Library Management System
Design Document

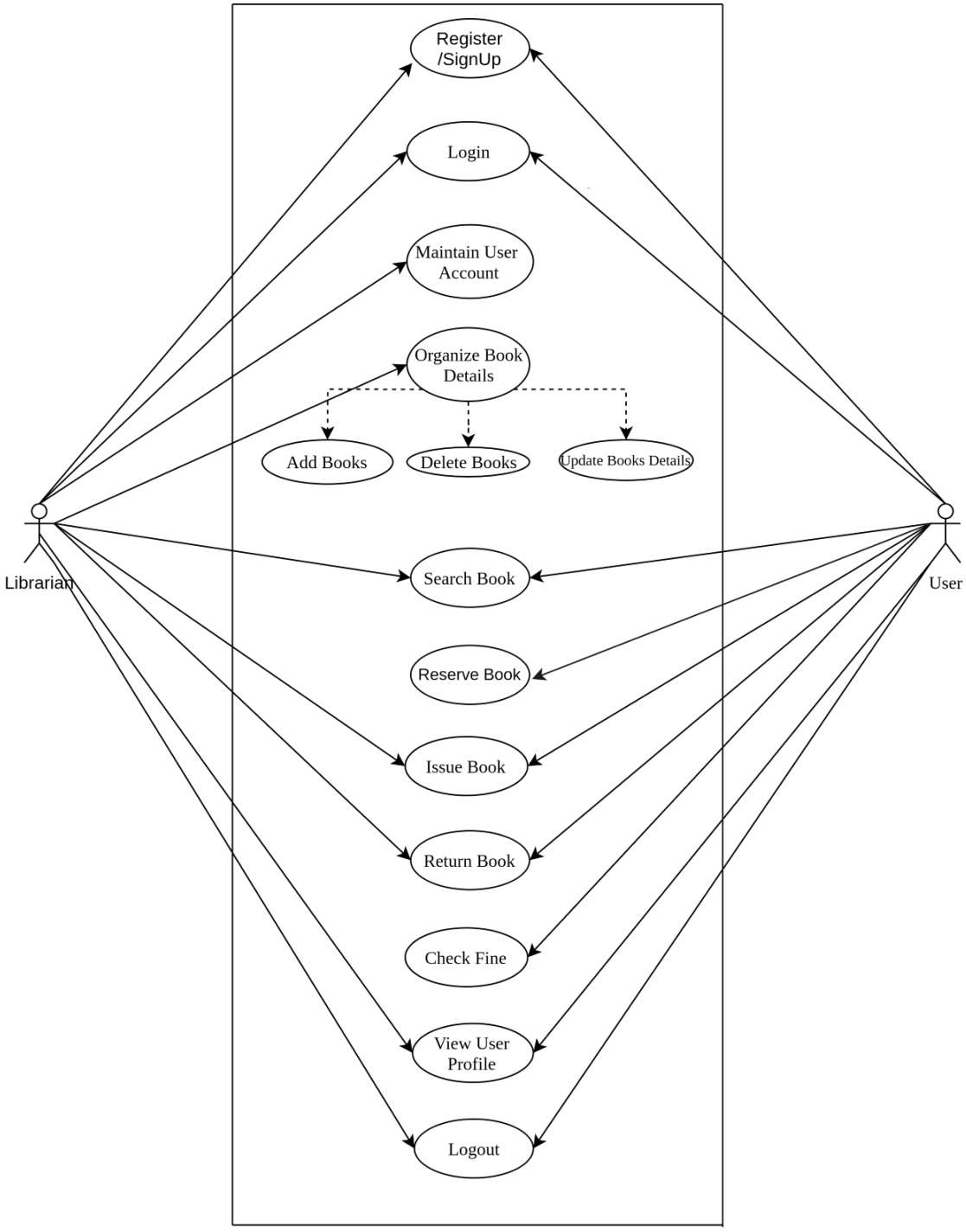
by
Ankita Paul MT2020053
Neha Kothari MT2020010
Rushi Shah MT2020113
Shubhi Maheshwari MT2020167
Shivam Govind Prasad Juyal MT2020018

Introduction:

The aim of developing this web application is to design, develop, and implement an online library book management system. The main purpose of this project is to maintain an easy circulation of books between the library and its users that may include students and faculty members. The current system being operated manually requires each student to register in the library, get a library card, select the book, and manually enter the issue/return date in catalogue and fine calculation. Also, the librarian needs to maintain the record of huge volumes of books and track the issue/return for each copy.

Content:

1. Use-Case Diagram :

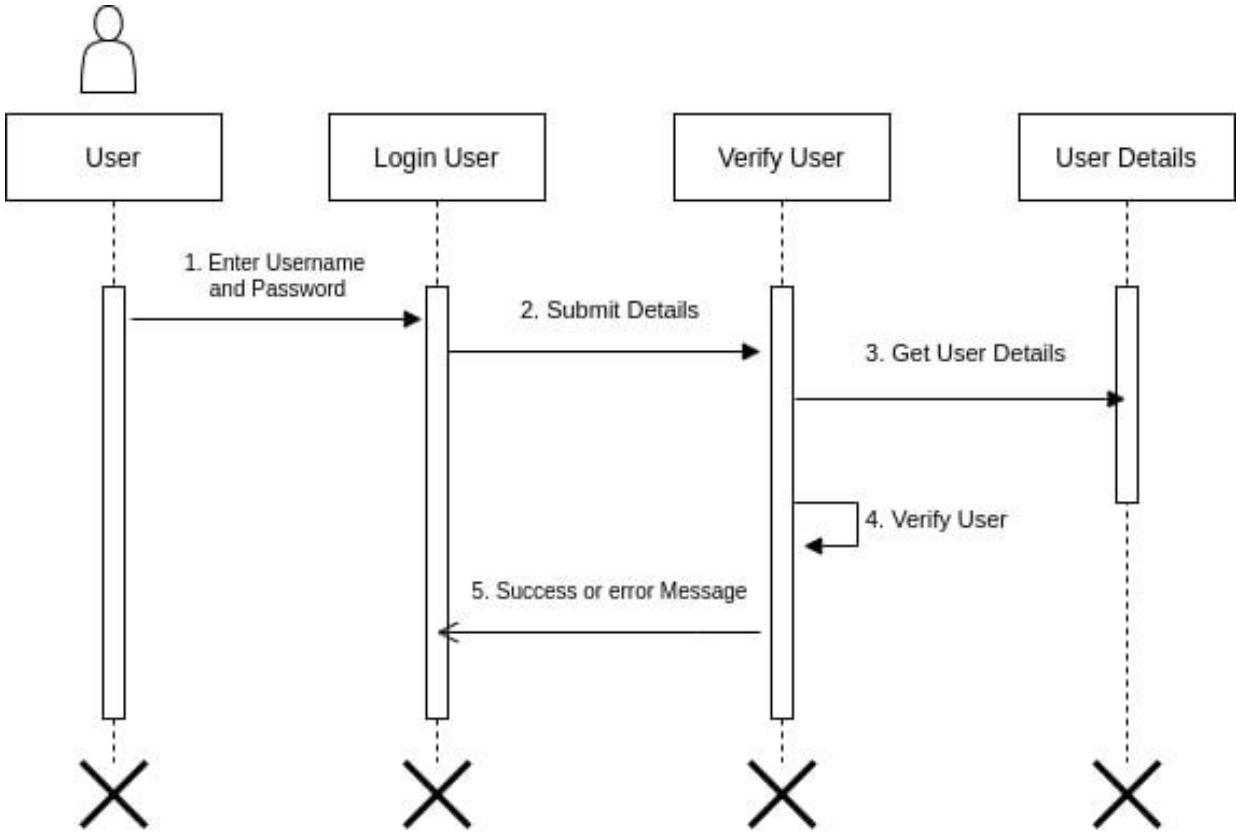


2. Sequence Diagrams:

2.1 Login User

Process:

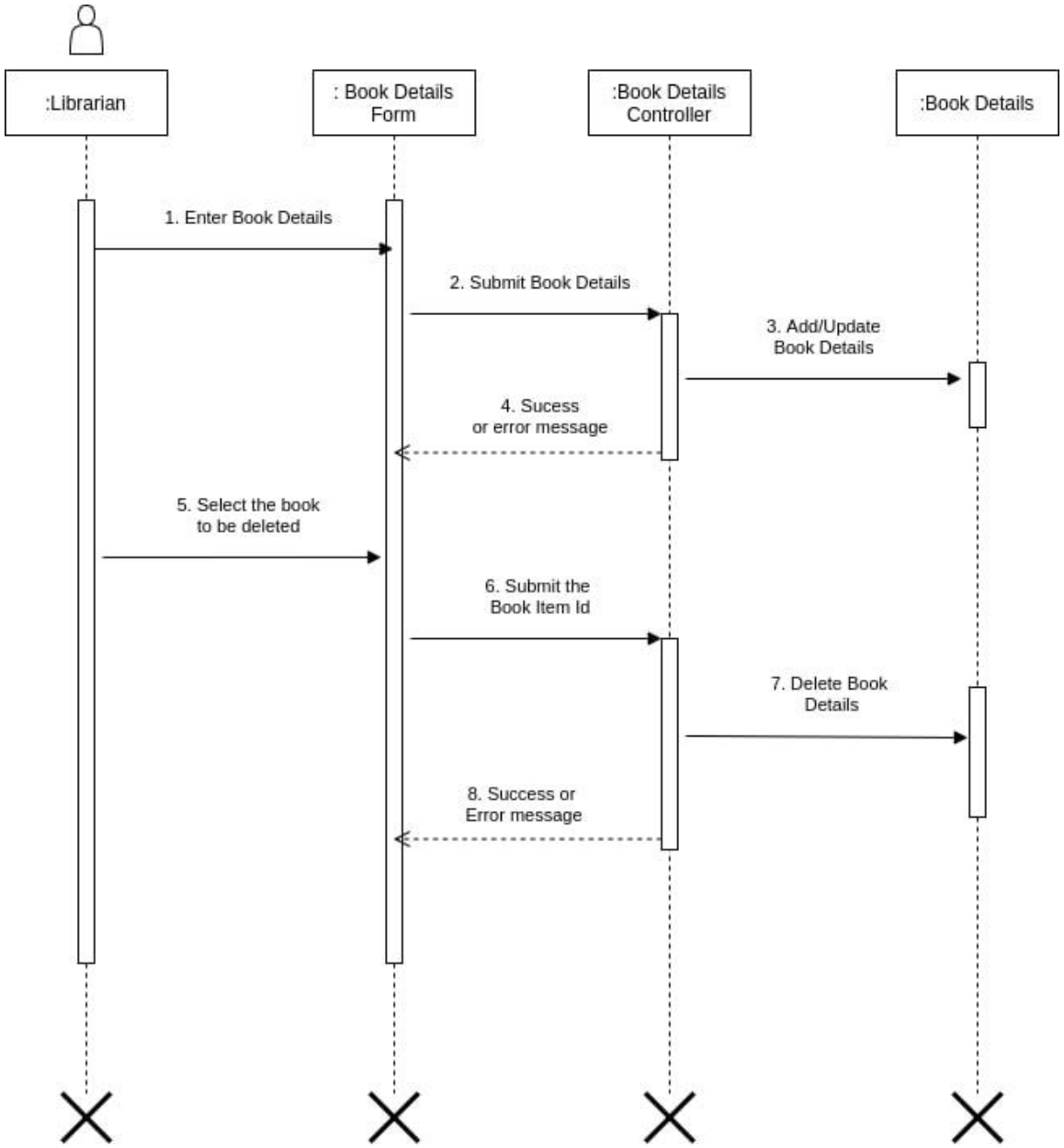
1. Enter username and password.
2. Submit details.
3. Get user details from the user's data store.
4. Verify user details.
5. Issue a success or error message.



2.2 Maintain Books

Process

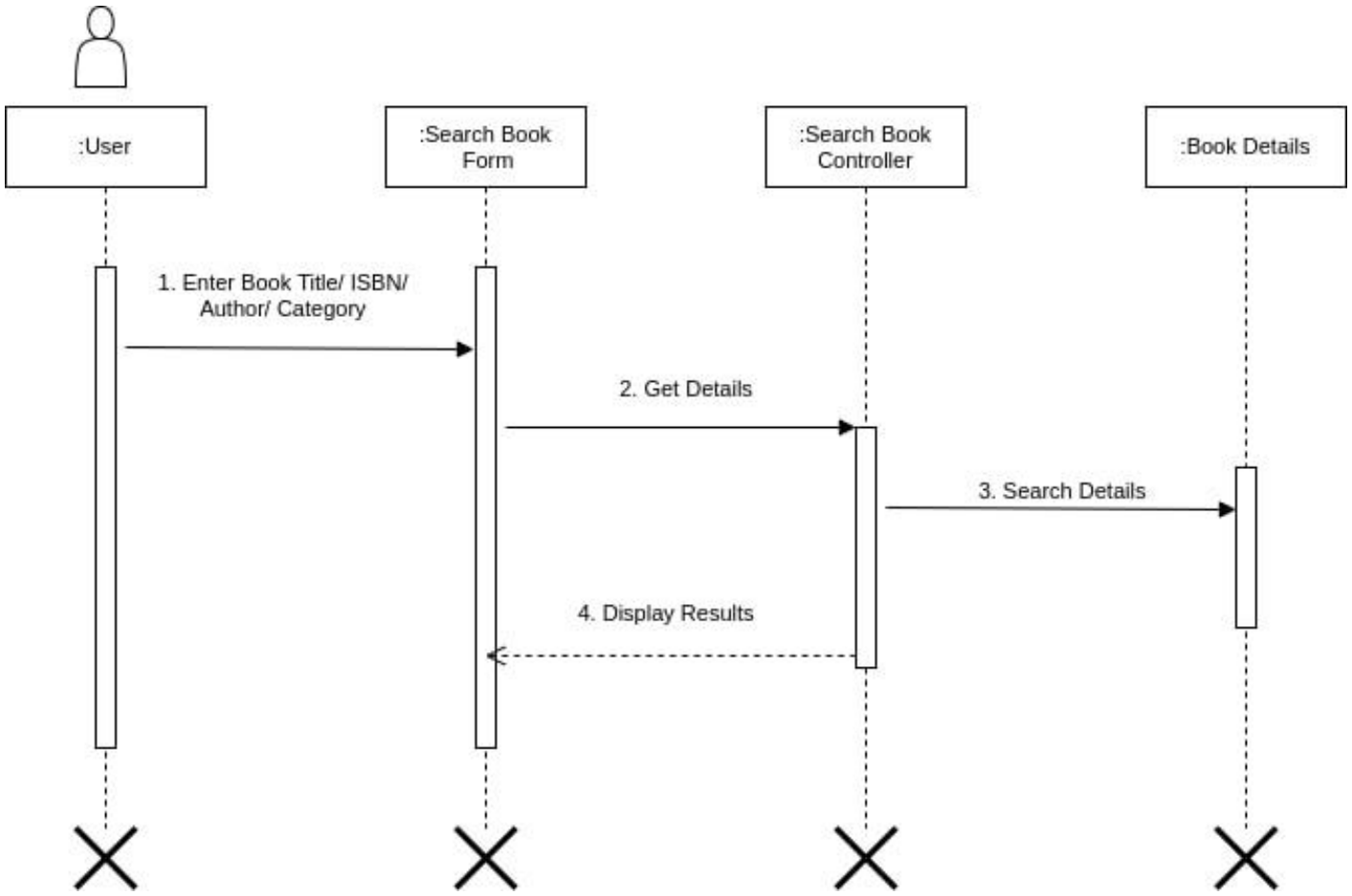
1. Enter Book Details
2. Submit Book Details
3. Add/Update Book Details
4. Return Success or error message
5. For deletion, select the book to be deleted
6. Submit Book Item ID
7. Delete the Book
8. Return success or error message



2.3 Search Book

Process

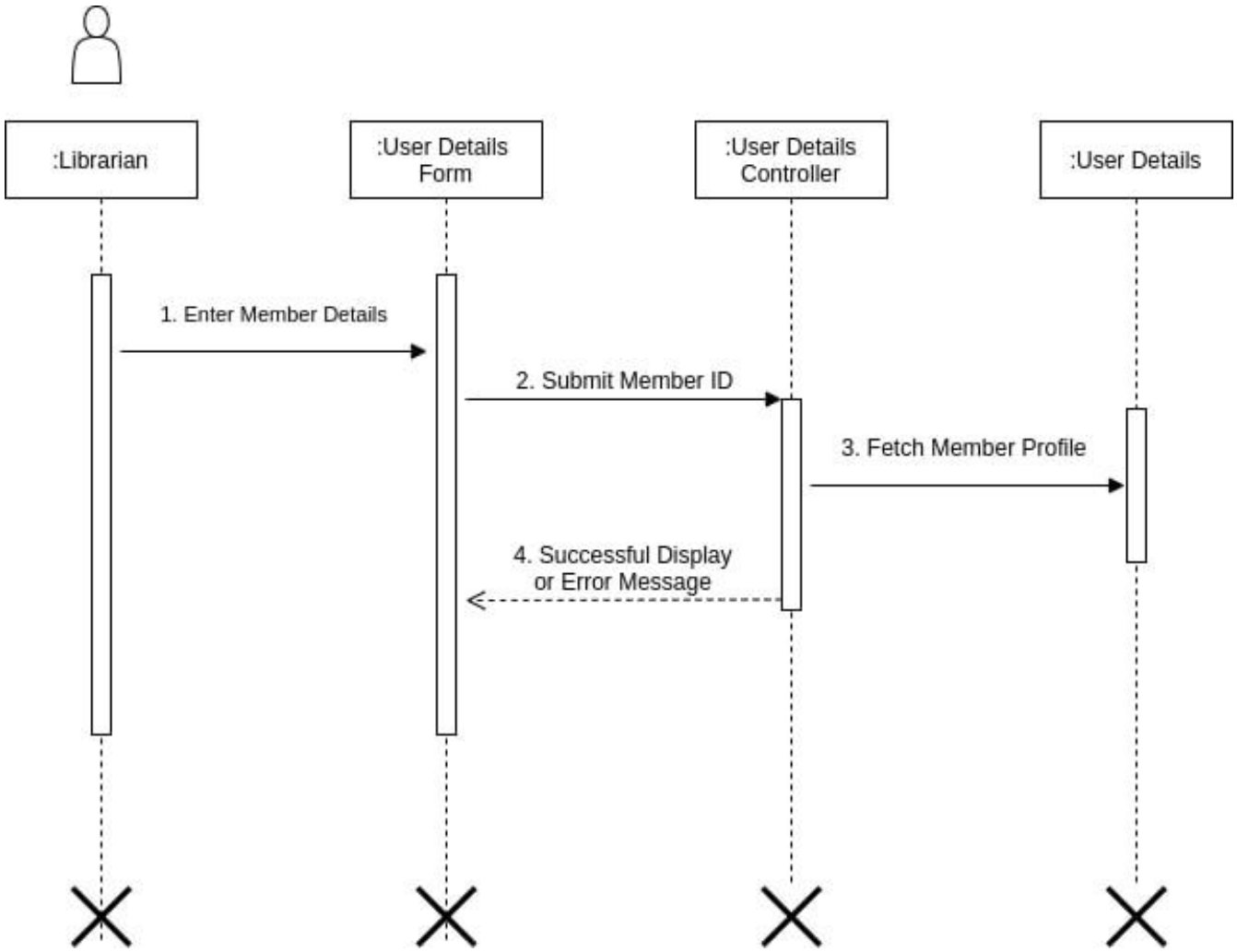
1. Enter the book title, ISBN, category, author
2. Submit details
3. Search for Book
4. Display successful search results or error message



2.4 Manage User Account

Process

1. Enter Member Details
2. Submit Member ID
3. Fetch Member Profile that includes list of books issued/reserved/fine pending
4. Return successful search results or error message

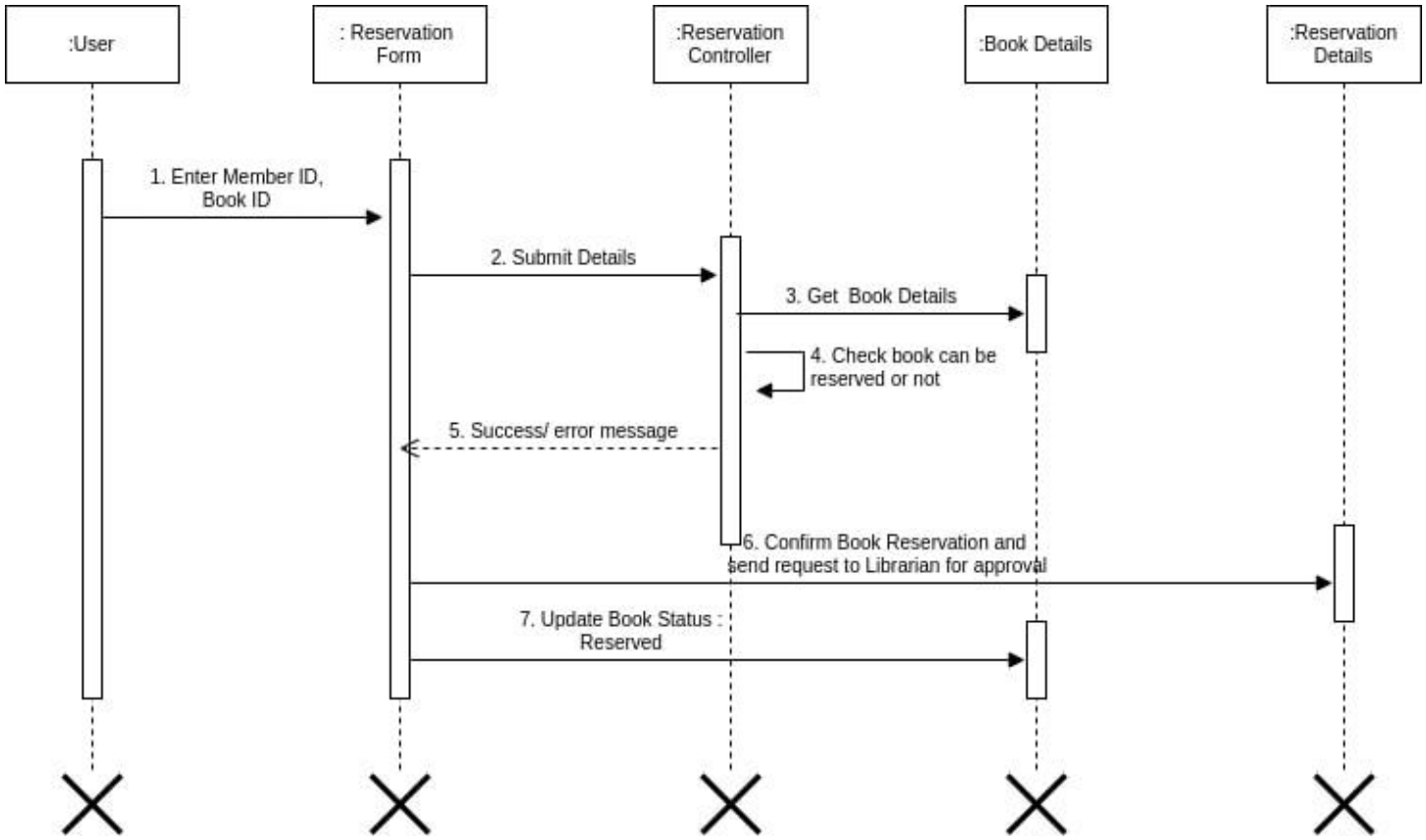


2.5 Reserve Book

The member can reserve the book if available and send the issue book request to the librarian.

Process

1. Enter member ID and book ID.
2. Submit details.
3. Get book details.
4. Check whether the book can be reserved or not.
5. Success or error message.
6. Confirm reservation details on success and send approval requests to the Librarian.
7. Update Book status : Reserved

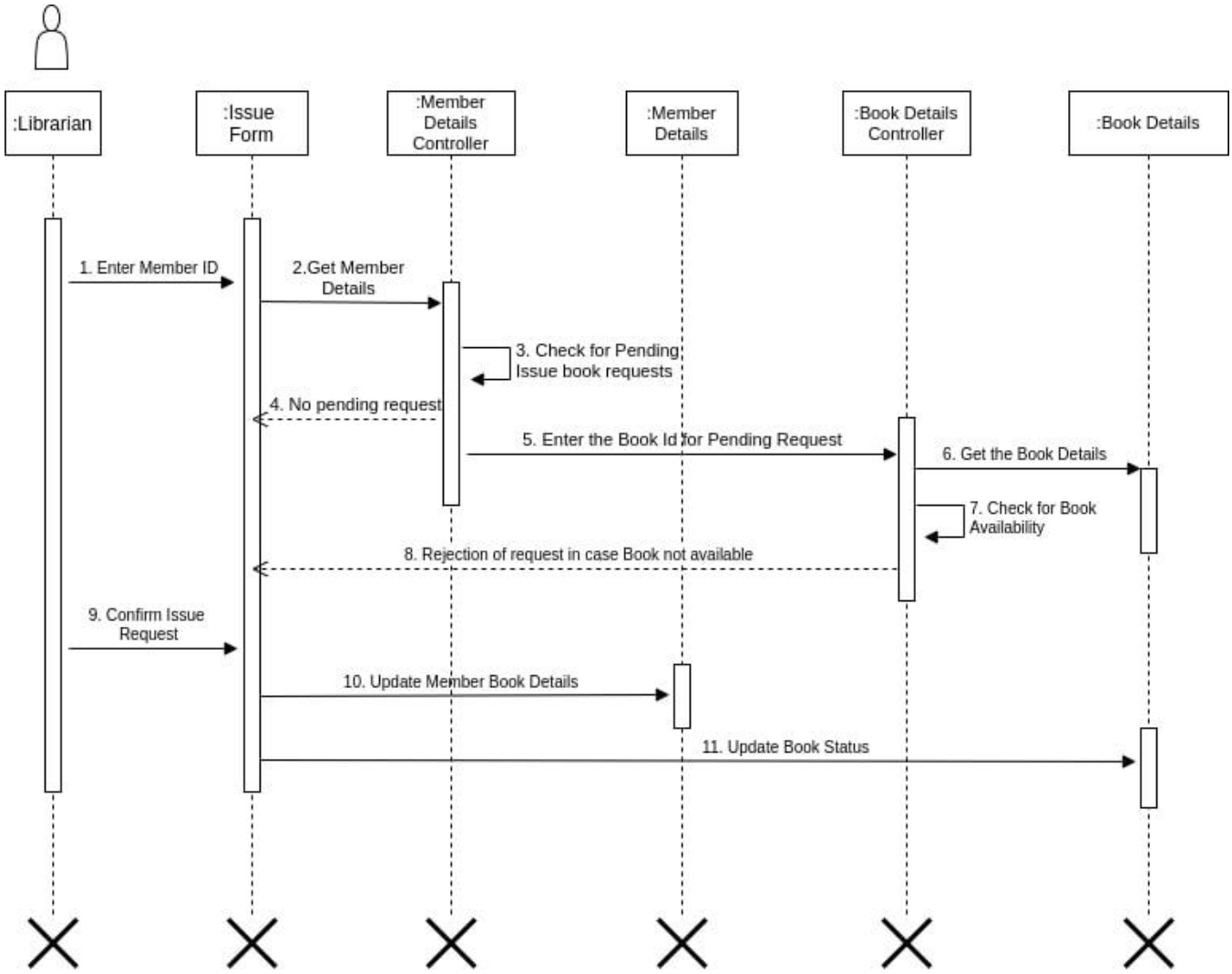


2.6 Reserve-Issue Book

The member reserves the book for issue by the librarian and the librarian approves/rejects the book issue request. In case the request is not approved in less than 48 hours, the book automatically becomes available.

Process

1. Enter Member ID
2. Get Member Book Details
3. Check for pending issue approval requests
4. Enter the Book ID for pending book issue request (if exist)
5. Get the Book details
6. Check for the availability of the book
7. Reject the approval request if book not available
8. If available, confirm the issue request
9. Update the Book Status : Issued
10. Update the Member Book Details

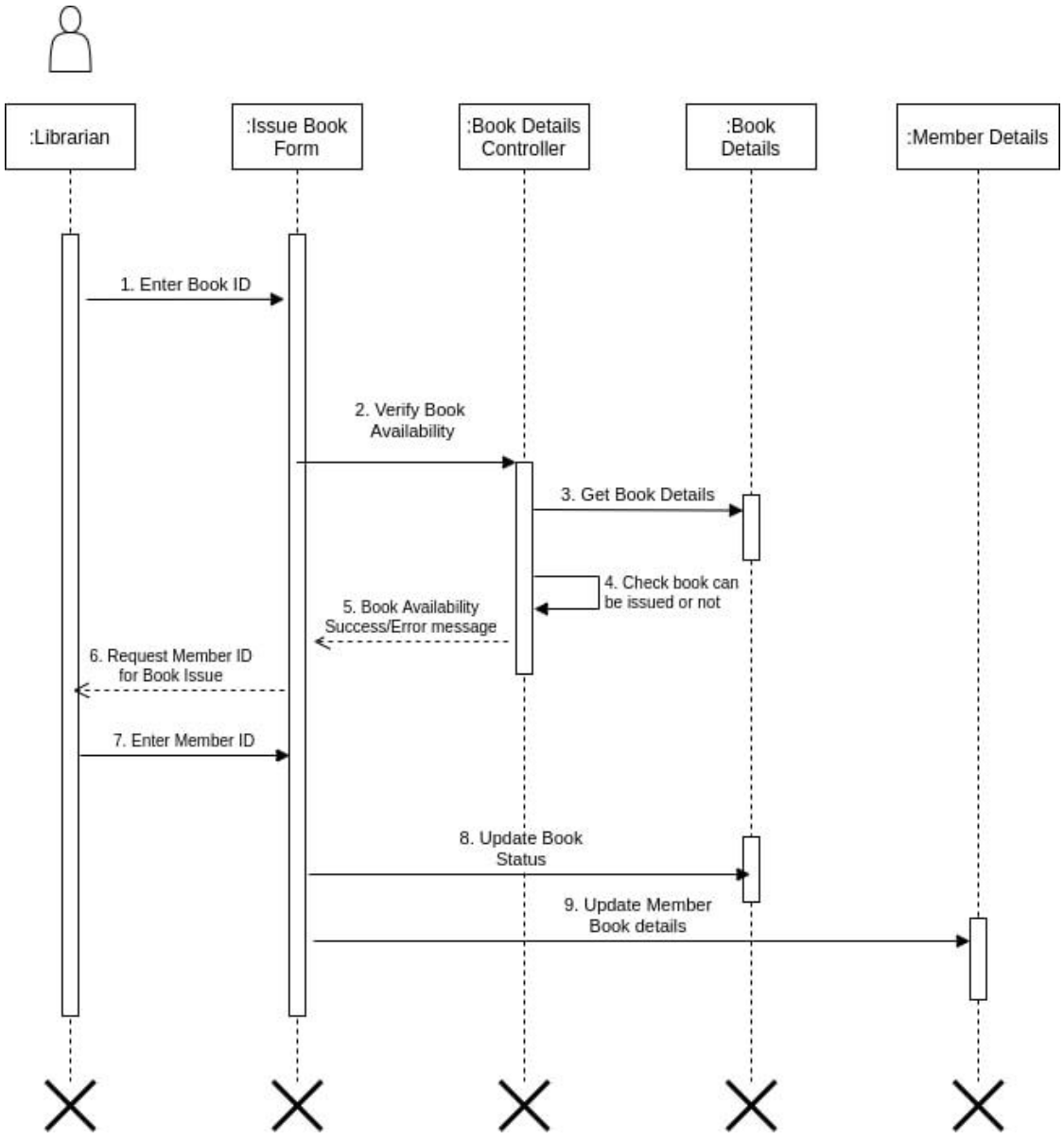


2.7 Direct-Issue Book

A member goes to the librarian directly for issuing the book.

Process

1. Enter the Book ID
2. Get the book details
3. Check for the book availability
4. If Book available, request for the Member ID
5. Librarian provides the Member ID
6. Confirm the book issue
7. Update the book status : Issued
8. Update the member book details

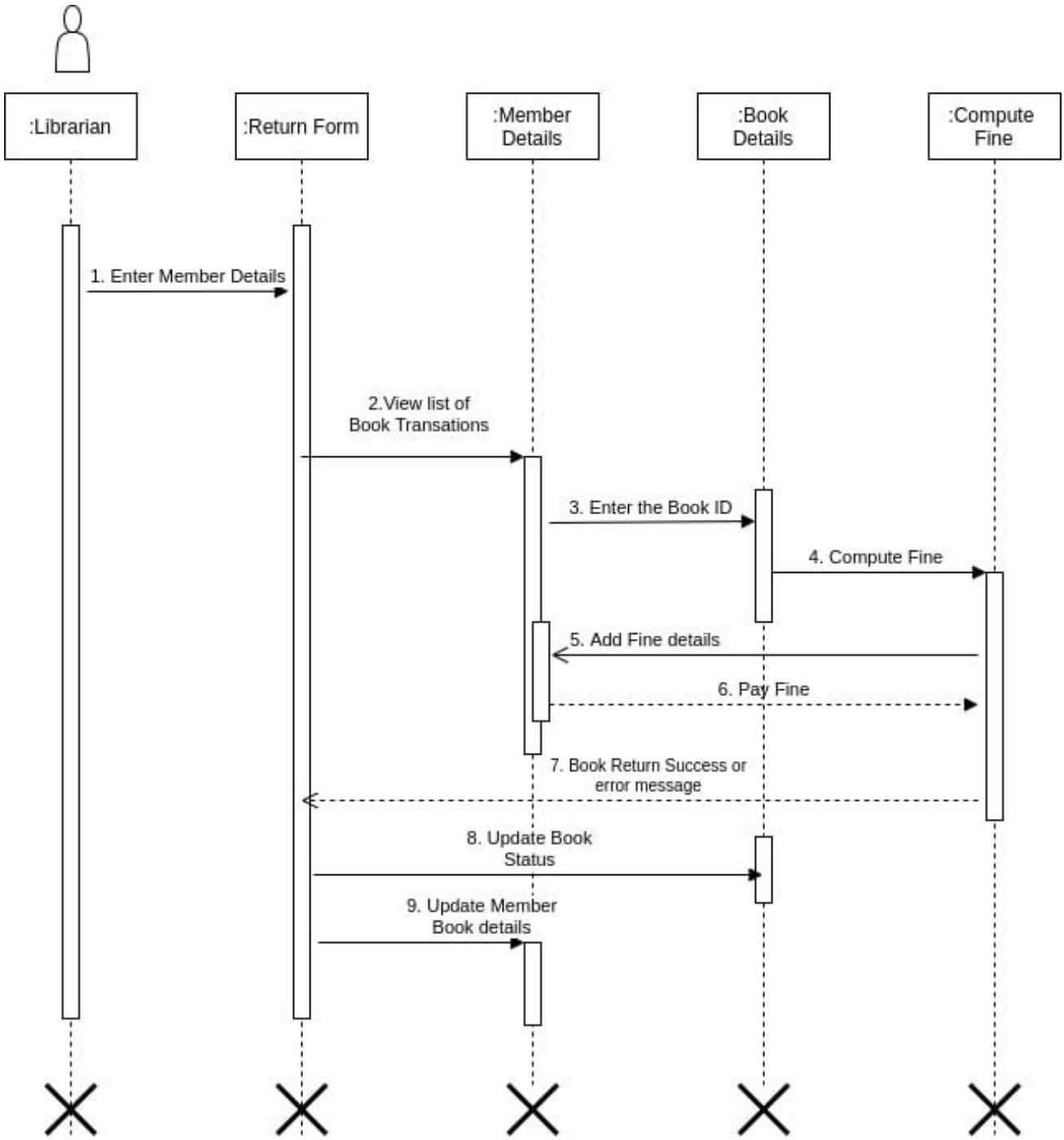


2.8 Return Book

The librarian confirms the book return when the user requests it, computes the fine if the actual return date exceeds specified book return date and accepts the fine payment.

Process

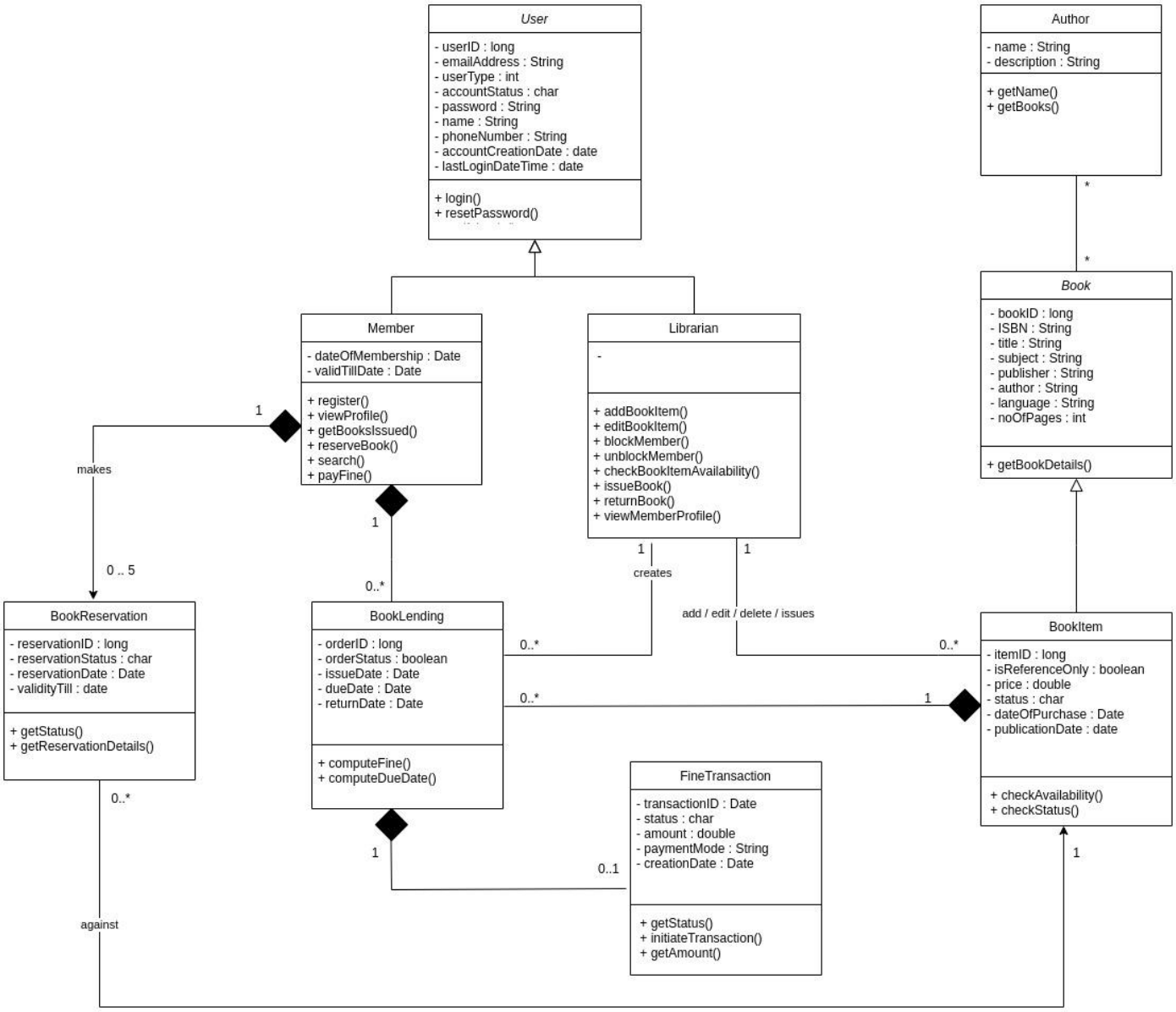
1. Enter the Member ID
2. A list of book transactions is displayed that includes issued books, reserved book details
3. Enter the book ID of the book to be returned
4. Compute the fine on the book (if exists)
5. Add the fine details for the member
6. Accept the fine payment from the member.
7. Confirm the book return if fine is paid or return the error message
8. Update the book status : Available
9. Update the member book details


:Librarian
:Return Form
:Member Details
:Book Details
:Compute Fine
1. Enter Member Details
2.View list of Book Transations
3. Enter the Book ID
4. Compute Fine
5. Add Fine details
6. Pay Fine
7. Book Return Success or error message
8. Update Book Status
9. Update Member Book details

3. Class Diagram:

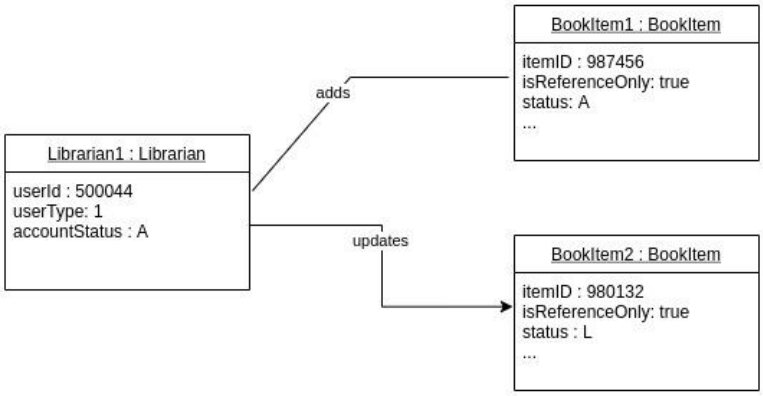
Here are the main classes of our Library Management System:

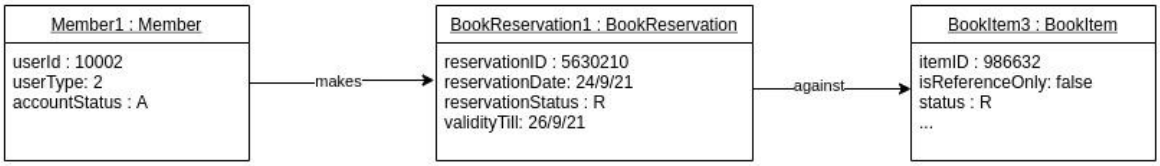
- **Book:** The basic building block of the system. Every book will have ISBN, Title, Subject, Publishers, etc.
- **BookItem:** Any book can have multiple copies, each copy will be considered a book item in our system. Each book item will have a unique book_item_id.
- **User:** We will have two types of user accounts in the system, one will be a general member, and the other will be a librarian.
- **BookReservation:** Responsible for managing reservations against book items.
- **BookLending:** Manage the checking-out of book items and will be processed as orders.
- **FineTransaction:** This class will be responsible for calculating and collecting fines from library members and keeping a record of the transaction done.
- **Author:** This class will encapsulate a book author.

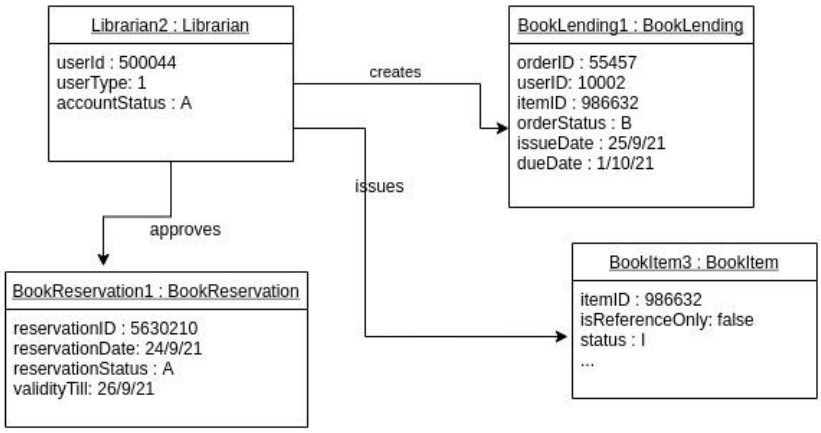

User
- userID : long
- emailAddress : String
- userType : int
- accountStatus : char
- password : String
- name : String
- phoneNumber : String
- accountCreationDate : date
- lastLoginDateTime : date
+ login()
+ resetPassword()
Author
- name : String
- description : String
+ getName()
+ getBooks()
*
*
Book
- bookID : long
- ISBN : String
- title : String
- subject : String
- publisher : String
- author : String
- language : String
- noOfPages : int
+ getBookDetails()
Member
- dateOfMembership : Date
- validTillDate : Date
+ register()
+ viewProfile()
+ getBooksIssued()
+ reserveBook()
+ search()
+ payFine()
Librarian
-
+ addBookItem()
+ editBookItem()
+ blockMember()
+ unblockMember()
+ checkBookItemAvailability()
+ issueBook()
+ returnBook()
+ viewMemberProfile()
1
makes
0 .. 5
1
0..*
1
creates
1
add / edit / delete / issues
BookReservation
- reservationID : long
- reservationStatus : char
- reservationDate : Date
- validityTill : date
+ getStatus()
+ getReservationDetails()
BookLending
- orderID : long
- orderStatus : boolean
- issueDate : Date
- dueDate : Date
- returnDate : Date
+ computeFine()
+ computeDueDate()
0..*
0..*
BookItem
- itemID : long
- isReferenceOnly : boolean
- price : double
- status : char
- dateOfPurchase : Date
- publicationDate : date
+ checkAvailability()
+ checkStatus()
0..*
0..*
1
1
1
0..1
FineTransaction
- transactionID : Date
- status : char
- amount : double
- paymentMode : String
- creationDate : Date
+ getStatus()
+ initiateTransaction()
+ getAmount()
against

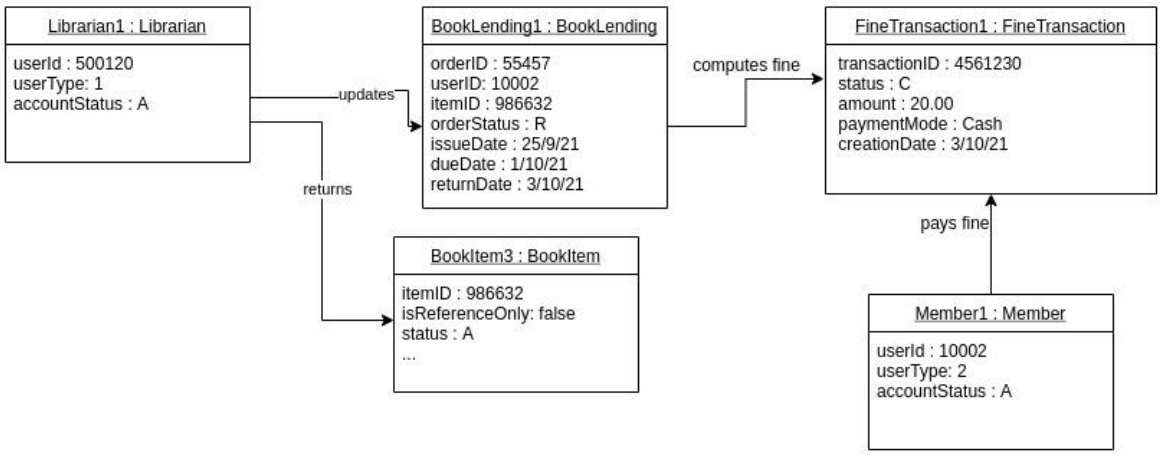
4. Object Diagram:

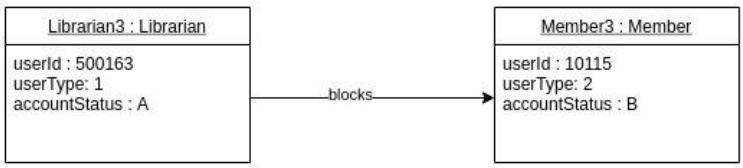
- A Librarian (userType = 1) can add, edit or delete a book. The status of the BookItem implies whether it's Available('A'), Lost('L'), Reserved('R') or Issued('I').
- A Librarian can block a Member. The accountStatus of that member would change from 'A'(Active) to 'B'(blocked). If a librarian unblocks a blocked Member, the accountStatus would change back from 'B' to 'A'.
- If the membership validity of a member expires, then the accountStatus would change to 'I'(Inactive).
- A Member (usertype = 2 and accountStatus = 'A') can reserve a BookItem if it's isReferenceOnly = false and if it's status is 'A'. The reservation is valid for 2 days from the day of reservation. The reservationStatus of the BookReservation object will be 'R'(Reserved).
- A Librarian can approve a BookReservation and change it's reservationStatus to 'A'(Approved) and issue the BookItem to the Member.
- If the BookItem isn't issued within 2 days, the reservationStatus will change to 'D'(Denied), the reservation will be invalid and the BookItem's status would change back to 'A'(Available).
- When a Librarian issues a BookItem, the status of the BookItem changes to 'I', and a new BookLending object is created. The BookLending object has details of the MemberId (userId of the member), the itemId (bookItemId), issuedBy (userId of the librarian), the issueDate and the dueDate.
- When a BookItem is returned, the returnDate is updated and the status of the BookItem changes back to 'A'(available). The fine computed depends on the positive difference between the return date and the due date of the BookLending object.
- A Member has to pay the Fine for the corresponding BookLending order and the transaction details are updated. If the fine is paid, the status of the transaction is 'C'(completed), else it will be 'P'(Pending).


BookItem1 : BookItem
itemID : 987456
isReferenceOnly: true
status: A
...
adds
Librarian1 : Librarian
userId : 500044
userType: 1
accountStatus : A
updates
BookItem2 : BookItem
itemID : 980132
isReferenceOnly: true
status : L
...


Member1 : Member
userId : 10002
userType: 2
accountStatus : A
makes
BookReservation1 : BookReservation
reservationID : 5630210
reservationDate: 24/9/21
reservationStatus : R
validityTill: 26/9/21
against
BookItem3 : BookItem
itemID : 986632
isReferenceOnly: false
status : R
...

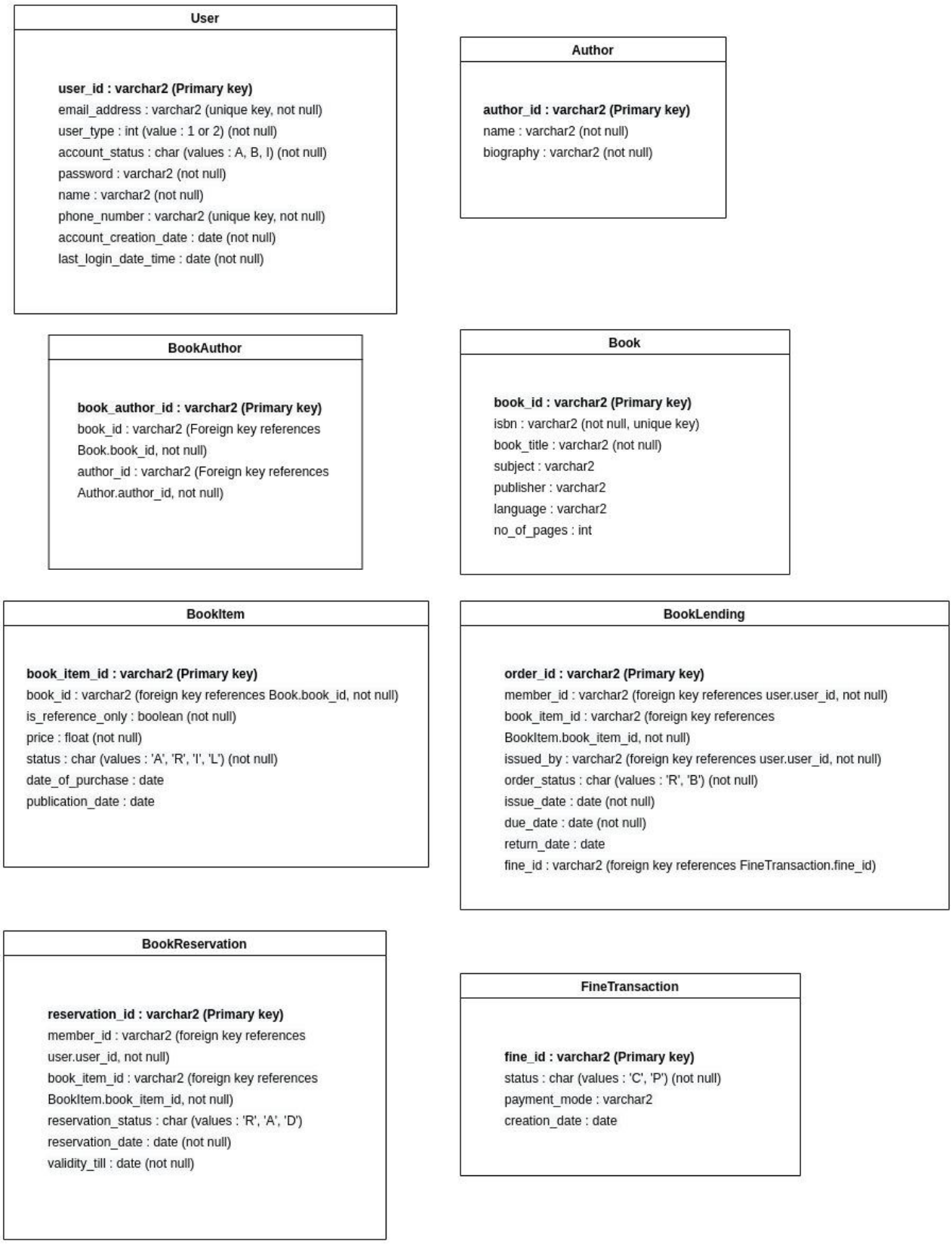

Librarian2 : Librarian
userId : 500044
userType: 1
accountStatus : A
creates
BookLending1 : BookLending
orderID : 55457
userID: 10002
itemID : 986632
orderStatus : B
issueDate : 25/9/21
dueDate : 1/10/21
issues
approves
BookReservation1 : BookReservation
reservationID : 5630210
reservationDate: 24/9/21
reservationStatus : A
validityTill: 26/9/21
BookItem3 : BookItem
itemID : 986632
isReferenceOnly: false
status : I
...


Librarian1 : Librarian
userId : 500120
userType: 1
accountStatus : A
updates
BookLending1 : BookLending
orderID : 55457
userID: 10002
itemID : 986632
orderStatus : R
issueDate : 25/9/21
dueDate : 1/10/21
returnDate : 3/10/21
computes fine
FineTransaction1 : FineTransaction
transactionID : 4561230
status : C
amount : 20.00
paymentMode : Cash
creationDate : 3/10/21
returns
BookItem3 : BookItem
itemID : 986632
isReferenceOnly: false
status : A
...
pays fine
Member1 : Member
userId : 10002
userType: 2
accountStatus : A


Librarian3 : Librarian
userId : 500163
userType: 1
accountStatus : A
blocks
Member3 : Member
userId : 10115
userType: 2
accountStatus : B

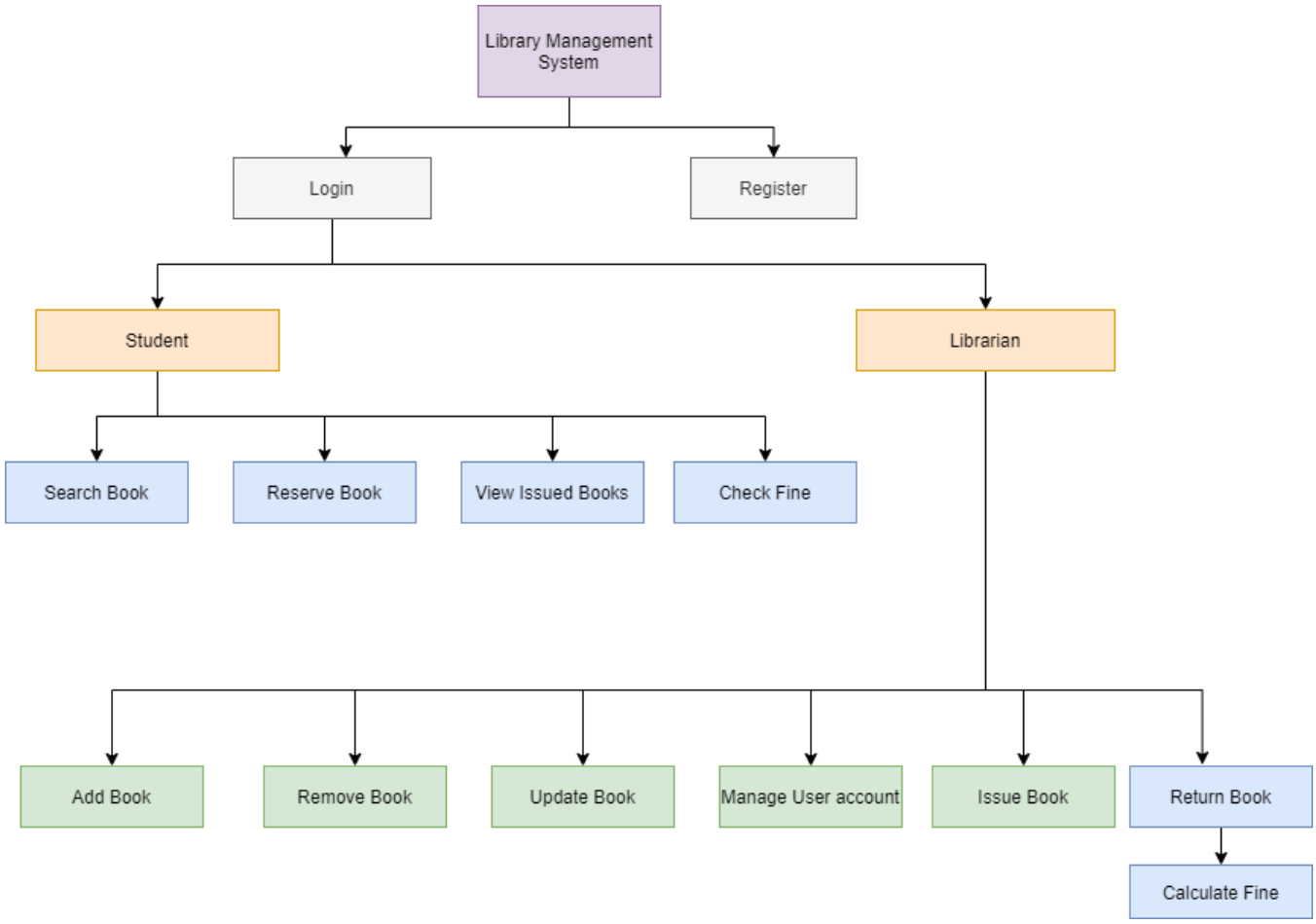
5. Data Model

Following is the relational data model that we have created for the Library Management System.

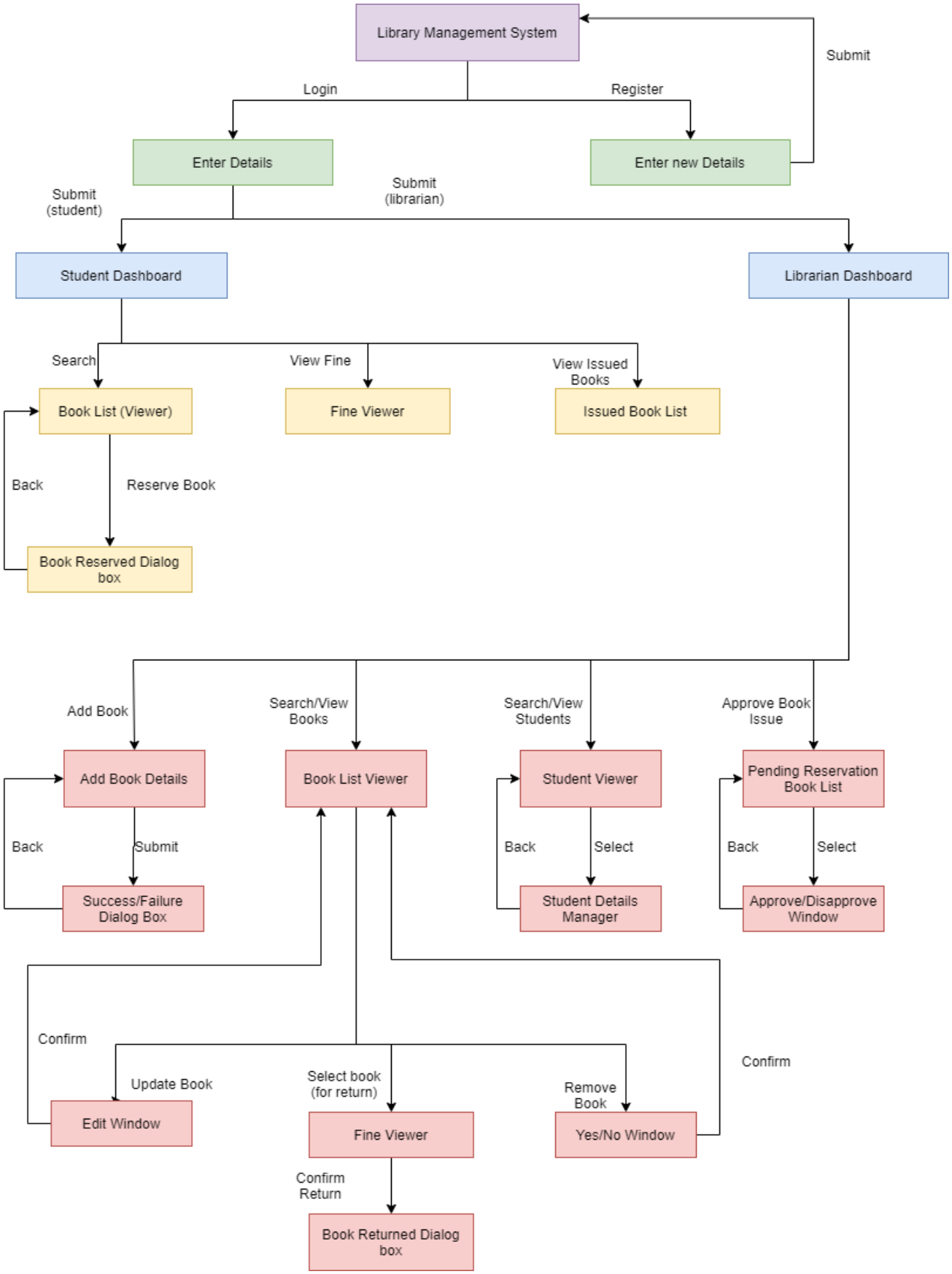


6. UI Design

6.1 Hierarchy of Screens:

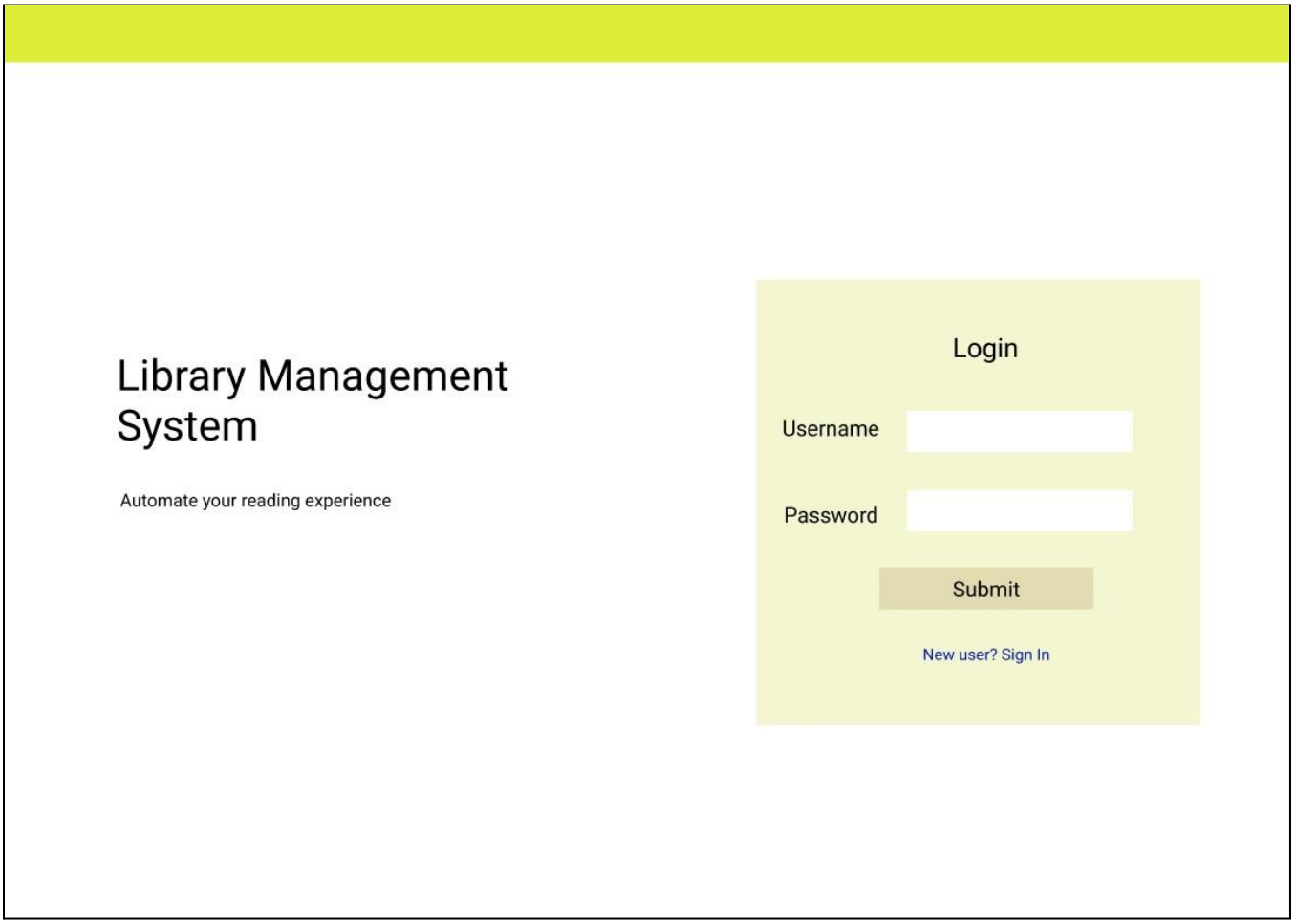


6.2 Navigational Web Diagram:

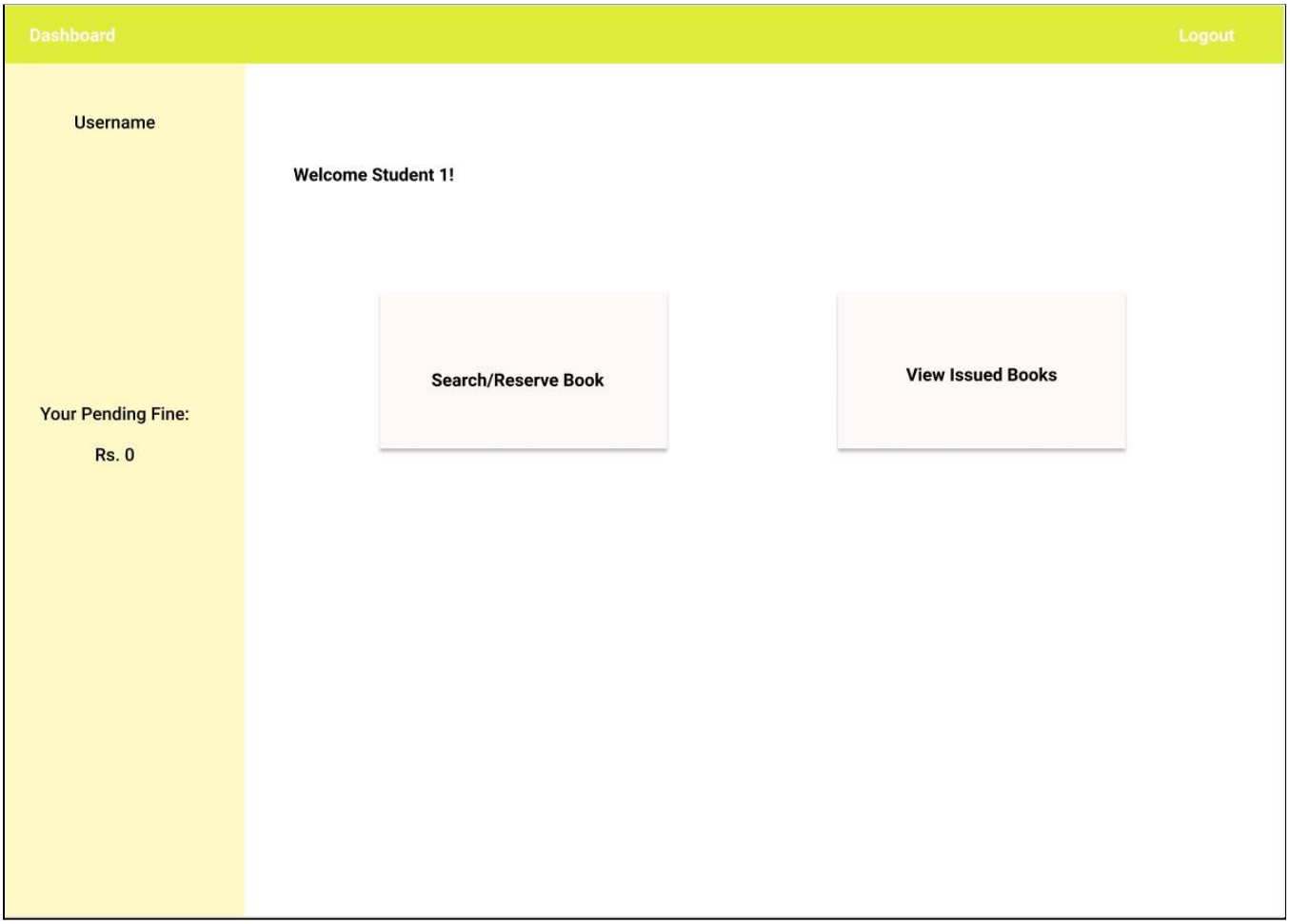


6.3 Screen Sketches

6.3.1 Welcome screen - The first screen visible when a user enters. This screen can lead to three pages, student dashboard or librarian dashboard (based on whether the username/email id stored in the database is a student or a librarian), or to the registration page (not depicted here) which looks similar to the login page.

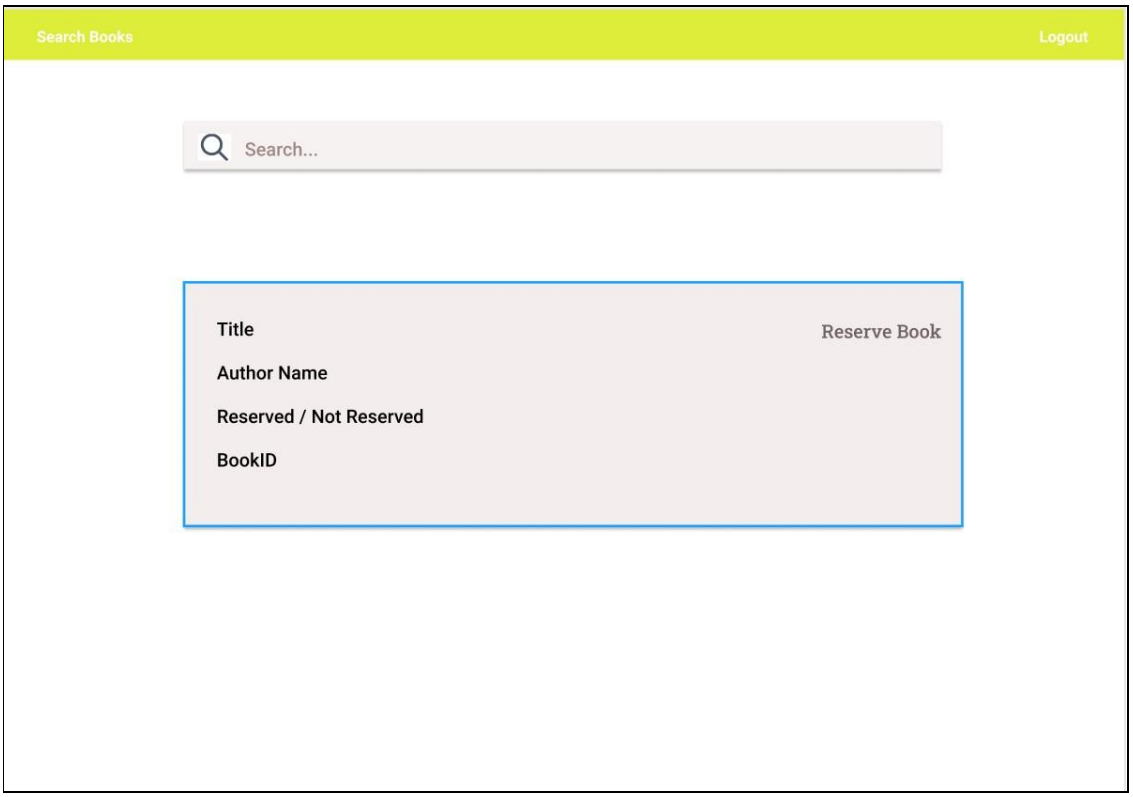


6.3.2 Student Dashboard - Screen is visible to students after login. It displays the current total fine accumulated for the student. It can move to search/ reserve book page or to view issued book page



6.3.2a Search / Reserve Books: Students can search for the book he or she wants. After this the required book appears (if it is present in the system) and it can be selected to reserve the book.

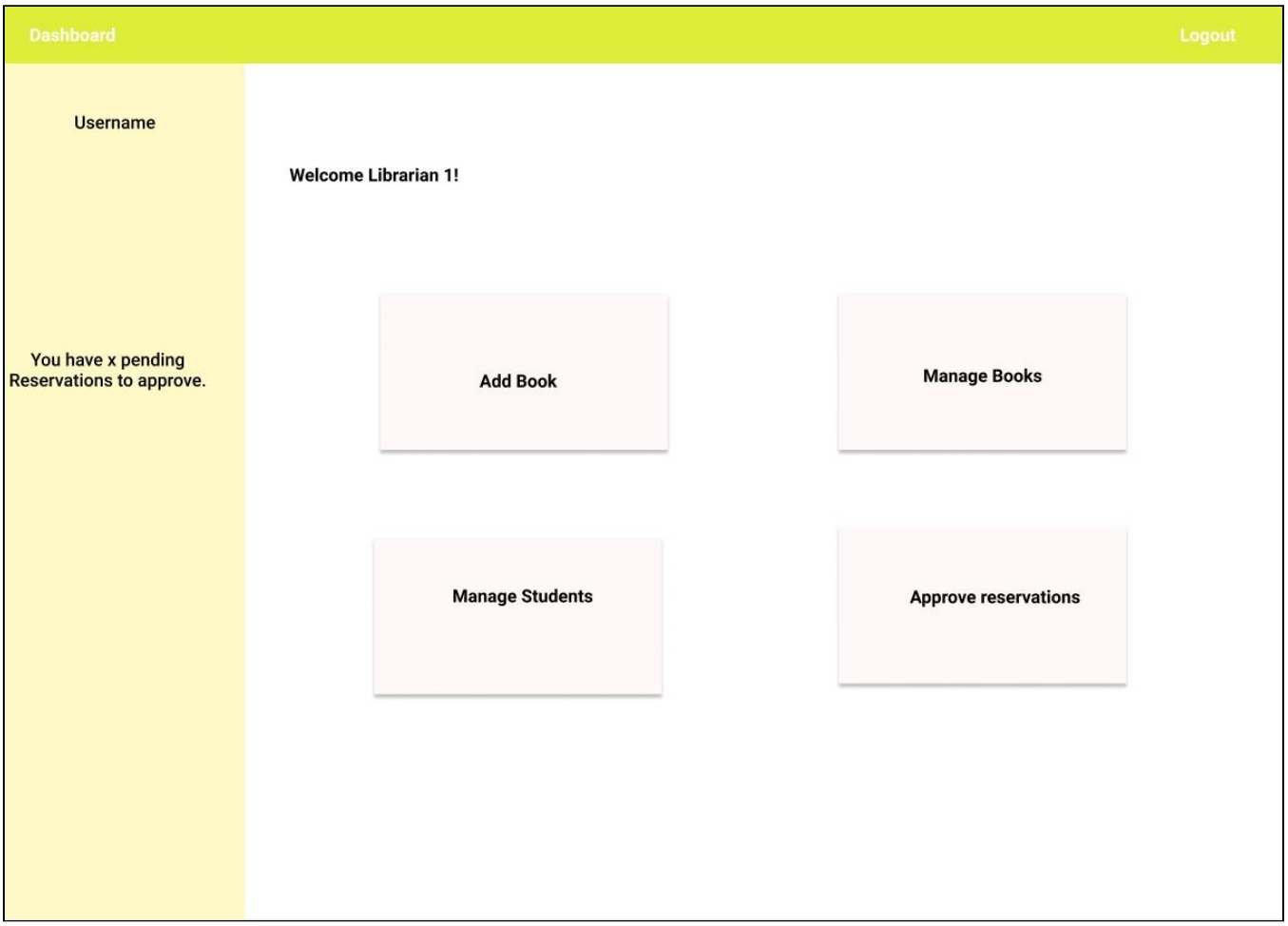
6.3.2b View Issued Books : It shows a list of all the books issued by the student and their information.



Search/ Reserve Book

View Issued Books

6.3.3 Librarian Dashboard : It is the first screen visible after the librarian logs in. It shows how many books are requested by the students to be issued (pending book issues) and can further move on to other pages with different functionalities.



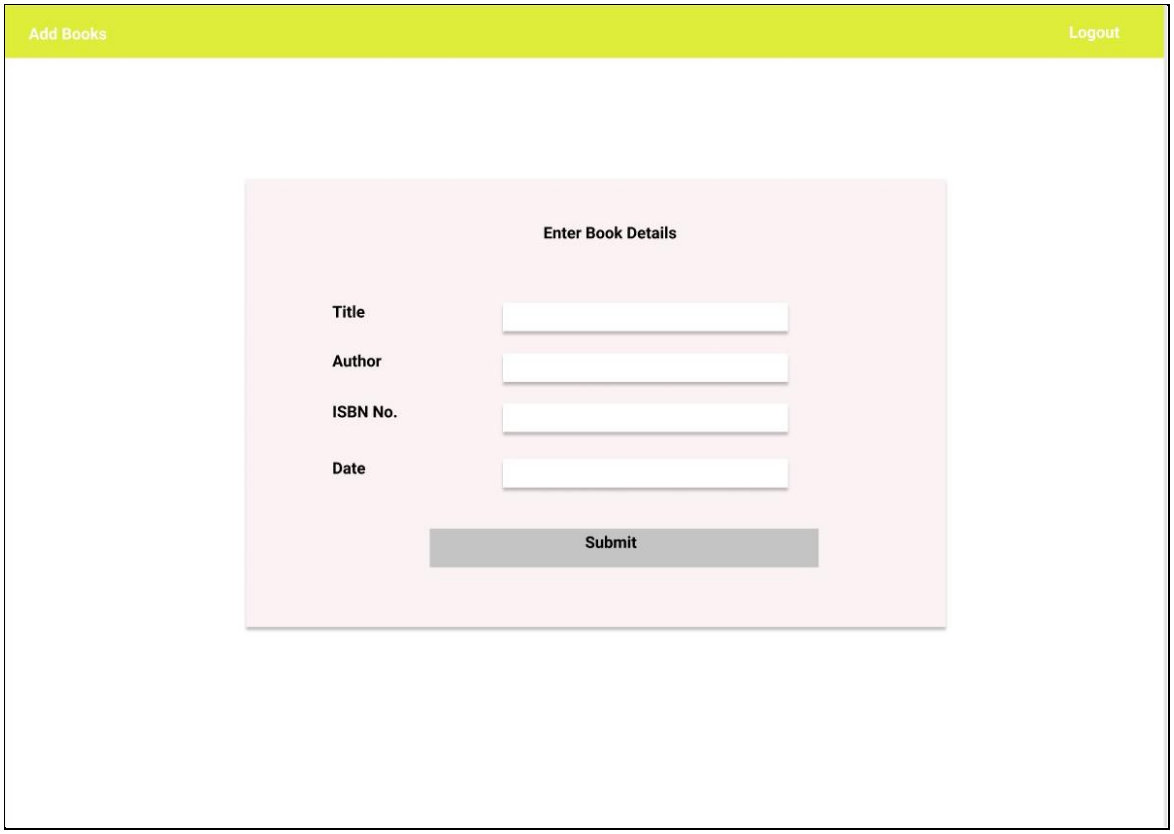
6.3.3a Add Book - Allows the Librarian to add a new book to the system.

6.3.3b Manage books - Allows the Librarian to search for a book and then perform the following:

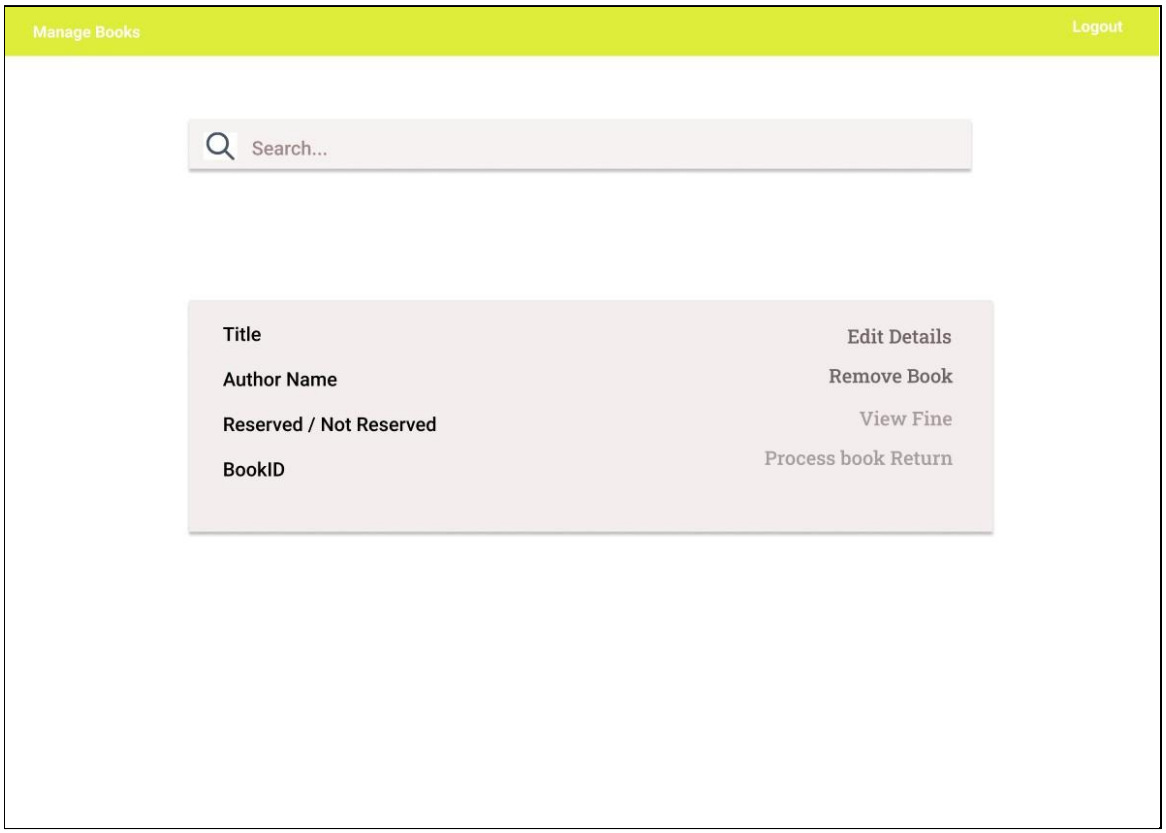
a. Edit details of the books
b. Remove the book from the system, i.e. students will no longer be able to access the book
c. In case the book has been issued, view fine associated to the book (This feature is available only if the book has been issued by some student)
d. Process book return - updates the system to show that the book has been returned by the student. (This feature is available only if the book has been issued by some student)

6.3.3c Manage Users - Allows the Librarian to search for a user and view all details of the user.

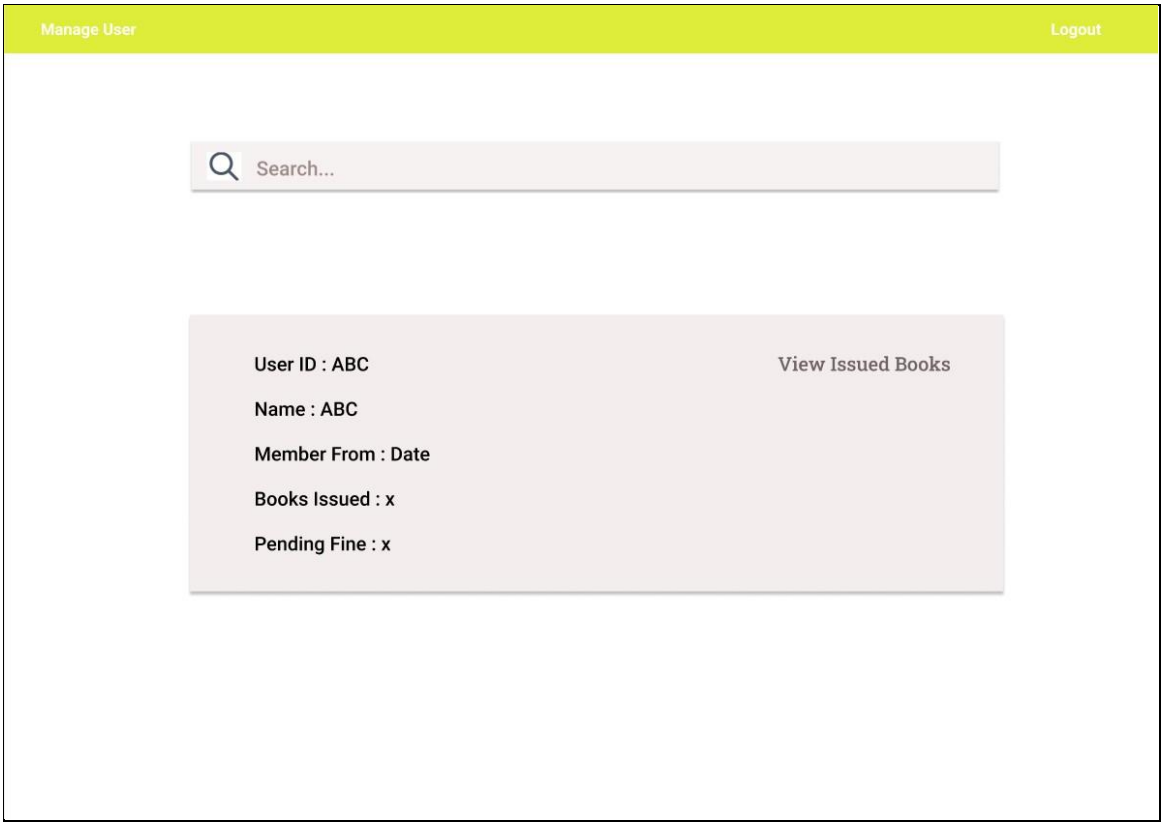
6.3.3d Approve Reservations - When a student reserves a book, it is approved by the librarian by entering the student id and bookid to first check if the data is correct. Once the check is performed, the approve button appears, allowing the librarian to approve the book issue.



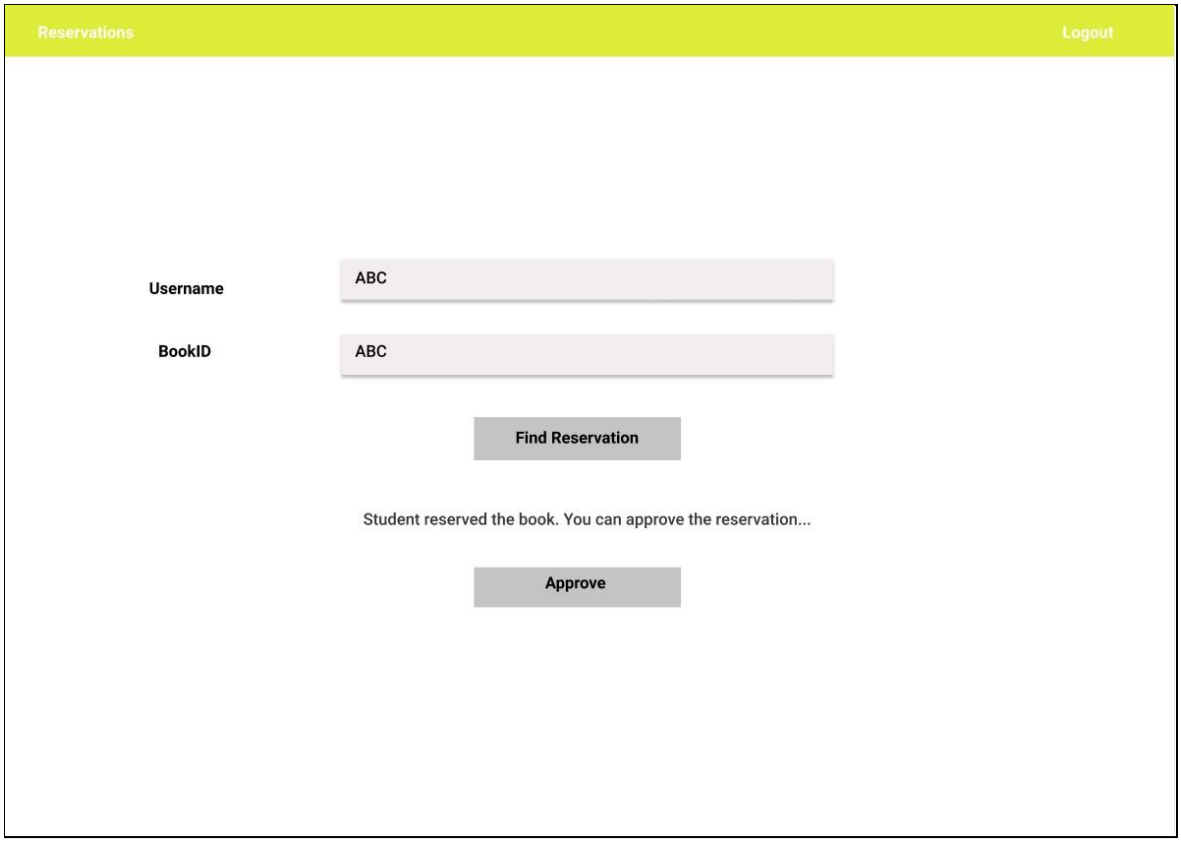
Add Books



Manage Books



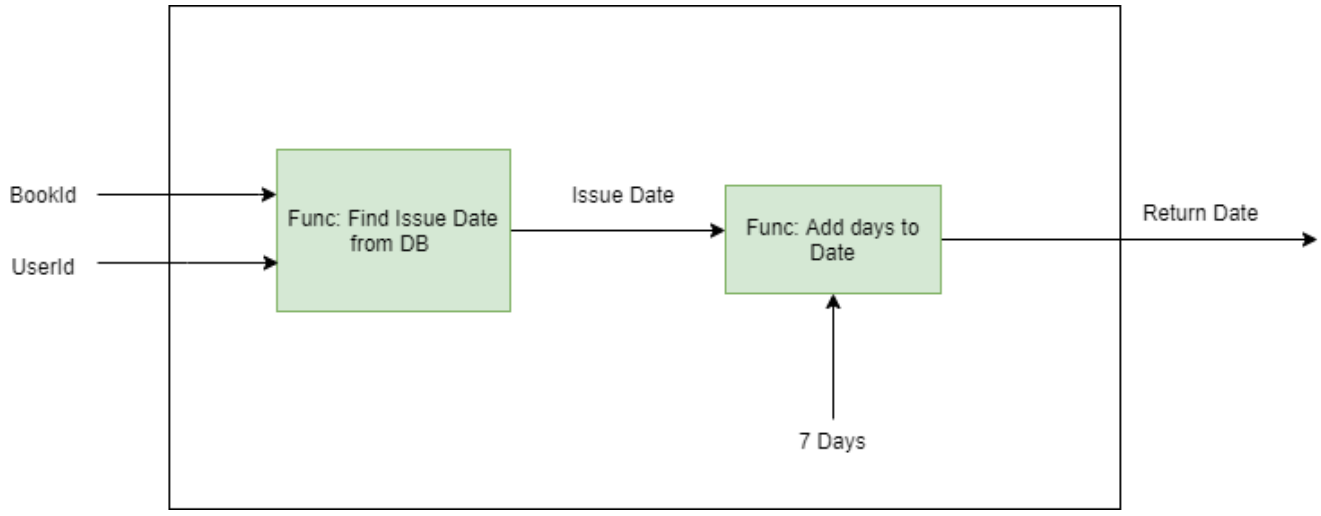
Manage User



Approve Reservations

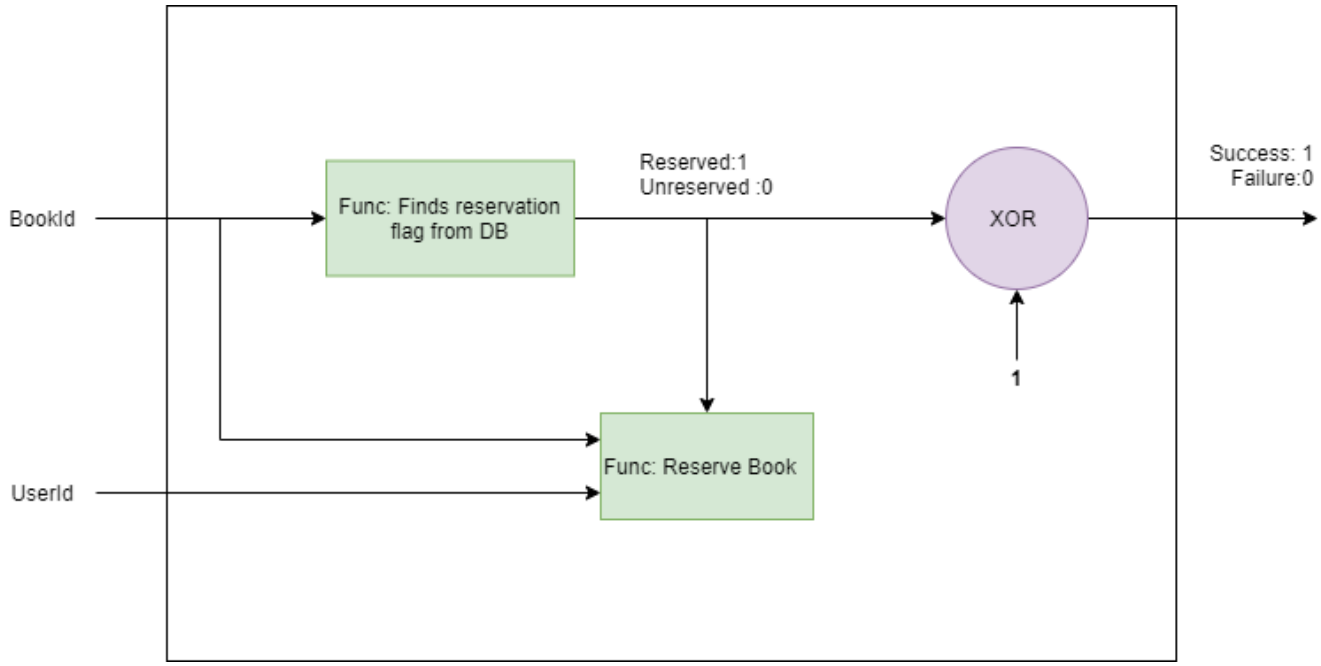
7. Functional Design

7.1 Finding the return date after a book has been issued.



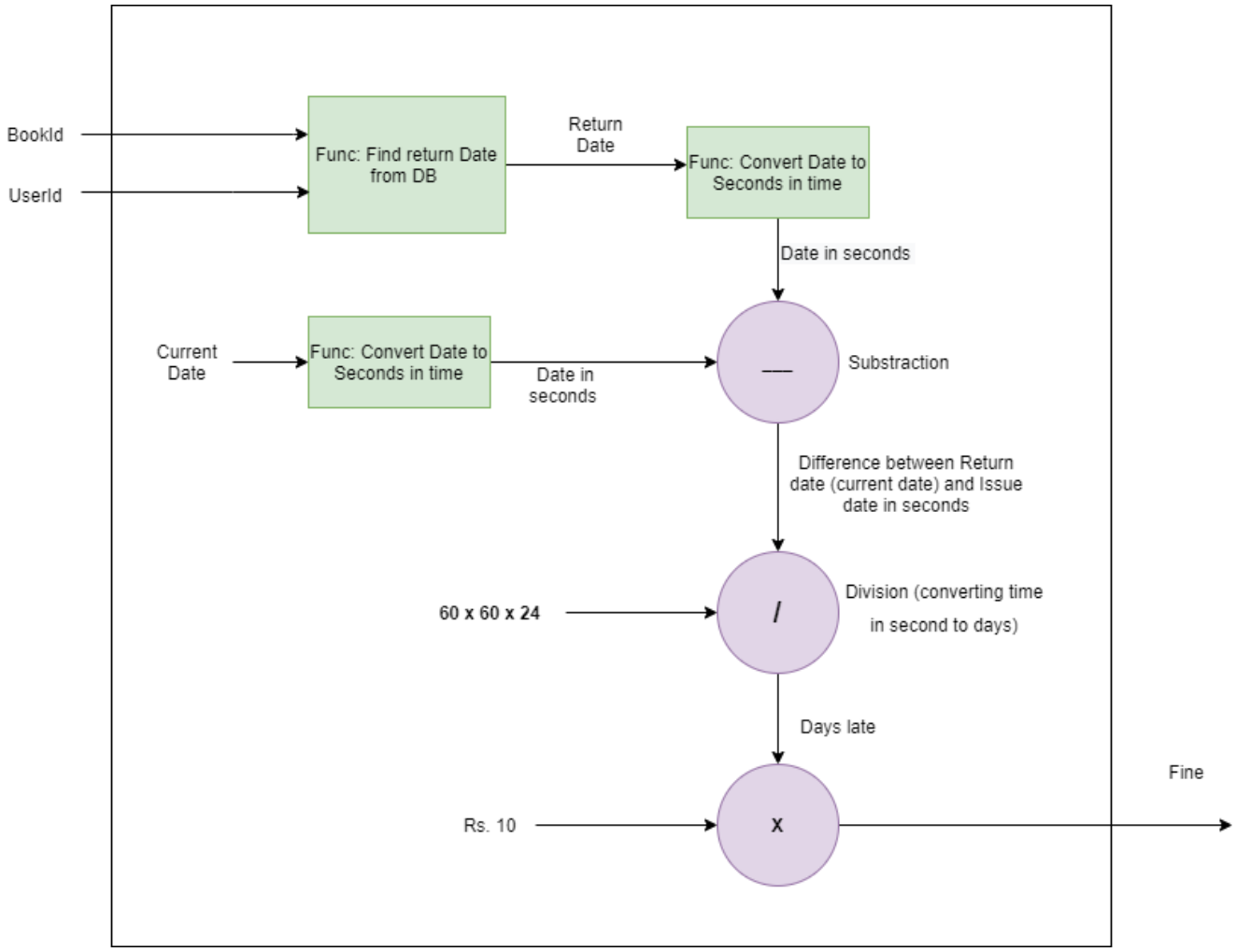
There are two high level functions in the above block diagram. The first function can return the issue date associated with a bookId and userId (which are the input arguments). The second function adds x number of days to the input date, and returns the final date.

7.2 Reserving a book by the student



The function takes in information about the user and book. First it checks whether the book has been reserved or not, and returns the reserved flag. If the book has been reserved, then it sends a failure signal (failure : 0) as an output. Otherwise, in the high level function: reserve book, it updates the database about the status of the book (makes it reserved) , and associates the userid with book id. This function updates the database and has no output. Overall, the caller gets a success signal as output (success :1).

7.3 Calculation of fine

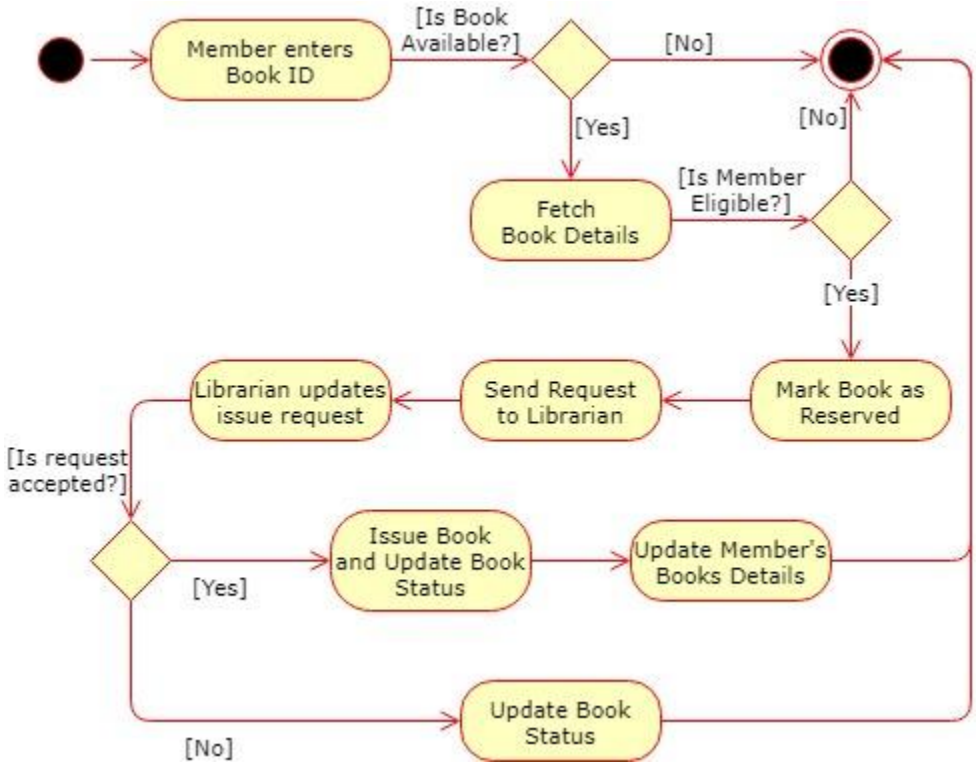


This takes in bookID, userID and returns the fine associated with the book.

8. Activity Diagram :

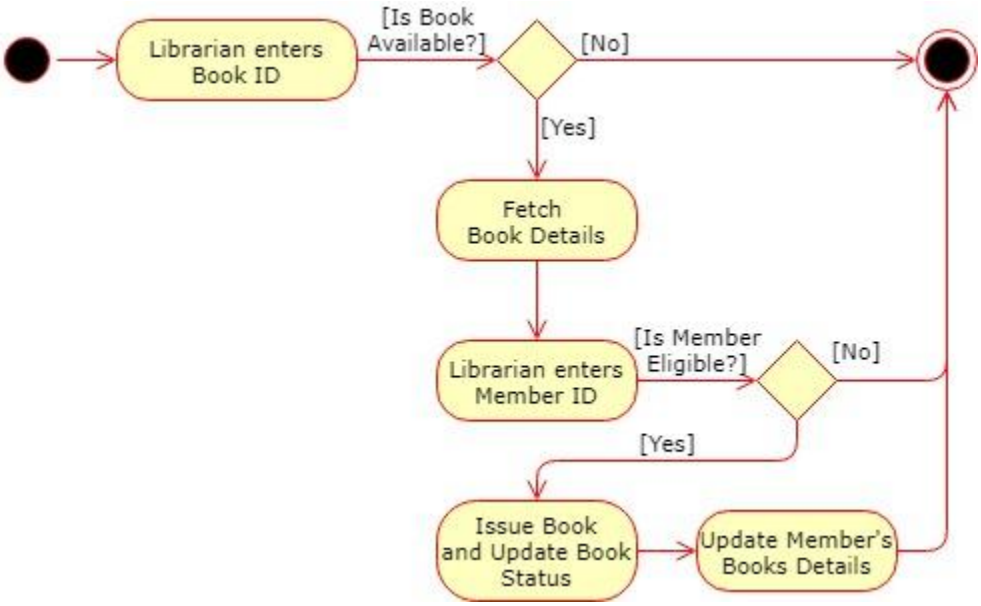
8.1 Activity Diagram to Reserve/Issue Book by member :

The Member will first need to reserve the Book he/she wants to issue. The Book will be marked reserved and the Librarian will receive Issue Request. If the Librarian approves it, the details will be updated accordingly, otherwise the request will be rejected and the book will be unreserved.

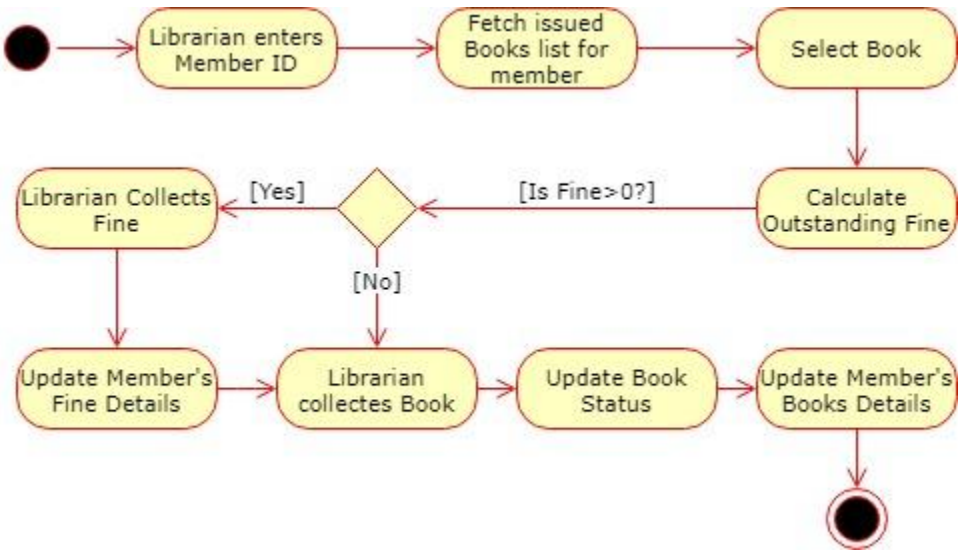


8.2 Activity Diagram to Directly issue Book by librarian:

The Librarian can directly issue an available book on premises to a member without the need for the member to reserve the book beforehand using the member's ID for transaction record.



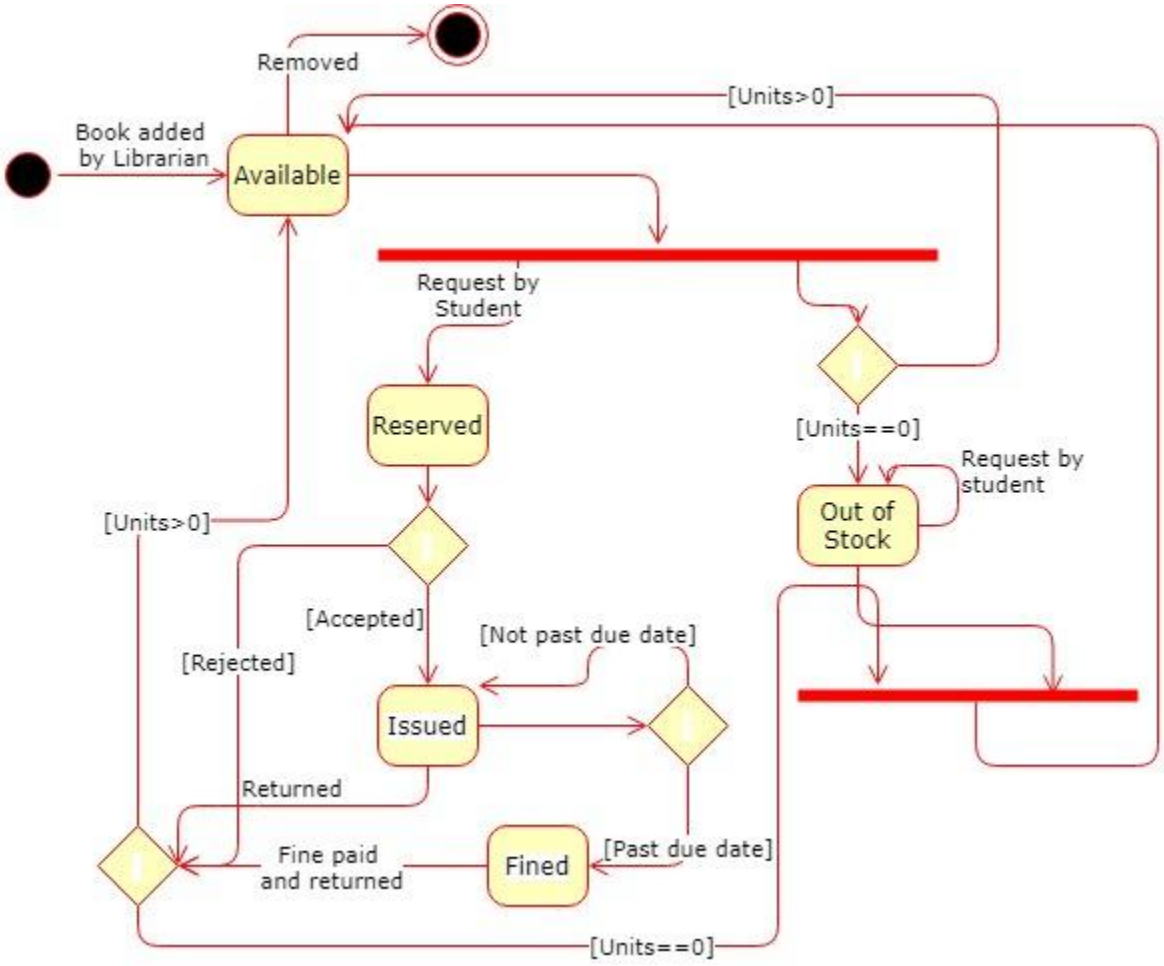
8.3 Activity Diagram to return book:



9. State Diagram:

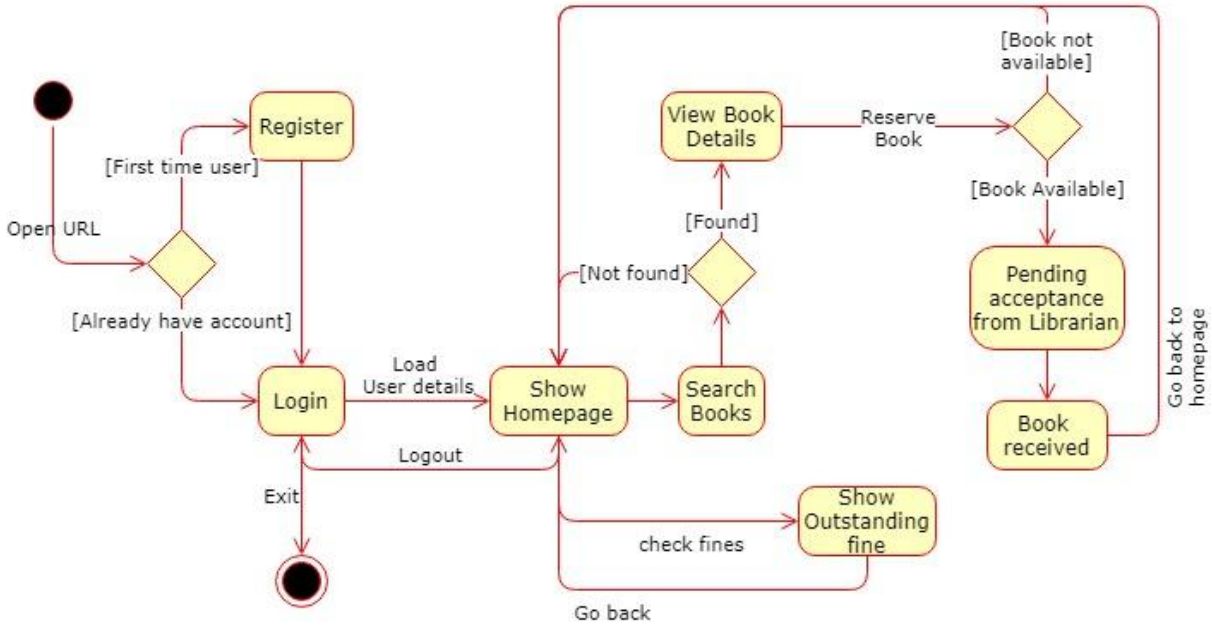
9.1 Book State Diagram :

Outlines the states the Book object can be in, and shows the transition elements between these states.



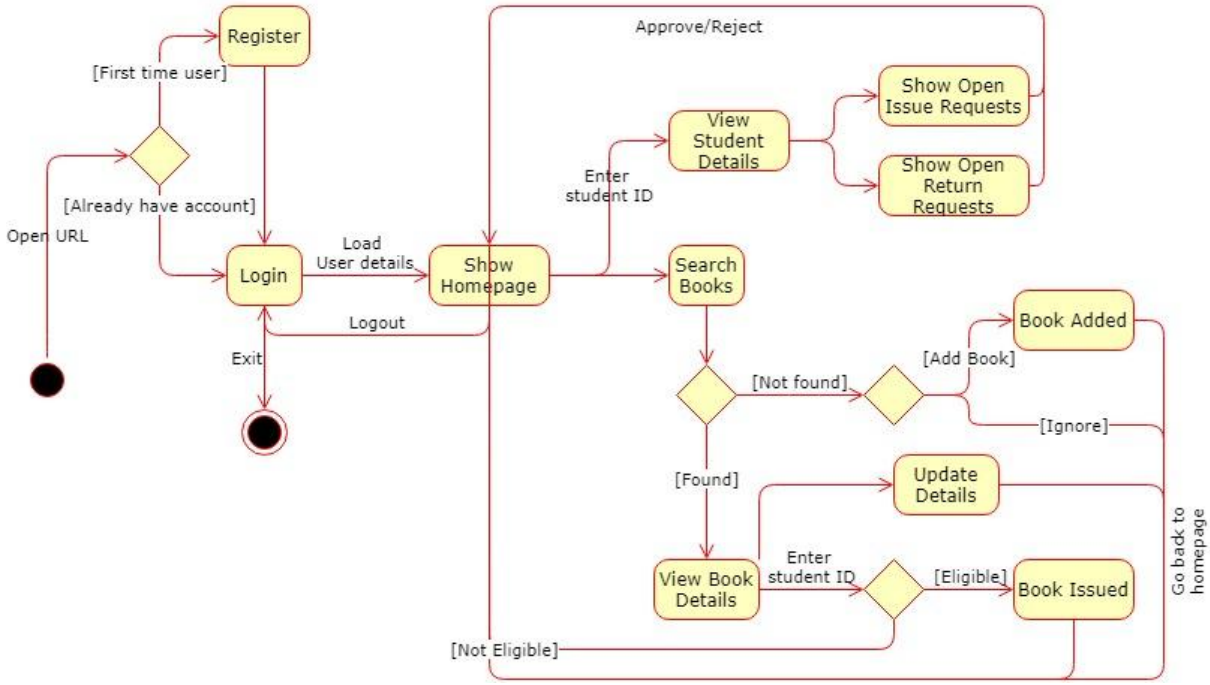
9.2 State Diagram for Student/Member users:

Shows the various states for the members while using the interface to search, issue or return books.



9.3 State Diagram for Librarian :

Shows the various states for the librarian while using the interface for facilitating library functionalities.



10. Deployment Architecture Diagram :

The Architecture will have a Spring boot server to handle backend functionalities. The front-end server will implement React to render web pages that will be delivered to the User's Web Browser. MySQL will serve as the RDBMS for this system.