

Full Stack Application Development (S2-23_SEZG503) (Regular)

Assignment 1

Group Members –

1. Sayantan Biswas
2. Nikhil Devare

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 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 Of Application

Our book exchange platform would be having the following components –

Entities –

- Participants
- Books

Features –

- Registration of participants
- Listing of books
- Searching for books
- Sending an exchange request

User Stories and Acceptance criteria

Epic 1: Login and Registration

Registration –

User story 1.1 – 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.

Login –

User story 1.2 – 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 – 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 – (Optional). 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.

Epic 3: Book Searching

User story 3.1 – As an existing user, I should be able to add a book in my library.

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

Acceptance Criteria –

- Input fields – Search box, Available (optional), Owner (optional)
- User needs to provide the name of the book 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.

Epic 4: Sending and receiving book exchange requests

Send Exchange request –

User story 4.1 – 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 Available flag for the book is False.
- When user clicks ‘Exchange’ button, a request is sent to the owner of the book.

Receive Exchange request –

User story 4.2 – As an existing user, I should be able to receive exchange request to 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.
- I should be able to get a notification when I receive an exchange request.
- When the user chooses to accept the exchange request, they should be able to see the books listed under the owner who has asked for exchange and user should be able to select one book from the other user’s book listing for exchange.

Design and Architecture, System operations, and services identified.

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

Please see the architecture diagram.