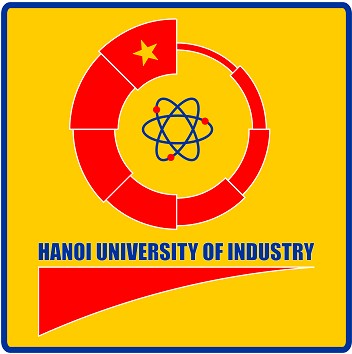


**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

------------------------------------------



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI**

**Xây dựng ứng dụng Web quản lý nhân sự**

**dùng ASP.NET Core, Angular, MySQL**

|  |  |
| --- | --- |
| **GVHD:** | **ThS. Nguyễn Đức Lưu** |
| **Sinh Viên:** | **Nguyễn Quốc Đạt** |
| **Lớp:** | **2018DHKTPM01** |
| **Khóa:** | **13** |
| **Mã sinh viên:** | **2018600273** |
|  |  |

Hà Nội - Năm 2022

|  |  |
| --- | --- |
|  |  |
| Nguyễn Quốc Đạt | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------** |
| SinhVienIT |
| ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC  NGÀNH KỸ THUẬT PHẦN MỀM |
|  |
| **Xây dựng ứng dụng Web quản lý nhân sự** |
| **dùng ASP.NET Core, Angular, MySQL** |
|  |
| **CBHD: *ThS.* Nguyễn Đức Lưu** |
| KỸ THUẬT PHẦN MỀM | **Sinh viên: Nguyễn Quốc Đạt** |
| **Mã số sinh viên: 2018600273** |
|  |
|  |
|  |
| Hà Nội – Năm 2022 |
|  |

# **LỜI CẢM ƠN**

Để hoàn thành báo cáo này, em xin gửi lời cảm ơn chân thành đến:

Ban giám hiệu trường Đại học Công nghiệp Hà Nội vì đã tạo điều kiện về cơ sở vật chất với hệ thống thư viện hiện đại, đa dạng các loại sách, tài liệu thuận lợi cho việc tìm kiếm, nghiên cứu thông tin.

Xin cảm ơn giảng viên hướng dẫn - Thầy Nguyễn Đức Lưu đã tận tình hướng dẫn cũng như cung cấp tài liệu, thông tin khoa học cần thiết để em có thể hoàn thành báo cáo này.

Lời cuối cùng, em xin kính chúc thầy nhiều sức khỏe, thành công và hạnh phúc.

*Em xin chân thành cảm ơn*

**MỤC LỤC**

[**LỜI CẢM ƠN** 1](#_Toc104324166)

[**MỤC LỤC** 2](#_Toc104324167)

[**DANH MỤC BẢNG BIỂU** 5](#_Toc104324168)

[**DANH MỤC HÌNH ẢNH** 6](#_Toc104324169)

[**MỞ ĐẦU** 8](#_Toc104324170)

[**i.** **Tên đề tài** 8](#_Toc104324171)

[**ii.** **Lý do chọn đề tài** 8](#_Toc104324172)

[**iii.** **Mục đích của đề tài** 9](#_Toc104324173)

[**iv.** **Bố cục đề tài** 9](#_Toc104324174)

[**CHƯƠNG 1: GIỚI THIỆU CÔNG NGHỆ SỬ DỤNG** 11](#_Toc104324175)

[**1.1. Angular** 11](#_Toc104324176)

[**1.2. ASP.NET Core** 12](#_Toc104324177)

[**1.3. MySQL** 14](#_Toc104324178)

[**CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG** 18](#_Toc104324179)

[**2.1. Yêu cầu** 18](#_Toc104324180)

[**2.2. Biểu đồ use case** 18](#_Toc104324181)

[**2.2.1. Biểu đồ use case tổng quan** 18](#_Toc104324182)

[**2.2.2 Biểu đồ phân rã Use Case** 20](#_Toc104324183)

[**2.3. Mô tả chi tiết Use Case** 24](#_Toc104324184)

[**2.3.1. Use Case Đăng nhập** 24](#_Toc104324185)

[**2.3.2. Use case Quản lý nhân viên** 25](#_Toc104324186)

[**2.3.3. Mô tả Quản lý hợp đồng** 28](#_Toc104324187)

[**2.3.4. Use Case Quản lý cơ cấu tổ chức** 30](#_Toc104324188)

[**2.3.5. Use case Quản lý Vị trí công việc** 33](#_Toc104324189)

[**2.3.6. Use case Quản lý Vai trò người dùng** 35](#_Toc104324190)

[**2.3.7. Use case Quản lý người dùng** 38](#_Toc104324191)

[**2.4. Các biểu đồ các lớp tham gia vào một Use Case (VOPC)** 40](#_Toc104324192)

[**2.4.1. Biểu đồ VOPC Use Case đăng nhập** 40](#_Toc104324193)

[**2.4.2. Biểu đồ VOPC Use Case quản lý nhân viên** 40](#_Toc104324194)

[**2.4.3. Biểu đồ VOPC Use Case quản lý hợp đồng** 41](#_Toc104324195)

[**2.4.4. Biểu đồ VOPC Use Case quản lý cơ cấu tổ chức** 42](#_Toc104324196)

[**2.4.5. Biểu đồ VOPC Use Case quản lý vị trí công việc** 42](#_Toc104324197)

[**2.4.6. Biểu đồ VOPC Use Case quản lý vai trò người dùng** 42](#_Toc104324198)

[**2.4.7. Biểu đồ VOPC Use Case quản lý người dùng** 44](#_Toc104324199)

[**2.5. Các biều đồ trình tự** 45](#_Toc104324200)

[**2.6. Thiết kế cơ sở dữ liệu** 52](#_Toc104324201)

[**2.6.1. Bảng users** 52](#_Toc104324202)

[**2.6.2. Bảng contracts** 53](#_Toc104324203)

[**2.6.3. Bảng organizationunits** 54](#_Toc104324204)

[**2.6.4. Bảng jobpositions** 55](#_Toc104324205)

[**2.6.5. Bảng roles** 55](#_Toc104324206)

[**2.6.6. Bảng user\_role** 55](#_Toc104324207)

[**2.6.7. Mô hình thiết kế vật lý** 55](#_Toc104324208)

[**2.7. Thiết kế giao diện** 56](#_Toc104324209)

[**2.7.1. Giao diện đăng nhập** 56](#_Toc104324210)

[**2.7.2. Giao diện Xem Tổng quan** 57](#_Toc104324211)

[**2.7.3. Giao diện Quản lý nhân viên** 58](#_Toc104324212)

[**2.7.4. Giao diện Quản lý hợp đồng** 58](#_Toc104324213)

[**2.7.5. Giao diện Quản lý vai trò người dùng** 59](#_Toc104324214)

[**2.7.6. Giao diện Quản lý Người dùng** 60](#_Toc104324215)

[**2.7.7. Giao diện Quản lý cơ cấu tổ chức** 60](#_Toc104324216)

[**2.7.8. Giao diện Quản lý vị trí công việc** 61](#_Toc104324217)

[**CHƯƠNG 3: THỬ NGHIỆM** 62](#_Toc104324218)

[**3.1. Cài đặt hệ thống** 62](#_Toc104324219)

[**3.1.1. Cài đặt giao diện** 62](#_Toc104324220)

[**3.1.2. Cài đặt dịch vụ** 65](#_Toc104324221)

[**3.2 Thử nghiệm** 66](#_Toc104324222)

[**3.2.1. Chức năng đăng nhập** 66](#_Toc104324223)

[**3.2.2. Phân hệ tổng quan** 67](#_Toc104324224)

[**3.2.3. Phân hệ nhân viên** 67](#_Toc104324225)

[**3.2.4. Phân hệ hợp đồng** 68](#_Toc104324226)

[**3.2.5. Phân hệ thiết lập** 70](#_Toc104324227)

[**KẾT LUẬN** 74](#_Toc104324228)

[**TÀI LIỆU THAM KHẢO** 75](#_Toc104324229)

# **DANH MỤC BẢNG BIỂU**

[Bảng 2. 1. Bảng người dùng 55](#_Toc104236544)

[Bảng 2. 2. Bảng hợp đồng 56](#_Toc104236545)

[Bảng 2. 3. Bảng cơ cấu tổ chức 57](#_Toc104236546)

[Bảng 2. 4. Bảng vị trí công việc 58](#_Toc104236547)

[Bảng 2. 5. Bảng vai trò 58](#_Toc104236548)

[Bảng 2. 6. Bảng vai trò người dùng 58](#_Toc104236549)

# **DANH MỤC HÌNH ẢNH**

[Hình 2- 1. Biểu đồ Use Case Tổng quát 19](#_Toc104324230)

[Hình 2- 2. Use Case Đăng nhập 21](#_Toc104324231)

[Hình 2- 3. Use Case Quản lý nhân viên 21](#_Toc104324232)

[Hình 2- 4. Use Case Quản lý hợp đồng 22](#_Toc104324233)

[Hình 2- 5. Quản lý cơ cấu tổ chức 22](#_Toc104324234)

[Hình 2- 6. Quản lý vị trí công việc 23](#_Toc104324235)

[Hình 2- 7. Use Case Quản lý vai trò người dùng 23](#_Toc104324236)

[Hình 2- 8. Use Case Quản lý người dùng 24](#_Toc104324237)

[Hình 2- 9. Biểu đồ VOPC Use Case đăng nhập 40](#_Toc104324238)

[Hình 2- 10. Biểu đồ VOPC Use Case quản lý nhân viên 41](#_Toc104324239)

[Hình 2- 11. Biểu đồ VOPC Use Case quản lý hợp đồng 41](#_Toc104324240)

[Hình 2- 12. Biểu đò VOPC Use Case quản lý cơ cấu tổ chức 42](#_Toc104324241)

[Hình 2- 13. Biểu đồ VOPC Use Case quản lý vị trí công việc 42](#_Toc104324242)

[Hình 2- 14. Biểu đồ VOPC Use Case quản lý vai trò người dùng 43](#_Toc104324243)

[Hình 2- 15. Biểu đồ VOPC Use Case quản lý người dùng 44](#_Toc104324244)

[Hình 2- 16. Biểu đồ trình tự Use Case đăng nhập 45](#_Toc104324245)

[Hình 2- 17. Biểu đồ trình tự Use Case quản lý nhân viên 46](#_Toc104324246)

[Hình 2- 18. Biểu đồ trình tự Use Case quản lý hợp đồng 47](#_Toc104324247)

[Hình 2- 19. Biều đồ trình tự Use Case quản lý cơ cấu tổ chức 48](#_Toc104324248)

[Hình 2- 20. Biều đồ trình tự Use Case quản lý vị trí công việc 49](#_Toc104324249)

[Hình 2- 21. : Biểu đồ trình tự Use Case quản lý vai trò người dùng 50](#_Toc104324250)

[Hình 2- 22. Biểu đồ trình tự Use Case quản lý người dùng 51](#_Toc104324251)

[Hình 2- 23. Giao diện Đăng nhập 57](#_Toc104324252)

[Hình 2- 24. Giao diện Tổng quan 57](#_Toc104324253)

[Hình 2- 25. Giao diện Danh sách nhân viên 58](#_Toc104324254)

[Hình 2- 26. Giao diện Danh sách hợp đồng 59](#_Toc104324255)

[Hình 2- 27. Giao diện Danh sách vai trò người dùng 59](#_Toc104324256)

[Hình 2- 28. Giao diện Danh sách người dùng 60](#_Toc104324257)

[Hình 2- 29. Giao diện Danh sách cơ cấu tổ chức 60](#_Toc104324258)

[Hình 2- 30. Giao diện Danh sách vị trí công việc 61](#_Toc104324259)

[Hình 3- 1. Chọn next để tiếp tục cài đặt 62](#_Toc104325177)

[Hình 3- 2.Chọn next để tiếp tục cài đặt 63](#_Toc104325178)

[Hình 3- 3.Chọn next để tiếp tục cài đặt 63](#_Toc104325179)

[Hình 3- 4.Cài đặt hoàn tất, chọn Finish để kết thúc 64](#_Toc104325180)

[Hình 3- 5. Kiểm tra NodeJS đã được cài thành công vào hệ thống 64](#_Toc104325181)

[Hình 3- 6.Downloads 65](#_Toc104325182)

[Hình 3- 7. Danh sách hiển thị 66](#_Toc104325183)

[Hình 3- 8.Màn hình đăng nhập 66](#_Toc104325184)

[Hình 3- 9.Màn hình tổng quan 67](#_Toc104325185)

[Hình 3- 10.Màn hình danh sách nhân viên 68](#_Toc104325186)

[Hình 3- 11.Form thêm/sửa thông tin nhân viên 68](#_Toc104325187)

[Hình 3- 12.Màn hình danh sách hợp đồng 69](#_Toc104325188)

[Hình 3- 13.Form thêm/sửa hợp đồng 70](#_Toc104325189)

[Hình 3- 14.Màn hình thiết lập Người dùng 70](#_Toc104325190)

[Hình 3- 15.Màn hình thiết lập Vai trò 71](#_Toc104325191)

[Hình 3- 16.Màn hình danh sách vị trí công việc 72](#_Toc104325192)

[Hình 3- 17.Form thêm/sửa vị trí công việc 72](#_Toc104325193)

[Hình 3- 18.Màn hình danh sách cơ cấu tổ chức 73](#_Toc104325194)

**MỞ ĐẦU**

1. **Tên đề tài**

*Xây dựng ứng dụng Web quản lý nhân sự sử dụng Angular, ASP.NET và MySQL*

1. **Lý do chọn đề tài**

***Tại sao lại chọn Quản lý nhân sự?***

Quản lý nhân sự là một trong những tác vụ quan trọng đối với một công ty. Quản lý nhân sự tốt sẽ mang lại nhân sự chất lượng tốt – yếu tố cốt lõi của doanh nghiệp, nắm bắt tình hình của nhân viên, đưa ra chiến lược tuyển dụng và kiểm soát các mức chi phí phù hợp. Nói cách khác, quản lý nhân sự đóng vai trò quản trị và ảnh hưởng nhiều đến sự thành công của một tổ chức.

***Tại sao lại xây dựng ứng dụng web?***

- Các ứng dụng web có thể chạy trên nhiều nền tảng bất kể hệ điều hành hay thiết bị, miễn là trình duyệt tương thích.

- Tất cả người dùng đều được truy cập cùng một phiên bản, điều này giúp loại bỏ mọi sự cố tương thích.

- Chúng không được cài đặt trên ổ cứng, do đó loại bỏ được các hạn chế về không gian.

- Chúng giảm chi phí cho cả doanh nghiệp và người dùng cuối do doanh nghiệp không cần sự support và bảo trì cho các ứng dụng web này, chưa kể chúng cũng ko đòi hỏi yêu cầu cao ở máy tính người dùng cuối để có thể thực thi.

***Tại sao lại sử dụng Angular, ASP.NET Core và MySQL?***

- Angular: là một Open-source farmework miễn phí chuyên dụng cho mục đích xây dựng ứng dụng web Single Page Application (SPA). Angular được phát triển từ những năm 2009 và được duy trì bởi Google. Đây được xem là một trong những farmework front-end mạnh mẽ nhất chuyên dụng nhất cho các lập trình viên.

- ASP.NET Core: là một Open-source framework đa nền tảng cho việc xây dựng các trang web, ứng dụng và dịch vụ web được phát hành bởi Microsoft. ASP.NET Core được thiết kế tích hợp với nhiều client side frameworks một cách liên tục trong đó có Angular.

- MySQL: là hệ quản trị cơ sở dữ liệu mã nguồn mở lớn nhất thế giới, đặc biết được dùng rất nhiều cho các dự án, ứng dụng, website trên internet.

**→** *Nội dung trên chính là cơ sở để lựa chọn đề tài* **“*Xây dựng ứng dụng Web quản lý nhân sự sử dụng Angular, ASP.NET và MySQL*” *.***

1. **Mục đích của đề tài**

* Tìm hiểu và áp dụng các công nghệ: Angular, ASP.NET Core, MySQL để xây dựng ứng dụng web:
  + Sử dụng Angular để xây dựng Single Page Application
  + Sử dụng ASP.NET Core để viết API
  + Sử dụng MySQL để lữu trữ, truy xuất dữ liệu
* Xây dựng ứng dụng web quản lý nhân sự đáp ứng được các chức năng:
  + Xem dữ liệu tổng quan nhân sự
  + Xem, thêm, sửa, xoá nhân viên và các thông tin liên quan
  + Thiết lập người dùng, vai trò cơ cấu tổ chức, vị trí công việc

1. **Bố cục đề tài**

(Nội dung chính chia thành 3 chương)

**Chương 1:** Giới thiệu công nghệ sử dụng

Giới thiệu các công nghệ sử dụng để xây dựng ứng dụng: Angular, ASP.NET Core, MySQL

**Chương 2:** Phân tích thiết kế hệ thống

Phân tích tích yêu cầu, mô tả các biểu đồ use case, biểu đồ trình tự, sơ đồ hoạt động và thiết kế giao diện của từng phân hệ, chức năng ứng dụng quản lý nhân sự

**Chương 3:** Cài đặt và thử nghiệm

Mô tả cách cài đặt, thử nghiệm sản phẩm.

**CHƯƠNG 1: GIỚI THIỆU CÔNG NGHỆ SỬ DỤNG**

**1.1. Angular**

Angular là một Open-source farmework miễn phí chuyên dụng cho mục đích xây dựng ứng dụng web Single Page Application (SPA). Angular được phát triển từ những năm 2009 và được duy trì bởi Google. Đây được xem là một trong những farmework front-end mạnh mẽ nhất chuyên dụng nhất cho các lập trình viên.

Các tính năng cốt lõi của AngularJS:

**Data-binding**: Nó tự động đồng bộ hóa dữ liệu giữa thành phần model và view.

**Scope**: Là những đối tượng hướng đến model, nó hoạt động như là cầu nối giữa controller và view.

**Controller**: Đây là những tính năng của AngularJS mà được giới hạn tới một scope cụ thể.

**Service**: AngularJS hoạt động với một vài dịch vụ (service) có sẵn , ví dụ $http để tạo XMLHttpRequests. Nó là các singleton object mà được khởi tạo duy nhất một lần trong ứng dụng.

**Filter**: Nó lựa chọn (hay là lọc) các tập con từ tập item trong các mảng và trả về các mảng mới.

**Directive**: Directive là các marker trong các phần tử DOM (như các phần tử, thuộc tính, css và nhiều hơn thế). Nó có thể dùng để tạo các thẻ HTML riêng phục vụ những mục đích riêng. AngularJS có những directive có sẵn như ngBind, ngModel…

**Template**:Là các rendered view với các thông tin từ controller và model. Nó có thể được sử dụng trong các file riêng rẽ (ví dụ như index.jsp) hoặc nhiều view với một trang sử dụng "partials".

**Routing**: Là khái niệm của sự chuyển dịch qua lại các view.

**Model View Whatever**: MVC là một mô hình thiết kế để phân chia các ứng dụng thành nhiều phần khác nhau (gọi là Model, View và Controller), một phần sử dụng với một nhiệm vụ nhất định. AngularJS không triển khai MVC theo cách truyền thống, mà gắn liền hơn với Model-View-ViewModel. Nhóm phát triển AngularJS đã đặt tên vui cho mô hình này là Model View Whatever.

**Deep Linking**: Cho phép bạn mã hóa trạng thái các ứng dụng trên địa chỉ URL để nó có thể được bookmark. Các ứng dụng có thể được phục hồi lại từ các địa chỉ URL với cùng một trạng thái.

**Dependency Injection**: AngularJS có sẵn một hệ thống con dependency injection để giúp các lập trình viên tạo ra các ứng dụng dễ phát triển, dễ hiểu và kiểm tra.

**1.2. ASP.NET Core**

ASP.NET Core là một phiên bản mới của ASP.NET, được phát hành bởi Microsoft và là một mã nguồn mở trên GitHub. ASP.NET Core được sử dụng để phát triển khuôn khổ website và có thể thích ứng với nhiều trình duyệt khác nhau như Windows, Mac hoặc Linux kể cả trên nền tảng MVC. Ban đầu, phiên bản này có tên là ASP.NET 5 nhưng sau đó được đổi tên thành ASP.NET Core.

ASP.NET Core được thiết kế để tối ưu development framework cho những ứng dụng cái mà được chạy on-promise hay được triển khai trên đám mây. ASP.NET Core bao gồm các thành phần theo hướng module với mục đích tối thiểu tài nguyên và tiết kiệm chi phí khi phát triển. Đồng thời, ASP.NET Core cũng là một mã nguồn mở, một xu thế mà các ngôn ngữ lập trình hiện nay hướng đến.

**Những lợi ích khi sử dụng ASP.NET Core**

**Xây dựng web UI và web API với ASP.NET Core MVC**

* Người dùng có thể tạo ra những ứng dụng web có thể thực hiện testing theo mô hình MVC.
* Xây dựng HTTP services hỗ trợ nhiều định dạng và đầy đủ những hỗ trợ cho nội dung của dữ liệu trả về.
* Razor cung cấp ngôn ngữ tạo Views hiệu quả.
* Tag Helper cho phép code server side tham gia vào quá trình tạo và render phần tử HTML.
* Model Binding có thể tự động ánh xạ dữ liệu từ HTTP request tới những tham số của method action.
* Model validation thực hiện validate client và server một cách tự động.

**Phát triển client-side**

ASP.NET Core được thiết kế tích hợp với nhiều client side frameworks một cách liên tục bao gồm AngularJS, Bootstrap và KnockoutJS.

**Những lợi ích khác**

Ngoài ra, ASP.NET Core còn được đánh giá cao khi mang trong mình nhiều lợi ích khác như:

* Là khung mẫu và gọn gàng hơn kiến trúc nhiều tầng.
* Là một khung công tác với mã nguồn mở.
* Dễ dàng tạo ứng dụng ASP.NET đa nền tảng trên Windows, Mac và Linux.
* Cấu hình là môi trường sẵn sàng cho đám mây.
* Khả năng lưu trữ trên: IIS, HTTP.sys, Kestrel, Nginx, Apache và Docker.

**1.3. MySQL**

**MySQL là gì?**

MySQL là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (Relational Database Management System, viết tắt là RDBMS) hoạt động theo mô hình client-server. **[RDBMS](https://en.wikipedia.org/wiki/Relational_database_management_system" \t "_blank)** là một phần mềm hay dịch vụ dùng để tạo và quản lý các cơ sở dữ liệu (Database) theo hình thức quản lý các mối liên hệ giữa chúng.

MySQL là một trong số các phần mềm RDBMS. RDBMS và MySQL thường được cho là một vì độ phổ biến quá lớn của MySQL. **[Các ứng dụng web lớn nhất](https://stackshare.io/mysql" \t "_blank)** như Facebook, Twitter, YouTube, Google, và Yahoo! đều dùng MySQL cho mục đích lưu trữ dữ liệu. Kể cả khi ban đầu nó chỉ được dùng rất hạn chế nhưng giờ nó đã tương thích với nhiều hạ tầng máy tính quan trọng như Linux, macOS, Microsoft Windows, và Ubuntu.

**Database**

Database là tập hợp dữ liệu theo cùng một cấu trúc. Hãy thử nghĩ về việc chụp hình tự sướng: bạn nhấn nút chụp ảnh về chính bản thân bạn. Hình ảnh là dữ liệu, thư viện lưu ảnh là cơ sở dữ liệu. Cơ sở dữ liệu, hay database, là nơi chứa và sắp đặt dữ liệu. Dữ liệu được đặt trong một bộ dữ liệu chung, dataset, được tổ chức sắp xếp giống như một bảng tính vậy. Mỗi “bảng” này có liên hệ với nhau theo cách nào đó. Vì vậy từ Relational (liên hệ) trong RDBMS có ý nghĩa như vậy. Nếu phần mềm không hỗ trợ mô hình dữ liệu quan hệ với nhau như vậy thì gọi là DBMS.

**Open source**

Open source dịch là mã nguồn mở, có nghĩa là ai cũng có thể dùng và chỉnh sửa nó. Bất kỳ ai cũng có thể cài đặt phần mềmn ày. Bạn cũng có thể học cách tùy chỉnh phần mềm theo nhu cầu của bạn. Tuy nhiên, giấy phép GPL ([**GNU Public License**](https://www.gnu.org/licenses/gpl-3.0.en.html)) quyết định bạn có thể làm gì tùy vào điều kiện nhất định. Phiên bản thương mại cũng được xuất bản nếu bạn cần thêm chủ quyền linh hoạt và hỗ trợ cao cấp.

**Mô hình Client-server**

Máy tính cài đặt và chạy phần mềm RDBMS được gọi là client (máy khách). Mỗi khi chúng cần truy cập dữ liệu, chúng kết nối tới máy chủ (server) RDBMS. Cách thức này chính là mô hình “client-server”.

**MySQL Server**

MySQL Server là máy tính hay một hệ các máy tính cài đặt phần mềm MySQL dành cho server để giúp bạn lưu trữ dữ liệu trên đó, để máy khách có thể truy cập vào quản lý. Dữ liệu này được đặt trong các bảng, và các bảng có mối liên hệ với nhau. MySQL server nhanh, an toàn, đáng tin cậy. Phần mềm MySQL cũng miễn phí và được phát triển, phân phối và hỗ trợ bởi Oracle Corporation.

**MySQL Client**

MySQL client không hẵn phải cài phần mềm MySQL của Oracle mà là nói chung của mọi phần mềm có thể thực hiện truy vấn lên một MySQL server và nhận kết quả trả về. MySQL client điển hình là đoạn mã PHP script trên một máy tính hay trên cùng server dùng để kết nối tới cơ sở dữ liệu MySQL database. Phpmyadmin cũng là một MySQL client có giao diện người dùng. Một số công cụ miễn phí dùng làm MySQL là:

* MySQL Workbench (Mac, Windows, Linux), Miễn phí, mã nguồn mở
* Sequel Pro (Mac), miễn phí, mã nguồn mở
* HeidiSQL (Windows; chạy trên Mac hoặc Linux bằng WINE emulator), miễn phí
* phpMyAdmin (web app), miễn phí, mã nguồn mở

**SQL**

[**MySQL và SQL không giống nhau**](https://www.hostinger.vn/huong-dan/su-khac-biet-giua-mysql-va-sql-server/). Hãy nhớ, MySQL là một trong các phần mềm RDBMS, hoạt động theo mô hình client-server. Nhưng, làm thế nào clietn và server liên lạc với nhau trong môi trường của RDBMS? Chúng sử dụng ngôn ngữ truy vấn có cấu trúc chung – Structured Query Language (SQL). Nếu bắt gặp một thương hiệu nào đó có SQL kèm theo, như PostgreSQL, Microsoft SQL server, chúng thường là những thương hiệu sử dụng syntax của SQL. RDBMS software lại được viết bằng ngôn ngữ lập trình, nhưng luôn sử dụng SQL làm ngôn ngữ chính để tương tác với database. Bản thân MySQL được viết bằng C và C++. Hãy tưởng tượng các nước Nam Mỹ, chúng có vị trí địa lý và lịch sử khác nhau, nhưng đều nói tiếng Tây Ban Nha.

Nhà khoa học máy tính Ted Codd tạo ra SQL vào đầu thập niên 1970s với IBM theo mô hình liên hệ. Nó được sử dụng rộng khắp vào năm 1974 và nhanh chóng thay thế các ngôn ngữ lỗi thời khác như ISAM và VISAM. SQL nói server phải làm gì với dữ liệu. Giống khi bạn nhập mật khẩu để đăng nhập vào trang quản lý của WordPress, trong trường hợp này, lệnh SQL sẽ nói cho server biết cần phải thực hiện một số hành động:

* Truy vấn dữ liệu – data query: yêu cầu một thông tin cụ thể trong database có sẵn.
* Điều hành dữ liệu – data manipulation: thêm, xóa, sửa, sắp xếp, và các tác vụ điều hành khác để chỉnh sửa dữ liệu, giá trị, và cách hiển thị.
* Loại dữ liệu – data identity: định nghĩa loại dữ liệu, như là đổi từ dữ liệu số thành số nguyên. Nó cũng bao gồm định nghĩa một schema hoặc định nghĩa mối quan hệ của từng bảng trong cơ sở dữ liệu
* Quyền truy cập data – data access control: cung cấp biện pháp bảo mật để bảo vệ dữ liệu, bao gồm việc quyết định ai có thể xem thông tin dữ liệu nào trong database

**CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

**2.1. Yêu cầu**

Yêu cầu chức năng:

- Xem dữ liệu tổng quan nhân sự

- Xem, thêm, sửa, xoá nhân viên và các thông tin liên quan

- Thiết lập người dùng, vai trò, cơ cấu tổ chức, vị trí công việc

Yêu cầu phi chức năng:

- Giao diện đẹp, thuận tiện cho người dùng

- Thao tác thực hiện các chức năng mượt mà

**2.2. Biểu đồ use case**

**2.2.1. Biểu đồ use case tổng quan**

Diagram

Description automatically generated

Hình 2- 1. Biểu đồ Use Case Tổng quát

Use Case Đăng nhập: Cho phép người dùng đăng nhập vào hệ thống

Use Case Quản lý nhân viên: Cho phép người dùng quản lý thông tin nhân viên

Use Case Quản lý hợp đồng: Cho phép người dùng quản lý thông tin hợp đồng lao động

Use Case Xem tổng quan: Cho phép người dùng xem tổng quan biểu dồ dữ liệu báo cáo

Use Case Xem báo cáo: Cho phép người dùng xem chi tiết dữ liệu từng báo cáo

Use Case Quản lý cơ cấu tổ chức: Cho phép quản trị ứng dụng quản lý cơ cấu tổ chức

Use Case Quản lý vị trí công việc: Cho phép quản trị ứng dụng quản lý vị trí công việc

Use Case Quản lý vai trò người dùng: Cho phép quản trị ứng dụng quản lý các vai trò của người dùng

Use Case Quản lý người dùng: Cho phép quản trị ứng dụng quản lý người dùng truy cập ứng dụng thông tin nhân sự

Use Case kích hoạt tài khoản: Cho phép người dùng kích hoạt tài khoản và tạo mật khẩu để đăng nhập vào ứng dụng khi được người quản trị thêm mới

**2.2.2 Biểu đồ phân rã Use Case**

Use Case Đăng nhập

Diagram

Description automatically generated

Hình 2- 2. Use Case Đăng nhập

Use Case Quản lý nhân viên

Diagram

Description automatically generated

Hình 2- 3. Use Case Quản lý nhân viên

Use Case Quản lý hợp đồng

Diagram

Description automatically generated

Hình 2- 4. Use Case Quản lý hợp đồng

Use Case Quản lý cơ cấu tổ chức

Diagram

Description automatically generated

Hình 2- 5. Quản lý cơ cấu tổ chức

Use Case Quản lý vị trí công việc

Diagram

Description automatically generated

Hình 2- 6. Quản lý vị trí công việc

Use Case Quản lý vai trò người dùng

Diagram

Description automatically generated

Hình 2- 7. Use Case Quản lý vai trò người dùng

Use Case Quản lý người dùng

Diagram

Description automatically generated

Hình 2- 8. Use Case Quản lý người dùng

**2.3. Mô tả chi tiết Use Case**

**2.3.1. Use Case Đăng nhập**

UseCase này cho phép người đăng nhập vào hệ thống

* Luồng sự kiện
  + Luồng cơ bản

Use Case này bắt đầu khi người dùng nhập tên đăng nhập và mật khẩu sau đó click button “Đăng nhập”. Hệ thống sẽ kiểm tra tài khoản và mật khẩu trong bảng users.Nếu tài khoản tồn tại và mật khẩu chính xác hệ thống sẽ đưa người dùng vào giao diện ứng dụng, nếu sai thì sẽ thông báo “Tài khoản hoặc mật khẩu không hợp lệ”.

* + Luồng rẽ nhánh

Tại bất kỳ thời điểm nào trong quá trình thực hiện Use Case nếu kết nối với cơ sở dữ liệu bị lỗi thì sẽ hiển thị thông báo lỗi và Use Case kết thúc. Nếu trong quá trình đăng nhập người dùng nhập sai tài khoản hoặc mật khẩu thì hệ thống sẽ hiển thị thông báo lỗi và Use Case kết thúc.

* Các yêu cầu đặc biệt

Không có.

* Tiền điều kiện

Không có.

* Hậu điều kiện

Không có.

* Điểm mở rộng

Không có.

**2.3.2. Use case Quản lý nhân viên**

Use Case này cho phép người dùng xem danh sách, xem chi tiết, thêm, sửa, xóa, tìm kiếm, xuất khẩu các nhân viên trong bảng users

* **Luồng sự kiện**
* Luồng cơ bản

1. Use Case này bắt đầu khi người dùng nhấn vào nút “Nhân viên” trên thanh menu sidebar. Hệ thống lấy thông tin của các nhân viên trong bản users và hiển thị lên màn hình
2. Xem chi tiết Nhân viên

Người dùng nhấn vào một bản ghi trên màn hình danh sách Nhân viên.Hệ thống sẽ lấy thông tin chi tiết của nhân viên và hiển thị trên màn hình chi tiết Nhân viên

1. Thêm Nhân viên:
2. Người dùng nhấn vào nút “Thêm” trên màn hình danh sách Nhân viên.Hệ thống hiển thị màn hình yêu cầu nhập thông tin chi tiết cho Nhân viên(Mã nhân viên, Họ và tên, Ngày sinh, Giới tính, Số điện thoại, Số CMT, ...)
3. Người dùng nhập thông tin cho Nhân viên và nhấn vào nút “Lưu”.Hệ thống sẽ tạo ra một bản ghi mới trong bảng users và hiển thị danh sách Nhân viên được cập nhật.
4. Sửa Nhân viên
5. Người dùng nhấn vào nút “Sửa” trên một dòng Nhân viên hoặc trên màn hình chi tiết Nhân viên.Hệ thống hiển thị một màn hình với các thông tin cũ của Nhân viên lên các ô nhập dữ liệu
6. Người dùng nhập thông tin mới cho Nhân viên và nhấn vào nút “Cập nhật”.Hệ thống sẽ cập nhật thông tin của Nhân viên trong bảng users và hiển thị danh sách Nhân viên đã cập nhật.
7. Xóa Nhân viên
8. Người dùng nhấn vào nút “Xóa” trên một dòng Nhân viên.Hệ thống sẽ hiển thị một màn hình yều cầu xác nhận xóa.
9. Người dùng nhấn vào nút “Đồng ý” Hệ thống sẽ xóa Nhân viên được chọn khỏi bảng users và hiển thị danh sách Nhân viên đã được cập nhật.
10. Tìm kiếm nhân viên

Khi người dùng nhập từ khóa tìm kiếm trên ô tìm tiếm và nhấn “Enter” hệ thống sẽ lấy ra các bản ghi trùng với từ khóa tìm kiếm và hiển thị ra danh sách nhân viên

1. Xuất khẩu danh sách Nhân viên

Người dùng nhấn vào nút “Xuất khẩu” trên màn hình danh sách Nhân viên.Hệ thống sẽ lấy tất cả danh sách Nhân viên trong bảng users và xuất ra file có định dạng .xlsx.

* Luồng rẽ nhánh

1. Tại bước 3b hoặc 4b nếu người dùng nhập dữ liệu thông tin Nhân viên không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại.Người dùng có thể nhập lại hoặc nhấn nút “Hủy” để kết thúc Use Case.
2. Tại bước 3b hoặc 4b trong luồng cơ bản nếu người dùng nhấn nút “Hủy: Hệ thống sẽ bỏ qua thao tác thêm mới hoặc cập nhật và hiển thị danh sách Nhân viên trong bảng users.
3. Tại bước 5b nếu người dùng nhấn vào nút “Hủy” thì hệ thống sẽ bỏ qua thao tác xóa, ẩn màn hình xác nhận xóa và hiển thị danh sách Nhân viên trong bảng users.
4. Tại bất kỳ thời điểm nào trong quá trình thực hiện Use Case nếu kết nối đến cơ sở dữ liệu bị lỗi thì hệ thống sẽ hiển thị thông báo lỗi và Use Case kết thúc.
5. Tại bước 2, 3, 4, 5 nếu người dùng không được phân quyền thao tác với các tính năng thì Hệ thống sẽ hiển thị thông báo cảnh báo không có quyền thực hiện chức năng và Use Case kết thúc.

* **Các yêu cầu đặc biệt**

Use Case này chỉ cho phép người dùng có quyền thao tác với phân hệ Nhân viên thực hiện

* **Tiền điều kiện**

Người dùng phải đăng nhập vào hệ thống và có quyền thao tác với phân hệ Nhân viên

* **Hậu điều kiện**

Nếu Use Case kết thúc thành công thì thông tin về nhân viên sẽ được cập nhật vào trong cơ sở dữ liệu

* **Điểm mở rộng**

Không có

**2.3.3. Mô tả Quản lý hợp đồng**

Use Case này cho phép người dùng xem danh sách, xem chi tiết, thêm, sửa, xóa, tìm kiếm, xuất khẩu các Hợp đồng trong bảng contracts

* **Luồng sự kiện**
* Luồng cơ bản

1. Use Case này bắt đầu khi người dùng nhấn vào nút “Hợp đồng” trên thanh menu sidebar. Hệ thống lấy thông tin của các hợp đồng trong bản contracts và hiển thị lên màn hình
2. Xem chi tiết Hợp đồng

Người dùng nhấn vào một bản ghi trên màn hình danh sách Hợp đồng.Hệ thống sẽ lấy thông tin chi tiết của Hợp đồng và hiển thị trên màn hình chi tiết Hợp đồng

1. Thêm Hợp đồng:
2. Người dùng nhấn vào nút “Thêm” trên màn hình danh sách Hợp đồng Hệ thống hiển thị màn hình yêu cầu nhập thông tin chi tiết cho Hợp đồng (Số hợp đồng, Họ và tên, thời hạn hợp đồng, Ngày bắt đầu, Ngày kết thúc, Lương cơ bản, Lương đóng bảo hiểm, ...)
3. Người dùng nhập thông tin cho Hợp đồng và nhấn vào nút “Lưu”.Hệ thống sẽ tạo ra một bản ghi mới trong bảng contracts và hiển thị danh sách Hợp đồng được cập nhật.
4. Sửa Hợp đồng
5. Người dùng nhấn vào nút “Sửa” trên một dòng Hợp đồng hoặc trên màn hình chi tiết Hợp đồng.Hệ thống hiển thị một màn hình với các thông tin cũ của Hợp đồng lên các ô nhập dữ liệu
6. Người dùng nhập thông tin mới cho Hợp đồng và nhấn vào nút “Cập nhật”.Hệ thống sẽ cập nhật thông tin của Hợp đồng trong bảng contracts và hiển thị danh sách Hợp đồng đã cập nhật.
7. Xóa Hợp đồng
8. Người dùng nhấn vào nút “Xóa” trên một dòng Hợp đồng.Hệ thống sẽ hiển thị một màn hình yều cầu xác nhận xóa.
9. Người dùng nhấn vào nút “Đồng ý” Hệ thống sẽ xóa Hợp đồng được chọn khỏi bảng contracts và hiển thị danh sách Hợp đồng đã được cập nhật.
10. Tìm kiếm Hợp đồng

Khi người dùng nhập từ khóa tìm kiếm vào ô tìm kiếm và nhấn “Enter” Hệ thống sẽ lấy các bản ghi hợp đồng trùng với từ khóa tìm kiếm và hiển thị ra danh sách hợp đồng

1. Xuất khẩu danh sách Hợp đồng

Người dùng nhấn vào nút “Xuất khẩu” trên màn hình danh sách Hợp đồng.Hệ thống sẽ lấy tất cả danh sách Hợp đồng trong bảng contracts và xuất ra file có định dạng .xlsx.

* Luồng rẽ nhánh

1. Tại bước 3b hoặc 4b nếu người dùng nhập dữ liệu thông tin Hợp đồng không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại.Người dùng có thể nhập lại hoặc nhấn nút “Hủy” đễ kết thức Use Case.
2. Tại bước 3b hoặc 4b trong luồng cơ bản nếu người dùng nhấn nút “Hủy: hệ thống sẽ bỏ qua thao tác thêm mới hoặc cập nhật và hiển thị danh sách Hợp đồng trong bảng contracts.
3. Tại bước 5b nếu người dùng nhấn vào nút “Hủy” thì hệ thống sẽ bỏ qua thao tác xóa, ẩn màn hình xác nhận xóa và hiển thị danh sách Hợp đồng trong bảng contracts.
4. Tại bất kỳ thời điểm nào trong quá trình thực hiện Use Case nếu kết nối đến cơ sở dữ liệu bị lỗi thì hệ thống sẽ hiển thị thông báo lỗi và Use Case kết thúc.
5. Tại bước 2, 3, 4, 5 nếu người dùng không được phân quyền thao tác với các tính năng thì Hệ thống sẽ hiển thị thông báo cảnh báo không có quyền thực hiện chức năng và Use Case kết thúc.

* **Các yêu cầu đặc biệt**

Use Case này chỉ cho phép người dùng có quyền thao tác với phân hệ Hợp đồng thực hiện.

* **Tiền điều kiện**

Người dùng phải đăng nhập vào hệ thống và có quyền thao tác với phân hệ Hợp đồng.

* **Hậu điều kiện**

Nếu Use Case kết thúc thành công thì thông tin về Hợp đồng sẽ được cập nhật vào trong cơ sở dữ liệu.

* **Điểm mở rộng**

Không có

**2.3.4. Use Case Quản lý cơ cấu tổ chức**

Use Case này cho phép người quản trị ứng dụng xem danh sách, xem chi tiết, thêm, sửa, xóa, tìm kiếm các Cơ cấu tổ chức trong bảng organizationunits

* **Luồng sự kiện**
* Luồng cơ bản

1. Use Case này bắt đầu khi người quản trị ứng dụng nhấn vào nút “Thiết lập” “Cơ cấu tổ chức” trên thanh menu sidebar. Hệ thống lấy thông tin của các Cơ cấu tổ chức trong bản organizationunits và hiển thị lên màn hình
2. Xem chi tiết Cơ cấu tổ chức

Quản trị ứng dụng nhấn vào một bản ghi trên màn hình danh sách Cơ cấu tổ chức.Hệ thống sẽ lấy thông tin chi tiết của Cơ cấu tổ chức và hiển thị trên màn hình chi tiết Cơ cấu tổ chức

1. Thêm Cơ cấu tổ chức:
2. Quản trị ứng dụng nhấn vào nút “Thêm” trên màn hình danh sách Cơ cấu tổ chức Hệ thống hiển thị màn hình yêu cầu nhập thông tin chi tiết cho Cơ cấu tổ chức (Mã cơ cấu tổ chức, Tên cơ cấu tổ chức, Thuộc đơn vị, địa chỉ, ...)
3. Quản trị ứng dụng nhập thông tin cho Cơ cấu tổ chức và nhấn vào nút “Lưu”.Hệ thống sẽ tạo ra một bản ghi mới trong bảng contracts và hiển thị danh sách Cơ cấu tổ chức được cập nhật.
4. Sửa Cơ cấu tổ chức
5. Quản trị ứng dụng nhấn vào nút “Sửa” trên một dòng Cơ cấu tổ chức hoặc trên màn hình chi tiết Cơ cấu tổ chức.Hệ thống hiển thị một màn hình với các thông tin cũ của Cơ cấu tổ chức lên các ô nhập dữ liệu
6. Quản trị ứng dụng nhập thông tin mới cho Cơ cấu tổ chức và nhấn vào nút “Cập nhật”.Hệ thống sẽ cập nhật thông tin của Cơ cấu tổ chức trong bảng organizationunits và hiển thị danh sách Cơ cấu tổ chức đã cập nhật.
7. Xóa Cơ cấu tổ chức
8. Quản trị ứng dụng nhấn vào nút “Xóa” trên một dòng Cơ cấu tổ chức.Hệ thống sẽ hiển thị một màn hình yều cầu xác nhận xóa.
9. Quản trị ứng dụng nhấn vào nút “Đồng ý” Hệ thống sẽ xóa Cơ cấu tổ chức được chọn khỏi bảng organizationunits và hiển thị danh sách Cơ cấu tổ chức đã được cập nhật.
10. Tìm kiếm Cơ cấu tổ chức

Khi người quản trị nhập từ khóa tìm kiếm vào ô tìm kiếm và nhấn “Enter” Hệ thống sẽ lấy các bản ghi Cơ cấu tổ chức thỏa mãn và hiển thị lên danh sách.

* Luồng rẽ nhánh

1. Tại bước 3b hoặc 4b nếu người dùng nhập dữ liệu thông tin Cơ cấu tổ chức không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại.Người dùng có thể nhập lại hoặc nhấn nút “Hủy” đễ kết thức Use Case.
2. Tại bước 3b hoặc 4b trong luồng cơ bản nếu người dùng nhấn nút “Hủy: hệ thống sẽ bỏ qua thao tác thêm mới hoặc cập nhật và hiển thị danh sách Cơ cấu tổ chức trong bảng organizationunits.
3. Tại bước 5b nếu người dùng nhấn vào nút “Hủy” thì hệ thống sẽ bỏ qua thao tác xóa, ẩn màn hình xác nhận xóa và hiển thị danh sách Cơ cấu tổ chức trong bảng organizationunits.
4. Tại bất kỳ thời điểm nào trong quá trình thực hiện Use Case nếu kết nối đến cơ sở dữ liệu bị lỗi thì hệ thống sẽ hiển thị thông báo lỗi và Use Case kết thúc.

* **Các yêu cầu đặc biệt**

Use Case này chỉ cho phép người quản trị ứng dụng thực hiện.

* **Tiền điều kiện**

Người dùng phải đăng nhập vào hệ thống và có vai trò quản trị ứng dụng

* **Hậu điều kiện**

Nếu Use Case kết thúc thành công thì thông tin về Cơ cấu tổ chức sẽ được cập nhật vào trong cơ sở dữ liệu.

* **Điểm mở rộng**

Không có

**2.3.5. Use case Quản lý Vị trí công việc**

Use Case này cho phép người quản trị ứng dụng xem danh sách, xem chi tiết, thêm, sửa, xóa, tìm kiếm các Vị trí công việc trong bảng jobpositions

* **Luồng sự kiện**
* Luồng cơ bản

1. Use Case này bắt đầu khi người quản trị ứng dụng nhấn vào nút “Thiết lập” “Vị trí công việc” trên thanh menu sidebar. Hệ thống lấy thông tin của các Vị trí công việc trong bản jobpositions và hiển thị lên màn hình
2. Xem chi tiết Vị trí công việc

Quản trị ứng dụng nhấn vào một bản ghi trên màn hình danh sách Vị trí công việc.Hệ thống sẽ lấy thông tin chi tiết của Vị trí công việc và hiển thị trên màn hình chi tiết Vị trí công việc

1. Thêm Vị trí công việc:
2. Quản trị ứng dụng nhấn vào nút “Thêm” trên màn hình danh sách Vị trí công việc Hệ thống hiển thị màn hình yêu cầu nhập thông tin chi tiết cho Vị trí công việc (Mã Vị trí công việc, Tên Vị trí công việc,Cơ cấu tổ chức,...)
3. Quản trị ứng dụng nhập thông tin cho Vị trí công việc và nhấn vào nút “Lưu”.Hệ thống sẽ tạo ra một bản ghi mới trong bảng contracts và hiển thị danh sách Vị trí công việc được cập nhật.
4. Sửa Vị trí công việc
5. Quản trị ứng dụng nhấn vào nút “Sửa” trên một dòng Vị trí công việc hoặc trên màn hình chi tiết Vị trí công việc.Hệ thống hiển thị một màn hình với các thông tin cũ của Vị trí công việc lên các ô nhập dữ liệu
6. Quản trị ứng dụng nhập thông tin mới cho Vị trí công việc và nhấn vào nút “Cập nhật”.Hệ thống sẽ cập nhật thông tin của Vị trí công việc trong bảng jobpositions và hiển thị danh sách Vị trí công việc đã cập nhật.
7. Xóa Vị trí công việc
8. Quản trị ứng dụng nhấn vào nút “Xóa” trên một dòng Vị trí công việc.Hệ thống sẽ hiển thị một màn hình yều cầu xác nhận xóa.
9. Quản trị ứng dụng nhấn vào nút “Đồng ý” Hệ thống sẽ xóa Vị trí công việc được chọn khỏi bảng jobpositions và hiển thị danh sách Vị trí công việc đã được cập nhật.
10. Tìm kiếm Vị trí công việc

Khi người quản trị nhập từ khóa tìm kiếm vào ô tìm kiếm và nhấn “Enter” Hệ thống sẽ lấy các bản ghi Vị trí công việc thỏa mãn và hiển thị lên danh sách.

* Luồng rẽ nhánh

1. Tại bước 3b hoặc 4b nếu người dùng nhập dữ liệu thông tin Vị trí công việc không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại.Người dùng có thể nhập lại hoặc nhấn nút “Hủy” đễ kết thức Use Case.
2. Tại bước 3b hoặc 4b trong luồng cơ bản nếu người dùng nhấn nút “Hủy: hệ thống sẽ bỏ qua thao tác thêm mới hoặc cập nhật và hiển thị danh sách Vị trí công việc trong bảng jobpositions.
3. Tại bước 5b nếu người dùng nhấn vào nút “Hủy” thì hệ thống sẽ bỏ qua thao tác xóa, ẩn màn hình xác nhận xóa và hiển thị danh sách Vị trí công việc trong bảng jobpositions.
4. Tại bất kỳ thời điểm nào trong quá trình thực hiện Use Case nếu kết nối đến cơ sở dữ liệu bị lỗi thì hệ thống sẽ hiển thị thông báo lỗi và Use Case kết thúc.

* **Các yêu cầu đặc biệt**

Use Case này chỉ cho phép người quản trị ứng dụng thực hiện.

* **Tiền điều kiện**

Người dùng phải đăng nhập vào hệ thống và có vai trò quản trị ứng dụng

* **Hậu điều kiện**

Nếu Use Case kết thúc thành công thì thông tin về Vị trí công việc sẽ được cập nhật vào trong cơ sở dữ liệu.

* **Điểm mở rộng**

Không có

**2.3.6. Use case Quản lý Vai trò người dùng**

Use Case này cho phép người quản trị ứng dụng xem danh sách, xem chi tiết, thêm, sửa, xóa, tìm kiếm các Vai trò người dùng trong bảng jobpositions

* **Luồng sự kiện**
* Luồng cơ bản

1. Use Case này bắt đầu khi người quản trị ứng dụng nhấn vào nút “Thiết lập” “Vai trò người dùng” trên thanh menu sidebar. Hệ thống lấy thông tin của các Vai trò người dùng trong bản jobpositions và hiển thị lên màn hình
2. Xem chi tiết Vai trò người dùng

Quản trị ứng dụng nhấn vào một bản ghi trên màn hình danh sách Vai trò người dùng.Hệ thống sẽ lấy thông tin chi tiết của Vai trò người dùng và hiển thị trên màn hình chi tiết Vai trò người dùng

1. Thêm Vai trò người dùng:
2. Quản trị ứng dụng nhấn vào nút “Thêm” trên màn hình danh sách Vai trò người dùng Hệ thống hiển thị màn hình yêu cầu nhập thông tin chi tiết cho Vai trò người dùng (Tên Vai trò người dùng, Mô tả, Quyền trùy cập dữ liệu)
3. Quản trị ứng dụng nhập thông tin cho Vai trò người dùng và nhấn vào nút “Lưu”.Hệ thống sẽ tạo ra một bản ghi mới trong bảng contracts và hiển thị danh sách Vai trò người dùng được cập nhật.
4. Sửa Vai trò người dùng
5. Quản trị ứng dụng nhấn vào nút “Sửa” trên một dòng Vai trò người dùng hoặc trên màn hình chi tiết Vai trò người dùng.Hệ thống hiển thị một màn hình với các thông tin cũ của Vai trò người dùng lên các ô nhập dữ liệu
6. Quản trị ứng dụng nhập thông tin mới cho Vai trò người dùng và nhấn vào nút “Cập nhật”.Hệ thống sẽ cập nhật thông tin của Vai trò người dùng trong bảng jobpositions và hiển thị danh sách Vai trò người dùng đã cập nhật.
7. Xóa Vai trò người dùng
8. Quản trị ứng dụng nhấn vào nút “Xóa” trên một dòng Vai trò người dùng.Hệ thống sẽ hiển thị một màn hình yều cầu xác nhận xóa.
9. Quản trị ứng dụng nhấn vào nút “Đồng ý” Hệ thống sẽ xóa Vai trò người dùng được chọn khỏi bảng jobpositions và hiển thị danh sách Vai trò người dùng đã được cập nhật.
10. Tìm kiếm Vai trò người dùng

Khi người quản trị nhập từ khóa tìm kiếm vào ô tìm kiếm và nhấn “Enter” Hệ thống sẽ lấy các bản ghi Vai trò người dùng thỏa mãn và hiển thị lên danh sách.

* Luồng rẽ nhánh

1. Tại bước 3b hoặc 4b nếu người dùng nhập dữ liệu thông tin Vai trò người dùng không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại.Người dùng có thể nhập lại hoặc nhấn nút “Hủy” đễ kết thức Use Case.
2. Tại bước 3b hoặc 4b trong luồng cơ bản nếu người dùng nhấn nút “Hủy: hệ thống sẽ bỏ qua thao tác thêm mới hoặc cập nhật và hiển thị danh sách Vai trò người dùng trong bảng jobpositions.
3. Tại bước 5b nếu người dùng nhấn vào nút “Hủy” thì hệ thống sẽ bỏ qua thao tác xóa, ẩn màn hình xác nhận xóa và hiển thị danh sách Vai trò người dùng trong bảng jobpositions.
4. Tại bất kỳ thời điểm nào trong quá trình thực hiện Use Case nếu kết nối đến cơ sở dữ liệu bị lỗi thì hệ thống sẽ hiển thị thông báo lỗi và Use Case kết thúc.

* **Các yêu cầu đặc biệt**

Use Case này chỉ cho phép người quản trị ứng dụng thực hiện.

* **Tiền điều kiện**

Người dùng phải đăng nhập vào hệ thống và có vai trò quản trị ứng dụng

* **Hậu điều kiện**

Nếu Use Case kết thúc thành công thì thông tin về Vai trò người dùng sẽ được cập nhật vào trong cơ sở dữ liệu.

* **Điểm mở rộng**

Không có

**2.3.7. Use case Quản lý người dùng**

Use Case này cho phép người quản trị ứng dụng xem danh sách, thêm, xóa, phân quyền người dùng ứng dụng

* **Luồng sự kiện**
* Luồng cơ bản

1. Use Case này bắt đầu khi người dùng nhấn vào nút “Thiết lập” “Quản lý người dùng” trên thanh menu sidebar. Hệ thống lấy thông tin của các hợp đồng trong bảng users có tích chọn là người dùng và trạng thái là chờ xác nhận hoặc đang hoạt động
2. Thêm người dùng:
3. Người quản trị nhấn vào nút “Thêm” trên màn hình danh sách Người dùng Hệ thống hiển thị màn hình cho phép người quản trị ứng dụng chọn ra những nhân viên có chọn trường là người dùng và chọn vai trò cho người dùng
4. Sau khi nhấn “Thêm” Hệ thống sẽ gửi một email kích hoạt tài khoản đên email tài khoản của người dùng mới được thêm và chuyển trạng thái của người dùng đó về chờ xác nhận.
5. SửaNgười dùng
6. Người quản trị nhấn vào nút “Sửa” trên một dòng danh sách người dùng, Hệ thống sẽ cho phép người quản trị sửa lại thông tin vai trò người dùng
7. Người quản trị cập nhật thông tin vai trò mới cho người dùng, Hệ thống sẽ cập nhật dữ liệu vào trong cơ sở dữ liệu trong bảng users
8. Xóa người dùng
9. Người quản trị nhấn vào nút “Xóa” trên một dòng Người dùng.Hệ thống sẽ hiển thị một màn hình yều cầu xác nhận xóa.
10. Người dùng nhấn vào nút “Đồng ý” Hệ thống sẽ cập nhật lại trường InActive trong bản users thành true và người dùng đó không thể đăng nhập vào ứng dụng

* Luồng rẽ nhánh

1. Tại bước 3b hoặc 4b nếu người dùng nhập dữ liệu thông tin Hợp đồng không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại.Người dùng có thể nhập lại hoặc nhấn nút “Hủy” đễ kết thức Use Case.
2. Tại bước 3b hoặc 4b trong luồng cơ bản nếu người dùng nhấn nút “Hủy: hệ thống sẽ bỏ qua thao tác thêm mới hoặc cập nhật và hiển thị danh sách Hợp đồng trong bảng contracts.
3. Tại bước 5b nếu người dùng nhấn vào nút “Hủy” thì hệ thống sẽ bỏ qua thao tác xóa, ẩn màn hình xác nhận xóa và hiển thị danh sách Hợp đồng trong bảng contracts.
4. Tại bất kỳ thời điểm nào trong quá trình thực hiện Use Case nếu kết nối đến cơ sở dữ liệu bị lỗi thì hệ thống sẽ hiển thị thông báo lỗi và Use Case kết thúc.

* **Các yêu cầu đặc biệt**
* Use Case này chỉ cho phép người quản trị ứng dụng thực hiện.
* **Tiền điều kiện**
* Người dùng phải đăng nhập vào hệ thống và có vai trò quản trị ứng dụng
* **Hậu điều kiện**

Nếu Use Case kết thúc thành công thì sẽ gửi một email kích hoạt tài khoản đến email của người dùng

* **Điểm mở rộng**

Nếu người dùng trước đây đã kích hoạt tài khoản và bị xóa khỏi danh sách người dùng, nếu người quản trị thêm lại thì người đó có thể sử dụng tài khoản để đăng nhập luôn. Hệ thống sẽ không gửi lại email kích hoạt tài khoản nữa.

**2.4. Các biểu đồ các lớp tham gia vào một Use Case (VOPC)**

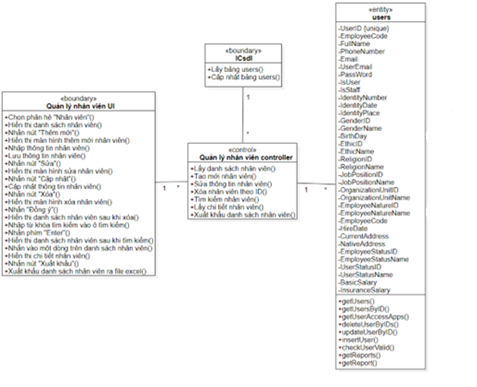
**2.4.1. Biểu đồ VOPC Use Case đăng nhập**

Diagram

Description automatically generated

Hình 2- 9. Biểu đồ VOPC Use Case đăng nhập

**2.4.2. Biểu đồ VOPC Use Case quản lý nhân viên**



Hình 2- 10. Biểu đồ VOPC Use Case quản lý nhân viên

**2.4.3. Biểu đồ VOPC Use Case quản lý hợp đồng**

Diagram

Description automatically generated

Hình 2- 11. Biểu đồ VOPC Use Case quản lý hợp đồng

**2.4.4. Biểu đồ VOPC Use Case quản lý cơ cấu tổ chức**

Diagram

Description automatically generated

Hình 2- 12. Biểu đò VOPC Use Case quản lý cơ cấu tổ chức

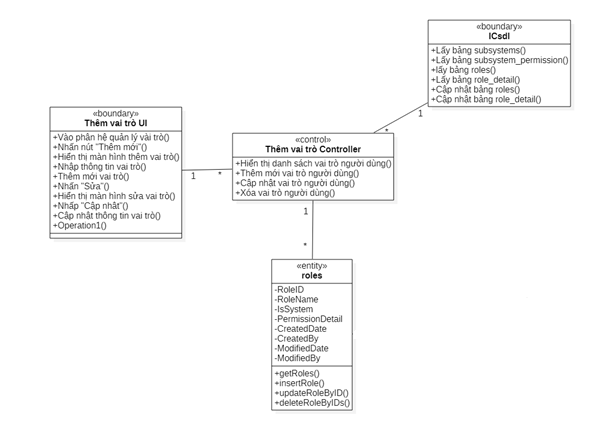
**2.4.5. Biểu đồ VOPC Use Case quản lý vị trí công việc**

Diagram

Description automatically generated

Hình 2- 13. Biểu đồ VOPC Use Case quản lý vị trí công việc

**2.4.6. Biểu đồ VOPC Use Case quản lý vai trò người dùng**



Hình 2- 14. Biểu đồ VOPC Use Case quản lý vai trò người dùng

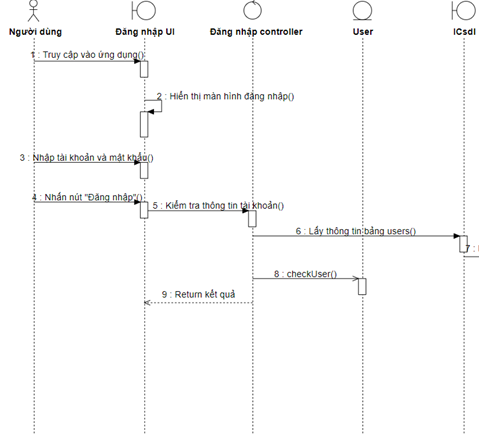
**2.4.7. Biểu đồ VOPC Use Case quản lý người dùng**

Diagram

Description automatically generated

Hình 2- 15. Biểu đồ VOPC Use Case quản lý người dùng

**2.5. Các biều đồ trình tự**



Hình 2- 16. Biểu đồ trình tự Use Case đăng nhập

A picture containing text, map, indoor, light

Description automatically generated

Hình 2- 17. Biểu đồ trình tự Use Case quản lý nhân viên

A picture containing text, indoor, map

Description automatically generated

Hình 2- 18. Biểu đồ trình tự Use Case quản lý hợp đồng

Timeline

Description automatically generated

Hình 2- 19. Biều đồ trình tự Use Case quản lý cơ cấu tổ chức

A picture containing graphical user interface

Description automatically generated

Hình 2- 20. Biều đồ trình tự Use Case quản lý vị trí công việc

A picture containing text, map, indoor

Description automatically generated

Hình 2- 21. : Biểu đồ trình tự Use Case quản lý vai trò người dùng

Diagram

Description automatically generated

Hình 2- 22. Biểu đồ trình tự Use Case quản lý người dùng

**2.6. Thiết kế cơ sở dữ liệu**

**2.6.1. Bảng users**

Bảng 2. 1. Bảng người dùng

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| UserID | INT | Khóa chính ID nhân viên/người dùng |
| EmployeeCode | Varchar(50) | Mã nhân viên |
| FullName | Varchar(50) | Họ và tên |
| Email | Varchar(50) | Email |
| PhoneNumber | Varchar(50) | Số điện thoại |
| GenderID | INT | ID Giới tính |
| GenderName | Varchar(50) | Giới tính |
| BirthDay | DATE | Ngày sinh |
| IdentityNumber | Varchar(50) | Số CMT/CCCD |
| IdentityDate | DATE | Ngày cấp CMT/CCCD |
| IdentityPlace | TEXT | Nơi cấp CMT/CCCD |
| EthicID | INT | ID Dân tộc |
| EthicName | Varchar(50) | Dân tộc |
| ReligionID | INT | ID Tôn giáo |
| ReligionName | Varchar(50) | Tôn giáo |
| IsUser | TINYINT | Có là người dùng không |
| UserEmail | Varchar(50) | Email đăng nhập |
| PassWord | Varchar(50) | Mật khẩu đăng nhập |
| EmployeeStatusID | INT | ID Trạng thái nhân viên |
| EmployeeStatusName | Varchar(50) | Trạng thái nhân viên |
| UserStatusID | INT | ID Trạng thái tài khoản |
| UserStatusName | Varchar(50) | Trạng thái tài khoản |
| OrganizationUnitID | INT | ID Cơ cấu tổ chức |
| OrganizationUnitName | Varchar(50) | Tên cơ cấu tổ chức |
| JobPositionID | INT | ID Vị trí công việc |
| JobPositionName | Varchar(50) | Tên vị trí công việc |
| EmployeeNatureID | INT | ID Tính chất lao động |
| EmployeeNatureName | Varchar(50) | Tính chất lao động |
| HireDate | DATE | Ngày thử việc |
| ReceiveDate | DATE | Ngày chính thức |
| OutDate | DATE | Ngày nghỉ việc |
| BasicSalary | DOUBLE | Lương cơ bản |
| InsuranceSalary | DOUBLE | Lương đóng bảo hiểm |
| CurrentAddress | TEXT | Địa chỉ hiện tại |
| NativeAddress | TEXT | Địa chỉ thường trú |
| IsDeleted | TINYINT | Có phải đã xó không |
| CreatedDate | DATE | Thời gian tạo |
| CreatedBy | Varchar(50) | Người tạo |
| ModifiedDate | DATE | Thời gian sửa |
| ModifiedBy | Varchar(50) | Người sửa |

**2.6.2. Bảng contracts**

Bảng 2. 2. Bảng hợp đồng

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| ContractID | INT | Khóa chính |
| ContractNo | Varchar(50) | Số hợp đồng |
| ContractName | Varchar(50) | Tên hợp đồng |
| UserID | INT | ID Nhân viên |
| FullName | Varchar(50) | Tên Nhân viên |
| JobPositionID | INT | Vị trí công việc |
| JobPositionName | Varchar(50) | Vị trí công việc |
| ContractUnitID | INT | Đơn vị ký hợp đồng |
| ContractUnitName | Varchar(50) | Đơn vị ký hợp đồng |
| ContractPeriodID | INT | ID Thời hạn hợp đồng |
| ContractPeriodName | Varchar(50) | Thời hạn hợp đồng |
| ContractTypeID | INT | ID Loại hợp đồng |
| ContractTypeName | Varchar(50) | Loại hợp đồng |
| FromDate | DATE | Từ ngày |
| ToDate | DATE | Đến ngày |
| BasicSalary | Double | Lương cơ bản |
| InsuranceSalary | Double | Lương đóng bảo hiểm |
| IsDeleted | TINYINT | Đã xóa |
| CreatedDate | DATE | Thời gian tạo |
| CreatedBy | Varchar(50) | Người tạo |
| ModifiedDate | DATE | Thời gian sửa |
| ModifiedBy | Varchar(50) | Người sửa |

**2.6.3. Bảng organizationunits**

Bảng 2. 3. Bảng cơ cấu tổ chức

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| OrganizationUnitID | INT | Khóa chính |
| OrganizationUnitName | Varchar(50) | Tên cơ cấu tổ chức |
| OrgCode | TEXT | Mã cơ cấu tổ chức |
| ParentID | INT | ID cơ cấu cha |
| Address | TEXT | Địa chỉ |
| Description | TEXT | Mô tả |

**2.6.4. Bảng jobpositions**

Bảng 2. 4. Bảng vị trí công việc

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| JobpositionID | INT | Khóa chính |
| JobpositionName | Varchar(50) | Tên vị trí công việc |
| OrganizationUnitID | INT | ID Cơ cấu tổ chức |
| OrganizationUnitName | Varchar(50) | Cơ cấu tổ chức |
| Description | TEXT | Mô tả |

**2.6.5. Bảng roles**

Bảng 2. 5. Bảng vai trò

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| RoleID | INT | Khóa chính |
| RoleName | Varchar(50) | Tên vai trò |
| PermissionDetail | LONG TEXT | Danh sách quyền |
| Description | TEXT | Mô tả |

**2.6.6. Bảng user\_role**

Bảng 2. 6. Bảng vai trò người dùng

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| UserRoleID | INT | Khóa chính |
| UserID | INT | ID Người dùng |
| RoleID | INT | ID Vai trò |

**2.6.7. Mô hình thiết kế vật lý**

Diagram

Description automatically generated

Hình 23. Mô hình thiết kế vật lý

**2.7. Thiết kế giao diện**

**2.7.1. Giao diện đăng nhập**

Graphical user interface

Description automatically generated with medium confidence

Hình 2- 23. Giao diện Đăng nhập

**2.7.2. Giao diện Xem Tổng quan**

Chart, bar chart

Description automatically generated

Hình 2- 24. Giao diện Tổng quan

**2.7.3. Giao diện Quản lý nhân viên**

A picture containing table

Description automatically generated

Hình 2- 25. Giao diện Danh sách nhân viên

**2.7.4. Giao diện Quản lý hợp đồng**

Table

Description automatically generated

Hình 2- 26. Giao diện Danh sách hợp đồng

**2.7.5. Giao diện Quản lý vai trò người dùng**

Graphical user interface, table

Description automatically generated

Hình 2- 27. Giao diện Danh sách vai trò người dùng

**2.7.6. Giao diện Quản lý Người dùng**

Table

Description automatically generated

Hình 2- 28. Giao diện Danh sách người dùng

**2.7.7. Giao diện Quản lý cơ cấu tổ chức**

Table

Description automatically generated

Hình 2- 29. Giao diện Danh sách cơ cấu tổ chức

**2.7.8. Giao diện Quản lý vị trí công việc**

Graphical user interface, text, application, email

Description automatically generated

Hình 2- 30. Giao diện Danh sách vị trí công việc

**CHƯƠNG 3: THỬ NGHIỆM**

**3.1. Cài đặt hệ thống**

**3.1.1. Cài đặt giao diện**

Để chạy được ứng dụng Angular ta cần phải cài đặt NodeJS theo các bước sau:

- Bước 1: truy cập địa chỉ <https://nodejs.org/en/download/> để tải xuống bộ cài

- Bước 2: sau khi tải xuống thành công, thực hiện chạy bộ cài

Graphical user interface, text, application

Description automatically generated

Hình 3- 1. Chọn next để tiếp tục cài đặt

Graphical user interface, text, application, email

Description automatically generated

Hình 3- 2.Chọn next để tiếp tục cài đặt

Graphical user interface, text, application, email

Description automatically generated

Hình 3- 3.Chọn next để tiếp tục cài đặt

Graphical user interface, text, application

Description automatically generated

Hình 3- 4.Cài đặt hoàn tất, chọn Finish để kết thúc

- Bước 3: Kiểm tra NodeJS đã được cài thành công vào hệ thống:

Text

Description automatically generated

Hình 3- 5. Kiểm tra NodeJS đã được cài thành công vào hệ thống

- Bước 4: mở project bằng visual studio

- Bước 5: sử dụng lệnh “npm insall” để cài đặt các gói cần thiết

- Bước 6: sử dụng lệnh “npm run start” để chạy project, mở trình duyệt lên và truy cập địa chỉ localhost:4200 để mở ứng dụng

**3.1.2. Cài đặt dịch vụ**

Để chạy một ứng dụng ASP.NET Core thì ta cần sử dụng IDE Visual Studio:

- Bước 1: Truy cập địa chỉ <https://visualstudio.microsoft.com/downloads/> , chọn phiên bản phù hợp để thực hiện tải xuống bộ cài

Graphical user interface, application

Description automatically generated

Hình 3- 6.Downloads

- Bước 2: Sau khi tải bộ cài thành công nhấp đúp chuột để mở bộ cài, quá trình cài đặt sẽ diễn ra tự động

- Bước 3: Sau khi cài đặt thành công, mở ứng dụng Visual Studio Installer, tại danh sách hiển thị, tích chọn ASP.NET như hình

Graphical user interface, application

Description automatically generated

Hình 3- 7. Danh sách hiển thị

- Bước 4: Lưu lại và đợi quá trình cài đặt hoàn tất

- Bước 5: Mở project bằng Visual studio và nhấn tổ hợp phím ctrl + F5 để chạy.

**3.2 Thử nghiệm**

**3.2.1. Chức năng đăng nhập**

Graphical user interface, application

Description automatically generated

Hình 3- 8.Màn hình đăng nhập

**3.2.2. Phân hệ tổng quan**

Xem số lượng nhân sự biến động theo tháng/quý/năm

Chart, bar chart

Description automatically generated

Hình 3- 9.Màn hình tổng quan

**3.2.3. Phân hệ nhân viên**

Chức năng:

- Xem danh sách nhân viên

- Tìm kiếm nhân viên

- Thêm nhân viên

- Sửa nhân viên

- Xoá nhân viên

- Xuất khẩu

A screenshot of a computer

Description automatically generated

Hình 3- 10.Màn hình danh sách nhân viên

Graphical user interface, text, email

Description automatically generated

Hình 3- 11.Form thêm/sửa thông tin nhân viên

**3.2.4. Phân hệ hợp đồng**

Chức năng:

- Xem danh sách hợp đồng

- Tìm kiếm hợp đồng

- Thêm hợp đồng

- Sửa hợp đồng

- Xoá hợp đồng

- Xuất khẩu

Graphical user interface, application, email

Description automatically generated

Hình 3- 12.Màn hình danh sách hợp đồng

Table

Description automatically generated

Hình 3- 13.Form thêm/sửa hợp đồng

**3.2.5. Phân hệ thiết lập**

**3.2.5.4. Thiết lập Người dùng**

A screenshot of a computer

Description automatically generated

Hình 3- 14.Màn hình thiết lập Người dùng

**3.2.5.3.Thiết lập vai trò**

Graphical user interface, text, application, email

Description automatically generated

Hình 3- 15.Màn hình thiết lập Vai trò

**3.2.5.1. Thiết lập vị trí công việc**

Chức năng:

- Xem danh sách vị trí công việc

- Tìm kiếm vị trí công việc

- Thêm vị trí công việc

- Sửa vị trí công việc

- Xoá vị trí công việc

- Xuất khẩu

Graphical user interface, text, application, email

Description automatically generated

Hình 3- 16.Màn hình danh sách vị trí công việc

Graphical user interface, text, application, email

Description automatically generated

Hình 3- 17.Form thêm/sửa vị trí công việc

**3.2.5.2. Thiết lập cơ cấu tổ chức**

- Xem danh sách cơ cấu tổ chức theo cây cơ cấu

- Tìm kiếm cơ cấu tổ chức cha, con

- Thêm cơ cấu tổ chức cha, con

- Sửa cơ cấu tổ chức

- Xoá cơ cấu tổ chức

Graphical user interface, text, application

Description automatically generated

Hình 3- 18.Màn hình danh sách cơ cấu tổ chức

**KẾT LUẬN**

Kết quả đạt được:

- Áp dụng các công nghệ: Angular, ASP.NET Core, MySQL để xây dựng một ứng dụng web

- Xây dựng được một ứng dụng quản lý nhân sự có các chức năng:

+ Xem, thêm, sửa, xoá nhân viên

+ Xem dữ liệu tổng quan

+ Thiết lập người dùng, vai trò, cơ cấu tổ chức, vị trí công việc

Hạn chế:

- Do hạn chế về mặt thời gian nên nghiệp vụ ứng dụng còn hạn chế

- Ứng dụng chưa đáp ứng được hết yêu cầu của các tổ chức/doanh nghiệp lớn

Hướng phát triển

- Tích hợp với các công nghệ: AI, Big Data, Blockchain

- Mở rộng nghiệp vụ ứng dụng để đáp ứng được yêu cầu của các tổ chức/doanh nghiệp lớn, cụ thể là bổ sung thêm các phân hệ:

+ Báo cáo

+ Bổ nhiệm

+ Miễn nhiệm

+ Thuyên chuyển

+ Khen thưởng

+ Sự cố

+ Phúc lợi

**TÀI LIỆU THAM KHẢO**

* Sách, giáo trình chính:

[1]. Phát triển phần mềm mã nguồn mở, Vũ Thanh Nguyên, Nguyễn Công Hoan, Phan Trung Hiếu, Lê Đình Tuấn, ĐHQG HCM 2017

[2]. Vũ Thị Dương, Phùng Đức Hòa, Nguyễn Thị Hương Lan, Giáo trình Phân tích thiết kế hướng đối tượng, NXB Khoa học và Kỹ thuật, 2015.

[3]. Nguyễn Thị Thanh Huyền, Ngô Thị Bích Thúy, Giáo trình Cơ sở dữ liệu, NXB Giáo dục VN, 2011.

- Sách, tài liệu tham khảo:

[1]. Bộ slide bài giảng phần mềm mã nguồn mở (Bộ môn CNPM – Khoa CNTT)

[2]. Peter Eeles, Kelli Houston and Wojtek Kozaczynski, Building J2EE Applications with the Rational Unified Process, Pearson Education Corporate Sales Division, 2002.

[3]. Graeme Simsion, Graham Witt, Data Modeling Essentials, Morgan Kaufmann Publishers, 2005.

[4]. Alan Dennis, Barbara Haley Wixom, Roberta M. Roth, Systems Analysis Design 5th Edition, John Wiley & Sons Inc, 2009.