

ReadNTrade Application

Full Stack Application Development Project on Book Exchange Platform

13th May, 2024

—

Full Stack Application Development
(S2-23_SEZG503)

Group - 35

—

Sayantan Biswas (2023mt93245)
Nikhil Devare (2023mt93244)

Introduction

Book swapping or book exchange is the practice of a swap of books between one person and another.

Practiced among book groups, friends, and colleagues at work, it provides an inexpensive way for people to exchange and rent books, find out about new books and obtain a new book to read without having to pay.

Our app aims at making this task easier and connect book lovers to each other so that they can connect with each other and share an experience with the other person and, even bond over it.



Description

Entities

Our book exchange platform would be having the following entities –

- Participants
- Books

Features

The features of this application are -

- Registration of participants
- Login
- Listing of books at a specific price for rent
- Listing of books for exchange
- Searching for books
- Rent a book
- Exchange a book with one of owner's book
- Sending an exchange request
- Approving the exchange request by the owner of the requested book
- Ability to handle multiple exchange requests
- Check the history and status of exchanges and rent, the owner has undertaken

Software/Tool used for creating this app:

- Frontend – NodeJS, React
- Backend - Java
- Database - MySQL
- Authentication – OAuth2, JWT
- API

Application code is available in GitHub URL as per below:

- HTTPS: https://github.com/yo-sayantan/FSAD_Project_BookExchange_Group35.git
- SSH: `git@github.com:yo-sayantan/FSAD_Project_BookExchange_Group35.git`
- GitHub CLI: `gh repo clone yo-sayantan/FSAD_Project_BookExchange_Group35`
- GitHub Link: https://github.com/yo-sayantan/FSAD_Project_BookExchange_Group35
- Drive Link: https://drive.google.com/file/d/1RDMFK3ZLG3So9qQcicEpdYXi_AjJBvJ/view?usp=sharing
- YouTube Link: <https://www.youtube.com/watch?v=nN1a3WyyQHU>

Contribution

- Design – Sayantan Biswas
- Backend – Sayantan Biswas
- Frontend – Sayantan Biswas
- Integration – Sayantan Biswas, Nikhil Devare
- Testing - Nikhil Devare
- Documentation - Nikhil Devare

Features

User Stories and Acceptance criteria

Epic 1: Login and Registration

User story 1.1 - Registration

As a new user, I should be able to register for the Book Exchange Platform application.

Acceptance Criteria –

- Input fields – Username, email ID, and password.
- Submit button should be disabled by default and only after entering the required fields – username and password, it should be enabled.
- Password criteria should be fulfilled – it should have at least one capital letter, one small letter, one number and one special character. Maximum it can contain 16 characters and minimum it should have 8 characters.
- If Password does not fulfil the criteria, and user clicks on ‘Submit’ then we should throw an error saying, “Password does not match the given criteria.”
- After providing correct username and password and clicking on ‘Submit’ button, user should get a message saying Registration is successful.

User story 1.2 - Login

As an existing user, I should be able to login to the Book Exchange Platform application.

Acceptance Criteria –

- Input fields – Username (email ID) and password.
- Login button should be disabled by default and only after entering the required fields – username and password, it should be enabled.
- Password field should be masked.
- If incorrect combination of username and password is provided and user clicks on ‘Login’ button, then error message should be displayed saying “Incorrect username or password.”
- After providing correct username and password and clicking on ‘Login button, user should be able to navigate on the homepage on the app.

Epic 2: Book Listing

User story 2.1 – List Books

As an existing user, I should be able to add a book in my library.

Acceptance Criteria –

- Input fields – Title, Author, Genre, Condition, Available, Price
- ‘Add book’ button
- Title, Author, Genre are string inputs which may contain up to 256 characters. They may contain numbers or special characters as well.
- Condition should accept inputs – Brand new, Good, Fair, Poor. If any other inputs are provided and user clicks on the ‘Add book’ button, then he should get an error saying, “Please provide correct condition value.”
- Available will be a Boolean value stating whether the book is currently available with you or already exchanged.
- Price, in case, if someone wants to purchase the book from you instead of exchange.
- After providing the correct input details and clicking on ‘Add book’ button, we should be able to add the book to library of books on the app.
- After adding a book, that book will be available to all the users for rent or exchange in the platform.

Epic 3: Book Searching

User story 3.1 – Book Searching

As an existing user, I should be able to search for books other users have added.

Acceptance Criteria –

- Input fields – Search box, we can enter any of these as search criteria – book name, author name and genre.
- User needs to provide the name of the book or author or genre they wish to exchange in the search bar.
- Available and Owner are optional fields and will adhere to the criteria mentioned in book listing epic.
- If no book is available for the provided search query, then user should get a message saying, “The book you are trying to search for is currently unavailable.”
- If the book is available, we should be able to see the details of the book – owner, author, genre, condition, whether it is available and whether it is up for purchase or if it for rent and how much.

Epic 4: Book Rent

User story 4.1 – Book Searching

As an existing user, I should be able to search for books and take a book for rent with the suggested price other users have added.

Acceptance Criteria –

- Search for the book name, if available, select the book, and click on rent.
- The owner of the book can see that the book has been rented and to whom it has been rented and is marked as availability – false.
- The book that has been rented will not be available in the search results for the other users, until the owner makes it available status – true
- The renter can see the history of requests for rent in the dashboard

Epic 5: Book Exchange

User story 5.1 – Sending and receiving book exchange requests

As an existing user, I should be able to send exchange request to another user.

Acceptance Criteria –

- ‘Exchange’ button should be available on the book details page
- The button should only be enabled if Available flag for the book is True.
- The button should only be disabled if the book is not selected by the user.
- When user clicks ‘Exchange’ button, a request is sent to the owner of the book.
- The owner can see the pending requests for exchange and which book is up for exchange from the requestee instead.
- The owner can see multiple exchange requests from multiple users.

User story 5.2 – Accepting Book Exchange requests

As an existing user, I should be able to receive exchange request from another user.

Acceptance Criteria –

- If another user sends an exchange request, then as a user, I should be able to see it when I see the details of that book put in exchange in the dashboard.
- The owner can accept any one request that he/she might seem good in terms of the book he/she is receiving instead.
- Once an owner accepts a request, all other requests for the same book made by other users will be rejected.
- Once an exchange request is accepted by the owner the two books involved in the transaction will switch ownership and will be available in each other’s portal as availability - false.
- The previous owner of the book can see this transaction in the History.
- The new owner of the book can see the requests being accepted in the dashboard.

Design, System Operations and Services

The design and architecture for the application is as follows –

Frontend -

- The frontend of the application will allow a smooth experience for the users to list their books and browse through the catalogue of books.
- It would be communicating with the backend through RESTful APIs to perform all the actions on the platform like adding a book to the repository, searching for a book, etc.
- HTML + CSS using React components is being used for this.

Backend –

- All the business logic of the application will be handled on the backend side.
- Requests from frontend, interacting with the database for searching or adding books, querying the database for incoming and outgoing exchange requests will be happening through the backend.
- The backend will expose the RESTful APIs for frontend to use.
- Java Spring framework can be used to handle this.

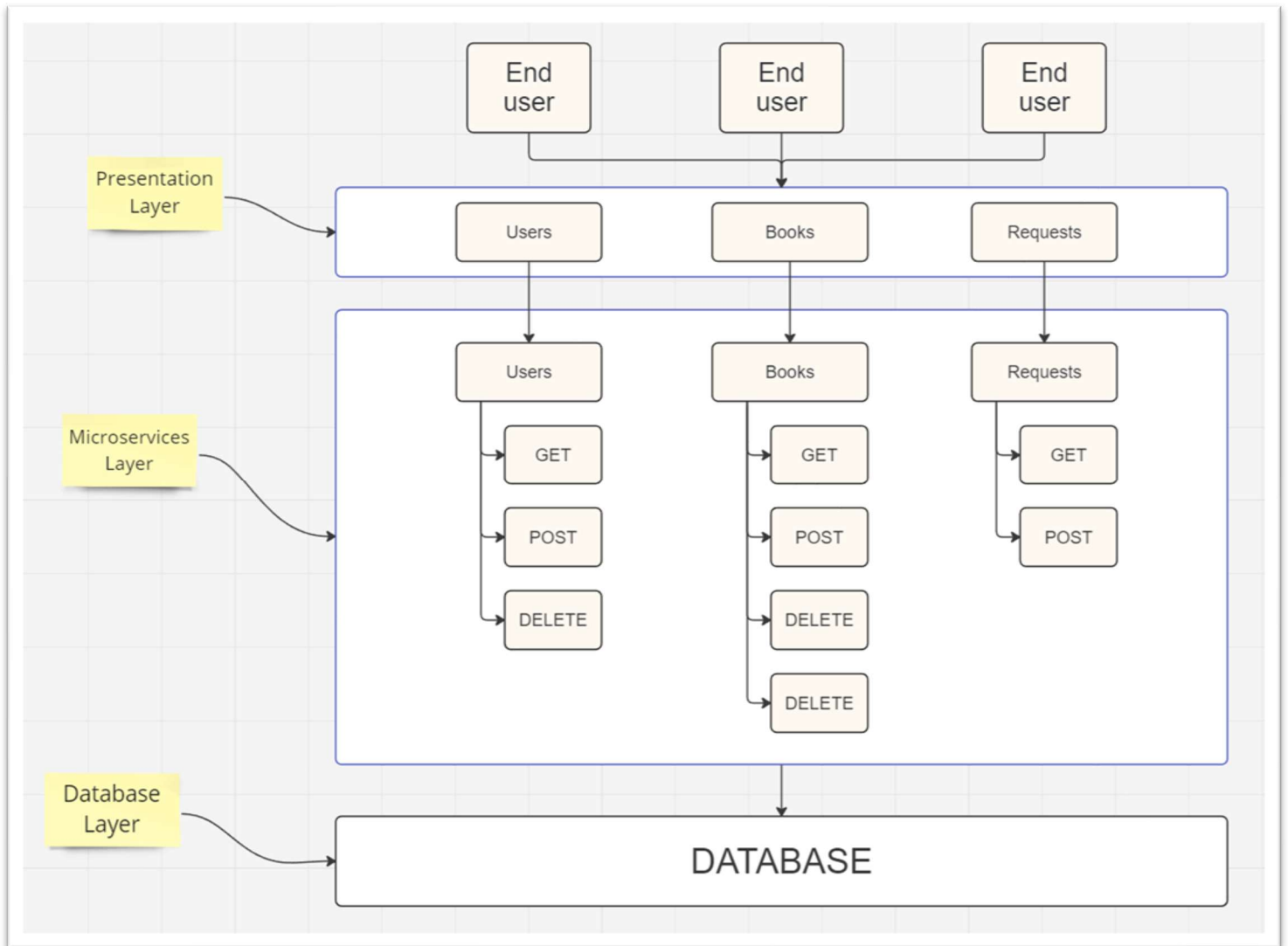
Database –

- Database layer would be consisting of multiple tables like the book repository, incoming requests, outgoing requests, etc.
- For structured way, a basic database like PostgreSQL can be used.

Authentication –

- Implement a robust authentication system to secure user login using Spring JWT authentication techniques.
- OAuth 2.0 or JWT (JSON Web Tokens) can be used to handle authentication and authorization.

Architecture



REST Endpoints

REST APIs endpoints with URI and HTTPS methods

No.	Entity	HTTP Method	URI
1	User	POST	/authenticate
		GET	/userinfo
		POST	/signup
2	Books	GET	/{"id"}
		GET	/getAllBooksForAUser
		GET	/getAllAvailableBooks
		POST	/addBook
		POST	/changeStatus
		DELETE	/{"id"}
3	Exchange Requests	GET	/getAllExchangeRequests
		GET	/getAllRequestsForAUser
		GET	/getAllRequestsFromUser
		GET	/{"id"}
		POST	/rentBook
		POST	/updateExchangeRequest