SW Engineering CSC648-848 Spring 2025

GatorXchange
team-10
Milestone 2 Part I

Angelie Bautista	Team Lead	abautista13@sfsu.edu
Chan-Chun Yen	GitHub Master	cyen@sfsu.edu
Lap Pham	Back End Lead	lpham15@sfsu.edu
Alvin Tulud	Front End Lead	atulud@sfsu.edu

History Table

Date Submitted:	3/25/2025
Date Revised:	N/A

1. Executive Summary

Taking advantage of the demand for buying and selling products or saving costs for students and staff at SFSU, GatorXchange was created to allow everyone working and studying at the university to exchange items they need. This platform maximizes economic benefits, reduces costs, promotes item reuse, and, most importantly, strengthens connections among members of the SFSU community.

With GatorXchange, users can browse currently listed products or filter based on what they want. They can then create an account to upload products they wish to sell or send a message to a seller if they're interested in buying a product. If the seller agrees, they can schedule a meeting on campus to complete a transaction, offering both parties a chance to connect. Admins can moderate the product postings, approving it before it can go live and deleting any inappropriate products or users. Additionally, our app provides the unique feature that generates the recommendation list for users based on a SFSU class number, and the recent search history to improve using experience.

The GatorXchange team consists of four skilled individuals. Angelie Bautista, the Project Lead, has passion for game development and leadership. As president of the Game Development Club, Angelie's enthusiasm for collaboration and experience in management drives the team's vision and progress. Alvin Tulud, the Front-End Lead, specializes in game development and UX/UI design. He is experienced in Unity game development and web technologies that further enhances his ability to create intuitive, user-friendly interfaces for the platform. Lap Pham, the Back-End Lead, has a strong background in full-stack development and data science research. Lap ensures GatorXchange has a robust back-end structure and database, with his expertise in Java, Python, C++, and web development. Lastly, Chan-Chun Yen, the GitHub Master, has strong knowledge in system programming and cloud computing. His experience with C, C++, Java, and AWS cloud services helps maintain the project's codebase and scalability, ensuring smooth transitions within GitHub. This team combines technical expertise and a collaborative spirit to create an online shopping experience that enhances connectivity and convenience for the SFSU community.

2. List of main data items and entities — data glossary/description

User Types

Unregistered Users (UU) - Can browse and search for products.
Registered Users (RU) - Inherits permissions of unregistered users. Users are also allowed to upload products for approval.
Admins (AD) - Has all privileges, able to access and modify permissions for all registered users, can approve products for registered users

Key Data Entries

User Information - Information that the registered user used to create and authenticate themselves. Has sub items of

- ID
- Name
- Email
- Password
- Access (Normal, Admin)

Product - An article or item meant for selling. Has sub items of

- Seller
- Title
- Category
- Date
- Price
- Class (Optional)
- Description (Optional)

Category - A group of products with common characteristics (Book, Furniture, Tools, Room…) Has sub items of

- Category
- Type

Transaction - A listing that has been completed. Has sub items of

- Transaction
- Buyer
- Product
- Date

Message - A message the buyer sends to a seller to indicate interest. Has sub items of

- Message
- Buyer
- Product
- Content
- Date

<u>Other Elements</u>

Register/Registration - The creation of a registered user by inputting user information

Recommendation List - A list of products that were filtered based on class number

Listings - Approved products up for sale.

Dashboard - Section where registered users can check their products and messages they may have received

Search - Finding a product through keywords

Filters - Removes unwanted products with a category while searching

Class Number - the ID for a specific class at SFSU

Seller - Registered user who uploaded the product

Buyer - Registered user who sent a message to a seller

3. List high-level functional requirements

Priority 1:

- Unregistered Users
  1. UU shall be able to browse all available products
  2. UU shall be able to search the available products
  3. UU shall be able to register an account by SFSU email to buy or sell products
- Registered Users
  5. RU shall be able to sell/buy products
  7. RU shall be able to send a message to other RU
  8. RU shall be able to sort results based on relevance, date, or price
  9. RU shall be able to input a class number and receive a recommendation list based on the input
- Admin
  11. Admins shall be able to approve or remove RU's products
- System
  12. System shall support user authentication and authorization
  13. System should provide filters and search functionality
  14. System shall be able to store products

Priority 2:

- Registered Users
  4. RU shall be able to access and update their user information (Name, email, password)
  6. RU shall be able to remove their products.
- Admin
  10. Admins shall be able to manage or remove RU.

Priority 3:

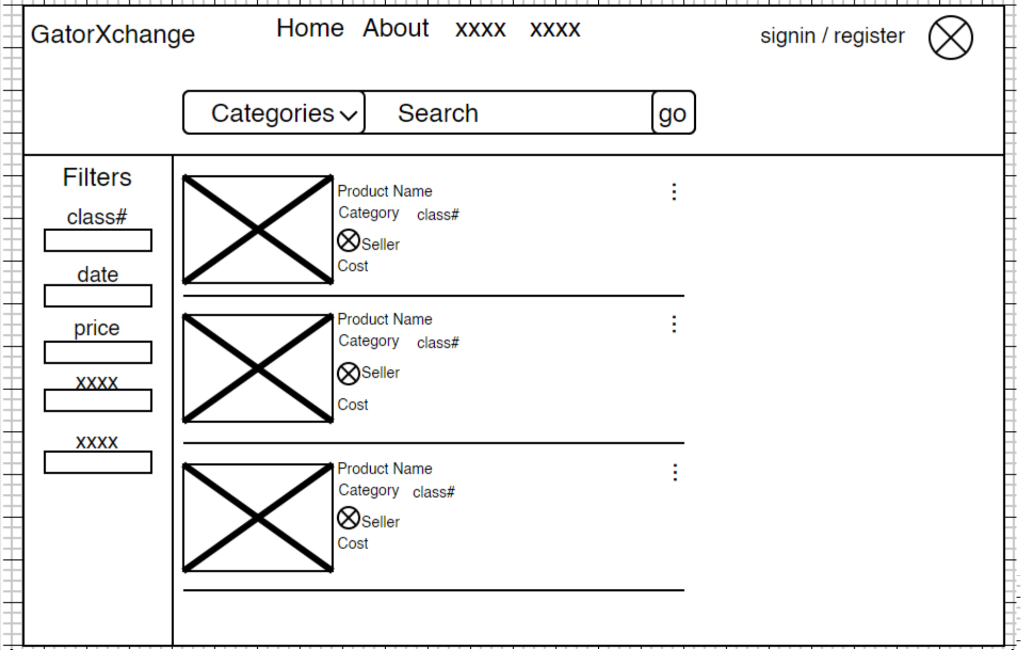
- System
  15. System shall alert Admin when new products are added.

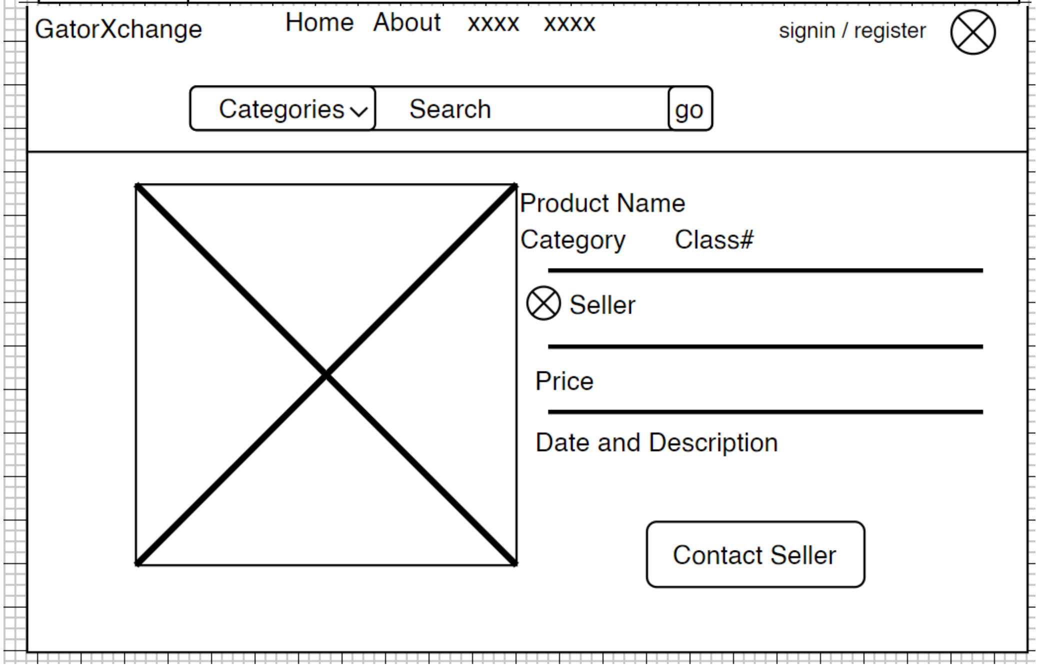
4. UI Storyboards for each main use case

1. Buying an item

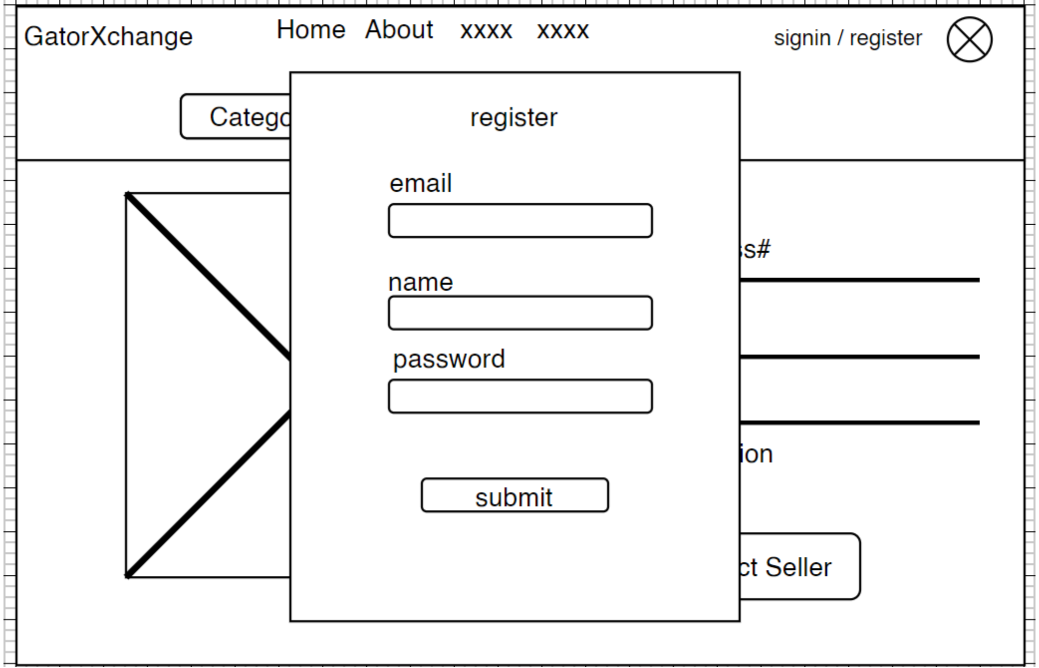
Buying an item

User looks through listings

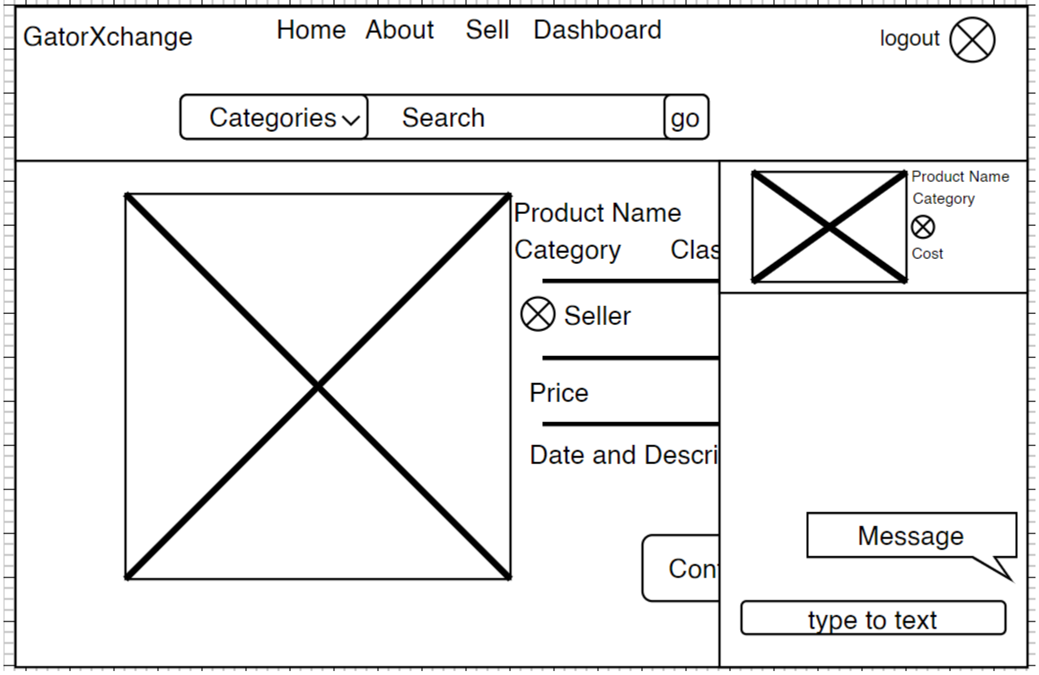




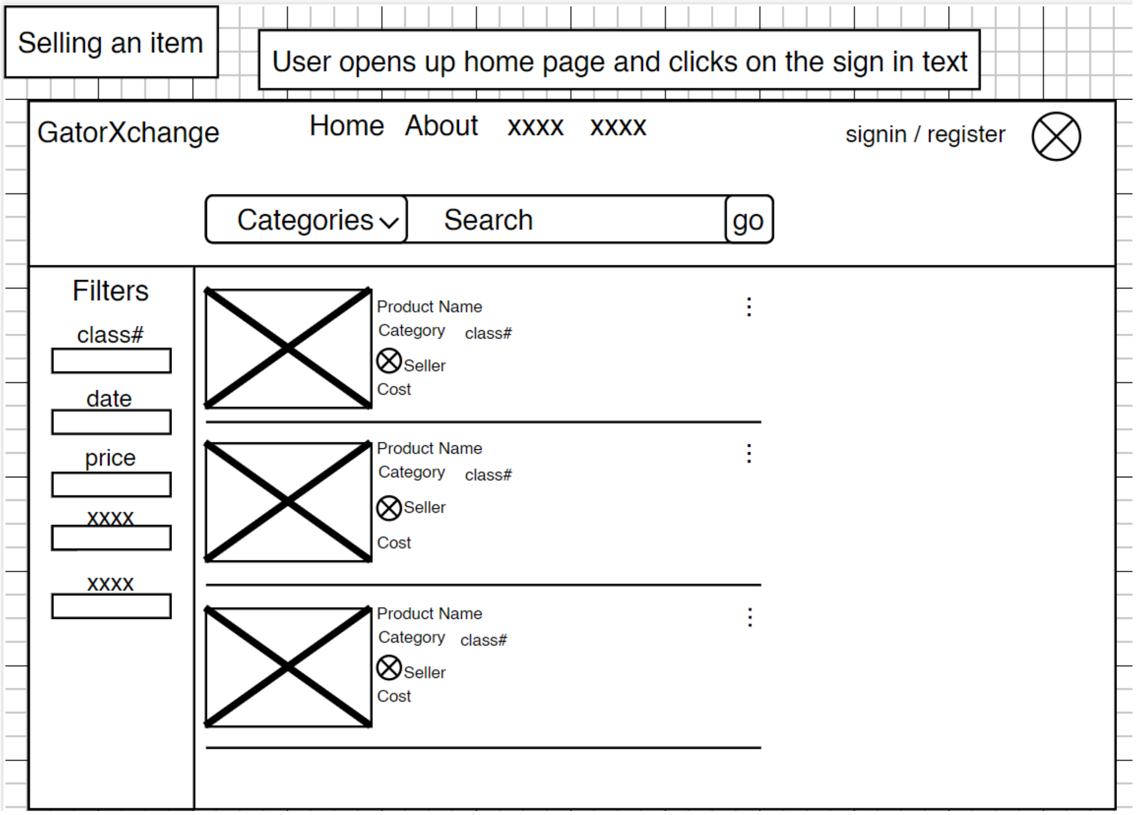
User has to register before contacting seller



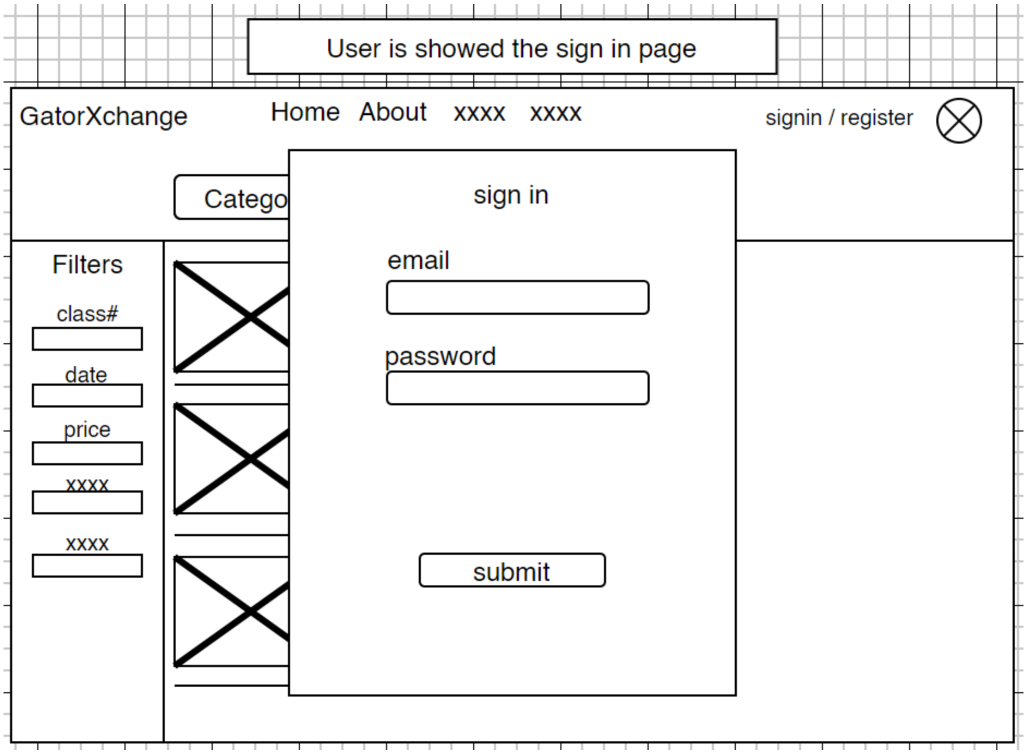
User messages seller



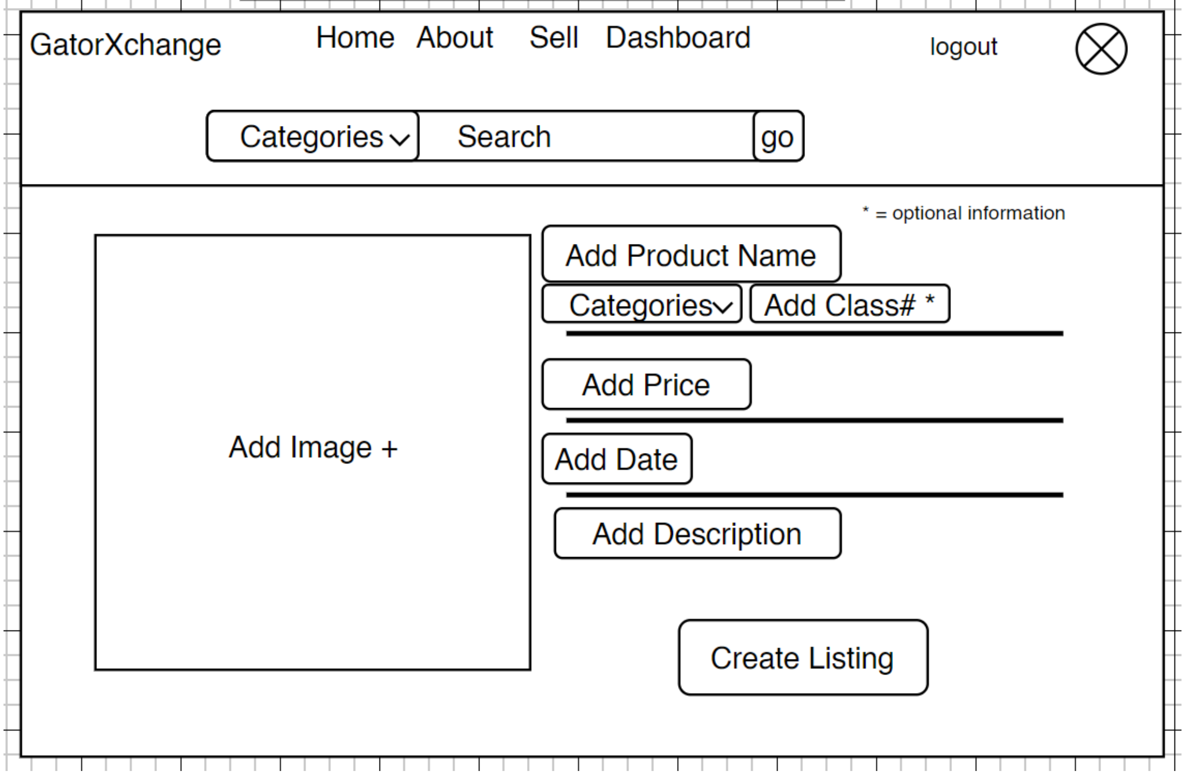
2. Selling an item


Selling an item

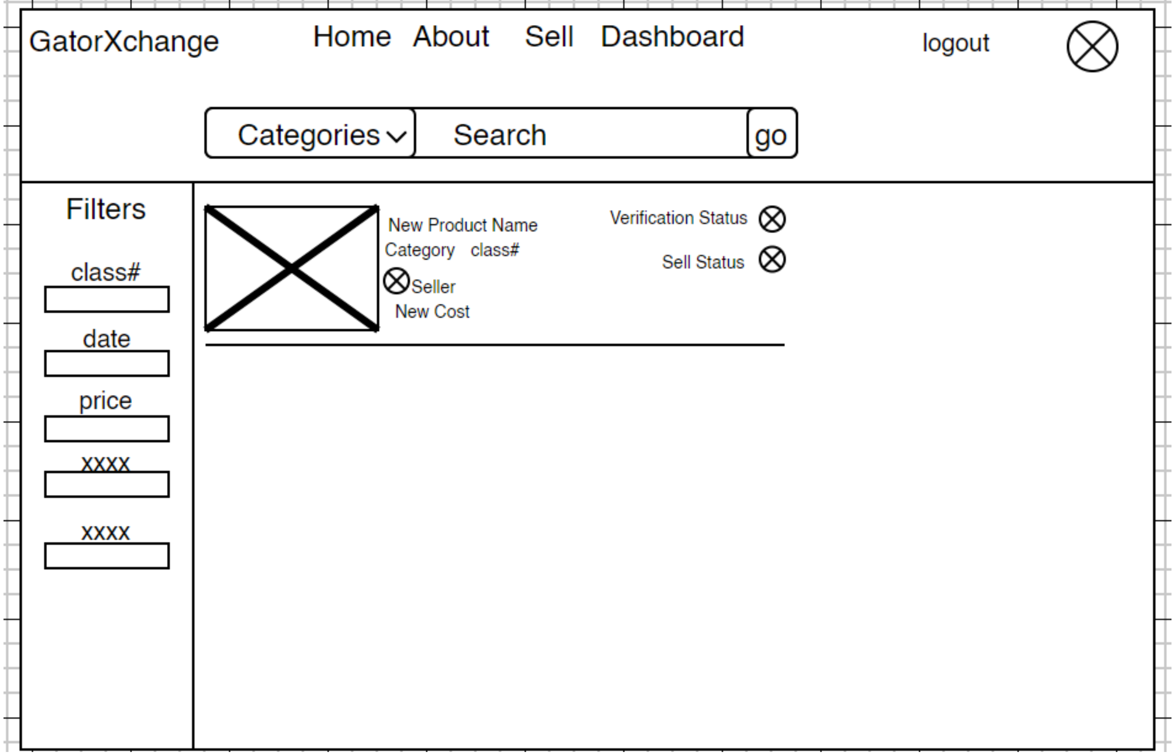
User opens up home page and clicks on the sign in text


User is showed the sign in page

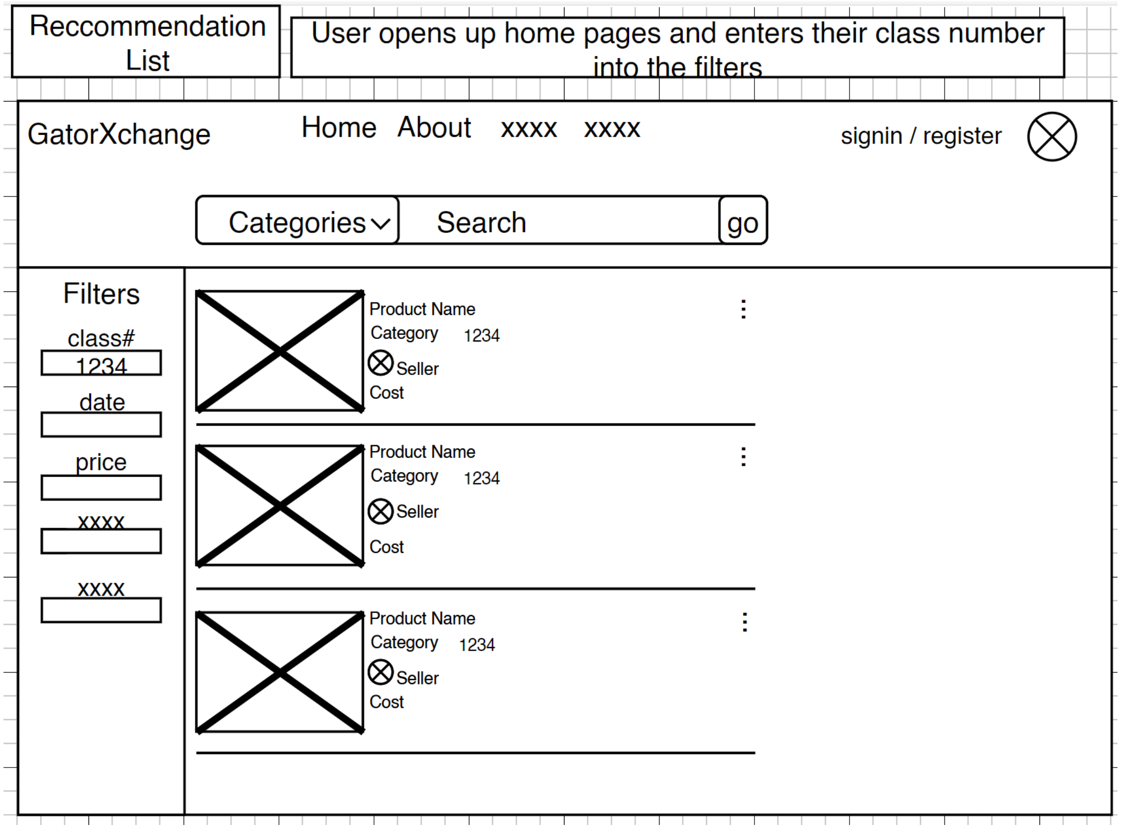
User clicks on Sell Tab



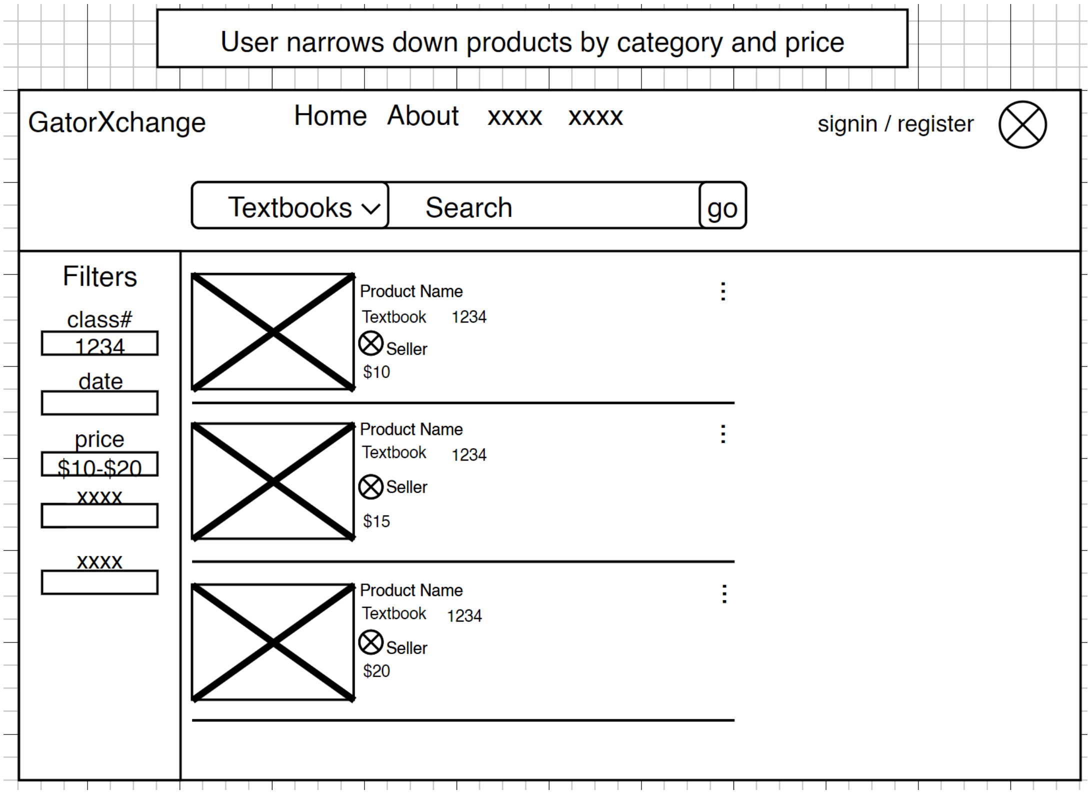
User clicks on Dashboard Tab



3. Recommendation List


Reccommendation List

User opens up home pages and enters their class number into the filters


User narrows down products by category and price

4. Admin Access

The Admin UI will be done through Workbench.

5. High-level Architecture, Database Organization summary only

Database Organization:

User: - can have 2 types
- userId (PK)
- Email address
- Password
- UserRole (USER and ADMIN)

Product:
- productId (PK)
- **Title**
- **Description**
- userId - (FK)
- CategoryId - (FK)
- **Price**
- Class
- Quantity

Image_Product:
- image_product_ID (PK)
- productId (FK)
- Name

Image_User:
- image_user_ID
- user_ID
- name

Category: (Strong)
- categoryId (PK)
- Type

Conversation:
- conversation_ID (PK)
- productId (FK)
- buyerId (userID - FK)
- sellerId (userID - FK

Message:

- messageId (PK)
- conversation_Id (FK)
- Content
- create_At

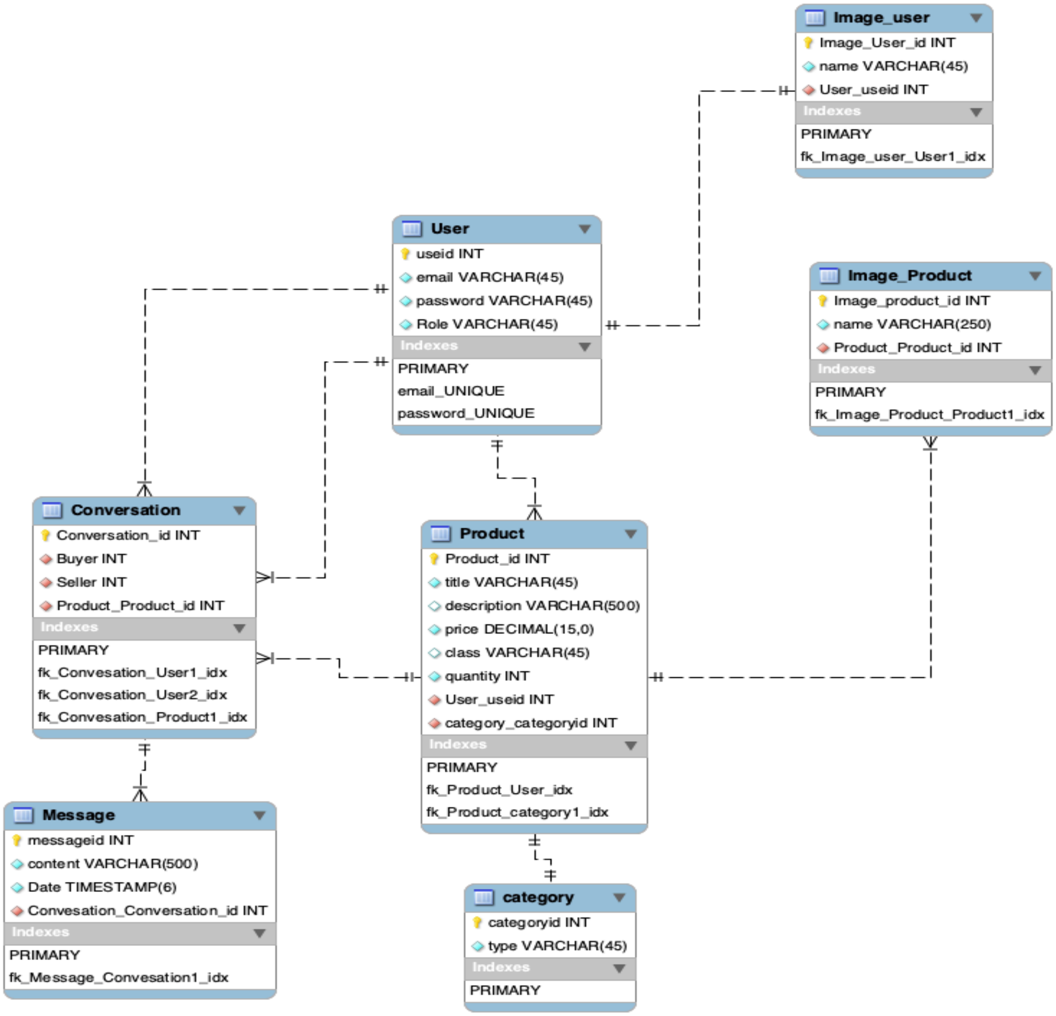
Media storage

We will store images in a file system with JPG and PNG format. Each image's filename will be stored in the image_Product and Image_User tables, allowing retrieval by referencing the stored name. These media files will be stored in a static directory and accessed via predefined paths in the application.

Search/filter architecture and implementation:

The system will display all products sorted in alphabetical order.
Users can search for a product based on category, keyword, or both. The system will retrieve the data based on the user's selection and present it below the search bar.

For the coding:
We will use an SQL query to search for a product based on its ID and utilize `LIKE '%{req}%'` to filter specific items preferred by the user.



6. Key risks

Schedule Risks

Our group has only four team members compared to the five that most groups have and our schedules are incompatible. Thus, meetings are limited and very restrictive, while our general manpower is smaller. To mitigate this problem, the scope of our overall project is small. We will try to have more votes and discussions through Discord, then writing the determined tasks in our Trello.

7. Project management

For Milestone 2, we split our four person team into two pairs: one that focuses on part 1 and one that focuses on part 2. The two focusing on part 1 are front-end adjacent, while the two starting part 2 are back-end adjacent. This allows both tasks to be started as soon as possible and separately. We check in at the midpoint before delivery, as well as the day of delivery.

We use group meetings to bring these two pairs together and make sure we're on the same page. Both groups express information that they need and things they have accomplished. We also discuss the instructions and key pieces of information we need to decide together.

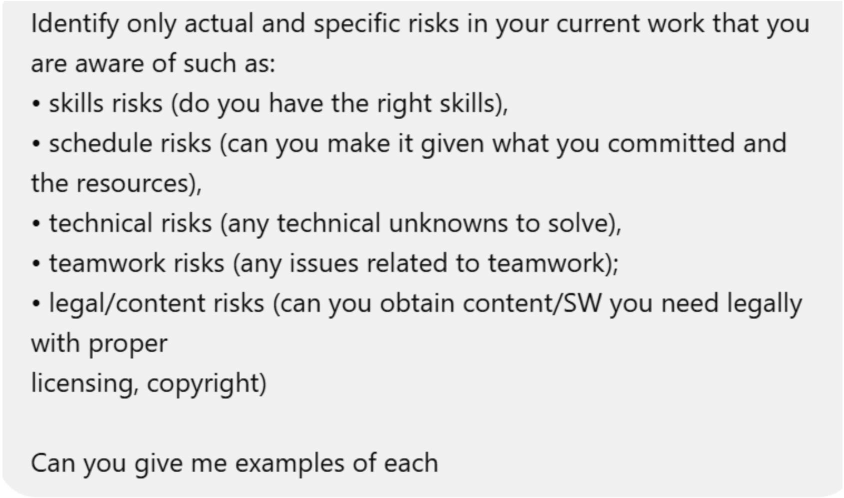
For this task and future tasks, we will continue to use Discord for immediate correspondence. We use Trello to delegate and pinpoint major tasks, and we use Zoom during our online meetings for its whiteboard capabilities.

8. Use of GenAI tools like ChatGPT and copilot

For all tasks, ChatGPT-4 was used.
It was used for:

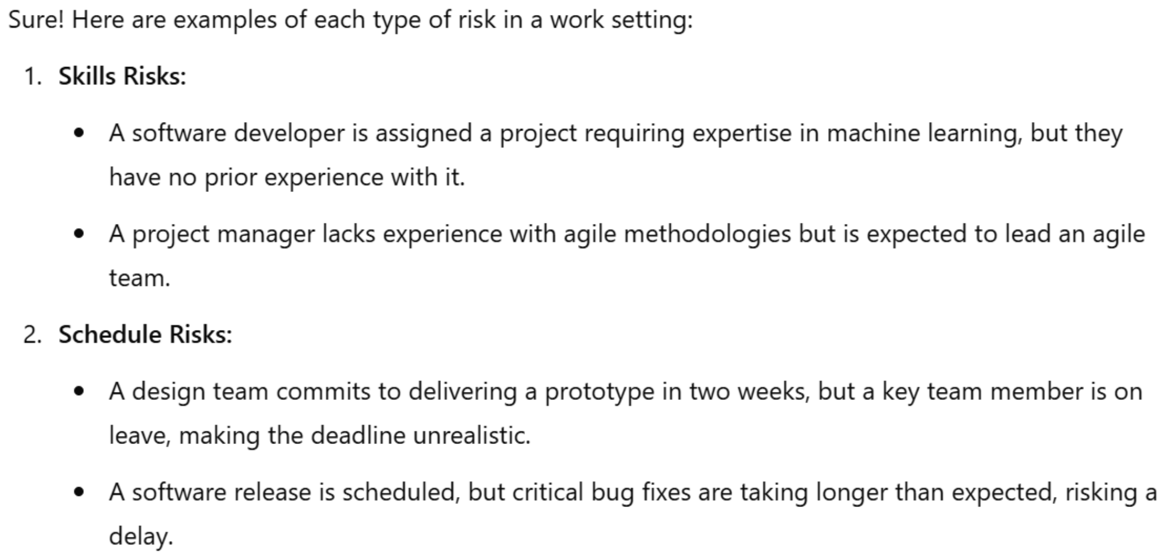
- Finding Ideas - HIGH

Finding Ideas
We used ChatGPT to help name examples of key risks. It helped with setting a tone of what kind of risks we should look out for, as well as gathering ideas of things we might be missing.
Screenshot of the interaction is shown below:



Identify only actual and specific risks in your current work that you are aware of such as:
• skills risks (do you have the right skills),
• schedule risks (can you make it given what you committed and the resources),
• technical risks (any technical unknowns to solve),
• teamwork risks (any issues related to teamwork);
• legal/content risks (can you obtain content/SW you need legally with proper
licensing, copyright)

Can you give me examples of each



Sure! Here are examples of each type of risk in a work setting:

1. **Skills Risks:**
   - A software developer is assigned a project requiring expertise in machine learning, but they have no prior experience with it.
   - A project manager lacks experience with agile methodologies but is expected to lead an agile team.
2. **Schedule Risks:**
   - A design team commits to delivering a prototype in two weeks, but a key team member is on leave, making the deadline unrealistic.
   - A software release is scheduled, but critical bug fixes are taking longer than expected, risking a delay.

9. Team Lead Checklist to be completed by team lead

DONE - So far all team members are fully engaged and attending team sessions when required

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 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