A red oval with a star in the middle

AI-generated content may be incorrect.

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN 1**

**--o0o--**

**KHOA CÔNG NGHỆ THÔNG TIN I**

**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](#_Toc220949051)

[Lập Bảng các actor và các chức năng tương ứng 1](#_Toc220949052)

[Lập Bảng các lớp, các thuộc tính và các chức năng trong lớp đó 1](#_Toc220949053)

[Biểu đồ lớp Phân tích > 50 lớp 5](#_Toc220949054)

[Vẽ 1 biểu đồ hoạt động và 1 biểu đồ tuần tự. Trình bày ý nghĩa các biểu đồ này 5](#_Toc220949055)

[Biểu đồ hoạt động: 5](#_Toc220949056)

[Biểu đồ tuần tự: 6](#_Toc220949057)

[Chương 2: Data model và Data base 7](#_Toc220949058)

[Data model 7](#_Toc220949059)

[Database MySQL 8](#_Toc220949060)

[Chương 3: Thiết kế 9](#_Toc220949061)

[Trình bày hiểu biết về DAO, MVC và mô hình monolithic 9](#_Toc220949062)

[Sinh code Django 12](#_Toc220949063)

[Chương 4: Cài đặt và triển khai 12](#_Toc220949064)

[Tương ứng code và thiết kế 12](#_Toc220949065)

[Màn hình giao diện 14](#_Toc220949066)

[14](#_Toc220949067)

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

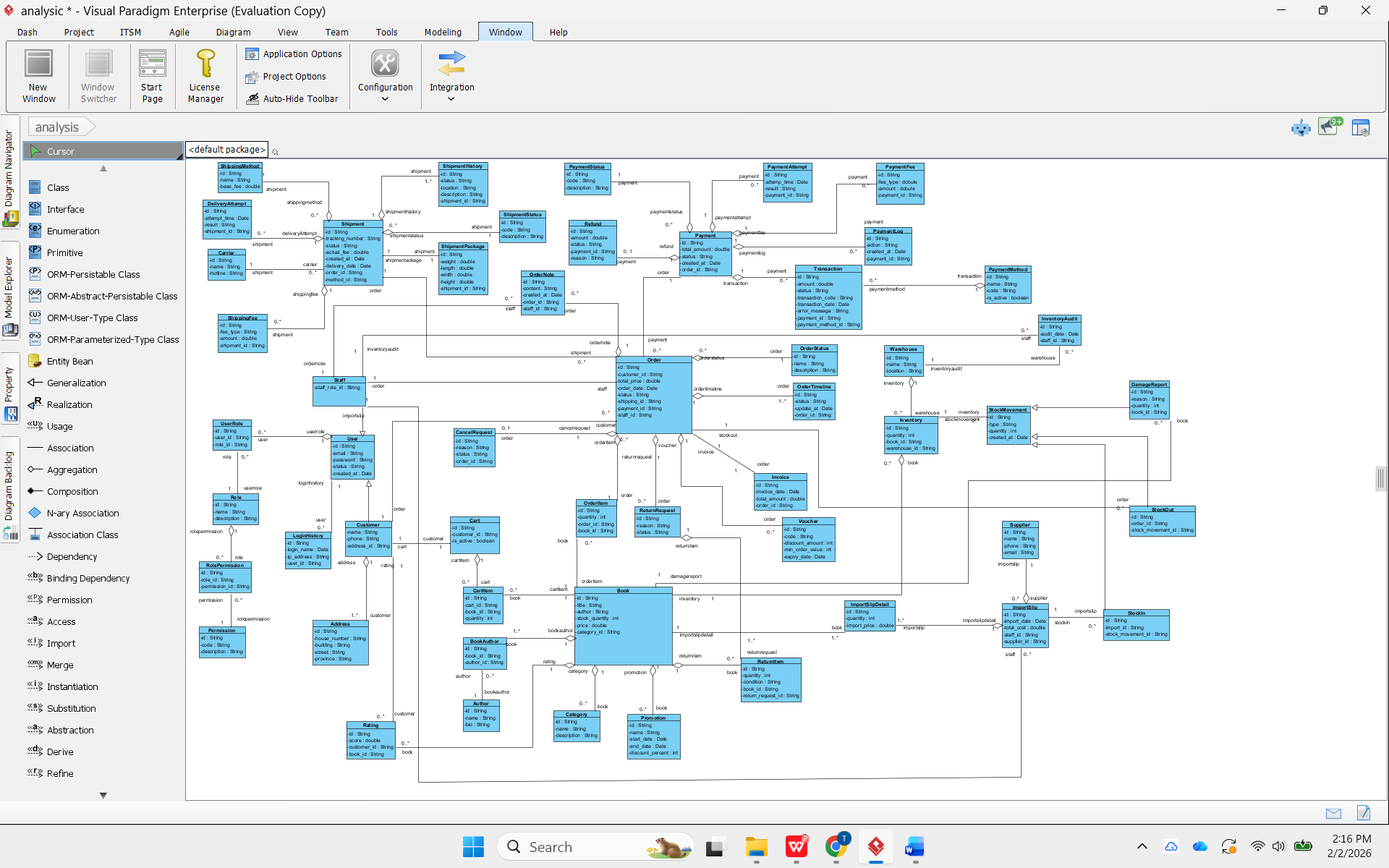
## Lập Bảng các actor và các chức năng tương ứng

|  |  |
| --- | --- |
| **Actor** | **Chức năng** |
| Khách hàng | * 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 | * 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 | * 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 | * Tạo yêu cầu thanh toán cho đơn hàng |
| Hệ thống vận chuyển | * Nhận yêu cầu giao hàng từ đơn hàng * Lấy thông tin vận chuyển |
| Module kho & tồn kho | * 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 | * 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

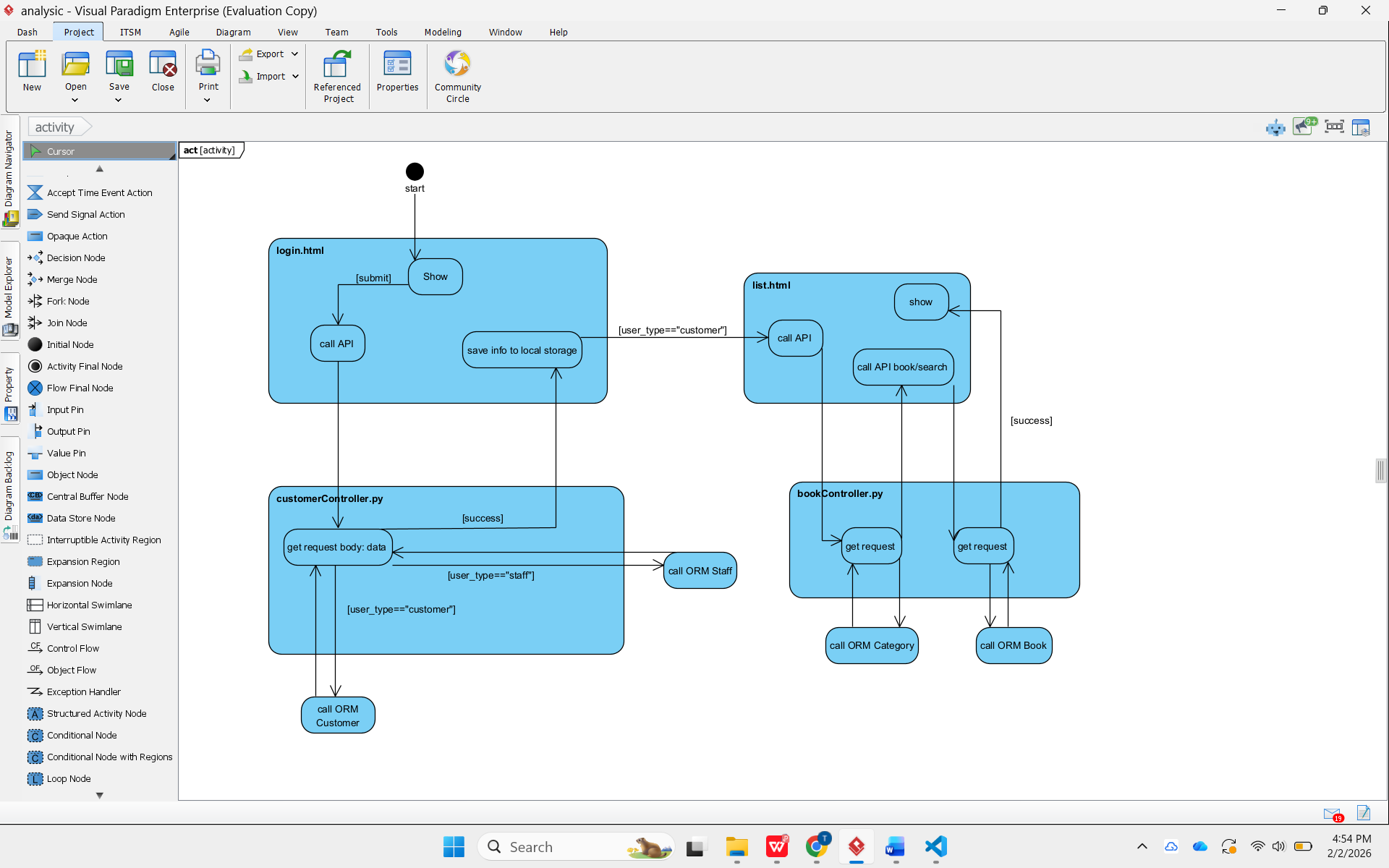


## Vẽ 1 biểu đồ hoạt động và 1 biểu đồ tuần tự. Trình bày ý nghĩa các biểu đồ này

### Biểu đồ hoạt động:

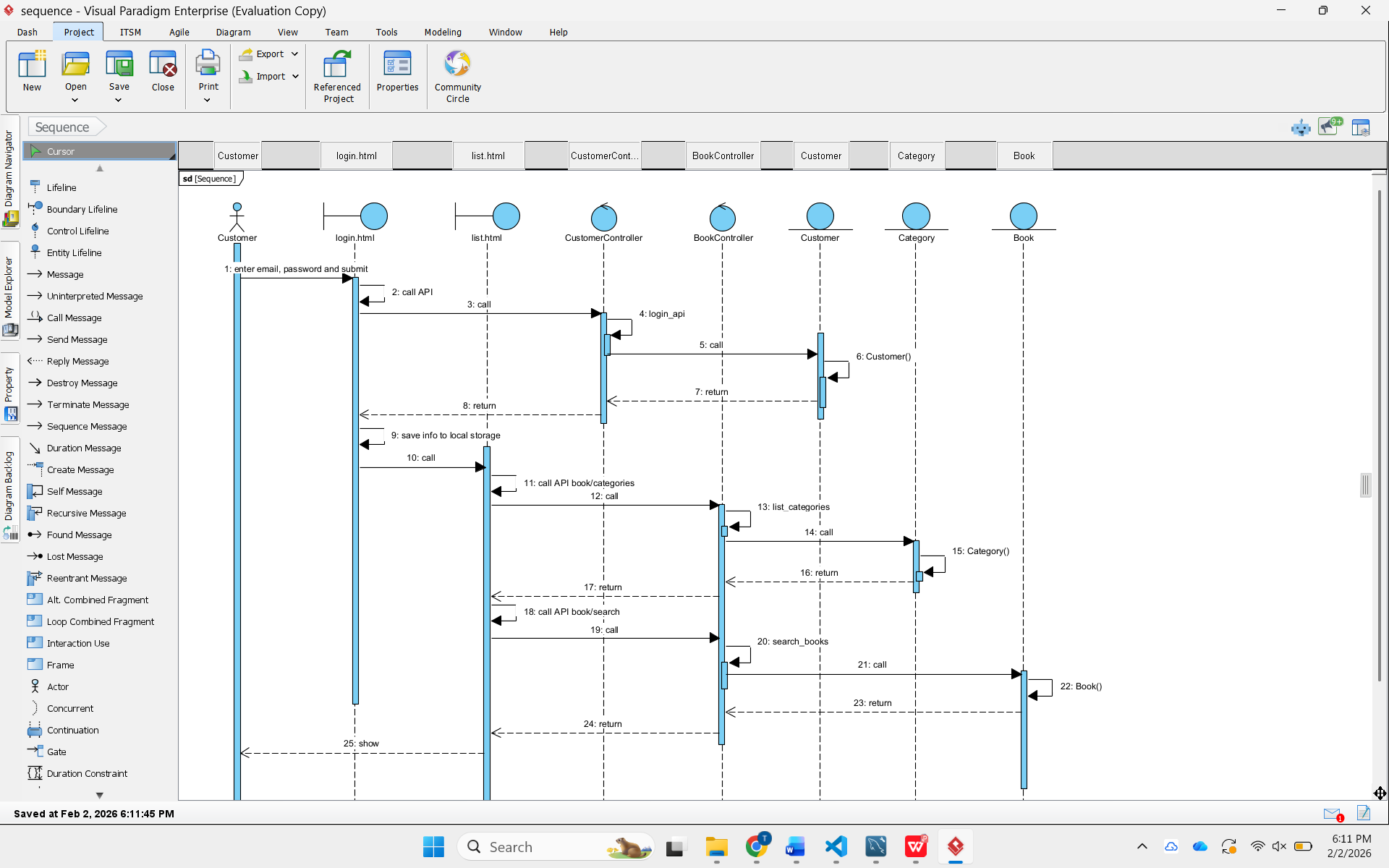
Biểu diễn các hoạt động và sự đồng bộ, chuyển tiếp các hoạt động của hệ thống trong một lớp hoặc kết hợp giữa các lớp với nhau trong một chức năng cụ thể.

* 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



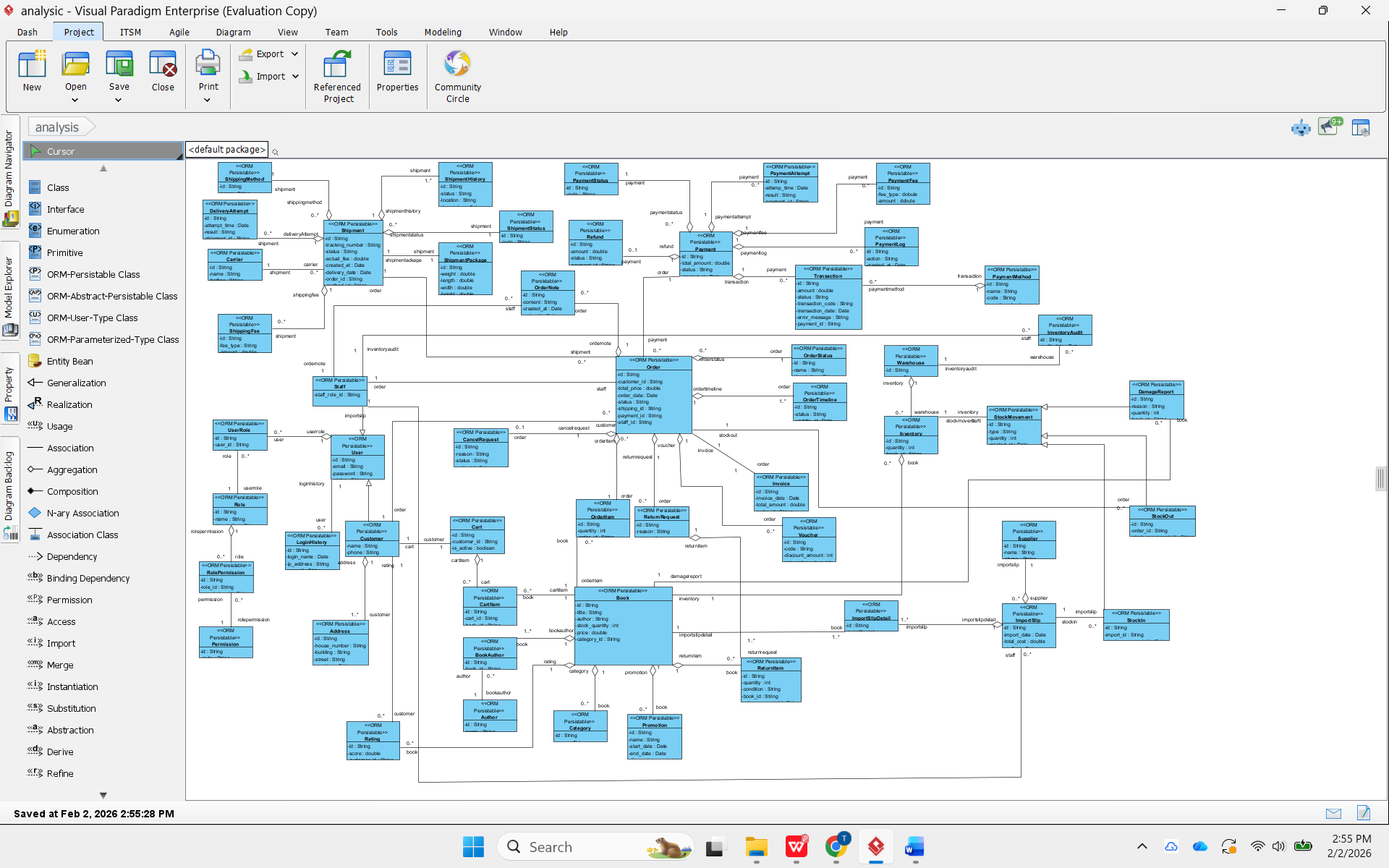
### Biểu đồ tuần tự:

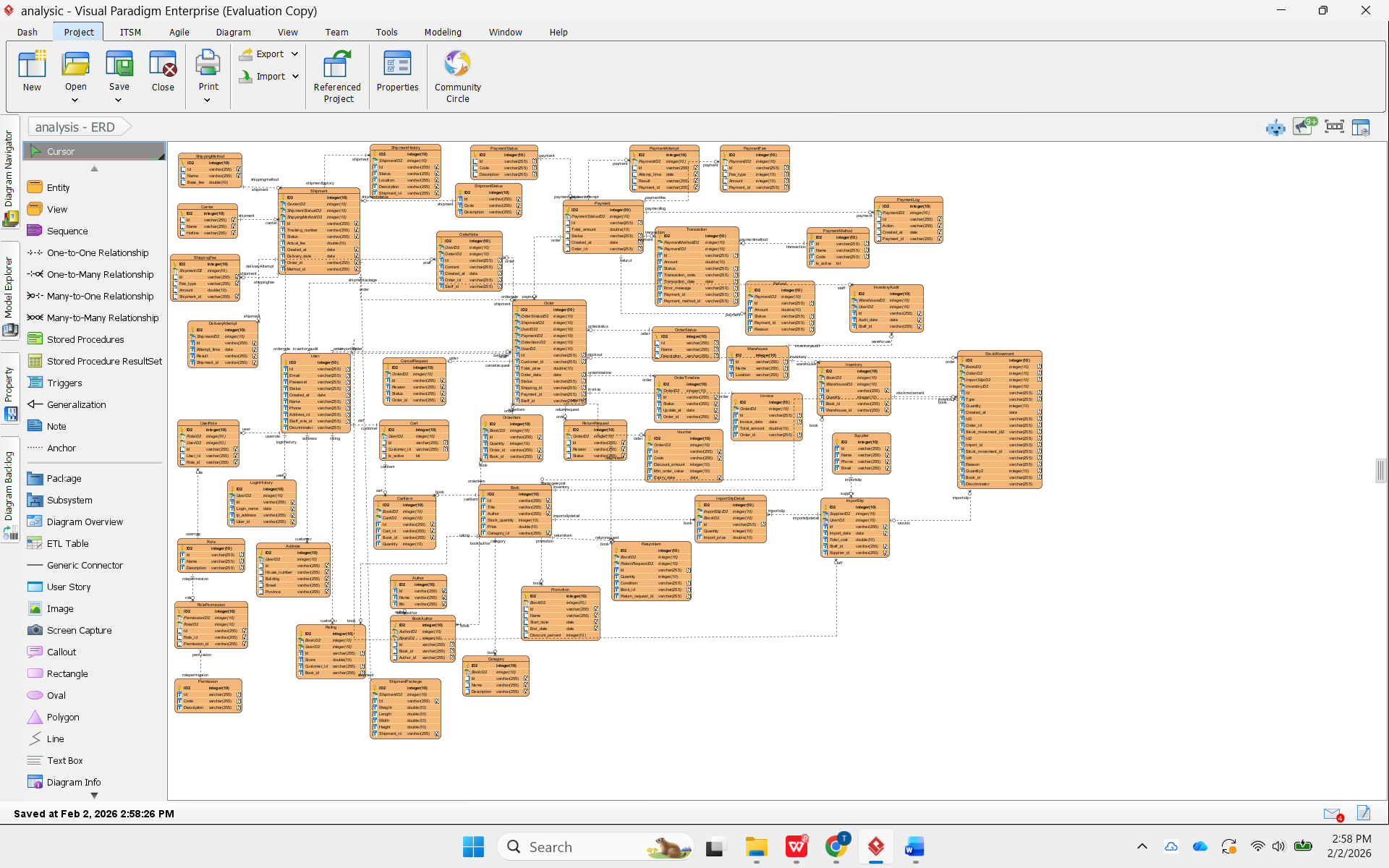
Biểu diễn mối quan hệ giữa các đối tượng, giữa các đối tượng và tác nhân theo thứ tự thời gian. Biểu đồ tuần tự nhấn mạnh thứ tự thực hiện của các tương tác.



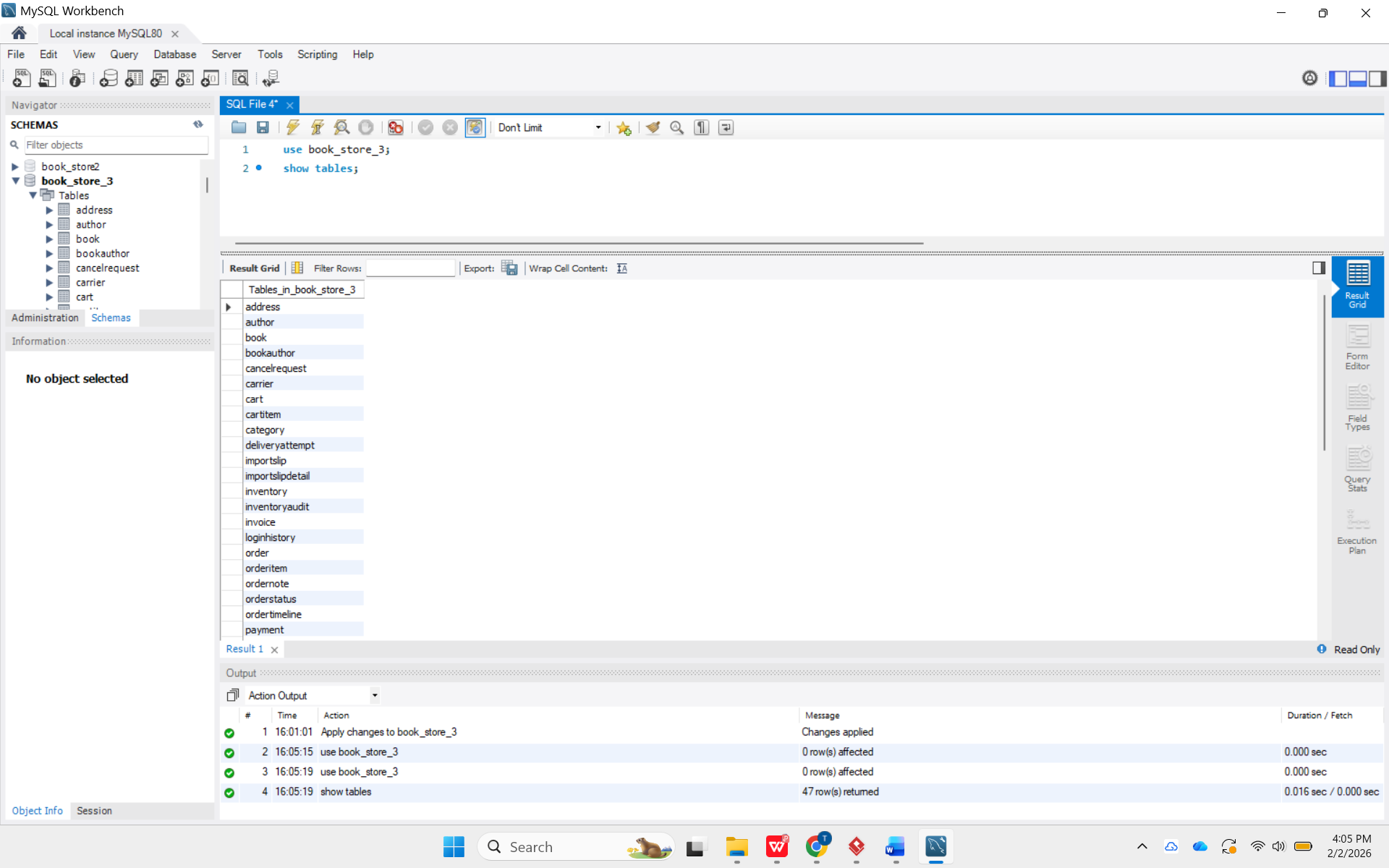
# Chương 2: Data model và Data base

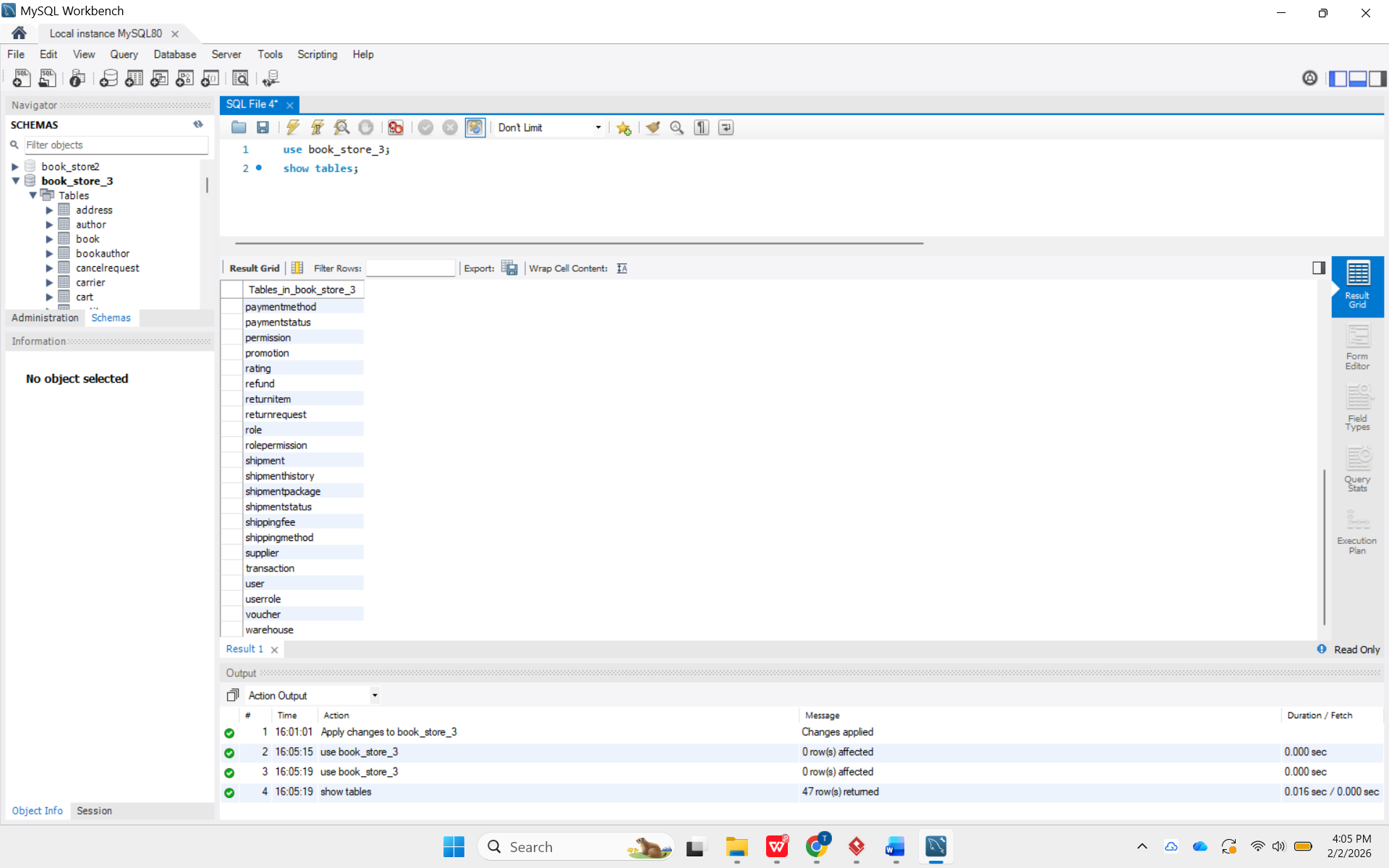
## Data model





## 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)

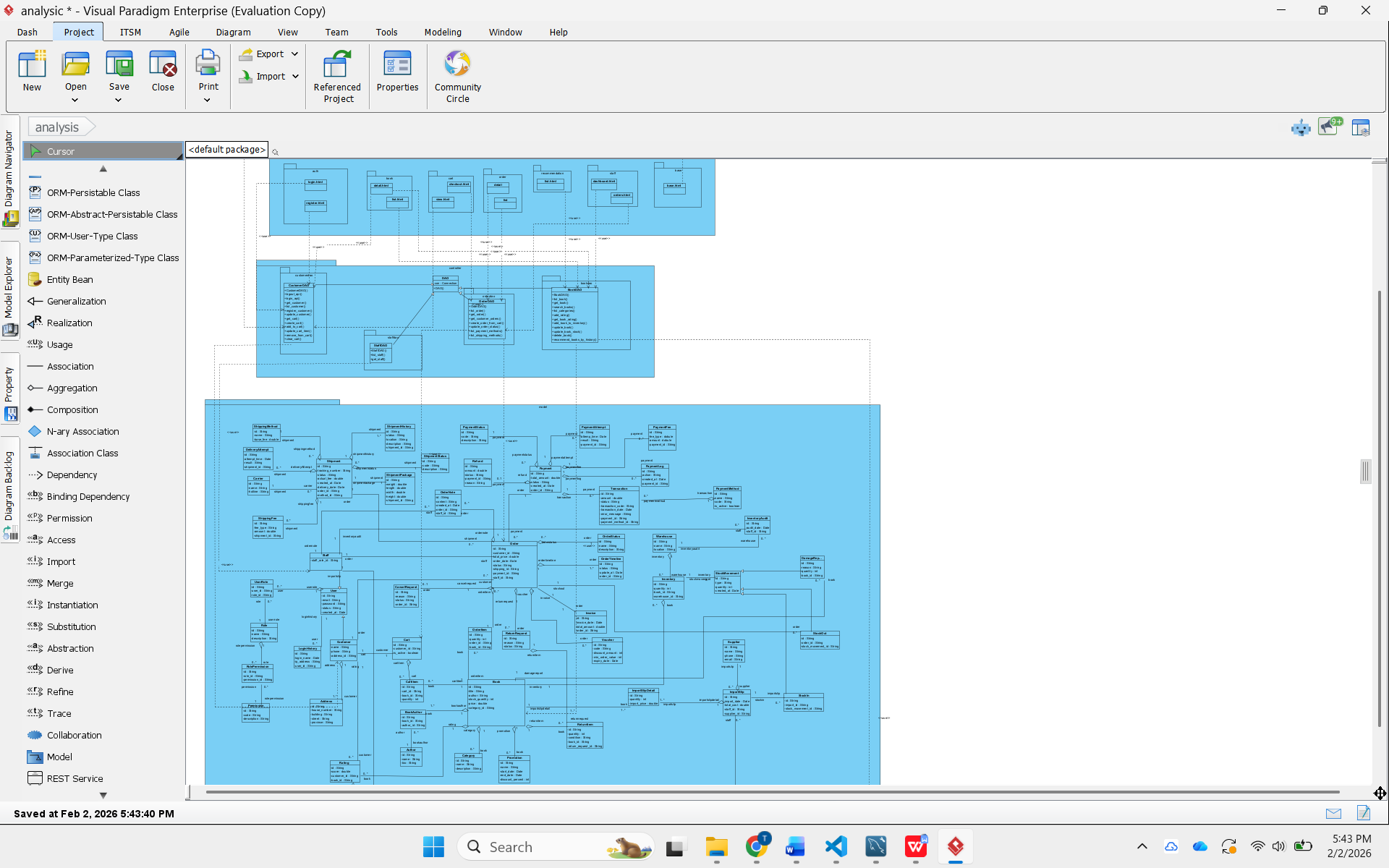
MVC là mô hình kiến trúc giúp chia ứng dụng thành ba phần riêng biệt nhằm tăng tính rõ ràng và dễ bảo trì trong quá trình phát triển.

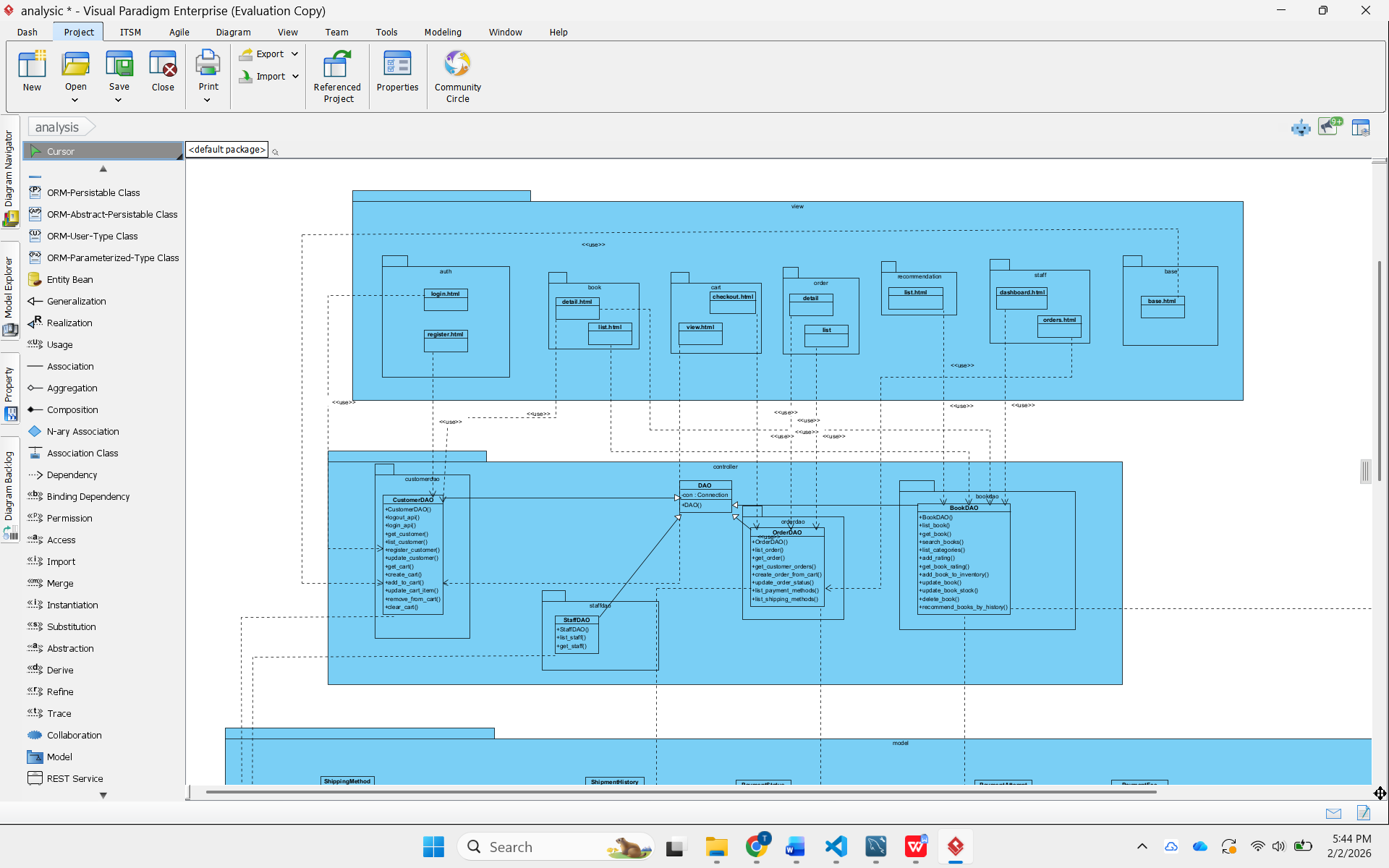
* Model: Phụ trách quản lý dữ liệu và xử lý logic nghiệp vụ của hệ thống. Model thường làm việc trực tiếp với cơ sở dữ liệu thông qua DAO hoặc ORM. Ví dụ trong hệ thống có thể là các đối tượng như User, Book, Order.
* View: Đảm nhiệm phần giao diện, hiển thị thông tin và tương tác với người dùng, thường 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.

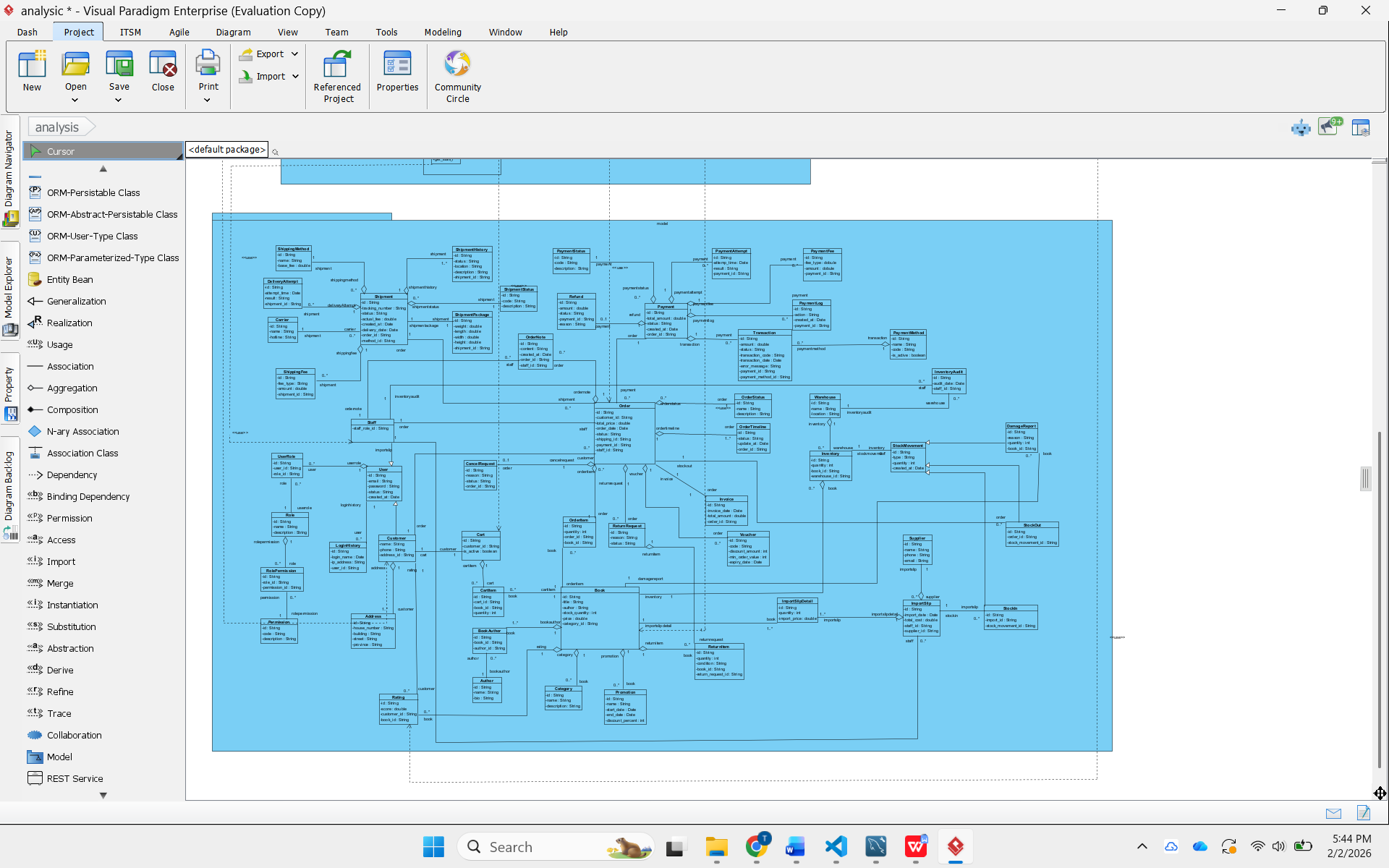
c) Lớp DAO (Data Access Object)

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.

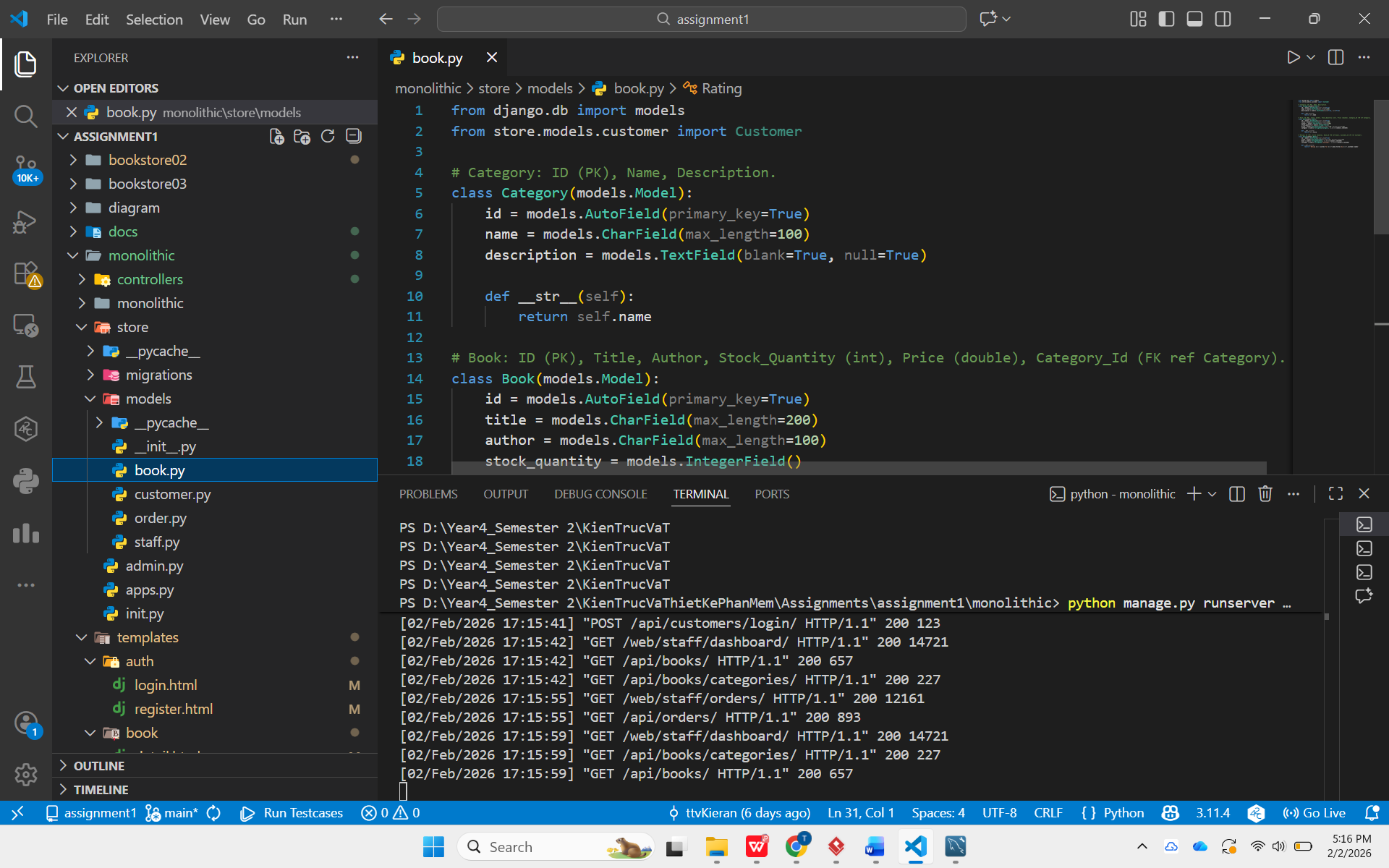
* Vai trò: DAO cung cấp các hàm thao tác dữ liệu cơ bản như thêm, sửa, xóa và truy vấn (CRUD) mà không để lộ chi tiết về câu lệnh SQL hay cách kết nối cơ sở dữ liệu.
* Áp dụng trong bài tập: Trong sơ đồ tuần tự, các lớp như BookDAO được sử dụng để làm việc với dữ liệu.
* Lợi ích: Việc thay đổi hệ quản 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.





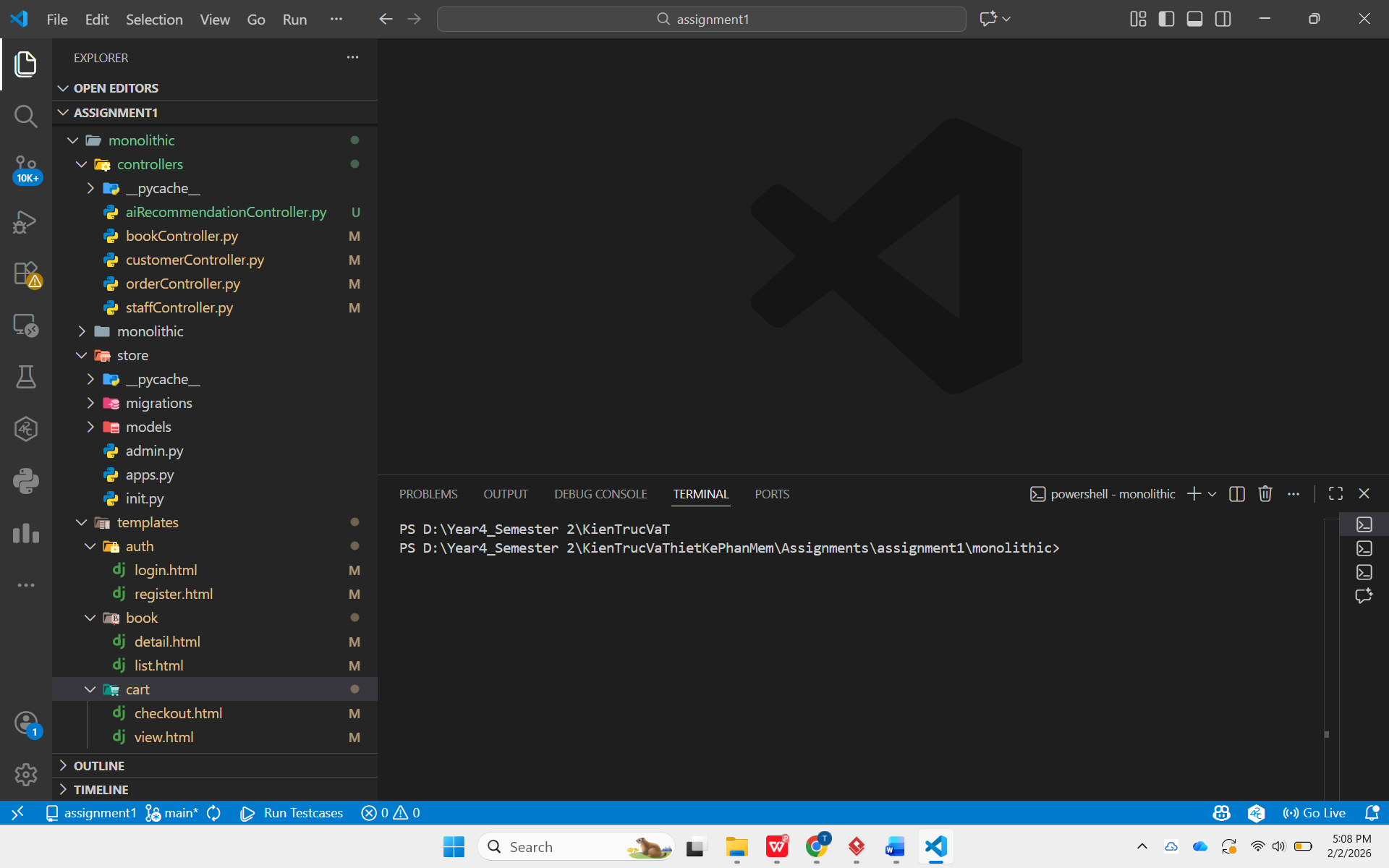


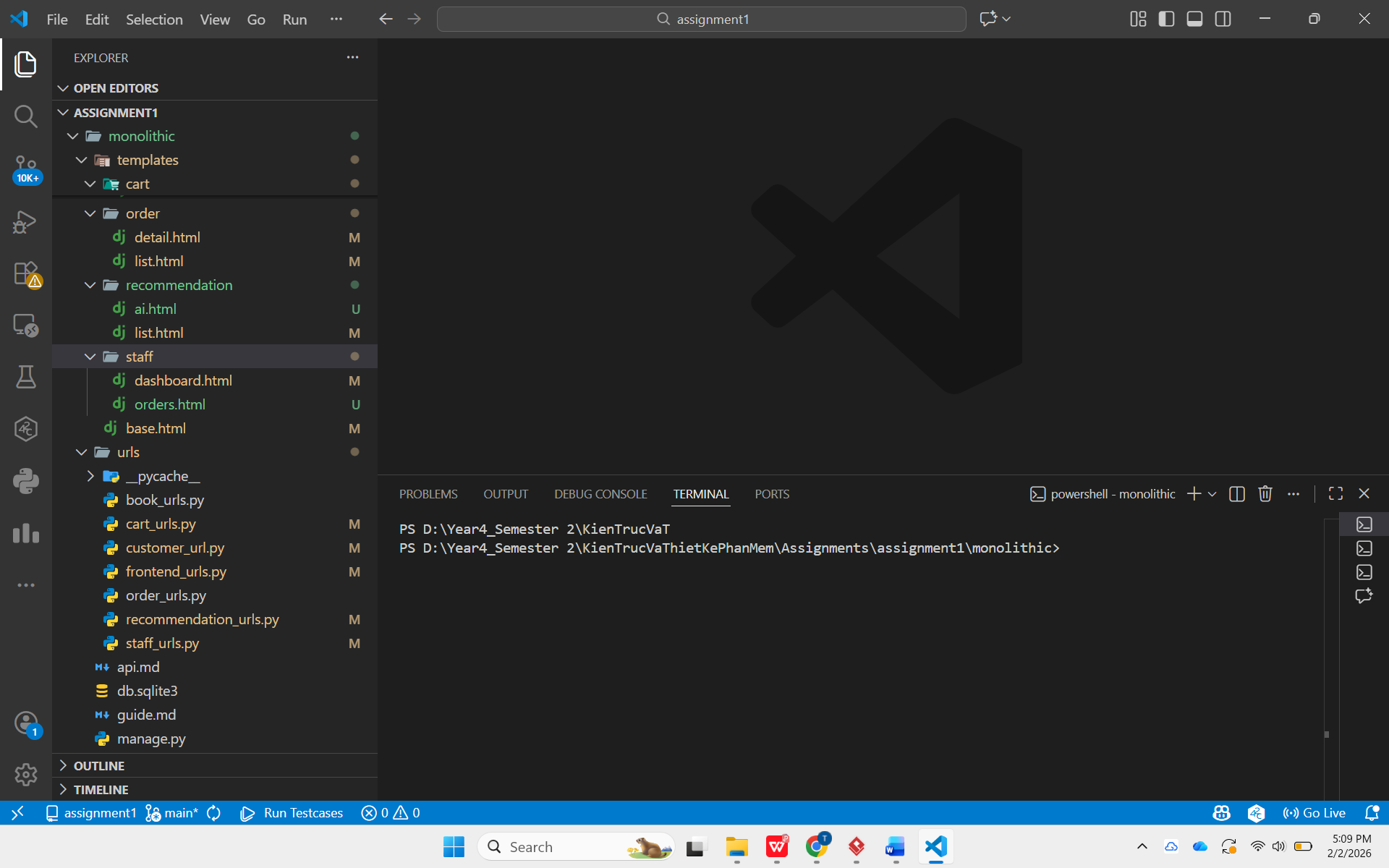
## Sinh code Django



# Chương 4: Cài đặt và triển khai

## Tương ứng code và thiết kế





## Màn hình giao diện

## 

