

Library Management System

-By Prasad B. Badwar



Tools & Technologies

Language : Java



Framework: Spring Boot Rest API, Spring Data JPA



spring boot

Database: PostgreSQL 14.7



PostgreSQL

Software : Eclipse



Introduction

The Simple and Small Library Management System Application is a software solution that is designed to manage library resources effectively. The objective of this system is to simplify the library management process, reduce administrative overhead, and improve the overall user experience. This business requirement document outlines the functional and non-functional requirements of the Simple and Small Library Management System Application.

Functional Requirements

User Management

Book Return

Book Management

Book Reservation

Book Search

Book Loan
Management

Book Borrowing

Reports

Database Tables

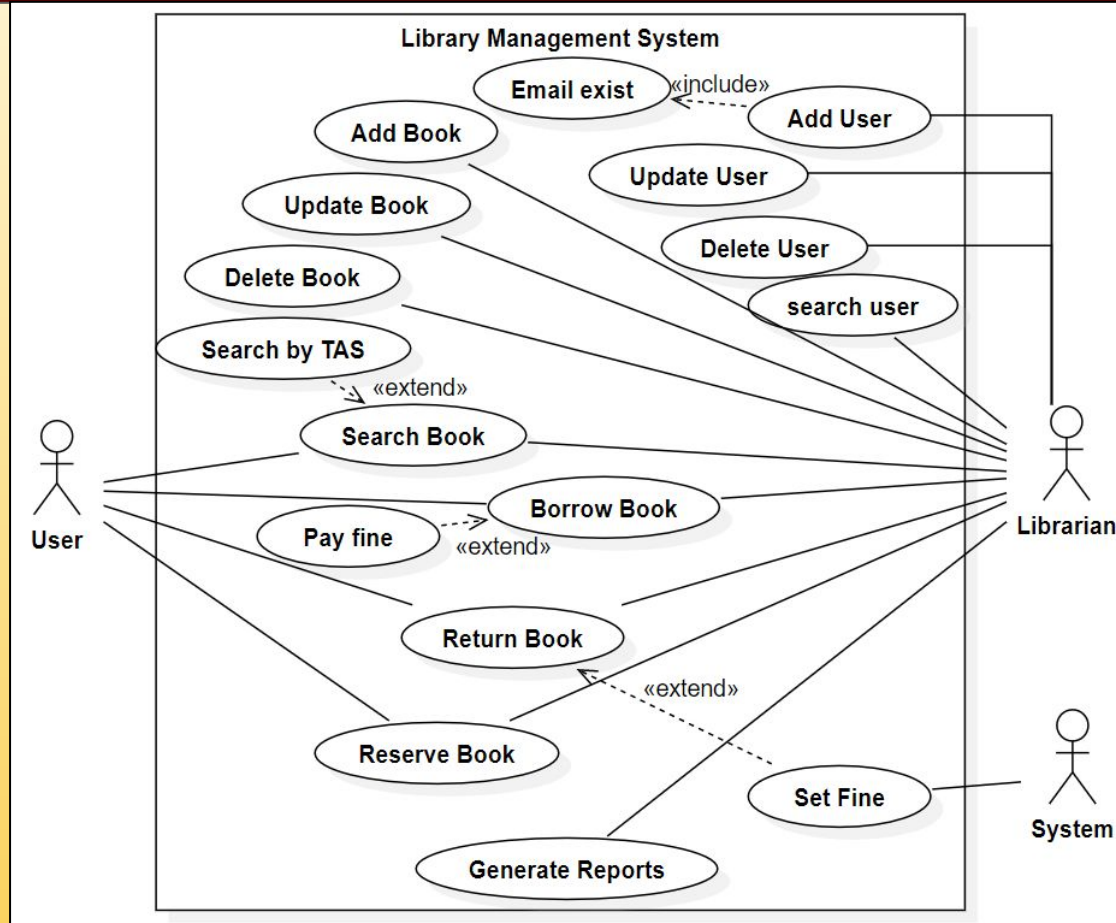
```
library=# \d
```

List of relations

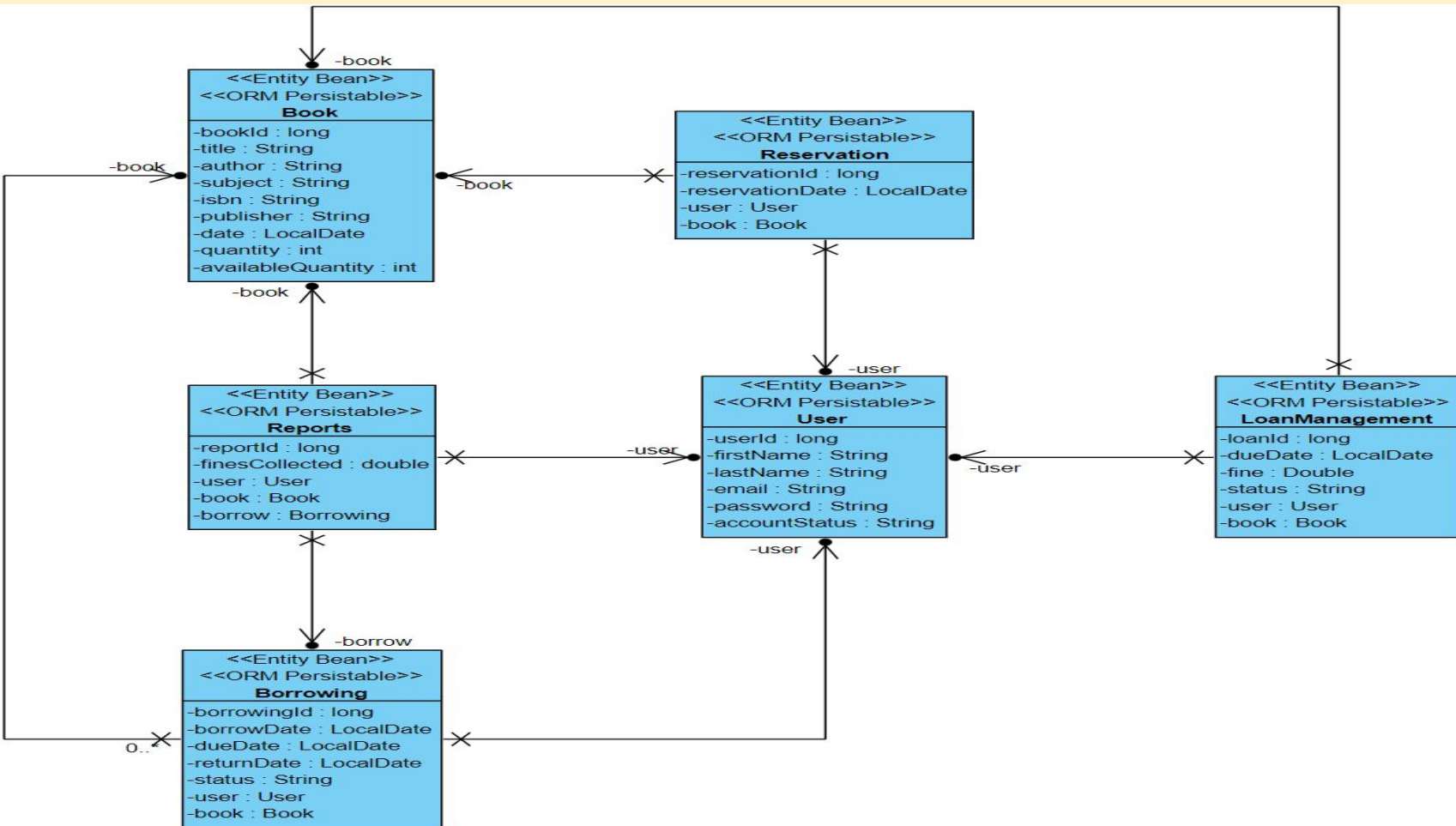
Schema	Name	Type	Owner
public	book	table	postgres
public	borrowing	table	postgres
public	hibernate_sequence	sequence	postgres
public	loan_management	table	postgres
public	reports	table	postgres
public	reservation	table	postgres
public	users	table	postgres

(7 rows)

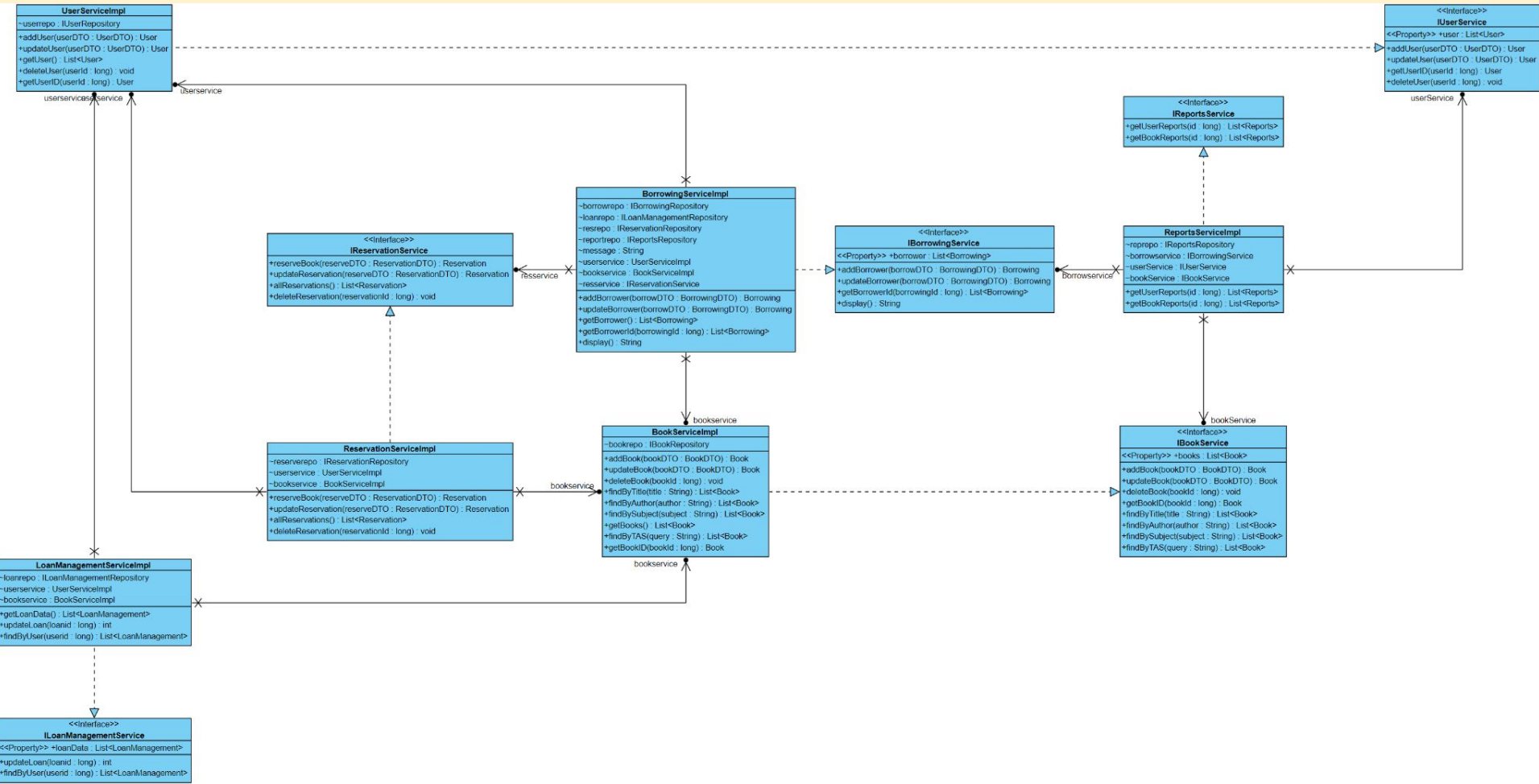
UML Diagrams - Use Case



UML Diagrams - Class Diagram(Entity)

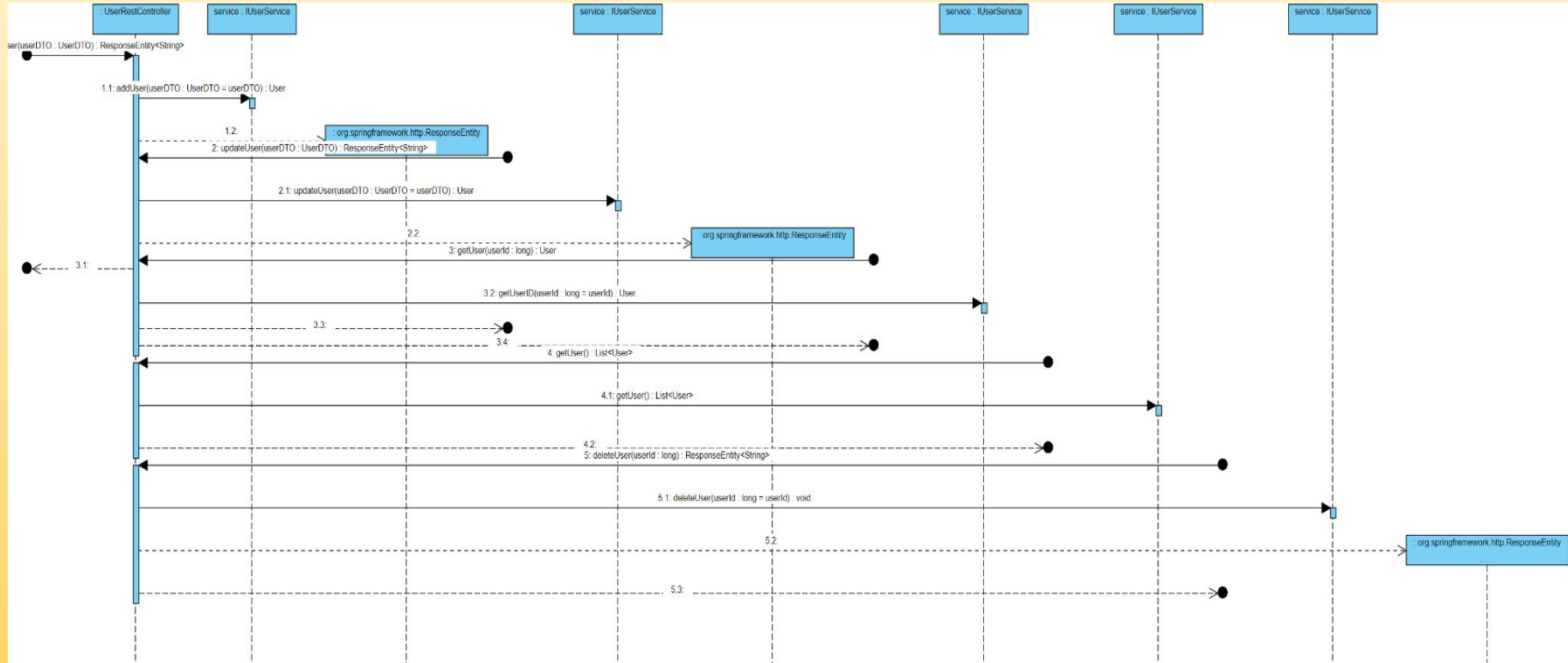


UML Diagrams - Class Diagram(Service)

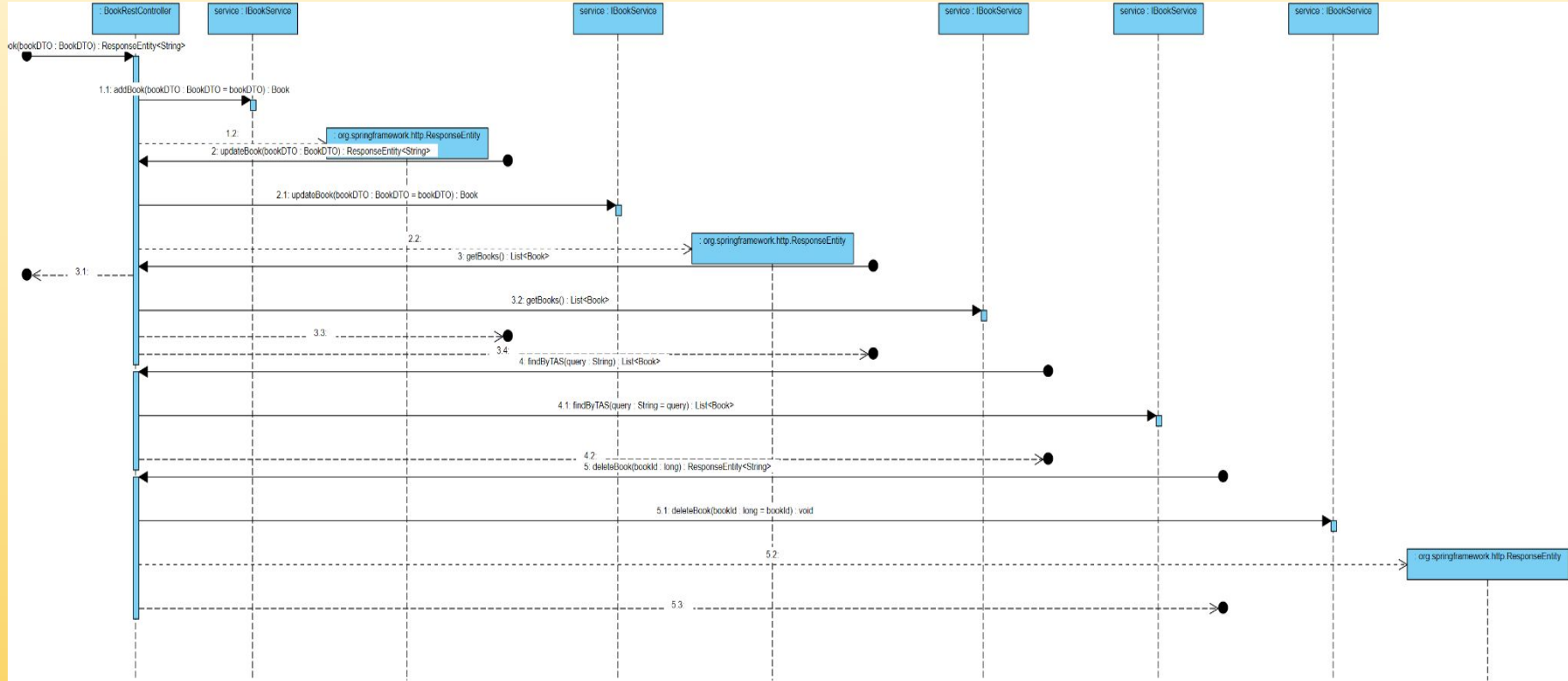


UML Diagrams - Sequence Diagram

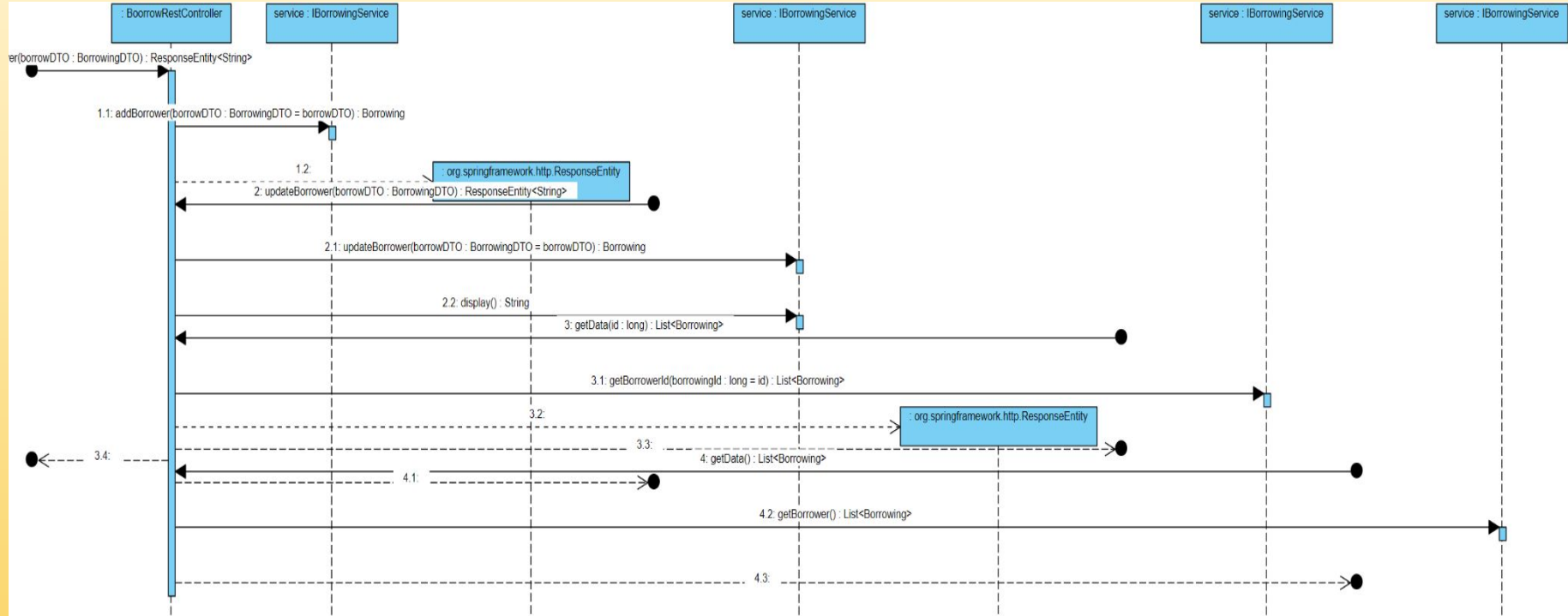
1. User Rest Controller:



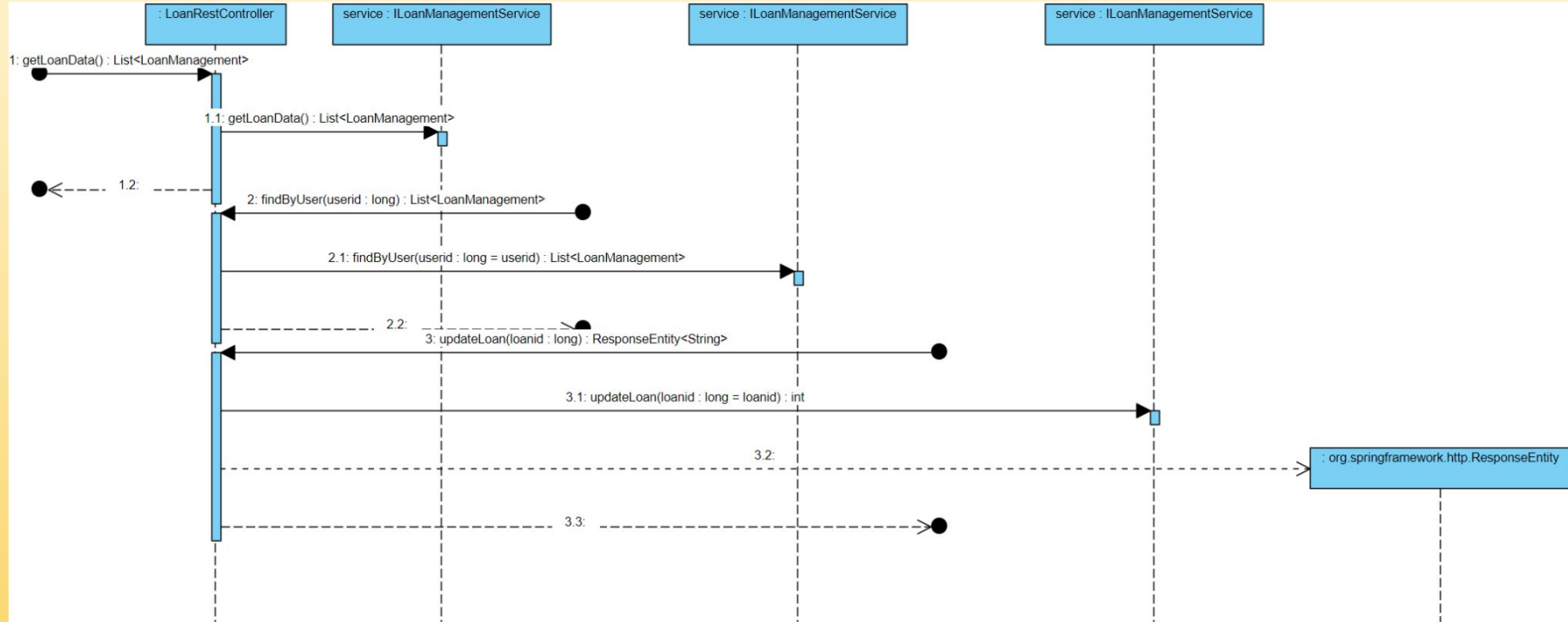
2. Book Rest Controller:



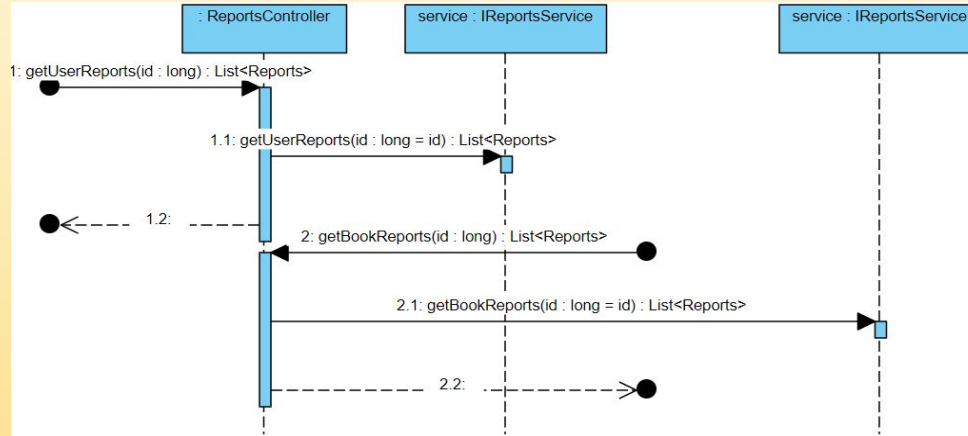
3. Borrowing Rest Controller:



4. Loan/Fine Rest Controller:



4. Report Rest Controller:



Project Snapshots

user-rest-controller



PUT /lms/user/updateuser

POST /lms/user/adduser

GET /lms/user/getusers

GET /lms/user/getuserid/{userId}

DELETE /lms/user/deleteuser/{userId}

POST

/lms/user/adduser

Parameters

No parameters

Request body required

application/json

```
{
  "userId": 4,
  "firstName": "John",
  "lastName": "Smith",
  "email": "john@gmail.com",
  "password": "John@smith123",
  "accountStatus": "Inactive"
}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
'http://localhost:9090/lms/user/adduser' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
  "userId": 4,
  "firstName": "John",
  "lastName": "Smith",
  "email": "john@gmail.com",
  "password": "John@smith123",
  "accountStatus": "Inactive"
}'
```

Request URL

http://localhost:9090/lms/user/adduser

Server response

Code

Details

200

Response body

User Added Successfully!!

GET

/lms/user/getuserid/{userId}

Parameters

Cancel

Name

Description

userId * required

integer(\$int64)

1

(path)

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
'http://localhost:9090/lms/user/getuserid/1' \
-H 'accept: */*'
```

Request URL

http://localhost:9090/lms/user/getuserid/1

Server response

Code

Details

200

Response body

```
{
  "userId": 1,
  "firstName": "Prasad",
  "lastName": "Badwar",
  "email": "prasad@gmail.com",
  "password": "123456",
  "accountStatus": "Active"
}
```

Download

Project Snapshots

book-rest-controller



PUT /lms/book/updatebook

POST /lms/book/addbook

GET /lms/book/getbyTAS/{query}

GET /lms/book/getbooks

GET /lms/book/findtitle/{title}

GET /lms/book/findsubject/{subject}

GET /lms/book/findauthor/{author}

DELETE /lms/book/deletebook/{bookId}

POST

/lms/book/addbook

Parameters

Cancel

Reset

No parameters

Request body required

application/json

```
{  "bookId": 2,  "title": "Origin of Species",  "author": "Charles Darwin",  "subject": "Biology",  "isbn": "#82Bio",  "publisher": "Darwin",  "date": "2023-04-14",  "quantity": 1,  "availableQuantity": 1}
```

Execute

Responses

Code	Description	Links
200	OK	No links

Request URL

http://localhost:9090/lms/book/addbook

Server response

Code	Details
200	<div>Response body<div>Book Added Successfully!!</div></div>

GET

/lms/book/getbyTAS/{query}

Parameters

Cancel

Name	Description
query <small>* required</small>	Biology

Execute

Clear

Responses

Curl

curl -X 'GET' \ 'http://localhost:9090/lms/book/getbyTAS/Biology' \ -H 'accept: */*'

Request URL

http://localhost:9090/lms/book/getbyTAS/Biology

Server response

Code	Details
200	<div>Response body<div>[{ "bookId": 2, "title": "Origin of Species", "author": "Charles Darwin", "subject": "Biology", "isbn": "#82Bio", "publisher": "Darwin", "date": "2023-04-14", "quantity": 1, "availableQuantity": 1 }]</div></div>

Project Snapshots

boorrow-rest-controller



PUT /lms/borrow/returnbook

POST /lms/borrow/boorrowbook

GET /lms/borrow/getBorrowers

GET /lms/borrow/getBorrowers/{id}

GET /lms/borrow/getBorrowers/{id}

Parameters

Cancel

Name Description

id * required

integer(\$int64)

1

(path)

Execute

Clear

200

Response body

```
{
  "borrowingId": 1,
  "user": {
    "userId": 1,
    "firstName": "Prasad",
    "lastName": "Badwar",
    "email": "prasad@gmail.com",
    "password": "123456",
    "accountStatus": "Active"
  },
  "book": {
    "bookId": 1,
    "title": "Java",
    "author": "Javed sir",
    "subject": "Programming",
    "isbn": "#B1java",
    "publisher": "Manipal",
    "date": "2023-04-14",
    "quantity": 2,
    "availableQuantity": 2
  },
  "borrowDate": "2023-04-14",
  "dueDate": "2023-04-14",
  "returnDate": "2023-04-15",
  "status": "Returned"
}
```



Download

POST /lms/borrow/boorrowbook

Parameters

Cancel

Reset

No parameters

Request body ^{required}

application/json

```
{
  "borrowingId": 2,
  "userId": 1,
  "bookId": 1,
  "borrowDate": "2023-04-17",
  "dueDate": "2023-04-17",
  "returnDate": "2023-04-17",
  "status": "string"
}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:9090/lms/borrow/boorrowbook' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "borrowingId": 2,
    "userId": 1,
    "bookId": 1,
    "borrowDate": "2023-04-17",
    "dueDate": "2023-04-17",
    "returnDate": "2023-04-17",
    "status": "string"
  }'
```

Request URL

http://localhost:9090/lms/borrow/boorrowbook

Server response

Code

Details

400

undocumented Error:

Response body

Sorry, you can't borrow Book.Please pay fine first...



Download

POST

/lms/borrow/boorrowbook

Parameters

Cancel

Reset

No parameters

Request body required

application/json

```
{
  "borrowingId": 2,
  "userId": 2,
  "bookId": 2,
  "borrowDate": "2023-04-19",
  "dueDate": "2023-04-20",
  "returnDate": "2023-04-20",
  "status": "string"
}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
'http://localhost:9090/lms/borrow/boorrowbook' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
  "borrowingId": 2,
  "userId": 2,
  "bookId": 2,
  "borrowDate": "2023-04-19",
  "dueDate": "2023-04-20",
  "returnDate": "2023-04-20",
  "status": "string"
}'
```

Request URL

```
http://localhost:9090/lms/borrow/boorrowbook
```

Server response

Code

Details

200

Response body

User Borrowed Book Successfully!!

PUT

/lms/borrow/returnbook

Parameters

Cancel

Reset

No parameters

Request body required

application/json

```
{
  "borrowingId": 2,
  "userId": 2,
  "bookId": 2,
  "borrowDate": "2023-04-19",
  "dueDate": "2023-04-20",
  "returnDate": "2023-04-20",
  "status": "string"
}
```

Execute

Clear

Responses

Curl

```
curl -X 'PUT' \
'http://localhost:9090/lms/borrow/returnbook' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
  "borrowingId": 2,
  "userId": 2,
  "bookId": 2,
  "borrowDate": "2023-04-19",
  "dueDate": "2023-04-20",
  "returnDate": "2023-04-20",
  "status": "string"
}'
```

Request URL

```
http://localhost:9090/lms/borrow/returnbook
```

Server response

Code

Details

200

Response body

User Returned Book Successfully!!

Project Snapshots

reservation-rest-controller



PUT /lms/reserve/updatereservation

POST /lms/reserve/reservebook

GET /lms/reserve/getallreservation

DELETE /lms/reserve/deleterreservation/{reserveId}

POST

/lms/reserve/reservebook

Parameters

Try it out

No parameters

Request body required

application/json

Example Value

Schema

```
{
  "reservationId": 0,
  "userId": 0,
  "bookId": 0,
  "reservationDate": "2023-04-20"
}
```

Responses

Code	Description	Links
200	OK	No links

Media type

/

Controls Accept header.

Example Value

Schema

GET

/lms/reserve/getallreservation

Parameters

Cancel

No parameters

Execute

Responses

Code	Description	Links
200	OK	No links

Media type

/

Controls Accept header.

Example Value

Schema

```
[
  {
    "reservationId": 0,
    "user": {
      "userId": 0,
      "firstName": "string",
      "lastName": "string",
      "email": "string",
      "password": "string",
      "accountStatus": "string"
    },
    "book": {
      "bookId": 0,
      "title": "string",
      "author": "string",
      "subject": "string",
      "isbn": "string",
      "publisher": "string",
      "date": "2023-04-20",
      "quantity": 0,
      "availableQuantity": 0
    },
    "reservationDate": "2023-04-20"
  }
]
```

Project Snapshots

loan-rest-controller



PUT /lms/book/payfine/{loanid}

GET /lms/book/getloandata

GET /lms/book/finebyuserid/{userid}

GET /lms/book/getloandata

Parameters

Cancel

No parameters

Execute

Clear

Responses

Curl

```
curl -X 'GET' \  
'http://localhost:9090/lms/book/getloandata' \  
-H 'accept: */*'
```

Request URL

http://localhost:9090/lms/book/getloandata

Server response

Code

Details

200

Response body

```
{  
  "username": "badwar",  
  "email": "prasad@gmail.com",  
  "password": "123456",  
  "accountStatus": "Active"  
},  
"book": {  
  "bookId": 1,  
  "title": "Java",  
  "author": "Javeed sir",  
  "subject": "Programming",  
  "isbn": "#B1java",  
  "publisher": "Manipal",  
  "date": "2023-04-14",  
  "quantity": 2,  
  "availableQuantity": 2  
},  
"dueDate": "2023-04-20",  
"fine": 500,  
"status": "Fined"
```



Download

PUT

/lms/book/payfine/{loanid}

Parameters

Cancel

Name

Description

loanid * required

integer(\$int64)

(path)

16

Execute

Clear

Responses

Curl

```
curl -X 'PUT' \  
'http://localhost:9090/lms/book/payfine/16' \  
-H 'accept: */*'
```

Request URL

http://localhost:9090/lms/book/payfine/16

Server response

Code

Details

200

Response body

User Paid Fine successfully!



Download

Project Snapshots

reports-controller



GET

/lms/book/getUserReport/{id}

GET

/lms/book/getBookReport/{id}

Thank you!

