

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN 1

--o0o--



KIẾN TRÚC VÀ THIẾT KẾ
PHẦN MỀM

Assignment 3

Họ và tên	: Tạ Trường Vũ
Mã sinh viên	: B22DCCN918
Giảng viên	: Trần Đình Quế
Khóa	: 2022-2027
Hệ	: Đại học chính quy
Chuyên ngành	: Công nghệ phần mềm

Hà Nội, 2026

MỤC LỤC

Chương 1: Xác định và phân tích yêu cầu (không đầy đủ).....	1
Lập Bảng các actor và các chức năng tương ứng.....	1
Lập Bảng các lớp, các thuộc tính và các chức năng trong lớp đó.....	1
Biểu đồ lớp Phân tích > 50 lớp.....	5
Vẽ 1 biểu đồ hoạt động và 1 biểu đồ tuần tự. Trình bày ý nghĩa các biểu đồ này.....	5
Biểu đồ hoạt động:.....	5
Biểu đồ tuần tự:.....	6
Chương 2: Data model và Data base.....	7
Data model.....	7
Database MySQL.....	8
Chương 3: Thiết kế.....	9
Trình bày hiểu biết về DAO, MVC và mô hình monolithic.....	9
Sinh code Django.....	12
Chương 4: Cài đặt và triển khai.....	12
Tương ứng code và thiết kế.....	12
Màn hình giao diện.....	14
.....	14

Chương 1: Xác định và phân tích yêu cầu (không đầy đủ)

Lập Bảng các actor và các chức năng tương ứng

Actor	Chức năng
Khách hàng	<ul style="list-style-type: none">• Đăng ký / đăng nhập hệ thống• Quản lý thông tin cá nhân và địa chỉ giao hàng• Tìm kiếm, lọc sách theo danh mục, tác giả• Xem chi tiết sách, đánh giá• Thêm sách vào giỏ hàng, cập nhật số lượng giỏ hàng• Đặt hàng và thanh toán• Theo dõi trạng thái đơn hàng
Nhân viên	<ul style="list-style-type: none">• Quản lý thông tin sách và danh mục sách• Nhập sách từ nhà cung cấp vào hệ thống• Xử lý đơn hàng (xác nhận, đóng gói)
Quản trị viên	<ul style="list-style-type: none">• Quản lý tài khoản người dùng (Customer, Staff)• Quản lý danh mục sách,• Quản lý đơn hàng và trạng thái đơn hàng
Hệ thống thanh toán	<ul style="list-style-type: none">• Tạo yêu cầu thanh toán cho đơn hàng
Hệ thống vận chuyển	<ul style="list-style-type: none">• Nhận yêu cầu giao hàng từ đơn hàng• Lấy thông tin vận chuyển
Module kho & tồn kho	<ul style="list-style-type: none">• Theo dõi tồn kho theo từng sách và kho• Ghi nhận nhập kho, xuất kho• Cập nhật số lượng tồn kho sau mỗi biến động
Module tư vấn & gợi ý	<ul style="list-style-type: none">• Phân tích lịch sử mua hàng của khách hàng• Gợi ý sách phù hợp cho khách hàng• Đề xuất sách bán chạy hoặc sách liên quan

Lập Bảng các lớp, các thuộc tính và các chức năng trong lớp đó

Lớp	Thuộc tính	Methods
Customer	id,username,password,name, address_id	List_customers(), get_customer(), register_customer(),

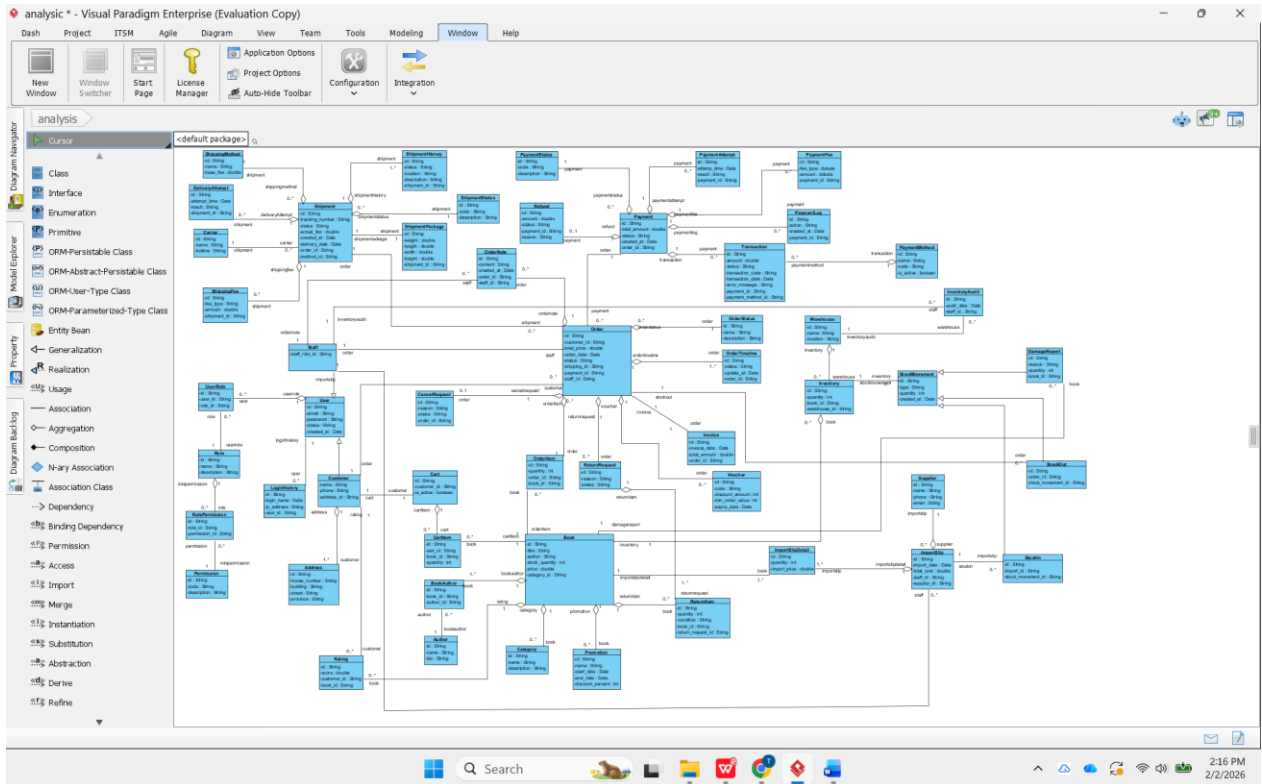
		update_customer(), login_api(), logout_api()
Staff	id, name, role	List_staff(), get_staff()
Address	id, house_number, street, district, province	addAddress(), updateAddress(), removeAddress()
Cart	id, customer_id, is_active	addItem(), removeItem(), clearCart(), computeTotal()
CartItem	id, cart_id, book_id, quantity	updateQuantity(), removeItem()
Book	id, name, author, stock_quantity, price, category_id	getDetail(), updateStock(), updatePrice()
Category	id, name, description	addCategory(), updateCategory(), removeCategory()
Review	id, rating, comment, customer_id, book_id	addReview(), updateReview(), deleteReview()
Order	id, customer_id, total_price, order_date, status, shipping_id, payment_id, staff_id	createOrder(), updateStatus(), cancelOrder()
OrderItem	id, order_id, book_id, quantity	updateQuantity()
OrderStatus	id, name, description	changeStatus()
OrderTimeline	id, status, updated_at, order_id	recordTimeline()
OrderNote	id, content, created_at, order_id, staff_id	addNote()
Invoice	id, invoice_date, total_amount, order_id	generateInvoice(), exportInvoice()
CancelRequest	id, reason, status, order_id	createRequest(), approve(), reject()

Payment	id, total_amount, status, created_at, order_id	createPayment(), updateStatus(), confirmPayment()
Transaction	id, amount, status, transaction_code, transaction_date, error_message, payment_id, payment_method_id	processTransaction(), recordResult()
PaymentMethod	id, name, code, is_active	enableMethod(), disableMethod()
PaymentStatus	id, code, description	updateStatus()
PaymentLog	id, action, created_at, payment_id	logAction()
PaymentFee	id, fee_type, amount, payment_id	calculateFee()
PaymentAttempt	id, attempt_time, result, payment_id	recordAttempt()
Refund	id, amount, reason, status, payment_id	createRefund(), processRefund()
Shipment	id, tracking_number, status, actual_fee, created_at, delivery_date, order_id, method_id	createShipment(), updateStatus()
ShippingMethod	id, name, base_fee	calculateFee()
ShipmentStatus	id, code, description	updateStatus()
ShipmentHistory	id, status, location, description, shipment_id	addHistory()
DeliveryAttempt	id, attempt_time, result, shipment_id	recordAttempt()
Carrier	id, name, hotline	assignShipment()
ShipmentPackage	id, weight, length, width, height, shipment_id	calculateVolume()
Supplier	id, name, phone, email	supplyBook(), updateInfo()

Import	id, import_date, total_cost, staff_id, supplier_id	createImport(), confirmImport()
ImportDetail	id, quantity, import_price, import_id, book_id	updateQuantity()
Warehouse	id, name, location	addWarehouse(), updateWarehouse()
Inventory	id, quantity, book_id, warehouse_id	updateQuantity(), checkStock()
StockMovement	id, type, quantity, created_at	recordMovement()
StockIn	id, import_id, stock_movement_id	recordStockIn()
StockOut	id, order_id, stock_movement_id	recordStockOut()
DamageReport	id, reason, quantity, book_id	reportDamage()
InventoryAudit	id, audit_date, staff_id	performAudit(), generateReport()
Promotion	id, name, start_date, end_date, discount_percent	applyPromotion(), deactivate()
Voucher	id, code, discount_amount, min_order_value, expiry_date	validateVoucher(), applyVoucher()
ReturnRequest	id, reason, status	createRequest(), approve(), reject()
Reservation	id, quantity, condition, book_id, return_request_id	reserveBook(), releaseReservation()
Author	id, name, bio	updateAuthorInfo()
BookAuthor	id, book_id, author_id	linkAuthor(), unlinkAuthor()
Role	id, name, description	createRole(), updateRole()
Permission	id, code, description	createPermission()
RolePermission	id, role_id, permission_id	assignPermission(), revokePermission()
UserRole	id, user_id, role_id	assignRole(), removeRole()

LoginHistory	id, login_time, ip_address, user_id	recordLogin()
---------------------	-------------------------------------	---------------

Biểu đồ lớp Phân tích > 50 lớp

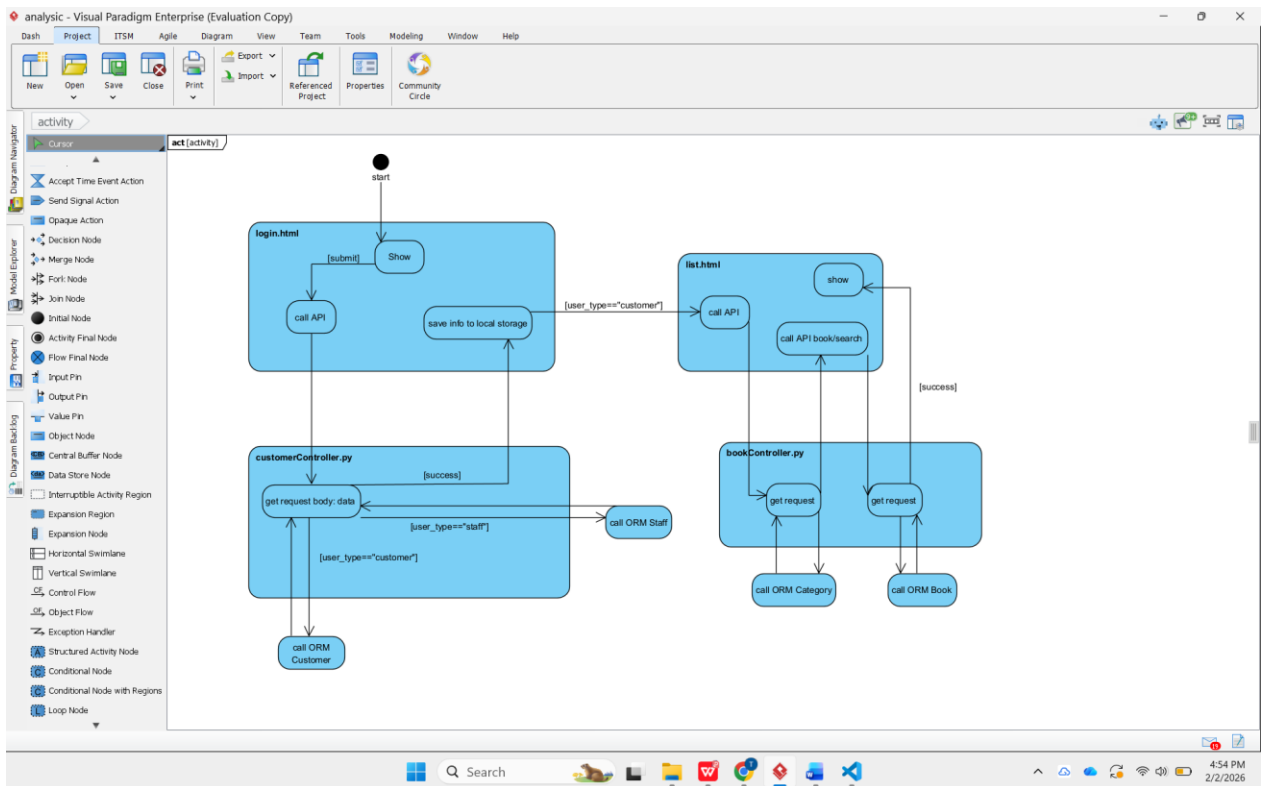


Vẽ 1 biểu đồ hoạt động và 1 biểu đồ tuần tự. Trình bày ý nghĩa các biểu đồ này

Biểu đồ hoạt động:

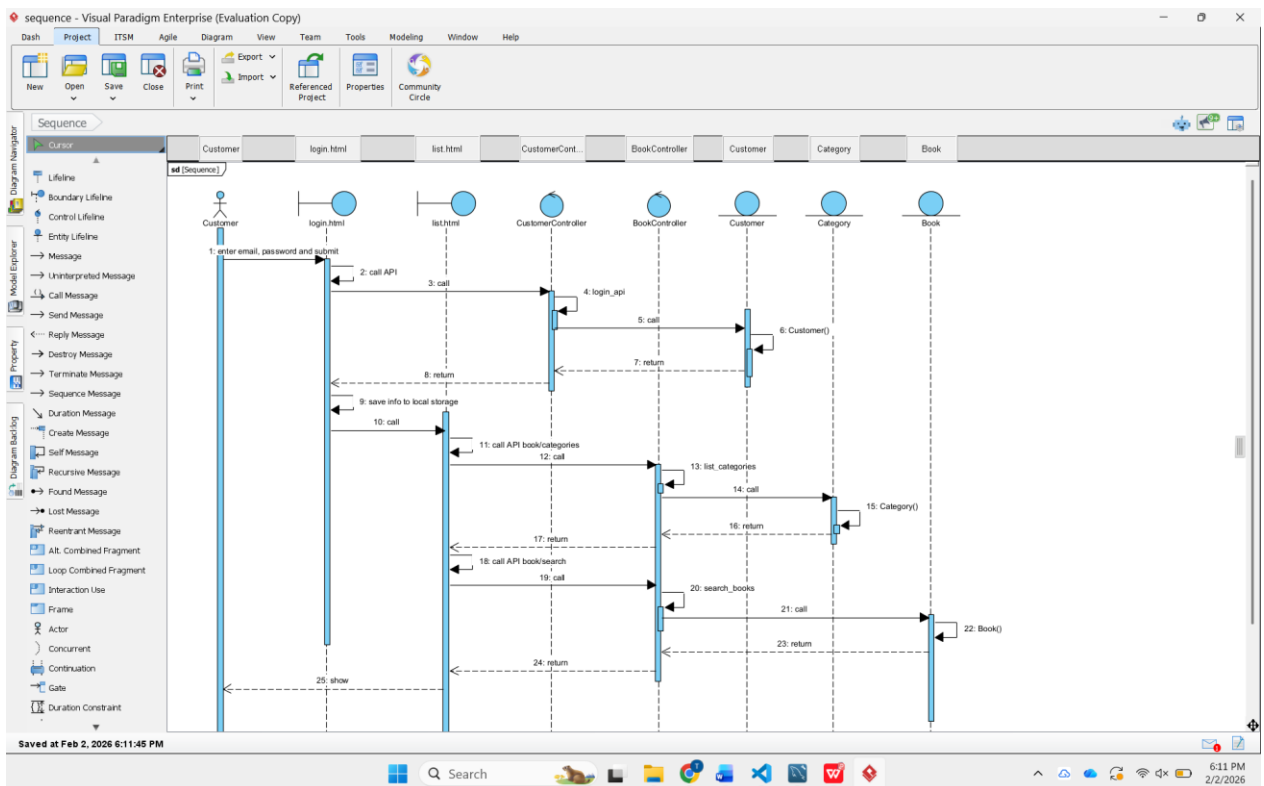
Biểu diễn các hoạt động và sự đồng bộ, chuyển tiếp các hoạt động của hệ thống trong một lớp hoặc kết hợp giữa các lớp với nhau trong một chức năng cụ thể.

- Xác định các hành động phải thực hiện trong phạm vi một phương thức
- Để xác định công việc cụ thể của một đối tượng
- Để chỉ ra một nhóm hành động liên quan của các đối tượng được thực hiện như thế nào và chúng sẽ ảnh hưởng đến những đối tượng nằm xung quanh

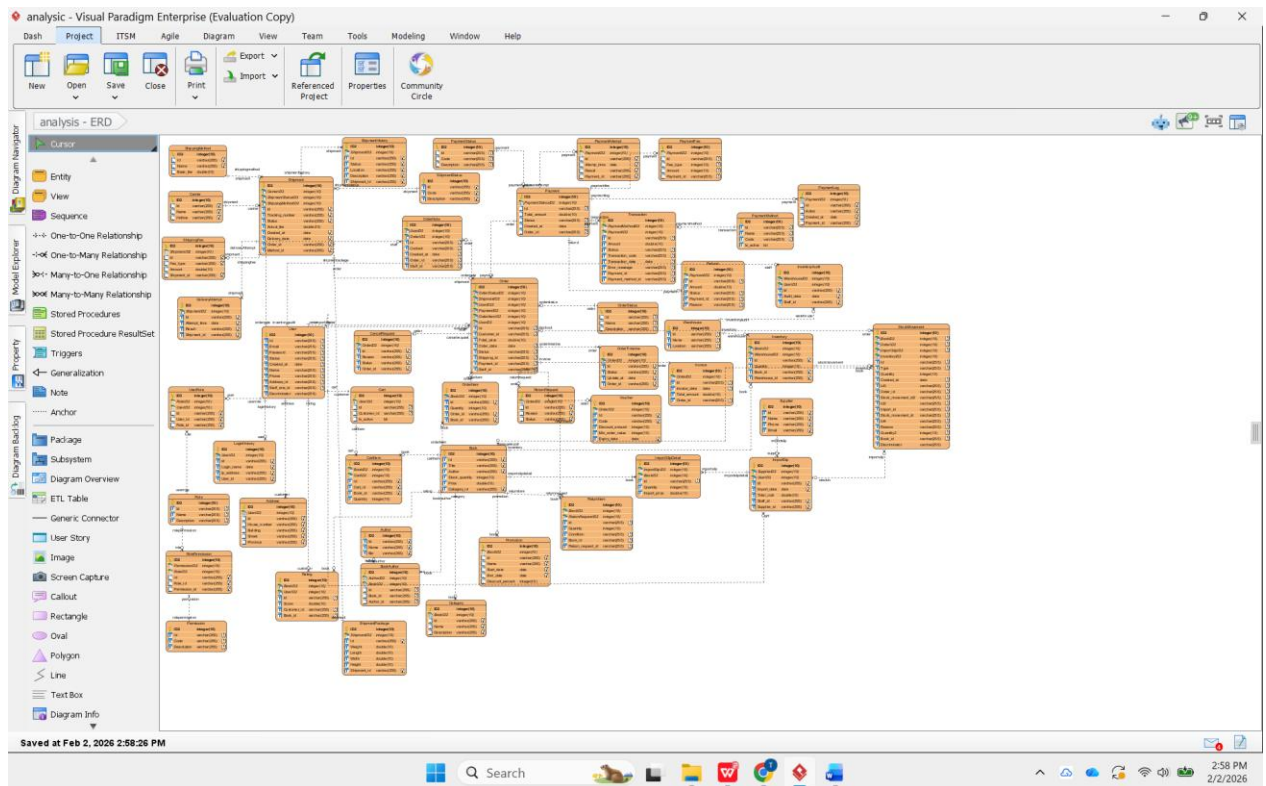
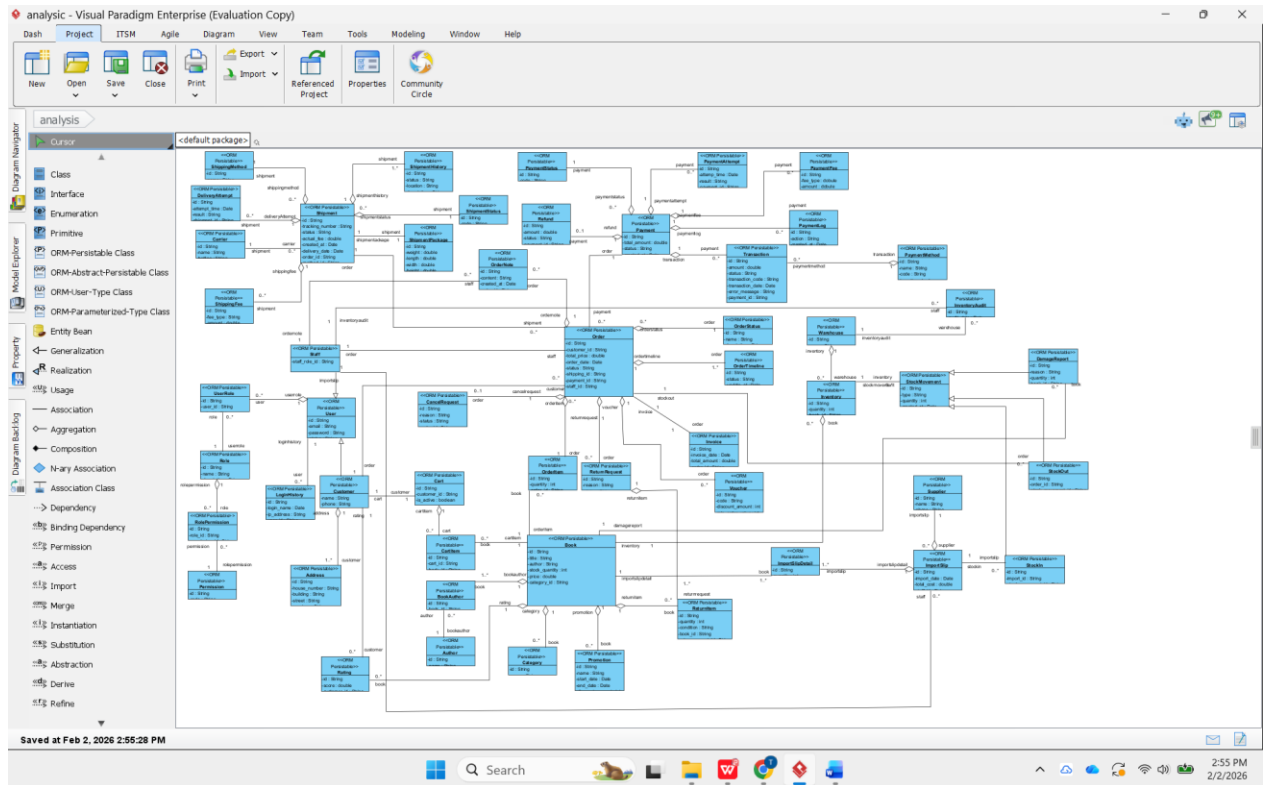


Biểu đồ tuần tự:

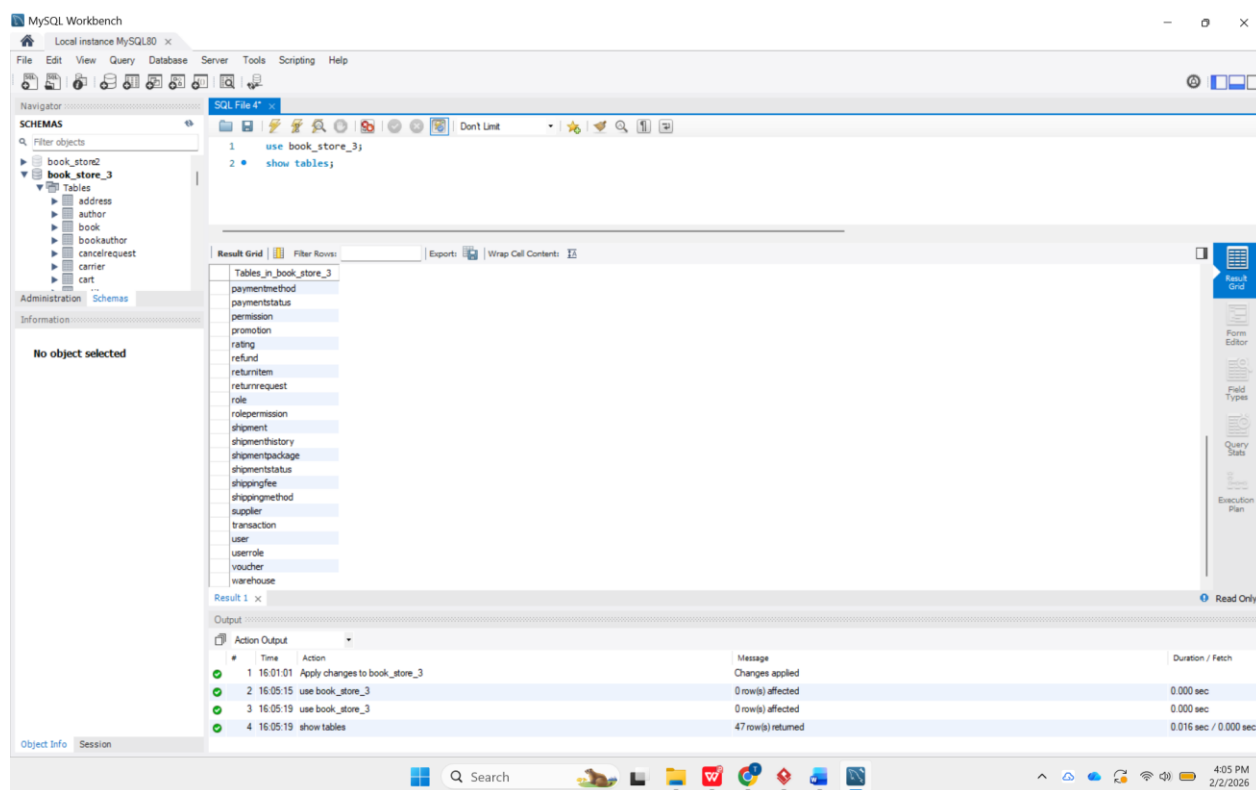
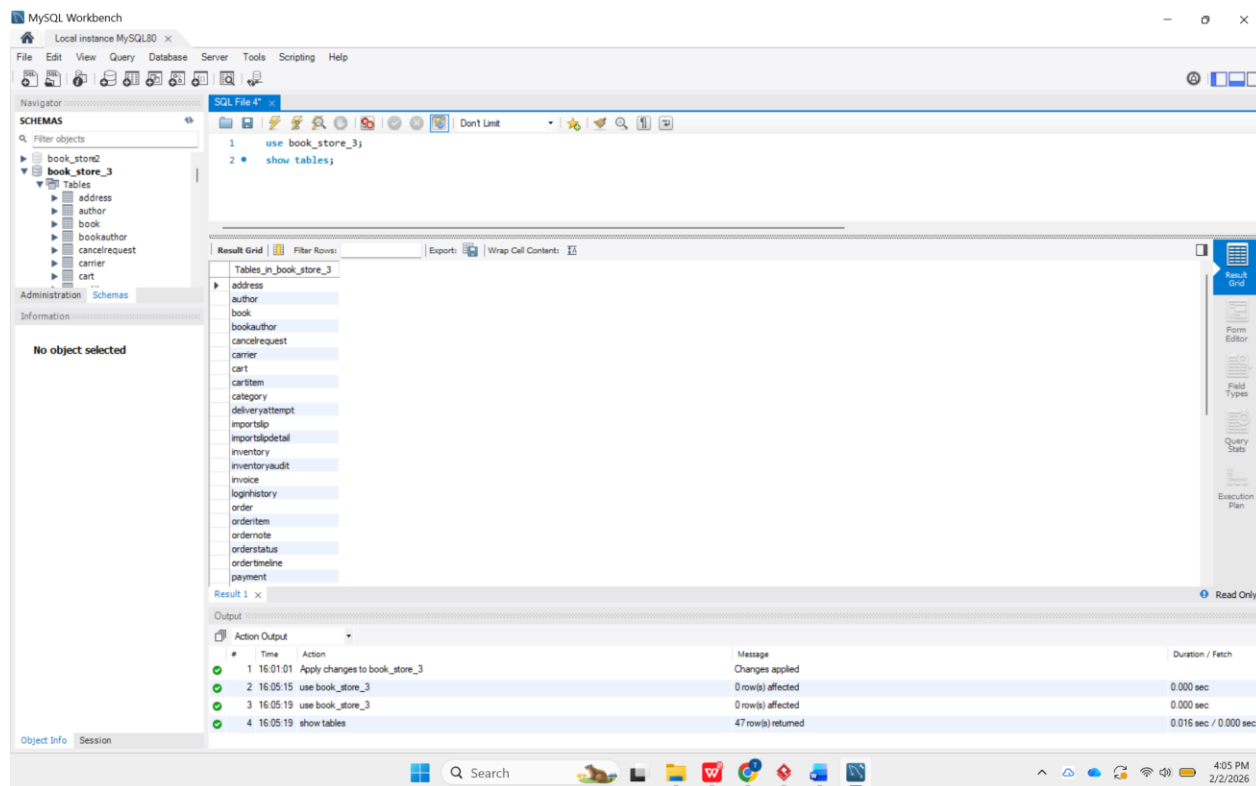
Biểu diễn mối quan hệ giữa các đối tượng, giữa các đối tượng và tác nhân theo thứ tự thời gian. Biểu đồ tuần tự nhấn mạnh thứ tự thực hiện của các tương tác.



Data model



Database MySQL



Chương 3: Thiết kế

Trình bày hiểu biết về DAO, MVC và mô hình monolithic

a) Mô hình Monolithic

Mô hình Monolithic là kiểu kiến trúc truyền thống, trong đó toàn bộ ứng dụng được xây dựng và triển khai như một khối thống nhất. Các thành phần như giao diện người dùng, xử lý nghiệp vụ và truy cập cơ sở dữ liệu đều nằm chung trong một project và chạy trong cùng một tiến trình.

- **Đặc điểm:** Tất cả mã nguồn được gom lại trong một ứng dụng duy nhất, không tách riêng thành các module độc lập.
- **Ưu điểm:** Dễ tiếp cận và triển khai ở giai đoạn đầu; việc kiểm thử và chạy chương trình cũng đơn giản vì chỉ cần build và deploy một gói duy nhất.
- **Nhược điểm:** Khi hệ thống mở rộng và phức tạp hơn thì việc bảo trì trở nên khó khăn; chỉ một lỗi nhỏ cũng có thể ảnh hưởng đến toàn bộ ứng dụng; ngoài ra việc sử dụng nhiều công nghệ khác nhau cho từng phần của hệ thống cũng không thuận tiện.

b) Mô hình MVC (Model – View – Controller)

MVC là mô hình kiến trúc giúp chia ứng dụng thành ba phần riêng biệt nhằm tăng tính rõ ràng và dễ bảo trì trong quá trình phát triển.

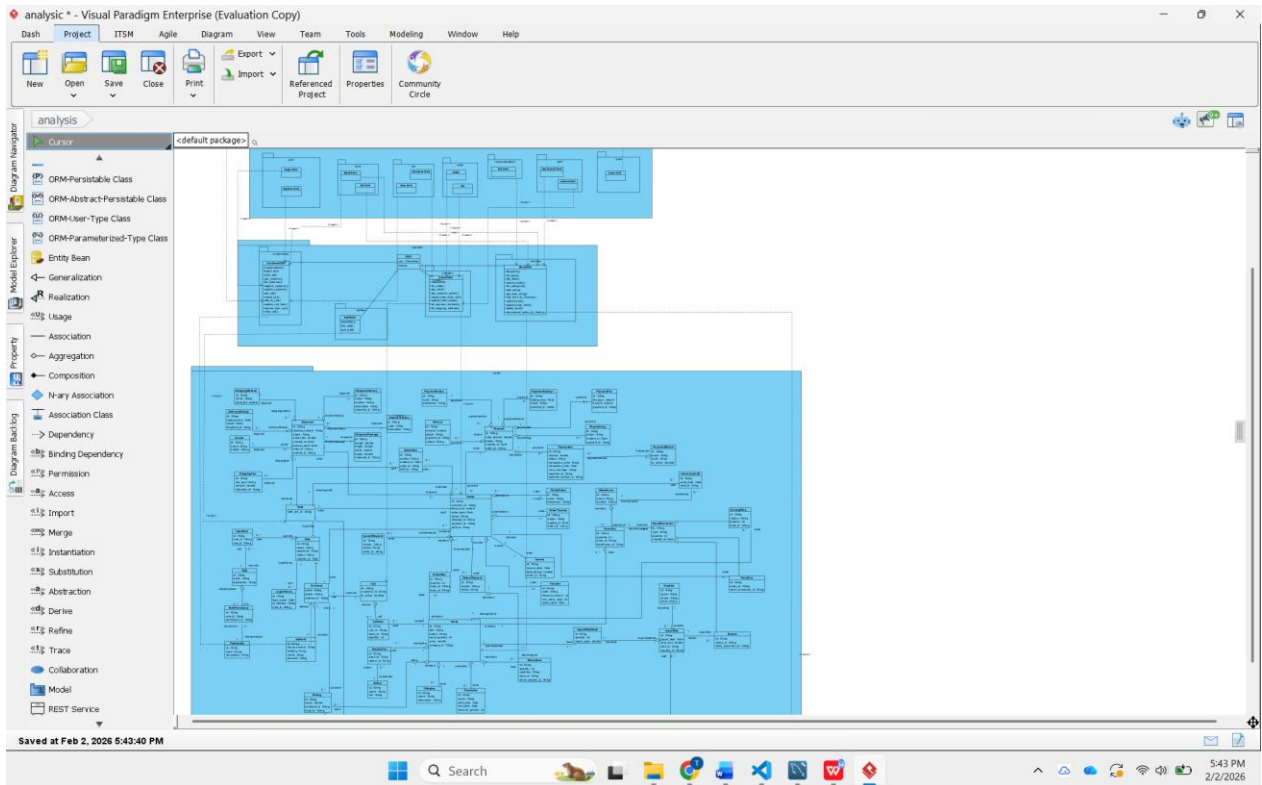
- **Model:** Phụ trách quản lý dữ liệu và xử lý logic nghiệp vụ của hệ thống. Model thường làm việc trực tiếp với cơ sở dữ liệu thông qua DAO hoặc ORM. Ví dụ trong hệ thống có thể là các đối tượng như User, Book, Order.
- **View:** Đảm nhiệm phần giao diện, hiển thị thông tin và tương tác với người dùng, thường được xây dựng bằng HTML, CSS và JavaScript. Trong bài tập, View có thể là các trang như login.html, registry.html.
- **Controller:** Đóng vai trò trung gian giữa View và Model. Controller nhận yêu cầu từ người dùng thông qua View, gọi Model để xử lý dữ liệu và sau đó trả kết quả về View phù hợp.

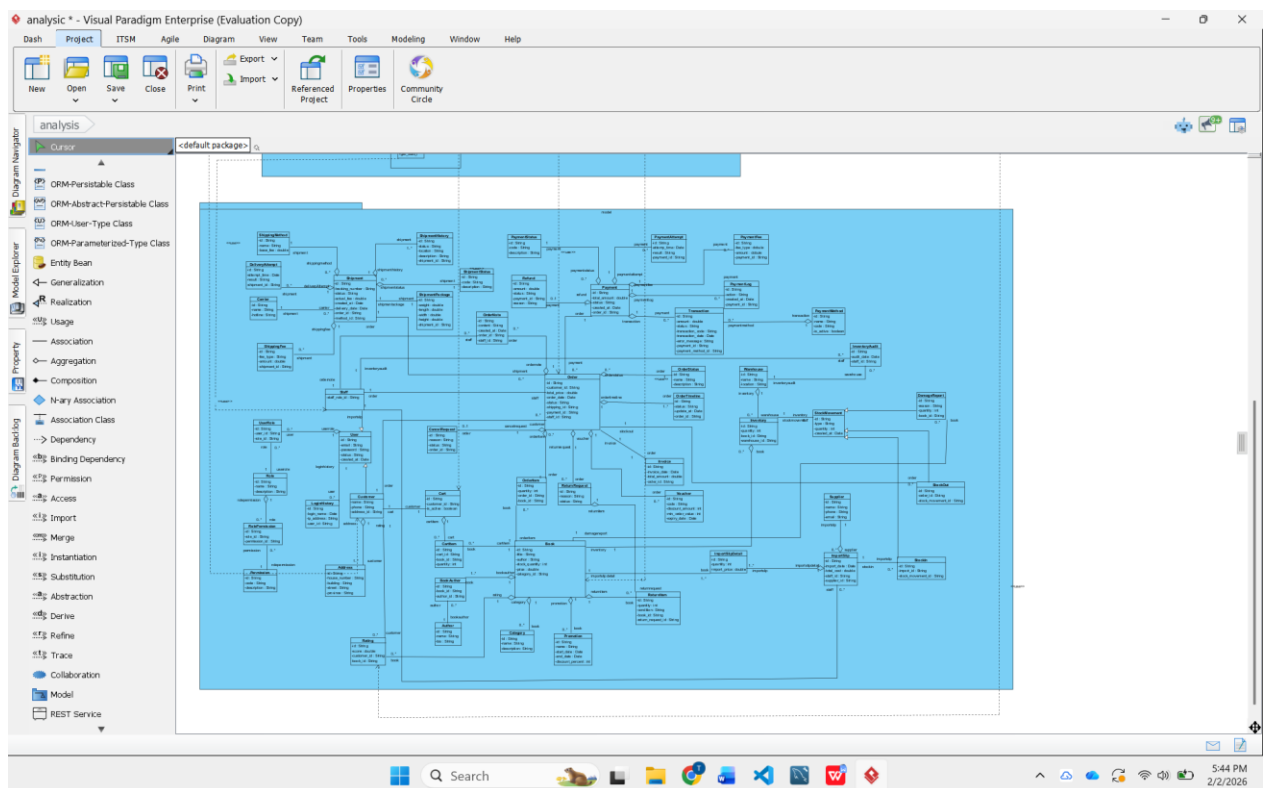
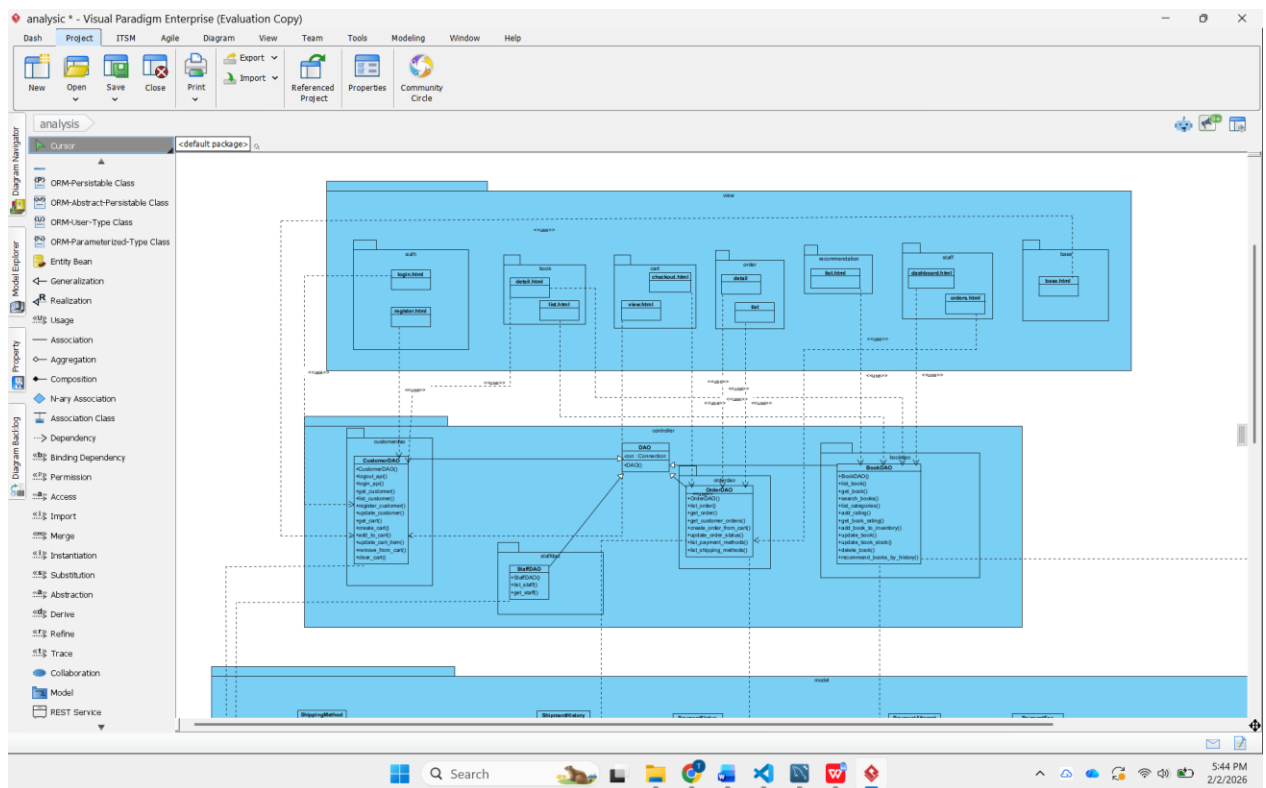
c) Lớp DAO (Data Access Object)

DAO là một mẫu thiết kế dùng để tách riêng phần làm việc với cơ sở dữ liệu ra khỏi phần xử lý nghiệp vụ của ứng dụng.

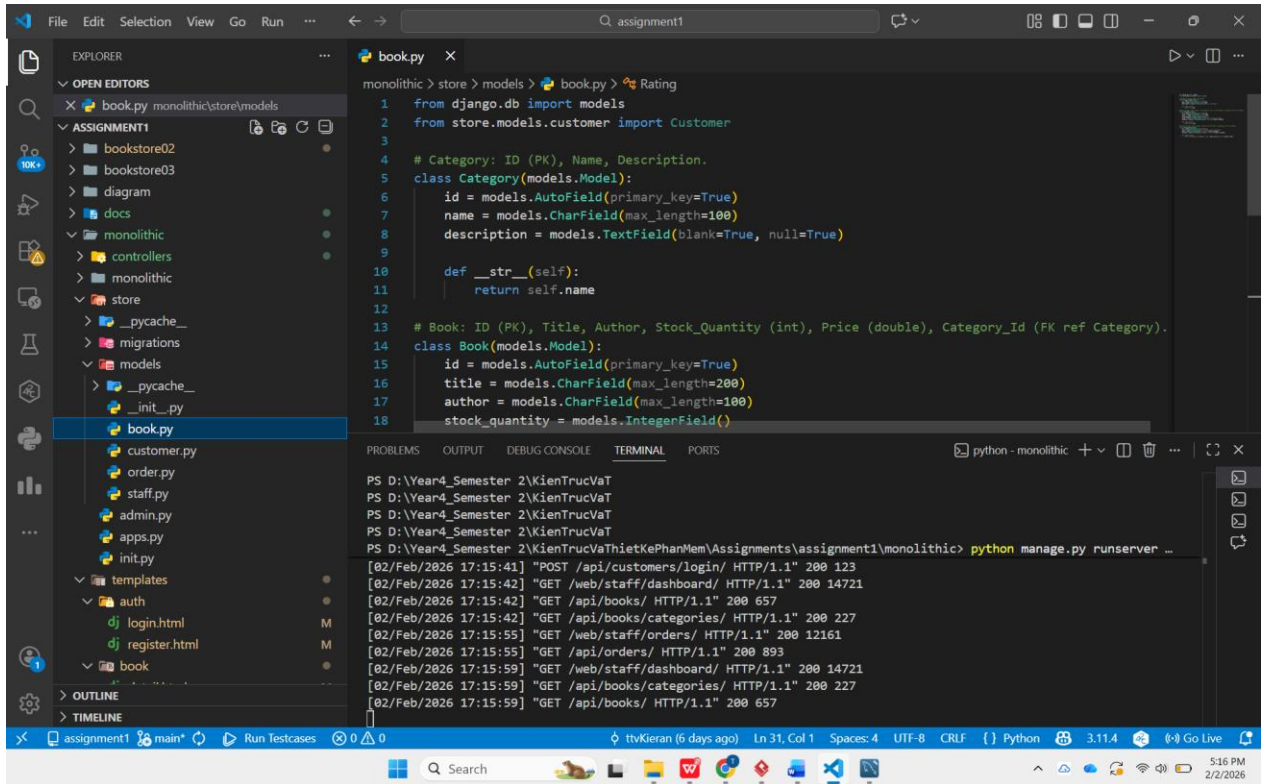
- **Vai trò:** DAO cung cấp các hàm thao tác dữ liệu cơ bản như thêm, sửa, xóa và truy vấn (CRUD) mà không để lộ chi tiết về câu lệnh SQL hay cách kết nối cơ sở dữ liệu.

- Áp dụng trong bài tập: Trong sơ đồ tuần tự, các lớp như BookDAO được sử dụng để làm việc với dữ liệu.
- Lợi ích: Việc thay đổi hệ quản trị cơ sở dữ liệu (ví dụ từ MySQL sang PostgreSQL theo yêu cầu bài toán) sẽ dễ dàng hơn, vì chỉ cần chỉnh sửa mã nguồn trong các lớp DAO mà không ảnh hưởng đến các phần khác của hệ thống.





Sinh code Django

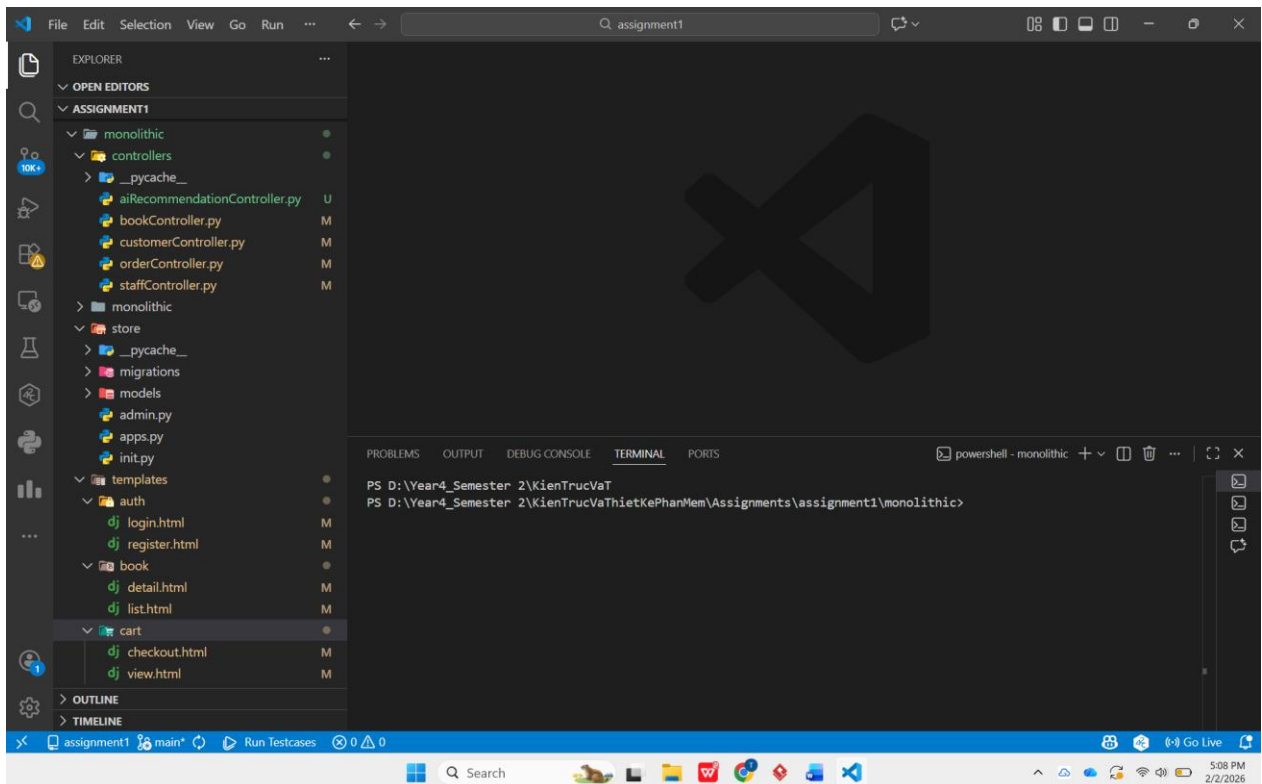


```
monolithic > store > models > book.py > Rating
1 from django.db import models
2 from store.models.customer import Customer
3
4 # Category: ID (PK), Name, Description.
5 class Category(models.Model):
6     id = models.AutoField(primary_key=True)
7     name = models.CharField(max_length=100)
8     description = models.TextField(blank=True, null=True)
9
10     def __str__(self):
11         return self.name
12
13 # Book: ID (PK), Title, Author, Stock_Quantity (int), Price (double), Category_Id (FK ref Category).
14 class Book(models.Model):
15     id = models.AutoField(primary_key=True)
16     title = models.CharField(max_length=200)
17     author = models.CharField(max_length=100)
18     stock_quantity = models.IntegerField()
```

```
PS D:\Year4_Semester 2\KienTrucVaT ThietKePhanMem\Assignments\assignment1\monolithic> python manage.py runserver ...
[02/Feb/2026 17:15:41] "POST /api/customers/login/ HTTP/1.1" 200 123
[02/Feb/2026 17:15:42] "GET /web/staff/dashboard/ HTTP/1.1" 200 14721
[02/Feb/2026 17:15:42] "GET /api/books/ HTTP/1.1" 200 657
[02/Feb/2026 17:15:42] "GET /api/books/categories/ HTTP/1.1" 200 227
[02/Feb/2026 17:15:55] "GET /web/staff/orders/ HTTP/1.1" 200 12161
[02/Feb/2026 17:15:55] "GET /api/orders/ HTTP/1.1" 200 893
[02/Feb/2026 17:15:59] "GET /web/staff/dashboard/ HTTP/1.1" 200 14721
[02/Feb/2026 17:15:59] "GET /api/books/categories/ HTTP/1.1" 200 227
[02/Feb/2026 17:15:59] "GET /api/books/ HTTP/1.1" 200 657
```

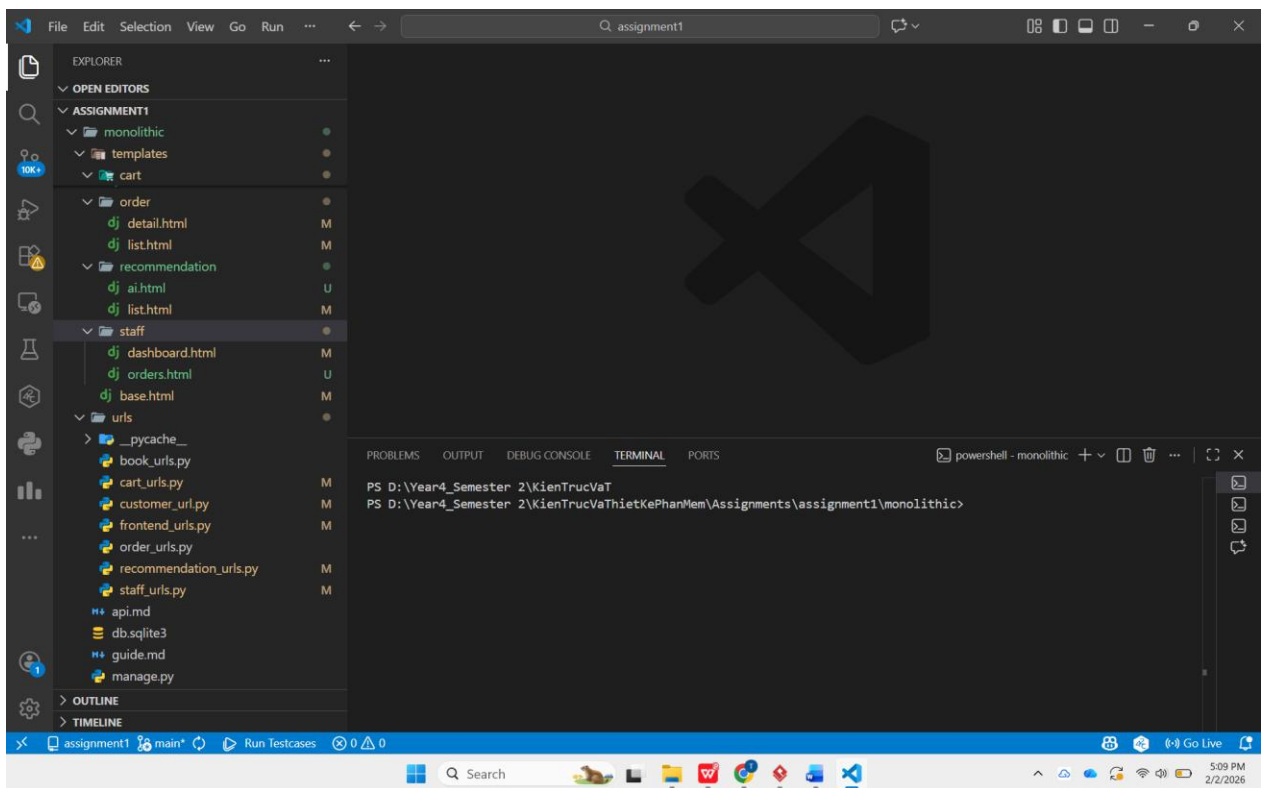
Chương 4: Cài đặt và triển khai

Tương ứng code và thiết kế

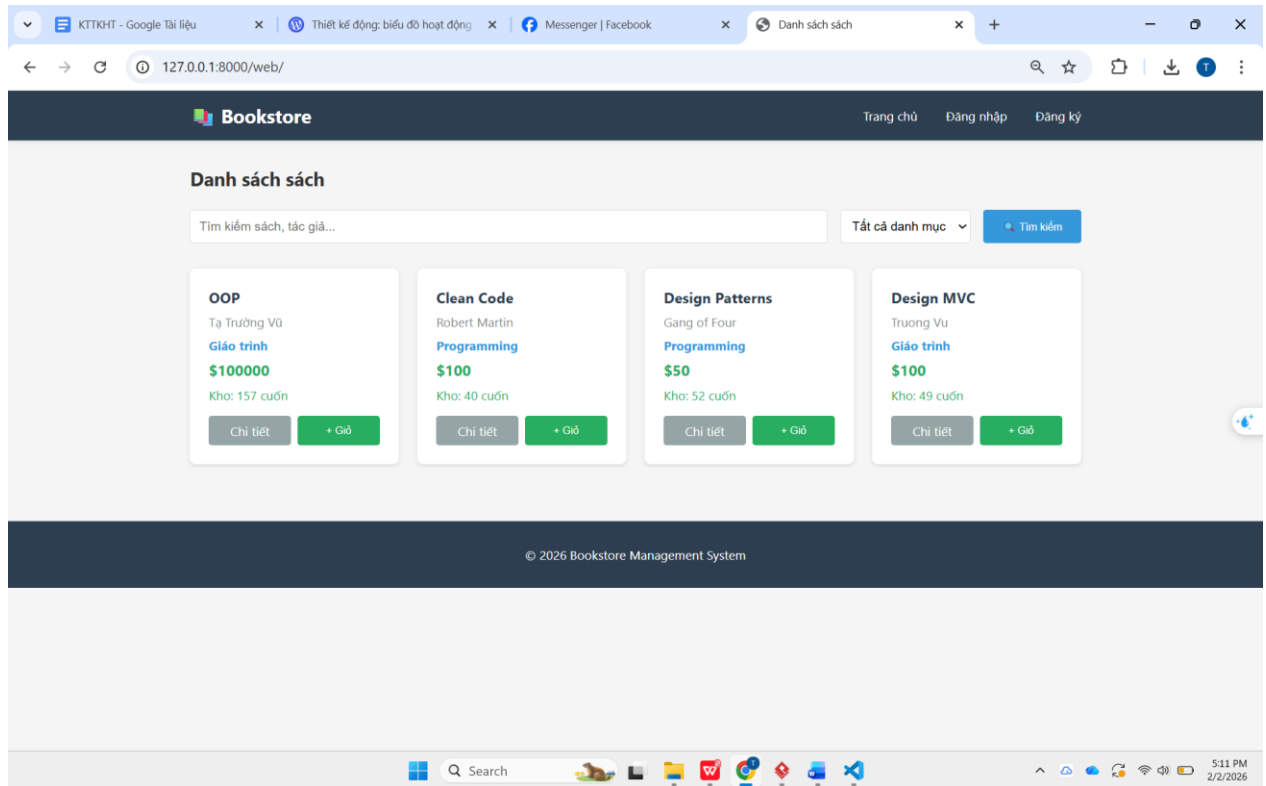


```
monolithic > controllers > aiRecommendationController.py
1 from django.db import models
2 from store.models.customer import Customer
3
4 # Category: ID (PK), Name, Description.
5 class Category(models.Model):
6     id = models.AutoField(primary_key=True)
7     name = models.CharField(max_length=100)
8     description = models.TextField(blank=True, null=True)
9
10     def __str__(self):
11         return self.name
12
13 # Book: ID (PK), Title, Author, Stock_Quantity (int), Price (double), Category_Id (FK ref Category).
14 class Book(models.Model):
15     id = models.AutoField(primary_key=True)
16     title = models.CharField(max_length=200)
17     author = models.CharField(max_length=100)
18     stock_quantity = models.IntegerField()
```

```
PS D:\Year4_Semester 2\KienTrucVaT ThietKePhanMem\Assignments\assignment1\monolithic> python manage.py runserver ...
[02/Feb/2026 17:15:41] "POST /api/customers/login/ HTTP/1.1" 200 123
[02/Feb/2026 17:15:42] "GET /web/staff/dashboard/ HTTP/1.1" 200 14721
[02/Feb/2026 17:15:42] "GET /api/books/ HTTP/1.1" 200 657
[02/Feb/2026 17:15:42] "GET /api/books/categories/ HTTP/1.1" 200 227
[02/Feb/2026 17:15:55] "GET /web/staff/orders/ HTTP/1.1" 200 12161
[02/Feb/2026 17:15:55] "GET /api/orders/ HTTP/1.1" 200 893
[02/Feb/2026 17:15:59] "GET /web/staff/dashboard/ HTTP/1.1" 200 14721
[02/Feb/2026 17:15:59] "GET /api/books/categories/ HTTP/1.1" 200 227
[02/Feb/2026 17:15:59] "GET /api/books/ HTTP/1.1" 200 657
```



Màn hình giao diện



KTTKHT - Google Tài liệu

Thiết kế động: biểu đồ ho

Messenger | Facebook

Giỏ hàng

Select customer to change

127.0.0.1:8000/web/cart/

Bookstore

Trang chủ

AI Gợi ý

Gợi ý

Giỏ hàng

Đơn hàng

Trương Vũ

Đăng xuất

Giỏ hàng của bạn

Sách	Giá	Số lượng	Tổng	Hành động
OOP undefined	\$100000	2	\$200000	Xóa
Tổng cộng:			\$200000	
← Tiếp tục mua sắm			Thanh toán →	

© 2026 Bookstore Management System

KTTKHT - Google Tài liệu

Thiết kế động: biểu đồ ho

Messenger | Facebook

Lịch sử đơn hàng

Select customer to change

127.0.0.1:8000/web/orders/

Bookstore

Trang chủ

AI Gợi ý

Gợi ý

Giỏ hàng

Đơn hàng

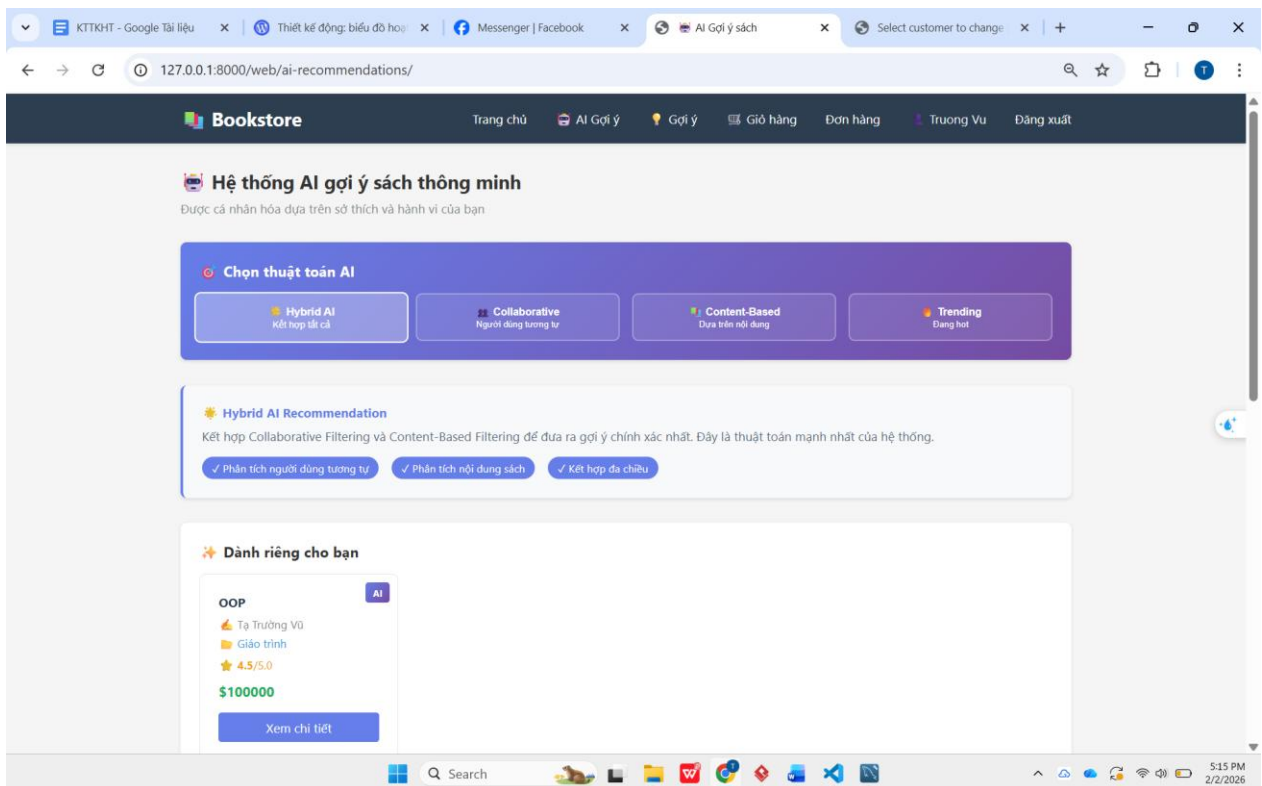
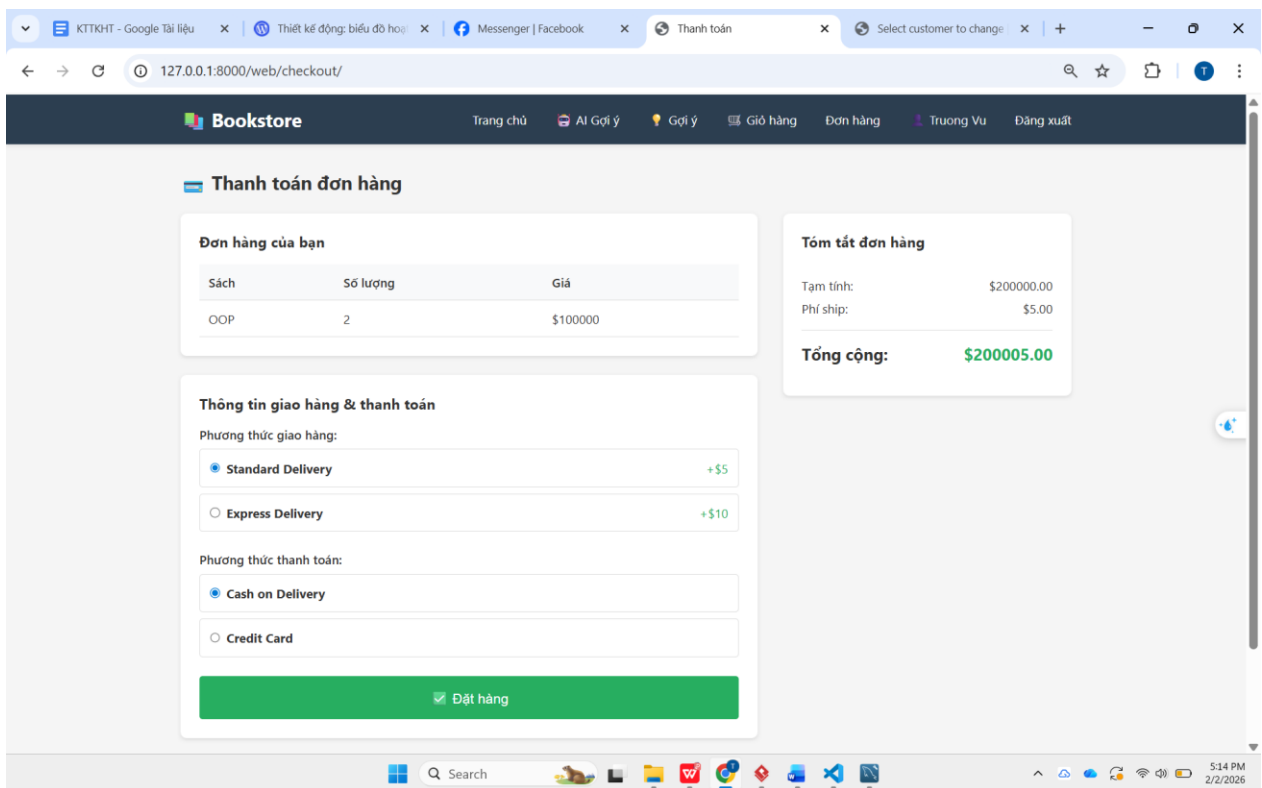
Trương Vũ

Đăng xuất

Lịch sử đơn hàng

Mã đơn	Ngày đặt	Tổng tiền	Trạng thái	Giao hàng	Thanh toán	Hành động
#6	26/1/2026	\$100105	Pending	Standard Delivery	Cash on Delivery	Chi tiết
#5	26/1/2026	\$405	Pending	Standard Delivery	Cash on Delivery	Chi tiết
#4	26/1/2026	\$100105	Pending	Standard Delivery	Cash on Delivery	Chi tiết
#3	26/1/2026	\$100010	Pending	Express Delivery	Credit Card	Chi tiết

© 2026 Bookstore Management System



Bookstore Trang chủ Quản lý kho Quản lý đơn hàng Admin Staff Đăng xuất

Quản lý kho hàng

+ Thêm sách mới

Tên sách: Tác giả:

Giá (\$): Số lượng:

Danh mục:

Thêm sách

Kho sách hiện tại

ID	Tên sách	Tác giả	Danh mục	Giá	Tồn kho	Hành động
1	OOP	Tạ Trường Vũ	Giáo trình	\$100000	157	
2	Clean Code	Robert Martin	Programming	\$100	40	
3	Design Patterns	Gang of Four	Programming	\$50	52	
4	Design MVC	Truong Vu	Giáo trình	\$100	49	

Bookstore Trang chủ Quản lý kho Quản lý đơn hàng Admin Staff Đăng xuất

Quản lý đơn hàng

Lọc theo trạng thái:

Mã đơn	Khách hàng	Nhân viên	Ngày đặt	Tổng tiền	Trạng thái	Hành động
#1	Nguyen Van B	Admin Staff	26/1/2026	\$300005	Shipping	Xem -- Cập nhật --
#2	Nguyen Van B	Admin Staff	26/1/2026	\$55	Pending	Xem -- Cập nhật --
#3	Truong Vu	Admin Staff	26/1/2026	\$100010	Pending	Xem -- Cập nhật --
#4	Truong Vu	Admin Staff	26/1/2026	\$100105	Pending	Xem -- Cập nhật --
#5	Truong Vu	Admin Staff	26/1/2026	\$405	Pending	Xem -- Cập nhật --
#6	Truong Vu	Admin Staff	26/1/2026	\$100105	Pending	Xem -- Cập nhật --

© 2026 Bookstore Management System