

**Project Report**  
**on**  
**UML Modeling using COMET Methodology**  
**Online Book Shopping (E-kart Library): Book Buy**

**SYSC 5708**  
**Model-Driven Development of Real-Time and Distributed Software**

**Department of Systems and Computer Engineering**  
**Carleton University**

**Submitted to: Dr. Farhana Islam & Dr. Adnan Faisal**

**Submitted by:**

**Vithika Joshi**  
**Student Id: 300036426**  
**Master of Engineering,**  
**Electrical and Computer Engineering,**  
**ECE, University of Ottawa**

**Navneet Kaushal**  
**Student Id: 101094963**  
**Master of Engineering, Electrical and**  
**Computer Engineering,**  
**SYSC, Carleton University**

## Table of Contents

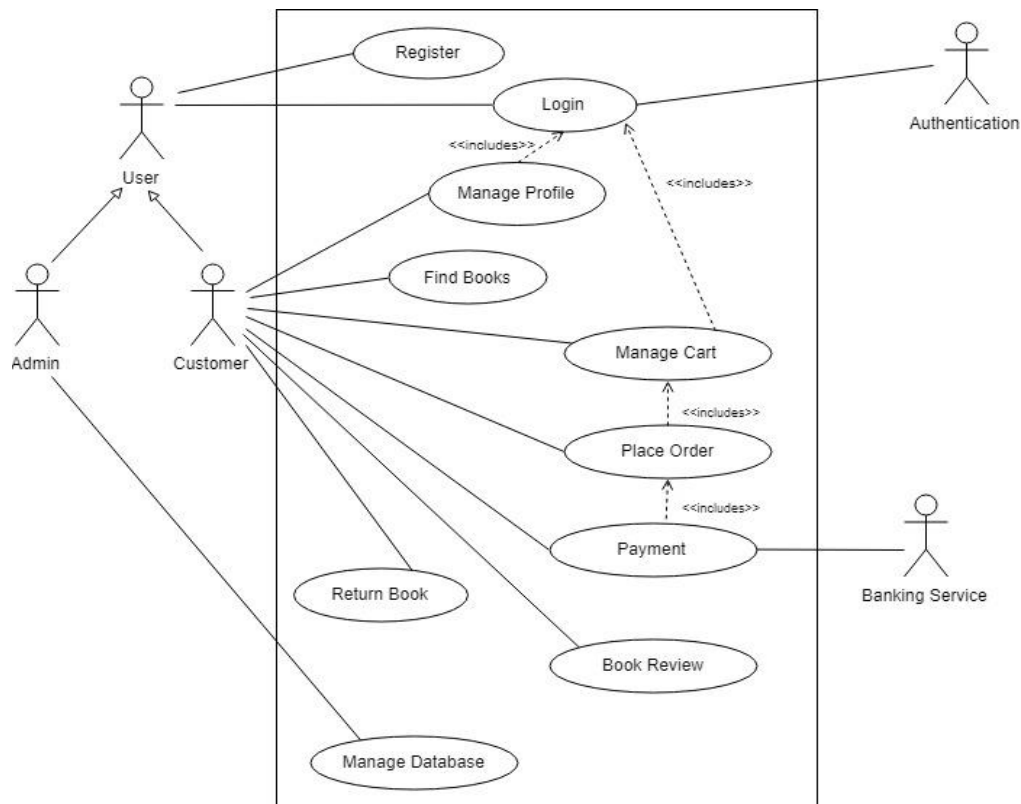
I.	PROBLEM STATEMENT.....	3
II.	REQUIREMENT MODEL.....	3
	1. USE CASE DIAGRAM.....	3
	2. TEXTUAL DESCRIPTION.....	4
III.	ANALYSIS STATIC MODEL.....	4
	1. CLASS AND ENTITY DIAGRAM.....	4
	2. SYSTEM CONTEXT MODEL.....	6
	3. EXTERNAL AND BOUNDARY CLASSES.....	7
IV.	ANALYSIS DYNAMIC MODEL.....	7
	1. SEQUENCE DIAGRAM.....	7
	2. STATE MACHINE DIAGRAM.....	13
V.	DESIGN MODEL.....	18
	1. SOFTWARE ARCHITECTURE.....	18
	2. CONCURRENT TASK ARCHITECTURE.....	19
	3. DEPLOYMENT DIAGRAM.....	20

## I. PROBLEM STATEMENT

This project aims to develop an **online book shopping** system named **BOOK BUY** which allows the web customer to purchase or rent books online. There can be two types of users of this system: Customer and Admin. Customer needs to log in to the website using its unique login credentials to purchase or rent books, similarly, an admin needs to log in to make changes to the website or for managing the database of the whole system. Customer can browse through the website without log in to the system as well, however, in such cases customer will not be able to place its orders since item cannot be added to the shopping cart without user authentication. Customer can browse through the product catalog and add the items to the shopping cart. Customer can proceed to the checkout as long as his shopping cart is not empty. Once customer enters its credit card details for payment, the bank authenticates the details provided by the customer. Once authentication has been done, the order is placed, and an Order Id is generated in the name of the Customer. While creating a new account, a customer needs to provide full name, phone number, and billing address details. Customer can also log in to the system to maintain his account information such as changing phone number, address and to check the status of his orders.

## II. REQUIREMENTS MODEL

### 1. USE CASE DIAGRAM



*Figure 1. Use Case Model of Book Buy*

As per the given problem statement, a high-level Use-Case diagram is drawn in Figure 1.

## 2. TEXTUAL DESCRIPTION

These two, complex, use-cases are selected, and their description are available in the COMET template:

### Use case name: Find Books

- **Actor:** Customer
- **Preconditions:** Correct URL is requested; Website gets launched successfully
- **Description:**
  1. Customer logs in to the website using its credentials
  2. Customer searches for a booking using the search bar
  3. System checks the availability of the book
- **Alternatives:** If requested book is not available, system displays “Book unavailable” error message
- **Postcondition:** Requested book is displayed by the system

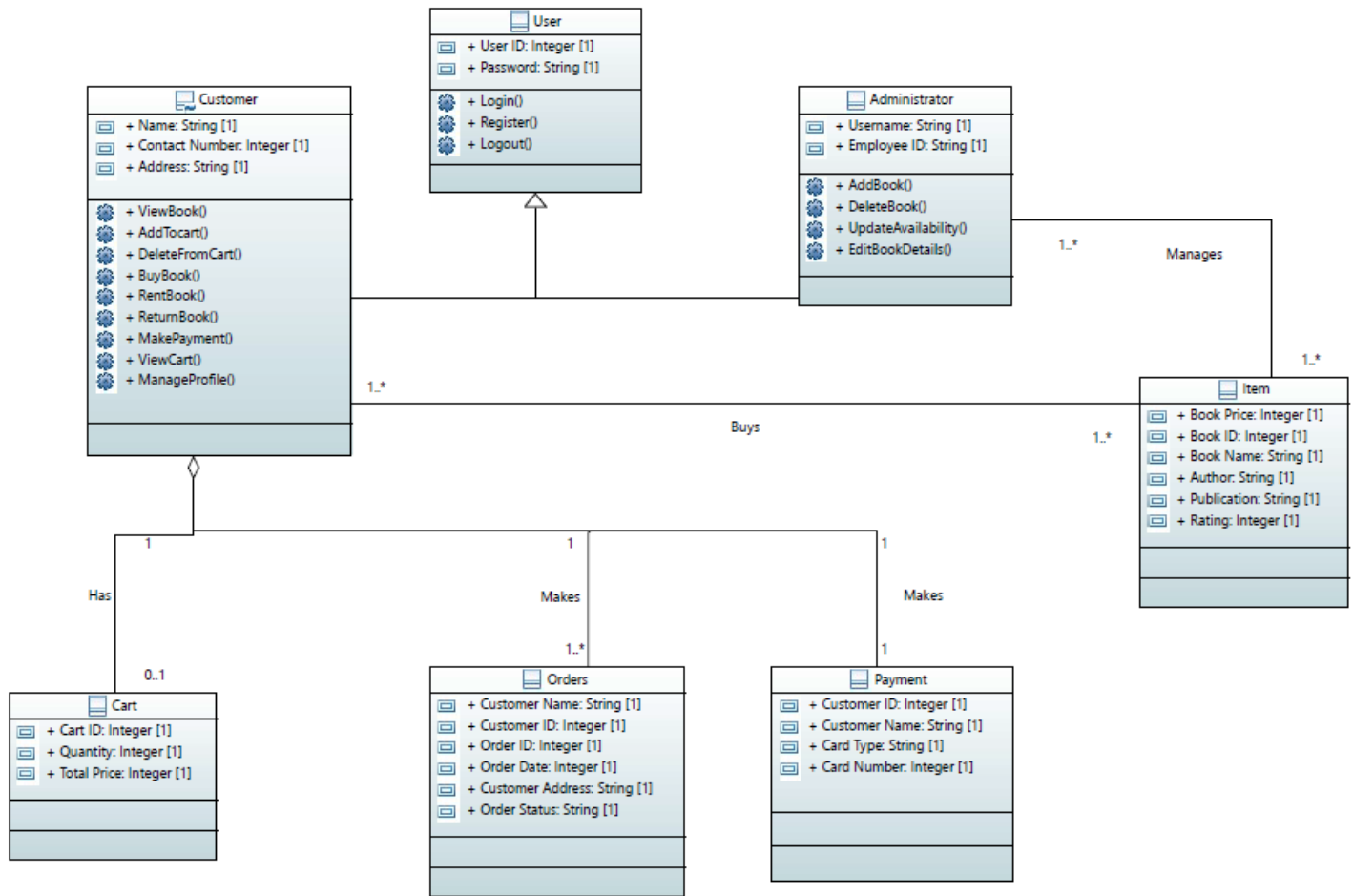
### Use case name: Payment

- **Actor:** Customer
- **Preconditions:** Customer should be logged in to the system and items should be added to the cart
- **Description:**
  1. Customer proceeds towards payment by clicking the payment button
  2. Customer selects a payment method and provides banking details for order confirmation
- **Alternatives:** If banking details are not authenticated, payment is not confirmed
- **Postcondition:** Payment is confirmed, and order is placed

## III. ANALYSIS STATIC MODEL

### 1. CLASS AND ENTITY DIAGRAM

A class diagram is a type of static structure diagram that defines the structure of a system by showing the system’s classes, their attributes, operations (or methods), and the relationships among objects. The class diagram, as shown in Figure 2, depicts the various classes of the online shopping system – User, Customer, Admin, Item, Cart, Orders, and Payment. Customer and Admin classes are specializations of User class, which means that both the classes are inherited from the User class. Admin has a simple association with the Item class.

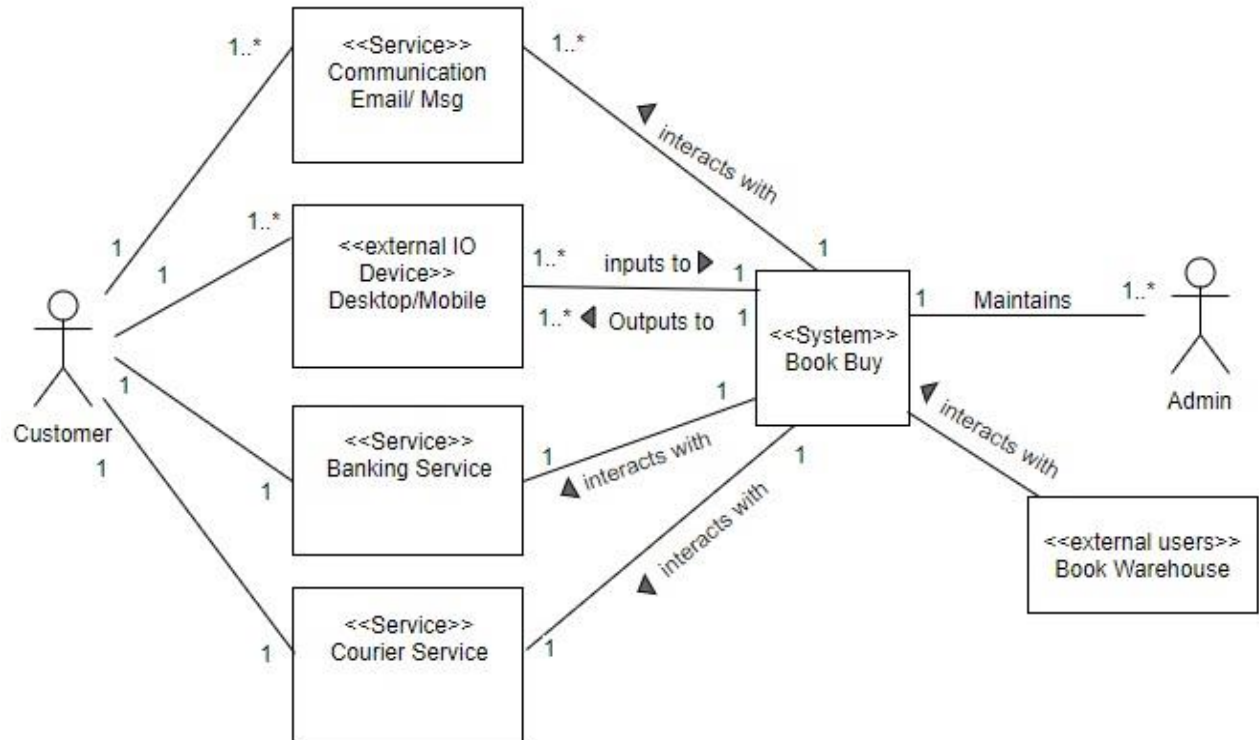


**Figure 2. Class/Entity Diagram of Book Buy**

One or more instances of Admin class is related to one or more instances of the class Item. In the similar way, Customer class is associated with Cart, Orders and Payment class. One instance of the customer class is associated with zero or one instance of class cart. There can be a scenario in which a user has not logged in to the website, in such cases cart is not allotted to the customer, therefore, there can be zero instances in this case. One instance of customer class is associated with one or more instance of Orders class since, one customer can make multiple orders. Similarly, an instance of customer class is associated with an instance of Payment class. Figure 2 is also an entity relationship diagram since it depicts how entities relate to each other within a system.

## 2. SYSTEM CONTEXT MODEL

A system context diagram is a diagram that shows system as a whole and it represents all the external entities that may interact with the system. Our online shopping system (Book Buy) and its interaction with all external entities is shown in Figure 3. The various external entities interacting with the system are as follows: Communication Service, Banking Service, Courier Service, IO devices and Book Warehouse.



*Figure 3. Context Model of Book Buy*

The communication service is used to communicate with the user regarding order confirmation, order cancellation, order dispatch etc. The customer interacts with the website using IO devices. These can be mobile, desktop, tablets etc. The user inputs using one or more such devices and in the similar way system outputs the result to one or more such devices. Payment is initiated by the customer by interacting with the Banking Services. Moreover, for the payment confirmation the system interacts with a banking service. Similarly, system and customer interacts with the Courier Service for the delivery of orders. Also, it can be seen in Figure 3 that one or more administrators maintains the system and book warehouse are the external users that also interacts with the system.

### 3. EXTERNAL AND BOUNDARY CLASSES

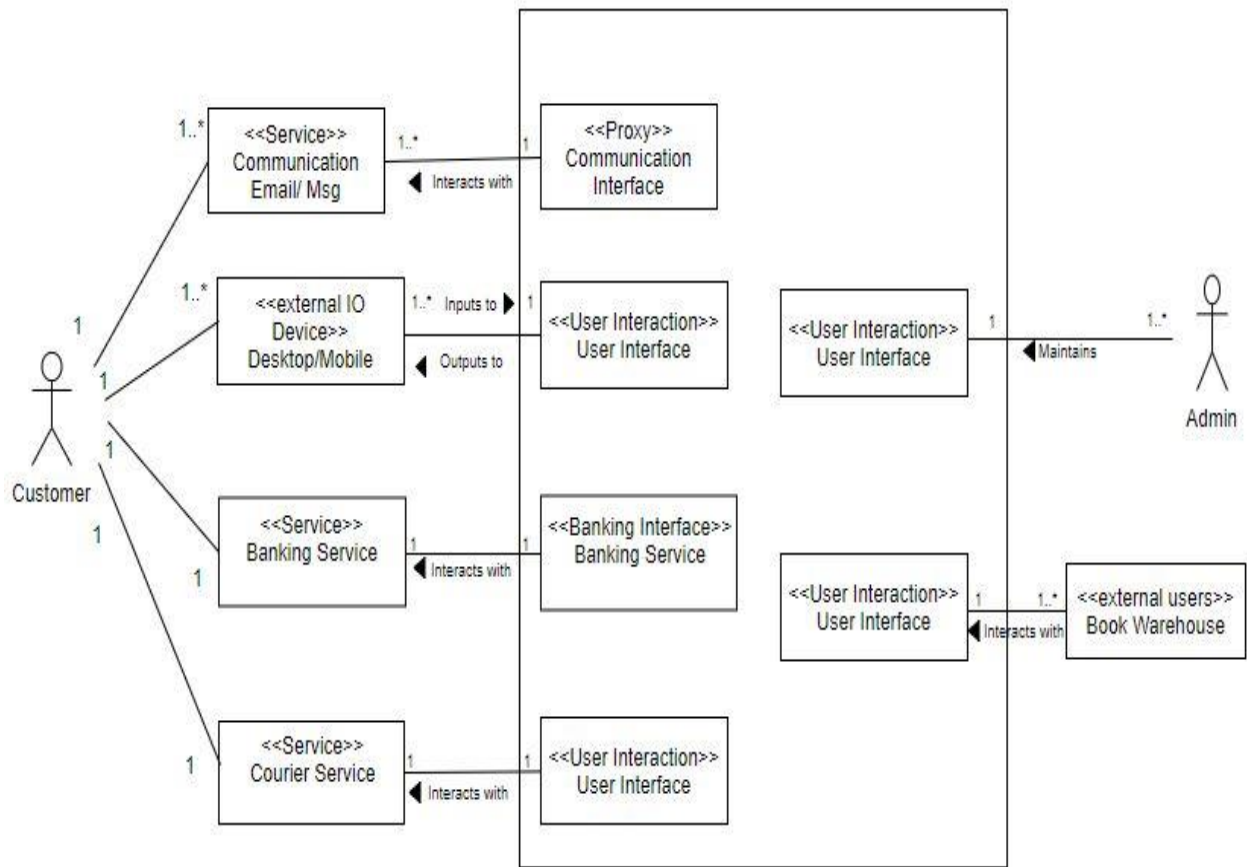


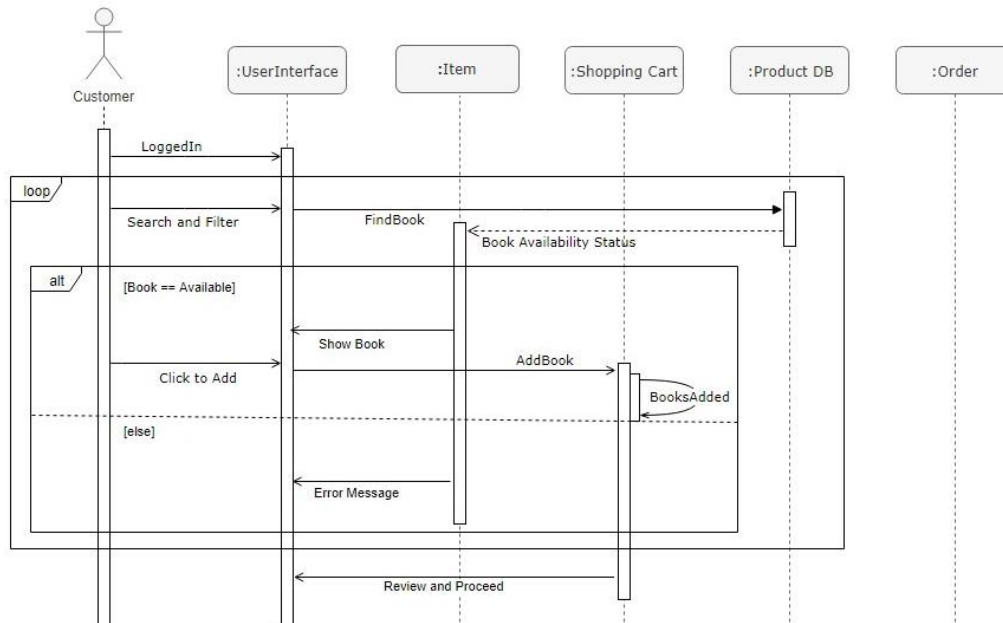
Figure 4. Boundary Class Diagram of Book Buy

From the system context model, boundary objects inside the system has been added that interact with the external objects as shown in Figure 4.

## IV. ANALYSIS DYNAMIC MODEL

### 1. SEQUENCE DIAGRAM

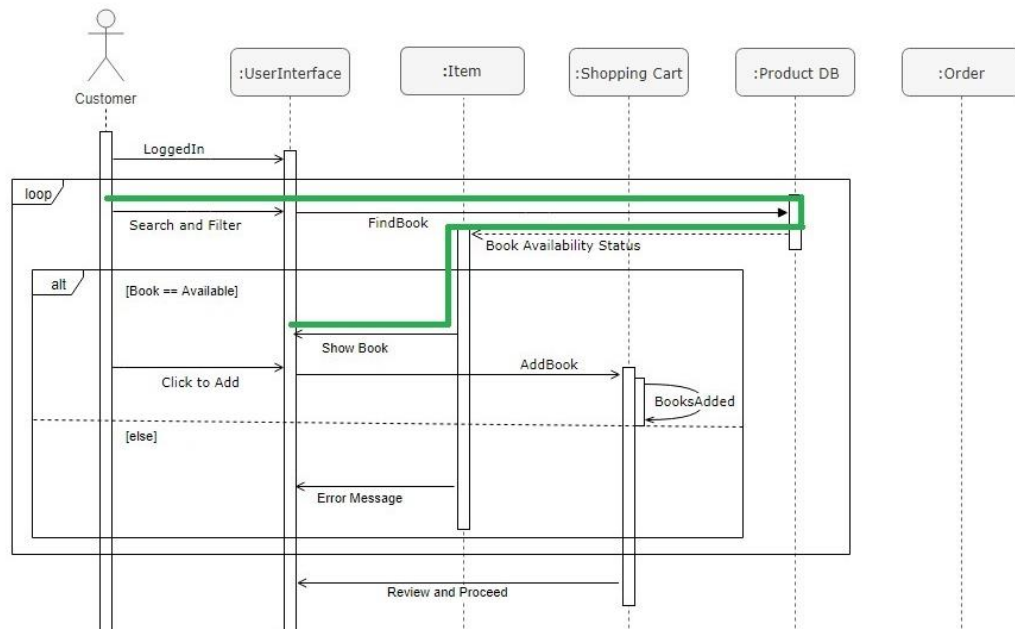
Sequence diagrams are drawn for the two selected use cases: Find books and Payment. Figure 5 shows the sequence diagram for Find Books use-case whereas Figure 8 shows the sequence diagram continuing the Payment use-case.



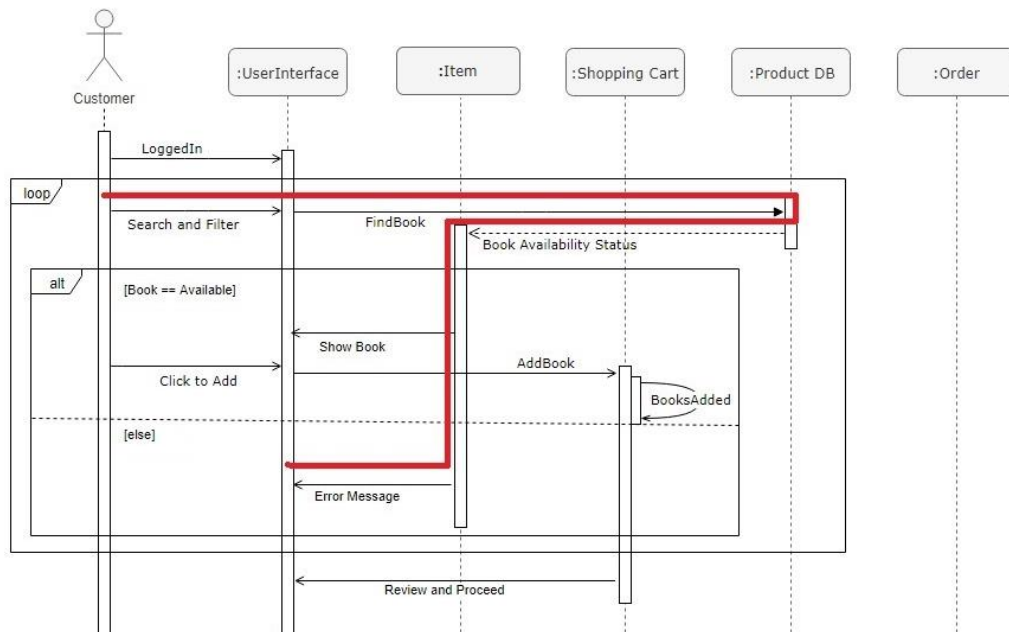
**Figure 5. Sequence Diagram of Find Books Use Case**

The horizontal axis illustrates the Interaction partners and the vertical axis is the chronological order of the interaction. An Asynchronous message is sent from the object of customer class to the object of User Interface class (logged in). Another Asynchronous message “Search and Filter” is sent from object of customer class to User Interface class object. The User Interface class object sends a synchronous message “Find Books” to the Product DB. It then waits for a response from the Product DB. The Product DB then sends a response message of the book availability status to the Item class. The loop block in the sequence diagrams shows a repeated interaction, whereas, alt block depicts an alternative interaction. The alt block checks for the following guard condition [Book == Available], if this conditions equals true then an asynchronous message “show book” is sent by object of Item class to the user interface class. The customer can then add the book to its cart which can be done by sending asynchronous message “add book” from user interface class to shopping cart. If the guard condition doesn’t hold true, an error message is sent from the Item class to the User Interface. For placing the order, the customer sends an asynchronous message to User Interface and User Interface to order class. Asynchronous message “SelectPaymentMethod” is sent by the object of order class to the User Interface class for selecting a payment method. Once a method is selected and user is ready to pay, a new object Banking Service is created for authentication and processing of all banking related details. The new object sends an asynchronous message “RequestBankDetails” to the User Interface. The customer then provides the banking credentials to the User Interface and it sends an asynchronous message “MakePayment” to the Banking Service object. The Banking Service object sends a response message regarding the Payment Status. The guard condition [Payment == Done] is checked, if true, “order placed successfully” message is sent by Order class to User Interface, otherwise, “Order on Hold” message is sent.



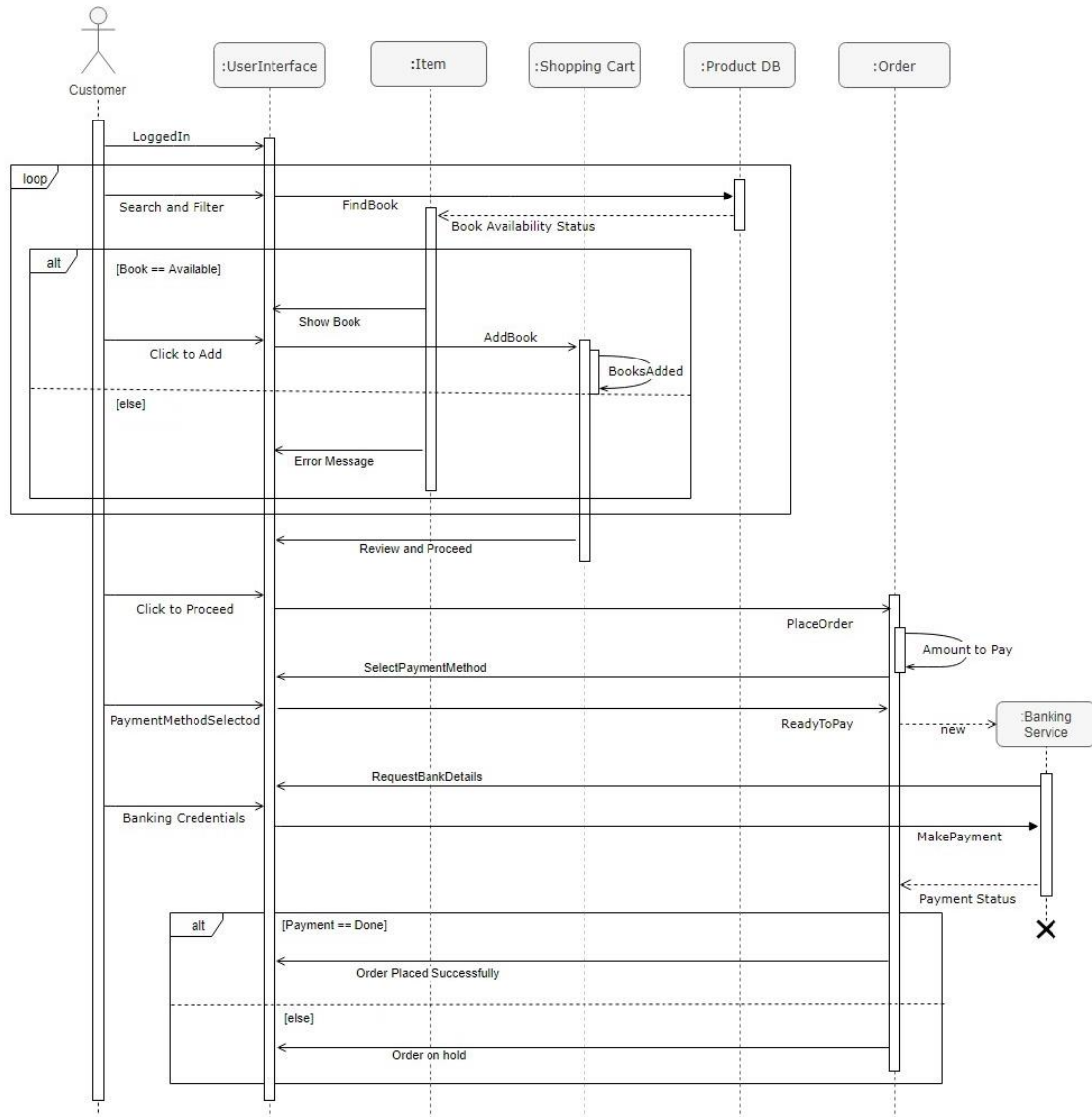


**Figure 6. Sequence Diagram of Find Books Use Case – Happy Path**

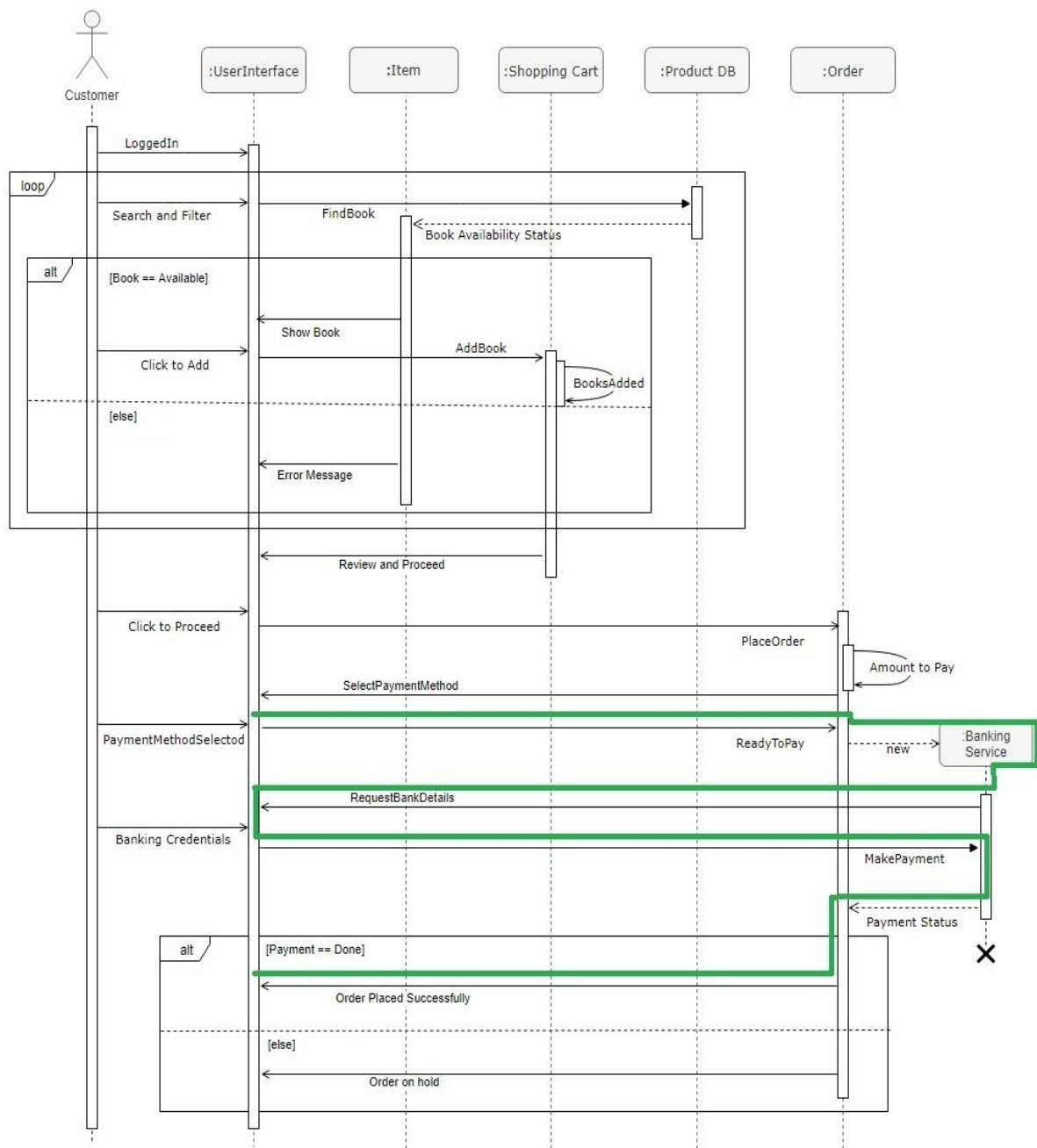


**Figure 7. Sequence Diagram of Find Books Use Case – Unhappy Path**

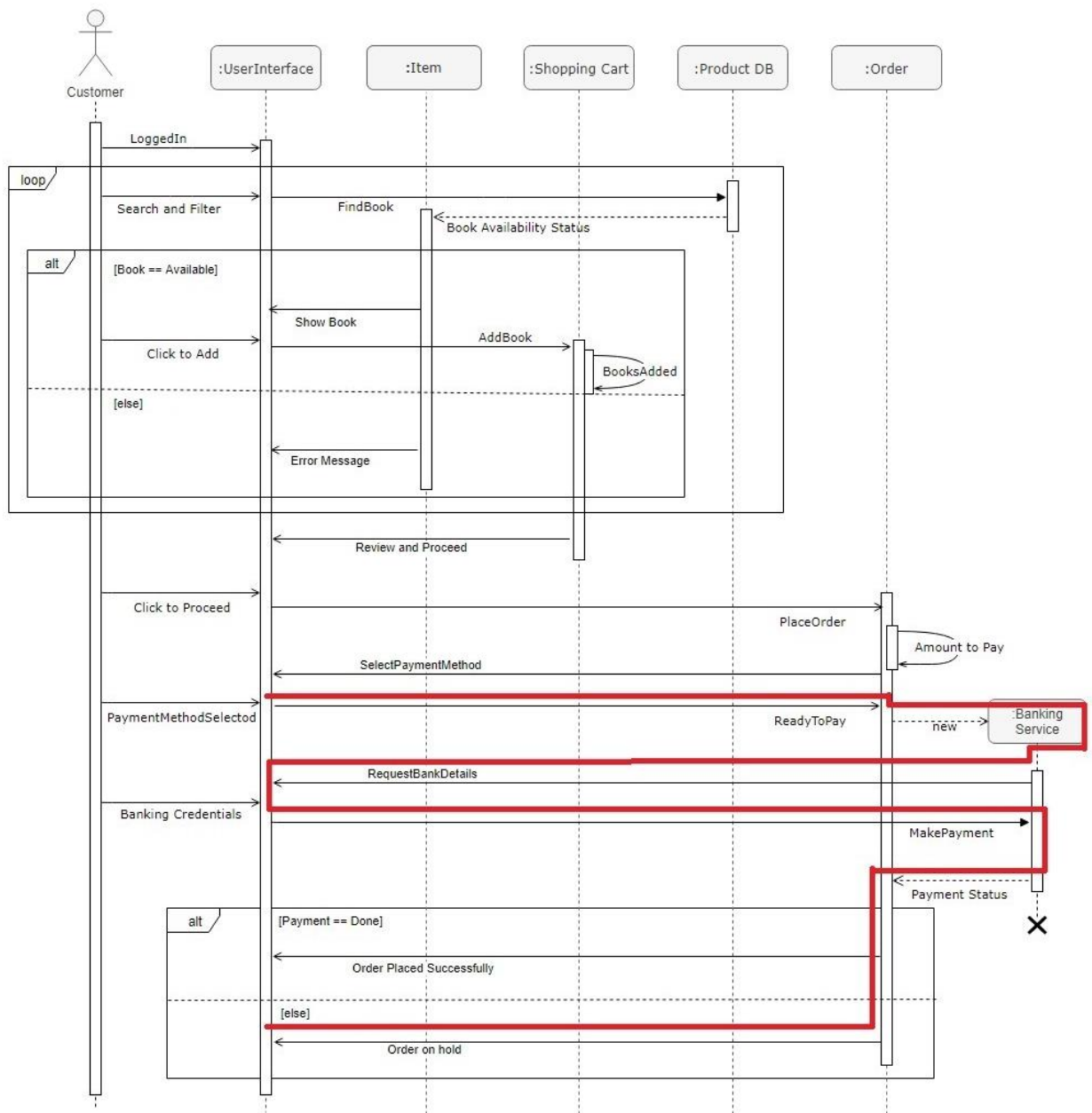
Figure 6 and 7 depict the happy and unhappy path of the use case Find Books. Happy path is a path which executes without exception and produces an expected output as can be seen marked by green line in the diagram. The red line in the diagram shows the occurrence of an exception, therefore, the expected output was not received, and an error message is thrown by the system. The happy and unhappy path of the use case “Payment” are shown in Figure 9 and 10 respectively.



**Figure 8. Sequence Diagram including Payment Use-Case**

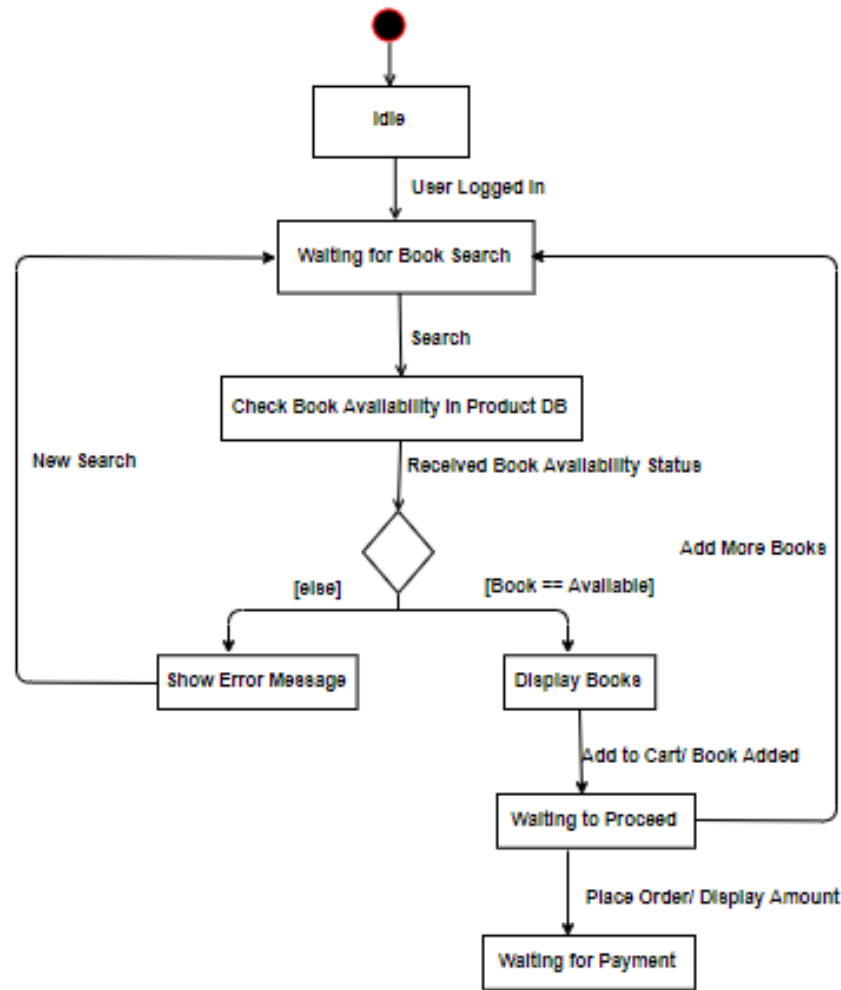


**Figure 9. Sequence Diagram including Payment Use-Case – Happy Path**



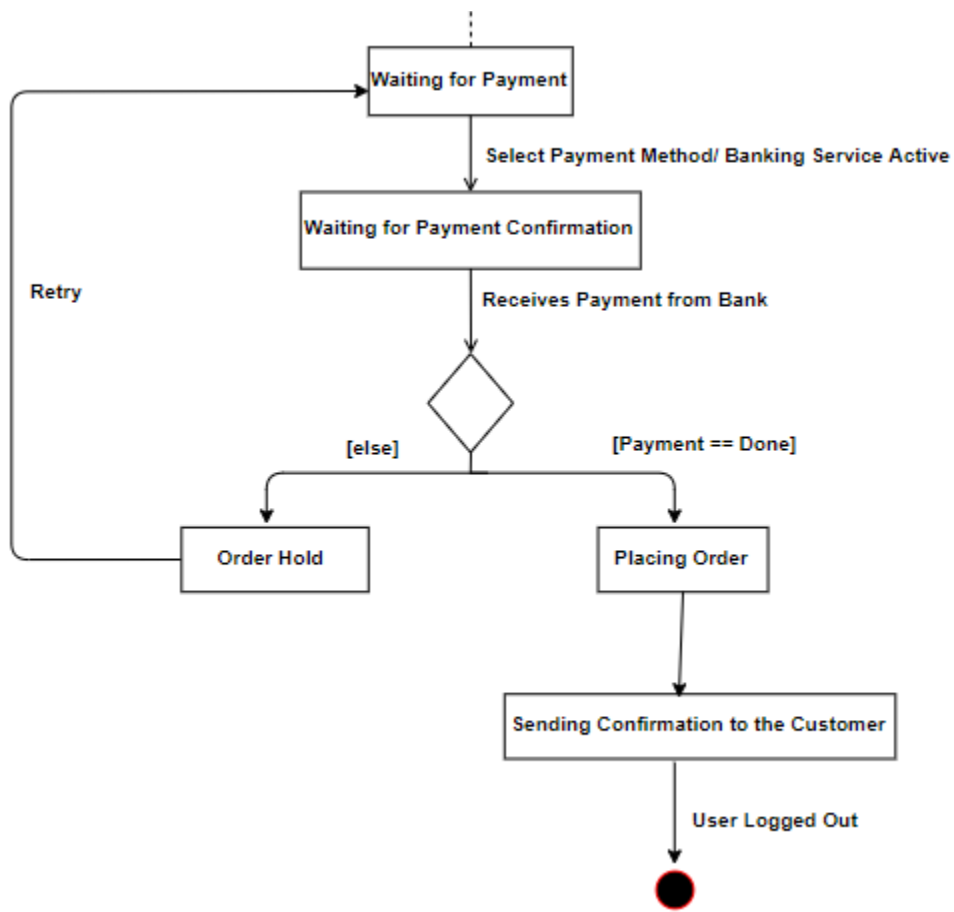
**Figure 10. Sequence Diagram including Payment Use-Case – Unhappy Path**

## 2. STATE MACHINE DIAGRAM



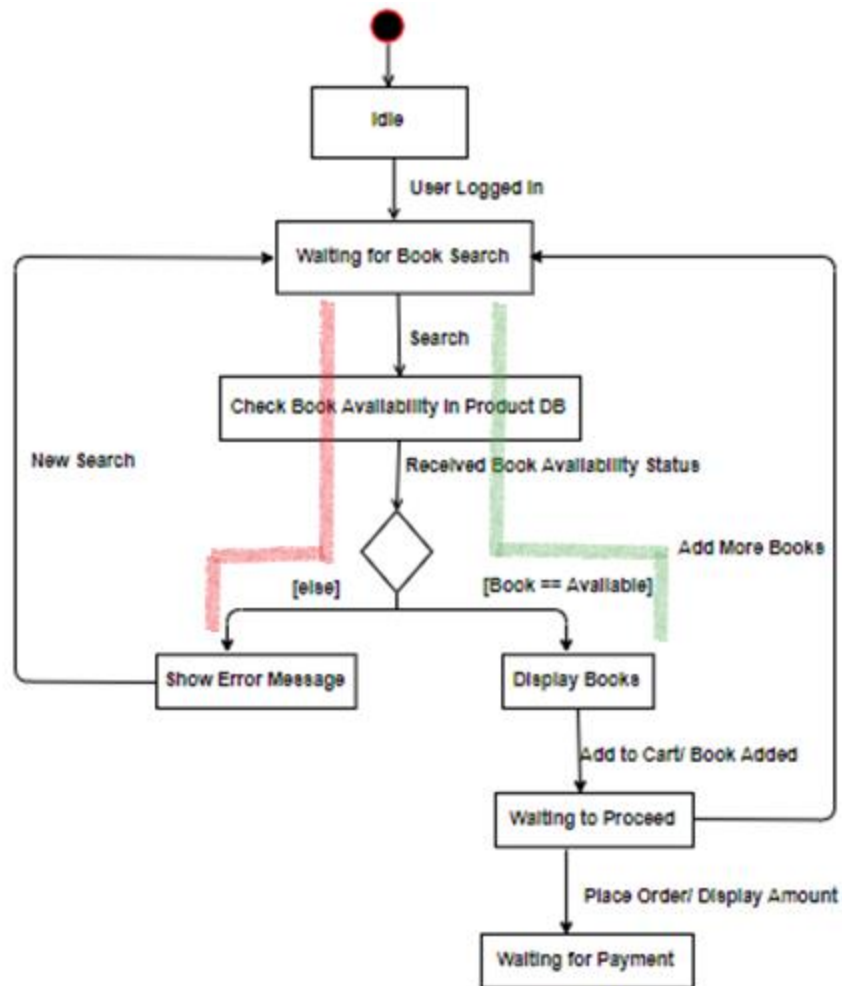
*Figure 11. State Machine Diagram of Find Books Use-Case*

The state machine diagrams illustrate different states of the system. Initially the system is in Idle state. When user logs in, the system transitions to the waiting state. Meanwhile the user searches for book and after the search the system checks the availability of book in the database. There can be two conditions, either the book is available, or book is unavailable. If the book is available, the system transitions to the “Display Books” state. Otherwise, it moves to state “Show Error Message” and after a new search again goes back to waiting state. If user adds the book to the Cart, the system waits for the user to proceed and complete the payment. The system remains in “waiting for payment” state until user selects a payment method. It can be observed that a **clear consistency is maintained between the sequence diagrams and the state machine diagrams** of the use cases.



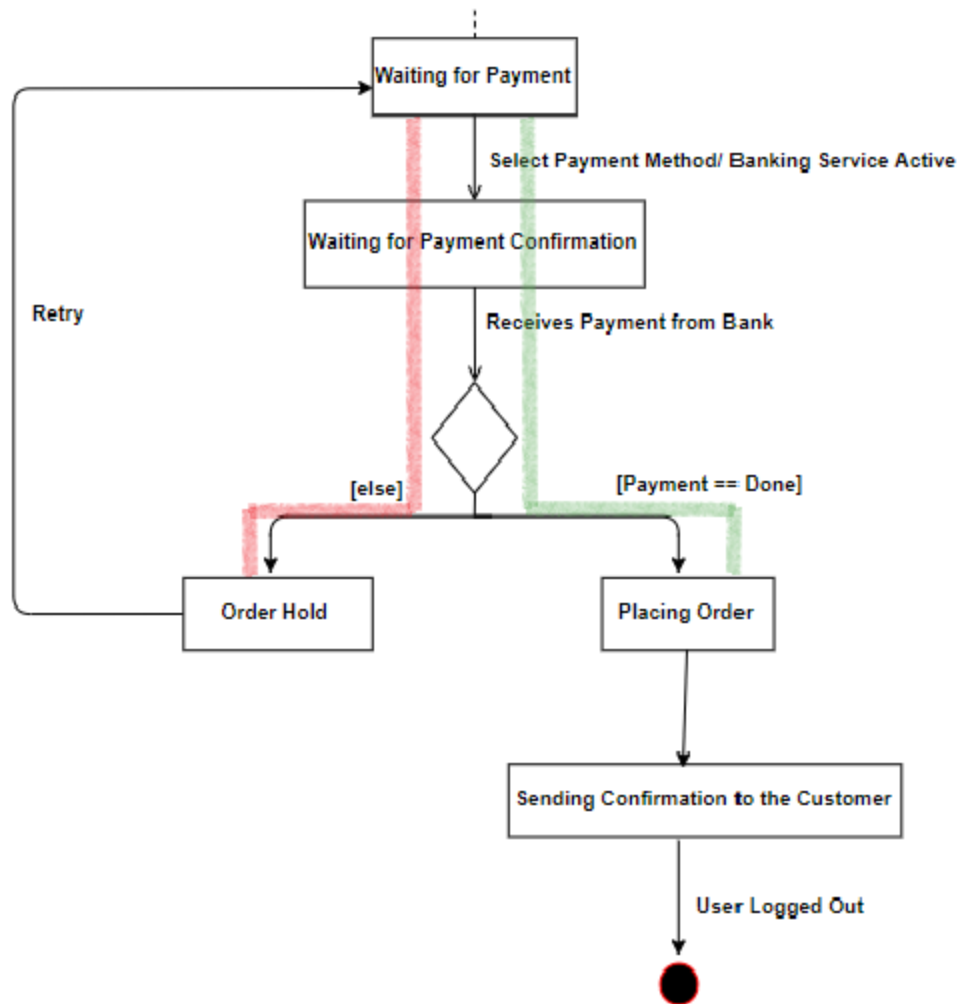
*Figure 12. State Machine Diagram of Payment Use-Case*

Once a payment method is selected, the system waits for the payment confirmation by the bank. If bank gives the payment confirmation, the system transitions to “Placing order” state and then “Sending confirmation to the customer”. On the other hand, if payment is not received by the bank the system transitions to “Order Hold” status. If the user tries to retry the payment procedure, the system again jumps back to the “Waiting for Payment” state.



*Figure 13. State Machine Diagram of Find Books Use-Case – Happy and Unhappy path*

Figure 13 and 14 represents the happy and unhappy path in the state machine diagrams. The path labelled in green is the happy path ,whereas, path labelled in red is the unhappy path.



***Figure 14. State Machine Diagram of Payment Use-Case – Happy and Unhappy path***

Figure 15 shows the state machine diagram of the integration of both the use cases – Find Books and Payment. The sky-blue colored section holds the portion of Find Books state machine diagram whereas the section marked with grey color represents the states of the system during Payment use case.



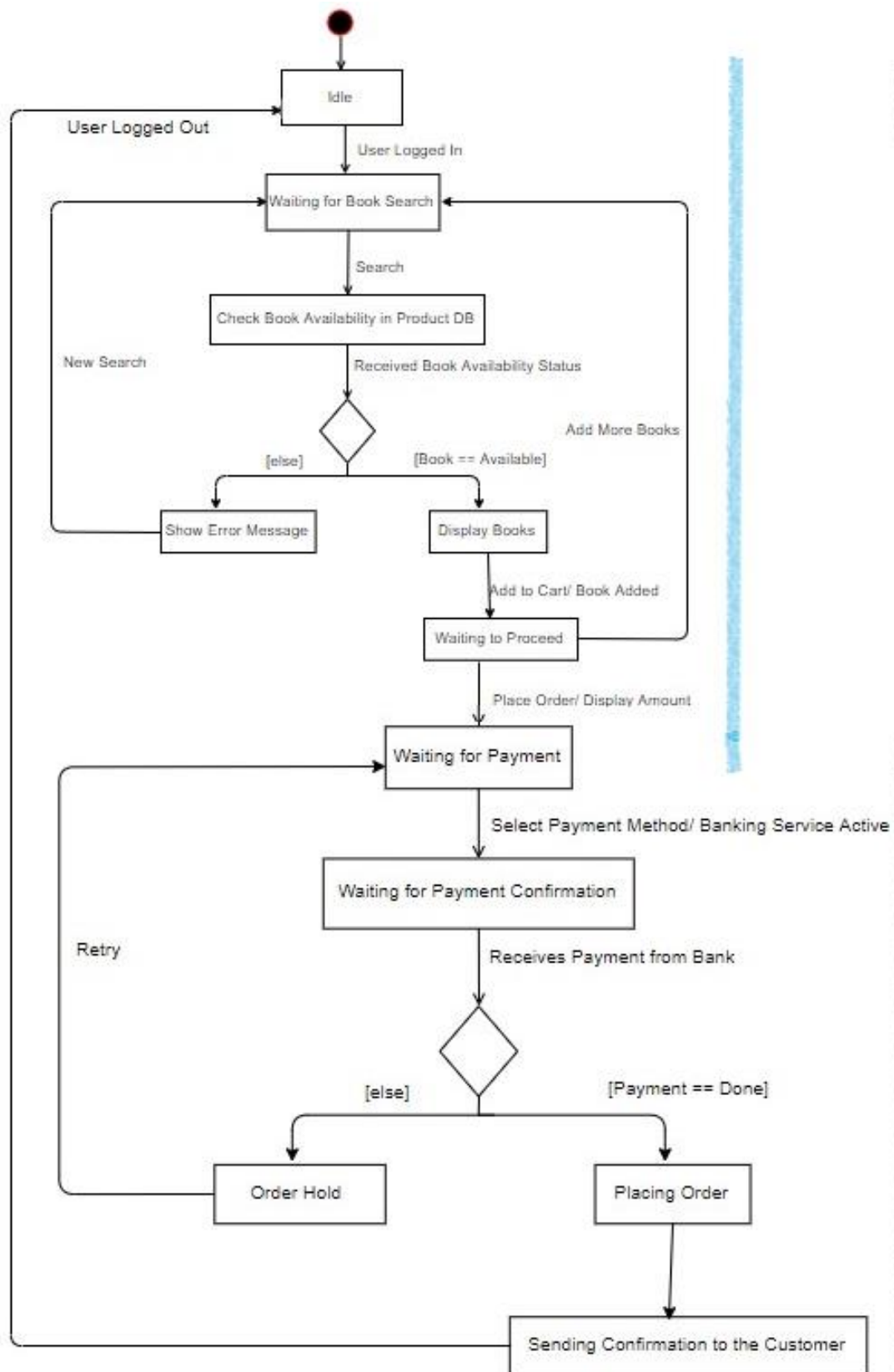
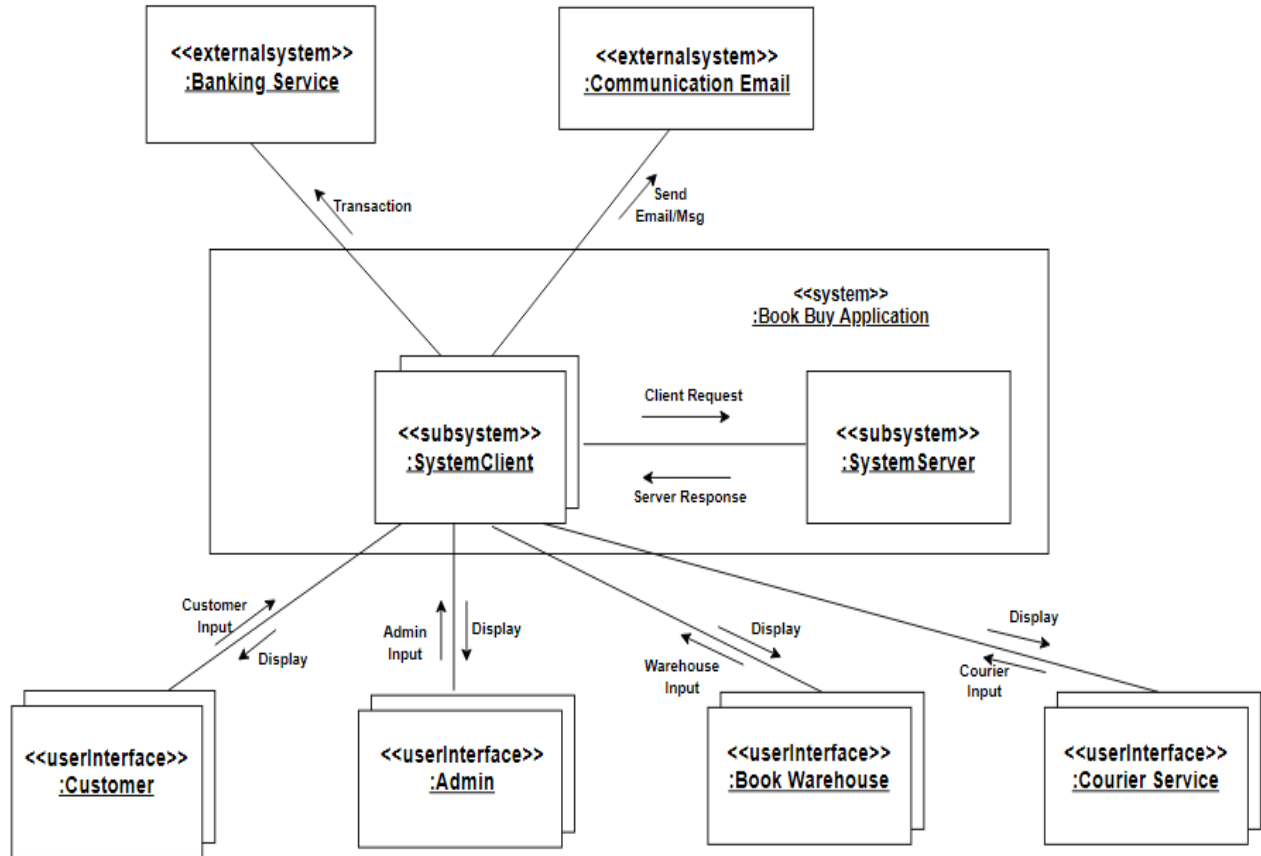


Figure 15. Integration: Find Books and Payment

## V. DESIGN MODEL

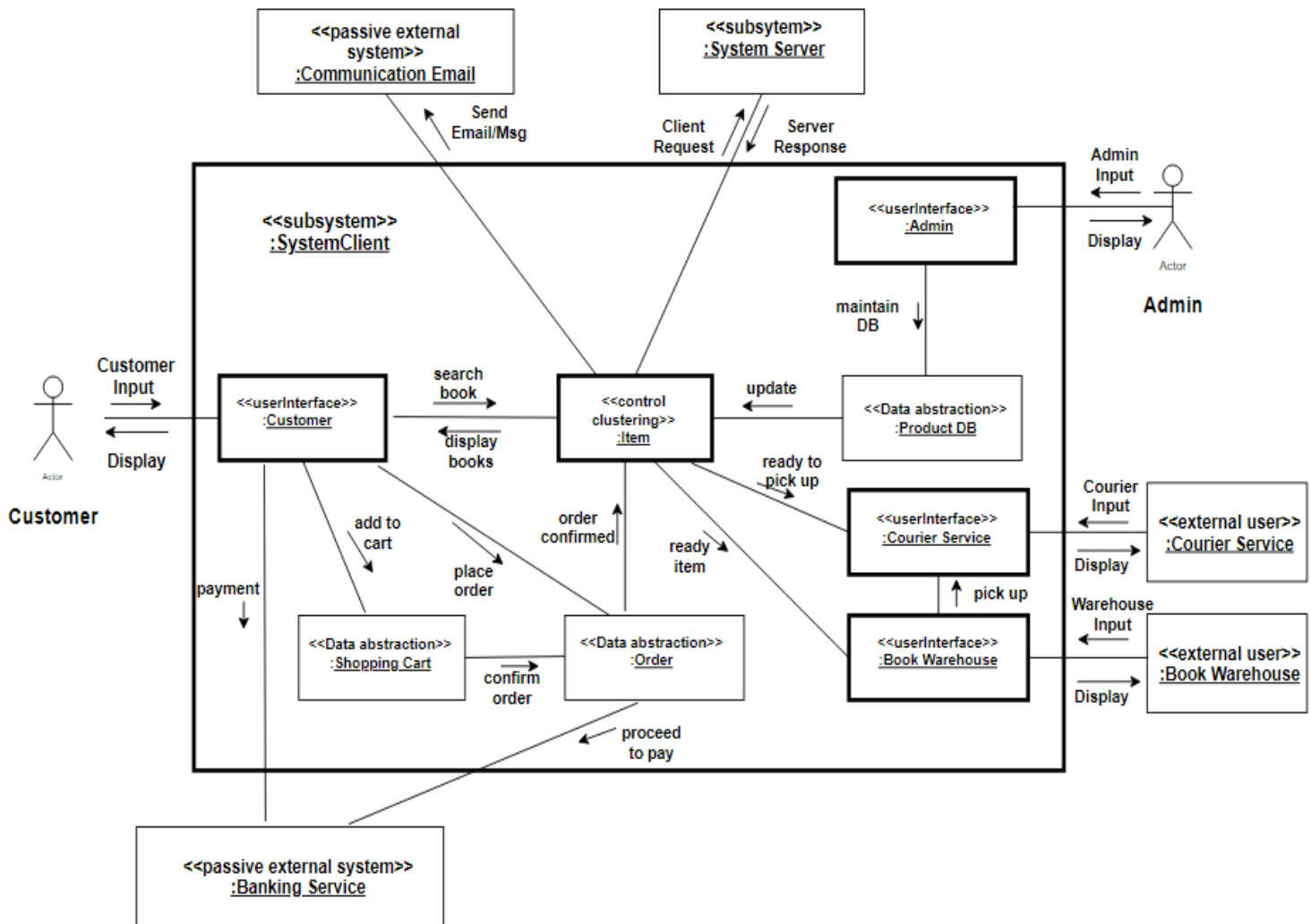
### 1. SOFTWARE ARCHITECTURE



*Figure 16. Overall Software Architecture*

Figure 15 is the Software Architecture diagram. The application has been decomposed into subsystems and various interfaces are defined between them. Clients and servers are allocated to different subsystems. The client subsystem requests the server subsystem and the server subsystem responds accordingly. The client subsystem is further connected to; Banking service for transactions, Customer/Admin for input and corresponding output, Book Warehouse for warehouse input and display, Courier Service for its input and corresponding display.

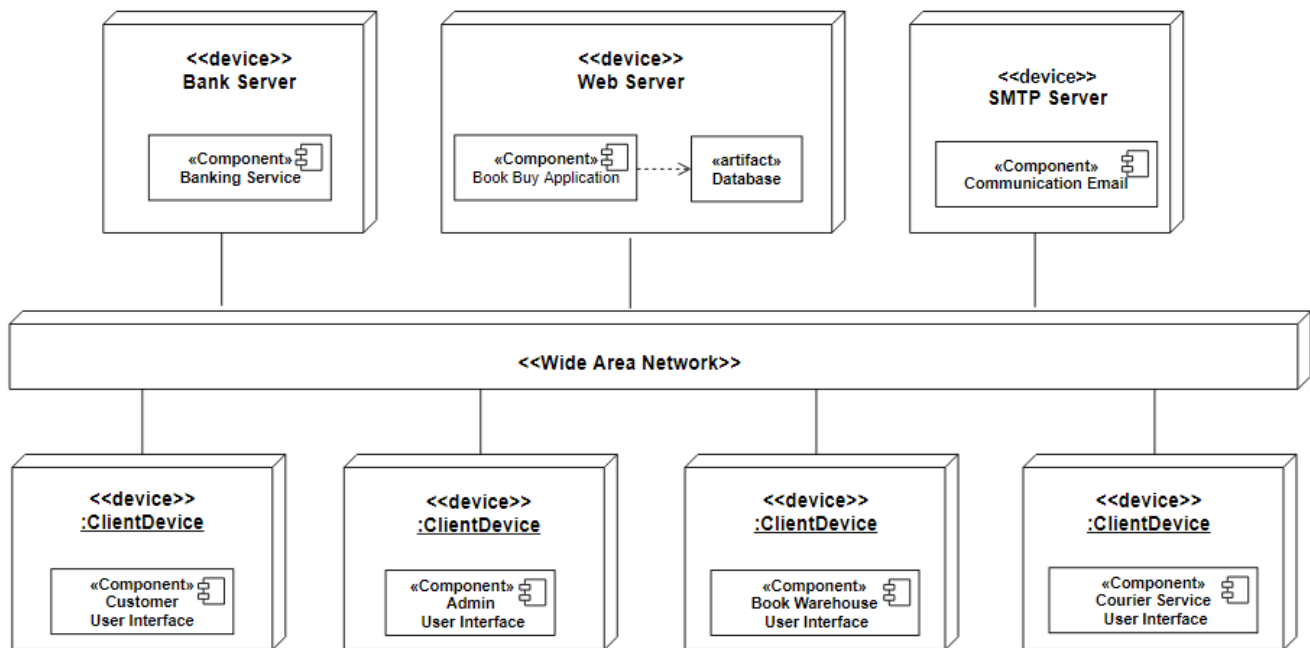
## 2. CONCURRENT TASK ARCHITECTURE



*Figure 17. Concurrent Task Architecture*

Figure 16 is concurrent task architecture where concurrent tasks are designed. This is a lower-level structure diagram for each subsystem that shows their internal structure. A high-level structure diagram was illustrated in Figure 15. The Customer inputs to the user interface and receives display as an output from the same. Similar behavior can be seen for the admin. Item class interacts with external system (communication Email) for sending emails. Client server relationship is observed between Item class and subsystem “System Server”. Similarly, the interactions among different classes and between classes and external systems can be observed in Figure 16.

### 3. DEPLOYMENT DIAGRAM



*Figure 18. Deployment Diagram – Distrbuted Software Architecture*

Figure 17 represents the deployment diagram of the system. Deployment diagram shows the existing hardware components, the software that run on each node and how different systems are connected. It can be seen from the above diagram that different servers that are required for functioning of the system are: Bank Server, Web Server and SMTP Server. Bank server is required for all banking related tasks. Web server is required for the website to function properly. SMTP stands for Simple Mail Transfer Protocol. It is a communication protocol for electronic mail transfer. The clients/users are connected to user interface through the client devices. In the above diagram, it can be seen that Customer and Admin are internal users, whereas, Book Warehouse and Courier Service are the external users. All the client devices are connected to the servers through a Wide Area Network.