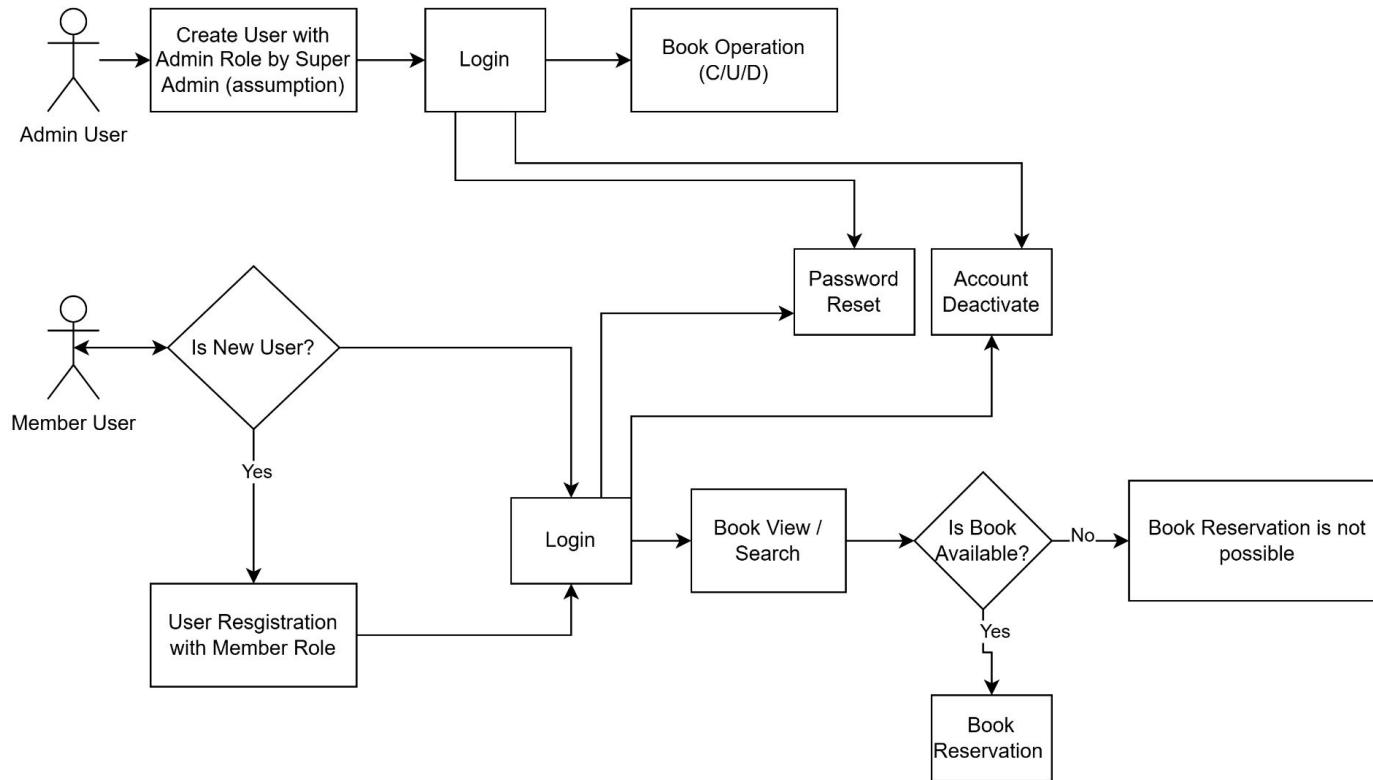


Library Book Reservation System

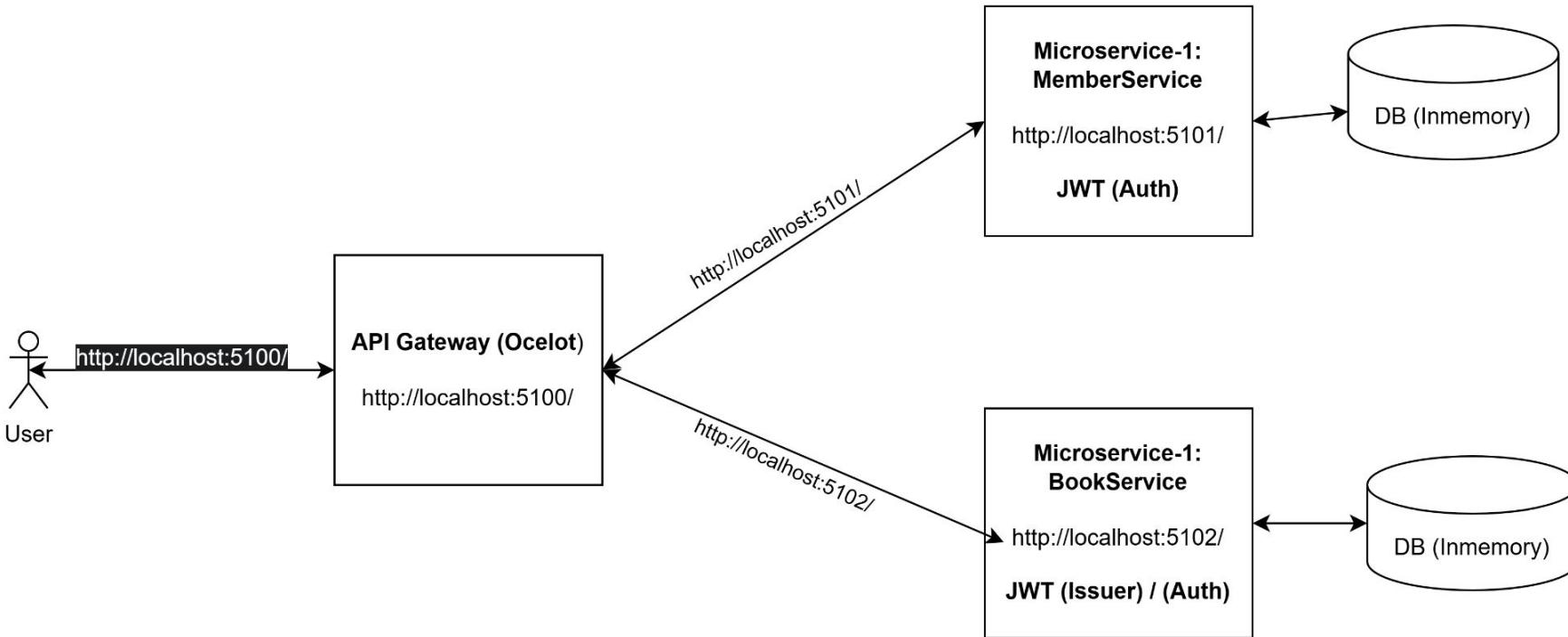
Library Book Reservation System

High-Level Workflow Design



Library Book Reservation System

Software Architecture



List of completed components on the Test Projects

- Microservice projects with multi-tier architecture
 - Member Service project
 - Book Service project
- API Gateway project using Ocelot
- Implemented JWT based authentication & role-based authorization
- Implemented Pagination
- Implemented Serilog for daily rolling logs
- Implemented error handling with logging

List of Role and Method names

Allow Anonymous can access the methods below:

- Member > Userlogin
- Member > UserAdd
- Member > AdminAdd

Admin Role can access the methods below:

- Book > Add Book
- Book > Update Book
- Book > Delete Book
- Member > Update
- Member > GetbyUserID
- Member > GetAll
- Member > Update

Member Role can access the methods below:

- Reservation > AddReservation
- Book > Search Books

Both roles can access the methods below:

- Book > Get book by ID
- Book > Get all books
- Member > useraccountdeactivate
- Member > UserPasswordReset

Postman collection list : Collection file attached for your reference

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar displays a tree structure of collections: 'LBRS' (selected), 'Members', 'Books', and 'Reservations'. A blue box highlights the 'GET http://localhost:5100/api/books/searchbooks' entry under the 'Books' collection. An orange box highlights the entire 'LBRS' collection structure. The main workspace shows a detailed view of the 'Books' collection. The top navigation bar includes 'Home', 'Workspaces', 'API Network', 'New', 'Import', and a search bar. The right panel shows an 'Overview' tab for 'LBRS / Books / http://localhost:5100/api/books/searchbooks'. The 'Params' tab is selected, showing query parameters: genre (sof), author (eri), pageNumber (1), and pageSize (2). The 'Body' tab shows a JSON response with a success message and data containing a single book record.

LBRS

- Members
 - POST http://localhost:5100/api/members/memberadd
 - POST http://localhost:5100/api/members/adminadd
 - POST http://localhost:5100/api/members/userlogin
 - GET http://localhost:5100/api/members/getbyuserid
 - GET http://localhost:5100/api/members/getall
 - POST http://localhost:5100/api/members/update
 - POST http://localhost:5100/api/members/useraccountdeactivate
 - POST http://localhost:5100/api/members/userpasswordreset
 - GET http://localhost:5100/api/members/healthcheck
- Books
 - POST http://localhost:5100/api/books/add
 - POST http://localhost:5100/api/books/update
 - POST http://localhost:5100/api/books/delete
 - GET http://localhost:5100/api/books/searchbooks
 - GET http://localhost:5100/api/books/getall
 - GET http://localhost:5100/api/books/get
- Reservations
 - POST http://localhost:5100/api/reservations/add

Postman Echo

LBRS / Books / http://localhost:5100/api/books/searchbooks

GET http://localhost:5100/api/books/searchbooks?genre=sof&author=eri&pageNumber=1&pageSize=2

Params

Key	Value
genre	sof
author	eri
pageNumber	1
pageSize	2

Body

```
{ "success": true, "message": "Books retrieved successfully.", "data": [ { "bookId": "0e413a81-13f0-4d70-a6c9-79c7352eb27e", "title": "Domain-Driven Design", "isbn": "978-0321125224", "author": "Eric Evans", "genre": "Software Design", "publishedYear": 2003, "publisher": "Addison-Wesley", "description": "Edited something", "totalCopies": 1, "availableCopies": 1, "isActive": true, "isAvailable": true } ] }
```

Thank you