GATOR JOE'S

Team 13

Milestone 2

Software Engineering CSC 648-848
Fall 2019 - Section 02

Piero Calenzani (Team Lead): pcalenzani@gmail.com

Grant Kennedy (Front-end Lead)

Mark Hammond (Back-end Lead)

Chuhao 'William' Yang (Front-end, Github Master)

Varun Sura (Front-end)

Zhuoxin Tan (Back-end)

Date Submitted	Date Revised	Description
10/18/2019	10/24/2019	First draft

1. Functional Requirements

Priority 1 – Must Have

Unregistered Users:

4. Users shall be able to view listings.
    - 4.1 Users shall be able to browse all active listings.
    - 4.2 Users shall be able to sort by price, category, class, etc.

8. Users shall be able to register.
    - 8.1 Users will only be prompted to registered when attempting to buy and sell.

Registered Users:

11. Registered users shall be able to post listings.
    - 11.1 Registered users shall be able to buy listings.

12. Registered users shall be able to log in and log out.

22. Registered users shall be able to send and receive messages.

Admin Users:

19. Administrators shall be required to approve listings.
    - 19.1 Administrators shall be able to deny listings.

21. Administrators shall be able to delete listings.
    - 21.1 Administrators shall be able to delete users.

Priority 2 – Desired

Unregistered Users:

1. Website shall display recent listings on home page.

2. Website shall show the images of listings.
    - 2.1 Website shall show a preview image of listings when browsing.

5. Website shall display any description/information for listings.

Registered Users:

6. Users shall be able to search.
    - 6.1 Users shall be able to search within a category.

16. Registered users shall be able to enter the information of their listings.
    - 16.1 Registered users shall be prompted for the proper information of items.

21. Registered users shall be able to view their own listings on home page.
    - 21.1 Registered users shall be able to cancel their listings.

Priority 3 – Opportunistic

Registered Users:

13. Registered users shall be able to view order status.

14. Registered users shall be able to view order history.

15. Registered users shall be able to open a shopping cart.
    - 15.1 Registered users shall be able to add or remove items from their cart.

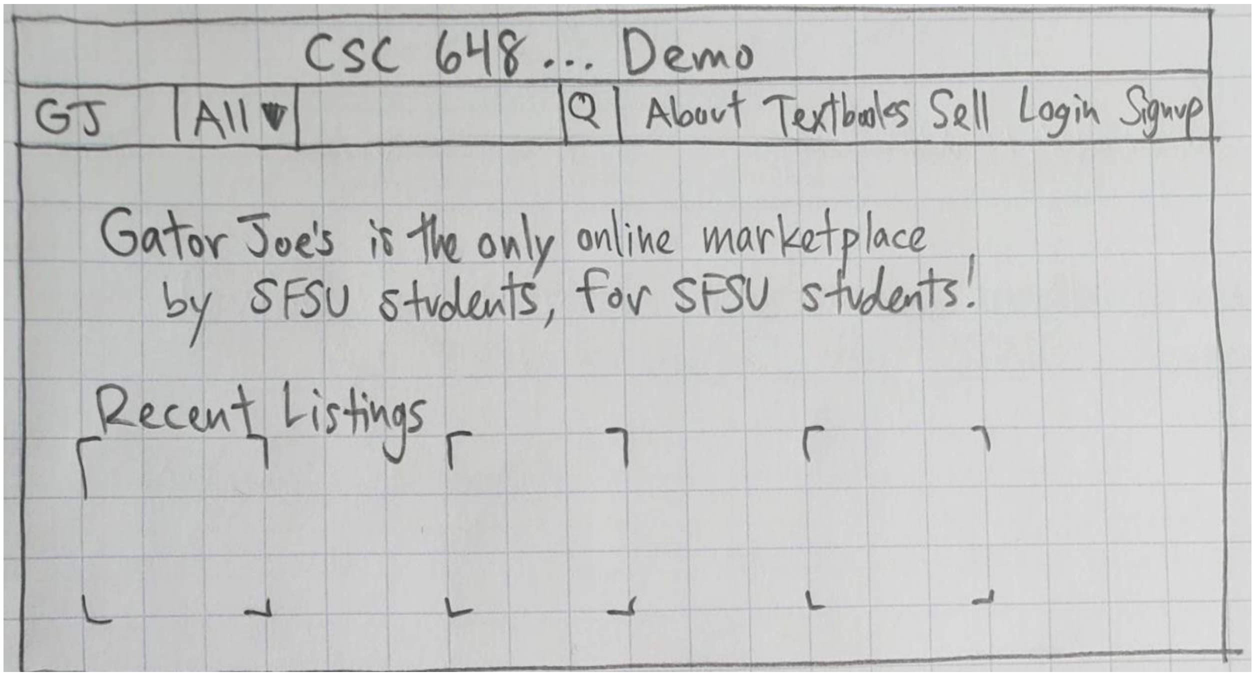
Admin Users:

18. Administrators shall be able to monitor all user listings.

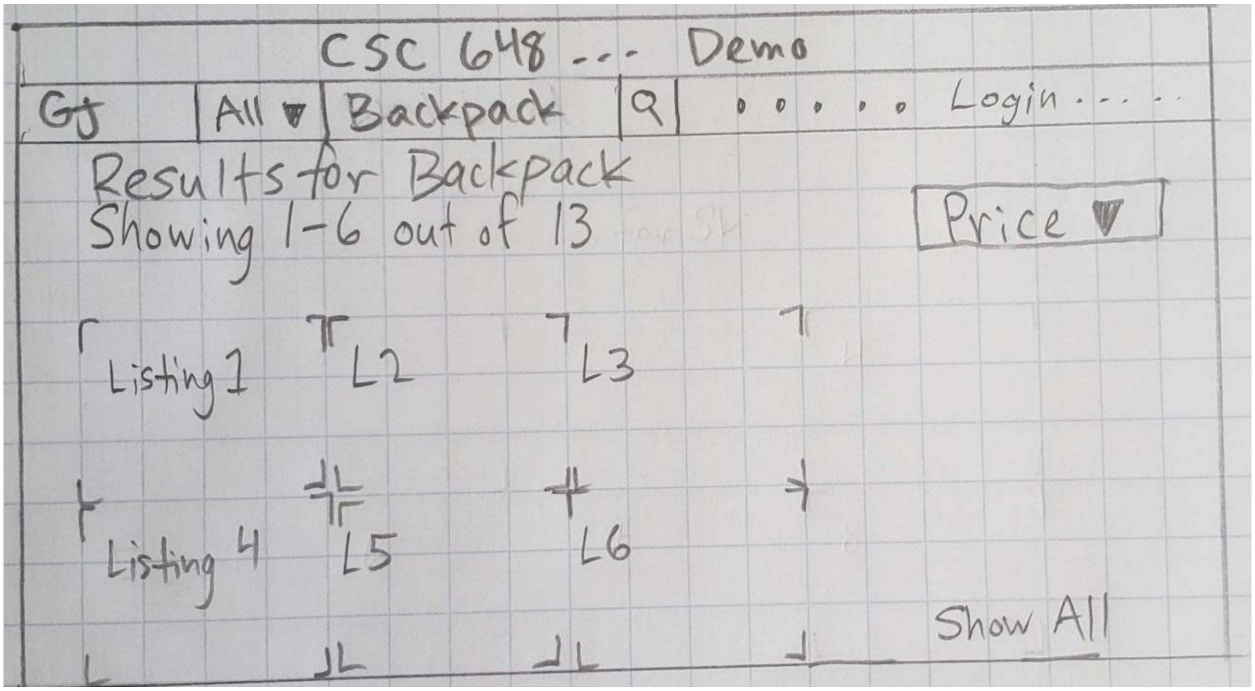
20. Administrators shall be able to cancel orders and transactions.

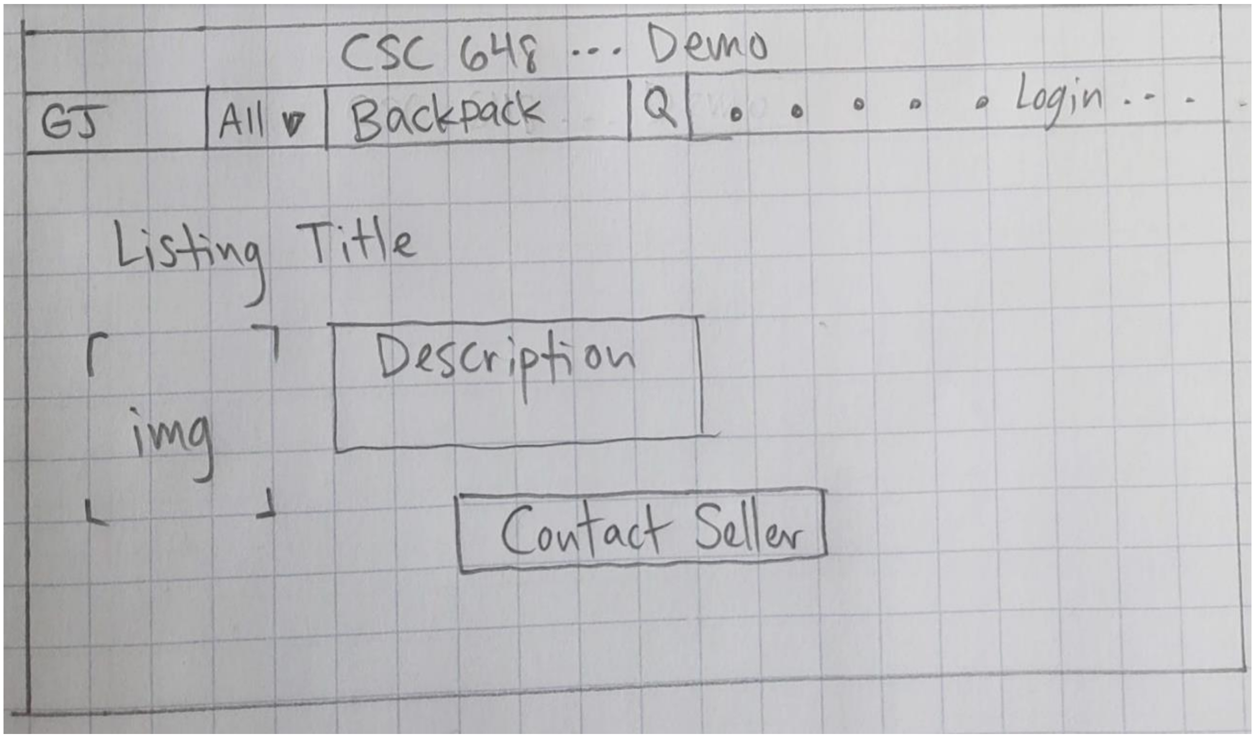
2. UI Mockups and Storyboards

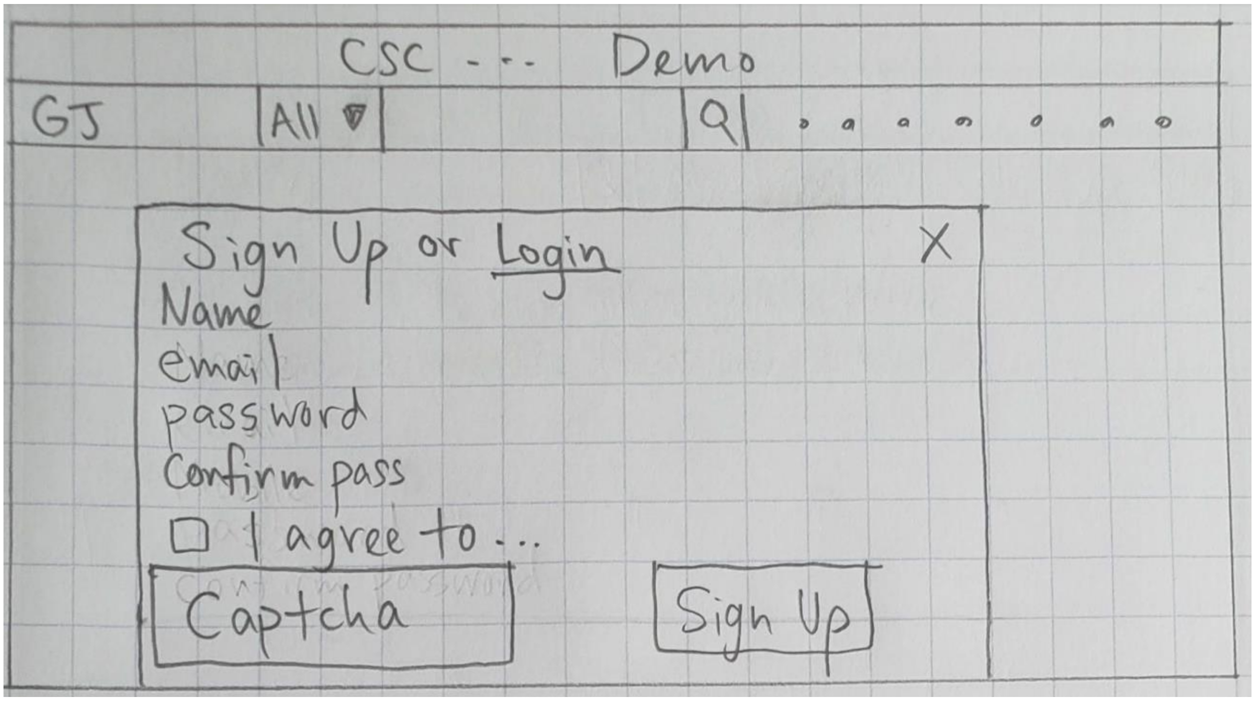
Use Case #1: Lily opens the application and searches for backpacks. She sorts by price and quickly browses through the picture of each search result until finding one she wants. After checking the listing and moving to purchase, she is prompted to register. The registration is quick and asks for a small amount of info before prompting her payment. After paying, she is given the seller's email and contacts them to agree on an exchange spot.

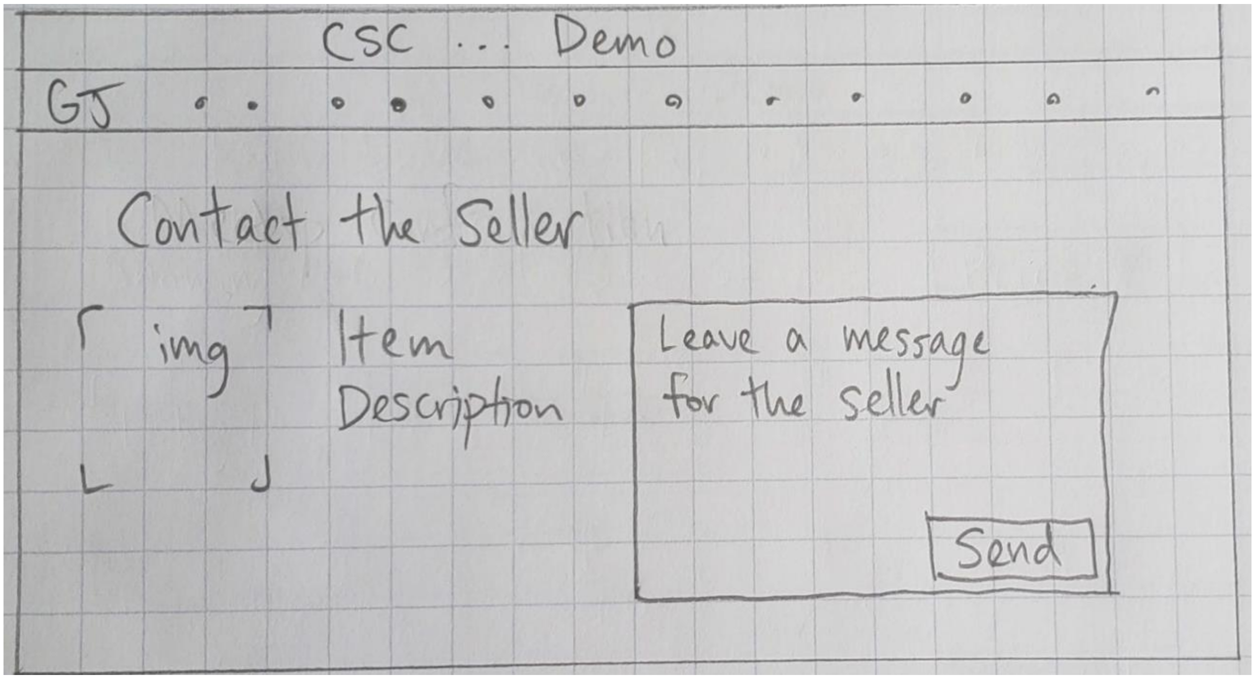


Gator Joe's is the only online marketplace by SFSU students, for SFSU students!

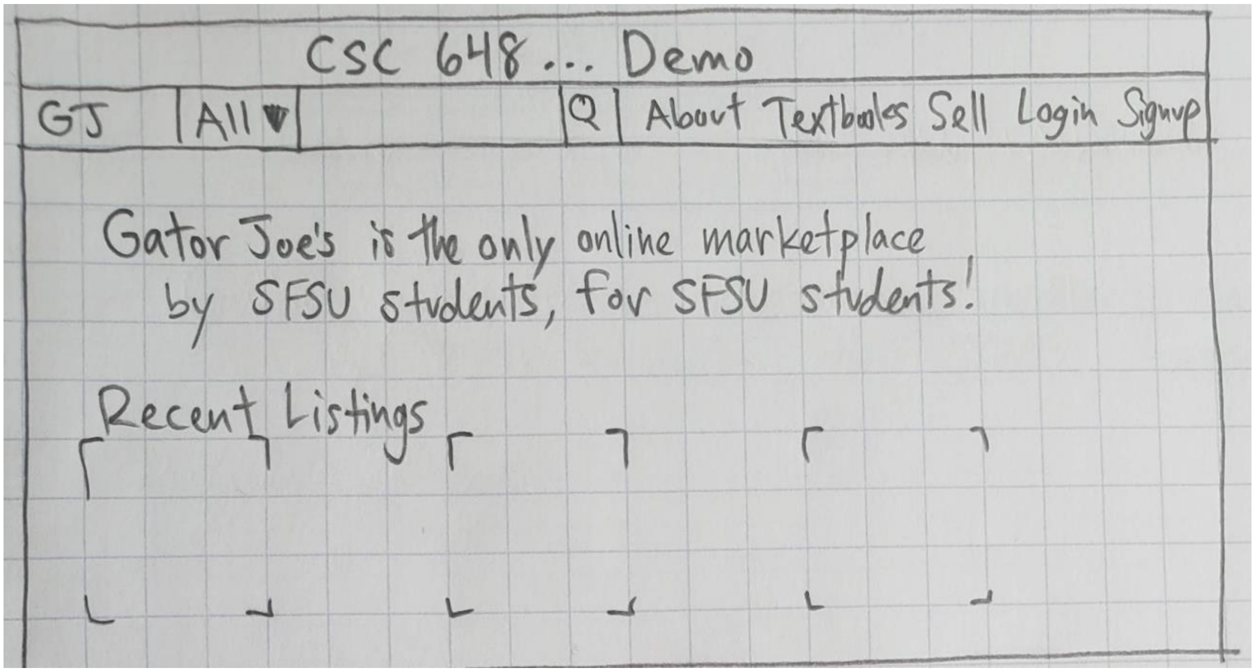


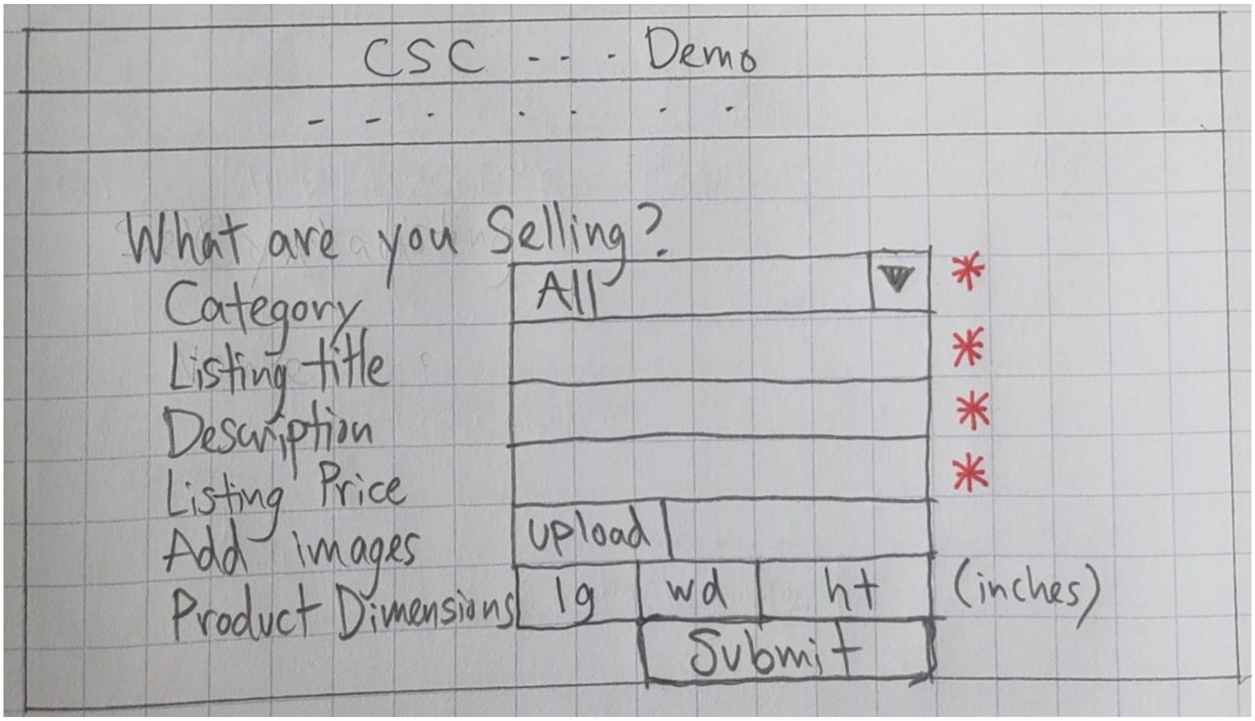

CSC 648 ... Demo
GJ
All ▾
Backpack
Q
Login
Listing Title
img
Description
Contact Seller

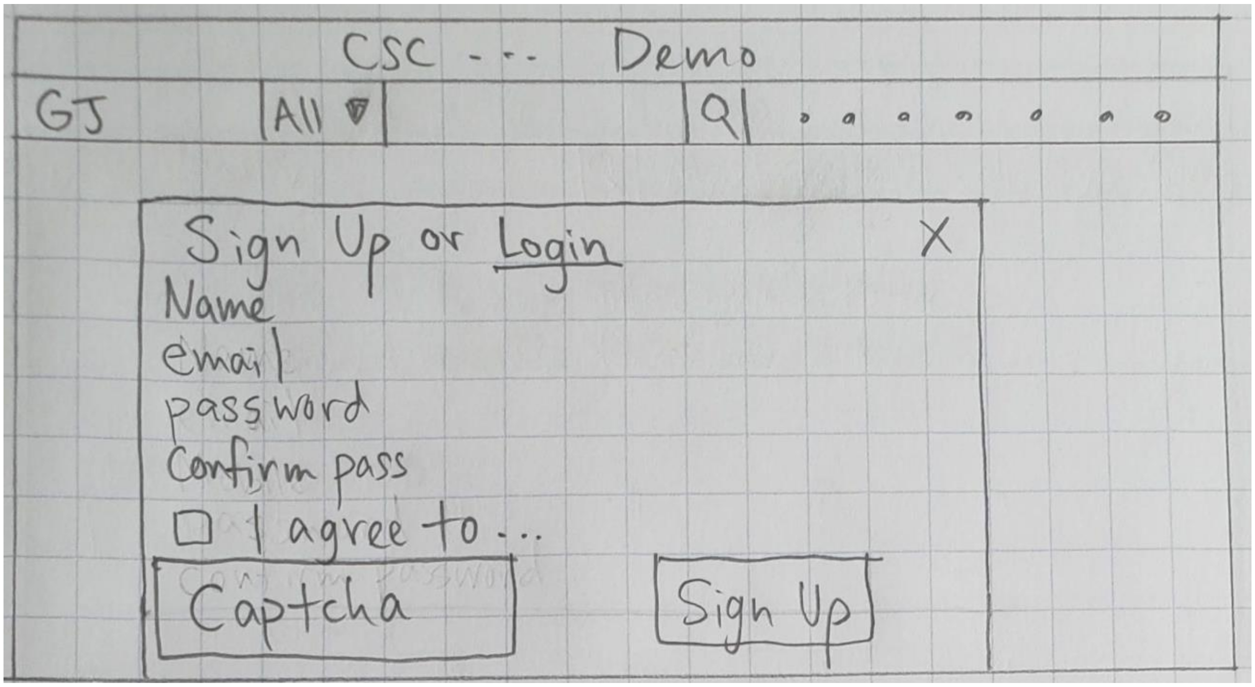

CSC ... Demo
GJ
All ▾
Q
Sign Up or Login
X
Name
email
password
Confirm pass
□ I agree to ...
Captcha
Sign Up

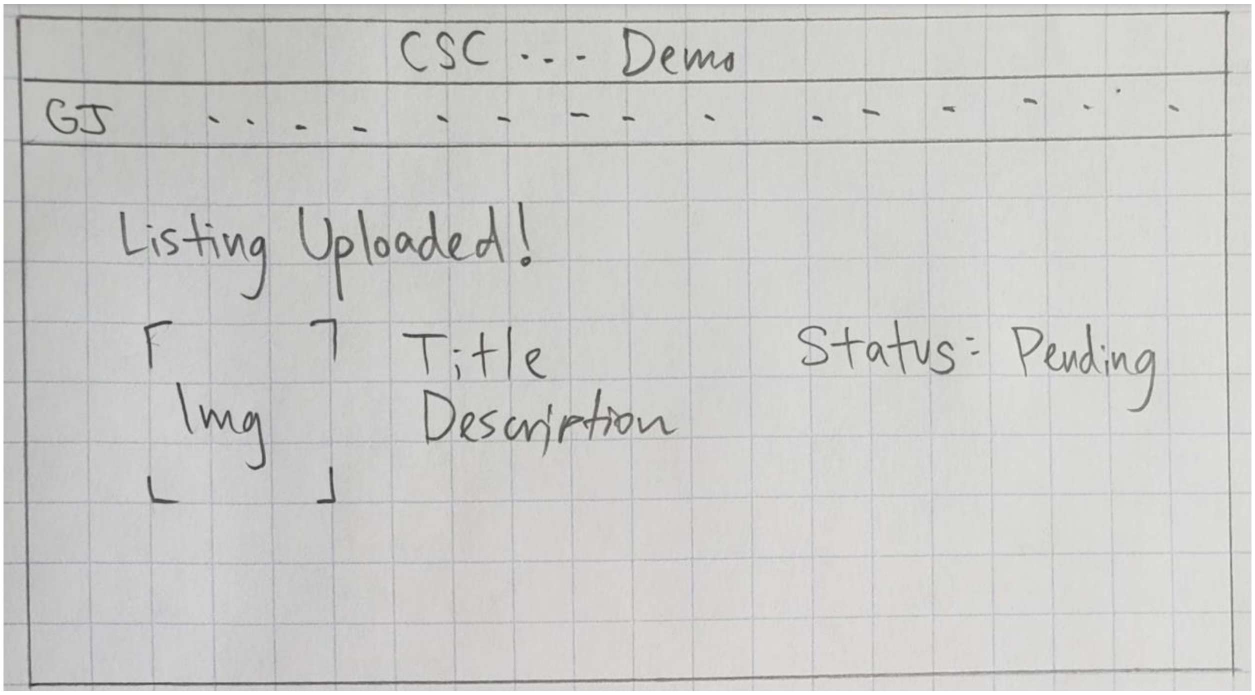

CSC ... Demo
GJ
Contact the Seller
img
Item
Description
Leave a message
for the seller
Send

Use Case #2: Noah opens the application and navigates to the seller page where he fills out the all the information for a nightstand he is selling. He is able to upload multiple pictures showcasing his item and also lists the furniture dimensions. After finalizing his listing, he is prompted to register and promptly fills it out. He is met with a confirmation that tells him his listing is waiting for approval.

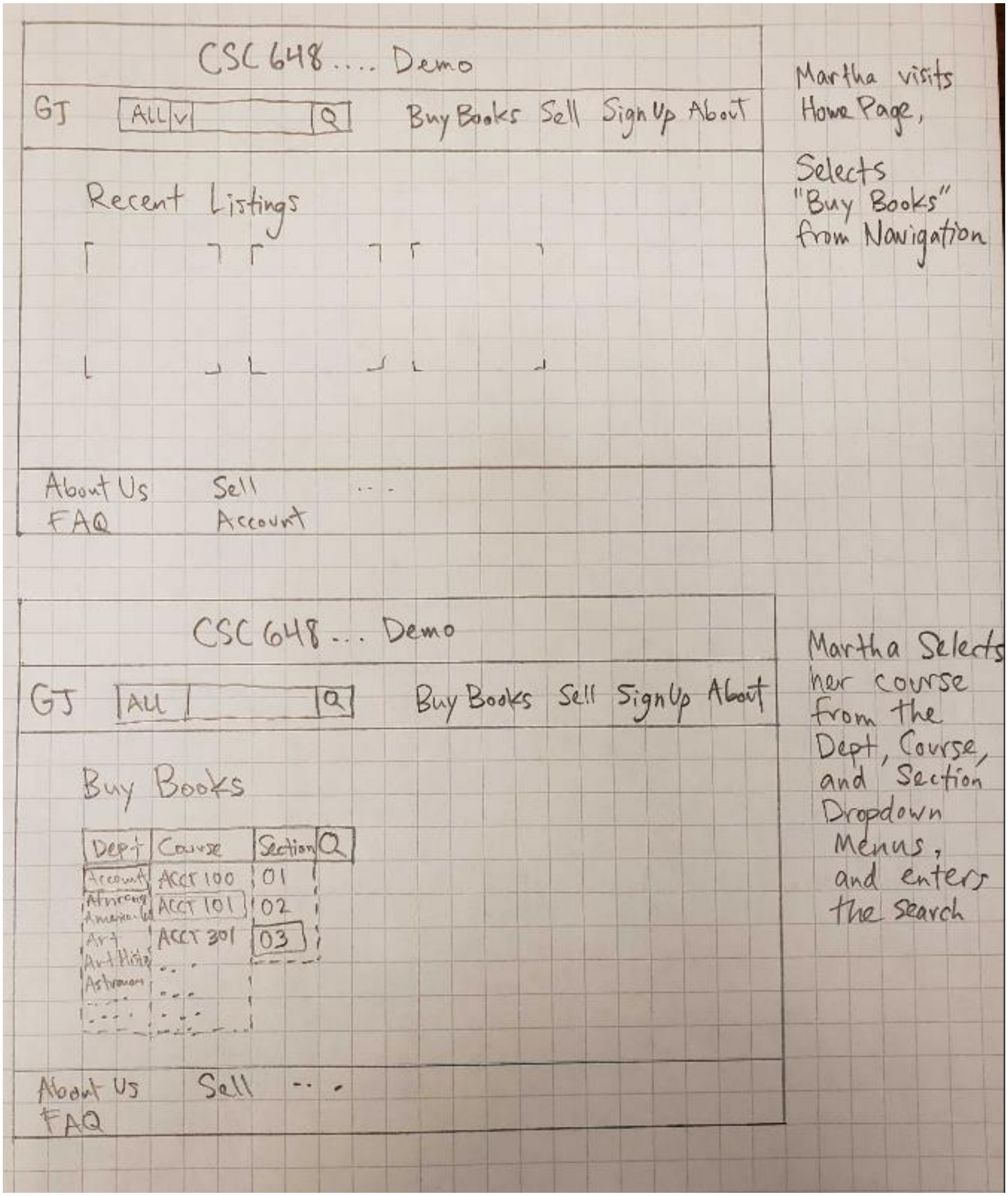

CSC 648 ... Demo
GJ
All ▼
Q
About
Textbooks
Sell
Login
Signup
Gator Joe's is the only online marketplace by SFSU students, for SFSU students!
Recent Listings




CSC ... Demo
GJ
All ▾
Q
Sign Up or Login
X
Name
email
password
Confirm pass
□ I agree to ...
Captcha
Sign Up

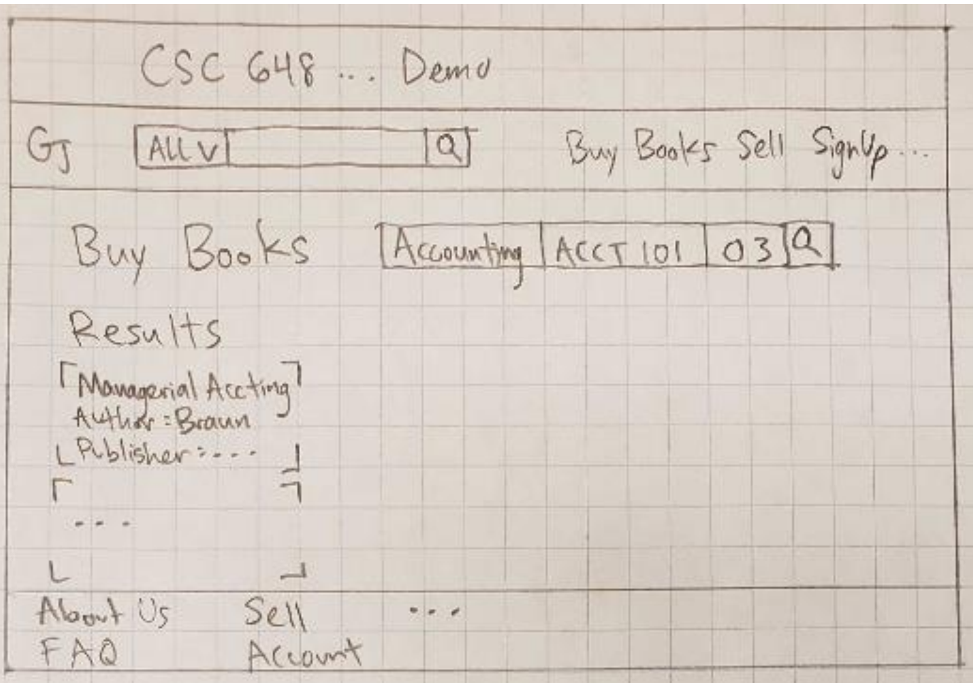

CSC ... Demo
GJ
Listing Uploaded!
Img
Title
Description
Status: Pending

Use Case #3: Martha opens the application and browses under the textbook category. She fills in her class' id and section into the sorting options and quickly finds the textbook needs. She opens the listing and is prompted to register or log. She fills in her login credentials and is taken to the payment page. After paying, she is given the seller's email to decide on a meeting spot.

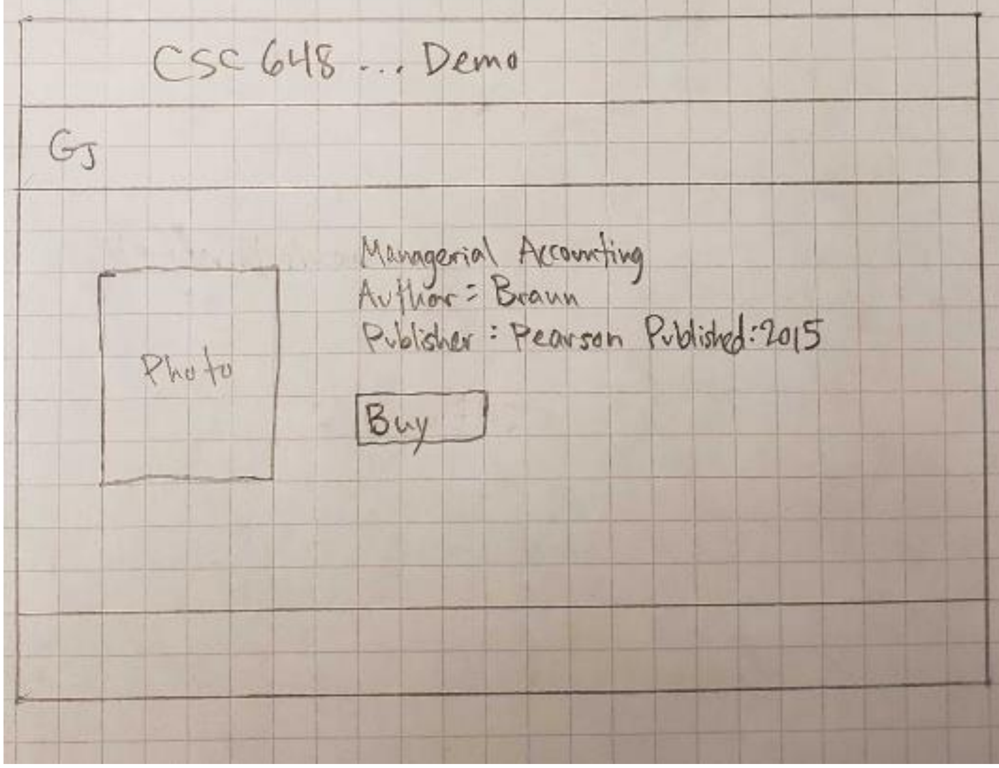


Martha visits Home Page, Selects "Buy Books" from Navigation

Martha Selects her course from the Dept, Course, and Section Dropdown Menus, and enters the search

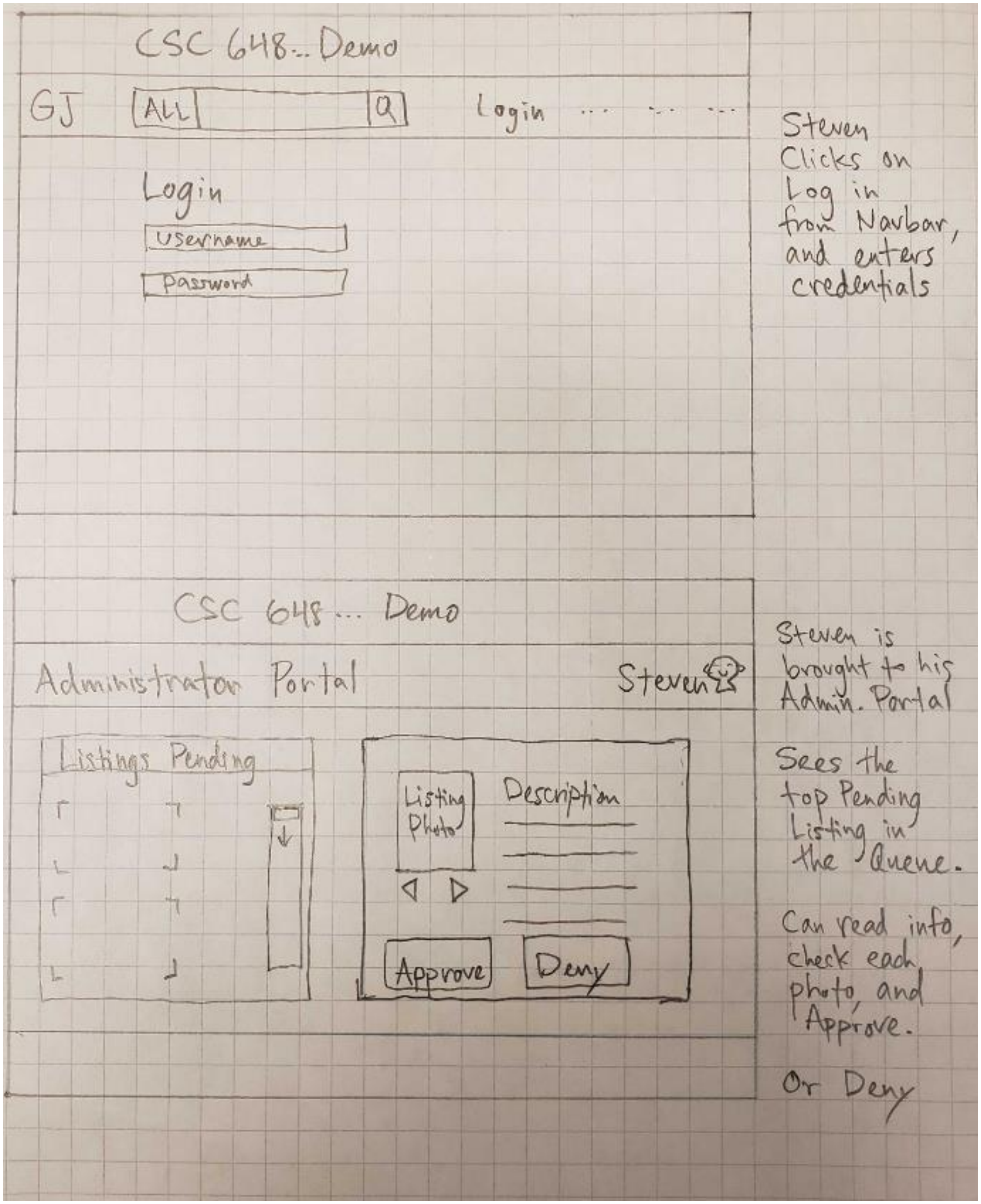


Martha Selects a listing from the search results



Martha can buy this book from the descripti page

Use Case #4: After opening the application, Steven logs in from the home page and is met with the first pending listing of the day. He checks the title, description and all uploaded images to ensure they are consistent and appropriate. He approves the listing and moves on to the next, which has an inappropriate picture attached. This time, Steven denies the listing and writes a quick sentence to explain why. When he finishes going through all listings, he notified that his work is done for the day.



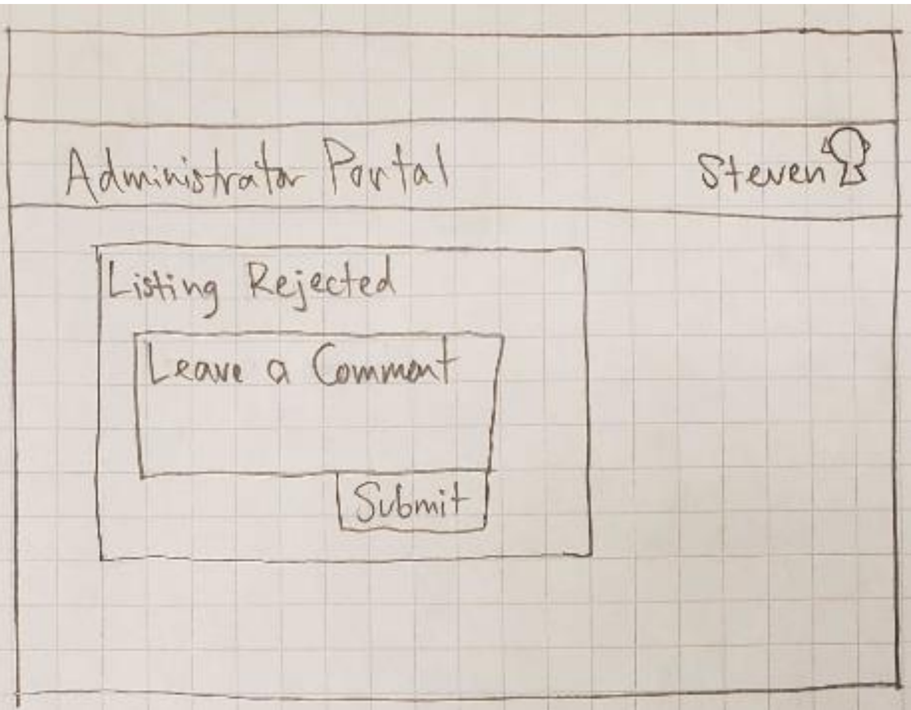
Steven Clicks on Log in from Navbar, and enters credentials

Steven is brought to his Admin. Portal

Sees the top Pending Listing in the Queue.

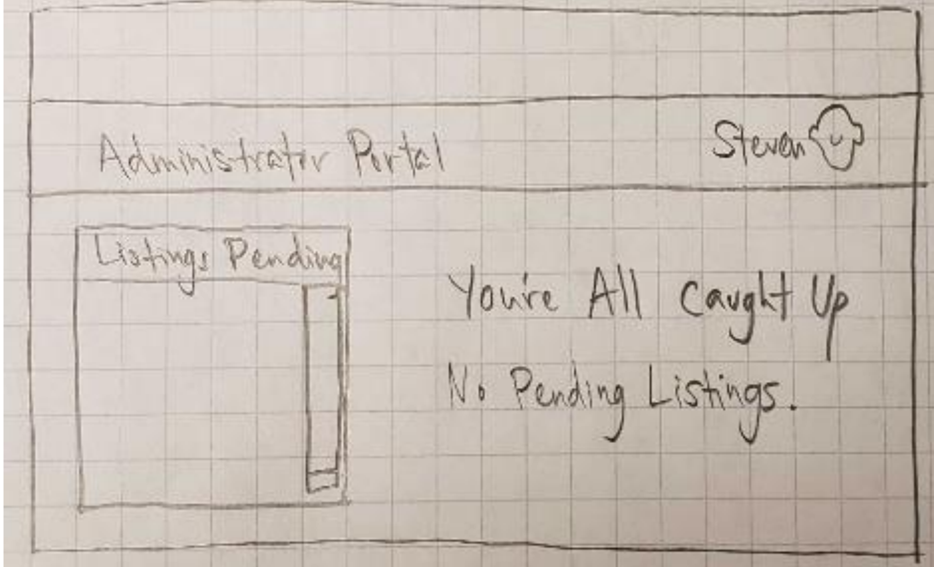
Can read info, check each photo, and Approve.

Or Deny



If Deny, Steven is prompted to leave a feedback comment explaining why the post was rejected.

After Submitted, Steen is directed back to the other Pending Listings.



When all Listings have been reviewed, Steven is dismissed by this page.

3. High Level Architecture, Database Organization

The organization of our database will be focused on two primary tables, with multiple secondary tables acting in support. The first primary table, RegisteredUsers, will contain a int field for the primary key ID and string fields for the login, password (which will be encrypted), name, and email. Administrators will be part of a separate Admin table. The primary key for the admin table will be a foreign key connected to the RegisteredUsers ID column.

The second primary table will be the Listing table. The Listing table will contain int fields for the listing ID, number of items sold, and the foreign keys for the Category_ID and Seller_ID (from RegisteredUsers). There will be string fields for the title and description. A Boolean entry for whether or not an administrator has approved the listing will be included. Finally, a photo column for the listing will be included as a binary blob.

Category will be its own table. It will only contain an ID and category name. There are two support tables containing addition information for listings. Both will have their primary IDs be a foreign key from the Listings table. The Book table will contain columns for author, class, section, and any addition prudent information for selling textbooks. The Furniture table will contain a dimensions column.

Orders will be separate from listings to differentiate them as listings that have been purchased. The Order table will contain the have columns for three foreign keys (in addition to its own ID primary key). There will be a column for the Listing.ID, the buyer's RegisteredUser.ID, and the seller's RegisteredUser.ID.

Search will primarily be handled by SQL queries and %like functions. The terms from the database that will be searched come from the columns Listing.Title, Listing.Description, and all Book column outside of the ID column. We will need to create APIs for creating new users and creating new listings.

RegisteredUsers

ID: int
Password: string
Name: string
Email: string

Admin

Admin_ID: int (foreign key: RegisteredUsers.ID)

Category

Category_ID: int
Category_Name: string

Listing
ID: int
Title: string
Description: string
Category_ID: int (foreign key: Category.Category_ID)
Sold: int
Approved: boolean
Seller_ID: int (foreign key: RegisteredIsers.ID)
Photo: binary blob

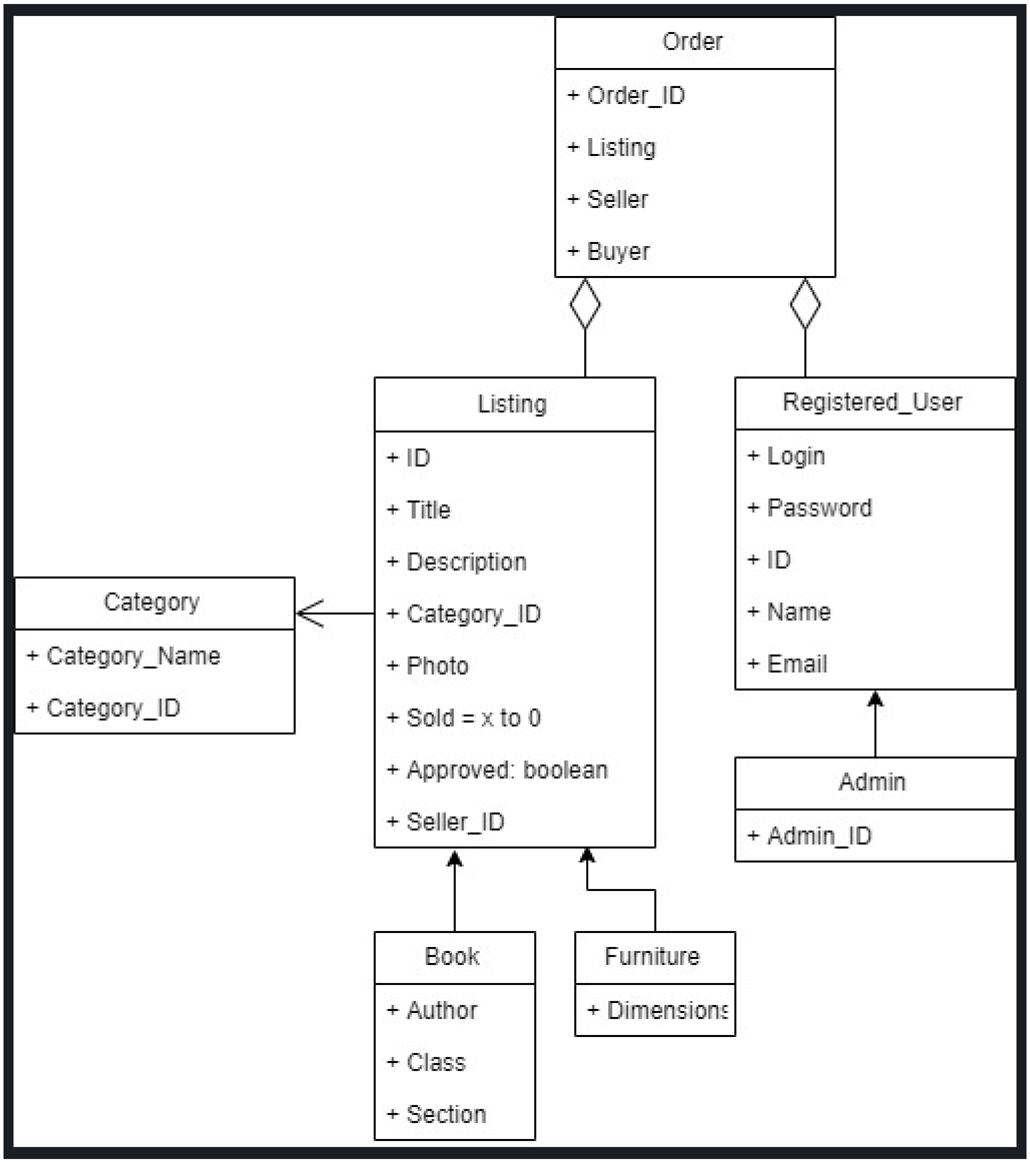
Book
ID: int (foreign key: Listing.ID)
Author: string
Title: string
Class: string
Section: int
ISBN: int

Furniture
ID: int (foreign key: Listing.ID)
Dimensions (string)

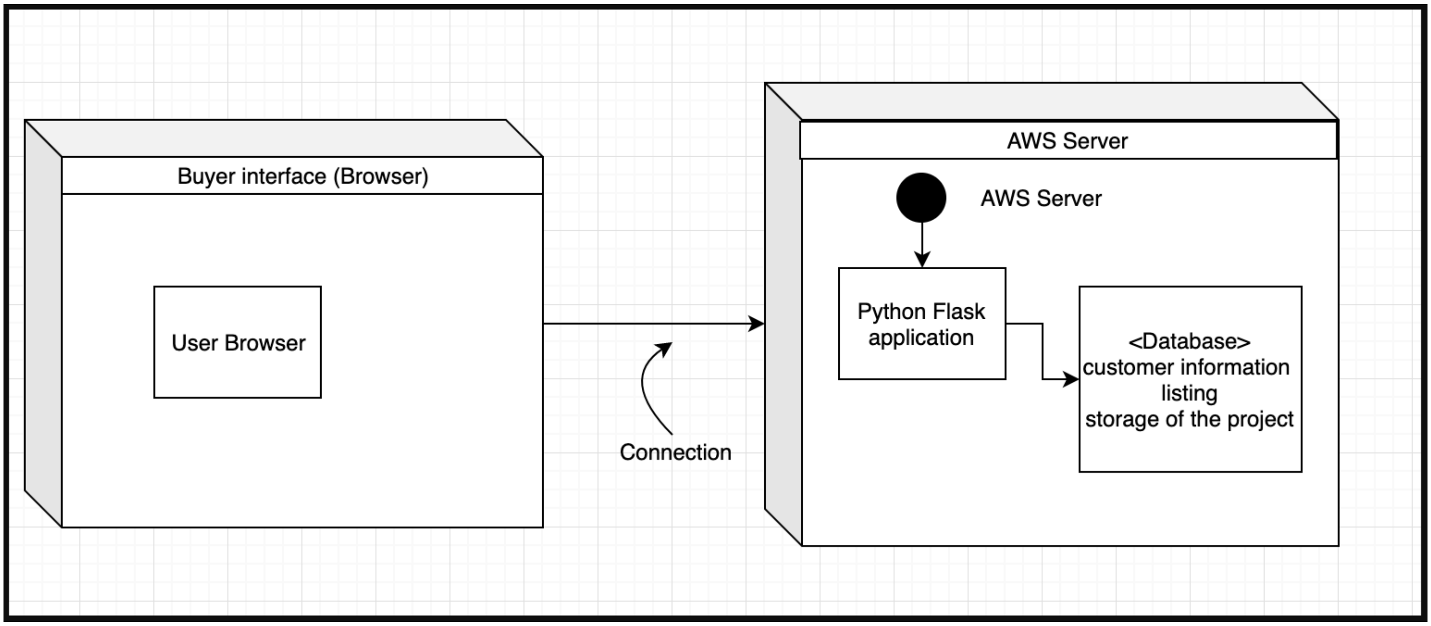
Order (Purchased Listings)
ID: int
Listing: int (foreign key: Listing.ID)
Seller: int (foreign key: RegisteredUser.ID)
Buyer: int (foreign key: RegisteredUser.ID)

4. High Level UML Diagrams

UML Class Diagram:



UML Component and Deployment Diagram:



5. Key Risks for Project

Skill Risk

Currently, our most prominent risk is that most of the team is learning or re-learning the languages used for the project. Between Boostrap, MySQL and Python Flask, our team members are not fully familiar with all three, which means a lot of learning is necessary along with project development. This issue can be resolved by taking careful looks through tutorial and documentation of each language. The development team will also be following the priority structure to ensure that the key features are completed by deployment and minimize time spent learning how to do extra features. In addition, code reviews will help inexperienced members learn from the more experienced ones within a particular language.

Schedule Risk

So far, we have committed to a few extra features in our project that extend the original scope of the application. Examples are functionalities like shopping cart, order history and furniture dimensions. As a safe way of managing expectations, those functionalities have been made as priority 2 or 3 to keep the focus on the primary functions of the application before extending ourselves to opportunistic features. Regardless, all extra features are intended for development if time permits.

Teamwork/Technical Risk

Some team members do not have a lot of experience working on projects as a group. A potential risk to our application is dealing with consistency and uniformity when combining everyone's parts together. Code reviews help with the understanding of each other's code but for merging it together, extensive testing and careful git merging will be required. In our current setup, front and back-end teams will have separate branches to commit too in order to avoid overwriting work.

6. Project Management

This team is managed by Piero, team lead, through Slack and Trello. Slack is used for announcements, checking progress and all general communication among the team. The slack workspace set up for the project has separate chat channels for front-end, back-end, git requests, and announcements. Regular check-ins are done through Slack to ensure that team is progressing and on track for what the current milestone may be. Trello is used as a clear way to display and keep track of all relevant tasks and their present priorities. Any work assigned to team members will be communicated in person, announced on Slack and reflected on our Trello dashboard.

Project work is assigned to either the back-end or front-end leads who are responsible to split up the work with the corresponding members of their sub-team. All major milestones and deadlines will have repository branches corresponding to the back and front teams to avoid any code discrepancies while they work on their assigned tasks. Team lead and back-end lead are responsible for any issues with the webserver itself.