Library Management Project Requirements

- Create a book catalog in library
    - Create with number of quantities
- Get a book details
    - Return the number of books left
- Add a user who can rent a book
- A particular book will be present for a fixed quantity
- User should be able to rent a book
- User can rent only one book
- User should be able to return a book
- User should return a book before 7 days
- User has to pay 5rs/day fine if book is not returned within 7 days
- User should be able to pay the fine
- User can only rent a book again if fine is fully paid
- User should be able to get fine

Design of API for the above requirements

- API to create book :
    - POST /api/books

      {

      title, author, quantity

      }

- API to get book details :
  - GET /api/books?title=””
- API to create user :
  - POST /api/user

    {

        username

    }
- API to get user details :
  - GET /api/user?username=””
- API to create order :
  - POST /api/order

    {

        username, title

    }
- API to delete/return order :
  - Delete /api/order/username/{user}/titlr/{title}
- API to pay fine :
  - PUT /api/fine

    {

        Username, title, amount

    }
- API to get fine details:
  - /api/fine/user/{username}/title/{title}

Table structure for the above design :

Book

id	Auto Increment, Big int
title	VARCHAR(225), NOT NULL, UNIQUEt
author	VARCHAR(225), NOT NULL
quantity	Big int

User

id	Auto Increment, Big int
username	VARCHAR(225), NOT NULL
fine_id	Big int
Order_Id	Big int

Order

id	Auto Increment, Big int
user_id	Big int
book_id	Big int

Fine

id	Auto Increment, Big int
amount	int

- user to book -> ManyToOne -> Order
- user to fine -> OneToOne

- Design approach -> Inline Architecture (bottom up layer)