

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. Transaction
 - a. Description: Records when a product has been marked as sold and to whom. Helps with reviews and messaging.
 - b. Key Attributes:
 - i. transaction_id (PK)
 - ii. buyer_id (FK to User)
 - iii. seller_id (FK to User)
 - iv. product_id (FK to Product)
 - v. agreed_price
 - vi. status (e.g. completed, canceled)
 - vii. timestamp
- 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

The image displays two side-by-side UI mockups on a light gray grid background. Both mockups are contained within rounded rectangular frames.

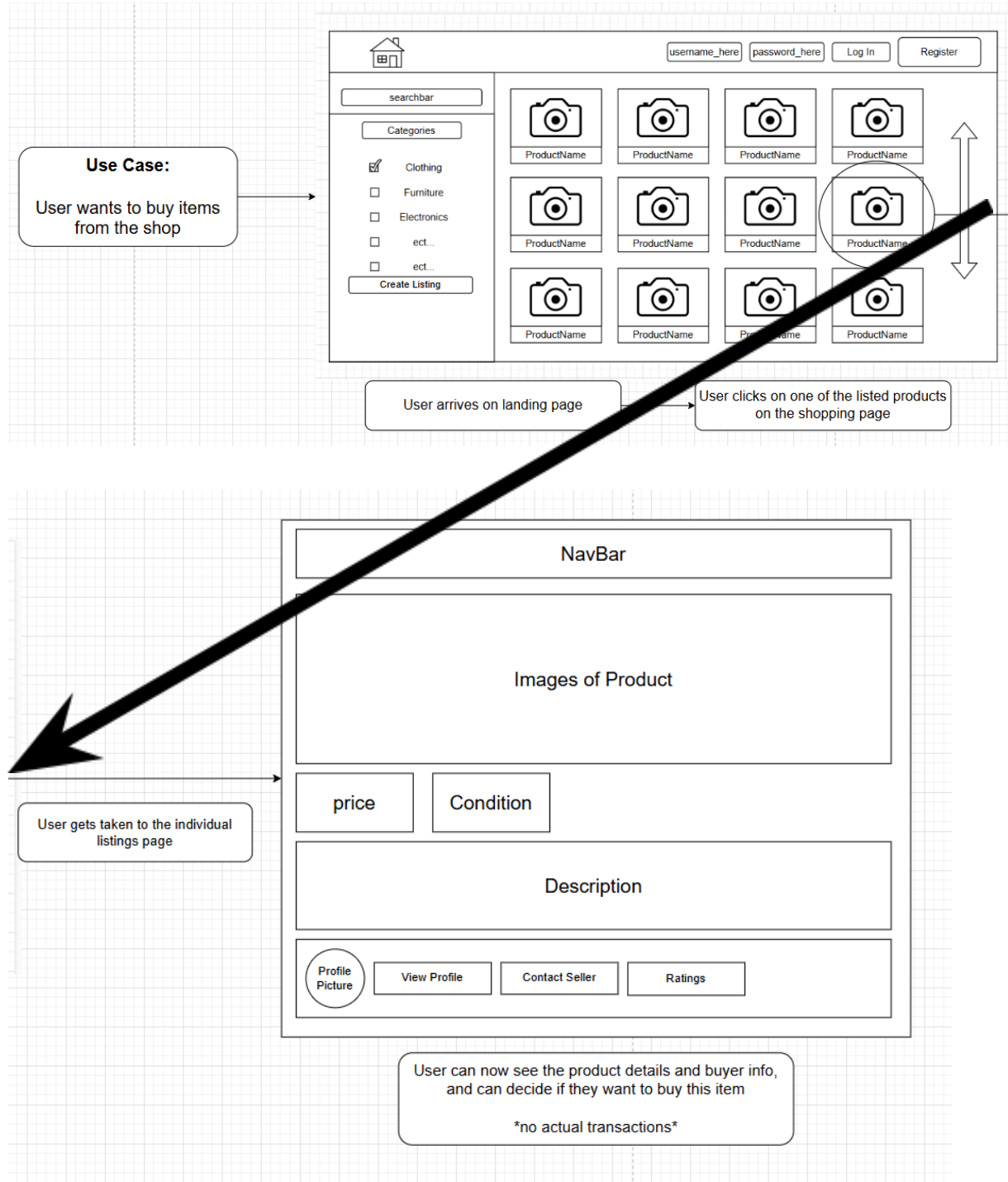
Left Mockup (Login Screen):

- It features two input fields stacked vertically: the top one is labeled "SFSU email" and the bottom one is labeled "Password".
- Below these fields is a "Login" button.
- A horizontal line separates the login section from the registration section below.
- The registration section contains the text "New to ThrifterSFSU?" followed by a link labeled "Create Account".

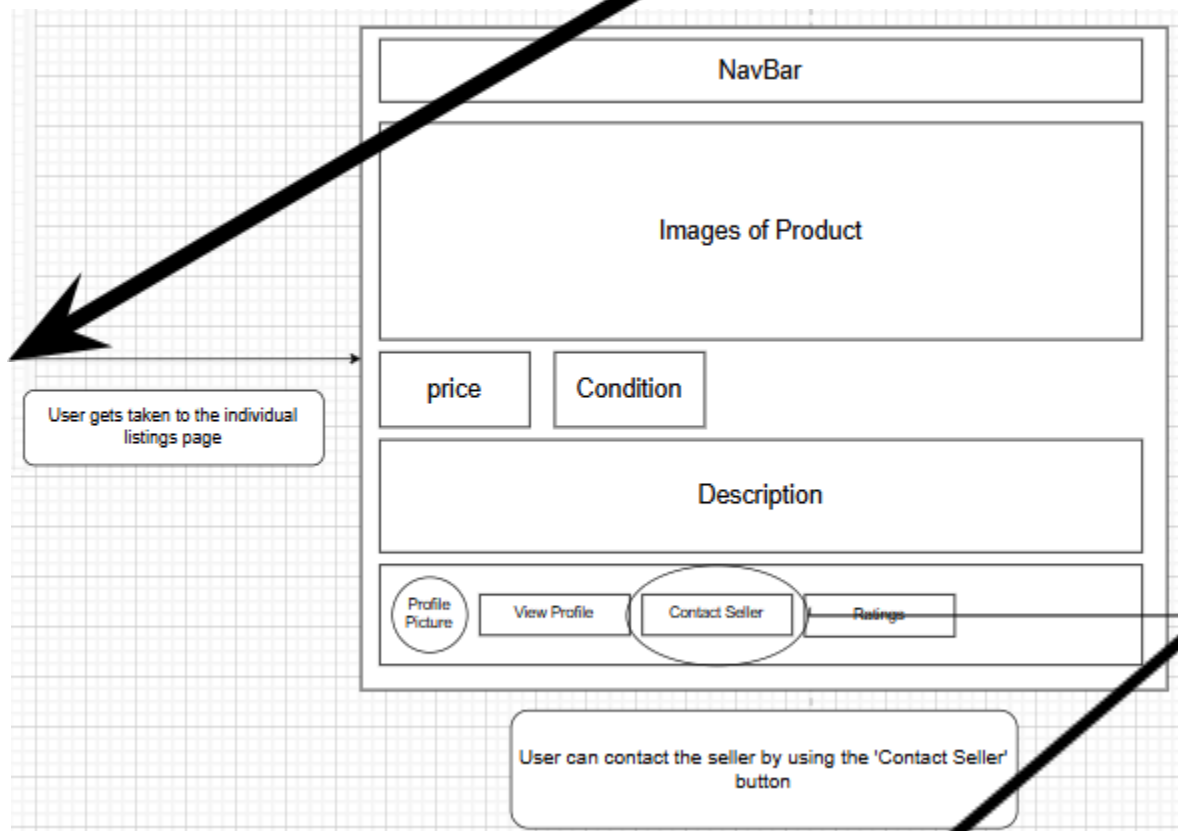
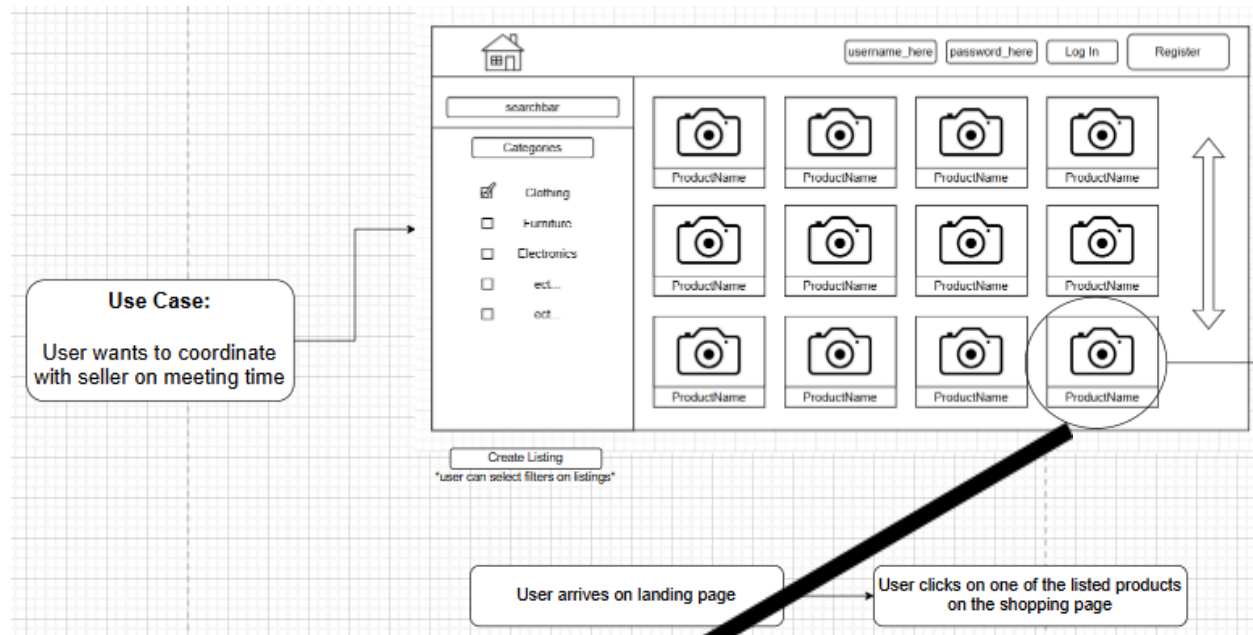
Right Mockup (Registration Screen):

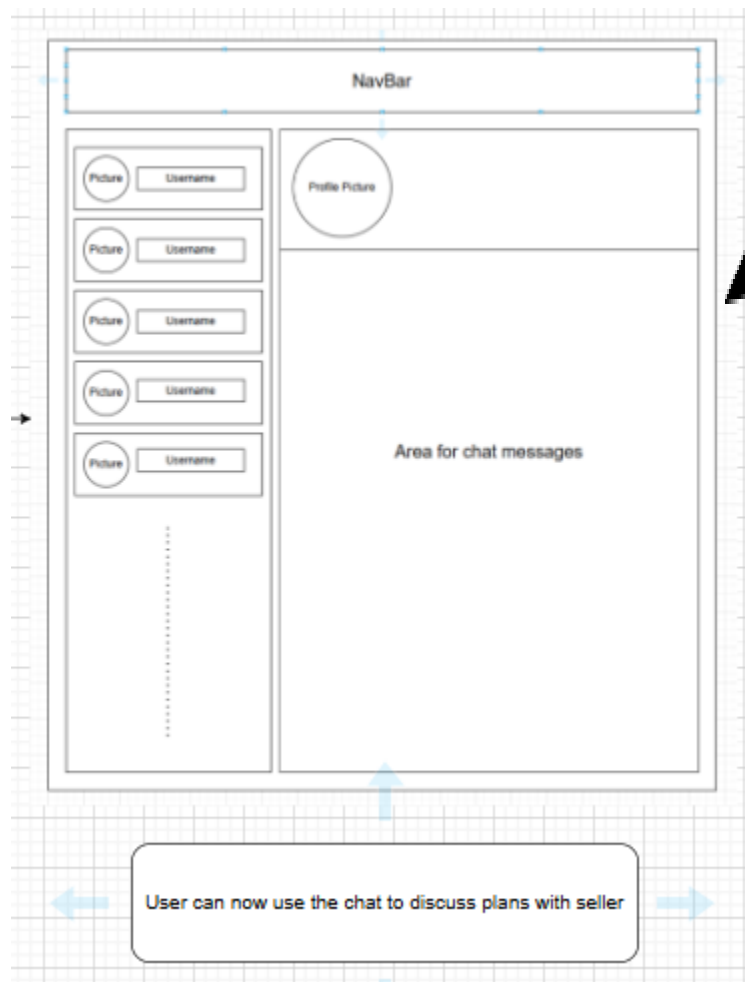
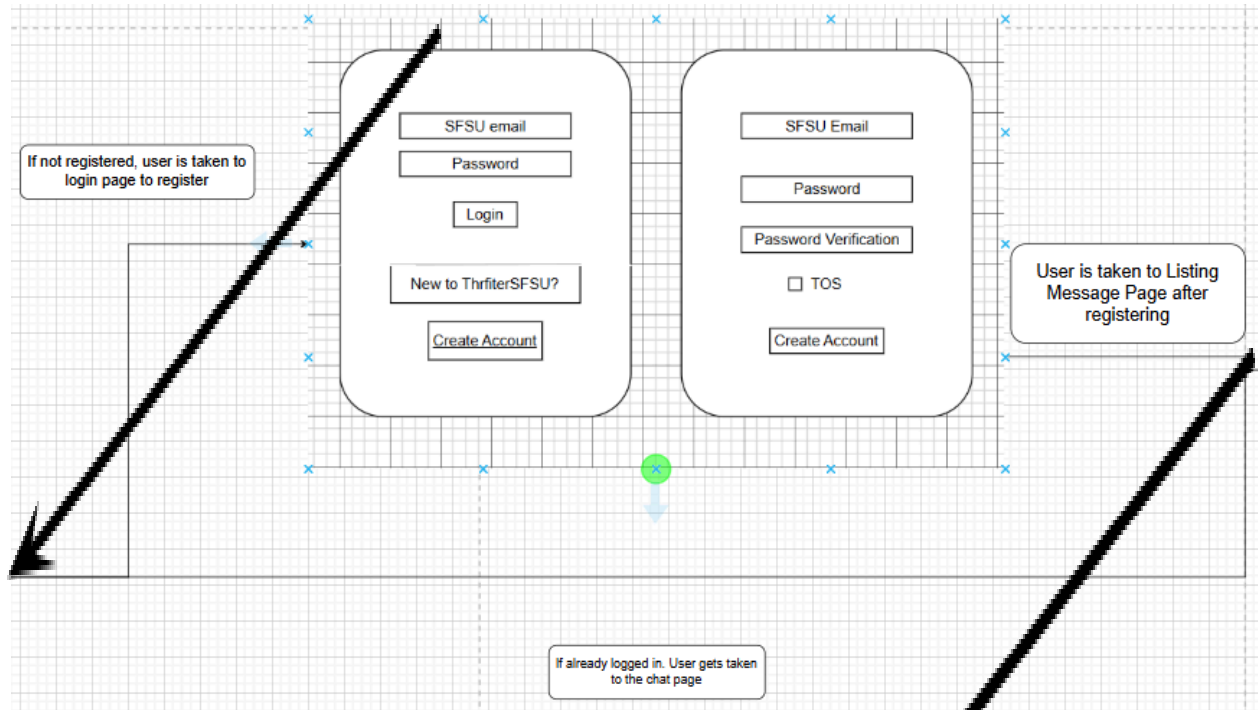
- It features three input fields stacked vertically: the top one is labeled "SFSU Email", the middle one is labeled "Password", and the bottom one is labeled "Password Verification".
- Below the "Password" field is a checkbox labeled "TOS".
- At the bottom is a "Create Account" button.

1.

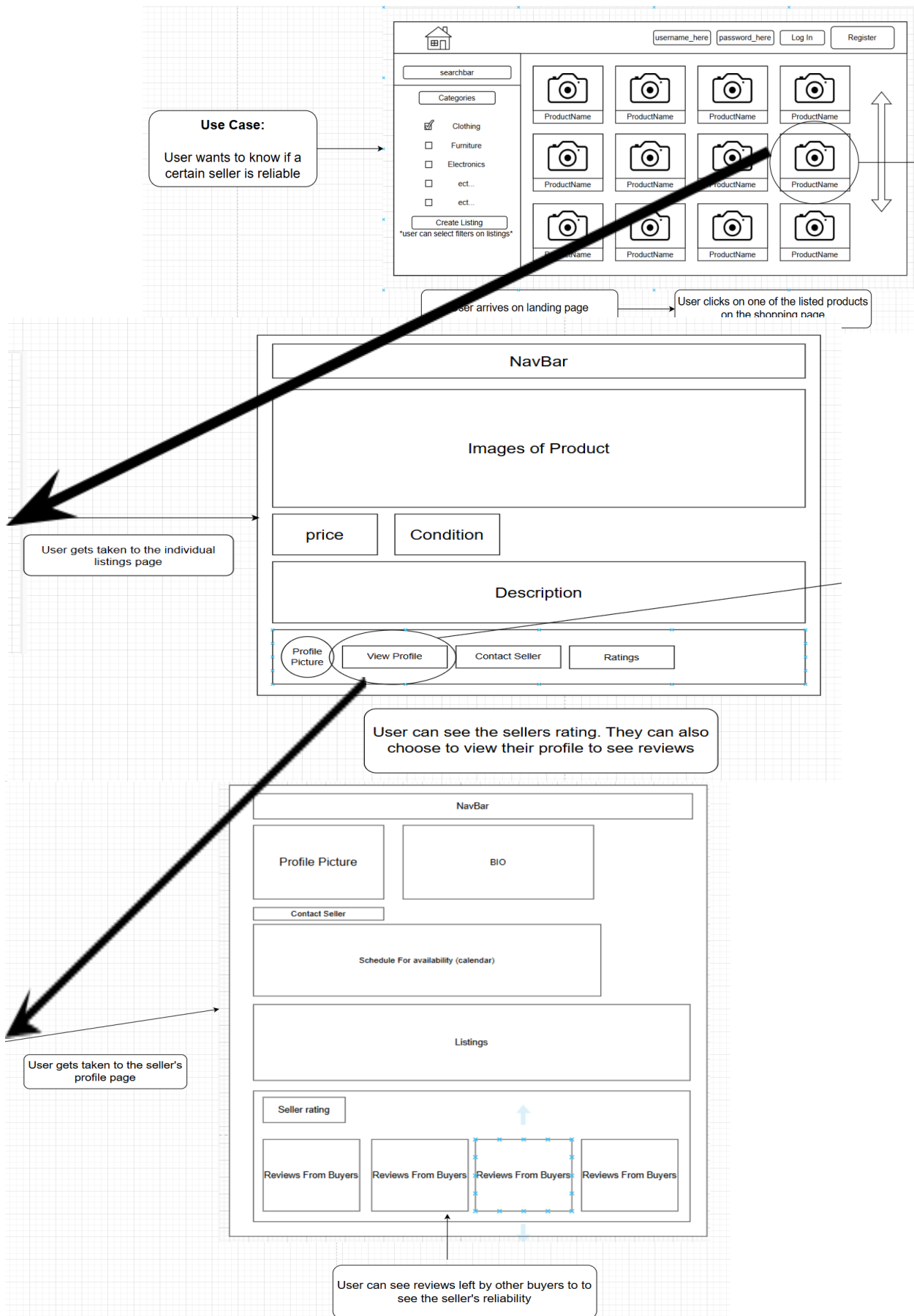


2.

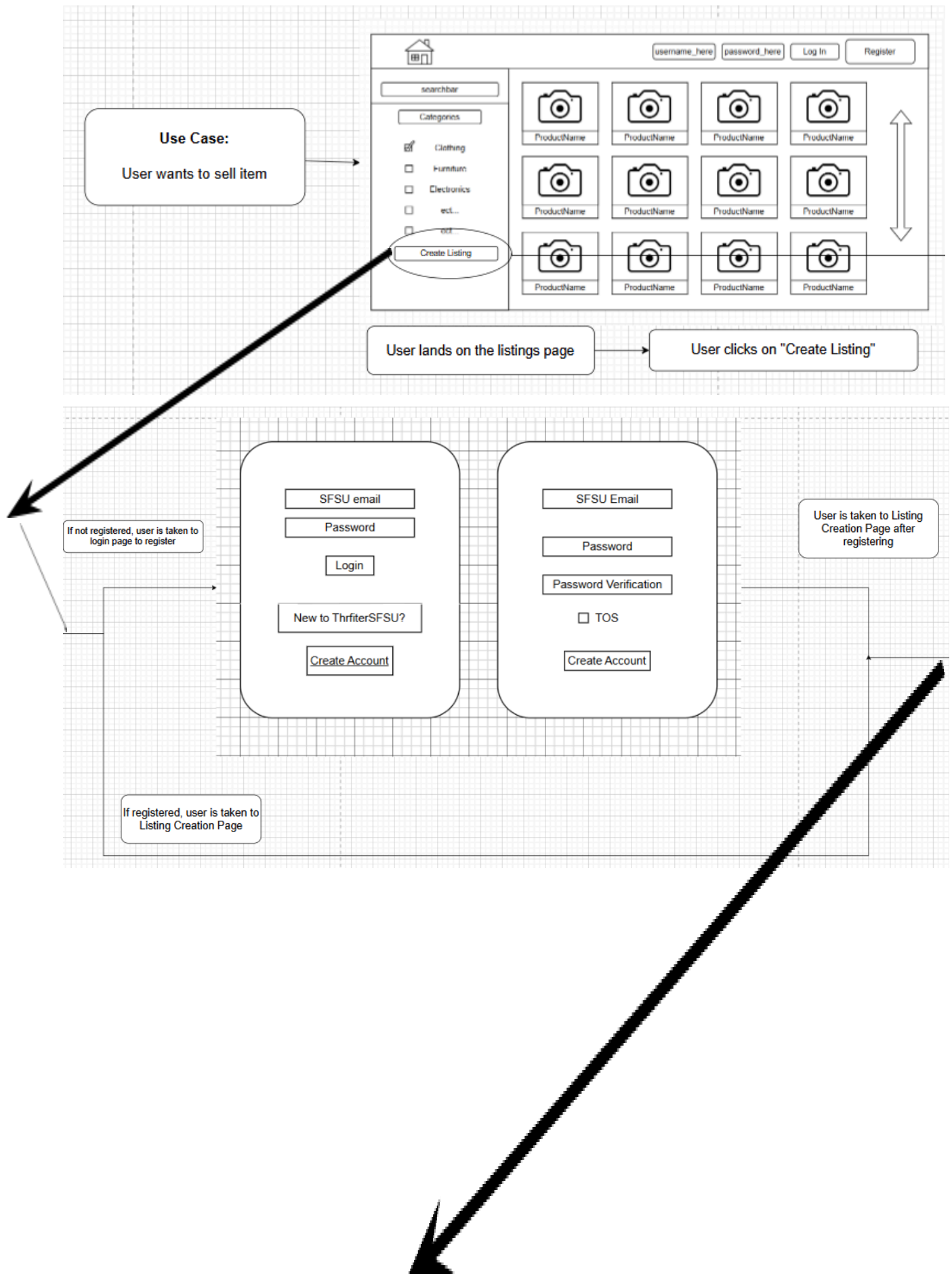




3.



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

OpenTask

CSC-648-SFSU/csc648-fa25-0104-team15Private

Hluui opened 2 weeks ago · edited by Hluui

Edits

☒ 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

Assignees

JosephCVA

anvan882

Labels

No labels

Type

Task

Projects

CSC648-0104-team15

StatusIn Progress

PriorityChoose an option

SizeChoose an option

EstimateEnter number...

IterationChoose an iteration

Start dateNo date

End dateMar 25, 2025

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 out 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