The building of the web software will take place in 4 parts:

1) Requirement engineerin:

Will take all required information from the online book reselling about how they want the website to function, look and what kind of features it will need.

2) System design

Will design the system UML diagrams using software's like magicdraw to give a prototype and diagrams to show the relationship between different classes of how this website will work and communicate with users

3) Implementation

Will design the website using the above prototypes and diagrams.

4) Testing:

Will test the implemented website in different and check if there are errors in the and will be fixed before releasing the the online book reselling.

PART 1: Overview & Software Requirements Specification

Introduction:

A) Purpose:

To develop a virtual bookstore that facilitate online functions like browse books, buying books, uploading books, through some modules suitable for a user's role like visitor, customer, administrator.

B) Project scope :

User:

- Enable user to register with basic registration details
- Enable user to login into the system in order to access the system.
- Once user is logged into the, he/she may view all the added books with their details.
- User can surf various book of their choice and buy them.
- System allows user to sell their books online buy added the book name and its details.
- Enable user to upload a book if he/she wants to sell.
- All the purchase history of user will be displayed with details.
- If any buyer is interested in buying a book or anyone buys a book from the user then buyer's details will be displayed.

Admin:

- admin need to login into the system in order to access the system.
- Admin can view all the added books online with their details
- System allows admin to view all the transaction details of buying and selling a book.
- All the registered user details will be displayed to the admin

C) Glossary and Abbreviations :

- Used UML notations

D) List of the System Stakeholders :

- Administrators
- Owers

2) Software Requirement Pattern:

3) Functional USER Requirements:

Register: Allows Account Registration

1- The system shall allow a non-registered user to create a new account.

2- The system shall require the following information from the user:

Name, password, re-enter password, E-mail.

3 The system shall ask the user for a username and password.

4 The system shall confirm the username and password are acceptable (username not taken).

5 The system shall store the information in the database.

Login: Allows Account Login

1. The system shall allow a registered user to log-in to their account.
2. The system shall require a username and password from the user.
3. The system will verify the username and password, and the user will be considered logged-in.

Search: Allows user to search

1. The system shall allow a user to search for books by title, author, subject, course number, professor, or ISBN number.
2. The search results will include a picture of the front cover, along with the title, author, price, availability, and condition of the book.

View browse: Enable user to view books

1 When user logged in , will be able to see a list of books with (tittle, auother)

2 View browse can be limited by user search

3 Every viewed book has details and add to cart buttons

Add to Shopping Cart: Allows users to add books to shopping cart

1. The system shall allow a registered and logged-in user to temporarily save books that are being considered for purchase into a list associated with their account
2. When viewing the shopping cart list, the system shall display the total price of the books in the cart
3. Delete from Shopping Cart: Allows users to delete books from shopping cart
4. The system shall allow a registered and logged-in user to remove any unwanted books from their shopping cart.
5. The system shall provide the user a way to select one of the books in his/ her cart for deletion.
6. After the user has indicated the particular book to be deleted from their shopping cart, the stored list representing the shopping cart should be updated by removing the list, and the display should be updated to show only the books remaining in the cart.

Checkout: Allows user to checkout

1. The system shall allow a registered and logged-in user to purchase books that are in their shopping cart.
2. The credit card information provided during account registering phase is used for charging the total number of the books in the shopping cart.
3. The system shall allow the user to modify their credit card information if they so desire (before the card is charged).
4. The user will receive a confirmation number.
5. The books (titles and prices and quantities) being purchased will be displayed to the screen.

Update Account Information: Allows users to update account information

1. The system shall allow a user to update their account information.
2. The user shall be allowed to view and change their name, mailing address, billing address, credit card type, credit card number, expiration date, and security code.
3. The user shall be able to change their password by entering the old one once, and a new one twice.

Shipping Status: Allows users to view shipping status

1. The system shall allow a registered and logged-in user to view the shipping status of orders they have placed within the last 30 days.
2. The user must enter a confirmation number to view the shipping status of their order.

3. The system shall then display the shipping date, mailing address, projected delivery date, and status (in transit, delivered, etc.)

Account Purchase History: Allows users to view account purchase history

1. The system shall allow a registered and logged-in user to view purchases made with their account within the last two years.
2. The system shall display the date of purchase, creidit card type, credit card number, expiration date for each order.
3. The system shall display the title of the book, price, and quantity for each book purchased in a given order.

Selling: allow users to add books to sell

1 System allows user to sell their books online by adding the book name and its details

2 System allows user to upload a book if he/she wants to sell a book.

Contact Us: Allows users to contact the online book reselling employees

a. The system shall allow a user to view various methods of contacting the store.
b. The system shall display the store address, telephone number, email address, and directions on how to get to the store.
c. The system should also display a map showing the location of the store.

Logout: Allows users account to logout

1. The system shall allow the registered and logged-in user to exit his/her account, so that access to operations requiring a user to be logged in are now disabled.

Help: Allows users to get help by showing FAQs

1. The system shall allow the user to view an overview of how to used the various operations defined above.
2. The system must allow the user to select one of the operations by name.
3. The system shall then display information on how to use that operation

Functional ADMIN Requirements:

Login: Administrators (web developers) will be required to login at all times

1. The system shall allow a registered admin to log-in to their account.
2. The system shall require a username and password from the admin.
3. The system will verify the username and password, and the admin will be considered logged-in.

View books: Admin can view all the added books online with their details

View transactions: System allows admin to view all the transaction details of buying and selling a book.

View users: All the registered user details will be displayed to the admin

4) Non Functional Requirements:

It is essential for this system to conform to user's needs and demands. Requirements Analysis produced the following non-functional requirements.

Performance Requirements:

• System login/logout shall take less than max 5 seconds.

• Searches shall return results within max 7 seconds.

• Orders shall be processed within max 7 seconds.

• System shall support approximately 5000 simultaneous users.

Reliability

The average time to failure shall be minimum of 30 days. In the event that a server does crash, a backup server will be up and running within an hour.

Availability

The Online Book Reselling shall be available to users 24 hours a day, 7 days a week, with the exception of being down for maintenance no more than one hour a week. If the system crashes, it should be back up within one hour

Security

Users will be able to access only their own personal information and not that of other users. Purchases will be handled through a secure server to ensure the protection of user's credit card and personal information.

Maintainability

Any updates or defect fixes shall be able to be made on server-side computers only without any patches required by the user.

The System Must be Easy to Use

Users should be able to learn this system very quickly. It should be fairly intuitive. The reports should be clear and precise, and not overly technical.

The System Should not be overly technical

The information produced by the system should be understandable by anyone with general knowledge. Details should be written in a very understandable fashion.

5) **Domain Requirements Specification:** www.onlinebookreselling.com

6) **Design & Implementation Constraints:**

1 easy to use

2 attractive interface

3 max response time 7s
4language can be written

5- skilled programmers

7) **System Evolution:**

Anticipated changes: total users will increase

How should any anticipated changes in the future: due to users increasing we will need more powerful hardware, continuous maintain and virus defender.

8) **requirements discovery:** all Design & Implementation Constraints

9) **requirements validation techniques:**

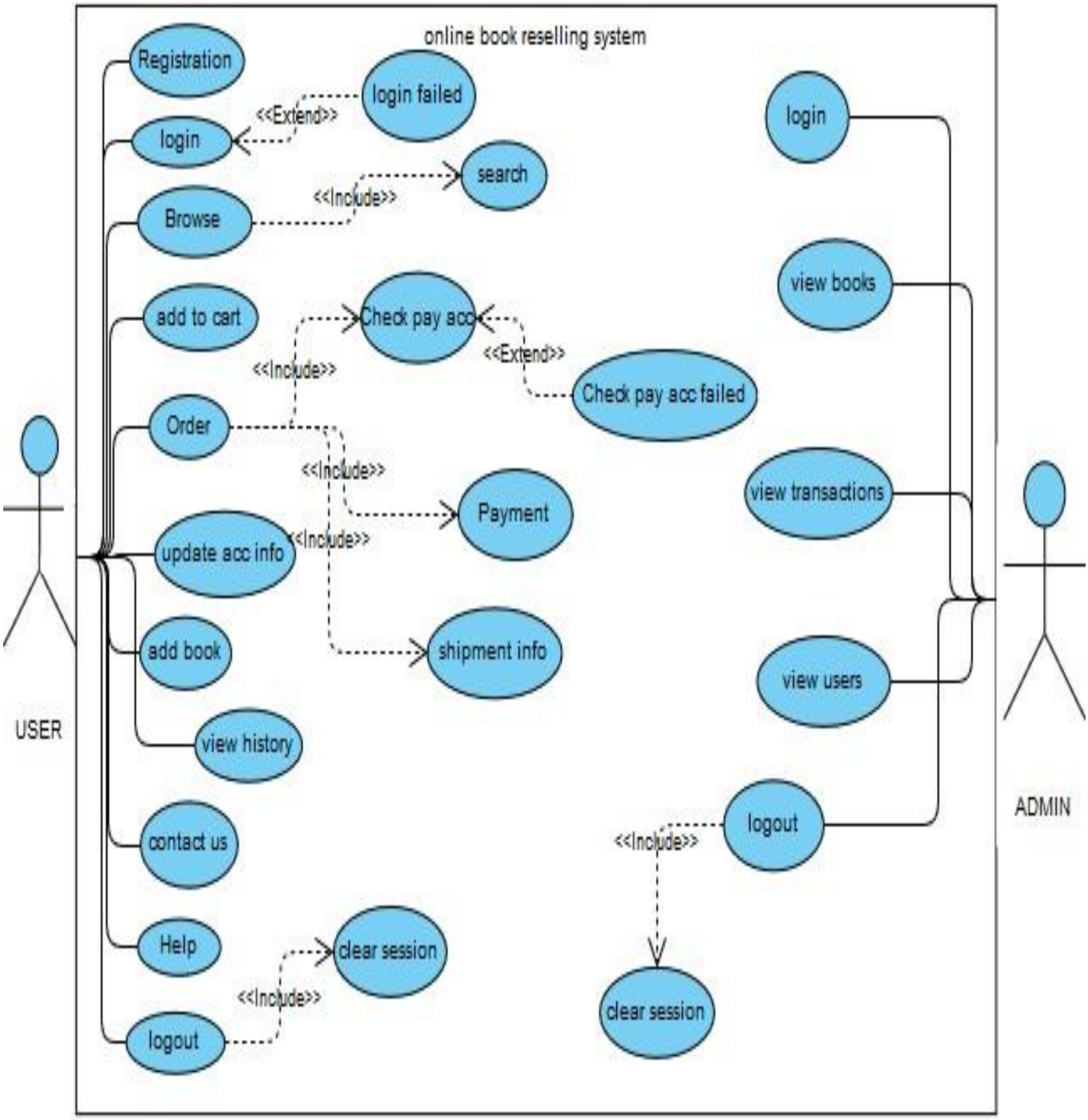
Test case generation: writing SQL test cases for verifying and testing database functionalities

Prototyping: the form of the product’s interface, front-end design.

PART 2: System Design & Models:

10) Functional Diagrams:

A) Use-Case Diagram:



B) Detailed USER Use-Cases Description:

1) Register

Purpose	If the user doesn't have an account then he will be asked to register
Actor	User
Input	: The user will enter details in the registration form according to the required fields. The fields include: 1. Username 2. Password 3.confirm password 4.Email
Output	After registration the user will be directed to the main home page

2) Login (USER)

Purpose	If the user wants to get access to all the functionalities of Online Book Store he should login using his username and password.
Actor	User
Input	The user will enter his username and password
Output	If it is a successful login the user will be directed to the main home page. Else if the user enters invalid information he will be asked to re enter information

3) Login failed

Purpose	If there are any problem in logging return massage
Actor	User
Input	Username and password
Output	Incorrect username or password

4) Browse

Purpose	List books, names, titels, aouthers to user
Actor	User
Input	Browse request
Output	Lbooks list

5) search

Purpose	A user can search for a book of his choice by selecting category and title. Then a select query is used to retrieve data from the database and display the selected information.
Actor	User
Input	The user will select a category and enter title in a text box provided
Output	The system will display the books which matches the selected search criteria

6) Add to cart

Purpose	Checkbox to be added to order
Actor	User
Input	Click checkbox
Output	Selected or not selected book

7) Order

Purpose	Start buying books process
Actor	User
Input	Order request
Output	the new book will be inserted in the website

8) Check pay acc

Purpose	Contact bank to determine if account payment information is valid
Actor	User
Input	Credit card info
Output	Accepted transaction

9) Check pay acc failed

Purpose	Display a massage if account not valid for payment
Actor	User
Input	Invalid payment return operation
Output	Failed massage

10) payment

Purpose	If the user wants to purchase a book then he can place an order by selecting the add to shopping cart button and entering the quantity required under the book description.
Actor	User
Input	The user will enter the quantity required and click the add to shopping cart button.
Output	The order will be added to the user’s shopping cart

11) Shipment info

Purpose	Enable user to enter shipment data
Actor	User
Input	Shipment address
Output	Shipment address

12) Update Acc info

Purpose	If the user wants to change his personal account information then he can update his/her selected fields and the entire data will be updated.
Actor	User
Input	The user will update his account information
Output	The system will update the entered information

13) Add Book

Purpose	If the user wants to add a book then he/she can upload a book
Actor	User
Input	If user wants to add a book the he/she should click the add book button and enter book details
Output	the new book will be inserted in the website

14) View history

Purpose	List transaction history
Actor	User
Input	Request history action
Output	Historical transaction

15) Contact us

Purpose	Display contact info
Actor	User
Input	Request event
Output	Contact info page

16) Help

Purpose	Provide help service to user
Actor	User
Input	Help request
Output	Help page contant

17) Clear session

Purpose	Secure system
Actor	User
Input	Logout event
Output	Cleared session

18) Logout (USER)

Purpose	If the user wants to end his session and sign out of the website then he can use the logout option
Actor	User
Input	The user will click the logout button
Output	The user's account session comes to an end and he should login again if he wants to enter into the website.

Detailed ADMIN Use-Cases Description:

1) Login

Purpose	If the user wants to end his session and sign out of the website then he can use the logout option
Actor	Admin
Input	The user will click the logout button
Output	The user's account session comes to an end and he should login again if he wants to enter into the website.

2) View books

Purpose	If the user wants to end his session and sign out of the website then he can use the logout option
Actor	Admin
Input	The user will click the logout button
Output	The user's account session comes to an end and he should login again if he wants to enter into the website.

3) View transactions

Purpose	If the user wants to end his session and sign out of the website then he can use the logout option
Actor	Admin
Input	The user will click the logout button
Output	The user’s account session comes to an end and he should login again if he wants to enter into the website.

4) View users

Purpose	If the user wants to end his session and sign out of the website then he can use the logout option
Actor	Admin
Input	The user will click the logout button
Output	The user’s account session comes to an end and he should login again if he wants to enter into the website.

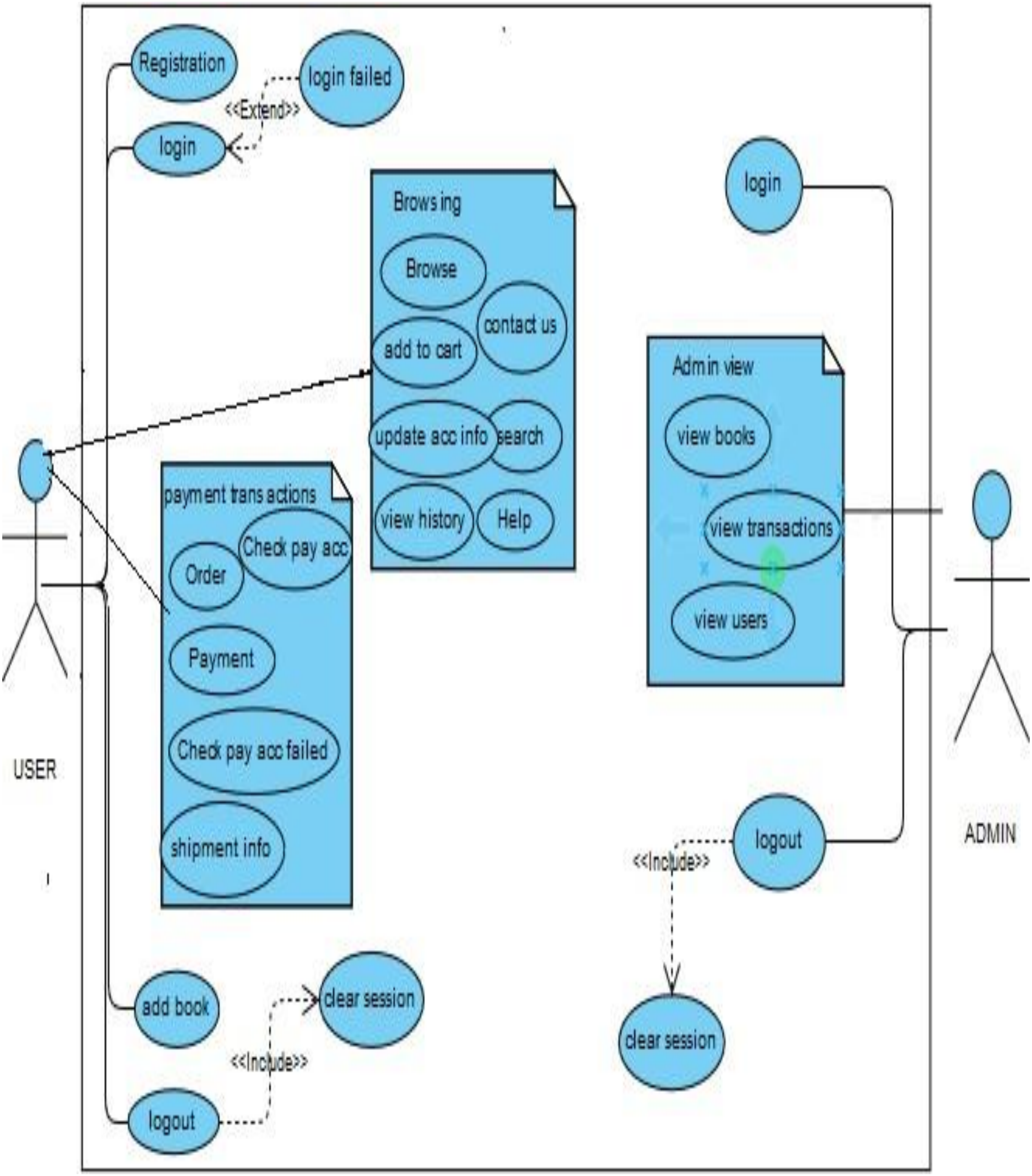
5) Logout

Purpose	If the user wants to end his session and sign out of the website then he can use the logout option
Actor	Admin
Input	The user will click the logout button
Output	The user’s account session comes to an end and he should login again if he wants to enter into the website.

6) Clear sessions

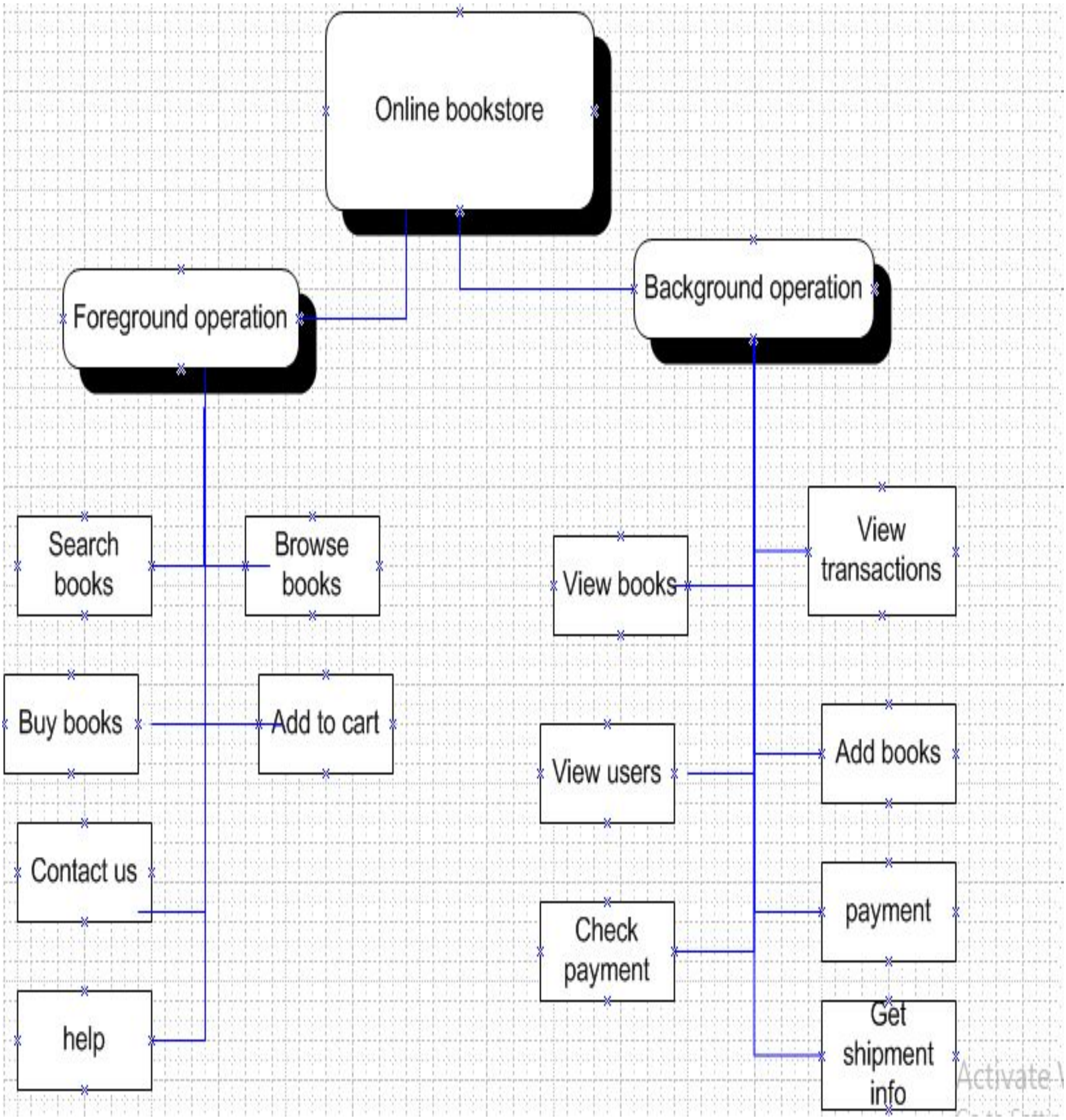
Purpose	If the user wants to end his session and sign out of the website then he can use the logout option
Actor	Admin
Input	The user will click the logout button
Output	The user’s account session comes to an end and he should login again if he wants to enter into the website.

c) Package Diagram grouping relevant Use-Cases into Packages:

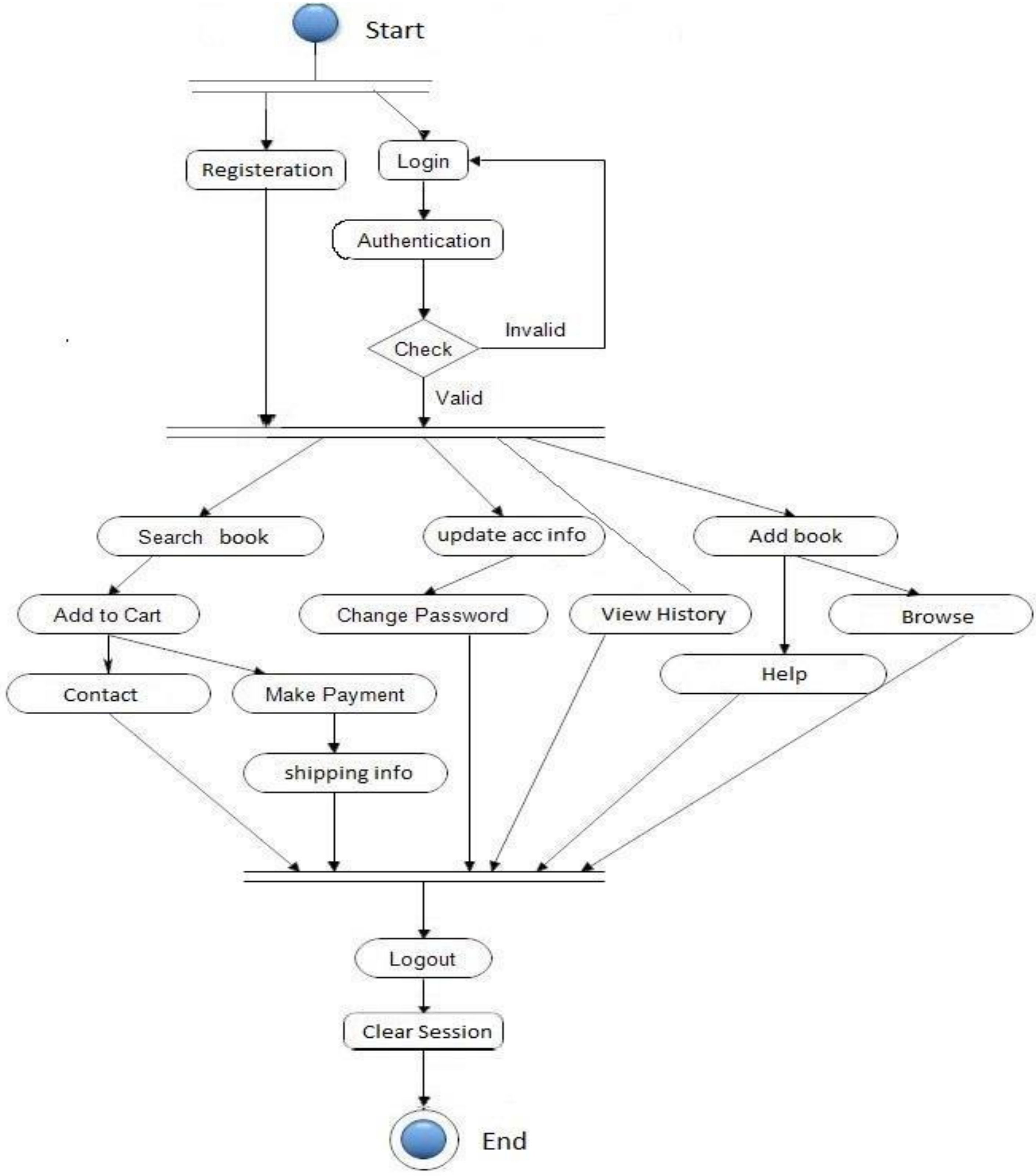


11) Structural & Behavioural Diagrams:

a) System Architecture:



b) Activity Diagrams:



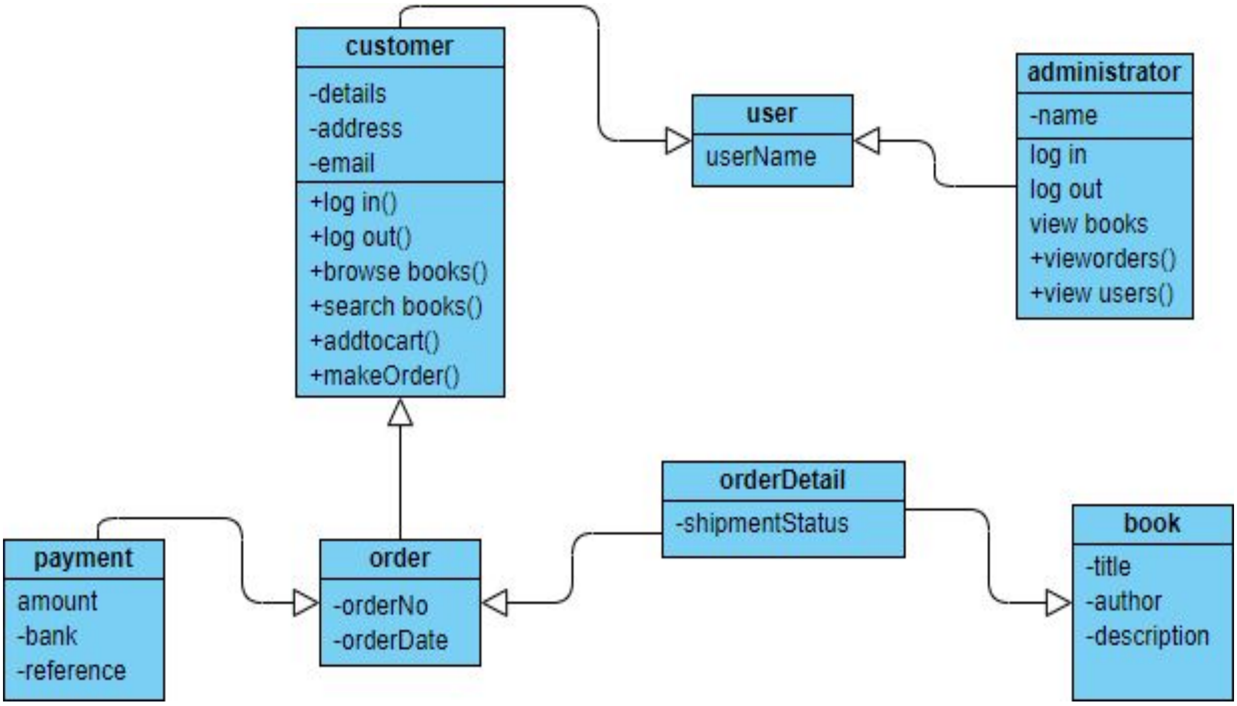
Design Pattern:

MVC pattern architecture gives us the idea of separation of concern, it helps us to implement the separation of concern among the model, view and controller classes within applications.

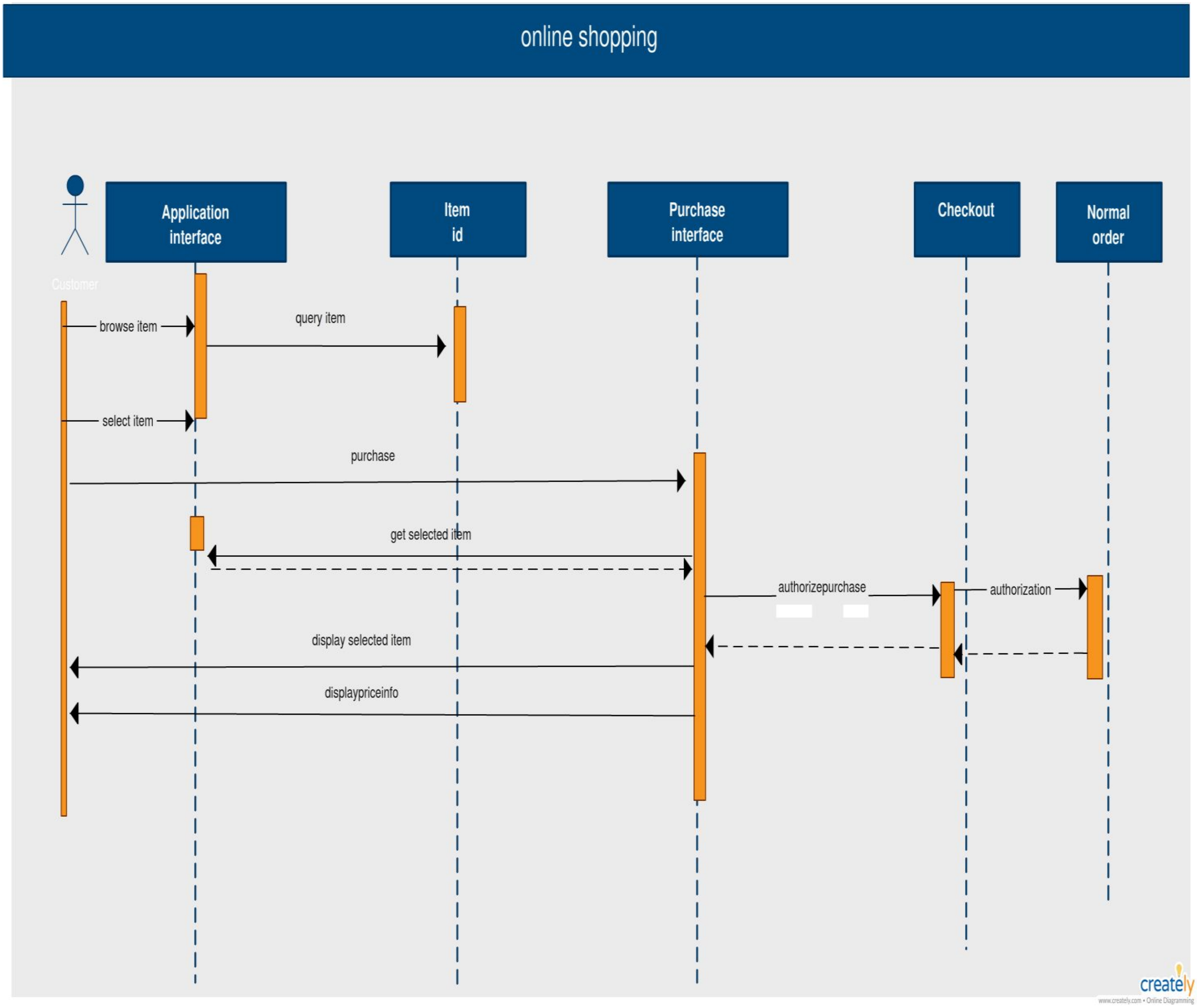
Separation of concern makes it easy for us to test our application as relation among different components of application is clearer and coherent. MVC help us to implement a test-driven development approach, in which we implement automated test cases before we write the code. These unit test cases help us predefine and verify requirements of new code before writing it.

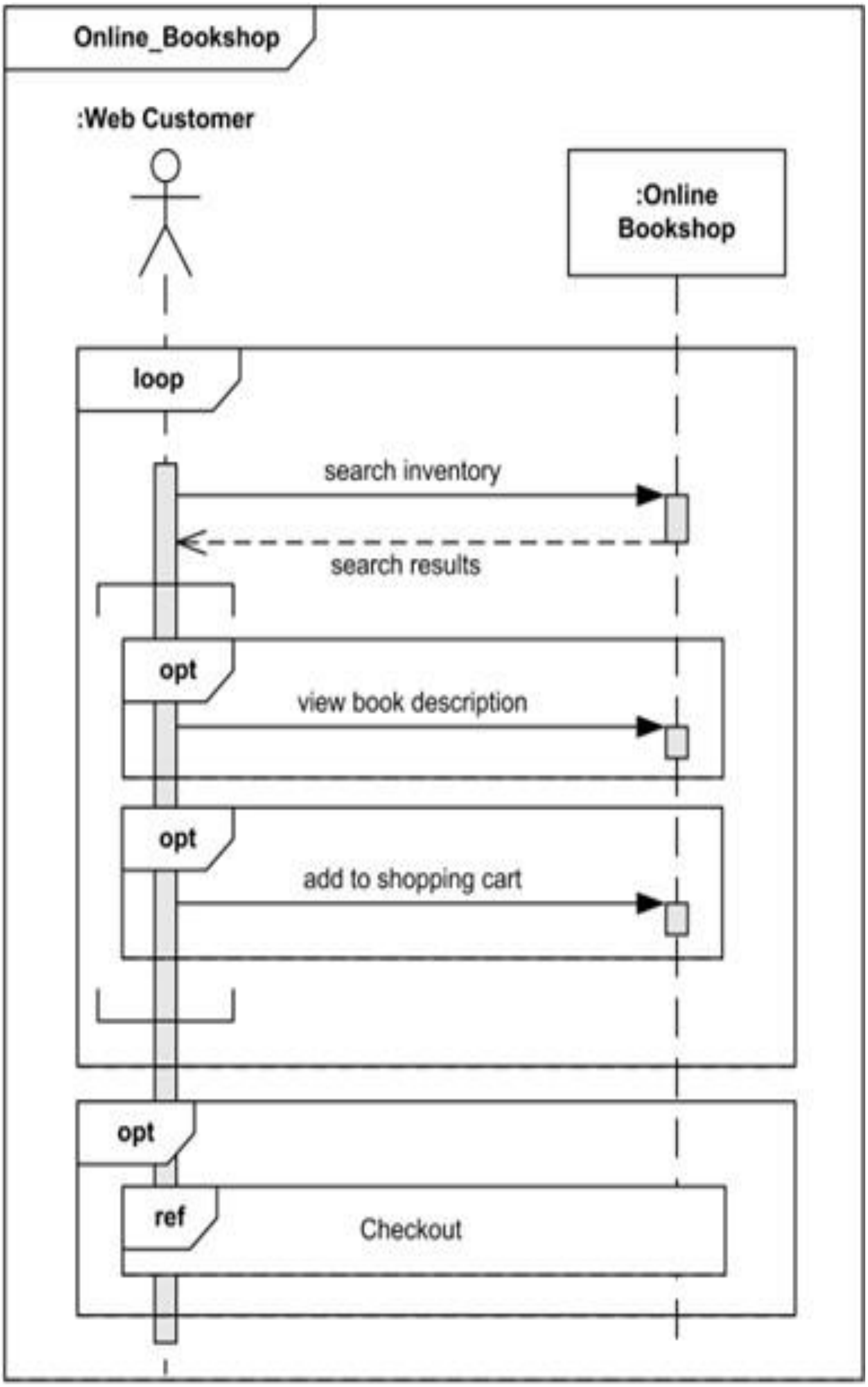
If we are making an application with enough serious stimulating on the client side to refuse to go along with JavaScript alone. If we are developing an application which have a very high lifting on the server side and a little communication on the client side then we should not use the MVC pattern architecture, instead we should use simple setup such as web-based form model

Class diagram:

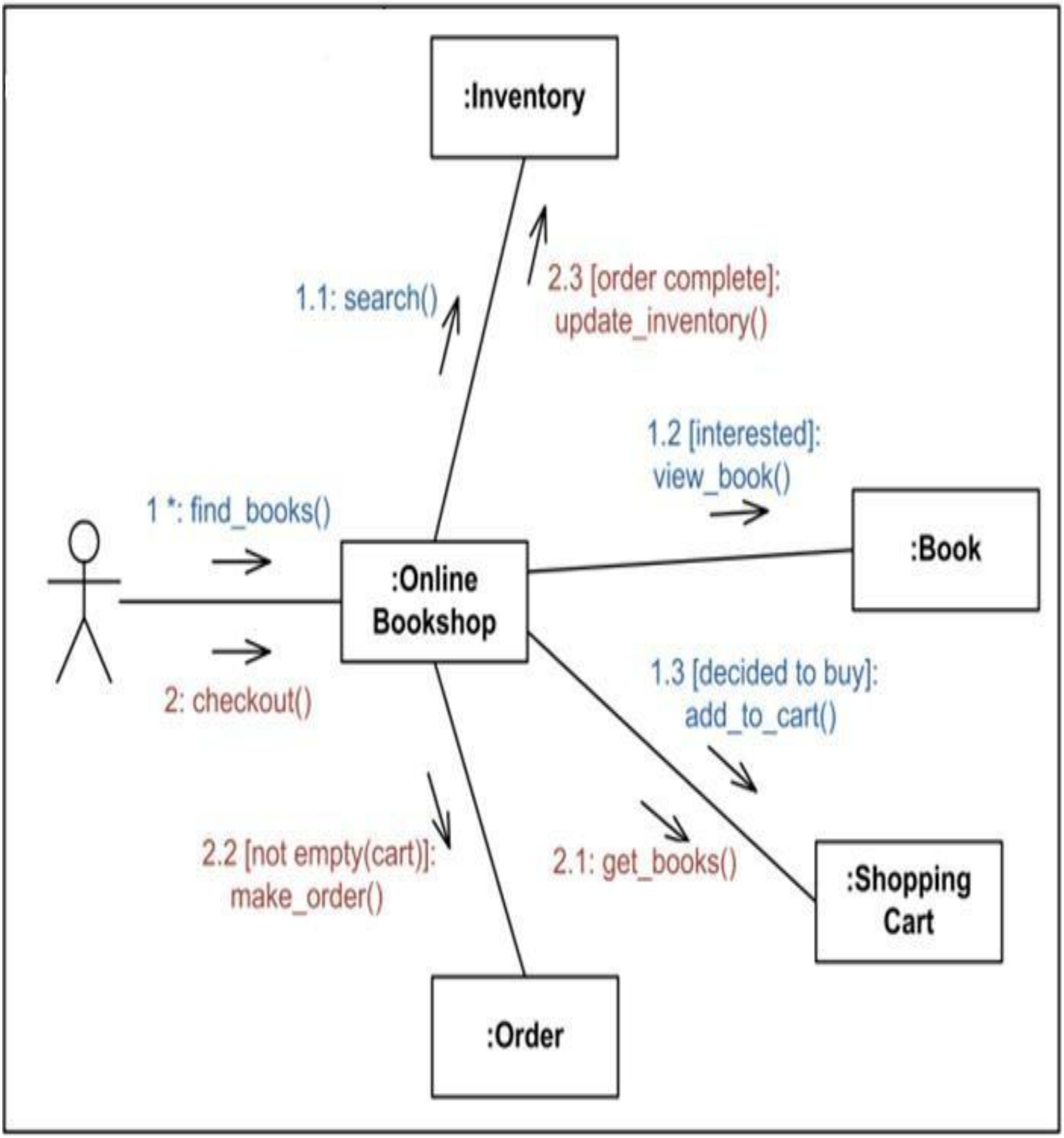


Sequence Diagram:

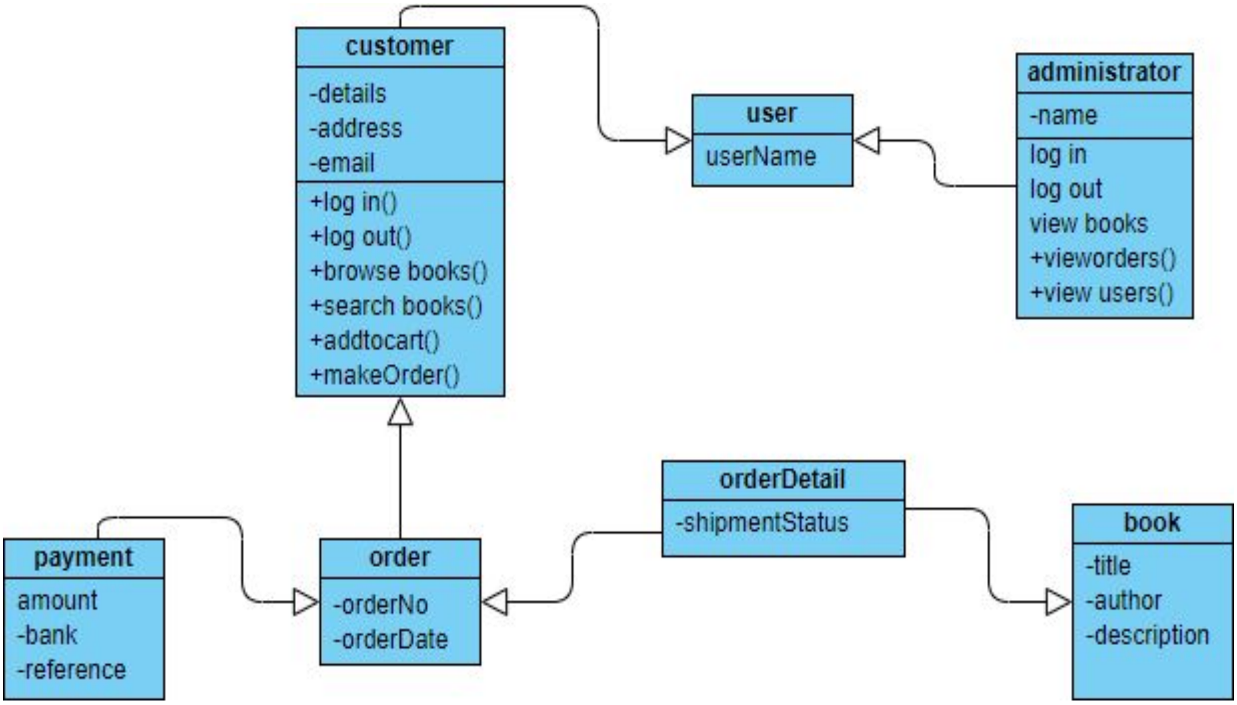



Online_Bookshop
:Web Customer
:Online Bookshop
loop
search inventory
search results
opt
view book description
opt
add to shopping cart
opt
ref
Checkout

Communication Diagram:



Class diagram:


customer
-details
-address
-email
+log in()
+log out()
+browse books()
+search books()
+addtocart()
+makeOrder()
user
userName
administrator
-name
log in
log out
view books
+vieworders()
+view users()
payment
amount
-bank
-reference
order
-orderNo
-orderDate
orderDetail
-shipmentStatus
book
-title
-author
-description