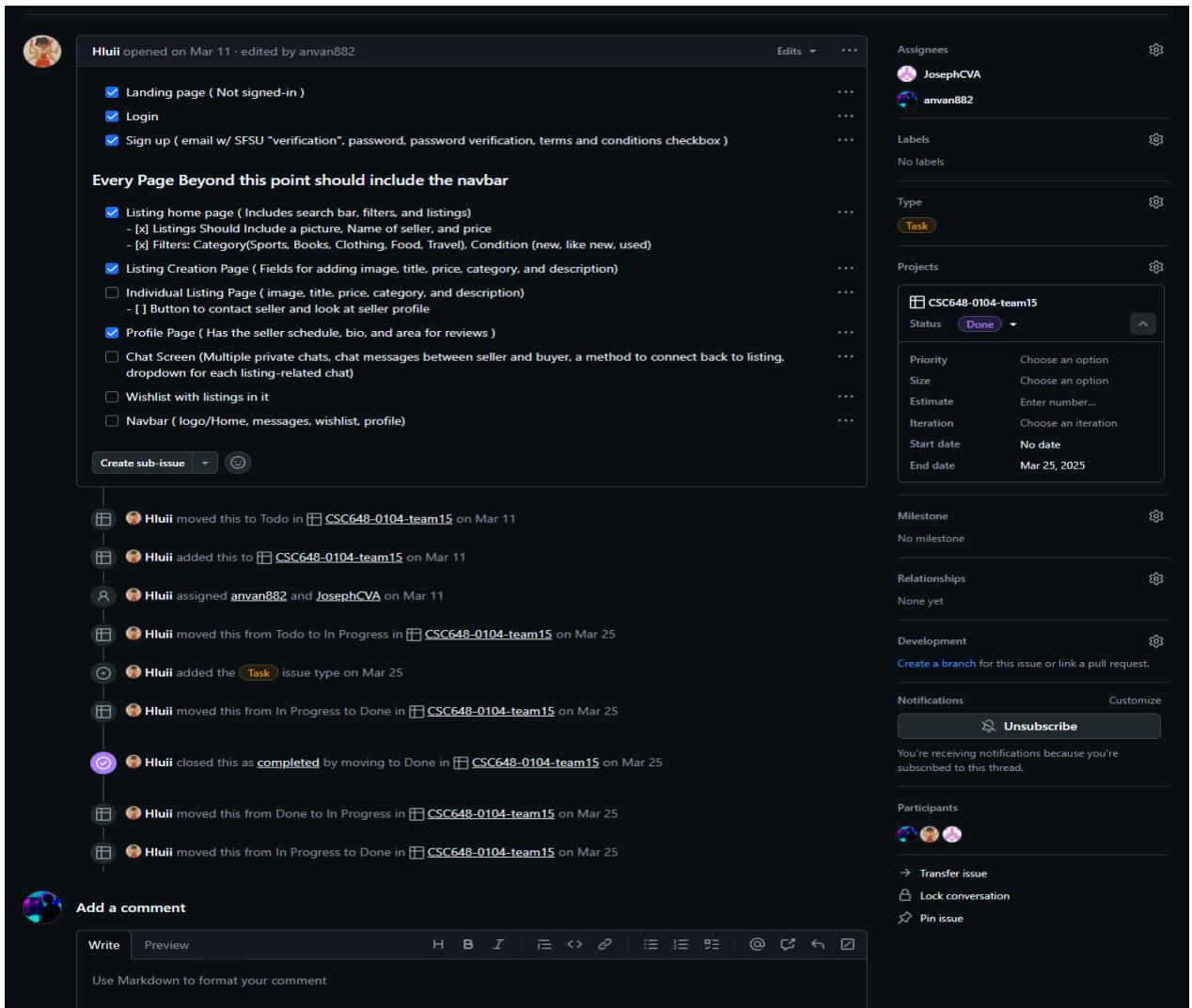
Timeline

- Hluii moved this to Todo in CSC648-0104-team15 on Mar 11
- Hluii added this to CSC648-0104-team15 on Mar 11
- Hluii assigned anvan882 and JosephCVA on Mar 11
- Hluii moved this from Todo to In Progress in CSC648-0104-team15 on Mar 25
- Hluii added the Task issue type on Mar 25
- Hluii moved this from In Progress to Done in CSC648-0104-team15 on Mar 25
- Hluii closed this as completed by moving to Done in CSC648-0104-team15 on Mar 25
- Hluii moved this from Done to In Progress in CSC648-0104-team15 on Mar 25
- Hluii moved this from In Progress to Done in CSC648-0104-team15 on Mar 25

Add a comment

Write Preview

Use Markdown to format your comment



CSC-648-SFSU / Projects / CSC648-0104-team15

Type [] to search

Add status update | Discard | Save

Backlog | Team capacity | Roadmap | Current iteration | My items | New view

Filter by keyword or by field

Todo 0 / 5 Estimate: 0

This item hasn't been started

In Progress 2 / 5 Estimate: 0

This is actively being worked on

- csc648-fa25-0104-team15 #54 Milestone 5
- csc648-fa25-0104-team15 #53 Milestone 4

Done 9 Estimate: 0

This has been completed

- csc648-fa25-0104-team15 #49 QA Testing
- csc648-fa25-0104-team15 #28 M3 Frontend Work
- csc648-fa25-0104-team15 #2 Server is up 24/7
- csc648-fa25-0104-team15 #21 Milestone 2 part 2
- csc648-fa25-0104-team15 #50 Code Review
- csc648-fa25-0104-team15 #16 Milestone 1 PDF review
- csc648-fa25-0104-team15 #19 Make revisions on Milestone 1 document
- csc648-fa25-0104-team15 #20 Milestone 2 Part 1
- csc648-fa25-0104-team15 #17 Wireframe

+ Add item

+ Add item

+ Add item

8. Team member self-assessment and contributions

Annison Van

Self-assessment and Contributions

 Annison Van
To: Ⓜ Joseph Velasco Alhambra; Ⓜ Joseph Harmon Shur; Ⓜ Hilary H Y Lui; Ⓜ Siddharth Padmanabhuni

Role: Front-End Lead

Contributions to Team Project and Teamwork

- Led front-end development using HTML, CSS, and Flask templates, converting wireframes into responsive, accessible pages.
- Built and refined core features such as the wishlist and messaging interfaces with a focus on user experience.
- Integrated front-end pages with Flask backend routes for data rendering and form submissions.
- Helped manage Git workflows by resolving merge conflicts and reviewing pull requests.
- Participated in stand-ups, coordinated tasks, and incorporated feedback to improve layout and usability.

(b) Number of GitHub Dev Branch Submissions: 9 commits

Explanation:
While my commit count is on the lower side, that's because I focused on developing larger features locally and submitted them in consolidated, fully-tested commits. This kept the repository history clean and reduced noise from small or partial changes. My commits represented complete and functional pieces of the front-end, prioritizing quality and clarity over quantity.

(c) Main Challenges

One of the biggest challenges was integrating the front-end with Flask without using JavaScript. It made dynamic behavior more difficult, requiring workarounds using only routes and templates. Keeping design consistent across the team also took extra coordination, especially when overlapping UI sections were involved. Git version control presented early issues with branch syncing and merge conflicts, but we improved over time through better communication and workflow habits.

(d) GenAI Usage and Usefulness

We used GenAI tools like ChatGPT early in the project to generate user personas, use cases, and design ideas, which sped up planning and gave us a clear direction. It also came in handy during development for debugging, solving layout issues, or clarifying concepts we were unfamiliar with. AI helped us move faster without relying on guesswork and was a useful support tool throughout the process.

(e) Improvements for Next Time

Next time, I'd use project management tools like GitHub Issues and Project Boards more consistently to organize tasks and assign responsibilities. While we had regular communication, we didn't always document who was doing what, which led to confusion and delays. I also learned the value of smaller, more frequent commits and clearer commit messages to help the team stay aligned. These are all practices I plan to improve moving forward.

(f) Additional Notes

My GitHub activity may seem low, but that doesn't reflect the scope of my contributions. I often tested and polished my work locally before pushing, and each commit delivered complete, working front-end features. My focus was always on quality, completeness, and maintaining clean code integration with the back-end.

Joseph H Shur

 Joseph Harmon Shur
To: Ⓜ Hilary H Y Lui; Ⓜ Joseph Velasco Alhambra; Ⓜ Annison Van; Ⓜ Sid Padmanabhuni

A. Role: Backend Team Member/Developer

- Filled out frontend CSS for dashboard/profile page
- Filled out dummy data for tables on both main search/home page and profile/dashboard before connecting backend to make sure team could see how our CSS would be applied to the site
- Helped with user contribution of ideas for main landing page, frontend formatting & messaging page throughout milestones
- Wrote a list of dev needs & requirements for high-level functional requirements throughout milestones starting at M1
- Took notes to base group meeting around M3 meeting with CTO
- Completed Self-check on best practices for security for M4
- Assisted Backend Lead Sid in backend lead in routes and backend using Python
- B. Number of submissions made to GitHub team Dev. Branch: 5-10. It wasn't too many submissions, since I mostly focused on a few things and didn't end up implementing too much backend code (Sid worked on most of it pretty early on), so I mostly contributed to more specific and frontend components of the website.
- C. The challenges that I specifically encountered were figuring out exactly what to do and how to contribute. While I had my work cut out for me, it was difficult to figure out where my work began and others ended. I honestly felt at times I didn't contribute as much as I wanted to, but I tried my best to contribute whenever I was asked.
- D. My use of AI wasn't too difficult but it definitely made things a lot more streamlined and quicker for the entirety of the project. It helped our group get our feet off the ground and have a more clear idea of what the general appearance of our website and UI/UX would ultimately become in the end - so definitely worth the time.
- E. What I would do better in the future is take more initiative; I mostly let the assignments fall to me based on what wasn't done, but next time I would like to get more hands-on and take up more than what was asked of me. That isn't to say I didn't learn anything about Python or backend at all, but I wish I gave myself the proper chance to implement it more directly.
- F. No other notes (explained the low commit count in section B above), but I enjoyed working with my group and would definitely collaborate with them again if presented the opportunity.

Joseph V Alhambra

 Joseph Velasco Alhambra
To: Ⓜ Sid Padmanabhuni; Ⓜ Joseph Harmon Shur; Ⓜ Hilary H Y Lui; Ⓜ Annison Van

a) Role: Front End Dev. Collaborated with Annison Van (Front-End Lead) developing personas, use cases, & wire frame models for the site. Collaborated with Annison Van (Front-End Lead) on site designs/design reviews through Figma. Developed implementation of landing page. Collaborated with Sid's (Back-End Lead) version of landing page with dummy UI for back-end implementation. Added filters UI, 'Add Listings' option, and mobile view compatibility to landing page. Developed the 'Add New Listings' page and listing upload UI to the site. Developed the 'Individual Listing' page to the site.

b) Around 5-10 submissions were made, but not to the team Dev Branch directly. Front end development of the site was done in multiple branches that were merged to one singular 'staging' branch. Submissions were made as entire branches that would be reviewed and merged by the team lead and/or the back-end lead through pull requests. Additional pushes would be made based on needed visual fixes or recommendations from other team members.

c) Once the back-end started to get developed, we all needed to set up extra installations and do extra preparations in order to set up local instances of the project that would work with our current working project. It had to be done, but I would keep forgetting the steps to reinstall the project every time I needed to clone or pull.

d) GenAI was used to help develop personas and use cases. It was very useful for helping with more mundane tasks unrelated to the actual code of the project. GenAI would give examples of people who would use our site and potential problems/needs the may face when using our site.

e) I think being able to invest more time into the team project was something I could have done better. At the beginning of the semester I had lost my role as Front End Lead since I couldn't dedicate the time to the role because of classes demanding super early deadlines at the time.

f) Submissions were only made after initial submission either for UI fixes or requests from other team members. I would receive tasks on developing certain pages and adding UI. I would submit progress to a separate branch to be reviewed and merged to a 'staging' branch where the team's working code was collected. Additional submissions would be made either to fix any UI issues or apply other team member requests until approval. A lower count meant that earlier submissions were approved and no further development was discussed at the time.

Sid Padmanabhuni

Self-Assessment & Contributions



Siddharth Padmanabhuni

To: Hilary H Y Lui; Joseph Harmon Shur; Joseph Velasco Alhambra; Annison Van

Reply Reply all Forward | ...

Mon 5/19/2025 5:49 PM

Hi Team,

As part of our self-assessment for the project, please find my summary below:

a) Contributions to team project and teamwork:

- Role: Backend Lead, Github Master, DevOps Lead
 - Designed and implemented all RESTful APIs using Flask to support core application features.
 - Integrated the backend APIs with the HTML templates to enable full-stack functionality.
 - Maintained and organized the Git repository, including regular merges, PR reviews, and issue tracking.
 - Set up CI/CD pipelines to automate testing and deployment, improving development efficiency.
 - Deployed the application on an AWS EC2 instance and managed server uptime and configurations.
 - Enforced branch protection rules to ensure safe code merges and maintain main branch integrity.
 - Provided cross-functional support and ensured alignment between backend and front-end development teams.
 - Fixed and optimized CSS styles to improve the application's UI/UX consistency across pages.

b) GitHub Dev Branch Submissions:

- Number of submissions: 28 commits, 11,918 ++, 4,908 --

c) Main Challenges Encountered:

One of the main challenges was coordinating backend changes with front-end development timelines, especially while maintaining the stability of the main branch. Managing AWS server configurations also posed some learning curves, especially during deployment and testing phases. I have also had some issues with the uptime and reliability of the EC2 server since it is the free tier.

d) Experience with GenAI:

I used GenAI extensively throughout the project. It was instrumental in helping me write the Flask APIs more efficiently by generating boilerplate code and offering debugging suggestions. I also leveraged the GPT-4o image generation tool to create UX mockups, which I then used as input to have GenAI generate corresponding CSS and HTML structure. This repetitive design-to-code workflow significantly sped up front-end development. Once the static templates were generated, integrating them with Flask routes became much smoother and faster. It was also helpful, technically, for example in setting up our reverse proxy or CI/CD pipelines for our EC2 instance.

e) Improvements for Next Time:

Next time, I would prioritize creating better documentation for onboarding and setup processes, so team members can get started more independently rather than relying on one-on-one help. This would save time and improve consistency across the team. I would also improve task delegation, especially to the backend developer I worked with, to ensure responsibilities are more evenly distributed and reduce bottlenecks during development.

Thanks,
Sid

Reply Reply all Forward

Hilary Lui



Hilary H Y Lui

To: Siddharth Padmanabhuni; Annison Van; Joseph Harmon Shur; Joseph Velasco Alhambra

Reply Reply all Forward | ...

Tue 5/20/2025 11:31 PM

Hi Team,

Below is my self-assessment and team lead feedback for our final project submission. Thank you all for your hard work!

Role: Team Lead

(a) Contributions to Team Project and Teamwork

- Wrote and organized all documentation from M1 to M5
- Managed team communication and coordinated weekly meetings
- Communicated regularly with our CTO/CEO to ensure alignment on deliverables
- Taught the team how to create pull requests and set up GitHub Projects for task management
- Presented the final project at Demo Day
- Oversaw website design consistency and guided decisions on frontend and backend elements
- Reviewed several pull requests and contributed to the "About Me" pages
- Helped plan and execute the QA testing process
- Consolidated GenAI outputs into documentation and helped the team use it responsibly

(b) Number of GitHub Dev Branch Submissions:

- Number of submissions: 7 commits, 72 ++, 0 --

c) Main Challenge (Individual)

My biggest personal challenge was procrastination. I often delayed preparation for meetings, which meant some decisions were made during meetings instead of beforehand. This sometimes led to less efficient use of time and delayed task assignments. I also struggled with finding the right balance between being hands-on with the project and stepping back to delegate. I sometimes felt guilty about not writing more code, but I also recognized that my leadership role was meant to focus on coordination and guidance.

d) GenAI Experience (Individual)

GenAI (ChatGPT) was extremely helpful, especially for interpreting vague documentation and assignment instructions. It gave me structured templates to start from and helped improve the quality and tone of written content. I used it frequently to edit and refine documentation, prepare task outlines, and draft professional communication.

e) What I'd Do Differently (Individual)

If I could do it again, I would assign tasks as soon as a milestone was published and use team meetings to check on progress instead of waiting until the meeting to assign work. Better preparation before meetings would have allowed me to use that time more efficiently and implement GitHub Projects more effectively from the start.

f) Additional Comments

Although my GitHub commit count was low, most of my time was spent managing the team, preparing documentation, and coordinating with stakeholders. I focused on supporting the team and ensuring each milestone was delivered with complete written material.

Overall Feedback from Team Lead**c) Team Challenges**

One of the major team-wide challenges was scheduling conflicts. Our chosen meeting time often overlapped with dinner hours, which led to delays and reduced attendance. This made it hard to maintain consistent momentum and often required us to follow up individually after meetings to clarify responsibilities.

d) GenAI Use (Team Perspective)

GenAI was a huge help with writing support, especially for QA documentation, code review summaries, and milestone reports. It helped structure our ideas and turn rough outlines into polished deliverables. It required supervision, especially for technical writing, to ensure consistency with our project's style and requirements. GenAI also helped us understand and modify large chunks of code, which made development more manageable.

e) What I'd Do Differently as Team Lead

Next time, I'd make better use of GitHub Projects from the start, and stay consistent with tracking and updating it throughout the milestone cycle. Halfway through the project, I shifted task assignment to meetings and Google Docs, but that created gaps in accountability. If I had been more prepared before each meeting, I could have integrated GitHub Projects more effectively and made follow-ups easier.

Best Regards,

Your Team Lead, Hilary