Software Requirements Specification

for

Online Book Recommend and Reading System

Version 1.0 approved

Prepared by Siri Chandana Priya

18CS10008

IIT Kharagpur

25-01-2020

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The main purpose of this book recommendation system is to recommend books to the buyer that suits their interest. This recommendation works offline and stores recommendation in the user's web profile.

1.2 Document Conventions

This SRS document uses a few different font sizes for a clear distinction. Important topics, headers, and words are highlighted so as to imply their significance. Higher-level requirements are given higher priority which is clearly visible by the detailed description of the requirement.
Font-style-size used for

- Headings are Times-Bold-18
- Sub-headings are Times-Bold-16
- Text and other's are Times-Regular-14

1.3 Intended Audience and Reading Suggestions

The document is intended for the developers, project managers, marketing staff, users and testers, and documentation writers.
Brief Summary of the whole SRS document can be summarized within the following sections
(Users are requested to follow this sequence for better understanding of the document).
Section 1: Introduction which covers the purpose, conventions of the document and a brief view of the project scope and references.
Section 2: This section is about the overall product description which includes product perspective, functions, operating environment design and implementation constraints.
Section 3: External Interface Requirements giving a brief introduction to the user, hardware, software and communications interfaces.
Section 4: Brief about the features of the system.
Section 5: Provides a list of Non-functional requirements.
Section 6: Other requirements.

1.4 Product Scope

The Software Requirements Specification captures all the requirements in a single document. The Online book Recommendation system that is to be developed provides the users of the system with the information of the books. The user has to register in order to use the facilities that will be provided by the system. Buyers profile records will be maintained. We find out the category of books from users buying history and give suggestions further. And feedback will also be taken from time to time.

1.5 References

SRS IEEE TEMPLATE is used to construct this Document.

2. Overall Description

2.1 Product Perspective

The goal of this recommendation system is to predict the buyer's interest and recommends the books accordingly. This book recommendation has considered many parameters like the content of the book and the quality of the book by doing collaborative filtering of ratings by the other buyers. This recommender system also uses associative model to give stronger recommendations. This system does not have a performance problem since it built the recommendations offline.

2.2 Product Functions

The following are the methods initiated by the user:

- Registration/Login
- View Book Gallery
- Add to cart/Buy
- Online Payment
- Orders

The following are the methods initiated by the Admin:

- Login
- Add Books
- View Books
- Remove Books
- View Users

2.3 User Classes and Characteristics

This System has two User classes, User and Admin.

2.4 Operating Environment

- Hardware Requirement :
  - i3 Processor-Based Computer or Higher
  - Memory - 1GB RAM
  - Hard Drive: 50GB
  - Monitor
  - Internet Connection
- Software Requirement
  - Windows 7 or Higher
  - Visual Studio
  - SQL Server, Java, python

2.5 Design and Implementation Constraints

In this system, collaborative filtering will be used. In its original and simplest implementation, this approach recommends to the active user, items that other users with similar tastes liked in the past. The similarity in the taste of two users is calculated based on the similarity in the rating history of the users. User-based methods rely on the opinion of like-minded users to predict a rating and generate ratings.

2.6 User Documentation

The user manual will be provided in online help support. This project is intended for one type of user. Without proof of identity, user can’t register for this.

2.7 Assumptions and Dependencies

It is assumed that the user will have the basic resources required for this software, this includes:

- Active Internet Connection
- A browser to access the internet
- A desktop system or a cell phone to work on

3. External Interface Requirements

3.1 User Interfaces

- The design or layout of every form will be very clear and very interactive to the user.
- In the login window, the user can easily enter the desired password and login name.
- Then it will give the successful login message.
- There will be a proper collection of GUI(graphical user interface).
- The user will be able to search the required book from the collection.
- All the profile settings options will be available to the user.
- This will provide better security data because the menu window will be displaying according to the login.
- Users can easily save their data into the database and keep track of the records of purchase etc...
- This software will be easily understandable and operable by the user.
- Admin can log-in with a valid user name and password into the system.
- Admin has the authority to add new books with basic details.
- Admin can see all the available book details.

3.2 Hardware Interfaces

There are no special hardware interface requirements.

3.3 Software Interfaces

The Graphical user interface will be constructed upon HTML5, CSS, and JavaScript. The collaborative filtering and backend will be done using python. Further details provided in 2.7 section of this document.

3.4 Communications Interfaces

The communication interfaces include e-mail and web browsers. E-mail is required for carrying out the necessary communications with the user and the web browser is required to send the notifications to the user.

4. System Features

4.1 Registration / Login:

4.1.1 Description

The user needs to create an account by signing up with his e-mail account. This is the first step for any user as the user needs to have an account to read or purchase any book.

4.1.2 Functional Requirements

- Ask for User details
- Allocate Space in the server memory
- A function to check if the user name is unique
- A function to verify the email id of the user
- Ask the user to create a password
- A function to personalize users profile
- A function to reset the password-Forgot password

4.1.3 Input and Output

Input: User Details to register and particular ID and password for login.
Output: Successful login or Incorrect Login

4.2 View Book Gallery and Edit Profile

4.2.1 Description

After successful login users can see all the available books to buy online. The book gallery contains the front image of the book, price, name, author, etc. Clicking on a particular book's name will show a brief overview of that book. Users can change his/her profile info like username password etc...whenever he/she wishes to.

4.2.2 Functional Requirements

- User can view all the available books
- User can view only the interested books of a particular genre
- Users can view the overview of the book by clicking on the book.
- A function to edit the user profile
- User has to save the edited profile info
- In case of resetting the password, the system will ask to re-enter the password.

4.2.3 Input and Output

Input: Search book or Advanced Book Search and Edit profile option.
Output: image of the book searched and profile updated.

4.3 Recommendations by seeing User Ratings

4.3.1 Description

Take the user ratings of the book while reading i.e how satisfied the reader is with the book. When a user comes to the app again the he can view the ratings of the books and proceed.

4.3.2 Functional Requirements

- collects the information of ratings/feedback
- Recommends to the user by showing the ratings.

4.3.3 Input and Output

Input: ask customer for rating a particular book
Output: Thanks for rating and ratings will be displayed in the gallery.

4.4 Add to cart/Buy now

4.4.1 Description

Users can buy books with two options like add to cart and buy now. Within cart option users can add or remove items and after confirmation payment procedures start.

4.4.2 Functional Requirements

- Take the input of the book
- Display the cost and ask for confirmation

4.4.3 Input and Output

Input: Adding wished book to cart in the profile
Output: Book Added confirm checkout further

4.5 Online Payment

4.5.1 Description

Users can buy the book by doing online payment with different payment options.

4.5.2 Functional Requirements

- Asks for payment option

- Performs payment
- Adds purchased book to the user's gallery

4.5.3 Input and Output

Input: choose Payment methods to purchase

Output: Payment Successful and book will be displayed in the user's gallery

4.6 My orders

4.6.1 Description

Gives an overview of the existing user's gallery and books purchased.

4.6.2 Functional Requirements

- A function that piles the overall user's gallery.

4.6.3 Input and Output

Input: click the profile and my orders

Output: Overview of the books in the gallery and orders.

4.7 Admins Login

4.7.1 Description

Admin can login with a valid username and password into the system.

4.7.2 Functional Requirements

- Similar to the Functional Requirements of the user login Refer to Section **4.1.**

4.7.3 Input and Output

Input: Admin details to login into the system

Output: Successful login or Invalid login

4.8 Add Books

4.8.1 Description

Admin has the authority to add new books to sell online with image, price and basic details.

4.8.2 Functional Requirements

- Function to add the new book to existing booklist
- Giving access of the book to the user's.

4.8.3 Input and Output

Input: Admin can view the book set
Output: Updated book set will be visible

4.9 View Books and Users

4.9.1 Description

Admin can see all the available books with its details like author, name, price. Admin can view all the registered users with their details.

4.9.2 Functional Requirements

- Function to give access of all the book details to the Admin.
- Admin can view the details of the books, authors …
- Function that makes admin view all the registered users and their details like email id for future reference.

4.9.3 Input and Output

Input: Admin views all the users ,books and order database
Output: Updated list of books in the database.

4.10 Exchange of Book

4.10.1 Description

When the user is done reading with the book he can exchange the book with the books with similar or less cost and the balance amount will be displayed as points in the user's profile.

4.10.2 Functional Requirements

- Function to take check the user's choice of required book with the cost of the returned book.
- Function to replace the previous book with a new book and update the points balance in the user's profile.

4.10.3 Input and Output

Input: Book the User wants to Exchange
Output: The exchanged new book in the user's gallery.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The software must satisfy the following constraints

- The User must be able to access only his\her routine.
- Only the person who has an account can access the book library.

5.2 Safety Requirements

There is a Feedback form available where users can report any bugs they have encountered so that the developers can fix it in the next release. There is no risk of any threat that can be issued by the system. Since it is a web-based application, there will definitely be chances of internet-based threats. These need to be minimized as much as possible.

5.3 Security Requirements

The system is built with login credentials so that any authorized user can access the system. All the personal data stored for a particular user will be encrypted by using the standard encryption algorithm. This needs to be protected from any possible data theft.

5.4 Software Quality Attributes

The Software must work with at least a Chrome and Firefox browser.Users must have a fast internet connection. The Software must operate on both windows and ubuntu Operating systems.

5.5 Business Rules

No business rules are required for this Software.

6. Other Requirements

6.1 Data and Category Requirement

There are different categories of users namely users and admins. Depending upon the category of the user the access rights are decided. It means if the user is an administrator then he can be able to modify the data, add books, etc...

No other Specific Requirements.

Appendix A: Glossary

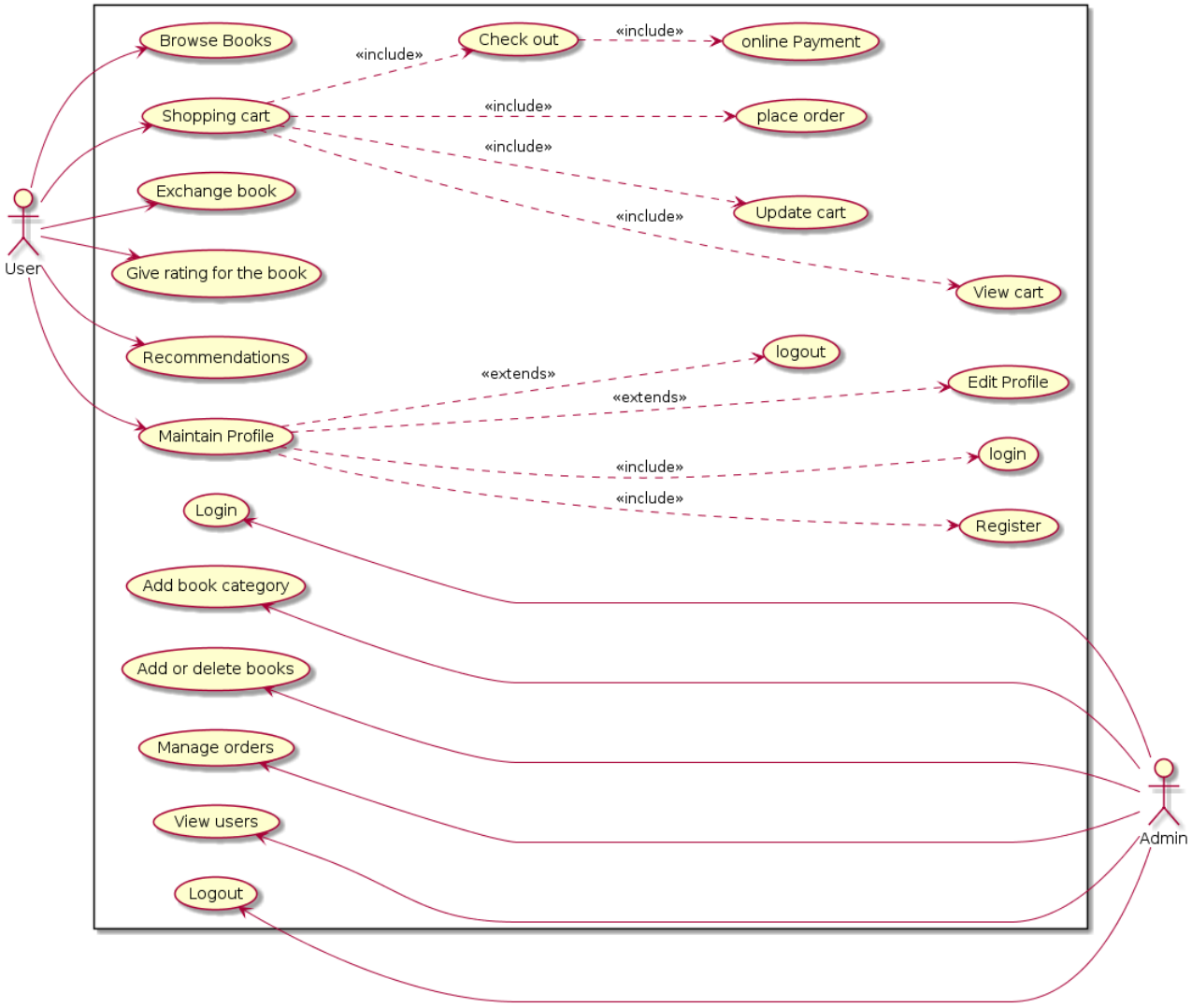
- Administrator/Admin: A login id representing a user with user administration privileges to the software.
- User: A general login id assigned to most users.
- SQL: Structured query language used to retrieve information from a database.
- SQL Server: A server used to store data in an organized format.
- Interface: Something used to communicate across different mediums.

Appendix B: Analysis Models

Appendix C: To Be Determined List

7. System Design and Documentation (SDD)

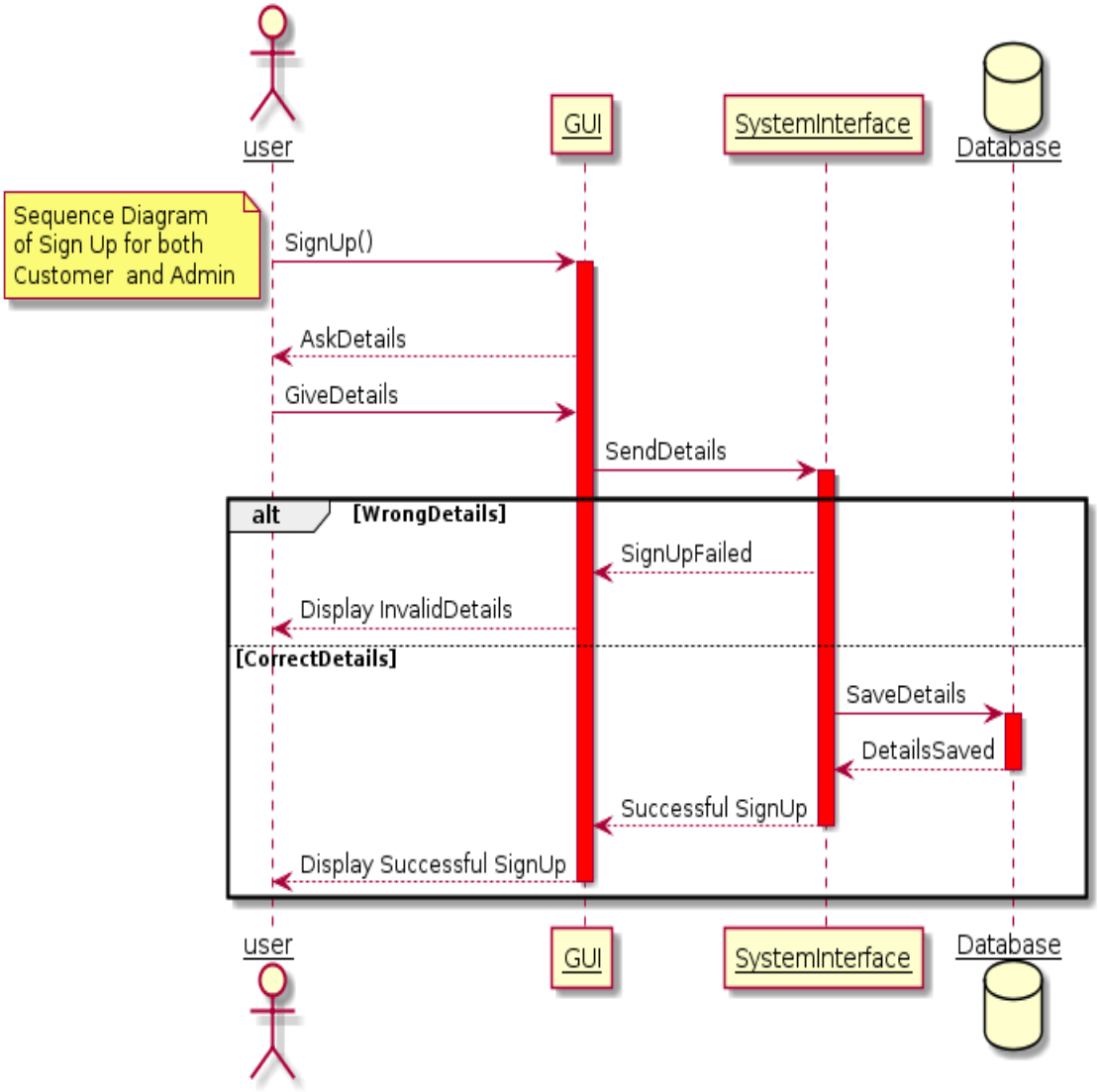
- **Use Case Diagram:** User and admin are the two users included in this software. Below diagram shows the use case representation.



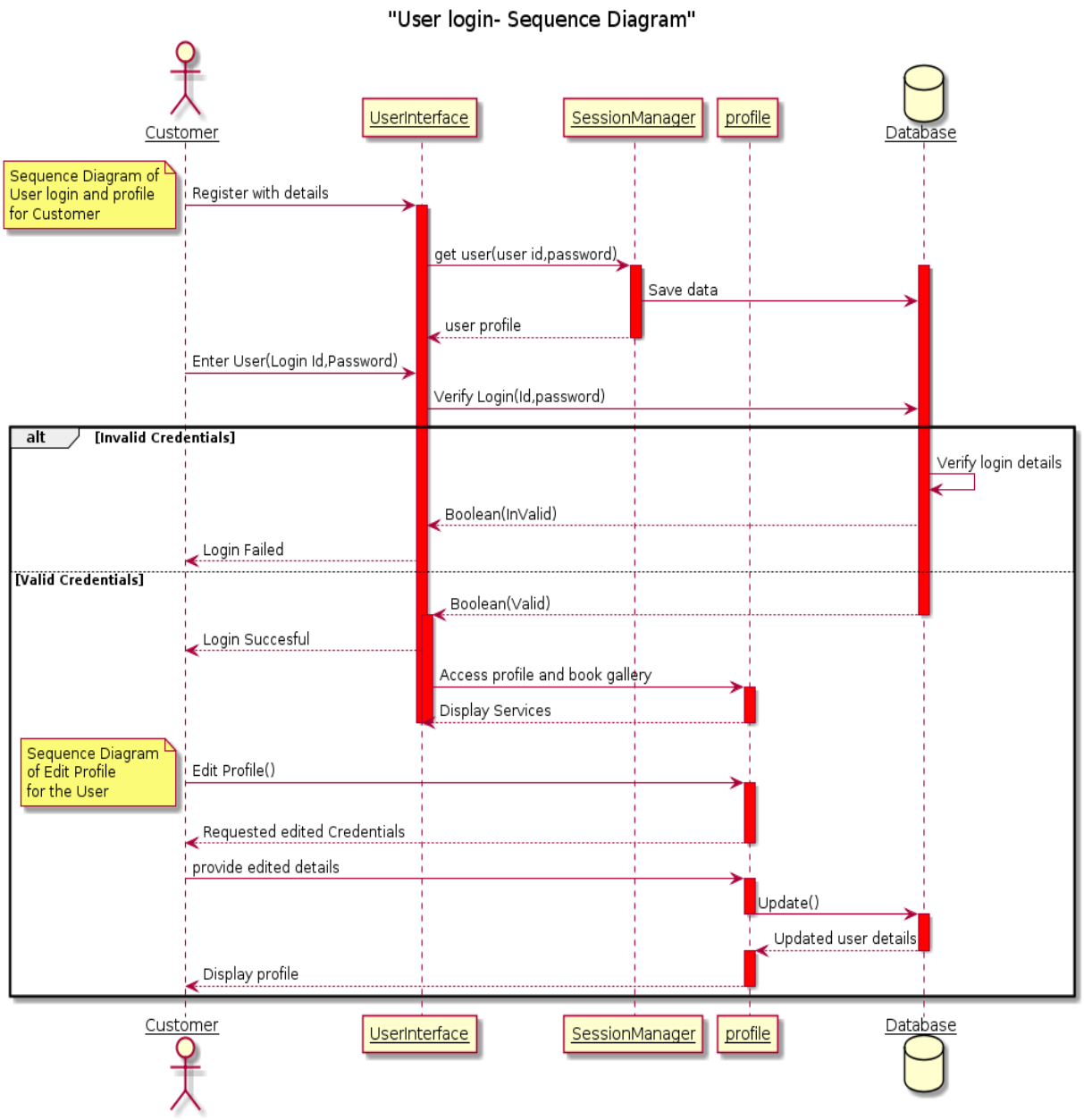
The Use Cases Used describes the particular functionality described in the SRS document (System features context part).Each use case serves its purpose input and output respectively.

- **Sequence Diagrams:**

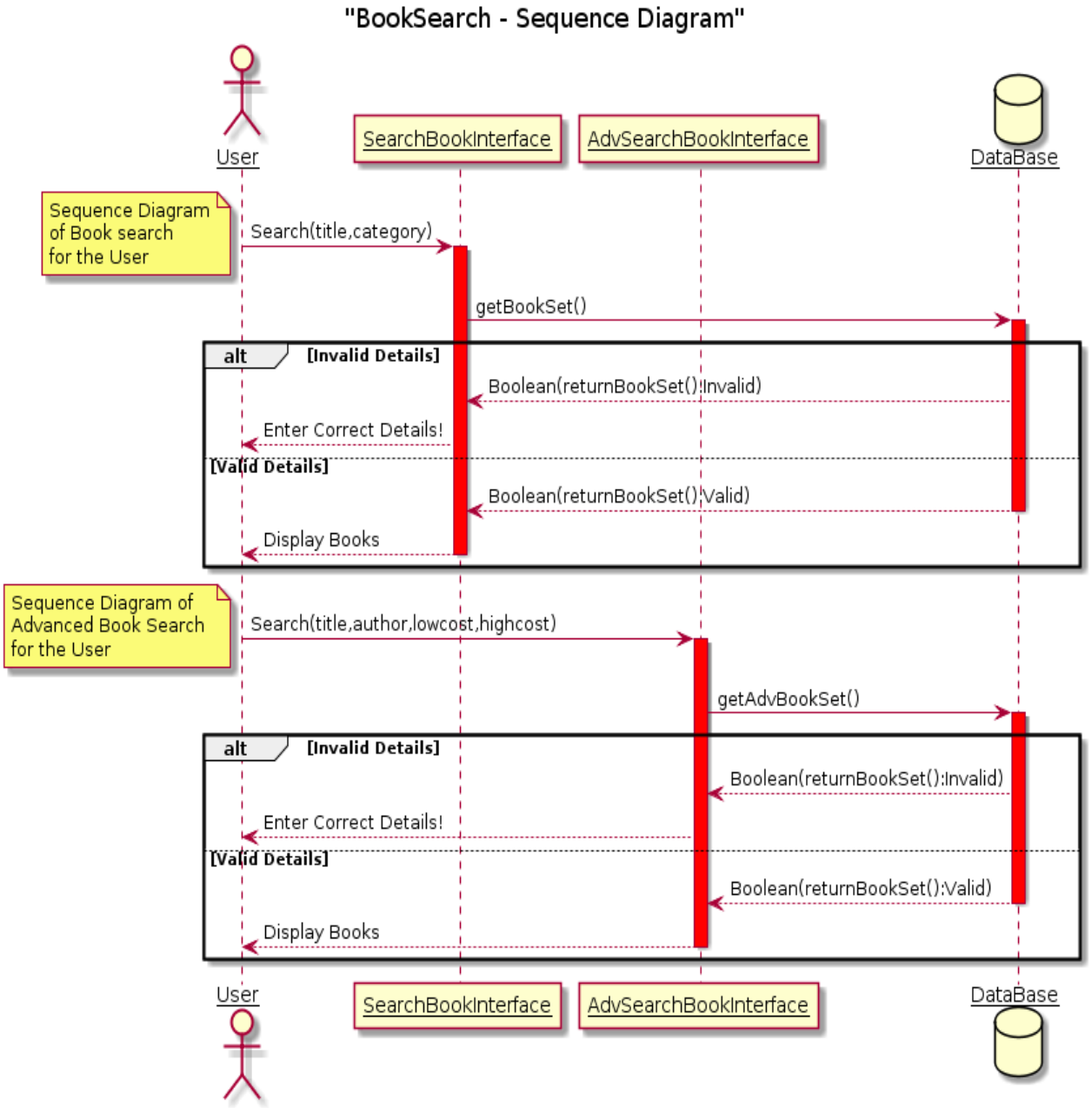
1. **Registration or sign Up:**



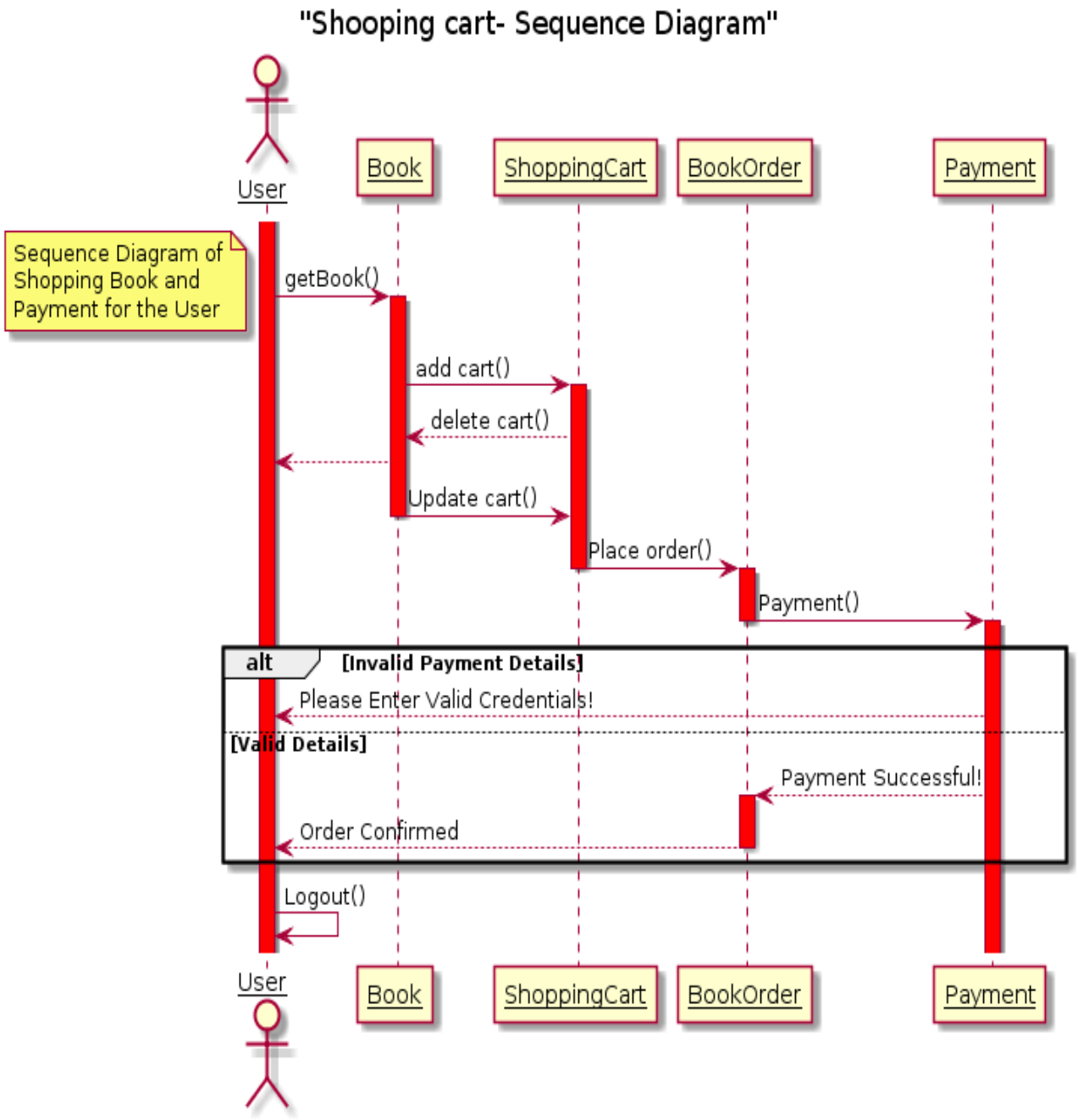
2. User Login , Edit Profile and View Gallery:



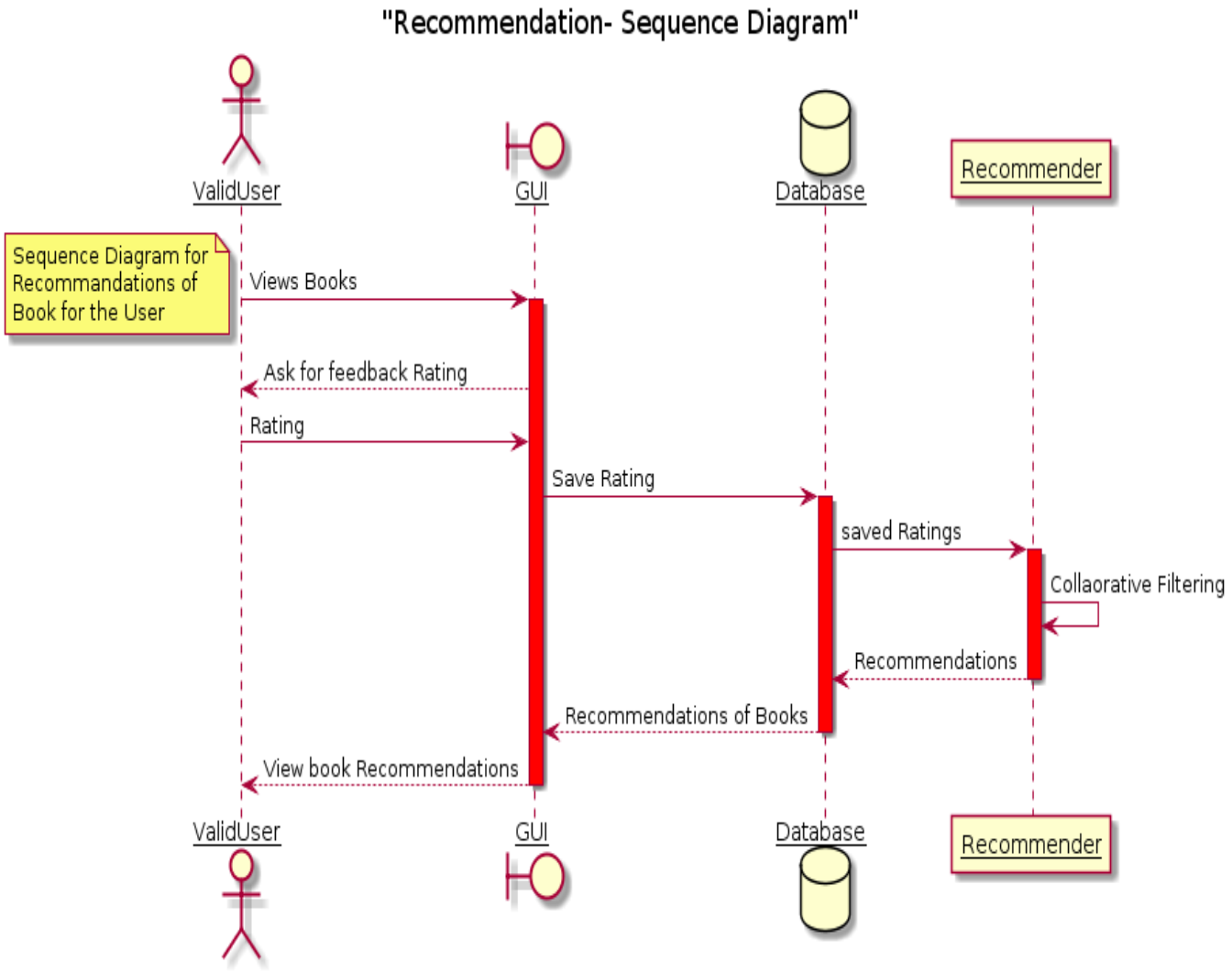
3. Search Book and Advanced Book Search:



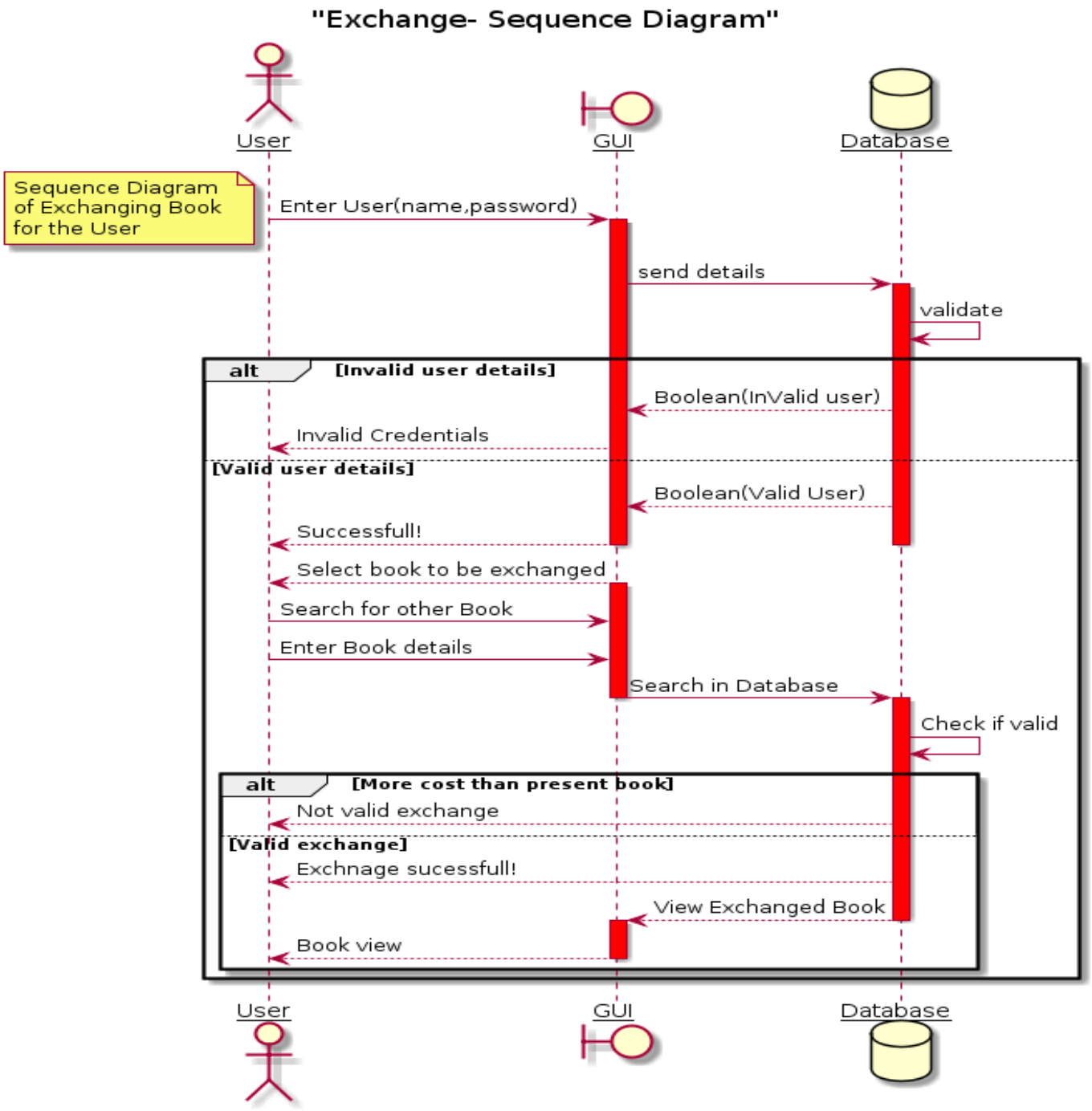
4. Shopping cart :



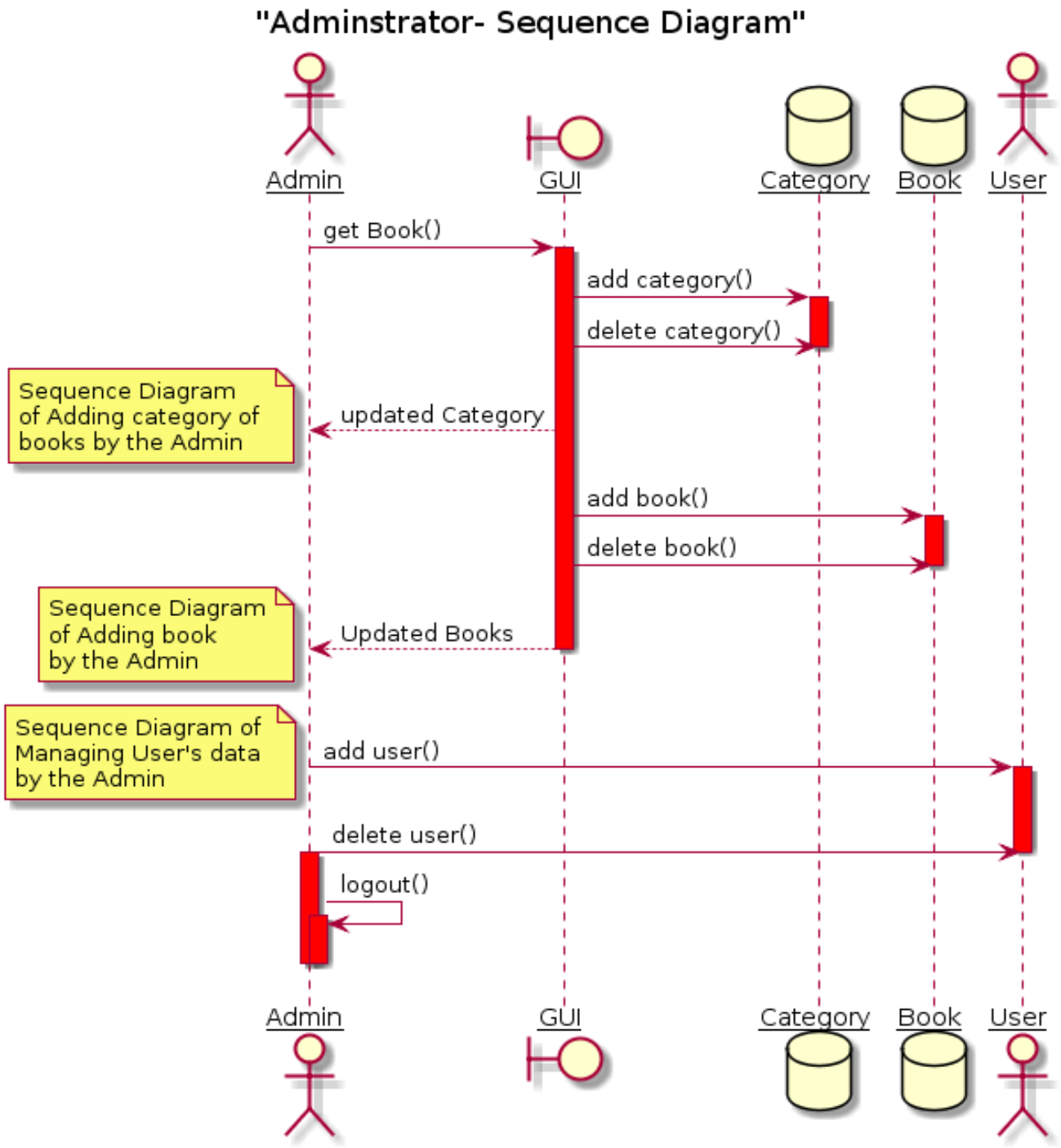
5. Recommendation:



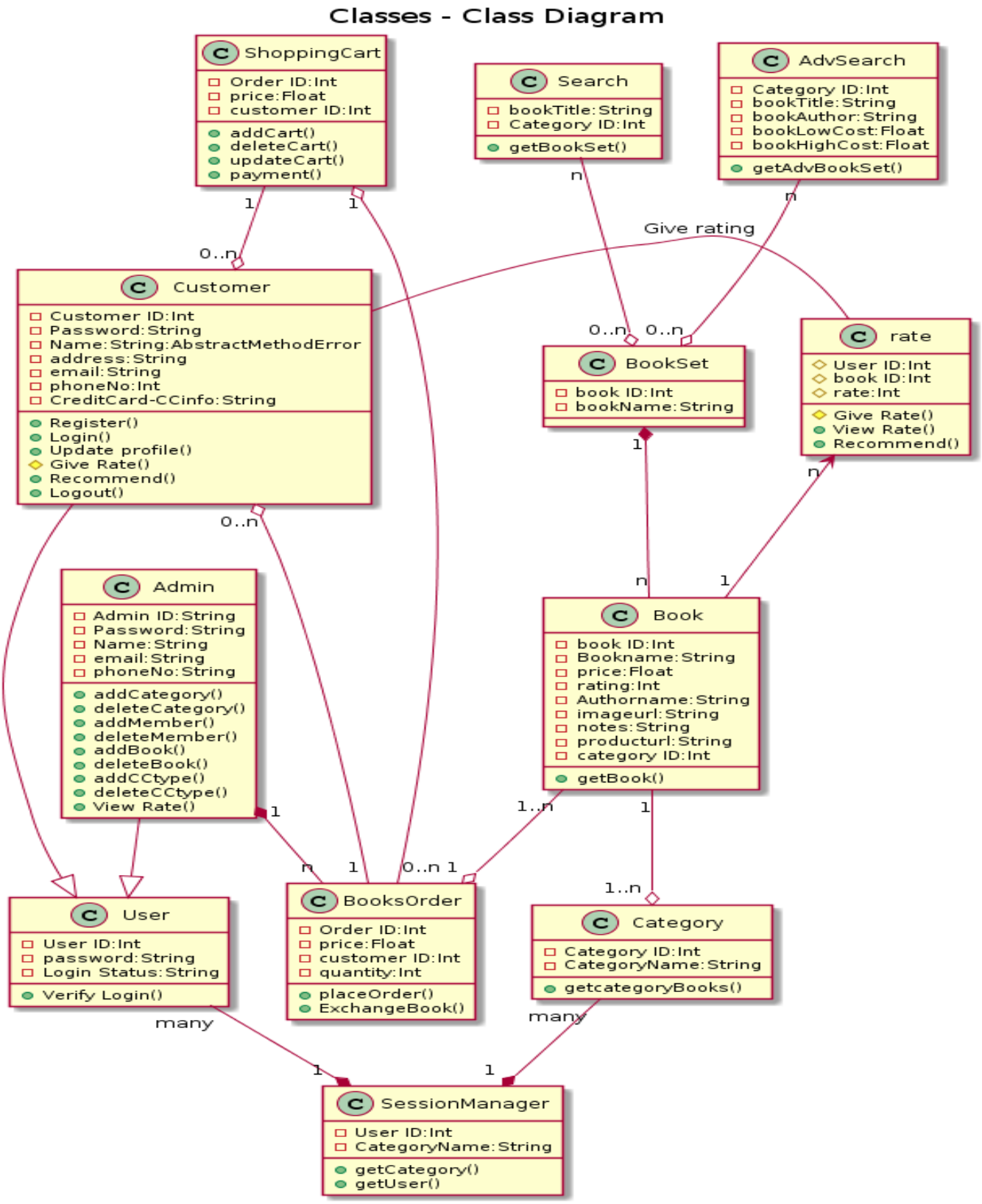
6. Exchange of Book by the User:



7. Administrator Adding Book category, Book and User's data:



- **Class Diagram:**



Specification:

1. Class Category
   Attributes
   CategoryID: integer
   CategoryName:string

   Operations
   getCategoryBooks(bookID:int)

2. Class Book
   Attributes
   bookID:integer
   categoryID:integer
   bookName:string
   authorName:string
   price:float
   imageurl:string
   producturl:string
   rating:int

   Operations
   getBook():Boolean

3. Class shoppingCart
   Attributes
   orderID:integer
   customerID:integer
   price:float

   Operations
   addCart():Boolean
   deeleteCart():Boolean
   UpdateCart():Boolean
   Payment(CCtype)

4. Class BookOrder
   Attributes
   orderID:integer
   customerID:integer
   price:float
   quantity:integer
   operations
   placeOrder(BO:BookOrder):Boolean

5. Class Search
   Attributes
   bookTitle:string
   categoryID:integer

   Operations
   getBookset():Boolean

6. Class AdvSearch
   Attributes
   bookTitle:String
   CategoryID:integer
   bookAuthor:String
   bookLowCost:float
   bookHighCost:float

   Operations
   getBooksetbyAdv():Boolean

7. Class BookSet
   Attributes
   bookID:Int
   bookName:String

8. Class User
   Attributes
   UserID:string
   Password:string
   LoginStatus:string

   Operations
   Verifylogin():Boolean

9. Class Adminstrator < User
   Attributes
   AdminID:string
   Password:string
   Name:string
   Email:string
   phoneNo:integer

   Operations
   addcategory():Boolean
   deletecategory():Boolean
   addMemeber():Boolean

deleteMember():Boolean
addBook():Boolean
deleteBook():Boolean
addCCtype():Boolean
deleteCCtype():Boolean
viewrate(rating:int)

10. Class Customer < user
    Attributes
    customerID:string
    password:string
    Name:string
    Address:string
    Email:string
    phoneNo:integer
    CCinfo:string

    Operations
    Register():Boolean
    Login():Boolean
    updateProfile(customerID:string,Name:string,address:string,email:string,phoneNo:integer,CCinfo:string):Boolean

11. Class rate
    UserID:integer
    BookID:integer
    Rate:integer

    Operations
    GiveRate(rating:integer):Boolean
    ViewRate():Boolean

--Associations
--Each book should belong to only one category.
Associate bookCategory between
Category[1] role subcategory
Book[1..*] role allBook
Justification: As each book should belong to particular category. It is a Aggregation.
--Each BooksOrder should contain atleast one Book
Association BooksOrderHasBook between
BooksOrder[1] role the BookOrder
Book[1..*] role the Book
Justification: As to order a book i.e in BooksOrder atleast one Book should be present.It is a Composition.

--Each BooksOrder should belong to exactly one Customer
Association CustomerHasOrder between
Customer[1] role belongstocustomer
BooksOrder[0..*] role thecustomerbook
Justification: Inorder to order the Book Customer should have an account and register and then in his/her profile the customer can order books. It is a Aggregation.

--Each Shoppingcart should belong to only one Customer
Association CustomerrelatedtoShoppingCart between
Customer[1] role thecustomer
ShoppingCart[0..*] role thecart
Justification: Valid Customer's profile contains his/her shopping cart i.e every customer has his/her own shopping cart. It is an Aggregation.

--Each Shoppingcart should have atleast one BooksOrder
Association ShoppingCartHasOrder between
ShoppingCart[1] role thecustomer
ShoppingCart[0..*] role thecart
Justification: In order to checkout Shoppingcart Customer's BooksOrder should contain atleast one Book. It is a Aggregation.

--Each search should result some bookset
Association searchHassomebookset between
Search[1] role thesearch
BookSet[0..*] role thesearchset
Justification: Search results some set of books. It is an Aggregation.

--Each Advsearch should result some bookset
Association AdvsearchHassomebookset between
Advsearch[1] role the Advsearch
BookSet[0..*] role the Advsearchset
Justification: AdvSearch also results some set of books with related search details. It is an Aggregation.

--Every User is connected to single session manager
Association session managerHasUser between
User[many] role the User
Session manager[1] role the session manager
Justification: Every User including Customer and Admin are connected to Session manager. It is an Composition.

--Every category is connected to session manager
Associate session managerHasCategory between
Category[many] role the Category
Session manager[1] role the session manager

Justification: Every User including Customer and Admin are connected to Session manager. It is an Composition.

--Each BooksOrder is known to Admin
Association BooksOrder between
BooksOrder[n] role the BooksOrder
Admin[1] role the Admin
Justification: Admin has access to all BooksOrder. It is a Composition.

--Each book results some set of ratings by user
Association ratings between
Rate[n] role the rate
Book[1] role the Book
Justification: Customer gives some set of ratings for the books. It is an Extension.

--Each Book should belong to some Bookset
Association Bookset between
Bookset[1] role the Bookset
Book[n] role the Book
Justification: Each book belongs to some specific BookSet. It is a Composition .

--Constraints
--Each user should have different userID
--Each Book should belong to exactly one Category
--Each Book should have a different bookID
--The OrderID for each Order must be different
--Each BookOrder should have some books
--Quantity should always be a positive value
--Each BookOrder belongs to exactly one customer
--Quantity should always be a positive value
--Each Shoppingcart belongs to only one Customer
--In search Low price must be less than High price
--Price should always be positive value