|  |
| --- |
| **TRƯỜNG ĐẠI HỌC XÂY DỰNG HÀ NỘI**  **KHOA CÔNG NGHỆ THÔNG TIN**  **BỘ MÔN CÔNG NGHỆ PHẦN MỀM**  **🙡🕮🙣**  Káº¿t quáº£ hÃ¬nh áº£nh cho logo trÆ°á»ng Äáº¡i há»c xÃ¢y dá»±ng  **ĐỒ ÁN MÔN : THIẾT KẾ KIẾN TRÚC PHẦN MỀM**  **Đề Tài:** Quản lý nhân sự  **Nhóm Đồ Án :** **Nhóm 11**    **Giảng viên hướng dẫn : Ths Phạm Hữu Tùng**  **Thành viên nhóm : Nguyễn Hồng Phong - 0199566**    **Đinh Quang Khải - 0192866**      Hà Nội, 5 / 2025 |

Mục Lục

[**Lời nói đầu** 3](#_Toc199174074)

[**Chương 1.Bản đặc tả yêu cầu bài toán** 4](#_Toc199174075)

[**I . Tổng quan về đề tài** 4](#_Toc199174076)

[**II.Bài toán nghiệp vụ hệ thống quản lý sản phẩm** 5](#_Toc199174077)

[**III. Phân tích chức năng:** 5](#_Toc199174078)

[**Chương 2 : Thiết kế UML** 7](#_Toc199174079)

[**I . Use case diagram** 7](#_Toc199174080)

[**II. Class Diagram** 13](#_Toc199174081)

[III. Sequence diagram 13](#_Toc199174082)

[**Chương 3. Thiết kế hệ thống** 28](#_Toc199174083)

[**I.Mô hình hệ thống clean architecture** 28](#_Toc199174084)

[**II.Ưu và nhược điểm của hệ thống clean architecture** 30](#_Toc199174085)

[**III.Thiết kế giao diện người dùng** 31](#_Toc199174086)

[**Chương 4. Triển khai và Cài đặt** 37](#_Toc199174087)

[I. Môi trường triển khai 37](#_Toc199174088)

[II. Cài đặt hệ thống 37](#_Toc199174089)

[**Chương 5. Kết quả và đánh giá** 41](#_Toc199174090)

[I. Kết quả thử nghiệm hệ thống 41](#_Toc199174091)

[II. Đánh giá hiệu quả hệ thống 42](#_Toc199174092)

[**Chương 6 . Kết luận + hướng phát triển trang web** 42](#_Toc199174093)

[**I. Kết luận** 42](#_Toc199174094)

[**II. Hướng phát triển sản phẩm** 43](#_Toc199174095)

# **Lời nói đầu**

Nhân loại đang có những bước tiến lớn, và công nghệ thông tin chính là một trong những nền móng quan trọng để đưa con người sang một kỉ nguyên mới – một kỉ nguyên công nghệ.

Giải pháp được đưa chính là xây dựng chương trình quản lý nhân sự bao gồm các chức năng cơ bản (thêm, sửa, xóa nhân viên, phòng ban, tìm kiếm thông tin nhân viên, phòng ban theo tên, id, phân trang, xuất file Excel, quản lý hợp đồng, quản lý bảng lương), và đây cũng chính là nội dung chính của đồ án của chúng em. Mục đích của chúng em là đưa ra 1 chương trình quản lý nhân sự cơ bản, dễ dàng sử dụng, những bước chuẩn bị và xây dựng được thể hiện rõ ràng hơn trong tài liệu này.

Đồ án được thực hiện bởi những sinh viên chưa có nhiều kinh nghiệm. Vì vậy trong quá trình thực hiện, có thể sẽ có những sai sót. Chúng em kính mong thầy có thể cho chúng em những góp ý để chúng tôi có thể hoàn thiện bài một cách tốt nhất.

# **Chương 1.Bản đặc tả yêu cầu bài toán**

## **I . Tổng quan về đề tài**

1. Lý do hình thành đề tài

Ngày nay khi thế giới bước sang thời đại công nghệ 4.0 thì công nghệ thông tin ngày càng phát triển mạnh mẽ và trở nên phổ biến trên toàn thế giới. Những thành tựu mà công nghệ thông tin mang lại làm thay đổi mọi mặt của đời sống xã hội, xâm nhập vào nhiều lĩnh vực khoa học, kĩ thuật cũng như trong cuộc sống thường ngày. Nó trở thành công cụ đắc lực không thể thiếu trong nhiều lĩnh vực hoạt động như: quản lý, nghiên cứu, điều khiển máy móc,….. và đặc biệt trong công tác quản lý nói chung và trong việc quản lý nhân sự nói riêng. Hoạt động quản lý nhân sự là một lĩnh vức quản lý vô cùng rộng và có nhiều nghiệp vụ phức tạp. Nhận thức được thực trạng đó nên nhóm chúng em quyết định thực hiện đề tài “Chuyên đề tổng hợp’’đồ án “Thiết kế kiến trúc phần mèm” với nội dung là hoạt động quản lý nhân sự Do hoạt động quản lý nhân sự là một lĩnh vực rộng cộng với thời gian có hạn nên nếu có thiếu sót gì mong thầy có thể góp ý cho nhóm chúng em để chúng em có thể hoàn thiện để tài được tốt nhất và rút kinh nghiệm cho các dự án sau .

1. Các công cụ sử dụng

* **Frontend**:
  + React
  + Angular – cli
  + Angular – grid
* **Backend**:
  + ASP .NET
  + Node js
  + Java (Spring Boot)
  + Postman
* **Database**:
  + SQLserver

3. Yêu cầu bổ sung

* Hỗ trợ API RESTful cho các ứng dụng khác kết nối .

## **II.Bài toán nghiệp vụ hệ thống quản lý sản phẩm**

1. Mô tả bài toán

**Quản lý nhân sự** là một nhu cầu cấp thiết và không thể thiếu trong thời đại công nghệ 4.0. Việc quản lý nhân sự sẽ giúp người dùng nắm bắt được các thông tin cần thiết về nhân viên, phòng ban, hợp đồng, bảng lương, từ đó hỗ trợ cho việc vận hành hệ thống nhân sự hiệu quả, thuận tiện trong việc theo dõi và ra quyết định.

* Quản trị viên sẽ có thể thêm, sửa, xóa thông tin nhân viên, phòng ban, hợp đồng lao động, bảng lương.
* Quản trị viên có thể quản lý danh sách nhân viên và phòng ban trên hệ thống.
* Người dùng (quản lý, nhân viên có phân quyền cụ thể) có thể xem thông tin nhân viên, hợp đồng, phòng ban và bảng lương tùy theo phạm vi quyền hạn.

**2. Nhu cầu quản lý chức năng:**

* Quản lý danh sách nhân viên và phòng ban
* Quản lý thông tin nhân viên: xem, sắp xếp, thêm, sửa, xóa, tìm kiếm
* Quản lý thông tin phòng ban
* Quản lý hợp đồng lao động
* Quản lý bảng lương
* Xuất file Excel thông tin nhân viên, phòng ban

**3. Phạm vi**   
Quản lý toàn bộ thông tin liên quan đến nhân viên, phòng ban, bảng lương và hợp đồng trong hệ thống.

**4. Đối tượng:**

* **Quản trị viên:** Có toàn quyền với hệ thống, bao gồm quản lý nhân viên, phòng ban, hợp đồng và bảng lương.

## **III. Phân tích chức năng:**

**1. Phân tích yêu cầu hệ thống:**

**1.1. Yêu cầu chức năng:**

**a. Yêu cầu quản lý danh sách nhân sự**

* Xem danh sách nhân viên:
  + Danh sách nhân viên theo tên
  + Danh sách nhân viên theo phòng ban

**b. Yêu cầu quản lý thông tin nhân sự**

* Thông tin nhân viên bao gồm:  
  id, employeeName, departmentId
* Các chức năng chính:
  + Xem thông tin nhân viên
  + Thêm nhân viên (thông tin đầy đủ)
  + Xóa nhân viên
  + Sửa thông tin nhân viên
  + Tìm kiếm nhân viên
  + Xuất file Excel thông tin nhân viên

**c. Quản lý phòng ban**

* Thêm phòng ban
* Sửa thông tin phòng ban
* Xóa phòng ban
* Xem danh sách phòng ban
* Xuất file Excel thông tin phòng ban

**d. Quản lý hợp đồng lao động**

* Thêm hợp đồng
* Sửa hợp đồng
* Xóa hợp đồng
* Tìm kiếm hợp đồng
* Xem danh sách hợp đồng theo nhân viên

**e. Quản lý bảng lương**

* Thêm bảng lương
* Sửa bảng lương
* Xóa bảng lương
* Xem chi tiết bảng lương theo nhân viên

**1.2. Yêu cầu chất lượng:**

* **Giao diện:** Thân thiện, hiện đại, phù hợp với một hệ thống quản lý nhân sự chuyên nghiệp.
* **Tốc độ xử lý:** Nhanh, tối ưu truy xuất dữ liệu, không gây chậm trễ cho người dùng.
* **Tính ổn định:** Việc thay đổi một chức năng không làm ảnh hưởng đến các chức năng khác.
* **Sao lưu & phục hồi:** Hệ thống có khả năng sao lưu và phục hồi cơ sở dữ liệu khi có sự cố.
* **Tính mở rộng:** Dễ dàng nâng cấp, thay đổi chức năng hoặc giao diện khi cần thiết.

**1.3. Yêu cầu chức năng cụ thể:**

* Quản lý thông tin nhân viên (thêm, sửa, xóa, tìm kiếm)
* Quản lý danh sách phòng ban
* Quản lý hợp đồng lao động
* Quản lý bảng lương nhân viên
* Đăng nhập hệ thống và phân quyền sử dụng

# **Chương 2 : Thiết kế UML**

## **I . Use case diagram**

1. Use case tổng quan hệ thống

A blue rectangular object with black lines and dots

AI-generated content may be incorrect.

1. Quản lý phòng ban

A blue rectangle with black text and circles

AI-generated content may be incorrect.

1. Quản lý nhân viên

A screenshot of a computer

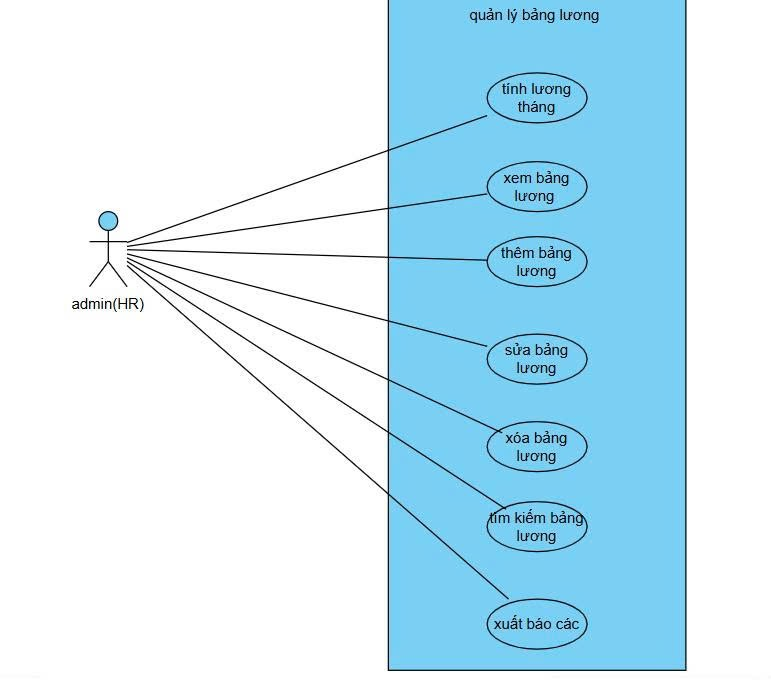
AI-generated content may be incorrect.

1. Quản lý hợp đồng

A blue paper with black text

AI-generated content may be incorrect.

1. Quản lý bảng lương



## **II. Class Diagram**

A computer screen shot of a computer

AI-generated content may be incorrect.

## III. Sequence diagram

1.Quản lý nhân viên

a.Tìm kiếm thông tin nhân viên

A diagram of a diagram

AI-generated content may be incorrect.

b.Sửa thông tin nhân viên

A diagram of a diagram

AI-generated content may be incorrect.

c.Xóa thông tin nhân viên

A diagram of a project

AI-generated content may be incorrect.

2.Quản lý phòng ban

a.Tìm kiếm phòng ban

A diagram of a diagram

AI-generated content may be incorrect.

b.Thêm phòng ban

A diagram of a diagram

AI-generated content may be incorrect.

c.Sửa thông tin phòng ban

A diagram of a diagram

AI-generated content may be incorrect.

d.Xóa phòng ban

A diagram of a diagram

AI-generated content may be incorrect.

3.Quản lý hợp đồng

a.Tìm kiếm hợp đồng

A diagram of a diagram

AI-generated content may be incorrect.

b.Thêm hợp đồng

A diagram of a diagram

AI-generated content may be incorrect.

c.Sửa hợp đồng

A diagram of a diagram

AI-generated content may be incorrect.

d.Xóa hợp đồng

A diagram of a diagram

AI-generated content may be incorrect.

4.Quản lý bảng lương

a.Tìm kiếm bảng lương

A diagram of a diagram

AI-generated content may be incorrect.

b.Thêm bảng lương

A diagram of a diagram

AI-generated content may be incorrect.

c.Sửa bảng lương

A diagram of a diagram

AI-generated content may be incorrect.

c.Xóa bảng lương

A diagram of a diagram

AI-generated content may be incorrect.

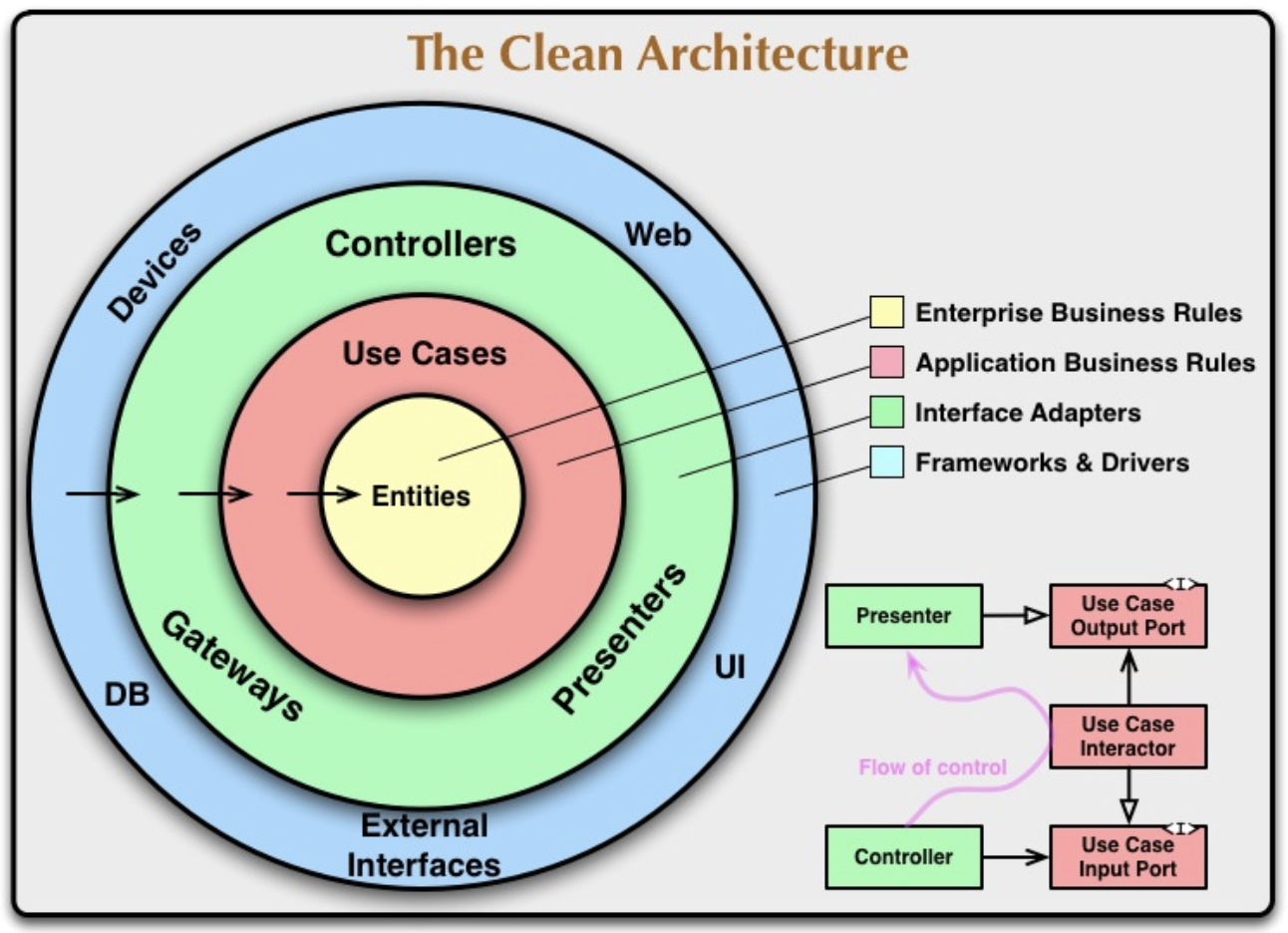
# **Chương 3. Thiết kế hệ thống**

## **I.Mô hình hệ thống clean architecture**

Hệ thống được xây dựng theo mô hình kiến trúc clean architecture

**Clean Architecture** là một kiến trúc ứng dụng rất nổi tiếng dựa trên nguyên lý loại bỏ sự lệ thuộc giữa các đối tượng cũng như các layer trong ứng dụng. Nguyên lý này kế thừa và phát triển dựa trên **Dependency Inversion** - nguyên lý nổi tiếng trong **SOLID**.

Trong kiến trúc **Clean Architecture** bao gồm 4 layer được đại diện thông qua các vòng tròn đồng tâm. Các vòng tròn ở trong sẽ không hề biết gì về các vòng tròn bên ngoài. Nguyên tắc "hướng tâm" này được minh hoạ như sau:



Từ trong ra ngoài **Clean Architecture** sẽ bao gồm: **Entities**, **Use Cases**, **Interface Adapters** và **Frameworks & Drivers**.

**Entities** là layer trong cùng, cũng là layer quan trọng nhất. Entity chính là các thực thể hay từng đối tượng cụ thể và các rule business logic của nó. Trong [**OOP**](https://200lab.io/blog/oop-la-gi), đây chính là **Object** cùng với các **method** và **properties** tuân thủ nguyên tắc **Encapsulation** - chỉ bên trong Object mới có thể thay đổi trạng thái (**State**) của chính nó.

**Use Cases** là layer chứa các business logic ở cấp độ cụ thể từng Use Case (hay application).Các business logic của Use Case đương nhiên cũng sẽ không quan tâm và lệ thuộc vào việc dữ liệu đến từ đâu, dùng các thư viện nào làm apdapter, dữ liệu thể hiện thế nào,... Vì đấy là nhiệm vụ của layer **Interface Adapters**.

**Interface Adapters** chính là layer phụ trách việc chuyển đổi các format dữ liệu để phù hợp với từng Use Case và Entities. Các format dữ liệu này có thể dùng cho cả bên trong hoặc ngoài ứng dụng.

**Frameworkd & Drivers** là tầng ngoài cùng, tổ hợp các công cụ cụ thể phục vụ cho từng nhu cầu của end user như: thiết bị (devices), web, application, databases,... Trong kiến trúc Clean Architecture thì ở tầng này là "nhẹ" nhất vì chúng ta không cần phải viết quá nhiều code.

## **II.Ưu và nhược điểm của hệ thống clean architecture**

**Ưu điểm của Clean Architecture**

* **Chia để trị rất hiệu quả trong ứng dụng lớn**: Trong Clean Architecture thì code tầng nào thì ở đúng tầng nấy. Hạn chế được việc "code ở đâu cũng là code, chạy được là được". Nếu làm tốt được các bài toán nhỏ thì không có bài toán lớn nào không giải quyết được.
* **Rất dễ maintain và mở rộng**: Việc tìm kiếm bug và lỗi logic sẽ trở nên dễ dàng và nhanh hơn, file code sẽ không nhiều vì chỉ làm đúng việc của nó. Vì các tầng độc lập với nhau thông qua các Interfaces nên việc mở rộng hoặc thay đổi các tầng sẽ không ảnh hưởng tới nhau. Điều này hạn chế các breaking change cũng như phải viết lại code (refactoring).
* **Rất dễ làm Unit Test**: Các logic business của các tầng trong Clean Architecture chính là các Unit Test cần được kiểm thử rất cẩn thận. Vì sự độc lập thông qua Interfaces nên các mock test rất dễ triển khai. Việc này được thực hiện thông qua implement lại để coverage được tất cả các trường hợp.

**Nhược điểm của Clean Architecture**

* **Cồng kềnh và phức tạp**: Điều dễ thấy nhất là Clean Architecture không hề dễ sử dụng, phải viết nhiều lớp (class/object) hơn. Trong trường hợp ứng dụng của bạn quá đơn giản, ít tính năng, vòng đời ngắn thì chọn lựa kiến trúc này có thể mang lại những rắc rối không cần thiết.
* **Tính trừu tượng cao**: Vấn đề này gọi là **indirect**. Trừu tượng càng cao thì tiện cho các developers nhưng sẽ gây ảnh hưởng không nhỏ tới tốc độ thực thi (performance). Ngoài ra cũng không thể code nhanh, vội vã "mì ăn liền" được mà phải tạo đủ các Interfaces.
* **Khó tuyển người**: Sử dụng Clean Architecture sẽ cần tuyển dụng developer thấu hiểu về kiến trúc này. Nguyên tắc Dependency Inversion rất dễ bị xâm phạm vì sự hạn chế kiến thức, sự bất cẩn hoặc vì thời gian cần triển khai tính năng quá ít.

## **III.Thiết kế giao diện người dùng**

Thiết kế giao diện người dùng: giao diện người dùng được thiết kế để dễ dàng sử dụng, dễ dàng thao tác:

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

# **Chương 4. Triển khai và Cài đặt**

## I. Môi trường triển khai

| Thành phần | Phiên bản khuyến nghị | Ghi chú |
| --- | --- | --- |
| Hệ điều hành | Windows 11 |  |
| .NET SDK | 8.0 | Dùng cho API C# với Entity Framework |
| Node.js | v20.11.1 | Kèm npm 10.x+ |
| Angular CLI | 16.x | Cài toàn cục: npm install -g @angular/cli |
| AG-Grid (angular-grid-form) | 32.x | Xuất Excel, lọc, tìm kiếm bảng dữ liệu |
| SQL Server | 2019 hoặc 2022 | Express hoặc Developer Edition |
| Entity Framework Core | 8.0 | Code-first migrations và seeding |
| Swagger / Swashbuckle | V3.36.0 | Tài liệu và thử nghiệm API |
| Postman | Phiên bản mới nhất | Test REST endpoints |

## II. Cài đặt hệ thống

4.2.1. Yêu cầu hệ thống

* Phần mềm
  + .NET SDK 8.0
  + Node.js 16.x+ và npm 10.x+
  + SQL Server 2019/2022
  + Visual Studio 2022 hoặc VS Code
  + Angular CLI
  + Postman
  + (Tuỳ chọn) Excel 2016+ để kiểm thử xuất file
* Quyền truy cập
  + Quyền tạo và quản trị database trên SQL Server
  + Quyền cài đặt gói NuGet và npm packages

4.2.2. Các bước cài đặt chương trình

A. Thiết lập database

1. Cài SQL Server
   * Cài bản Express hoặc Developer
   * Tạo user và cấp quyền admin
2. Tạo database
   * Tạo database mới (ví dụ HRManagementDB)
   * Cấu hình chuỗi kết nối trong appsettings.json:

json

Sao chépChỉnh sửa

"ConnectionStrings": {

"DefaultConnection": "Server=.;Database=HRManagementDB;Trusted\_Connection=True;"

}

1. Chạy migration & seeding

cd path/to/API

* Tạo DB

dotnet ef migrations add InitialCreate

>> dotnet ef database update

* Chạy ứng dụng

dotnet run --urls=http://localhost:5002

B. Cài đặt backend (.NET Web API)

1. Mở solution
   * Dùng Visual Studio hoặc VS Code mở thư mục API/.
2. Cài dependencies

cd API

dotnet restore

1. Cấu hình Swagger
   * Đảm bảo trong Program.cs đã có:

builder.Services.AddEndpointsApiExplorer();

builder.Services.AddSwaggerGen();

...

app.UseSwagger();

app.UseSwaggerUI();

1. Chạy APIdotnet run --urls "https://localhost:5002;http://localhost:5002"
   * Truy cập https://localhost:5002/swagger để test các endpoint:
     + Nhân viên: add, edit, delete, search
     + Phòng ban
     + Hợp đồng
     + Bảng lương

C. Cài đặt frontend (Angular)

1. Cài Angular CLI : npm install -g @angular/cli
2. Cài dependencies

cd path/to/Client

npm install

1. Cấu hình environment
   * Mở src/environments/environment.ts, chỉnh apiUrl trỏ tới backend

export const environment = {

production: false,

Base apiUrl: 'http://localhost:5002/api

};

1. Chạy ứng dụng

ng serve

* + Ứng dụng sẽ mở http://localhost:4300
  + Tại giao diện, bạn có thể:
    - Thêm / Sửa / Xóa / Tìm kiếm nhân viên, phòng ban, hợp đồng, bảng lương
    - Xuất file Excel qua tính năng AG-Grid export

D. Kiểm thử với Postman

1. Import Collection
   * Import file Postman collection kèm code
2. Chạy thử từng endpoint:
   * GET /employees
   * POST /employees/add-employee
   * PUT /employees/{id}
   * DELETE /employees/delete-employee/{id}
   * Tương tự cho departments, contracts, salaries
3. Kiểm tra xuất Excel
   * Gọi endpoint lấy dữ liệu
   * So sánh file xuất với dữ liệu trên UI

Kết quả: Sau khi hoàn tất, bạn sẽ có một hệ thống quản lý nhân sự đầy đủ chức năng (CRUD, tìm kiếm, export Excel) chạy ổn định trên .NET + Angular với SQL Server.

# **Chương 5. Kết quả và đánh giá**

## I. Kết quả thử nghiệm hệ thống

1. Kiểm thử chức năng CRUD
   * Nhân viên:
     + Thêm mới, sửa, xóa nhân viên hoạt động chính xác (Test cases: TC\_EMP\_ADD, TC\_EMP\_UPDATE, TC\_EMP\_DELETE) thành công 100%.
     + Tìm kiếm nhân viên theo tên và phòng ban trả về kết quả nhanh, đúng dữ liệu mẫu.
   * Phòng ban:
     + Các thao tác thêm, sửa, xóa và tìm kiếm phòng ban (TC\_DEP\_\*) thực hiện ổn định, không phát sinh lỗi.
   * Hợp đồng:
     + CRUD hợp đồng (TC\_CON\_\*) hoạt động đúng business logic, ngày bắt đầu trước ngày kết thúc được kiểm tra chặt.
   * Bảng lương:
     + Tạo mới, cập nhật, xóa lương, tính tổng lương tự động chính xác.
     + Tìm kiếm theo ID và tên nhân viên trả kết quả đúng.
2. Kiểm thử hiệu năng
   * Thời gian phản hồi cho các API CRUD < 200ms với < 100 bản ghi.
3. Kiểm thử bảo mật
   * Đảm bảo chỉ cho phép truy cập HTTP methods đúng (GET, POST, PUT, DELETE).
   * API trả về mã lỗi thích hợp (400, 404, 500) khi dữ liệu không hợp lệ.

## II. Đánh giá hiệu quả hệ thống

1. Tính ổn định và tin cậy
   * Hệ thống chạy mượt, ít lỗi.
   * Sử dụng middleware tập trung xử lý Exception giúp ghi log và phản hồi nhất quán.
2. Tính mở rộng và bảo trì
   * Kiến trúc tách biệt giữa backend (C#/.NET) và frontend (Angular) thuận tiện cho phát triển, nâng cấp.
   * Sử dụng EF Core Migrations dễ dàng quản lý schema, seed dữ liệu mẫu.
3. Tính năng đáp ứng yêu cầu
   * Đầy đủ CRUD, tìm kiếm, export Excel đáp ứng hoàn toàn yêu cầu quản lý nhân sự.
   * Swagger & Postman hỗ trợ test & tài liệu hóa API nhanh chóng.
4. Hiệu quả về mặt thời gian và chi phí
   * Sử dụng công cụ miễn phí/opensource (Angular, EF Core, SQL Server Express) giảm chi phí bản quyền.
5. Hạn chế và hướng cải tiến
   * Chưa có tính năng phân quyền chi tiết theo vai trò.
   * UX cần tinh chỉnh responsive cho mobile.
   * Có thể tích hợp báo cáo BI, biểu đồ để trực quan hóa dữ liệu.

# **Chương 6 . Kết luận + hướng phát triển trang web**

## **I. Kết luận**

Qua thời gian thực hiện, nhóm em đã hoàn thành **đồ án môn Thiết kế Kiến trúc Phần mềm** theo đúng yêu cầu và thời gian quy định. Chúng em đã thực hiện được các công việc sau:

* Đặt vấn đề, tìm hiểu, chọn hướng đi phù hợp cho bài toán quản lý nhân sự.
* Phân tích và thiết kế hệ thống.
* Xây dựng hệ thống quản lý nhân sự gồm các chức năng như: quản lý thông tin nhân viên, phòng ban, hợp đồng, bảng lương và đăng nhập, xuất file Excel thông tin nhân viên, phòng ban bám sát với các yêu cầu cơ bản về chức năng của một hệ thống quản lý nhân sự.

Mặc dù nhóm em đã cố gắng hoàn thiện và phát triển hệ thống theo cách tốt nhất có thể, nhưng do kiến thức thực tiễn về việc vận hành các hệ thống nhân sự trong môi trường doanh nghiệp còn hạn chế, nên hệ thống vẫn khó tránh khỏi những thiếu sót. Vì vậy, nhóm em rất mong thầy/cô có thể đưa ra những góp ý để nhóm em cải thiện và hoàn thiện sản phẩm tốt hơn.

## **II. Hướng phát triển sản phẩm**

Do giới hạn về mặt thời gian nên sản phẩm hiện tại của nhóm em vẫn chưa đạt mức hoàn chỉnh như kỳ vọng. Tuy nhiên, trong tương lai gần, nhóm em sẽ tiếp tục phát triển hệ thống quản lý nhân sự với các chức năng bổ sung như:

* Phân quyền chi tiết cho các loại tài khoản (quản trị viên, trưởng phòng, nhân viên).
* Xuất bảng lương ra file Excel.
* Thêm chức năng quản lý đánh giá năng lực nhân viên.
* Thống kê số lượng nhân viên theo phòng ban, hợp đồng lao động còn hiệu lực.
* Tự động gửi thông báo hết hạn hợp đồng hoặc cần gia hạn.

**3. Hạn chế**

Mặc dù dự án đã đạt được những kết quả đáng kể, nhưng vẫn còn tồn tại một số hạn chế:

* **Tính năng còn giới hạn:** Do thời gian gấp rút, hệ thống quản lý nhân sự mới chỉ có một số chức năng cơ bản như thêm, sửa, xóa, tìm kiếm nhân viên, phòng ban, hợp đồng, bảng lương và đăng nhập quản trị viên; chưa đáp ứng được hết các nhu cầu chuyên sâu của một hệ thống quản lý nhân sự thực tế.
* **Tốc độ xử lý:** Khi có nhiều người dùng truy cập cùng lúc hoặc lượng dữ liệu lớn, tốc độ xử lý vẫn chưa tối ưu.

Những kết quả đạt được cùng với định hướng phát triển sẽ là cơ sở để nhóm em tiếp tục hoàn thiện dự án trong tương lai, hướng tới xây dựng một nền tảng quản lý nhân sự chuyên nghiệp, linh hoạt và hiệu quả hơn.