ĐẶC TẢ HỆ THỐNG QUẢN LÝ TÀI LIỆU

# 1. Giới thiệu

Hệ thống Quản lý Tài liệu (Document Management System - DMS) là một ứng dụng giúp tổ chức lưu trữ, quản lý, phân quyền truy cập và tra cứu tài liệu một cách hiệu quả. Hệ thống được phát triển trên nền tảng ASP.NET MVC, áp dụng kiến trúc Repository – Service – Controller nhằm đảm bảo tính tách biệt, dễ bảo trì và mở rộng.

# 2. Kiến trúc hệ thống

Hệ thống được chia thành các tầng chính:  
- Repository: giao tiếp trực tiếp với DbContext, thực hiện các thao tác CRUD.  
- Service: sử dụng repository, xử lý logic nghiệp vụ.  
- Controller: nhận yêu cầu từ người dùng, gọi Service và trả về View.  
- Helper: chứa các hàm tiện ích dùng chung.  
- ViewModel: lớp trung gian để truyền dữ liệu giữa Controller và View.

# 3. Các tầng chi tiết

## 3.1 Repository

Chịu trách nhiệm truy xuất dữ liệu từ cơ sở dữ liệu thông qua Entity Framework.

## 3.2 Service

Xử lý logic nghiệp vụ, gọi repository để thao tác dữ liệu và trả về kết quả cho Controller.

## 3.3 Controller

Gọi Service, xử lý request/response, ánh xạ dữ liệu vào ViewModel và hiển thị ra View.

## 3.4 Helper

Chứa các hàm tiện ích như phân trang, định dạng ngày giờ, validate dữ liệu.

## 3.5 ViewModel

Là lớp trung gian giúp tách dữ liệu hiển thị với dữ liệu trong database.

# 4. Use Case chính

- Quản lý phòng ban (Department): thêm, sửa, xóa mềm, xem chi tiết.  
- Quản lý người dùng (User): đăng ký, đăng nhập, phân quyền.  
- Quản lý tài liệu (Document): upload, download, phân loại, gắn phòng ban.  
- Phân quyền (Role & Permission): kiểm soát quyền truy cập tài liệu.  
- Tìm kiếm & phân trang: hỗ trợ tìm nhanh theo từ khóa, lọc theo trạng thái.

# 5. Mô hình dữ liệu

Các bảng chính trong hệ thống:  
- Departments: quản lý thông tin phòng ban.  
- Users: quản lý tài khoản người dùng.  
- Roles: định nghĩa vai trò.  
- Permissions: định nghĩa quyền.  
- Documents: lưu trữ tài liệu.  
- Bảng liên kết: UserRoles, RolePermissions, DocumentDepartments.

# 6. Hướng dẫn mở rộng và bảo trì

- Khi thêm module mới, chỉ cần tạo Repository, Service, Controller tương ứng.  
- Tách logic nghiệp vụ phức tạp vào Service để Controller gọn nhẹ.  
- Sử dụng ViewModel để đảm bảo dữ liệu hiển thị an toàn, tránh lộ dữ liệu nhạy cảm.  
- Có thể tích hợp API để cho phép hệ thống khác kết nối và khai thác dữ liệu.