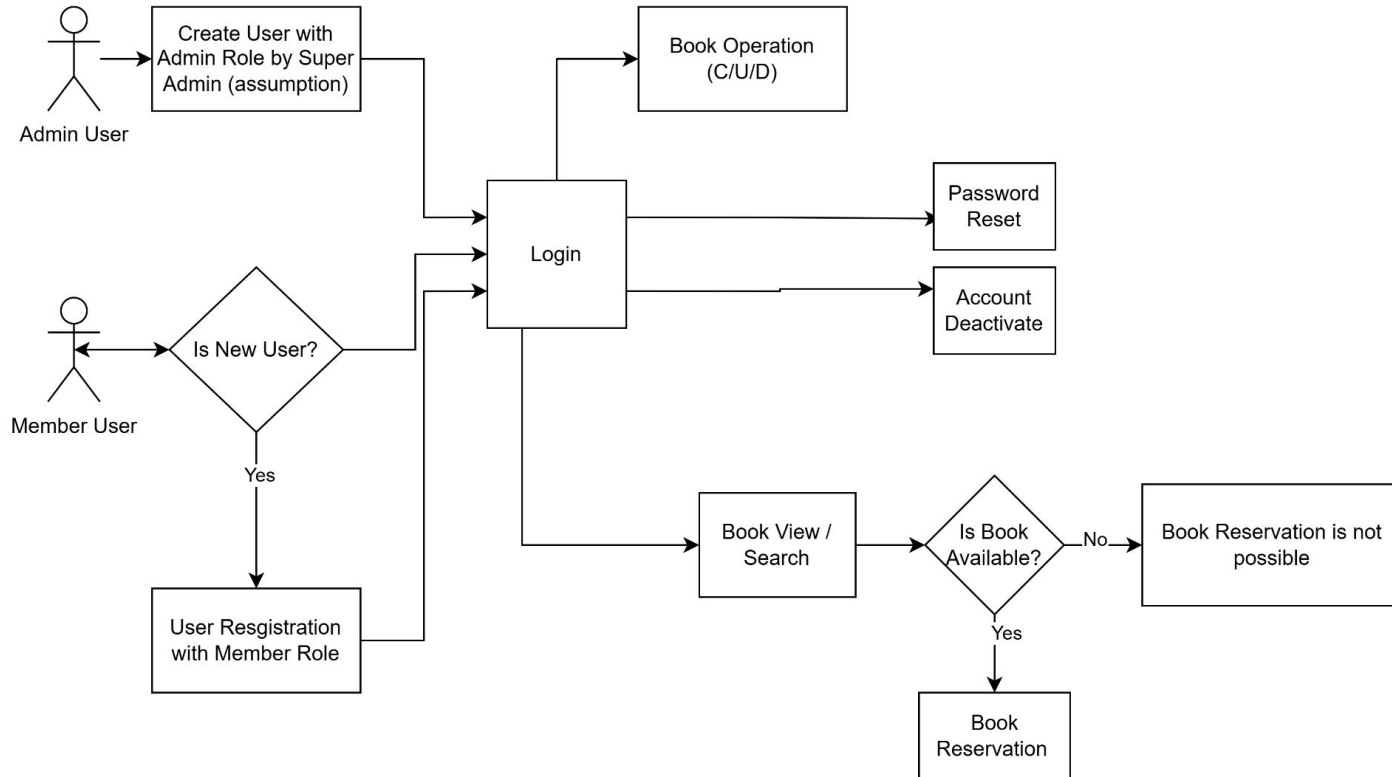


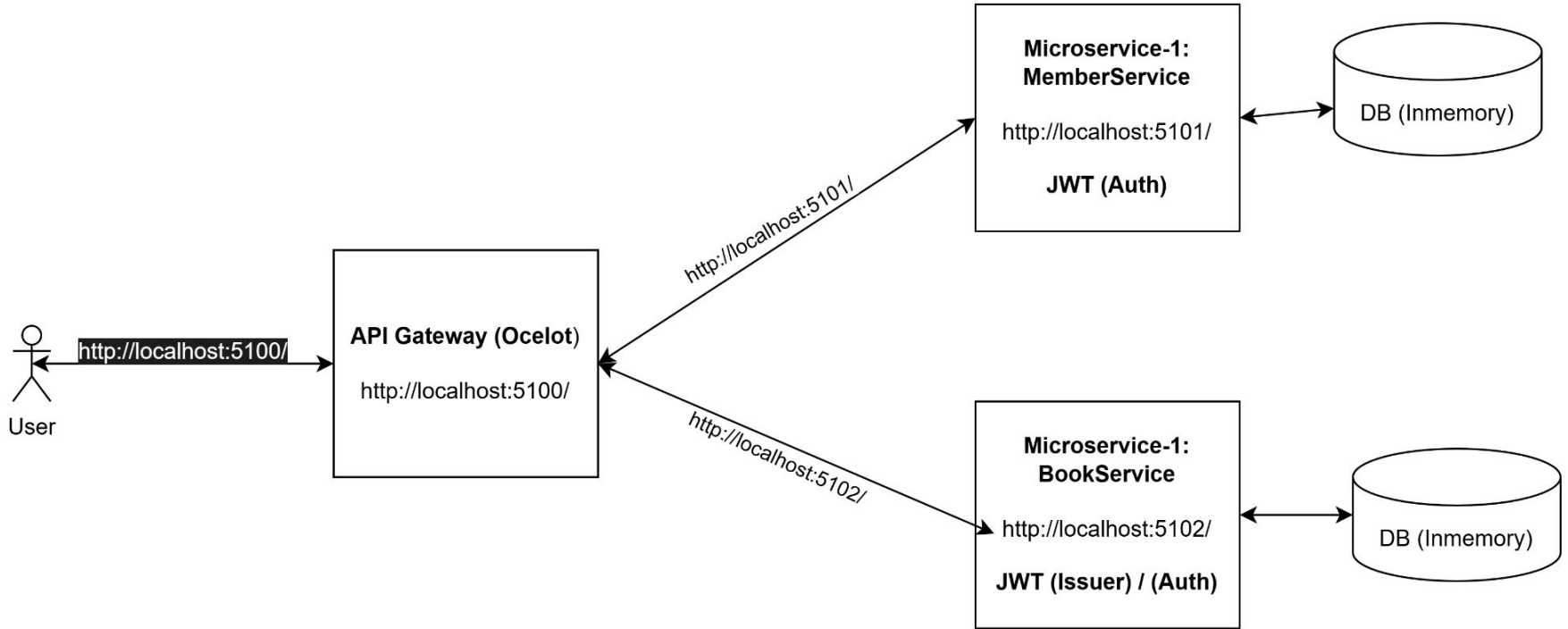
# Library Book Reservation System

# 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
- Book > GetReservedBooks

## 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 displays the Postman interface with a collection named 'LBRS' on the left and a detailed view of a GET request on the right.

**LBRS Collection:**

- 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`
  - GET `http://localhost:5100/api/reservations/getreservedbooks`

**Request Details:**

- Method:** GET
- URL:** `http://localhost:5100/api/reservations/getreservedbooks`
- Body:** none (selected), form-data, x-www-form-urlencoded, raw, binary, GraphQL

**Response (JSON):**

```
1 {
2   "success": true,
3   "message": "Reserved books retrieved successfully.",
4   "data": [
5     {
6       "reservationId": "b960ff82-d625-4d73-ba79-7f7621b7850f",
7       "userId": "ce8dbf8c-1faa-44f8-a19e-5650847201f0",
8       "remarks": "Reserved for a week",
9       "reservationStatusListDTO": [
10        {
11          "reservationStatusId": "ae7df08c-b2ab-44c9-8589-f58eb728bc82",
12          "reservationStatusType": "Reserved",
13          "createdDate": "2026-01-12T13:51:48.6616849Z",
14          "createdByUserID": "ce8dbf8c-1faa-44f8-a19e-5650847201f0"
15        }
16      ],
17       "bookDetailResViewDTO": {
18         "bookId": "35a5d886-a6c7-441a-b0b6-1c76d2d16170",
19         "title": "Refactoring: Improving the Design of Existing Code",
20         "isbn": "978-0201485677",
21         "author": "Martin Fowler",
```

Thank you