

# Mutual Book Exchange Portal

Siddharth Kolipara  
Samanyu Segu  
Abhinav Rahul  
Amogh Nyapathy

2019A8PS1258H  
2019AAPS0329H  
2019A8PS1354H  
2019AAPS0322H

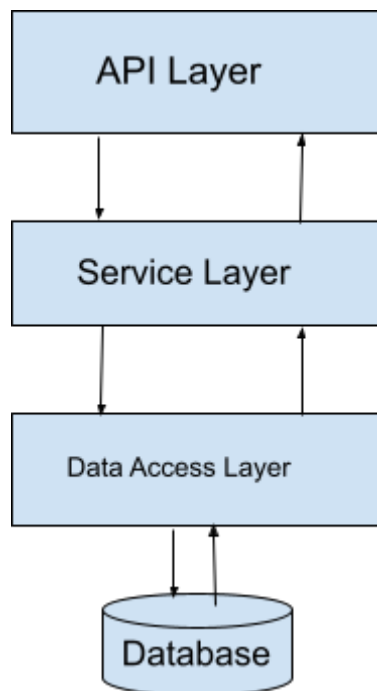
## Tech Stack

Front End -> HTML, CSS, JS, Bootstrap4, JSP

Backend -> Java, SpringBoot, JPA, MySQL

Hosting -> Heroku

## Backend Overview

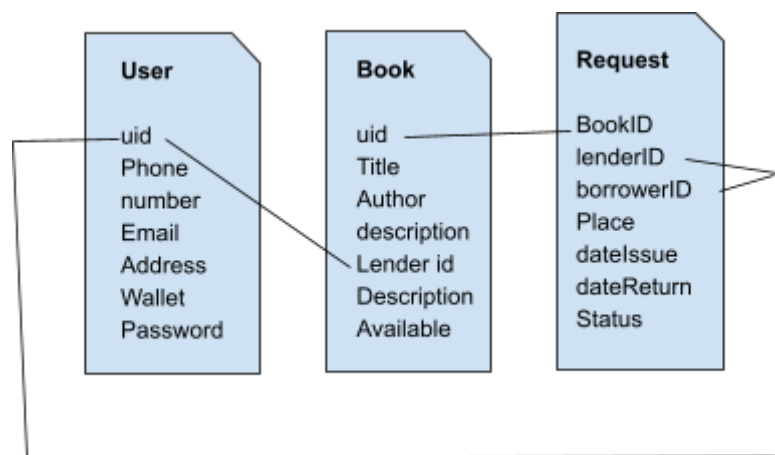


Backend is built entirely in Java with SpringBoot, using MySQL as a database. The various layers are described as follows

1. API Layer
  - a. This will have the API routing logic of the app, such as get, post, put, and delete requests
  - b. This will not have much of the business logic of the app, rather it will have routing logic, to call the functions from the service layer.
  - c. For example, routes for the create, read, update and delete functionalities of the books published by a user will be served with get, post, put, and delete requests from this layer, whereas the functionality of these routes will be given from the service layer.

2. Service Layer
  - a. This contains the business logic of the application.
  - b. When a request is encountered from the api layer, the service layer processes the request and calls the data access layer to query the necessary data from the database
3. Data Access layer
  - a. The data access layer receives control from the service layer to access the necessary tables in the database with the help of JPA

## Database Schemas



## Class Details

### User

1. User Class
  - a. Members:
    - i. UID
    - ii. Phone
    - iii. Email
    - iv. Address
    - v. password
    - vi. IsAdmin
    - vii. Wallet
2. User Controller Class - API layer
  - a. Members
    - i. Declaration of user service class
  - b. Methods
    - i. Get all Users
    - ii. Get user details

- iii. Create User
  - iv. Update user details
  - v. Update Wallet
  - vi. Delete User
  - vii. Generate User report
- 3. User Service Class - Service Layer
  - a. Members
    - i. Declaration of user data access layer class
  - b. Methods
    - i. Business logic for the methods in controller class, using data access class
- 4. User DataAccess Class - Data access layer
  - a. Methods
    - i. JDBC/ JPA calls to the database such as selectAll(), updateOne(), etc

## Book

- 1. Book Class
  - a. Members
    - i. UID
    - ii. Title
    - iii. Author
    - iv. description
    - v. Lender id
    - vi. Status
- 2. Book Controller Class - Api Layer
  - a. Members
    - i. Call to book service layer
  - b. Methods
    - i. Create book
    - ii. Update book
    - iii. Delete book
    - iv. Get book by id
    - v. Get all books
    - vi. Get book by lender id
    - vii. Get book by title
    - viii. Get book by author
    - ix. Get all books by status
- 3. Book service class and book data access class hold similar functionality as described in user class

## Request

- 1. request class
  - a. Members

- i. UID
  - ii. BookID
  - iii. lenderID
  - iv. borrowerID
  - v. Place of exchange
  - vi. daterequest
  - vii. dateReturn
  - viii. Status
2. request controller class - API layer
  - a. Methods
    - i. Create request
    - ii. Update request
    - iii. Delete request
    - iv. Get by UID
    - v. Get by BookID
    - vi. Get by LenderID
    - vii. Get by BorrowerID
    - viii. Get by Status
    - ix. Accept Request
    - x. Extend request
    - xi. Charge Penalty
3. The service and data access layer have similar functionality as previously described.

## Authentication

1. Sign up class
2. Sign in class

## Roles of Users

### 1. User

- Create, update, view and delete own profile
- Create update, view and delete their own book listings
- Create, Update, delete, view, return, and extend a request that they created
- View and accept requests that other users have placed on their listed books
- View catalogue of books
- Add money to wallet

### 2. Admin

Apart from user functionalities

- View List of all users
- View list of all book requests with status
- View report on each individual user
- Delete, update any book
- Delete, update any non admin user

Following the main class groups mentioned above, the backend functionality for the webapp is handled.

## Front-End

### Screens

1. Sign Up
2. Sign in
3. Home Screen (Book Catalogue)
4. Search Screen
5. Book Request Screen
6. My Profile Screen
7. My Profile Update/Delete Screen
8. My Books View/Upload/Edit/Delete Screen
9. Request - View, Create,edit,accept,delete screen
10. Update wallet and pay fines screen
11. Report Generation Screen

### SiteMap

