Revision #: 1 Date: March 21, 2019

Software Engineering CSC648/848 Spring 2019

Milestone 2

LitLister

Team 4 - LitLister Team
Vismay Patel - Team Lead, Backend Engineer - vpatel3@mail.sfsu.edu
John Mendoza - Back End lead, Back End Engineer - Jmendo12@mail.sfsu.edu
Jesus Garnica - Front End Lead, Front End Engineer - jgarnica1@mail.sfsu.edu
Edwin Menjivar - GitHub Master, Back End Engineer - emen15@mail.sfsu.edu
Leonid Grekhov - Front End Engineer - lgrekhov@mail.sfsu.edu
Michael Winata - Front End Engineer - mwinata@mail.sfsu.edu

Milestone & Version	Date Submitted For Review	Date Revised
M1 V1	3/14/19	3/19/19
M1 V2	3/19/19	-
M2 V1	4/04/19	4/09/19
M2 V2	4/09/19	-

Table of Contents:

Content **Page(s)**

1. Data Definitions 2
2. Functional Requirements 4
   - Functional Specs 4
   - Non-functional Specs 6
3. UI Mockups and Storyboards 10
4. High level Architecture, Database Organization 24
5. High Level UML Diagrams 28
6. Key Project Risks 30
7. Project Management 32

1. Data Definitions

Datum	Definition
Client	
● User	The use of this term by itself will refer to someone who has registered with the site. A user can be one of multiple types, including: general, buyer, seller, customer service, or admin
● Guest / Unregistered User	User that has not registered with the site, can browse the listings with limited details and public user profiles, but cannot make purchases, or sales.
● General User	User that has registered with the site, can browse the listings, public user profiles, but cannot make purchases and sales.
● Buyer / Buying User	User that has registered with the site, can browse the listings public user profiles, configure personal profile, and payment information, and make purchases.
● Seller / Selling User	User that has registered with the site, can browse the listings, public user profiles, own sale log, configure personal profile, payment information, sale information and own listing, and make purchases and sales.
Company Employee	
● Customer Service	Employee that can observe buyer to seller exchanges of information and order history. Responsible to filter and to reply user inputs before passing them to the Admin Customer service representative will be available during business hours. They receive a notification every time a seller/buyer user disputes a transaction. Depending on the transaction, and after reviewing both sides, the customer service user can decide whether to refund the transaction, and fees. The customer service user also receives notifications when any user decides to contact customer service. The customer service user is able to advice the users in any of their problems, and they are able to fix any mistakes on their accounts.

● Admin	Employee that has full access to the website. This employee can ban users and remove illegal profiles and lists.
Other Data Definitions	
● Privacy Policy	Policies dealing with what data we collect, why we collect it, and what we do with the data
● Help	A document explaining other customers’ common issues
● Meeting Location	A public place inside the San Francisco State University Campus
● Approved credentials	Email verified as a valid San Francisco State University email
● Contact Customer Service	A service offered to email customer service representatives when having any problems with the website
● Book Listing	A post that contains the information of any book being sold
● Book	The book the user will be buying or selling. The system will maintain a list of validated published books, the user can only sell or buy book that are and stored in the database
● Author	The person who has written the book.

2. Functional Requirements

Functional Specs

	Specs	Priority
	For guests/unregistered users:	
1.	Only guests/unregistered who are SFSU students and faculty can sign up with their school email address.	**1**
2.	[Combined with feature 1, reason: De Morgan logic]	**-**
3.	Guests/unregistered users shall be able to search for enlisted books. 3.1. Search books by combinations of title and isbn	**1**
4.	[Combined with feature 5 -- check 5.2, reason: feature 5 covers feature 4]	**-**
5.	Guests/unregistered users shall not be able to see all information about the listings. 5.1. Display title, cover image, and isbn 5.2. Hide lister/seller information	**1**
	For general users:	
1.	General users shall be able to login using their approved credentials.	**1**
2.	General users shall be able to search for enlisted books. 2.1. Search books by combinations of title, isbn, book conditions and seller id.	**1**
3.	[Combined with feature 4, reason: feature 4 covers feature 3]	**-**
4.	General users shall be able to see all information about the listings. 4.1. Display title, cover image, isbn, price, book conditions and seller id	**1**
5.	General users shall be able to buy books, by adding payment information.	**1**
6.	General users shall be able to sell books, by adding payment information.	**1**
7.	General users shall be able to contact customer service.	**2**
	For seller user:	
1.	Seller users shall be able to list books for sale.	**1**

2. Seller users shall be able to edit all information about their listing. 2.1. Edit price and book conditions	**1**
3. [Combined with feature 2, reason: feature 2 covers feature 3]	**-**
4. Seller users shall be able to specify a meeting time that works for them.	**1**
5. Seller users shall be able to accept and decline meeting locations/time after buyer user accepts the listing.	**1**
6. Seller users shall be able to delete a listing.	**1**
7. Seller users shall be able to dispute a transaction.	**2**
For buyer user:	
1. Buyer user shall be able to accept and purchase any listed item.	**1**
2. Buyer users shall be able to accept the available time, or contact the seller user if the chosen time doesn't work for them.	**1**
3. Buyer users shall be able to select one of the public meeting locations from a drop down list.	**1**
4. Buyer users shall be able to dispute a transaction.	**2**
5. Buyer users shall be able to cancel the transaction within a reasonable time.	**1**
For customer service user:	
1. Customer service users shall be able to see all listings.	**2**
2. Customer service users shall be able to see all information related to each listing.	**2**
3. Customer service users shall be able to edit/remove any listing.	**2**
4. Customer service users shall be able to contact general users.	**2**
5. Customer service users shall be able to refund transactions.	**3**
6. Customer service users shall be able to edit general users' accounts.	**3**
7. Customer service users shall be able to suspend/ban general users' accounts.	**3**

	For admin user:	
1.	Admin users shall be able to see all listings.	**2**
2.	Admin users shall be able to see all information related to each listing.	**2**
3.	Admin users shall be able to edit/remove any listing.	**2**
4.	Admin users shall be able to contact general users.	**2**
5.	Admin users shall be able to adjust transactions.	**2**
6.	Admin users shall be able to approve new accounts for creation	**3**
7.	Admin users shall be able to edit general users’ accounts.	**3**
8.	Admin users shall be able to remove/suspend/ban general users’ accounts.	**3**
9.	Admin users shall be able to edit/remove customer service users’ accounts.	**3**

Non-functional Specs

1. **Security**
    a. Login shall be required to make purchases
    b. User’s shall verify their emails when registering an account
    c. User’s shall be able to set a display name different than their email
    d. User’s emails shall not be displayed by default
    e. Passwords shall be encrypted before storing in the database
    f. User’s session timeout limit shall be decided by the administrator
    g. User’s session shall only be ended by code design
    h. User can login to multiple sessions on different devices
    i. Content uploaded by users shall be audited by the administrator
    j. User’s payment information shall be encrypted
    k. This site shall not accept any third party cookies
    l. The meeting time and places users make with each other to exchange books shall not be revealed to other users not involved in the transaction
2. **Audit**
    a. New registrations shall be audited by the administrator
    b. New registrations shall be approved by the administrator
    c. Users shall not be able to login to administrator accounts
    d. New sale listings shall be approved by the administrator
3. **Performance**
    a. The site loading time shall be less than 2 seconds for all screens

   b. Application shall be able to retrieve information from the database and react in a timely manner.
   c. The site shall handle requests asynchronously following a REST format

4. **Capacity**
   a. The total data storage for the site shall not exceed 80% of the server's capacity for this site
   b. The website shall be capable of handling at least 50 users
   c. The website shall be scalable, so that new features can be added easily
5. **Reliability**
   a. Downtime for maintenance shall be less than 3 hours per month
   b. Downtime for maintenance shall not affect the site's main functionality
   c. In all cases, users shall be informed of downtime for maintenance, either via an announcement on the main page, or e-mail
6. **Recovery**
   a. In case of a total site failure, the whole site shall be shut down for revision.
   b. If the site is broken, the mean time to recovery shall not exceed one day.
   c. User data is the most valuable aspect and priority will be placed on recovering such data in case of total failure.
7. **Data Integrity**
   a. Database tables shall be backed up weekly
   b. Administrator shall be able to execute a recovery if needed
   c. Image sizes shall be restricted to at most 1 megabyte
   d. Images shall be uploaded in jpg, jpeg, or png formats
   e. Images will be saved on Amazon's s3 storage server
   f. URLs to image will be stored on the database
8. **Compatibility**
   a. The site shall be compatible with the last version of the Safari browser version 11.1.2
   b. The site shall be compatible with the last version of the Firefox browser version 64
   c. The site shall be compatible with the last version of the Chrome browser version 73
   d. Third party applications shall not be able to modify any content that may affect the site compatibility
   e. Content should be able to be ignored by most popular ad-block services.
   f. The site shall be able to account for any other compatibility issues created as a result of browser updates in the future
   g. The site should be compatible to escalate to new databases
9. **Conformance with Coding Standards**
   a. Architecture and design standards shall meet all the requirements listed under the High-level system architecture and technologies used section of this document
   b. Design pattern is to be strictly enforced with all aspects of the site.

c. Appropriate documentation must be created for all code that is individually written for future maintenance.
d. Production code shall not have any log or output to the console.
e. All errors must not halt the web application without appropriate error handling.
f. Only working code that meets all code standards shall be submitted to the main branch of the project repository
g. Code shall be thoroughly tested and debugged before being considered working code
h. All internal errors and exceptions encountered when writing or modifying code shall be stored in a log
i. Any error that can affect the site's functionality shall be reported to the user
j. Errors shall be handled in a way that does not affect site functionality
k. The whole production cycle of the site shall be finished at least one week before the delivery date
l. The site shall be tested and debugged as a whole product at least one week before the delivery date
m. The site shall not be launched without all priority one features finished and working
n. All major changes to the application shall be discussed by the team and communicated to the class CTO.

10. **Look and Feel Standards**
    a. The application and it's layouts shall look professional
    b. The site shall be simple, so that it is usable to a wide range of users, and all previously mentioned parties
    c. Targeted users will be the main priority for ensuring usability and readability.
    d. Elements on screen shall meet the compatibility standards of all supported browsers
    e. Elements on screen shall meet the compatibility standards of all supported browsers on mobile devices
    f. Elements on screen shall be aesthetically pleasing
    g. The site shall be able to work correctly without mouse interaction
    h. The site shall be able to work correctly without keyboard interaction
    i. Elements in screen shall be resized automatically without user interaction when being loaded in all the different platforms supported by the site
    j. Application's user interface shall make it easy for user to find what they are looking for.
11. **Internalization / Localization Requirements**
    a. The default language of the site shall be English
    b. The site shall only allow SFSU students to register accounts initially
    c. The site shall be scalable to incorporate other schools into the user base
12. **Scalability**
    a. There shall be a whitelist of accepted school e-mails that can be updated to incorporate other schools

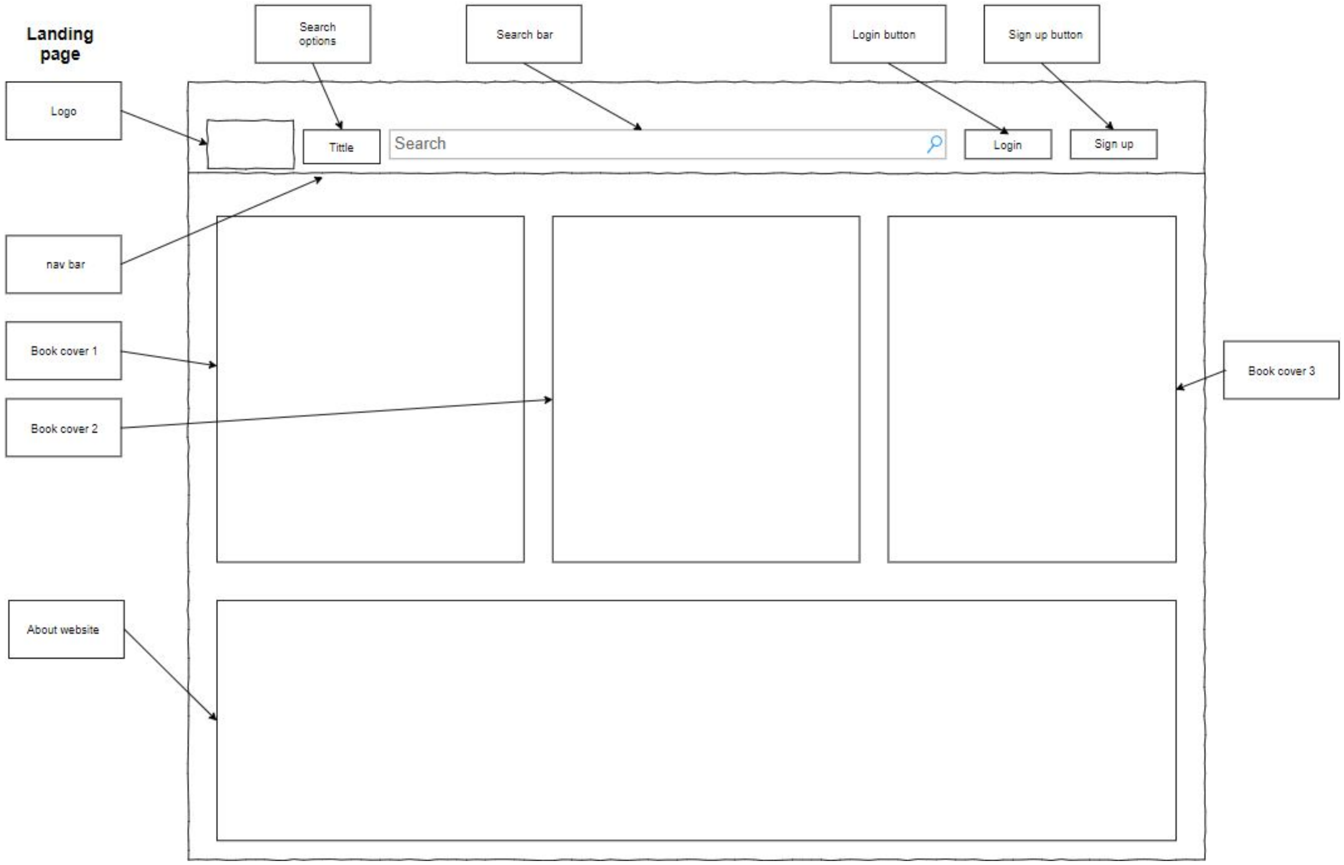
b. The whitelist shall be stored in a file on the server
c. The CPU instance and storage capacity shall be updated to be able to handle a large amount of users if needed

13. Web Site Policies

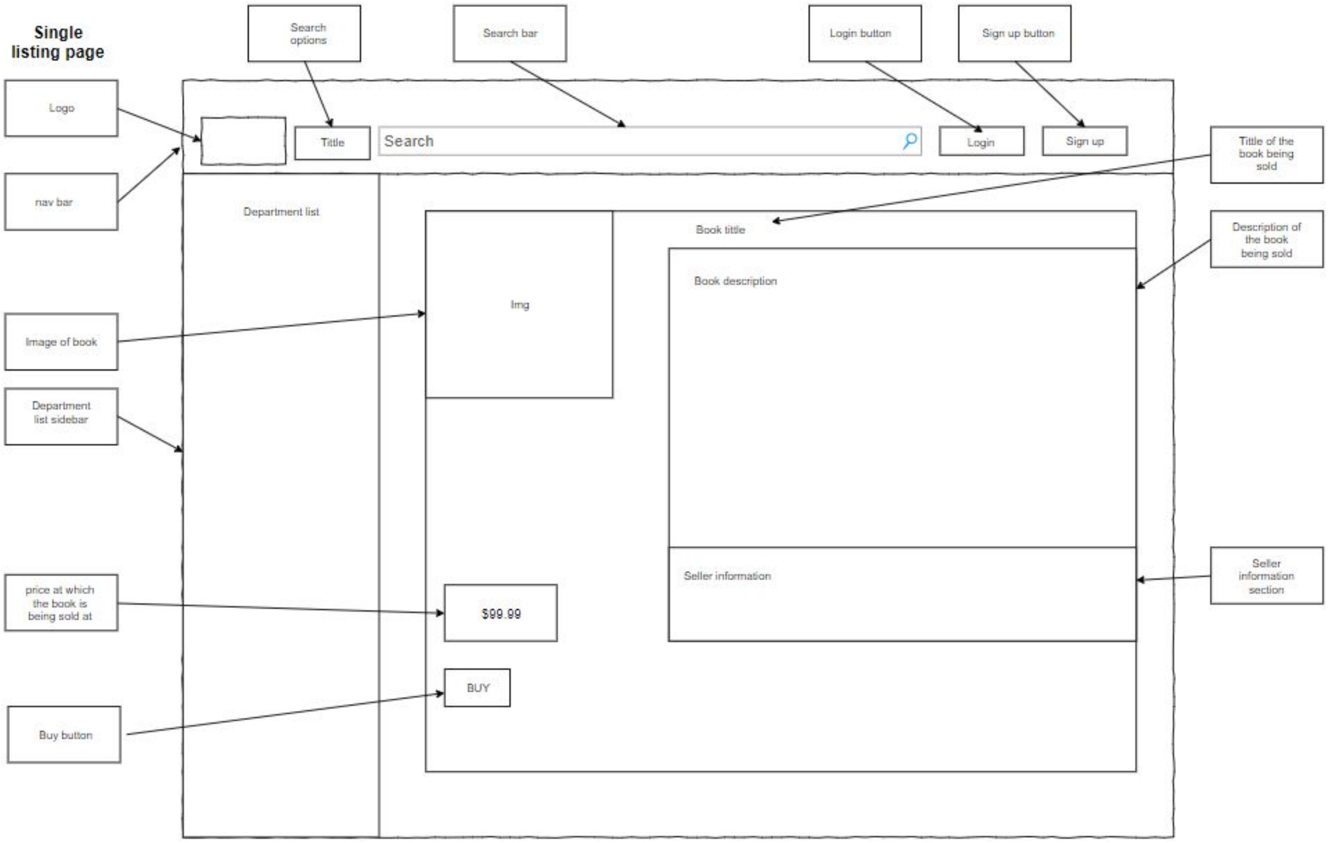
a. A link to the policies of this site shall be always visible in all its pages to be accessible by all the parties
b. Users payment information shall be kept confidential and secure
c. The website shall allow users to register an account.
d. Email verification shall be implemented upon registration.
e. User’s shall agree to application’s privacy policy before using the product.

3. UI Mockups and Storyboard

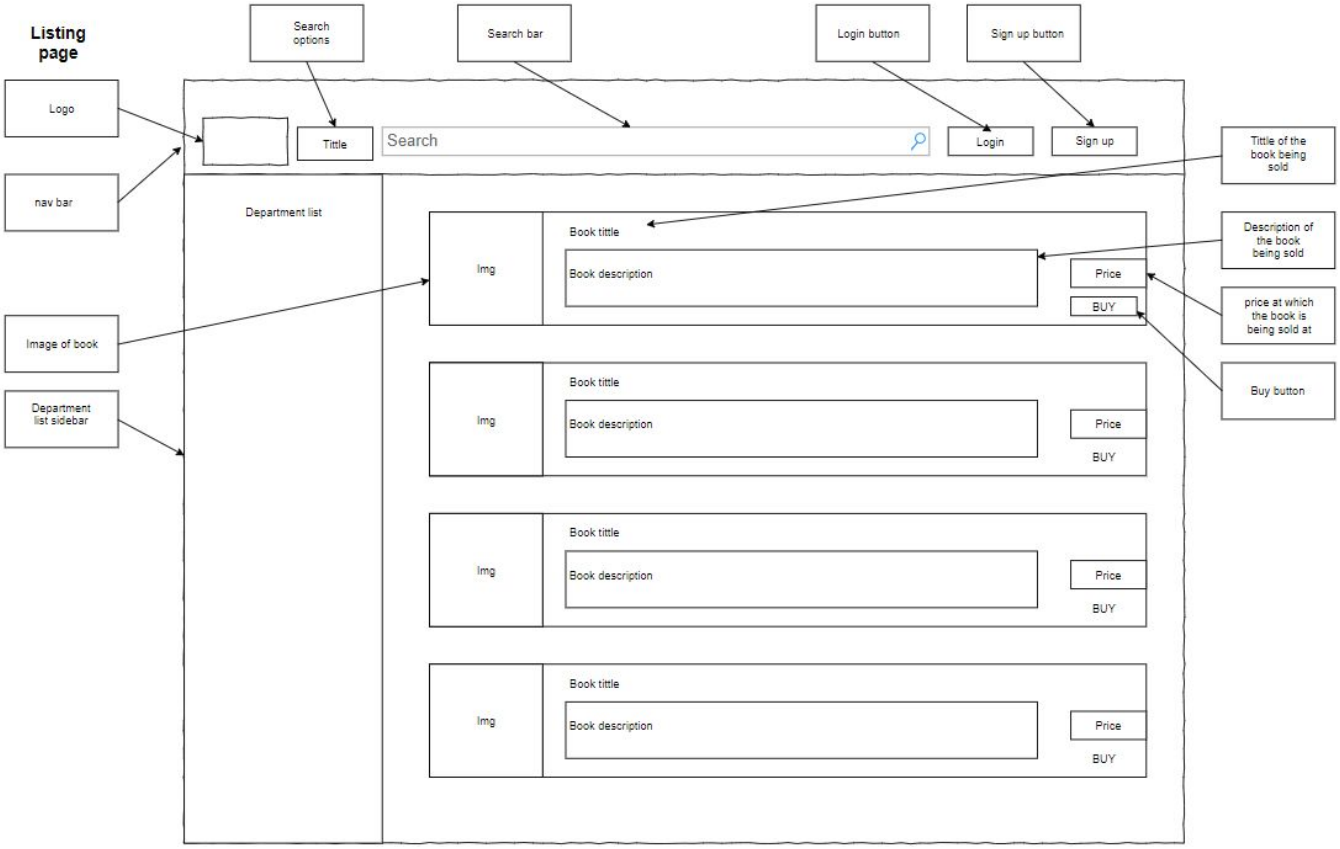
Lobby



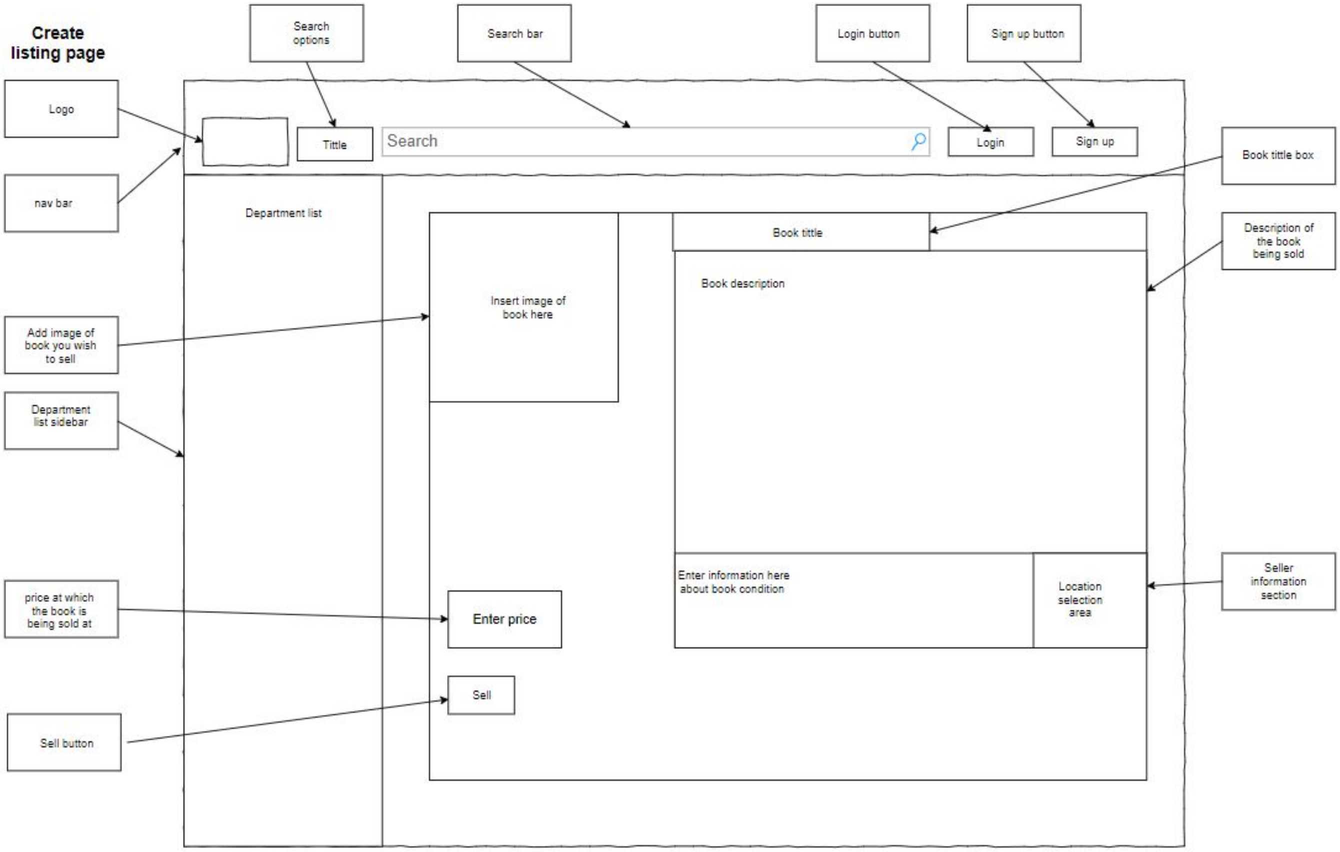
Single listing



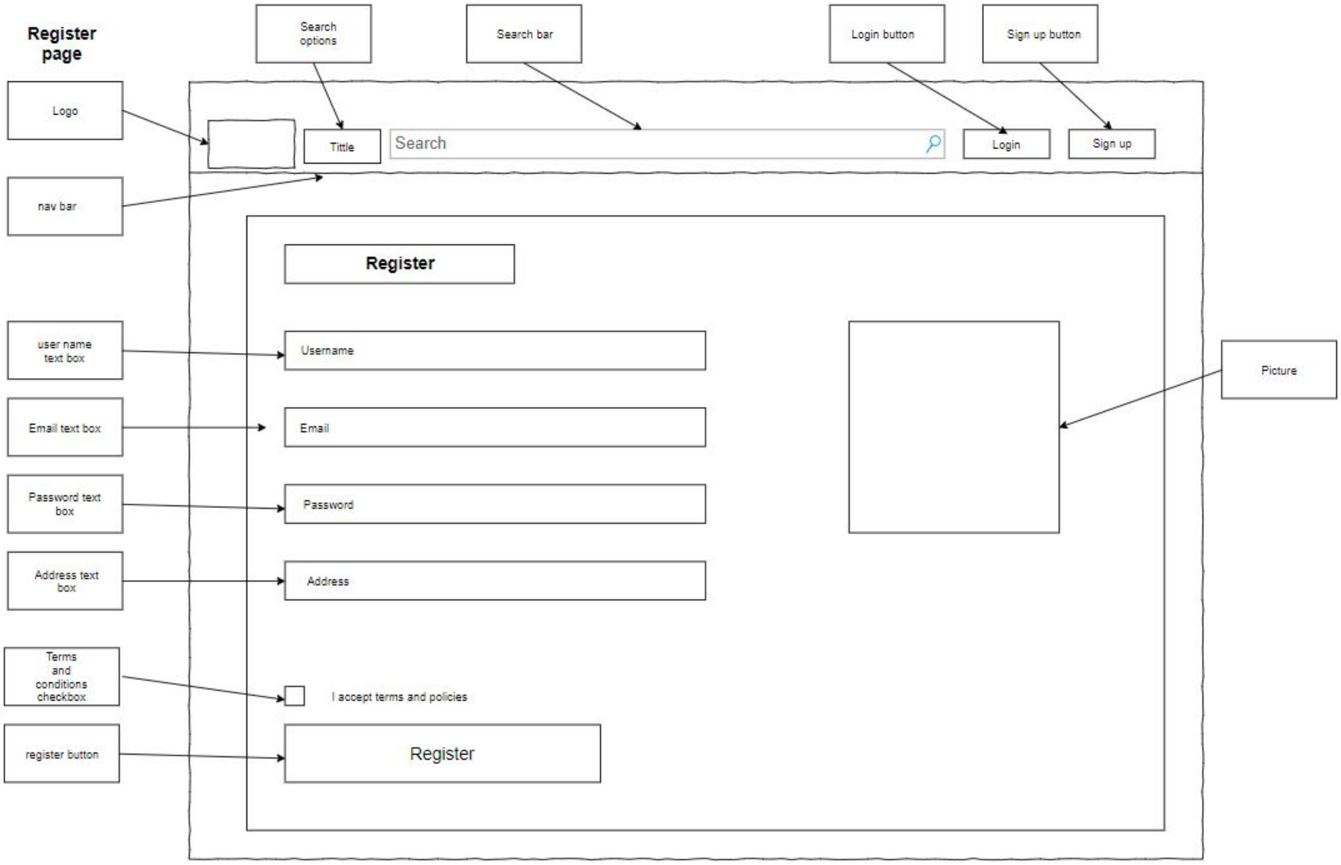
Listing


Listing page
Search options
Search bar
Login button
Sign up button
Logo
nav bar
Tittle
Search
Login
Sign up
Tittle of the book being sold
Department list
Book tittle
Img
Book description
Price
BUY
Description of the book being sold
price at which the book is being sold at
Image of book
Buy button
Department list sidebar

Sell request



Registration

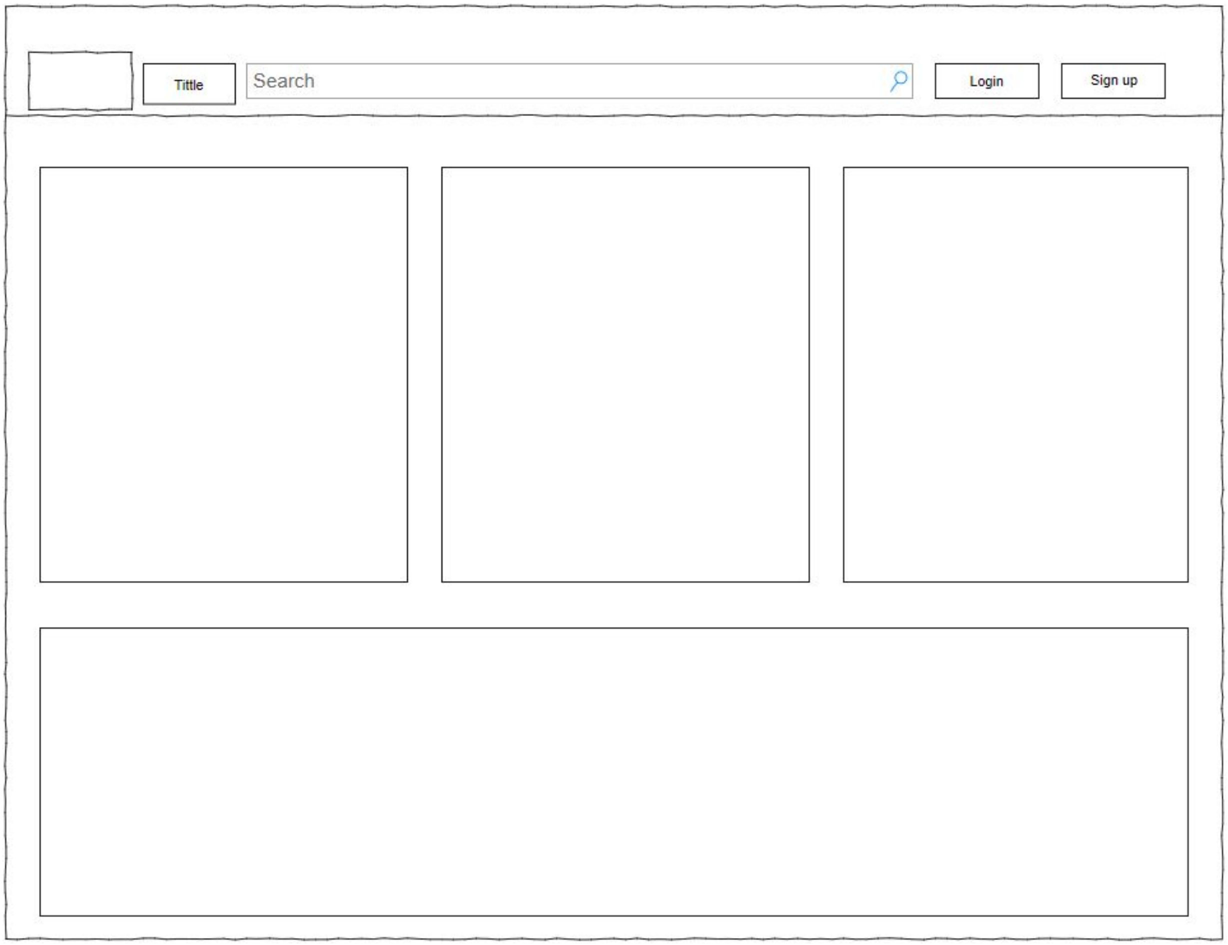


Storyboards

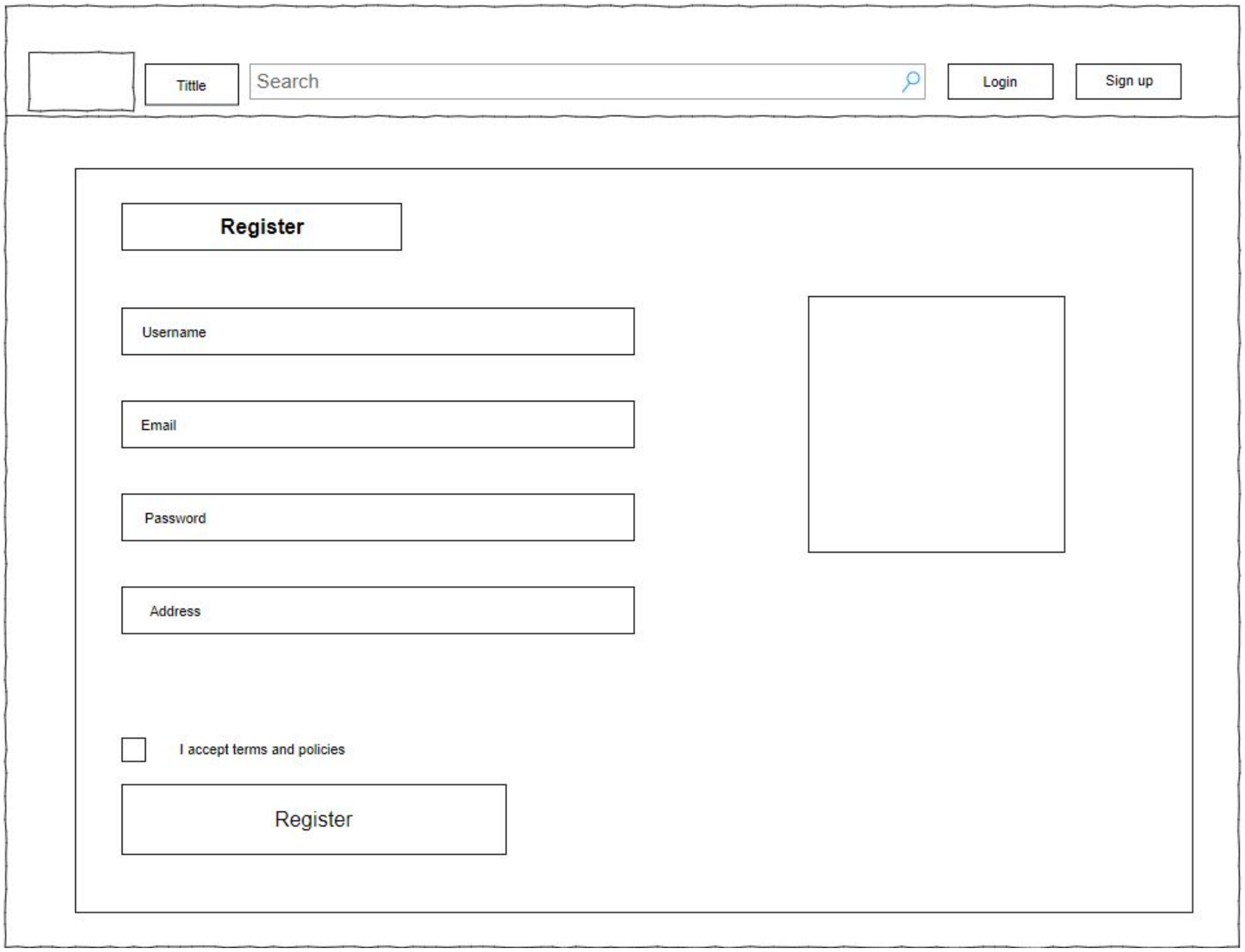
Guest / Unregistered User:

Philip is a junior at SFSU. He has heard of a new platform called “LitLister”. He has been buying books through Amazon and occasionally through the school bookstore. He looks up this new platform on his laptop and searches for a book he will need for the semester. He can not yet see the book prices, because he is not a registered user. He is curious about the prices so he decides it is probably worth it to create an account. He creates and verifies his account using his SFSU e-mail, and he is now free to browse any part of the site he’d like to.

Philip is on the landing page:



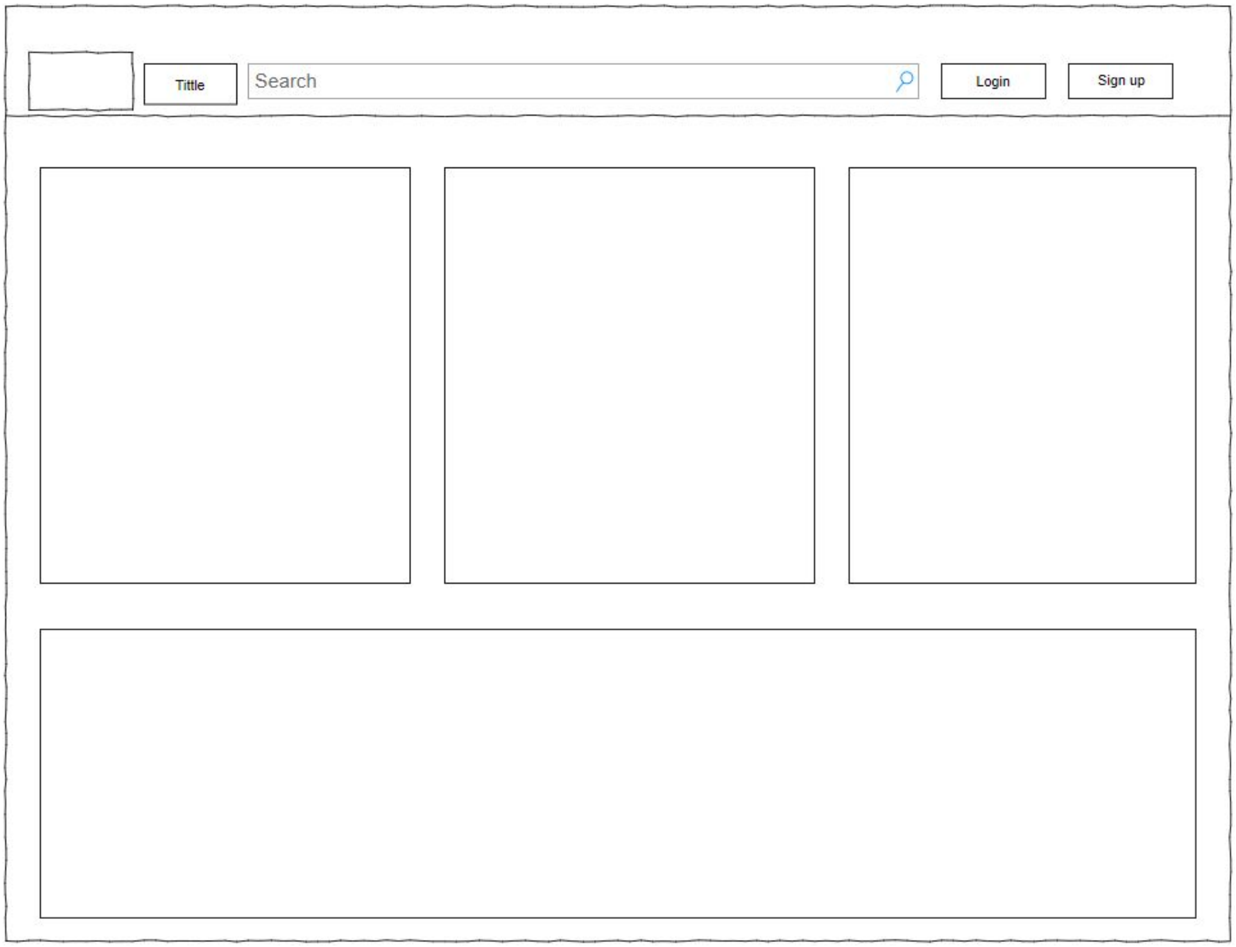
Philip Clicks on register and is taken to the registration page:



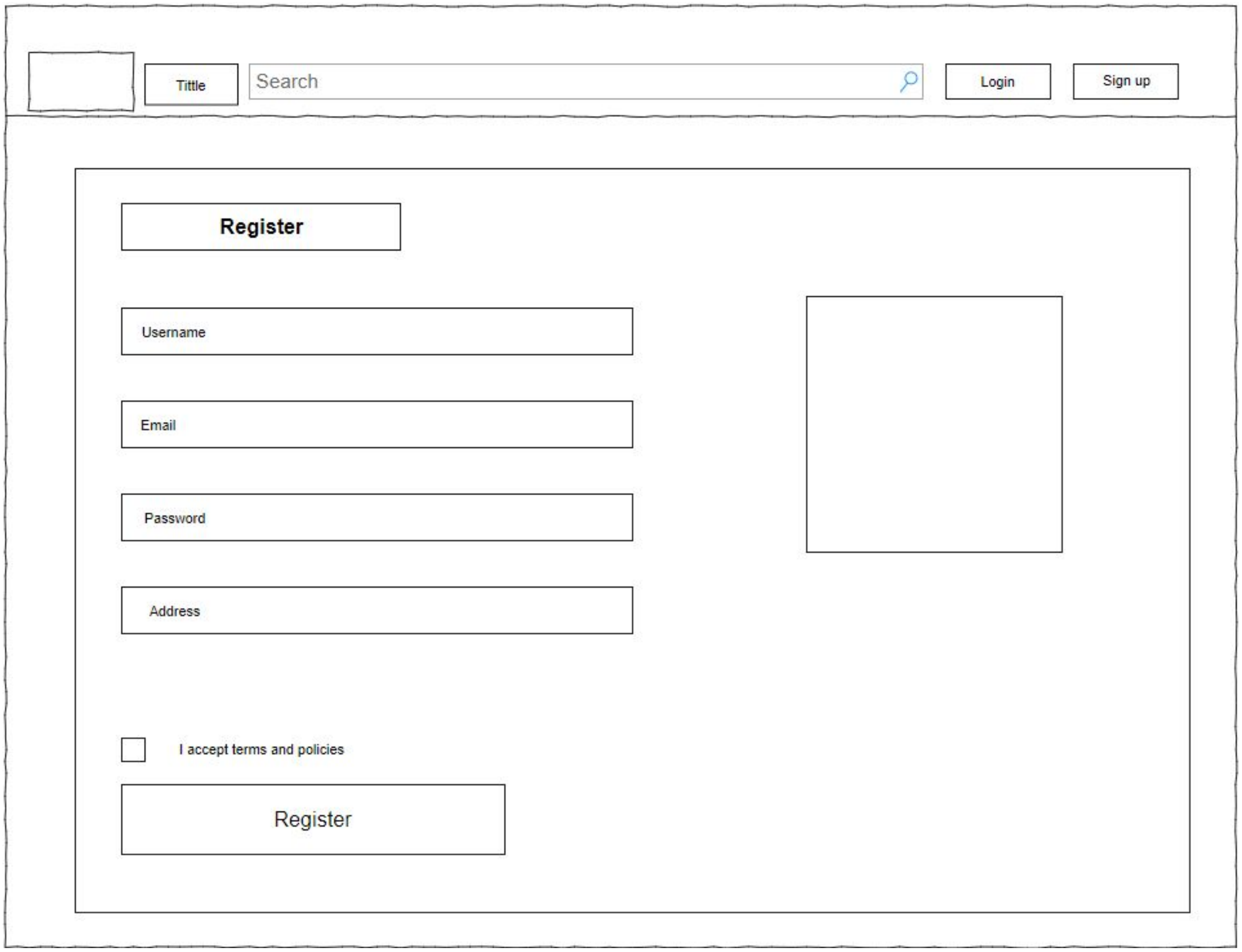
General User/ Registered User Buying a book:

Jake has been browsing through LitLister for awhile as a guest. He finds a book that he needs, but he is not sure whether he wants to buy the book immediately because the price is not visible. Jake registers an account at LitLister. Jake can see the prices on the list now, so he navigates to a listing for a book he needs and purchases the book. Jake and the seller meet and Jake receives his book. At the end of the year, Jake no longer needs the book, so he returns to LitLister, logs in to his account, and creates a listing to sell his book. A buyer purchases Jake's book, and Jake and the buyer arrange a time and place to meet to exchange the book using the website. Jake and the buyer meet, and Jake exchanges his book.

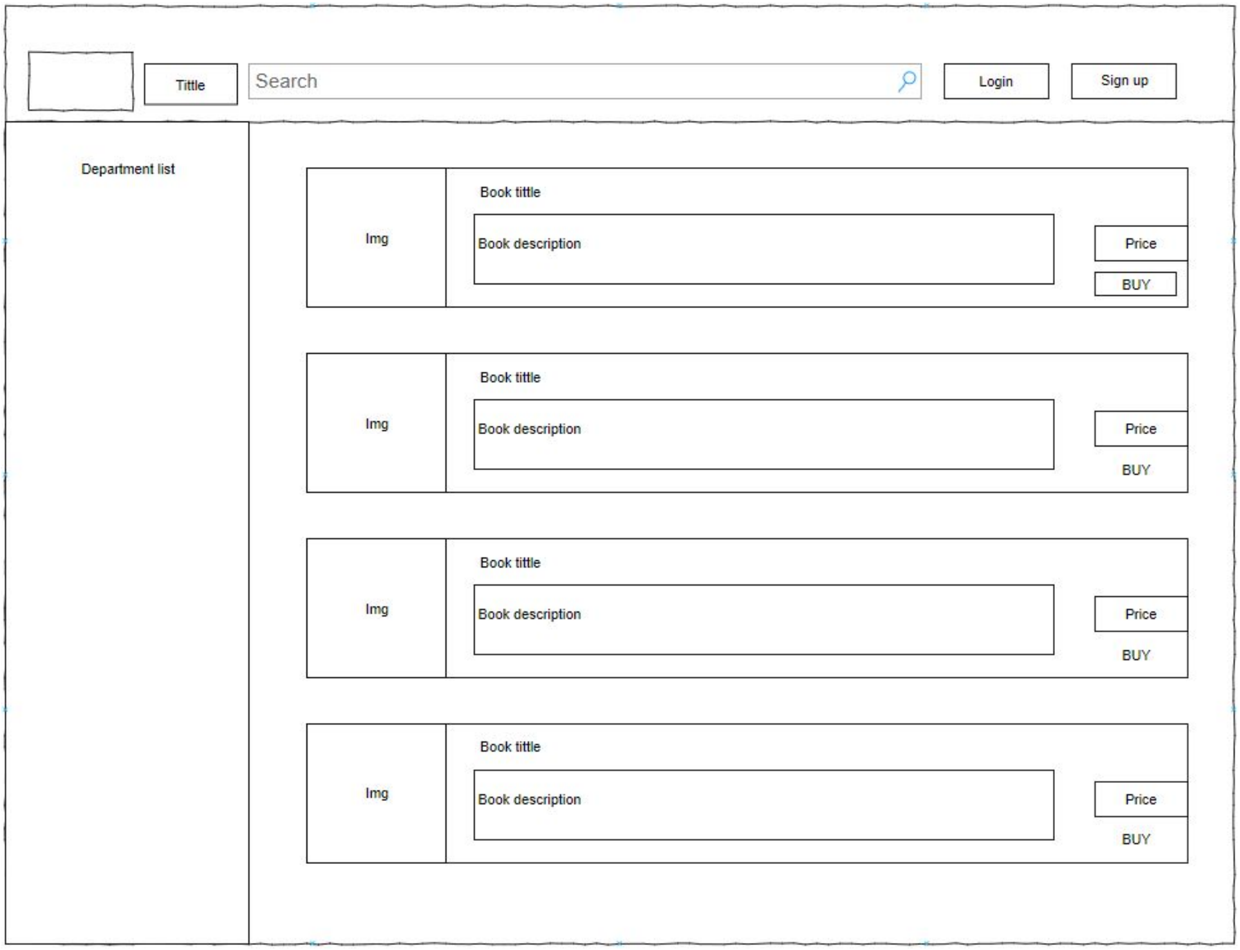
Jake is browsing from the lobby page:



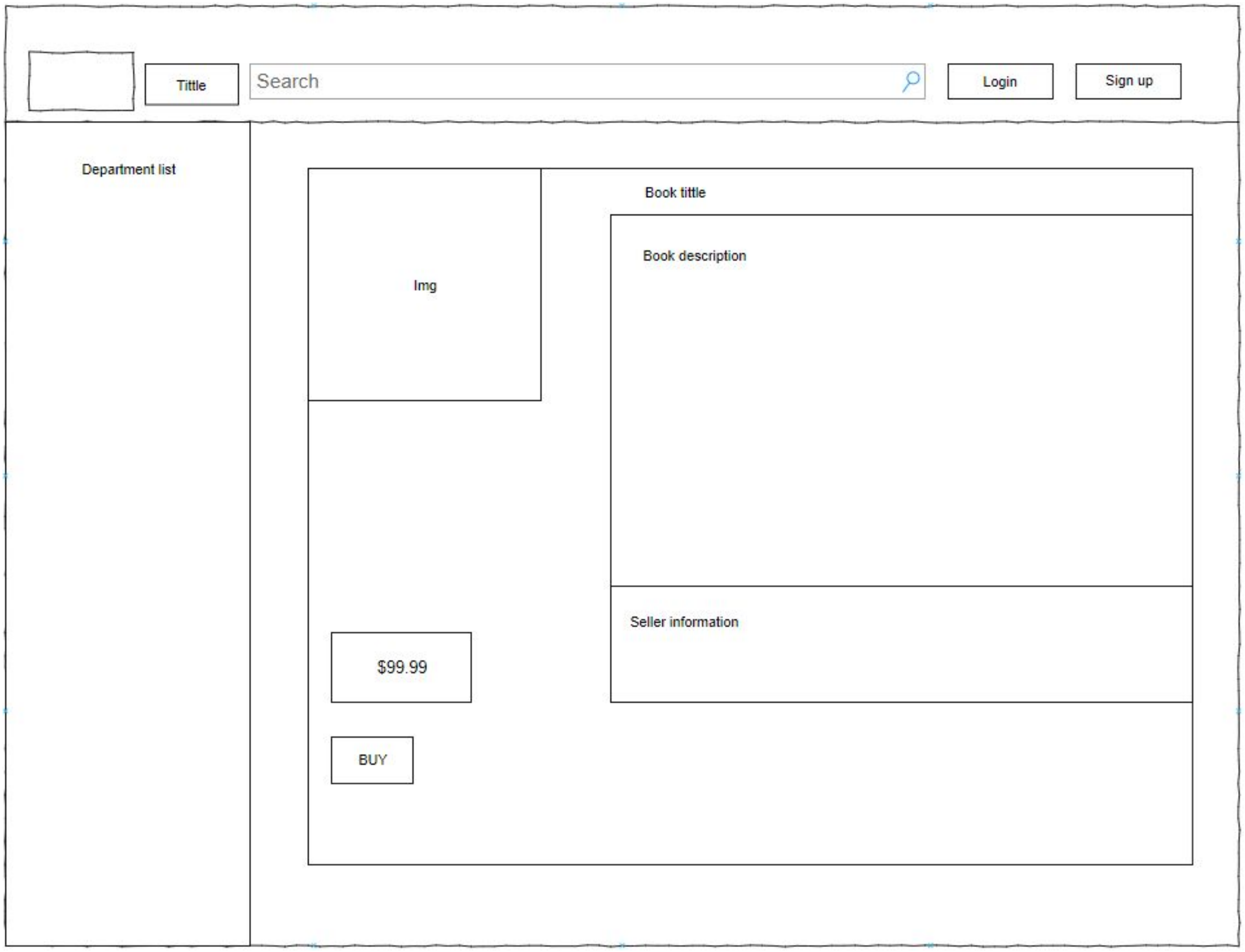
Jake finds a book he wants and registers:



Jake finds book in listing page:



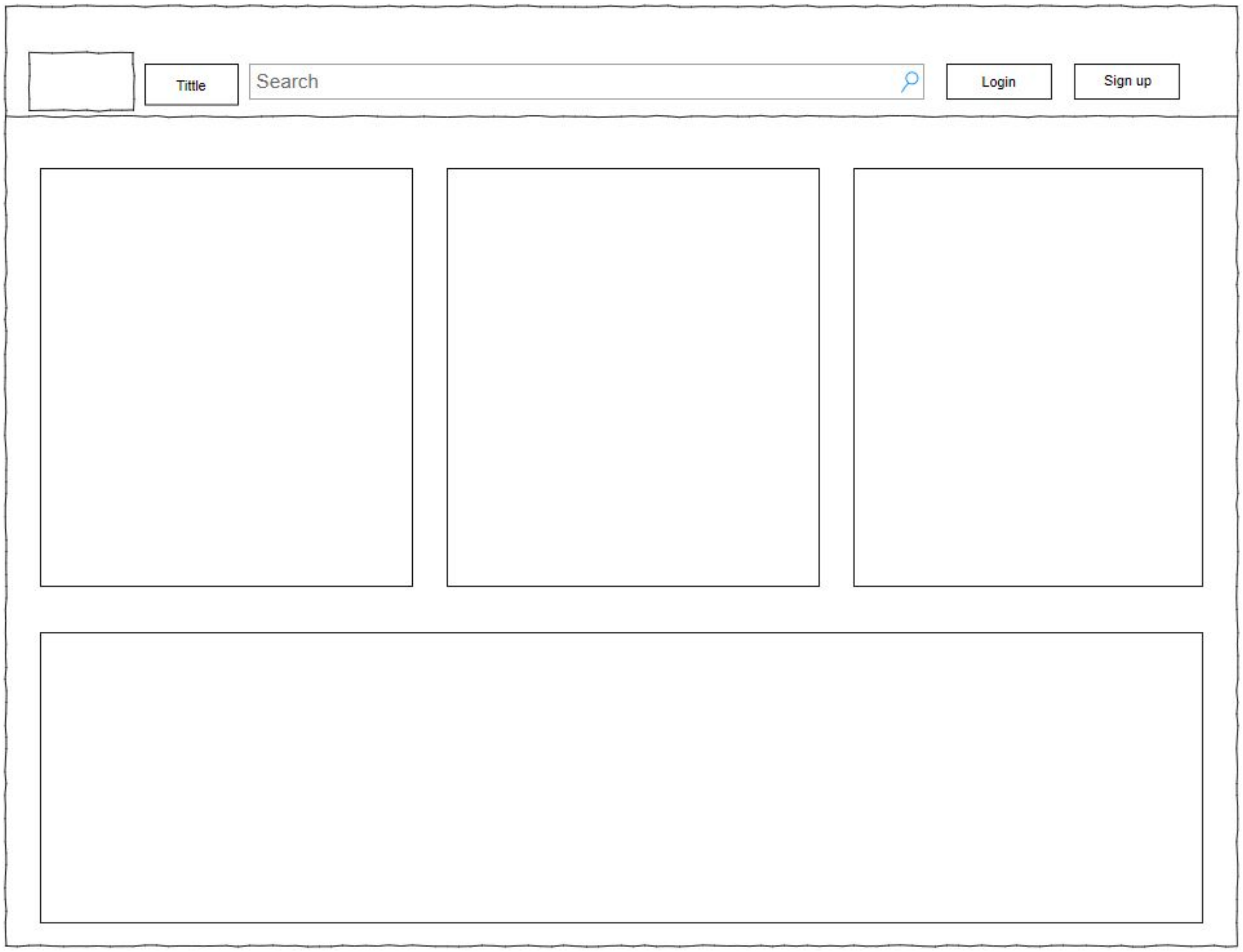
Jake buys book through single listing page:



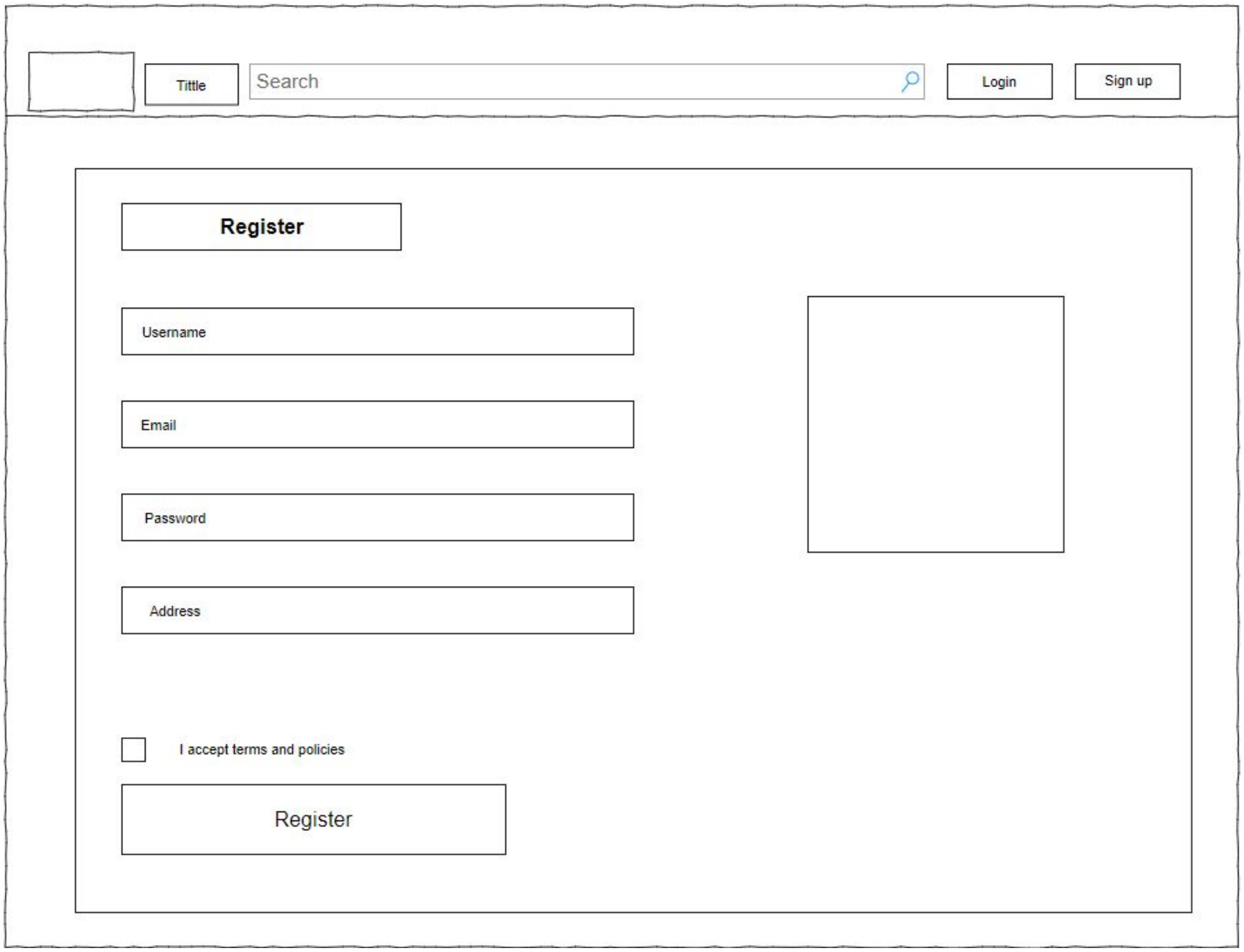
Seller / Selling User:

Jane is a senior at SFSU, and she has just finished her last semester at the school. She has many left over books from all her classes and decides that she needs to sell them to make up some of the money she spent on college. She visits our website and sees that there are options for selling and exchanging books. She decides to make an account and lists her books for sale through the website. After selecting the time and location she would like to sell at, another student sees the posting, and contacts Jane about the posting. Jane and the buyer meet at the designated location and exchange the book concluding the transaction.

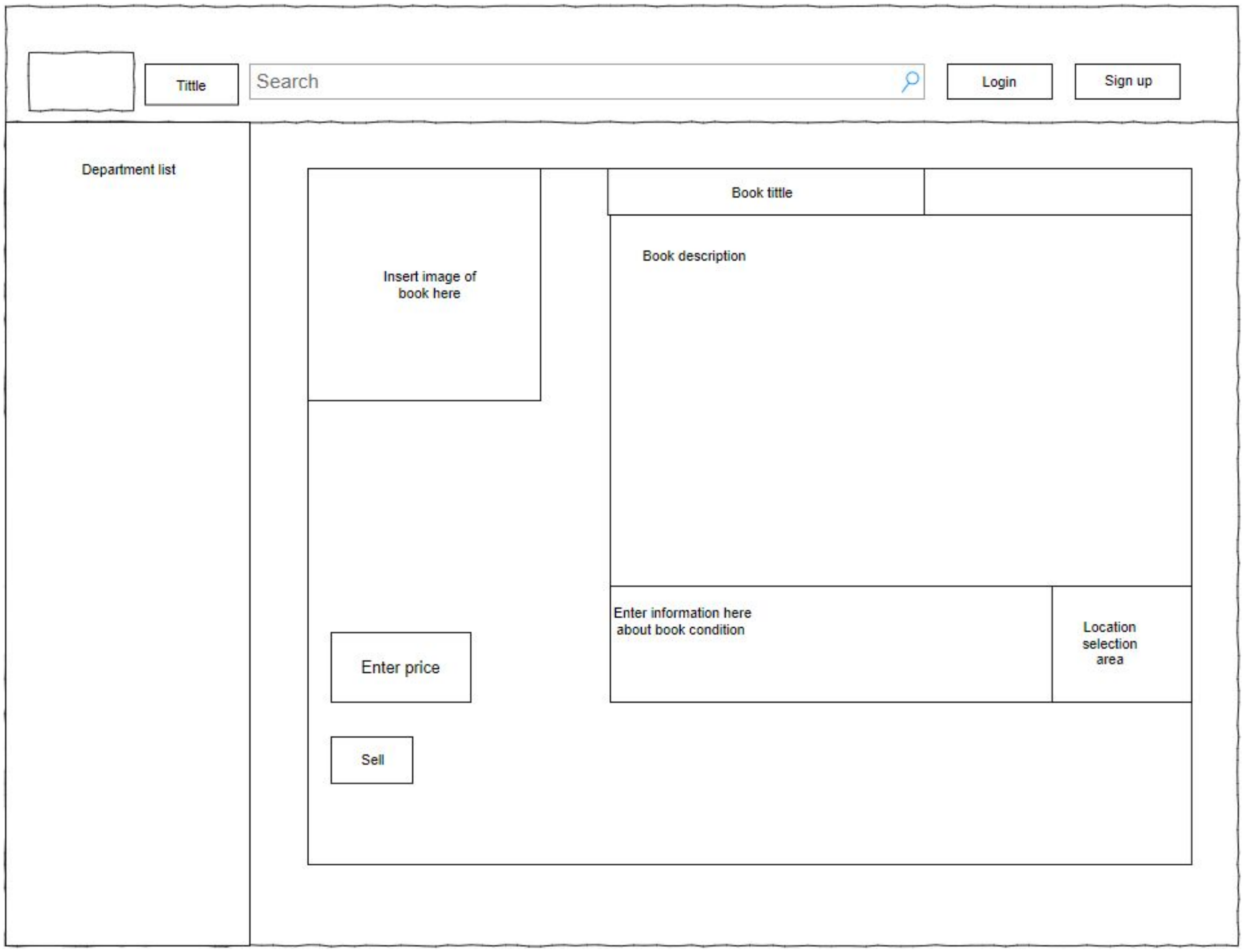
Jane on the lobby page:



Jane registers:



Jane Create a listing for a book she wishes to sell:

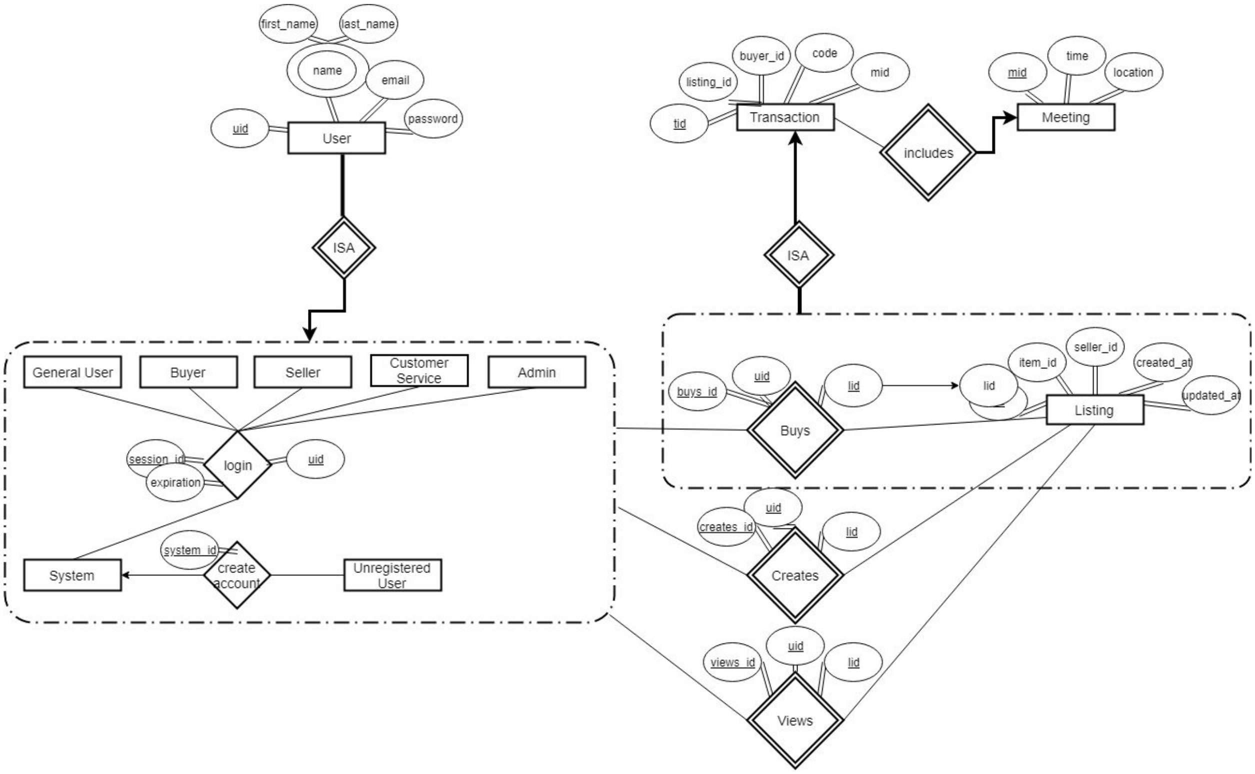


4. High Level Architecture, Database Organization

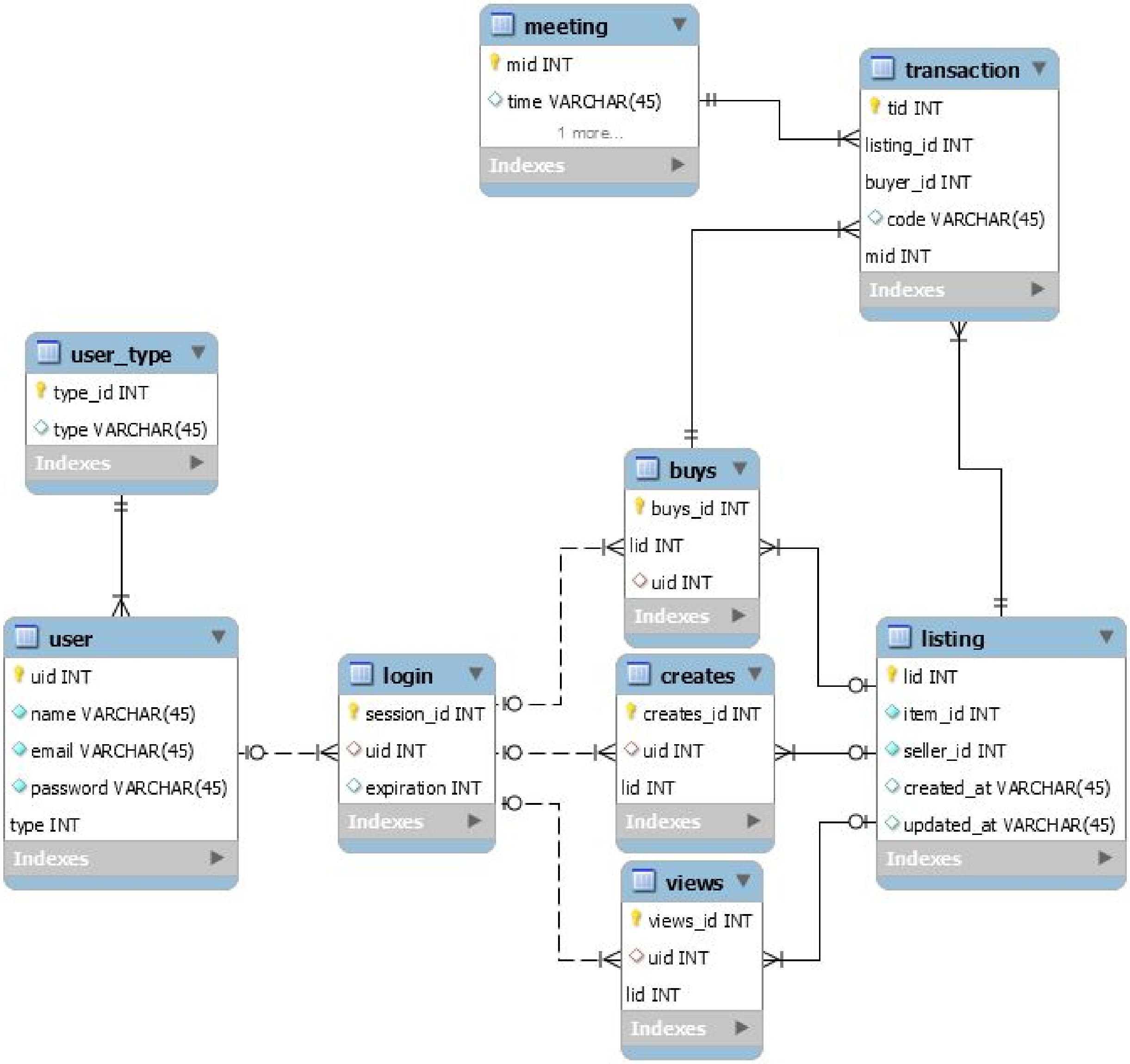
- **DB Organization:**
    - **Business Rules:**
        1. Multiple unregistered users can create one account into the website.
        2. Each user must have only one type.
        3. Multiple registered users can login into the website.
        4. Multiple users can view the multiple listings.
        5. Multiple users can buy multiple listings.
        6. Multiple users can create multiple listings.
        7. A single transaction is made up of only one buyer.
        8. A buyer can buy only one listing per transaction.
        9. A transaction must contain at least one listing.
        10. A single transaction must include only one meeting.

    - **Entities and Attributes:**
        - Strong entities:
            - User: uid: PK, user_type, name: composite(first name, last name), email, password
        - Weak entities:
            - Listing:
                - lid: PK, item_id, seller_id, created_at, updated_at
            - Transaction
                - tid: PK, listing_id, buyer_id, code, meeting_id
            - Meeting:
                - mid: PK, location, time
        - Relationships:
            - Login, Creates, Buys, Views, ISA, Includes, Create Account

- **Entity Relationship Diagram:**



- **Database Model:**



- **DBMS to use:**
    - Our team will be using MySQL to create the database, because it is one of the easiest DBMS to use, and our back-end team is very familiar with it.

- **Media Storage:**
    - Our media resources will be stored within the server’s file system using S3. The database will only store links to media resources.
- **Search/filter architecture and implementation:**
    - Our algorithm will use contains to search for matching substrings to a user’s search. The search items will be organized based on how old the listing is, the price, the posting user’s rating, and based on how the user would like to organize the listings; for example, the user shall be able to sort by price, or date. The database terms that will be searched include name of the book, isbn, the department category, author. We will be using sequelize to code our migrations, seeds, and connections.
- **Your Own APIs:**
    - Some of our APIs that we will be implementing are a login API, a search API, and a database API.
- **Non-trivial algorithms:**
    - The default algorithm we will implement for sorting listings will sort them by the lister’s ratings; we will implement this by using Sequelize’s “order” API. For example, this would be achieved through the following code:

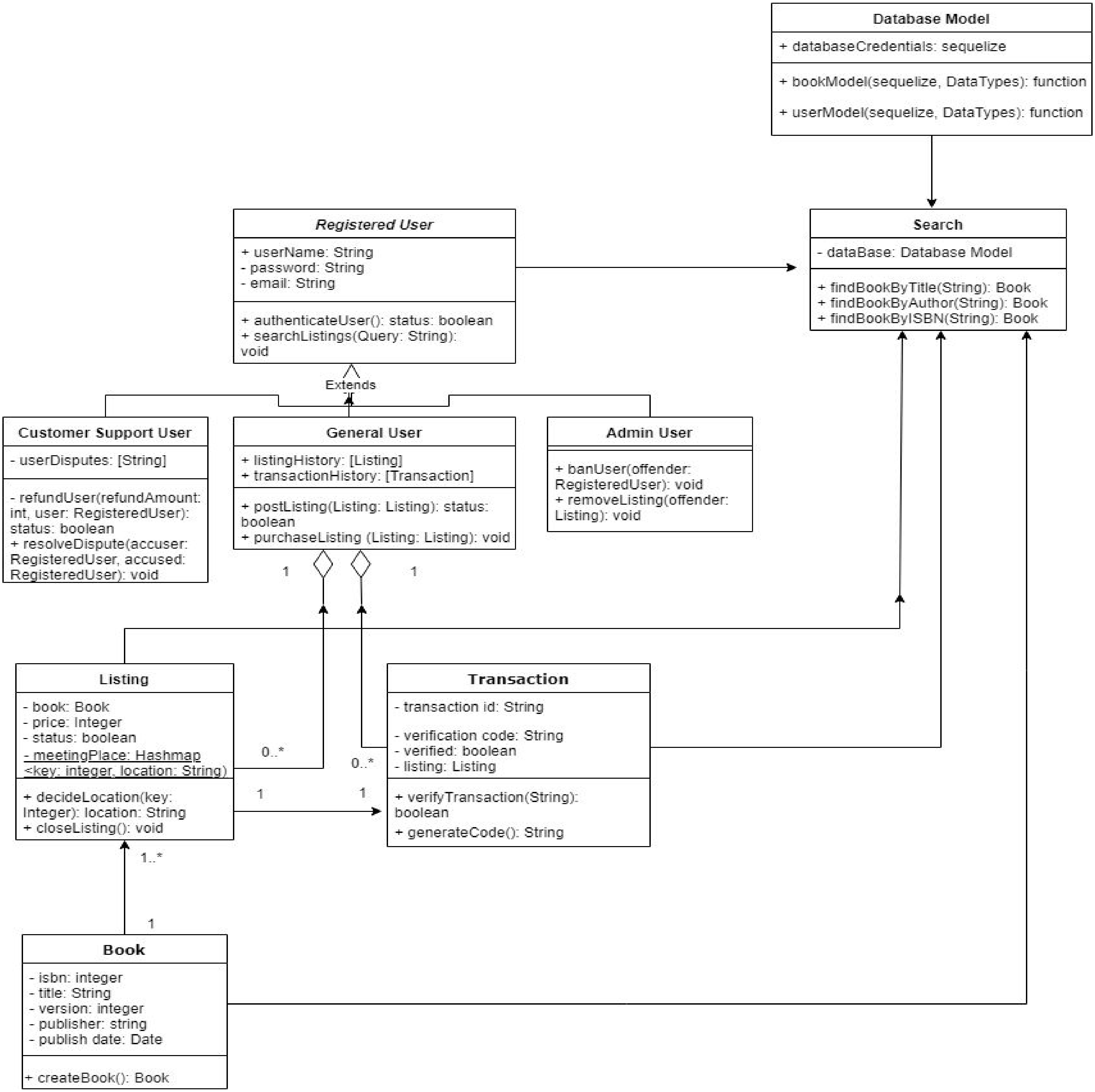
      ```
      db.book.findAll({
              where:{ title }).then ( books => books.getLister ({ order:
      rating });
      ```
- **Changed/Edited frameworks:**
    - Sequelize for creating our database API. For example:
        - Create a User Account:

          ```
          db.user.create({firstname, lastname, email, password});
          ```
        - Create a Book:

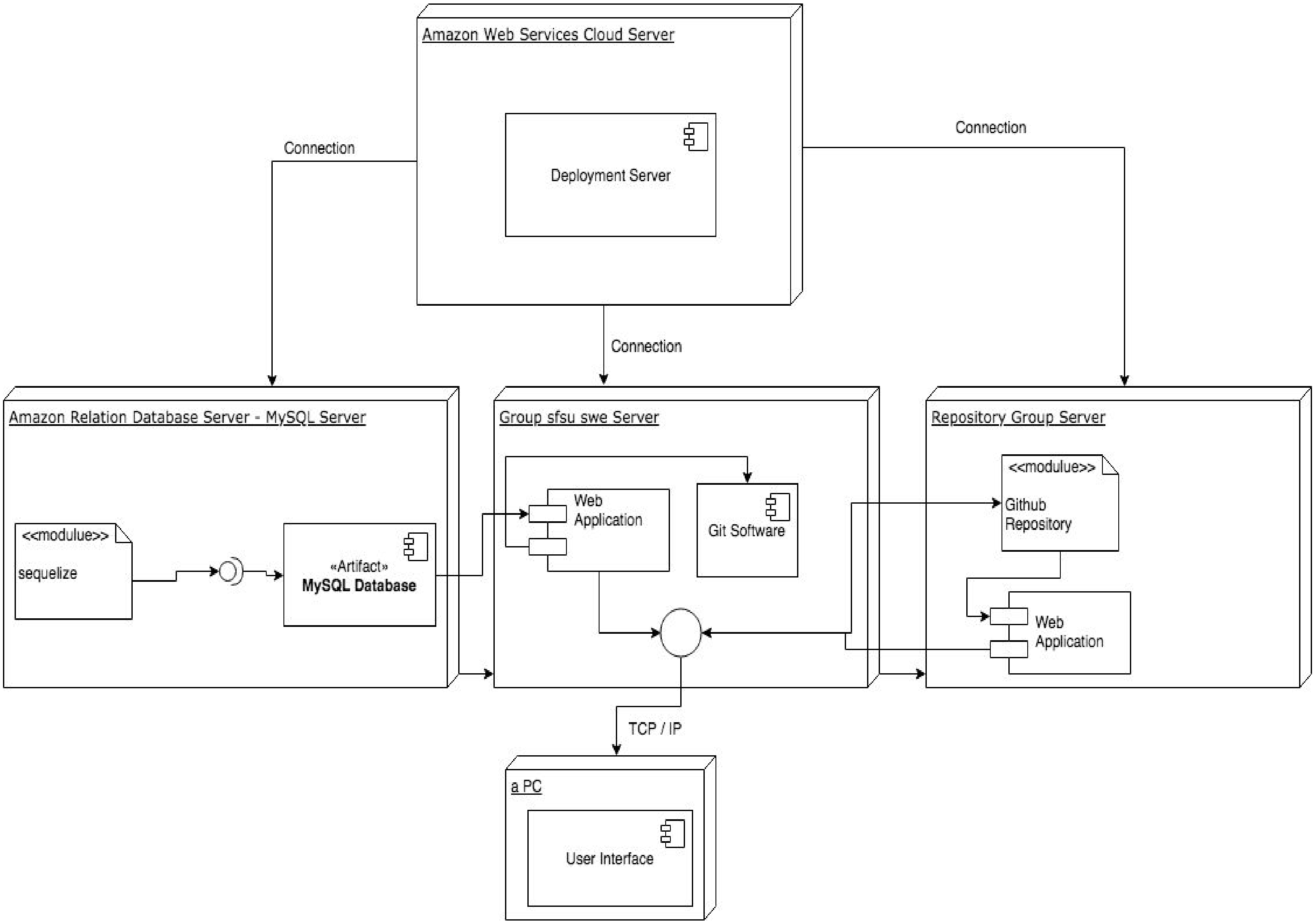
          ```
          db.book.create({title, isbn, author, description});
          ```

5. High Level UML Diagrams

a. Class Diagrams


Database Model
+ databaseCredentials: sequelize
+ bookModel(sequelize, DataTypes): function
+ userModel(sequelize, DataTypes): function
Search
- dataBase: Database Model
+ findBookByTitle(String): Book
+ findBookByAuthor(String): Book
+ findBookByISBN(String): Book
Registered User
+ userName: String
- password: String
- email: String
+ authenticateUser(): status: boolean
+ searchListings(Query: String): void
Extends
Customer Support User
- userDisputes: [String]
- refundUser(refundAmount: int, user: RegisteredUser): status: boolean
+ resolveDispute(accuser: RegisteredUser, accused: RegisteredUser): void
General User
+ listingHistory: [Listing]
+ transactionHistory: [Transaction]
+ postListing(Listing: Listing): status: boolean
+ purchaseListing (Listing: Listing): void
Admin User
+ banUser(offender: RegisteredUser): void
+ removeListing(offender: Listing): void
1
1
Listing
- book: Book
- price: Integer
- status: boolean
- meetingPlace: Hashmap <key: integer, location: String)
+ decideLocation(key: Integer): location: String
+ closeListing(): void
0..*
0..*
1
1
Transaction
- transaction id: String
- verification code: String
- verified: boolean
- listing: Listing
+ verifyTransaction(String): boolean
+ generateCode(): String
1..*
1
Book
- isbn: integer
- title: String
- version: integer
- publisher: string
- publish date: Date
+ createBook(): Book

B. Component and Deployment Diagrams



6. Key Project Risks

- Team Management
  - We will need to use some sort of management software to keep track of who is responsible for jobs and tasks, or we may accidently do someone else's task. It would be a waste of time if two people worked on something at the same time and essentially implemented the same feature twice.
  - Trello or Asana could be very useful in keeping track of these tasks. We will have to pick just one depending on the preference of the team.
- Legal Risks
  - One of our main legal risks is handling taxes and regulations; the main purpose of our website is to be a business, so we must ensure that we follow legal regulations and pay taxes on our revenue.
  - Another legal risk that we face is Amazon claiming ownership of our website. Since we are using a free AWS instance, Amazon can take our website from us under their terms of services. We can avoid this risk by keeping our site to a small scale.
- Technical Risks
  - One technical risk we face is managing our user's data; this includes their personal data as well as their payment data. We will adopt industry standards and refine our knowledge on software security, so that we can mitigate this risk.
  - Another 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 were using. We can avoid this risk by setting standards on how we write our code.
- Skill Risks
  - One skills risk that we face is learning the new technologies that we have chosen for this project. Not every team member is fully knowledgeable of how to use these technologies, and only one person on our team has worked with full-stack development, so most of our team is taking their first steps in learning this

development process. We can handle this risk by reading and watching tutorials to solidify our knowledge of the technologies we’re using in this project.

    - Another skills risk that we face in this project is that no one on our team has ever sold books before. We lack the domain knowledge on how to sell books and what makes a successful book seller. We can take on this risk by analyzing what makes other online book vendors successful, and by going above and beyond what they do.
- 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. One solution for this risk is to have a weekly meeting where everyone can meet together and work on their own tasks. This is especially key since our team seems to be much more productive when everyone has arrived.
    - Another schedule risk that we have face is learning all the technology we need to and completing all work before any deadlines. Learning how to effectively use the technology we need to, and implementing the features we would like to will take some time, so we have to make sure that all of that is done before the deadlines for the class. We can handle this risk by tackling our work as early on as possible, and by learning what we need to as early as possible.

7. Project Management

We are managing our team’s tasks by meeting up as much as we can in person, communicating through messaging applications, and by delegating tasks using Github Projects.

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 version 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 Slack since the creation of our team, but we migrated over to Discord due to the really convenient voice chat capabilities and the improved general chat. 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. Discord allows us to communicate quickly and effectively to handle issues and work on tasks as a team.

For the rest of the semester, we intend on following this same pattern of meeting up in person and trying to assign as much as we can there. We will also utilize Github Projects since a majority of the team is familiar with it. Github Projects will be useful for making sure everyone is doing only what they are assigned to do and not do someone else’s work too. We will follow our pattern of meeting up, communicating with discord on our own time, and utilizing Github Projects to work together as a team towards our goal of creating a great software product.