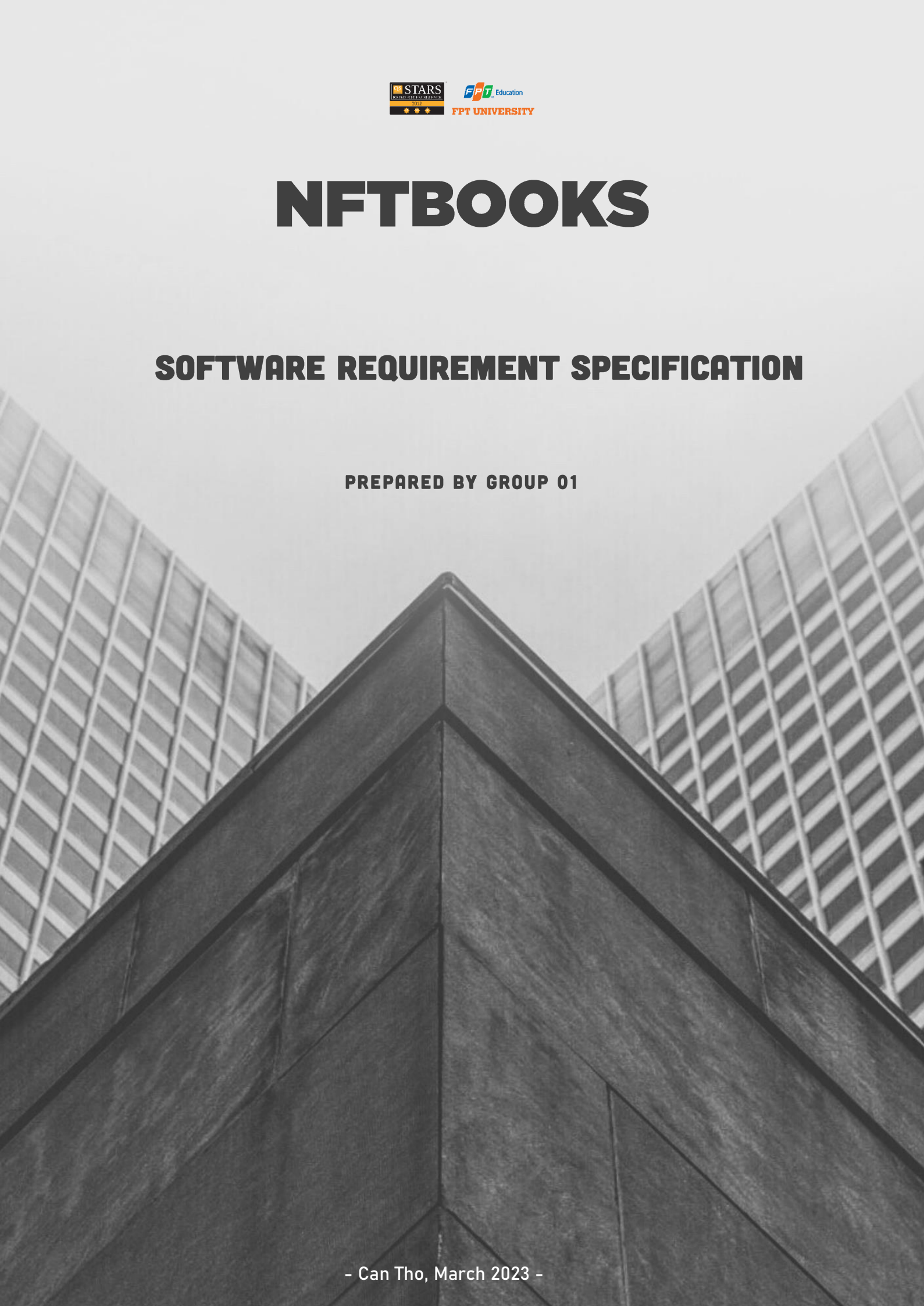
FPT UNIVERSITY

NFTBOOKS

SOFTWARE REQUIREMENT SPECIFICATION

PREPARED BY GROUP 01



- Can Tho, March 2023 -

Table of Contents

I. Overview 4

1. Code Packages/Namespaces 4

2. Database schema 4

II. Code Designs 9

1. Login 9

a. Class Diagram 9

b. Class Specifications 9

c. Sequence Diagram(s) 11

d. Database queries 12

2. REGISTER 12

a. Class Diagram 12

b. Class Specifications 12

c. Sequence Diagram 14

d. Database queries 15

3. SEARCH 15

a. Class Diagram 15

b. Class Specifications 15

c. Sequence Diagram 16

d. Database queries 17

4. BOOK MANAGE 17

a. Class Diagram 17

b. Class Specifications Data.BookCoinConnect 18

c. Sequence Diagram 20

d. Database queries 20

5. USER MANAGE 21

a. Class Diagram 21

b. Class Specifications 22

c. Sequence Diagram 23

d. Database queries 23

6. VIEW DETAIL BOOK 24

a. Class Diagram 24

b. Class Specifications 25

c. Sequence Diagram 26

d. Database queries 27

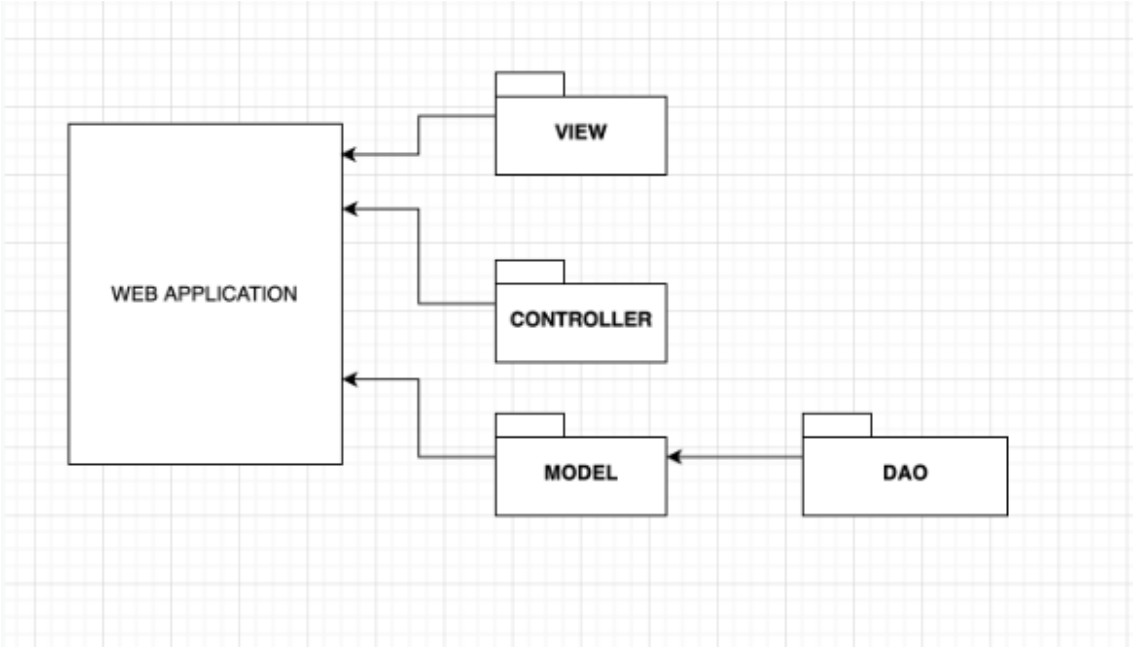
7. VIEW PERSONAL INFORMATION 27
a. Class Diagram 28
b. Class Specifications 28
c. Sequence Diagram 29
8. CHANGE PERSONAL INFORMATION 31
a. Class Diagram 31
b. Class Specifications 31
c. Sequence Diagram 34
d. Database queries 34
9. VIEW HISTORY BUYING 35
a. Class Diagram 35
b. Class Specifications 36
c. Sequence Diagram 37
d. Database queries 38
10. VIEW CART 39
a. Class Diagram 39
b. Class Specifications 40
c. Sequence Diagram 42
d. Database queries 42
11. ADD BOOK TO CART 43
a. Class Diagram 43
b. Class Specifications 44
c. Sequence Diagram 45
d. Database queries 45
12. VIEW DETAIL ORDER 46
a. Class Diagram 46
b. Class Specifications 47
c. Sequence Diagram 49
d. Database queries 49
13. ENTER SHIPPING INFORMATION 50
a. Class Diagram 50
b. Class Specifications 51
c. Sequence Diagram 52
d. Database queries 53
14. CANCEL THE ORDER 53

a. Class Diagram 54
b. Class Specifications 55
c. Sequence Diagram 57
d. Database queries 57
15. ORDER APPROVAL 58
a. Class Diagram 58
b. Class Specifications 58
c. Sequence Diagram 60
d. Database queries 61
16. VIEW AND EDIT INFORMATION OF MANAGER, CUSTOMER, BOOK 62
a. Class Diagram 62
b. Class Specifications 63
c. Sequence Diagram 65
d. Database queries 65
III. Database Tables 66
1. Category 66
2. Contact 66
3. Feed_Back 66
4. Footer 67
5. KhoHang 67
6. Menu 67
7. MenuType 67
8. Messenger 67
9. Muon_Tra 68
10. NhaCungCap 68
11. NhapHang 68
12. Order_Detail 69
13. Orders 69
14.Quyen 69
15. SanPham 69
16.Order_Detail 70
--The End-- 70

I. Overview

1. Code Packages/Namespaces

Provide the package diagram for each sub-system. The content of this section including the overall package diagram, the explanation, package and class naming conventions in each package. Please see the sample and description table format



below – following Java project naming convention.

Figure 1. Code packages/Namespaces

Package descriptions & package class naming conventions

No	Package	Description
01	VIEW	Display model data to the user and also enables them to modify them
02	CONTROLLER	Handles the user request. The user uses the view and raises an HTTP request, which will be handled by the controller
03	MODEL	Represents the shape of the data(class or object)
04	DAO	Handles everything update to database(SQL Server 2019)

2. Database schema

Provide the tables relationship – following SQL Server database naming convent

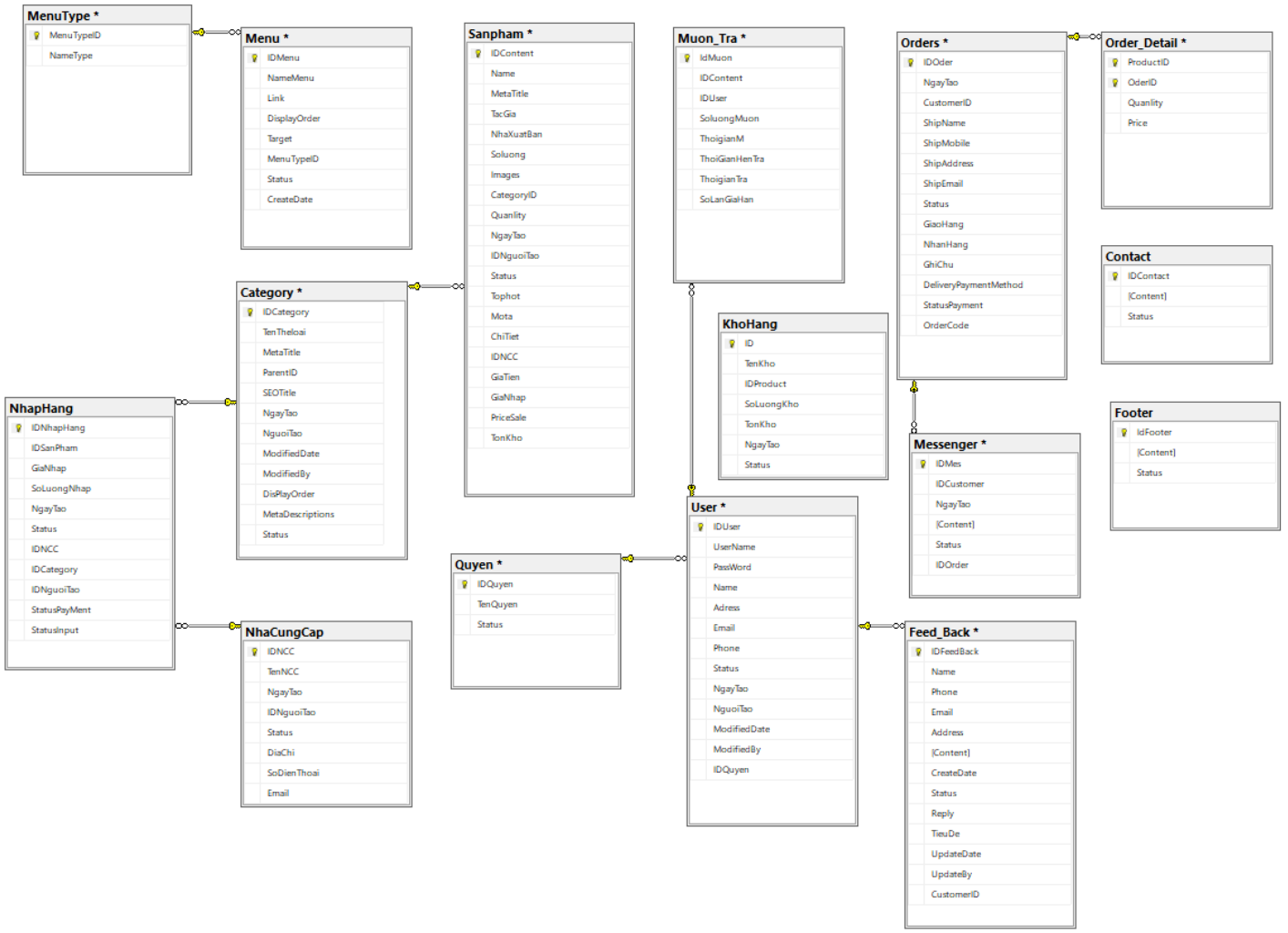


Figure 2. Database Schema

Table descriptions & package class naming conventions are as below

No	Table	Description
01	Category	Category has the task of classifying products/services, linking content/websites with common themes. • Primary keys: IDCategory<bigint> Foreign key: none
02	Contact	This is the table that contains all the information about the book. • Primary keys: IDFeedBack<bigint> Foreign key: none
03	Feed_Back	This is the table to store all customer information • Primary keys: IDFeedBack<bigint> Foreign key: none
04	Footer	This is the table Save web footer information • Primary keys: IdFooter< bigint> Foreign key: none
05	KhoHang	This is the table inventory management • Primary keys: ID< bigint> Foreign key: none
06	Menu	this is the table Save web menu information • Primary keys: IDMenu< bigint> Foreign key: none
07	MenuType	This is the table Save web menu location information • Primary keys: MenuTypeID< bigint> Foreign key: none
08	Messenger	This is the table Store supplier information • Primary keys: IDMes<bignint>

		Foreign key: none
09	Muon_Tra	This is the user's return request table • Primary keys: IdMuon< bigint > Foreign key: none
10	NhaCungCap	this is the table Save supplier information • Primary keys: IDNCC< bigint > Foreign key: none
11	NhapHang	This table is to Save information about goods to be imported. • Primary keys: IDNhapHang< bigint> Foreign key: none
12	Order_Detail	this table is to Save order details • Primary keysProductID < bigint > Foreign key: OderID< bigint >
13	Orders	This table is to Save order information • Primary keys: IDOder < bigint > Foreign key: none
14	Quyen	This table is to Saves permission information. • Primary keys: IDQuyen < bigint > Foreign key: none
15	Sanpham	This table is to Save book information • Primary keys: IDContent < bigint > Foreign key: none
16	User	This table is to Store customer information • Primary keys: IDUser < bigint > Foreign key: none

II. Code Designs

Convention:

Status order:
0: cancel
1: ordered
2: order confirmed
3: received the goods

Status customer:
0: admin
1: manager
2: customer

1. Login

a. Class Diagram

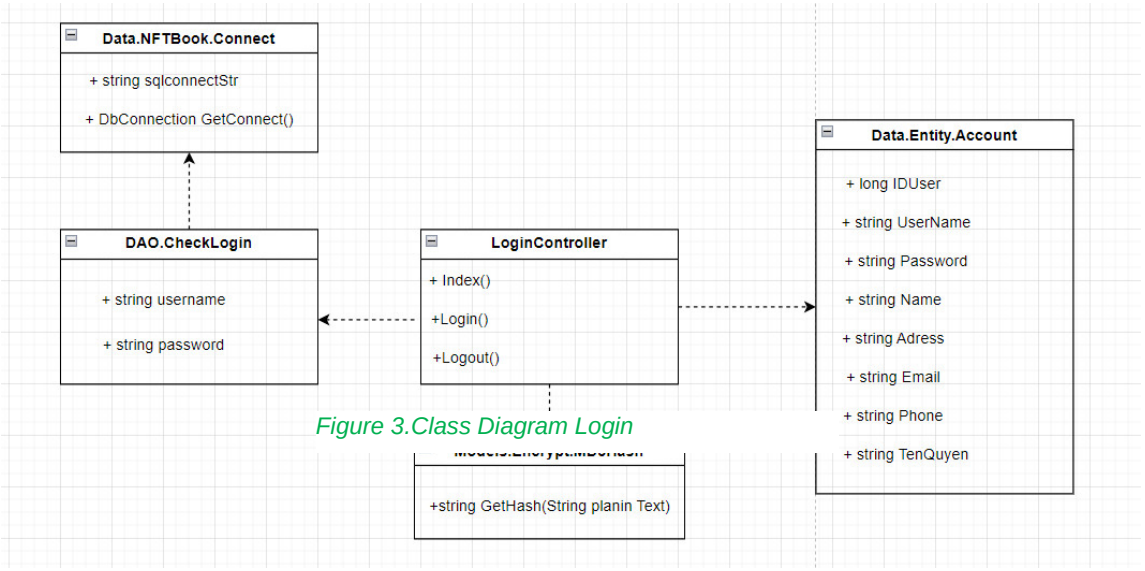


Figure 3.Class Diagram Login

b. Class Specifications

LoginController

No	Method	Description
01	Index()	function return login page view Input: None Output: Login page
02	Logout()	function that redirects access to the logout page Input: None Output: Logout page
03	Login()	function of checking login information and redirecting user access when logging into the system Input: username, password Output: log-in results and access navigation

Models.Encrypt.MD5Hash

No	Method	Description
01	GetHash()	function to encrypt user's login password according to MD5 standard Input: Password Output: MD5 encrypted password

DAO.CheckLogin

No	Method	Description
01	CheckLogin()	Function to check and match the user's login information when logging in and the login information in the database Input: username, MD5 encrypted password Output: user account information available on the system

Data.BookConnect

No	Method	Description
01	GetConnect()	function to connect to the system database Input: None Output: database connection command

c. Sequence Diagram(s)

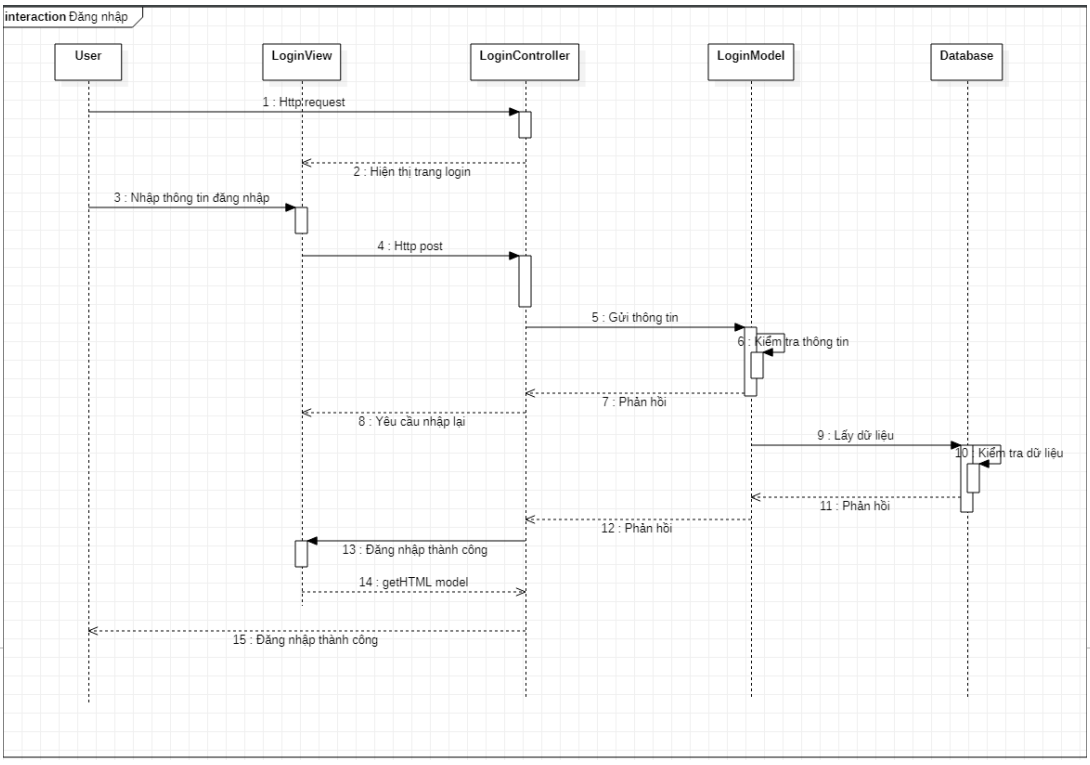


Figure 4.Sequence Diagram Login

d. Database queries

- Login

SELECT [UserName],[PassWord] FROM [QLTV_BTL].[dbo].[User]

2. REGISTER

a. Class Diagram

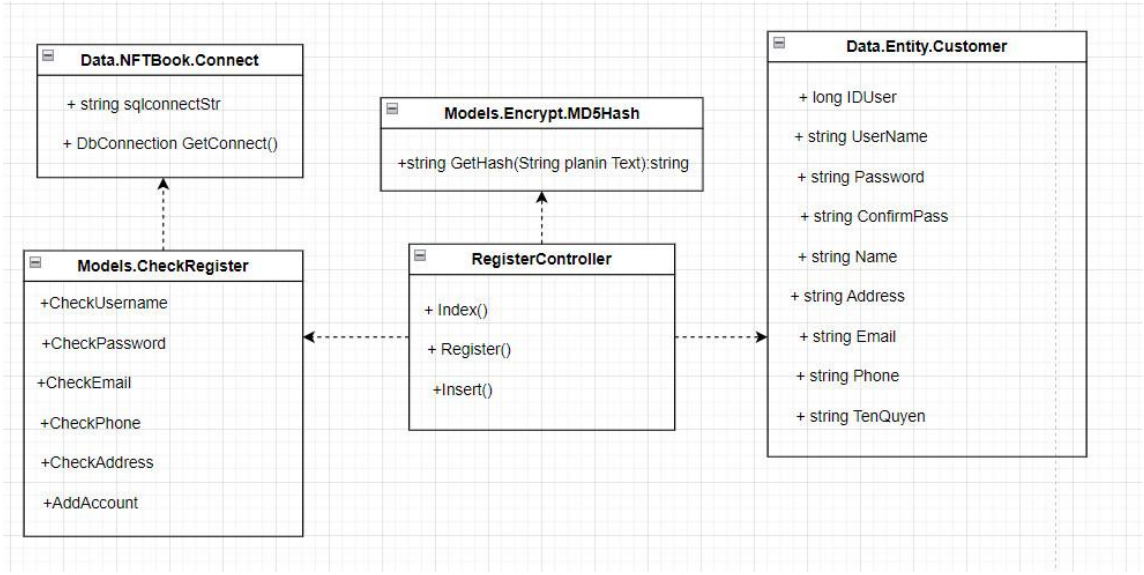


Figure 5. Class Diagram Register

b. Class Specifications

RegisterController

No	Method	Description
01	Index()	function return login page view Input: None Output: Login page

02	Register()	function transfer information through the registration form for users to fill in the missing information by get method Input: name, email, dob, password, repassword. Output: Message register success and redirect login page
03	Insert()	insert is used to add a new row of data to a table in the database. Input: None Output: information of account

Data.BookCoinConnect

No	Method	Description
01	GetConnect()	function to connect to the system database Input: None Output: database connection command

Models.CheckRegister

No	Method	Description
01	CheckRegister ()	check registration information exists or not Input: Identifier Output: True/False

03	AddAccount(s)	new account creation function Input: acc - information account– method register Output: None

Models.EnCrypt.MD5Hash

No	Method	Description
01	GetHash()	function to encrypt user's login password according to MD5 standard Input: Passwword Output: MD5 encrypted password

c. Sequence Diagram

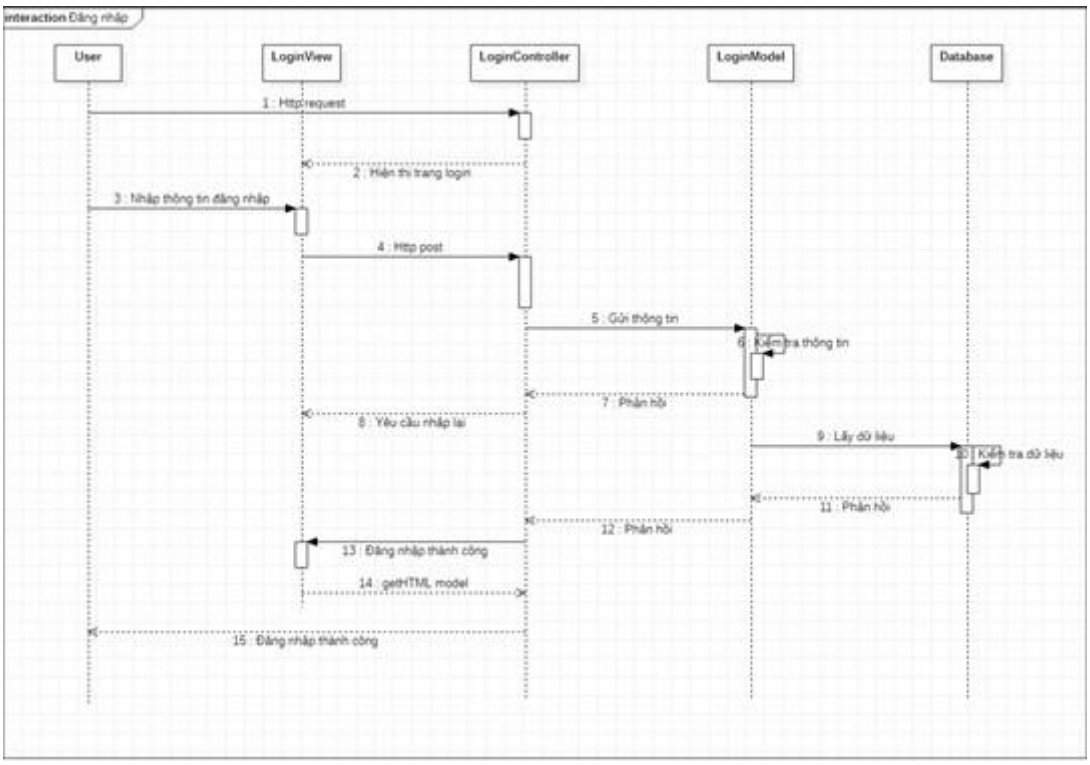


Figure 6. Sequence Diagram Register

d. Database queries

- Register

```
INSERT INTO [dbo].[User] ([IDUser], [UserName], [PassWord], [Name], [Adress], [Email], [Phone], [Status], [NgayTao], [IDQuyen])
VALUES (ID_bigint, UserName_input, Password_input, Name_input, adress_input, mail_input, phone_input, 1, date_now, 2);
```

3. SEARCH

a. Class Diagram

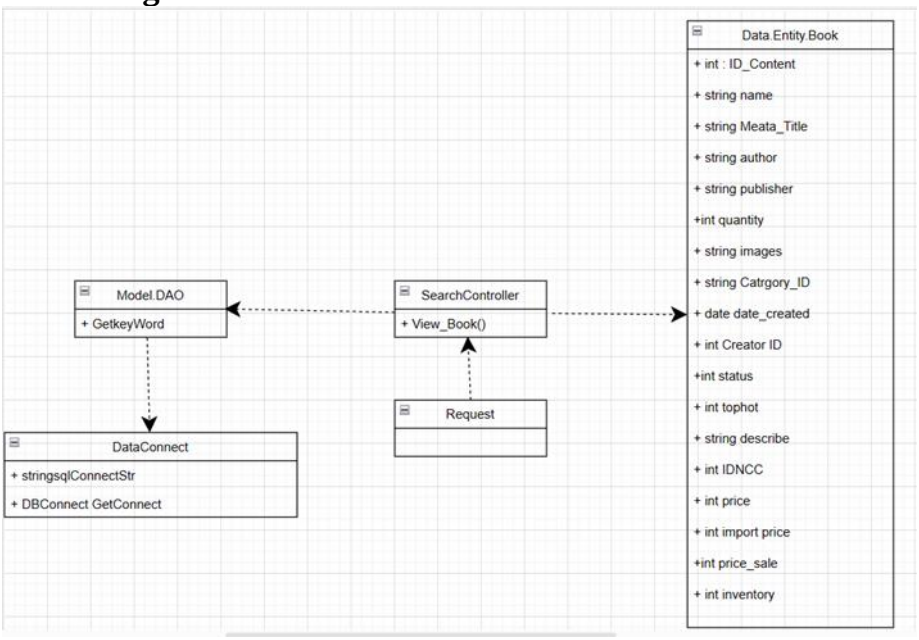


Figure 7. Class Diagram Search

b. Class Specifications

Data.BookCoinConnect

No	Method	Description
01	GetConnect()	function to connect to the system database Input: None Output: database connection command

SearhController

No	Method	Description
01	Index()	function return searh view page Input: None Output: view page

Model.DAO

No	Method	Description
01	GetBook(book_id)	Function get list of infor book Input: book_id Output: database connection command

c. Sequence Diagram

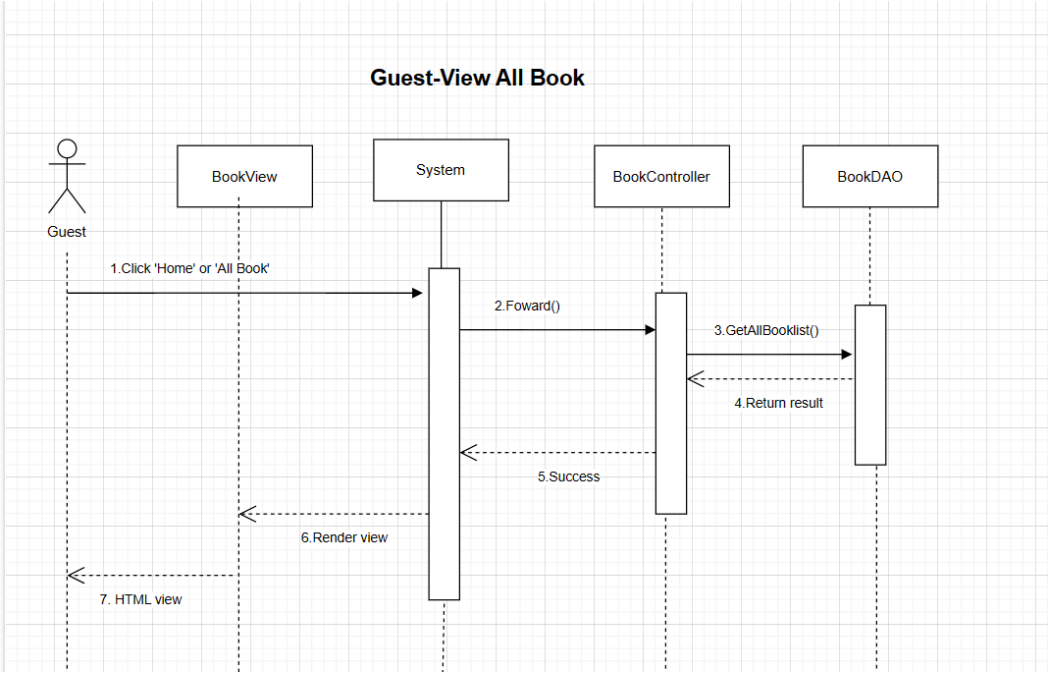


Figure 8. Sequence Diagram Search

d. Database queries

- Search

```
SELECT *
FROM Sanpham WHERE Sanpham.Name LIKE N'%name_input%';
```

4. BOOK MANAGE

a. Class Diagram

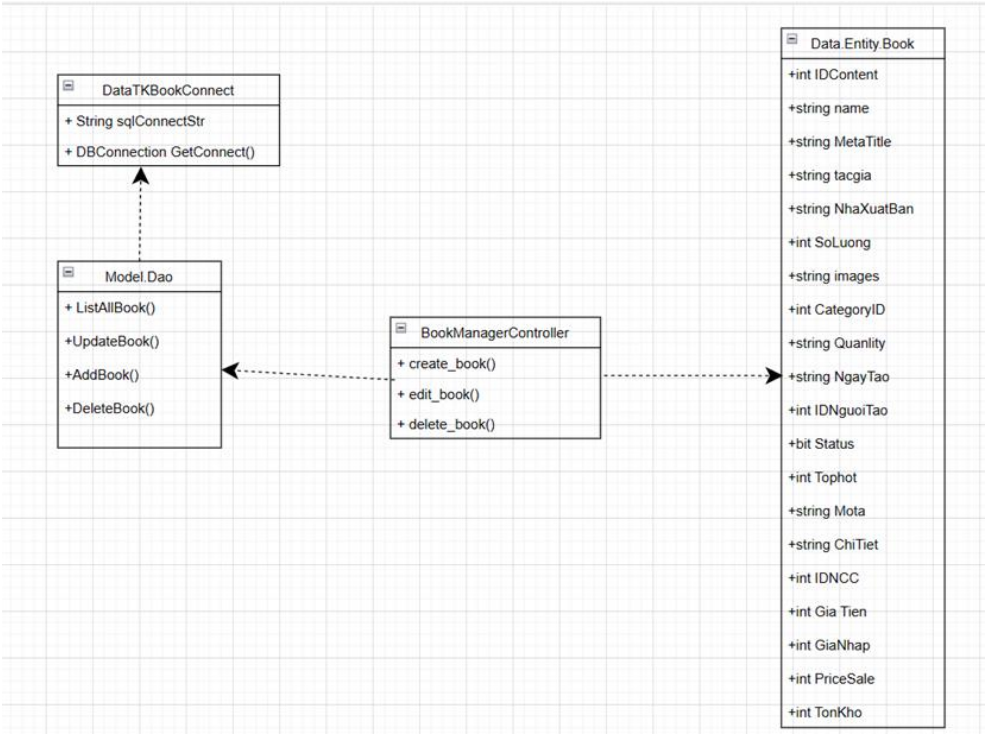


Figure 9. Class Diagram Book Manage

b. **Class Specifications**

Data.BookCoinConnect

No	Method	Description
01	GetConnect()	function to connect to the system database Input: None Output: database connection command

BookManagerController

No	Method	Description
01	Index()	function return book management view page Input: None Output: view page

Models.DAO

No	Method	Description
01	ListAllBook()	function get list all ebook Input: book_id,name, price, category,detail_book Output: list ebooks

02	UpdateBook()	Function to update ebook Input: book_id,name,author_id,price, category, size,detail_book Output: None
03	AddBook();	Function to add new ebook Input:book_id,name,author_id,price, category, size,detail_book Output: None
04	Delete(int book_id)	Function to delete ebook Input: B_ID Output: None

CustomerManagerController

No	Method	Description
01	Index()	function return customer management view page Input: None Output: view page

c. Sequence Diagram

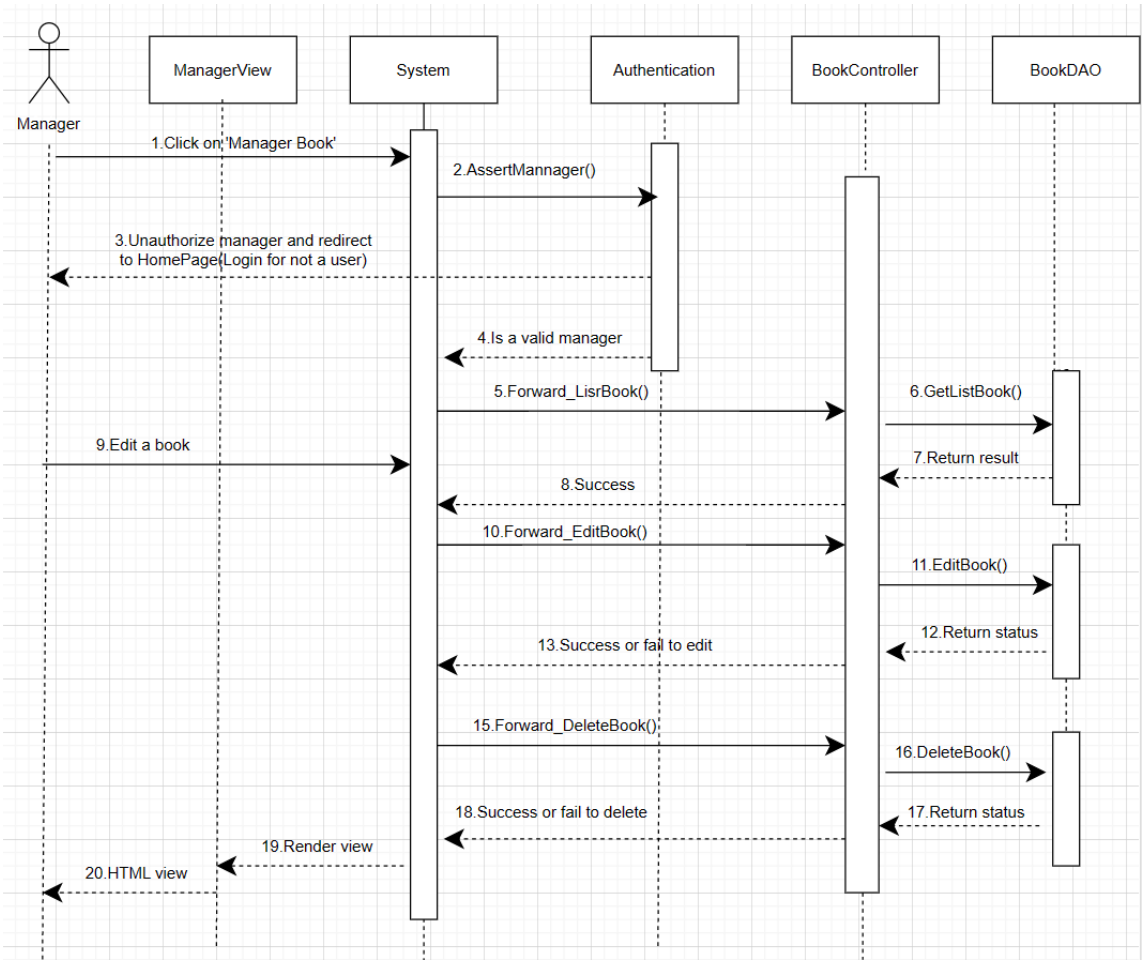


Figure 10. Sequence Diagram Book manage

d. Database queries

- **View all book**

```
SELECT * FROM Sanpham;
```

- **View detail book**

```
SELECT * FROM Sanpham Where IDContent = 'IDContent_input';
```

- **Update book**

```
Update Sanpham
Set
Name = name_input,
MetaTitle = MetaTitle_input,
TacGia = TacGia_input,
NhaXuatBan = NhaXuatBan_input,
Soluong = Soluong_input,
Images = Images_input,
```

CategoryID = CategoryID_input,
Quanlity = Quanlity_input,
Status = Status_input,
Tophot = Tophot_input,
Mota = Mota_input,
ChiTiet = ChiTiet_input,
IDNCC = IDNCC_input,
GiaTien = GiaTien_input,
GiaNhap = GiaNhap_input,
PriceSale = PriceSale_input,
TonKho = TonKho_input
where Sanpham.IDContent = IDContent_input;

- **Add book**

INSERT INTO [dbo].[Sanpham] ([IDContent], [Name], [MetaTitle],
[TacGia], [NhaXuatBan],
[Soluong], [Images], [CategoryID], [Quanlity], [NgayTao], [IDNguoiTao],
[Status], [Tophot],
[Mota], [ChiTiet], [IDNCC], [GiaTien], [GiaNhap], [PriceSale], [TonKho])
VALUES (ID_input, Name_input, title_input, tacgia_input, xuanban_input,
soluong_input, image_input, CatId_input, Quanlity_input, time_input,
IdNguoitao_input, status_input, tophot_input, mota_input, chitiet_input,
IDNCC_input, Gia_input, GIanhap_input, sale_input, tonkho_input);

5. USER MANAGE

a. Class Diagram

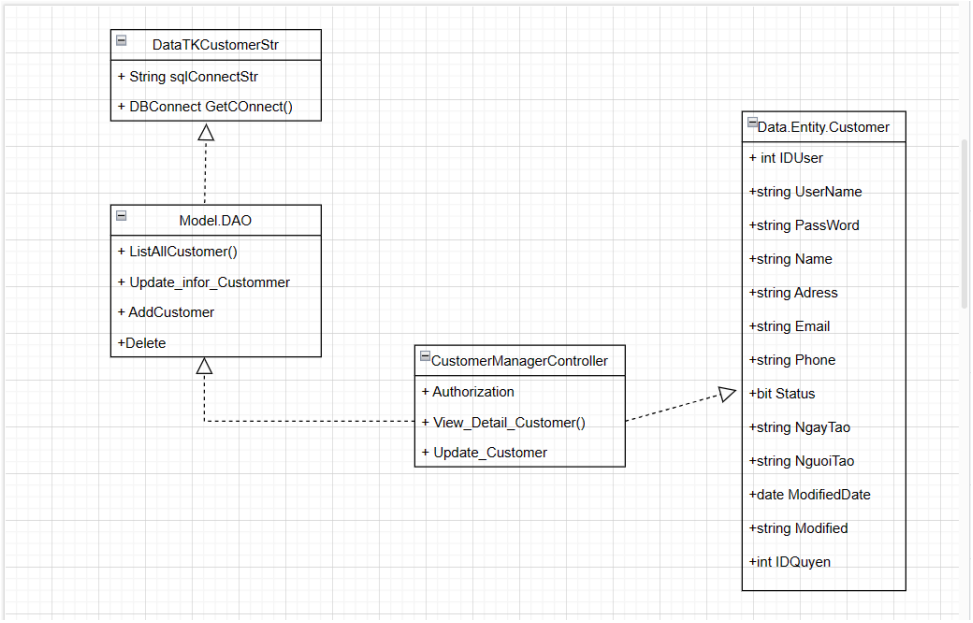


Figure 11. Class Diagram Customer Manage

b. Class Specifications

Models.DAO

No	Method	Description
01	ListAllCustomer	function get list all customer Input: None Output: list customers
02	UpdateInfoCustomer	Function to update information customer Input: name,email,phone_number, password Output: None
03	AddBook	Function to add new customer Input:name,email,phone_number, password Output: None
04	Delete(int book_id)	Function to delete customer Input: Book_ID Output: None

Data.BookCoinConnect

No	Method	Description
01	GetConnect()	function to connect to the system database Input: None Output: database connection command

c. Sequence Diagram

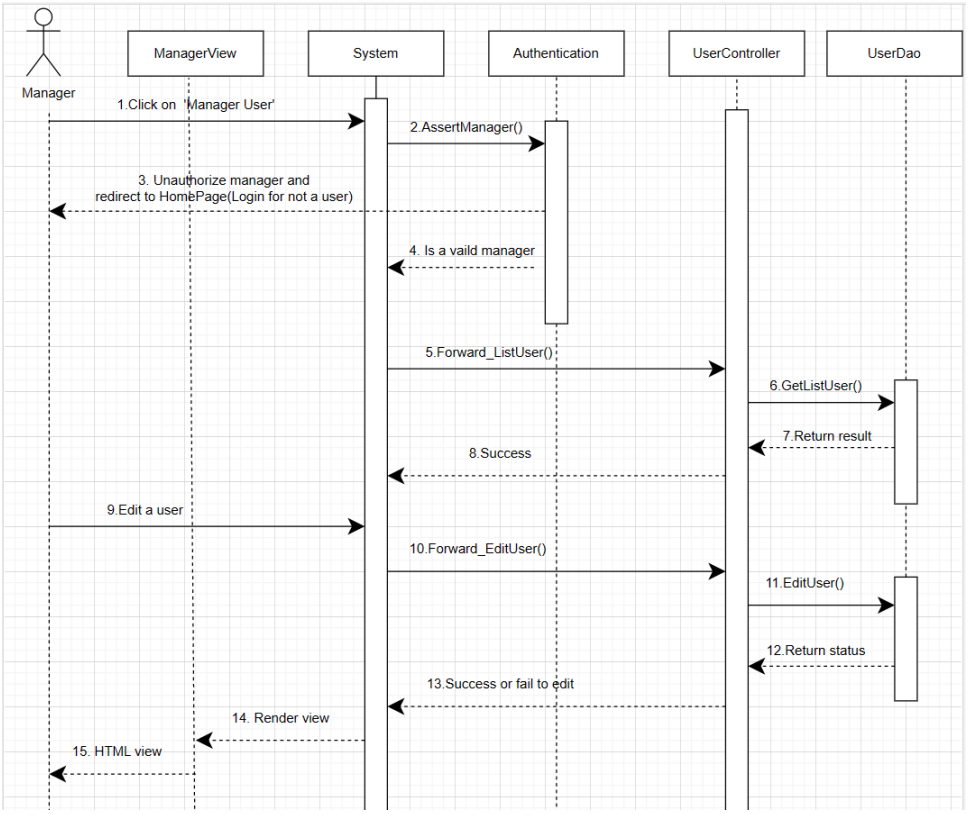


Figure 12. Sequence Diagram Book Manage

d. Database queries

- **View and Edit information Customer**

```
UPDATE [dbo].[User]
SET [User].Name = name_input,
password = password_input,
Name = Name.input,
Adress = Adress_input,
```

Email = Email_input,
Phone = Phone_input,
Where [User].IDUser = IDUser_input;

6. VIEW DETAIL BOOK

a. Class Diagram

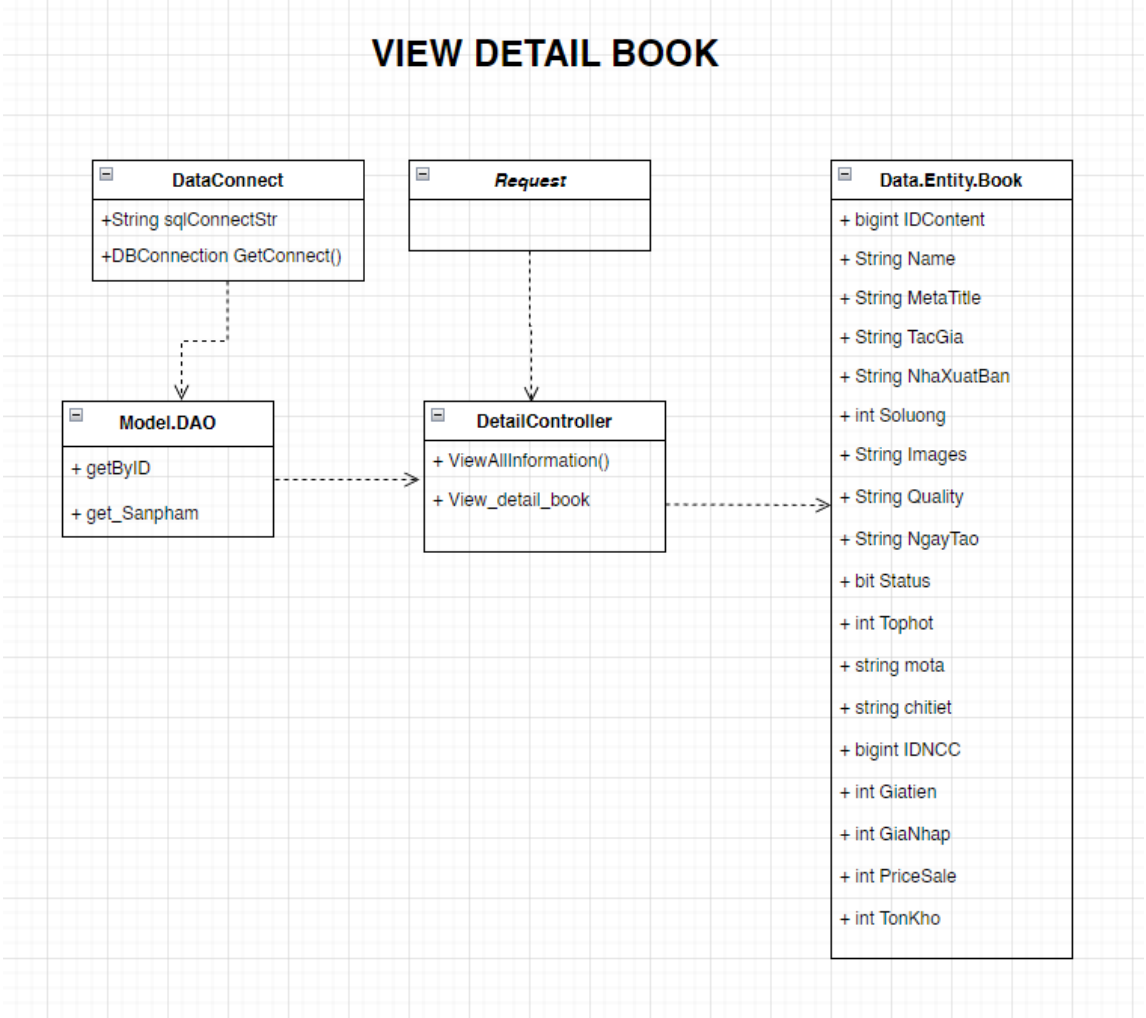


Figure 13. Class Diagram View Detail Book

b. Class Specifications

Data.BookCoinConnect

No	Method	Description
01	GetConnect()	function to connect to the system database Input: None Output: database connection command

DetailController

No	Method	Description
01	ViewAllInformation()	function return all the details of the book view page Input: None Output: view page
02	View_detail_book	function return all the details of the book view page Input: None Output: view page

Models.DAO

No	Method	Description
01	+ get_Sanpham(string name, string MetaTitle, string TacGia, string NhaXuatBan, int Soluong, string Images, int CategoryID, string Quanlity, bool Status, string ChiTiet, string IDNCC, int GiaTien, int GiaNhap, int PriceSale, int TonKho)	function get list all details of the book Input: none Output: list details of book
02	+ getByID	Function to display a detailed list of books Input: none Output: list details of book

c. Sequence Diagram

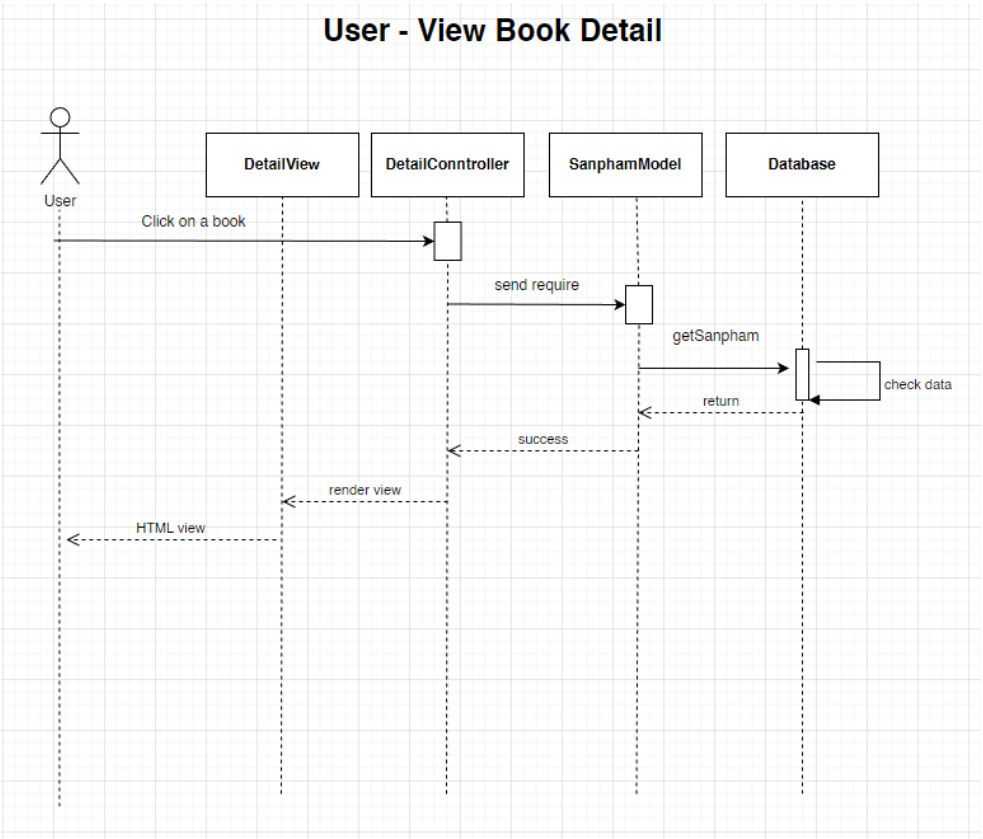


Figure 14. Sequence Diagram View detail book

d. Database queries

- View detail book (comment, rate, author,...)

```
select * from Sanpham
where sanpham.IDContent=id_inpuit;
```

7. VIEW PERSONAL INFORMATION

a. Class Diagram

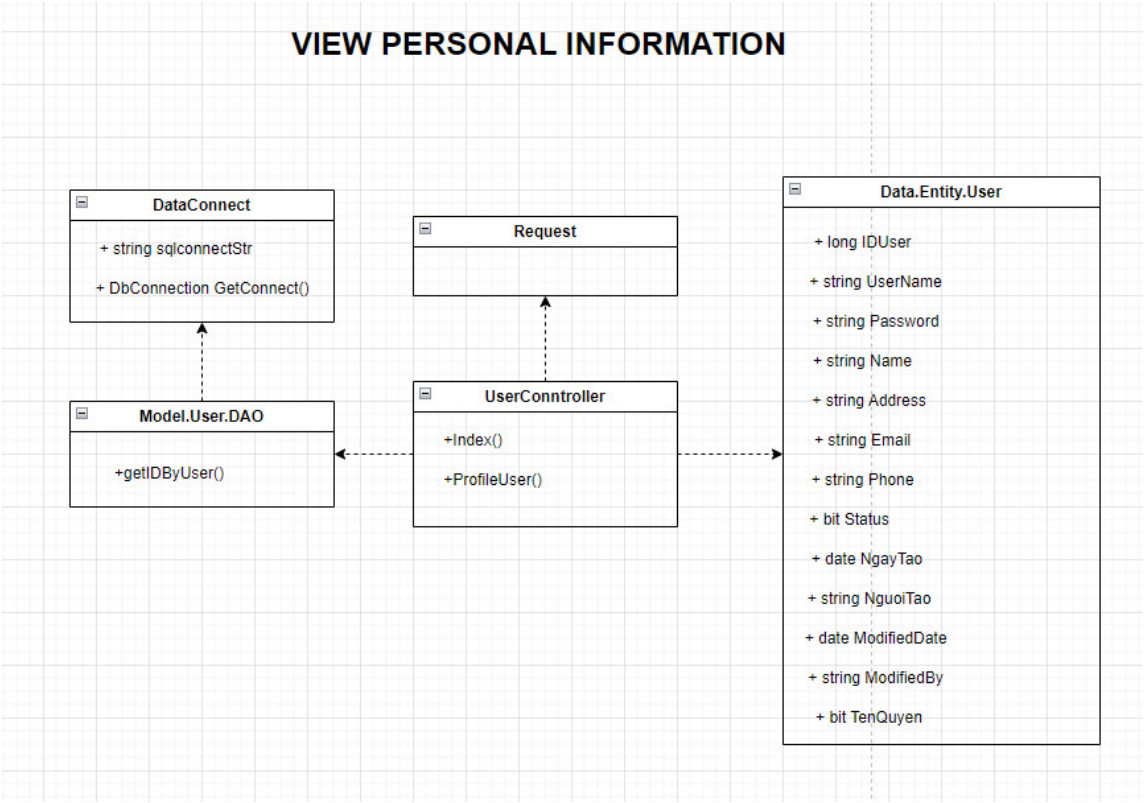


Figure 15. Class Diagram View Personal Infor

b. Class Specifications

Data.Connect

No	Method	Description

01	GetConnect()	function to connect to the system database Input: None Output: database connection command

InformationController

No	Method	Description
01	Index()	function return all the details of the book view page Input: None Output: view page

Model.DAO

No	Method	Description
01	Show_detail_customer(string detail_customer)	Function get list of infor customer Input: detail_customer Output: display list of infor customer

c. Sequence Diagram

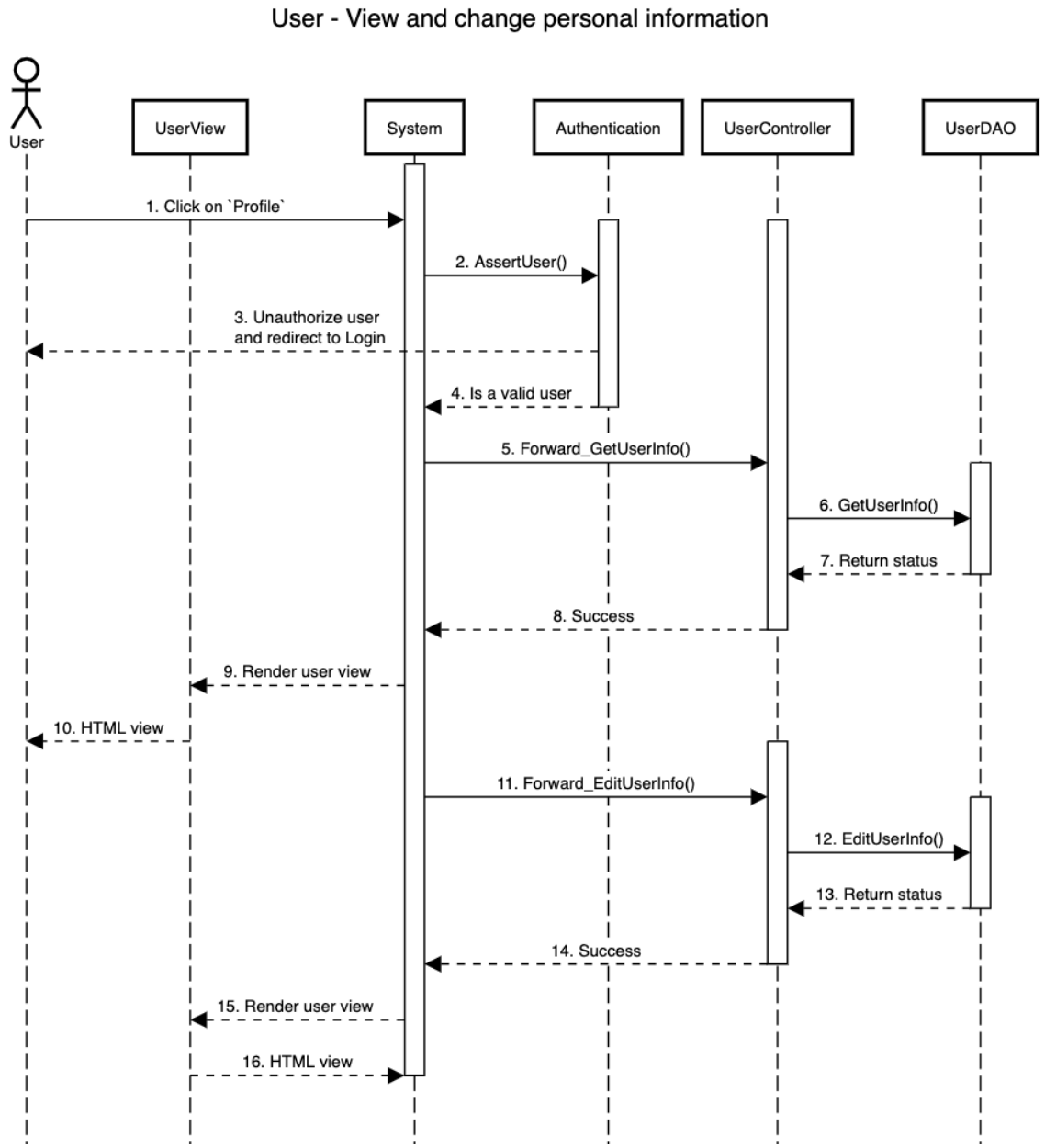


Figure 16. Sequence Diagram View and Change Personal Information

d. Database queries

- View personal information

```
select * from [User]
where [User].IDUser = id_input;
```

8. CHANGE PERSONAL INFORMATION

a. Class Diagram

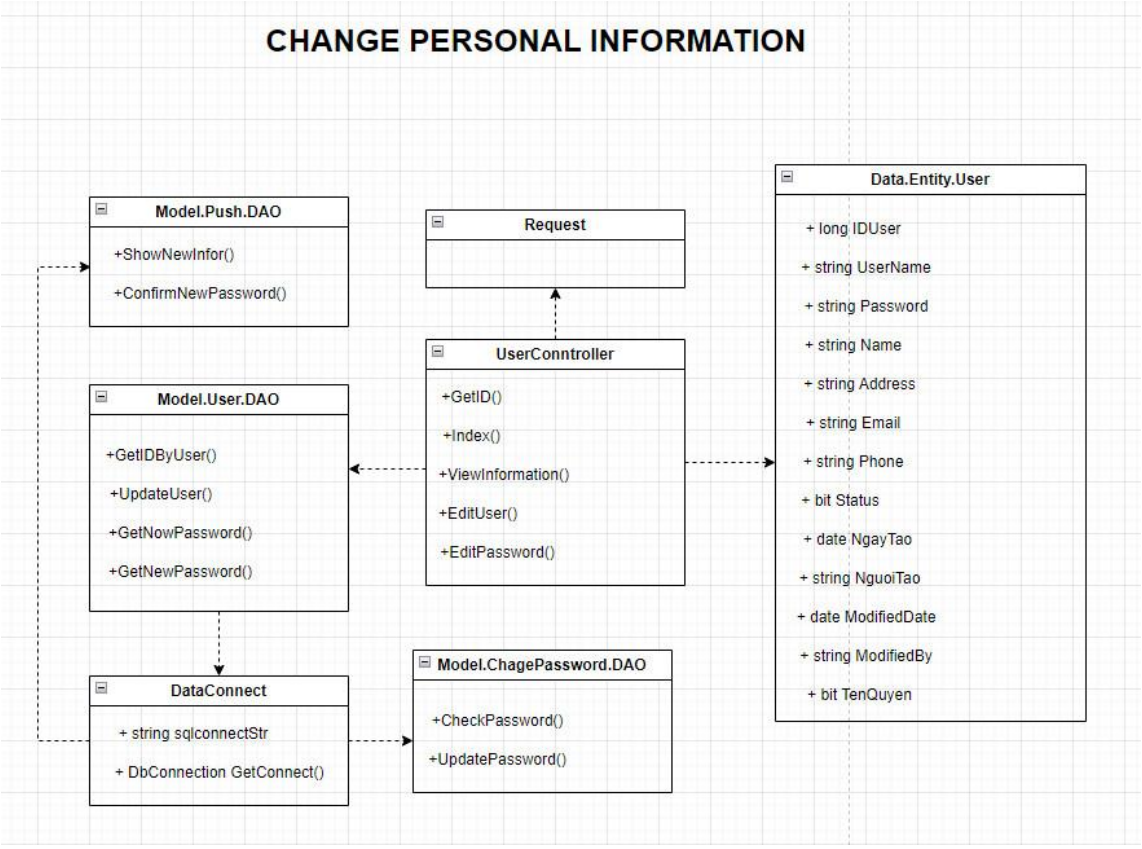


Figure 17. Class Diagram Change Personal Information

b. Class Specifications

Data.BookConnect

No	Method	Description
01	GetConnect()	function to connect to the system database Input: None Output: database connection command

UserController

No	Method	Description
01	GetID()	function get ID Input: User_ID Output: database connection command
02	Index()	function return all information Input: None Output: view page
03	View_Infor()	Function to change information Input : User_ID Output : None
04	Edit User	Function to change information Input : User_ID Output : None
05	Edit pass	Function to change information Input : Pass_ID Output : None

Models.User.DAO

No	Method	Description
01	GetInfor(string name,string email,string phone_number,string password)	function get information Input: User_id Output: information of customer
02	UpdateInfor(name, email,phone_number, password)	Function to update information Input : name,email,phone_number, password Output : None
03	GetNewPassword(string password)	function get new password Input : Password Output:none

c. Sequence Diagram

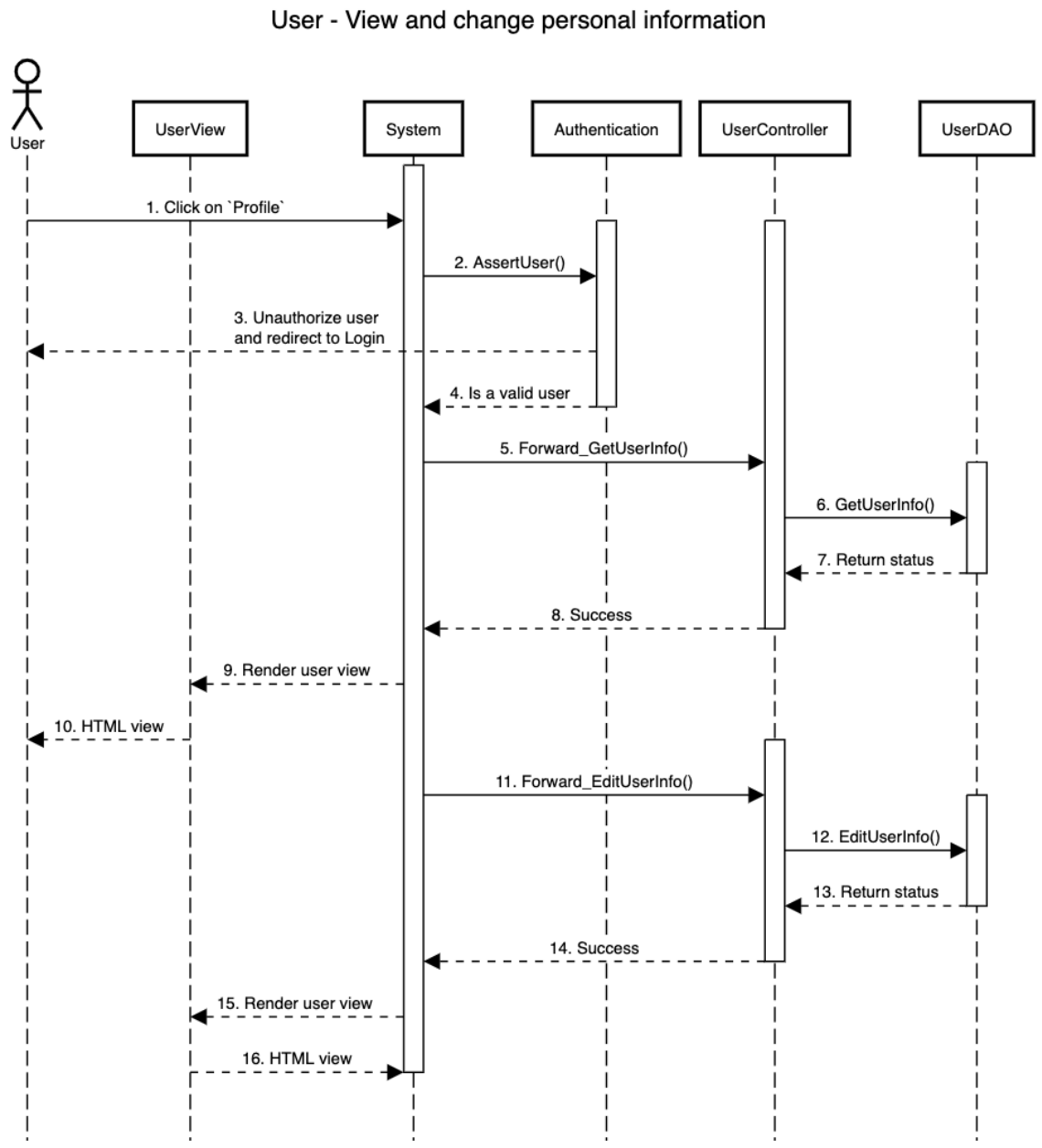


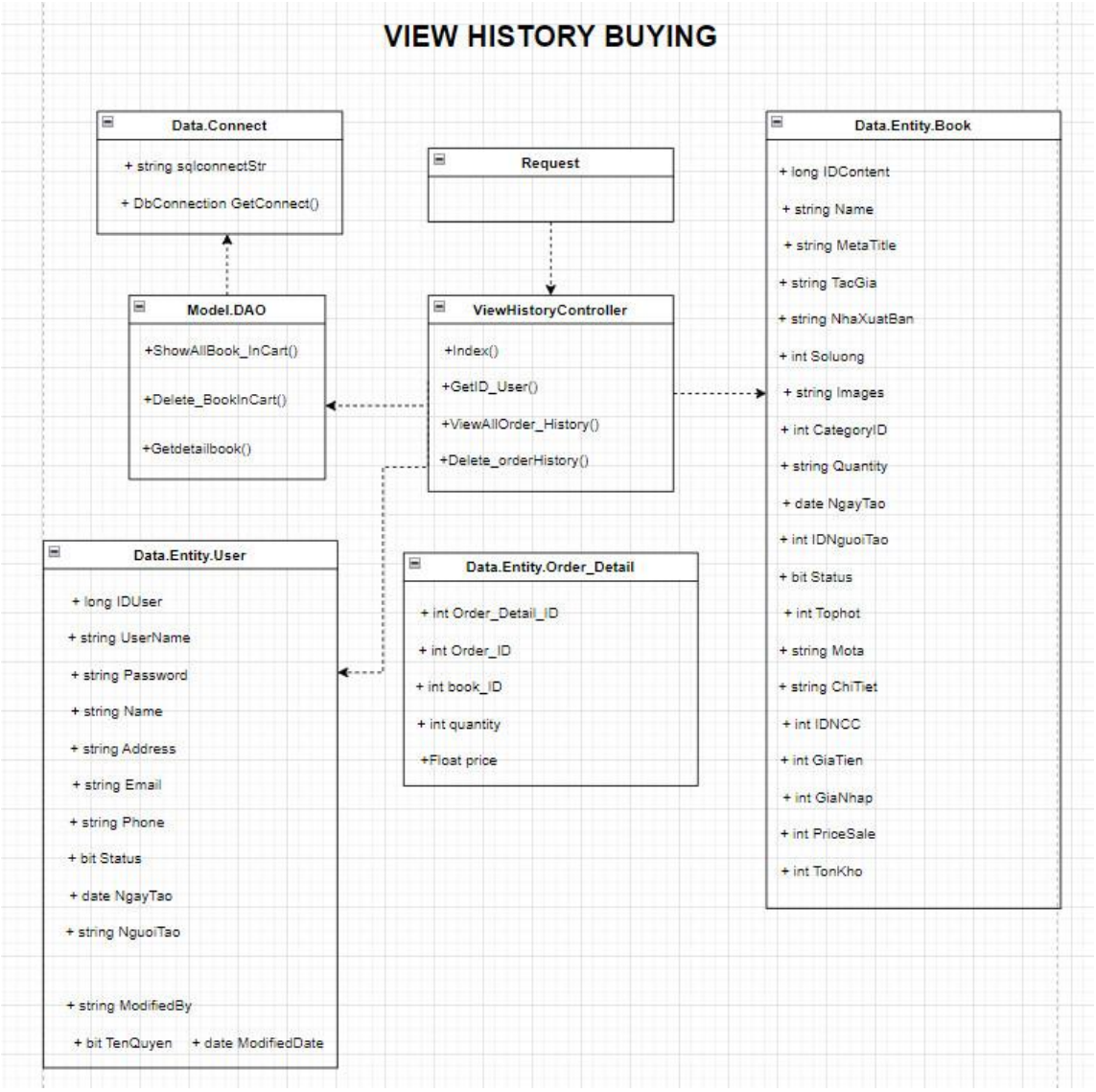
Figure 18. Sequence Diagram View And Change Personal Information

d. Database queries

UPDATE *User*
SET *name = name_input*,phone_number = phone_number_input,
password = password_input, email = mail_input
WHERE *user_id = user_id_input*;

9. VIEW HISTORY BUYING

a. Class Diagram



b. Class Specifications

Data.BookCoinConnect

No	Method	Description
01	GetConnect()	function to connect to the system database Input: None Output: database connection command

ViewHistoryController

No	Method	Description
01	GetID_customer()	function to get ID of customer Input: user_id Output: none

Figure 19. Class Diagram View History Buying

02	ViewAllOrder ()	Function to get all orders Input: order_id Output: none
03	ViewInfor_Cart ()	function to view cart information Input: address Output: display cart information

Models.DAO

No	Method	Description
01	Index()	function return all books are in the order history Input: book_id Output: view page
02	Delete_BookInCart(int book_id)	Function to delete book Input: book_id Output: delete book

c. Sequence Diagram

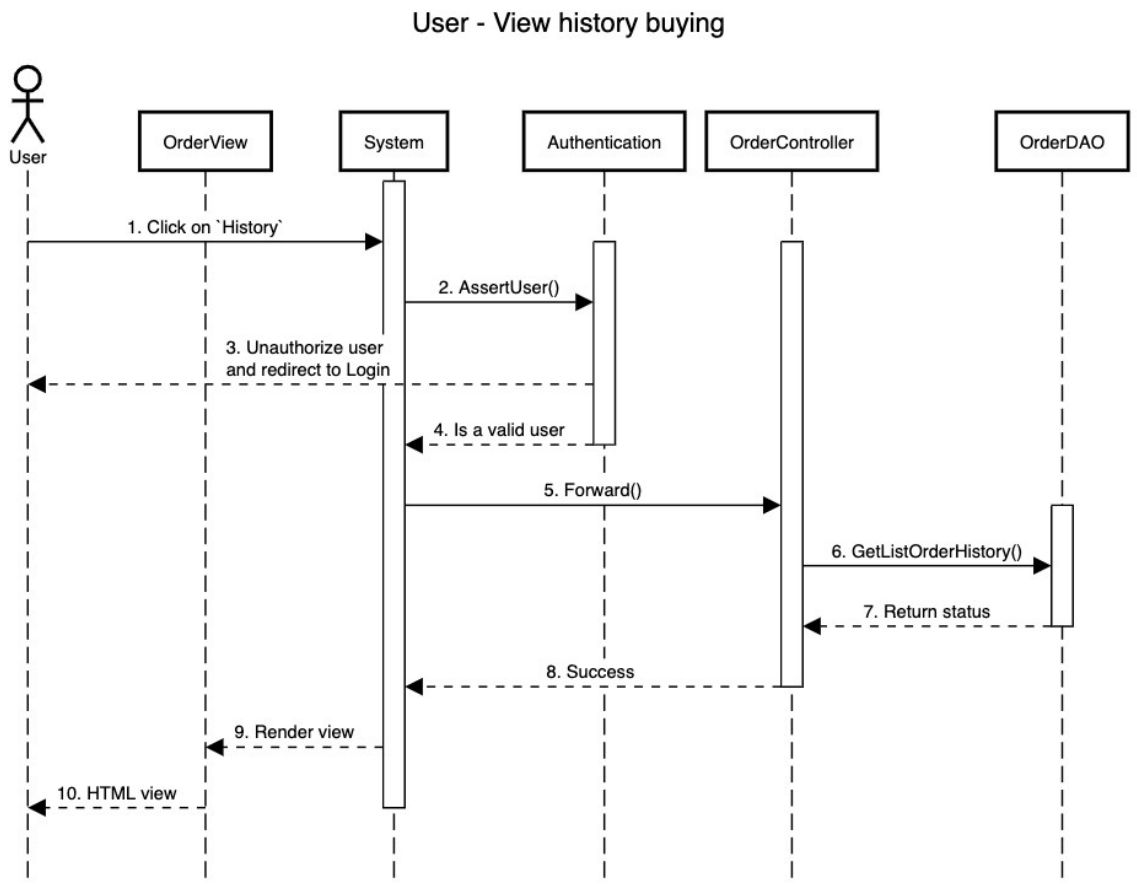


Figure 20. Class Diagram View history buying

d. Database queries

```
select * from Orders
where Orders.IDOder = IDOder_input;
```

10.VIEW CART

a. Class Diagram

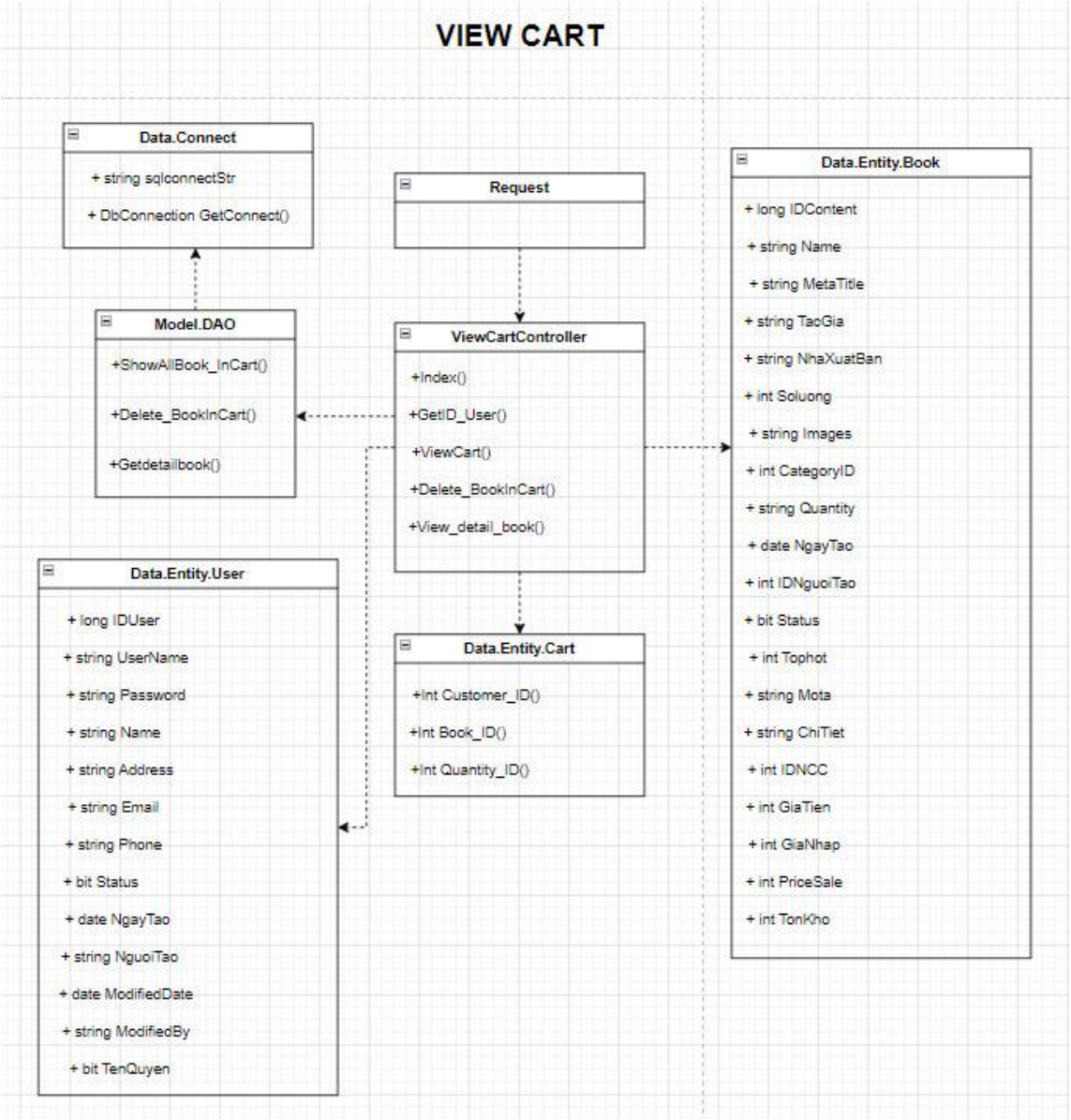


Figure 21. Class Diagram VIew Cart

b. Class Specifications

Data.BookCoinConnect

No	Method	Description
01	GetConnect()	function to connect to the system database Input: None Output: database connection command

ViewCartController

No	Method	Description
01	GetID_customer()	function get customer ID Input: customer_id Output: none
02	Viewcart()	function to see what's in the cart Input: book_id Output : display books in cart
03	Delete_BookInCart	Function to delete book Input: book_id Output: None

Models.DAO

No	Method	Description
01	Index()	function return all books are in the cart Input: book_id Output: view page
02	Delete_BookInCart(int book_id)	Function to delete book Input: book_id Output: delete book
03	GetdetailBook(string detail_book)	function to get details about a book Input : detail_book Output : none

c. Sequence Diagram

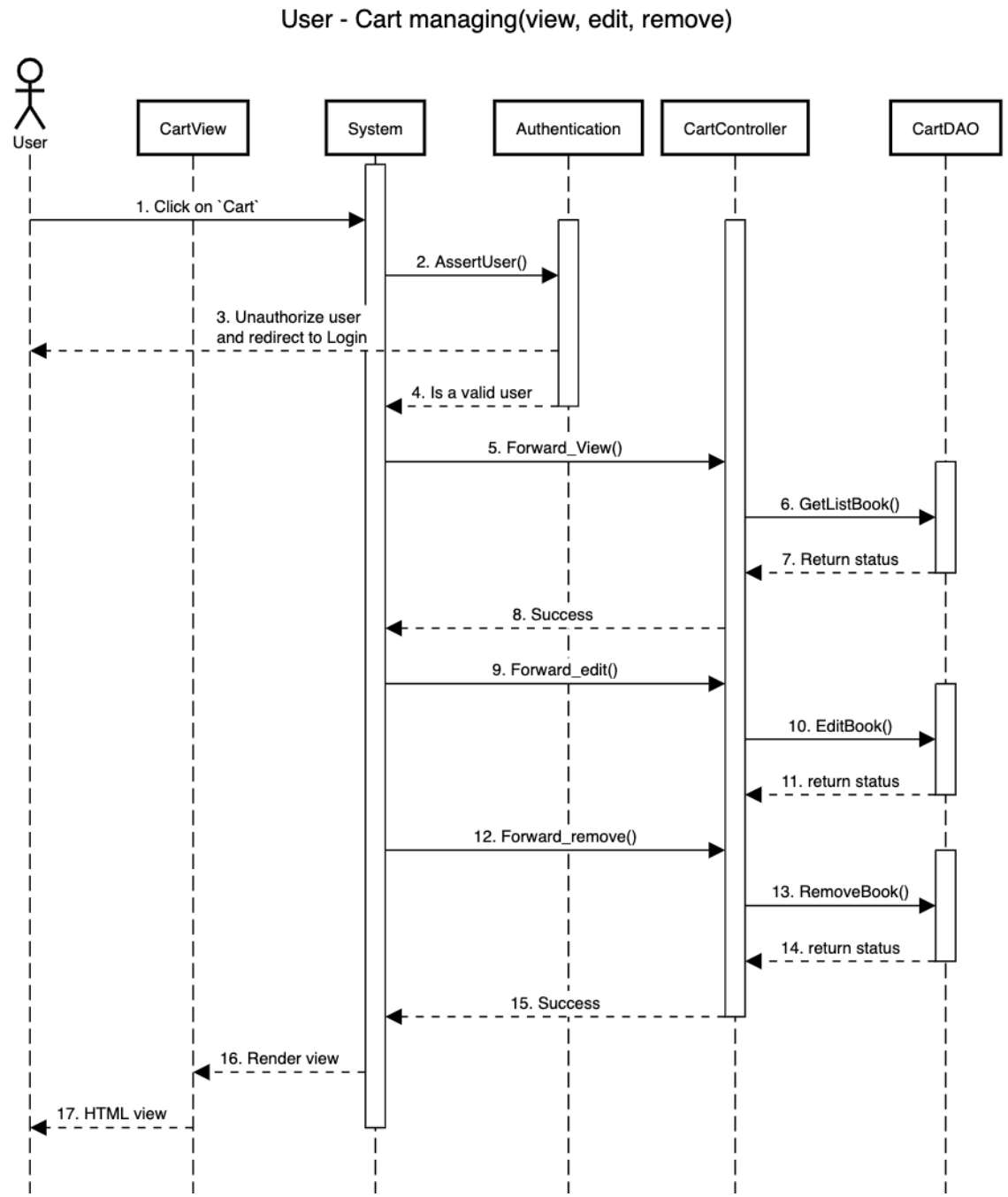


Figure 22. Sequence Diagram Cart Manage

d. Database queries

```
SELECT * FROM CART LEFT JOIN BOOK USING (book_id)
WHERE customer_id = customer_id_input
```

11. ADD BOOK TO CART

a. Class Diagram

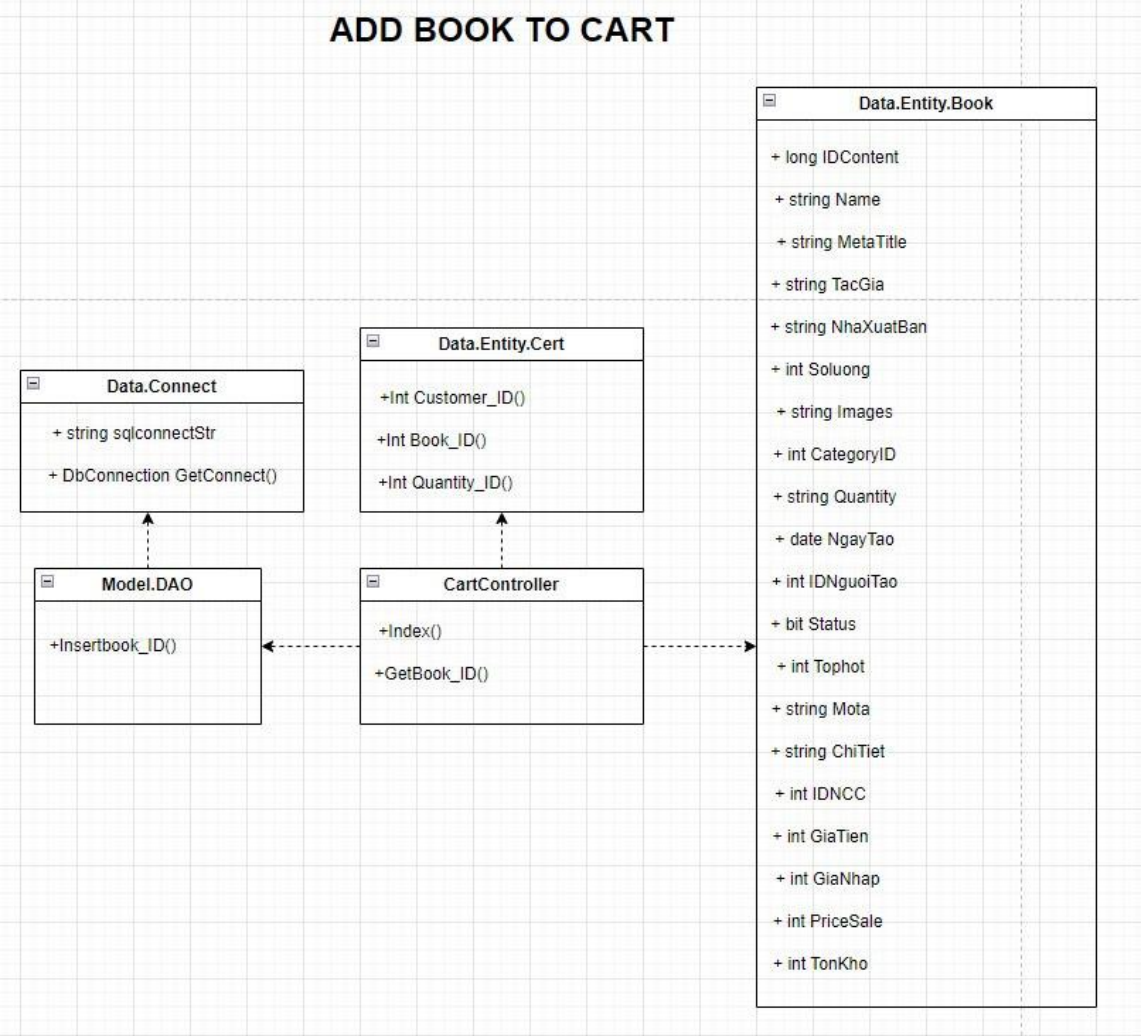


Figure 23. Class Diagram add book to Cart

b. Class Specifications

Data.BookCoinConnect

No	Method	Description
01	GetConnect()	function to connect to the system database Input: None Output: database connection command

AddBookController

No	Method	Description
01	Getbook_ID()	function get book id Input: customer_id Output: none

Models.DAO

No	Method	Description
01	Index()	function return all books are in the cart Input: book_id Output: view page

02	Delete_BookInCart(int book_id)	Function to delete book Input: book_id Output: delete book
03	GetdetailBook(string detail_book)	function to get details about a book Input : detail_book Output : none

c. Sequence Diagram

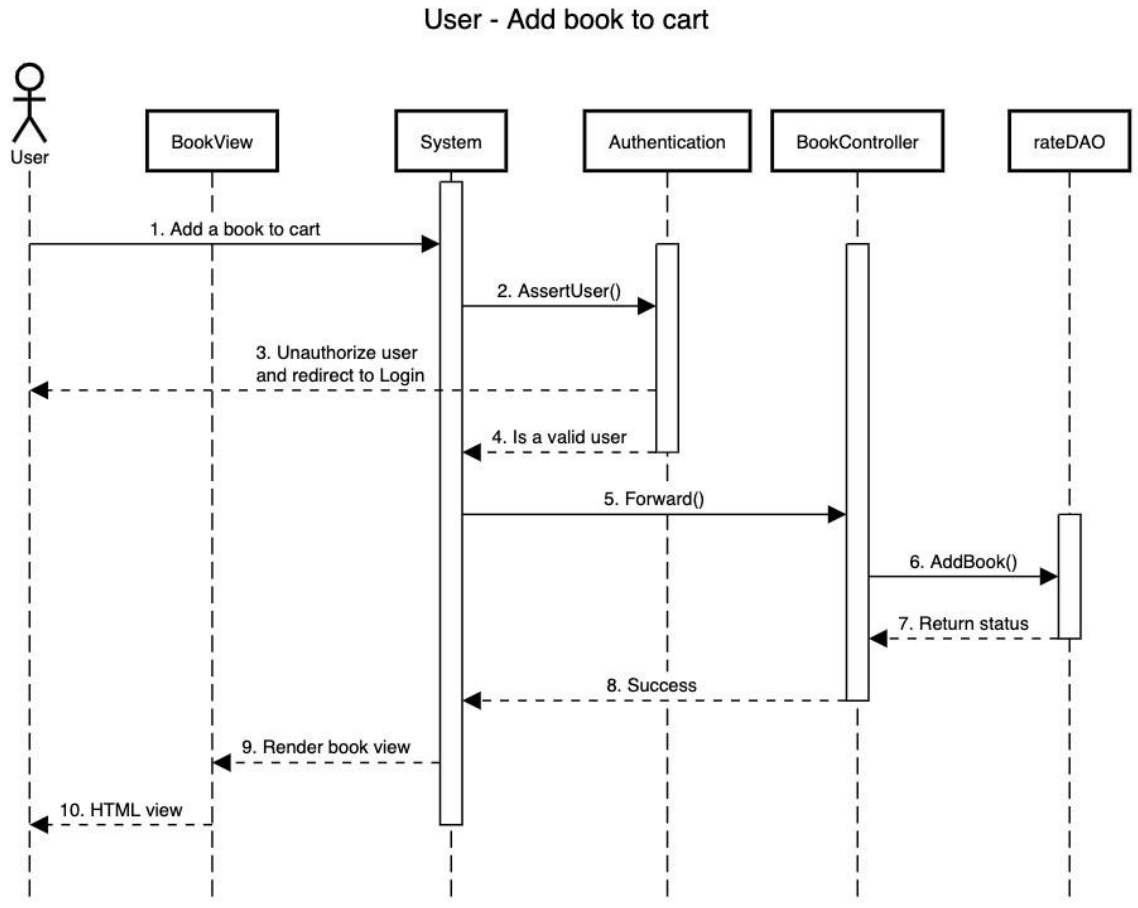


Figure 24. Class Diagram add book to cart

d. Database queries

- **Add book**

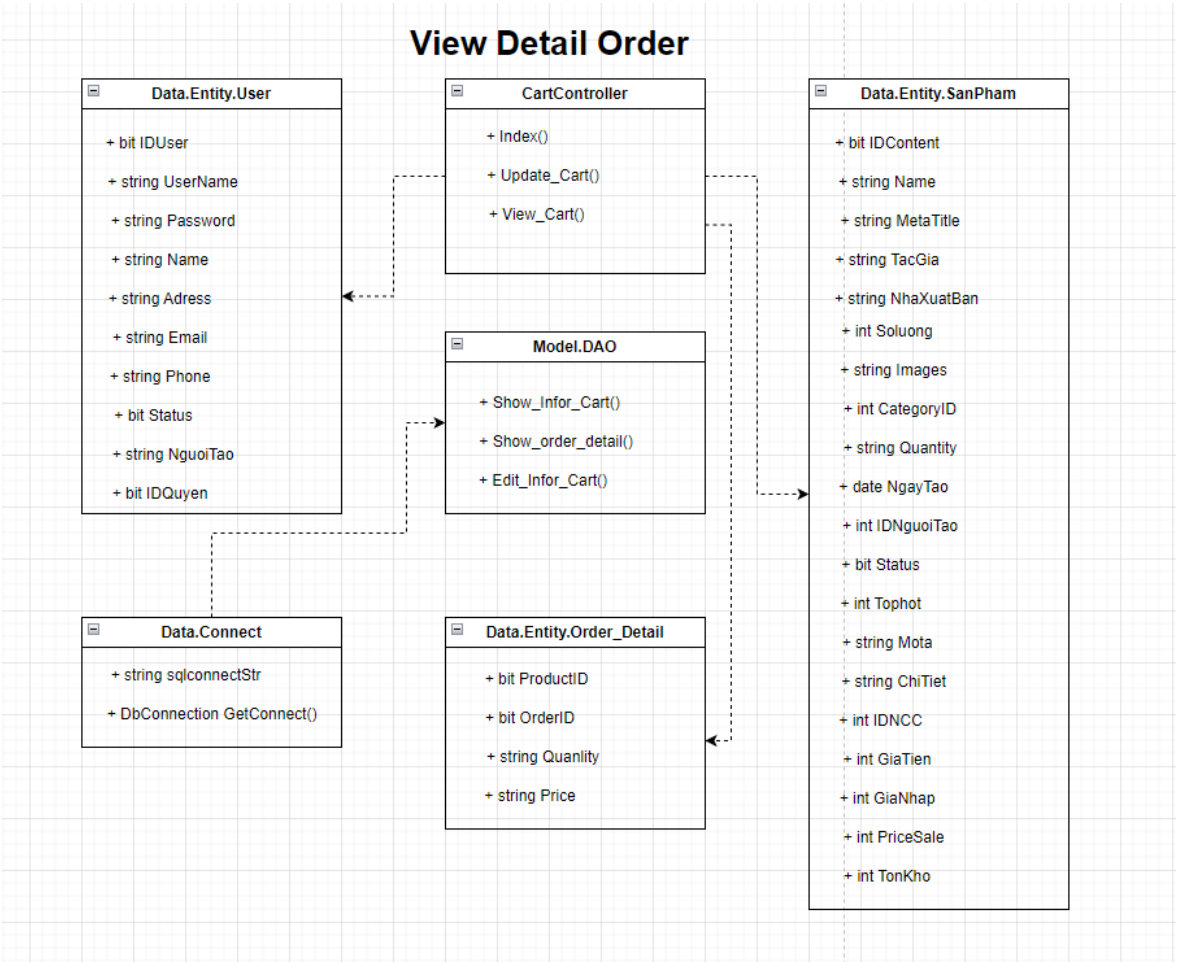
  INSERT INTO *CART* (*customer_id*, *book_id*, *quantity*)
  VALUES (*customer_id_input*, *book_id_input*, quantity_input);

UPDATE *CART*
SET *quantity = quantity_input*
WHERE *cart_id = cart_id_input*;

- **Remove Book in Cart**
  DELETE FROM CART WHERE customer_id = customer_id_input and book_id = book_id_input

12. VIEW DETAIL ORDER

a. Class Diagram



b. Class Specifications

Figure 25. Class Diagram View Detail Order

Data.BookCoinConnect

No	Method	Description
01	GetConnect()	function to connect to the system database Input: None Output: database connection command

CartController

No	Method	Description
01	Index()	function return login page view Input: None Output: Login page
02	Update_Cart()	Function to update book Input: cart information book Output: none
03	View_Cart()	function to view cart information Input: None Output: display cart information

Model.DAO

No	Method	Description
01	Show_Info_Cart()	Function to display cart information Input: address Output: display cart information
02	Show_order_detail()	Function to display order detail Input : order_id Output : display order detail
03	Edit_Infor_Cart()	Function to edit cart information Input: address Output: display cart information

c. Sequence Diagram

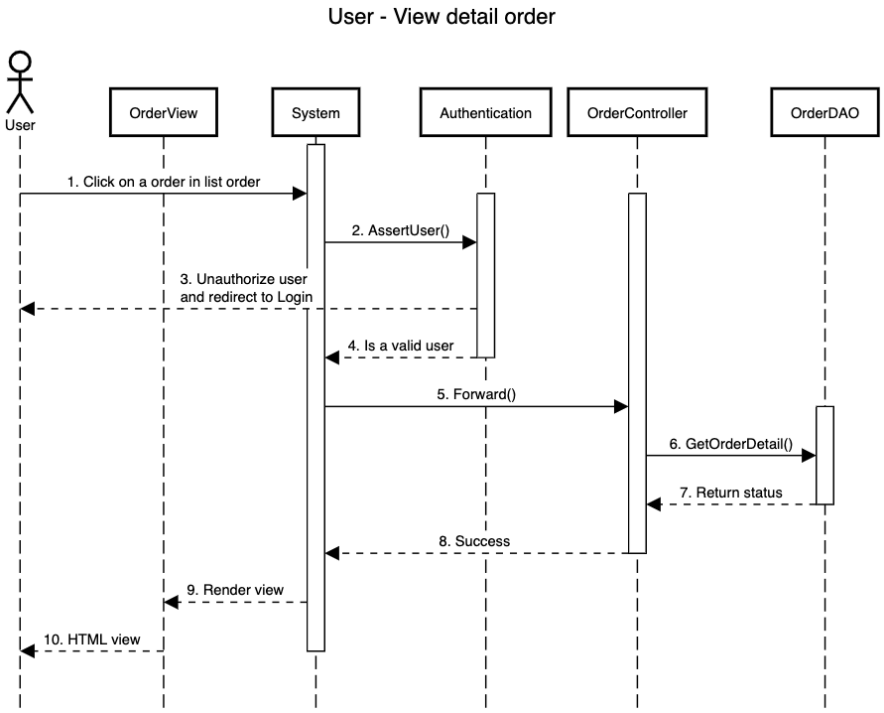


Figure 26. Sequence Diagram View detail order

d. Database queries

```
SELECT * FROM ORDER_DETAIL LEFT JOIN BOOK
USING(BOOK_ID) WHERE order_id = order_id_input
```

13. ENTER SHIPPING INFORMATION

a. Class Diagram

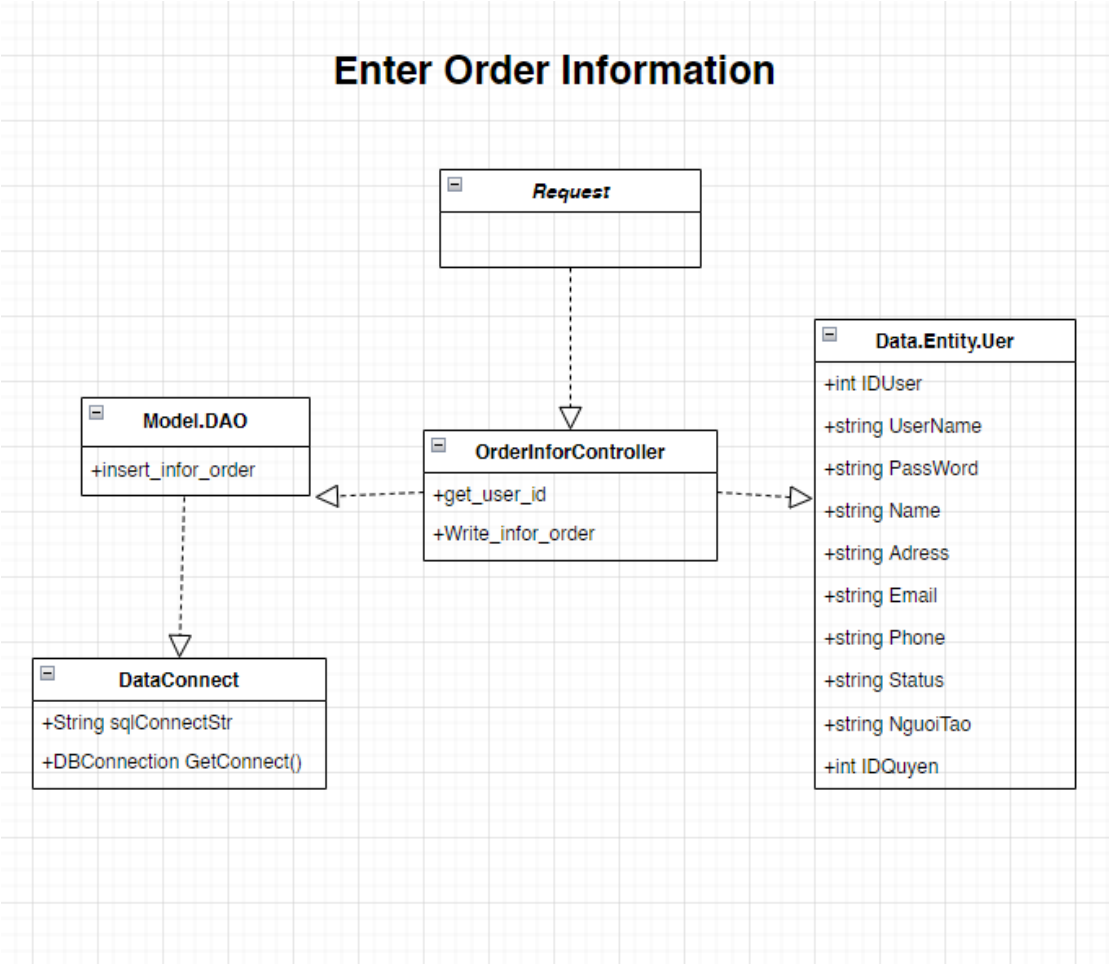


Figure 27. Class Diagram Enter Order Information

b. Class Specifications

Data.BookCoinConnect

No	Method	Description
01	GetConnect()	function to connect to the system database Input: None Output: database connection command

ShippingInforController

No	Method	Description
01	Getcustomer_ID()	function get customer ID Input: customer_id Output: none
02	WriteInfor_Shipping()	function allows users to write information Input:none Output : database connection command

Model.Received.DAO

No	Method	Description
01	Insert_Infor_Shipping(int customer_id,string name,string email,string phone_number,string address)	Function to insert information shipping Input: customer_id,name,email,phone_number,address Output:display shipping information
02	Show_order_detail(int order_detail_id)	Function to display order detail Input : order_id Output : display order detail

c. **Sequence Diagram**

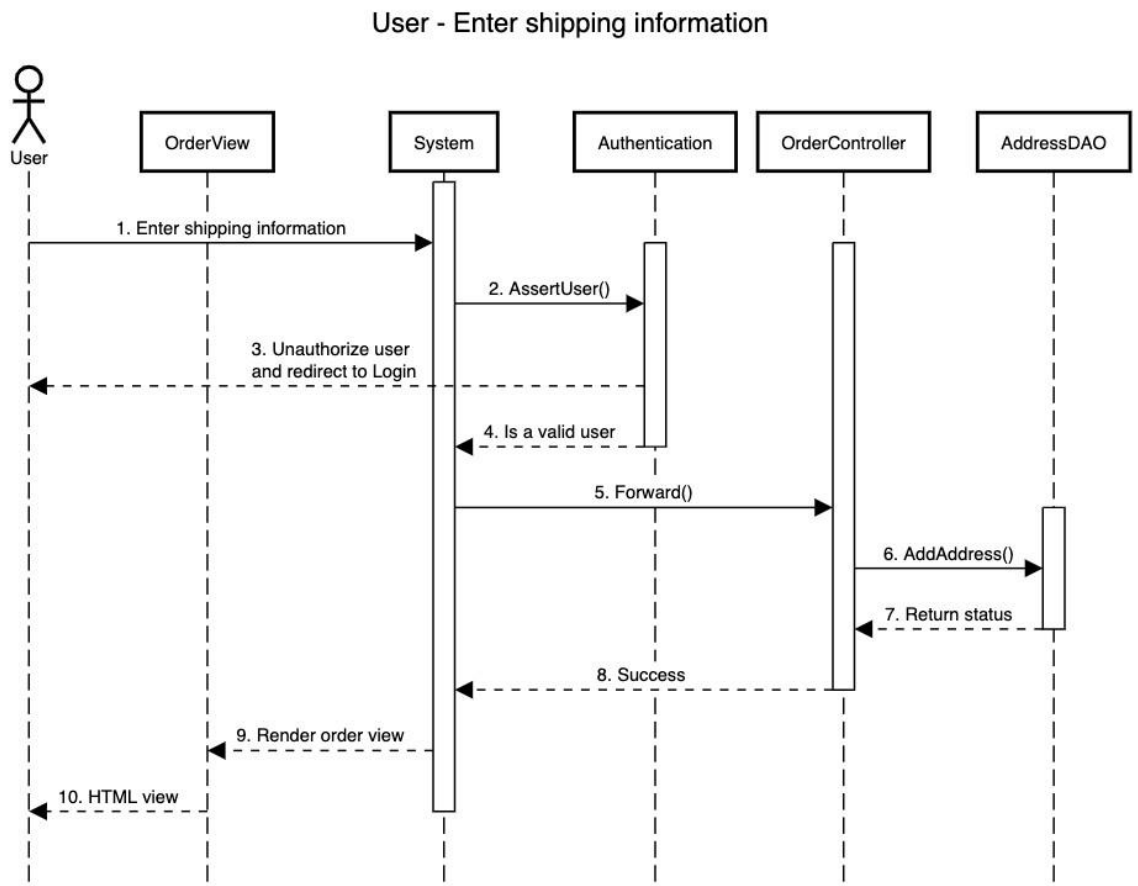


Figure 28. Sequence Diagram enter shipping information

d. Database queries

INSERT INTO *ADDRESS* (*address*)
VALUES (*address_input*);
⇒ address_id

INSERT INTO CUSTOMER_*ADDRESS* (*address_id, customer_id*)
VALUES (*address_id_input, customer_id*_input);

14. CANCEL THE ORDER

a. Class Diagram

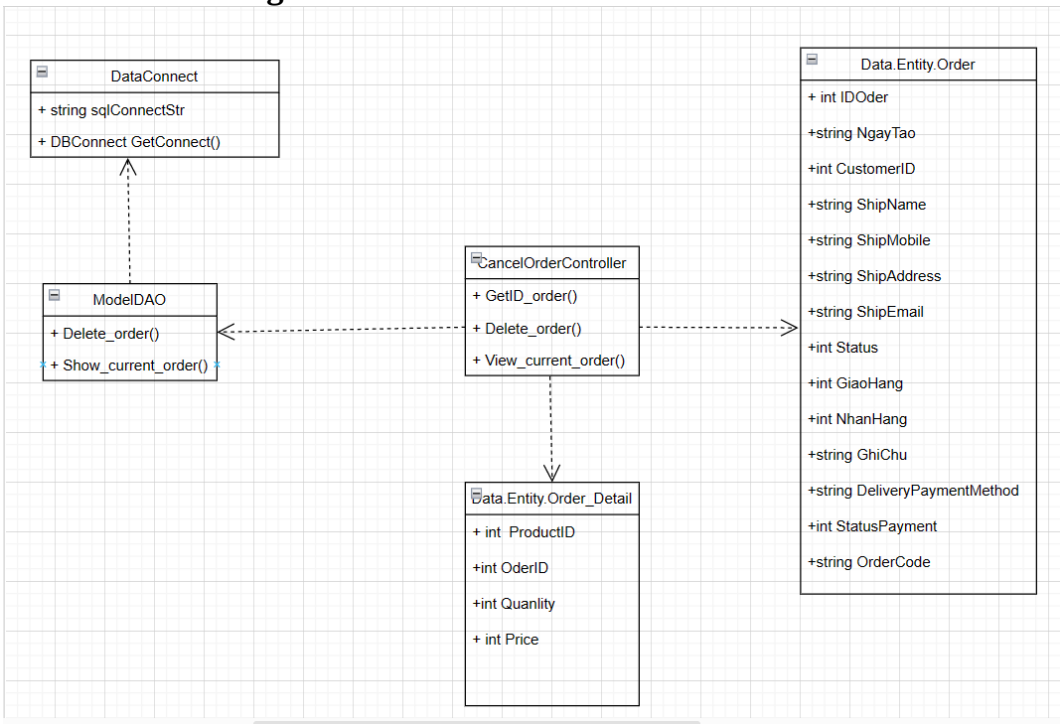


Figure 29. Class Diagram Cancel The order

b. Class Specifications

Data.Connect

No	Method	Description
01	GetConnect()	function to connect to the system database Input: None Output: database connection command

CancelController

No	Method	Description
01	GetID_order()	Function to get id of order Input: order_id Output: none
02	Delete_order()	Function to delete order Input : order_id Output : delete order

03	View_current_order()	function to view the current order Intput : order_id Output : display the current order

Models.DAO

No	Method	Description
01	Delete_order(int order_id)	function to delete order Input: order_id Output: delete order
02	Show_current_order(int order_detail,int order_id,int book_id,int quantity,float price)	function to view the current order Input: none Output: display the current order

c. Sequence Diagram

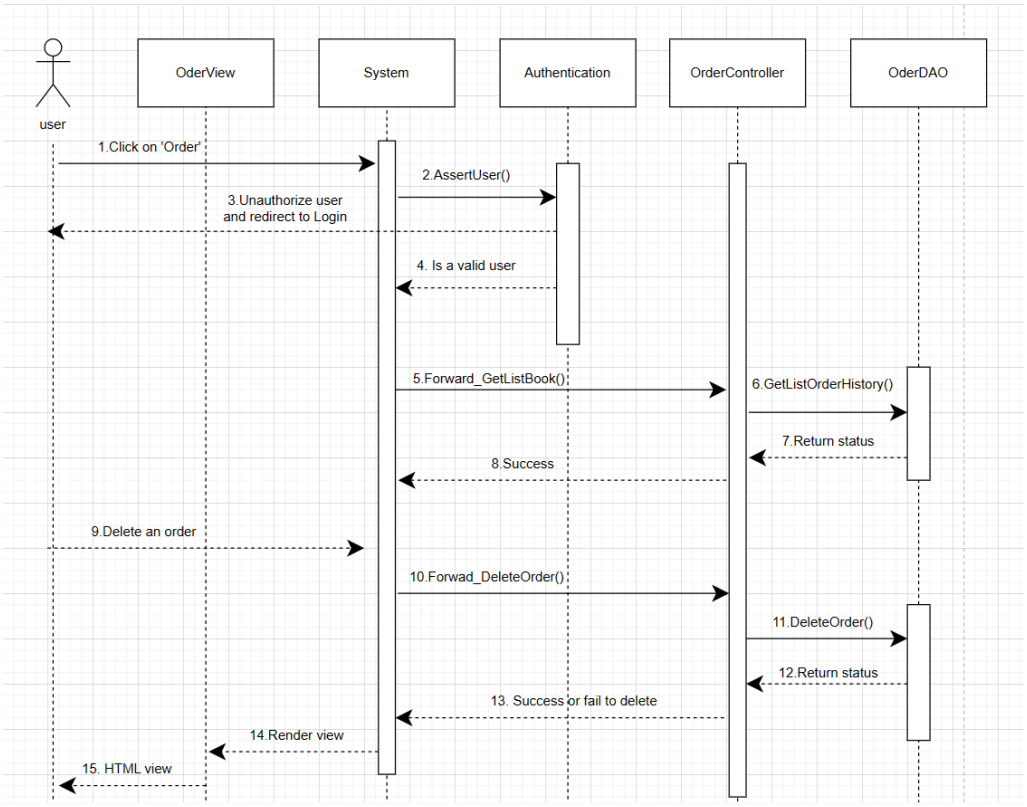


Figure 30. Sequence Diagram Cancel order

d. Database queries

UPDATE ORDER SET status = 0 WHERE ORDER_ID = ORDER_ID_INPUT

15. ORDER APPROVAL

a. Class Diagram

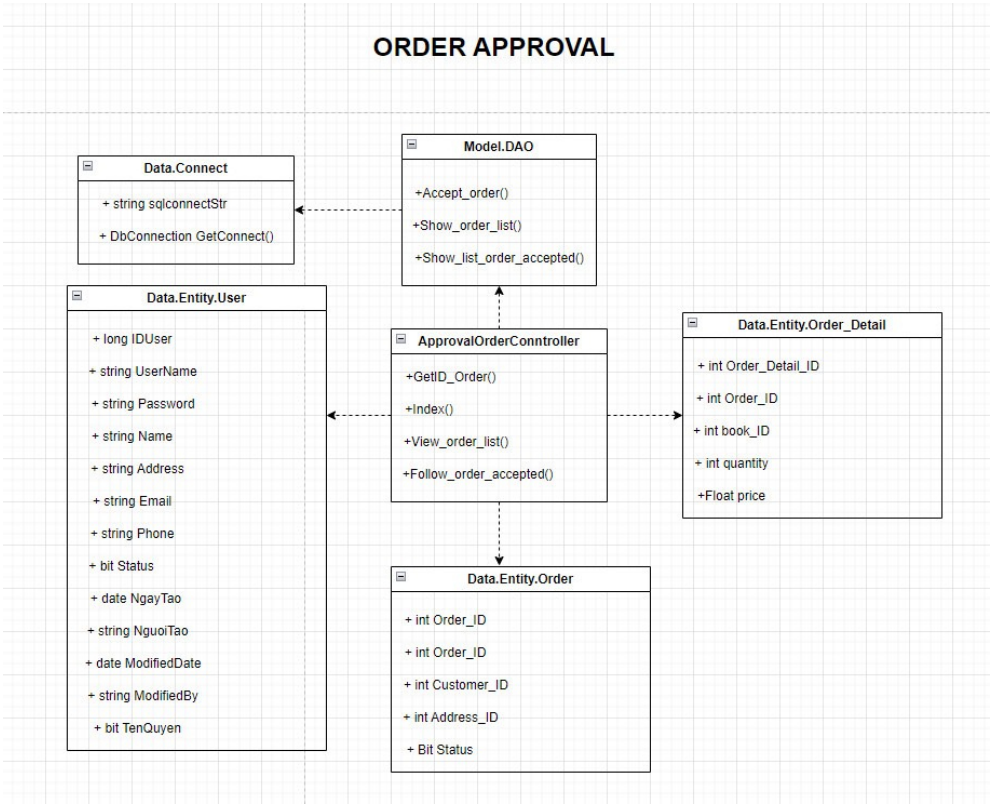


Figure 31. Class Diagram Order approval

b. Class Specifications

Data.BookCoinConnect

No	Method	Description
01	GetConnect()	function to connect to the system database Input: None Output: database connection command

ApprovalOrderController

No	Method	Description
01	GetID_order()	function get order ID Input: order_id Output: none
02	View_order_list()	function to see order list Input: order_id Output : display order list
03	Follow_order_accepted()	Function to order tracking function accepted Input: order_id Output: tracking function accepted

Models.DAO

No	Method	Description
01	Accept_order()	function get order accepted Input: none Output: order accepted

02	show_order_list(int order_id,int book_id,int quantity,float price)	Function to view order list Input: none Output: view order list
03	show_list_order_accepted()	function to display list order accepted Input : detail_order Output :display list order accepted

c. Sequence Diagram

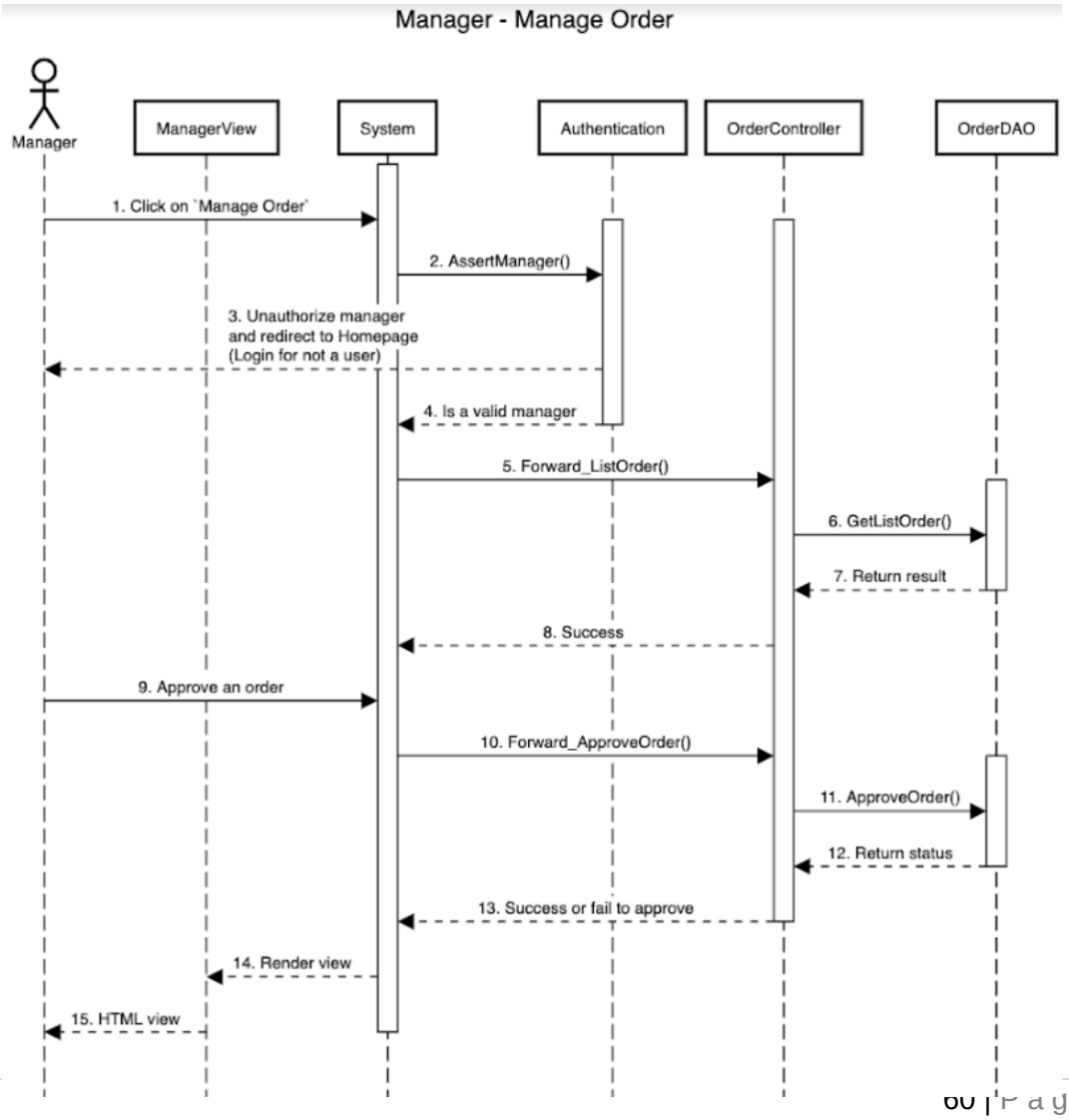


Figure 32. Sequence Diagram Approval

d. Database queries

```
UPDATE ORDER
SET status = 2
WHERE order_id = order_id_input

UPDATE ORDER
SET status = 3
WHERE order_id = order_id_input
```

16. VIEW AND EDIT INFORMATION OF MANAGER, CUSTOMER, BOOK

a. Class Diagram

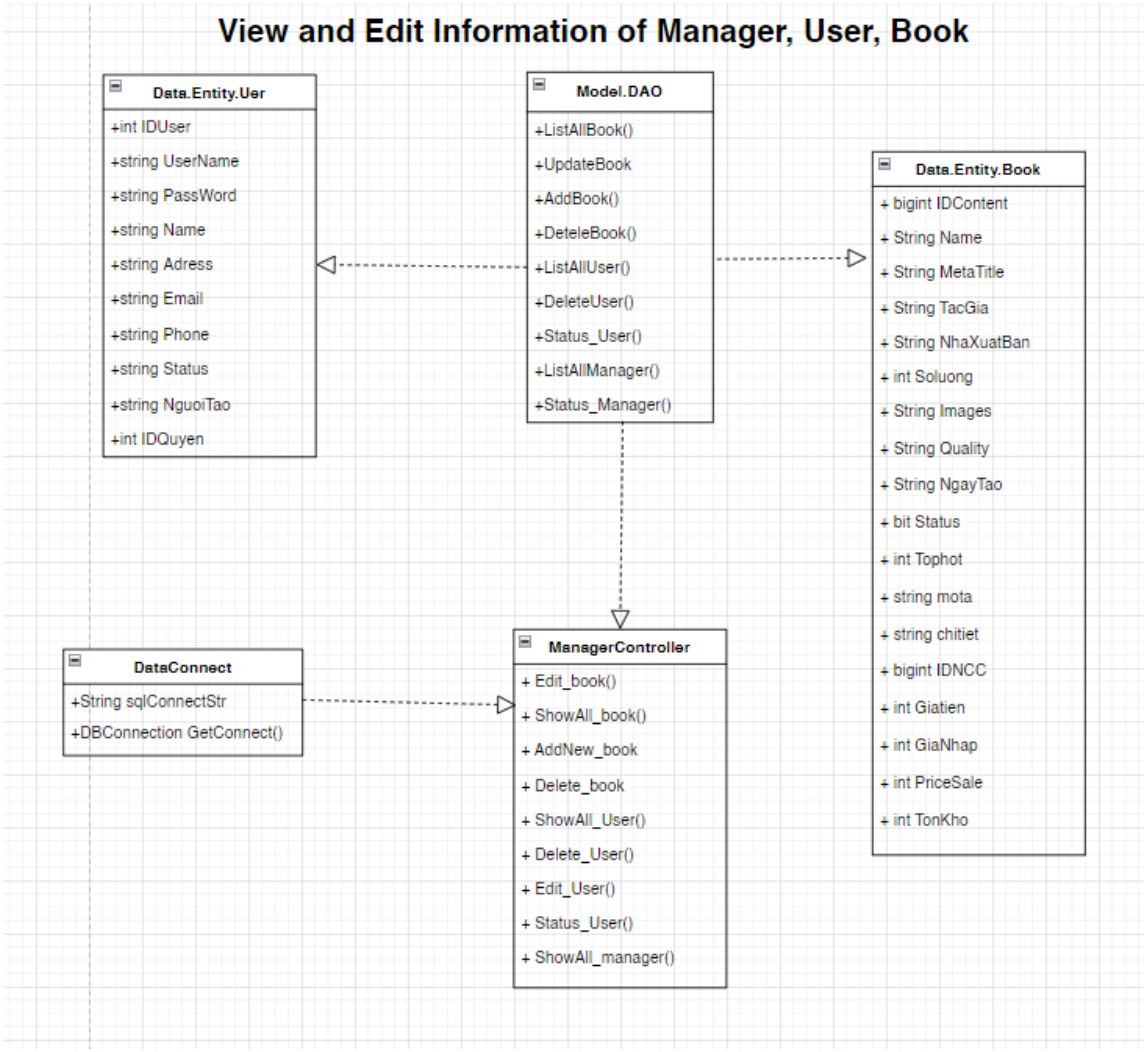


Figure 33. Class Diagram VIEW AND EDIT INFORMATION OF MANAGER, CUSTOMER, BOOK

b. Class Specifications

Data.Connect

No	Method	Description
01	GetConnect()	function to connect to the system database Input: None Output: database connection command

ManagerController

No	Method	Description
01	Index()	function return book management view page Input: None Output: view page

Models.DAO

No	Method	Description
01	ListAllBook()	function get list all ebook Input: book_id,name,author_id,price, category, size,detail_book Output: list ebooks

02	UpdateBook(book_id,name,author_id,price, category, size,detail_book)	Function to update ebook Input: book_id,name,author_id,price, category, size,detail_book Output: None
03	AddBook(int book_id,string name,int author_id,int price,string, category, size,detail_book);	Function to add new ebook Input:book_id,name,author_id,price, category, size,detail_book Output: None
04	Delete(int book_id)	Function to delete ebook Input: book_id Output: None
05	ListAllCustomer(string name, string email ,string phone_number, string password)	function get list all customer Input: None Output: list customers
06	Delete(int customer_id)	Function to delete customer Input: customer_id Output: None

07	ListAllManager(string,string email ,string phone_number, string password)	function get list all Manager Input: None Output: list customers
08	Status_ Manager()	Manager Status function Input:None Output : Output: database connection command

c. **Sequence Diagram**
d. **Database queries**

```
UPDATE User
SET name = name_input, image = image_input, phone_number =
phone_number_input, password = password_input
WHERE user_id = user_id_input;

UPDATE BOOK
SET name = name_input, image = image_input, phone_number =
phone_number_input, password = password_input
WHERE book_id = book_id_input;
```

III. Database Tables

1. Category

No	Field name	Data Type	Not Null
01	IDCategory	bigint	
02	TenTheloai	Nvarchar	
03	MetaTitle	Varchar	X
04	ParentID	bigint	X
05	SEOTitle	Varchar	X
06	NgayTao	Date	X
07	NguoiTao	Varchar	X
08	ModifiedDate	Date	X
09	ModifiedBy	Varchar	X
10	DisPlayOrder	Int	X
11	MetaDescriptions	Nvarchar	X
12	Status	Bit	

2. Contact

No	Field name	Data Type	Not Null
01	IDContact	Bigint	
02	Content	Nvarchar	X
03	Status	Bit	X

3. Feed_Back

No	Field name	Data Type	Not Null
01	IDFeedBack	bigint	
02	Name	Nvarchar	X
03	Phone	Varchar	X
04	Email	Varchar	X
05	Address	Nvarchar	X
06	Content	Nvarchar	X
07	CreateDate	Datetime	X
08	Status	Bit	X
09	Reply	Nvarchar	X
10	TieuDe	Nvarchar	X
11	UpdateDate	Datetime	X
12	UpdateBy	Varchar	X
13	CustomerID	Bigint	X

4. **Footer**

No		Field name	Data Type	Not Null
01		IdFooter	bigint	
02		Content	Nvarchar	
03		Status	Bit	

5. **KhoHang**

No	Field name	Data Type	Not Null
01	ID	bigint	
02	TenKho	nvarchar	X
03	IDProduct	bigint	X
04	SoLuongKho	int	
05	TonKho	int	
06	NgayTao	datetime	X
07	Status	bit	X

6. **Menu**

No	Field name	Data Type	Not Null
01	IDMenu	bigint	
02	NameMenu	nvarchar	
03	Link	varchar	X
04	DisplayOrder	int	X
05	Target	varchar	X
06	MenuTypeID	bigint	X
07	Status	bit	
08	CreateDate	datetime	X

7. **MenuType**

No	Field name	Data Type	Not Null
01	MenuTypeID	bigint	
02	NameType	varchar	X

8. **Messenger**

No	Field name	Data Type	Not Null
01	IDMes	bigint	
02	IDCustomer	bigint	X
03	NgayTao	datetime	X
04	Content	nvarchar	X
05	Status	bit	X
06	IDOrder	bigint	X

9. Muon_Tra

No	Field name	Data Type	Not Null
01	IdMuon	bigint	
02	IDContent	bigint	
03	IDUser	bigint	
04	SoluongMuon	int	
05	ThoigianM	datetime	X
06	ThoiGianHenTra	datetime	X
07	ThoigianTra	datetime	X
08	SoLanGiaHan	int	

10. NhaCungCap

No	Field name	Data Type	Not Null
01	IDNCC	bigint	
02	TenNCC	nvarchar	
03	NgayTao	date	X
04	IDNguoiTao	bigint	X
05	Status	bit	
06	DiaChi	nvarchar	X
07	SoDienThoai	varchar	X
08	Email	varchar	X

11. NhapHang

No	Field name	Data Type	Not Null
01	IDNhapHang	bigint	
02	IDSanPham	bigint	X
03	GiaNhap	int	X
04	SoLuongNhap	int	X
05	NgayTao	datetime	X
06	Status	int	X
07	IDNCC	bigint	X
08	IDCategory	bigint	X
09	IDNguoiTao	bigint	X
10	StatusPayMent	bit	X
11	StatusInput	bit	X

12. Order_Detail

No	Field name	Data Type	Not Null
01	ProductID	bigint	
02	OderID	bigint	
03	Quanlity	int	X
04	Price	int	X

13. Orders

No	Field name	Data Type	Not Null
01	IDOder	bigint	
02	NgayTao	datetime	X
03	CustomerID	bigint	X
04	ShipName	Nvarchar	X
05	ShipMobile	Nvarchar	X
06	ShipAddress	Nvarchar	X
07	ShipEmail	Nvarchar	X
08	Status	Int	X
09	GiaoHang	Int	X
10	NhanHang	Int	X
11	GhiChu	Nvarchar	X
12	DeliveryPaymentMethod	Nvarchar	X
13	StatusPayment	Int	X
14	OrderCode	Varchar	X

14.Quyen

No	Field name	Data Type	Not Null
01	IDQuyen	bigint	
02	TenQuyen	Nvarchar	
03	Status	Bit	

15. SanPham

No	Field name	Data Type	Not Null
01	IDContent	bigint	
02	Name	nvarchar	
03	MetaTitle	varchar	X
04	TacGia	nvarchar	
05	NhaXuatBan	nvarchar	
06	Soluong	int	
07	Images	varchar	X
08	CategoryID	bigint	X
09	Quanlity	nvarchar	X
10	NgayTao	datetime	

11	IDNguoiTao	bigint	X
12	Status	bit	
13	Tophot	int	
14	Mota	nvarchar	X
15	ChiTiet	nvarchar	X
16	IDNCC	bigint	X
17	GiaTien	int	
18	GiaNhap	int	X
19	PriceSale	int	X
20	TonKho	int	X

16.Order_Detail

No	Field name	Data Type	Not Null
01	IDUser	bigint	
02	UserName	varchar)	
03	PassWord	varchar	
04	Name	nvarchar	
05	Adress	nvarchar	
06	Email	varchar	
07	Phone	varchar	
08	Status	bit	
09	NgayTao	date	X
10	NguoiTao	varchar	X
11	ModifiedDate	date	X
12	ModifiedBy	varchar	X
13	IDQuyen	bigint	

--The End--