

SW Engineering CSC648/848 Spring 2021

iShareBooks

Team 04

Milestone#: Milestone 2

Date: 04/01/2021

History Table

Milestone	Date
M1V2	03/10/2021
M1V1	03/05/2021
M2V2	04/05/2021
M2V1	04/01/2021
M2V2	04/07/2021

Table of Contents

1. Data Definitions V2	4
2. Functional Requirements V2	6
3. UI Mockups and Storyboards	11
4. High level database architecture and organization	22
5. High Level APIs and Main Algorithms	27
6. High Level UML Diagram	29
7. High Level Application Network and Deployment Diagrams	30
8. Identify actual key risks for your project at this time	31
9. Project Management	33
10. Detailed List of Contributions	34

Team Information

Team Member Name	Role	Email
Yangesh KC	Team Lead/Documentation Expert	ykc@mail.sfsu.edu
Danish Siddiqui	FrontEnd Lead	dsiddiqui@mail.sfsu.edu
Zaid Alkhatib	BackEnd Lead	zalkhatib@mail.sfsu.edu
Aryanna Brown	FrontEnd Member	abrown22@mail.sfsu
Mark Jovero	BackEnd Member / Documentation Expert	mjovero@mail.sfsu.edu
Pramod Khatri	FrontEnd Member	pkhatri1@mail.sfsu.edu
Abishek Neralla	BackEnd Member	aneralla@mail.sfsu.edu
Yuhua Wu	FrontEnd Member	ywu23@mail.sfsu.edu

1. Data Definitions V2

1. **General User:** A user who can generally look up the website and has the least privilege to use website features. A general user may **register** to gain additional privilege
 - 1.1. **Registration:** A general user shall be able to create an account.
 - 1.2. **Account Contains:**
 - 1.2.1. **Username:** Needs a username to create an account.
 - 1.2.2. **Email:** Needs a unique email to create an account.
 - 1.2.3. **Password:** Needs a password that gets hashed for security purposes.
 - 1.2.4. **Accept Terms of Use:** Must select this to create the account.
2. **Registered User:** A user who can look up the website and has the most privilege to use website features. A general user shall be able **login** in order to use all of these features and services.
 - 2.1. Can login to the system.
 - 2.1.1. **Email:** Needs to login with a unique email.
 - 2.1.2. **Password:** Needs to provide a password.
 - 2.2. Has ability to start transactions for textbooks.
 - 2.3. **Buyers:** All registered users are buyers by default. They can buy textbooks.
 - 2.4. **Sellers:** Once a user posts a book for sale, they are considered a seller as long as they have at least 1 book for sale.
 - 2.5. **Profile:** A profile contains **username, email, ratings, and recent transactions.**
 - 2.5.1. **Username:** The username of a logged in user.
 - 2.5.2. **Email:** The unique email of a logged in user.
 - 2.5.3. **Ratings:** The ratings given to this registered user by other registered users.
 - 2.5.4. **Recent Transactions:** The overall transactions made by this registered user.
3. **Textbook:** The item that is sold/traded by sellers, and bought by buyers/registered users.
 - 3.1. **On Sale Textbooks:** Books can be placed for other registered users to buy.
 - 3.2. **Traded Textbook:** Books can be traded for other books.
 - 3.3. **Auctioned Textbooks:** Sellers can set their own price, and buyers can place offers for it.
 - 3.4. **Free Textbooks:** Buyers can buy free textbooks.
 - 3.5. **Images:** Textbook image is a single image that contains the cover only.
 - 3.5.1. **Size:** 300 x 300 & 500 x 500
 - 3.5.2. **Storage:** Local storage and database
 - 3.5.3. **Format:** jpg/png

4. **Textbook Posts:** A textbook post shall be posted by a registered user who is a seller, trader, for auction, or for free.
 - 4.1. **Contains:** Title, Author, Image, Condition, Price.
 - 4.1.1. **Title:** The title of a textbook
 - 4.1.2. **Author:** The author of a textbook.
 - 4.1.3. **Image:** The cover image of a textbook.
 - 4.1.4. **Price:** The price of a textbook (for sale and for auction only).
5. **Payment Type:** A payment type shall be selected by all the registered users at the time of transaction.
 - 5.1. **Types Include:** Credit Card, Bank Account, Debit Card.
 - 5.1.1. **Credit Card:** Payment shall be paid with a credit card.
 - 5.1.2. **Bank Account:** Payment shall be paid with a bank account.
 - 5.1.3. **Debit Card:** Payment shall be paid with a debit card.
6. **Receipt:** A proof-of-purchase document after completion of transaction.
 - 6.1. **Receipt Includes:** Username, Price, Item info.
 - 6.1.1. **Username:** The user's name who bought an item.
 - 6.1.2. **Price:** Price of the item.
 - 6.1.3. **Item info:** The information of an item (textbook).
7. **Chat Bot:** An assistance bot which shall be able to answer basic website related questions to all kinds of users.
 - 7.1. **Speech:** The chatbot shall be able to speak back.
 - 7.2. **Text:** The chatbot shall be able to reply back with a text.
8. **Tutoring Subscription:** A tutoring subscription is a way of making revenue from this website and shall only be subscribed by the registered users.
 - 8.1. **Includes:** Video, Username, Title.
 - 8.2. **Video:** A tutorial video shall be posted by the registered users.
 - 8.2.1. **Size:** 1000px at full screen, 700 px as default.
 - 8.2.2. **Format:** .mp4
 - 8.3. **Username:** The username of whoever posted the tutorial video.
 - 8.4. **Title:** The title of the tutorial video.

2. Functional Requirements V2

Priority 1

1. General User:

- 1.1. A general user shall be able to search books without creating an account.
- 1.2. A general user shall be able to register for an account.
- 1.3. A general user shall visit most pages of websites.
- 1.4. A general user shall ask the live chat box for website-related questions.
- 1.5. A general user shall be able to search for the textbooks available for the auction.
- 1.6. A general user shall be able to search for the textbooks available for the trade.
- 1.7. A general user shall be able to search for the textbooks available for free.
- 1.8. A general user shall be able to filter searches.
- 1.9. A general user shall be able to view the privacy policy.
- 1.10. A general user shall be able to search a textbook with a department.
- 1.11. A general user shall be able to select the textbooks without creating an account.
- 1.12. A general user shall be able to view only the inquiries related to the post.
- 1.13. A general user shall comply with the website terms and conditions.

2. Registered User

- 2.1. A registered user shall be able to log in to the website account.
- 2.2. A registered user shall be able to log out from the website account.
- 2.3. A registered user shall be able to edit his/her account profile.
- 2.4. A registered user shall be able to list books for sale.
- 2.5. A registered user shall be able to list books for trade.
- 2.6. A registered user shall be able to list books for a free giveaway.
- 2.7. A registered user shall be able to buy books from other sellers.
- 2.8. A registered user shall be able to trade books with other sellers.
- 2.9. A registered user shall be able to get books free of cost from other sellers.
- 2.10. A registered user shall be able to edit/modify the book listing.
- 2.11. A registered user shall be able to reply to inquiries from the registered users.
- 2.12. A registered user shall be able to ask the live chatbot for website-related questions.
- 2.13. A registered user shall be able to filter searches.
- 2.14. A registered user shall be able to search books by department names.
- 2.15. A registered user shall have a rating.
- 2.16. A registered user shall be able to get a receipt of their transactions.
- 2.17. A registered user shall be able to post a tutorial video for subscription.
- 2.18. A registered user shall be able to purchase a subscription for tutorial video.

3. Profile

- 3.1. A profile shall be created for all registered users.
- 3.2. A profile shall be created when the account is made.
- 3.3. A profile rating shall be visible to all registered users.
- 3.4. A profile shall be deleted from the system.
- 3.5. A profile shall have a 5 star rating of the registered users by default.
- 3.6. A profile shall be edited by the registered user.
- 3.7. A profile shall be keeping the records of the textbooks listed.
- 3.8. A profile shall be keeping the records of the textbooks bought.
- 3.9. A profile shall be added to favorites/save by other registered users.
- 3.10. A profile's recent transaction shall be visible to other registered user
- 3.11. A profile shall be able to see their favourites.

4. Payment Type

- 4.1. A payment type shall be a bank account.
- 4.2. A payment type shall be a debit card.
- 4.3. A payment type shall be a credit card.
- 4.4. A payment type shall be selected by all buyers.
- 4.5. A payment type shall belong to each registered user.
- 4.6. A payment type shall be selected at the time of order confirmation.

5. Textbook

- 5.1. A textbook shall be posted on sale only by the registered users.
- 5.2. A textbook shall be accessible with the textbook name.
- 5.3. A textbook shall be accessible with the department name.
- 5.4. A textbook shall be posted for auctions by the registered user.
- 5.5. A textbook shall be posted for trade by the registered user.
- 5.6. A textbook shall be posted for free by the registered user.
- 5.7. A textbook shall be purchased by the registered users.
- 5.8. A textbook shall be publicly available to all registered users.
- 5.9. A textbook shall be publicly available to all general users.

6. Free Books

- 6.1. A free textbook shall be listed as free of cost.
- 6.2. A free textbook shall be visible to registered users.
- 6.3. A free textbook shall be visible to the general user.
- 6.4. A free textbook shall have a finalized purchase.

7. Traded Books

- 7.1. A traded Book shall be available for all general users.
- 7.2. A traded textbook shall be exchanged between registered users.

- 7.3. A traded textbook shall be in use or in new condition.
- 7.4. A traded textbook shall be listed as available for trade.
- 7.5. A traded textbook's trade shall be finalized.
- 7.6. A traded textbook shall be removed after the trade is done.
- 7.7. A traded textbook shall be listed as available for trade.

8. Textbook Image

- 8.1. A textbook image shall be resized.
- 8.2. A textbook image shall be the image of a book.
- 8.3. A textbook image shall belong to each textbook post.
- 8.4. A textbook image shall be posted by the seller.
- 8.5. A textbook image shall be clearly visible to all users.
- 8.6. A textbook image shall be compatible in any device.

9. Receipt

- 9.1. A receipt shall be made for all purchased textbooks.
- 9.2. A receipt shall belong to each order of the textbook.
- 9.3. A receipt shall be uniquely made for an order.
- 9.4. A receipt shall be a part of the checkout process.

10. ChatBot

- 10.1. A chat bot shall be able to help all kinds of users.
- 10.2. A chatbot shall be able to speak or text.
- 10.3. A chatbot shall be able to navigate through the website pages.

11. Security Upgrades/Upkeep

- 11.1. Database security and Encryption.
- 11.2. Speed, Caching and Optimization.
- 11.3 UI/UX.

Priority 2

1. General User:

- 1.1. A general user shall be able to subscribe to the website via email.

2. Registered User:

- 2.1. A registered user shall be able to report general users and their illegitimate activities to the website support team.

- 2.2. A registered user shall be able to report the registered users and their illegitimate activities to the website support team.
- 2.3. A registered user shall be able to block general users.
- 2.4. A registered user shall be able to block the registered users.
- 2.5. A registered user shall be able to invite their friends by message.
- 2.6. A registered user shall be able to subscribe to the website via email.

3. Profile

- 3.1. A profile shall be tracking the inventory for all posts.
- 3.2. A profile shall be editable for all of its fields.

4. Payment Type

- 4.1. A payment type shall be selected in real time.
- 4.2. A payment type shall be accepting real card numbers.

5. Textbook

- 5.1. A textbook shall be posted on auction by the registered users.
- 5.2. A textbook shall be searched by ISBN number.
- 5.3. A textbook shall be returned to the seller.

6. Free Books

- 6.1. A free textbook shall be claimed on a first come first serve basis.
- 6.2. A free textbook shall be removed after a registered user has claimed it.

7. Traded Books

- 7.1. A traded textbook shall meet the seller's requirements.

8. Textbook Image

- 8.1. A textbook image shall be an appropriate image.

9. Receipt

- 9.2. A receipt shall be sent to the email.
- 9.3. A receipt shall be printed out via email.

10. ChatBot

- 10.1. A chatbot shall be able to work on all browsers.
- 10.2. A charbot shall be able to help with any website related question.

11. Tutorial Video

- 11.1. A tutorial video shall be recorded by the registered users.

- 11.2. A tutorial video shall be available on search.
- 11.3. A tutorial video shall be posted by the registered users.
- 11.4. A tutorial video shall be subscribed by the registered users.

12. Auctioned Books

- 12.1. An auctioned book shall be posted on auction by the registered users.
- 12.2. A auctioned book shall be purchased by the highest bidder.
- 12.3. An auction book shall be available on search to all users.

Priority 3

1. General User:

- 1.1. A registered user shall be able to subscribe for a website membership.
- 1.2. A general user shall be able to search for books with any keyword.

2. Registered User:

- 2.1. A registered user shall be able to subscribe for a membership.
- 2.2. A registered user shall be able to live chat with other registered users.
- 2.3. A registered user shall be able to make an offer for auction.
- 2.4. A registered user shall be able to save the books.
- 2.5. A registered user shall be able to search for books with any keyword.

3. Profile:

- 3.1. A profile shall be reported by the registered users.

4. Subscriber:

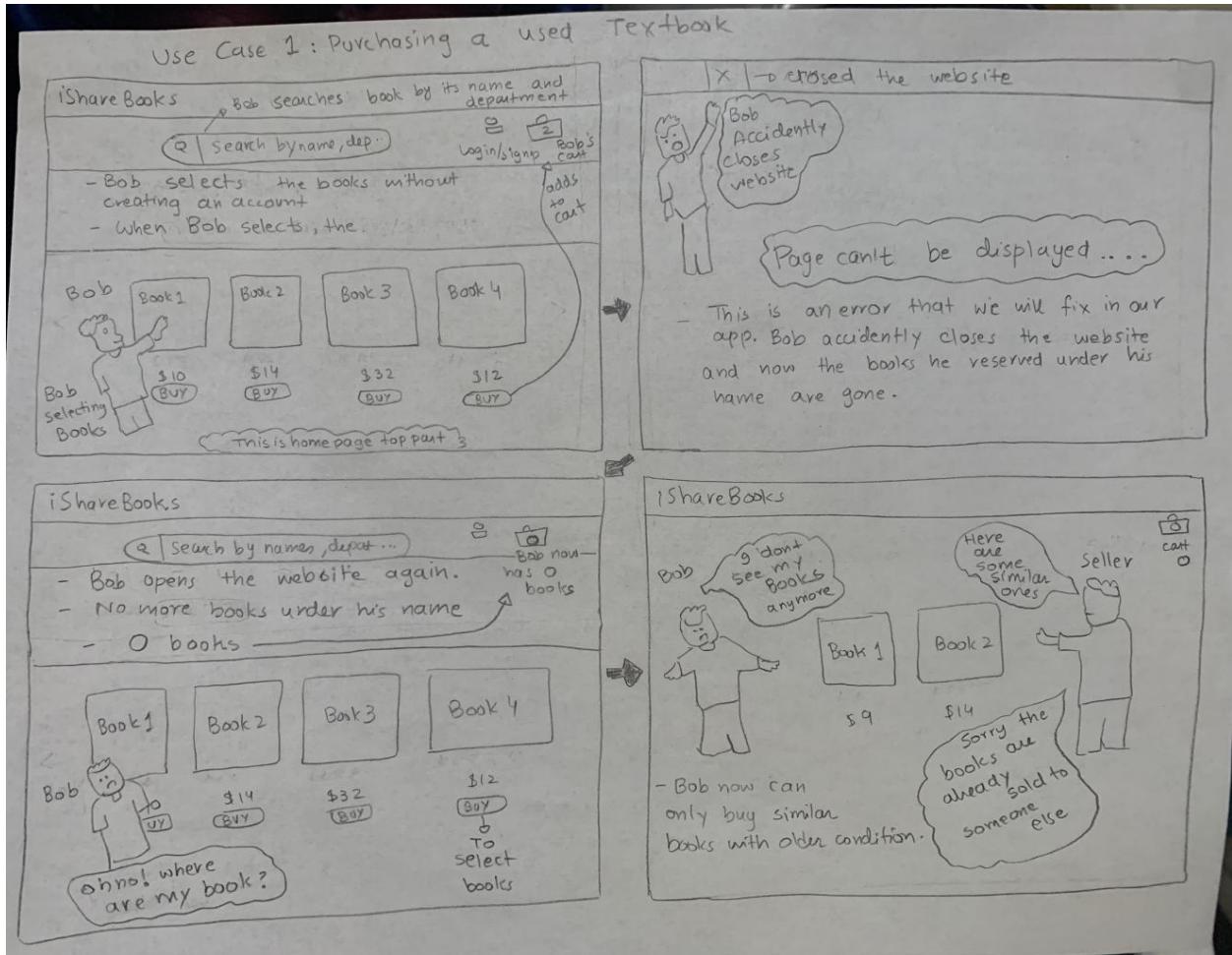
- 4.1. Subscriber shall have access to paid content.
- 4.2. Subscriber shall have access to tutoring videos.
- 4.3. Subscriber shall have a membership type.

5. Security Upgrades/Upkeep

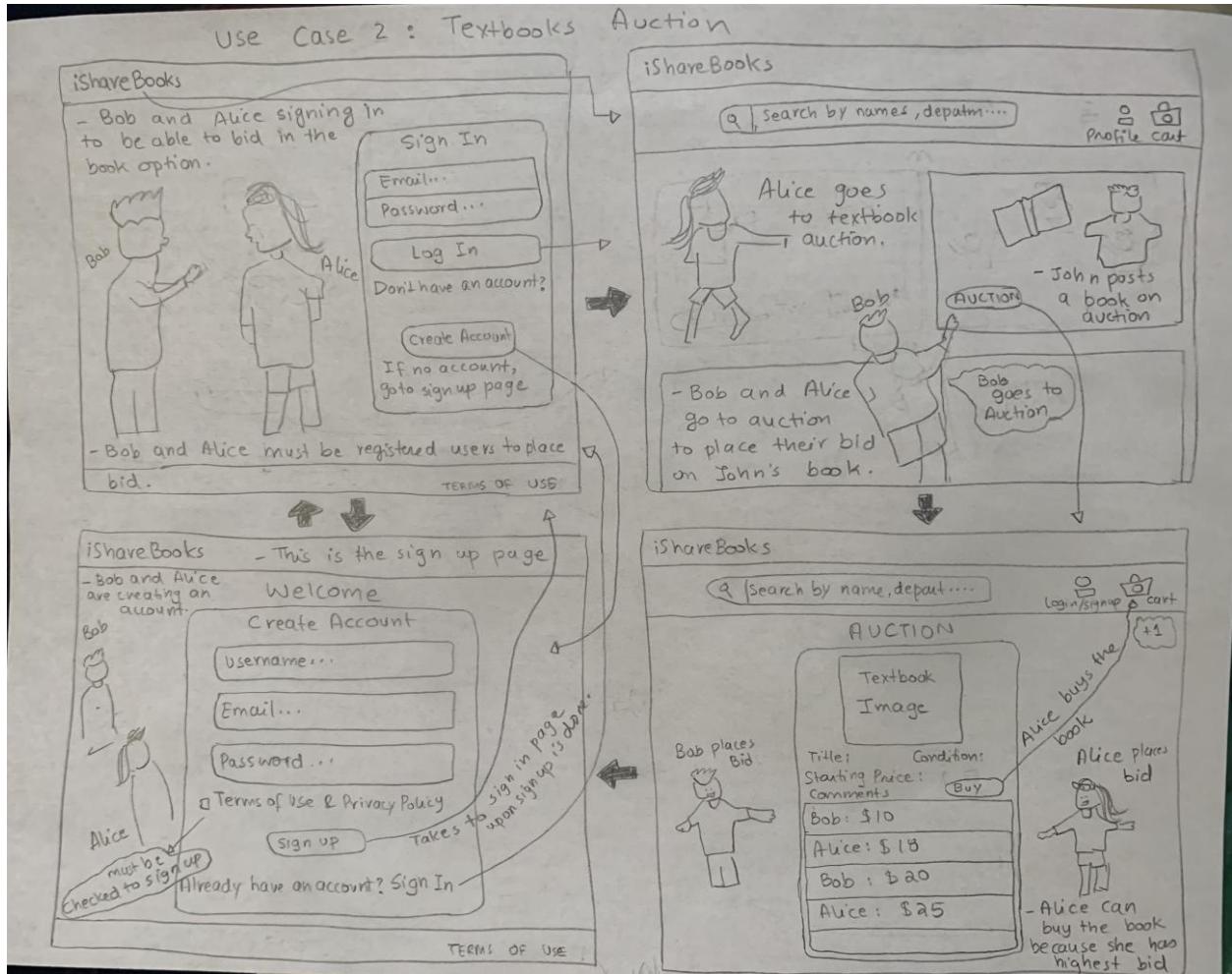
- 5.1. Infrastructure Upgrades
- 5.2. Feature Upgrades

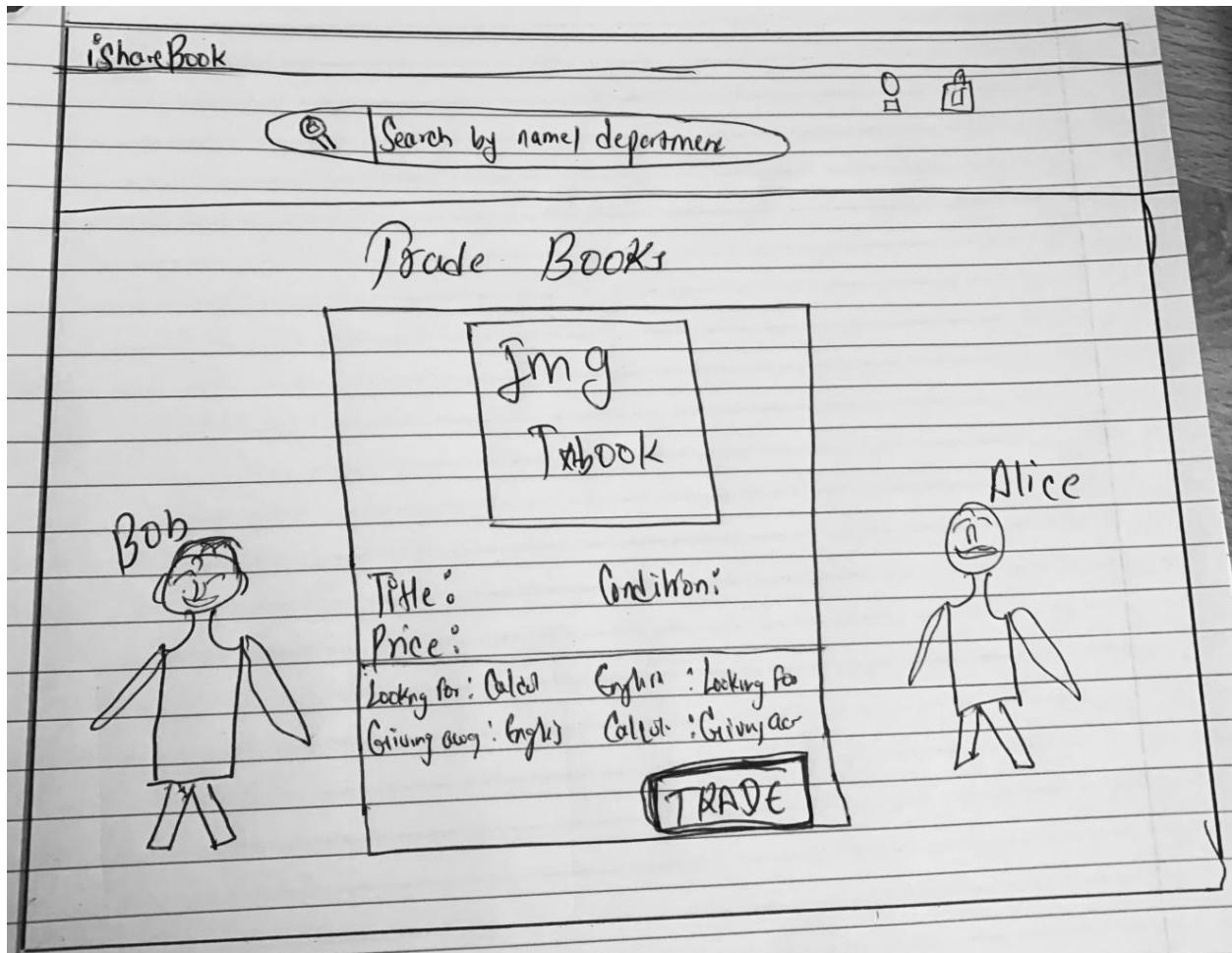
3. UI Mockups and Storyboards

Use Case 1:

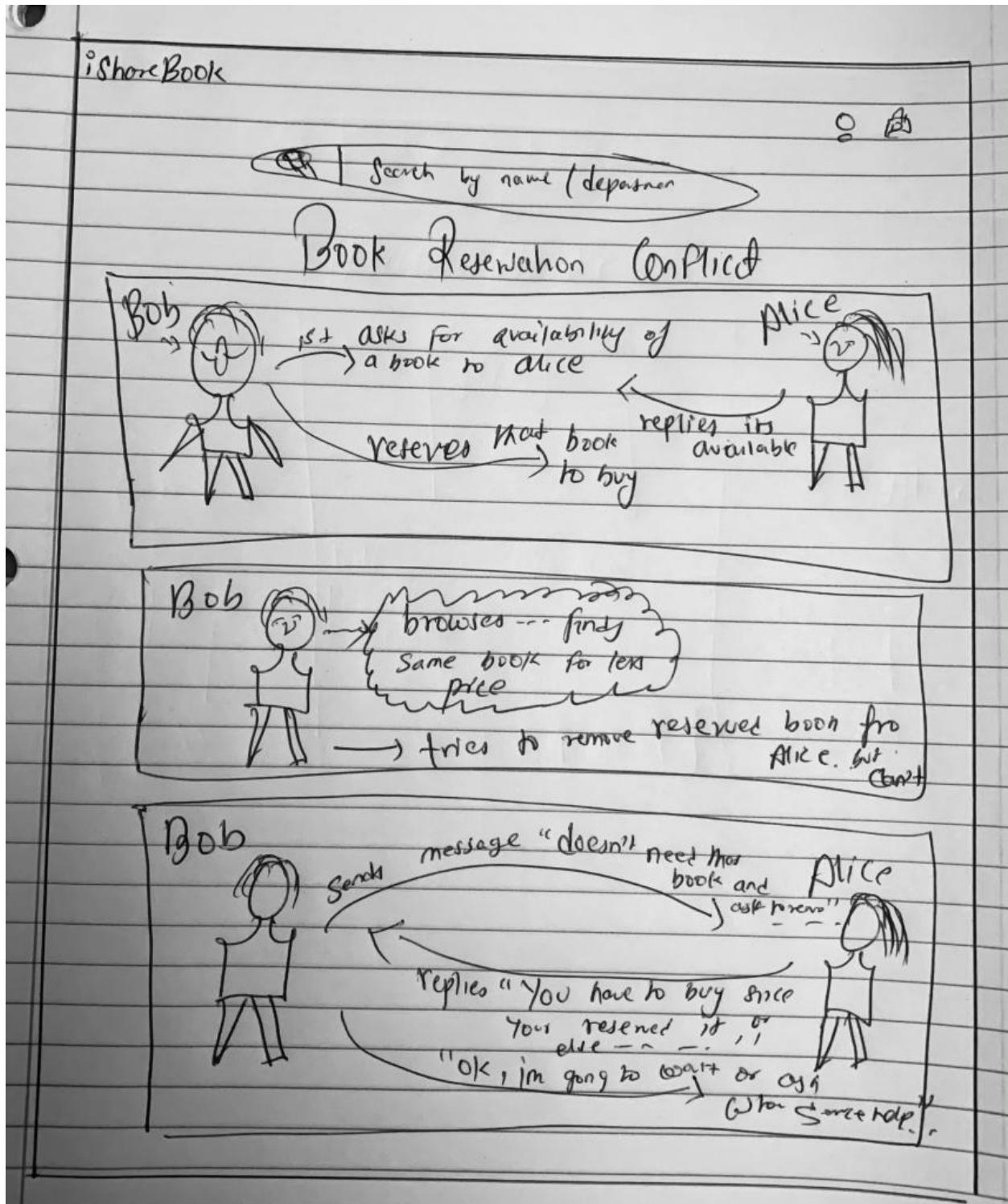


Use Case 2:

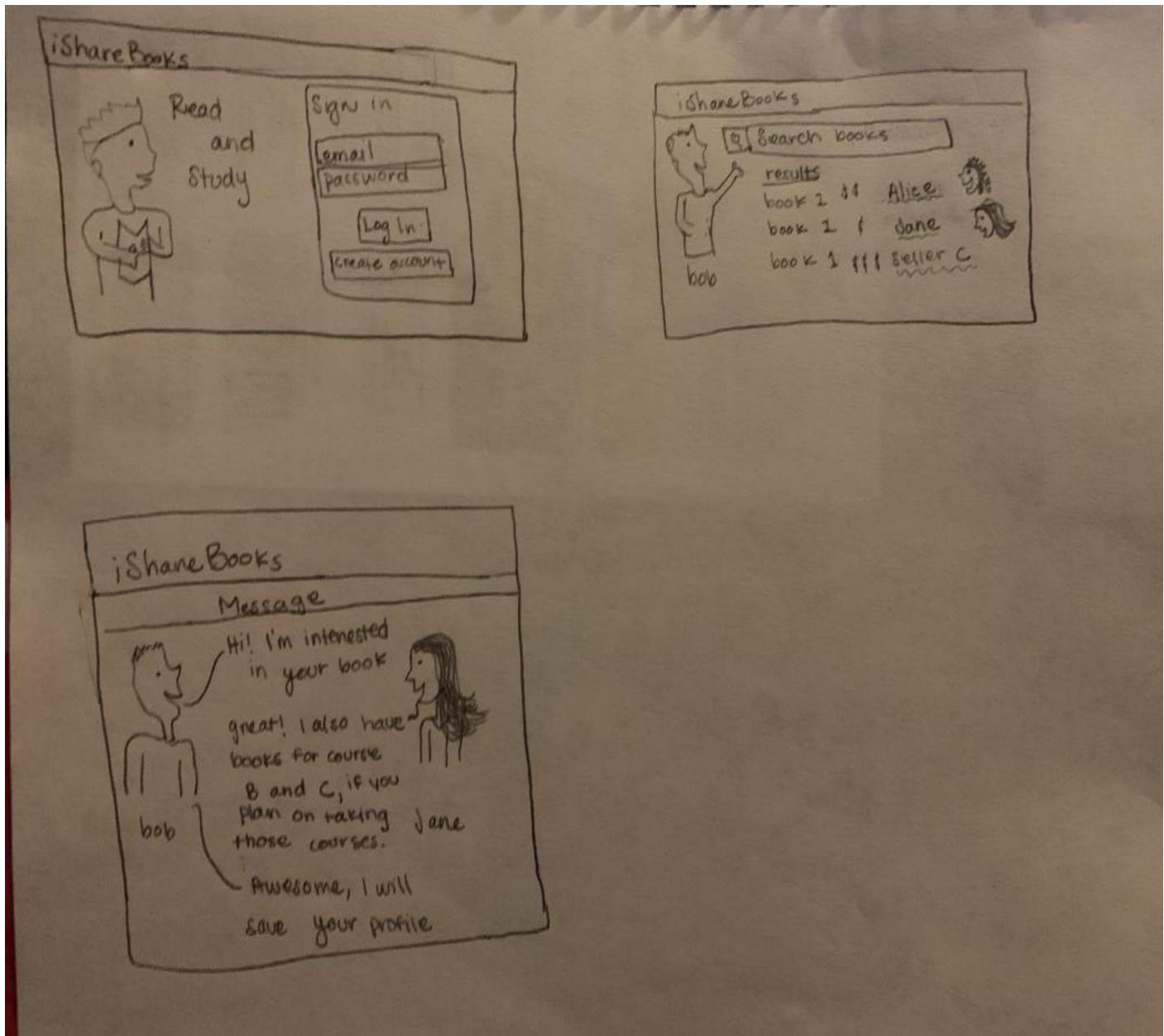


Use Case 3:

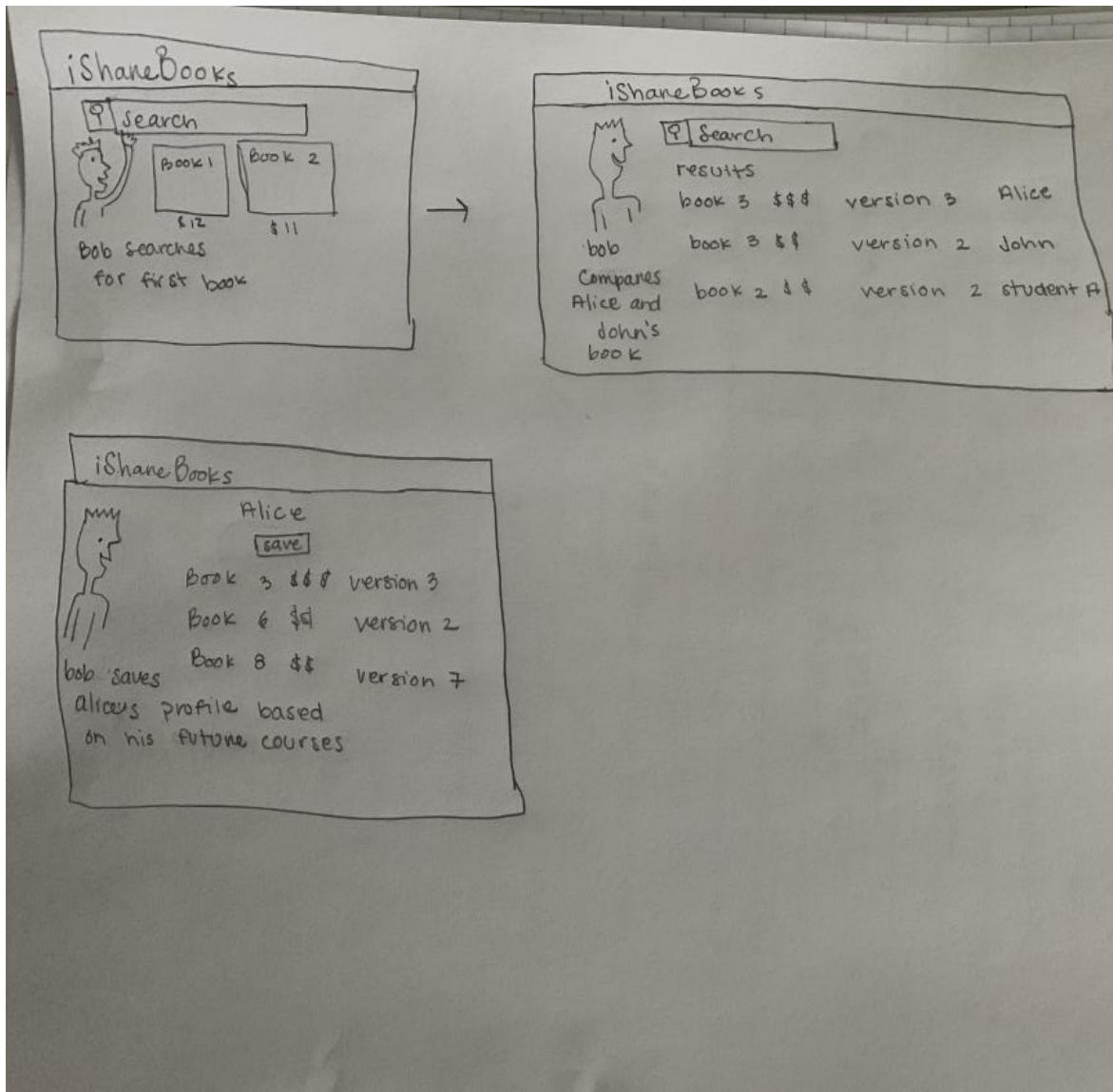
Use Case 4:



Use Case 5:



Use Case 6:

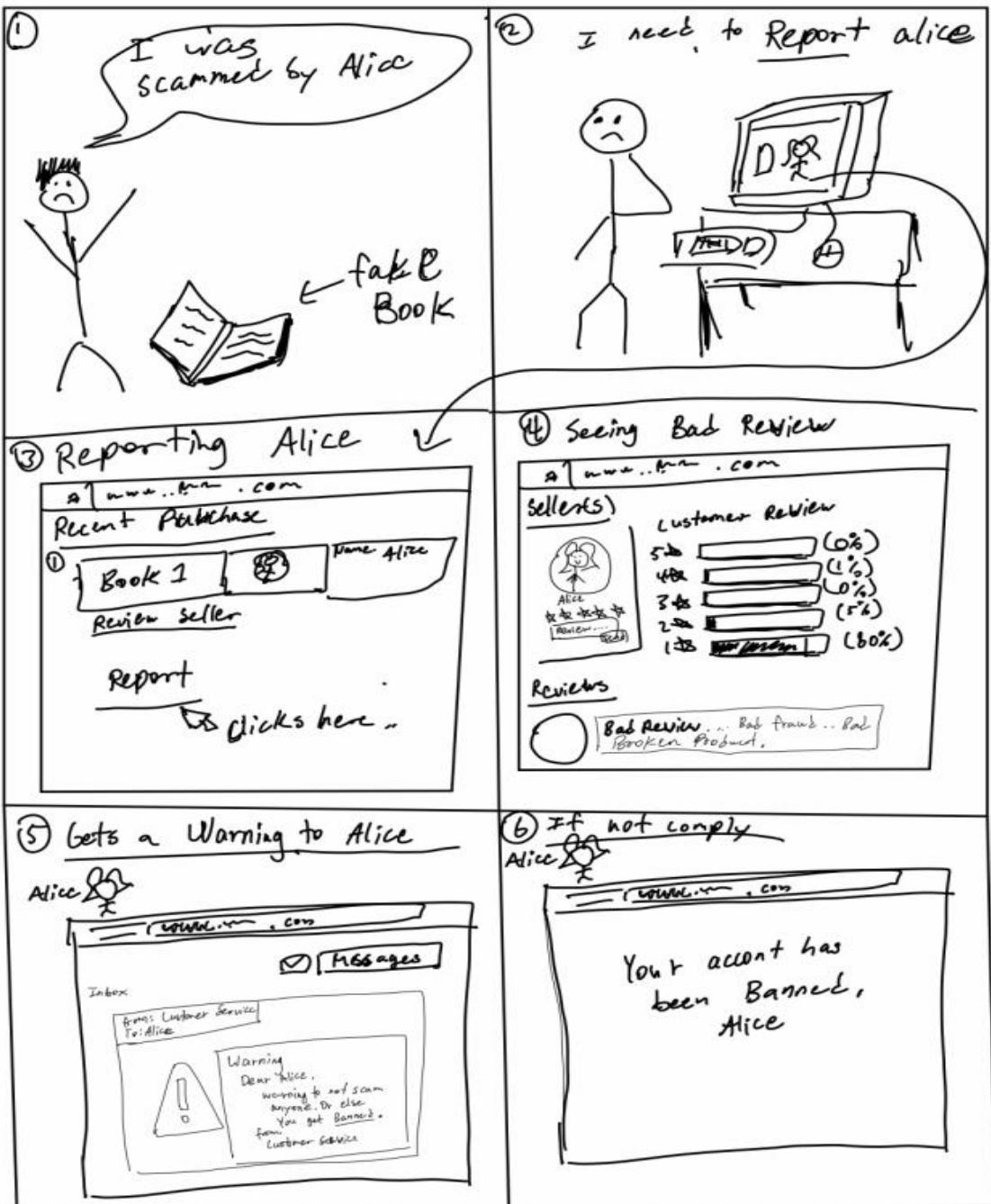


Use Case 7:

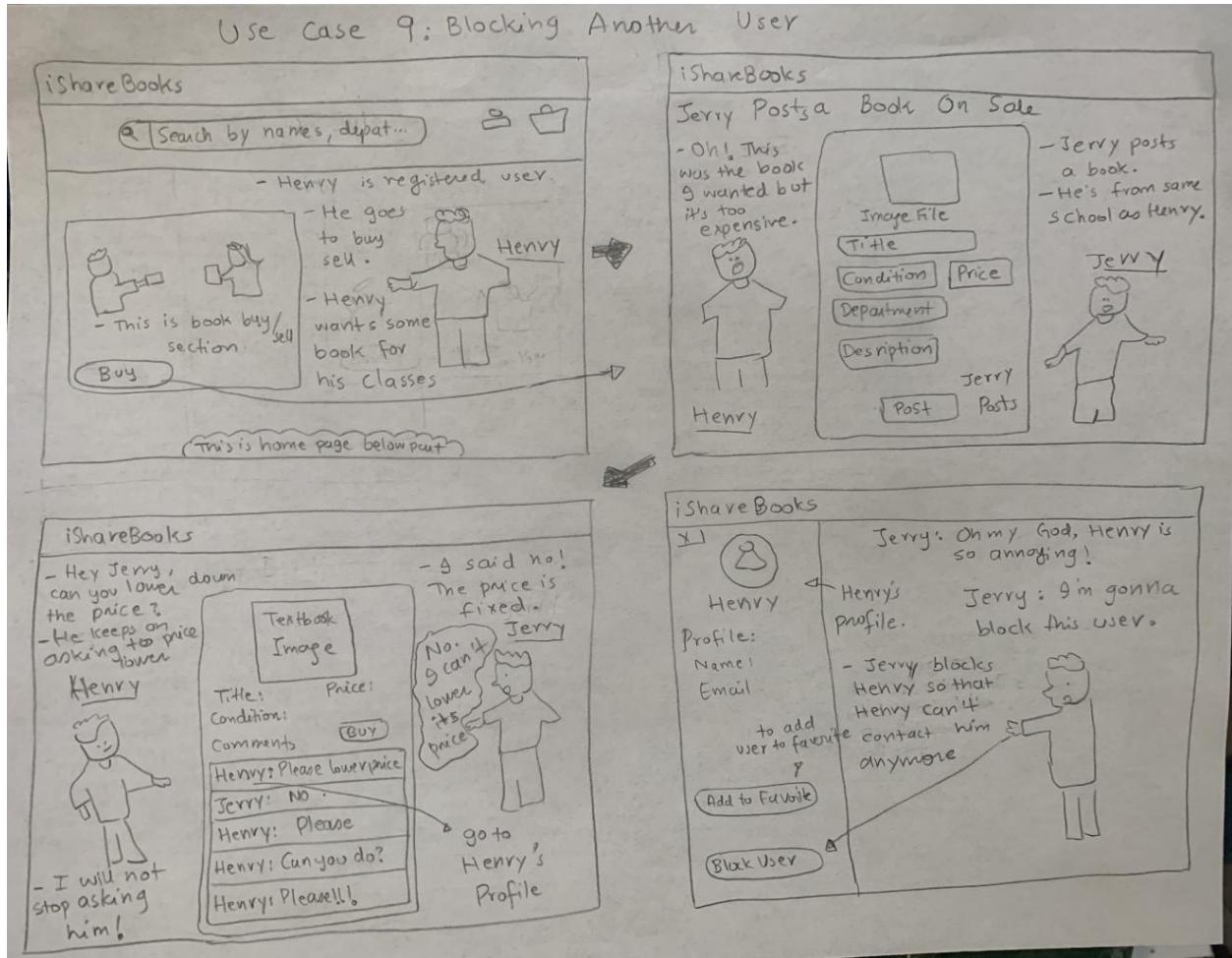
Use
Case 7

Use Case 8:

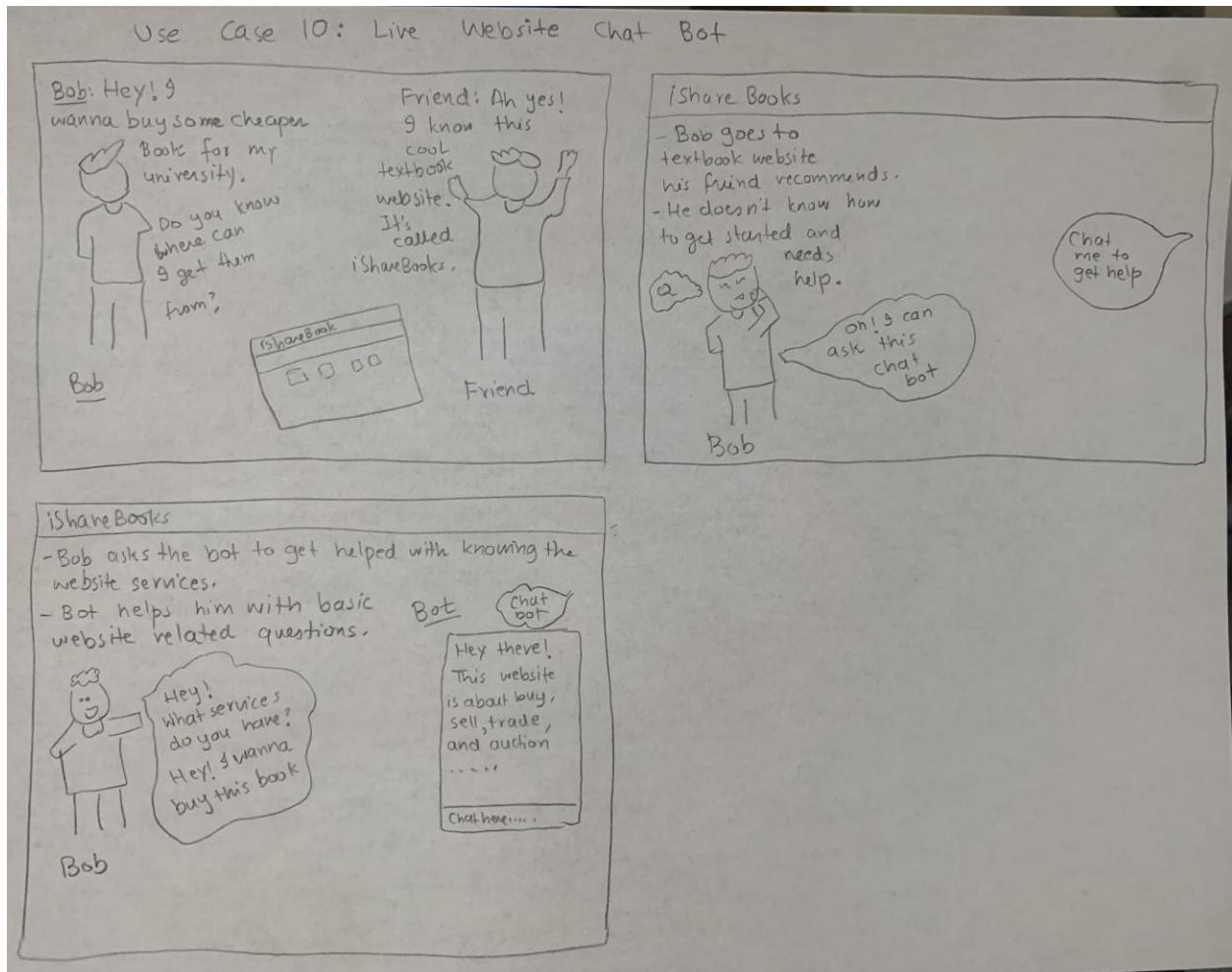
Use
Case 8



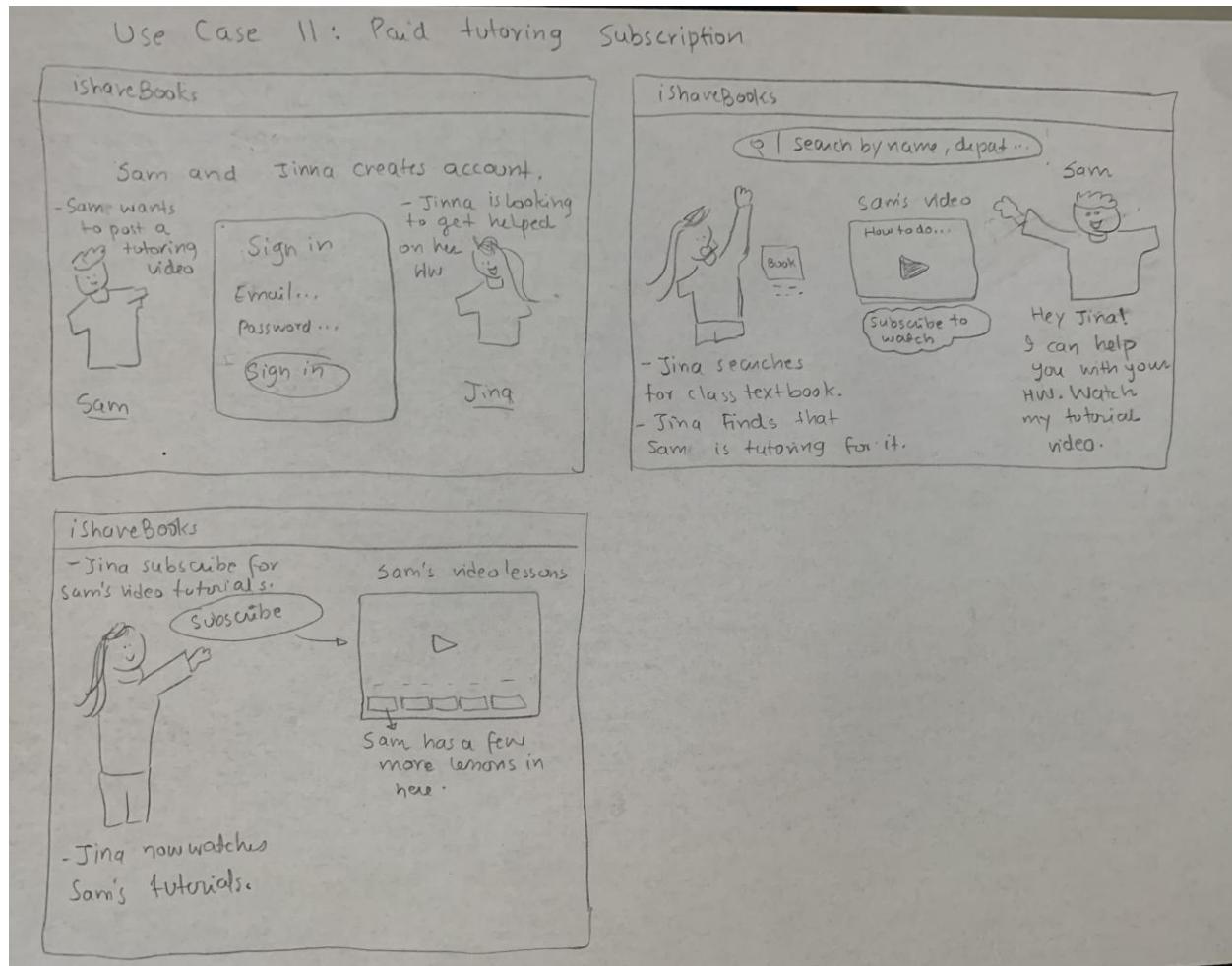
Use Case 9:



Use Case 10:



Use Case 11:



4. High level database architecture and organization

User Rating System

1. BUSINESS RULES:

1. Registered User

- a. A registered user shall be either a buyer or a seller.
- b. A registered user shall be able to log into one account.
- c. A registered user shall have a rating.

2. Buyer

- a. A buyer shall be able to buy zero or many books.
- b. A buyer shall be able to post many user ratings.
- c. A buyer shall be able to post a payment type.

3. Seller

- a. A seller shall be able to sell many books.
- b. A seller shall be able to post many user ratings.

4. Textbook

- a. A textbook shall be listed by many sellers.
- b. A textbook shall be sold by many sellers.
- c. A textbook shall be purchasable by one registered user.

5. Transaction

- a. A transaction shall be posted after the completion of one purchase.
- b. A transaction shall have many receipts.

6. Rating

- a. A user rating shall be posted by many buyers or sellers.
- b. A user rating shall belong to one registered user.

7. Payment Type:

- a. A payment type shall belong to one buyer.

8. Receipt:

- a. A receipt shall be sent to one seller.
- b. A receipt shall belong to one transaction.

2. ENTITIES

1. Registered User (STRONG)

- a. User_id: To identify each user's actions and login status (int).
- b. Email: To identify each user and save their credentials in the database (string &|| int).
- c. Password (string &|| int)

2. Buyer (WEAK)

- a. Buyer_id: To identify each buyer's actions and login status (int).
- b. Username: To be used for buying and transactions (string &|| int).

3. Seller (WEAK)

- a. Seller_id: To identify each seller's actions and login status (int).
- b. Username: To be used for selling and transactions (string &|| int).

4. Book (STRONG)

- a. Book_id: To identify each book uniquely (int).
- b. Title: To be saved in the database and displayed on the frontend and search results (string &|| in).
- c. Author: To be saved in the database and displayed on the frontend (string).
- d. Department: To be saved in the database and displayed on the frontend and search results (string).
- e. Isbn: Null for now, will be added when the seller posts a book (in).
- f. Cost: Cost that goes with each book to be sold (int/double).

5. Transactions (WEAK)

- a. Transaction_id: To identify an individual transaction (int).

6. Rating (WEAK)

- a. Rating_id: To identify each rating coming after the checkout is done (int).
- b. Rating: The rating count which is calculated by taking the average of all the ratings for a user (int).
- c. Comment: Optional but will be working as a feedback for each seller (string &|| int).
- d. Transaction_id: Will reference the transaction which was done before rating the seller (int).
- e. To: To a seller. (string &|| int)
- f. From: From a buyer (string &|| int).

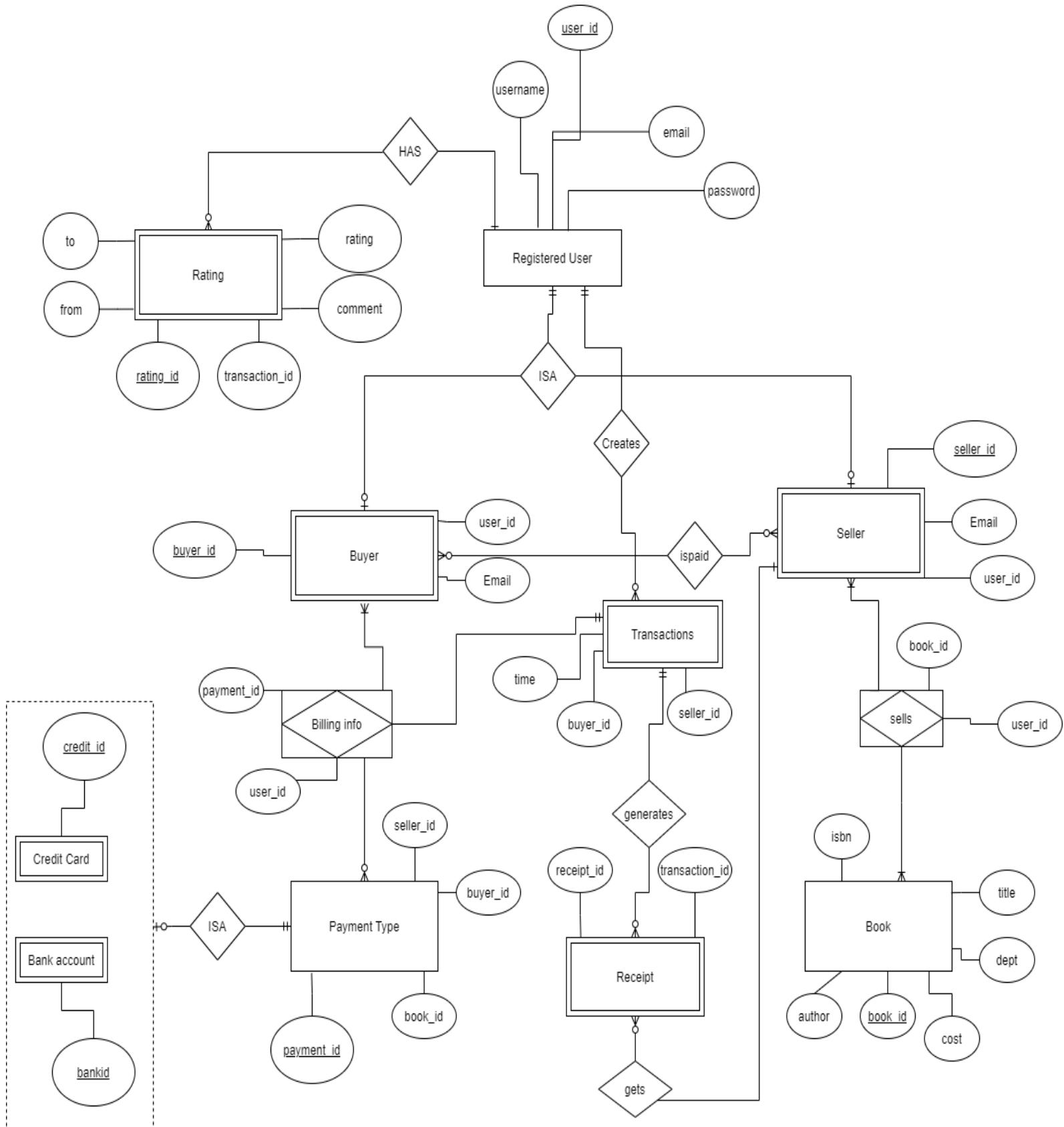
7. Payment Type (STRONG)

- a. Payment_id: To identify the payment type (int).
- b. Seller_id: To identify which seller is picking the payment type for the transaction (int).
- c. Type: The type of the payment type that the buyer will pick (string).

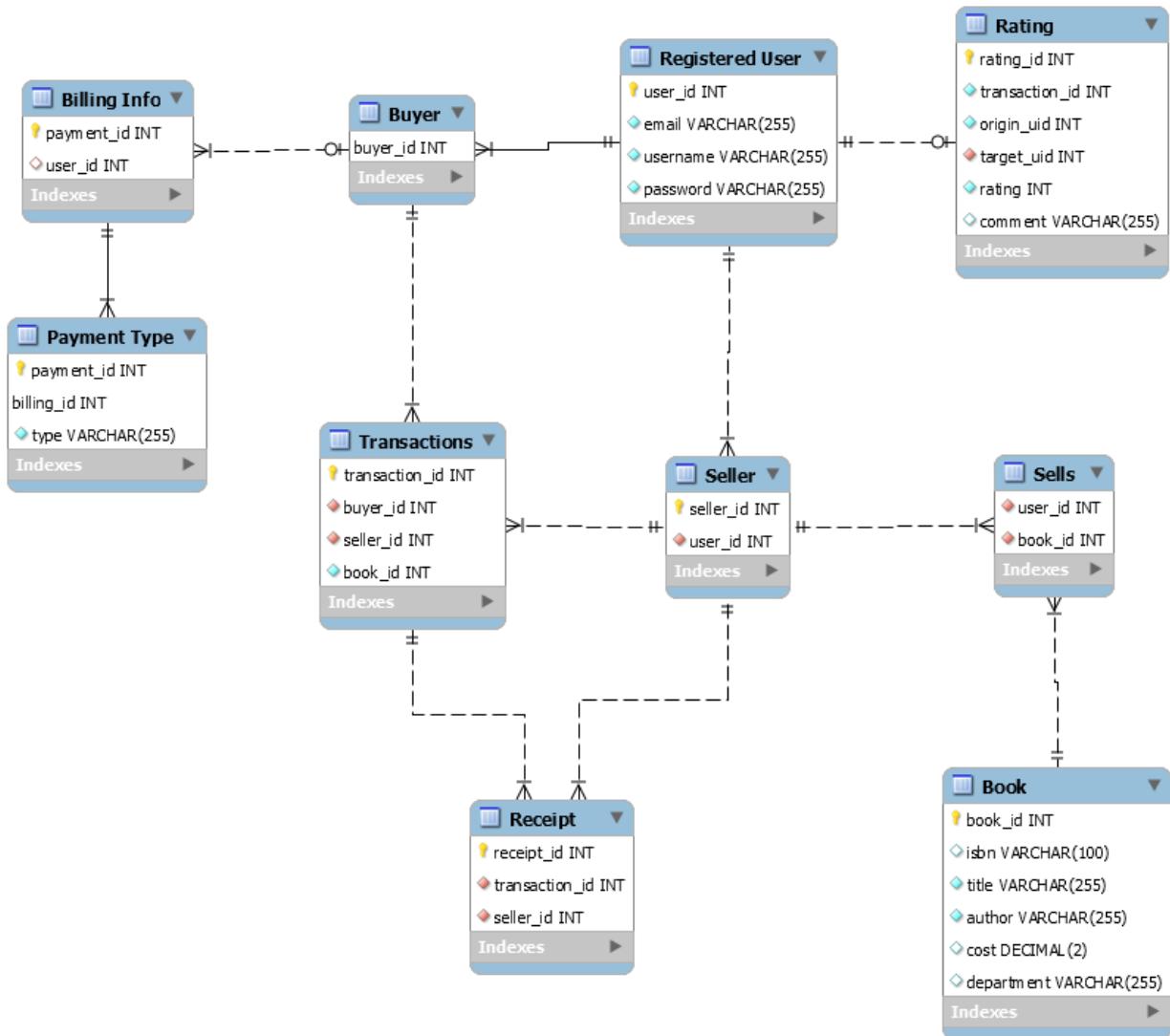
8. Receipt (WEAK)

- a. Receipt_id: To identify each receipt for a checkout (string).
- b. Transaction_id: Needs it to know which transaction the receipt is for (int).

3. ENTITY RELATIONSHIP DIAGRAM (ERD)



DATABASE MODEL



DBMS:

We will use MySQL to create the database because there are already multiple powerful tools such as MySQL Workbench that allow us to easily create and test the integrity of each entity relation.

Media Storage

1. Book Cover Image
 - a. Storage: MEDIUMBLOB
 - b. Types Supported: JPEG, PNG
 - c. Size Limit: 300KB

Search Filter & Architecture Implementation

1. Search Algorithm:
 - a. The search algorithm will consist of user input into the search bar. The user may input book attributes, departments, and/or seller's attributes. After parsing and confirming validity of the input, multiple queries will be sent to the database. All valid queries will return a list of book objects (**discuss). A book object will store the book's attributes (i.e., title, author, department, cost, and seller).

Organization of User Search Results:

1. Search results shall be organized in the order as follows:
 - b. By Title
 - c. By Author
 - d. By ISBN
 - e. By Cost
 - f. By Seller (username/rating)
 - g. By Department (English, Literature, History, CompSci, etc.)

DB Search Terms (by table):

1. Book
 - a. title: title of book
 - b. author - author of book
 - c. department - department of book
 - d. isbn - International Standard Book Number
 - e. cost: price of book
2. Ratings
 - a. user_id (target and source user_id)
3. Transactions
 - a. transaction_id

5. High Level APIs and Main Algorithms

Auth API

- **Register:**
 - **Post request:** When the user registers, the backend receives a post request with user's name, email and password. If the user inputs are valid, the backend saves the data in the database and sends a success message. Otherwise the backend responds with a failure message.
- **Login:**
 - **Post request:** When the user logs in, the backend receives a post request with the user's email and password. The backend responds with the login validation whether the user exists in the database, their email is correct, and/or their password is correct. If the user doesn't exist in the database, or the login credentials are invalid then the backend sends a failure response.

Selling/ Trade/ Free APIs

- **Posting Books:**
 - **Post request:** When the user posts a book, the backend receives a post request with the book's data. The backend saves the data in the database and responds with a success message.
 - **Get request:** When the backend receives a get request, the backend sends an array of data that is to be displayed on the frontend UI.

Buying API

- **Payment:**
 - **Post request:** When the user checks out, the backend receives a post request of the item and the amount and sends a success or failure message based on the user input.

User Issues API

- **Reporting a User:**
 - **Post request:** When a user reports another user, the backend receives the 2 users information and the data gets saved in the database to be reviewed. The user who reported receives a success message.
- **Website Bugs**
 - **Post Request:** When the user reports a bug, the backend receives the information and saves it and the database to be reviewed. The user who reported the bug receives a success message.

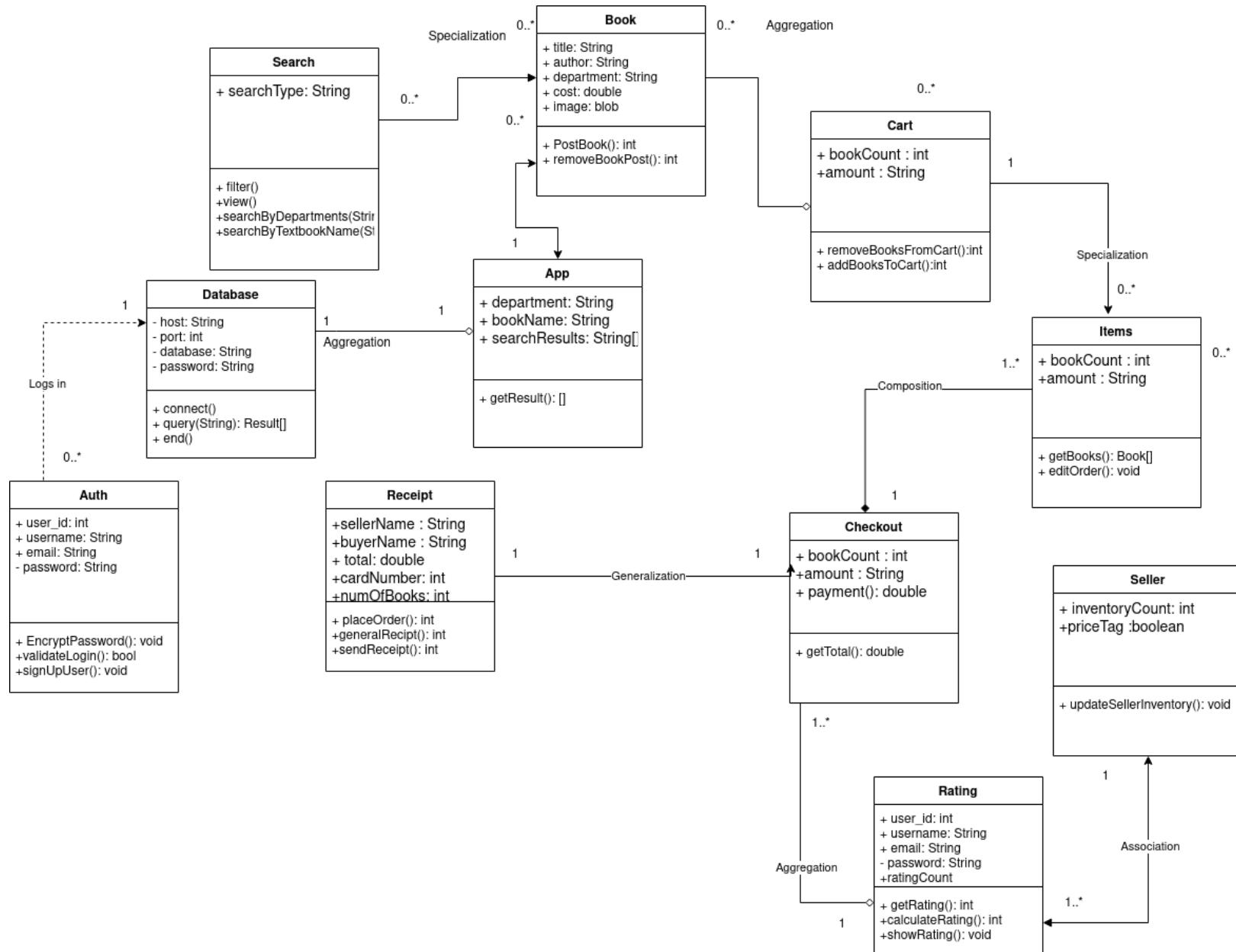
Books Search API

- **Search by Name**
 - **Get request:** When a user searches for a book by the book name. The backend will receive a get request. If the book exists in the database, the backend sends the book's data as a response. If the book doesn't exist in the database the backend sends a failed to find response.
- **Search by Department**
 - **Get request:** When a user searches for books by a department. The backend will receive a get request. If the department has books, the backend sends data of all the books of the department as a response. If the department doesn't have any books, the backend sends a failed to find response.

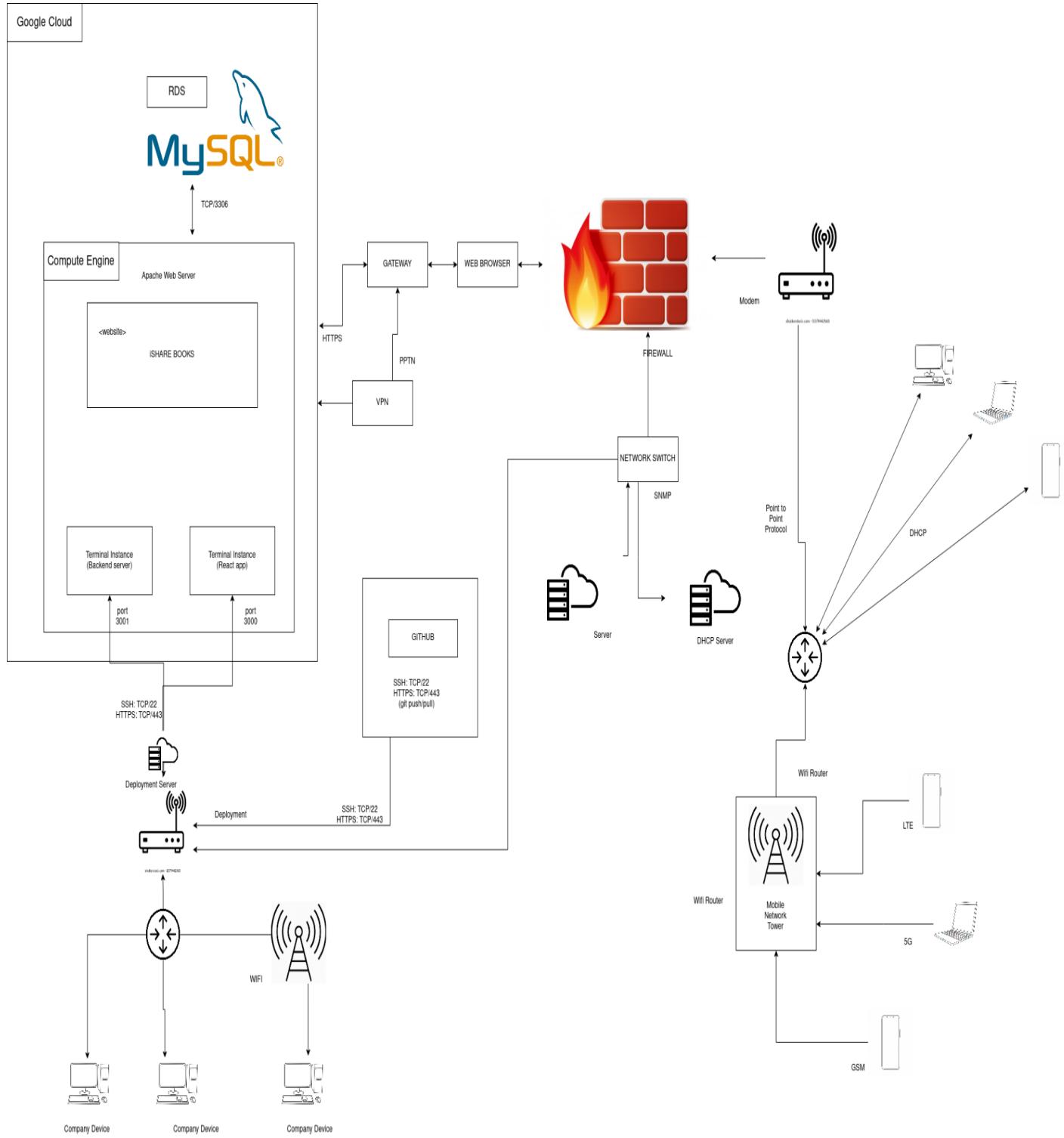
Rating System API

- **Rating a User**
 - **Post request:** When a buyer rates a seller, the backend will receive a post request of the rating data. The backend will save the data in the database and sends a success response.
 - **Get request:** When a buyer clicks on a seller's profile, the backend will receive a get request with the user's ID. The backend sends the user rating as a response

6. High Level UML Diagram



7. High Level Application Network and Deployment Diagrams



8. Identify actual key risks for your project at this time

1. skills risks (do you have the right skills):

- a. Developers don't have a team work spirit.

Solutions:

- i. Talk about how we work in a team.
- ii. Create a good management tool to make it easier to work as a team.

- b. Developers don't have enough coding skills:

Solutions:

- i. Offer them resources like online courses to learn.
- ii. Encourage them to ask questions when stuck.
- iii. Support them emotionally.

2. schedule risks (can you make it given what you committed and the resources)

- a. There is little to no time when everyone is available.

Solution:

- i. Use Trello to assign everyone a task so that everyone is on track. We will also use Discord to update everyone on what happened in a meeting.

3. technical risks (any technical unknowns to solve)

- a. Backend:

- i. The rating system is slightly more complex than we thought. More discussion with the backend team will be needed to figure out how exactly we will track each user's rating. Some questions to answer:

- 1. Will feedback (comments) be tracked?
- 2. What algorithm to run to compute ratings?
- 3. How to handle modifying ratings (i.e. change/delete rating)?

- ii. Database:

- 1. No backup: If a database breaks, we have no transaction logs.
- 2. No code to detect SQL injections.

Solution:

Research how to properly prevent these database issues and apply them to our database.

4. teamwork risks (any issues related to teamwork);

- a. Scheduling Conflicts.

5. legal/content risks (can you obtain content/SW you need legally with proper licensing copyright). Tell me how do you plan to resolve each actual risk you have. The key is to resolve risks as soon as possible. (Note that we will provide you with a basic set of images)

a. General

- i. When displaying our catalogue of available books for trade/sale, we may run into issues where the front cover image is subject to copyright laws.

Solution:

We will consider Fair Use laws to analyze the legality of our use of images to display book catalogue..

(<https://www.copyrightlaws.com/legally-using-images-in-libraries/>)

- ii. Fraud: We may contact major book distributors to request permission to use the front cover image to display our book catalogue.

1. No legal Terms of Use page and privacy, and data collection policy page.
2. No way to detect if someone might be selling the stolen books.
3. No way to detect copyright and stolen content.

Solution:

We will consider adding legal terms and conditions in our application along with Data collection policy. We will abide by the local, state, federal and international laws on digital content.

- iii. API use. We will use several API, some made in-house, others sourced from external developers.

Solution:

We will ensure each sourced API has the correct copyright usage for our application and that we have the author's permission if required.

9. Project Management

1. We are using Discord to manage our scheduling, make announcements, update team goals & status and to ensure everyone is up-to-date. We have created different channels for each section and have given everyone a role to make them easy to manage.
2. To manage tasks and their statuses, we will use Trello. We add what we will be doing and what we have done.
3. We used Calendly to plan office hours and schedule group meetings.
4. Used Zoom to interact and have video discussion, share screen.
5. We used Github to compare and share the code with the team. Everyone has their own branch and everyone pushes to their branch and at the end we merge to master so we don't have issues with the code.

10. Detailed List of Contributions

1. Yangesh KC -Team Lead

ykc@mail.sfsu.edu

As a Team Lead:

- Accountable for following M2 guidelines and ensuring all requirements are met.
- Directly supervise team members with daily regular interactions, including managing teams work load.
- Managed and oversaw team progress setting up deadlines, group discussion and resolving issues.

As a Team Member:

Milestone 2:

1. Main contributor to Network and Deployment Diagrams.
2. Partial Contribution to Identifying Key risks and planning mitigations.
3. Contributed to UML diagrams, Project Management Section.
4. Partial contribution to ERD and Database Model.

Vertical Prototype:

1. Implemented Suggestion Feature in the search bar.
2. Main Contributor to the Footer Section.
3. Partial contribution for Project Deployment.

2. Danish Siddiqui -Frontend Lead

dsiddiqui@mail.sfsu.edu

Milestone 2:

1. Partial contribution to the Data definitions, Mockups and Storyboards, UML and Network diagrams.
2. Contributed to peer review.

Vertical Prototype:

1. Front-end Vertical SW Prototype Architecture Design
2. Reviewed and Contributed to login/signup validation and implemented Landing page components.

3. Implemented Search Bar Functionality and Filter Components
4. Helped to resolve front-end css and routing issues.

3. Zaid Alkhatib -Backend Lead

zalkhatib@mail.sfsu.edu

Milestone 2:

1. Main Contributor to the High Level API and Algorithms section.
2. Partial contribution to Identifying risks, Network Diagrams and Project Management details.
3. Partial Contribution to ERD section

Vertical Prototype:

1. Build Login and Registration for the backend.
2. Created AWS database
3. Assisted Frontend to resolve data retrieval issue.

4. Mark Jovero -Backend Support

mjovero@mail.sfsu.edu

Milestone 2:

1. Contributed to high level architecture and design.
2. Partial Contribution to UML, Network Diagrams and Identifying Key Risks.

Vertical Prototype:

1. In charge of Project Deployment via AWS.
2. Main contributor to books.js
3. Contributed to backend image archive and rendering.
4. Assisted to debug and resolve Media Storage in the backend.

5. Abishek Neralla -Backend Member
aneralla@mail.sfsu.edu

Milestone 2:

1. Partial contribution to UML and Database Model Diagrams.
2. Contributed to group discussions and peer review.

Vertical Prototype:

1. Contributed to the services section on the front-end.
2. Implemented UI/UX for the services section.

6. Pramod Khatri -Frontend Member
pkhatri@mail.sfsu.edu

Milestone 2:

1. Partial contribution on Mockups and storyboards.
2. Partial Contributed to data definitions V2.

Vertical Prototype:

1. Implemented UI/UX and validation for the Login page.
2. Contributed to the FAQ section on the front-end.

7. Aryanna Brown -Frontend Member
abrown22@mail.sfsu.edu

Milestone 2:

1. Contributed to Mock-ups and storyboards.
2. Partial Contribution to data definitions V2.

Vertical Prototype:

1. Implemented UI/UX and validation for the Registration page.

8. Yuhua Wu -Frontend Member
ywu23@mail.sfsu.edu

Milestone 2:

1. Contributed to storyboard and UML group discussions.

Vertical Prototype:

1. Designed and Implemented the UI/UX services section.
2. Implemented the four different buttons for services section
3. Service section included responsive and interactive features