SW Engineering CSC648-848 Spring 2025
Team 16: “Gator Goods”
Milestone 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
M1	March 12, 2025	March 22, 2025

Table of Contents

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 sellers. 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.

Personae

Persona Summary: Jake (Buyer - Non-technical Professor)

(Generated by ChatGPT, 2025)

About Jake:

- Middle age professor who has basic tech proficiency, such as emails, social media, some Canvas understanding
- Does not like complex application, prefers simple but quick interface due to busy schedule
- Typically buys breakfast and lunch from the school cafeteria or nearby vendors
- Prefers quick and efficient service
- Would like to share favorite items with students or other teachers

Goals and Scenario:

- Decide to find affordable teaching materials for students and wants to quickly find the items using a search bar to see if any items are being sold by seller on campus instead navigating complex menus to find items
- Quickly needs a meal before class starts, would like to schedule a specific time and pickup location for food delivery
- Ordered items and he would like to track purchase to know exactly when it will arrive
- Wants to ensure item quality and trustworthiness of the seller by viewing rating and customer reviews before making a purchase

Persona Summary: John Taylor (Seller - Non-technical Vendor)

(Generated by ChatGPT, 2025)

About John:

- Age: 45
- Not really technically proficient, has a hard time navigating sites
- Owner of a cafe on SFSU campus
- Lives in SF
- Long time business owner, so he knows how a cafe should run
- Doesn't have time to learn the intricacies of apps

Goals and Scenario:

- Is a long time owner of a cafe on campus, and has decided that he wants to use the app to sell to more customers in order to keep his profits high
- Wants to make as much profit from his business as possible
- Needs to sell his food to as many students as he can on campus
- Wants to implement a "order ahead" app that allows for food pickups from his store to various paying customers around the campus
- Doesn't want to have to navigate too many layers of an app in order to process orders, would much rather be linked directly to an orders page
- Doesn't want to have to worry about constantly making posts to advertise their product

Persona Summary: Ariel (Seller - Non-technical Professor)

(Generated by ChatGPT, 2025)

About Ariel:

- Teacher at SFSU, perhaps not well-versed in technology
- Age range likely around 40s-50s(?)
- She's very knowledgeable and passionate about the subject she teaches, although very tired from the long strenuous years of being in SFSU.
- The way she go about things in her classes could be considered "old-school" (pen-and-paper assignments/quizzes, reading from a physical and seemingly older edition of a textbook, and discussions with very little usage of a laptop/projector, oftentimes preferring the faithful method of using marker and whiteboard)
- Despite this, she does have basic knowledge of HOW to use websites and other social media, though very likely she doesn't frequent it unless necessary.

Goals and Scenario:

- Having found out of the new SFSU-centric website, where students and faculty alike can sell and purchase products, Ariel's interest may be piqued.
- This would likely be a way to connect with others around campus– even giving her a means to sell to other teachers who may need their stuff more.
    - This could also be the perfect place to get rid of any class material she no longer needs (textbooks, extra papers,teaching material, etc)
- There is now an incentive to learn how to use this website as a seller, but also due to the centered community social aspect of it.

Persona Summary: Alex (Buyer - Technical Student)

(Generated by ChatGPT, 2025)

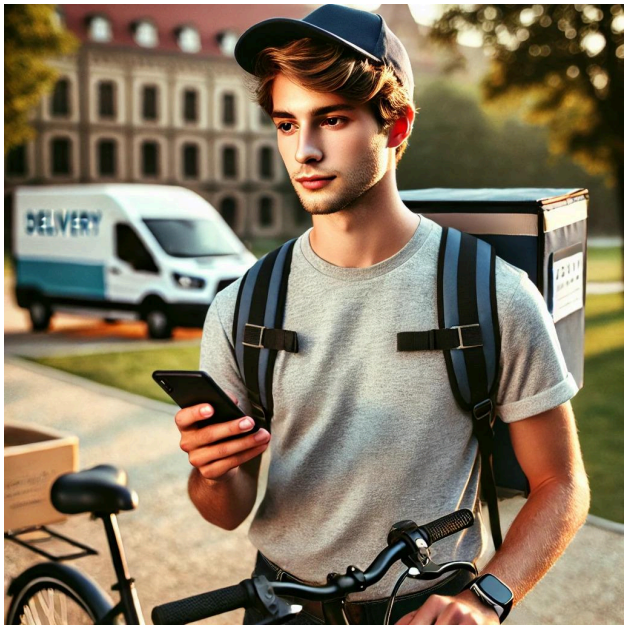
About Alex:

- Enrolled as a Full-Time Student in Computer Science/Engineering.
- Comfortable with mobile applications and technology.
- Prefers seamless and intuitive interfaces.
- Oftenly busy with classes, projects, meetings and events.
- Irregular meal schedules.
- Familiar with online payment systems, cashback offers and loyalty programs.

Goals and Scenario:

- Wants to easily buy/sell tech gadgets, books, and accessories or order food with a smooth, real-time updated interface.
- Looks for student discounts, affordable deals, and low-cost delivery while avoiding hidden fees.
- Prefers verified users and reliable peer-to-peer transactions to avoid scams and ensure safe purchases.

Persona Summary: Steven (Student Courier)



(Generated by ChatGPT, 2025)

About Steven:

- Steven is our main avenue for delivery services.
- Steven has average technical skills (knowledge of login processes, emails, moderate social media user)
- Student in his early 20's likely a Junior/Sophomore
- Balances schoolwork with his part time delivery job
- Prefers apps that are intuitive and easy to navigate

Goals and Scenario:

- Needs a clear, real-time map and estimated delivery times to optimize his routes
- Prefers a system that allows him to manage his delivery routes and schedule
- Needs a clear concise map displaying the pickup locations as well as delivery locations.
- Uses a bike/scooter to get around campus for deliveries
- Wants to ensure he gets fair payment and tips for his services
- Wants to receive and accept delivery requests through an intuitive system

High-Level Use Cases

Use Case 1: A professor buying furniture

Jake is an instructor at SFU with limited technical skills. He wants to buy furniture for his office but has a busy schedule and no time to visit an in-person store. Instead, he turns to the university Buy and Sell website, which serves SFSU faculty, staff, and students. The app provides a convenient way to browse available furniture and communicate with sellers before making a purchase

Jake opens the app and uses the search bar to find a specific type of **product**, such as a chair. Since convenience and availability are his top priorities, he applies filters to refine his search results. These filters help him narrow down the **listings** based on factors like price, condition, and delivery methods. After refining the listings, he browses through them, **checking images, prices, and seller information**. He wants to ensure that the chair he chooses meets his needs, so he carefully examines the product descriptions and looks for seller reviews. Positive **reviews** from previous buyers reassures him, while negative reviews make him more cautious.

Once he finds a chair that matches his preferences, he takes a closer look at the details. He checks the description to see if the chair is new or used and whether it has any visible damage. To ensure he is making the right choice, he decides to contact the seller directly. He sends a **message** asking about the chair's condition, dimensions, and any defects that may not be visible in the photos. The seller responds promptly, providing additional details and confirming that the chair is in good condition.

Now that Jake is confident in his purchase, he considers his options for receiving the chair. He can either pick it up in person or have it delivered to his office. Since he has a packed schedule and doesn't want to risk missing a pickup time, he opts for delivery. He chooses his **delivery instructions**, which includes a time, building, and room number, then leaves a special request for the delivery person to knock on the door. The professor appreciates that the payment options are simple and secure, making the entire transaction smooth and hassle-free

Use Case 2: Graduating Student Vendor

Jake is a graduating student at SFSU. He wants to sell items he no longer needs, such as a chair and a couch, as he is moving out. He first creates an account and becomes a **registered user** on the app, and then logs in. Since he wants to be a **vendor**, he then

lists his **products** with **descriptions, prices, and images**. A **buyer** browsing the app selects his chair and messages the vendor to place an **order**, choosing delivery instead of pickup. The app processes the **order** and notifies Jake when the **seller** confirms the **order**. He prepares the furniture for delivery and uses the app to coordinate with the assigned **courier**. Once the furniture is delivered, the delivery driver confirms the delivery (**buyer** confirms receipt) and Jake receives his payment. Later, the **buyer** may leave a **review** detailing his experience with the purchase and Jake.

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

Ariel, as a **vendor**, would like to present a **product** (teaching materials, such as papers, notes, agendas, etc) that she wants to sell on this website and that this will mostly be centered around SFSU faculty. She creates an account to start a new seller's product page, newly registered on the website. From there, Ariel creates a **listing** of her **product**, customized with an image (perhaps more than one) and a description. She provides a picture of her **product** as well as a description on what it is, the uses of it, and its price. After putting down how much of the **product** she has for stock, the new **listing** of her **product** is now public for any potential **buyers** to see it and express their interest.

Upon interest piqued and a deal made between herself and **buyer**, Ariel is assigned a **courier**, to which she'll be able to frequently update on for **delivery instructions**. This continues up to the transaction made. For future reference, rinse and repeat until out of **product.**

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

Alex, a busy technical student, uses the app to quickly order food or buy tech essentials like laptops, accessories, or textbooks. With a tight budget, Alex looks for affordable deals, filters products by price and **condition**, and picks the best option. For food, Alex selects a restaurant, special request for the order, and checks estimated delivery **time**. After placing an order, Alex gets **real-time updates** and can **chat** with the seller or delivery driver for any questions. If picking up an item, Alex arranges a meet-up time with the seller. Once the food or product is received, Alex checks the **condition**, confirms the order, and leaves a **review**. This smooth process saves time, ensures secure transactions, and makes buying and delivery hassle-free for a busy student.

Use Case 5: Fulfilling of Delivery Request

Steven, a **student courier**, uses the app to accept **special requests** based on his availability. After receiving a new **order** notification, he reviews the pickup location, delivery details, and costs before confirming the job.

Once accepted, the app provides step-by-step navigation via Google Maps, guiding Steven to the seller's location for pickup. He verifies the **order** with the **seller**, pays for the **products** and marks it as "Picked Up" in the system, triggering an update for the buyer.

Since he delivers within the campus area, the app optimizes pedestrian-friendly routes, helping him find the fastest way to the drop-off location. If needed, he can message the buyer for precise handoff coordination.

Upon arrival, Steven confirms the delivery, completes the transaction, and receives payment for the **products** as well as his delivery fee. His successful deliveries are automatically updated in his profile, allowing him to track his progress and have access to previous deliveries if problems may arise.

List of Main Data Items & Entries

Data Glossary/Description

- **Registered user:** *Store their personal information like email, name, transaction history (parent entity of buyer, vendor, courier)*
    - 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
    - Address: A segment of registration in which the User would put in their current address
- **Buyer**: *A registered user that is able to buy products, and access a list of past transactions.*
    - Reviews : A reference to all reviews made by the buyer.
- **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
    - Transaction: A reference to all past transactions made with the vendor
- **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: A reference at which the Courier will be able to schedule themselves as when they are available to make a delivery
- **Product**: *the item in which the seller would like to list.*
    - Description: a brief explanation of the product
    - Price: value of the product determined by seller
    - Condition (Perfect Condition, Almost New?, Used): state of the product
    - Image: a visual representation of the product
    - Discount: percentage of price which the seller is willing to reduce
    - Seller: the user who will list the product
- **Review**: *The Review can be provided by a **Buyer** who has recently made a purchase and will be publicly displayed on product listings to help future buyers make informed decisions.*
    - Rating: *"Star" count [can be a 1-5 scale]*
        - **Not an item,** but think it would be fun to envision/consider making a gator-related gimmick for these reviews/ratings

    - Comment: written feedback from the user
- **Listing**: *A **listing** represents a specific product that a vendor offers for sale within the marketplace.*
    - Stock: Quantity Available
    - 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? (Out of Stock, In stock)
    - Price: selling price
    - Approval Status (pending, approved, denied)
- **Order**: *Represents a collection of products a buyer has added to their cart for purchase*
    - Product: A list of items selected by the user for purchase
    - Amount purchased: A specified amount of each product the user would like to buy
    - Price to-be-paid: The final amount calculated based on the selected items and their prices
- **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 Instruction:** *Tracks the assigned courier and the designated drop-off location for the order*
    - Time: The expected time for the courier to complete the delivery
    - Building: SFSU building where the order will be delivered
    - Room: SFSU room in the building for drop-off
    - Special request: Instructions from the buyer regarding delivery
    - Courier: The user responsible for handling the delivery
- **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
    - Time: Tracks when each review action was performed
- **Unregistered Use**r: *A visitor who can browse the platform and create an account but does not have a stored personal information until they completed the signup page*

High-level Functioning Requirements

1. Unregistered users shall be able to create an account.
2. Unregistered users shall be able to provide verification of SFSU status with an SFSU email.
3. Buyers, vendors and couriers shall be registered users.
4. Registered users shall provide an address for delivery/billing purposes.
5. A Vendor shall be able to create a listing for at least one product.
6. A Vendor shall be able to update the discount on a product.
7. A Vendor shall be able to delete listings.
8. A Vendor shall be able to view sales analytics and transaction history.
9. A Buyer shall be able to write at least one review for a product.
10. A Buyer shall be able to post at most one rating for a product.
11. A Buyer shall be able to directly message at least one Vendor.
12. A Buyer shall be able to make many Transactions.
13. A Buyer shall be able to report an issue with transaction/product.
14. A Courier shall be able to pick-up a product for a Vendor.
15. A Courier shall be able to deliver products to a Buyer.
16. A Courier shall be able to schedule at least one Availability Status.
17. Buyers shall be able to access their past transactions.
18. Vendors shall be able to see their past transactions.
19. System shall require all listing to be approved after creation
20. Admins shall be able to approve new posts.
21. Listings can be viewed by all users.
22. Products can only be added to a Buyer’s cart.

Non-functioning Requirements

High-level non-functional specifications (how the app is delivered and other constraints) that MUST be adhered to:

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 email 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.

Competitive Analysis

(functions/features only, not business or marketing)

Feature	eBay	Amazon	Etsy	Our Future Product
Post a Listing	+	+	+	+
Delivery	+	+	+	+
Messaging	+	-	+	++
Order Status	+	+	+	+
Image Uploading	+	+	+	+

+ Feature exists || **++** superior || **-** does not exist

Feature Descriptions

- **Post a listing**
  - In our app, we would work on the implementation of creating and posting a listing akin to that of our competitors. Visibly similar to that of posting on Etsy, we would make certain that– upon registration and confirmation– a user would be able to post a listing by providing them the option of uploading a picture of their product(s), details such as whether or not the item was handmade/bought/used or new, additional description for said item, as well as listing down how much of their product they have on-hand. Once satisfied with how their product listing looks, we'd include the additional detail of making sure they are analyzed by administration before being confirmed or denied publicity on our app.

- **Delivery**
  - Our application introduces a delivery option to provide a more convenient and efficient experience for buyers and vendors. This feature connects the buyer, vendor, and courier ensuring a smooth transaction process. Once the buyer decides on buying a product the buyer can choose to get the item delivered. Where both the buyer and vendor must agree to proceed with the delivery. After the mutual agreement, the system assigns a courier to pick up the item from the vendor and deliver it to the buyer. The courier

is selected based on availability to ensure the item is transported efficiently. The application provides real-time tracking, allowing the buyer to monitor the status of the delivery. This feature simplifies the buying process by reducing the need for in-person exchanges while ensuring a reliable delivery method.

- **Messaging**
    - For our app, we are implementing a messaging feature that enables a seamless communication between vendor, seller, and couriers throughout the buying and delivery process. Unlike Amazon, our platform allows users to contact sellers directly while browsing product listings, allowing them to ask questions or concerts of a product before making a purchase. Additionally, our messaging system facilitates real-time communication between buyers and couriers during delivery, ensuring that customers can provide instructions or ask updates on the delivery process. Couriers can also reach out to sellers if they need clarification on pickup details. This functionality sets us apart from platforms like Amazon, eBay, and Etsy, which do not offer this level of direct interaction. This messaging feature enhances the experience of all the users using this app.

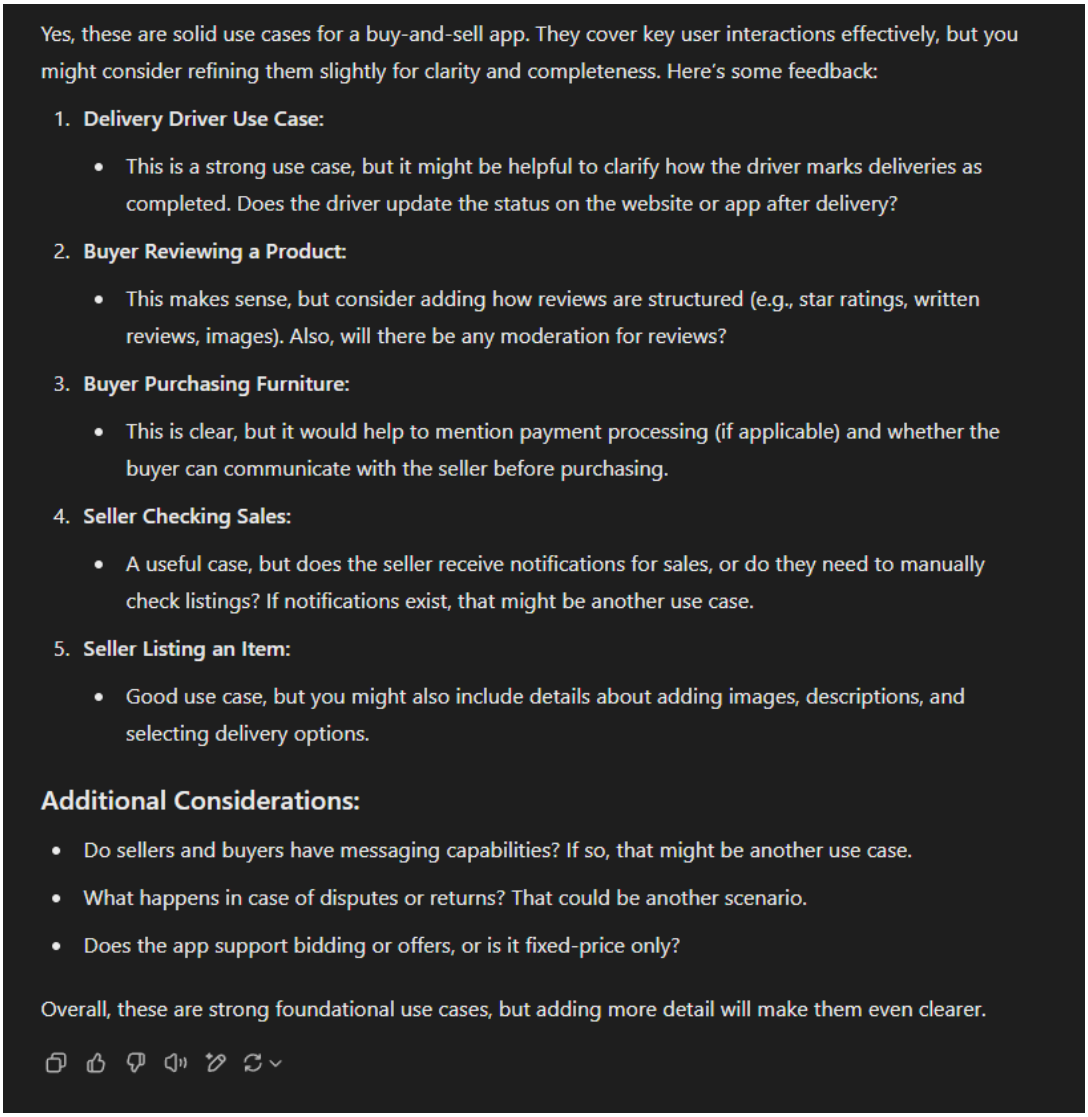
- **Order Status**
    - In our app, the Order Status feature will provide real-time updates at every transaction step with a dedicated ***progress bar*** that clearly tracks each phase from order placement through final delivery. While Amazon uses milestone indicators and eBay/Etsy primarily use text-based updates, our approach offers a more intuitive, visual representation of order movement. This means customers can quickly see where their purchase stands without needing to parse multiple status messages or emails, creating a simpler and more reassuring experience.

- **Image Uploading**
    - Our image uploader is all about making the process quick, hassle free, and high quality, perfect for everything from product listings to photo documentation. Just drag and drop your files, check out a real-time preview, and you're good to go. We automatically optimize each image so it stays sharp, loads fast, and maintains a professional look. Plus, with support for multiple file types and resolutions, you'll have the flexibility to upload exactly what you need.

High-level System Architecture & Technologies

- **Deployment Cloud Server Host**: AWS t2.micro, 1 vCPU, 1 Gib Memory
- **Operating System**: Ubuntu v22.04
- **Database**: MySQL v8.0.40
- **Web Server**: Express.js v5.01
- **Server-Side Language**: JavaScript vES6
- **Additional Technologies:**
    - **Web Framework:** React
    - **IDE**: Visual Studio Code
- **Browser Support**
    - Google Chrome
    - Firefox
- **External Open Source API:**
    - Google Maps API

Use of GenAI Tools



Yes, these are solid use cases for a buy-and-sell app. They cover key user interactions effectively, but you might consider refining them slightly for clarity and completeness. Here's some feedback:

1. **Delivery Driver Use Case:**
   - This is a strong use case, but it might be helpful to clarify how the driver marks deliveries as completed. Does the driver update the status on the website or app after delivery?
2. **Buyer Reviewing a Product:**
   - This makes sense, but consider adding how reviews are structured (e.g., star ratings, written reviews, images). Also, will there be any moderation for reviews?
3. **Buyer Purchasing Furniture:**
   - This is clear, but it would help to mention payment processing (if applicable) and whether the buyer can communicate with the seller before purchasing.
4. **Seller Checking Sales:**
   - A useful case, but does the seller receive notifications for sales, or do they need to manually check listings? If notifications exist, that might be another use case.
5. **Seller Listing an Item:**
   - Good use case, but you might also include details about adding images, descriptions, and selecting delivery options.

Additional Considerations:

- Do sellers and buyers have messaging capabilities? If so, that might be another use case.
- What happens in case of disputes or returns? That could be another scenario.
- Does the app support bidding or offers, or is it fixed-price only?

Overall, these are strong foundational use cases, but adding more detail will make them even clearer.

In Milestone 1, we used ChatGPT to provide feedback on our use cases to ensure we were on the right track and did not require additional changes/review. We took a divide and conquer approach to completing each use case, and some teammates thought the review was helpful. In this case, we found ChatGPT helpful (Medium) as it provided an extra set of eyes before moving on to the next milestone requirements.

Team & Roles

Name	Roles	Front-end or Back-end?	Email
Martha Martin Pablo	Team Lead	Front-end	mmartinpablo@sfsu.edu
Anthony Mingus	Back-end Lead	Back-end	amingus@sfsu.edu
Jaideen-Jace Bondoc	Front-end lead, Documentation Organizer	Front-end	jbondoc3@sfsu.edu
Vansh Virani	GitHub Master	Front-end	vvirani@sfsu.edu
Garvin Zhao	Floater	Front-end and Back-end	gzhao@sfsu.edu

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).*

- All team members are fully engaged and attending team sessions when required
    - Done
- 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
- Team reviewed class slides on requirements and use cases before drafting Milestone 1
    - Done
- Team reviewed non-functional requirements from “How to start…” document and developed Milestone 1 consistently
    - Done
- Team lead checked Milestone 1 document for quality, completeness, formatting and compliance with instructions before the submission
    - Done
- Team lead ensured that all team members read the final M1 and agree/understand it before submission
    - Done
- 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