# SW Engineering CSC648/848 Fall 2019

# Gator-Aid

# Section 02

# Team 09

(local)


TEAM LEAD

Nayan Pandey

npandey1@mail.sfsu.edu

Nikhil Mohan – Front-end Lead

George Pernov – Back-end Lead

Stephanie Sechrist – GitHub Master

Ali Nasralla

Jackie Huang


Milestone 2

10/18/2019

| Revision | Date | Description |
|---|---|---|
| - | 10/18/2019 | Initial Submission |
| | | |

# Table of Contents

## 1. Functional Requirements – Prioritized

**Priority 1**

**Unregistered Users**

1. Shall be able to browse items by category
2. Shall be able to view listings
3. Shall be able to register an account with an SFSU email account

**Registered Users (includes above)**

4. Shall be able to log in
5. Shall be able to post items for sale as listings
6. Shall be able to contact sellers
7. Shall be able to respond to messages from buyers

**Administrator (includes above)**

8. Shall be able to approve or reject listings

**Priority 2**

**Unregistered Users**

1. Shall be able to sort listings by price and post date
2. Shall be able to search items

**Registered Users (includes above)**

3. Shall be able to remove their own listings

**Administrator (includes above)**

4. Shall be able to remove listings

**Priority 3**

**Unregistered Users**

1. Shall be able to search for books by department and/or course number
2. Shall be able to edit their own listings

**Registered Users (includes above)**

3. Shall be able to flag listings as inappropriate

**Administrator (includes above)**

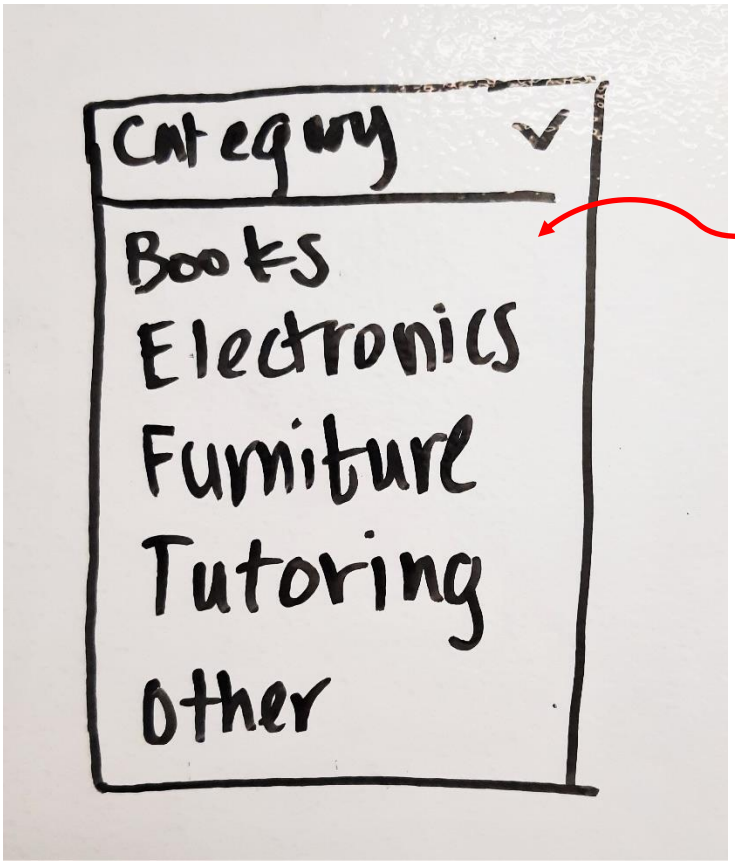
4. Shall be able to ban accounts.

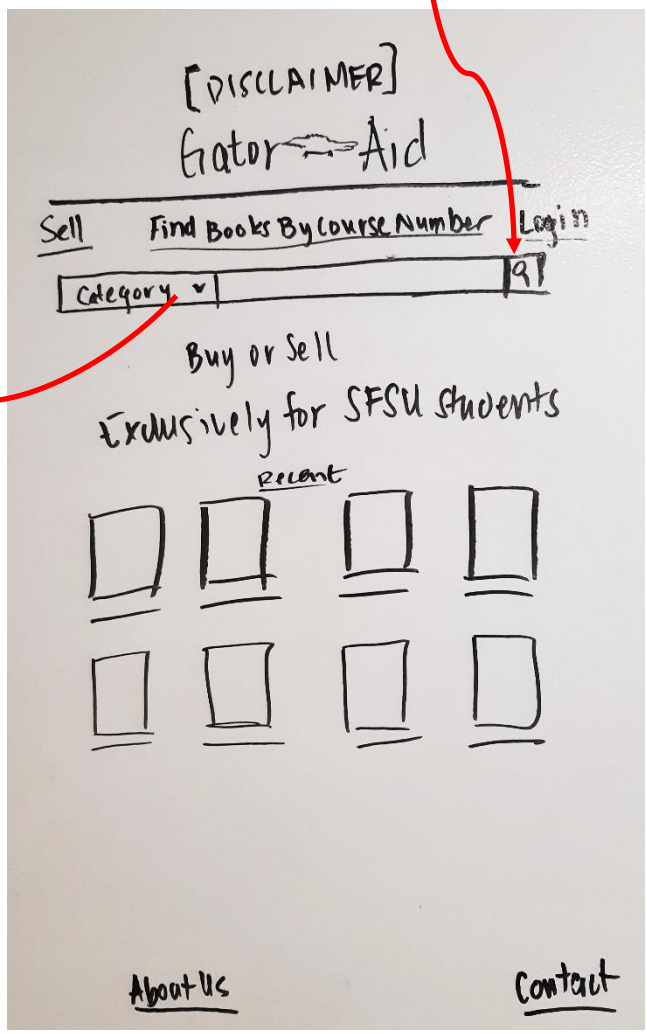
## 2. UI Mockups and Storyboards

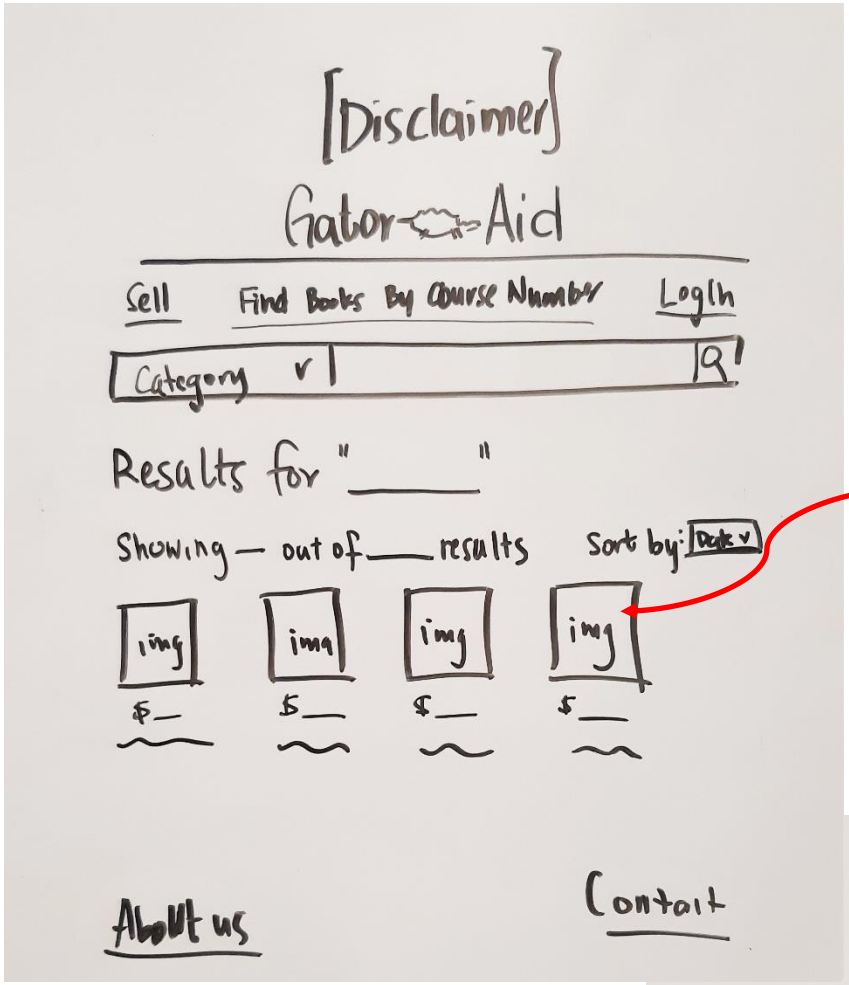
Use Case 1: Unregistered User

Alexa wants to buy a certain electronic item. She searches Gator-Aid after selecting the Electronics category, finds the item she wants, and clicks on the listing. She clicks Buy and is prompted to log in or register. Once she does, she can message the seller.

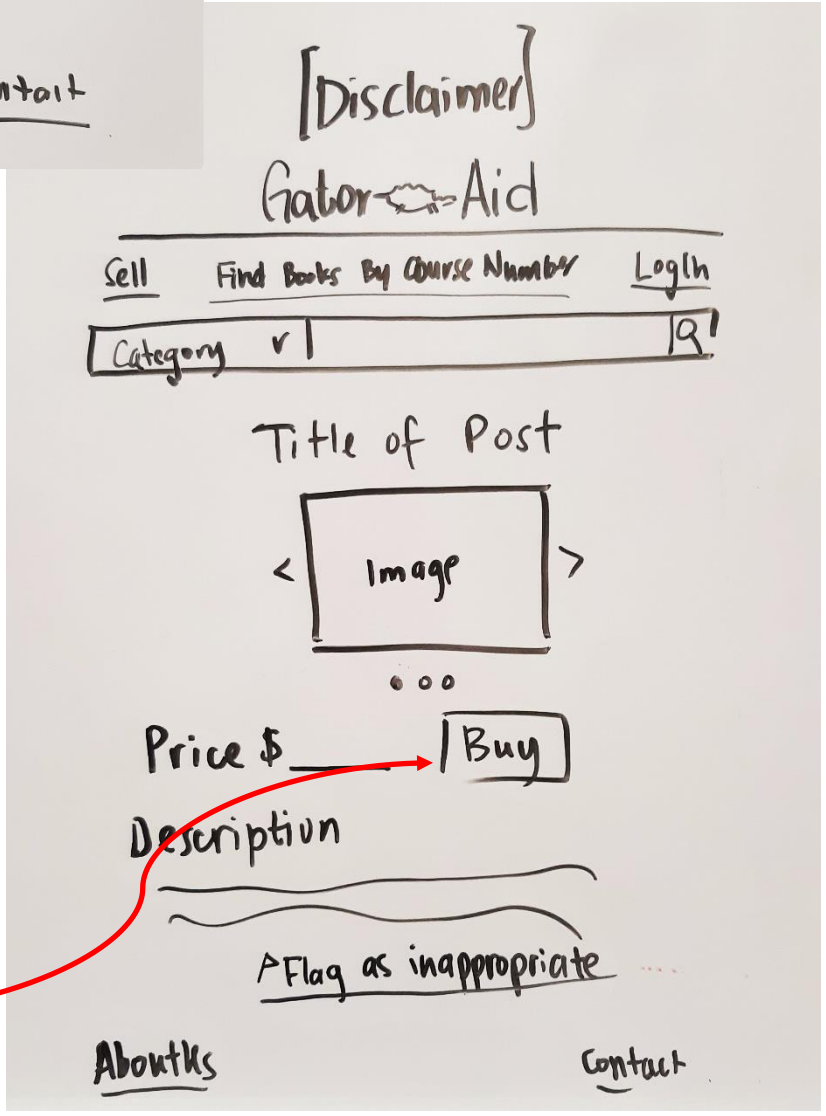
Alexa selects "Electronics" from drop-down Category list (seen left), types in her search query, and clicks the search icon.



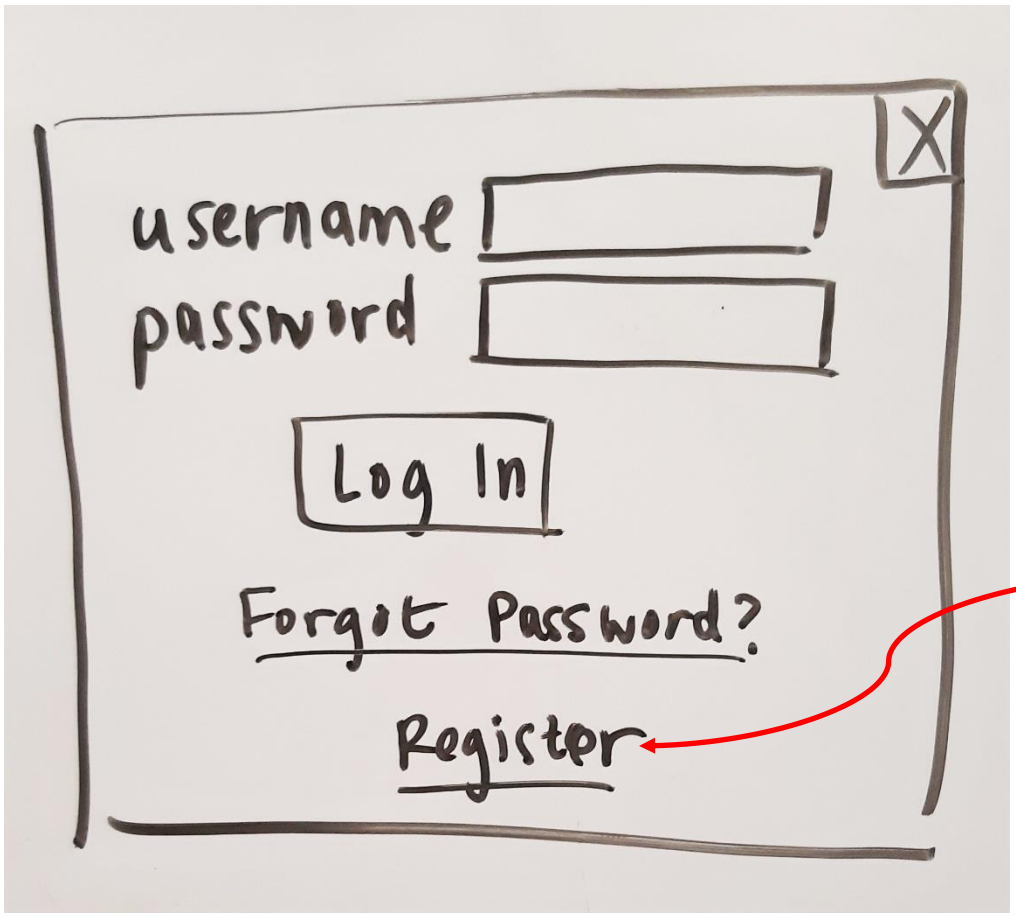



Clicks listing

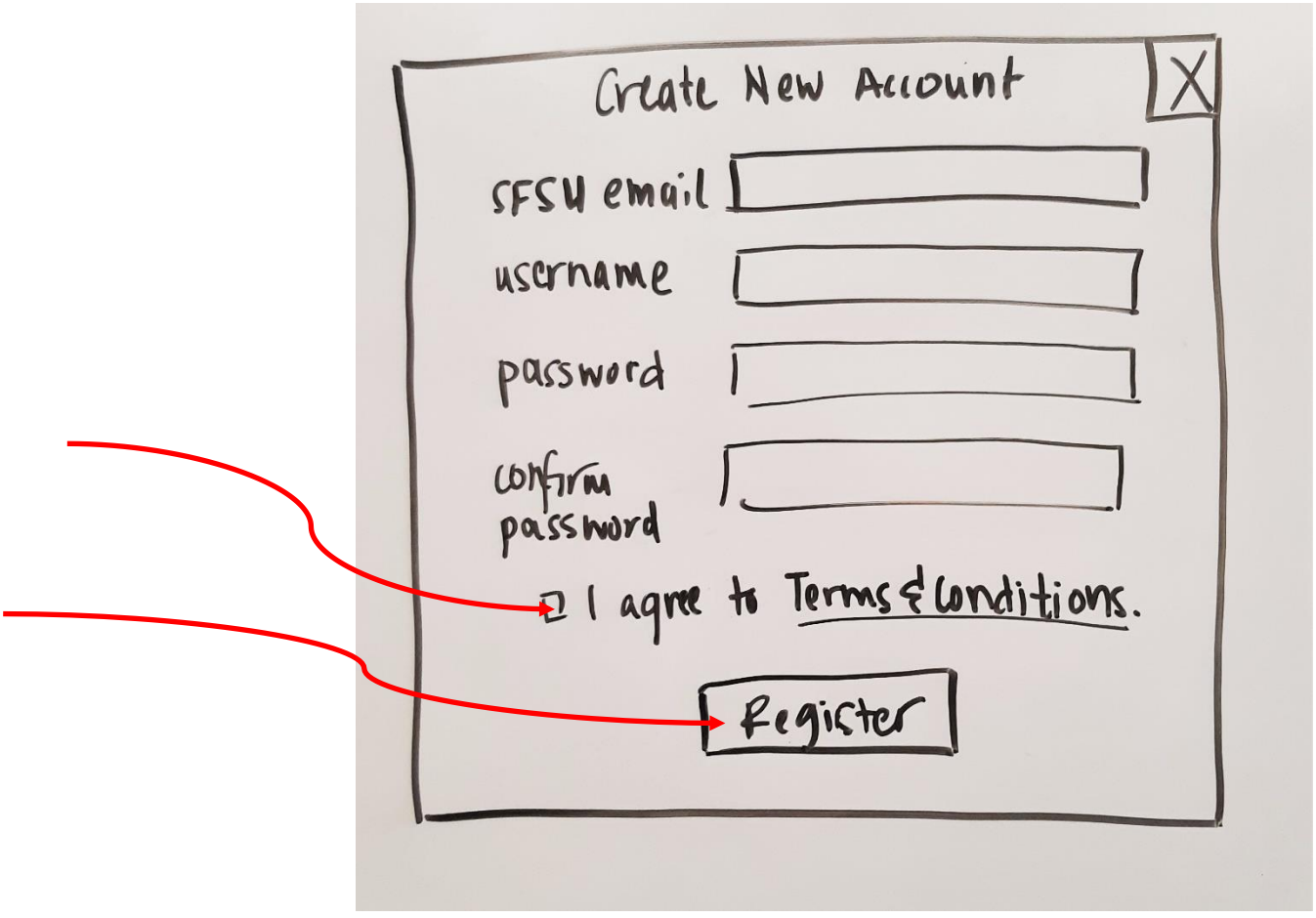


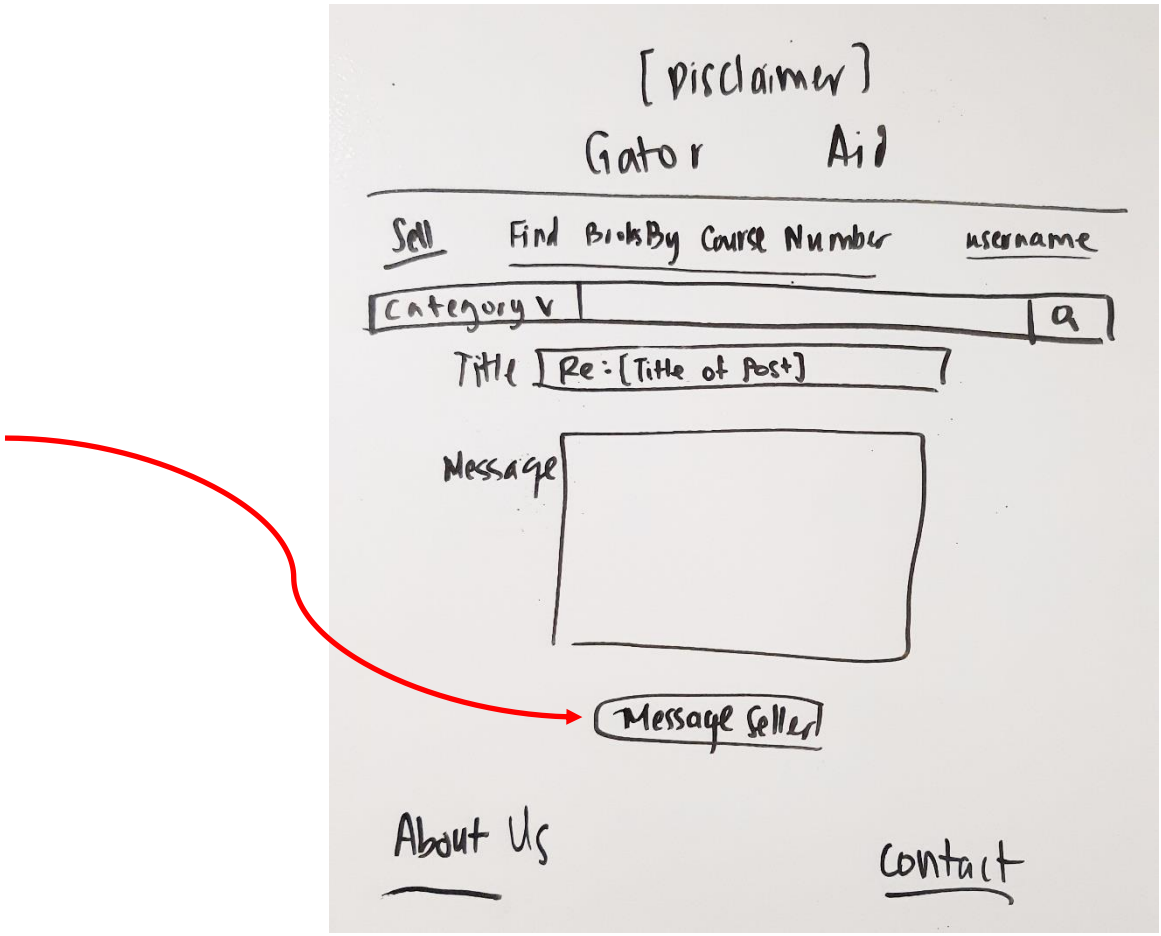

Clicks buy button

Log-in pop up



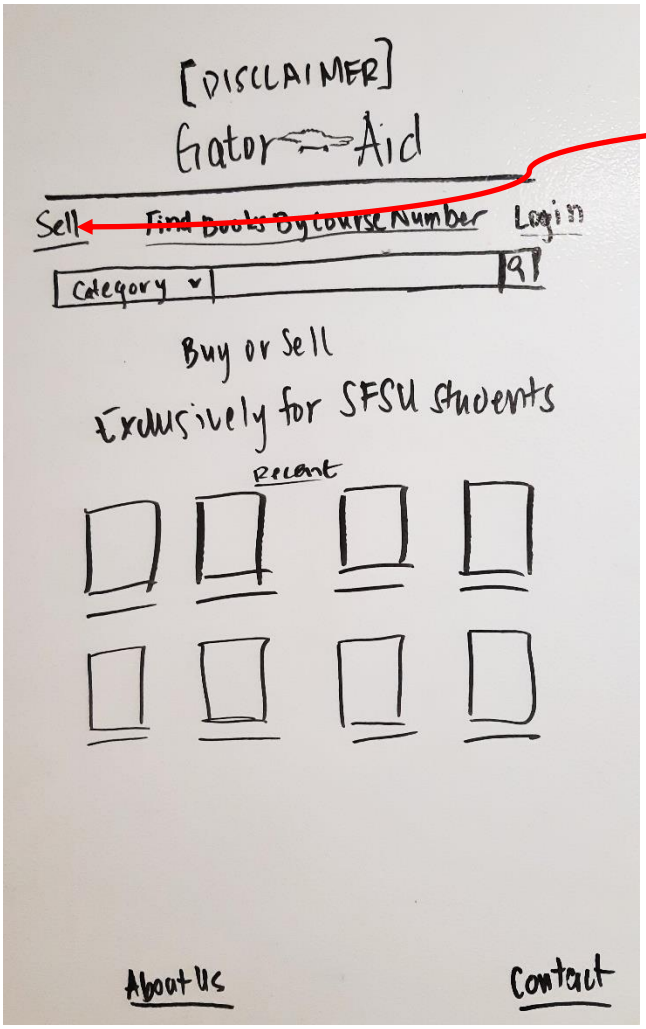

Registration pop up

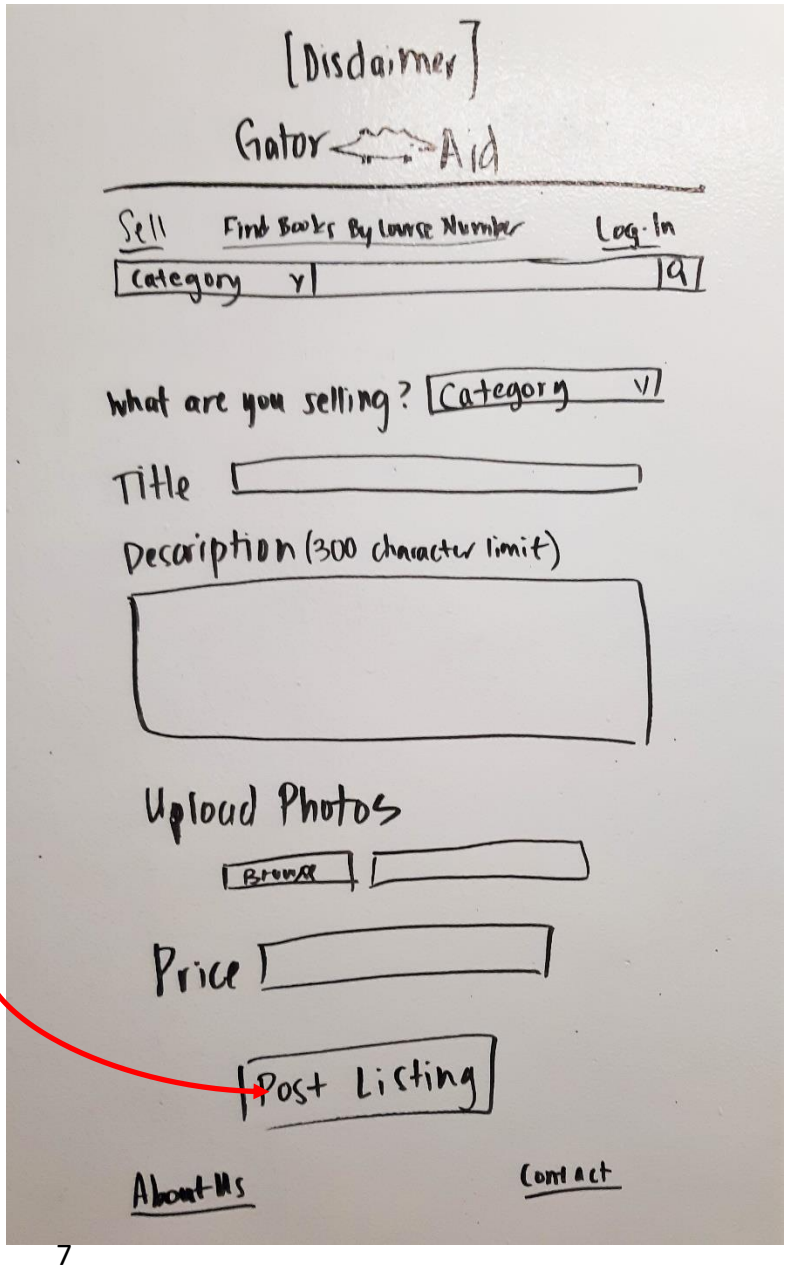



[Disclaimer]
Gator Aid
Sell
Find BooksBy Course Number
username
Category
Title
Re: [Title of Post]
Message
Message Seller
About Us
Contact

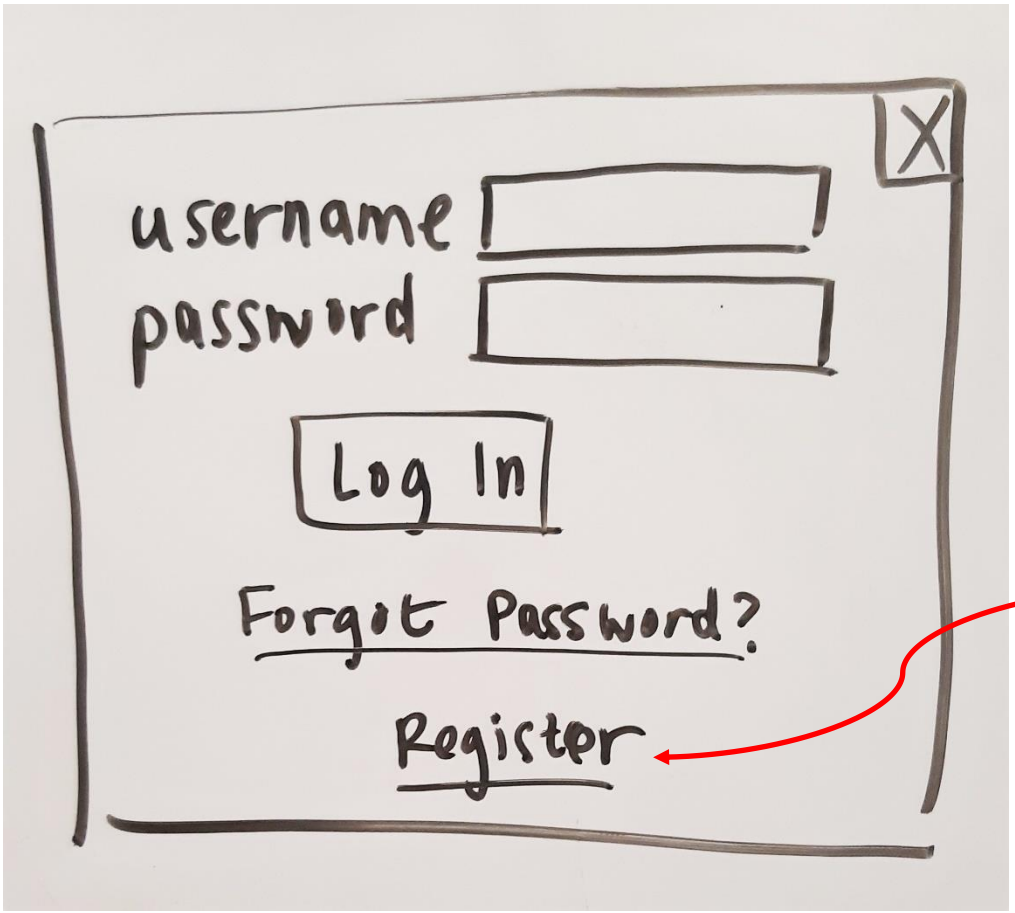
Use Case: User/Seller

Jennifer wants to sell her used textbooks. After clicking sell, she has to fill out the necessary fields and upload a photograph of the book. After clicking Post Listing, she must login or register. Her information is saved after doing so, and she clicks Post Listing again.

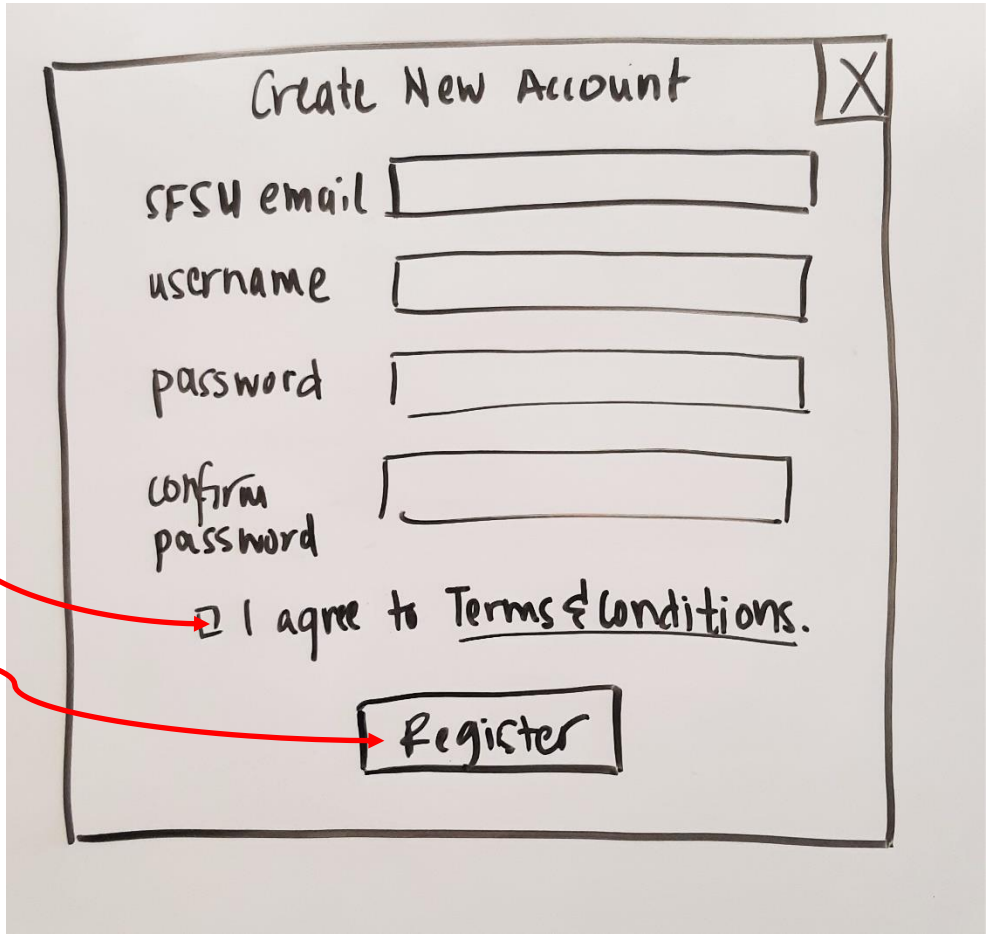


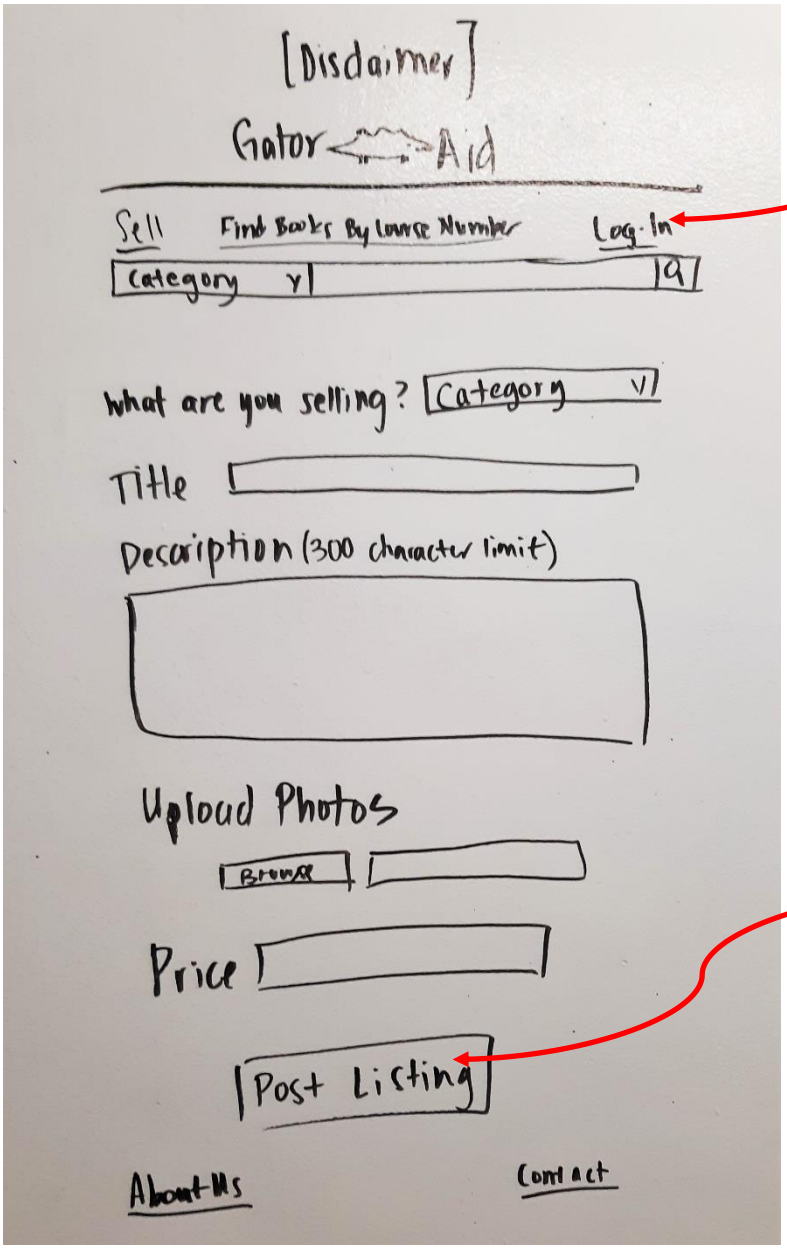



Log-in pop up




Registration pop up

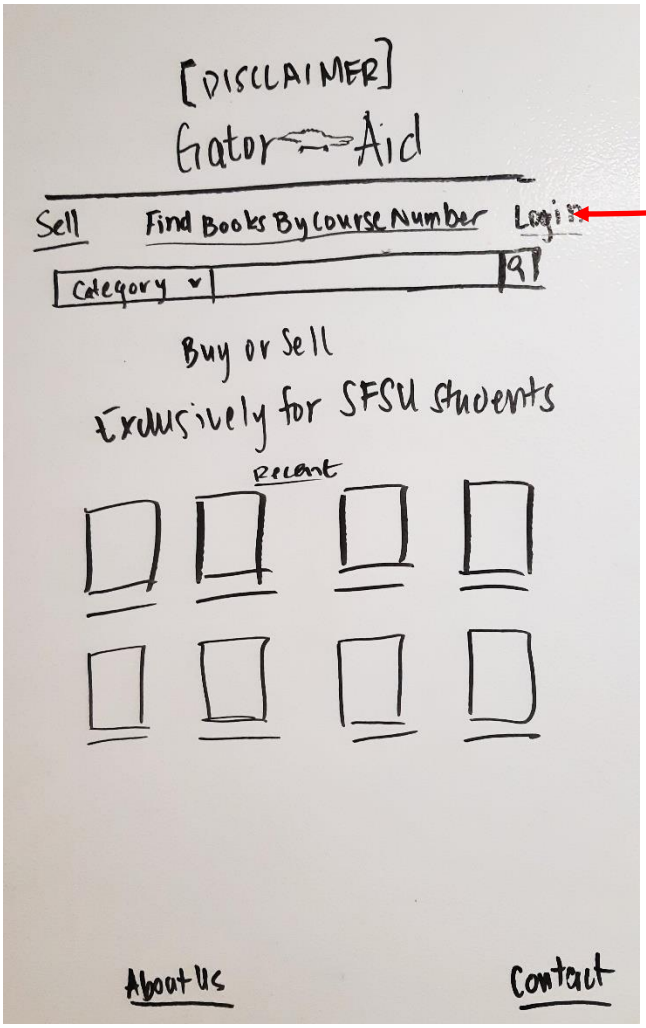





Would now show username

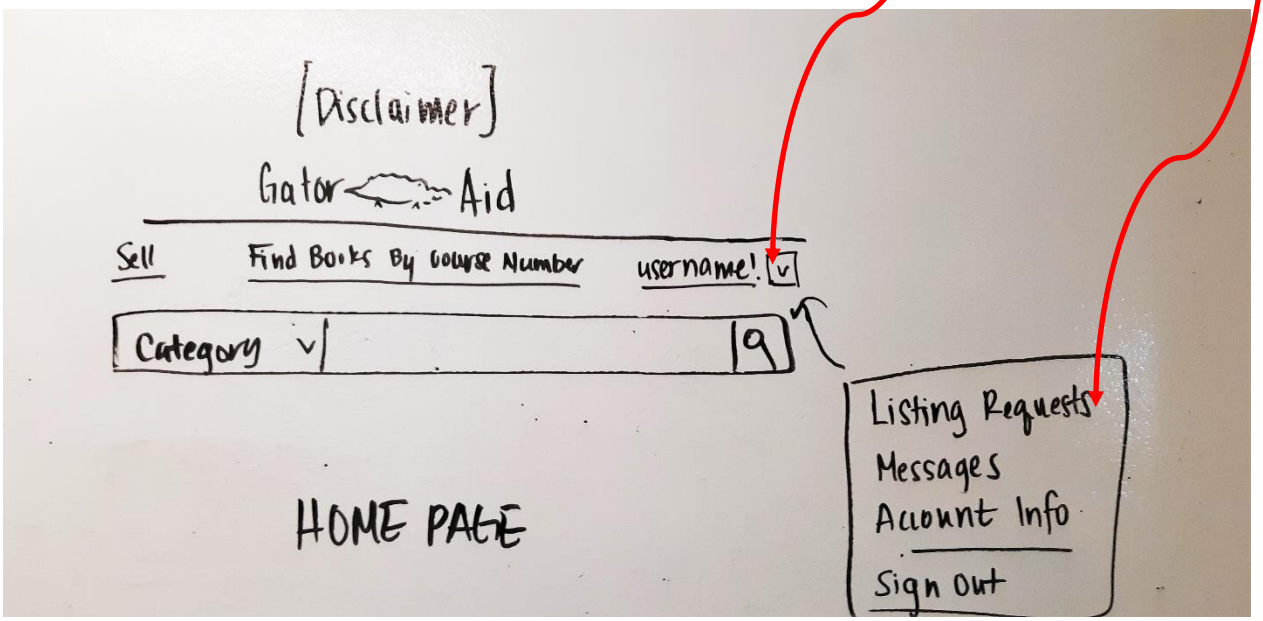
Use case: Administrator

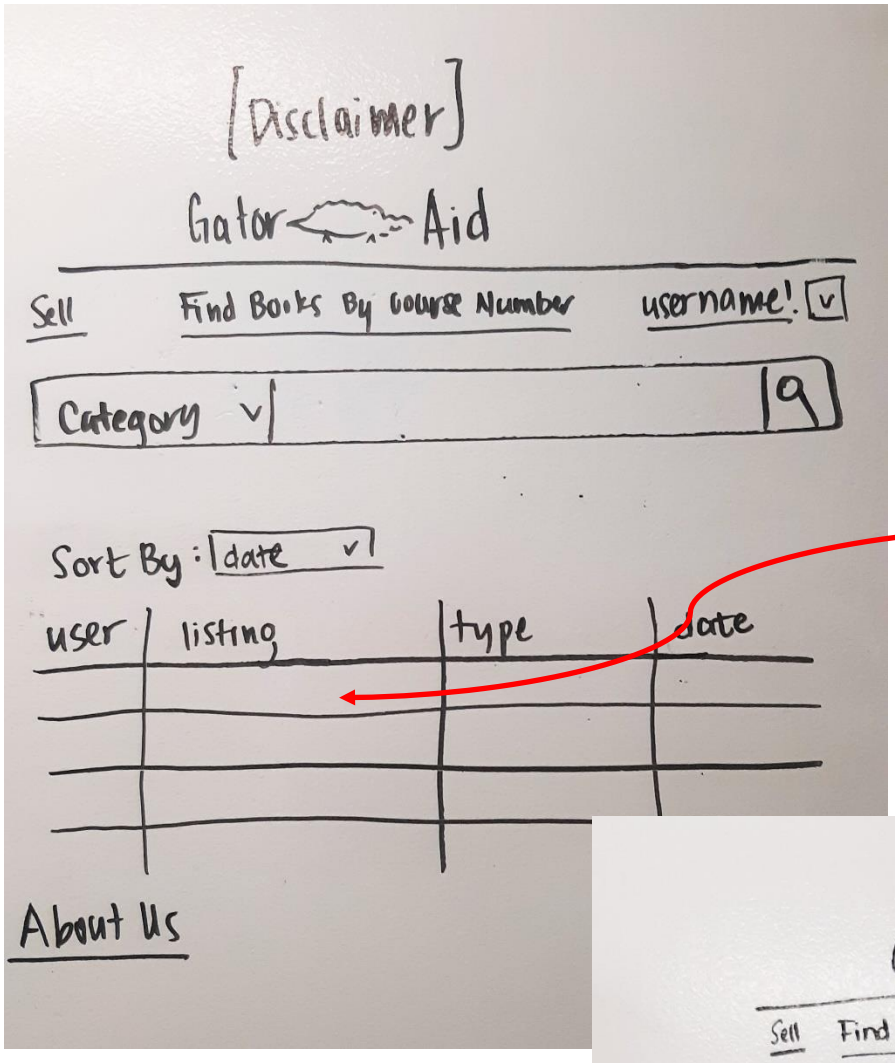
John logs in and sees that he has a notification (!) about new listings. On the listing requests page, he sees listings waiting to be approved. He clicks the titles to view each listing and either approves or rejects the listing. After doing so, he is returned to the Listing Requests page.



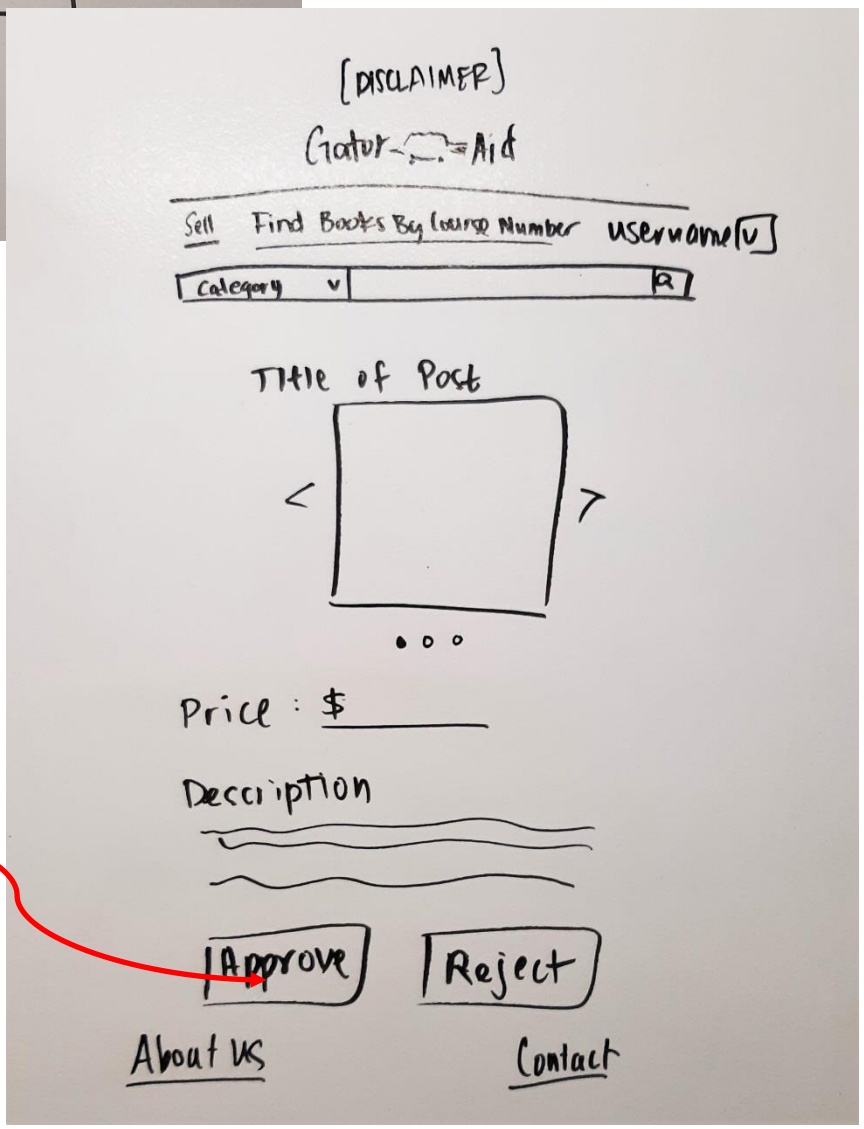

Clicks pull-down menu to access Listing Requests

Clicks Listing Requests






Clicks title of listing



# 3. High Level Architecture & Database Organization

## 3.1 Database Organization

Users:

| Name | Data Type | Default View | NULL |
|---|---|---|---|
| **user_id** | uuid | Auto Increment | No |
| **first_name** | varchar(30) | | No |
| **last_name** | varchar(30) | | No |
| **email** | varchar(30) | | No |
| **password** | char(32) | | No |

Listings:

| Name | Data Type | Default View | NULL |
|---|---|---|---|
| **listing_id** | uuid | Auto increment | No |
| **user_id** | uuid | | No |
| **date_created** | date | Current Timestamp | No |
| **description** | text | | Yes |
| **price** | decimal(7,2) | | Yes |
| **image** | Varchar(255) | NULL | Yes |
| **confirm_listing** | boolean | False | No |
| **title** | Text | | No |

Messaging:

| Name | Data Type | Default Value | NULL |
|---|---|---|---|
| **message_id** | uuid | Auto Increment | No |
| **sending_user** | uuid | | No |
| **listing_id** | uuid | | No |
| **message** | varchar(255) | | No |

## 3.2 Media Storage

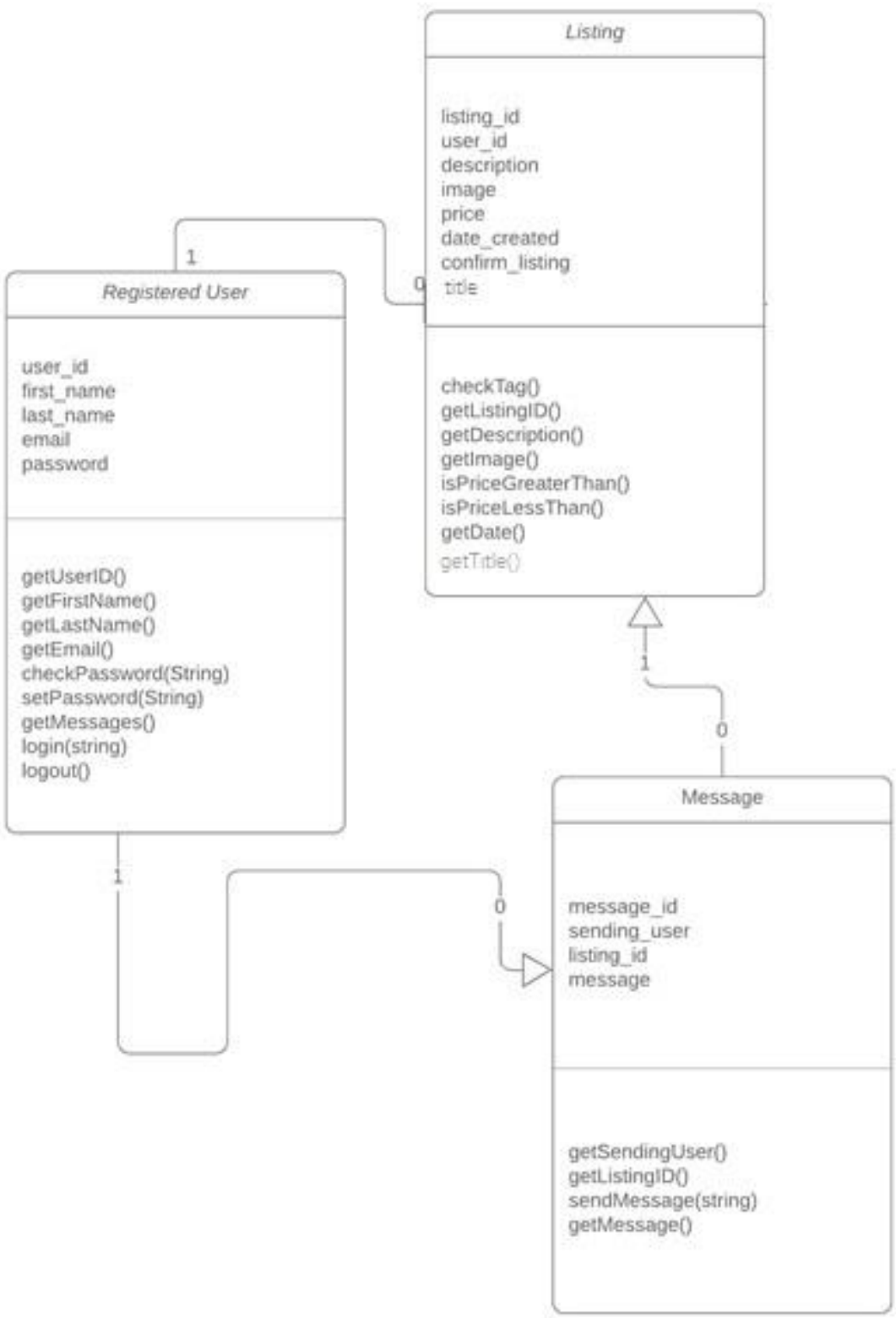
All media will be kept in a file system format with the database containing the file path names pointing to all the media being used.

## 3.3 Search Architecture and Implementation

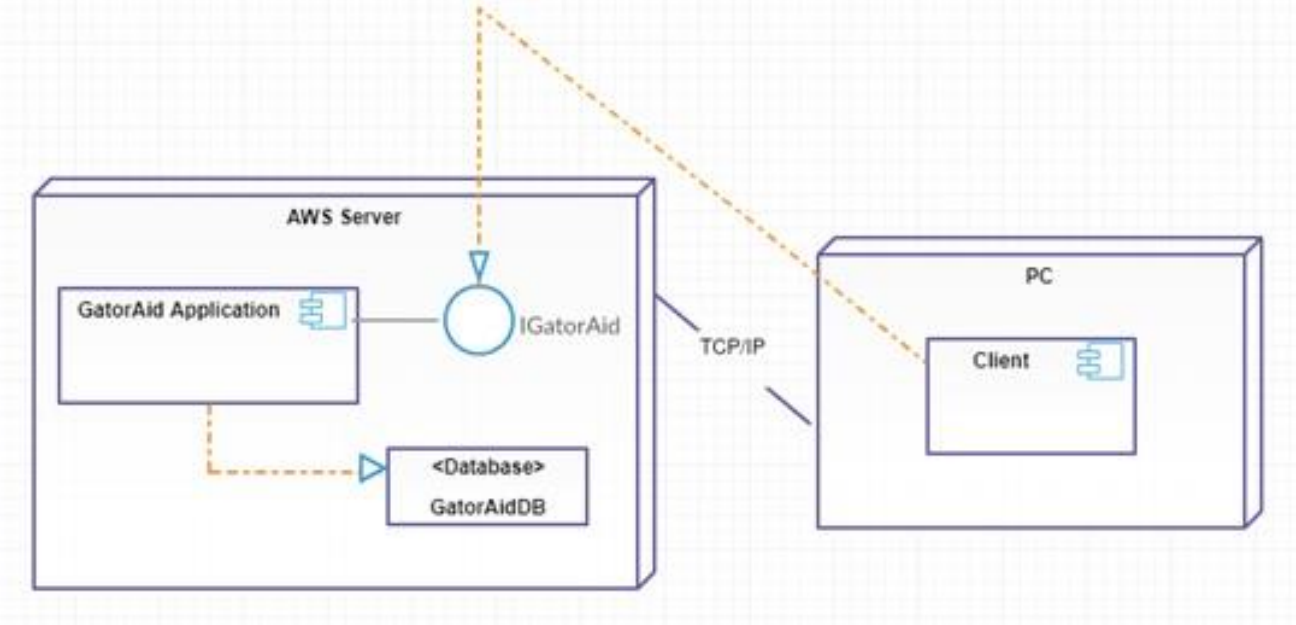
- If the user does not input anything in the search bar or choose a category, the display page will show a variety of listings from all categories, sorted by the date created.
- If the user chooses a category but does not input anything in the search bar, the display page will show a variety of listings from the specified category, sorted by the date created.
- If the user enters a keyword in the search bar, listings whose title or description contains the keyword will be listed, sorted by the date created. This is implemented using the %LIKE operator.

# 4. High Level UML Diagrams

## 4.1 High-Level UML Class Diagram



## 4.2 UML Component and Deployment Diagram



# 5. Key Risks

## 5.1 Skill Risks

A major skill risk that we have is learning the new technologies that we have chosen for this project. We each have a variant degree of knowledge on the technologies that we have chosen. This can cause implications when coding is a big part of the project. However, since there are hundreds of helpful documentation and guides on these specs that we have chosen, it is very easy to get an understanding of how it works. Furthermore, since each group member has different knowledge levels, we are able to help each other when implementing the application.

## 5.2 Schedule Risks

All of us meeting at the same time is a struggle due to our differing school and work schedules. Often only 2 or 3 teammates at a time can meet at school. It is rare for all of the teammates to be there. Our solution to this risk is to come up with a weekly meeting where everyone can participate. If we absolutely cannot do this, we are able to have a group call using Discord to discuss the project as well.

## 5.3 Technical Risks

A technical risk we face is conforming to coding standards and best practices. This is a challenge because we all have our unique coding style, and we are also learning new technologies, so we may not yet know the best practices for each language we are using. We can avoid this risk by setting standards on how we write our code.

Another technical risk we have encountered is that the website that we are creating is showing different results when running it on the server host versus running it on the localhost. This can be easily solved by asking the CTO why this is happening.

## 5.4 Teamwork Risks

One of the main teamwork risks that we have is keeping track of the progress for each specific group member task. There is also a possibility that two or more members might be doing the same task while no one is doing the other task. In order to solve this risk, we can start using Trello. Trello is a useful software to show explicitly which member has which task and they are able to add info on the task to show what their progress is.

## 5.5 Legal/Content Risks

As of this moment, we believe that we are not in any legal/content risk when developing this project. There could be risks of a client losing their credit information when buying items, but as of now, our site does not accept any kind of payment.  Also, the site has disclaimers that it is a class project to avoid any liability.

## 6. Project Management

We are managing our team tasks by meeting up as much as we can either in person or communicating through the messaging application Discord.

When we meet up in person, we work on a majority of the tasks together and are able to get some of them done fully. This can range from proofreading to brainstorming to getting everybody up to speed on the current milestone of the project. We also ask for information on concepts we are not clear about. Meeting together as a team allows us to complete tasks together efficiently and well.

We communicated through Discord from the very beginning, as we feel Discord is really convenient due to improved general chat and has an effective voice/video chat capability. Also, we have assigned specific channels in Discord for front-end and back-end teams, where each team can work independently. Discord also allows us to create multiple “voice channels” on the same server, so we can communicate effectively with each other when working outside of class. We also use Discord to note really essential tasks or jobs that must get done ASAP. In the future when the project size expands, we will also use Trello to manage tasks. Discord and Trello together allow us to communicate quickly and effectively to handle issues and work on tasks as a team.

For the rest of the semester, we intend to meet up in person and trying to assign as much as we can there. We will also utilize GitHub from Milestone 3 and on. The GitHub master will be working on managing all the conflicts if there are any. For the rest of the project, we will follow our pattern of meeting up, communicating with Discord on our own time, managing tasks with Trello, and utilizing GitHub to work together as a team towards our goal of creating a great software product.