

**PROJECT REPORT**

# Book Store

|  |  |
| --- | --- |
| **Semester:** | Programming Fundamentals |
| **Class:** | PF1126 |
| **Group:** | TBN Group |
| **Instructor** | Nguyen Dinh Cuong |
| **Team members:** | Tran Luu Phuc Thanh  Bui Huy Binh  Phan Thi Kim Nga |

# Index

[Book Store 1](#_Toc173700635)

[Index 2](#_Toc173700636)

[I.Project introduction 5](#_Toc173700637)

[II.Analyze System Requirements 6](#_Toc173700638)

[1.Use Case 7](#_Toc173700639)

[**1.1 Use Case** 7](#_Toc173700640)

[**1.2 Login Description** 8](#_Toc173700641)

[**1.3 Create Account Description** 9](#_Toc173700642)

[**1.4 Update Account Info Description** 10](#_Toc173700643)

[**1.5 Search Book Description** 11](#_Toc173700644)

[**1.6 Create Order Description** 12](#_Toc173700645)

[**1.7 View Order Description** 13](#_Toc173700646)

[**1.8 Change User Status Description** 14](#_Toc173700647)

[**1.9 Update Book Description** 15](#_Toc173700648)

[**1.10 Add Book Description** 16](#_Toc173700649)

[2.Activity Diagram: 17](#_Toc173700650)

[**2.1 Create account** 17](#_Toc173700651)

[**2.2 Login** 18](#_Toc173700652)

[**2.3 Add Book** 19](#_Toc173700653)

[**2.4 Update Book Info** 20](#_Toc173700654)

[**2.5 Change User Status** 21](#_Toc173700655)

[**2.6 Update Account Info** 22](#_Toc173700656)

[**2.7 Create Order** 23](#_Toc173700657)

[III.Design Details 24](#_Toc173700658)

[1.UI Design 24](#_Toc173700659)

[**1.1 Main Menu** 24](#_Toc173700660)

[**1.2 Sign In** 24](#_Toc173700661)

[**1.3 Log In** 25](#_Toc173700662)

[**1.4 Manager Menu** 25](#_Toc173700663)

[**1.5 Manage Book Menu** 26](#_Toc173700664)

[**1.6 Add Book** 26](#_Toc173700665)

[**1.7 Update Book Info** 27](#_Toc173700666)

[**1.8 Change User Status** 28](#_Toc173700667)

[**1.9 User Menu** 29](#_Toc173700668)

[**1.10 Purchase Menu** 29](#_Toc173700669)

[**1.11 Create Order** 30](#_Toc173700670)

[**1.12 User Info Menu** 30](#_Toc173700671)

[**1.13 Update Account Info** 31](#_Toc173700672)

[2.Code Design (Class Diagram) 32](#_Toc173700673)

[3.Sequence Diagram 33](#_Toc173700674)

[**3.1 Login** 33](#_Toc173700675)

[**3.2 Sign In** 34](#_Toc173700676)

[**3.3 Add Book** 35](#_Toc173700677)

[**3.4 Change User Status** 36](#_Toc173700678)

[**3.5 Update Account Info** 37](#_Toc173700679)

[**3.6 Update Book Info** 38](#_Toc173700680)

[**3.7 Purchase Book** 39](#_Toc173700681)

[4.Database Design 40](#_Toc173700682)

[**a.Entity Relationship Diagram** 40](#_Toc173700683)

[**b.Database Design Details** 41](#_Toc173700684)

[IV.Test 43](#_Toc173700685)

[1. Sign In Test 43](#_Toc173700686)

[**1.1 Sign In Test 1** 43](#_Toc173700687)

[**1.2 Sign In Test 2** 43](#_Toc173700688)

[2. Login Test 44](#_Toc173700689)

[**2.1 Login Test 1** 44](#_Toc173700690)

[**2.2 Login Test 2** 44](#_Toc173700691)

[3. Book Search Test 45](#_Toc173700692)

[**3.1 Seacrh by Name Test 1** 45](#_Toc173700693)

[**3.2 Seacrh by Name Test 2** 45](#_Toc173700694)

[**3.3 Seacrh by Category Test 1** 46](#_Toc173700695)

[**3.4 Seacrh by Category Test 2** 46](#_Toc173700696)

[4. Create Order Test 47](#_Toc173700697)

[**4.1 Create Order Test 1** 47](#_Toc173700698)

[**4.2 Create Order Test 2** 47](#_Toc173700699)

[5. Update Account Info Test 48](#_Toc173700700)

[6. Add Book Test 48](#_Toc173700701)

[**6.1 Add Book Test 1** 48](#_Toc173700702)

[**6.2 Add Book Test 2** 49](#_Toc173700703)

[7. Update Book Info Test 49](#_Toc173700704)

[8. User Search Test 50](#_Toc173700705)

[**8.1 User Seacrh Test 1** 50](#_Toc173700706)

[**8.2 User Seacrh Test 2** 50](#_Toc173700707)

[9. Change User Status Test 51](#_Toc173700708)

[**9.1 Deactivate User Status Test** 51](#_Toc173700709)

[**9.2 Activate User Status Test** 51](#_Toc173700710)

[V.Task Assign 52](#_Toc173700711)

[VI.Installation Instructions 53](#_Toc173700712)

[1.Deployment Diagram 53](#_Toc173700713)

[2.Installation steps 53](#_Toc173700714)

# Project introduction

This is a simple simulation of a book selling system with some basic features including: create orders, user login, view book list, search book, view order list, view order details.

1. Proposed System

* Creating a program to Manager manage store
* Creating a program to User create order

1. The scope of the project to be applied

* Make for actor are Manager and User

1. System Name

* Store management system

1. Deployment Environment

* Windows, Mac

1. Development Tools

* Visual Studio Code
* Draw.io Diagram
* MySQL Workbench 8.0 CE

1. Customer Requirements (System features)

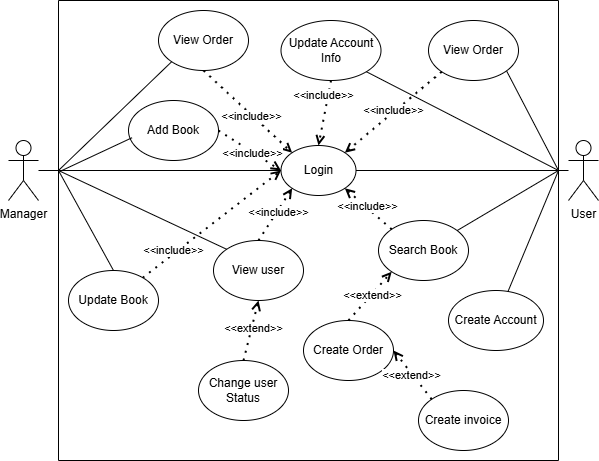
* Manager login
* Manage users
* Manage books
* Show order list
* User login
* Search books
* Create order

# Analyze System Requirements

The system is designed to facilitate the management of book inventory, sales, and customer information for a small bookstore. The system provides functionalities for the manager such as adding and updating books, managing customer information, and recording sales transactions. As for the user, the system allows them to view the books in store and create order, updating account info and view order history.

1. **Use Case**

**1.1 Use Case**



### **1.2 Login Description**

|  |  |
| --- | --- |
| Name | Login |
| ID | UC\_001 |
| Description | Used to access program and define the feature accessible depending on the type of account |
| Actor | Manager and User |
| Organizational Benefits | To define the feature and data that manager and user can access in the program |
| Triggers | Select login option from the main menu |
| Preconditions | Must enter correct username and password, the database must be connected, account must be made available |
| Postconditions | Success: Manager or User logged into the system, must display menu according to account type  Fail: Manager or User haven’t logged into the system |
| Main Course | 1. Input username and password  2. Verify login  3. Show menu |
| Alternate Courses | AC1: Enter wrong username or password  1. Display message error  2. Ask for retry  3. Re-enter request  AC2: username and password exist  1. Show menu |
| Exceptions | EX1: Input username/password  1. Display “Invalid” and ask for retry  2. Request to re-enter  EX1: Disconnect to database  1. Display error message  2. Return to main menu |

### **1.3 Create Account Description**

|  |  |
| --- | --- |
| Name | Create account |
| ID | UC\_002 |
| Description | Used to create an user account for access the system |
| Actor | User |
| Organizational Benefits | To create an user that can access the program |
| Triggers | Select sign in option from the main menu |
| Preconditions | Must enter all the info that required by the system |
| Postconditions | Success: User added into the database, return to main menu  Fail: User haven’t added into the database |
| Main Course | 1. Input username, password, full name, phone number, address  2. Verify info  3. Return to main menu |
| Alternate Courses | AC1: Username exist  1. Display message  2. Return to main menu |
| Exceptions | EX1: Input info  1. Display “Invalid” and ask for re-enter until correct  2. Request to re-enter  EX1: Disconnect to database  1. Display error message  2. Return to main menu |

### **1.4 Update Account Info Description**

|  |  |
| --- | --- |
| Name | Update account info |
| ID | UC\_003 |
| Description | Used to change user account info |
| Actor | User |
| Organizational Benefits | To change info that user want |
| Triggers | Select account info option from the user menu, then select update account info |
| Preconditions | Must be an available user |
| Postconditions | Success: Change the info selected by the user  Fail: Info remains unchanged |
| Main Course | 1. Select the info need to change  2. Enter new info  3. Verify info  4. Return to user menu |
| Alternate Courses | AC1: Select exit  1. Return to user menu |
| Exceptions | EX1: Input info  1. Display “Invalid” and ask for re-enter until correct  2. Request to re-enter  EX1: Disconnect to database  1. Display error message  2. Return to main menu |

### **1.5 Search Book Description**

|  |  |
| --- | --- |
| Name | Search Book |
| ID | UC\_004 |
| Description | Used to search for books |
| Actor | Manager and User |
| Organizational Benefits | Allow for faster access to books and other feature |
| Triggers | Logged in to the system |
| Preconditions | Must be an available account |
| Postconditions | Success: Show list of valid books  Fail: No valid book |
| Main Course | 1. Input book name/category wants to find  2. Show list found |
| Alternate Courses | AC1: Data doesn’t exist in the database  1. Display “Book not found” |
| Exceptions | EX1: Disconnect to database  1. Display error message  2. Return to menu |

### **1.6 Create Order Description**

|  |  |
| --- | --- |
| Name | Create order |
| ID | UC\_005 |
| Description | Used to create order for book |
| Actor | User |
| Organizational Benefits | Allow for create transaction of book |
| Triggers | Logged in to the system with an user account |
| Preconditions | Must have at least one book created |
| Postconditions | Success: Create order complete  Fail: No order created |
| Main Course | 1. Search book  2. Show list found  3. Select a book from list  4. Enter quantity to buy  5. Confirm order |
| Alternate Courses | AC1: Quantity entered larger than available stock or invalid number  1. Display error message  2. Request for re-enter  AC2: Quantity entered is 0  1.No order created  2. Return to menu  AC3: Exit from selection list  1. Press “ESC” key |
| Exceptions | EX1: Disconnect to database  1. Display error message  2. Return to menu |

### **1.7 View Order Description**

|  |  |
| --- | --- |
| Name | View Order |
| ID | UC\_006 |
| Description | Used to display order list |
| Actor | Manager and User |
| Organizational Benefits | Allow for access to recorded order data |
| Triggers | Logged in to the system |
| Preconditions | Must be an available account |
| Postconditions | Success: Show list of orders  Fail: No valid order |
| Main Course | 1. Trigger the function  2. Show list found |
| Alternate Courses | AC1: Data doesn’t exist in the database  1. Display nothing  2. Press any key to go back |
| Exceptions | EX1: Disconnect to database  1. Display error message  2. Return to menu |

### **1.8 Change User Status Description**

|  |  |
| --- | --- |
| Name | Change user status |
| ID | UC\_007 |
| Description | Used to deactivate or activate an user account |
| Actor | Manager |
| Organizational Benefits | Allow for control over user access to the data and feature |
| Triggers | Logged into the system with a manager account |
| Preconditions | Must have at least one user created |
| Postconditions | Success: Deactivate or activate the selected user account  Fail: No change in user status |
| Main Course | 1. Input name of user  2. Show list found  3. Select an user from the list  4. Select activate or deactivate  5. Confirm selection |
| Alternate Courses | AC1: Data doesn’t exist in the database  1. Display nothing  2. Press any key to go back  AC2: User already deactivated or activated  1. No change |
| Exceptions | EX1: Disconnect to database  1. Display error message  2. Return to menu |

### **1.9 Update Book Description**

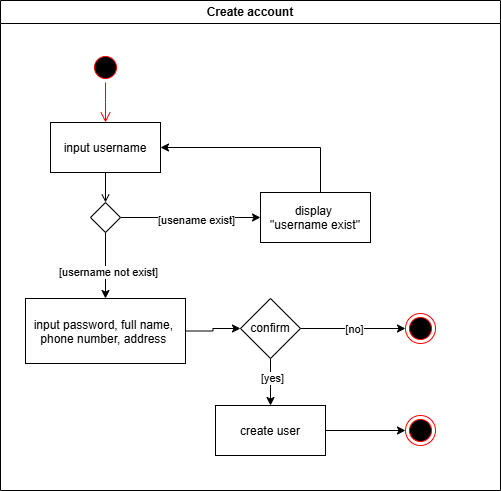
|  |  |
| --- | --- |
| Name | Update book |
| ID | UC\_008 |
| Description | Used to change book info |
| Actor | Manager |
| Organizational Benefits | To update books info |
| Triggers | Logged into the system with a manager account |
| Preconditions | Must have at least one book created |
| Postconditions | Success: Change the selected info of a book  Fail: No change in book info |
| Main Course | 1. Input name of book  2. Show list found  3. Select a book from the list  4. Select an info to change  5. Input new info  6. Confirm change |
| Alternate Courses | AC1: Data doesn’t exist in the database  1. Display nothing  2. Press any key to go back  AC2: Exit from list selection  1. Press “ESC” key to go back |
| Exceptions | EX1: Disconnect to database  1. Display error message  2. Return to menu |

### **1.10 Add Book Description**

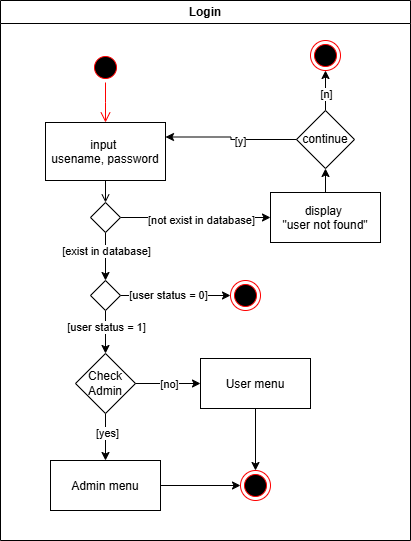
|  |  |
| --- | --- |
| Name | Add book |
| ID | UC\_009 |
| Description | Used to add new book into database |
| Actor | Manager |
| Organizational Benefits | Allow for new books to be added |
| Triggers | Logged into the system with a manager account |
| Preconditions | Must be a manager account |
| Postconditions | Success: Added a new book into database  Fail: No book added |
| Main Course | 1. Input info of book required by the system  2. Confirm |
| Alternate Courses | AC1: Data exist in the database  1. Display “book exist”  2. Press any key to go back |
| Exceptions | EX1: Disconnect to database  1. Display error message  2. Return to menu |

1. **Activity Diagram:**

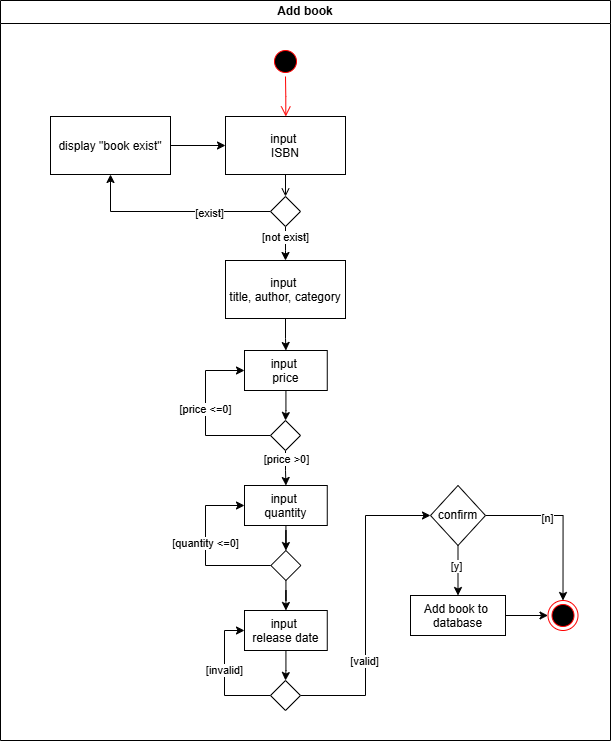
**2.1 Create account**



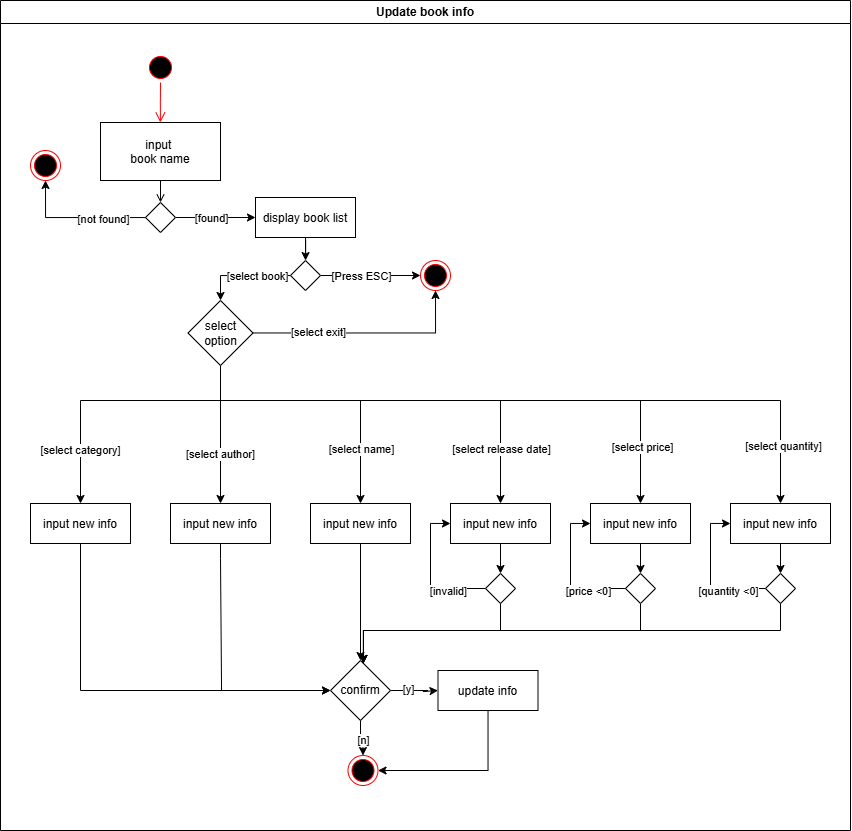
### **2.2 Login**



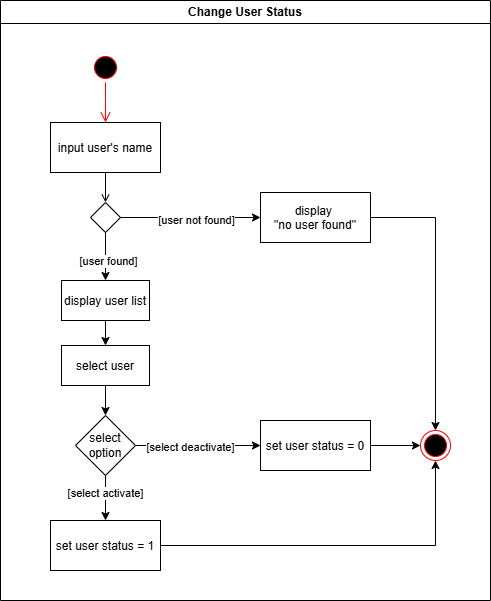
### **2.3 Add Book**



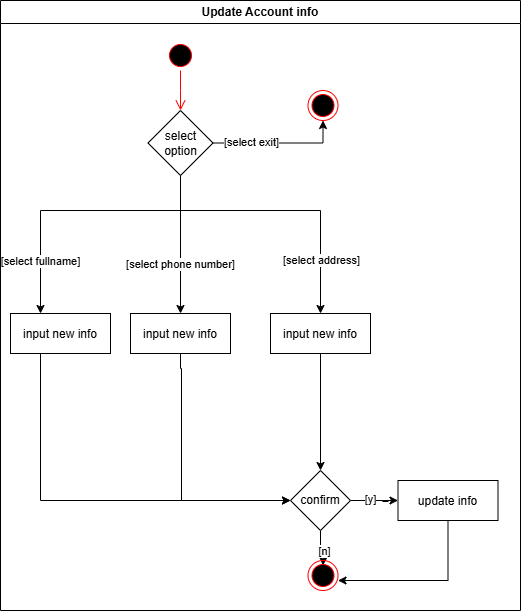
### **2.4 Update Book Info**



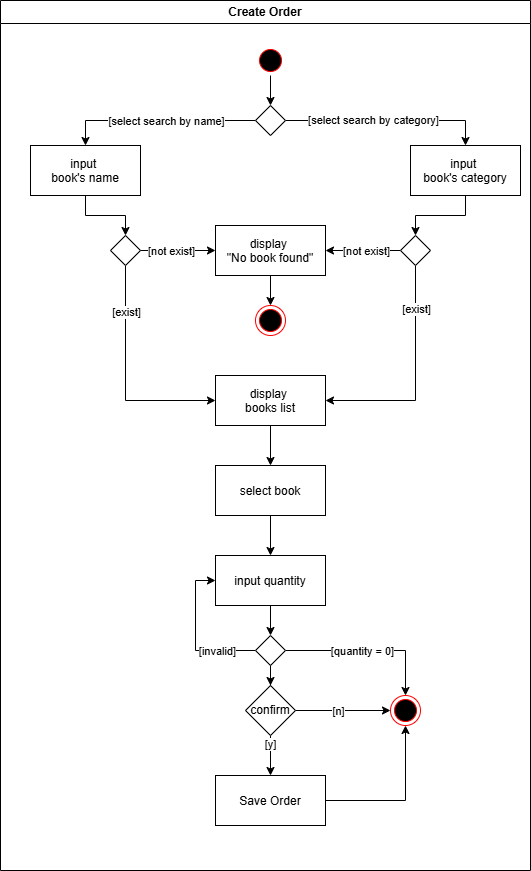
### **2.5 Change User Status**



### **2.6 Update Account Info**



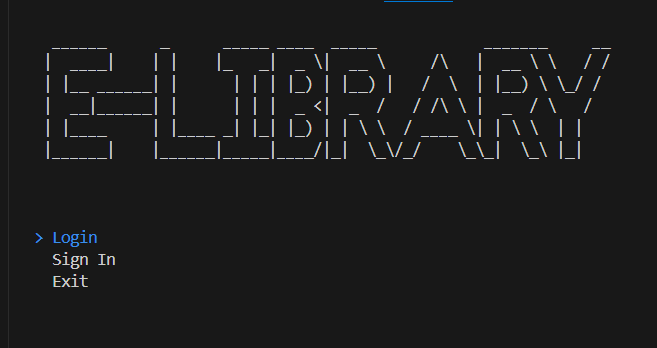
### **2.7 Create Order**



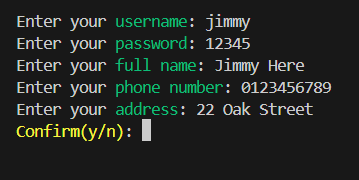
# **Design Details**

1. **UI Design**

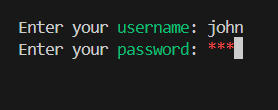
**1.1 Main Menu**



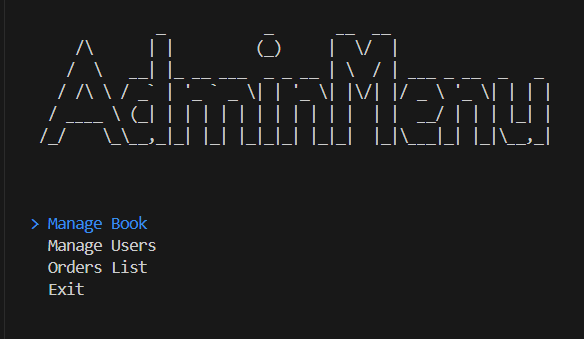
### **1.2 Sign In**



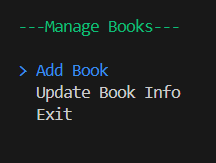
### **1.3 Log In**



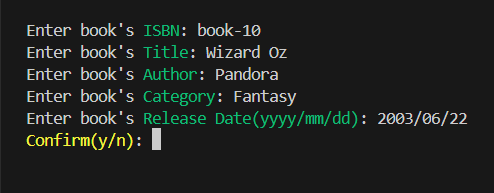
### **1.4 Manager Menu**



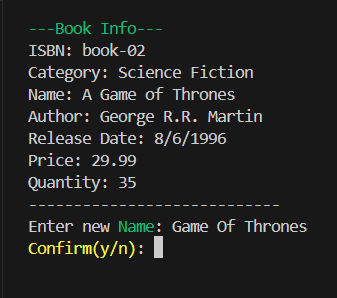
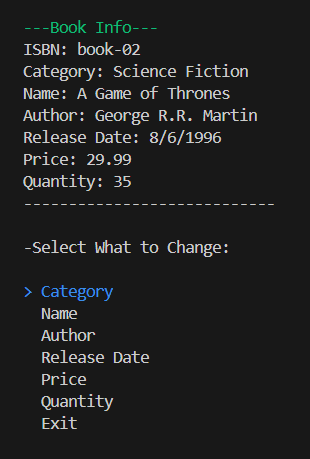
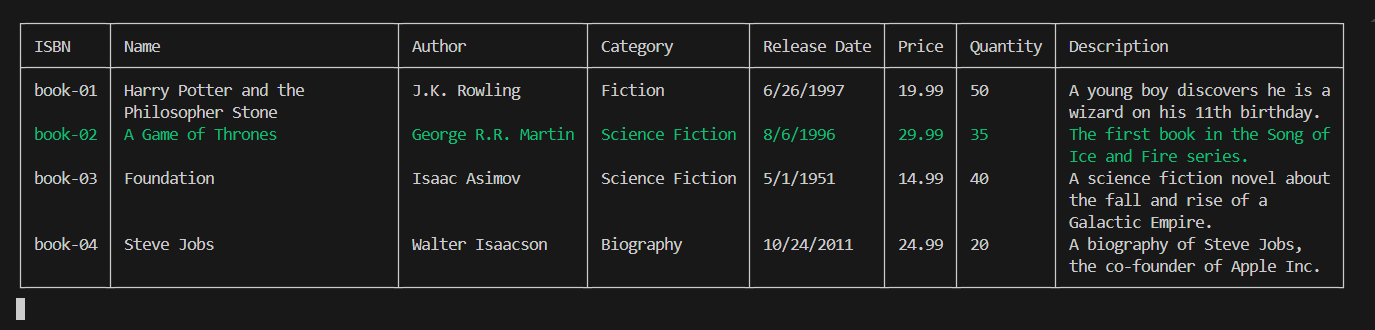
### **1.5 Manage Book Menu**



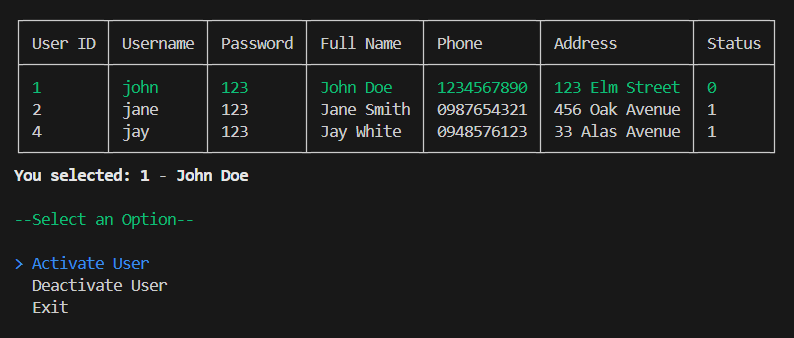
### **1.6 Add Book**



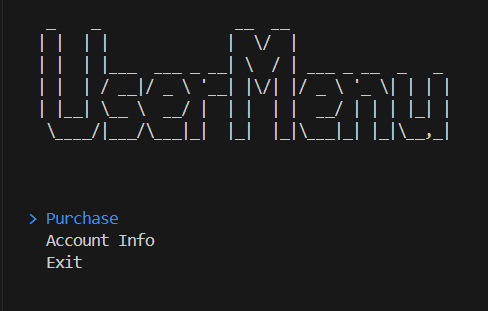
### **1.7 Update Book Info**



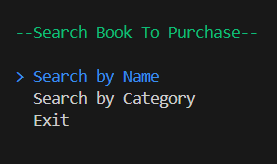
### **1.8 Change User Status**



### **1.9 User Menu**



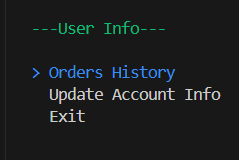
### **1.10 Purchase Menu**



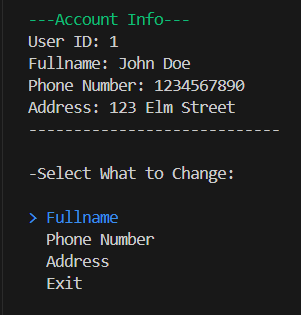
### **1.11 Create Order**



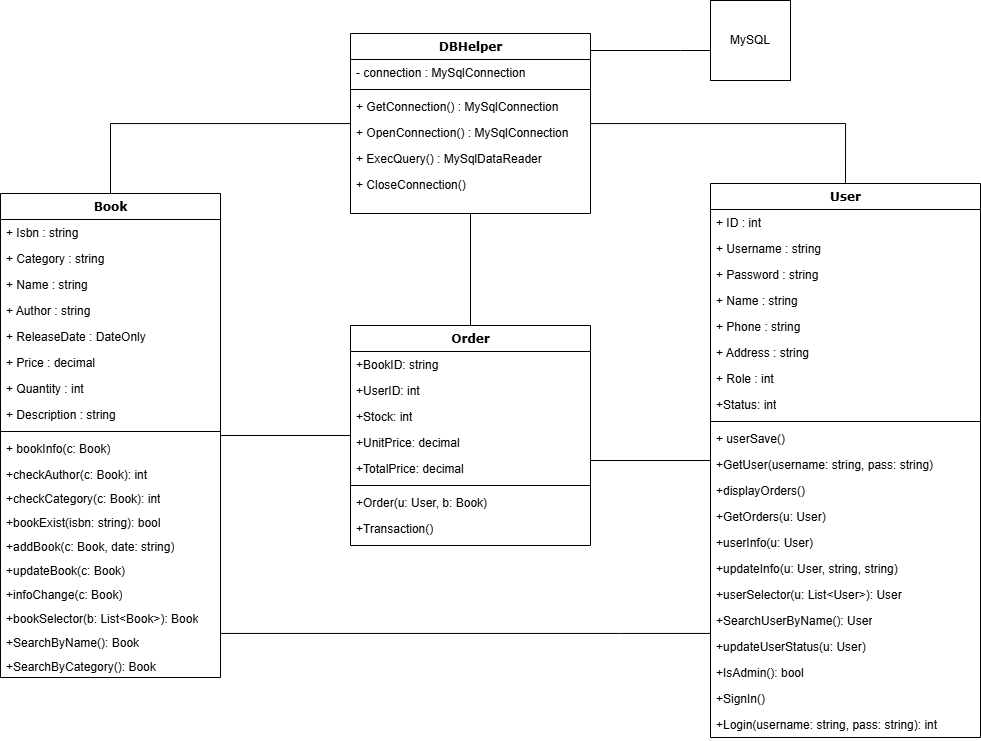
### **1.12 User Info Menu**



### **1.13 Update Account Info**

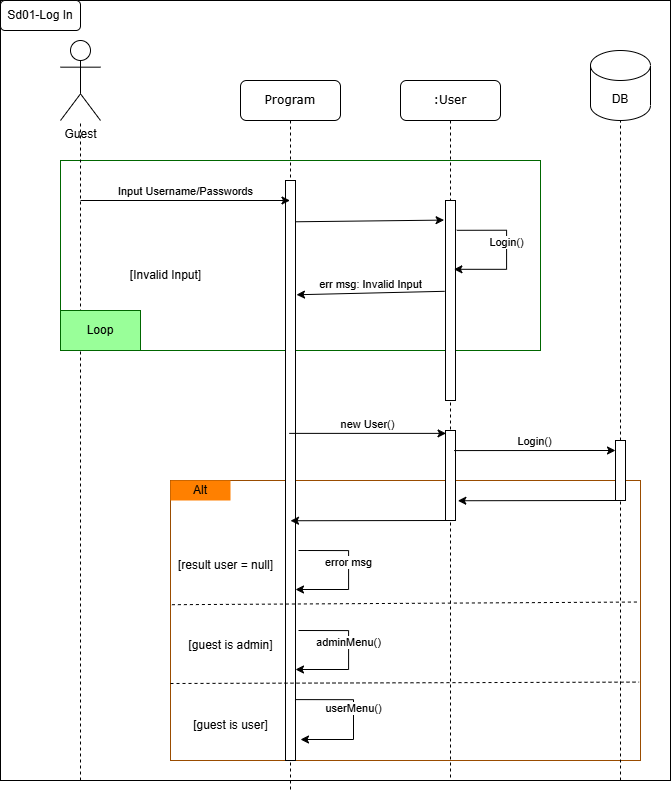


1. **Code Design (Class Diagram)**

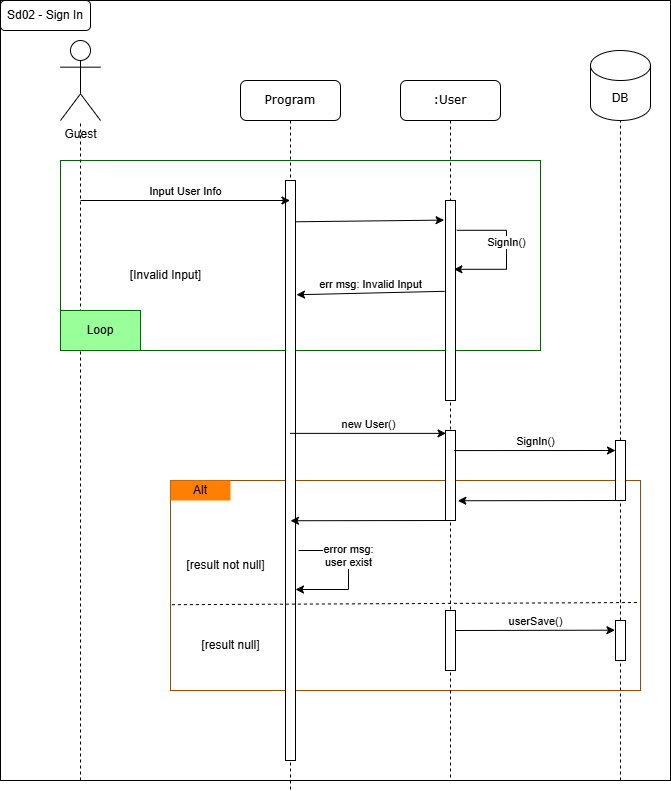


1. **Sequence Diagram**

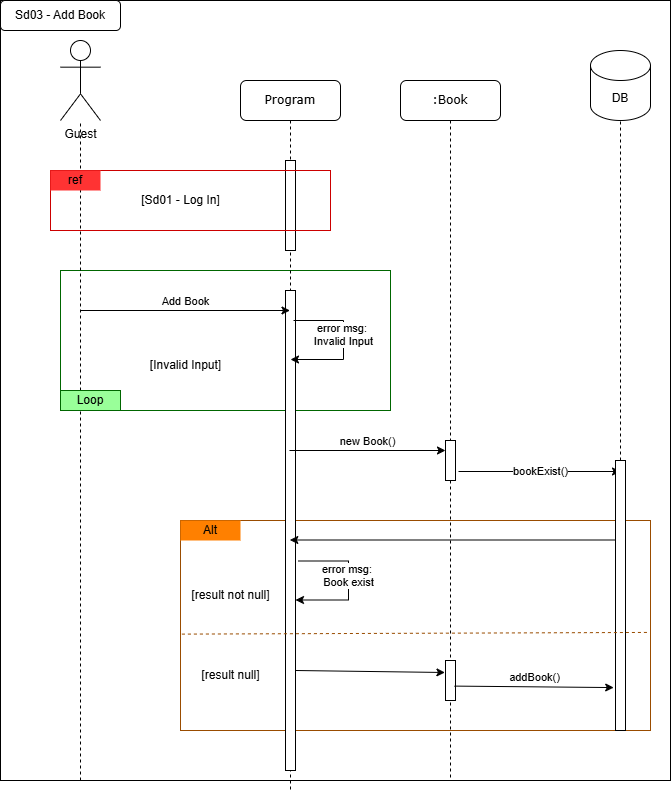
### **3.1 Login**



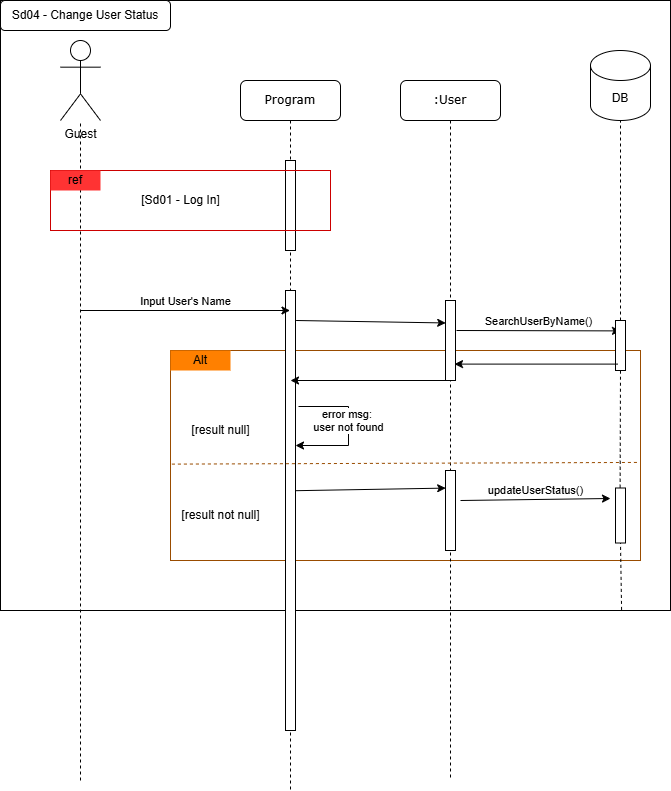
### **3.2 Sign In**



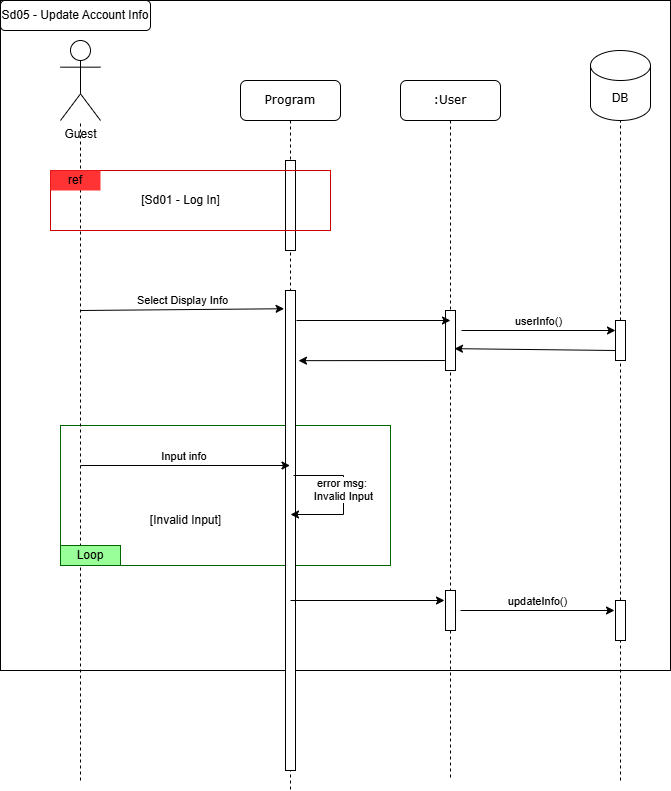
### **3.3 Add Book**



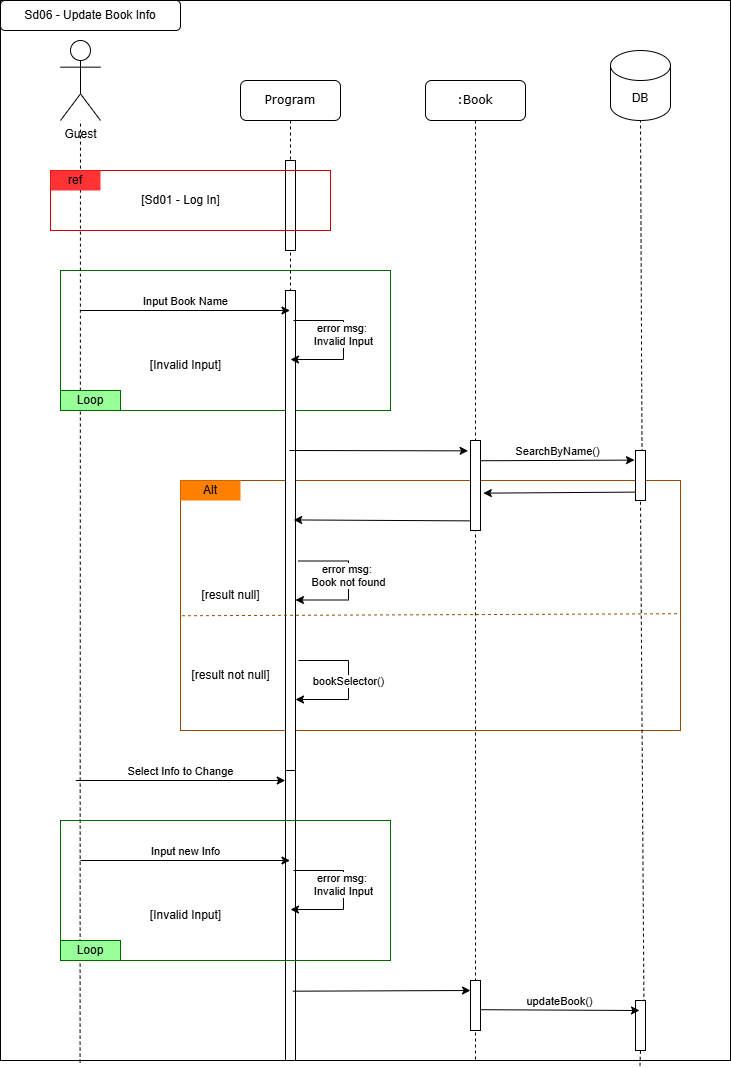
### **3.4 Change User Status**



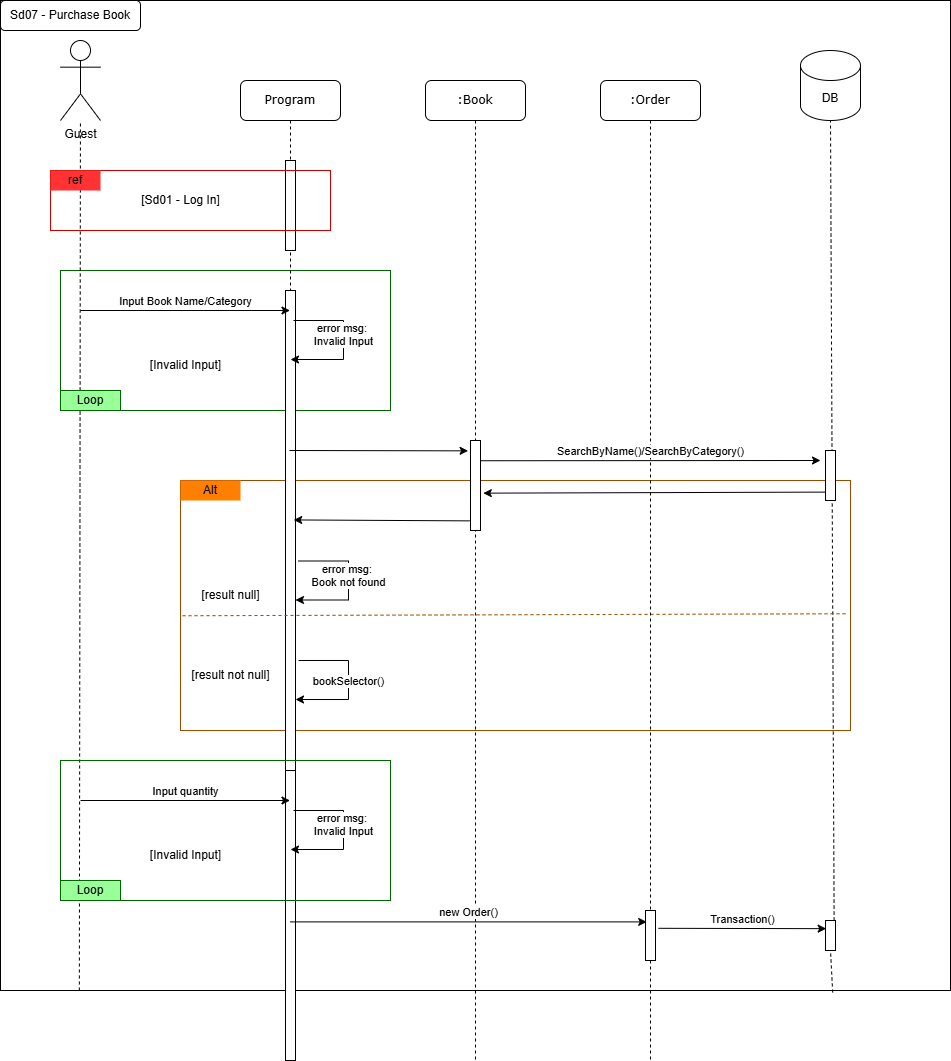
### **3.5 Update Account Info**



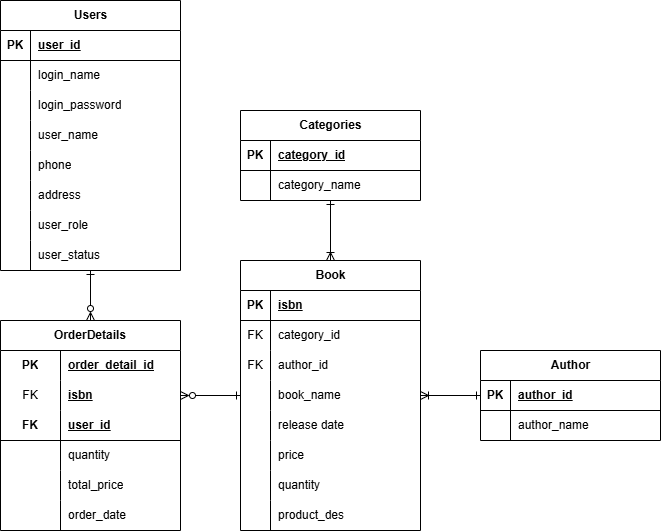
### **3.6 Update Book Info**



### **3.7 Purchase Book**



1. **Database Design**
   1. **Entity Relationship Diagram**



* 1. **Database Design Details**

|  |  |  |  |
| --- | --- | --- | --- |
| Users | | | |
| Column Name | Data Type | Constraints | Description |
| User\_id | int | Primary Key, Autoincrement |  |
| Login\_name | varchar(50) | NOT NULL, UNIQUE | Username for login |
| Login\_password | varchar(50) | NOT NULL | Password for login |
| User\_name | varchar(100) | NOT NULL | Fullname |
| Phone | varchar(20) | NOT NULL | Phone Number |
| Address | varchar(255) | NOT NULL |  |
| User\_role | int | DEFAULT 0 | 0: User  1: Admin |
| User\_status | int | DEFAULT 1 | 0: Not active  1: Active |

|  |  |  |  |
| --- | --- | --- | --- |
| Categories | | | |
| Column Name | Data Type | Constraints | Description |
| Category\_id | int | Primary Key, Autoincrement |  |
| Category\_name | varchar(100) | NOT NULL |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Author | | | |
| Column Name | Data Type | Constraints | Description |
| Author\_id | int | Primary Key, Autoincrement |  |
| Author\_name | varchar(100) | NOT NULL |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Book | | | |
| Column Name | Data Type | Constraints | Description |
| Isbn | varchar(20) | Primary Key | Book ID |
| Category\_id | int | Foreign key |  |
| Author\_id | int | Foreign Key |  |
| Book\_name | varchar(255) | NOT NULL |  |
| Release\_date | date | NOT NULL |  |
| Price | decimal(10,2) | NOT NULL | Unit price |
| Quantity | int | NOT NULL | Quantity in stock |
| Product\_des | text | DEFAULT ‘No desc’ | Description of book |

|  |  |  |  |
| --- | --- | --- | --- |
| OrderDetails | | | |
| Column Name | Data Type | Constraints | Description |
| Order\_detail\_id | int | Primary Key, Autoincrement |  |
| User\_id | int | Foreign key |  |
| Isbn | varchar(20) | Foreign Key |  |
| Order\_date | datetime | Default Current Date | Order Creation date |
| Total\_price | decimal(10,2) | NOT NULL | Total price of order |
| Quantity | int | NOT NULL | Quantity of order |

# **Test**

## **1. Sign In Test**

### **1.1 Sign In Test 1**

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | SignInTest01() |
| Test Case Description | This test case check for user info input |
| Preconditions | Username not exist in database |
| Test Case Input | “jim”, “123”, “Jimmy”, “0123456789”, “20 Oak Street”  “jake”, “123”, “Jake”, “0123456789”, “21 Oak Street” |
| Test Case Expected Output | New User |
| Test Case Steps | New User  Call SignIn method  Compare output with expected output |
| Default Value Preverving |  |

### **1.2 Sign In Test 2**

|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | SignInTest02() |
| Test Case Description | This test case check for user info input |
| Preconditions |  |
| Test Case Input | “jim”  “jake” |
| Test Case Expected Output | “Username Exist!” |
| Test Case Steps | New User  Call SignIn method  Compare output with expected output |
| Default Value Preverving |  |

## **2. Login Test**

### **2.1 Login Test 1**

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | LoginTest01() |
| Test Case Description | This test case check username and password correct or not |
| Preconditions | Username and password must exist in database |
| Test Case Input | “jim”, “123”  “admin1”, “123” |
| Test Case Expected Output | An User |
| Test Case Steps | New User  Call Login method  Compare output with expected output |
| Default Value Preverving |  |

### **2.2 Login Test 2**

|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | LoginTest02() |
| Test Case Description | This test case check username and password correct or not |
| Preconditions |  |
| Test Case Input | “jimm”, “123”  “jakee”, “123”  “jake”, “124” |
| Test Case Expected Output | “User Not Found!” |
| Test Case Steps | New User  Call Login method  Compare output with expected output |
| Default Value Preverving |  |

## **3. Book Search Test**

### **3.1 Seacrh by Name Test 1**

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | SearchByNameTest01() |
| Test Case Description | Search for book in the system |
| Preconditions | Logged in |
| Test Case Input | “ ”  “a”  “foundation” |
| Test Case Expected Output | List Book |
| Test Case Steps | New Book  Call SearchByName method  Compare output with expected output |
| Default Value Preverving |  |

### **3.2 Seacrh by Name Test 2**

|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | SearchByNameTest02() |
| Test Case Description | Search for book in the system |
| Preconditions | Logged in |
| Test Case Input | “1”  “#”  “]” |
| Test Case Expected Output | “No Book Found!” |
| Test Case Steps | New Book  Call SearchByName method  Compare output with expected output |
| Default Value Preverving |  |

### **3.3 Seacrh by Category Test 1**

|  |  |
| --- | --- |
| Test Case Number | 003 |
| Test Case Name | SearchByCategoryTest01() |
| Test Case Description | Search for book in the system |
| Preconditions | Logged in |
| Test Case Input | “1”  “#”  “]” |
| Test Case Expected Output | “No Book Found!” |
| Test Case Steps | New Book  Call SearchByCategory method  Compare output with expected output |
| Default Value Preverving |  |

### **3.4 Seacrh by Category Test 2**

|  |  |
| --- | --- |
| Test Case Number | 004 |
| Test Case Name | SearchByCategoryTest02() |
| Test Case Description | Search for book in the system |
| Preconditions | Logged in |
| Test Case Input | “ ”  “s”  “fiction” |
| Test Case Expected Output | List Book |
| Test Case Steps | New Book  Call SearchByCategory method  Compare output with expected output |
| Default Value Preverving |  |

## **4. Create Order Test**

### **4.1 Create Order Test 1**

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | CreateOrderTest01() |
| Test Case Description | Create order for a book with quantity to buy |
| Preconditions | Logged in |
| Test Case Input | “10”  “20” |
| Test Case Expected Output | New Order |
| Test Case Steps | New Order  Call Transaction method  Compare output with expected output |
| Default Value Preverving |  |

### **4.2 Create Order Test 2**

|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | CreateOrderTest02() |
| Test Case Description | Create order for a book with quantity to buy |
| Preconditions | Logged in |
| Test Case Input | “-1”  “0”  “9999999999999” |
| Test Case Expected Output | Null |
| Test Case Steps | New Order  Call Transaction method  Compare output with expected output |
| Default Value Preverving |  |

## **5. Update Account Info Test**

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | UpdateAccountTest01() |
| Test Case Description | Change selected info of user account |
| Preconditions | Logged in |
| Test Case Input | “Johnny”  “0123456789”  “54 Oak Street” |
| Test Case Expected Output | True |
| Test Case Steps | New Order  Call Transaction method  Compare output with expected output |
| Default Value Preverving |  |

## **6. Add Book Test**

### **6.1 Add Book Test 1**

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | AddBookTest01() |
| Test Case Description | Add a Book into database |
| Preconditions | Logged in |
| Test Case Input | “book-11”, “Wizard of Oz”, “Frank”, “Fantasy”, “1990/12/20”  “book-12”, “Algebra”, “William”, ”Science”, “2005/6/2” |
| Test Case Expected Output | True |
| Test Case Steps | New Book  Call AddBook method  Compare output with expected output |
| Default Value Preverving |  |

### **6.2 Add Book Test 2**

|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | AddBookTest02() |
| Test Case Description | Add a Book into database |
| Preconditions | Logged in |
| Test Case Input | “book-11”  “book-12” |
| Test Case Expected Output | False |
| Test Case Steps | New Book  Call AddBook method  Compare output with expected output |
| Default Value Preverving |  |

## **7. Update Book Info Test**

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | UpdateBookTest01() |
| Test Case Description | Change selected info of a book |
| Preconditions | Logged in |
| Test Case Input | “Wizard Oz”  “Fiction”  “Frank Baum”  “1996/12/22”  “19.99”  “100” |
| Test Case Expected Output | True |
| Test Case Steps | New Book  Call updateBook method  Compare output with expected output |
| Default Value Preverving |  |

## **8. User Search Test**

### **8.1 User Seacrh Test 1**

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | UserSearchTest01() |
| Test Case Description | Search for user in the database |
| Preconditions | Logged in |
| Test Case Input | “ ”  “j”  “jim” |
| Test Case Expected Output | List User |
| Test Case Steps | New User  Call SearchUserByName method  Compare output with expected output |
| Default Value Preverving |  |

### **8.2 User Seacrh Test 2**

|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | UserSearchTest02() |
| Test Case Description | Search for user in the database |
| Preconditions | Logged in |
| Test Case Input | “@”  “1” |
| Test Case Expected Output | Null |
| Test Case Steps | New User  Call SearchUserByName method  Compare output with expected output |
| Default Value Preverving |  |

## **9. Change User Status Test**

### **9.1 Deactivate User Status Test**

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | DeactivateUserStatusTest01() |
| Test Case Description | Change User Status |
| Preconditions | Logged in |
| Test Case Input |  |
| Test Case Expected Output | True |
| Test Case Steps | New User  Call updateUserStatus method  Compare output with expected output |
| Default Value Preverving |  |

### **9.2 Activate User Status Test**

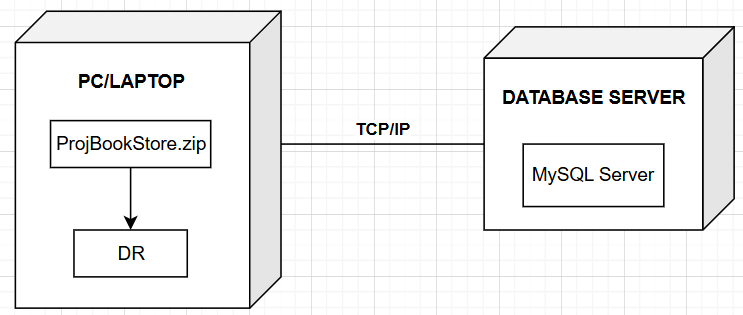
|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | ActivateUserStatusTest02() |
| Test Case Description | Change User Status |
| Preconditions | Logged in |
| Test Case Input |  |
| Test Case Expected Output | True |
| Test Case Steps | New User  Call updateUserStatus method  Compare output with expected output |
| Default Value Preverving |  |

# Task Assign

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TBN Group | Book Store | | | | | |
| No | Task name | Description | Start Date | End Date | Member | Self assessment |
| 1 | Design |  | July 1 | July 7 | Thanh |  |
| 2 | Code |  | July 1 | July 7 | Nga |  |
| 3 | Code |  | July 8 | July 20 | Thanh & Nga |  |
| 4 | Code |  | July 8 | July 15 | Binh |  |
| 5 | Control |  | July 21 | July 27 | Thanh & Binh |  |
| 6 | Write Report |  | July 21 | July 27 | Thanh |  |

# **Installation Instructions**

1. **Deployment Diagram**



1. **Installation steps**
   * Database Install
   * Server install
   * Application install