SW Engineering CSC648/848 Fall 2020

GatorMart

Team 6

Lothar Narins Team Lead

Aaron Colmenares Backend Lead

Allyson Leung Frontend Lead

Wilson Young GitHub Master

Saloni Bhatia Backend

Kevin Nunura Frontend

lnarins@mail.sfsu.edu

Milestone 2

October 1, 2020

Date Submitted: 10/15/2020

Date Revised: TBD

Table of Contents

Executive Summary

GatorMart is a sleek and functional website where SFSU students, staff, and faculty can go to buy, sell, or exchange select goods and services. By keeping a narrow focus and restricting registration to SFSU students and faculty, we are able to offer more streamlined services to our target audience, as well as greater security through SFSU email verification, which sets us apart from our more generalized competitors. The site allows buyers to contact sellers through a built-in messaging system. Users can browse through categories or search with keywords or for articles related to specific classes, and if they don't find what they're looking for, they may post a request, so that potential sellers can see what is in demand. Of course, all postings must be approved by an admin, so users are protected from anything untoward. We allow our users great flexibility in how they handle purchases by having them use their own payment method rather than through our website.

We as a team are very excited to bring these features to the potential users of GatorMart. Our team consists of aspiring Computer Science majors in the CSC 648 class. We are all very enthusiastic about bringing you a functional and reliable means to sell or buy items from other SFSU members. Our motivation behind this project is to help fellow SFSU members safely sell or buy their goods while also providing free resources to those who need it. We will strive to present the best possible product to help our community.

Main Data Items and Entities

GatorMart users are stored in the '**users**' table. They can be:

1. ***Anonymous users:*** Users that are not logged in. These have no entry in the '**users**' table. Their data is stored in cookies.
2. ***Active users:*** Users that are logged in. These get an entry in the '**sessions**' table when they log in.
3. ***Administrators:*** Active users with special privileges. Their entry in the '**users**' table has a usertype of admin.

GatorMart posts are stored in the '**posts**' table. These will be hidden from users until approved by an administrator. They can be:

4. ***For-sale:*** A post created by a registered user about an item that they want to sell. The user provides a description, image, and price of the item. There is an option for other users to message the poster that they are interested in buying the item.
5. ***Requests:*** A post created by a registered user in search of a specific item. Other users with access to the original request can fulfill this sale.
6. ***Class Resource:*** A post created by a registered user that offers supplemental tools and resources specific to a course ID.

Other items:

7. ***Messaging:*** A one-way system for users to communicate to another user they are interested in buying an item and provide the seller with their contact information. These are kept track of in the '**messages**' table.
8. ***Flag:*** A button functionality to flag posts not complying with the website terms and conditions. Administrators will be able to review flagged posts for inappropriate content. These will be stored in the '**flags**' table.

Low-priority items:

9. ***Wishlist:*** A compiled list of posts that an anonymous or active user wishes to save for future viewing. This will be stored in cookies.
10. ***Gator Review:*** Our website's user review system set in place for the safety of our clients. This system includes personal statements written by other users about a specific user as well as a five star rating system called Gator Stars. This is a priority 3 item which would be stored in the database in a '**reviews**' table (not shown).

Functional Requirements

Priority 1

1. **Anonymous Users**

   1.1 Shall be able to browse and search for items for sale.

   1.2 Shall be able to log in.

   1.3 Shall be able to register ONLY with an email ending in sfsu.edu.

   1.4 Shall be able to browse the website utilizing categorized filters.

2. **Active Users**

   2.1 Shall inherit all functionality allowed for anonymous users.

   2.2 Shall be able to post items for sale/share or request items, and such posts will be reviewed by the website administrator.

   2.3 Shall be able to privately message a user for an item they are interested in.

   2.4 Shall be able to log out of their account.

3. **Administrator**

   3.1 Shall inherit all functionality allowed for active users.

   3.2 Shall be required to use MySQLWorkbench to manage the website.

   3.3 Shall be able to approve or deny post submissions from users.

Priority 2

2. **Active Users**

   2.5 Shall be able to flag posts not complying with the website terms and conditions.

3. **Administrator**

   3.4 Shall be able to delete posts and users that go against the website policy and procedures.

   3.5 Shall be able to review flagged posts.

Priority 3

1. **Anonymous Users**

   1.5 Shall be able to add posts to their wishlist.

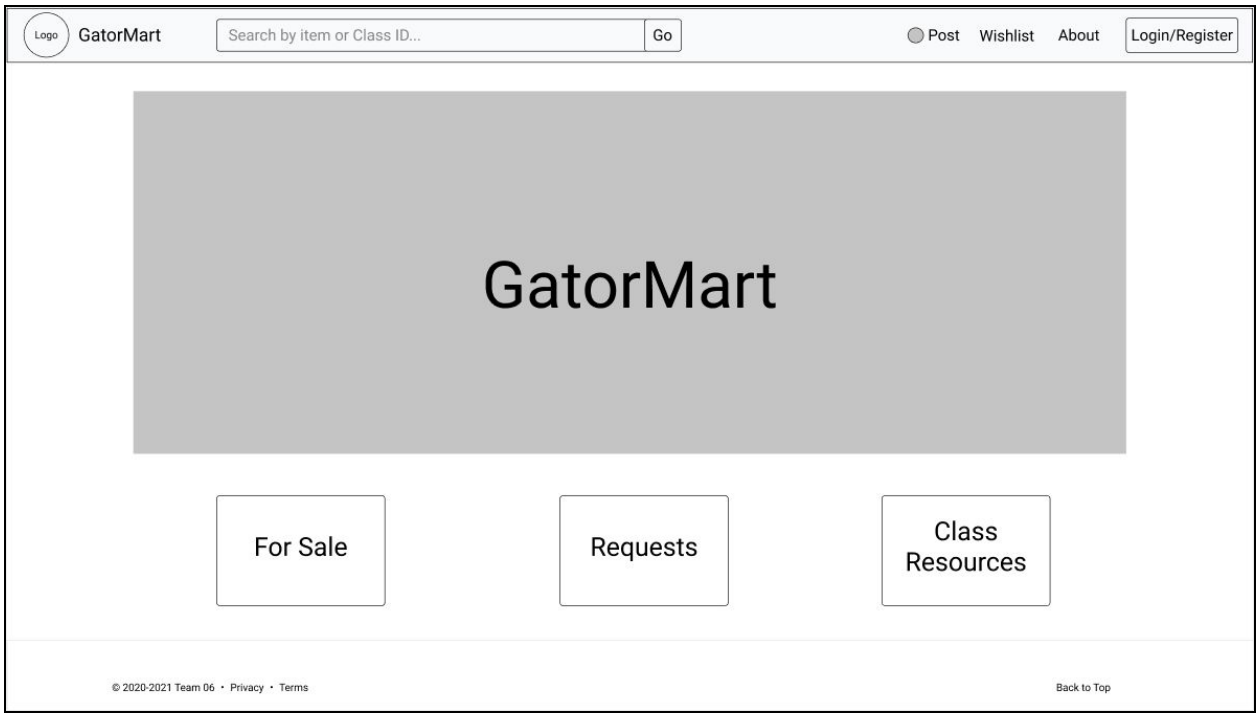
2. **Active Users**

   2.6 Shall be able to rate a user utilizing the Gator Review system.

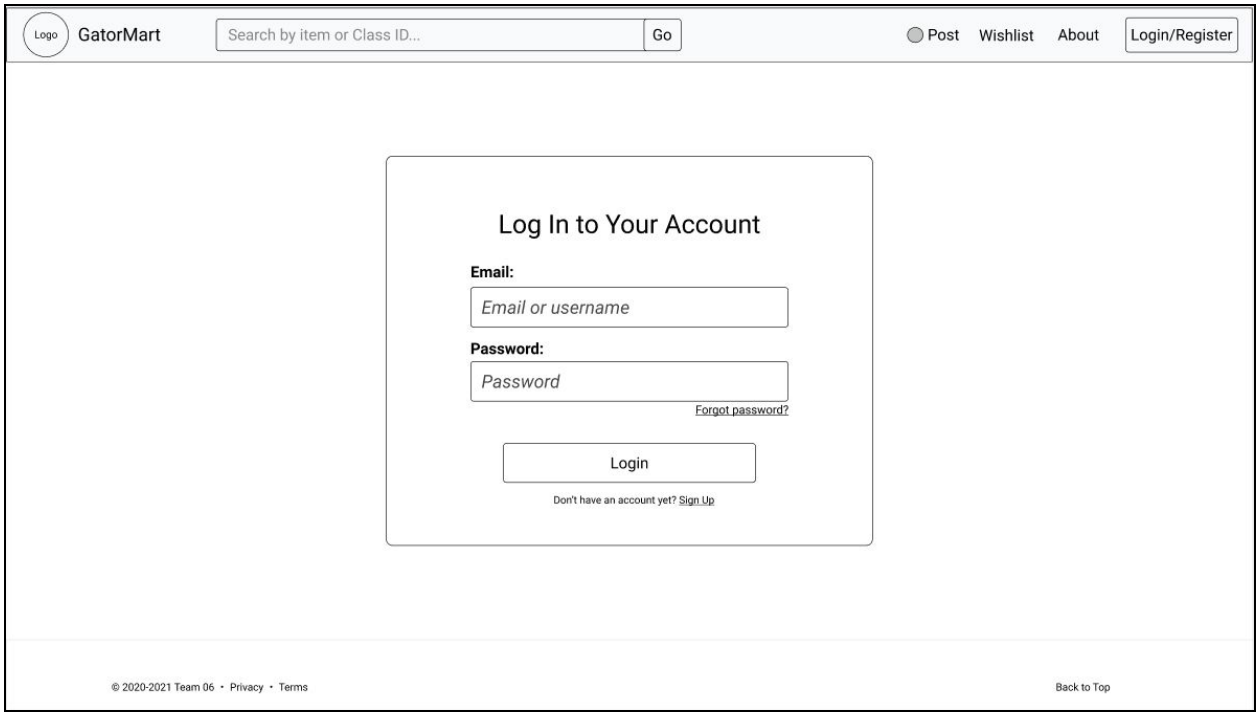
UI Mockups and Storyboards

Use Case 1: Register, Login, and Post
Dr. Mosk has just finished a faculty meeting discussing the emotional and financial impact on current San Francisco State (SFSU) Students as a result of the global pandemic and seasonal local wildfires. As a professor that cares for the wellbeing of his students, Dr. Mosk was able to get a textbook that was being offered for free by a reputable academic institution online approved by the Computer Engineering department for one of his courses. Knowing that the class occupancy was low due to textbook costs, Dr. Mosk decided to motivate future students to enroll in the course by making his cost-free materials known to the university community through GatorMart. He logs in to our website on his laptop and proceeds to create a post in GatorMart specific to his class ID but he is redirected to register or log in with an active account before doing so. Upon successfully registering with his SFSU email and logging in, he creates a post with the appropriate description that directs SFSU students to download the free course textbook from the reputable educational website. Dr. Mosk fills out all of the fields in his post submission form and is able to submit his post for admin approval.

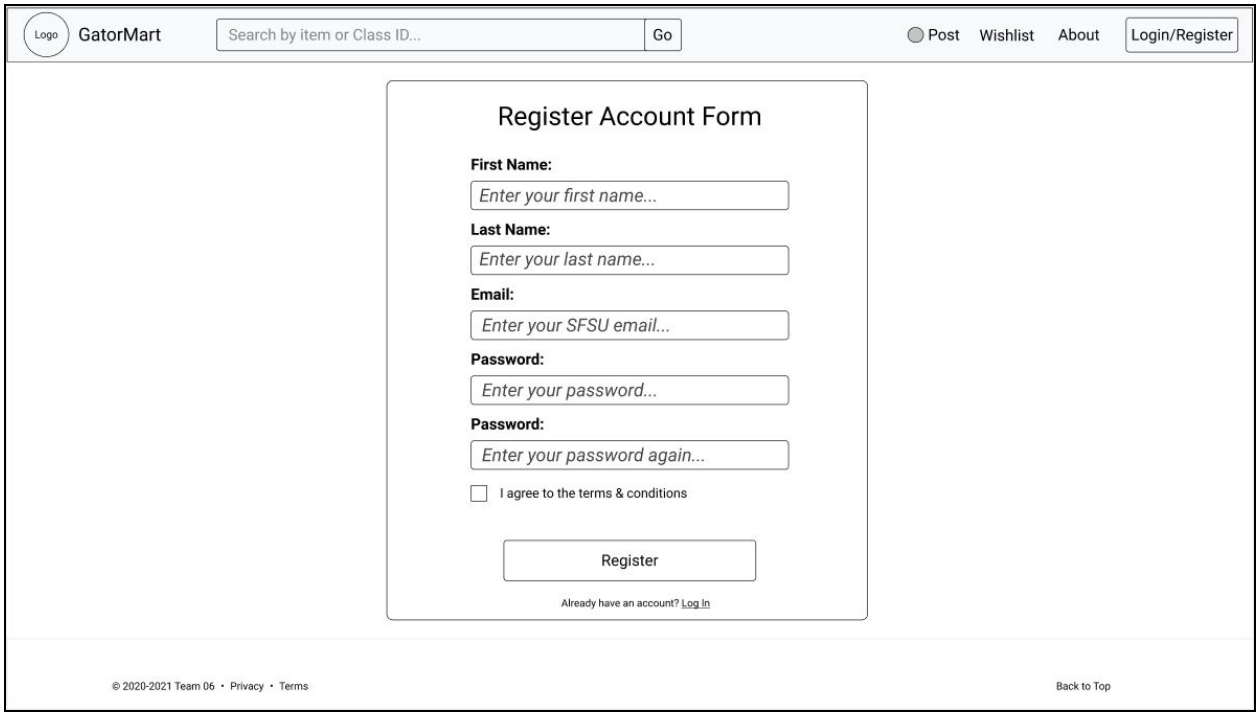
Landing/Home Page: Dr. Mosk travels to the GatorMart home page.



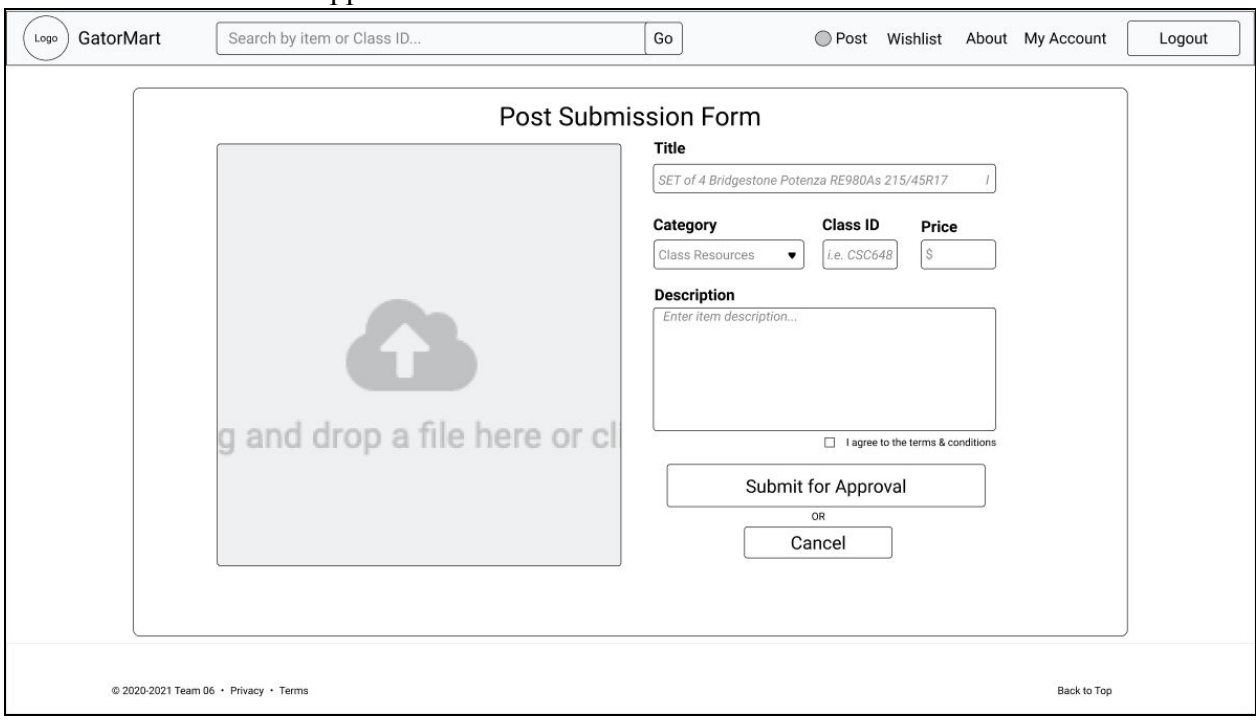
Login Page: Dr. Mosk attempts to create a post and is redirected to register or log in.



Registration Page: Dr. Mosk proceeds to register an account with his SFSU email.

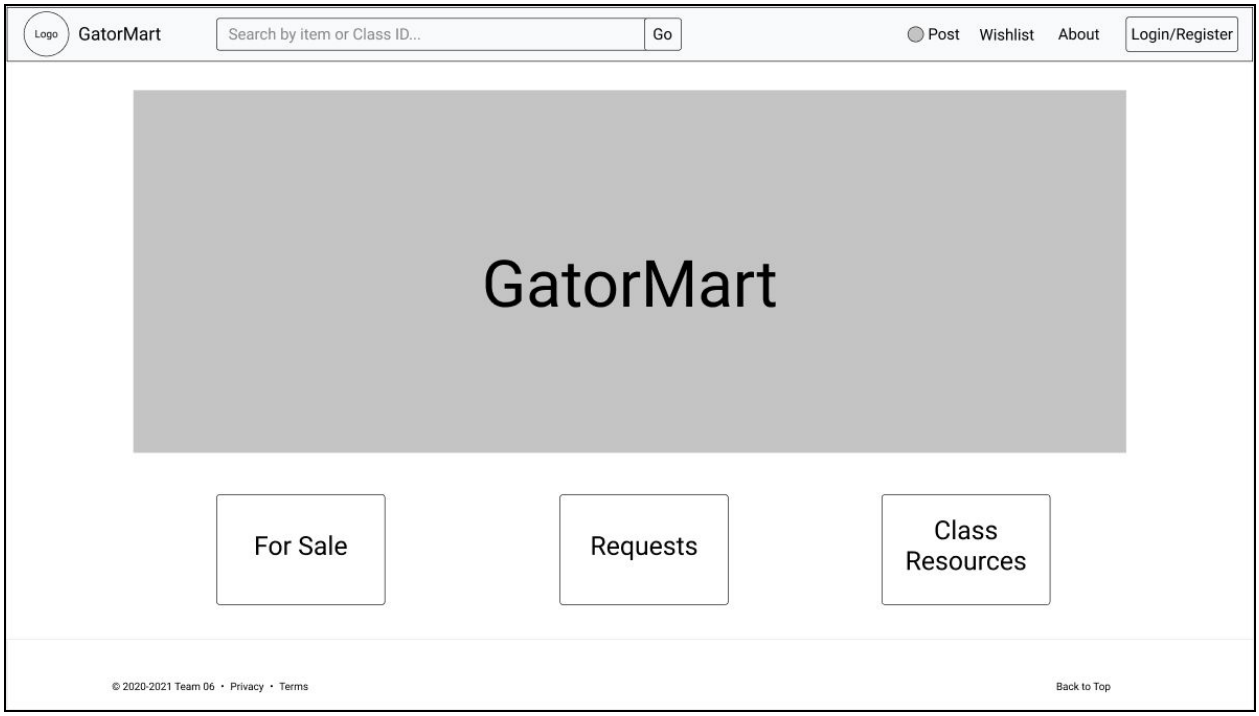


Posts Submission Page: Dr. Mosk clicks “Post” in the navbar then creates and submits a “class resource” for site admin approval.

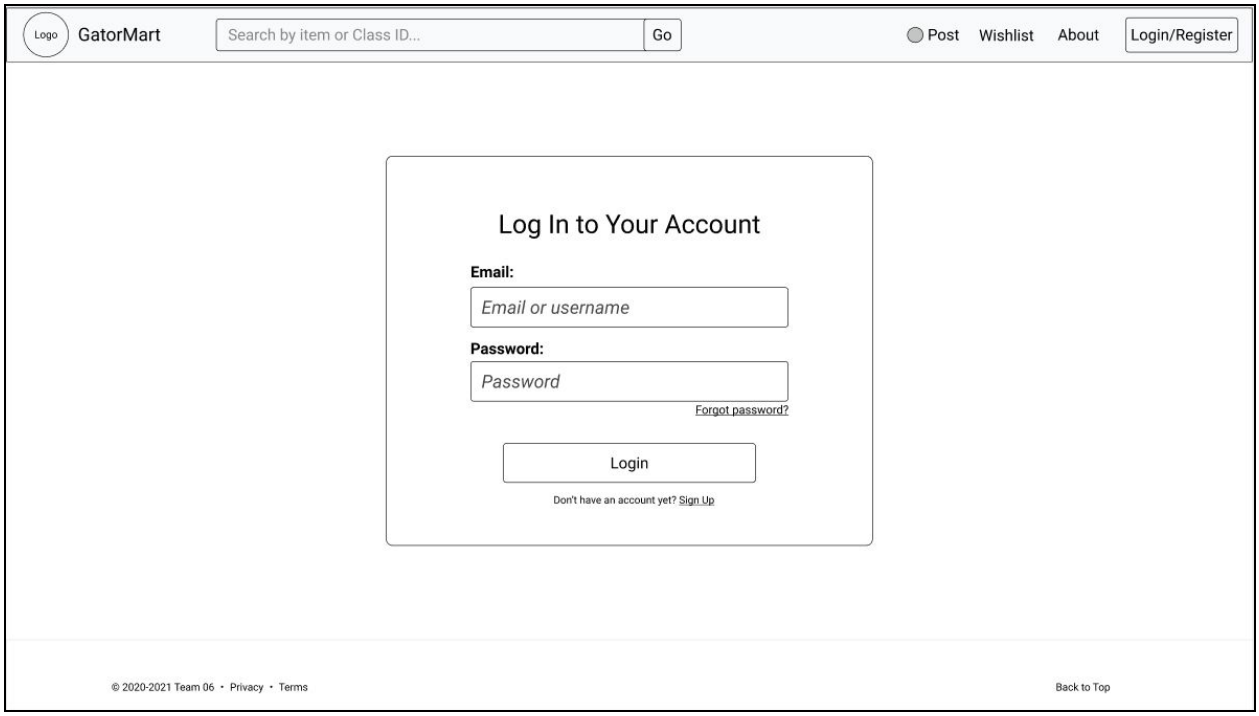


Use Case 2: Buying
Ricardo has just woken up late for his first day as a freshman at San Francisco State University (SFSU). He is excited for this day, but is also exhausted from finishing up an overnight shift the day prior at a local coffee shop and is now running late for school. As Ricardo is traveling on BART to SFSU, he realizes that he forgot to purchase the required textbook for his first class. He immediately goes to the GatorMart mobile site, logs in with his active account, and seamlessly searches for the title of the book he needs. He notices that there is one “for sale” post created by a reliable active user offering the textbook he required. Ricardo quickly utilizes the simple one-way messaging system from our site to inform the seller of his interest while providing his contact information for their scheduling convenience. Upon successful contact from the seller, both users set up a time to meet up at the university’s J. Paul Leonard Library which is on his way to his first class. Ricardo meets his seller at their public appointed place and time and safely purchases his textbook by paying his seller through an external money exchange application.

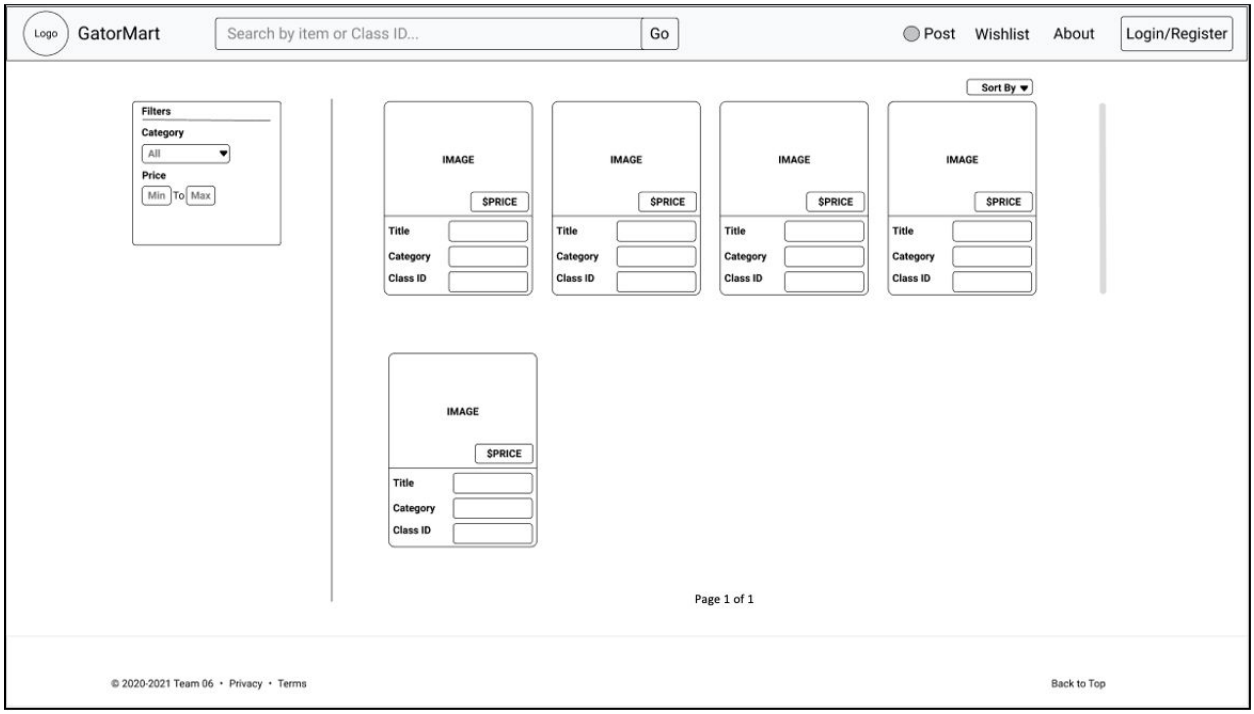
Landing/Home Page: Ricardo travels to the GatorMart home page.



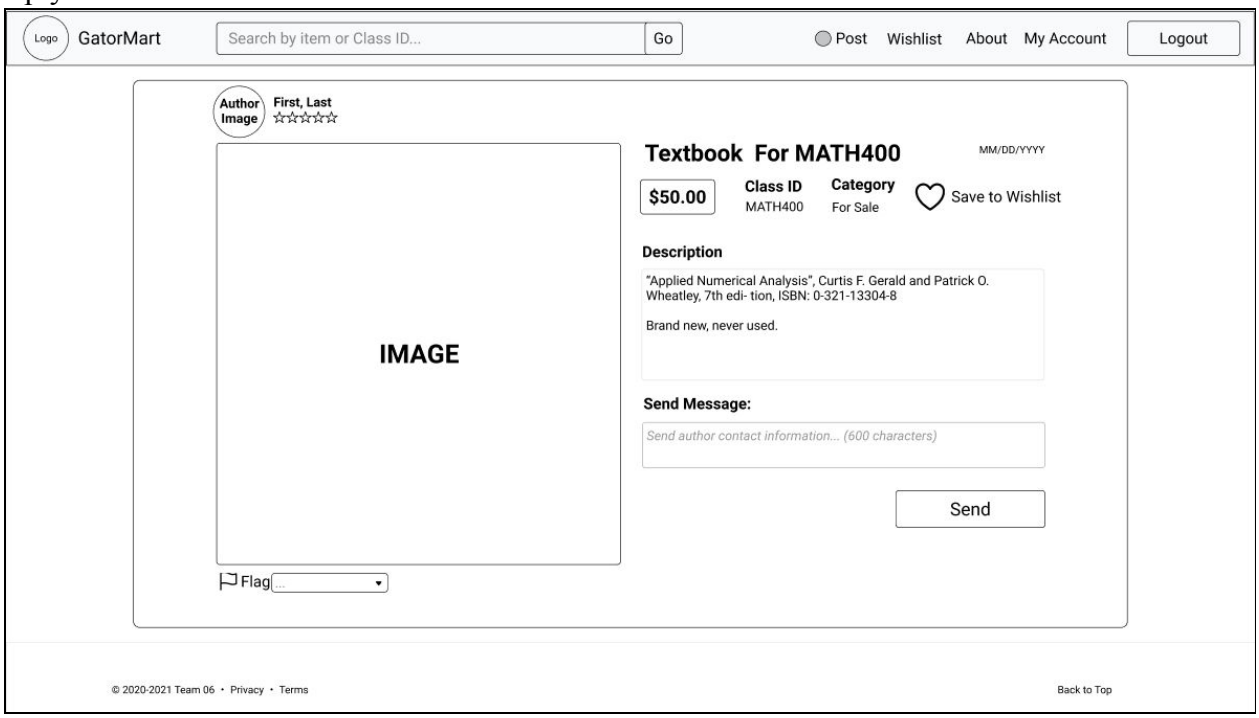
Login Page: Ricardo clicks the "Login/Register" button in the navbar and logs in with his active account.



Search Page: Ricardo uses the search bar to search for his class specific textbook by filtering the results by class ID.

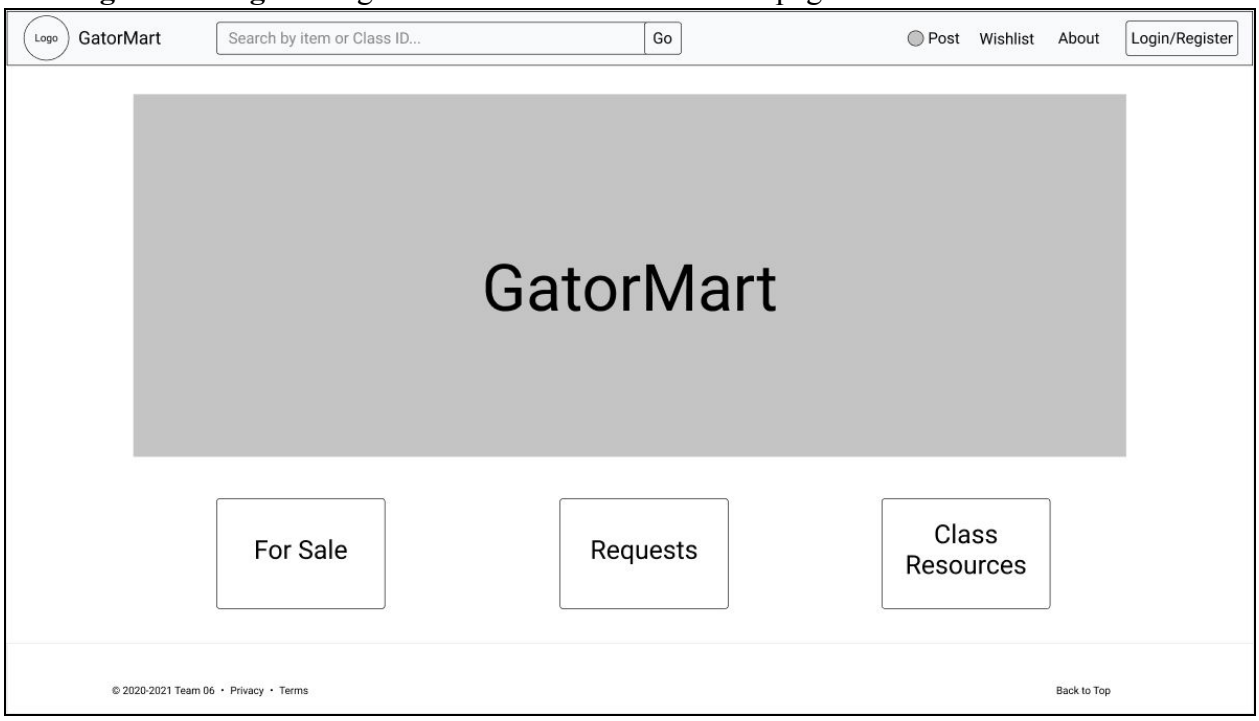


Item Page: Ricardo finds a post and sends a one-way message to the seller then waits for a reply.

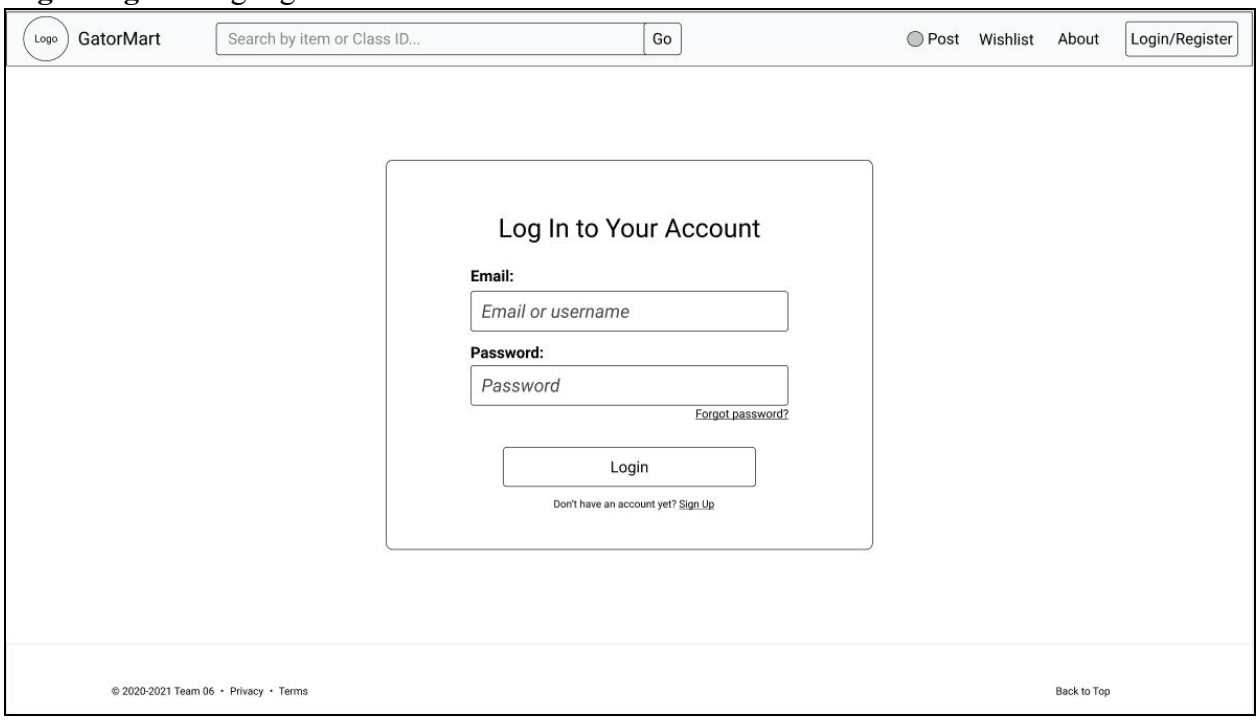


Use Case 3: Selling
Liang is an avid volunteer at her local nursing home and is currently preparing to relocate to Minnesota for a job opportunity. However, due to her constant volunteer roles, she does not have enough money to purchase a plane ticket to Minnesota. As she is cleaning out her room, Liang realizes that she cannot take multiple furniture pieces from her dorm. Liang decides to go onto the desktop GatorMart site and logs in with her already active account. She creates a "for sale" post for one piece of furniture, attaches an image from her phone of the furniture, and annotates their respective price point in the post description. After Liang's post is approved by site admin, an incoming freshman at SFSU interested in the item for sale immediately sends a message to Liang with their contact information. Liang and the freshman exchange information through their mutual desired means of communication and meet up to purchase one item from Liang's post. After the sale, Liang returns to the mobile GatorMart site, and because she is already logged in, she goes directly to her post and marks it as "sold" to prevent the post from showing up on future searches.

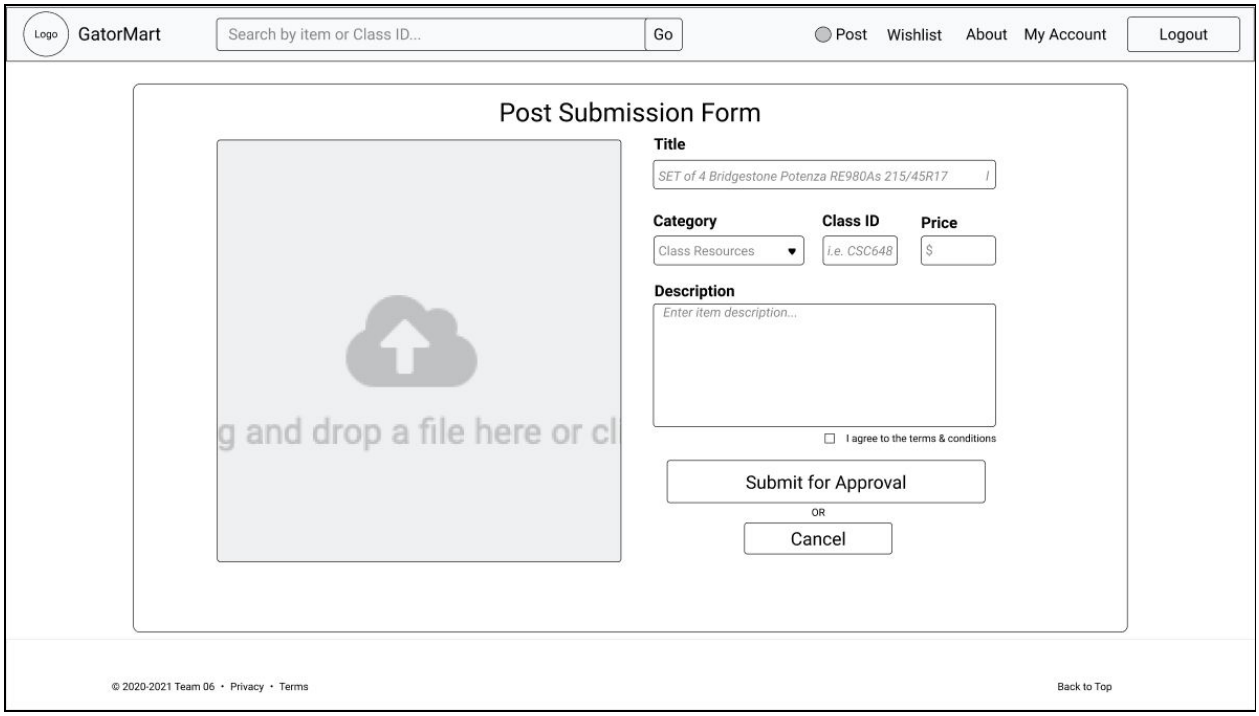
Landing/Home Page: Liang travels to the GatorMart home page.



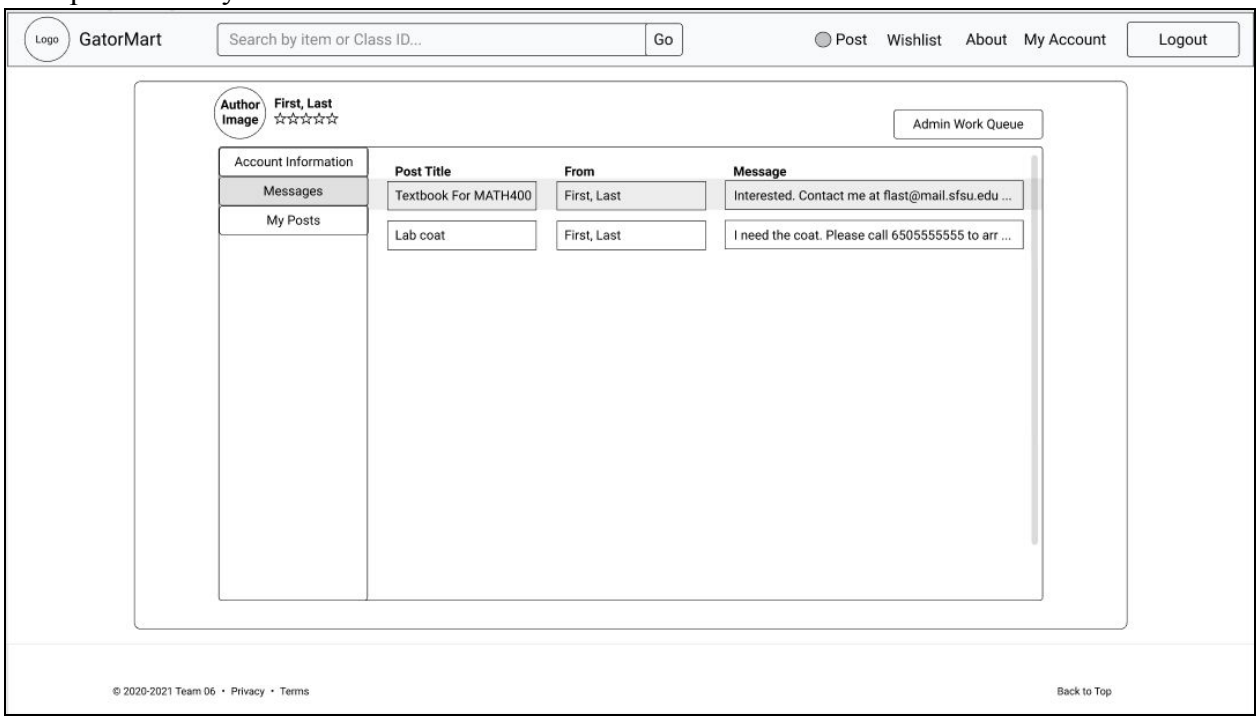
Login Page: Liang logs in with her active account.



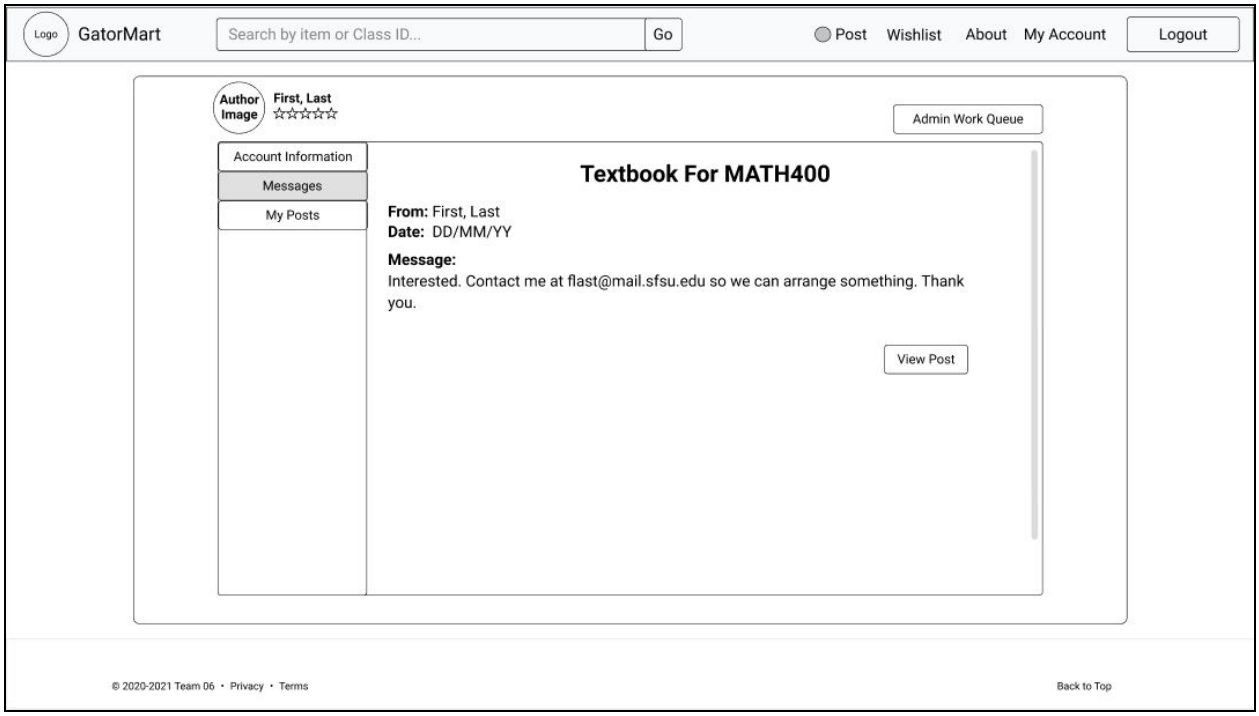
Posts Submission Page: Liang creates and submits a "for sale" post to site admin.



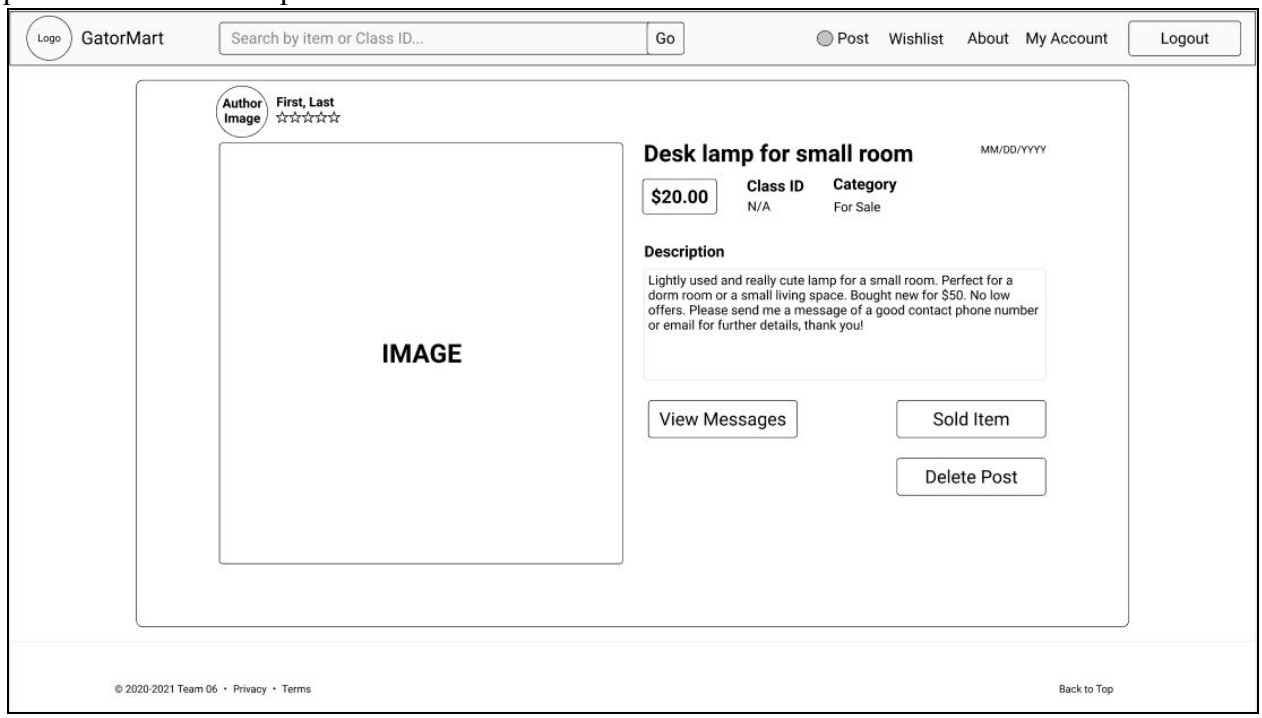
Profile Page (User Dashboard): Upon site admin approval of her post, Liang receives messages from potential buyers.



Profile Page (Message): Liang selects a message to read.

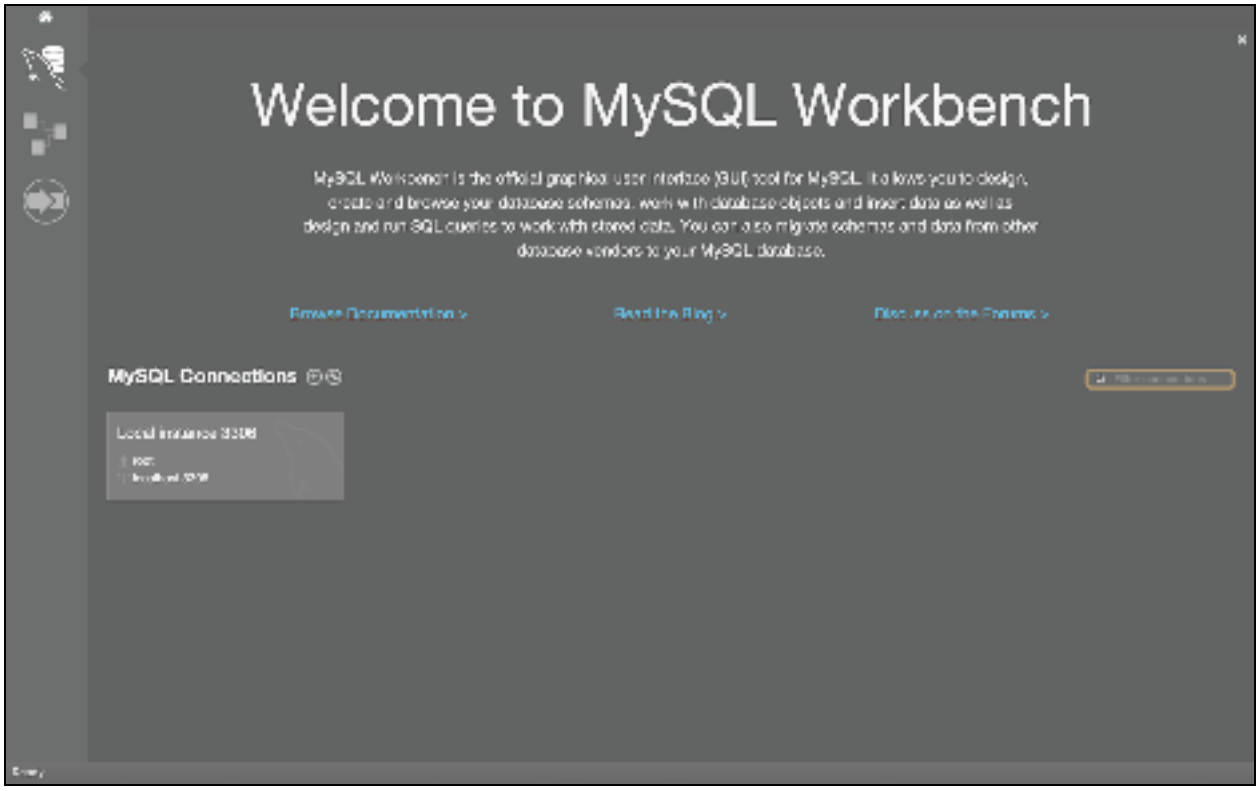


Item Page (Seller Point of View): Upon successful contact with a buyer and sale, Liang proceeds to mark her post as "sold".



Use Case 4: Administering Posts
Nadia is trying to keep up with her last semester set of courses at San Francisco State University (SFSU) while concurrently reviewing an influx of post submissions by faculty, staff, and students for the GatorMart website. As an experienced programmer, she was hired by GatorMart due to her knowledge in MySQL Workbench which is the main tool used to administer the website. At the beginning of her shift at GatorMart, Nadia logs on to our website's MySQLWorkbench and begins to search for posts pending approval and begins reviewing the content of each post. Nadia encounters a post with an inappropriate image and content. She denies the publication of this post and immediately deletes the seller's account for going against GatorMart terms and conditions. The next post Nadia reviews has a buyer requesting materials for a specific course ID which abides by GatorMart's terms and policies, she approves this post. Before finishing her shift, Nadia inspects posts "flagged" for inappropriate content by active users. She encounters a highly reported post that did not comply with the terms and conditions to a certain degree. She deletes the post and informs the poster of their error.

MySQLWorkbench (Admin): Nadia will be using this tool to approve or deny posts, inspect flagged posts, and manage users of the site.



High-Level System Architecture and Database

DB Organization:

GatorMart users stored in SQL table labeled '**users**':

id: Each user will have an associated id number with their account for backend uses that is auto generated upon account registration.

firstname: When signing up for the site, users will have to provide thor first name to be displayed on their account alongside posts.

lastname: Upon account creation, users must provide their names to be displayed on accounts and posts.

email: A user's school email must be provided and verified upon registration to be able to use the site.

password: Users need to create a password that will allow them to log into their accounts to post on the site.

rating: There will be a seller rating implemented into the site and the score will be tied to the account.

usertype: This field will show what type of user the account is and can grant admin permissions if it is the correct value.

profilepic: When creating an account on the site, the user will submit a profile picture to identify themselves to other users. This field will have the file path to the image.

active: This will be a field that will determine if the user has been banned from the site or not.

GatorMart posts stored in SQL table labeled '**posts**':

id: When a post is created by a user, it will be automatically assigned an id number for backend purposes.

title: When the user creates a post on the site, they will title their post with relevant information that other users will search for.

description: Upon creation of the post, the user will provide extra information about the item they are posting about. Things like condition and other relevant material.

price: The post will require a price associated for the item being posted about.

type: This field will be an option that the user creating the post will select to determine what kind of item the post will be. The different options include For-sale, Request, or Class Resource.

class: This will be a field for storing what class this item could potentially be useful for.

created: This will be the date associated with when the post was made for other users to determine relevance.

photopath: The user will be allowed to attach a picture to their post to display what the post is about. This will be the path to that image.

thumbnail: When in the main search page, users see a preview of the items being sold. The path to this lower resolution image will be stored here.
approved: The field will indicate if an admin has approved the post for display on the site.
fk_userid: When the user creates a post, their account id will be tied to the post to be displayed with the post and for backend purposes.

GatorMart messages stored in SQL table labeled '**messages**':
fk_sender: This field will store the user id of the person sending the message.
fk_receive: The field will store the user id of the person the message is being sent to.
created: This will be the date that the message will be sent on.
message: This will be the field that stores the message being sent between users.
fk_post: This will store the post id that the message will be about.

GatorMart sessions stored in SQL table labeled '**sessions**':
session_id: This will be the auto generated id for the session when a user logs in.
expires: This will be the value for when the session will no longer be valid.
data: This will store the cookie and information used to track the user session.

GatorMart flagged posts stored in SQL table labeled '**flags**':
reason: The reason that the post was flagged, an option from a provided list
fk_postid: The post id of the post that was flagged to be used in the backend.
date: The date that the post was flagged.

Media Storage:
Data will be kept in a file system rather than in DB BLOBs. This means file paths will be in the database leading to the images to be served to the user. There won't be any video, audio, or other form of media needed to be stored for the site.

Search/filter:
The search will work by a user submitting text to be compared to titles of posts on the site. This will be done with the SQL %like statement to try and find the closest item that the user is looking for. For filters, these will also be accomplished by using SQL statements that search for posts with desired fields.

Key Risks

1. **Skill Risks**
   Some team members have little to no experience in web development, and are concurrently taking CSC 317 for the first time this semester. This creates a skill risk, as some knowledge that is assumed may be missing or may be learned later than would be convenient for the project. To address this risk, we will look to anticipate what missing knowledge will be necessary to have, so that those lacking it can teach themselves as necessary.

2. **Schedule Risks**
   Many team members have very busy schedules and only have limited time to devote to the project. In order to manage this risk, we will keep each other informed via slack and during our team meetings about times when we are extra busy or when we have more time, so that we can plan accordingly to make sure things can get done on time.

3. **Technical Risks**
   Among our planned features is a wishlist that can be populated even by anonymous users, but that can then be stored in the database when a user registers. This presents a technical risk, as the implementation of this feature could be complicated. A further technical risk is that we would prefer to have a drag and drop style file upload for images, but are unsure as to how to implement that within our framework.

4. **Teamwork Risks**
   With any team it is possible for the members to have a difference of opinion that can lead to friction or even resentment. To mitigate this risk, it is important to ensure that every team member gets an opportunity to voice their opinions, and is listened to with respect and willingness to compromise.

Project Management

Our team has set up a slack channel for asynchronous communication between team members, and created a Trello board to keep track of which tasks are left to do, and which tasks are being worked on, and who is assigned to each task. We hold regular Zoom meetings on Thursdays in addition to the one on Wednesday during class. During these meetings we discuss what we want the project to look like, and what tasks need to get done by what deadline. We also use the meetings to work on tasks that need the whole team’s input, such as finalizing milestone documents, and reviewing each other's work.