Project Milestone 4
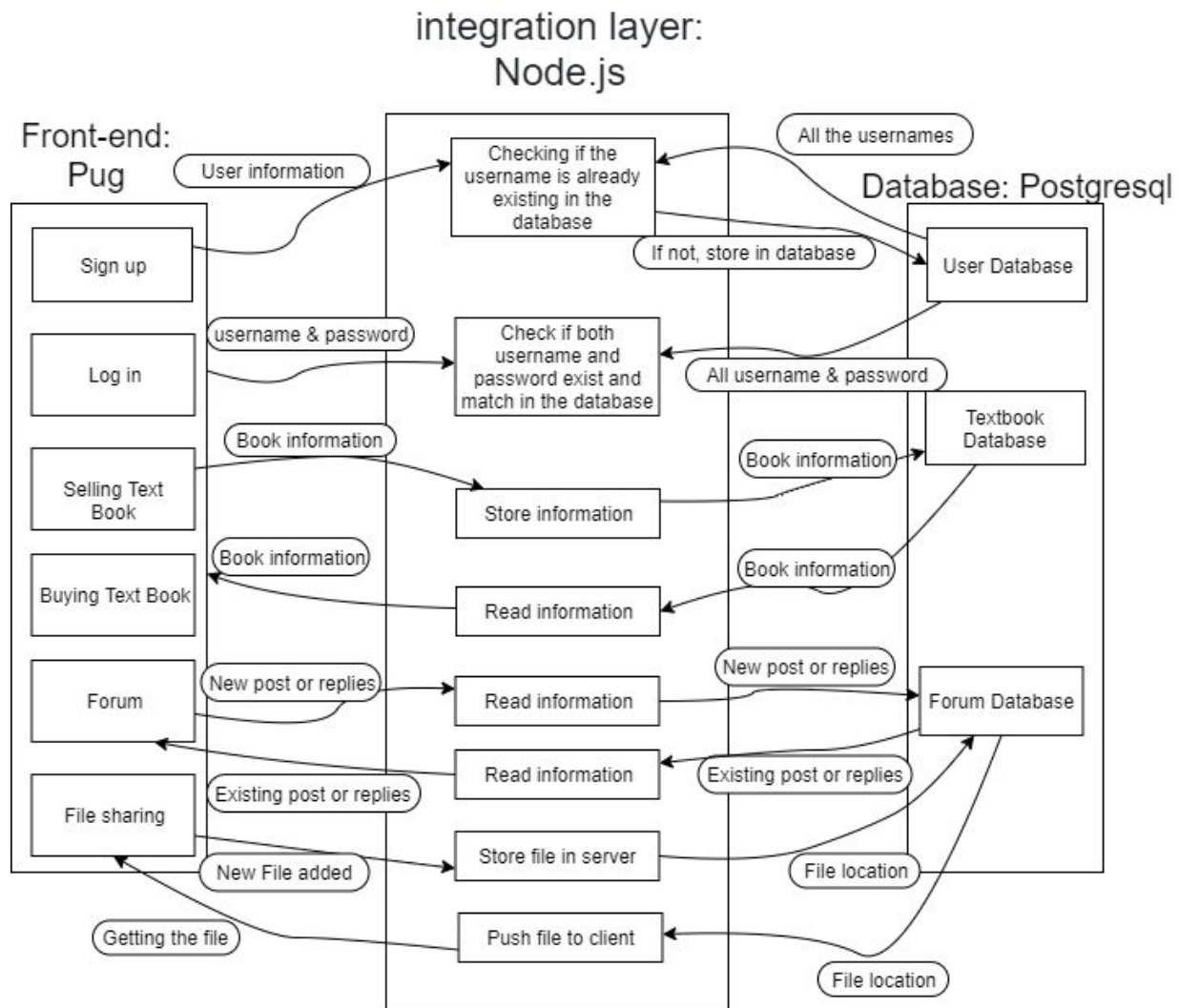
Revised List of Features (in order of priority)

1. Create Account
   a. Students may create accounts on TextBuddy through a registration page. They will have to provide information about themselves, which will be added to a database of users.
2. Sign In
   a. If a user already has an account, they can directly login. The login page will verify their username and password, and if it matches a user in the database, they will have access to the site.
3. Sell Textbooks/Notes
   a. Users can sell textbooks by posting listings for textbooks/notes that they have but no longer need. Information for a textbook being sold or a file being shared will be stored in the database system.
4. Buy Textbooks
   a. Users will be able to search/buy textbooks and notes. These materials will be separated by category, and the database will store the files and information that is displayed.
5. Forum
   a. A forum space separated by different categories of courses where people can discuss homework and other class ideas.
6. File sharing
   a. Users can upload or sell lecture notes, study guides, etc. This will take place within the listings for selling textbooks/notes. The users will have the option of filtering search to only see file uploads of notes.
7. Payment Page
   a. TextBuddy will not authorize payments. All payment transactions will occur off-site, and TextBuddy will simply be the means by which the users can see what is available for being sold and gain each other's contact information

Prominent changes made from the previous features list:
- We split the create account page and the sign-in/login page into two separate features to accommodate the amount of front-end and back-end implementation required for both.
- We decided that our website will not authorize payments and simply be a means of communication between users; our payment page describes this fact.
- The file-sharing feature will now be incorporated with our listings page (sell textbooks feature), and users will have the option of choosing between view textbook options and files for notes/notebooks.
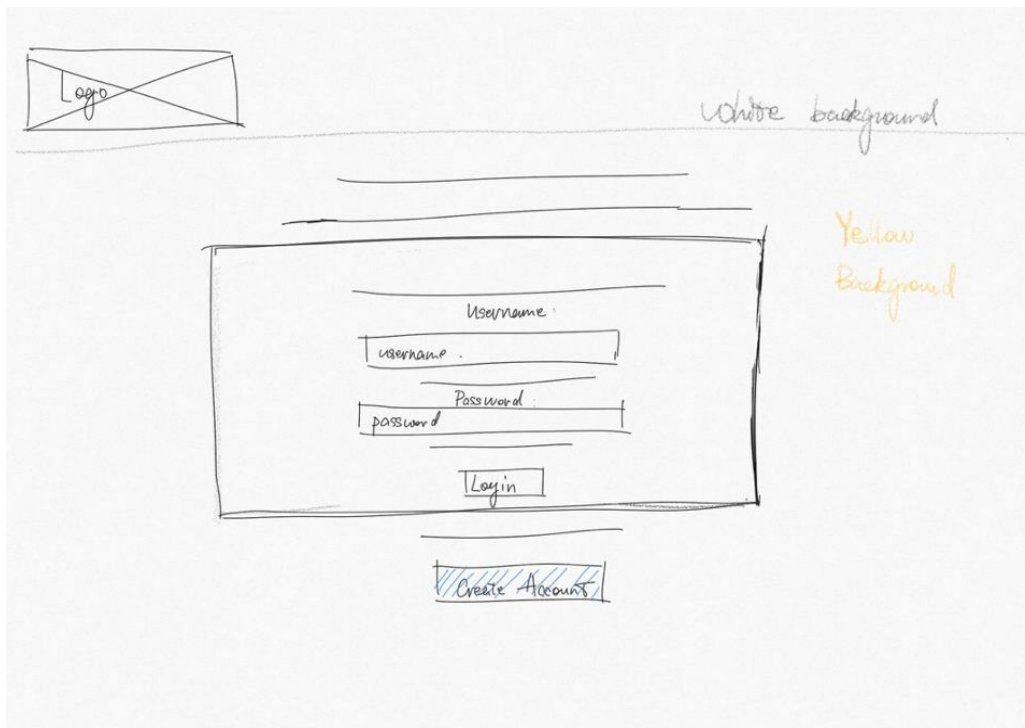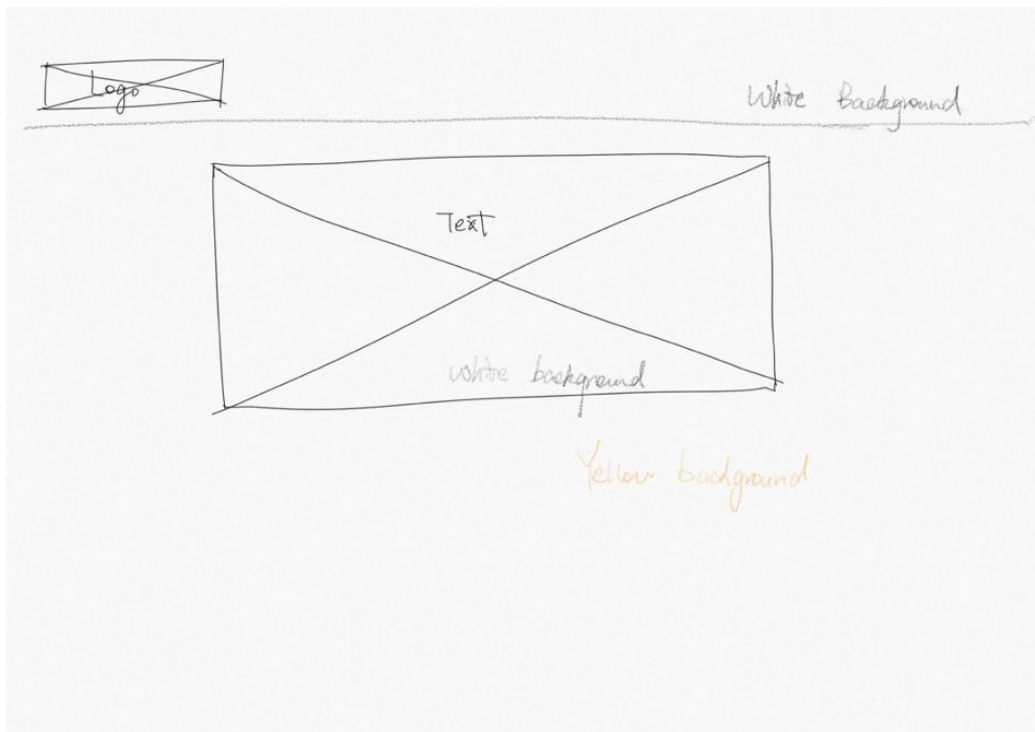
Architectural Diagram

Architecture Diagram

integration layer: Node.js

Front-end: Pug

Database: Postgresql

- Sign up
- Log in
- Selling Text Book
- Buying Text Book
- Forum
- File sharing

User information → Checking if the username is already existing in the database

All the usernames

If not, store in database → User Database

username & password → Check if both username and password exist and match in the database

All username & password

Book information → Store information

Book information → Textbook Database

Book information ← Read information

Book information

New post or replies → Read information

New post or replies → Forum Database

Existing post or replies ← Read information

Existing post or replies

New File added → Store file in server

File location

Getting the file → Push file to client

File location

Front End Design

1. **Login Page**

2. **Homepage**

3. Sign up Page

4. Homepage

5. Listing page

logo

≡ 🔍   Filter

Subject

Math

Computer

Physics

English

(Book Title)
(Author)
(Condition)

(Book Title)
(Author)
(Condition)

(Book Title)
(Author)
(Condition)

Yellow
Background.

6. Forum Page

logo

Subject.

Math

Computer

▷ CSCI 2400

▶ CSCI 3308

Physics

English

Topic

Content

Replies
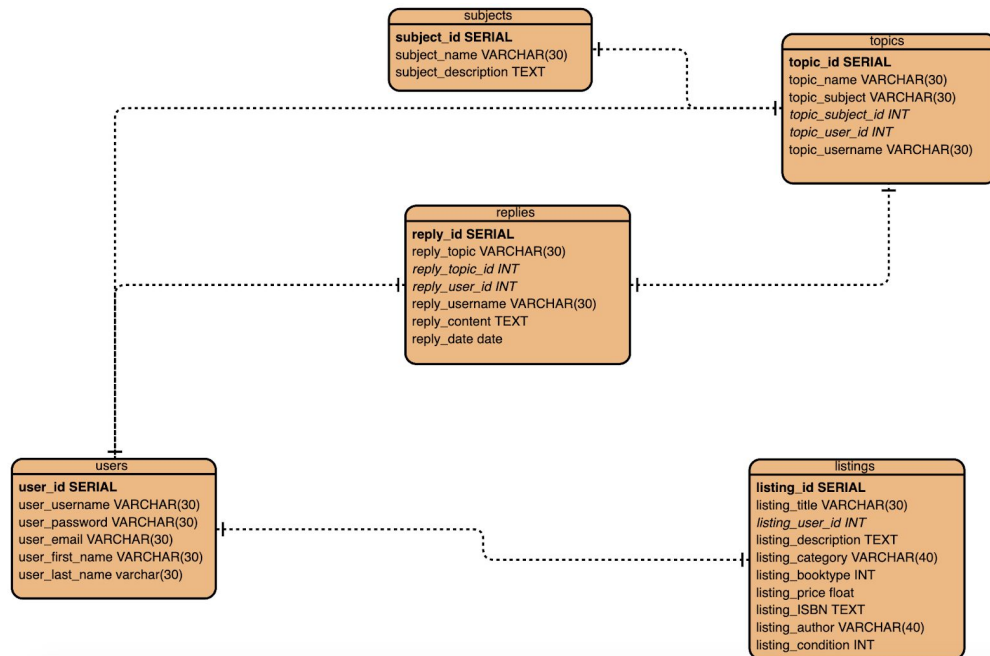
Yellow

Background.
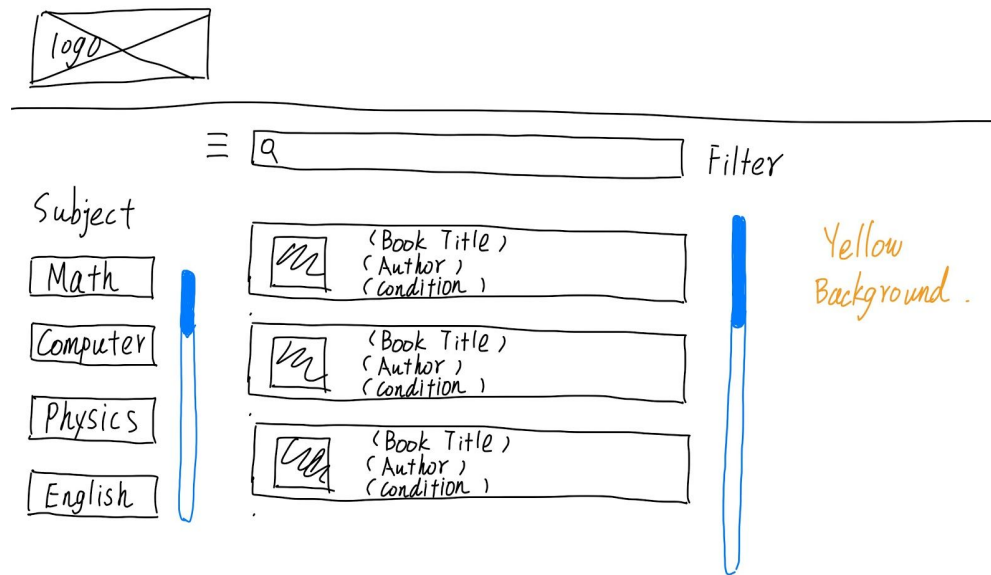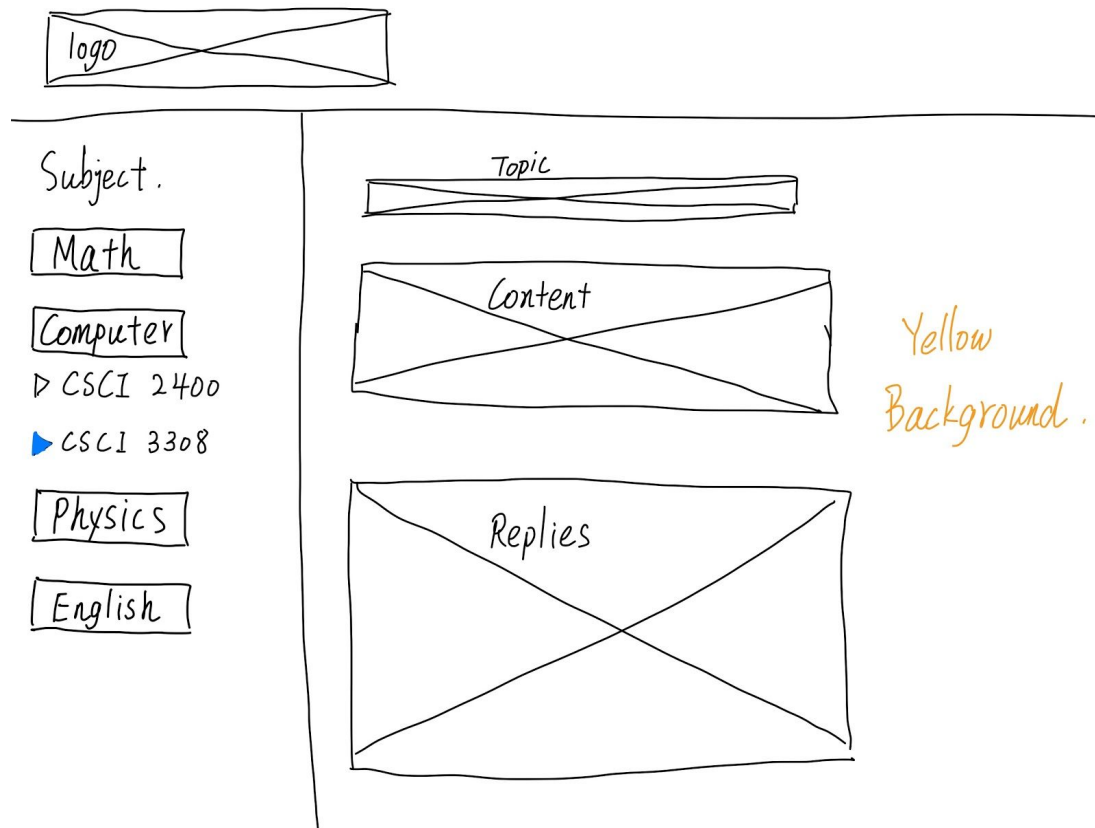
Web Service Design

- We do not plan to incorporate any web services or API's in our project. Our website will simply build on itself and its own resources/database system.

Database Design

- ❖ Our database is making use of Postgresql
- ❖ We have one database that we will connect too, with several tables that will be used for most of the information. The Entity Relationship Diagram will show these tables and their relationships in more detail

subjects
subject_id SERIAL
subject_name VARCHAR(30)
subject_description TEXT

topics
topic_id SERIAL
topic_name VARCHAR(30)
topic_subject VARCHAR(30)
topic_subject_id INT
topic_user_id INT
topic_username VARCHAR(30)

replies
reply_id SERIAL
reply_topic VARCHAR(30)
reply_topic_id INT
reply_user_id INT
reply_username VARCHAR(30)
reply_content TEXT
reply_date date

users
user_id SERIAL
user_username VARCHAR(30)
user_password VARCHAR(30)
user_email VARCHAR(30)
user_first_name VARCHAR(30)
user_last_name varchar(30)

listings
listing_id SERIAL
listing_title VARCHAR(30)
listing_user_id INT
listing_description TEXT
listing_category VARCHAR(40)
listing_booktype INT
listing_price float
listing_ISBN TEXT
listing_author VARCHAR(40)
listing_condition INT

- ❖ In the database we will be storing all user information, post information, listing information, and overhead topic information (as in the topic that forum posts will branch from)
  - ➢ All of the data within postgresql will be text, numbers, or date (aka usernames, passwords, emails, forum post titles or body, etc)
- ❖ We intend to allow for photo uploads for the forum and listings
  - ➢ We will do this by storing photos in a local directory (most likely the same directory that the web pages are stored under, because as of now data usage is not a concern)
  - ➢ To do this we will use the multer library within Node.js