**Work Integrated Learning Programmer**

**M.Tech Software Engineering**



**FULL STACK APPLICATION DEVELOPMENT**

**Submitted by**

**Raghunath Reddy G**

**2023MT93051**

Table of Contents

**Type chapter title (level 1)1**

Type chapter title (level 2)2

Type chapter title (level 3)3

**Type chapter title (level 1)4**

Type chapter title (level 2)5

Type chapter title (level 3)6

Problem statement: BOOK EXCHANGE PLATFORM

Book lovers frequently accumulate a collection of books they have read and look for other recommendations. They are always eager to explore new reading material. Traditional methods of exchanging books, such as local book swaps or lending among friends, are limited in scope and accessibility. Therefore, it is imperative to have a digital platform that can facilitate book exchanges on a larger scale. This platform should connect users with similar reading interests, enabling them to trade books easily and efficiently. This project aims to develop a full-stack web application that serves as a centralized platform for users to exchange, lend, and borrow books with other users. The platform should provide a user-friendly interface, robust search and recommendation features, and secure transaction capabilities.

***Key Features:***

**User Authentication**: Implement a secure user authentication system to allow users to register, log in, and manage their accounts.

**Book Listing**: Enable users to list books they want to exchange or lend, including details such as title, author, genre, condition, and availability status.

**Book Search:** Provide users with advanced search and filtering options to discover books based on criteria like genre, author, title, location, and availability.

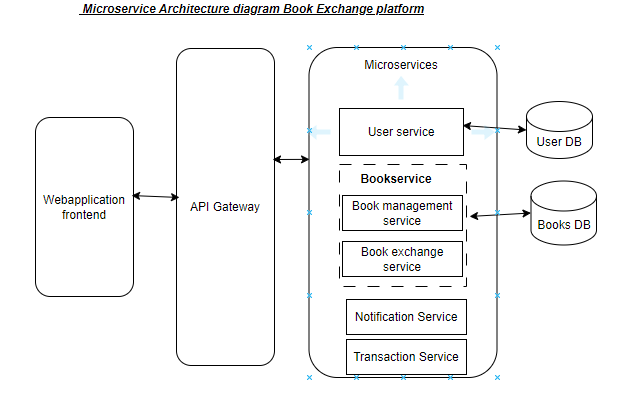
**Exchange Requests**: Allow users to send and receive exchange requests for specific books, including negotiation options for terms such as delivery method and duration.

**Messaging System**: Implement a messaging system to facilitate user communication regarding book exchanges, including negotiation details, logistics, and scheduling. (Mock API s can be used)

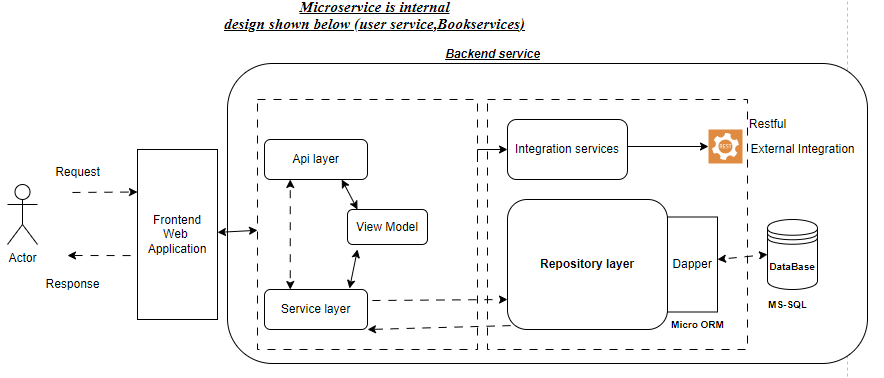
**User Profiles**: Enable users to create profiles with information about their reading preferences, favorite genres, and books they currently own or wish to acquire.

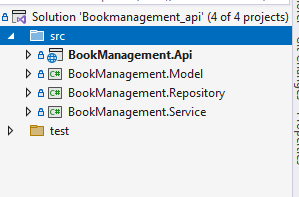
**Transaction Management**: Provide tools for users to track the status of their exchange transactions, including pending requests, accepted exchanges, and completed transactions.

***Architecture design:***



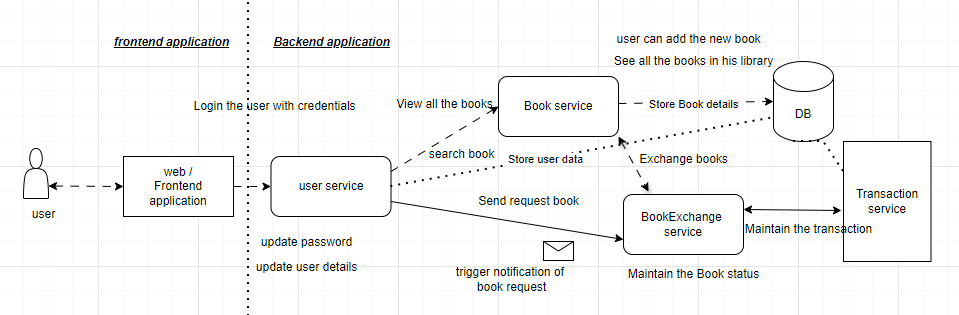
***Each Microservice is internal design shown below (user service,Bookservices)***



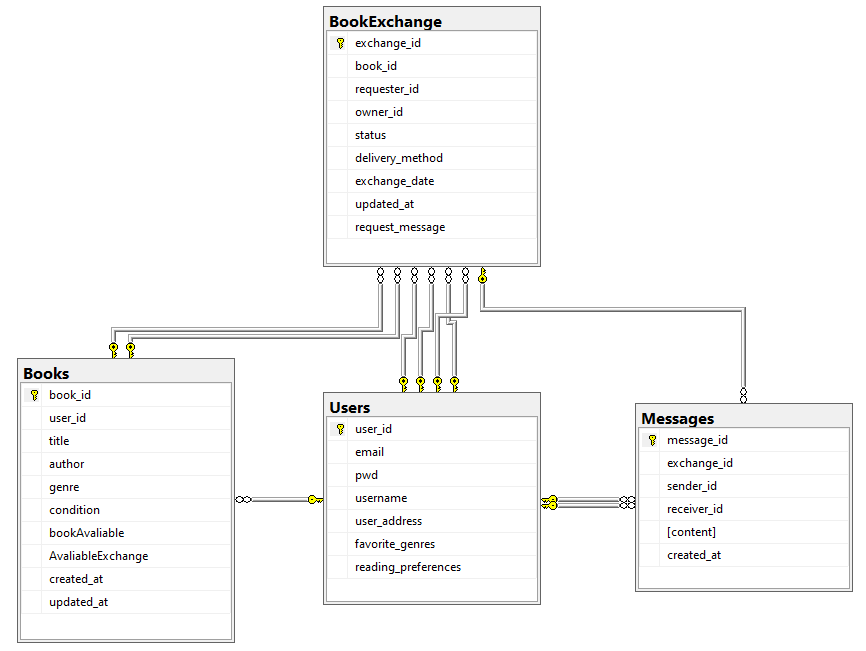


Thebackend service is designed using the microservicce each micro-service is independently developed & deployed

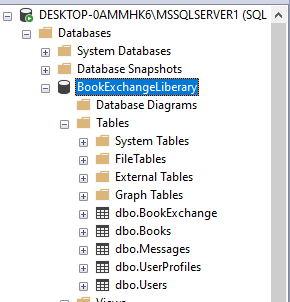
***Flow diagram:***



***Database Design:***



***MS-SQL Database design***



***User Story 1: User Authentication***

As a user,I want to securely register, log in, and manage my account, So that I can access and use the book exchange platform.

Acceptance Criteria:

The platform must allow users to register with a valid email and password.

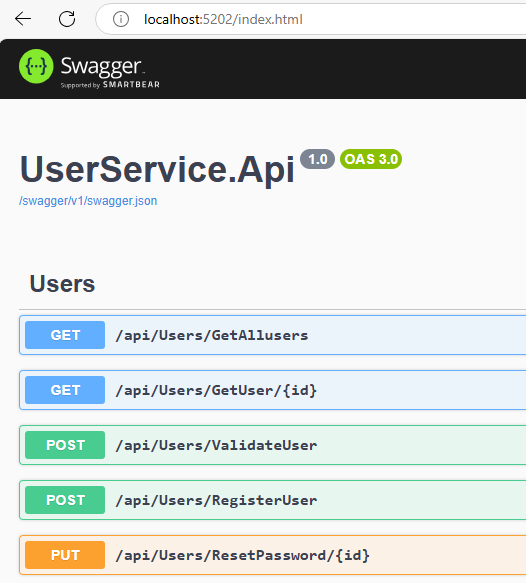
Passwords must be stored securely using encryption.

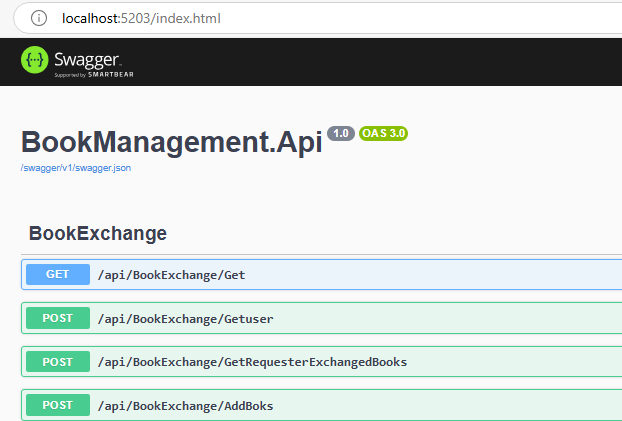
Users should be able to reset their password via a password recovery system.

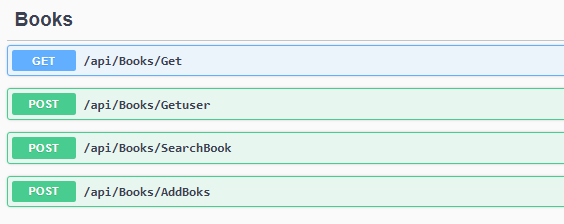
Users should be able to log out from their account.

|  |  |  |
| --- | --- | --- |
| **Entity** | **Http Method** | **URI** |
| Users | GET | **/api/Users/GetAllusers** |
| GET | **/api/Users/GetAllusers/${userid}** |
| POST | **/api/Users/RegisterUser** |
| PUT | **/api/users/resetPassword ${id}** |
| DELETE | **/api/users/${id}** |

***Swagger screen of the “user service”***







Front End

