



# **Dharmsinh Desai University**

Faculty of Technology, Department of Computer Engineering

B. Tech. CE Semester – VI

Subject: Object Oriented Software Engineering

Project Title:

## **Online Book Store**

By:

Langalia Trupti CE065 (17CEUOS080)

Drashti Menpara CE068 (18CEUOD007)

**Guided By:**

**Prof. Prashant M. Jadav**



# **Dharmsinh Desai University**

Faculty of Technology, Department of Computer Engineering

## **CERTIFICATE**

This is to certify that Object Oriented Software Engineering Project entitled “**Online Book Store**” is the bona fide report of work carried out by

**1) Drashti Menpara (18CEUOD007)**

**2) Langalia Trupti (17CEUOS080)**

Of Department of Computer Engineering, Semester VI, academic year 2019-2020, under our supervision and guidance.

---

### **Guide**

**Prof. Prashant M. Jadav**  
Associate Professor of Department of  
Computer Engineering, Dharmsinh  
Desai University, Nadiad

### **HOD**

**Dr. C. K. Bhensdadia**  
Head of Department of Computer  
Engineering, Dharmsinh Desai  
University, Nadiad

# Table of Contents

<b>1. Abstract.....</b>	<b>4</b>
<b>2. Introduction .....</b>	<b>5</b>
2.1 Project Details: Brief Introduction .....	5
2.2 Technology and Tools Used .....	5
<b>3. Software Requirement Specification .....</b>	<b>6</b>
3.1 Scope .....	6
3.2 System Functional Requirement .....	6
<b>4. Design .....</b>	<b>9</b>
4.1 Usecase Diagrams.....	9
4.2 Class Diagrams.....	10
4.3 Sequence Diagrams.....	11
4.4 Activity diagrams.....	12
4.5 State diagrams.....	14
4.6 E-R Diagram.....	15
4.7 Data dictionary.....	16
<b>5. Implementation Details .....</b>	<b>19</b>
5.1 Modules.....	19
5.2 Functional Prototypes.....	21
<b>6. Testing .....</b>	<b>24</b>
i. Testing Method.....	24
ii. Test Cases .....	24
<b>7. Screen-shots of the System.....</b>	<b>26</b>
<b>8. Conclusion .....</b>	<b>29</b>
<b>9. Limitations Of System .....</b>	<b>30</b>
<b>10. Future Extension to the System .....</b>	<b>31</b>
<b>11. Bibliography .....</b>	<b>32</b>

---

## 1. Abstract

---

The main objective of the project is to create an online book store that allows users to search and purchase a book online based on title. The selected books are displayed in a tabular format and the user can order their books. Using this Website the user can purchase a book online instead of going out to a book store.

There are many online book stores like Powell's, Amazon which were designed using Html. We want to develop a similar website using .NET, SQL Server. This application is developed using C# programming language and .NET framework. The Master page, data sets, data grids, user controls are used to develop the Online Book store.

The Online Book Store application provides customers with online shopping through a windows web form. A customer can create, sign in to his account, place items into a shopping cart and purchase. The Administrator will have additional functionalities when compared to the common user. He can add, delete and update the book details and also confirm a placed order.

---

## 2. Introduction

---

### 2.1 Brief Introduction



# Online Book Store

The Online Book Store application provides customers with online shopping through a windows web application. A customer can, create, sign in to his account, place items into a shopping cart and purchase. The Administrator will have additional functionalities when compared to the common user. He can add, delete and update the book details and also confirm a placed order.

### 2.2 Tools/ Technologies Used

Technologies:

.Net Windows form Application  
Sql Server

Tools: Visual Studio

---

## 3. Software Requirement Specifications

---

### **3.1 Purpose**

We developed this application so that any user can Buy book from home And also can see regular updates of book which book are in new market come.

### **3.2 Scope:**

Any user can buy the Book but need to login into the system and internet connection is required. User can open website in laptop or mobile form anywhere.

### **3.3 Types of users**

1. Admin
2. End User

### **3.4 System Functional Requirement**

#### **R1.Admin functionality:**

##### **R.1.0: Login:**

Admin can login in the system and access the system functionalities.

**Input:** username and password

**Output:** Acknowledgment

##### **R.1.1: Add Book:**

Admin can add the book in the system.

**Input:** Book title, price, preview, book poster, author name, description.

**Output:** Book is added.

##### **R.1.2: Remove Book:**

Admin can remove book which is not available in store.

**Input:** book title

**Output:** Book is removed.

##### **R.1.3: Update Book:**

**Input:** Book title , prize , book name, Author and description,

**Output:** Acknowledgment

**R.1.4: Register User:**

Admin can add user I the system which control the hole system.

**Input:** username, password, address , mobile number

**Output:** Acknowledgment

**R.1.5: Block the user:**

Which user are not authorized admin can block that user.

**Input:** username and userid

**Output:** Acknowledgment

**R.1.6: View ratings and feedback:**

**Input:** Book title

**Output:** rating and feedback of that book

**R2. Buyer Functionality:****R.2.0: View Book:**

When user open the site view the book which is available.

**Input:** book name, author , title

**Output:** Acknowledgment

**R.2.1: Buy Book:**

User can buy book after login into the system

**Input:** Book name, title , author

**Output:** Payment Receipt.

**R.2.2: Remove book from Cart:**

User can remove added book form cart.

**Input:** book name, title

**Output:** Acknowledgment

**R.2.3 : Login:**

Each user can login in the system and access the system functionalities.

**Input:** username and password

**Output:** Acknowledgment

**R.2.4 : Register:**

Each user can register in the system and access the system functionalities.

**Input:** username, email, mobile number, password.

**Output:** Acknowledgment

**R.2.5: Logout:**

**Input:** userid

**Output:** Acknowledgment

**R.2.6: View ratings and feedback:**

**Input:** Book title

**Output:** rating and feedback of that book

**R.2.7: Add ratings and feedback:**

**Input:** Book title

**Output:** rating and feedback of that book is added

**R.2.8: Add to cart**

**Input:** Book title

**Output:** Added to cart

**R.2.9: Make Payment****R.2.9.1: Make payment with cash on delivery:**

**Input:** Shipping address, Contact no. , Payable amount

**Output:** Your order is confirmed

**R.2.9.2: Make payment online:**

**Input:** Bank details, shipping address, Contact no. , Payable amount

**Output:** Payment done successfully

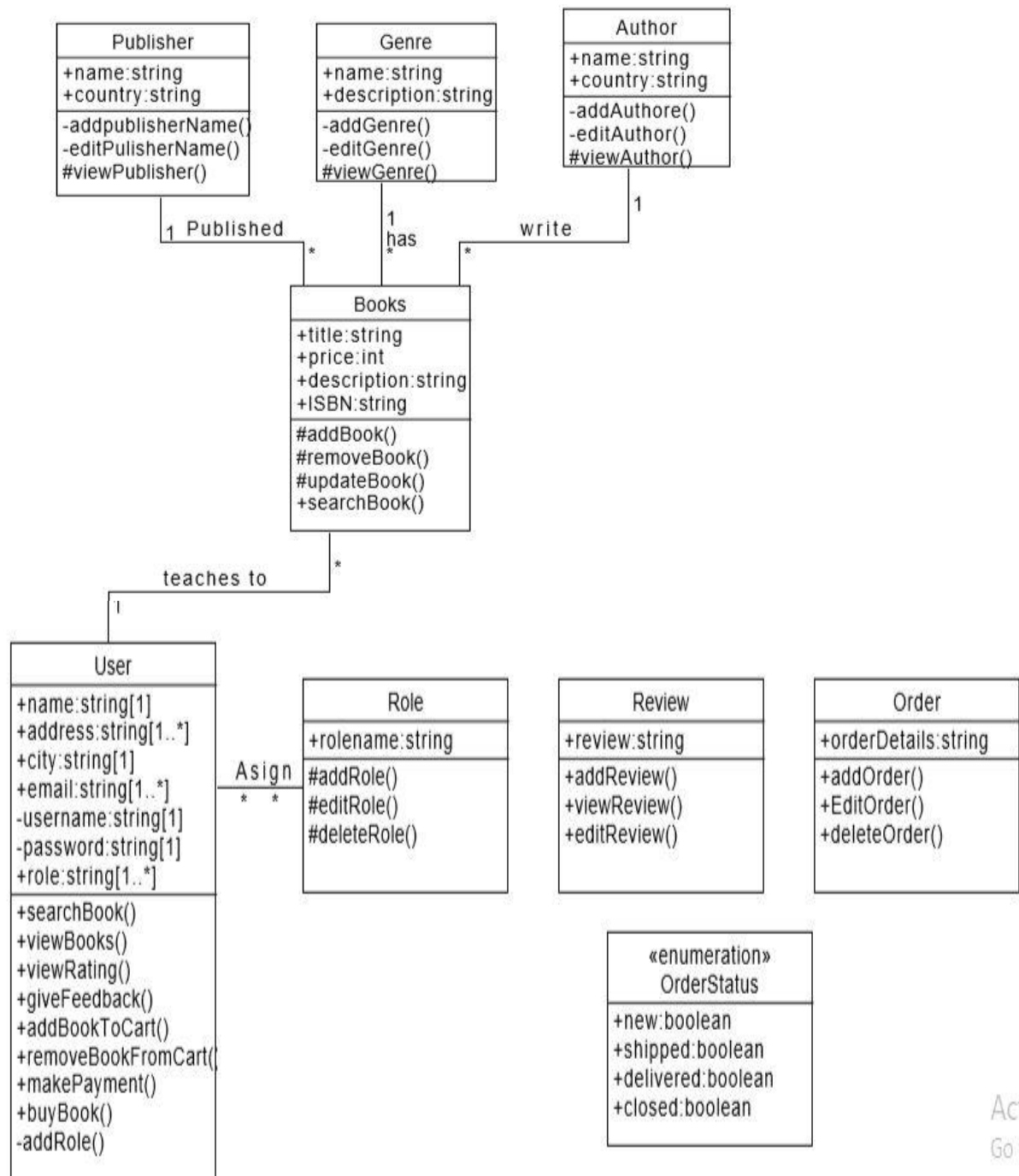


## 4. Design

### 4.1 Usecase Diagram:

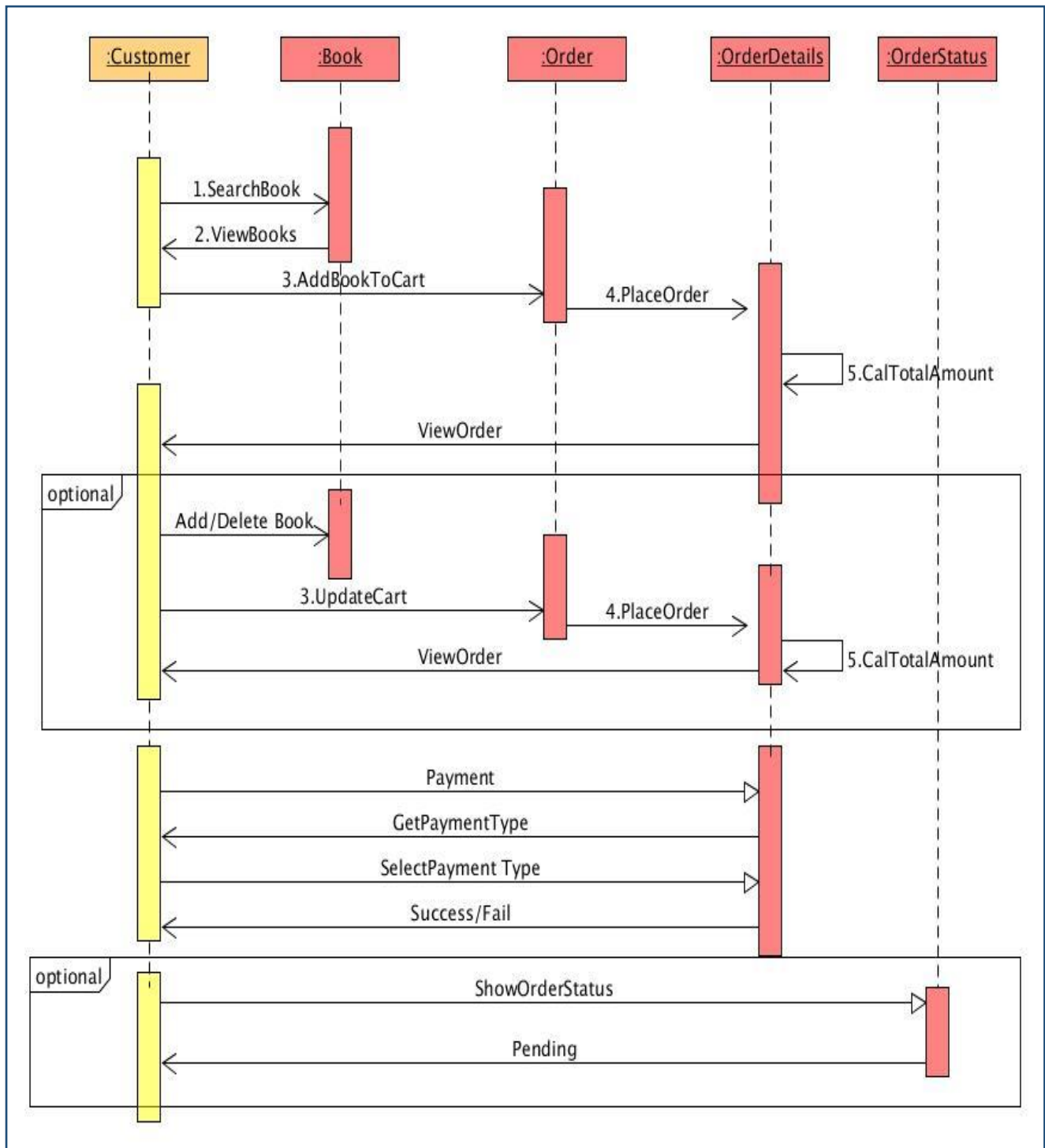


## 4.2 Class Diagram:



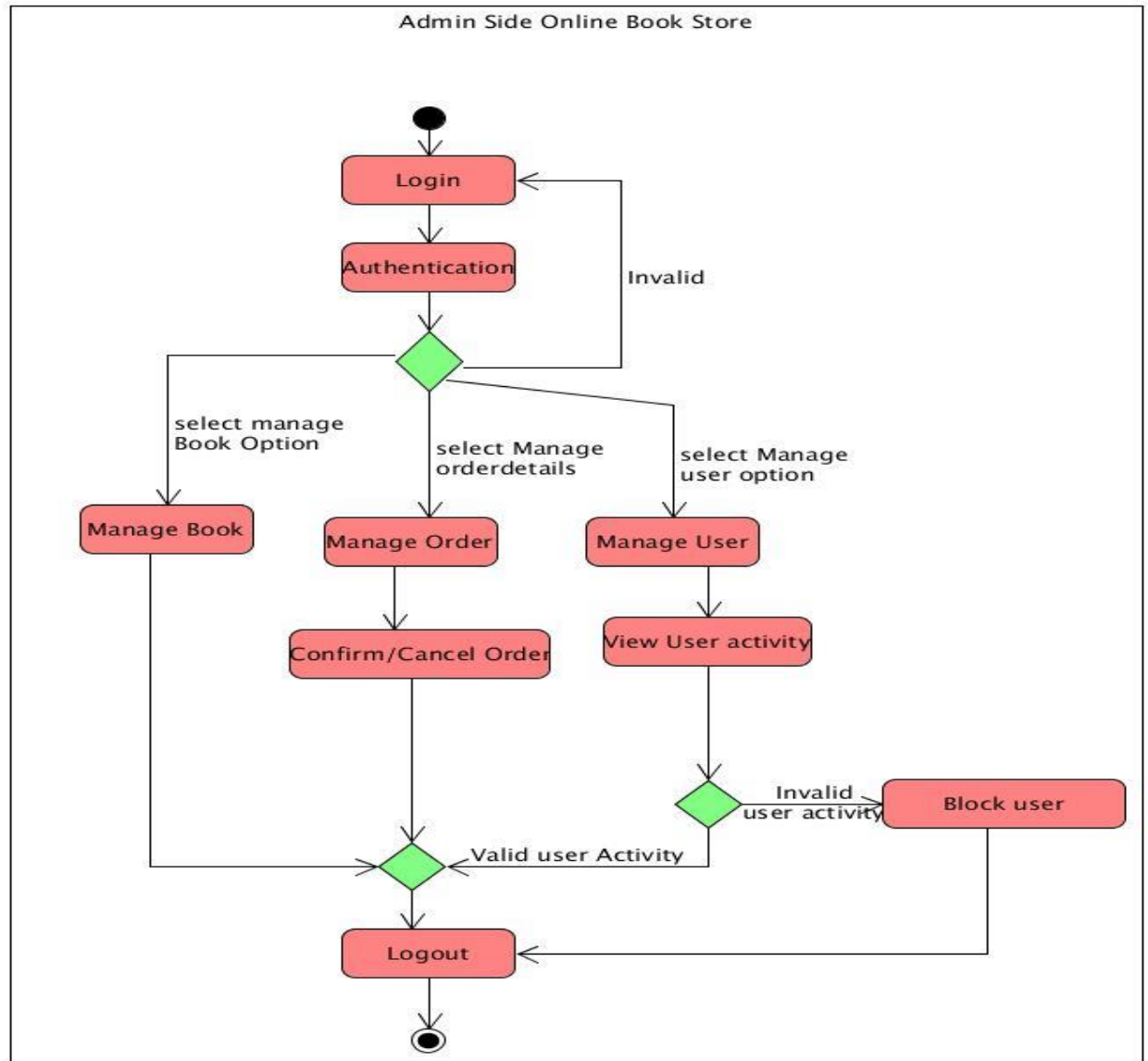
Acti  
Go to

### 4.3 Sequence Diagram:

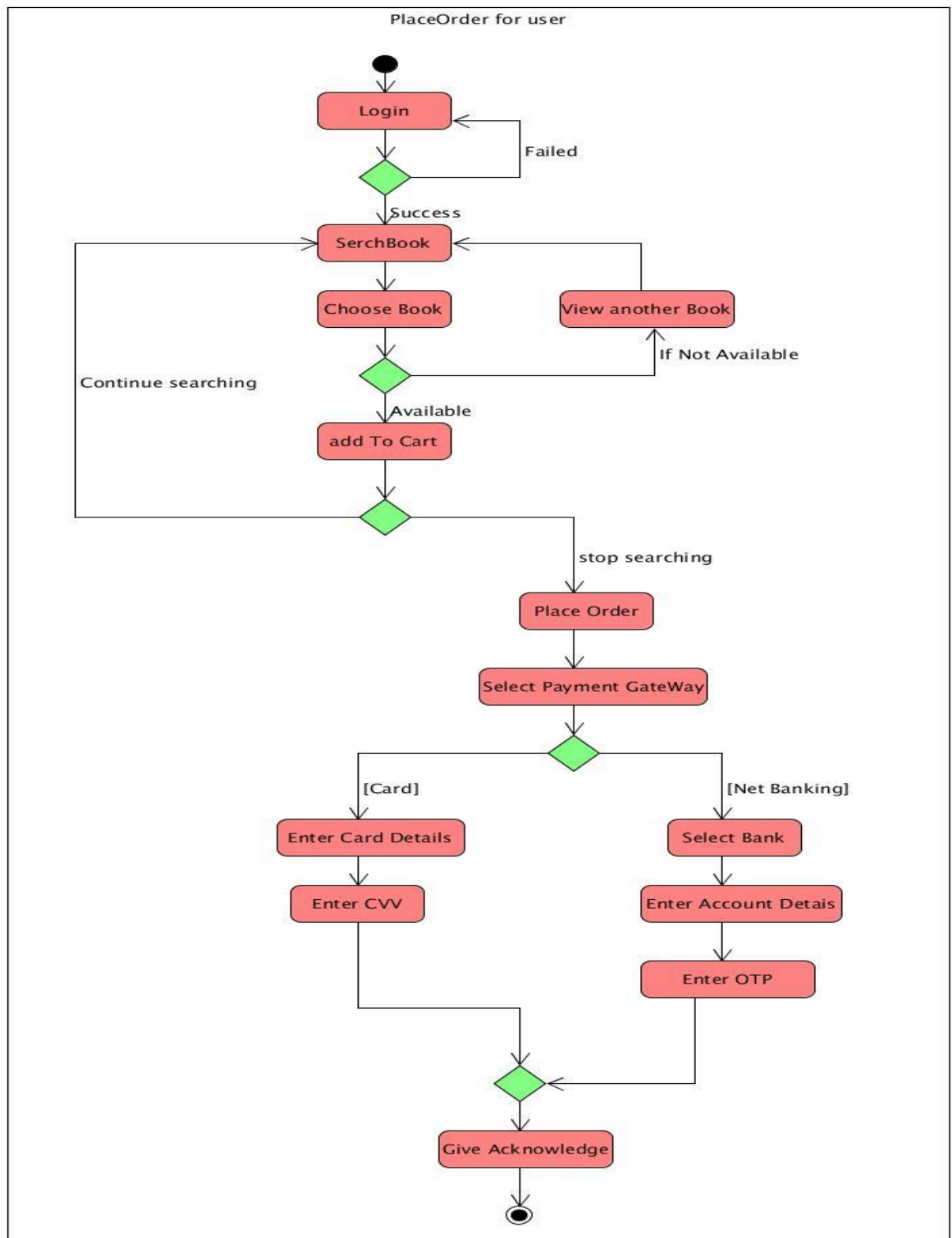


## 4.4 Activity Diagrams:

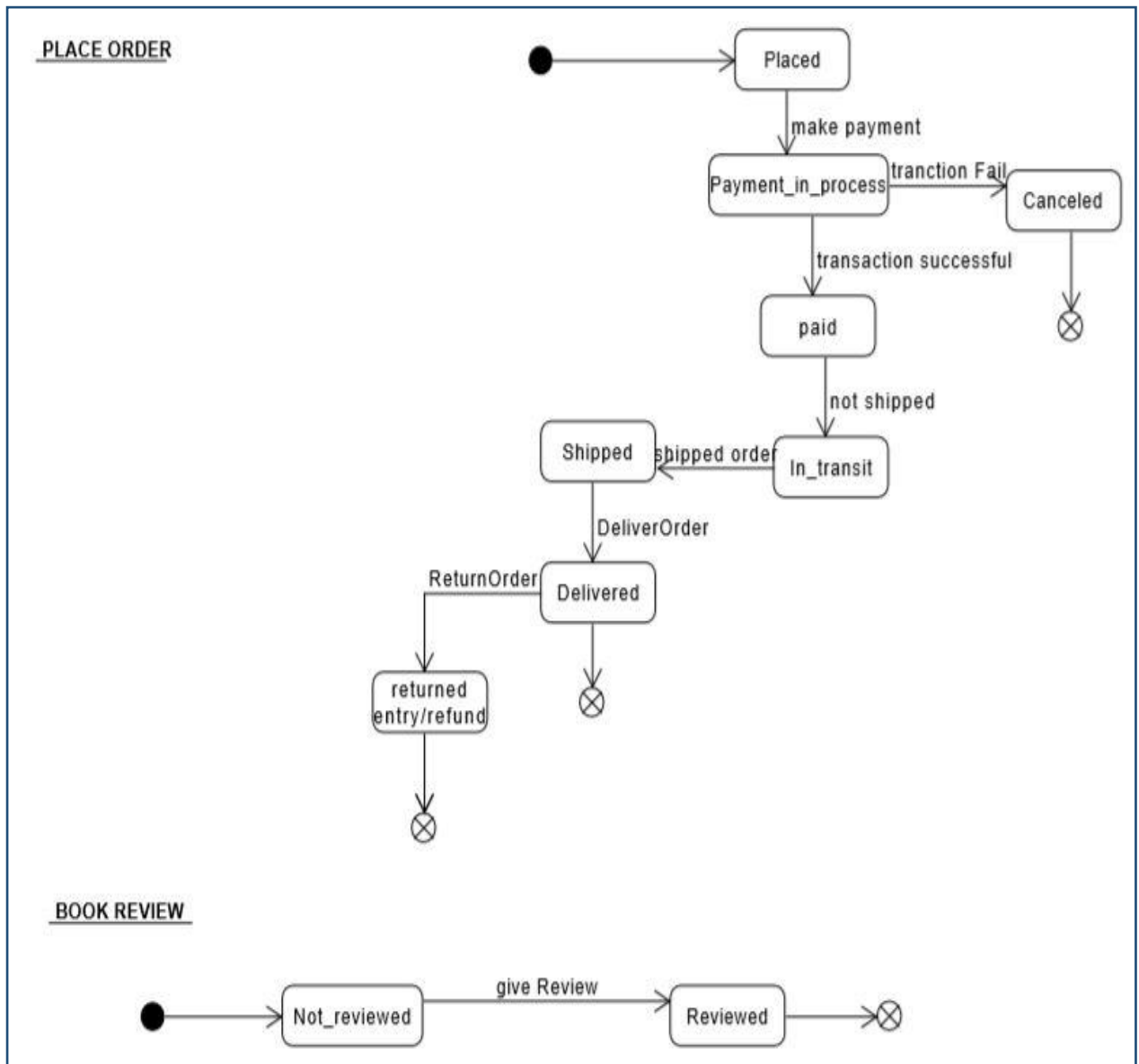
### 1. Admin side:



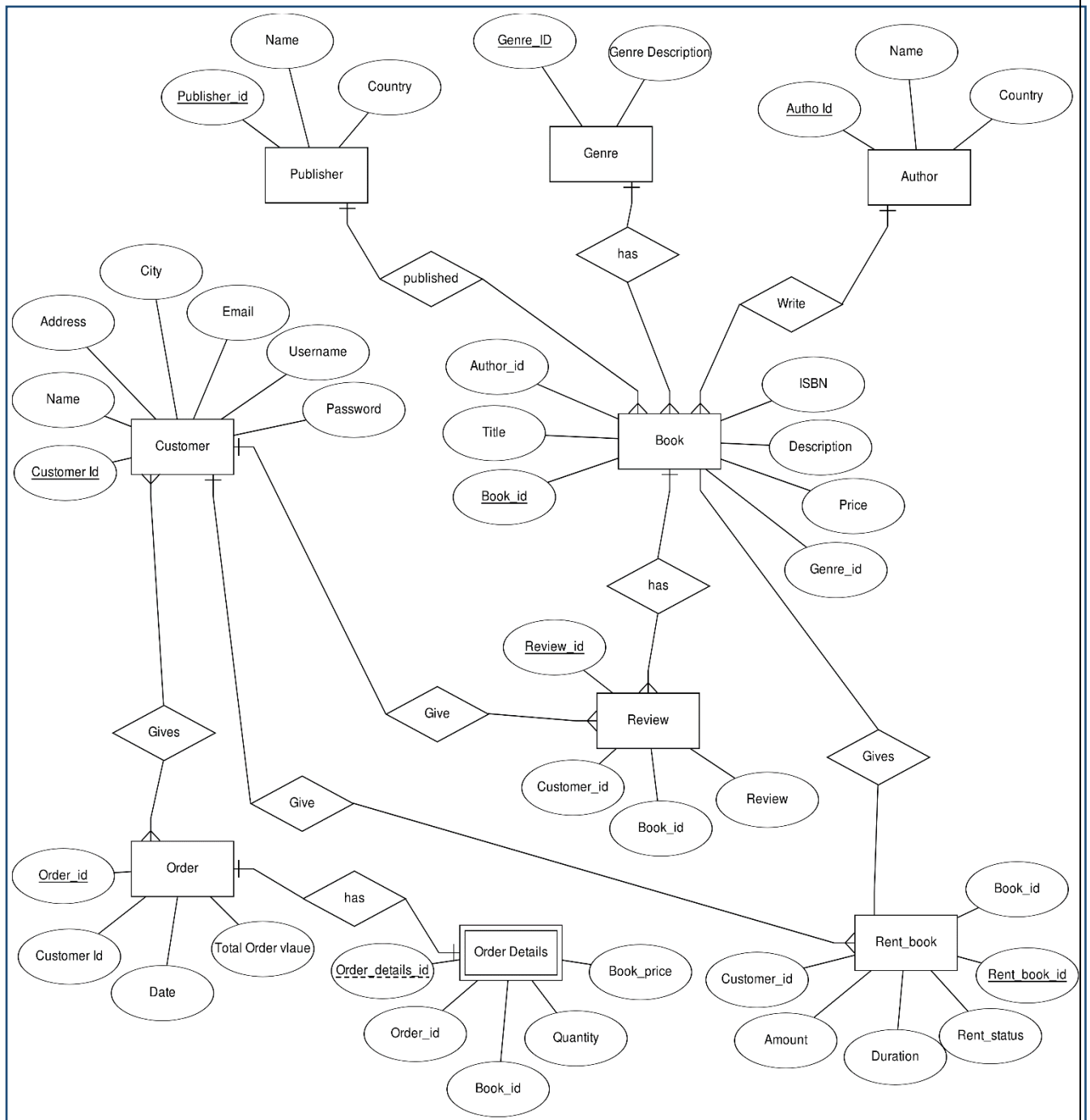
## 2.User Side:



## 4.5 State Diagram:



## 4.6 E-R Diagram:



#### 4.7 Data Dictionary:

Customer								
Sr.No.	Field Name	Data Type	Width	Required	Unique	PK/FK	Referenced Table	Description
1	Customer_id	Int	20	Yes	Yes	PK		
2	Name	Varchar	50	Yes				
3	Address	Varchar	200	Yes				
4	City	Varchar	50	Yes				
5	Email	Varchar	50	Yes				
6	Username	Varchar	50	Yes				
7	Password	Varchar	50	Yes				

Book								
Sr.No.	Field Name	Data Type	Width	Required	Unique	PK/FK	Referenced Table	Description
1	Book Id	Int	20	Yes	Yes	PK		
2	Author Id	Int	20	Yes		FK	Author	Reference with author table
3	Genre Id	Int	20	Yes	Yes	FK	Genre	Reference with genre table
4	Publisher Id	Int	20	Yes	Yes	FK	Publisher	Reference with publisher
5	Title	Varchar	50	Yes				
6	ISBN	Varchar	50	Yes				
7	Description	Varchar	50	Yes				
8	Prize	Int	30	Yes				



## Author

Sr.No.	Field Name	Data Type	Width	Required	Unique	PK/FK	Referenced Table	Description
1	Author Id	Int	20	Yes	Yes	PK		
2	Name	Varchar	50	Yes				
3	Country	Varchar	50	Yes				

## Publisher

Sr.No.	Field Name	Data Type	Width	Required	Unique	PK/FK	Referenced Table	Description
1	Publisher Id	Int	20	Yes	Yes			
2	Name	Varchar	50	Yes				
3	Country	Varchar	50	Yes				

## Genre

Sr.No.	Field Name	Data Type	Width	Required	Unique	PK/FK	Referenced Table	Description
1	Genre Id	Int	20	Yes	Yes	PK		
2	Description	Varchar	150	Yes				

## Order

Sr.No.	Field Name	Data Type	Width	Required	Unique	PK/FK	Referenced Table	Description
1	Order ID	Int	20	Yes	Yes	PK		
2	Customer ID	Int	20	Yes		FK	Customer	Reference With Customer
3	Date	Date	40	Yes				
4	Total Order Value	Int	50	Yes				

Customer_Order								
Sr.No.	Field Name	Data Type	Width	Required	Unique	PK/FK	Referenced Table	Description
1	customer_order_Id	Int	25	Yes	Yes	PK	Customer and order	Many to many with customer and order
2	customer_ID	Int	25	Yes	Yes	FK	Customer	
3	Order_ID	Int	25	Yes	Yes	FK	Order	

Order Details								
Sr.No.	Field Name	Data Type	Width	Required	Unique	PK/FK	Referenced Table	Description
1	Order details Id	Int	20	Yes	Yes	PK		
2	Order ID	Int	20	Yes		FK	Order	

---

## 5. Implementation Details

---

### 5.1 Module Description

#### ⇒ **Main User Module:**

This module gives the main home page of the system which provides basic information about the application. It Contains in page navigation to go to different Part of the pages. This module does not need authentication. It contains buttons for sign in and login.

Input : User's Selection

Output: Corresponding response

#### ⇒ **Registration Module:**

This module is used to store user's data to the database and enables the user to login to the system. All the fields in this module contains required validations. User can also navigate to login page if he/she already registered.

Input: User's Information

Output: User Registered and redirect to login page

Processing : Validate user's data and then storing them to database

#### ⇒ **Login Module:**

This module takes users credentials and then verifies it with registered users, if user is not registered the invalid credentials is shown else if they match with database then login user.

Input : User Credentials

Output: Logging user.

Processing : Verifying user credentials with the database.

### ⇒ **End User Home module:**

This module is accessed only by authenticated user. It is users home page . Users can manage their account. It provides user to see all the available books. User can add books to cart or purchase it. User can logout too.

Input: User Selection.

Output: Corresponding Response.

Processing: Validating Fields and then Updating the database

### ⇒ **Admin Home module :**

This module is accessed only by authenticated user. It is admin's home page . Admin can add, update or delete books. Admin can logout too.

Input : User Selection.

Output: Corresponding Response.

Processing : Validating Fields and then Updating the database

## 5.2 Functional Prototype :

Search and show books to user

```
2 references
public DataTable PopulateDataGridView(String search)
{
    string constr = @"Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=BookStoreDB;Integrated
    string query = "SELECT * FROM Book";
    query += " WHERE Name LIKE '%' + @ContactName + '%'";
    query += " OR @ContactName = ''";
    using (SqlConnection con = new SqlConnection(constr))
    {
        using (SqlCommand cmd = new SqlCommand(query, con))
        {
            cmd.Parameters.AddWithValue("@ContactName", search.Trim());
            using (SqlDataAdapter sda = new SqlDataAdapter(cmd))
            {
                DataTable dt = new DataTable();
                sda.Fill(dt);
                return dt;
            }
        }
    }
}
```

Add to cart

```
1 reference
public bool AddToCart(int bookid)
{
    Cart cart = new Cart();
    cart.BookId = bookid;
    cart.UserId = Global.Id;
    if(bookid <= 0)
    {
        MessageBox.Show("Please select a Book");
    }
    try
    {
        int n;
        SqlConnection con = new SqlConnection(@"Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=BookStoreDB;Integrated
        string cmdText = "Insert into [Cart] (BookId, UserId) values(@BookId, @UserId)";
        SqlCommand cmd = new SqlCommand(cmdText, con);
        cmd.Parameters.AddWithValue("@BookId", cart.BookId);
        cmd.Parameters.AddWithValue("@UserId", cart.UserId);
        using (con)
        {
            con.Open();
            n = cmd.ExecuteNonQuery();
        }
        if (n > 0)
        {
            MessageBox.Show("Added To Cart");
        }
        else
            MessageBox.Show("Updation Failed");
    }
}
```

## Buy book

```
1 reference
public void Buy()
{
    SqlConnection con = new SqlConnection(@"Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=BookStoreDB;Integrated Security=SSPI;");
    string cmdText = "Insert into [Purchases] (BookId,UserId) select [BookId], [UserId] FROM [Cart] where UserId = @UserId ";
    SqlCommand cmd = new SqlCommand(cmdText, con);
    cmd.Parameters.AddWithValue("@UserId", Global.Id);
    using (con)
    {
        con.Open();
        cmd.ExecuteNonQuery();
    }
}
```

## Insert book

```
1 reference
private void btninsert_Click(object sender, EventArgs e)
{
    string filename = System.IO.Path.GetFileName(openFileDialog1.FileName);
    if (IsValid())
    {
        SqlConnection con = new SqlConnection(@"Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=BookStoreDB;Integrated Security=SSPI;");

        string cmdText = "Insert into [Book] (Name, Price,Author,Genre,CoverPage) values('" +
        cmd = new SqlCommand(cmdText, con);
        image_upload();
        cmd.Parameters.AddWithValue("@photo", photo_array);
        con.Open();
        int n = cmd.ExecuteNonQuery();
        con.Close();
        if (n > 0)
        {
            MessageBox.Show("record inserted");
            GetRecord();
        }
        else
            MessageBox.Show("insertion failed");
    }
    Resetcontrol();
}
```

## Delete Book

1 reference

```
private void button1_Click(object sender, EventArgs e)
{
    if (bookid > 0)
    {
        string cmdText = "Delete from Book where Id=@id";
        SqlConnection con = new SqlConnection(@"Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=

        SqlCommand cmd = new SqlCommand(cmdText, con);
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.AddWithValue("@id", this.bookid);
        using (con)
        {
            con.Open();
            cmd.ExecuteNonQuery();
            MessageBox.Show("Record Deleted successfully.");
        }
        GetRecord(); Resetcontrol();
    }
    else
    {
        MessageBox.Show("select a record");
    }
}
```

---

## 6. Testing

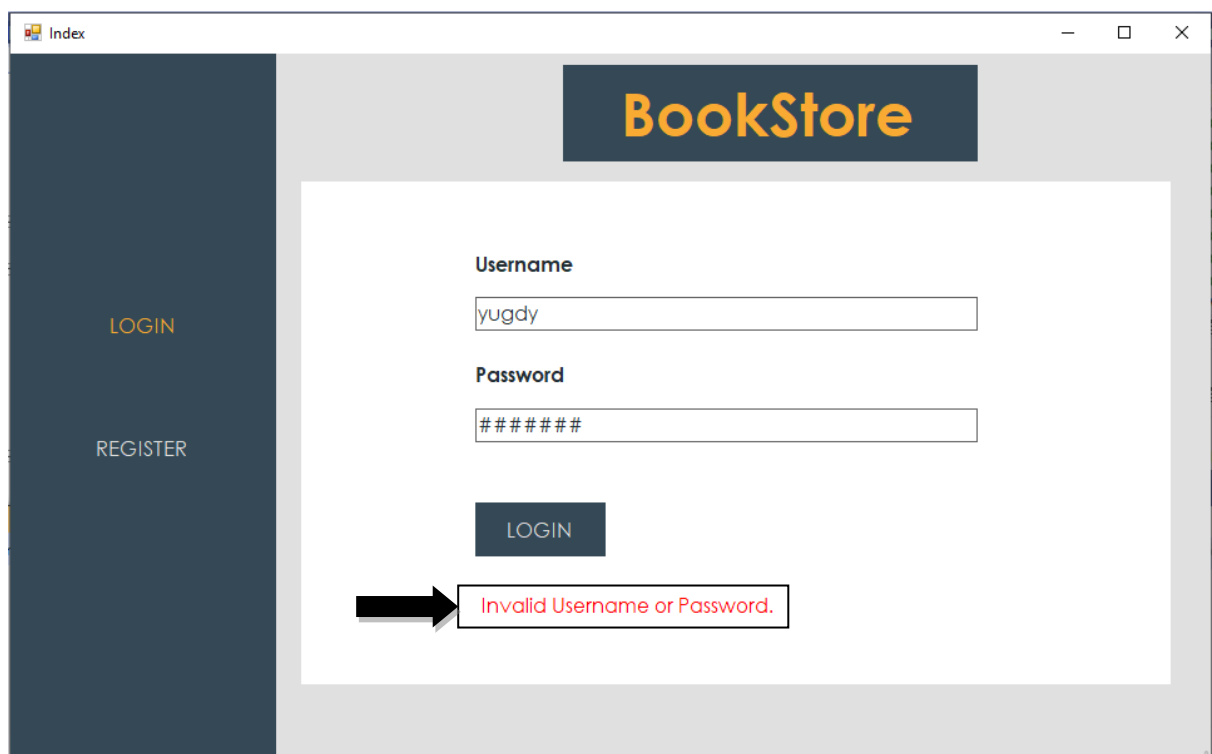
---

➤ **Testing Method** :

We have done structural testing and code based testing (White box testing) and Black box testing in our project.

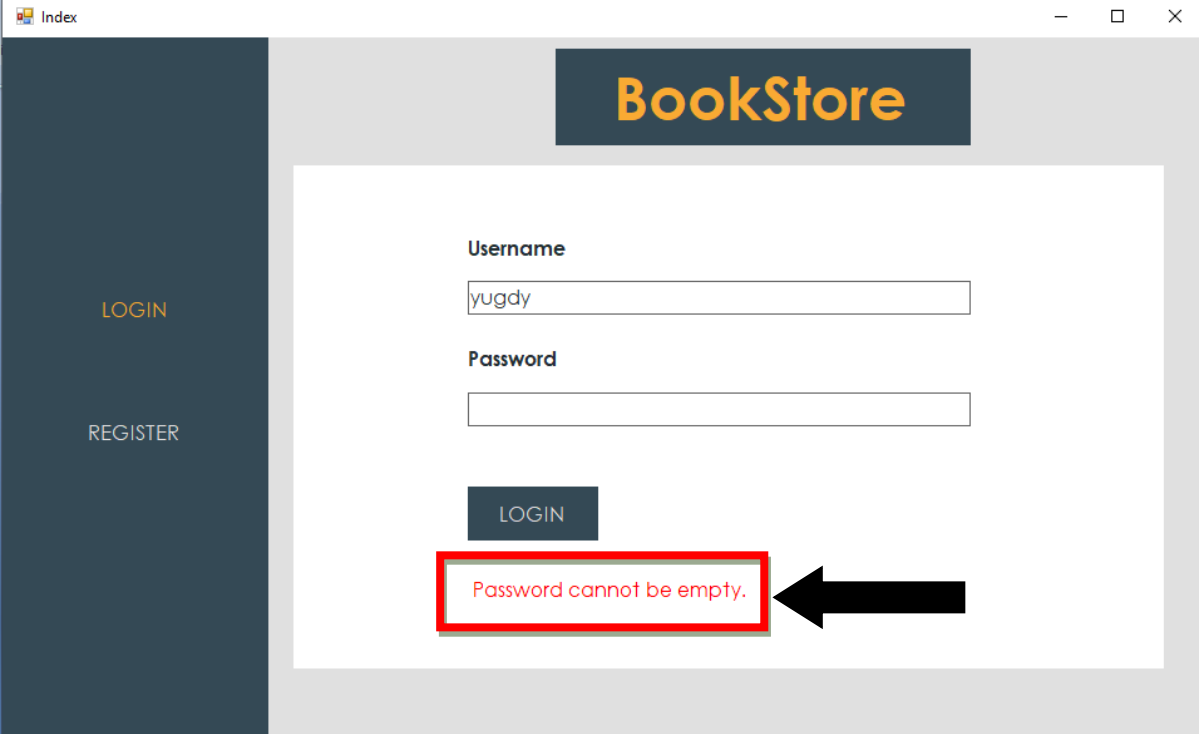
➤ **Test Cases** :

While logging in to the system if incorrect details are provided then it will not allow user to login.



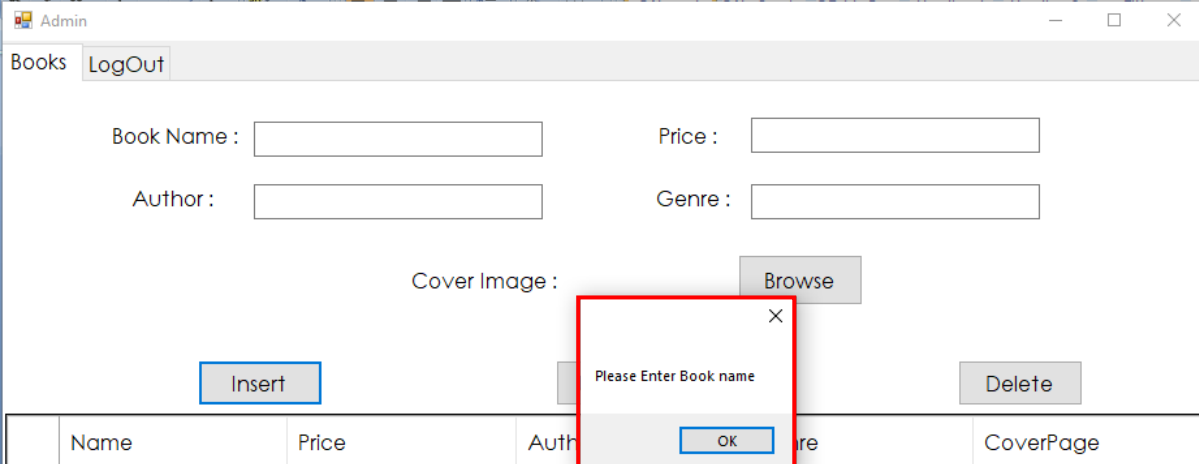


While logging in to the system if any of the fields are empty then error message will be shown.



The screenshot shows a web browser window titled "Index". The page has a dark blue sidebar on the left with "LOGIN" and "REGISTER" links. The main content area has a "BookStore" header. Below the header, there are input fields for "Username" (containing "yugdy") and "Password" (empty). A "LOGIN" button is below the password field. A red-bordered box highlights the error message "Password cannot be empty." with a black arrow pointing to it.

While adding a new book if one of the field of book details is empty then error message will be shown.

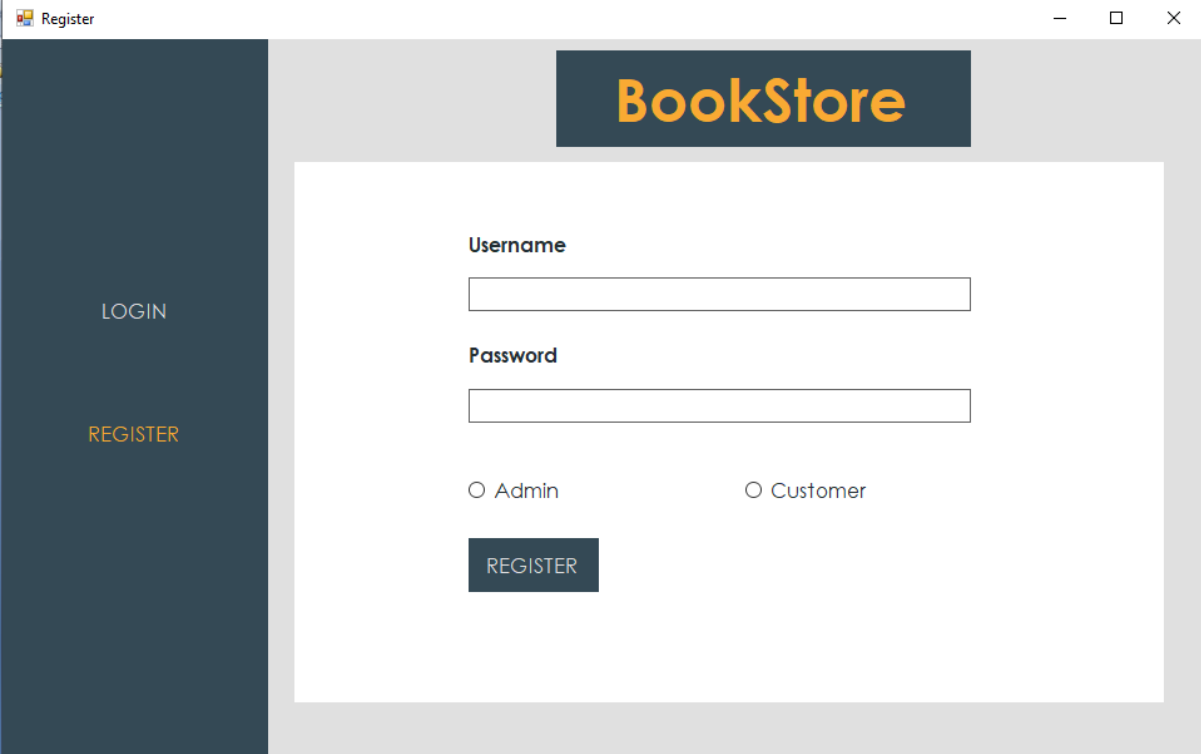


The screenshot shows a web browser window titled "Admin". The page has a "Books" tab and a "LogOut" button. Below the tab, there are input fields for "Book Name", "Price", "Author", and "Genre". There is a "Cover Image" label and a "Browse" button. Below these fields, there are "Insert", "Delete", and "OK" buttons. A red-bordered box highlights the error message "Please Enter Book name" with a black arrow pointing to it.

---

## 7. Screen-shots of the system

---



The screenshot shows a web application window titled "Register". The window has a dark blue sidebar on the left with the text "LOGIN" and "REGISTER" in white. The main content area has a light gray header with the "BookStore" logo in orange. Below the header, there is a registration form with the following elements:

- Username**: A text input field.
- Password**: A text input field.
- ☐ Admin
- ☐ Customer
- REGISTER**: A dark blue button with white text.

CustomerHome

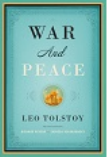

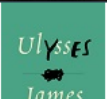
View Books

Cart

History

LogOut

Search:

	Name	Price	Author	Genre	CoverPage
	War and Peace	270	Leo Tolstoy	Historical Fiction	
	The Golden Com...	150	Philip Pullman	Adventure	
	Ulysses	125	James Joyce	Literary Fiction	

Add to Cart

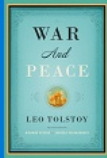

CustomerHome

View Books

Cart

History

LogOut

	Name	Price	Author	Genre	CoverPage
	War and Pea...	270	Leo Tolstoy	Historical Ficti...	
	The Kite Runner	270	Khaled Hosseini	Literary Fiction	

Total ammount : 540

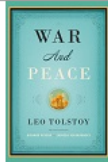

Remove Book

Buy

CustomerHome

View Books Cart History Logout

Books You Have Purchased :

	Name	Price	Author	Genre	CoverPage
	War and Peac...	270	Leo Tolstoy	Historical Ficti...	
*					

Admin


Books Logout

Book Name : The Kite Runner

Price : 270

Author : Khaled Hosseini

Genre : Literary Fiction

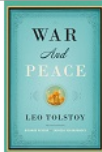


Cover Image : 

Browse

Insert

Update

Delete

	Name	Price	Author	Genre	CoverPage
	War and Peace	270	Leo Tolstoy	Historical Fiction	
	The Golden Comp...	150	Philip Pullman	Adventure	
	Ulysses	125	James Joyce	Literary Fiction	

---

## 8. Conclusion

---

The functionality implemented in the system was done after understanding all the system modules according to the requirements.

Functionality that are successfully implemented in the system are:

- Add, Update, Delete Books
- Sign Up, Login and Log Out
- Search Book
- Add book to or Remove book from cart
- Buy Book
- View History

After the Implementation and coding of the system, comprehensive testing was performed on the system to determine the loopholes and possible flaw of the system.

---

## 9. Limitation of the system

---

- We have not implemented payment method so user buys book just by clicking Buy button.
- Admin cannot block the user.

---

## **10. Future extension to the system**

---

- Actual Payment method
- Rating and feedback of book
- Better UI

---

## 11. Bibliography

---

### Websites:

- <https://www.c-sharpcorner.com/> - for understanding C#
- <https://stackoverflow.com/> - for solving errors
- <https://www.tutorialspoint.com/> - Some syntax