

## **Đồ án môn Phân tích và thiết kế phần mềm**

### **Thiết kế kiến trúc Ứng dụng quản lý thư viện**

Version 1.0

Sinh viên thực hiện:  
21424034 – Nguyễn Hiếu Nghĩa  
21424045 - Đinh Thành Phú  
21424052 – Đỗ Văn Thành  
21424067 - Nguyễn Minh Văn



Ứng dụng quản lý thư viện

Phiên bản: 1.0

Thiết kế kiến trúc

Ngày: 25/12/2022

## Bảng ghi nhận thay đổi tài liệu

Ngày	Phiên bản	Mô tả	Người thay đổi
25/12/2022	1.0	Tài liệu thiết kế kiến trúc	Nguyễn Minh Văn Đinh Thành Phú Nguyễn Hiếu Nghĩa Đỗ Văn Thành

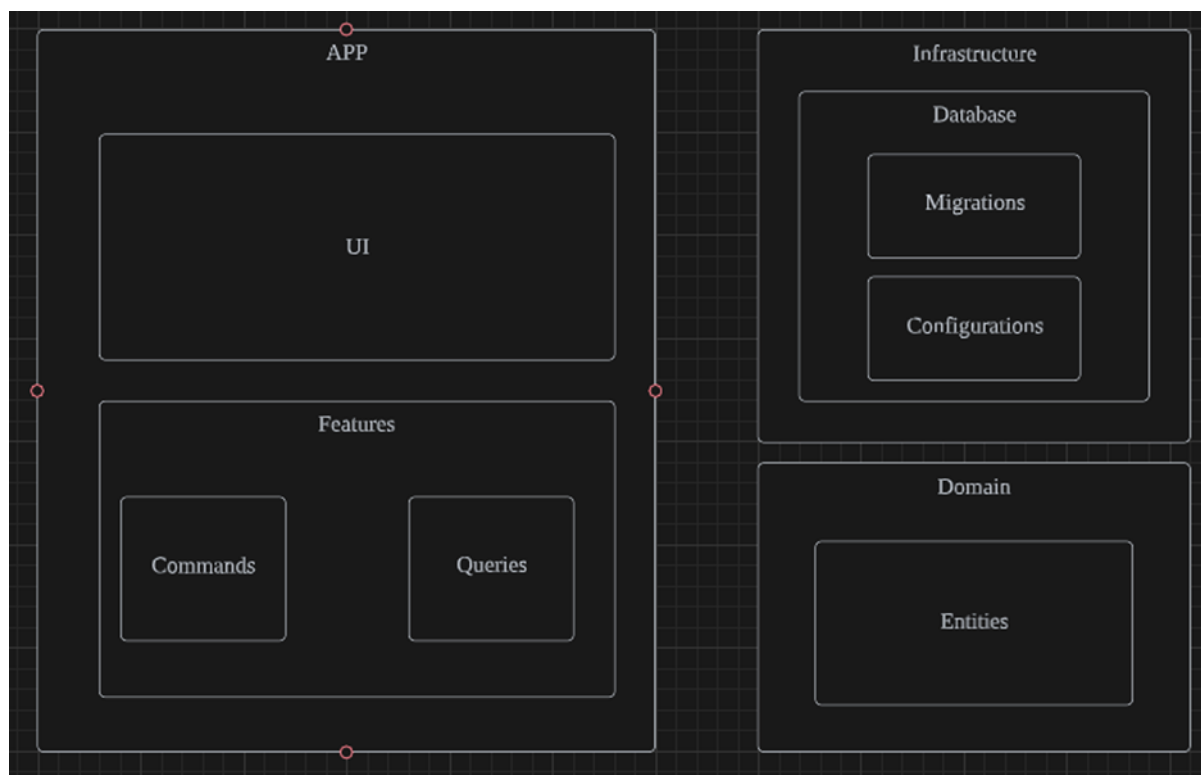


Ứng dụng quản lý thư viện	Phiên bản: 1.0
Thiết kế kiến trúc	Ngày: 25/12/2022

## Mục lục

- |  |   |
|--|---|
| 1. Kiến trúc hệ thống                            | 3 |
| 2. Mô tả chi tiết từng thành phần trong hệ thống | 4 |

## 1. Kiến trúc hệ thống



Sau đó, liệt kê danh sách các module, thành phần trong hệ thống:

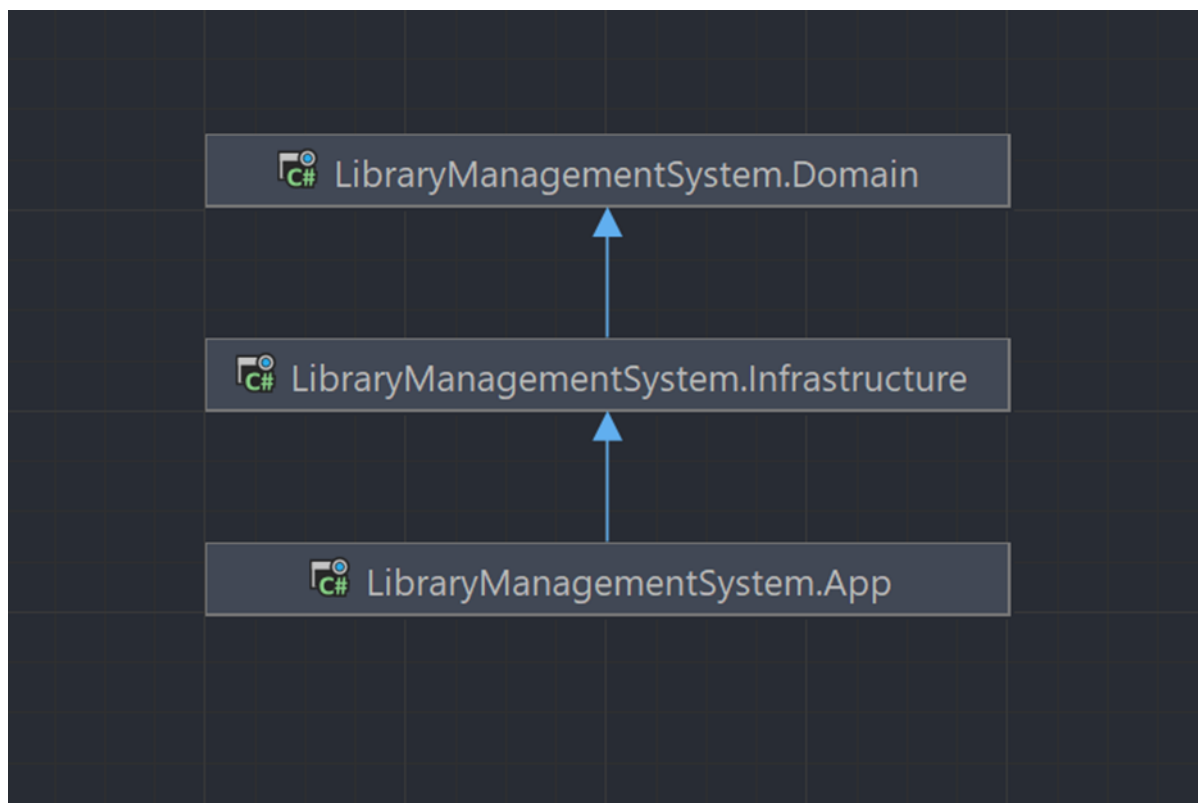
Ví dụ:

Thành phần	Diễn giải
Winform UI	<p>Giao diện được xây dựng bằng C# Winform</p> <p>UI chứa code quản lý layout cho giao diện và quản lý các sự kiện xảy ra trong ứng dụng</p> <p>Bao gồm cấu hình services cần dùng cho UI và cấu hình database provider</p>

Features	<p>Quản lý các feature chính của toàn bộ ứng dụng bao gồm</p> <ul style="list-style-type: none"> <li>- Author feature</li> <li>- Book feature</li> <li>- Book type feature</li> <li>- Cell card feature</li> <li>- Library card feature</li> <li>- Library configuration feature</li> <li>- Publisher feature</li> </ul> <p>Các feature thực hiện các xử lý liên quan tới nghiệp vụ (domain) của hệ thống, nhưng không có sự tương tác trực tiếp với tầng UI cũng như tầng hạ tầng (bao gồm cơ sở dữ liệu), toàn bộ code thực thi tại features là code C# thuần túy không có sự phụ thuộc lên các Framework như Winform, hay Entity Framework, Features có thể tách ra sử dụng với bất kỳ cơ sở dữ liệu nào và cũng có thể sử dụng cho ứng dụng Winform hoặc ứng dụng Web...</p> <p>Features được xây dựng dựa trên pattern CQRS ở mức đơn giản</p>
Feature commands	<p>Mỗi feature bao gồm nhiều commands, command chứa các phương thức thực hiện các thao tác tạo mới, cập nhật và xóa dữ liệu</p> <p>Ví dụ đối với Author feature, các commands bao gồm Create Author Command (tạo mới một tác giả), Update Author Command (cập nhật thông tin một tác giả đã tồn tại trong hệ thống), Delete Author Command (xóa một tác giả có trong hệ thống)</p>

	thông)
Feature queries	<p>Mỗi feature bao gồm nhiều queries, các query sử dụng với mục đích duy nhất là đọc dữ liệu, không bao gồm các thao tác thêm mới, chỉnh sửa, hay xóa dữ liệu</p> <p>Query bao gồm hai dạng query chính là truy vấn dữ liệu theo id của dữ liệu cần truy vấn, và truy vấn nhiều dữ liệu dựa vào các điều kiện tìm kiếm và hỗ trợ phân trang</p> <p>Ví dụ đối với Book feature, các queries bao gồm Get Book Query (truy vấn thông tin một đầu sách dựa vào Id của sách), List Books Query (truy vấn thông tin sách dựa vào các điều kiện tìm kiếm như tên sách, tên tác giả,.. và hỗ trợ tìm kiếm phân trang)</p>

## 2. Mô tả chi tiết từng thành phần trong hệ thống



Lớp đối tượng	Diễn giải
LibraryManagementSystem.App	<p>Bao gồm UI và features, cấu hình services, seed data, database provider (hệ thống sử dụng SQL Server)</p> <p>Kiến trúc cơ bản của hệ thống được dựa trên clean architecture thay vì mô hình 3 lớp truyền thống, toàn bộ cấu hình dependencies được cấu hình tại App, bao gồm các cấu hình services, cấu hình database provider, hệ thống từ đó có thể dễ dàng thay thế các services, hay database providers mà không ảnh hưởng tới các code nghiệp vụ vì toàn bộ code nghiệp vụ không phụ thuộc lên services, cụ thể chúng ta có thể thay từ SQL Server sang một hệ quản trị cơ sở dữ liệu quan hệ khác như MySQL hoặc PostgreSQL bằng cách thay đổi cấu hình mà</p>

	<p>không ảnh hưởng tới các phần code khác của hệ thống</p> <p>Pattern chính được sử dụng là CQRS có chức năng tách luồng đọc và ghi dữ liệu của ứng dụng ra làm hai luồng độc lập tách biệt, điều này hỗ trợ hệ thống dễ dàng mở rộng, có thể thay thế các triển khai của các command, query một cách dễ dàng, công việc maintain và code mới được tổ chức một cách tốt hơn</p>
LibraryManagementSystem. Infrastructure	<p>Cấu hình cơ sở dữ liệu</p> <p>Các cấu hình này là database agnostic, nghĩa là cấu hình không quan tâm tới việc sử dụng hệ quản trị cơ sở dữ liệu nào (các hệ quản trị cơ sở dữ liệu quan hệ), mọi hệ quản trị cơ sở dữ liệu được hỗ trợ đều sử dụng một cấu hình chung hỗ trợ việc mapping entities từ code sang bảng của cơ sở dữ liệu trở nên đơn giản và dễ dàng thay thế, việc dễ dàng thay thế database provider được thể hiện ở project App phía trên, việc chuyển đổi chỉ đơn giản là chuyển đổi việc đăng ký database services còn cấu hình không bị ảnh hưởng và không cần thay đổi</p>
LibraryManagementSystem. Domain	<p>Cấu hình domain entities, chứa các lớp thể hiện nghiệp vụ của ứng dụng và sẽ được map thành các bảng dữ liệu dưới database</p> <p>Chỉ chứa code C# và không phụ thuộc lên bất kỳ framework nào khác</p>