SW ENGINEERING CSC648/848
FALL 2019 Section 01

Gator Barter

Team # 04 (Local Team)

Aditya Bodi, Team Lead | abodi@mail.sfsu.edu
Alan Ng, Frontend Lead
Philip Yu, Frontend Engineer
Alex Kohanim, Backend Lead
Deming Yan, Full Stack Engineer
Akshay Kasar, Git Master- Backend Engineer
Tejasvi Belsare, Document Master- Backend Engineer

"Milestone 2"

Date: October 17th, 2019

Version	Date Submitted	Owner	Description
1.0	October 11th, 2019	Tejasvi	Initial draft with all sections and content
1.1	October 16th, 2019	Philip, Alan, Deming	Updated UI Mockups, storyboards
1.2	October 17th, 2019	Aditya	Document review and formatting
1.3	November 13th, 2019	Philip, Tejasvi	Changes as per Professor's feedback
1.4	November 22nd, 2019	Tejasvi	Document review, formatting and freeze

Table of Contents

1. Functional Requirements – prioritized

1.1 Priority 1

- Unregistered Users
  1. Unregistered users shall be able to browse the Gator Barter website.
  2. Unregistered users shall be able to search any item as per their need on the Gator Barter website.
  3. Unregistered users shall be able to search as per Category on the Gator Barter website.
  4. Unregistered users shall be able to see search results in an organized manner.
  5. Unregistered users shall be able to filter search results.
  6. Unregistered users shall be able to create an account on Gator Barter by using credentials.
  8. Unregistered users shall be able to initiate a post for selling an item onto the website which shall require registration to submit the post (lazy registration).

- Registered Users
  9. Satisfy all priority 1 functional requirements of unregistered users.
  10. Registered users shall be able to login into their Gator Barter account by using credentials.
  11. Registered users shall be able to post their item(s) for sale on Gator Barter website and see the item(s) they posted.
  13. Registered users shall be able to send messages for buying/trading.
  14. Registered users shall be able to trade the items with other items on Gator Barter website also can trade items for free.

- Administrator
  19. Administrator shall be required to approve or reject the pending item(s) in the selling queue.
  20. Administrator shall be able to remove inappropriate or illegal item(s).
  21. Administrator shall be able to block suspicious registered users.

1.2 Priority 2

- Unregistered Users
  7. Unregistered users shall be able to search for class materials using abbreviated course name and course number.

- Registered Users
  9. Satisfy all priority 2 functional requirements of unregistered users.
  15. Registered users shall be able to communicate with the Administrator.

16. Registered users shall be able to remove the items posted if they no longer wish to sell the item.
17. Registered users shall be able to mark the item as sold once sold out.

1.3 Priority 3

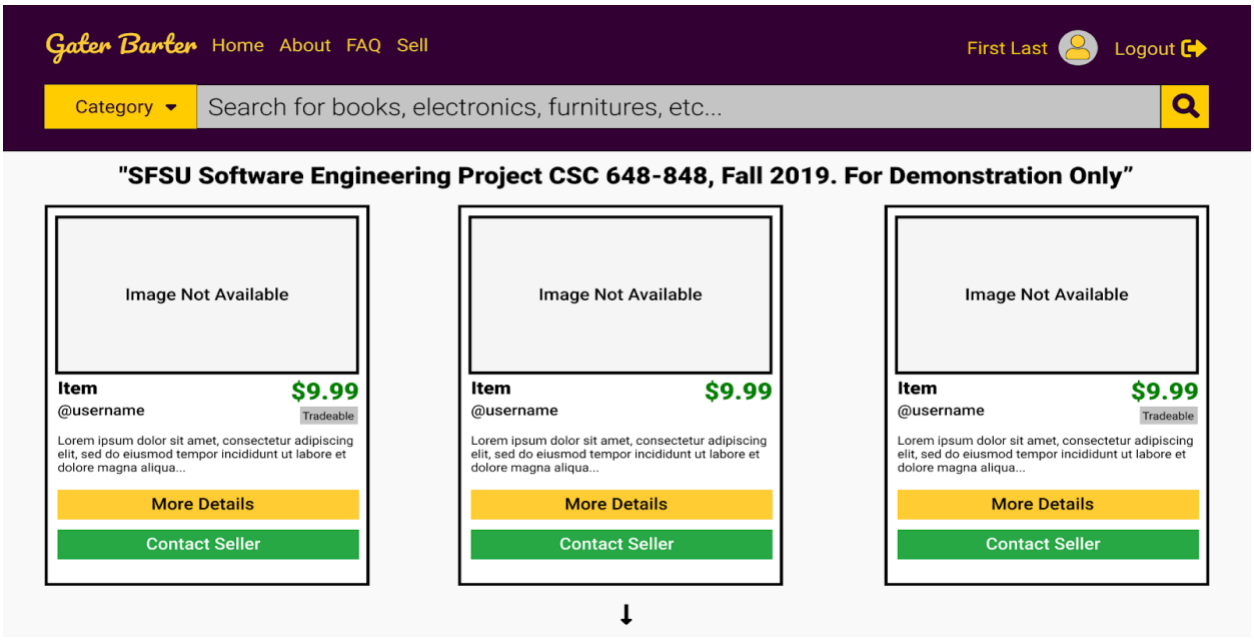
- Registered Users
  9. Satisfy all priority 2 functional requirements of registered users
  12. Registered users shall be able to check the list of their transaction history.
  18. Registered users shall be able to report other users to the Administrator for violating website policies and terms of service.

- Administrator
  22. Administrator shall be able to contact registered users if required.

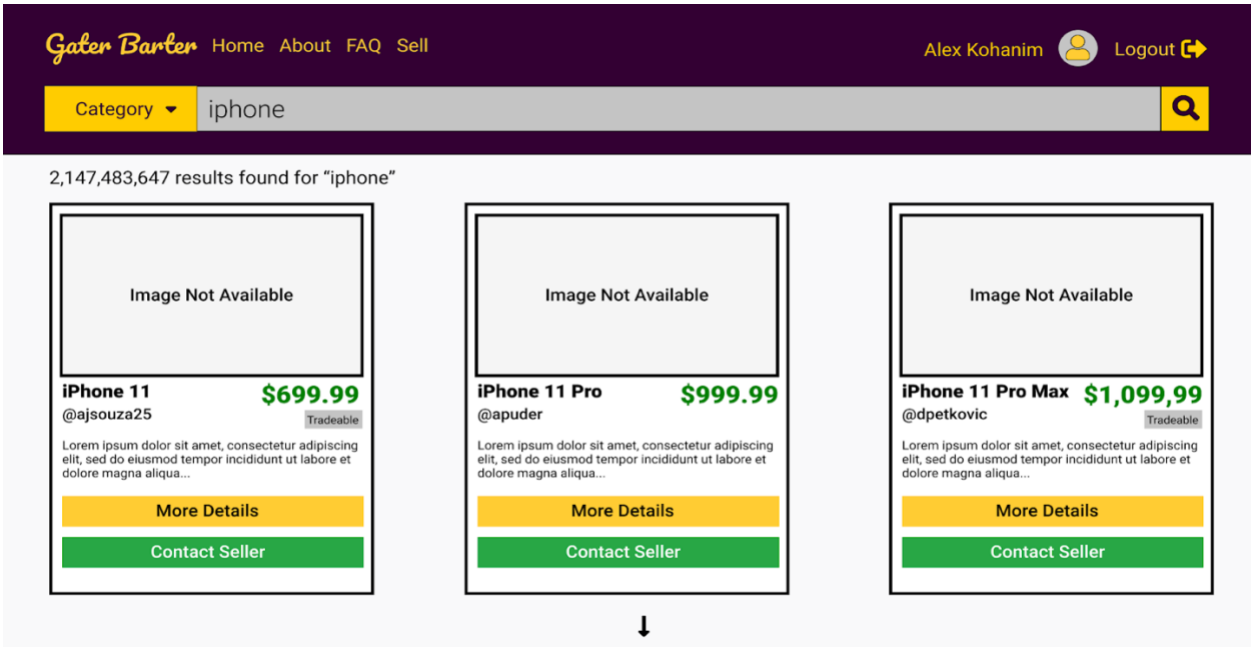
2. UI Mockups and Storyboards (high level only)

2.1 UI Mockups

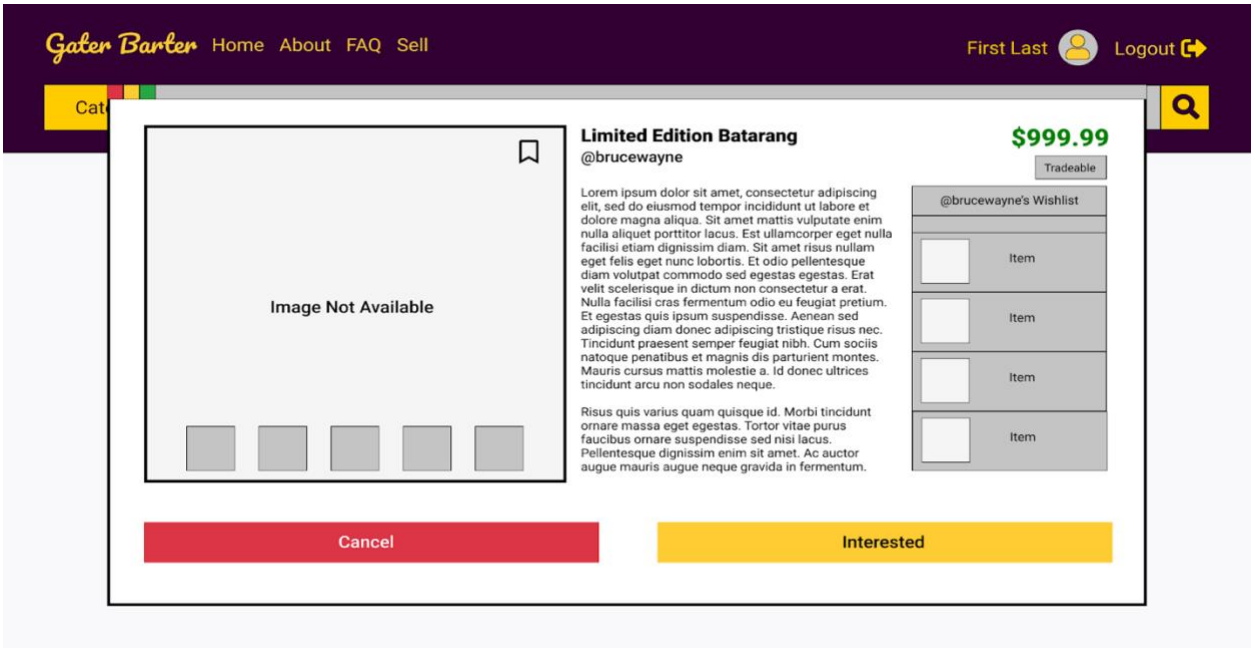
1. Home Page - This is the default home page for Gator Barter website where any user can browse and search by entering specific name or by category.



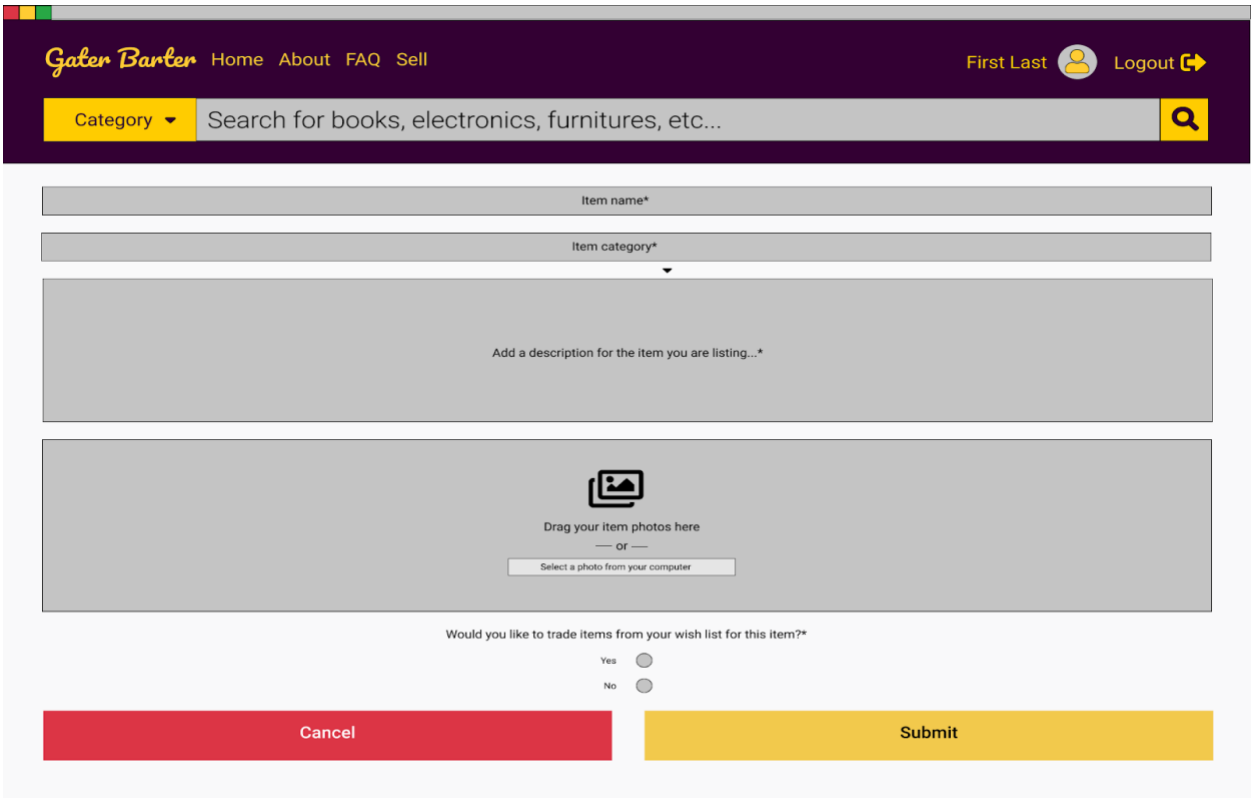
2. Search Page - This is the search page returned upon searching for a specific keyword (e.g. iPhone).



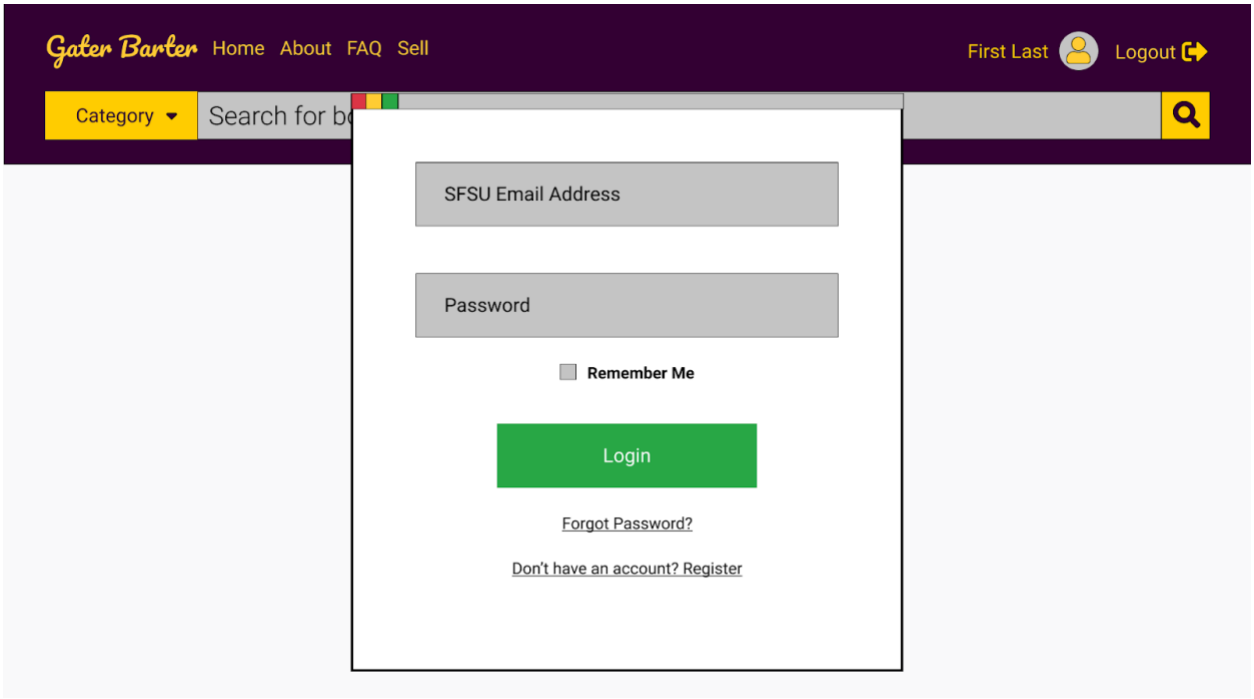
3. View Item Details Page - This is the Item detail page that opens in a new tab upon clicking View Details, where users can look up the details of the item.



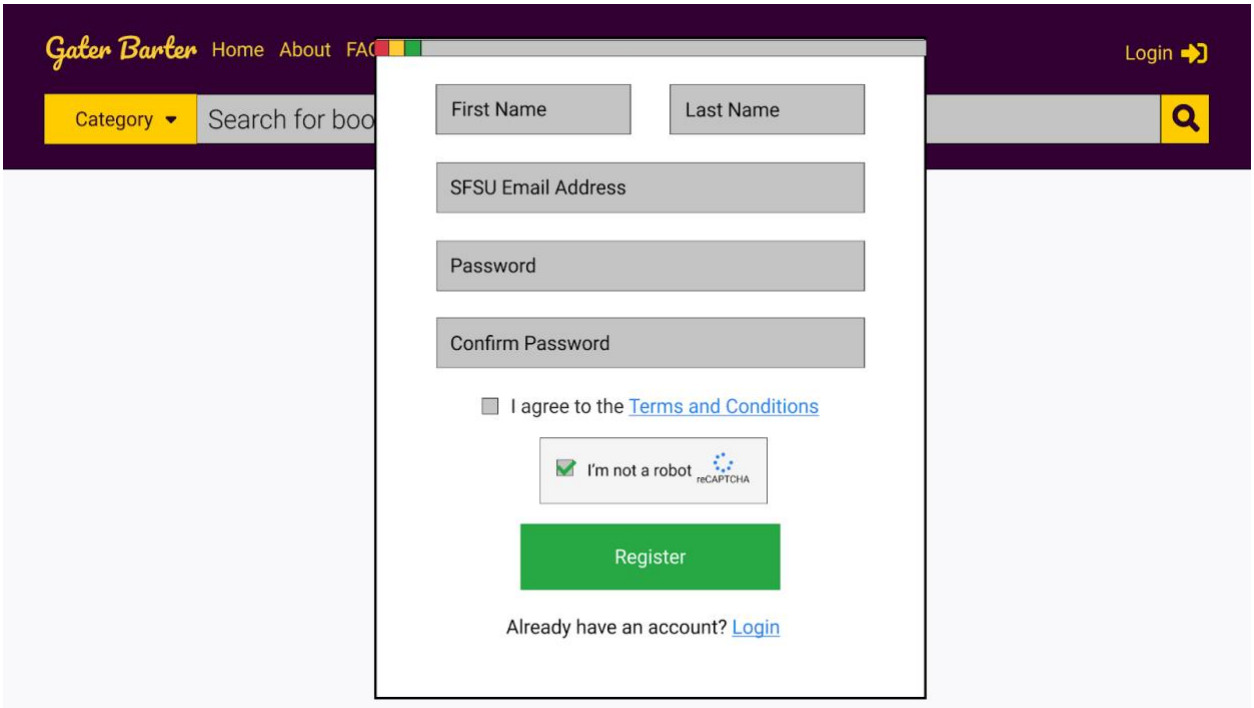
4. Post an Item Page - This page allows a registered user to post an item on the Gator Barter website by entering information in the fields such as Title, Price, Category, etc.



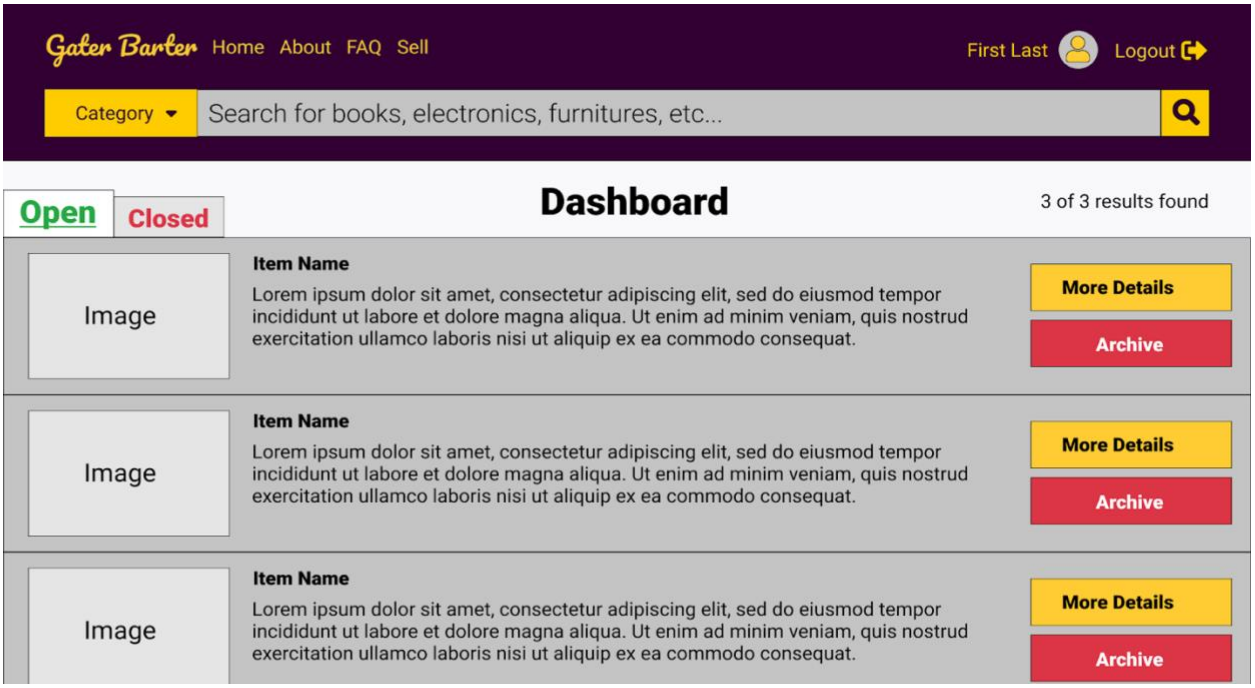
5. Login Page - This is the page for logging in with the credentials for registered users and administrators.



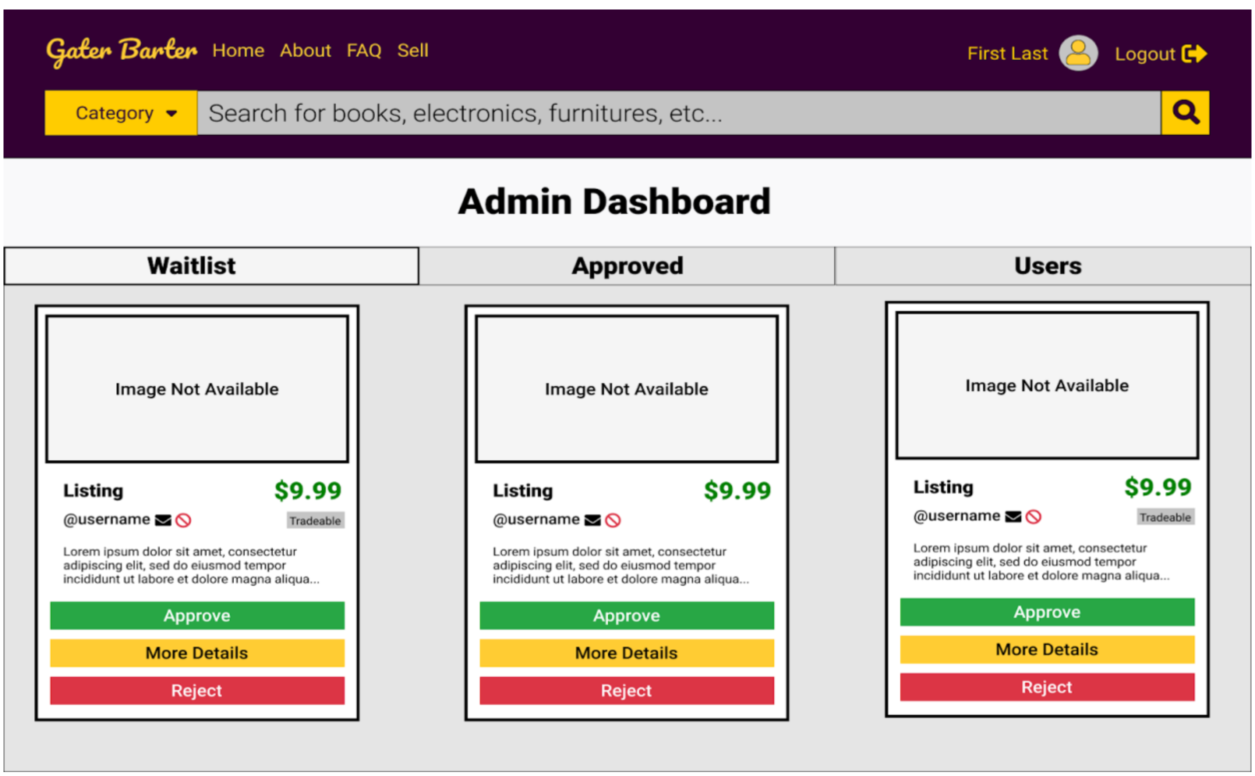
6. Registration Page - This is the registration page where unregistered users can create a Gator Barter account by entering details such as email and password.



7. Transaction History Page - This is the transaction history page for a registered user which shows all their items posted for sale, previously sold items or previously bought items.



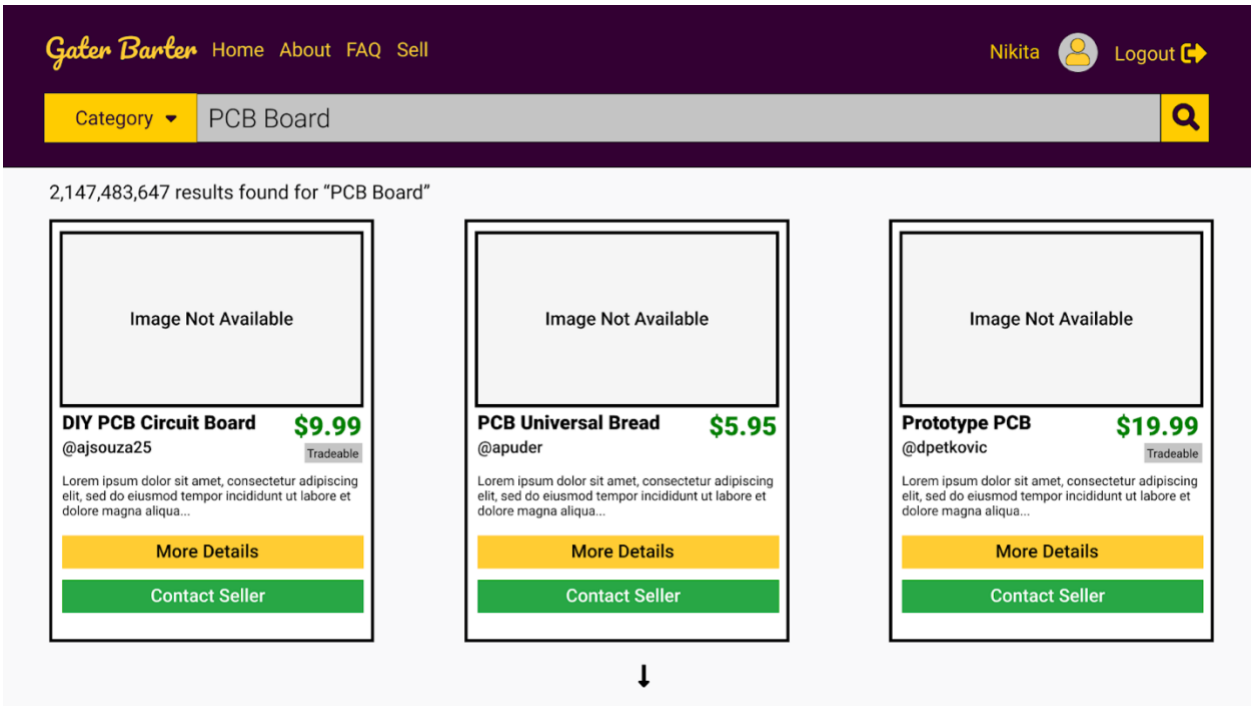
8. Admin Dashboard Page - This page is only displayed to administrators who has the privilege to approve or deny items posted by a registered user.



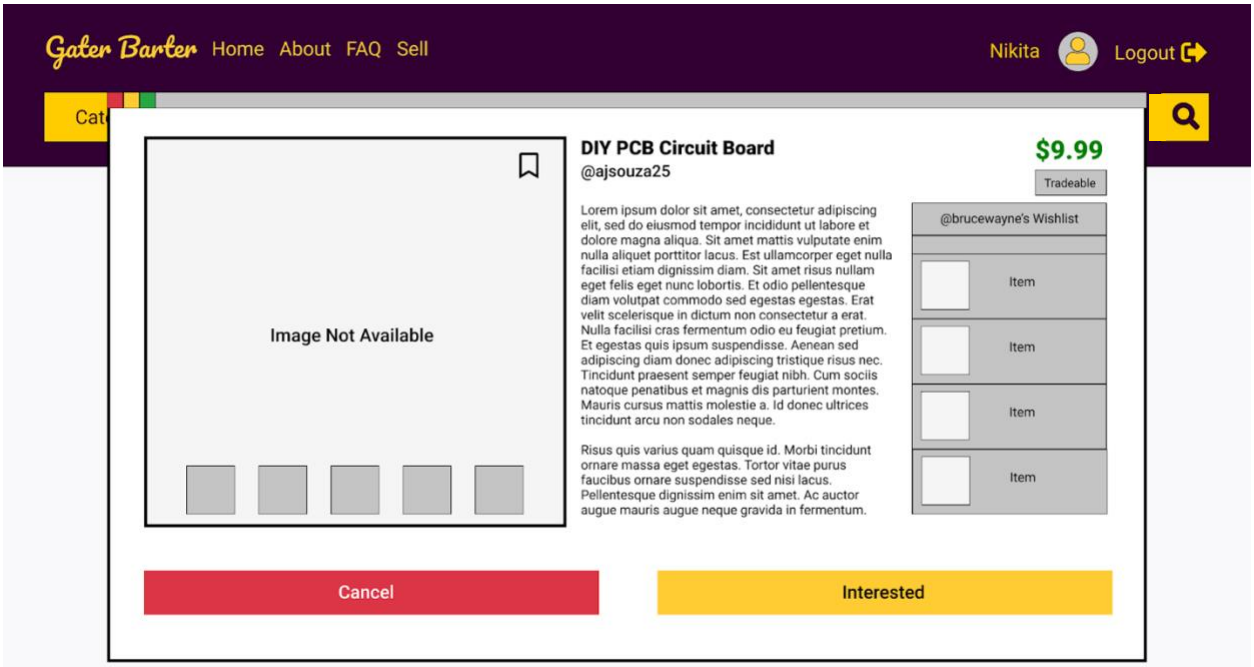
2.2 Storyboards

1. User Case 1:

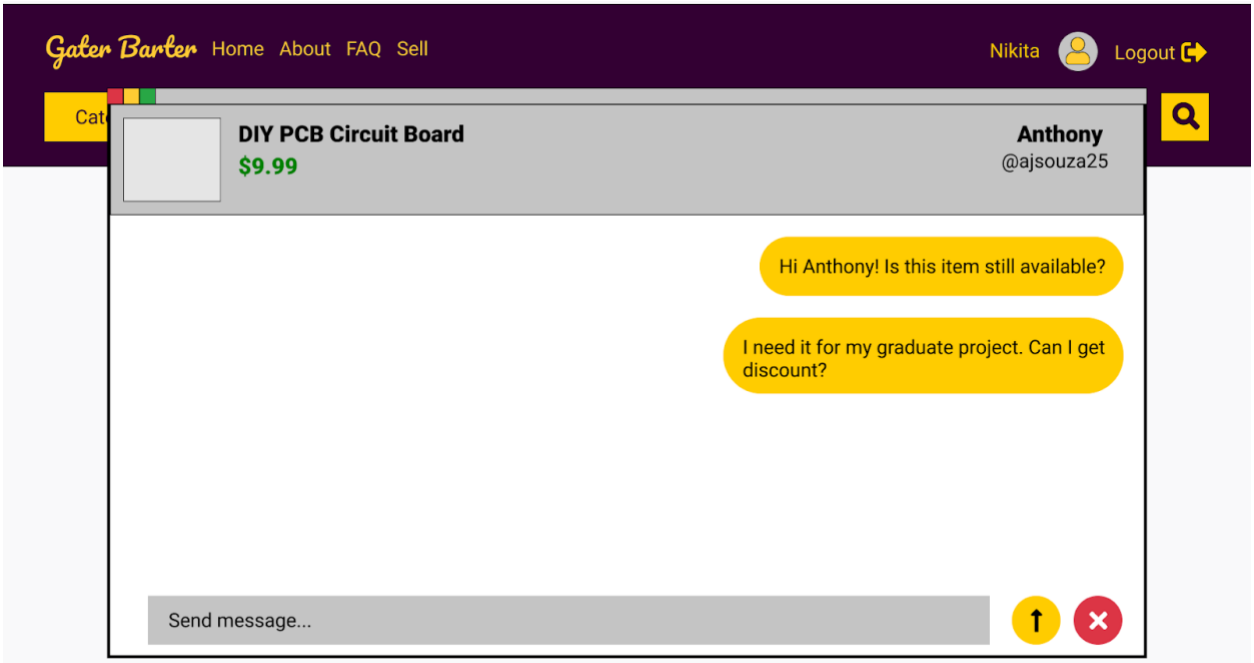
   Nikita is a graduate student from EECS department at SF State. She lives near campus. She needs a PCB board for her Embedded course project. As project deadline is closer, she doesn't have enough time to buy the new board. Buying online would be a slow process considering the shipping time. Also, used PCB board would serve her perfectly from this project's point of view. She needs it within the next 2 days, so that she can cover up her project work and 4 complete milestones on time.



Nikita searches for PCB boards from the home page and is shown a list of PCB boards. Here she can pick and choose from the list of sellers and price. When she has decided, she can click on interested to see a more detailed view.



In this example, Nikita is interested in Anthony's PCB board and clicked interested on his listing. She has a more detailed view on his listed item. She can confirm if she is satisfied with the price and description and then click on interested once again to initiate a chat with Anthony to begin the transaction.

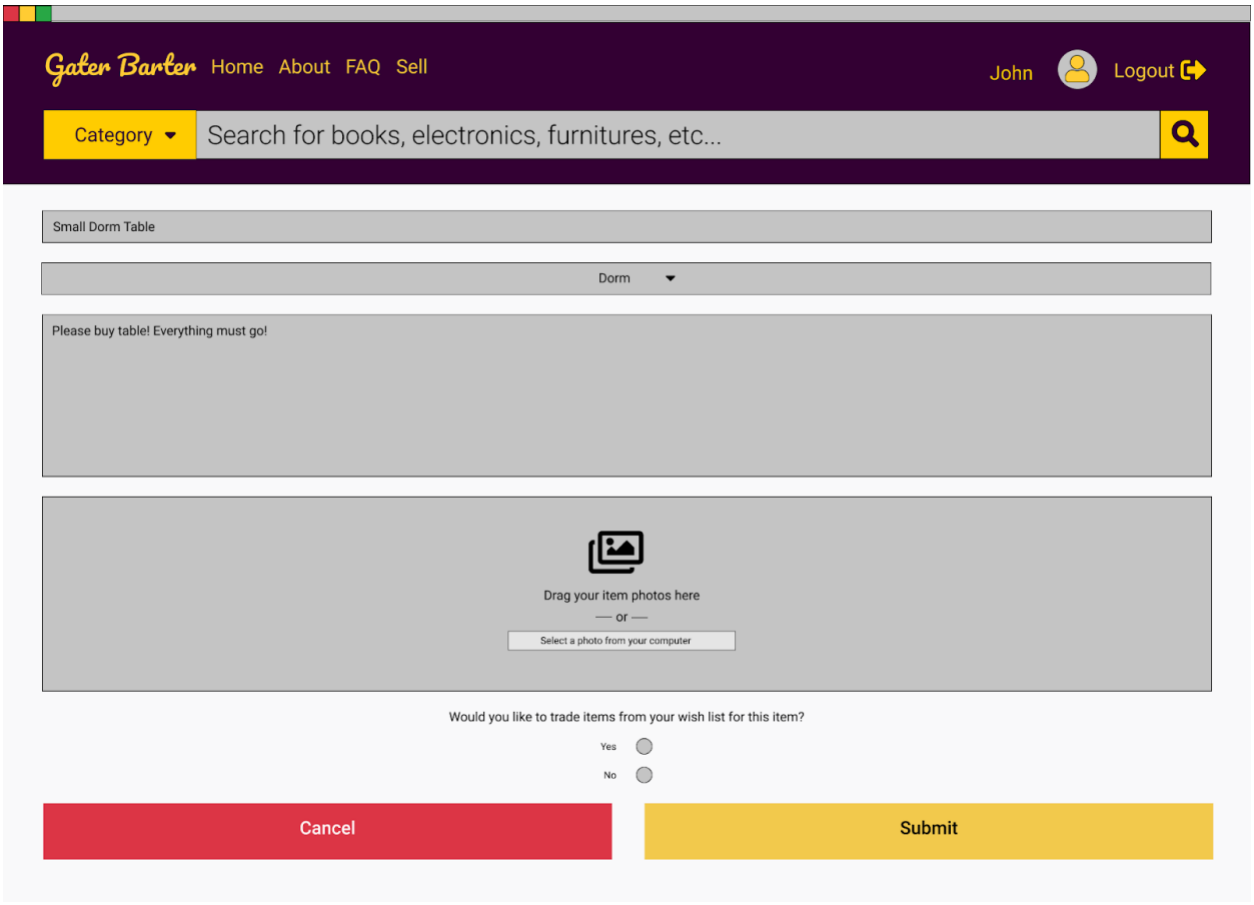


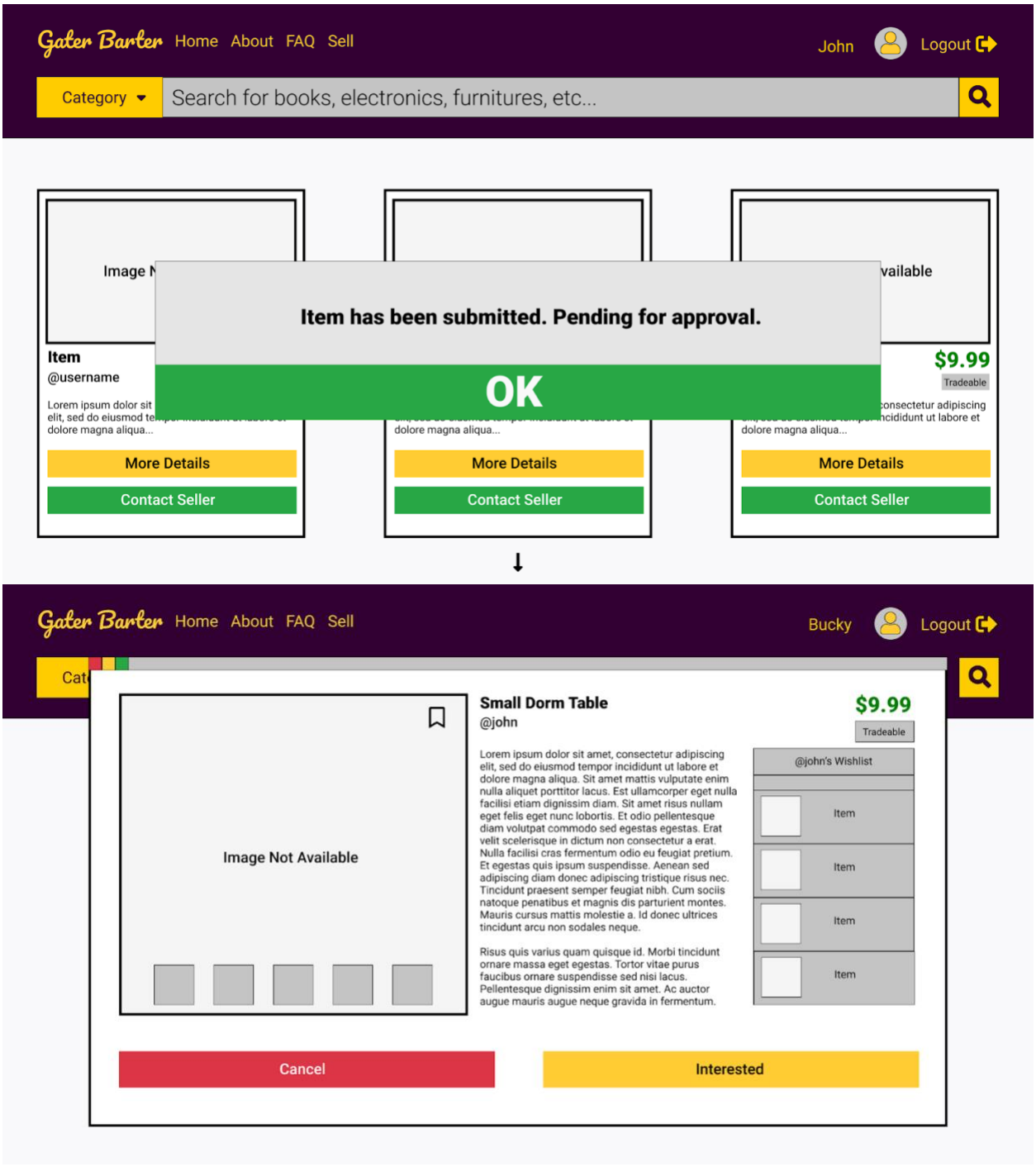
Here she would discuss with Jack if the item is still for sale and finalize the payment method and meeting place.

2. User Case 2:

John is a student from SF State Mathematics department. He recently graduated and got a great job offer in Texas. Now, he is moving out of SF State dorm and wants to sell his room items quickly before moving. Selling online would be a slow process. It would save a lot of time if he can sell those items to someone from the dorms or SF State itself.

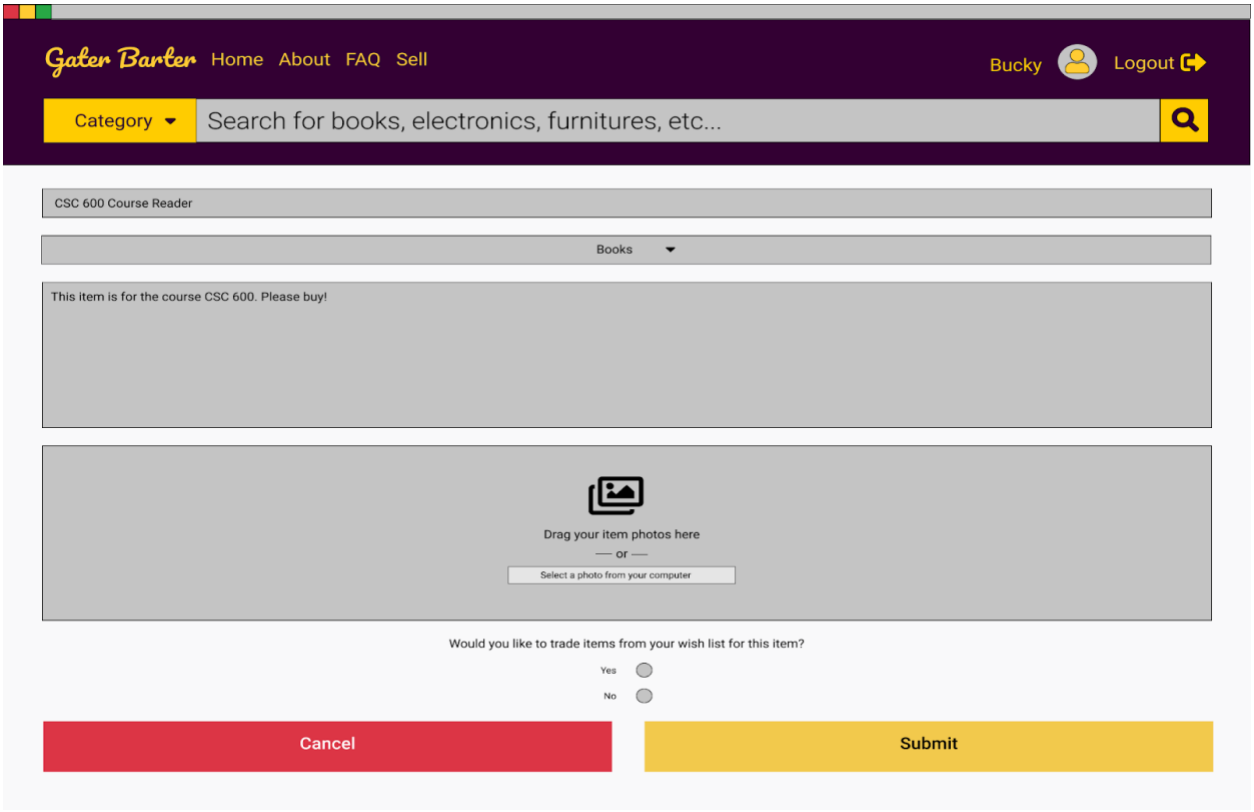
The following is the page displayed when the user posts an item to sell.

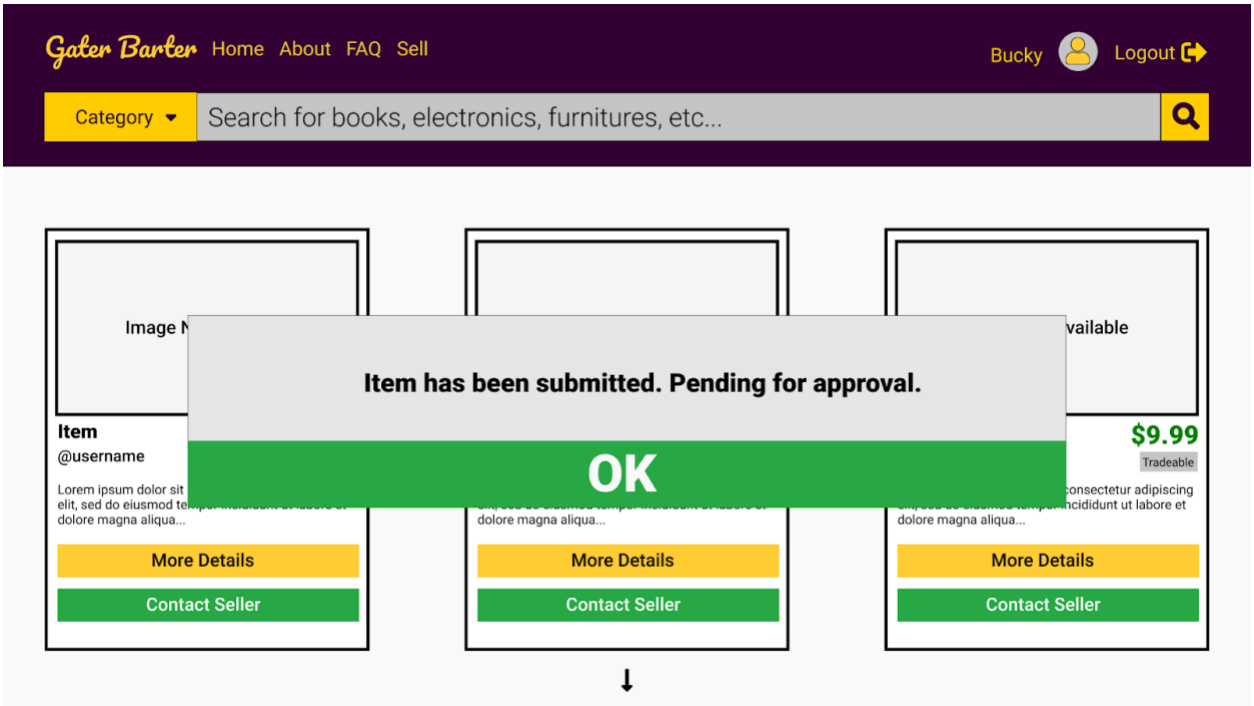




3. User Case 3:
   Bucky is a senior student from the Computer Science Department. He lives in San Jose and is on campus 4 days a week. Bucky has few CS course-related textbooks, which he doesn't need once he graduates. He wants to trade those readers/textbooks with some other stuff required for his new room. As course material is the same for courses, it would

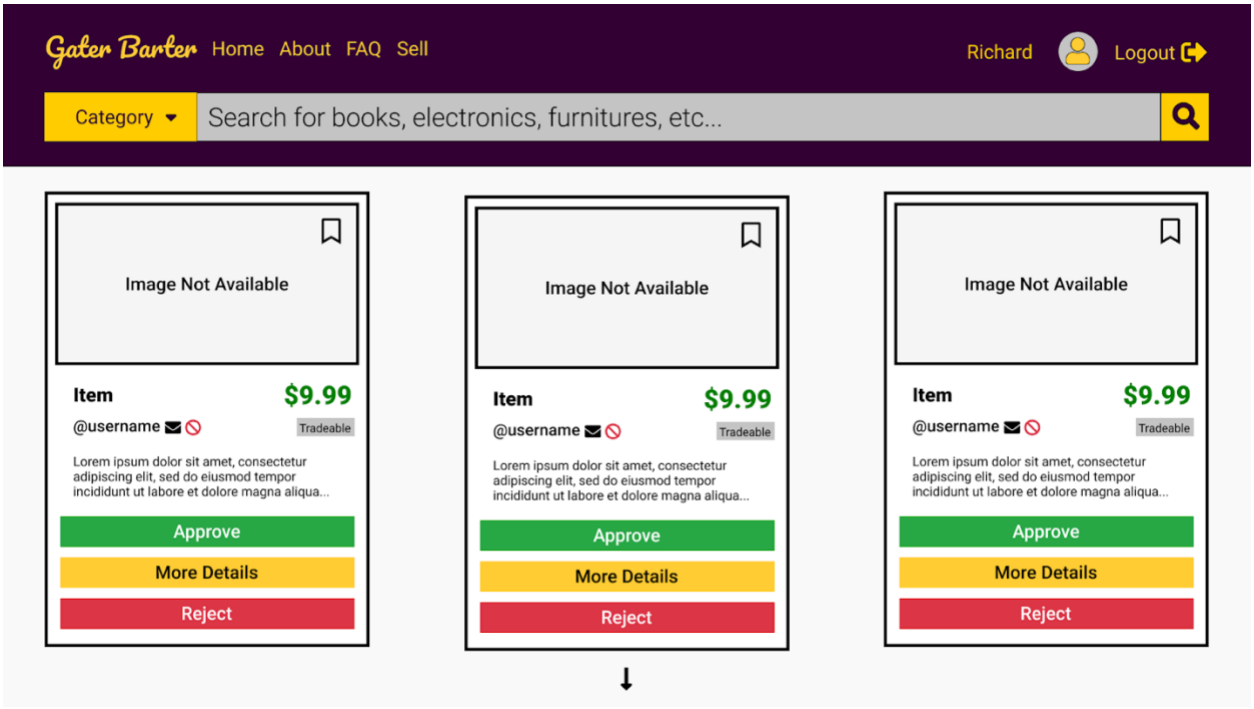
be good if someone from SF State CS department can trade those textbooks. As Bucky comes frequently to the school, he can trade things with the students with an ease.

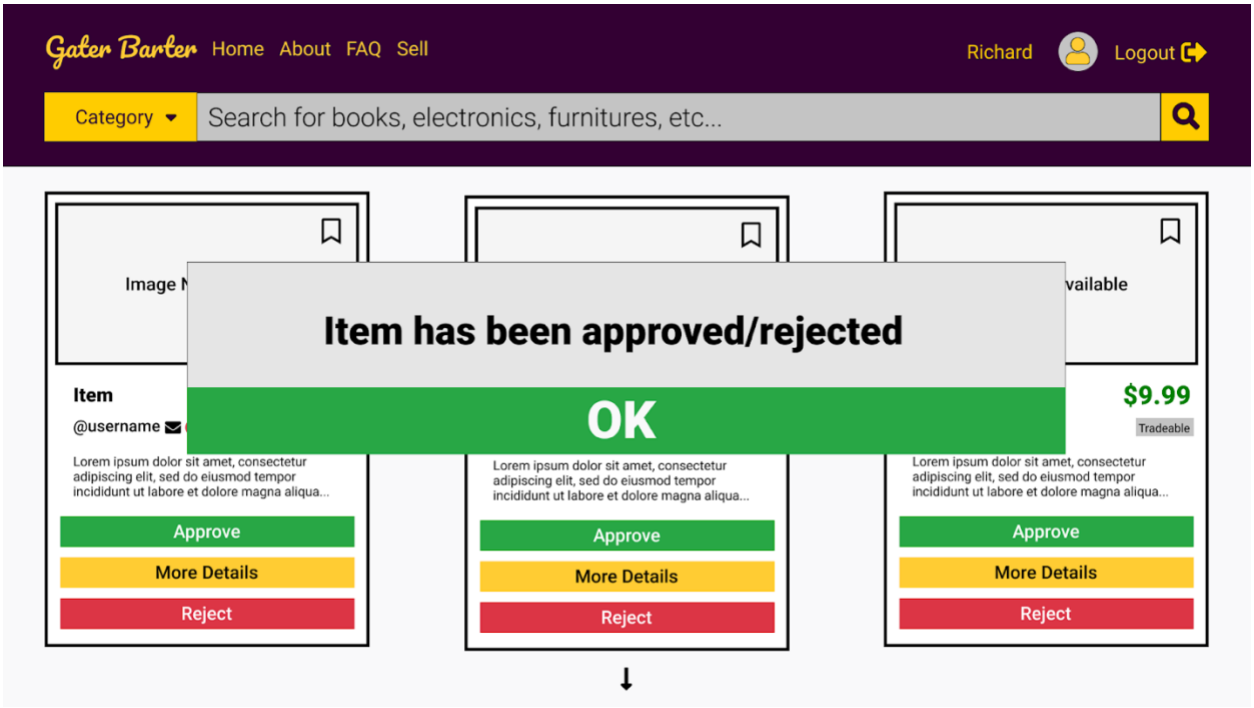




4. User Case 4:

   Richard is an Administrator on the Gator Barter website. He supervises the website and has the right to approve or reject the items posted by registered users.





3. High level Architecture, Database Organization

3.1 Database Organization

Database will contain below tables –

1. user - This is the user registration table when an unregistered user registers on Gator Barter.
   It will have following data fields –

   `u_id` bigint NOT NULL AUTO_INCREMENT
   `u_email` varchar(128) NOT NULL UNIQUE
   `u_pass` TEXT(256) NOT NULL
   `u_is_admin` int NOT NULL DEFAULT '0'
   `u_created_ts` TIMESTAMP
   `u_updated_ts` TIMESTAMP
   `u_fname` varchar(128) NOT NULL
   `u_lname` varchar(128) NOT NULL
   `u_status` int NOT NULL DEFAULT '1'
   PRIMARY KEY (`u_id`)

2. category – This table categorizes item. Whenever a registered user posts an item, he has to select the category for an item. Category selection would be mandatory for sellers and they would not be allowed to enter his choice for category.
   It will have the following data fields –

   `c_id` bigint NOT NULL AUTO_INCREMENT
   `c_name` varchar(128) NOT NULL
   `c_status` int NOT NULL
   PRIMARY KEY (`c_id`)

3. item –

   `i_id` bigint NOT NULL AUTO_INCREMENT
   `i_title` varchar(256) NOT NULL
   `i_desc` TEXT
   `i_price` double NOT NULL DEFAULT '0'
   `i_is_tradable` int NOT NULL DEFAULT '0'
   `i_u_id` bigint NOT NULL
   `i_created_ts` TIMESTAMP NULL DEFAULT NULL
   `i_updated_ts` TIMESTAMP NULL DEFAULT NULL
   `i_sold_ts` TIMESTAMP NULL DEFAULT NULL
   `i_status` int NOT NULL DEFAULT '1'
   `i_c_id` bigint NOT NULL DEFAULT '0'

PRIMARY KEY (`i_id`)
FOREIGN KEY (`i_u_id`) [u_id from user table]
FOREIGN KEY (`i_c_id`) [c_id from category table]

4. item_image

`ii_id` bigint NOT NULL AUTO_INCREMENT
`ii_image` blob
`ii_url` TEXT
`ii_i_id` bigint NOT NULL
`ii_status` int NOT NULL DEFAULT '1'
PRIMARY KEY (`ii_id`)
FOREIGN KEY (`ii_i_id`) [i_id from item table]

5. message

`m_id` bigint NOT NULL AUTO_INCREMENT
`m_text` TEXT
`m_sender_id` bigint NOT NULL
`m_receiver_id` bigint NOT NULL
`m_item_id` bigint NOT NULL
`m_item_id` bigint NOT NULL
`m_sent_ts` TIMESTAMP NULL DEFAULT NULL,
PRIMARY KEY (`m_id`)

3.2 Media Storage

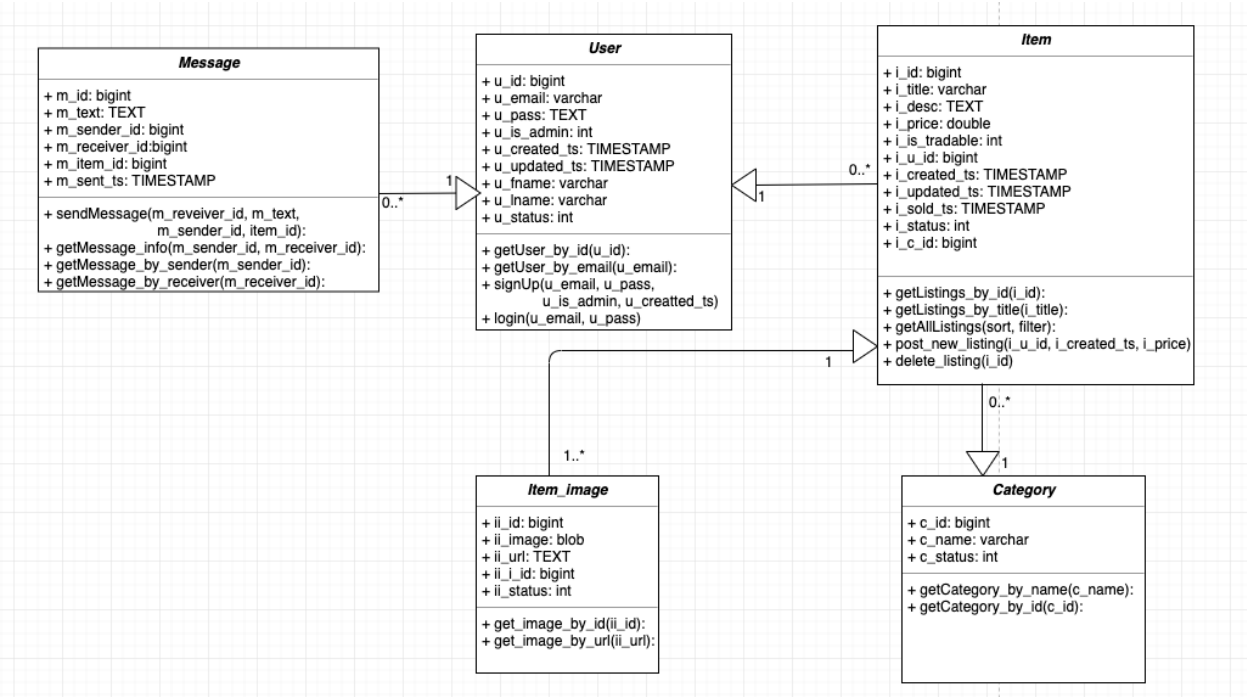
We shall use DB URLs for media storage. Standard images shall be stored locally. The server shall hold chat sessions locally up to some amount (1 GB tentatively). We shall create a thumbnail version of each item's image in order to speed up the load times of the front page.

3.3 Search/Filter Architecture and Implementation

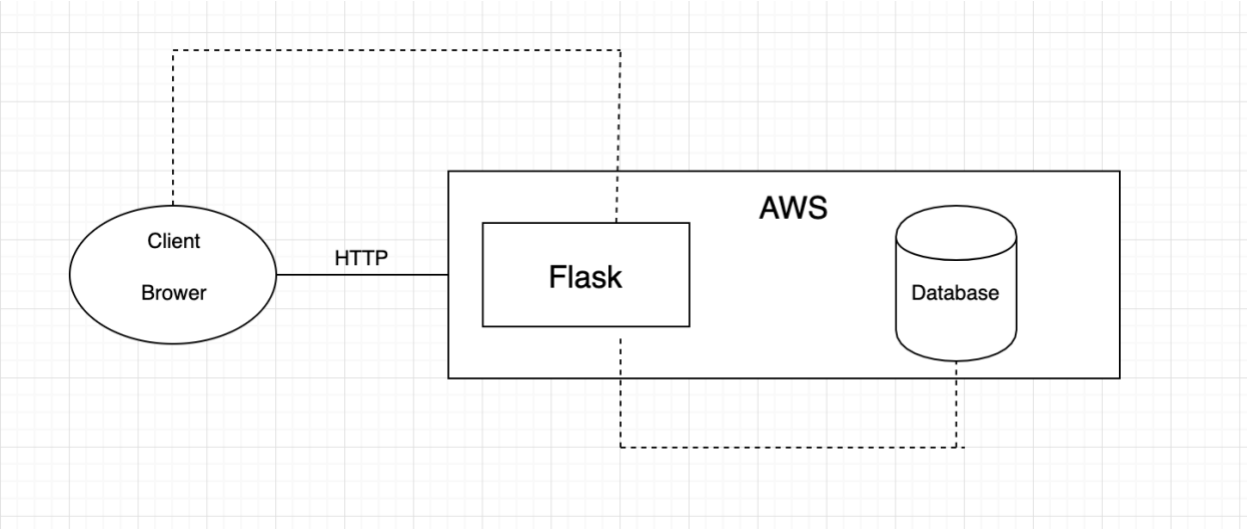
Our website will include a search box where a user can search by text typed in search field or by category. The user can search by some keywords such as item name, etc. which would then display the specific listings according to the keyword using %LIKE MYSQL query.

4. High Level UML Diagrams

4.1 UML Diagram



4.2 Component and Deployment Diagram



5 . Identify Actual Key Risks for Your Project at this Time

5.1 Skills Risks

Our team has development experience but have rarely designed such a buying/selling website completely. In this case, team members would study on their own and Vertical Prototype would serve as learning tool to all the team members. We would restrict the scope to priority 1 tasks. Developing this website would be an interesting and challenging environment for us to work end to end.

5.2 Schedule Risks

We have completed all the milestones till now with proper team meetings and communication between our team members. We would focus on all priority 1 tasks. Also, our team has set a Trello account for project management with all the team members. Hence, this part should not be challenging for us.

5.3 Technical Risks

The technical risk was associated with the git push /pull/ merge branches and the server deployment. However, we have identified a GitHub master who will handle the GitHub conflicts and also, a server master for deploying the working code remotely. Also, the vertical prototype will help as a learning tool for an end to end development and would resolve technical risks if any at an early stage of the development process.

5.4 Teamwork Risks

Our team works well together, and we have good communication between the backend and frontend team. There can be some risks if one of the team members is unavailable during crucial time and the dependency is high on that member. Our team lead takes care that everyone is on the same page.

5.5 Legal/Content Risks

If the site gets deployed publicly, the website shall prominently display the following text on all pages “SFSU- Software Engineering Project CSC 648-848, Fall 2019 - For Demonstration Only” in order to avoid any conflicts later.

6. Project management

We use Trello to help organize development tasks. All tasks are explained by the Team Lead and then delegated to the front-end engineers and back-end engineers. Front-end team lead takes the responsibility of completion of all front-end tasks and back-end team lead takes the responsibility of completion of all back-end tasks. For communication among team members, we use discord (a team communication platform). We have organized channels in discord as per the sub-teams like the front end, back end, documentation, database. For the documentation part, we use google docs so that everyone can work on their assigned tasks at one place.