HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN

-----ജ📖ഗ-----

Assignment 1

Giảng viên : Trần Đình Quế

Sinh viên thực hiện:

Họ Tên: Trần Đức Lợi

Mã sinh viên: B22DCCN511

Lớp : D22CNPM04

Nhóm học phần: 03

Nhóm bài tập: 03

Hà Nội, ngày 19/1/2026

I.Giới thiệu

Báo cáo này trình bày quá trình phân tích, thiết kế và triển khai hệ thống Book Store Web System theo ba kiến trúc phần mềm phổ biến:

- Monolithic Architecture
- Clean Architecture
- Microservices Architecture

Hệ thống được xây dựng bằng Django Framework và sử dụng MySQL làm hệ quản trị cơ sở dữ liệu. Thông qua bài tập này, sinh viên có cơ hội hiểu rõ cách tổ chức mã nguồn, thiết kế UML và áp dụng các kiến trúc phần mềm vào một hệ thống thực tế.

II. TỔNG QUAN BÀI TOÁN (PROJECT OVERVIEW)

1. Mô tả hệ thống

Book Store Web System là một hệ thống bán sách trực tuyến cho phép khách hàng:

- Đăng ký và đăng nhập tài khoản
- Xem danh sách các cuốn sách có sẵn
- Thêm sách vào giỏ hàng
- Xem nội dung giỏ hàng

Hệ thống hướng đến việc mô phỏng một website bán sách đơn giản, tập trung vào việc tổ chức kiến trúc phần mềm hơn là giao diện người dùng.

2. Các thực thể trong hệ thống (Entities)

Hệ thống bao gồm các thực thể chính sau:

- Customer
    - id
    - name
    - email
    - password
- Book
    - id
    - title
    - author
    - price
    - stock
- Cart
    - id

- customer_id
- created_at

- CartItem
  - id
  - cart_id
  - book_id
  - quantity

Các thực thể này được thiết kế để thể hiện mối quan hệ giữa khách hàng, sách và giỏ hàng trong hệ thống.

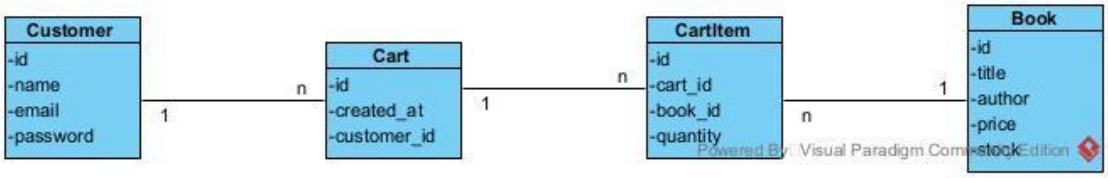
III. PHÂN TÍCH YÊU CẦU CHỨC NĂNG (FUNCTIONAL REQUIREMENTS)

Hệ thống đáp ứng các yêu cầu chức năng sau:

1. Customer Registration & Login
   Người dùng có thể đăng ký tài khoản mới và đăng nhập để sử dụng hệ thống.
2. View Book Catalog
   Người dùng có thể xem danh sách các sách hiện có trong cửa hàng.
3. Add Book to Shopping Cart
   Người dùng có thể thêm một hoặc nhiều sách vào giỏ hàng.
4. View Shopping Cart
   Người dùng có thể xem các sách đã thêm vào giỏ hàng cùng với số lượng tương ứng.

IV. THIẾT KẾ UML (UML DESIGN)

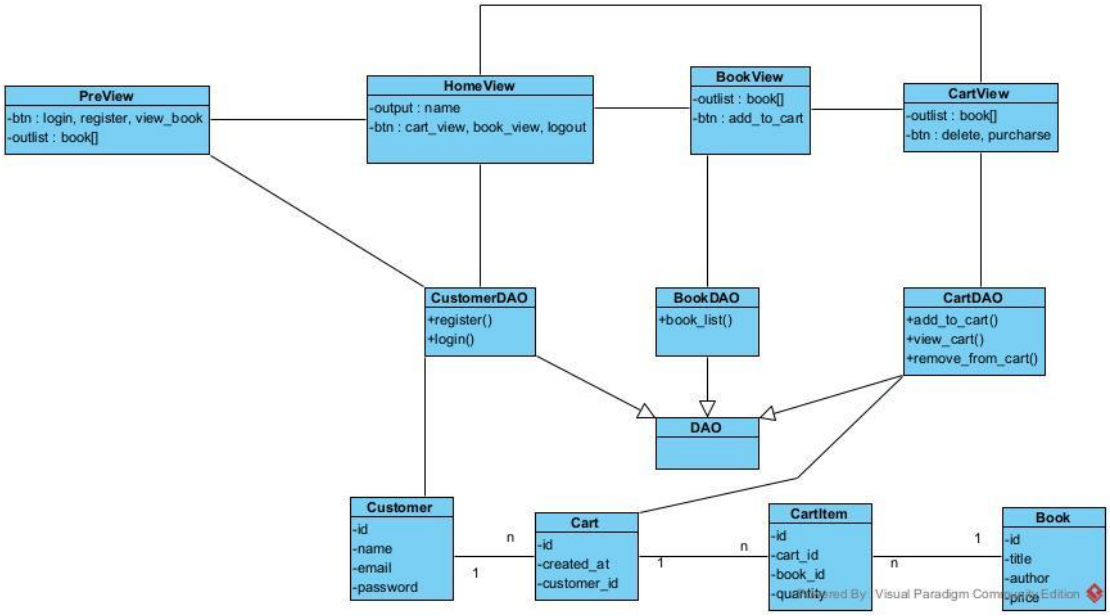
1. Class Diagram



2. Monolithic Architecture Diagram (MVC)

Trong kiến trúc Monolithic, toàn bộ hệ thống được triển khai trong một project Django duy nhất theo mô hình MVC (Model – View – Controller).

- Model: Quản lý dữ liệu và tương tác với cơ sở dữ liệu
- View: Xử lý giao diện và response
- Controller (View trong Django): Xử lý logic nghiệp vụ

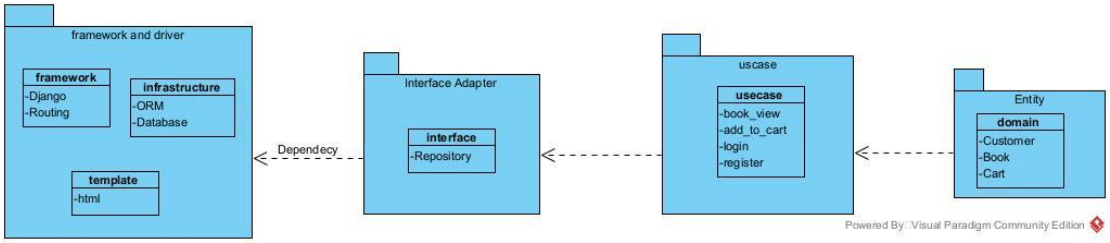


3. Clean Architecture Diagram

Clean Architecture chia hệ thống thành các layer độc lập, giúp giảm sự phụ thuộc vào framework và dễ bảo trì hơn.

Các layer chính:

- Domain: Chứa các entity và business rules cốt lõi
- UseCases: Xử lý nghiệp vụ của hệ thống
- Interfaces: Định nghĩa interface cho các thành phần
- Infrastructure: Làm việc với database và external services
- Framework (Django): Layer ngoài cùng



4. Microservices Architecture Diagram

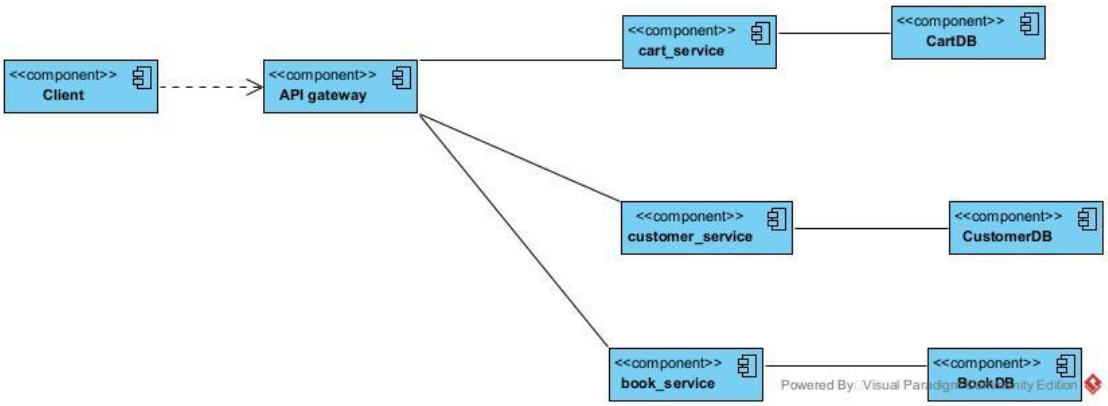
Trong kiến trúc Microservices, hệ thống được chia thành các service độc lập, mỗi service đảm nhiệm một chức năng riêng và giao tiếp với nhau thông qua REST API.

Các service bao gồm:

- customer-service
- book-service

- cart-service

Mỗi service có cơ sở dữ liệu riêng, đảm bảo tính độc lập và khả năng mở rộng.



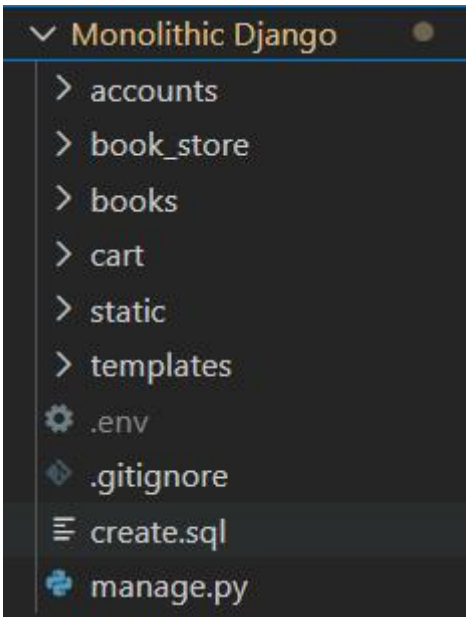
V. TRIỂN KHAI HỆ THỐNG (IMPLEMENTATION)

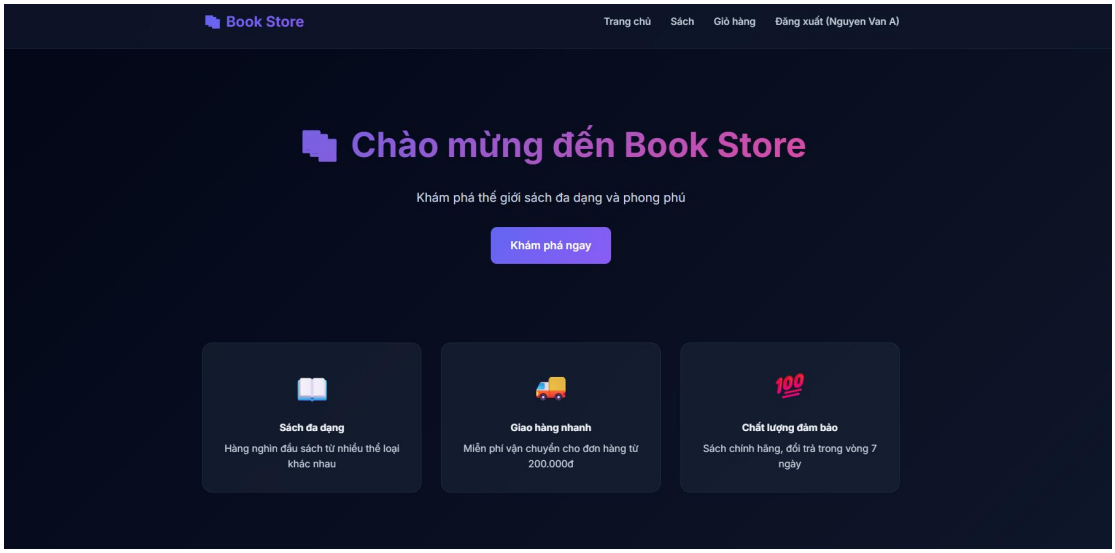
1. Version A – Monolithic Django

Phiên bản Monolithic được triển khai dưới dạng một project Django duy nhất với các app:

- accounts
- books
- cart

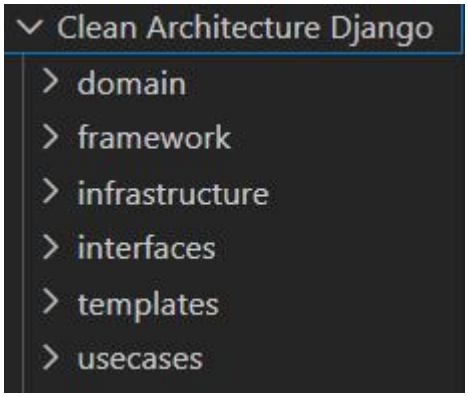
Toàn bộ hệ thống sử dụng chung một cơ sở dữ liệu MySQL. Kiến trúc này đơn giản, dễ triển khai nhưng khó mở rộng khi hệ thống lớn dần.



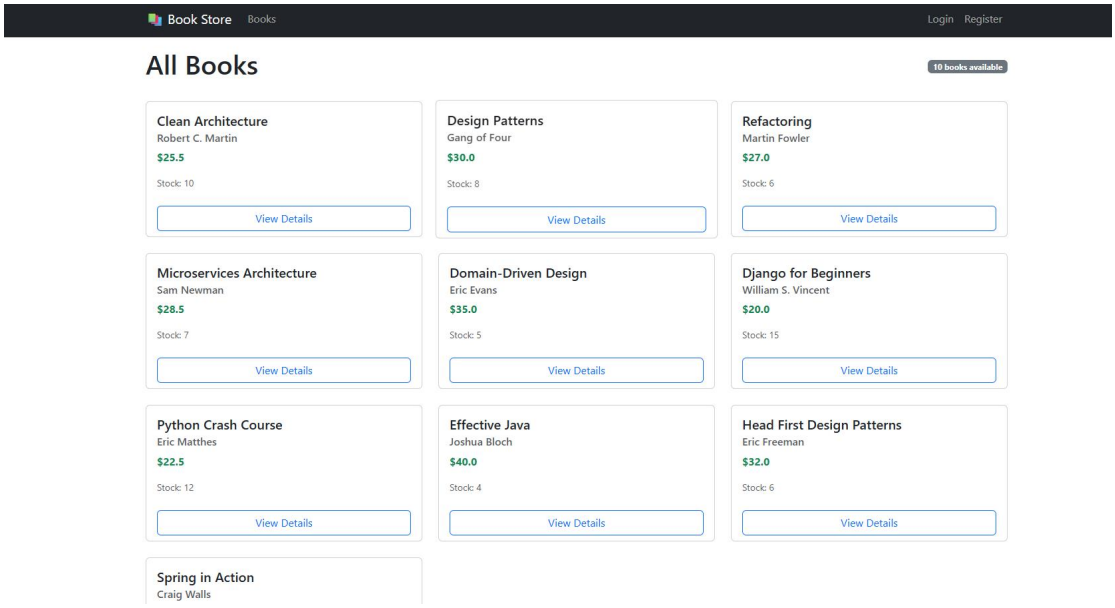


2. Version B – Clean Architecture Django

Phiên bản Clean Architecture được tổ chức theo cấu trúc:



Kiến trúc này giúp tách biệt logic nghiệp vụ và framework, dễ dàng bảo trì và mở rộng hệ thống trong tương lai.

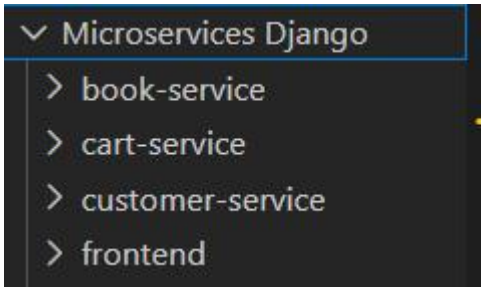


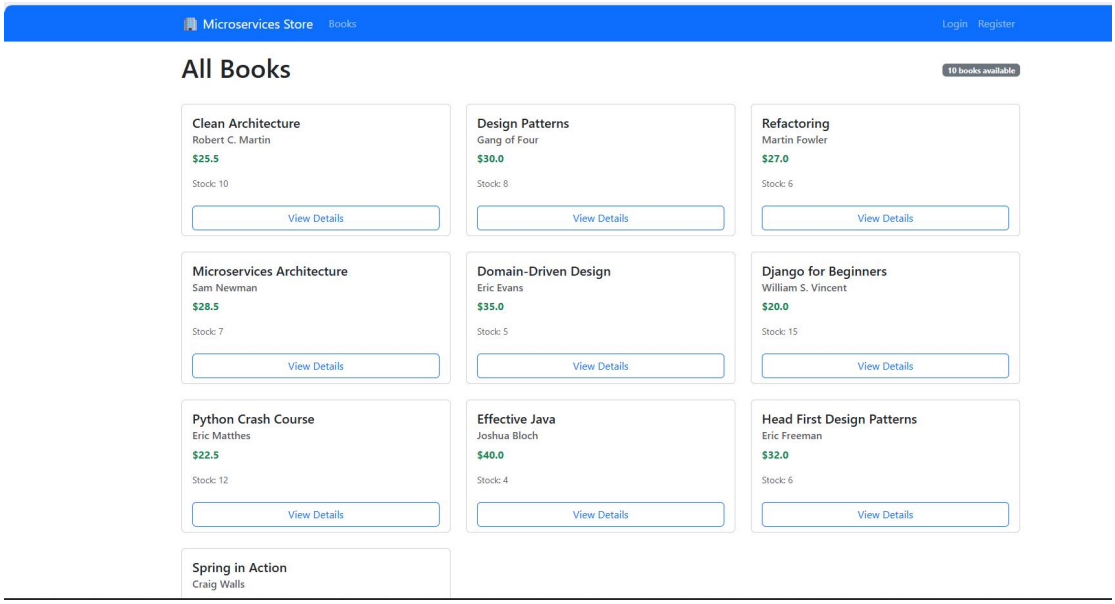
3. Version C – Microservices Django

Phiên bản Microservices chia hệ thống thành các service độc lập:

- customer-service
- book-service
- cart-service

Các service giao tiếp với nhau thông qua REST API. Mỗi service có database riêng và có thể triển khai độc lập.

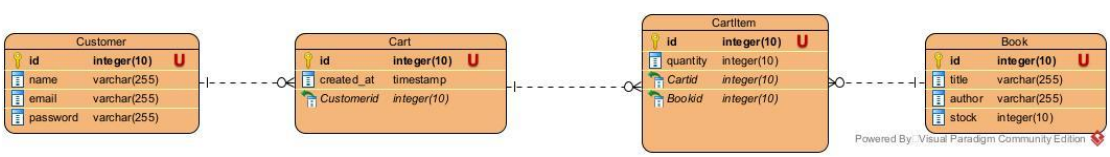




VI. THIẾT KẾ CƠ SỞ DỮ LIỆU (DATABASE DESIGN)

Hệ thống sử dụng MySQL làm hệ quản trị cơ sở dữ liệu.
Mỗi phiên bản (Monolithic, Clean, Microservices) đều có cơ sở dữ liệu riêng nhưng cùng 1 cấu trúc.

- Các bảng được thiết kế tương ứng với các entity
- Cung cấp SQL script để tạo bảng



VII. SO SÁNH CÁC KIẾN TRÚC

Tiêu chí	Monolithic	Clean Architecture	Microservices
Độ phức tạp	Thấp	Trung bình	Cao
Dễ bảo trì	Thấp	Cao	Cao
Khả năng mở rộng	Kém	Tốt	Rất tốt
Triển khai	Dễ	Trung bình	Phức tạp

VIII. KẾT LUẬN (CONCLUSION)

Qua bài Assignment này, hệ thống Book Store Web System đã được thiết kế và triển khai thành công theo ba kiến trúc phần mềm khác nhau. Mỗi kiến trúc có những ưu điểm và hạn chế riêng:

- Monolithic phù hợp với hệ thống nhỏ, đơn giản
- Clean Architecture phù hợp với dự án cần bảo trì lâu dài
- Microservices phù hợp với hệ thống lớn, yêu cầu mở rộng cao

Bài tập giúp sinh viên hiểu sâu hơn về kiến trúc phần mềm và cách áp dụng lý thuyết vào thực tế.