

**PROJECT REPORT**

# Book Shop Create Order System

|  |  |
| --- | --- |
| **Semester:** | Programming Fundamentals |
| **Class:** | PF1121 |
| **Group:** | H&T Team |
| **Instructor** | Nguyễn Xuân Sinh |
| **Team members:** | 114010122034 - Nguyễn Hoàng Hiệp  1140101122010 - Đinh Thanh Tuấn |

# Index

Book Store

Index 2

I. Project introduction 3

1. Proposed System
2. The scope of the project to be applied.
3. System Name
4. Deployment Environment
5. Development Tools
6. Customer Requirements (System features)

II. Analyze System Requirements 4

1. Use Case

2. Descriptions

2.1 Login Description 5

2.2 Create Order Description 6

2.3 View Staff Revenue in Day Description 7

3.Activity Diagram

3.1 Login 10

3.2 Create order 11

3.3 View Staff Revenue in Day. 12

III. Design Details 13

1. UI Design
2. UI Main Menu
3. UI Create Order
4. UI Add new Customer.
5. UI Payment
6. UI Invoices
7. UI View revenue in day

IV. Test 29

V. Task Assign 50

# Project Introduction

In the context of strong development of technology and digitization, offline book selling applications have become an indispensable part of bookstores. This application gives readers new experiences and unique utilities.

An offline book selling application is a useful tool that brings many important benefits such as: Convenience and time saving, information security, ease of use.

This system will provide some basic features including:  
 - Log in  
 - Create order  
 - View revenue of staff for the day.

1.Proposed System

* Creating a program for Staff create order and confirm payment or cancel order, then confirm export invoices.
* Creating a program for Staff to view their revenue.

2.The scope of the project to be applied.

* The scope of the project includes the development of core functions of the system, such as user authentication, data management, and reporting.

3.System Name

* System named “Book shop create order system”.

4.Deployment Environment

* The system will be deployed on a personal computer platform, ready and easily accessible. And can only be accessed on 1 device

5.Development Tools

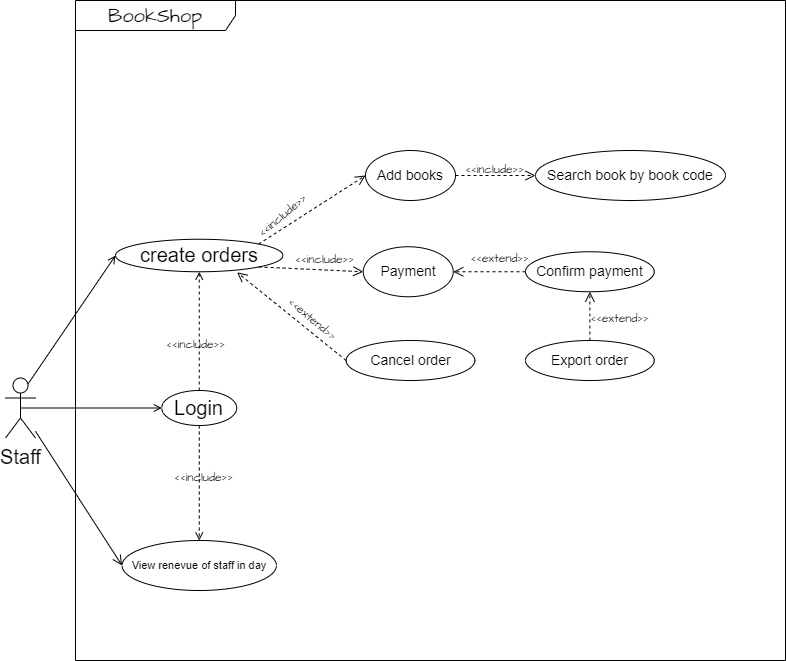
* Visual Studio Code
* Draw.io Diagram
* MySQL Workbench 8.0 CE.

6.Customer Requirements (System features)

* Staff login
* Create order.
* View staff revenue in day
* Show invoices.
* Confirm payment or cancel payment.

# II.Analyze System Requirements

1.Use Case



2.1 Login descriptions:

|  |  |
| --- | --- |
| Use Case Name | Login |
| Use Case ID | UC-01 |
| Description | Used to allows staff access to program and define the accessible depending on type of their account |
| Actor | Staff |
| Organizational Benefits | This feature allow staff to create invoices and staff can check all invoices and totals of the shifts |
| Triggers | This feature will automatically activate when the program started |
| Preconditions | If you enter correct username and password, database must be connected automatically, account must be made available |
| Postconditions | Success: Staff have logged into the system; the main menu of the program must be displayed.  Fail: Staff haven’t logged in the system, and their must re-enter correct username and password |
| Main Course | 1. Enter username and password  2. Verify account  3.Show main menu |
| Alternate Courses | Alternate paths through the system  AC1: Wrong username or password  1. Display error message  2. Show login form again  AC2: Correct username and password  1. Login success message  2. Show main menu  3. Re-enter request |
| Exceptions | EX1: Input wrong Username/password  1. Display: “Invalid username or password !!!, please check again and re-enter.” And ask re-enter until enter correct.  2. Request to re-enter username/password.  3  EX2: Not exist in database.  1. Show message: "Invalid Username / Password".  2. Request to connect internet connection again. |

2.2 Create order descriptions.

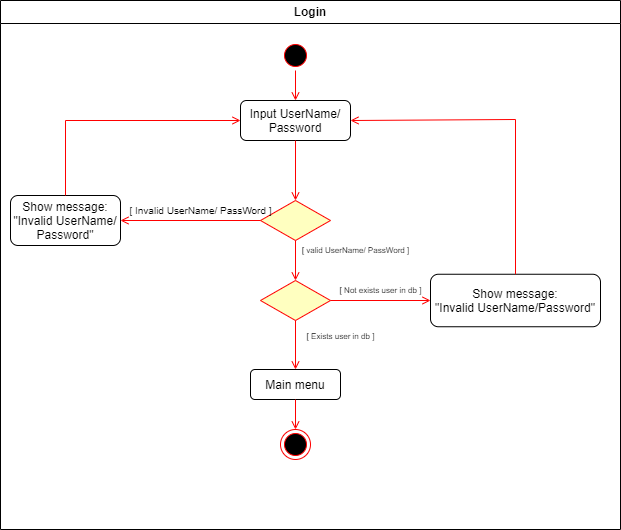
|  |  |
| --- | --- |
| Use Case Name | Create order |
| Use Case ID | UC-02 |
| Description | Used to allow employees to create invoices and pay bills to customers. |
| Actor | Staff |
| Organizational Benefits | This feature allows staff to search, select, view books requested by customers and create order slips. |
| Triggers | This feature would show all main menu, search results, books information. |
| Preconditions | If you choose available books, the payment page would be open. |
| Postconditions | Success: After staff create the orders, Payment page would be display.  Fail: Main menu would be re-open. |
| Main Course | 1. Search books  2. Choose books  3. Create order bills |
| Alternate Courses | AC1: Create order fail.  1. Display fail message  2. Request create order again.  AC2: Create order success.  1. Success message display.  2. Show main menu |
| Exceptions | EX1: Out of stock  1. Display: “This book is out of stock. You can buy other books, or you can pre-order”.  2. Request to buy other books or pre-order.  EX2: Cancel order.  1. Cancel the order  2.Display” Cancel successful !!!” |

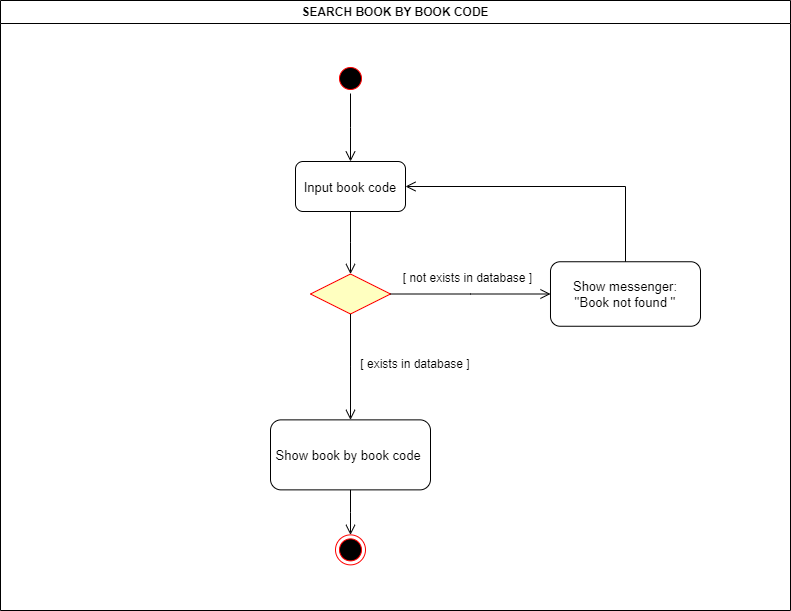
2.3 View Staff Revenue in Day description

|  |  |
| --- | --- |
| Use Case Name | View Staff Revenue in Day |
| Use Case ID | UC-03 |
| Description | Used to see history revenue in Staff login shift |
| Actor | Staff |
| Organizational Benefits | This feature allows staff to check again revenue |
| Triggers |  |
| Preconditions | The feature will let you check again your revenue in your shift |
| Postconditions | Success: Show all revenue in your shift in a day |
| Main Course | View your own revenue |
| Alternate Courses | AC1: View success.  List order of Staff’s revenue will be display. |
| Exceptions | EX1: Haven’t any order created.  1 Show empty table. |

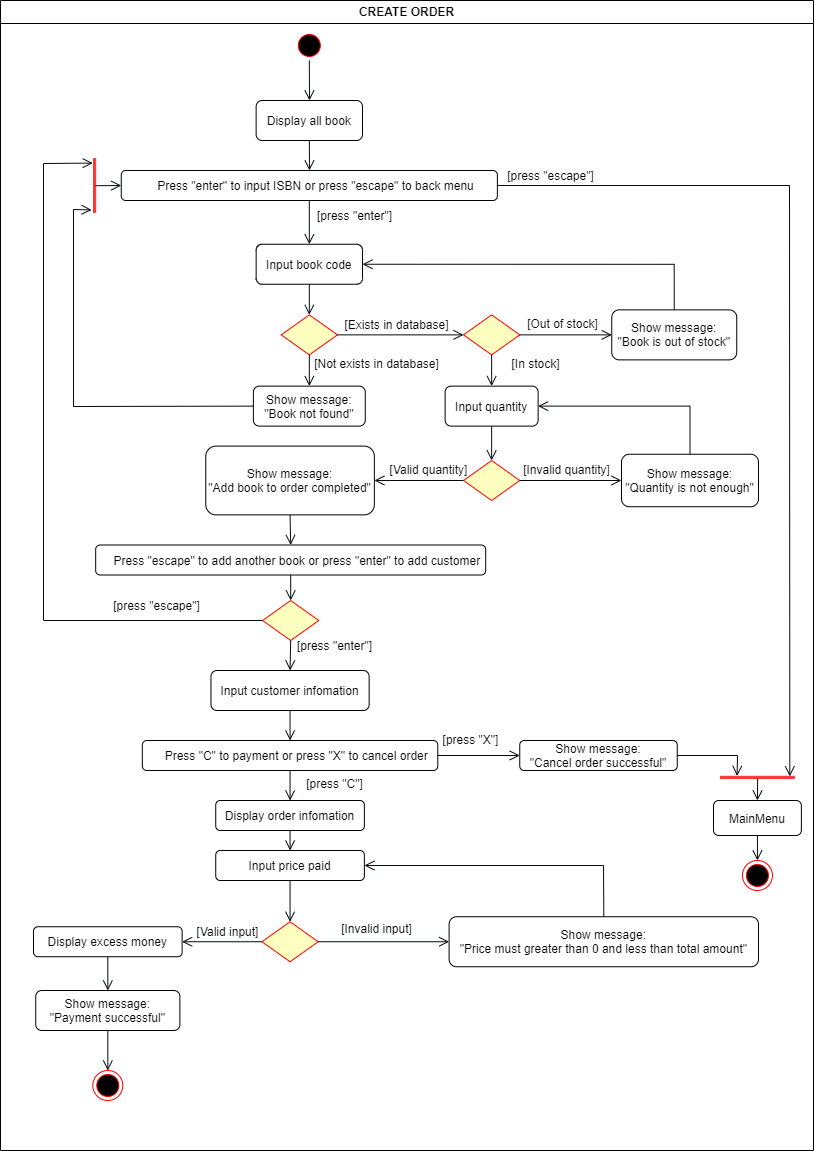
3.Activity Diagram:

3.1 Login



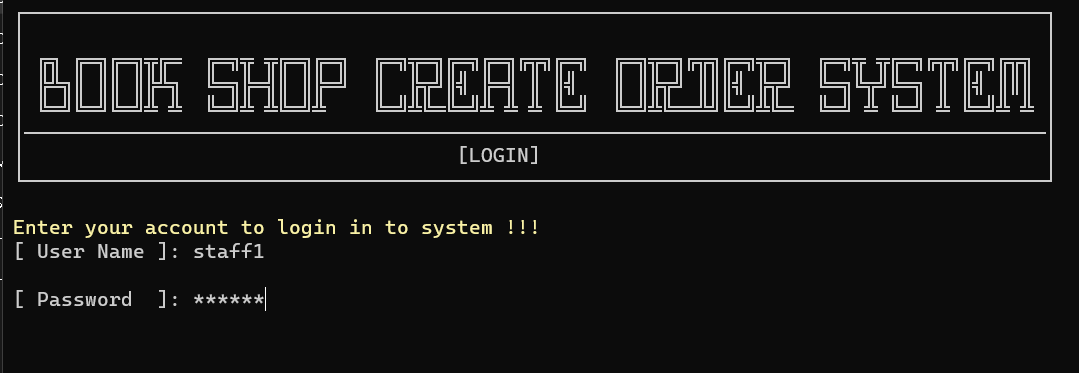
****3.2 Search Book by Book Code

2.3 Create order:

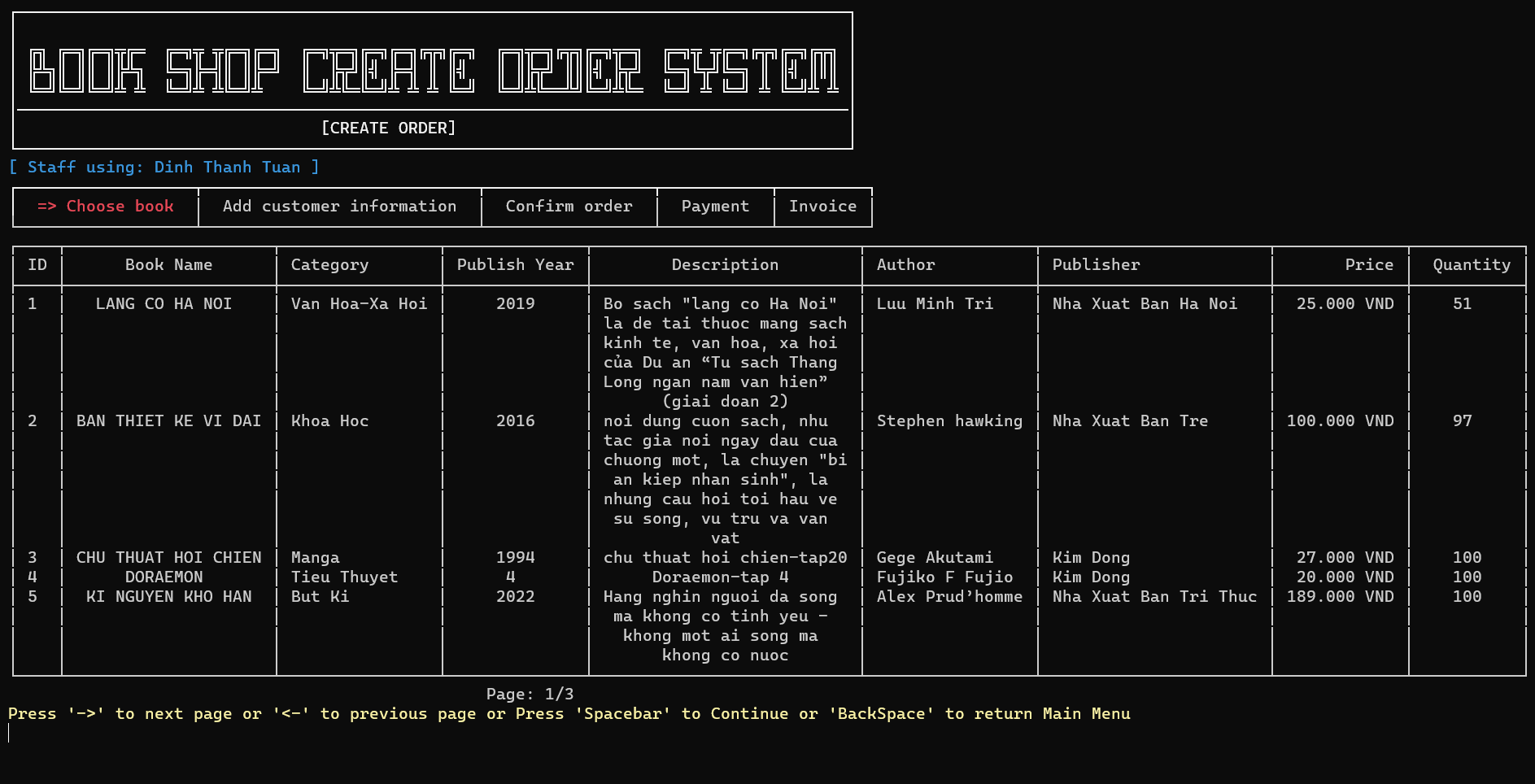


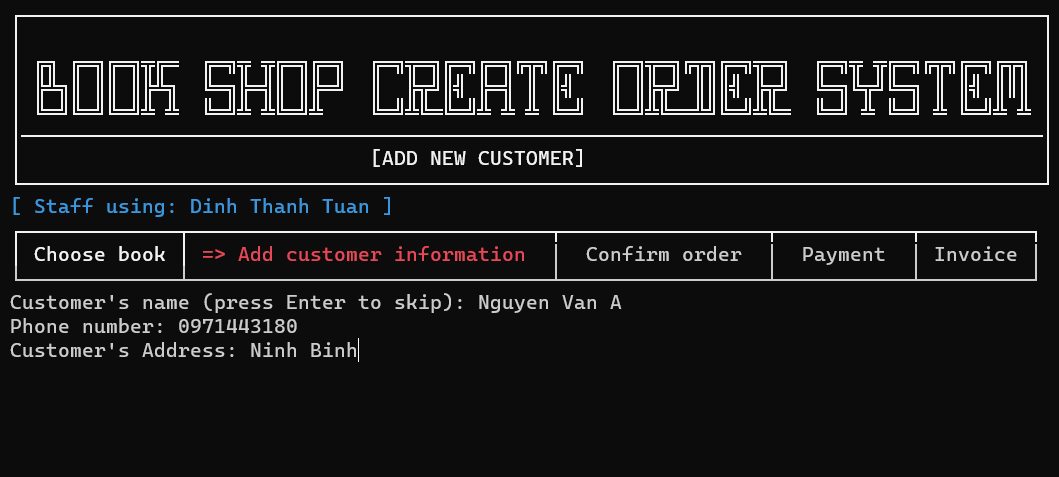
# III.Analyze System Requirements

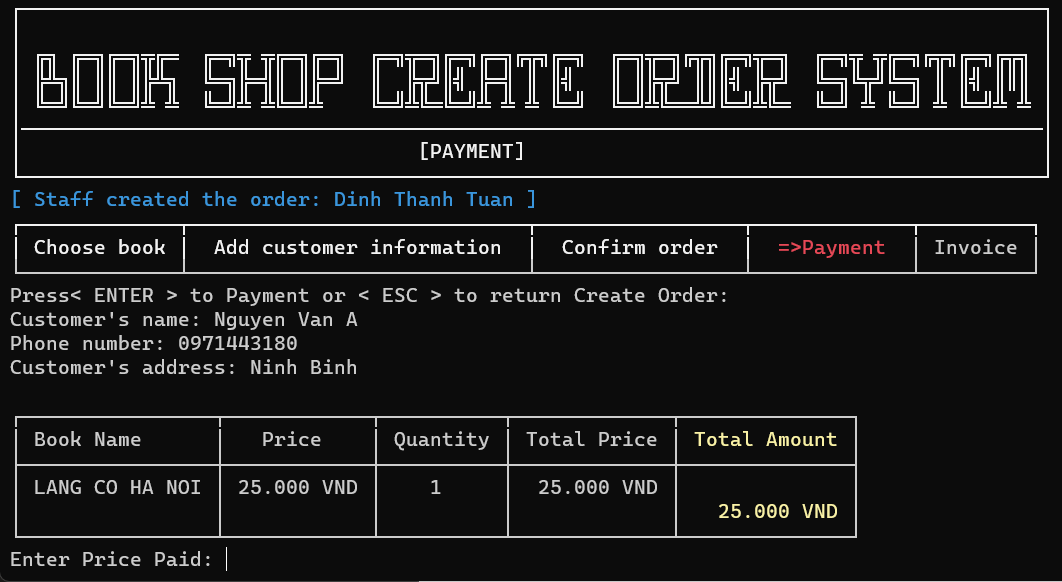
1. UI Design

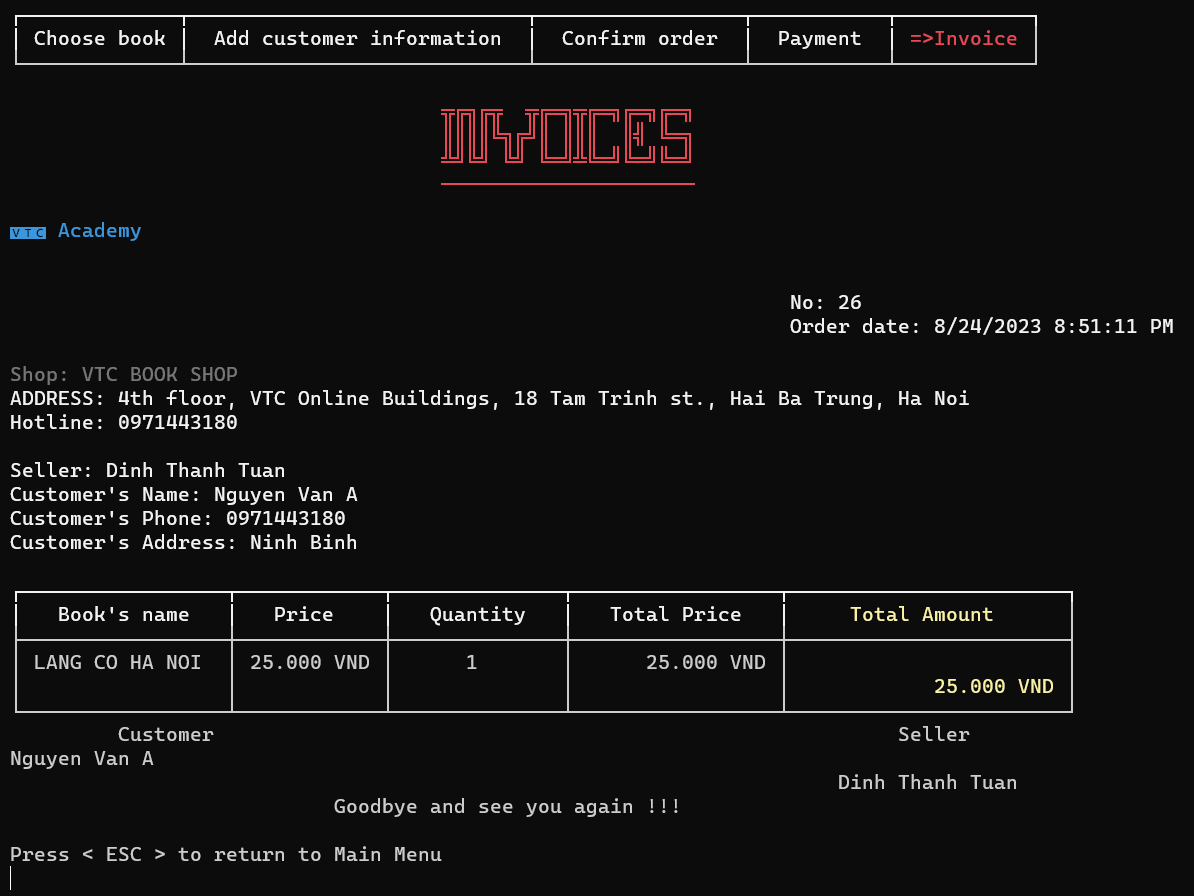
1.1 UI Login

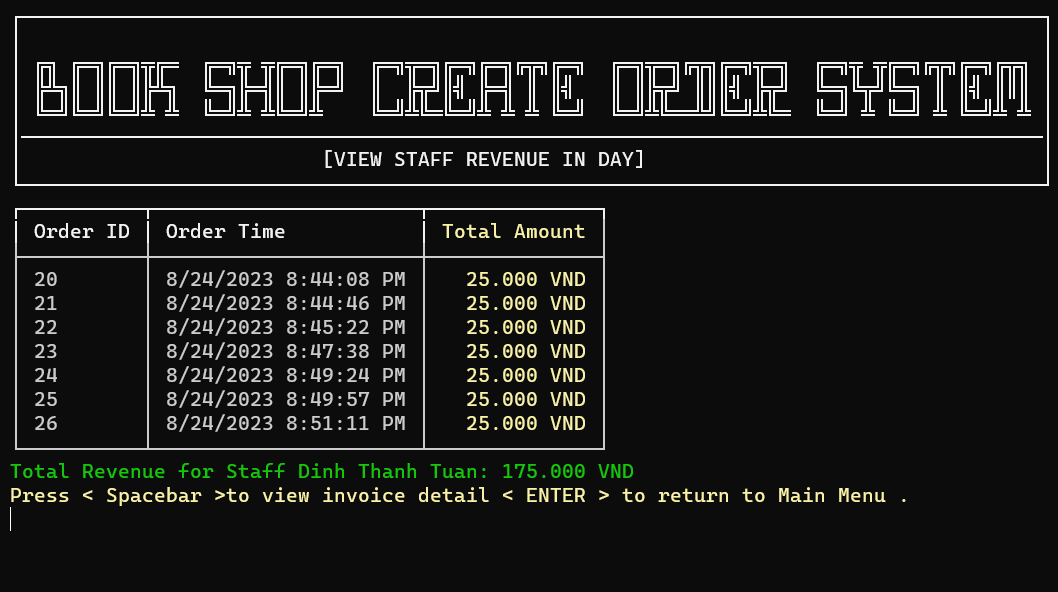
1.2. UI Main Menu

1.3. UI Create order

1.4. UI Add new customer.

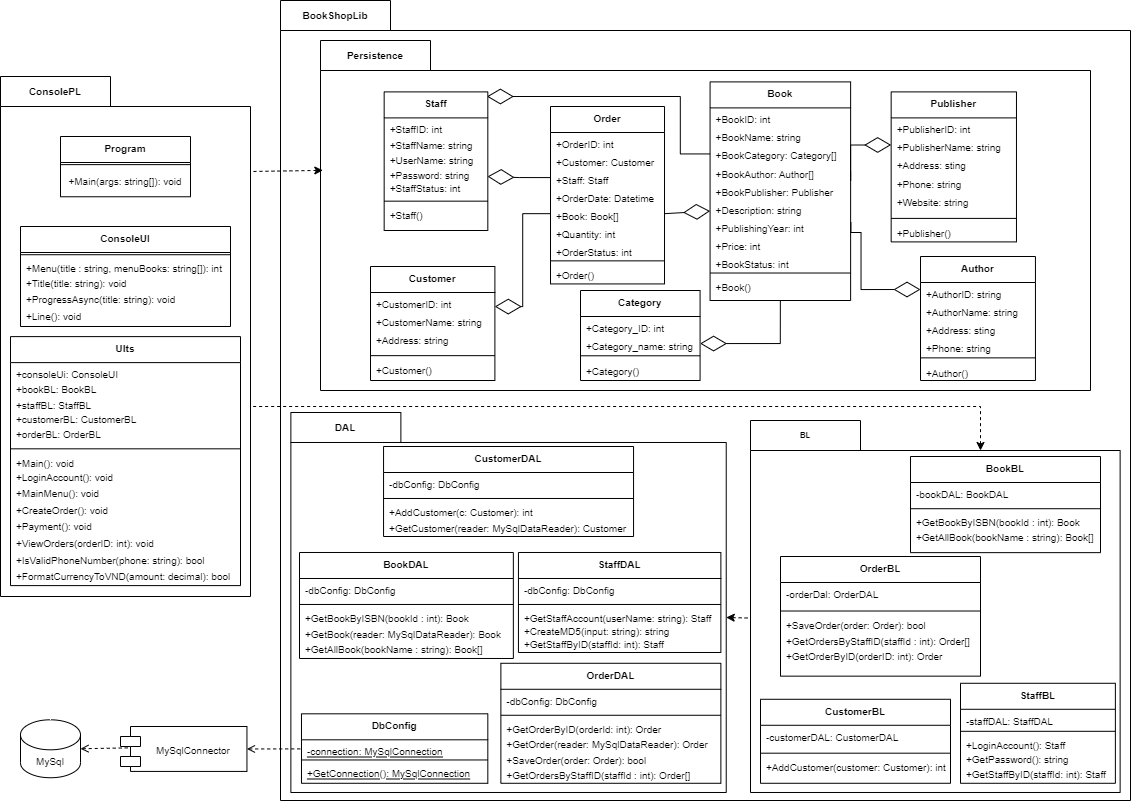
1.5. UI Payment

1.6. UI Invoice:

1.7. View revenue of staff in day UI:

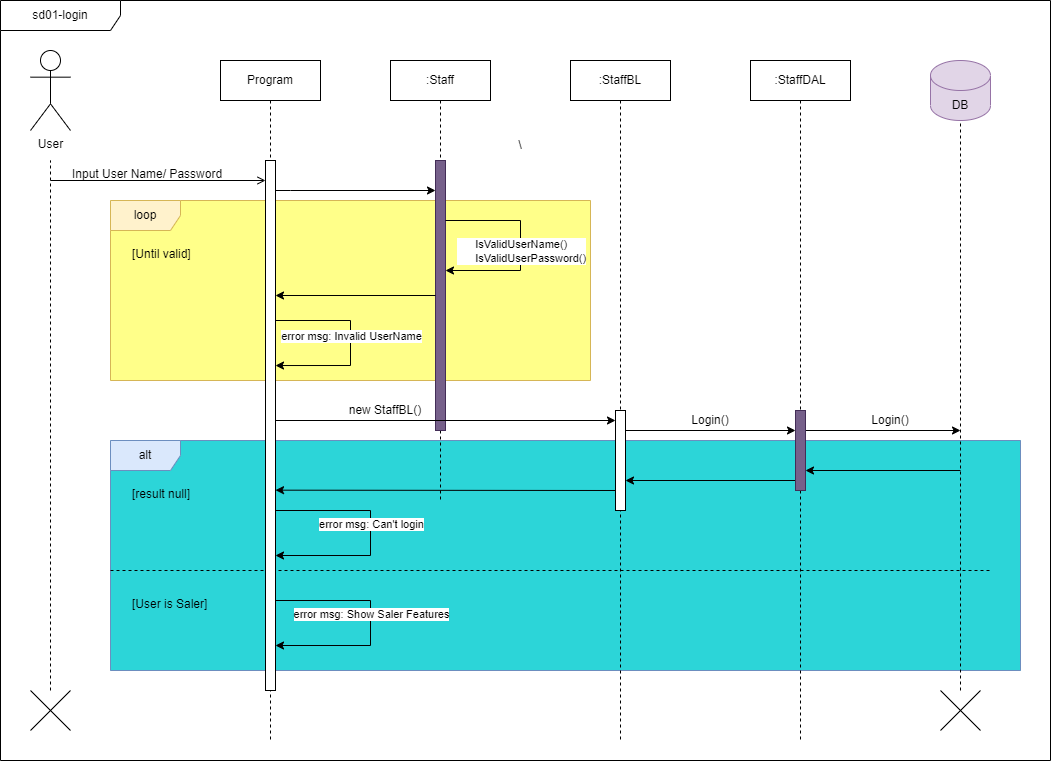
1. Code Design (Class Diagram)

Class Diagram

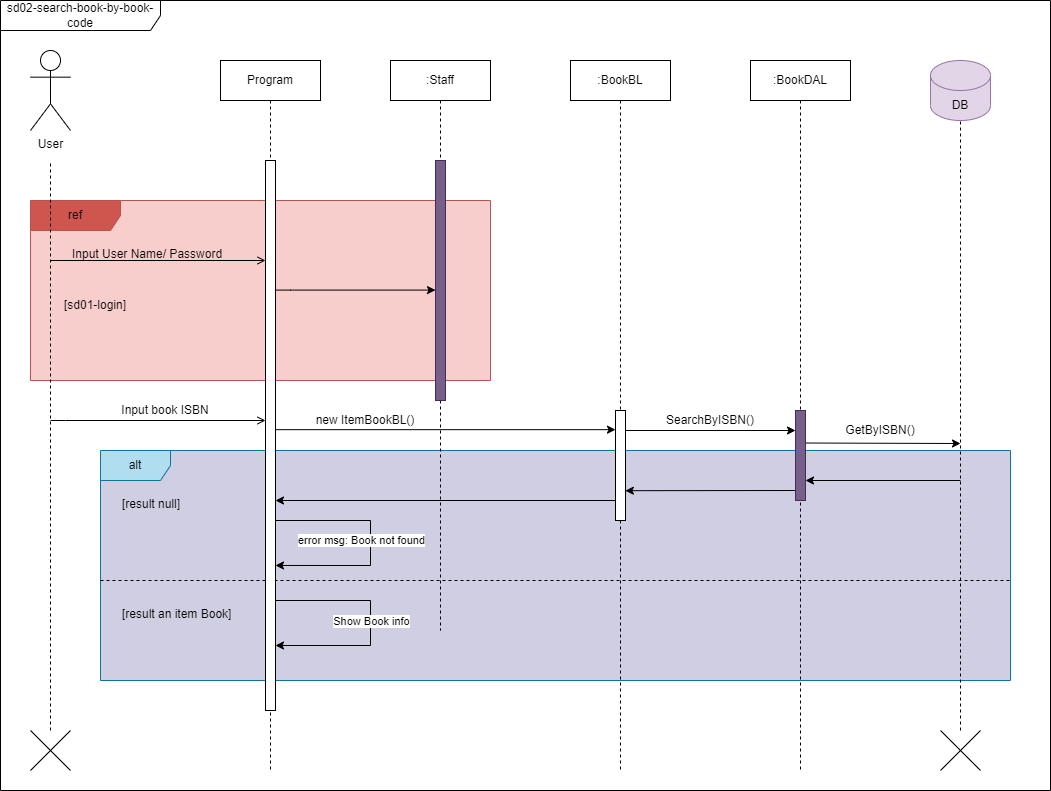


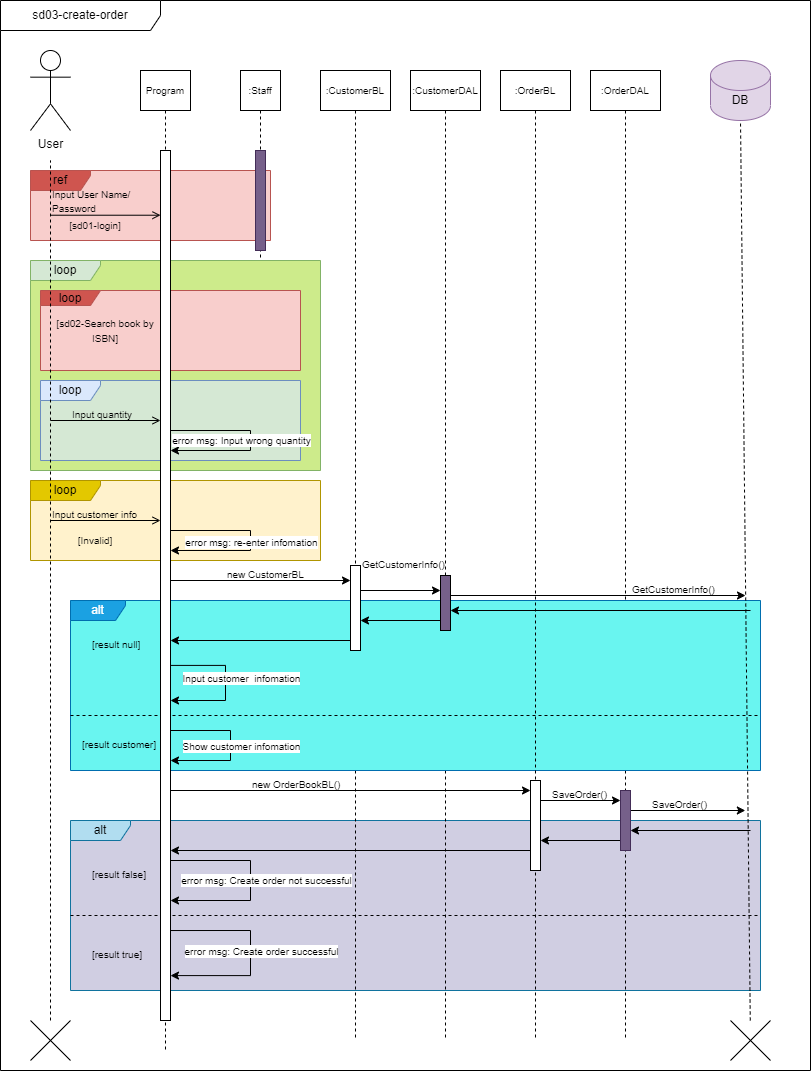
1. Sequence Diagram

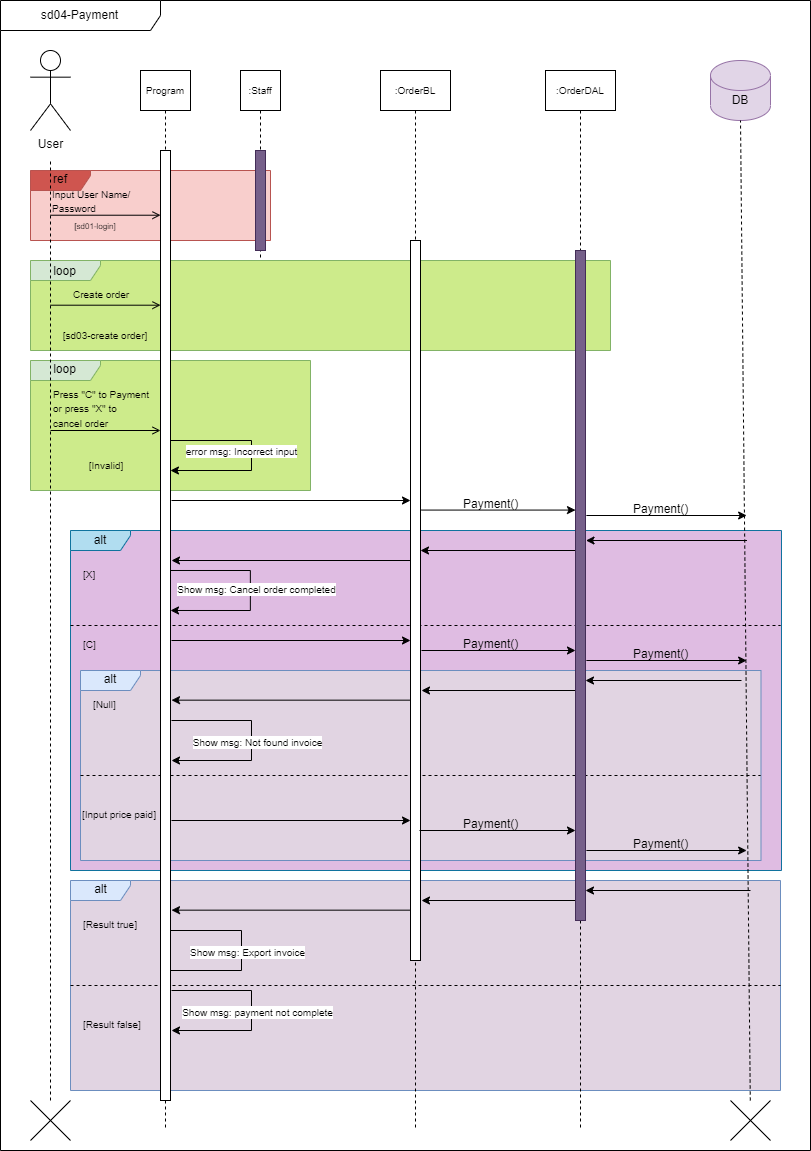
3.1 Login:



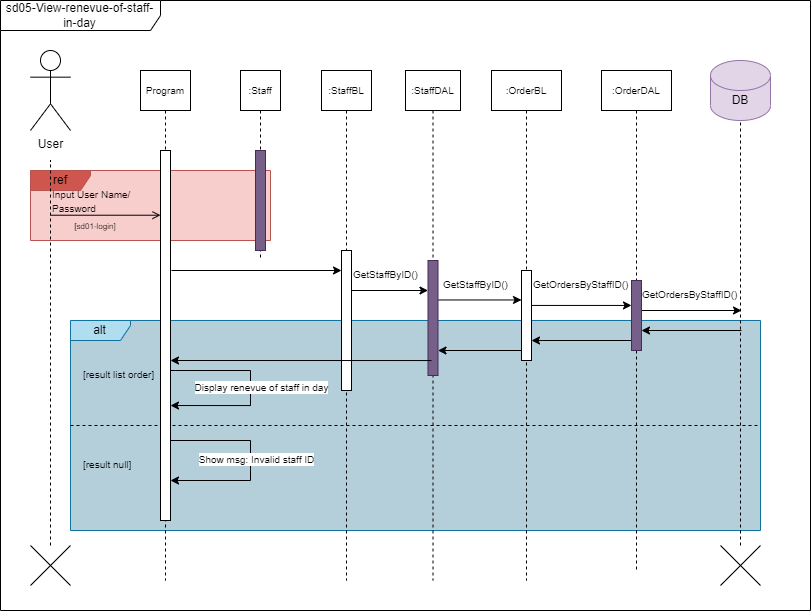
3.2 Search book by Code:

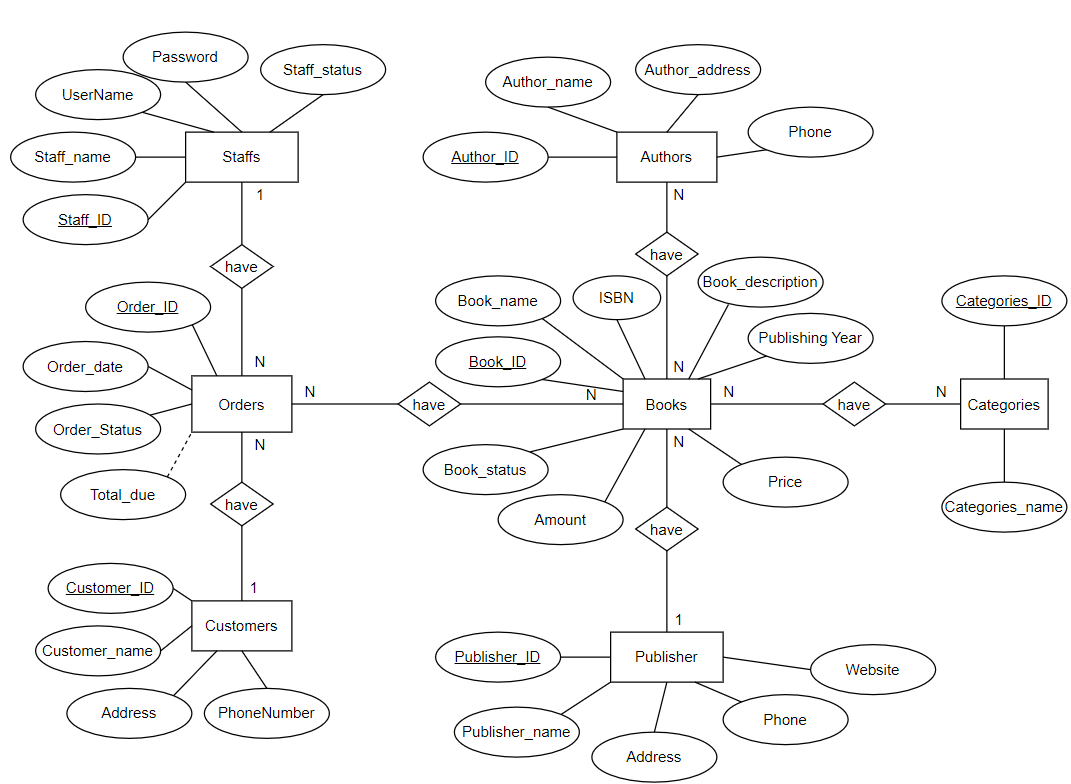


3.3 Create order:  


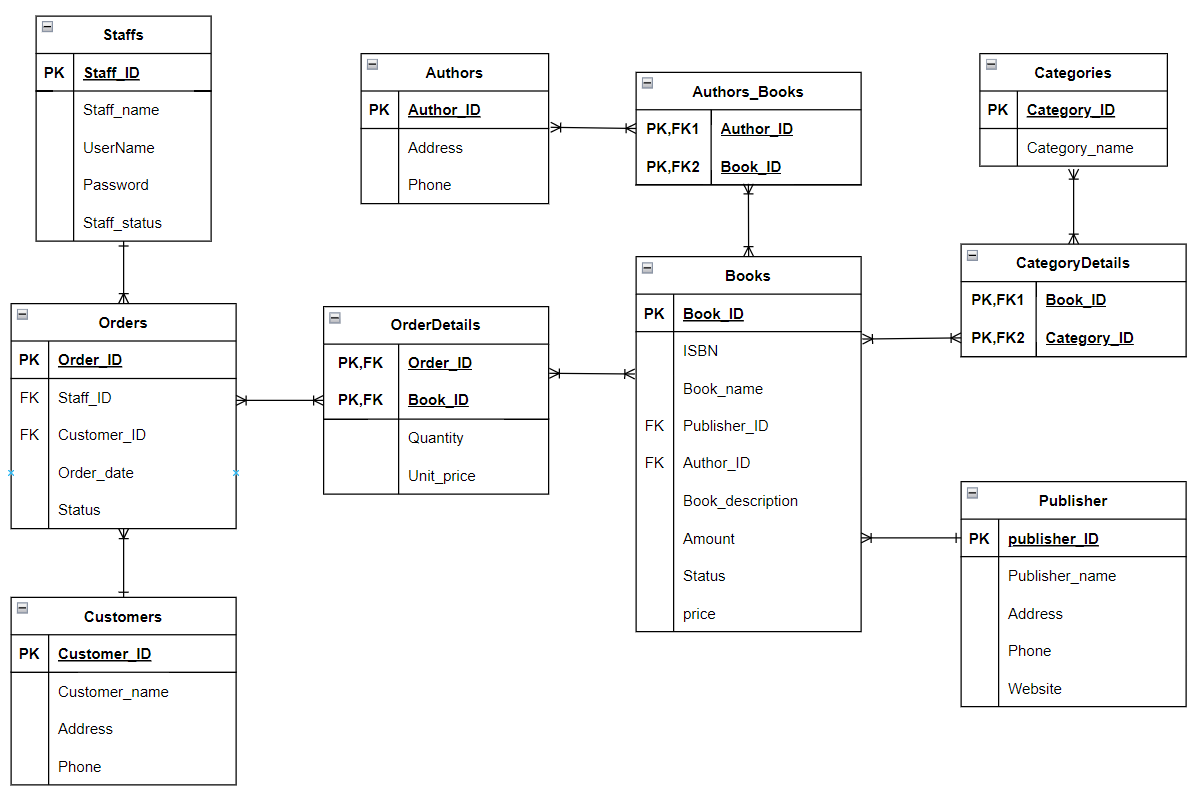
3.4 Payment:

3.5 View revenue of staff in day:

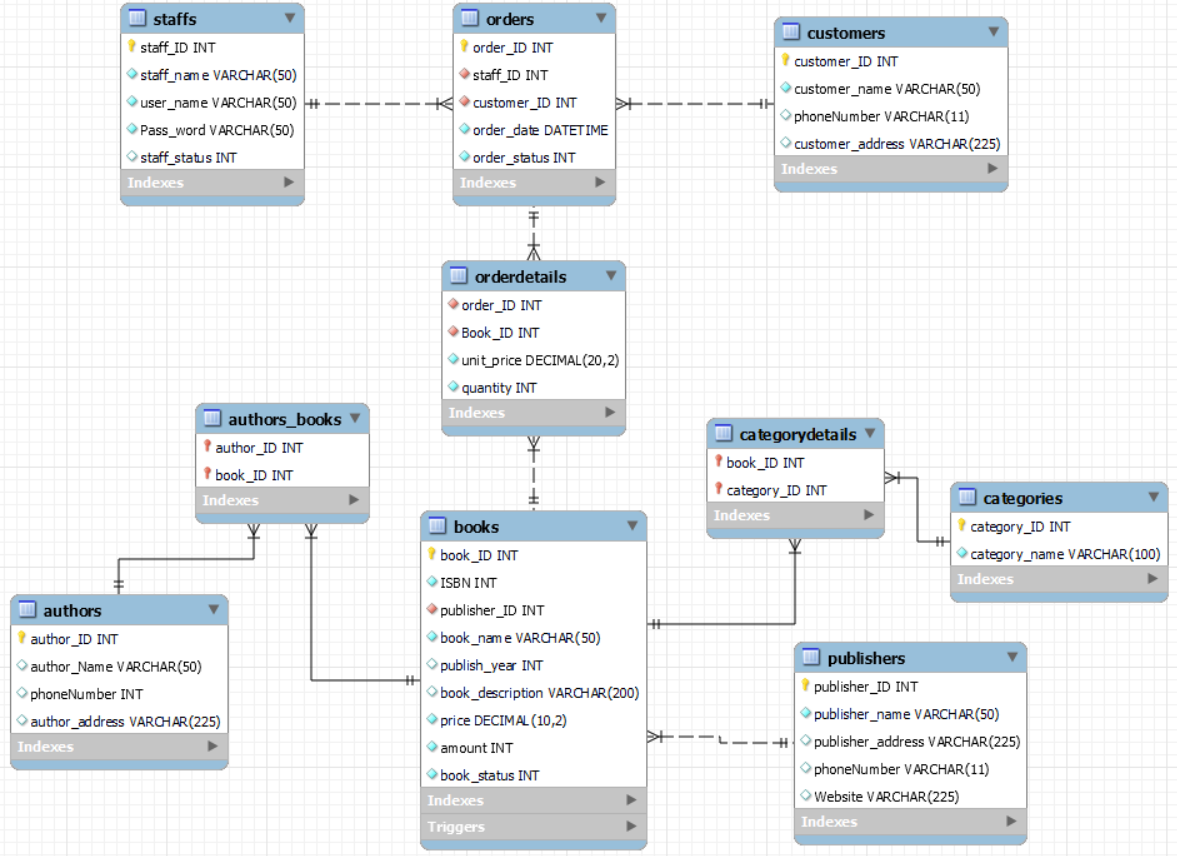


1. Database Design
   1. Entity Relationship Diagram

Or



b. Database Design Details



|  |  |  |  |
| --- | --- | --- | --- |
| Staffs | | | |
| Column Name | Data Type | Constraints | Description |
| Staff\_ID | int | Primary Key |  |
| Staff\_name | varchar(50) | NOT NULL |  |
| UserName | varchar(225) | NOT NULL |  |
| Password | varchar(225) | NOT NULL |  |
| Staff\_status | int | NOT NULL |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Customers | | | |
| Column Name | Data Type | Constraints | Description |
| Customer\_ID | int | Primary Key |  |
| Customer\_name | varchar(50) | NOT NULL |  |
| PhoneNumber | varchar(11) | NOT NULL |  |
| Address | varchar(225) | NOT NULL |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Orders | | | |
| Column Name | Data Type | Constraints | Description |
| Order\_ID | int | Primary Key |  |
| Staff\_ID | int | NOT NULL, Foreign Key |  |
| Customer\_ID | int | NOT NULL, Foreign Key |  |
| Order\_date | datetime | NOT NULL |  |
| Order\_status | int | NOT NULL |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Authors | | | |
| Column Name | Data Type | Constraints | Description |
| Author\_ID | int | Primary Key |  |
| Author\_name | varchar(50) | NOT NULL |  |
| Address | varchar(225) | NOT NULL |  |
| PhoneNumber | varchar(11) | NOT NULL |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Publishers | | | |
| Column Name | Data Type | Constraints | Description |
| Publisher\_ID | int | Primary Key |  |
| Publisher\_name | varchar(50) | NOT NULL |  |
| Address | varchar(225) | NOT NULL |  |
| PhoneNumber | varchar(11) | NOT NULL |  |
| Website | varchar(225) | NOT NULL |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Books | | | |
| Column Name | Data Type | Constraints | Description |
| Book\_ID | int | Primary Key |  |
| ISBN | int | UNIQUE |  |
| Book\_name | Varchar(50) | NOT NULL |  |
| Publisher\_name | varchar(50) | NOT NULL, Foreign Key |  |
| Author\_name | varchar(50) | NOT NULL, Foreign Key |  |
| Book\_description | varchar(100) |  |  |
| Publishing\_Year | int | Not Null |  |
| Book\_status | int | NOT NULL |  |
| Price | decimal(10,2) | Price > 0 |  |
| Amount | int | NOT NULL |  |

|  |  |  |  |
| --- | --- | --- | --- |
| OrderDetails | | | |
| Column Name | Data Type | Constraints | Description |
| Order\_ID | int | Primary Key, Foreign Key |  |
| Book\_ID | int | Primary Key, Foreign Key |  |
| Quantity | varchar(225) | NOT NULL |  |
| Unit\_price | decimal(10.2) | Unit price > 0 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Authors\_Books | | | |
| Column Name | Data Type | Constraints | Description |
| Book\_ID | int | Primary Key, Foreign Key |  |
| Author\_ID | int | Primary Key, Foreign Key |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Authors\_Books | | | |
| Column Name | Data Type | Constraints | Description |
| Book\_ID | int | Primary Key, Foregin Key |  |
| Author\_ID | int | Primary Key, Foregin Key |  |

|  |  |  |  |
| --- | --- | --- | --- |
| CategoryDetails | | | |
| Column Name | Data Type | Constraints | Description |
| Book\_ID | int | Primary Key, Foregin Key |  |
| Category\_ID | int | Primary Key, Foregin Key |  |

# IV.Test

1. Login test

1.1 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 the database |
| Test Case Input | “staff1”, “staff1”  “staff2”, “staff2” |
| Test Case Expected Output | A staff |
| Test Case Steps | New Staff  New StaffDAL  Call Login method  Compare output with expected output |
| Default Value Preserving |  |

1.1 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 | Username and password must exist in the database |
| Test Case Input | “staff1”, “staff1”  “staff2”, “staff2”  “staff3”, “staff3” |
| Test Case Expected Output | Null |
| Test Case Steps | New Staff  New StaffDAL  Call Login method  Compare output with expected output |
| Default Value Preserving |  |

1.3 Login test 3

|  |  |
| --- | --- |
| Test Case Number | 003 |
| Test Case Name | LoginTest03() |
| Test Case Description | This test case check username and password correct or not |
| Preconditions | Username and password must exist in the database |
| Test Case Input | “123”, “321” |
| Test Case Expected Output | Null |
| Test Case Steps | New Staff  New StaffDAL  Call Login method  Compare output with expected output |
| Default Value Preserving |  |

1.4 Login test 4

|  |  |
| --- | --- |
| Test Case Number | 004 |
| Test Case Name | LoginTest04() |
| Test Case Description | This test case check username and password correct or not |
| Preconditions | Username and password must exist in the database |
| Test Case Input | “-01”, “32121” |
| Test Case Expected Output | Null |
| Test Case Steps | New Staff  New StaffDAL  Call Login method  Compare output with expected output |
| Default Value Preserving |  |

2. Test search book by book code

2.1 Test search book by book code 1

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | GetSearchBookByBookCode1() |
| Test Case Description | Search for book in the system |
| Preconditions | Book must exist in the database  Logged in |
| Test Case Input | “1234567890”  “234567890” |
| Test Case Expected Output | A book |
| Test Case Steps | New BookDAL  Call GetBookByBookCode method  Compare output with expected output |
| Default Value Preserving |  |

2.2 Test search book by book code 2

|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | GetSearchBookByBookCode2() |
| Test Case Description | Search for book in the system |
| Preconditions | Book must exist in the database.  Logged in |
| Test Case Input | “1234567890”  “2345678901”  “3456789012” |
| Test Case Expected Output | A book |
| Test Case Steps | New BookDAL  Call GetBookByBookCode method  Compare output with expected output |
| Default Value Preserving |  |

2.3 Test search book by book code 3

|  |  |
| --- | --- |
| Test Case Number | 003 |
| Test Case Name | GetSearchBookByBookCode3() |
| Test Case Description | Search for book in the system |
| Preconditions | Book must exist in the database.  Logged in |
| Test Case Input | “0”  “24445678901” |
| Test Case Expected Output | Null |
| Test Case Steps | New BookDAL  Call GetBookByBookCode method  Compare output with expected output |
| Default Value Preserving |  |

2.4 Test search book by book code 4

|  |  |
| --- | --- |
| Test Case Number | 004 |
| Test Case Name | GetSearchBookByBookCode4() |
| Test Case Description | Search for book in the system |
| Preconditions | Book must exist in the database  Logged in |
| Test Case Input | “1234567890”  “6789012345” |
| Test Case Expected Output | A book |
| Test Case Steps | New BookDAL  Call GetBookByBookCode method  Compare output with expected output |
| Default Value Preserving |  |

2.5 Test search book by book code 5

|  |  |
| --- | --- |
| Test Case Number | 005 |
| Test Case Name | GetSearchBookByBookCode5() |
| Test Case Description | Search for book in the system |
| Preconditions | Book must exist in the database.  Logged in |
| Test Case Input | “1111111111” |
| Test Case Expected Output | Null |
| Test Case Steps | New BookDAL  Call GetBookByBookCode method  Compare output with expected output |
| Default Value Preserving |  |

2.6 Test search book by book code 6

|  |  |
| --- | --- |
| Test Case Number | 006 |
| Test Case Name | GetSearchBookByBookCode6() |
| Test Case Description | Search for book in the system |
| Preconditions | Book must exist in the database.  Logged in |
| Test Case Input | “-1225452445” |
| Test Case Expected Output | Null |
| Test Case Steps | New BookDAL  Call GetBookByBookCode method  Compare output with expected output |
| Default Value Preserving |  |

2.7 Test search book by book code 7

|  |  |
| --- | --- |
| Test Case Number | 007 |
| Test Case Name | GetSearchBookByBookCode7() |
| Test Case Description | Search for book in the system |
| Preconditions | Book must exist in the database.  Logged in |
| Test Case Input | “-0” |
| Test Case Expected Output | Null |
| Test Case Steps | New BookDAL  Call GetBookByBookCode method  Compare output with expected output |
| Default Value Preserving |  |

3. Test create order

3.1 Create order test 1.

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | CreateOrderTest1() |
| Test Case Description | Create an order with at least one book and customer information |
| Preconditions | Book must exist in the database.  Logged in |
| Test Case Input |  |
| Test Case Expected Output | True |
| Test Case Steps | New Order  New OrderDAL  Call create order method  Compare output with expected output |
| Default Value Preserving |  |

3.2 Create order test 2.

|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | CreateOrderTest2() |
| Test Case Description | Create an order with at least one book and customer information |
| Preconditions | Book must exist in the database.  Logged in |
| Test Case Input |  |
| Test Case Expected Output | False |
| Test Case Steps | New Order  New OrderDAL  Call creates order method.  Compare output with expected output |
| Default Value Preserving |  |

3.3 Create order test 3.

|  |  |
| --- | --- |
| Test Case Number | 003 |
| Test Case Name | CreateOrderTest3() |
| Test Case Description | Create an order with at least one book and customer information |
| Preconditions | Book must exist in the database.  Logged in |
| Test Case Input |  |
| Test Case Expected Output | True |
| Test Case Steps | New Order  New OrderDAL  Call creates order method.  Compare output with expected output |
| Default Value Preserving |  |

3.4 Create order test 4.

|  |  |
| --- | --- |
| Test Case Number | 004 |
| Test Case Name | CreateOrderTest4() |
| Test Case Description | Create an order with at least one book and customer information |
| Preconditions | Book must exist in the database.  Logged in |
| Test Case Input |  |
| Test Case Expected Output | False |
| Test Case Steps | New Order  New OrderDAL  Call creates order method.  Compare output with expected output |
| Default Value Preserving |  |

3.5 Create order test 5.

|  |  |
| --- | --- |
| Test Case Number | 005 |
| Test Case Name | CreateOrderTest5() |
| Test Case Description | Create an order with at least one book and customer information |
| Preconditions | Book must exist in the database.  Logged in |
| Test Case Input |  |
| Test Case Expected Output | True |
| Test Case Steps | New Order  New OrderDAL  Call creates order method.  Compare output with expected output |
| Default Value Preserving |  |

3.6 Create order test 6.

|  |  |
| --- | --- |
| Test Case Number | 006 |
| Test Case Name | CreateOrderTest6() |
| Test Case Description | Create an order with at least one book and customer information |
| Preconditions | Book must exist in the database.  Logged in |
| Test Case Input |  |
| Test Case Expected Output | True |
| Test Case Steps | New Order  New OrderDAL  Call creates order method.  Compare output with expected output |
| Default Value Preserving |  |

3.7 Create order test 7.

|  |  |
| --- | --- |
| Test Case Number | 007 |
| Test Case Name | CreateOrderTest7() |
| Test Case Description | Create an order with at least one book and customer information |
| Preconditions | Book must exist in the database.  Logged in |
| Test Case Input |  |
| Test Case Expected Output | True |
| Test Case Steps | New Order  New OrderDAL  Call creates order method.  Compare output with expected output |
| Default Value Preserving |  |

4. Test view revenue of staff in day

4.1 Test view revenue of staff in day 1

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | GetOrderByStaffID() |
| Test Case Description | Display the list of orders that the staff created in the day |
| Preconditions | order has been created and is in the database.  Logged in |
| Test Case Input |  |
| Test Case Expected Output | List order |
| Test Case Steps | New OrderDAL  Call creates order method.  Compare output with expected output |
| Default Value Preserving |  |

4.2 Test view revenue of staff in day 2

|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | GetOrderByStaffID() |
| Test Case Description | Display the list of orders that the staff created in the day |
| Preconditions | order has been created and is in the database.  Logged in |
| Test Case Input |  |
| Test Case Expected Output | Null |
| Test Case Steps | New OrderDAL  Call creates order method.  Compare output with expected output |
| Default Value Preserving |  |

4.3 Test view revenue of staff in day 3

|  |  |
| --- | --- |
| Test Case Number | 003 |
| Test Case Name | GetOrderByStaffID() |
| Test Case Description | Display the list of orders that the staff created in the day |
| Preconditions | order has been created and is in the database.  Logged in |
| Test Case Input |  |
| Test Case Expected Output | List order |
| Test Case Steps | New OrderDAL  Call creates order method.  Compare output with expected output |
| Default Value Preserving |  |

# Task Assign (to each team member)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| H&T  Team | Book Shop Create Order System | | | | | |
| No | Task name | Description | Start Date | End Date | Member | Self-assessment |
| 1 | Write report | Write report on the Bookshop project | 24th June | 25th June | Hiep, Tuan |  |
| 2 | Use case, Activity, Sequence, class diagram |  | 26th June | 4th July | Tuan, Hiep |  |
| 3 | UI design | Using Spectre.Console to create a friendly UI. | 4th July | 5th July | Hiep |  |
| 4 | Database | Data MySQL | 4th July | 6th July | Tuan, Hiep |  |
| 5 | ERD | Entity Relationship Diagram | 7th July | 9th July | Tuan |  |
| 6 | Login ,Create order, Payment | Using MD5 for more protection | 10th July | 13th July | Hiep |  |
| 7 | Code search book by book code | Update search book by book code | 15th July | 17th July | Tuan |  |
| 8 | Code view revenue staff in day |  | 20th July | 23rd July | Hiep |  |
| 9 | Test case | Test main feature | 26th July | 29th July | Tuan |  |

# VI.Installation Instructions

**INSTALLATION INSTRUCTIONS FOR BOOK SHOP CREATE ORDER SYSTEM APPLICATION**

System Requirements :

* Operating System: Windows
* Version (.Net): .Net7.0

1. Deployment Diagram

A diagram of a data transfer

Description automatically generated

1. Installation steps
   * Database Install

* A screenshot of a computer

  Description automatically generatedDownload MySQL Community Sever in website <https://dev.mysql.com/downloads/mysql/>

* Choose the version suitable for your device and operating system and press Download.
* Extract the file and perform the installation process.
* Click "Next" in turn to navigate through the installation steps. You can change the installation options like installation directory, port, root password, etc.
* When the installation is complete, the installer will provide the option to configure MySQL and start the MySQL service.
* Open Command Prompt and run the command: mysql --version to check the installed MySQL version.
  + Application install.
* Click on this link <https://drive.google.com/file/d/1GvQ0LGIjb-EBSSNjHBOQ53fDFKwgmcMJ/view?usp=sharing>

and download BookShopCreateOrderSystem.rar.

* After downloaded, double-click the BookShopCreateOrderSystem.rar and copy the folder's path.
* Open Command Port and use the following command:

cd <paste the copied folder path>

mysql -u root -p < BookShopCreateOrderSystem.sql

then enter your password of root account in MySQL

* Go back to the folder, right click on BookShopCreateOrderSystem.exe and select Send to > Desktop(create shortcut) then go to desktop and use application