
SW Engineering CSC648-848 Spring 2025

Team 16: “Gator Goods”

Milestone 2 Part 1

Members:

Team Lead: Martha Martin Pablo
Front-end Lead: Jaideen-Jace Bondoc
Back-end Lead: Anthony Mingus
GitHub Master: Vansh Virani
Garvin Zhao

Team Lead Email:

mmartinpablo@sfsu.edu

Milestone	Dated Submitted	Date Revised
M2	March 25, 2025	

Table of Contents

Executive Summary	3
Main Data Items & Entries	4
High-level Functioning Requirements	7
UI Storyboards for Each Main Use Case	8
High level Architecture	18
Identify actual key risks for your project at this time	24
Project Management	25
Use of GenAI Tools	26
Team Lead Checklist	27

Executive Summary

Gator Goods is a localized online marketplace designed exclusively for the San Francisco State University (SFSU) community. It provides students, faculty, and staff with a trusted, campus-exclusive platform to buy, sell, and trade goods—including second-hand textbooks, electronics, dorm essentials, and handmade items. Unlike generic marketplaces, Gator Goods ensures that all users are verified SFSU members, creating a safe, affordable, and sustainable environment for transactions. By keeping commerce within the campus ecosystem, we improve accessibility, reduce waste, lower cost, and strengthen community connections.

A standout feature of Gator Goods is its integrated courier system, enabling registered SFSU members to earn extra income by delivering goods between buyers and vendors. This service provides a flexible, student-friendly job opportunity while eliminating the need for in-person meetups. Additionally, the platform supports local food vendors and student-run businesses by supporting an optional food delivery feature, allowing small restaurants and home-based cooks to reach a wider audience without relying on expensive third-party services.

At its core, Gator Goods is more than just a marketplace—it promotes sustainability, affordability, and community, making campus life more connected and resourceful. By providing a centralized, user-friendly platform, Gator Goods fosters trusted, campus-exclusive transactions, helping students save money, reduce waste, and support one another. Investing in this project means supporting a practical, student-led solution that directly benefits the entire SFSU ecosystem.

Our team created Gator Goods to bridge the gap between available resources and campus needs. As students we have searched for affordable goods off-campus, unaware that our peers may already have what we need. This lack of awareness leads to unnecessary spending and waste. We wish to minimize this challenge for the current and future SFSU community while connecting and strengthening the campus community.

Main Data Items & Entries

- **Registered user:** *A parent type of user that has some personal information stored for security purposes, as well as a naming scheme for the user.*
 - First name: User's first name, government-recognized
 - Last name: User's last name, government-recognized
 - Username: Chosen name by the user
 - Password: Where User would be able to apply a password to log in-and-out of their account
 - Sfsu email: To verify User as SFSU-recognized, either as student or faculty/staff
- **Buyer:** *A registered user that is able to place products from listings into their cart, buy products, and access a list of past transactions.*
 - Past Delivered: A record of only products that have been delivered to the buyer, as we cannot keep track of the things they pickup from vendors.
 - Incoming Package: A list of delivery instructions that are being addressed to the buyer that have yet to be delivered. Is a way to track their incoming packages.
- **Vendor:** *A registered user that is able to post listings and sell products on the system, and keep a record of all sales made with them.*
 - Listing: A reference to current listings posted by the vendor
 - Rating: An average score of the satisfaction rating given by buyers who have bought products from them.
 - Sold Item: A record of all listings that have been completed and sold by the vendor.
 - Pending Delivery Request: A list of all pending delivery requests from potential buyers.
- **Courier:** *A registered user that is able to be assigned deliveries between buyers and vendors, who takes action in retrieving the product and completing the transactional process of making sure the buyer receives it.*
 - Availability status: A reference at which the Courier will be able to mark themselves as either available to make a delivery, or off duty.
 - Delivery Instruction: An assigned instruction to the courier that allows them to retain and complete a delivery once assigned to them.
- **Product:** *A type of item that is sold by a vendor.*
 - Title: Name of the product
 - Category: A keyword that can help filter the item
 - Description: a brief explanation of the product
 - Image: a visual representation of the product
 - Vendor: the user who created the product

- **Review:** *The Review is provided by a buyer who has recently made a purchase and will be a post that is publicly displayed on vendors to help future buyers make informed decisions. (optional)*
 - Rating: “Star” count [can be a 1-5 scale]
 - Comment: written feedback from the user
 - Date created: Keeps track of when the rating was made
 - Vendor: The name of the vendor from which the buyer bought from.
 - Author: Keeps track of the buyer’s name and which review they left
- **Listing:** *Represents a specific instance of a product that a vendor offers for sale within the marketplace.*
 - Product: Reference which product this listing is for, A vendor may create multiple listings for the same product but with different conditions, such as new vs. used items.
 - Availability: Product is available or not? (Sold, Not Sold)
 - Condition (Perfect Condition, Almost New?, Used): state of the product
 - Price: The suggested selling price from the vendor.
 - Discount: percentage of price which the vendor is willing to reduce
 - Listing Status: Keeps track of the age of the listing, and allows the system to flag old listings to inquire if they’re still valid or sold.
 - Approval Status: (pending, approved, denied) Keeps track of the approval status of the listing by admins.
- **Direct Message:** *Allows the buyer, vendor and courier to communicate*
 - Sender: This will be identified whether the message is sent by a buyer, vendor, or courier
 - Receiver: This will identify whether the message is received by a buyer, vendor or courier
 - Content: The text of the message sent by the sender
 - TimeStamp: Records the date and time when message was sent
- **Delivery Request:** *Is a request sent from the buyer side to have a product they plan to buy delivered. (optional)*
 - Listing: A reference to the specific listing that is requested to be delivered.
 - Buyer: Keep track of which buyer is trying to buy the product, and who the delivery would be addressed to.
 - Vendor: Keep track of which vendor this product is coming from so the request can be processed by them.
 - Status: (pending, approved, denied) Keeps track of the approval status of the listing by the vendor.
 - Dropoff Building: The area that the courier will pickup the product for delivery

- **Delivery Instruction:** *Happens when a vendor accepts Delivery Request, handles the specifics of delivery for the courier, as well as the status tracking and age of the instruction for the buyer. (optional)*
 - Time Created: Keeps track of the age of each delivery so as to make sure that older deliveries aren't getting left open, can help in designating deliveries to couriers.
 - Pickup Location: The area that the courier will pickup the product for delivery
 - Dropoff Building: SFSU building where the order will be delivered
 - Product: A reference to the product that will be delivered by the courier
 - Quantity: The number of the product that will be delivered
 - Buyer special request: Description of special instructions from the buyer regarding delivery
 - Vendor special request: Description of special instructions from the vendor regarding delivery
 - Delivery Status: Will show the current stage of the delivery (unassigned, assigned, picked up, delivered)
 - Courier: The user responsible for handling the delivery
 - Vendor: Keep track of which vendor this product is coming from in case contact needs to happen
 - Buyer: Keeps track of who the package is addressed to so the delivery goes to the correct person.
- **Administrator:** *A user responsible for managing the buy and sell website, where they approve or reject user product listing and post*
 - Listing Reviewed: The records of listing that have been approved or rejected
- **Unregistered User:** *A visitor who can browse the platform and create an account but does not have a stored personal information until they have completed the signup page*

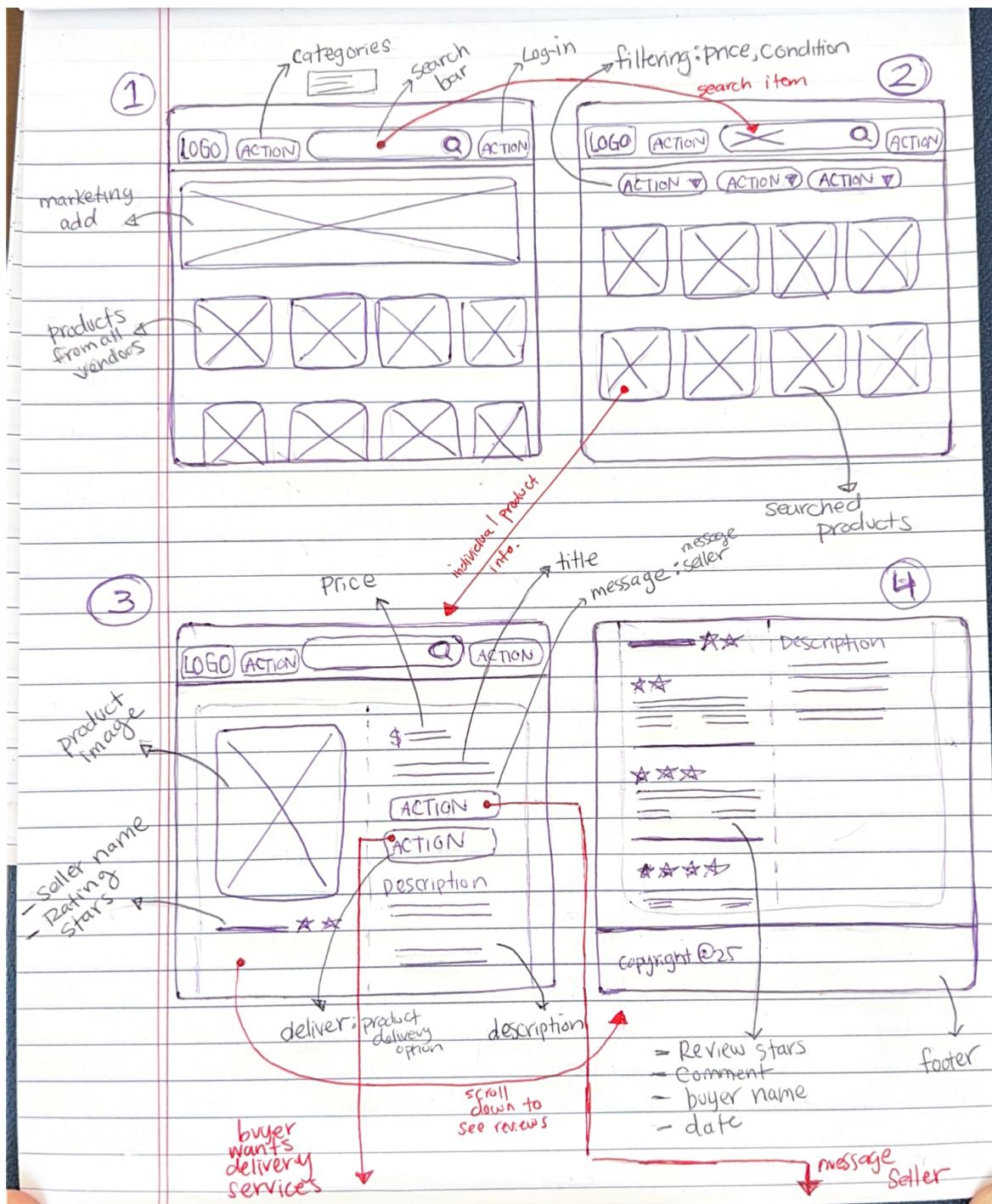
High-level Functioning Requirements

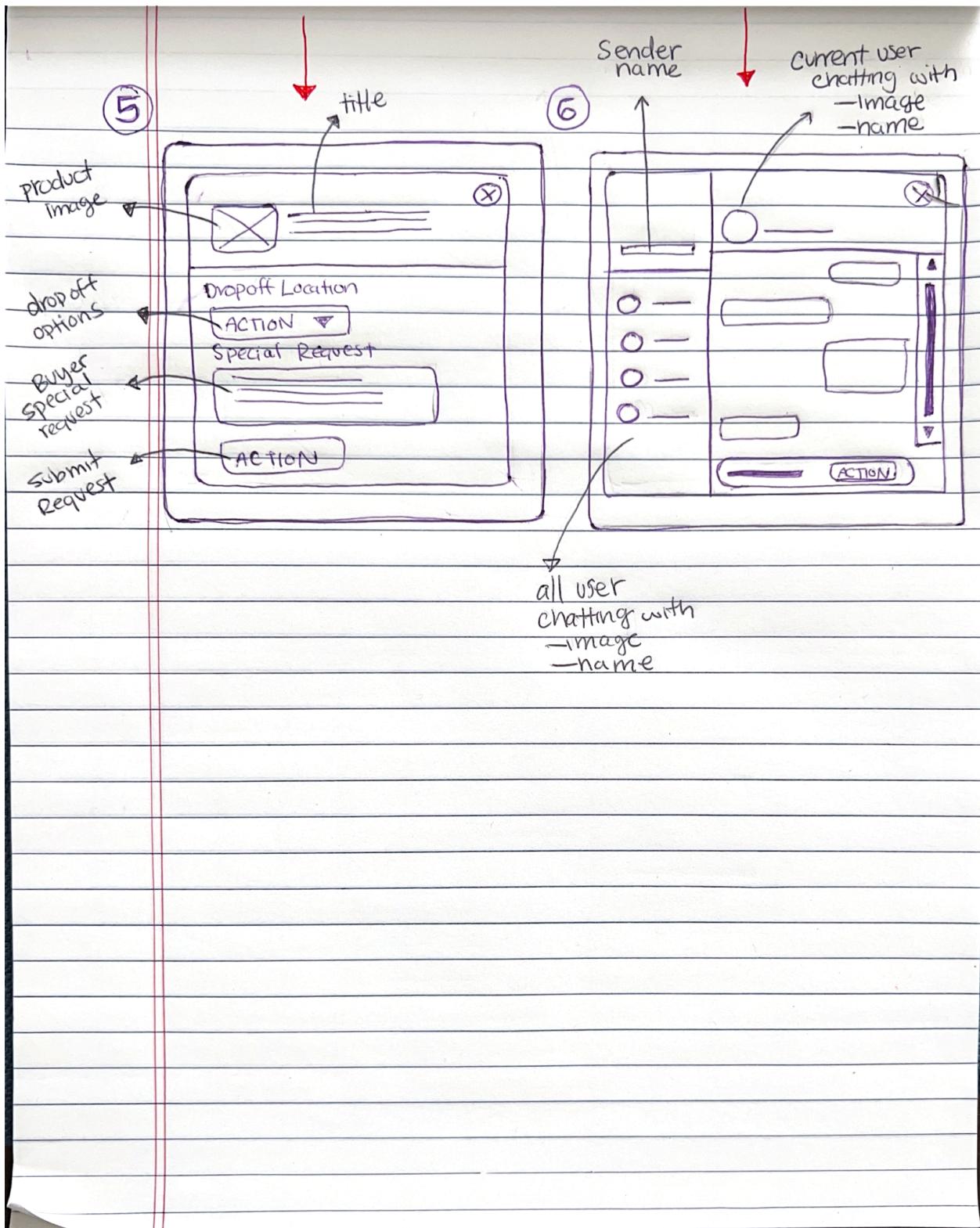
1. Priority 1
 - 1.1 Unregistered users shall be able to create an account.
 - 1.2 Unregistered users shall be able to provide verification of SFSU status with an SFSU email.
 - 1.3 All users shall be able to search for a product using categories.
 - 1.4 All users shall be able to search for a product using its name.
 - 1.5 All users shall be able to view listings.
 - 1.6 Buyers, vendors and couriers shall be registered users.
 - 1.7 A Vendor shall be able to create a listing for at least one product.
 - 1.8 A Vendor shall be able to delete listings.
 - 1.9 A Registered user shall be able to directly message other registered users.
 - 1.10 A Courier shall update their “Available to Deliver” status.
 - 1.11 A Courier shall be able to make many deliveries.
 - 1.12 Admins shall be able to approve new posts.
 - 1.13 System shall require all listings to be approved after creation
2. Priority 2
 - 2.1 All users shall be able to filter their searches for a more specific product.
 - 2.2 A Buyer shall be able to write at least one review for a product.
 - 2.3 A Buyer shall be able to post at most one rating for a product.
 - 2.4 Buyers shall be able to access their past delivered products.
 - 2.5 Vendors shall be able to see their sold products.
3. Priority 3
 - 3.1 A Vendor shall be able to update the discount on a product.
 - 3.2 A Vendor shall be able to view sales analytics.
 - 3.3 A Vendor shall be able to autofill products in listings with products they have already sold before.

UI Storyboards for Each Main Use Case

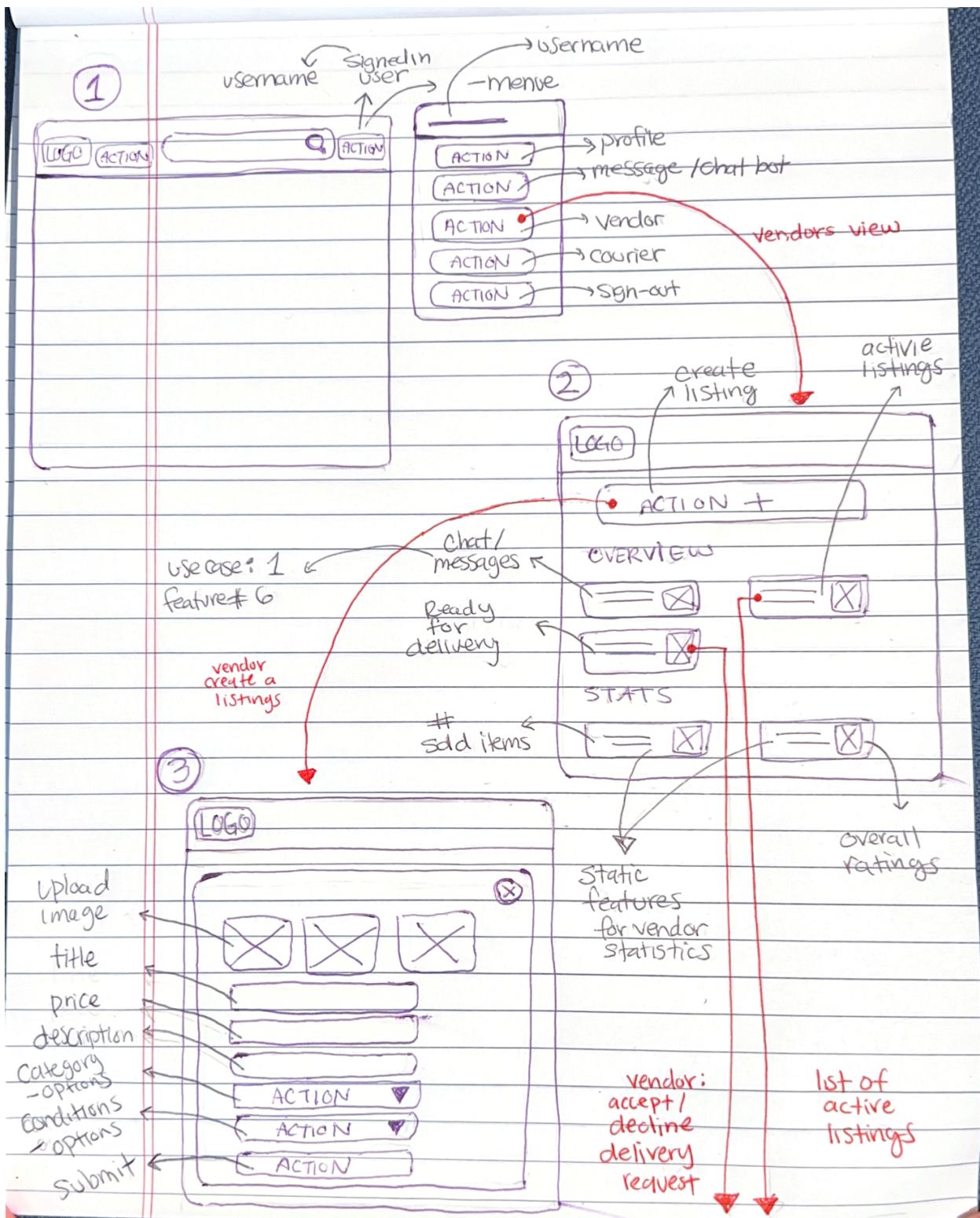
(low-fidelity B&W wire diagrams only)

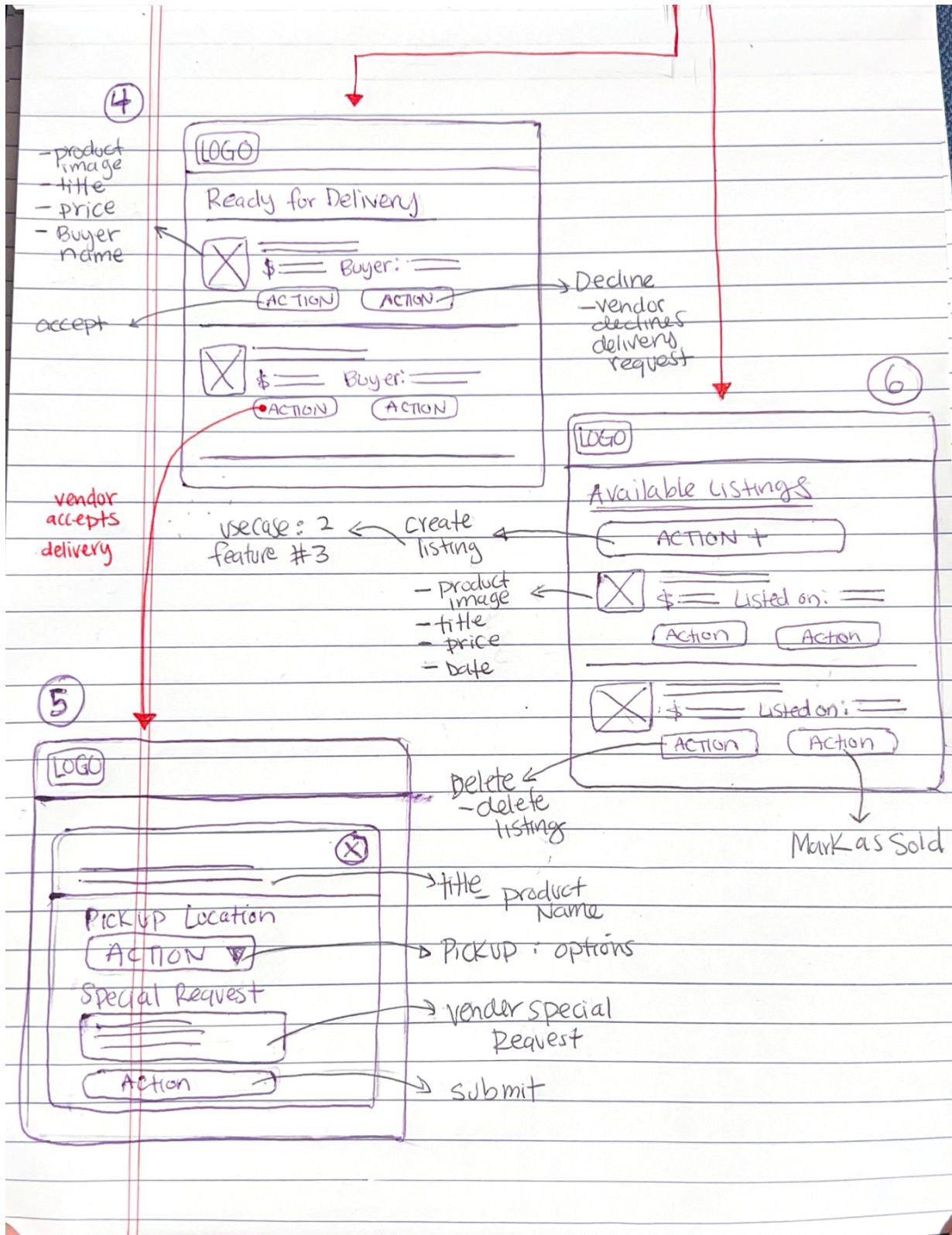
Use Case 1: A professor buying furniture





Use Case 2: Graduating Student Vendor





Use-Case 3: Professor Ariel is selling her teaching materials



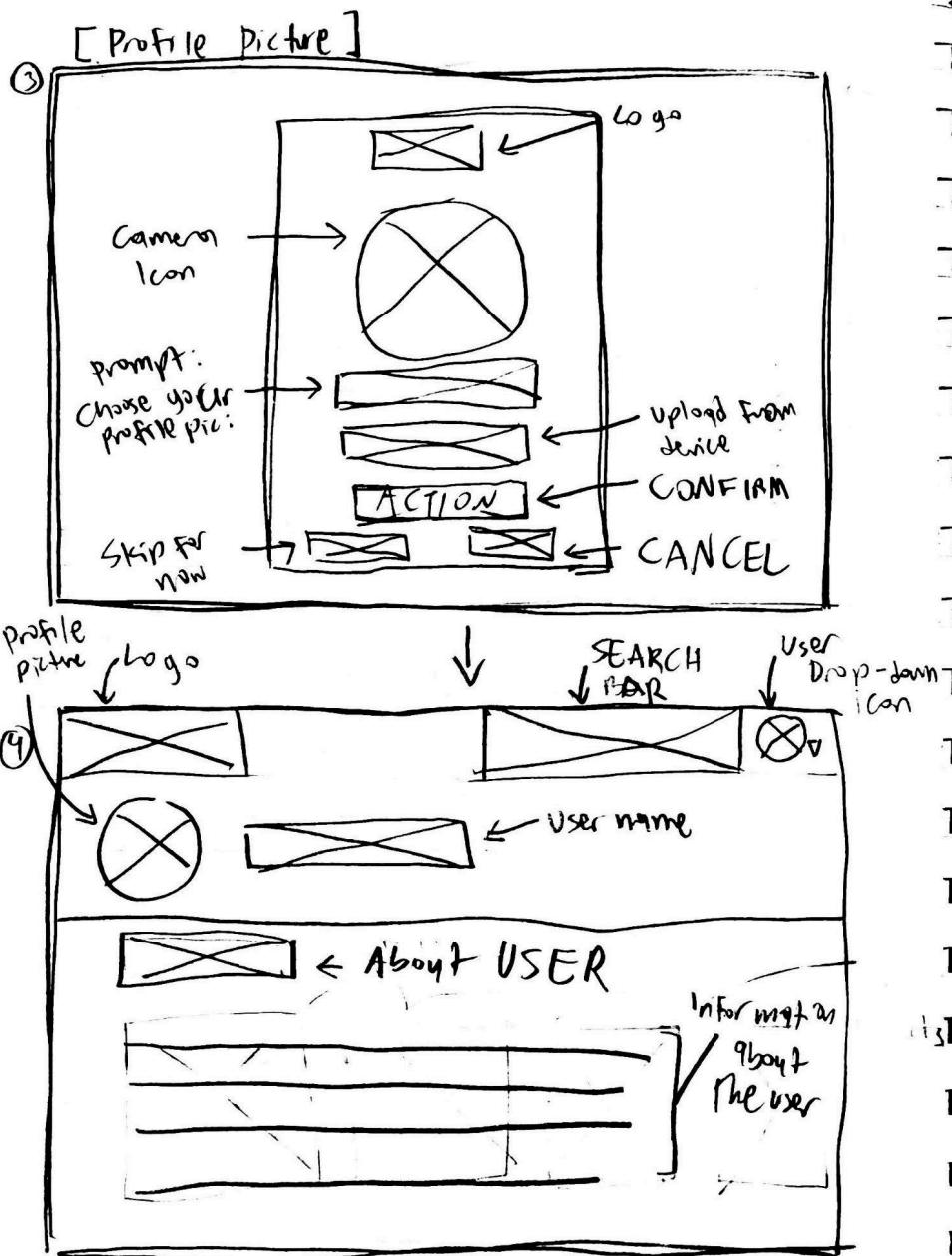
① Use Case 3: Ariel [Sign in Page]

A hand-drawn diagram of a sign-in page. At the top right is a logo. Below it is a large rectangular input field containing five rows, each with a red X over it. To the left of the input field is the text "Enter SFSU email" with an arrow pointing to the first row. Below that is "Enter password" with an arrow pointing to the second row. To the right of the input field is a button labeled "LOG-IN." with an arrow pointing to the third row. Below the input field is a link "Forgot Password?" with an arrow pointing to the fourth row. To the right of the input field is a link "Don't have an account?" with an arrow pointing to the fifth row. At the bottom left is a "CONTINUE" button.

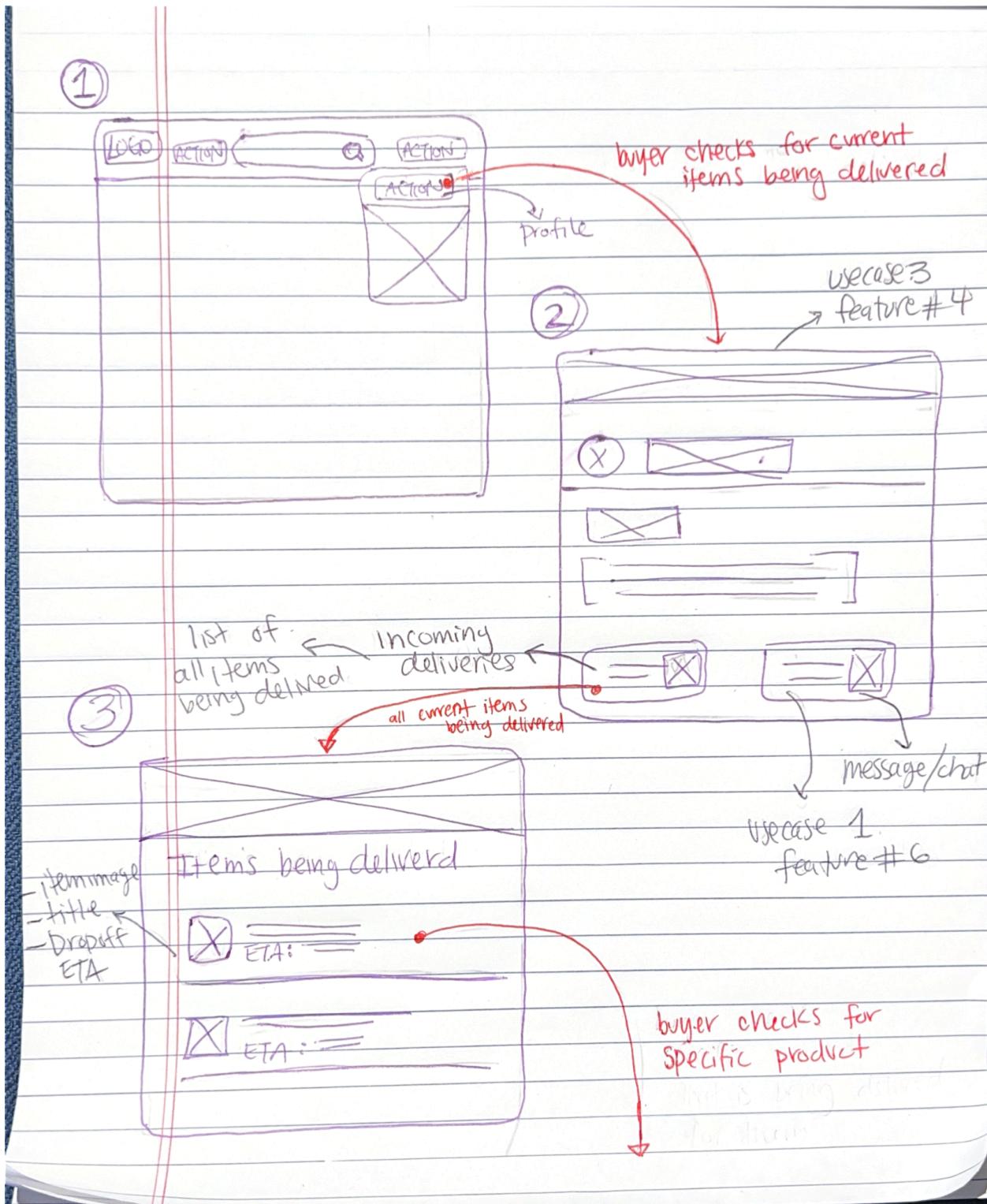


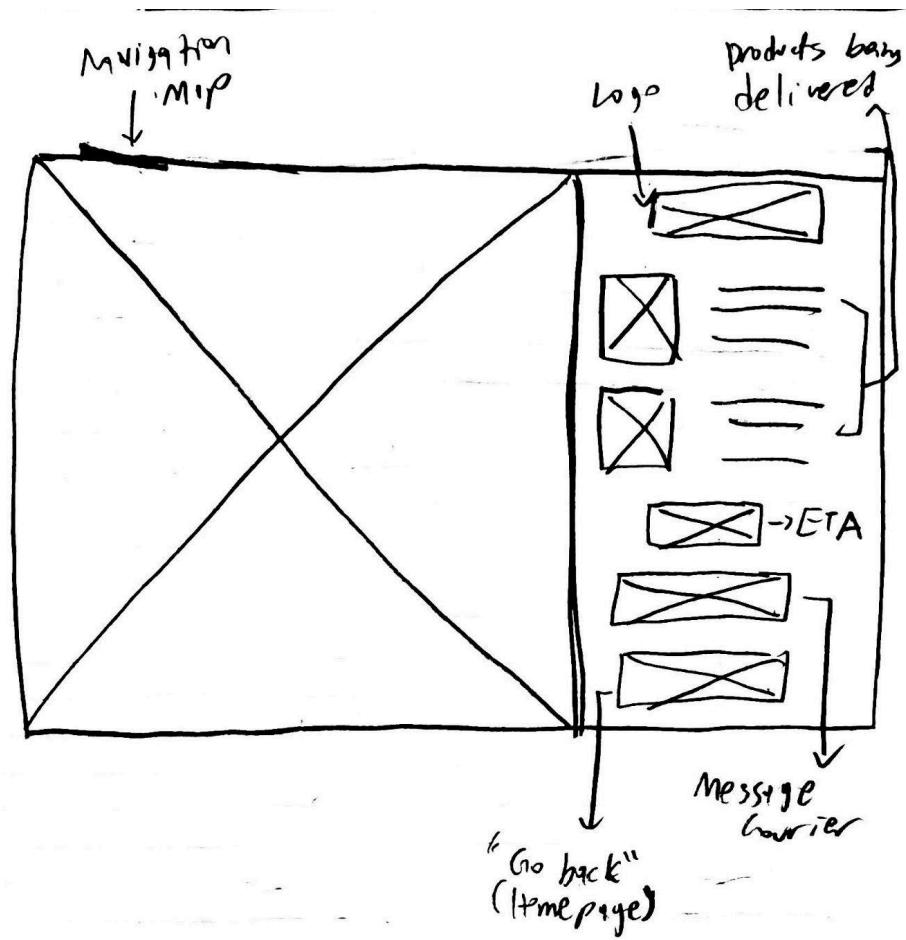
② [Sign-up Page]

A hand-drawn diagram of a sign-up page. At the top right is a logo. Below it is a large rectangular input field containing five rows, each with a red X over it. To the left of the input field is "First name" with an arrow pointing to the first row. Below that is "SFSU email" with an arrow pointing to the second row. Below that is "Password" with an arrow pointing to the third row. To the right of the input field is "Last name" with an arrow pointing to the fourth row. Below that is "User name" with an arrow pointing to the fifth row. To the right of the input field is "Confirm password" with an arrow pointing to the fourth row. To the right of the input field is "CANCEL" with an arrow pointing to the fifth row. At the bottom left is a "CONTINUE" button, and at the bottom center is an "ACTION" button.

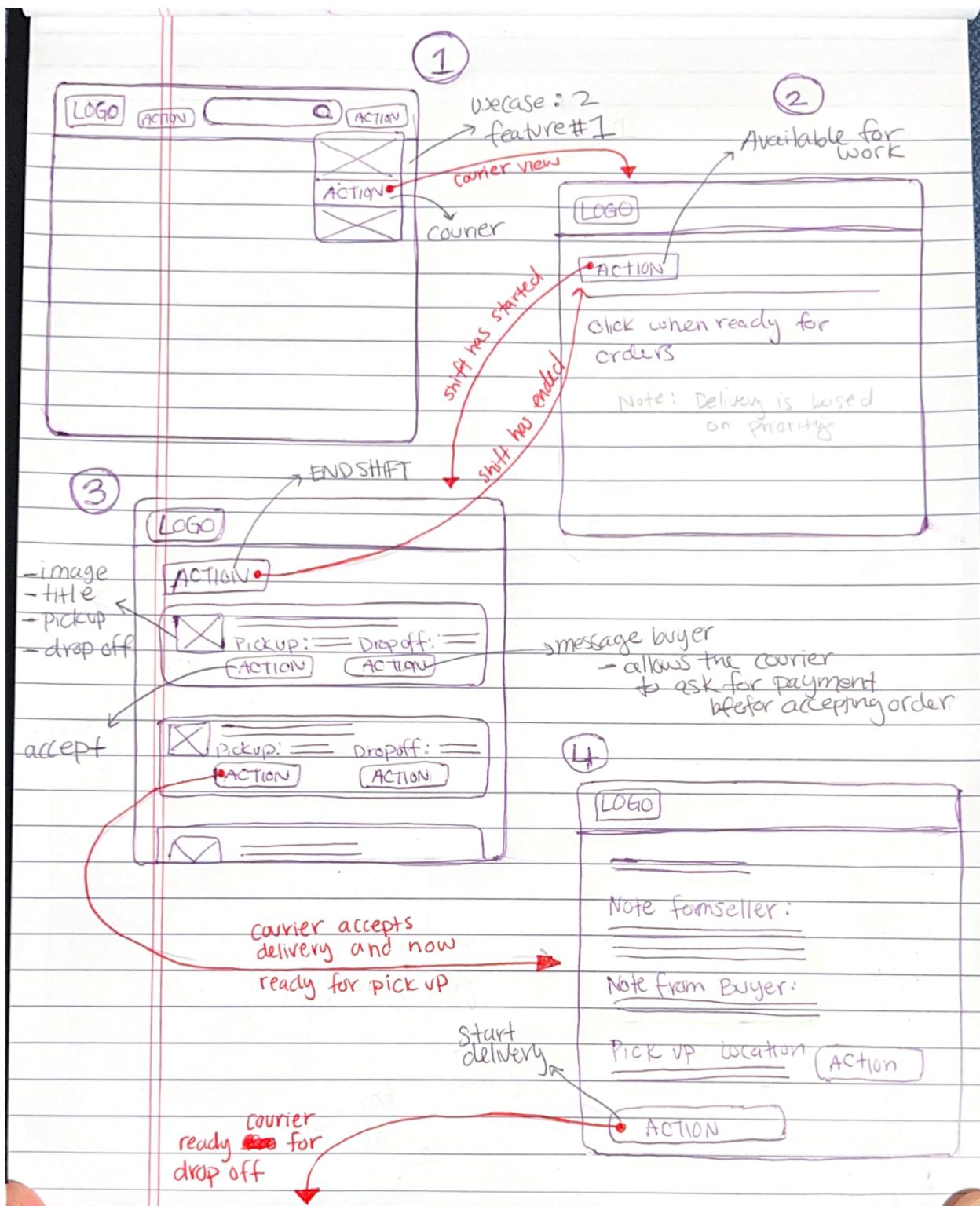


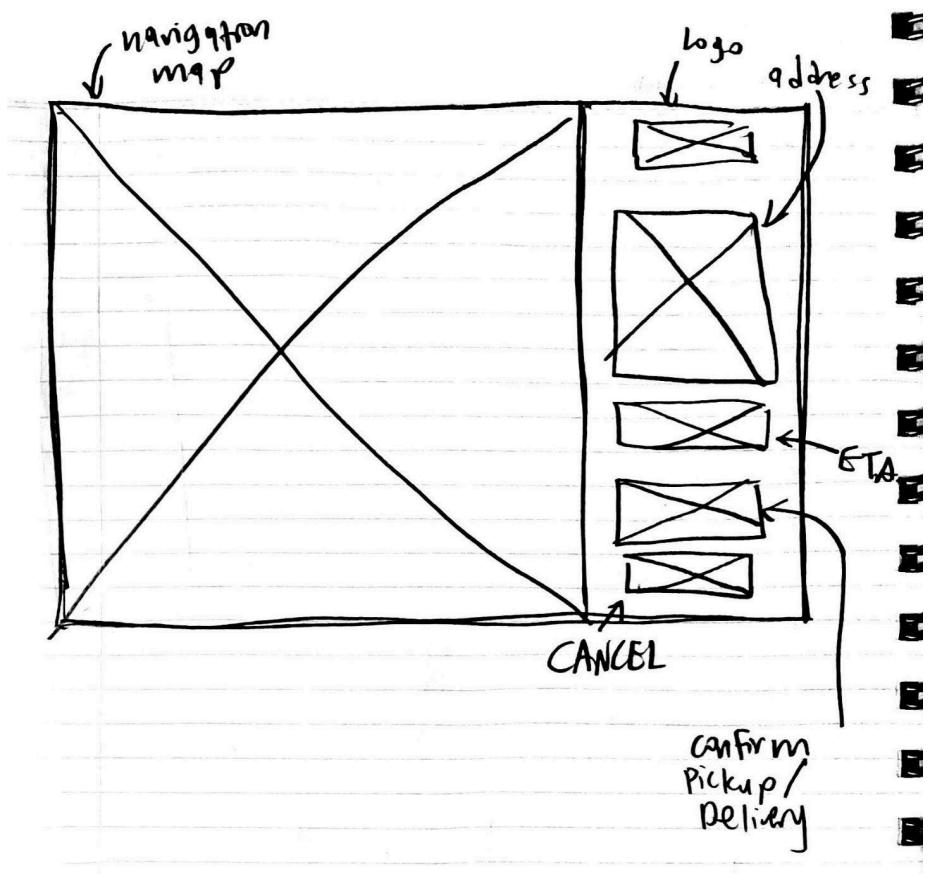
Use-Case 4: Alex wants to buy a product/food





Use Case 5: Fulfilling of Delivery Request





Admin UI

No mockup needed at this time. Admin UI will be worked on using Workbench.

High level Architecture, Database Organization summary only

1. Users Table

Stores information related to users (both registered and unregistered).

- **Table Name:** users
- **Columns:**
 - user_id (Primary Key) - Unique identifier for each user.
 - first_name - First name of the user.
 - last_name - Last name of the user.
 - username - Chosen username by the user.
 - password - Hashed password for user authentication.
 - sfsu_email - SFSU email (for verification purposes).
 - registration_date - Timestamp when the user registered.
 - is_verified - Boolean indicating if the user's email is verified.

2. Vendors Table

Stores information about vendors.

- **Table Name:** vendors
- **Columns:**
 - vendor_id (Primary Key) - Unique identifier for each vendor.
 - rating - An average score of the vendor based on buyer reviews.
 - image_url - URL of the vendor's image (if using a file system for media storage).
 - user_id (Foreign Key) - Reference to the users table, identifying the vendor selling this product.
 - listing (Foreign Key) - A reference to all active listings made by the vendor.
 - sold_item (Foreign Key) - A reference to all previously sold listings by the vendor.
 - pending_delivery_request (Foreign Key) - A reference to any delivery requests for current listings by potential buyers.

3. Buyers Table

Stores information about buyers.

- **Table Name:** buyers
- **Columns:**
 - buyer_id (Primary Key) - Unique identifier for each buyer.
 - image_url - URL of the buyer's image (if using a file system for media storage).
 - user_id (Foreign Key) - Reference to the users table, identifying the buyer

- past_delivered (Foreign Key) - A reference to previous products that have been delivered to the user.
- incoming_package (Foreign Key) - A reference to all delivery instructions that are addressed to the buyer and not delivered yet.

4. Courier Table

Stores information about couriers.

- **Table Name:** courier
- **Columns:**
 - courier_id (Primary Key) - Unique identifier for each courier.
 - image_url - URL of the courier's image (if using a file system for media storage).
 - user_id (Foreign Key) - Reference to the users table, identifying the vendor selling this product.
 - availability_status - Status of Couriers ability to work (e.g., "Available", "Unavailable").
 - Delivery_instruction_id (Foreign Key) - A reference to an active delivery instruction that the courier is responsible for carrying out.

5. Products Table

Stores information about products listed by vendors.

- **Table Name:** products
- **Columns:**
 - product_id (Primary Key) - Unique identifier for each product.
 - description - A brief description of the product.
 - category - An enumerated category that allows the search function to filter by type of product (clothing, food, furniture, electronics).
 - title - A name for the product.
 - image_url - URL of the product's image (if using a file system for media storage).
 - vendor_id (Foreign Key) - Reference to the users table, identifying the vendor selling this product.

6. Listings Table

Stores the product listings that vendors post.

- **Table Name:** listings
- **Columns:**
 - listing_id (Primary Key) - Unique identifier for each listing.
 - listing_status - The listing status (e.g., "Active", "Sold", "Delisted")
 - product_id (Foreign Key) - Reference to the products table.
 - availability - Status of the product (e.g., "In Stock", "Out of Stock").
 - price - Selling price for this specific listing.

- discount - Discount percentage applied to the product (if applicable).
- approval_status - The approval status of the listing (e.g., "Pending", "Approved", "Denied").
- listing_date - Timestamp when the listing was posted.
- Reviewer (Foreign Key) - A reference to the admin that approved or denied the listing.

7. Reviews Table

Stores reviews left by buyers for vendors.

- **Table Name:** reviews
- **Columns:**
 - review_id (Primary Key) - Unique identifier for each review.
 - author_id (Foreign Key) - Reference to the users table, identifying the buyer who wrote the review.
 - vendor_id (Foreign Key) - Reference to the users table, identifying the vendor the review is about.
 - rating - Rating given by the buyer (1-5 stars).
 - comment - Optional written feedback from the buyer.
 - review_date - Timestamp when the review was submitted.

8. Direct Messages Table

Stores messages between buyers, vendors, and couriers.

- **Table Name:** direct_messages
- **Columns:**
 - message_id (Primary Key) - Unique identifier for each message.
 - sender_id (Foreign Key) - Reference to the users table, identifying the sender.
 - receiver_id (Foreign Key) - Reference to the users table, identifying the receiver.
 - content - The text content of the message.
 - timestamp - Timestamp when the message was sent.

9. Delivery Instructions Table

Stores the delivery details for a transaction, managed by a courier.

- **Table Name:** delivery_instructions
- **Columns:**
 - delivery_id (Primary Key) - Unique identifier for the delivery instruction.
 - courier_id (Foreign Key) - Reference to the users table, identifying the courier assigned to the delivery.

- buyer_id (Foreign Key) - A reference to the buyer that the delivery is addressed to.
- product_id (Foreign Key) - A reference to the product that is being picked up from the vendor.
- pickup - Identifies the area from which a product will be picked up from.
- dropoff - SFSU building where the product will be delivered.
- quantity - A count of the number of a product being picked up for delivery.
- buyer_special_request - Any special instructions for delivery from the buyer.
- vendor_special_request - Any special instructions for delivery from the vendor.
- delivery_status - An enumeration that will display whether a delivery instruction has been assigned or unassigned, picked up or delivered.
- timestamp - Timestamp when the delivery was started.

10. Administrators Table

Stores information related to administrators who manage the platform.

- **Table Name:** administrators
- **Columns:**
 - admin_id (Primary Key) - Unique identifier for each admin.
 - user_id (Foreign Key) - Reference to the users table.
 - listing_reviewed (Foreign Key) - A reference to all listings approved by the admin.
 - role - Administrator's specific role (e.g., "Super Admin", "Listing Moderator").

11. Delivery Request

Provides a pending stage for a vendor to approve for a potential buyer with the buyer's information.

- **Table Name:** Request
- **Columns:**
 - delivery_request_id (Primary Key) - Unique identifier for the delivery request.
 - buyer_id (Foreign Key) - Reference to buyer's table, identifying the buyer associated with the delivery request.
 - vendor_id (Foreign Key) - Reference to the vendor's table, identifying the vendor associated with the potential delivery.
 - status - A stored approval state identifying if this request is approved to become an instruction or not.
 - dropoff_building - A buyer determined location on campus that they would prefer it get delivered to.
 - listing_id (Foreign Key) - A reference to the specific listing that is chosen to be delivered.

Media Storage: File System vs. BLOBs

- **Images and Video/Audio:**
 - Images will likely be stored in the **file system**, with URLs stored in the database (e.g., image_url in the products table).
 - **Video and Audio** content, if needed, will likely be handled in a similar fashion using file storage. However, large video/audio files could be stored as **BLOBs** in the database if required.

Search/filter architecture and implementation:

We will be using SQL and the %like operator in order to allow users to search through our database for listings. We will organize search terms using a keyword based approach, where the user can input a word into the search bar, and if it is a substring of any named product, it will appear. We will also have predetermined categories such as furniture, electronics, food, and clothing in order to allow further / alternate filtering for the search feature.

The following DB terms can be searched and filtered by:

product.title: Name of a product.

product.category: The type of product.

vendor.username: The name of the vendor selling the product.

The code for a search query would look similar to the following:

```
SELECT p.title, p.category, u.username AS vendor_username, v.vendor_id, u.user_id  
FROM products p  
JOIN vendors v ON p.vendor_id = v.vendor_id  
JOIN users u ON v.user_id = u.user_id  
WHERE p.title LIKE '%search_term%'  
    OR p.category LIKE '%search_term%'  
    OR u.username LIKE '%search_term%';
```

This code would allow a general query to be matched with their respective field given our architecture.

Significant Algorithms or Processes

1. Rating Algorithm:

- The rating system will allow buyers to rate vendors using a 1-5 star scale. The system will calculate an average rating for each vendor based on all reviews and display it on the vendor's page.

2. Approval Process:

- Vendor product listings will go through an approval process, where an admin must approve or reject a listing. This is tracked by the approval_status column in the listings table.

3. Search Algorithm:

- The search filter will be based on the title (as provided by the vendor). It will perform an exact match search, meaning that results will only be returned if the title contains the specified word(s).

4. Delivery Order Filtering:

- Products will be filtered to determine which needs to be delivered first by following a FIFO (First In, First Out) approach, ensuring that the earliest received items are delivered first.

Identify Actual Key Risks For Your Project at This Time

- **Skills Risk:** Our team, made up of students, has a good basic understanding of programming and software, but some of us haven't worked with more advanced tools or techniques. This gap means we might take longer to complete certain parts of the project, and we may need extra time for learning, group help, or advice from our professors.
- **Schedule Risk:** Given our project deadlines and the interdependent nature of our work, any delay in one phase could have a domino effect on our overall timeline. Tight deadlines mean that unforeseen issues or slower than expected progress in one segment could jeopardize our ability to complete the project on time, making careful planning and contingency measures essential.
- **Technical Risk:** Our project will use different software tools and might include external APIs that could have compatibility issues or bugs. Since we are students and may not have immediate access to expert technical support, these unexpected problems could slow our progress unless we carefully test everything early on.
- **Teamwork Risk:** Working as a team of students from different disciplines and possibly varying schedules poses a risk to our project's cohesion and productivity. Challenges like miscommunication, coordination difficulties, or uneven workload distribution can impact our overall progress. To mitigate these risks, we need to establish clear roles, consistent communication channels, and regular meetings to ensure everyone is aligned and contributing effectively.
- **Legal/Content Risk:** The project requires the use of various third-party resources, including content and software, which must be properly licensed and accredited. Ensuring that all external materials comply with copyright and licensing requirements is crucial to avoid academic and legal complications. We must diligently verify that all resources are legally obtained and appropriately cited to safeguard the integrity of our project.

Project Management

To enhance task management and improve efficiency, we will gather feedback from the team to identify what is working and what needs improvement. This feedback will help us refine our process and look for additional tools if needed.

To ensure our task management is organized and efficient, we are currently using the following tools:

1. **Trello**: Trello is our primary task management tool that allows us to assign tasks, set deadlines, and track progress. As project manager, I will ensure that each task card is updated regularly with status changes and send follow-ups to the team to keep everyone informed.
2. **Discord**: Discord serves as our communication platform, allowing real-time discussion and updates. Regular check-ins in both group and personal chat helps us stay aligned on task progress and address any challenges.
3. **Google Drive**: Google Drive houses all our essential project documentation, such as webpage inspiration, email drafts, weekly meeting agendas, brainstorming notes, module documentation. This ensures that all relevant information is well-organized and easily accessible for team members.

Use of GenAI Tools

For this project, our entire team completed the research, drafting, and revisions through our own collaborative efforts, drawing on our collective knowledge and resources without using any AI tools. Each member contributed ideas, participated in group discussions, and provided feedback during peer reviews to ensure the work remained original and accurate. We are solely responsible for the content presented here, adhering to academic integrity and demonstrating our commitment to producing genuine team-based results without automated or AI-generated assistance.

Team Lead Checklist

*For each item below the team lead must answer with only one of the following:
DONE/OK; or **ON TRACK** (meaning it will be done on time, and no issues perceived);
or **ISSUE** (you have some problems, and then define what is the problem with 1-3
lines).*

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