SW Engineering CSC648-848

Spring 2024

Campus Buy/Sell Application: SwiftSell
Team 04

Team Members:
Aymane Arfaoui, Team Lead (aarfaoui@sfsu.edu)
Markus Reyer, Github Master
Amandeep Singh, Front End Lead
Alexis Alvarez, Back End
David Daly, Back End Lead

Milestone 2

Submitted	Revised
March 23rd 2024	April 2nd 2024

1. Executive Summary

There are many challenges when it comes to buying, selling and trading items related to university functions. Our team has realized that it is very difficult for students, faculty and staff at San Francisco State University to have access to a safe online marketplace that caters to their needs as members of the university. Other online marketplaces often do not offer products such as used textbooks that are usable for SFSU classes and users are forced to interact with strangers with no ties to their school's community which increases their risk to their safety. This is where our team has come up with the online marketplace SwiftSell. Our project is motivated by the need to create a safe, convenient, and user-friendly marketplace that provides unique functions specially designed for the SFSU community. SwiftSell will differ from other online retailers by providing services specifically tailored to the needs of college students, faculty and staff members. Doing so, SwiftSell seeks to enhance the overall experience at SFSU.

SwiftSell will offer a range of functions and services tailored to the needs of the members of SFSU. These include verified user profiles with the use of SFSU emails, local listings convenient for users to complete their transactions, as well as a dashboard for sellers to be able to view messages from prospective buyers. The platform will also feature search categories, including textbooks searchable by class, furniture, and electronics. The platform will also provide the user with the ability to list and accept job listings such as assisting in moving furniture sold on our marketplace and listing tutoring services for university classes. By providing a market focused on student, staff, and faculty needs, SwiftSell aims to be the all in one platform for local school transactions and provide additional services related to the campus.

Our team is composed of driven and innovative students from diverse backgrounds, all united by a shared vision of enhancing the student experience at SFSU. With an emphasis on the user experience, we aim to create an easy to use and attractive marketplace for users to interact with. With expertise in Computer Science and the guidance of our CEO Dragutin Petkovic and CTO Anthony Souza, we are equipped to develop and launch a successful marketplace platform. Together, we are committed to making SwiftSell a valuable resource for members of the SFSU community.

2. List of main data items and entities

Items for Sale: Objects available for sale by registered users.
Associated data fields include:

- Owner (registered user who created the post)
- Category (e.g., Furniture, Electronics, Textbooks, Services)
- Description (e.g., description of item, tutoring subject and hours)
- Price (price of item or hourly rate for services)
- Picture (photo of item or image of flyer for services)
- Thumbnail (generated from the user-supplied image)
- Live (is the item visible on the site? - admin function)
- Date posted
- Item id

Dashboard: A screen view within the website, available to registered users, which displays messages sent to the user and items posted by the user
.

Messages: Communication between two registered users on the website.
Associated data fields include:

- Sender
- Recipient
- Associated item
- Content of message
- Timestamp of when it was sent
- Unread/read by the recipient
- Message id

Register/Registration: The process whereby an unregistered user becomes a registered user by creating a username and password.

User: Visitors to the web application are part of one of three groups: unregistered users, registered users, and admin.
Associated data fields include:

- Email address (sfsu.edu)
- Username
- First name
- Last name
- Password
- Date joined
- User id

Unregistered User: An unregistered user is not allowed to sign into the website. The unregistered user is not allowed to login, post, or message, and cannot view the dashboard; the unregistered user is allowed to view items for sale, to fill out the form to post an item or send an item (but not submit the form), and to register.

Registered User: In addition to the actions allowed to an unregistered user, a registered user is allowed to sign into the website. Once signed in, a registered user is allowed to post items for sale (with admin moderation), message other registered users, and view the dashboard.

Admin: In addition to the actions allowed to a registered user, the admin shall have the responsibility to approve posts for sale that conform to terms of service.

3. Functional Requirements - prioritized

Priority 1

- Unreg user
  - An unregistered User shall be able to search for specific items.
  - An unregistered User shall be able to browse the website's items
  - An unregistered User shall be able to see images for items/services with accurate description
  - An unregistered User shall be able to categorize items for sale by area of use (electronics, textbooks, furniture, etc)
  - An unregistered User shall be able to sort items for sale by majors.
  - An unregistered User shall be able to filter through items based on price, date posted.
  - An unregistered user shall be able to register
- Reg user inherits all functions of unregistered user plus shall have these below.
  - A Registered User shall have a dashboard with information containing order history, messages, notifications, and the items they posted for sale.
  - A Registered User shall be able to post items and/or services(tutoring,delivery, etc) for sale with approval of admin
  - A Registered User who posts items for sale shall upload at least one image, a brief description of said item, and a reasonable price for the item.
  - A Registered User shall be able to message to sellers about items for sale.
  - A Registered User shall be able to post small jobs such as tutoring, moving furniture, etc.
  - A Registered User shall be informed if another user wants one of their items via in-site messaging.
  - A Registered User's post shall wait to be approved by the admin.

- A Registered User shall be able to delete a post without the approval of an admin.
- A registered User shall be able to login.

o Admin inherits all functions of registered user plus shall have these below

- An admin shall be required to approve or deny User posts to go online.
- An admin shall be able to delete users.

Priority 2

o Unregistered user

o Reg user

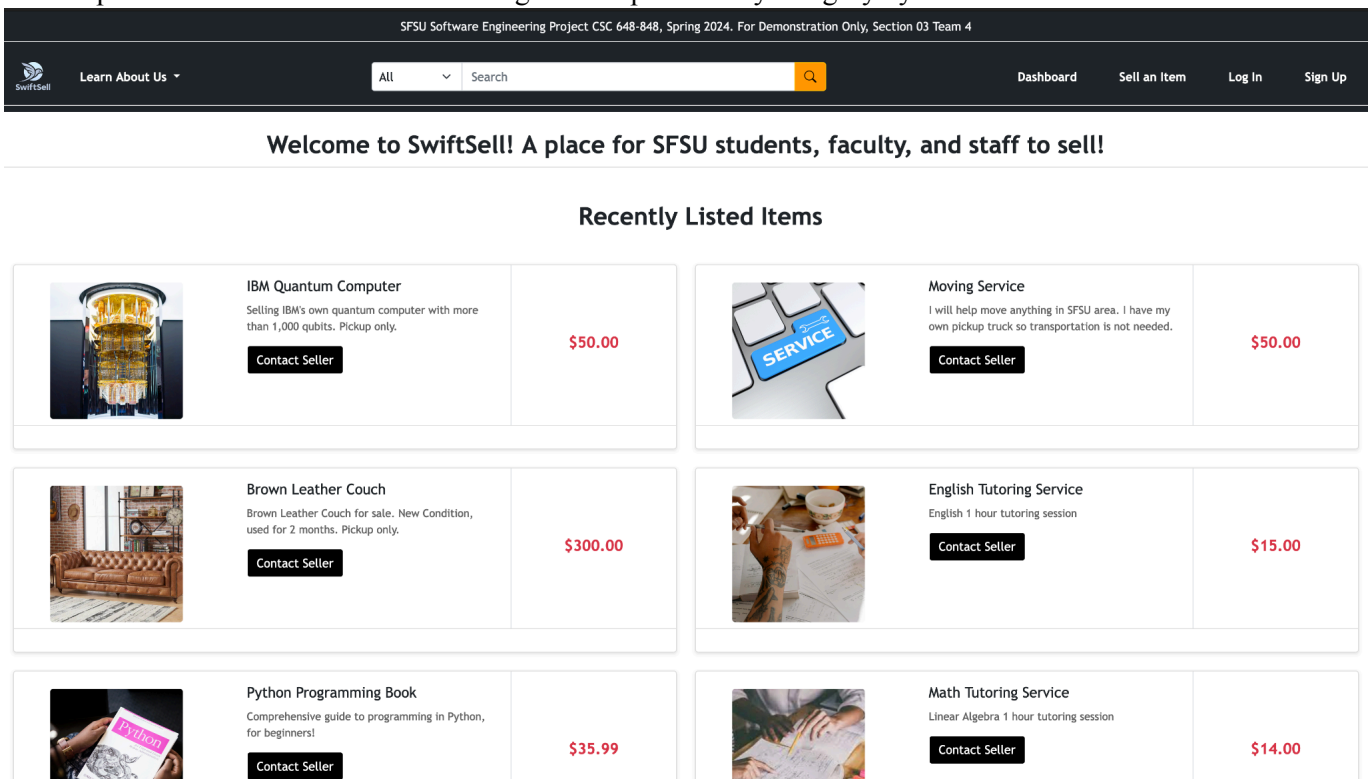
Priority 3

o Reg user

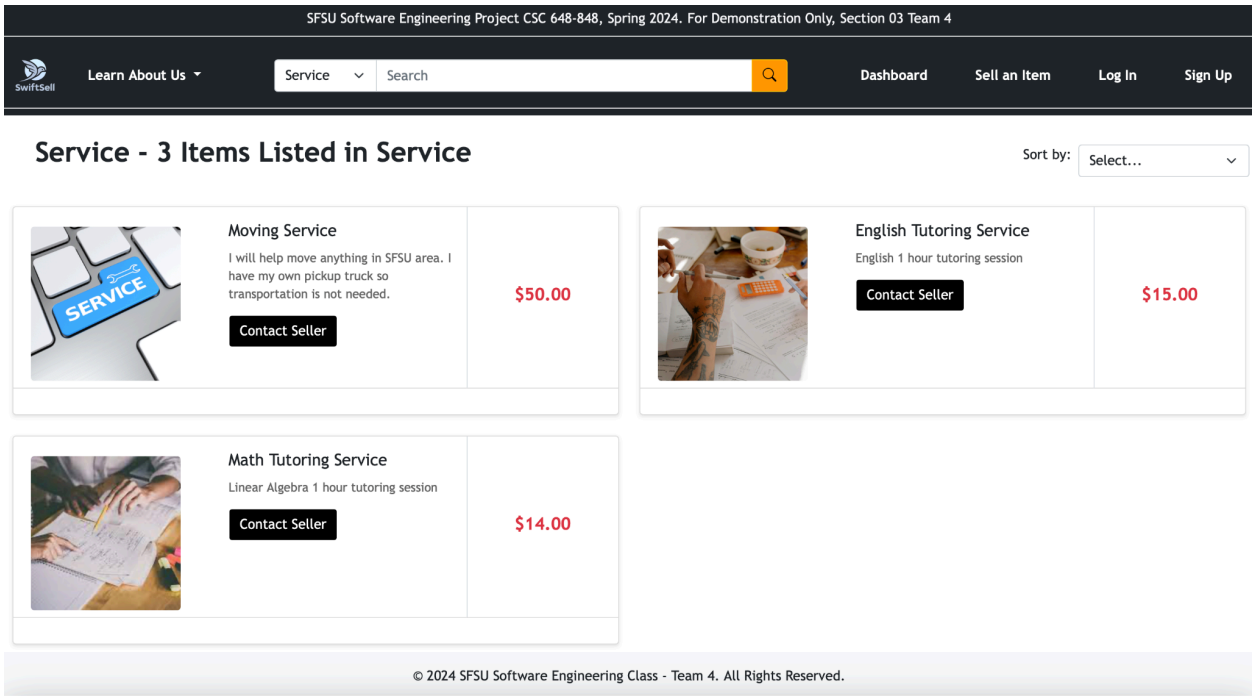
o Admin

4. UI Storyboards

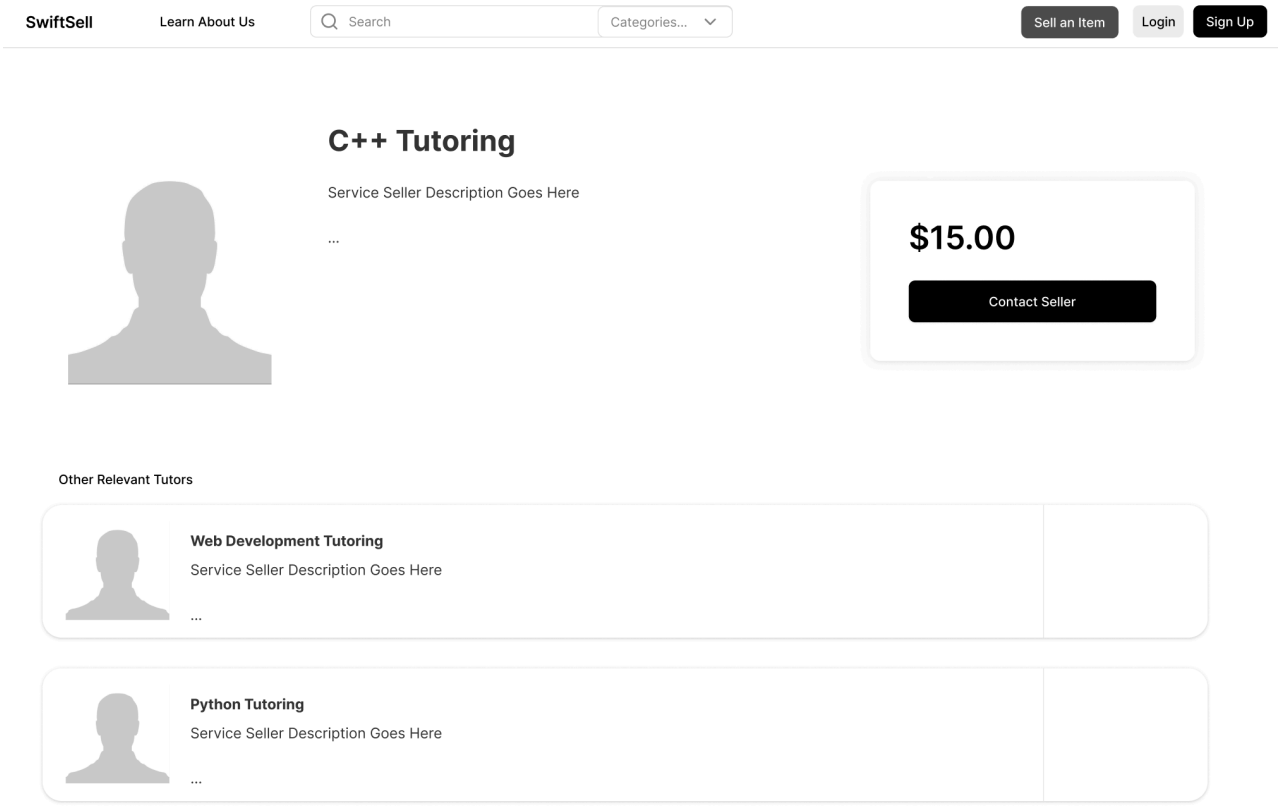
User is presented with home screen showing various products by category by default



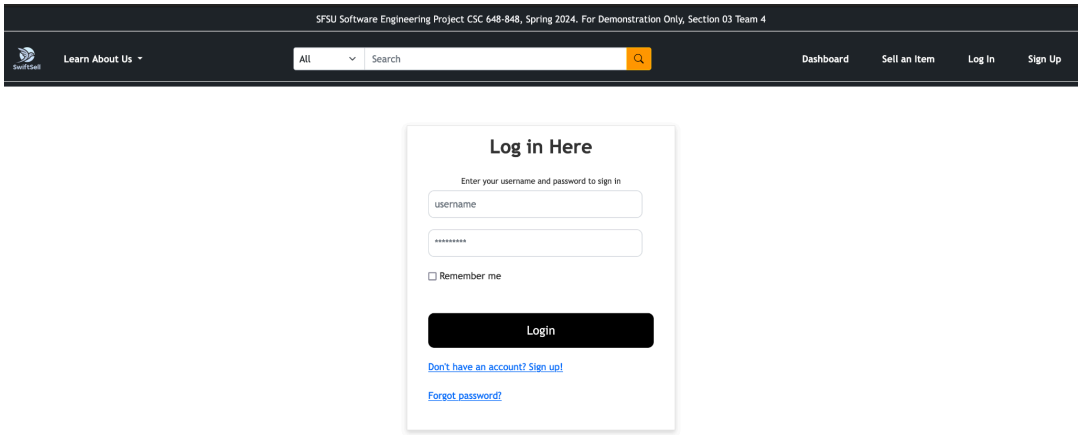
- User will be able to search for products/services showing SFSU specific function

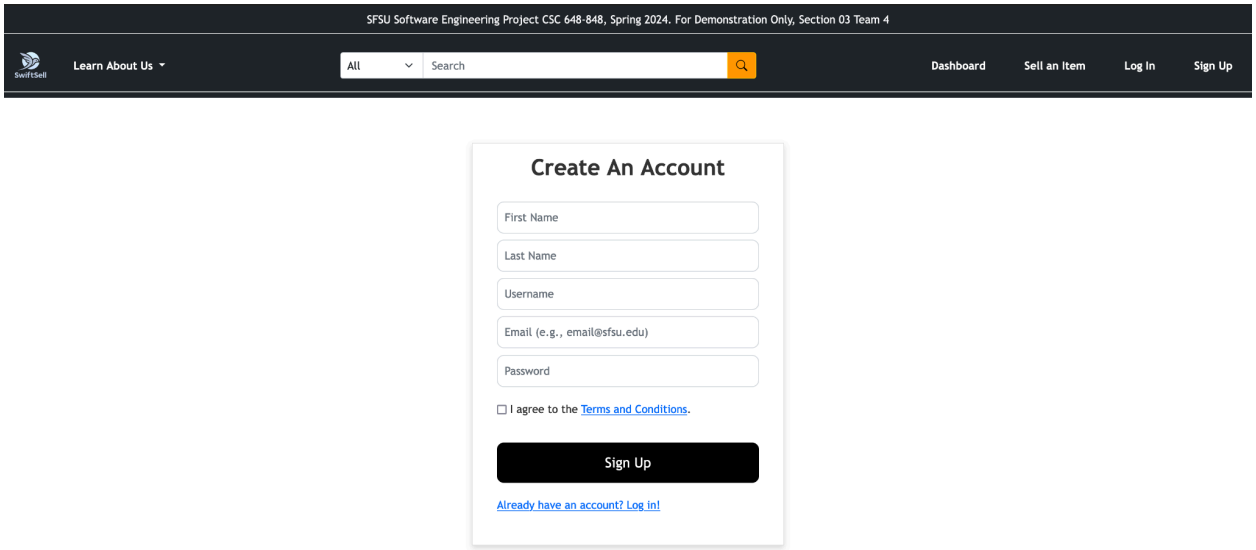


- Users shall be able to click on the offer to see the expanded view allowing the ability to contact the seller directly, other relevant listings will also be shown.

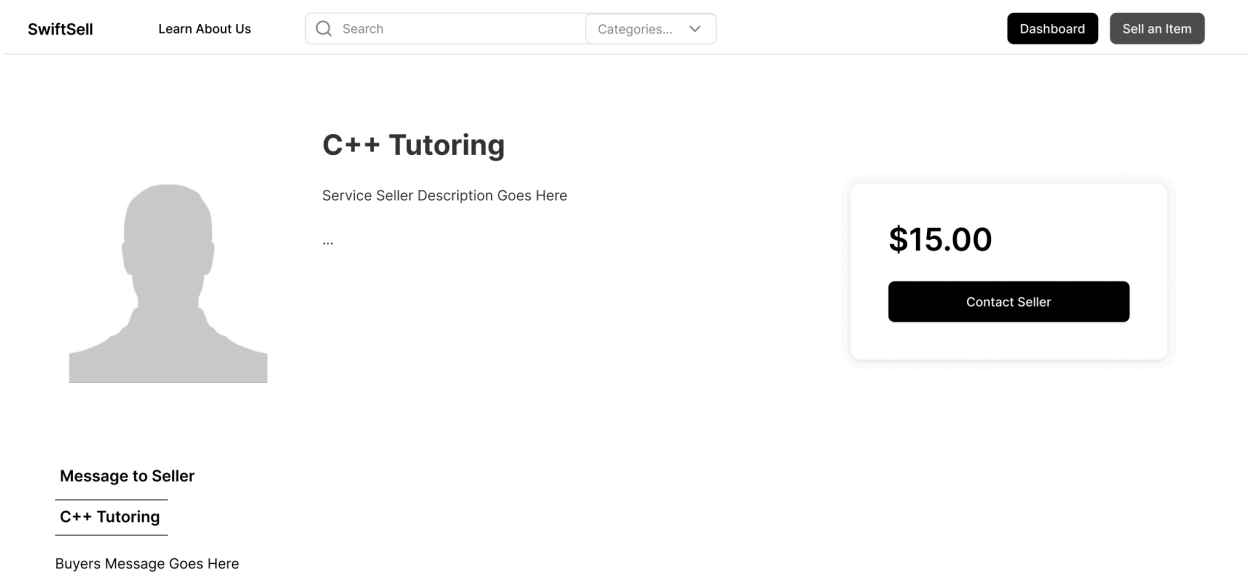


- User shall be required to log in or create an account when trying to contact the seller as they must be a part of SFSU.

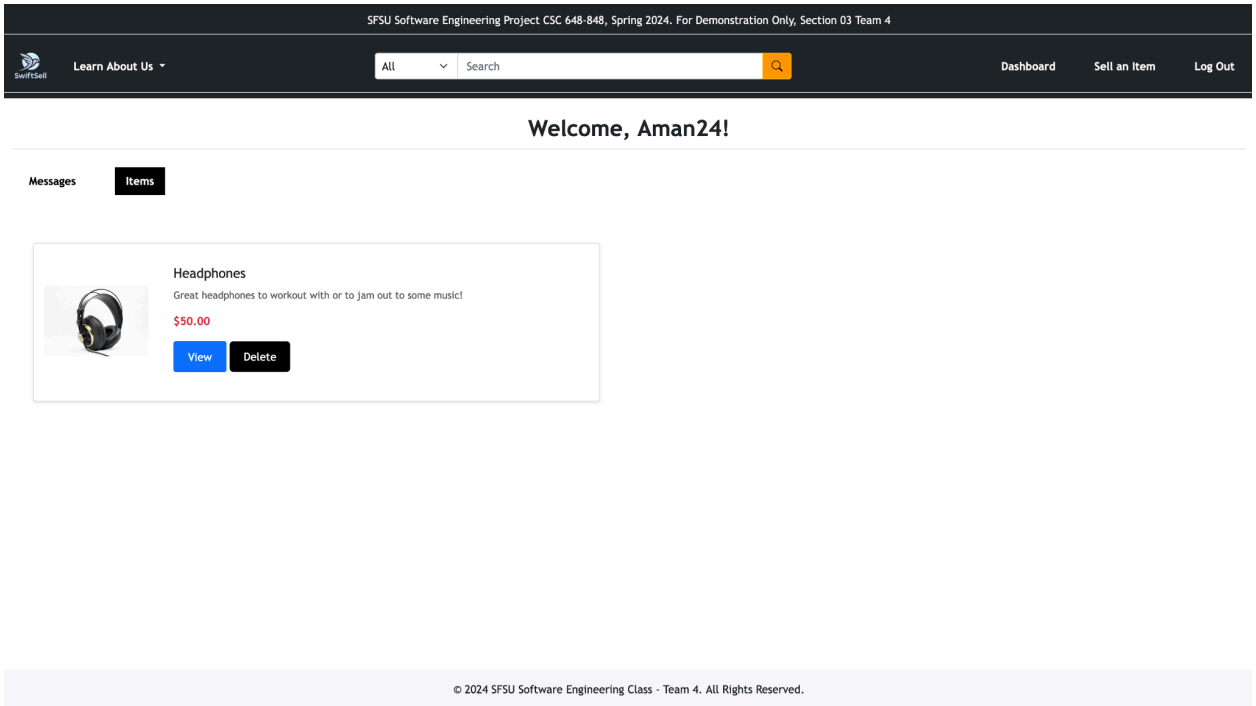




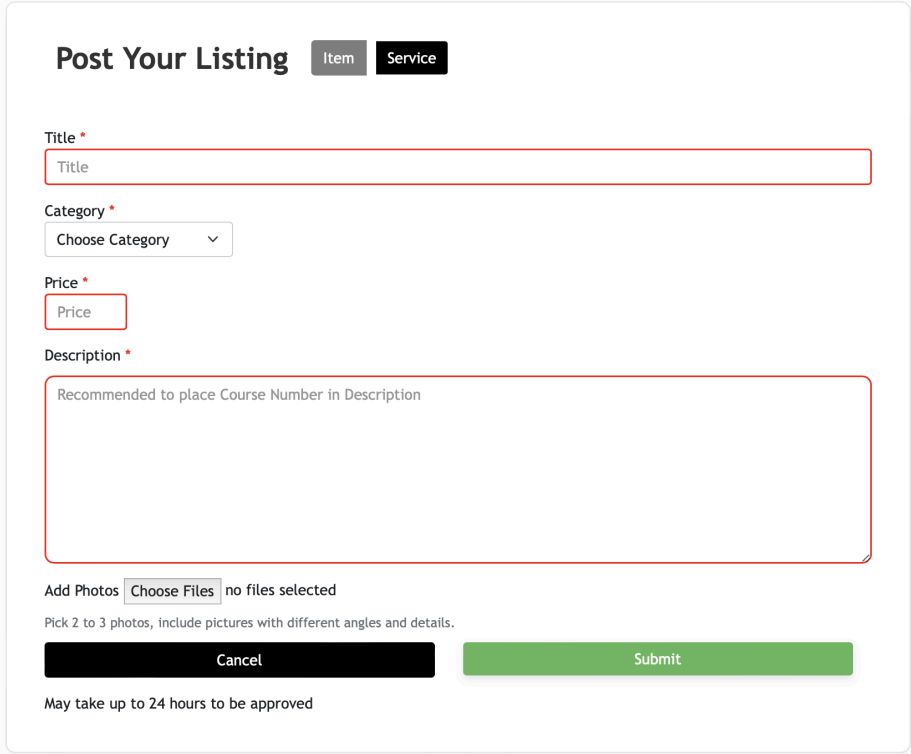
- Registered user shall be allowed to send a message directly to the seller, where they must input a way of communication.

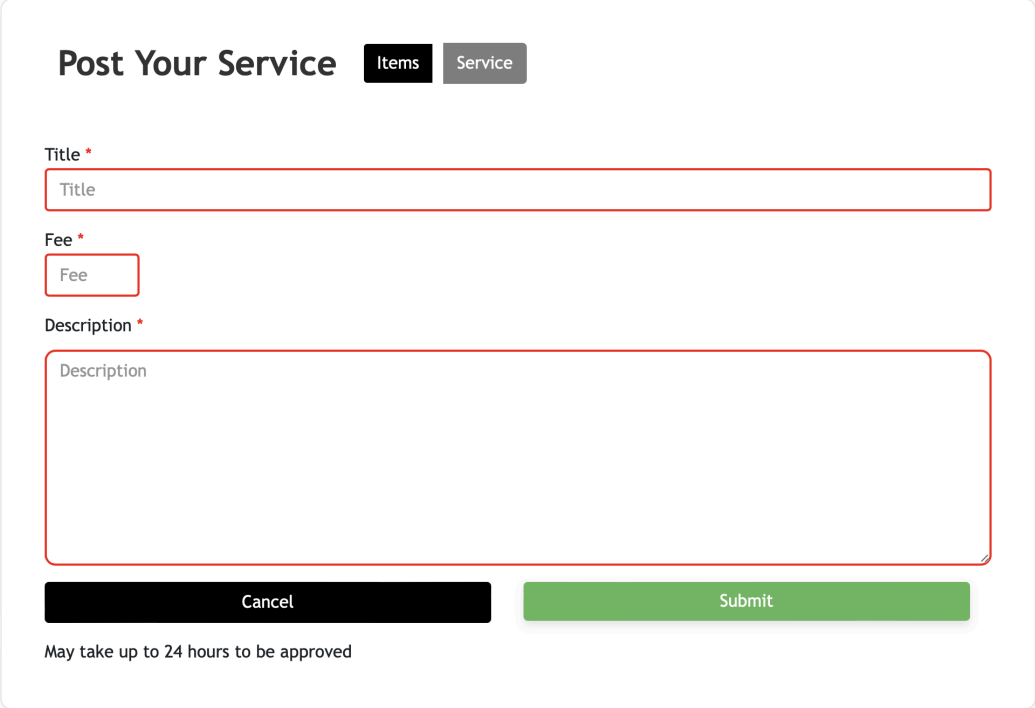


- A user’s dashboard shall contain a tab for messages and items showing the most recent inquiries on their items.



- Registered users will have the option to either post an item or service for sale where a tab at the top will indicate the type of entity they are selling





5. High Level Architecture

registered_user:

- user_id : INT AUTO_INCREMENT PRIMARY KEY, Unique identifier for each user.
- Username: VARCHAR(255), Username for other users to see
- first_name : VARCHAR(255), First name of the user
- last_name: VARCHAR(255), Last name of the user
- password: Hashed password for user authentication.
- email: VARCHAR(255), Email address of the user.
- registration_date: DATETIME(), Date and time when the user registered.
- last_login: Date and time of the user's last login.
- account_status: Status of the user account (active, suspended, etc.).

message:

- Message_id: INT PRIMARY KEY Unique identifier for each message.
- sender_id : INT FOREIGN KEY, Links to the UserID in the Registered Users table.
- recipient_id: INT FOREIGN KEY, Links to the UserID in the Registered Users table
- item_id: INT FOREIGN KEY, Links to the ItemID in the Items For Sale table
- content: VARCHAR(255), Text content of the message.
- message_date: DATETIME(), Date and time when the message was sent.

items_for_sale:

- item_id : Unique identifier for each item.
- seller_id : Links to the UserID in the Registered Users table.
- title: Title of the item.
- description: Description of the item.
- price: Price of the item.
- category_id: Category of the item
- availability: Availability status of the item (available, sold, pending, etc.).

- listed_date: Date and time when the item was listed.
- live: Is the posting visible on the website (admin function)
- high_res_image_url: URL of the high-resolution image of the item.
- thumbnail_image_url: URL of the thumbnail image of the item.

categories :

- Books
- Electronics
- Furniture
- Moving
- Tutoring

1. **Media Storage**

   We will be using file system approach to store our images.

   Storage Location: images will be stored directly on the server's file system. This entails creating a designated folder or directory within the server's file structure where these images will be saved.

   Accessing Images: To facilitate access to these stored images on the frontend of the application, Flask provides a mechanism to serve static files. Flask's send from directory function serves static files, including images.

   We will use relative pointers to root, and ensure file system protection.

2. **Search architecture and Implementation**

We will be using the preferred option of SQL %like method as described in architecture class slides.

In our search implementation, we will utilize the SQL function Like to enable users to search for terms and find results that are similar to their query.

When a user performs a search, we will use SQL queries with the LIKE operator to match search terms against the attributes of goods.

Implementation:

The backend will receive user queries, construct SQL queries with the LIKE operator, execute them against the database, and retrieve matching good entries.

6. Key Risks

Schedule Risk:

One of the potential issues is related to the schedule risk. Considering the fact that we would have to meet multiple times, especially regarding the coding, it is not always easy to find time considering how we all have different schedules. This is why we decided to separate tasks and reduce our meeting time where we would tackle problems instantly and quickly. If more time is needed, only the people who are assigned tasks can talk for longer periods of time. We constantly use discord to communicate (through chat). We also are experimenting using Trello to manage our tasks and see if it is a good fit for the needs of our team.

Skill Risks:

One of the potential risks we will probably face is related to skills. Considering the fact that it is the first time we work in a team project of this size. We have encountered some technical issues previously regarding the connection of our website to the amazon server. It is also the first time we work with a python framework flask. To address this issue, we will dedicate a discord channel specifically to issues our team might encounter in order to use our combined knowledge to try to fix it. Strong communication is key in solving such issues. We also plan on leveraging AI tools to guide us in solving our issues. Finally, if a certain problem persists, we will seek guidance from an experienced developer such as CTO (professor Souza).

7. Project Management

Our team has primarily made use of Discord to post updates on our project's progress. Our team lead has consistently set up meetings through Discord 1-2 times per week to allow the team to check in and ensure each team member is aware of the requirements that need to be completed for our project milestone. The meetings also serve as a place to assign tasks to team members so that each part of the project is being worked on. Discord is also used to send quick messages to the team of project updates, such as if a new branch has been created in the github, or if a problem has arisen. Using Discord has proven effective to keep the team updated through its direct messaging. Our team has also set up accounts with Trello and have begun posting checklist objectives and updating our "Doing" card to keep the team up to date on if a problem is being worked on. Our team plans to continue to make use of Trello, as it provides a team member the ability to check if a certain part of the project is being worked on and whether or not tasks have been completed. Our team is also able to add tasks to the Trello Dashboard which proves helpful for when new problems arise as updates can be lost in the Discord channel.

8. Use of genAI tools like ChgatGPT and copilot

- GenAI was used as a guide for the Executive Summary. Once feedback was given for the executive summary, the team modified the summary to fit the requirements for the milestone. Usefulness for this part was Medium as GenAI did not fully understand the the task at hand

- GenAI was used for suggestions on the story board in terms of what user shall see when logging into the page. Along with how to format the products, giving ideas on the design of our item “cards”. Usefulness in this scenario as well was low, as the real design was fully constructed by us and many changes were brought while making the mock-up

- GenAI was used to establish some concrete solutions in terms of the schedule risk and skill risks. GenAI was also used to provide examples of what the database tables should look like for a registered user, a sales_item, messages and categories. In this scenario, the usefulness of GenAI was High as it provided clear options to choose from to make our database.

- GenAI was used to provide critical and in-depth information as to how images should be stored as well as best practices for our search algorithm. GenAI also provided visual information as to how images could best be stored and managed in our project. GenAI also provided examples for implementing our search algorithm. Usefulness here was Medium as it guided the search on how to incorporate these implementations.

9.Team Lead Checklist:

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 |**DONE**