

TRƯỜNG CAO ĐẲNG THỰC HÀNH FPT POLYTECHNIC QUY NHƠN

----- □ □ □ -----



FPT POLYTECHNIC HỆ THỐNG QUẢN LÝ NHÂN VIÊN

NGÀNH PHÁT TRIỂN PHẦN MỀM

Giảng viên hướng dẫn:

Phạm Tiến Huy

Nhóm sinh viên thực hiện:

- Nguyễn Thị Bảo Ly PY00011
- Trương Minh Tiến PY00023

Lớp: SD19301

QUY NHƠN 2024

FPT POLYTECHNIC

1 Giới thiệu dự án	4
1.1 Giới thiệu văn phòng abc	4
1.2 Yêu cầu hệ thống	4
1.2.1 Mô hình triển khai	4
1.2.2 Yêu cầu phần cứng	5
1.2.3 Yêu cầu công nghệ	5
1.3 Yêu cầu nghiệp vụ	5
1.3.1 Quản lý thông tin nhân viên	5
1.3.2 Quản lý ứng viên	5
1.3.3 Quản lý loại hợp đồng	6
1.3.4 Quản lý phòng ban	6
1.3.5 Quản lý chức vụ	6
1.3.6 Thống kê	7
1.4 Use case	7
2 Thiết kế	8
2.1 Thiết kế CSDL	8
2.1.1 Xác định thực thể, thuộc tính	8
2.1.2 Thiết lập ERD	9
2.1.3 Biến đổi ERD sang lược đồ quan hệ	10
2.1.4 Tạo các bảng	12
2.2 Thiết kế giao diện	15
2.2.1 Sơ đồ tổ chức giao diện	15
2.2.2 Thiết kế giao diện cho các chức năng nghiệp vụ	17
3 Thực hiện dự án	27
3.1 Tạo CSDL với SQL Server	27
3.1.1 Sơ đồ quan hệ	27
3.1.2 Chi tiết các bảng	28
3.1.3 Các thủ tục lưu tổng hợp thống kê	32
3.2 Lập trình CSDL	32
3.2.1 JDBC	32
3.2.2 Entity class	32
3.2.3 DAO Class	41
3.3 Utils	65
3.3.1 Lớp tiện ích XDate	65
3.3.2 Lớp tiện ích Y	66
3.4 Lập trình nghiệp vụ	66
3.4.1 Xử lý Form X	66
3.4.2 Xử lý Form Y	67
4 Kiểm thử	67
4.1 Kiểm thử form X	67

4.2 Kiểm thử form Y	67
5 Đóng gói và triển khai	67
5.1 Hướng dẫn chuyển đổi jar thành exe	67
5.2 Hướng dẫn cài đặt triển khai	67
5.3 Hướng dẫn sử dụng phần mềm	67

BẢNG PHÂN CÔNG

CÔNG VIỆC THỰC HIỆN	SINH VIÊN THỰC HIỆN	ĐỘ HOÀN THIỆN
1. Phân tích yêu cầu khách hàng		
Chọn đề tài	Cả nhóm	100%
Phân tích đề tài	Cả nhóm	100%
Vẽ sơ đồ ERD	Ly	100%
Vẽ sơ đồ Use Case	Ly	100%
Tạo CSDL	Tiến	100%
Tìm dữ liệu cho CSDL	Ly	100%
Thiết kế Logo	Tiến	100%
Thiết kế giao diện	Cả nhóm	100%
2. Thực hiện dự án		
Lập trình chức năng login	Ly	100%
Lập trình chức năng đổi mật khẩu	Tiến	100%
Trang chủ chung panel	Cả nhóm	100%
Lập trình chức năng quản lý thông tin nhân viên	Ly	100%
Lập trình chức năng quản lý ứng viên	Tiến	100%
Lập trình chức năng quản lý loại hợp đồng	Ly	100%
Lập trình chức năng quản lý phòng ban	Tiến	100%
Lập trình chức năng quản lý chức vụ	Ly	100%
Lập trình chức năng quản lý tài khoản	Tiến	100%
Lập trình chức năng thống kê	Ly	100%
3. Báo cáo và đóng gói		
Viết test case	Ly	
Viết báo cáo	Cả nhóm	
Đóng gói sản phẩm	Tiến	

1 GIỚI THIỆU DỰ ÁN

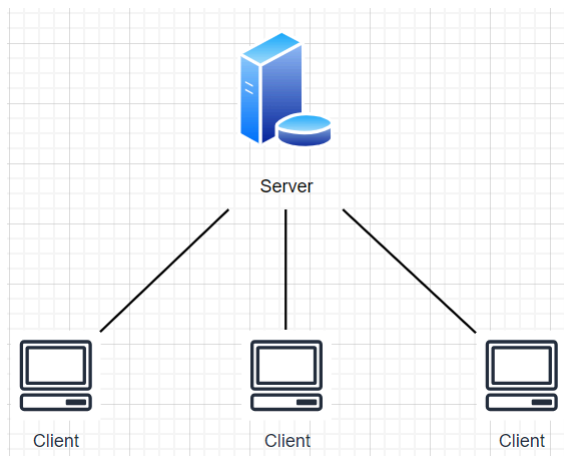
1.1 GIỚI THIỆU VĂN PHÒNG ABC

Với hơn năm năm kinh nghiệm, ABC đã xây dựng được danh tiếng vững chắc trong ngành và có đội ngũ nhân viên to lớn. Văn phòng luôn chú trọng đến việc phát triển nguồn nhân lực và không ngừng cải tiến các quy trình quản lý nhân sự để nâng cao hiệu quả hoạt động.

1.2 YÊU CẦU HỆ THỐNG

- **Hiệu suất:** Đảm bảo hệ thống hoạt động mượt mà và đáp ứng nhanh.
- **Khả năng mở rộng:** Hệ thống phải có khả năng mở rộng để đáp ứng nhu cầu phát triển của công ty.
- **Bảo mật:** Đảm bảo an toàn và bảo mật cho dữ liệu nhân sự.
- **Trải nghiệm người dùng:** Giao diện thân thiện và dễ sử dụng.

1.2.1 Mô hình triển khai



Mô hình khách chủ.

1.2.2 Yêu cầu phần cứng

- Cấu hình máy:
 - + Sử dụng bộ xử lý có hiệu suất cao.
 - + RAM từ 8GB trở lên.
- Hệ điều hành: Các hệ điều hành phổ biến.

1.2.3 Yêu cầu công nghệ

- Ứng dụng phải được thực hiện với công nghệ Swing và JDBC chạy trên mọi hệ điều hành với môi trường JDK tối thiểu 17.
- Hệ quản trị CSDL SQL Server 2019 trở lên.

1.3 YÊU CẦU NGHIỆP VỤ

1.3.1 Quản lý thông tin nhân viên

✓ MÔ TẢ CHỨC NĂNG:

Chức năng quản lý nhân viên được sử dụng để quản lý thông tin nhân viên. Yêu cầu của chức năng này là liệt kê danh sách nhân viên, xem thông tin chi tiết của mỗi nhân viên, thêm nhân viên mới, cập nhật thông tin hoặc xóa nhân viên.

✓ **DỮ LIỆU LIÊN QUAN:**

Thông tin của mỗi nhân viên gồm: Mã nhân viên, Họ tên, Ngày sinh, Giới tính, Địa chỉ, Số điện thoại, Email, Phòng ban, Chức vụ (gồm trưởng phòng và nhân viên bình thường).

✓ **ĐỐI TƯỢNG SỬ DỤNG:**

Trưởng phòng và nhân viên đều có thể sử dụng.

1.3.2 Quản lý ứng viên

✓ **MÔ TẢ CHỨC NĂNG:**

Chức năng quản lý ứng viên được sử dụng để quản lý thông tin ứng viên. Yêu cầu của chức năng này là liệt kê danh sách ứng viên, xem thông tin chi tiết của mỗi ứng viên, thêm ứng viên mới hoặc xóa ứng viên.

✓ **DỮ LIỆU LIÊN QUAN:**

Thông tin ứng viên: Bao gồm họ tên, ngày sinh, giới tính, địa chỉ, số điện thoại, email, vị trí ứng tuyển và các tài liệu liên quan khác.

✓ **ĐỐI TƯỢNG SỬ DỤNG:**

Trưởng phòng và nhân viên đều có thể sử dụng.

1.3.3 Quản lý loại hợp đồng

✓ **MÔ TẢ CHỨC NĂNG:**

Chức năng quản lý loại hợp đồng được sử dụng để quản lý các loại hợp đồng lao động. Yêu cầu của chức năng này là liệt kê danh sách các loại hợp đồng, xem thông tin chi tiết của mỗi loại hợp đồng.

✓ **DỮ LIỆU LIÊN QUAN:**

Thông tin của thân nhân gồm: Mã loại hợp đồng, Tên loại hợp đồng, Thời hạn hợp đồng.

✓ **ĐỐI TƯỢNG SỬ DỤNG:**

Trưởng phòng và nhân viên đều có thể sử dụng.

1.3.4 Quản lý phòng ban

✓ **MÔ TẢ CHỨC NĂNG:**

Quản lý phòng ban được sử dụng để quản lý thông tin về các phòng ban trong công ty. Yêu cầu của chức năng này là liệt kê danh sách phòng ban, xem thông tin chi tiết của mỗi phòng ban, thêm phòng ban mới, cập nhật thông tin hoặc xóa phòng ban.

✓ **DỮ LIỆU LIÊN QUAN:**

Thông tin của phòng ban gồm: Mã phòng ban, Tên phòng ban, Mô tả chức năng, Mã nhân viên trưởng phòng.

✓ **ĐỐI TƯỢNG SỬ DỤNG:**

Trưởng phòng và nhân viên đều có thể sử dụng.

1.3.5 Quản lý chức vụ

✓ **MÔ TẢ CHỨC NĂNG:**

Quản lý chức vụ được sử dụng để quản lý thông tin về các chức vụ có trong công ty. Yêu cầu của chức năng này là liệt kê danh sách chức vụ, xem thông tin chi tiết của mỗi chức vụ, thêm chức vụ mới, cập nhật thông tin hoặc xóa chức vụ.

✓ **DỮ LIỆU LIÊN QUAN:**

Thông tin của phòng ban gồm: Mã chức vụ, Tên chức vụ, Mô tả chức vụ.

✓ **ĐỐI TƯỢNG SỬ DỤNG:**

Trưởng phòng và nhân viên đều có thể sử dụng.

1.3.6 Thống kê

✓ **MÔ TẢ CHỨC NĂNG:**

Chức năng này cung cấp các báo cáo thống kê số lượng ứng viên cho từng vị trí, số lượng nhân viên theo từng loại hợp đồng, số lượng nhân viên của từng phòng ban, lương, độ tuổi. Thống kê bảng lương.

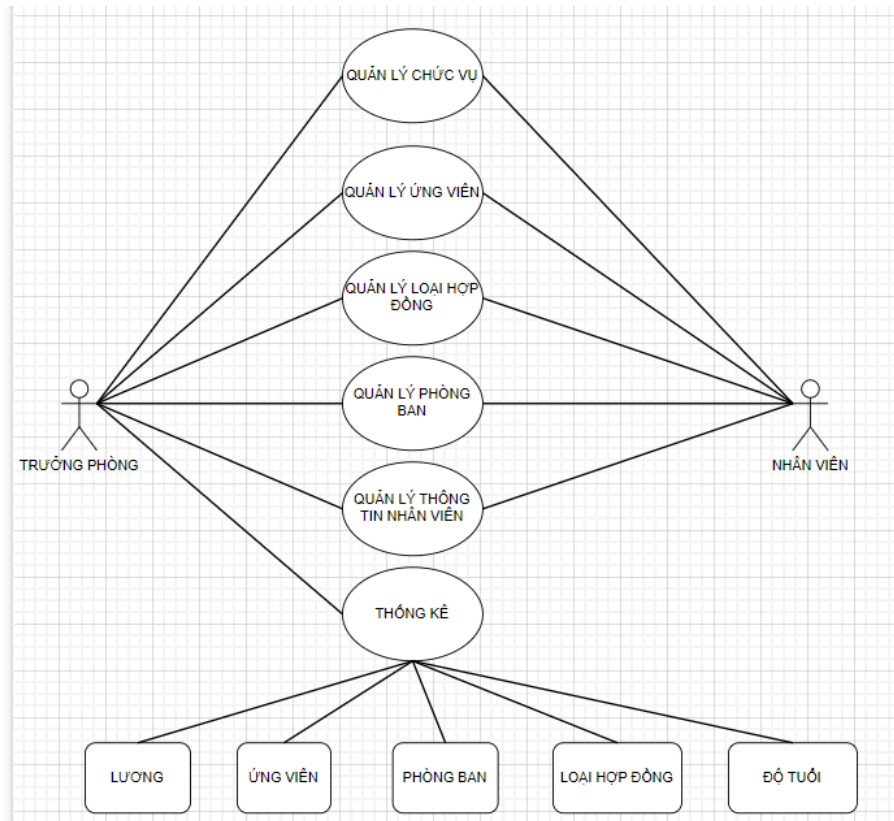
✓ **DỮ LIỆU LIÊN QUAN:**

Danh sách ứng viên, danh sách phòng ban, danh sách nhân viên, danh sách loại hợp đồng, danh sách chức vụ.

✓ **ĐỐI TƯỢNG SỬ DỤNG:**

Chỉ có trưởng phòng mới có thể sử dụng.

1.4 USE CASE



2 THIẾT KẾ

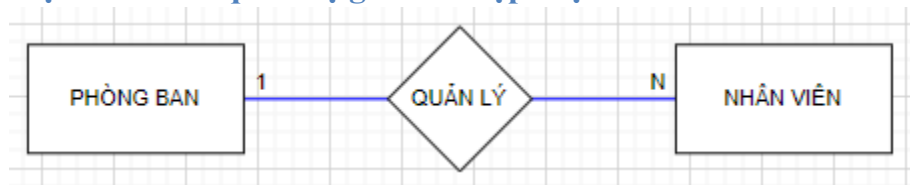
2.1 THIẾT KẾ CSDL

2.1.1 Xác định thực thể, thuộc tính

- Tập thực thể: **Phong_Ban**
Thuộc tính: maPhongBan, TenPhongBan, MoTa.
Thuộc tính khoá: **maPhongBan**
- Tập thực thể: **Chuc_Vu**
Thuộc tính: MaChucVu, TenChucVu, MoTa
Thuộc tính khoá: **MaChucVu**
- Tập thực thể: **Nhan_Vien**
Thuộc tính: maNhanVien, HoTen, GioiTinh, DiaChi, SDT, Email, NgaySinh, Luong, NgayBatDau, ngayKetThuc
Thuộc tính khoá: **maNhanVien**
- Tập thực thể: **Ung_Vien**
Thuộc tính: MaUngVien, HoTen, SDT, DiaChi, NgaySinh, GioiTinh, Email, MoTa.
Thuộc tính khoá: **MaUngVien**
- Tập thực thể: **Loai_Hop_Dong**
Thuộc tính: MaLoaiHopDong, ThoiHan, TenHopDong, MoTa
Thuộc tính khoá: **MaHopDong**
- Tập thực thể: **Tai_Khoan**
Thuộc tính: TenDangNhap, MatKhau
Thuộc tính khoá: **TenDangNhap**

2.1.2 Thiết lập ERD

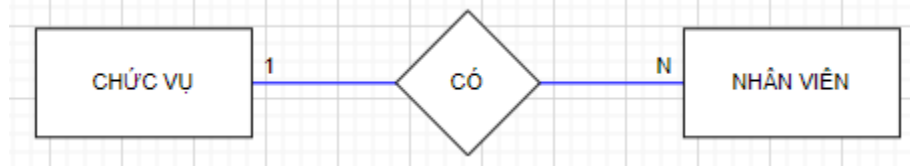
2.1.2.1 Xác định các mối quan hệ giữa các tập thực thể



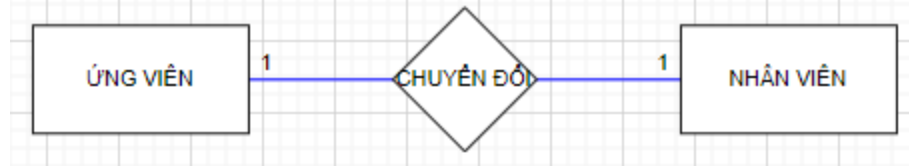
Một phòng ban có nhiều nhân viên, nhưng một nhân viên chỉ có một phòng ban.



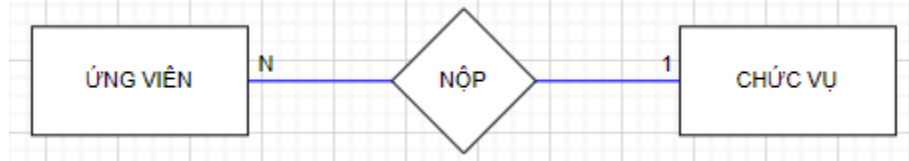
Một phòng ban có một trưởng phòng, và một nhân viên chỉ là trưởng phòng của một phòng ban.



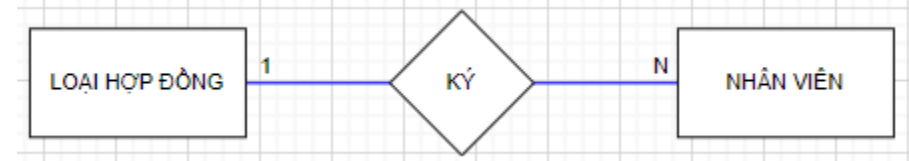
Một chức vụ có nhiều nhân viên, nhưng một nhân viên chỉ có một chức vụ.



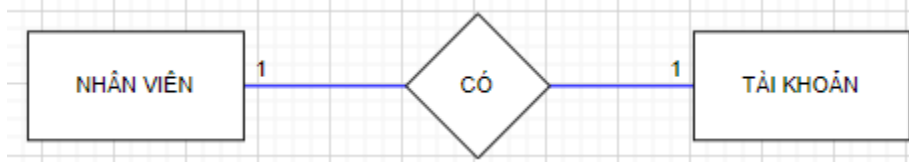
Một ứng viên trở thành một nhân viên, và một nhân viên chỉ là một ứng viên.



Một chức vụ có nhiều ứng viên nộp hồ sơ, nhưng một ứng viên chỉ nộp một chức vụ.

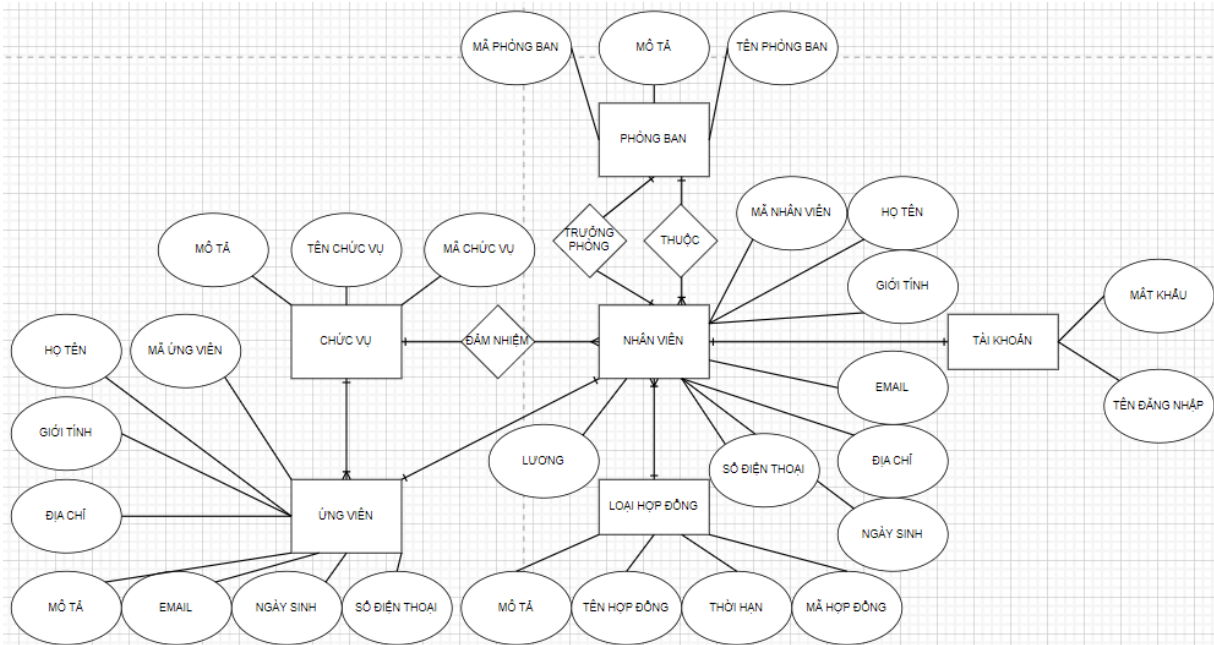


Một loại hợp đồng có thể có nhiều nhân viên ký, nhưng một nhân viên chỉ ký một loại hợp đồng.



Một nhân viên nhân sự chỉ có một tài khoản, và một tài khoản chỉ có thể là một nhân viên nhân sự.

2.1.2.2 Thiết lập ERD



2.1.3 Biến đổi ERD sang lược đồ quan hệ

PHONG_BAN (maPhongBan, tenPhongBan, moTa, **maTruongPhong**)

NHAN_VIEN (maNhanVien, hoTen, GioiTinh, DiaChi, SDT, Email, NgaySinh, Luong, NgayBatDau, ngayKetThuc, **maPhongBan**)

Quan hệ giữa **PHONG_BAN** và **NHAN_VIEN** là quan hệ 1-N không có thuộc tính. Để chuyển đổi sang lược đồ quan hệ thì thêm vào **NHAN_VIEN** thuộc tính khóa của bên **PHONG_BAN** làm khóa ngoại.

Quan hệ trường phòng của **NHAN_VIEN** và **PHONG_BAN** là quan hệ 1-1 không có thuộc tính. Để chuyển đổi sang lược đồ quan hệ thì thêm vào **PHONG_BAN** thuộc tính khóa của bên **NHAN_VIEN** làm khóa ngoại.

CHUC_VU (maChucVu, tenChucVu, moTa)

NHAN_VIEN (maNhanVien, hoTen, GioiTinh, DiaChi, SDT, Email, NgaySinh, Luong, NgayBatDau, ngayKetThuc, **maPhongBan**, **maChucVu**)

Quan hệ giữa **CHUC_VU** và **NHAN_VIEN** là quan hệ 1-N không có thuộc tính. Để chuyển đổi sang lược đồ quan hệ thì thêm vào **NHAN_VIEN** thuộc tính khóa của bên **CHUC_VU** làm khóa ngoại.

UNG_VIEN (maUngVien, HoTen, SDT, DiaChi, NgaySinh, GioiTinh, Email, MoTa)

NHAN_VIEN (maNhanVien, hoTen, GioiTinh, DiaChi, SDT, Email, NgaySinh, Luong, NgayBatDau, ngayKetThuc, **maPhongBan**, **maChucVu**, **maUngVien**)

Quan hệ giữa **UNG_VIEN** và **NHAN_VIEN** là quan hệ 1-1 không có thuộc tính. Để chuyển đổi sang lược đồ quan hệ thì thêm vào **NHAN_VIEN** thuộc tính khóa của bên **UNG_VIEN** làm khóa ngoại.

UNG_VIEN (maUngVien, HoTen, SDT, DiaChi, NgaySinh, GioiTinh, Email, MoTa, **maChucVu**)

CHUC_VU (maChucVu, tenChucVu, moTa)

Quan hệ giữa **CHUC_VU** và **UNG_VIEN** là quan hệ 1-N không có thuộc tính. Để chuyển đổi sang lược đồ quan hệ thì thêm vào **UNG_VIEN** thuộc tính khóa của bên **CHUC_VU** làm khóa ngoại.

LOAI_HOP_DONG (maLoaiHopDong, ThoiHan, TenHopDong, MoTa)

NHAN_VIEN (maNhanVien, hoTen, GioiTinh, DiaChi, SDT, Email, NgaySinh, Luong, NgayBatDau, ngayKetThuc, **maPhongBan**, **maChucVu**, **maUngVien**, **maLoaiHopDong**)

Quan hệ giữa **LOAI_HOP_DONG** và **NHAN_VIEN** là quan hệ 1-N không có thuộc tính. Để chuyển đổi sang lược đồ quan hệ thì thêm vào **NHAN_VIEN** thuộc tính khóa của bên **LOAI_HOP_DONG** làm khóa ngoại.

TAI_KHOAN (tenDangNhap, matKau, **maNhanVien**)

NHAN_VIEN (maNhanVien, hoTen, GioiTinh, DiaChi, SDT, Email, NgaySinh, Luong, NgayBatDau, ngayKetThuc, **maPhongBan**, **maChucVu**, **maUngVien**, **maLoaiHopDong**)

Quan hệ giữa **TAI_KHOAN** và **NHAN_VIEN** là quan hệ 1-1 không có thuộc tính. Vì **NHAN_VIEN** tham gia bán phần vào quan hệ này, **TAI_KHOAN** tham gia toàn phần vào quan hệ. Nên để chuyển đổi sang lược đồ quan hệ thì thêm vào **TAI_KHOAN** thuộc tính khóa của bên **NHAN_VIEN** làm khóa ngoại.

2.1.4 Kiểm tra chuẩn hóa đến 3NF

Để kiểm tra xem các bảng dữ liệu đã đạt được dạng chuẩn 3NF chưa, ta cần kiểm tra ba yếu tố chính của chuẩn hóa dữ liệu:

Yếu tố đầu tiên (1NF): Mọi giá trị trong bảng phải là nguyên tố, tức là không có giá trị nào có thể chia nhỏ thành các phần nhỏ hơn.

Yếu tố thứ hai (2NF): Các thuộc tính không khoá phải phụ thuộc hàm đầy đủ vào khoá chính. Mọi thuộc tính phải phụ thuộc vào toàn bộ khoá chính. Nếu khoá chính có nhiều hơn một cột, thì mọi thuộc tính phải phụ thuộc vào tất cả các cột của khoá chính, không chỉ một phần.

Yếu tố thứ ba (3NF): Mọi thuộc tính phải phụ thuộc trực tiếp vào khoá chính và không phụ thuộc vào bất kỳ thuộc tính phi khoá nào.

Dựa vào các mô tả trên, dữ liệu sẽ cần được chuẩn hóa để đạt đến dạng chuẩn 3NF.

a) Bảng: PHONG_BAN (maPhongBan, tenPhongBan, moTa, **maTruongPhong**)

Thuộc tính maPhongBan là khoá chính của quan hệ, **maTruongPhong** là khoá ngoại của quan hệ.

- Các thuộc tính trên không có giá trị lặp lại và mỗi ô chỉ chứa một giá trị → **đạt 1NF.**
- Trong quan hệ này không có phụ thuộc hàm bán phần vào khoá chính → **đạt 2NF.**
- Các thuộc tính không khoá phụ thuộc trực tiếp vào khoá à không phụ thuộc bất cầu → **đạt 3NF.**

b) Bảng: CHUC_VU (maChucVu, tenChucVu, moTa)

Thuộc tính maChucVu là khoá của quan hệ.

- Mỗi mã chức vụ sẽ có một tên chức vụ → không có thuộc tính nào chứa bộ đa giá trị → **đạt 1NF.**
- Trong quan hệ này có hai phụ thuộc hàm, và thuộc tính tenChucVu, moTa chỉ phụ thuộc duy nhất vào khoá chính maChucVu → không có phụ thuộc bán phần vào khoá chính → **đạt 2NF.**
- Thuộc tính tenChucVu, moTa được suy ra từ khoá → không phụ thuộc bất cầu → **đạt 3NF.**

c) Bảng: LOAI_HOP_DONG (maLoaiHopDong, ThoiHan, TenHopDong, MoTa)

Thuộc tính maLoaiHopDong là khoá chính của quan hệ.

- Các thuộc tính trên không có giá trị lặp lại và mỗi ô chỉ chứa một giá trị → **đạt 1NF.**
- Trong quan hệ này không có phụ thuộc hàm bán phần vào khoá chính → **đạt 2NF.**
- Các thuộc tính không khoá phụ thuộc trực tiếp vào khoá → không phụ thuộc bất cầu → **đạt 3NF.**

d) Bảng: TAI_KHOAN (tenDangNhap, matKhau, **maNhanVien**)

Thuộc tính tenDangNhap là khoá của quan hệ, **maNhanVien** là khoá ngoại của quan hệ.

- Các thuộc tính trên không có giá trị lặp lại và mỗi ô chỉ chứa một giá trị → **đạt 1NF.**
- Trong quan hệ này không có phụ thuộc hàm bán phần vào khoá chính → **đạt 2NF.**
- Các thuộc tính không khoá phụ thuộc trực tiếp vào khoá → không phụ thuộc bất cầu → **đạt 3NF.**

e) Bảng: UNG_VIEN (maUngVien, HoTen, SDT, DiaChi, NgaySinh, GioiTinh, Email, MoTa, **maChucVu**)

Thuộc tính maUngVien là khoá chính của quan hệ, **maChucVu** là khoá ngoại của quan hệ.

- Các thuộc tính trên không có giá trị lặp lại và mỗi ô chỉ chứa một giá trị → **đạt 1NF.**
- Trong quan hệ này không có phụ thuộc hàm bán phần vào khoá chính → **đạt 2NF.**
- Các thuộc tính không khoá phụ thuộc trực tiếp vào khoá → không phụ thuộc bất cầu → **đạt 3NF.**

f) Bảng: NHAN_VIEN (maNhanVien, hoTen, GioiTinh, DiaChi, SDT, Email, NgaySinh, Luong, NgayBatDau, ngayKetThuc **maPhongBan**, **maChucVu**, **maUngVien**, **maLoaiHopDong**)

Thuộc tính maNhanVien là khoá chính của quan hệ, **maChucVu**, **maPhongBan**, **maUngVien**, **maLoaiHopDong** là khoá ngoại của quan hệ.

- Các thuộc tính trên không có giá trị lặp lại và mỗi ô chỉ chứa một giá trị → **đạt 1NF.**
- Trong quan hệ này không có phụ thuộc hàm bán phần vào khoá chính → **đạt 2NF.**
- Các thuộc tính không khoá phụ thuộc trực tiếp vào khoá → không phụ thuộc bất cầu → **đạt 3NF.**

2.1.5 Tạo các bảng

2.1.5.1 Bảng Phong Ban

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
MaPhongBan	char(5)	PK	Mã phòng ban
TenPhongBan	nvarchar(100)	NOT NULL	Tên của phòng ban
Mota	nvarchar(max)		Mô tả thông tin của phòng ban
MaTruongPhong	char(5)	NOT NULL, FK	Trưởng phòng của phòng ban đó

2.1.5.2 Bảng Chức Vụ

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
MaChucVu	char(5)	PK	Mã của chức vụ
TenChucVu	nvarchar(100)	NOT NULL	Tên của chức vụ
MoTa	nvarchar(max)		Mô tả thông tin của chức vụ

2.1.5.3 Bảng Loại Hợp Đồng

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
maLoaiHopDong	char(5)	PK	Mã loại hợp đồng
thoiHan	int	NOT NULL	Thời hạn của hợp đồng
tenHopDong	nvarchar(50)	NOT NULL	Tên loại hợp đồng
moTa	nvarchar(max)		Mô tả loại hợp đồng

2.1.5.4 Bảng Tài Khoản

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
tenDangNhap	char(5)	PK	Tên đăng nhập
matKhau	nvarchar(50)	NOT NULL	Mật khẩu của tài khoản
maNhanVien	char(5)	FK, NOT NULL	Mã nhân viên của tài khoản.

2.1.5.5 Bảng Nhân Viên

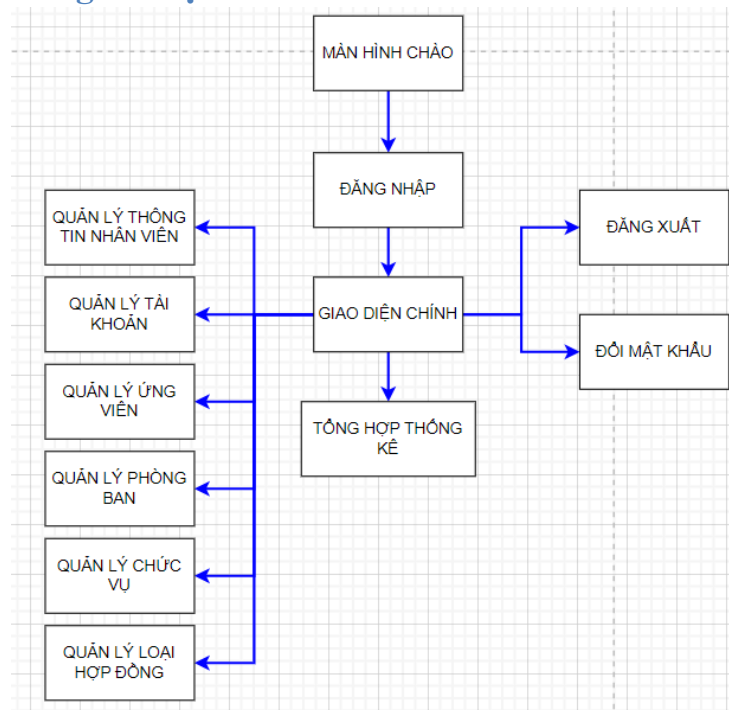
Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
maNhanVien	char(5)	PK	Mã của nhân viên
HoTen	nvarchar(100)	NOT NULL	Họ và tên của nhân viên
GioiTinh	bit default(0)	NOT NULL	Giới tính của nhân viên
DiaChi	nvarchar(max)	NOT NULL	Địa chỉ của nhân viên
SDT	varchar(11)	NOT NULL, UNIQUE	Số điện thoại của nhân viên
Email	varchar(100)	NOT NULL, UNIQUE	Email của nhân viên
ngaySinh	date	NOT NULL, FK	Ngày sinh của nhân viên
MaPhongBan	char(5)	NOT NULL, FK	Mã Phòng Ban
MaChucVu	char(5)	NOT NULL, FK	Mã chức vụ của nhân viên
MaUngVien	char(5)	NOT NULL, UNIQUE	Mã ứng viên của nhân viên
maLoaiHD	char(5)	NOT NULL	Mã loại hợp đồng
Luong	float	NOT NULL	Lương của nhân viên
NgayBatDau	date		Ngày bắt đầu hợp đồng
ngayKetThuc	date		Ngày kết thúc hợp đồng

2.1.5.6 Bảng Ung Viên

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
MaUngVien	char(5)	PK	Mã của ứng viên
HoTen	nvarchar(100)	NOT NULL	Họ và tên của ứng viên
SĐT	varchar(11)	NOT NULL, UNIQUE	Số điện thoại của ứng viên
DiaChi	nvarchar(max)	NOT NULL	Địa chỉ của ứng viên
NgaySinh	date	NOT NULL	Ngày sinh của ứng viên
GioiTinh	bit DEFAULT(0)	NOT NULL	Giới tính của ứng viên
Email	varchar(100)	NOT NULL ,UNIQUE	Email của ứng viên
Mota	nvarchar(max)		Mô tả ứng viên
MaChucVu	char(5)	NOT NULL ,FK	Mã vị trí chức vụ ứng tuyển

2.2 THIẾT KẾ GIAO DIỆN

2.2.1 Sơ đồ tổ chức giao diện

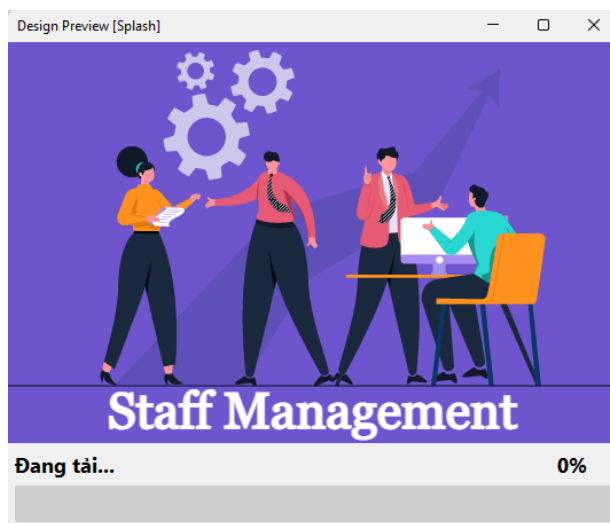


Mô tả sơ đồ:

- Giao diện chính là một cửa sổ chứa menu chính và các nút. Thông qua đó để đi đến các giao diện thành viên để thực hiện các chức năng trong hệ thống.
- Theo yêu cầu thì mọi người dùng phải đăng nhập trước khi sử dụng ứng dụng nên form đăng nhập xuất hiện trước để yêu cầu đăng nhập.
- Ngoài ra mỗi ứng dụng trong thời gian khởi động cần có một màn hình chào cùng với thanh tiến trình để người có cảm giác ứng dụng đang khởi động.

2.2.2 Thiết kế giao diện cho các chức năng nghiệp vụ

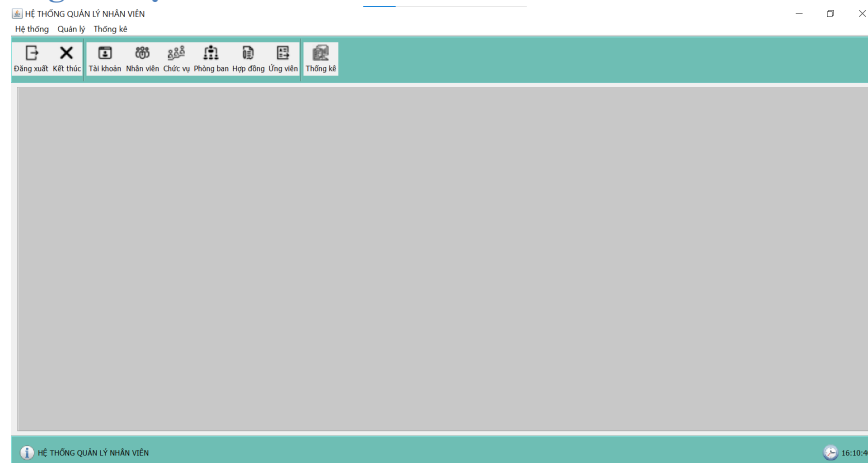
2.2.2.1 Cửa sổ chào



2.2.2.2 Cửa sổ đăng nhập

2.2.2.3 Cửa sổ đổi mật khẩu

2.2.2.4 Cửa sổ giao diện chính



2.2.2.5 Cửa sổ quản lý nhân viên

Quản lý nhân viên

QUẢN LÝ NHÂN VIÊN

THÔNG TIN NHÂN VIÊN **DANH SÁCH NHÂN VIÊN**

Mã nhân viên:

Họ và tên:

Giới tính: ☐ Nam ☒ Nữ Ngày sinh:

Số điện thoại:

Email:

Địa chỉ:

Lương (VND): Mã phòng ban: PB001

Mã ứng viên: Mã chức vụ:

Mã loại hợp đồng: LHD01 Ngày bắt đầu: Ngày kết thúc:

Quản lý nhân viên

QUẢN LÝ NHÂN VIÊN

THÔNG TIN NHÂN VIÊN **DANH SÁCH NHÂN VIÊN**

MÃ NHÂN VIÊN:

Mã nhân viên	Họ và tên	Giới tính	Địa chỉ	SĐT	Email	Ngày sinh	Mã phòng ban	Mã chức vụ	Mã ứng viên	Mã loại hợp...	Lương (VND)
NV001	Nguyễn Văn ...	Nam	234 Đường ...	0985654321	nguyenvana...	10-01-1990	PB001	CV001	UV001	LHD01	20,000,000
NV002	Nguyễn Văn ...	Nam	234 Đường ...	0987654321	nguyenvanw...	15-04-1990	PB002	CV001	UV009	LHD02	22,000,000
NV003	Hồ Thị Ung	Nữ	123 Đường ...	0765432598	hothiu@exa...	25-03-1988	PB002	CV004	UV012	LHD04	5,000,000
NV004	Trần Văn Tâm	Nam	789 Đường ...	0369552147	tranvant@ex...	10-08-1990	PB001	CV003	UV011	LHD02	12,000,000
NV005	Phạm Văn Vi...	Nam	456 Đường ...	0321456987	phamvanv@...	30-10-1993	PB001	CV002	UV013	LHD02	12,000,000
NV006	Nguyễn Thị Sa	Nữ	567 Đường ...	0123456789	nguyenthis@...	20-12-1995	PB001	CV002	UV010	LHD02	12,000,000
NV007	Trần Thị Xuân	Nữ	567 Đường ...	0123556789	tranthix@ex...	20-08-1993	PB003	CV002	UV014	LHD03	9,000,000
NV008	Hồ Văn Tinh	Nam	789 Đường ...	0369852147	hovan.y@ex...	10-12-1988	PB004	CV003	UV015	LHD02	11,000,000
NV009	Hồ Thị Dung	Nữ	123 Đường ...	0765432198	hothid@exa...	30-04-1995	PB005	CV005	UV004	LHD03	10,000,000
NV010	Trần Lê Chi	Nam	32 Ngõ Quỳ...	0857666776	lechi@gmail...	02-02-2002	PB002	CV002	UV002	LHD02	8,000,000
NV011	Nguyễn Thị ...	Nam	ahfdb	0753447644	adh@gmail.c...	02-09-1999	PB001	CV003	UV016	LHD02	10,000,000

2.2.2.6 Cửa sổ quản lý ứng viên

QUẢN LÝ ỨNG VIÊN

THÔNG TIN ỨNG VIÊN **DANH SÁCH ỨNG VIÊN**

Mã ứng viên:

Họ và tên:

Giới tính: ☐ Nam ☐ Nữ Ngày sinh:

Số điện thoại: Email:

Địa chỉ:

Mã chức vụ: CV001 Mô tả:

QUẢN LÝ ỨNG VIÊN

THÔNG TIN ỨNG VIÊN **DANH SÁCH ỨNG VIÊN**

MÃ ỨNG VIÊN:

Mã ứng viên	Họ và tên	Giới tính	Ngày sinh	Số điện thoại	Email	Địa chỉ	Mã chức vụ	Mô tả
UV001	Nguyễn Văn T...	Nam	1990-01-10	0987654321	nguyenvan3a...	234 Đường AB...	CV001	Ứng viên cho v...
UV002	Trần Thị Lam	Nữ	1993-05-20	0123456789	tranb3@exam...	567 Đường XY...	CV002	Ứng viên cho v...
UV003	Lê Văn Ca	Nam	1988-08-15	0369852147	levanc3@exa...	789 Đường UV...	CV004	Ứng viên cho v...
UV004	Hồ Thị Dung	Nữ	1995-04-30	0765432198	hoth3id@exa...	123 Đường D...	CV005	Ứng viên cho v...
UV005	Phạm Văn Nam	Nam	1992-11-10	0321456987	phamv3e@exa...	456 Đường GH...	CV003	Ứng viên cho v...
UV006	Nguyễn Thị Văn	Nữ	1993-07-25	0901234567	ngu3yenthif@...	789 Đường UV...	CV003	Ứng viên cho v...
UV007	Trần Văn Tiến	Nam	1990-02-15	0987653321	tra3nv@exa...	234 Đường AB...	CV001	Ứng viên cho v...
UV008	Lê Thị Hoài	Nữ	1993-06-20	0123436789	leth3lh@exam...	567 Đường XY...	CV002	Ứng viên cho v...
UV009	Nguyễn Văn M...	Nam	1990-04-15	0987694321	nguye3nvamw...	234 Đường AB...	CV001	Ứng viên cho v...
UV010	Nguyễn Thị Sa	Nữ	1990-04-15	0123406789	nguye3enthis@...	567 Đường KL...	CV002	Ứng viên cho v...
UV011	Trần Văn Tâm	Nam	1990-08-10	0369952147	tranv3ant@ex...	789 Đường UV...	CV003	Ứng viên cho v...
UV012	Hồ Thị Ung	Nữ	1988-03-25	0369852107	hot3hiu@exa...	123 Đường XY...	CV004	Ứng viên cho v...
UV013	Phạm Văn Vinh	Nam	1993-10-30	0321456977	pham3vamv@e...	456 Đường D...	CV002	Ứng viên cho v...
UV014	Trần Thị Xuân	Nữ	1993-08-20	0123456799	tran3thix@exa...	567 Đường XY...	CV002	Ứng viên cho v...
UV015	Hồ Văn Tĩnh	Nam	1988-12-10	0368852147	hov3an.y@exa...	789 Đường UV...	CV003	Ứng viên cho v...
UV016	Nguyễn Thị Bả...	Nữ	1999-09-02	0753447644	adh@gmail.com	ahfbd	CV003	Ứng viên cho v...

2.2.2.7 Cửa sổ quản lý loại hợp đồng

QUẢN LÝ HỢP ĐỒNG

Mã loại hợp đồng: LHD01

Tên loại hợp đồng: Hợp đồng vô thời hạn

Thời hạn (Tháng): 240

Ngày bắt đầu:

Mô tả: Hợp đồng không xác định thời hạn

Mã loại hợp đồng	Tên loại hợp đồng	Thời hạn (Tháng)	Ngày bắt đầu	Mô tả
LHD01	Hợp đồng vô thời hạn	240	Hợp đồng không xác định ...	
LHD02	Hợp đồng có thời hạn	12	Hợp đồng xác định thời hạn...	
LHD03	Hợp đồng mùa vụ	6	Hợp đồng thời vụ 6 tháng	
LHD04	Hợp đồng thử việc	2	Hợp đồng thử việc 2 tháng	

2.2.2.8 Cửa sổ quản lý phòng ban

Quản lý phòng ban

QUẢN LÝ PHÒNG BAN

THÔNG TIN PHÒNG BAN | DANH SÁCH PHÒNG BAN

Mã phòng ban
PB001

Tên phòng ban
Phòng Kỹ Thuật

Mã trưởng phòng
NV001

Mô tả
Phòng chuyên về kỹ thuật

Thêm **Sửa**
Xoá **Reset**

<< >>
< >

Quản lý phòng ban

QUẢN LÝ PHÒNG BAN

THÔNG TIN PHÒNG BAN | **DANH SÁCH PHÒNG BAN**

Mã phòng ban **Tìm kiếm**

Mã phòng ban	Tên phòng ban	Tên trưởng phòng	Mô tả
PB001	Phòng Kỹ Thuật	NV001	Phòng chuyên về kỹ thuật
PB002	Phòng Nhân Sự	NV002	Phòng chuyên về nhân sự
PB003	Phòng Kinh Doanh	NV007	Phòng chuyên về kinh doanh
PB004	Phòng Marketing	NV008	Phòng chuyên về marketing
PB005	Phòng IT	NV009	Phòng chuyên về công nghệ ...

2.2.2.9 Cửa sổ quản lý chức vụ

Quản lý chức vụ

QUẢN LÝ CHỨC VỤ

THÔNG TIN CHỨC VỤ | DANH SÁCH CHỨC VỤ

Mã chức vụ
CV001

Tên chức vụ
Trưởng phòng

Mô tả
Chịu trách nhiệm quản lý và điều hành phòng ban

Thêm **Sửa**
Xoá **Reset**

< << >> >



Mã chức vụ	Tên chức vụ	Mô tả
CV001	Trưởng phòng	Chịu trách nhiệm quản lý và điều hành ...
CV002	Nhân viên	Nhân viên làm việc trong phòng ban
CV003	Nhân viên kinh doanh	Thực hiện các công việc kinh doanh the...
CV004	Thực tập sinh	Sinh viên thực tập tại công ty
CV005	Chuyên Viên	Chuyên viên phụ trách một lĩnh vực cụ t...

2.2.2.10 Cửa sổ quản lý tài khoản




Tên đăng nhập	Mật khẩu	Mã nhân viên
us001	111111	NV002
us002	222222	NV003

2.2.2.11 Cửa sổ thống kê

THỐNG KÊ SỐ LIỆU

ỨNG VIÊN

ĐỘ TUỔI

LƯƠNG

HỢP ĐỒNG

PHÒNG BAN

Tên chức vụ

Trưởng phòng

Q Tìm kiếm

Số lượng ứng viên	Số lượng PASS	Số lượng FAIL
5	2	3

THỐNG KÊ SỐ LIỆU

ỨNG VIÊN

ĐỘ TUỔI

LƯƠNG

HỢP ĐỒNG

PHÒNG BAN

Tuổi từ

18

đến

60

Q Tìm kiếm

Số lượng nhân viên
11

THỐNG KÊ SỐ LIỆU

ỨNG VIÊN

ĐỘ TUỔI

LƯƠNG

HỢP ĐỒNG

PHÒNG BAN

Lương từ

10000000

đến

50000000

Q Tìm kiếm

Số lượng nhân viên
6

THỐNG KÊ SỐ LIỆU

ỨNG VIÊN

ĐỘ TUỔI

LƯƠNG

HỢP ĐỒNG

PHÒNG BAN

Loại hợp đồng	Số lượng nhân viên đã ký
Hợp đồng vô thời hạn	1
Hợp đồng có thời hạn	7
Hợp đồng mùa vụ	2
Hợp đồng vô thử việc	1

THỐNG KÊ SỐ LIỆU

ỨNG VIÊN

ĐỘ TUỔI

LƯƠNG

HỢP ĐỒNG

PHÒNG BAN

Tên phòng ban

Phòng Kỹ Thuật

Q Tìm kiếm

Số lượng nhân viên	Số lượng chức vụ	Họ tên trưởng phòng
5	3	Nguyễn Văn Thắng

3 THỰC HIỆN DỰ ÁN

3.1 TẠO CSDL với SQL SERVER

3.1.1 Sơ đồ quan hệ



- Tất cả các relationship của UPDATE đều là CASCADE.
- Tất cả các relationship của DELETE đều là NO ACTION ngoại trừ NhanVien - TaiKhoan là CASCADE.

3.1.2 Chi tiết các bảng

3.1.2.1 Bảng ChucVu

MÃ LỆNH TẠO BẢNG

```
CREATE TABLE Chuc_Vu (
    MaChucVu CHAR(5) PRIMARY KEY,
    TenChucVu NVARCHAR(100) NOT NULL,
    MoTa NVARCHAR(MAX) );
```

CÂU LỆNH SQL CƠ BẢN

```
INSERT INTO Chuc_Vu (MaChucVu, TenChucVu, MoTa) VALUES (?, ?, ?)
UPDATE Chuc_Vu SET TenChucVu = ?, MoTa = ? WHERE MaChucVu = ?
DELETE FROM Chuc_Vu WHERE MaChucVu = ?
SELECT * FROM Chuc_Vu WHERE MaChucVu = ?
```

3.1.2.2 Bảng PhongBan

MÃ LỆNH TẠO BẢNG

```
CREATE TABLE Phong_Ban (
    MaPhongBan CHAR(5) PRIMARY KEY,
    TenPhongBan NVARCHAR(100) NOT NULL,
    MoTa NVARCHAR(MAX),
    MaTruongPhong CHAR(5) NULL);
ALTER TABLE Phong_Ban
ADD CONSTRAINT FK_MaTruongPhong
FOREIGN KEY (MaTruongPhong)
REFERENCES Nhan_Vien(MaNhanVien)
ON DELETE NO ACTION ON UPDATE NO ACTION;
```

CÂU LỆNH SQL CƠ BẢN

```
INSERT INTO Phong_Ban (MaPhongBan, TenPhongBan, MoTa, MaTruongPhong) VALUES (?, ?, ?, ?)
UPDATE Phong_Ban SET TenPhongBan = ?, MoTa = ?, MaTruongPhong = ? WHERE MaPhongBan = ?
DELETE FROM Phong_Ban WHERE MaPhongBan = ?
SELECT * FROM Phong_Ban WHERE MaPhongBan = ?
```

3.1.2.3 Bảng NhanVien

MÃ LỆNH TẠO BẢNG

```
CREATE TABLE Nhan_Vien (
    MaNhanVien CHAR(5) PRIMARY KEY,
    HoTen NVARCHAR(100) NOT NULL,
    GioiTinh BIT DEFAULT(0) NOT NULL,
    DiaChi NVARCHAR(MAX) NOT NULL,
    SDT VARCHAR(11) NOT NULL UNIQUE,
    Email VARCHAR(100) NOT NULL UNIQUE,
    NgaySinh DATE NOT NULL,
    MaPhongBan CHAR(5) NOT NULL,
    MaChucVu CHAR(5) NOT NULL,
    MaUngVien CHAR(5) NOT NULL UNIQUE,
    MaLoaiHd CHAR(5) NOT NULL,
    Luong FLOAT NOT NULL,
    FOREIGN KEY (MaLoaiHd) REFERENCES Loai_Hop_Dong(MaLoaiHd) ON DELETE NO ACTION
ON UPDATE CASCADE,
    FOREIGN KEY (MaPhongBan) REFERENCES Phong_Ban(MaPhongBan) ON DELETE NO ACTION
ON UPDATE CASCADE,
    FOREIGN KEY (MaChucVu) REFERENCES Chuc_Vu(MaChucVu) ON DELETE NO ACTION ON
UPDATE CASCADE);
ALTER TABLE Nhan_Vien
ADD CONSTRAINT FK_Nhan_Vien_MaUngVien
FOREIGN KEY (MaUngVien) REFERENCES Ung_Vien(MaUngVien)
ON DELETE NO ACTION
ON UPDATE NO ACTION;
```

CÂU LỆNH SQL CƠ BẢN

```
INSERT INTO Nhan_Vien (maNhanVien, HoTen, GioiTinh, DiaChi, SDT, Email, NgaySinh, MaPhongBan,
MaChucVu, MaUngVien, MaLoaiHD, Luong) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
UPDATE Nhan_Vien SET HoTen = ?, GioiTinh = ?, DiaChi = ?, SDT = ?, Email = ?, NgaySinh = ?, MaPhongBan = ?,
MaChucVu = ?, MaUngVien = ?, MaLoaiHD = ?, Luong = ? WHERE maNhanVien = ?
DELETE FROM Nhan_Vien WHERE maNhanVien = ?
SELECT * FROM Nhan_Vien WHERE maNhanVien = ?
```

3.1.2.4 Bảng LoaiHopDong

MÃ LỆNH TẠO BẢNG

```
CREATE TABLE Loai_Hop_Dong (
    MaLoaiHd CHAR(5) PRIMARY KEY,
    ThoiHan INT NOT NULL,
    TenHopDong NVARCHAR(50) NOT NULL,
    MoTa NVARCHAR(MAX) );
```

CÂU LỆNH SQL CƠ BẢN

```
INSERT INTO Loai_Hop_dong (MaLoaiHd, ThoiHan, TenHopDong, MoTa) VALUES (?, ?, ?, ?)
UPDATE Loai_Hop_Dong SET TenHopDong = ?, ThoiHan = ?, MoTa = ? WHERE MaLoaiHd = ?
DELETE FROM Loai_Hop_Dong WHERE MaLoaiHd = ?
SELECT * FROM Loai_Hop_Dong WHERE MaLoaiHd = ?
```

3.1.2.5 Bảng TaiKhoan

MÃ LỆNH TẠO BẢNG

```
CREATE TABLE Tai_Khoan (
    TenDangNhap CHAR(5) PRIMARY KEY,
    MatKhau NVARCHAR(50) NOT NULL,
    MaNhanVien CHAR(5) NOT NULL,
    FOREIGN KEY (MaNhanVien) REFERENCES Nhan_Vien(MaNhanVien) ON DELETE CASCADE
ON UPDATE CASCADE );
```

CÂU LỆNH SQL CƠ BẢN

```
INSERT INTO Tai_Khoan (TenDangNhap, MatKhau, MaNhanVien) VALUES (?, ?, ?)
UPDATE Tai_Khoan SET MatKhau = ?, MaNhanVien = ? WHERE TenDangNhap = ?
DELETE FROM Tai_Khoan WHERE TenDangNhap = ?
SELECT * FROM Tai_Khoan WHERE TenDangNhap = ?
```

3.1.2.6 Bảng UngVien

MÃ LỆNH TẠO BẢNG

```
CREATE TABLE Ung_Vien (
    MaUngVien CHAR(5) PRIMARY KEY,
    HoTen NVARCHAR(100) NOT NULL,
    SDT VARCHAR(11) NOT NULL UNIQUE,
    DiaChi NVARCHAR(MAX) NOT NULL,
    NgaySinh DATE NOT NULL,
    GioiTinh BIT DEFAULT(0) NOT NULL,
    Email VARCHAR(100) NOT NULL UNIQUE,
    MoTa NVARCHAR(MAX),
    MaChucVu CHAR(5) NOT NULL,
    FOREIGN KEY (MaChucVu) REFERENCES Chuc_Vu(MaChucVu) ON DELETE NO ACTION ON
    UPDATE CASCADE);
```

CÂU LỆNH SQL CƠ BẢN

```
INSERT INTO Ung_Vien (MaUngVien, HoTen, GioiTinh, DiaChi, SDT, Email, NgaySinh, MaChucVu, MoTa)
VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)
UPDATE Ung_Vien SET HoTen = ?, GioiTinh = ?, DiaChi = ?, SDT = ?, Email = ?, NgaySinh = ?, MaChucVu =
?, MoTa = ? WHERE MaUngVien = ?
DELETE FROM Ung_Vien WHERE MaUngVien = ?
SELECT * FROM Ung_Vien WHERE MaUngVien = ?
```

3.1.3 Các thủ tục lưu tổng hợp thống kê

3.1.3.1 SP_sl_ungvien

Mã nguồn	<pre>CREATE PROC sp_sl_ungvien (@tencv NVARCHAR(100)) AS BEGIN SELECT COUNT(uv.MaUngVien) + COUNT(nv.MaUngVien) AS TongUngVien, COUNT(nv.MaUngVien) AS SLpass, COUNT(uv.MaUngVien) AS SLfail FROM Chuc_Vu cv LEFT JOIN Ung_Vien uv ON cv.MaChucVu = uv.MaChucVu LEFT JOIN Nhan_Vien nv ON uv.MaUngVien = nv.MaUngVien WHERE cv.TenChucVu like @tencv GROUP BY cv.MaChucVu, cv.TenChucVu END</pre>
Mô tả	Proc này được sử dụng để truy vấn số lượng ứng viên theo tên chức vụ
Tham số	@tencv là tên chức vụ cần truy vấn
Kết quả	[TongUngVien, SLpass, SLfail]

3.1.3.2 SP_sl_tuoi

Mã nguồn	<pre> CREATE PROC sp_sl_tuoi (@mintuoi INT, @maxtuoi INT) AS BEGIN SELECT count(*) SLNV FROM nhan_vien WHERE (year(GETDATE())-year(nhan_vien.ngaysinh))>@mintuoi AND (year(GETDATE())-year(nhan_vien.ngaysinh))<@maxtuoi END </pre>
Mô tả	Proc này được sử dụng để truy vấn số lượng nhân viên theo khoảng tuổi.
Tham số	@mintuoi là tuổi thấp nhất, @maxtuoi là tuổi cao nhất.
Kết quả	[SLNV]

3.1.3.3 SP_sl_luong

Mã nguồn	<pre> CREATE PROC sp_sl_luong (@minluong FLOAT, @maxluong FLOAT) AS BEGIN SELECT count(*) SLNV FROM nhan_vien WHERE Nhan_Vien.Luong>@minluong AND Nhan_Vien.Luong<@maxluong END </pre>
Mô tả	Proc này được sử dụng để truy vấn số lượng nhân viên theo khoảng lương.
Tham số	@minluong là lương thấp nhất, @maxluong là lương cao nhất.
Kết quả	[SLNV]

3.1.3.4 SP_sl_hd

Mã nguồn	<pre> CREATE PROC sp_sl_hd AS BEGIN SELECT TenHopDong, count(*) SLNV FROM Loai_Hop_Dong JOIN Nhan_Vien ON Loai_Hop_Dong.MaLoaiHd = Nhan_Vien.MaLoaiHd GROUP BY Nhan_Vien.MaLoaiHd, Loai_Hop_Dong.TenHopDong END </pre>
Mô tả	Proc này được sử dụng để truy vấn số lượng nhân viên theo loại hợp đồng.
Tham số	
Kết quả	[TenHopDong, SLNV]

3.1.3.5 SP_sl_pb

Mã nguồn	<pre> CREATE PROC sp_sl_pb(@tenpb NVARCHAR(100)) AS BEGIN SELECT count(nv.manhanvien) AS SLNV, count(DISTINCT nv.maChucVu) AS SLCV, tp.HoTen AS HOTENTRUONGPHONG FROM Phong_Ban LEFT JOIN Nhan_Vien tp ON Phong_Ban.MaTruongPhong = tp.MaNhanVien LEFT JOIN Nhan_Vien nv ON Phong_Ban.MaPhongBan = nv.MaPhongBan WHERE Phong_Ban.TenPhongBan LIKE @tenpb GROUP BY tp.HoTen END </pre>
Mô tả	Proc này được sử dụng để truy vấn số lượng nhân viên theo tên phòng ban.
Tham số	@tenpb là tên phòng ban cần truy vấn
Kết quả	[SLNV, SLCV, HoTenTruongPhong]

3.2 LẬP TRÌNH CSDL

3.2.1 JDBC

```

package UTILS;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

/**
 *
 * @author Admin
 */
public class TEST_CONNECT {
    public static void main(String[] args) throws ClassNotFoundException, SQLException {
        // TODO code application logic here
        String driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";
        String dbUrl = "jdbc:sqlserver://13.71.106.166:1433;databaseName=SD19301_Team5;encrypt=true;trustServerCertificate=true;";
        String admin = "usr01_team5";
        String pass = "qlnhanvien";
        try {
            Class.forName(Class.forName(driver));
            Connection a = DriverManager.getConnection(dbUrl, admin, pass);
            System.out.print("success");
        } catch (Exception e) {
            System.out.print("fail");
            e.printStackTrace();
        }
    }
}

```

3.2.2 Entity class

3.2.2.1 Class NhanVien

```

package CLASS;
import java.util.Date;
public class NhanVien {
    private String MaNhanVien;
    private String HoTen;
    private Date NgaySinh;
    private String SDT;
    private boolean GioiTinh;
    private String DiaChi;
    private String Email;
    private String MaPhongban;
}

```



```

private String MaChucVu;
private String MaUngVien;
private String MaLoaiHd;
private double Luong = 0;
public NhanVien() {
}
public NhanVien(String MaNhanVien, String HoTen, Date NgaySinh, String SDT, boolean GioiTinh, String DiaChi, String
Email, String MaPhongban, String MaChucVu, String MaUngVien, String MaLoaiHd, double Luong) {
    this.MaNhanVien = MaNhanVien;
    this.HoTen = HoTen;
    this.NgaySinh = NgaySinh;
    this.SDT = SDT;
    this.GioiTinh = GioiTinh;
    this.DiaChi = DiaChi;
    this.Email = Email;
    this.MaPhongban = MaPhongban;
    this.MaChucVu = MaChucVu;
    this.MaUngVien = MaUngVien;
    this.MaLoaiHd = MaLoaiHd;
    this.Luong = Luong;
}
public String getMaNhanVien() {
    return this.MaNhanVien;
}
public void setMaNhanVien(String MaNhanVien) {
    this.MaNhanVien = MaNhanVien;
}
public String getHoTen() {
    return this.HoTen;
}
public void setHoTen(String HoTen) {
    this.HoTen = HoTen;
}
public Date getNgaySinh() {
    return this.NgaySinh;
}
public void setNgaySinh(Date NgaySinh) {
    this.NgaySinh = NgaySinh;
}
public String getSDT() {
    return this.SDT;
}
public void setSDT(String SDT) {
    this.SDT = SDT;
}
public boolean isGioiTinh() {
    return this.GioiTinh;
}
public boolean getGioiTinh() {
    return this.GioiTinh;
}
public void setGioiTinh(boolean GioiTinh) {
    this.GioiTinh = GioiTinh;
}
public String getDiaChi() {
    return this.DiaChi;
}
public void setDiaChi(String DiaChi) {
    this.DiaChi = DiaChi;
}
public String getEmail() {

```

```

        return this.Email;
    }
    public void setEmail(String Email) {
        this.Email = Email;
    }
    public String getMaPhongban() {
        return this.MaPhongban;
    }
    public void setMaPhongban(String MaPhongban) {
        this.MaPhongban = MaPhongban;
    }
    public String getMaChucVu() {
        return this.MaChucVu;
    }
    public void setMaChucVu(String MaChucVu) {
        this.MaChucVu = MaChucVu;
    }
    public String getMaUngVien() {
        return this.MaUngVien;
    }
    public void setMaUngVien(String MaUngVien) {
        this.MaUngVien = MaUngVien;
    }
    public String getMaLoaiHd() {
        return this.MaLoaiHd;
    }
    public void setMaLoaiHd(String MaLoaiHd) {
        this.MaLoaiHd = MaLoaiHd;
    }
    public double getLuong() {
        return this.Luong;
    }

    public void setLuong(double Luong) {
        this.Luong = Luong;
    }
    @Override
    public String toString() {
        return this.MaLoaiHd; // hoặc trả về bất kỳ thông tin nào bạn muốn hiển thị
    }
}

```

3.2.2.2 Class UngVien

```

package CLASS;
import java.util.Date;
public class UngVien {
    private String MaUngVien;
    private String HoTen;
    private String SDT;
    private String DiaChi;
    private Date NgaySinh;
    private boolean GioiTinh;
    private String Email;
    private String MoTa;
    private String MaChucVu;
    public UngVien() {
    }
    public UngVien(String MaUngVien, String HoTen, String SDT, String DiaChi, Date NgaySinh, boolean GioiTinh, String
    Email, String MoTa, String MaChucVu) {
        this.MaUngVien = MaUngVien;
    }
}

```

```
this.HoTen = HoTen;
this.SDT = SDT;
this.DiaChi = DiaChi;
this.NgaySinh = NgaySinh;
this.GioiTinh = GioiTinh;
this.Email = Email;
this.MoTa = MoTa;
this.MaChucVu = MaChucVu;
}
public String getMaUngVien() {
    return MaUngVien;
}
public void setMaUngVien(String MaUngVien) {
    this.MaUngVien = MaUngVien;
}
public String getHoTen() {
    return HoTen;
}
public void setHoTen(String HoTen) {
    this.HoTen = HoTen;
}
public String getSDT() {
    return SDT;
}
public void setSDT(String SDT) {
    this.SDT = SDT;
}
public String getDiaChi() {
    return DiaChi;
}
public void setDiaChi(String DiaChi) {
    this.DiaChi = DiaChi;
}
public Date getNgaySinh() {
    return NgaySinh;
}
public void setNgaySinh(Date NgaySinh) {
    this.NgaySinh = NgaySinh;
}
public boolean isGioiTinh() {
    return GioiTinh;
}
public void setGioiTinh(boolean GioiTinh) {
    this.GioiTinh = GioiTinh;
}
public String getEmail() {
    return Email;
}
public void setEmail(String Email) {
    this.Email = Email;
}
public String getMoTa() {
    return MoTa;
}
public void setMoTa(String MoTa) {
    this.MoTa = MoTa;
}
public String getMaChucVu() {
    return MaChucVu;
}
public void setMaChucVu(String MaChucVu) {
```

```

        this.MaChucVu = MaChucVu;
    }
}

```

3.2.2.3 Class PhongBan

```

package CLASS;
public class PhongBan {
    private String MaPhongBan;
    private String TenPhongBan;
    private String MoTa;
    private String MaTruongPhong;
    public PhongBan() {
    }
    public PhongBan(String MaPhongBan, String TenPhongBan, String MoTa, String MaTruongPhong) {
        this.MaPhongBan = MaPhongBan;
        this.TenPhongBan = TenPhongBan;
        this.MoTa = MoTa;
        this.MaTruongPhong = MaTruongPhong;
    }
    public String getMaPhongBan() {
        return this.MaPhongBan;
    }
    public void setMaPhongBan(String MaPhongBan) {
        this.MaPhongBan = MaPhongBan;
    }
    public String getTenPhongBan() {
        return this.TenPhongBan;
    }
    public void setTenPhongBan(String TenPhongBan) {
        this.TenPhongBan = TenPhongBan;
    }
    public String getMoTa() {
        return this.MoTa;
    }
    public void setMoTa(String MoTa) {
        this.MoTa = MoTa;
    }
    public String getMaTruongPhong() {
        return this.MaTruongPhong;
    }
    public void setMaTruongPhong(String MaTruongPhong) {
        this.MaTruongPhong = MaTruongPhong;
    }
    @Override
    public String toString() {
        return this.MaPhongBan; // hoặc trả về bất kỳ thông tin nào bạn muốn hiển thị
    }
    public String toTenPb(){
        return this.TenPhongBan;
    }
}

```

3.2.2.4 Class ChucVu

```

package CLASS;
public class ChucVu {
    private String MaChucVu;
    private String TenChucVu;
    private String Mota;
    public ChucVu() {
    }
}

```

```

public ChucVu(String MaChucVu, String TenChucVu, String Mota) {
    this.MaChucVu = MaChucVu;
    this.TenChucVu = TenChucVu;
    this.Mota = Mota;
}
public String getMaChucVu() {
    return this.MaChucVu;
}
public void setMaChucVu(String MaChucVu) {
    this.MaChucVu = MaChucVu;
}
public String getTenChucVu() {
    return this.TenChucVu;
}
public void setTenChucVu(String TenChucVu) {
    this.TenChucVu = TenChucVu;
}
public String getMota() {
    return this.Mota;
}
public void setMota(String Mota) {
    this.Mota = Mota;
}
@Override
public String toString() {
    return this.MaChucVu; // hoặc trả về bất kỳ thông tin nào bạn muốn hiển thị
}
public String toTenCv(){
    return this.TenChucVu;
}
}

```

3.2.2.5 Class LoaiHD

```

package CLASS;
public class LoaiHD {
    private String MaLoaiHd;
    private int ThoiHan;
    private String TenHopDong;
    private String MoTa;
    public LoaiHD() {
    }
    public LoaiHD(String MaLoaiHd, int ThoiHan, String TenHopDong, String MoTa) {
        this.MaLoaiHd = MaLoaiHd;
        this.ThoiHan = ThoiHan;
        this.TenHopDong = TenHopDong;
        this.MoTa = MoTa;
    }
    public String getMaLoaiHd() {
        return this.MaLoaiHd;
    }
    public void setMaLoaiHd(String MaLoaiHd) {
        this.MaLoaiHd = MaLoaiHd;
    }
    public int getThoiHan() {
        return this.ThoiHan;
    }
    public void setThoiHan(int ThoiHan) {
        this.ThoiHan = ThoiHan;
    }
    public String getTenHopDong() {

```

```

        return this.TenHopDong;
    }
    public void setTenHopDong(String TenHopDong) {
        this.TenHopDong = TenHopDong;
    }
    public String getMoTa() {
        return this.MoTa;
    }
    public void setMoTa(String MoTa) {
        this.MoTa = MoTa;
    }
    @Override
    public String toString() {
        return this.MaLoaiHD; // hoặc trả về bất kỳ thông tin nào bạn muốn hiển thị
    }
}

```

3.2.2.6 Class TaiKhoan

```

package CLASS;
public class TaiKhoan {
    private String TenDangNhap;
    private String MatKhau;
    private String MaNhanVien;
    public TaiKhoan() {
    }
    public TaiKhoan(String TenDangNhap, String MatKhau, String MaNhanVien) {
        this.TenDangNhap = TenDangNhap;
        this.MatKhau = MatKhau;
        this.MaNhanVien = MaNhanVien;
    }
    public String getTenDangNhap() {
        return this.TenDangNhap;
    }
    public void setTenDangNhap(String TenDangNhap) {
        this.TenDangNhap = TenDangNhap;
    }
    public String getMatKhau() {
        return this.MatKhau;
    }
    public void setMatKhau(String MatKhau) {
        this.MatKhau = MatKhau;
    }
    public String getMaNhanVien() {
        return this.MaNhanVien;
    }
    public void setMaNhanVien(String MaNhanVien) {
        this.MaNhanVien = MaNhanVien;
    }
}

```

3.2.3 DAO Class

3.2.3.1 NhanVienDAO

```

package DAO;
import CLASS.NhanVien;
import UTILS.JDBCHelper;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

```

```

import java.util.logging.Level;
import java.util.logging.Logger;
public class NhanVienDAO extends HRDAO<NhanVien, String> {
    String INSERT_SQL="INSERT INTO Nhan_Vien (maNhanVien, HoTen, GioiTinh, DiaChi, SDT, Email, NgaySinh,
MaPhongBan, MaChucVu, MaUngVien, MaLoaiHD, Luong) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?);";
    String UPDATE_SQL="UPDATE Nhan_Vien SET HoTen = ?, GioiTinh = ?, DiaChi = ?, SDT = ?, Email = ?, NgaySinh =
?, MaPhongBan = ?, MaChucVu = ?, MaUngVien = ?, MaLoaiHD = ?, Luong = ? WHERE maNhanVien = ?";
    String DELETE_SQL = "DELETE FROM Nhan_Vien WHERE maNhanVien = ?";
    String SELECT_ALL_SQL = "SELECT * FROM Nhan_Vien";
    String SELECT_BY_ID_SQL = "SELECT * FROM Nhan_Vien WHERE MaNhanVien = ?";
    @Override
    public void insert(NhanVien entity) {
        try {
            JDBCHelper.update(INSERT_SQL,entity.getMaNhanVien(),entity.getHoTen(),entity.getGioiTinh(),entity.getDiaChi(),entity.g
etSDT(),entity.getEmail(),entity.getNgaySinh(),entity.getMaPhongban(),entity.getMaChucVu(),entity.getMaUngVien(),entity.g
etMaLoaiHd(),entity.getLuong());
        } catch (SQLException ex) {
            Logger.getLogger(NhanVienDAO.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    @Override
    public void update(NhanVien entity) {
        try {
            JDBCHelper.update(UPDATE_SQL,entity.getHoTen(),entity.getGioiTinh(),entity.getDiaChi(),entity.getSDT(),entity.getEmail(
),entity.getNgaySinh(),entity.getMaPhongban(),entity.getMaChucVu(),entity.getMaUngVien(),entity.getMaLoaiHd(),entity.get
Luong(),entity.getMaNhanVien());
        } catch (SQLException ex) {
            Logger.getLogger(NhanVienDAO.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    @Override
    public void delete(String id) {
        try{
            JDBCHelper.update(DELETE_SQL, id);
        }
        catch(SQLException ex) {
            Logger.getLogger(NhanVienDAO.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    @Override
    public NhanVien selectById(String id) {
        List<NhanVien> ds = this.selectBySql(SELECT_BY_ID_SQL, id);
        if(ds.isEmpty()){
            return null;
        }
        return ds.get(0);
    }
    public List<NhanVien> selectByten(String hoten){
        String sql="Select * from Nhan_Vien where hoten like ?";
        return this.selectBySql(sql,"%"+hoten+"%");
    }
    public List<NhanVien> selectByMaPb(String id) {
        String sql="Select * from Nhan_Vien where MaPhongBan like ?";
        return this.selectBySql(sql,"%"+id+"%");
    }
    public List<NhanVien> selectByMaCv(String id) {
        String sql="Select * from Nhan_Vien where MaChucVu like ?";
        return this.selectBySql(sql,"%"+id+"%");
    }
    public NhanVien selectByMaUv(String id) {
        List<NhanVien> ds = this.selectBySql("Select * from Nhan_Vien where MaUngVien like ?","%" +id+"%");
    }

```

```

        if(ds.isEmpty()){
            return null;
        }
        return ds.get(0);
    }
    public List<NhanVien> selectByMaLoaiHD(String id) {
        String sql="Select * from Nhan_Vien where MaLoaiHd like ?";
        return this.selectBySql(sql,"%"+id+"%");
    }
    @Override
    public List<NhanVien> selectAll() {
        return this.selectBySql(SELECT_ALL_SQL);
    }
    @Override
    protected List<NhanVien> selectBySql(String sql, Object... args) {
        List<NhanVien> ds = new ArrayList<NhanVien>();
        try{
            ResultSet rs = JDBCHelper.query(sql, args);
            while(rs.next()){
                NhanVien a = new NhanVien();
                a.setMaNhanVien(rs.getString("MaNhanVien"));
                a.setHoTen(rs.getString("HoTen"));
                a.setGioiTinh(rs.getBoolean("GioiTinh"));
                a.setDiaChi(rs.getString("DiaChi"));
                a.setSDT(rs.getString("SDT"));
                a.setEmail(rs.getString("Email"));
                a.setNgaySinh(rs.getDate("NgaySinh"));
                a.setMaPhongban(rs.getString("MaPhongBan"));
                a.setMaChucVu(rs.getString("MaChucVu"));
                a.setMaUngVien(rs.getString("MaUngVien"));
                a.setMaLoaiHd(rs.getString("MaLoaiHD"));
                a.setLuong(rs.getDouble("Luong"));
                ds.add(a);
            }
            rs.getStatement().getConnection().close();
            return ds;
        }
        catch (Exception e){
            throw new RuntimeException(e);
        }
    }
}

```

3.2.3.2 TaiKhoanDAO

```

package DAO;
import CLASS.TaiKhoan;
import UTILS.JDBCHelper;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
public class TaiKhoanDAO extends HRDAO<TaiKhoan, String>{
    String INSERT_SQL="INSERT INTO Tai_Khoan (TenDangNhap, MatKhau, MaNhanVien) VALUES (?, ?, ?);";
    String UPDATE_SQL="UPDATE Tai_Khoan SET MatKhau = ?, MaNhanVien = ? WHERE TenDangNhap = ?";
    String DELETE_SQL = "DELETE FROM Tai_Khoan WHERE TenDangNhap = ?";
    String SELECT_ALL_SQL = "SELECT * FROM Tai_Khoan";
    String SELECT_BY_ID_SQL = "SELECT * FROM Tai_Khoan WHERE TenDangNhap = ?";
    @Override

```



```

public void insert(TaiKhoan entity) {
    try {
        JDBCHelper.update(INSERT_SQL, entity.getTenDangNhap(), entity.getMatKhau(), entity.getMaNhanVien());
    } catch (SQLException ex) {
        Logger.getLogger(TaiKhoanDAO.class.getName()).log(Level.SEVERE, null, ex);
    }
}

@Override
public void update(TaiKhoan entity) {
    try {
        JDBCHelper.update(UPDATE_SQL, entity.getMatKhau(), entity.getMaNhanVien(), entity.getTenDangNhap());
    } catch (SQLException ex) {
        Logger.getLogger(TaiKhoanDAO.class.getName()).log(Level.SEVERE, null, ex);
    }
}

@Override
public void delete(String id) {
    try {
        JDBCHelper.update(DELETE_SQL, id);
    } catch (SQLException ex) {
        Logger.getLogger(TaiKhoanDAO.class.getName()).log(Level.SEVERE, null, ex);
    }
}

@Override
public TaiKhoan selectById(String id) {
    List<TaiKhoan> ds = this.selectBySql(SELECT_BY_ID_SQL, id);
    if(ds.isEmpty()){
        return null;
    }
    return ds.get(0);
}

@Override
public List<TaiKhoan> selectAll() {
    return this.selectBySql(SELECT_ALL_SQL);
}

@Override
protected List<TaiKhoan> selectBySql(String sql, Object... args) {
    List<TaiKhoan> ds = new ArrayList<TaiKhoan>();
    try {
        ResultSet rs = JDBCHelper.query(sql, args);
        while(rs.next()){
            TaiKhoan a = new TaiKhoan();
            a.setTenDangNhap(rs.getString("TenDangNhap"));
            a.setMatKhau(rs.getString("MatKhau"));
            a.setMaNhanVien(rs.getString("MaNhanVien"));
            ds.add(a);
        }
        rs.getStatement().getConnection().close();
        return ds;
    } catch (Exception e) {
        throw new RuntimeException(e);
    }
}

public TaiKhoan selectByMaNhanVien(String maNhanVien) {
    String sql = "SELECT * FROM Tai_Khoan WHERE MaNhanVien = ?";
    List<TaiKhoan> list = this.selectBySql(sql, maNhanVien);
    return list.size() > 0 ? list.get(0) : null;
}
}

```

3.2.3.3 HRDAO

```
abstract public class HRDAO <EntityType, KeyType> {
    abstract public void insert(EntityType entity);
    abstract public void update(EntityType entity);
    abstract public void delete(KeyType id);
    abstract public EntityType selectById(KeyType id);
    abstract public List<EntityType>selectAll();
    abstract protected List<EntityType>selectBySql(String sql, Object...args);
}
```

3.2.3.4 ChucVuDAO

```
public class ChucVuDAO extends HRDAO<ChucVu, String>{
    String INSERT_SQL="INSERT INTO Chuc_Vu (MaChucVu, TenChucVu, MoTa) VALUES (?, ?, ?);";
    String UPDATE_SQL="UPDATE Chuc_Vu SET TenChucVu = ?, MoTa = ? WHERE MaChucVu = ?";
    String DELETE_SQL = "DELETE FROM Chuc_Vu WHERE MaChucVu = ?";
    String SELECT_ALL_SQL = "SELECT * FROM Chuc_Vu";
    String SELECT_BY_ID_SQL = "SELECT * FROM Chuc_Vu WHERE MaChucVu = ?";
    @Override
    public void insert(ChucVu entity) {
        try {
            JDBCHelper.update(INSERT_SQL,entity.getMaChucVu(),entity.getTenChucVu(),entity.getMota());
        } catch (SQLException ex) {
            Logger.getLogger(ChucVuDAO.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    @Override
    public void update(ChucVu entity) {
        try {
            JDBCHelper.update(UPDATE_SQL,entity.getTenChucVu(),entity.getMota(),entity.getMaChucVu());
        } catch (SQLException ex) {
            Logger.getLogger(ChucVuDAO.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    @Override
    public void delete(String id) {
        try{
            JDBCHelper.update(DELETE_SQL, id);
        }
        catch(SQLException ex) {
            Logger.getLogger(ChucVuDAO.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    @Override
    public ChucVu selectById(String id) {
        List<ChucVu> ds = this.selectBySql(SELECT_BY_ID_SQL, id);
        if(ds.isEmpty()){
            return null;
        }
        return ds.get(0);
    }
    public List<ChucVu> selectByten(String hoten){
        String sql="Select * from Chuc_Vu where tenChucVu like ?";
        return this.selectBySql(sql,"%"+hoten+"%");
    }
    @Override
    public List<ChucVu> selectAll() {
        return this.selectBySql(SELECT_ALL_SQL);
    }
    @Override
```

```
protected List<ChucVu> selectBySql(String sql, Object... args) {
    List<ChucVu> ds = new ArrayList<ChucVu>();
    try{
        ResultSet rs = JDBCHelper.query(sql, args);
        while(rs.next()){
            ChucVu a = new ChucVu();
            a.setMaChucVu(rs.getString("MaChucVu"));
            a.setTenChucVu(rs.getString("TenChucVu"));
            a.setMota(rs.getString("MoTa"));
            ds.add(a);
        }
        rs.getStatement().getConnection().close();
        return ds;
    }
    catch (Exception e){
        throw new RuntimeException(e);
    }
}
```

3.2.3.5 PhongBanDAO

```
public class PhongBanDAO extends HRDAO<PhongBan, String>{
    String INSERT_SQL="INSERT INTO Phong_Ban (MaPhongBan, TenPhongBan, MoTa, MaTruongPhong) VALUES (?, ?, ?, ?);";
    String UPDATE_SQL="UPDATE Phong_Ban SET TenPhongBan = ?, MoTa = ?, MaTruongPhong = ? WHERE MaPhongBan = ?";
    String DELETE_SQL = "DELETE FROM Phong_Ban WHERE MaPhongBan = ?";
    String SELECT_ALL_SQL = "SELECT * FROM Phong_Ban";
    String SELECT_BY_ID_SQL = "SELECT * FROM Phong_Ban WHERE MaPhongBan = ?";
    @Override
    public void insert(PhongBan entity) {
        try {
            JDBCHelper.update(INSERT_SQL,entity.getMaPhongBan(),entity.getTenPhongBan(),entity.getMoTa(),entity.getMaTruongPhong());
        } catch (SQLException ex) {
            Logger.getLogger(PhongBanDAO.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    @Override
    public void update(PhongBan entity) {
        try {
            JDBCHelper.update(UPDATE_SQL,entity.getTenPhongBan(),entity.getMoTa(),entity.getMaTruongPhong(),entity.getMaPhongBan());
        } catch (SQLException ex) {
            Logger.getLogger(PhongBanDAO.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    @Override
    public void delete(String id) {
        try{
            JDBCHelper.update(DELETE_SQL, id);
        }
        catch(SQLException ex) {
            Logger.getLogger(PhongBanDAO.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    @Override
    public PhongBan selectById(String id) {
```

```

List<PhongBan> ds = this.selectBySql(SELECT_BY_ID_SQL, id);
if(ds.isEmpty()){
    return null;
}
return ds.get(0);
}
@Override
public List<PhongBan> selectAll() {
    return this.selectBySql(SELECT_ALL_SQL);
}
@Override
protected List<PhongBan> selectBySql(String sql, Object... args) {
    List<PhongBan> ds = new ArrayList<PhongBan>();
    try{
        ResultSet rs = JDBCHelper.query(sql, args);
        while(rs.next()){
            PhongBan a = new PhongBan();
            a.setMaPhongBan(rs.getString("MaPhongBan"));
            a.setTenPhongBan(rs.getNString("TenPhongBan"));
            a.setMoTa(rs.getNString("MoTa"));
            a.setMaTruongPhong(rs.getString("MaTruongPhong"));
            ds.add(a);
        }
        rs.getStatement().getConnection().close();
        return ds;
    }
    catch (Exception e){
        throw new RuntimeException(e);
    }
}
}
public List<PhongBan> selectByTenPhongBan(String id){
    String sql="Select * from Phong_Ban where TenPhongBan like ?";
    return this.selectBySql(sql,"%"+id+"%");
}
public List<PhongBan> selectByMaTruongPhong(String id){
    String sql="Select * from Phong_Ban where MaTruongPhong like ?";
    return this.selectBySql(sql,"%"+id+"%");
}
}
}

```

3.2.3.6 UngVienDAO

```

public class UngVienDAO extends HRDAO<UngVien, String>{
    String INSERT_SQL="INSERT INTO Ung_Vien (MaUngVien, HoTen, GioiTinh, DiaChi, SĐT, Email, NgaySinh,
    MaChucVu, MoTa) VALUES (?, ?, ?, ?, ?, ?, ?, ?);";
    String UPDATE_SQL="UPDATE Ung_Vien SET HoTen = ?, GioiTinh = ?, DiaChi = ?, SĐT = ?, Email = ?, NgaySinh = ?,
    MaChucVu = ?, MoTa = ? WHERE MaUngVien = ?";
    String DELETE_SQL = "DELETE FROM Ung_Vien WHERE MaUngVien = ?";
    String SELECT_ALL_SQL = "SELECT * FROM Ung_Vien";
    String SELECT_BY_ID_SQL = "SELECT * FROM Ung_Vien WHERE MaUngVien = ?";

    @Override
    public void insert(UngVien entity) {
        try {
            JDBCHelper.update(INSERT_SQL,entity.getMaUngVien(),entity.getHoTen(),entity.isGioiTinh(),entity.getDiaChi(),entity.getSDT(),entity.getEmail(),entity.getNgaySinh(),entity.getMaChucVu(),entity.getMoTa());
        } catch (SQLException ex) {
            Logger.getLogger(UngVienDAO.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}

```

```

@Override
public void update(UngVien entity) {
    try {
        JDBCHelper.update(UPDATE_SQL,entity.getHoTen(),entity.isGioiTinh(),entity.getDiaChi(),entity.getSDT(),entity.getEmail(),
        entity.getNgaySinh(),entity.getMaChucVu(),entity.getMoTa(),entity.getMaUngVien());
    } catch (SQLException ex) {
        Logger.getLogger(UngVienDAO.class.getName()).log(Level.SEVERE, null, ex);
    }
}

@Override
public void delete(String id) {
    try{
        JDBCHelper.update(DELETE_SQL, id);
    }
    catch(SQLException ex) {
        Logger.getLogger(UngVienDAO.class.getName()).log(Level.SEVERE, null, ex);
    }
}

@Override
public UngVien selectById(String id) {
    List<UngVien> ds = this.selectBySql(SELECT_BY_ID_SQL, id);
    if(ds.isEmpty()){
        return null;
    }
    return ds.get(0);
}

@Override
public List<UngVien> selectAll() {
    return this.selectBySql(SELECT_ALL_SQL);
}

@Override
protected List<UngVien> selectBySql(String sql, Object... args) {
    List<UngVien> ds = new ArrayList<UngVien>();
    try{
        ResultSet rs = JDBCHelper.query(sql, args);
        while(rs.next()){
            UngVien a = new UngVien();
            a.setMaUngVien(rs.getString("MaUngVien"));
            a.setHoTen(rs.getNString("HoTen"));
            a.setSDT(rs.getString("SDT"));
            a.setDiaChi(rs.getNString("DiaChi"));
            a.setNgaySinh(rs.getDate("NgaySinh"));
            a.setGioiTinh(rs.getBoolean("GioiTinh"));
            a.setEmail(rs.getString("Email"));
            a.setMoTa(rs.getNString("MoTa"));
            a.setMaChucVu(rs.getString("MaChucVu"));
            ds.add(a);
        }
        rs.getStatement().getConnection().close();
        return ds;
    }
    catch (Exception e){
        throw new RuntimeException(e);
    }
}

```

```

public List<UngVien> selectByHoTen(String hoten){
    String sql="Select * from Ung_Vien where HoTen like ?";
    return this.selectBySql(sql,"%"+hoten+"%");
}

public List<UngVien> selectByMaChucVu(String id) {
    String sql="Select * from Ung_Vien where MaChucVu like ?";
    return this.selectBySql(sql,"%"+id+"%");
}

public UngVien selectBySdt(String id) {
    List<UngVien> ds = this.selectBySql("Select * from Ung_Vien where SĐT like ?", id);
    if(ds.isEmpty()){
        return null;
    }
    return ds.get(0);
}

public UngVien selectByEmail(String id) {
    List<UngVien> ds = this.selectBySql("Select * from Ung_Vien where Email like ?", id);
    if(ds.isEmpty()){
        return null;
    }
    return ds.get(0);
}
}

```

3.2.3.7 ThongKeDAO

```

public class ThongKeDAO {
    private List<Object[]> getListOfArray(String sql,String[] cols,Object...args){
        try{
            List<Object[]> ds = new ArrayList<>();
            ResultSet rs = JDBCHelper.query(sql, args);
            while(rs.next()){
                Object[] value = new Object[cols.length];
                for(int i=0;i<cols.length;i++){
                    value[i]=rs.getObject(cols[i]);
                }
                ds.add(value);
            }
            rs.getStatement().getConnection().close();
            return ds;
        }
        catch (Exception e){
            throw new RuntimeException(e);
        }
    }

    public List<Object[]> getUngVien(String tencv){
        String sql = "{CALL sp_sl_ungvien(?)}" ;
        String[] cols = {"TongUngVien","SLpass","SLfail"};
        return this.getListOfArray(sql, cols,tencv);
    }

    public List<Object[]> getTuoi(int mintuoi, int maxtuoi){
        String sql = "{CALL sp_sl_tuoi(?,?)}" ;
        String[] cols = {"SLNV"};
        return this.getListOfArray(sql, cols,mintuoi,maxtuoi);
    }

    public List<Object[]> getLuong(double minluong, double maxluong){
        String sql = "{CALL sp_sl_luong(?,?)}" ;
    }
}

```

```
String[] cols = {"SLNV"};
return this.getListOfArray(sql, cols, minluong, maxluong);
}
public List<Object[]> getPhongBan(String tenpb){
String sql = "{CALL sp_sl_pb(?)}";
String[] cols = {"SLNV", "SLCV", "HOTENTRUONGPHONG"};
return this.getListOfArray(sql, cols, tenpb);
}
public List<Object[]> getHopDong(){
String sql = "{CALL sp_sl_hd()}";
String[] cols = {"TenHopDong", "SLNV"};
return this.getListOfArray(sql, cols);
}
}
```

3.2.3.8 LoaiHDDAO

```
public class LoaiHDDAO extends HRDAO<LoaiHD, String>{
String INSERT_SQL="INSERT INTO Loai_Hop_dong (MaLoaiHd, ThoiHan, TenHopDong, MoTa) VALUES (?, ?, ?, ?)";
String UPDATE_SQL="UPDATE Loai_Hop_Dong SET TenHopDong = ?, ThoiHan = ?, MoTa = ? WHERE MaLoaiHd = ?";
String DELETE_SQL = "DELETE FROM Loai_Hop_Dong WHERE MaLoaiHd = ?";
String SELECT_ALL_SQL = "SELECT * FROM Loai_Hop_Dong";
String SELECT_BY_ID_SQL = "SELECT * FROM Loai_Hop_Dong WHERE MaLoaiHd = ?";

@Override
public void insert(LoaiHD entity) {
try {
JDBCHelper.update(INSERT_SQL, entity.getMaLoaiHd(), entity.getThoiHan(), entity.getTenHopDong(), entity.getMoTa());
} catch (SQLException ex) {
Logger.getLogger(LoaiHDDAO.class.getName()).log(Level.SEVERE, null, ex);
}
}

@Override
public void update(LoaiHD entity) {
try {
JDBCHelper.update(UPDATE_SQL, entity.getTenHopDong(), entity.getThoiHan(), entity.getMoTa(), entity.getMaLoaiHd());
} catch (SQLException ex) {
Logger.getLogger(LoaiHDDAO.class.getName()).log(Level.SEVERE, null, ex);
}
}

@Override
public void delete(String id) {
try{
JDBCHelper.update(DELETE_SQL, id);
}
catch(SQLException ex) {
Logger.getLogger(LoaiHDDAO.class.getName()).log(Level.SEVERE, null, ex);
}
}

@Override
public LoaiHD selectById(String id) {
List<LoaiHD> ds = this.selectBySql(SELECT_BY_ID_SQL, id);
if(ds.isEmpty()){
return null;
}
}
```

```

        return ds.get(0);
    }

    @Override
    public List<LoaiHD> selectAll() {
        return this.selectBySql(SELECT_ALL_SQL);
    }

    @Override
    protected List<LoaiHD> selectBySql(String sql, Object... args) {
        List<LoaiHD> ds = new ArrayList<LoaiHD>();
        try{
            ResultSet rs = JDBCHelper.query(sql, args);
            while(rs.next()){
                LoaiHD a = new LoaiHD();
                a.setMaLoaiHd(rs.getString("MaLoaiHd"));
                a.setThoiHan(rs.getInt("ThoiHan"));
                a.setTenHopDong(rs.getString("TenHopDong"));

                a.setMoTa(rs.getString("MoTa"));

                ds.add(a);
            }
            rs.getStatement().getConnection().close();
            return ds;
        }
        catch (Exception e){
            throw new RuntimeException(e);
        }
    }
}

```

3.3 UTILS

3.3.1 Lớp tiện ích XDate

```

public class XDate {
    static SimpleDateFormat form= new SimpleDateFormat();
    public static Date toDate(String date, String pattern){
        try{
            form.applyPattern(pattern);
            return form.parse(date);
        }
        catch(Exception e){
            throw new RuntimeException(e);
        }
    }
    public static String toString(Date date,String pattern){
        form.applyPattern(pattern);
        return form.format(date);
    }
    public static Date addDays(Date date,long days){
        date.setTime(date.getTime()+days*24*60*60*1000);
        return date;
    }
}

```

3.3.2 Lớp tiện ích Auth

```

public class Auth {
    public static TaiKhoan user = null;
    public static NhanVienDAO nv;
}

```



```

public static void clear() {
    Auth.user = null;
    Auth.nv = null;
}
public static boolean isLogin() {
    return Auth.user != null;
}
public static boolean isManager(){
    NhanVienDAO nvdao = new NhanVienDAO();
    NhanVien nhanVien = nvdao.selectById(Auth.user.getMaNhanVien());
    if (Auth.user == null) {
        return false; // Không đăng nhập hoặc không có thông tin nhân viên
    }

    if (nhanVien == null) {
        return false; // Không tìm thấy nhân viên
    }
    return "CV001".equals(nhanVien.getMaChucVu().trim()); // Kiểm tra mã chức vụ
}
}

```

3.3.3 Lớp tiện ích MsgBox

```

public class MsgBox {
    public static void alert(Component parent, String message) {
        JOptionPane.showMessageDialog(parent, message, "EduSys", JOptionPane.INFORMATION_MESSAGE);
    }

    public static boolean confirm(Component parent, String message) {
        int result = JOptionPane.showConfirmDialog(parent, message, "EduSys", JOptionPane.YES_NO_OPTION,
        JOptionPane.QUESTION_MESSAGE);
        return result == JOptionPane.YES_OPTION;
    }

    public static String prompt(Component parent, String message) {
        return JOptionPane.showInputDialog(parent, message, "EduSys", JOptionPane.INFORMATION_MESSAGE);
    }
}

```

3.4 LẬP TRÌNH NGHIỆP VỤ

3.4.1 Xử lý Form Main

3.4.1.1 Xử lý button Login

```

public void openLogin(){
    Login a = new Login();
    openWindow(Desktop, a);
}

```

3.4.1.2 Xử lý button Đăng xuất

```

public void DangXuat(){
    Auth.clear();
    JInternalFrame[] frames = Desktop.getAllFrames();
    for (JInternalFrame frame : frames) {
        frame.dispose();
    }
    openLogin();
}

```

3.4.1.3 Xử lý button Chức vụ

```
public void openPositionManagement(){
    if(Auth.isLogin()){
        PositionManagement a = new PositionManagement();
        openWindow(Desktop, a);
    }else{
        MsgBox.alert(this,"Vui lòng đăng nhập");
    }
}
```

3.4.1.4 Xử lý button Hợp đồng

```
public void openContractManagement(){
    if(Auth.isLogin()){
        ContractManagement a = new ContractManagement();
        openWindow(Desktop, a);
    }
    else{
        MsgBox.alert(this,"Vui lòng đăng nhập");
    }
}
```

3.4.1.5 Xử lý button Nhân viên

```
public void openEmployeeManagement(){
    if(Auth.isLogin()){
        EmployeeManagement a = new EmployeeManagement();
        openWindow(Desktop, a);
    }
    else{
        MsgBox.alert(this,"Vui lòng đăng nhập");
    }
}
```

3.4.1.6 Xử lý button Phòng ban

```
public void openDepartmentManagement(){
    if(Auth.isLogin()){
        DepartmentManagement a = new DepartmentManagement();
        openWindow(Desktop, a);
    }
    else{
        MsgBox.alert(this,"Vui lòng đăng nhập");
    }
}
```

3.4.1.7 Xử lý button Tài khoản

```
public void openAccountManagement(){
    if(Auth.isLogin()){
        if(!Auth.isManager()){
            MsgBox.alert(this, "Bạn không có quyền xem quản lý tài khoản!");
            return;
        }
    }
    else{
        AccountManagement a = new AccountManagement();
        openWindow(Desktop, a);
    }
}
else{
    MsgBox.alert(this,"Vui lòng đăng nhập");
}
```

3.4.1.8 Xử lý button Ứng viên

```
public void openCandidateManagement() {
    if(Auth.isLogin()){
        CandidateManagement a = new CandidateManagement();
        openWindow(Desktop, a);
    }
    else{
        MsgBox.alert(this, "Vui lòng đăng nhập");
    }
}
```

3.4.1.9 Xử lý button Thống kê

```
public void openStatistics(int index) {
    if(Auth.isLogin()){
        if(!Auth.isManager()){

            MsgBox.alert(this, "Bạn không có quyền xem thống kê!");
            return;
        }
        else{
            Statistics a = new Statistics();
            openWindow(Desktop, a);
            a.selectTab(index);
        }
    }
    else{
        MsgBox.alert(this, "Vui lòng đăng nhập");
    }
}
```

3.4.1.10 Xử lý button Đổi mật khẩu

```
public void openChangePassword() {
    if(Auth.isLogin()){
        ChangePassword a = new ChangePassPass();
        openWindow(Desktop, a);
    }
    else{
        MsgBox.alert(this, "Vui lòng đăng nhập");
    }
}
```

3.4.2 Xử lý Form Splash

```
public class Splash extends javax.swing.JDialog {
    public Splash(java.awt.Frame parent, boolean modal) {
        super(parent, modal);
        initComponents();
        init();
    }

    void init(){
        setLocationRelativeTo(null);
        setTitle("Hệ thống quản lý nhân viên");
    }

    public void Loading(int value, String message){
        pgbLoading.setValue(value);
        lblStatus.setText(message);
        lblValue.setText(String.valueOf(value)+"%");
    }
}
```

```

        try {
            Thread.sleep(value*100);
        } catch (InterruptedException ex) {
        }
    }
}

private void formWindowOpened(java.awt.event.WindowEvent evt) {
    Thread t1 = new Thread () {
        public void run(){
            try {
                Loading(10,"Đang khởi động...");
                Loading (30, "Đang kết nối cơ sở dữ liệu...");
                Loading (50, "Đang tải cơ sở dữ liệu...");
                Loading (70,"Đang tải lên hệ thống...");
                Loading (100,"Hoàn thành!");
                dispose();
            }
            catch (Exception e){
            }
        }
    };
    t1.start();
}

```

3.4.3 Xử lý Form Login

3.4.3.1 Xử lý button Đăng nhập

```

TaiKhoanDAO dao = new TaiKhoanDAO();
void dangNhap(){
    String manv=txtTenDangNhap.getText();
    String mk=new String(txtMatKhau.getPassword());
    TaiKhoan tk = dao.selectById(manv);
    if(tk==null){
        MsgBox.alert(this,"Sai tên đăng nhập!");
    }
    else if(!mk.equals(tk.getMatKhau())){
        MsgBox.alert(this,"Sai mật khẩu!");
    }
    else{
        Auth.user=tk;
        this.dispose();
    }
}

```

3.4.3.2 Xử lý button Kết thúc

```

void ketThuc(){
    if(Msgbox.confirm(this,"Bạn muốn kết thúc ứng dụng?")){
        System.exit(0);
    }
}

```

3.4.4 Xử lý Form Đổi mật khẩu

3.4.4.1 Xử lý button Đổi mật khẩu

```

void doiMatKhau() {
    String manv = txtTenDangNhap.getText();
    String matKhau = new String(txtMatKhau.getPassword());
    String matKhauMoi1 = new String(txtMatKhau1.getPassword());
    String matKhauMoi2 = new String(txtMatKhau2.getPassword());
    if(manv==null|| matKhau==null || matKhauMoi1==null||matKhauMoi2==null){

```

```

        MsgBox.alert(this, "Không được để trống các trường dữ liệu!");
    }
    if (!manv.equalsIgnoreCase(Auth.user.getTenDangNhap())) {
        MsgBox.alert(this, "Sai tên đăng nhập");
    } else if (!matKhau.equals(Auth.user.getMatKhau())) {
        MsgBox.alert(this, "Sai mật khẩu");
    } else if (!matKhauMoi1.equals(matKhauMoi2)) {
        MsgBox.alert(this, "Xác nhận mật khẩu không đúng!");
    } else {
        Auth.user.setMatKhau(matKhauMoi1);
        dao.update(Auth.user);
        MsgBox.alert(this, "Đổi mật khẩu thành công!");
    }
}

```

3.4.4.2 Xử lý button Huỷ bỏ

```

void huyBo() {
    this.dispose();
};
}

```

3.4.5 Xử lý Form Nhân viên

3.4.5.1 Xử lý button Thêm

```

private void insert() {
    NhanVien nv = getForm();
    if (nv != null) {
        try {
            this.dao.insert(nv);
            this.fillTable();
            this.clearForm();
            MsgBox.alert(this, "Thêm mới thành công!");
        }
        catch (Exception e) {
            MsgBox.alert(this, "Thêm mới thất bại!");
            System.out.print(e);
        }
    }
}

```

3.4.5.2 Xử lý button Sửa

```

private void edit() {
    String manv = (String) tblNhanVien.getValueAt(this.row, 0);
    NhanVien nv = dao.selectById(manv);
    this.setForm(nv);
    tabs.setSelectedIndex(0);
    this.updateStatus();
}

```

3.4.5.3 Xử lý button Xóa

```

private void delete() {
    String manv = txtMaNV.getText().trim();
    if (manv.equals(Auth.user.getMaNhanVien())) {
        MsgBox.alert(this, "Bạn không được xóa chính bạn!");
    } else if (Msgbox.confirm(this, "Bạn muốn xóa nhân viên này?")) {
        try {
            dao.delete(manv);
            this.fillTable();
            MsgBox.alert(this, "Xóa thành công!");
        }
    }
}

```

```

        } catch (Exception e) {
            MsgBox.alert(this, "Xóa thất bại!");
        }
    }
    if(row>tblNhanVien.getRowCount()){
        row = tblNhanVien.getRowCount();
        this.edit();
    }
    else{
        this.edit();
    }
}

```

3.4.5.4 Xử lý button Reset

```

private void clearForm() {
    txtMaNV.setText("");
    txtHoTen.setText("");
    txtSdt.setText("");
    txtNgaySinh.setText("");
    txtLuong.setText("");
    txtEmail.setText("");
    txtDiaChi.setText("");
    txtFind.setText("");
    txtMaUv.setText("");
    rdoNam.setSelected(true);
    rdoNu.setSelected(false);
    this.row = -1;
    this.updateStatus();
}

```

3.4.5.5 Xử lý button First

```

private void first() {
    this.row = 0;
    this.edit();
}

```

3.4.5.6 Xử lý button Prev

```

private void prev() {
    if (this.row > 0) {
        this.row--;
        this.edit();
    }
}

```

3.4.5.7 Xử lý button Next

```

private void next() {
    if (this.row < tblNhanVien.getRowCount() - 1) {
        this.row++;
        this.edit();
    }
}

```

3.4.5.8 Xử lý button Last

```

private void last() {
    this.row = tblNhanVien.getRowCount() - 1;
    this.edit();
}

```

3.4.5.9 Xử lý button Tìm kiếm

```
private void timKiem() {
    this.fillFind();
    this.clearForm();
    this.row = -1;
    updateStatus();
}
private void fillFind() {
    DefaultTableModel model = (DefaultTableModel) tblNhanVien.getModel();
    model.setRowCount(0);
    String keyword = txtFind.getText();
    try {
        List<NhanVien> ds = new ArrayList<>();
        switch (cboFind.getSelectedIndex()) {
            case 0:
                NhanVien nv = dao.selectById(keyword);
                if (nv != null) {
                    ds.add(nv);
                }
                break;
            case 1:
                ds = dao.selectByten(keyword); // Danh sách nhân viên theo tên
                break;
            case 2:
                ds = (List<NhanVien>) dao.selectByMaPb(keyword); // Danh sách nhân viên theo mã phòng ban
                break;
            case 3:
                ds = (List<NhanVien>) dao.selectByMaCv(keyword); // Danh sách nhân viên theo mã chức vụ
                break;
            case 4:
                NhanVien nvByMaUv = dao.selectByMaUv(keyword); // Đối tượng nhân viên theo mã ứng viên
                if (nvByMaUv != null) {
                    ds.add(nvByMaUv); // Thêm vào danh sách nếu có kết quả
                }
                break;
            case 5:
                ds = (List<NhanVien>) dao.selectByMaLoaiHD(keyword); // Danh sách nhân viên theo mã loại hợp đồng
                break;
            default:
                this.fillTable();
                break;
        }
        for (NhanVien nvItem : ds) {
            Object[] row = {
                nvItem.getMaNhanVien(),
                nvItem.getHoTen(),
                nvItem.isGioiTinh() ? "Nam" : "Nữ",
                nvItem.getDiaChi(),
                nvItem.getSDT(),
                nvItem.getEmail(),
                XDate.toString(nvItem.getNgaySinh(), "dd-MM-yyyy"),
                nvItem.getMaPhongban(),
                nvItem.getMaChucVu(),
                nvItem.getMaUngVien(),
                nvItem.getMaLoaiHd(),
                nvItem.getLuong()
            };
            model.addRow(row);
        }
    } catch (Exception e) {
        MsgBox.alert(this, "Lỗi truy vấn dữ liệu!");
    }
}
```

```

        System.out.println(e);
    }
}

```

3.4.6 Xử lý Form Ứng viên

3.4.6.1 Xử lý button Thêm

Tương tự Nhân viên

3.4.6.2 Xử lý button Sửa

Tương tự Nhân viên

3.4.6.3 Xử lý button Reset

Tương tự Nhân viên

3.4.6.4 Xử lý button Xóa

Tương tự Nhân viên

3.4.6.5 Xử lý button Tìm kiếm

```

private void timKiem() {
    this.fillFind();
    this.clearForm();
    this.row = -1;
    updateStatus();
}

private void fillFind() {
    DefaultTableModel model = (DefaultTableModel) tblUngVien.getModel();
    model.setRowCount(0);

    String keyword = txtFind.getText().trim();
    try {
        List<UngVien> dsUV = new ArrayList<>();
        switch (cboFind.getSelectedIndex()) {
            case 0:
                UngVien uv = uvDAO.selectById(keyword);
                if (uv != null) {
                    dsUV.add(uv);
                }
                break;
            case 1:
                dsUV = uvDAO.selectByHoTen(keyword); // Danh sách nhân viên theo tên
                break;
            case 2:
                dsUV = (List<UngVien>) uvDAO.selectByMaChucVu(keyword); // Danh sách nhân viên theo mã chức vụ
                break;
            default:
                this.fillTable();
                break;
        }

        for (UngVien uv : dsUV) {
            Object[] row = {
                uv.getMaUngVien(),
                uv.getHoTen(),
                uv.isGioiTinh() ? "Nam" : "Nữ",
            }

```



```

        uv.getNgaySinh(),
        uv.getSDT(),
        uv.getEmail(),
        uv.getDiaChi(),
        uv.getMaChucVu(),
        uv.getMoTa()
    };
    model.addRow(row);
}
} catch (Exception e) {
    MsgBox.alert(this, "Lỗi truy vấn dữ liệu!");
    System.out.print(e);
}
}
}

```

3.4.6.6 Xử lý button First

Tương tự Nhân viên

3.4.6.7 Xử lý button Prev

Tương tự Nhân viên

3.4.6.8 Xử lý button Next

Tương tự Nhân viên

3.4.6.9 Xử lý button Last

Tương tự Nhân viên

3.4.6.10 Xử lý button Thêm nhân viên

```

void themNhanVien() {
    try {
        UV_NV them = new UV_NV(); // Truyền JDesktopPane
        them.setVisible(true);
        maUngVien = txtMaUngVien.getText();
        tenUv = txtHoTen.getText();
        ngaySinhUv = txtNgaySinh.getText();
        gioiTinhUv = rdoNam.isSelected();
        sdtUv = txtSdt.getText();
        emailUv = txtEmail.getText();
        dcUv = txtDiaChi.getText();
        cvUv = (String) cboMaChucVu.getSelectedItem();
    } catch (Exception ex) {
        Logger.getLogger(CandidateManagement.class.getName()).log(Level.SEVERE, null, ex);
    }
}
}

```

3.4.7 Xử lý Form Hợp đồng

3.4.7.1 Xử lý button First

Tương tự Nhân viên

3.4.7.2 Xử lý button Prev

Tương tự Nhân viên

3.4.7.3 Xử lý button Next

Tương tự Nhân viên

3.4.7.4 Xử lý button Last

Tương tự Nhân viên

3.4.8 Xử lý Form Tài khoản

3.4.8.1 Xử lý button Thêm

Tương tự Nhân viên

3.4.8.2 Xử lý button Sửa

Tương tự Nhân viên

3.4.8.3 Xử lý button Reset

Tương tự Nhân viên

3.4.8.4 Xử lý button Xoá

Tương tự Nhân viên

3.4.8.5 Xử lý button Tìm kiếm

```
void timKiem() {  
    String MaNhanVien = txtTimKiemMaNhanVien.getText();  
  
    TaiKhoan tk = tkDAO.selectByMaNhanVien(MaNhanVien);  
    if (tk != null) {  
        DefaultTableModel model = (DefaultTableModel) tblTaiKhoan.getModel();  
        model.setRowCount(0); // Xóa tất cả các hàng hiện tại  
        Object[] row = {  
            tk.getTenDangNhap(),  
            tk.getMatKhau(),  
            tk.getMaNhanVien()  
        };  
        model.addRow(row); // Thêm hàng mới với thông tin tìm được  
        MsgBox.alert(this, "Tìm thấy tài khoản!");  
    } else {  
        MsgBox.alert(this, "Không tìm thấy tài khoản với mã nhân viên này!");  
        fillTable(); // Tải lại toàn bộ bảng nếu không tìm thấy  
    }  
}
```

3.4.8.6 Xử lý button First

Tương tự Nhân viên

3.4.8.7 Xử lý button Prev

Tương tự Nhân viên

3.4.8.8 Xử lý button Next

Tương tự Nhân viên

3.4.8.9 Xử lý button Last

Tương tự Nhân viên

3.4.9 Xử lý Form Phòng ban

3.4.9.1 Xử lý button Thêm

Tương tự Nhân viên

3.4.9.2 Xử lý button Sửa

Tương tự Nhân viên

3.4.9.3 Xử lý button Reset

Tương tự Nhân viên

3.4.9.4 Xử lý button Xóa

Tương tự Nhân viên

3.4.9.5 Xử lý button Tìm kiếm

```
private void timKiem() {
    this.fillFind();
    this.clearForm();
    this.row = -1;
    updateStatus();
}

// Trong phương thức fillFind(), điều chỉnh xử lý khi chọn tìm kiếm theo mã ứng viên
private void fillFind() {
    DefaultTableModel model = (DefaultTableModel) tblPhongBan.getModel();
    model.setRowCount(0);

    String keyword = txtFind.getText();
    try {
        List<PhongBan> ds = new ArrayList<>();
        switch (cboFind.getSelectedIndex()) {
            case 0:
                PhongBan pb = pbDAO.selectById(keyword);
                if (pb != null) {
                    ds.add(pb);
                }
                break;
            case 1:
                ds = (List<PhongBan>) pbDAO.selectByTenPhongBan(keyword);
                break;
            case 2:
                ds = (List<PhongBan>) pbDAO.selectByMaTruongPhong(keyword);
                break;

            default:
                this.fillTable();
                break;
        }

        for (PhongBan pbItem : ds) {
            Object[] row = {
                pbItem.getMaPhongBan(),
```

```

        pbItem.getTenPhongBan(),
        pbItem.getMaTruongPhong(),
        pbItem.getMoTa(),
    };
    model.addRow(row);
}
} catch (Exception e) {
    MsgBox.alert(this, "Lỗi truy vấn dữ liệu!");
    System.out.println(e);
}
}
}

```

3.4.9.6 Xử lý button First

Tương tự Nhân viên

3.4.9.7 Xử lý button Prev

Tương tự Nhân viên

3.4.9.8 Xử lý button Next

Tương tự Nhân viên

3.4.9.9 Xử lý button Last

Tương tự Nhân viên

3.4.10 Xử lý Form Chức vụ

3.4.10.1 Xử lý button Thêm

Tương tự Nhân viên

3.4.10.2 Xử lý button Sửa

Tương tự Nhân viên

3.4.10.3 Xử lý button Reset

Tương tự Nhân viên

3.4.10.4 Xử lý button Xóa

Tương tự Nhân viên

3.4.10.5 Xử lý button Tìm kiếm

```

private void timKiem() {
    this.fillFind();
    this.clearForm();
    this.row = -1;
    updateStatus();
}

// Trong phương thức fillFind(), điều chỉnh xử lý khi chọn tìm kiếm theo mã ứng viên
private void fillFind() {
    DefaultTableModel model = (DefaultTableModel) tblChucVu.getModel();
    model.setRowCount(0);

    String keyword = txtFind.getText();

```

```
try {
    List<ChucVu> ds = new ArrayList<>();
    switch (cboFind.getSelectedIndex()) {
        case 0:
            ChucVu cv = dao.selectById(keyword);
            if (cv != null) {
                ds.add(cv);
            }
            break;
        case 1:
            ds = dao.selectByten(keyword); // Danh sách nhân viên theo tên
            break;
        default:
            this.fillTable();
            break;
    }

    for (ChucVu cv : ds) {
        Object[] row = {
            cv.getMaChucVu(),
            cv.getTenChucVu(),
            cv.getMota()
        };
        model.addRow(row);
    }
} catch (Exception e) {
    MsgBox.alert(this, "Lỗi truy vấn dữ liệu!");
    System.out.println(e);
}
}
```

3.4.10.6 Xử lý button First

Tương tự Nhân viên

3.4.10.7 Xử lý button Prev

Tương tự Nhân viên

3.4.10.8 Xử lý button Next

Tương tự Nhân viên

3.4.10.9 Xử lý button Last

Tương tự Nhân viên

3.4.11 Xử lý Form Thống kê

3.4.11.1 Xử lý tab Ứng viên

```
private void fillComboBoxCv() {
    DefaultComboBoxModel model = (DefaultComboBoxModel) cboCv.getModel();
    model.removeAllElements();
    try {
        List<ChucVu> list = cvdao.selectAll();
        for (ChucVu cv: list) {
            model.addElement(cv.toTenCv());
        }
    }
}
```

```

        catch (Exception e) {
            MsgBox.alert(this, "Lỗi truy vấn dữ liệu");
            e.printStackTrace();
        }
    }
}

private void fillTblUngVien() {
    DefaultTableModel model = (DefaultTableModel)tblUngVien.getModel();
    model.setRowCount(0);
    String selectedCv = (String) cboCv.getSelectedItem();
    System.out.println(selectedCv);
    List<Object[]> ds = tkdao.getUngVien(selectedCv);
    for(Object[] row:ds){
        model.addRow(new Object[] {row[0],row[1],row[2]});
    }
}

```

3.4.11.1.2 Xử lý tab Độ tuổi

```

private void fillTblTuoi() {
    DefaultTableModel model = (DefaultTableModel)tblTuoi.getModel();
    model.setRowCount(0);
    int min, max;
    if (txtMinTuoi.getText().isEmpty()) {
        min = 18;
        txtMinTuoi.setText("18");
    }
    else {
        min = Integer.parseInt(txtMinTuoi.getText());
    }
    if (txtMaxTuoi.getText().isEmpty()) {
        max = 60;
        txtMaxTuoi.setText("60");
    }
    else {
        max = Integer.parseInt(txtMaxTuoi.getText());
    }
    List<Object[]> ds = tkdao.getTuoi(min, max);
    for(Object[] row:ds){
        model.addRow(row);
    }
}

```

3.4.11.1.3 Xử lý tab Lương

```

private void fillTblLuong() {
    DefaultTableModel model = (DefaultTableModel)tblLuong.getModel();
    model.setRowCount(0);
    int min, max;
    if (txtMinLuong.getText().isEmpty()) {
        min = 5000000;
        txtMinLuong.setText("5000000");
    }
    else {
        min = Integer.parseInt(txtMinLuong.getText());
    }
    if (txtMaxLuong.getText().isEmpty()) {
        max = 50000000;
        txtMaxLuong.setText("50000000");
    }
    else {

```

```

        max = Integer.parseInt(txtMaxLuong.getText());
    }
    List<Object[]> ds = tkdao.getLuong(min, max);
    for(Object[] row:ds){
        model.addRow(row);
    }
}

```

3.4.11.1.4 Xử lý tab Hợp đồng

```

private void fillTblHopDong(){
    DefaultTableModel model = (DefaultTableModel)tblHopDong.getModel();
    model.setRowCount(0);
    List<Object[]> ds = tkdao.getHopDong();
    for(Object[] row:ds){
        model.addRow(row);
    }
}

```

3.4.11.1.5 Xử lý tab Phòng ban

```

private void fillComboBoxPb(){
    DefaultComboBoxModel model = (DefaultComboBoxModel) cboPb.getModel();
    model.removeAllElements();
    try {
        List<PhongBan> list = pbdao.selectAll();
        for (PhongBan pb: list) {
            model.addElement(pb.toTenPb());
        }
    }
    catch (Exception e) {
        MsgBox.alert(this, "Lỗi truy vấn dữ liệu");
        e.printStackTrace();
    }
}

private void fillTblPhongBan(){
    DefaultTableModel model = (DefaultTableModel)tblPhongBan.getModel();
    model.setRowCount(0);
    String selectedPb = (String) cboPb.getSelectedItem();
    List<Object[]> ds = tkdao.getPhongBan(selectedPb);
    for(Object[] row:ds){
        model.addRow(row);
    }
}

```

4 KIỂM THỬ

4.1 KIỂM THỬ FORM LOGIN

TH	MÔ TẢ TÌNH HUỐNG	DỮ LIỆU MẪU	KẾT QUẢ	FIX
1	Đăng nhập để trống tên và mật khẩu		Yêu cầu nhập	ok
2	Đăng nhập sai tên	us111, 111111	Sai tên đăng nhập	ok
3	Đăng nhập sai mật khẩu	us001, ahihi	Sai mật khẩu	ok

4	Đăng nhập đúng	us001, 111111	Đóng cửa sổ	ok
---	----------------	---------------	-------------	----

4.2 KIỂM THỬ FORM TÀI KHOẢN

TH	MÔ TẢ TÌNH HUỐNG	DỮ LIỆU MẪU	KẾT QUẢ	FIX
1	Mở form		Đổ dữ liệu vào Table. Form hiển thị tài khoản hàng đầu tiên trong table.	ok
2	Để trống form		Yêu cầu nhập	ok
3	Kiểm tra trùng tên đăng nhập	us001	Thông báo trùng	ok
4	Kiểm tra các định dạng	-1000,sdf@,...	Thông báo lỗi	ok
5	Nút điều hướng		Form thay đổi dữ liệu	ok
6	Kiểm tra sự kiện MouseClicked		Hiển thị thông tin Tài khoản lên form	ok
7	Tìm kiếm theo tiêu chí mã nhân viên	nv001	Hiển thị “Tài khoản” theo mã nhân viên và đổ lên Table	ok
8	Tìm kiếm theo tiêu chí mã nhân viên	bỏ trống/ Nhập sai	Đổ tất cả “Tài khoản” đang có lên Table	ok
9	Thêm	Nhập sai/ bỏ trống	Thông báo lỗi, Không thêm được	ok
10	Thêm	NV009, us003,33333	Thông báo thêm thành công	ok
11	Sửa	Nhập sai/bỏ trống	Thông báo lỗi, Không sửa được	ok
12	Sửa	thông tin đúng	Thông báo cập nhật thành công	ok
13	Click từ Table qua form. Xóa	Sửa tên đăng nhập sai	Thông báo xóa thất bại	ok
14	Click từ Table qua form		Thông báo xóa thành	ok

	Xóa		công	
15	Reset		Tắt cả dữ liệu đã có trong Textfield chuyển thành null	ok

4.3 KIỂM THỬ FORM NHÂN VIÊN

TH	MÔ TẢ TÌNH HUỐNG	DỮ LIỆU MẪU	KẾT QUẢ	FIX
1	Mở form		Đổ dữ liệu vào Table. Đổ dữ liệu vào ComboBox. Form hiển thị Nhân viên hàng đầu tiên trong Table.	ok
2	Để trống form		Yêu cầu nhập	ok
3	Kiểm tra trùng mã nhân viên	NV001	Thông báo trùng	ok
4	Kiểm tra các định dạng	-1000,sdf@,...	Yêu cầu nhập lại	ok
5	Nút điều hướng		Form thay đổi dữ liệu	ok
6	Kiểm tra sự kiện MouseClicked		Hiển thị lại thông tin Nhân viên lên form	ok
7	Tìm kiếm theo tiêu chí mã nhân viên, họ tên nhân viên, mã chức vụ, mã ứng viên, mã loại hợp đồng, mã phòng ban	UV001, Nguyễn Văn Thắng, CV001, NV001, PB001, LHD01	Hiển thị “Nhân viên” theo các tiêu chí và đổ lên Table	ok
8	Tìm kiếm theo tiêu chí mã nhân viên, họ tên nhân viên, mã chức vụ, mã ứng viên, mã loại hợp đồng, mã phòng ban	bỏ trống/ Nhập sai	Đổ tất cả “Nhân viên” đang có lên Table	ok
9	Thêm	Nhập sai/ bỏ trống	Thông báo lỗi, Không thêm được	ok
10	Thêm	NV012, Nguyễn	Thông báo thêm thành	ok

		Văn Thắng, 1, 234 Đường ABC Quận 1 TP.HCM, 0985643321, nguyena@exam- le.com, 1990-01-10, PB001, CV002, UV008, LHD01, 20000000,2024- 08-07	công	
11	Sửa	Nhập sai/bỏ trống	Thông báo lỗi, Không sửa được	ok
12	Sửa	thông tin đúng	Thông báo cập nhật thành công	ok
13	Click từ Table qua form Xóa	Sửa mã nhân viên sai	Thông báo xóa thất bại	ok
14	Click từ Table qua form Xóa		Thông báo xóa thành công	ok
15	Reset		Tất cả dữ liệu đã có trong Textfield chuyển thành null	ok

4.4 KIỂM THỬ FORM CHỨC VỤ

TH	MÔ TẢ TÌNH HUỐNG	DỮ LIỆU MẪU	KẾT QUẢ	FIX
1	Mở form		Đổ dữ liệu vào Table. Đổ dữ liệu vào ComboBox. Form hiển thị Chức vụ hàng đầu tiên trong Table.	ok
2	Để trống form		Yêu cầu nhập	ok
3	Kiểm tra trùng mã chức vụ	CV001	Thông báo trùng	ok
4	Kiểm tra các định dạng	CV00sd	Yêu cầu nhập lại	ok

5	Nút điều hướng		Form thay đổi dữ liệu	ok
6	Kiểm tra sự kiện MouseClicked		Hiện thị lại thông tin Chức vụ lên form	ok
7	Tìm kiếm theo tiêu chí mã chức vụ, tên chức vụ	CV0001, Trưởng phòng	Hiện thị “Chức vụ” theo các tiêu chí và đổ lên Table	ok
8	Tìm kiếm theo tiêu chí mã chức vụ, tên chức vụ	bỏ trống/ Nhập sai	Đổ tất cả “Chức vụ” đang có lên Table	ok
9	Thêm	Nhập sai/ bỏ trống	Thông báo lỗi, Không thêm được	ok
10	Thêm	CV005, Chuyên Viên, Chuyên viên phụ trách một lĩnh vực cụ thể	Thông báo thêm thành công	ok
11	Sửa	Nhập sai/bỏ trống	Thông báo lỗi, Không sửa được	ok
12	Sửa	thông tin đúng	Thông báo cập nhật thành công	ok
13	Click từ Table qua form Xóa	Sửa mã chức vụ sai	Thông báo xóa thất bại	ok
14	Click từ Table qua form Xóa		Thông báo xóa thành công	ok
15	Reset		Tất cả dữ liệu đã có trong Textfield chuyển thành null	ok

4.5 KIỂM THỬ FORM ỨNG VIÊN

TH	MÔ TẢ TÌNH HUỐNG	DỮ LIỆU MẪU	KẾT QUẢ	FIX
1	Mở form		Đổ dữ liệu vào Table. Đổ dữ liệu vào ComboBox. Form hiện thị Ứng	ok

			viên hàng đầu tiên trong Table.	
2	Đề trống form		Yêu cầu nhập	ok
3	Kiểm tra trùng mã ứng viên	UV001	Thông báo trùng	ok
4	Kiểm tra các định dạng	-1000,sdf@,...	Thông báo lỗi	ok
5	Nút điều hướng		Form thay đổi dữ liệu	ok
6	Kiểm tra sự kiện MouseClicked		Hiện thị thông tin Ứng viên lên form	ok
7	Tìm kiếm theo tiêu chí mã ứng viên, họ tên ứng viên, mã chức vụ	UV001, Nguyễn Văn Thắng, CV001	Hiện thị “Ứng viên” theo các tiêu chí và đổ lên Table	ok
8	Tìm kiếm theo tiêu chí mã ứng viên, họ tên ứng viên, mã chức vụ	bỏ trống/ Nhập sai	Đổ tất cả “Ứng viên” đang có lên Table	ok
9	Thêm	Nhập sai/ bỏ trống	Thông báo lỗi, Không thêm được	ok
10	Thêm	UV015, Hồ Văn Tình, 0368852147, 789 Đường UVW Quận 3 TP.HCM, 1988-12-10, 1, hov3an.y@example.com, Ứng viên cho vị trí Nhân Viên Kinh Doanh, CV003	Thông báo thêm thành công	ok
11	Sửa	Nhập sai/bỏ trống	Thông báo lỗi, Không sửa được	ok
12	Sửa	thông tin đúng	Thông báo cập nhật thành công	ok
13	Click từ Table qua form	Sửa mã ứng viên	Thông báo xóa thất	ok

	Xóa	sai	bại	
14	Click từ Table qua form Xóa		Thông báo xóa thành công	ok
15	Reset		Tất cả dữ liệu đã có trong Textfield chuyển thành null	ok
16	Thêm ứng viên này thành nhân viên	UV011, Trần Văn Tâm, 0369952147, 789 Đường UVW Quận 6 TP.HCM, 1990-08-10, 1, tranv3ant@exam ple.com, CV003	Hiện thị form Nhân Viên với các thông tin tương ứng.	ok

4.6 KIỂM THỬ FORM ĐÔI MẬT KHẨU

TH	MÔ TẢ TÌNH HUỐNG	DỮ LIỆU MẪU	KẾT QUẢ	FIX
1	Để trống form		Yêu cầu nhập	ok
2	Kiểm tra không trùng tên tài khoản	us001	Thông báo tên tài khoản phải trùng với tên đang đăng nhập.	ok
3	Kiểm tra trường dữ liệu mật khẩu hiện tại không đúng	sdfh	Yêu cầu nhập lại mật khẩu hiện tại	ok
4	Kiểm tra trường dữ liệu mật khẩu mới và xác nhận mật khẩu mới không đúng	123456, sdf	Yêu cầu nhập lại xác nhận mật khẩu mới	ok
5	Kiểm tra các trường dữ liệu mật khẩu đúng	111111, 123456, 123456	Hiện thị thông báo “Đổi mật khẩu thành công”.	ok

4.7 KIỂM THỬ FORM HỢP ĐỒNG

TH	MÔ TẢ TÌNH HUỐNG	DỮ LIỆU MẪU	KẾT QUẢ	FIX
----	------------------	-------------	---------	-----

1	Mở form		Đổ dữ liệu vào Table. Đổ dữ liệu vào ComboBox. Form hiển thị Loại hợp đồng hàng đầu tiên trong Table.	ok
2	Để trống form		Yêu cầu nhập	ok
3	Nút điều hướng		Form thay đổi dữ liệu	ok
4	Kiểm tra sự kiện MouseClicked		Hiển thị lại thông tin Tài khoản lên form	ok

4.8 KIỂM THỬ FORM PHÒNG BAN

TH	MÔ TẢ TÌNH HUỐNG	DỮ LIỆU MẪU	KẾT QUẢ	FIX
1	Mở form		Đổ dữ liệu vào Table. Đổ dữ liệu vào ComboBox. Form hiển thị Phòng ban hàng đầu tiên trong Table.	ok
2	Để trống form		Yêu cầu nhập	ok
3	Kiểm tra trùng mã phòng ban	PB001	Thông báo trùng	ok
4	Kiểm tra các định dạng	PB0233, NV87763	Thông báo lỗi	ok
5	Nút điều hướng		Form thay đổi dữ liệu	ok
6	Kiểm tra sự kiện MouseClicked		Hiển thị thông tin Phòng ban lên form	ok
7	Tìm kiếm theo tiêu chí mã phòng ban, tên phòng ban, mã trưởng phòng	PB001, Phòng Nhân sự, NV001	Hiển thị “Phòng ban” theo từng tiêu chí và đổ lên Table	ok
8	Tìm kiếm theo tiêu chí mã phòng ban, tên phòng ban, mã trưởng phòng	bỏ trống/ Nhập sai	Đổ tất cả “Phòng ban” đang có lên Table	ok

9	Thêm	Nhập sai/ bỏ trống	Thông báo lỗi, Không thêm được	ok
10	Thêm	PB005, Phòng IT, Phòng chuyên về công nghệ thông tin, NV009	Thông báo thêm thành công	ok
11	Sửa	Nhập sai/bỏ trống	Thông báo lỗi, Không sửa được	ok
12	Sửa	thông tin đúng	Thông báo cập nhật thành công	ok
13	Click từ Table qua form Xóa	Sửa mã phòng ban sai	Thông báo xóa thất bại	ok
14	Click từ Table qua form Xóa		Thông báo xóa thành công	ok
15	Reset		Tất cả dữ liệu đã có trong Textfield chuyển thành null	ok

4.9 KIỂM THỬ FORM THỐNG KÊ

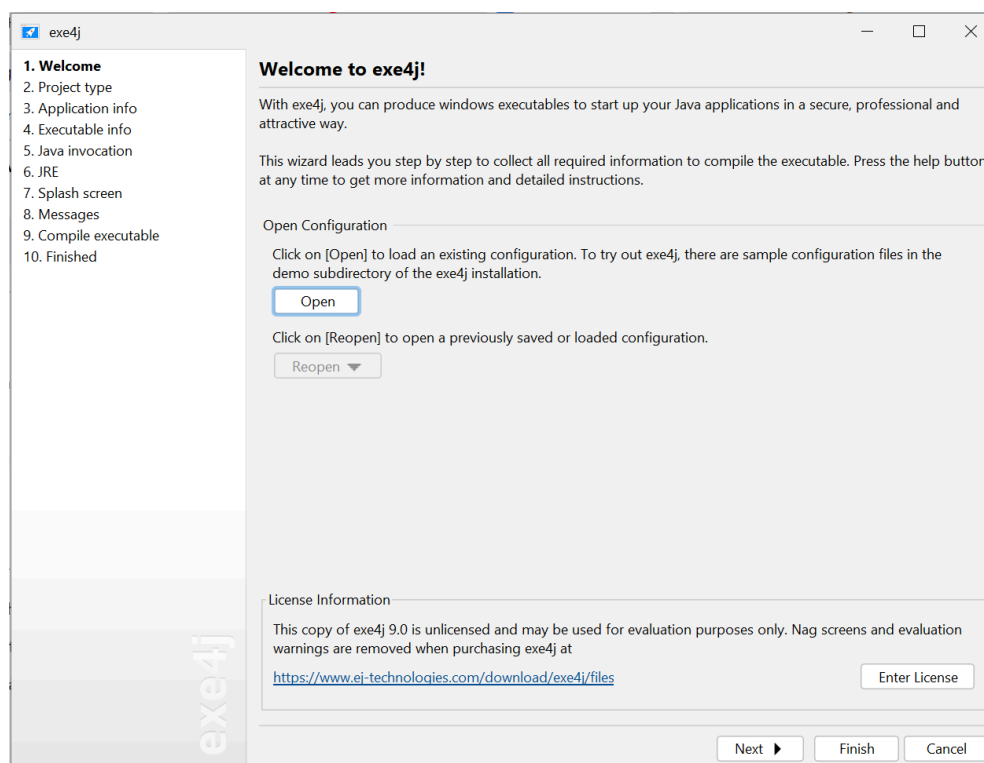
TH	MÔ TẢ TÌNH HUỐNG	DỮ LIỆU MẪU	KẾT QUẢ	FIX
1	Mở form		Đổ dữ liệu vào Table của các Tab. Đổ dữ liệu vào ComboBox của các Tab.	ok
2	Tab Ứng viên: Tìm kiếm theo tên chức vụ từ ComboBox	Trưởng phòng	Hiển thị tổng số lượng ứng viên đã nộp vào chức vụ này, số lượng ứng viên đậu, số lượng ứng viên rớt. Đổ lên Table	ok
3	Tab Độ tuổi: Tìm kiếm theo khoảng tuổi	18-60	Hiển thị số lượng nhân viên nằm trong khoảng tuổi này.	ok

			Đổ lên Table	
4	Tab Lương: Tìm kiếm theo khoảng lương	5.000.000-20.000.000	Hiển thị số lượng nhân viên nằm trong khoảng lương này. Đổ lên Table	ok
5	Tab Hợp đồng		Hiển thị số lượng nhân viên theo từng loại hợp đồng. Đổ lên Table	ok
6	Tab Phòng ban: Tìm kiếm theo tên phòng ban từ ComboBox	Phòng Nhân sự	Hiển thị số lượng nhân viên, số lượng chức vụ, và họ tên của phòng ban đó. Đổ lên Table	ok

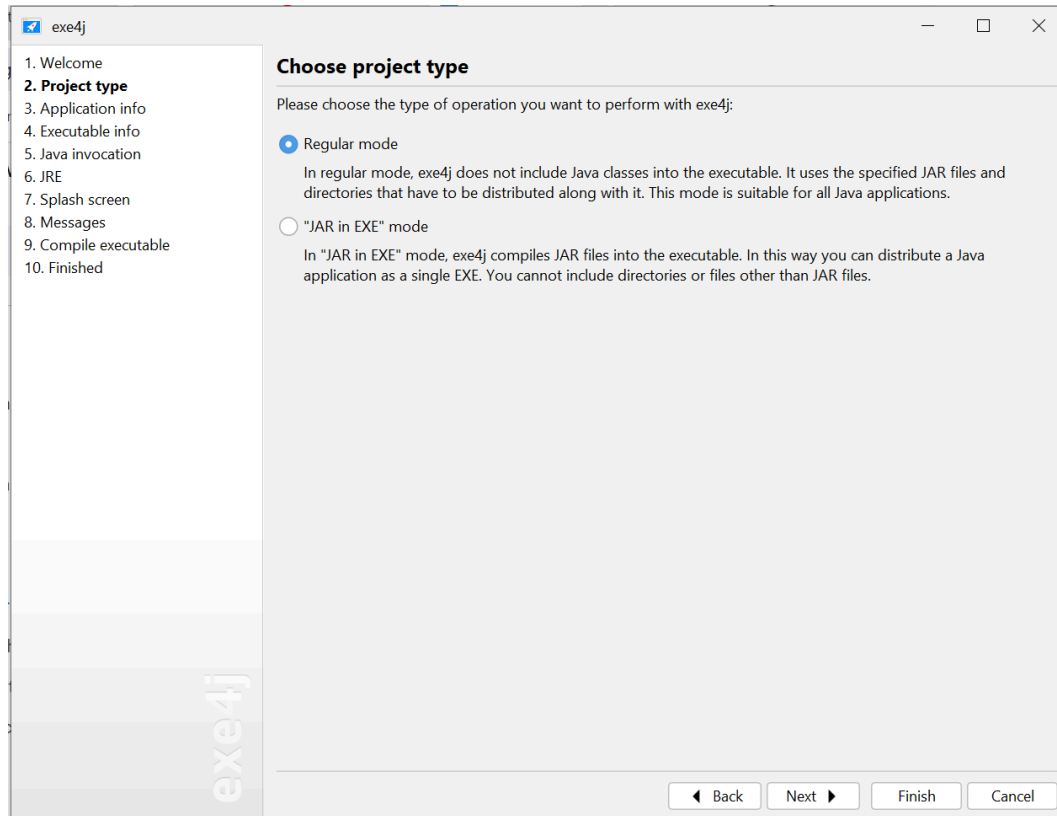
5 ĐÓNG GÓI VÀ TRIỂN KHAI

5.1 HƯỚNG DẪN CHUYỂN ĐỔI JAR THÀNH EXE

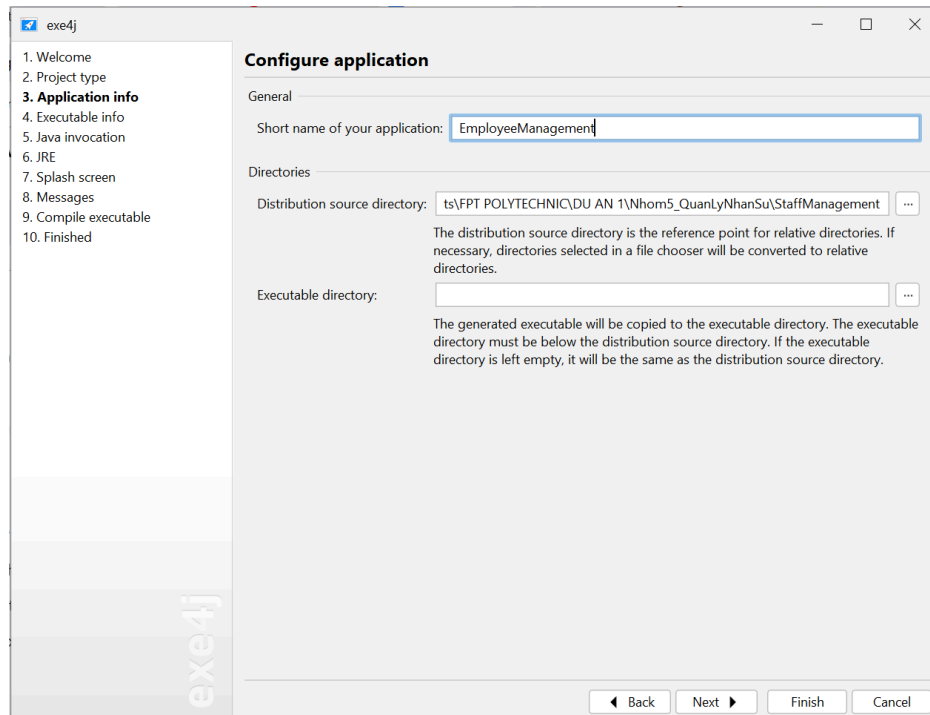
- Tải exe4j, chạy chương trình, chọn Next.



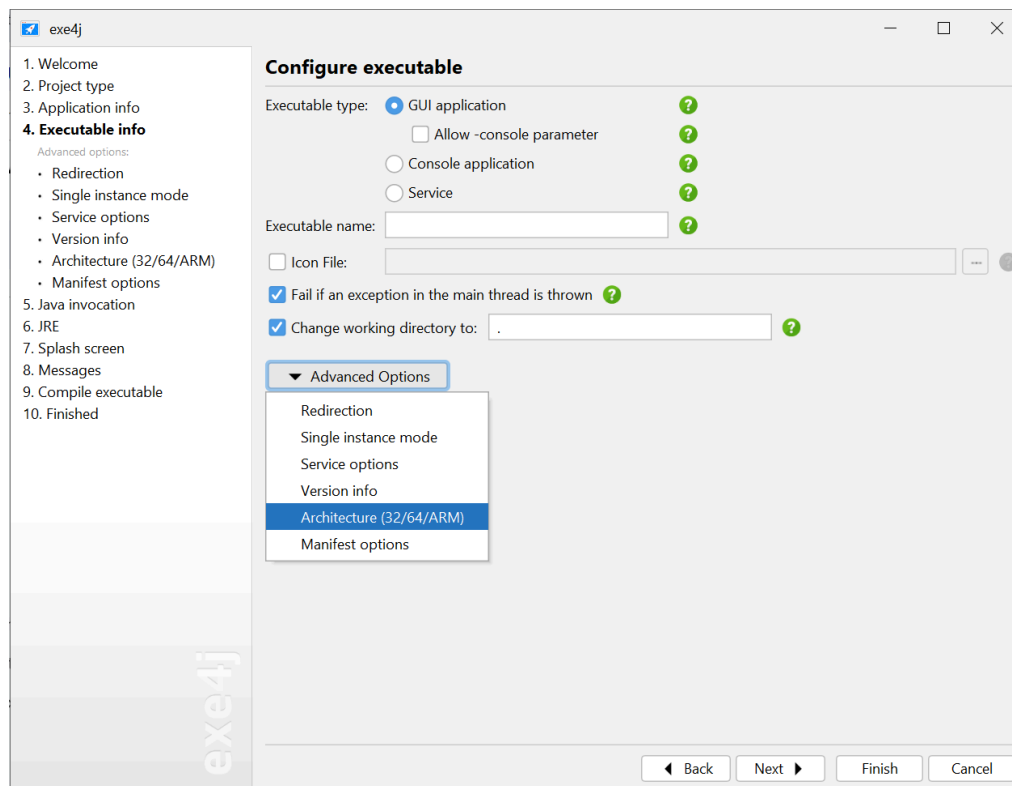
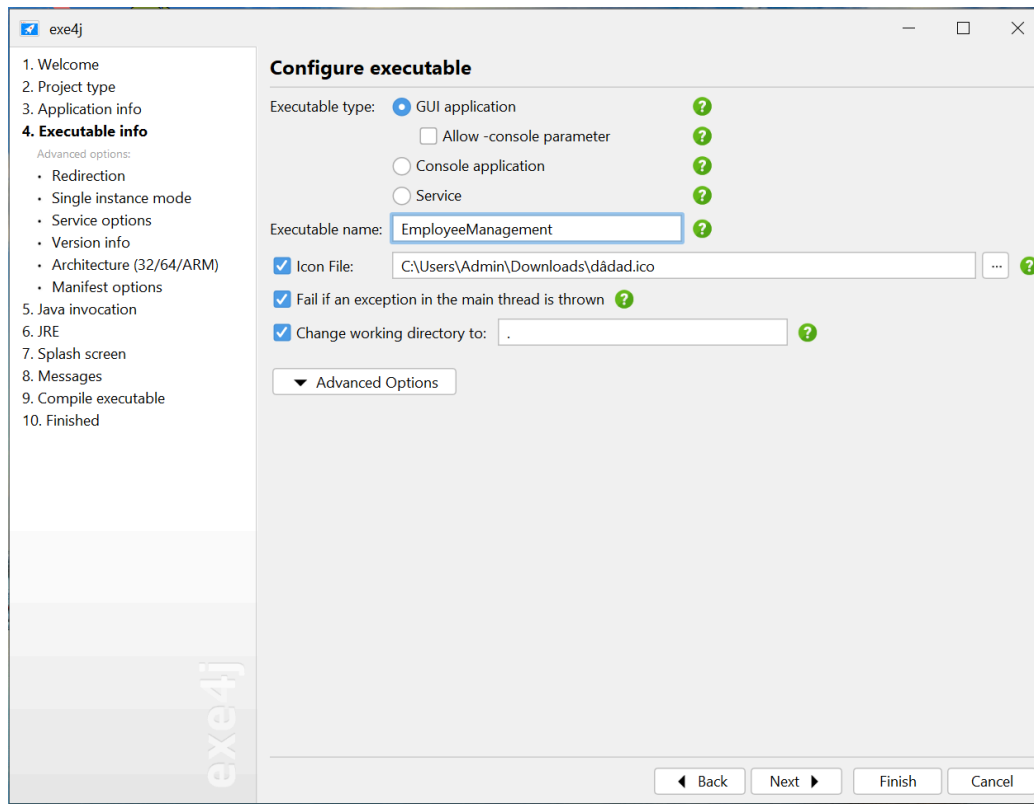
- Chọn Next



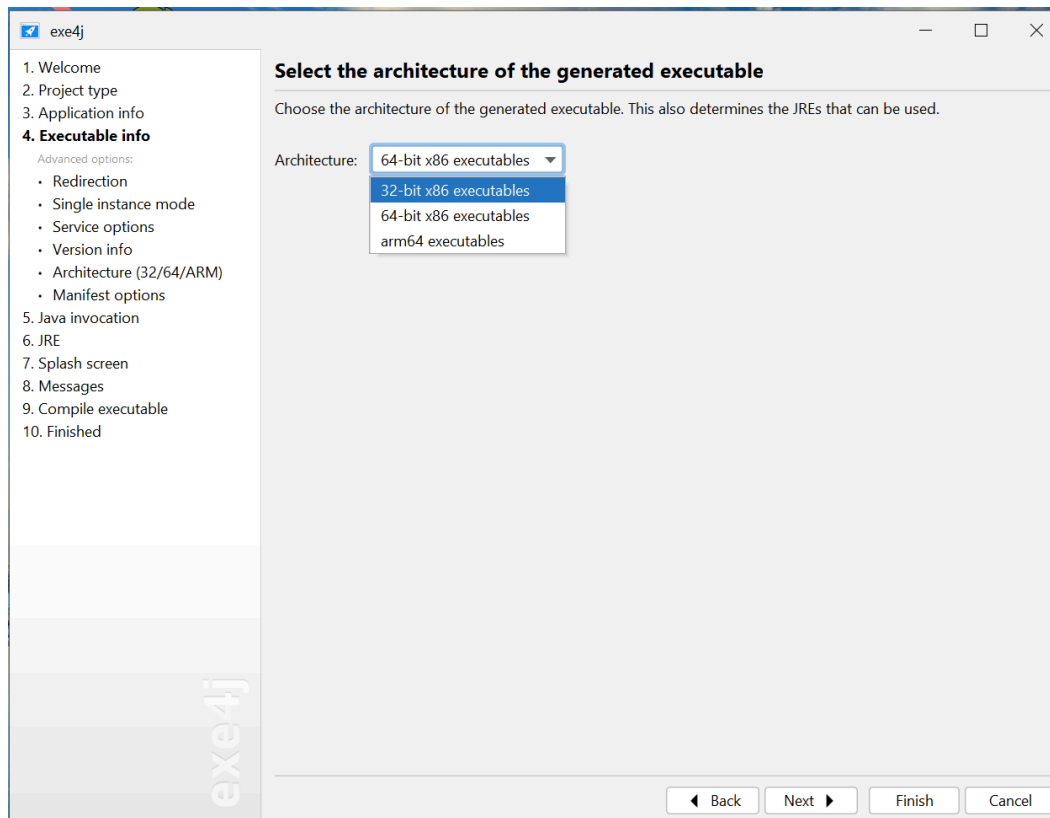
- Đặt tên ngắn cho file, chọn đường folder chứa file jar, chọn Next.



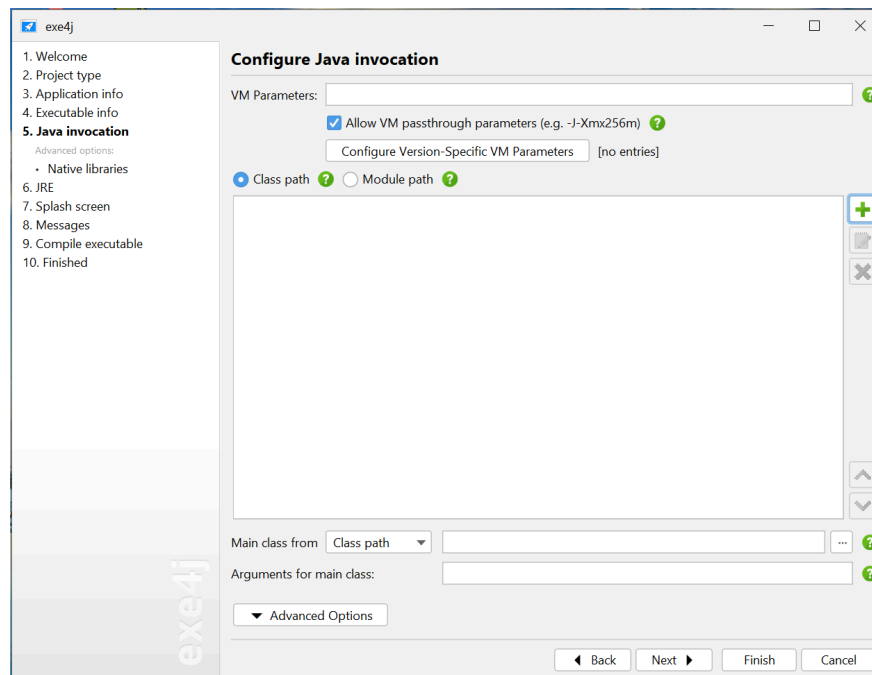
- Nhập tên exe, chọn icon cho file. Chọn Advance → Architecture.



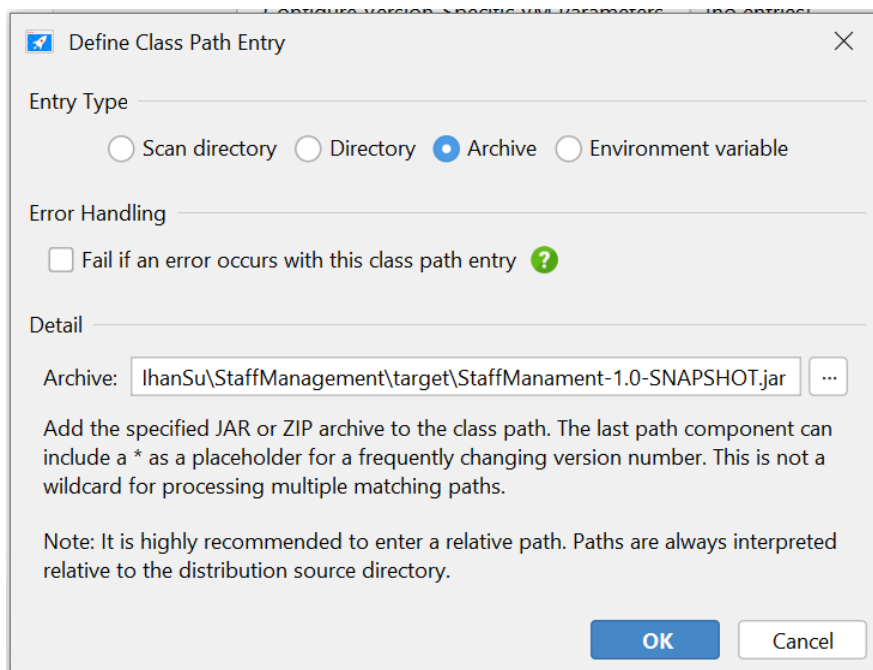
- Chọn 64-bit, chọn Next



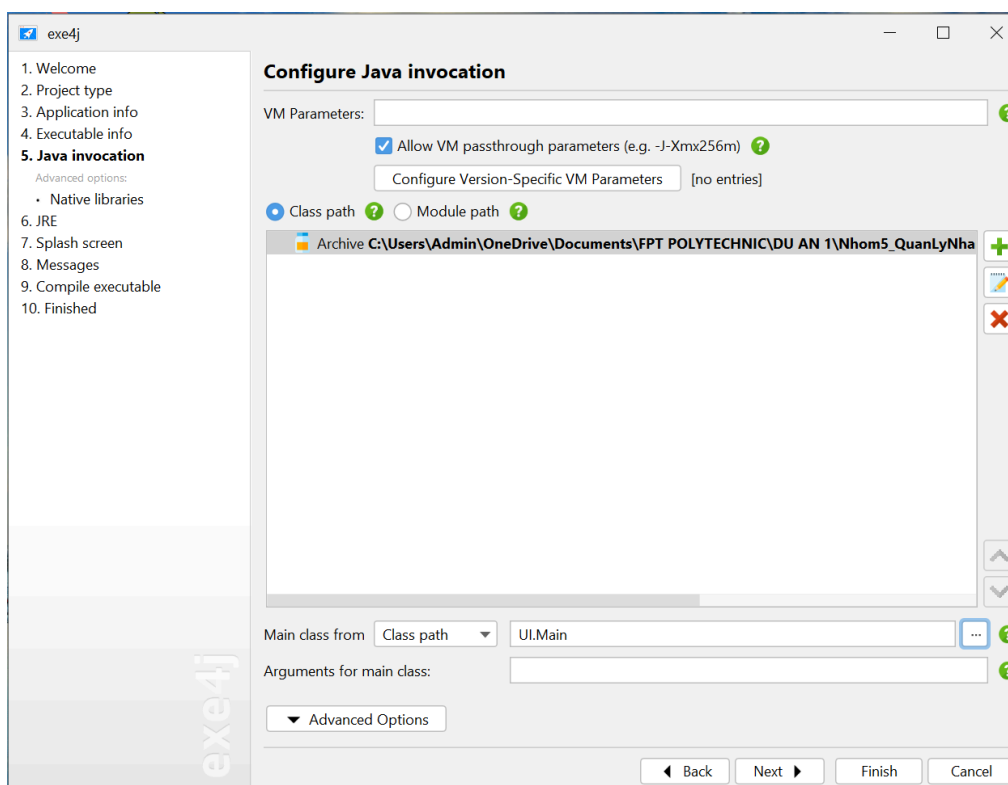
- Chọn +



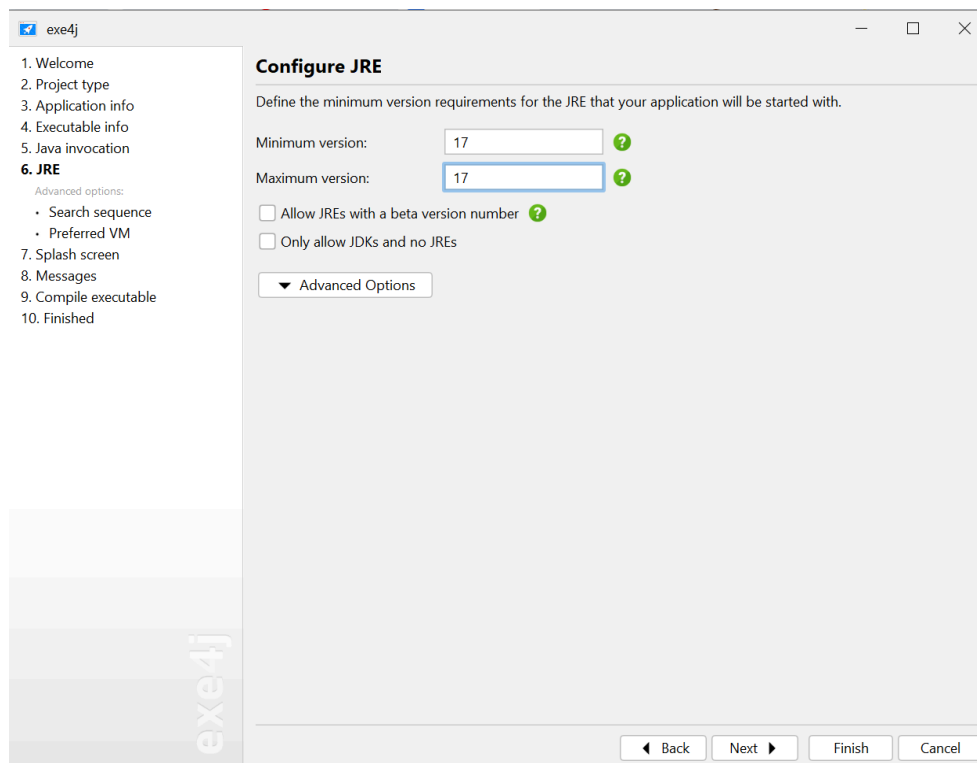
- Chọn file jar



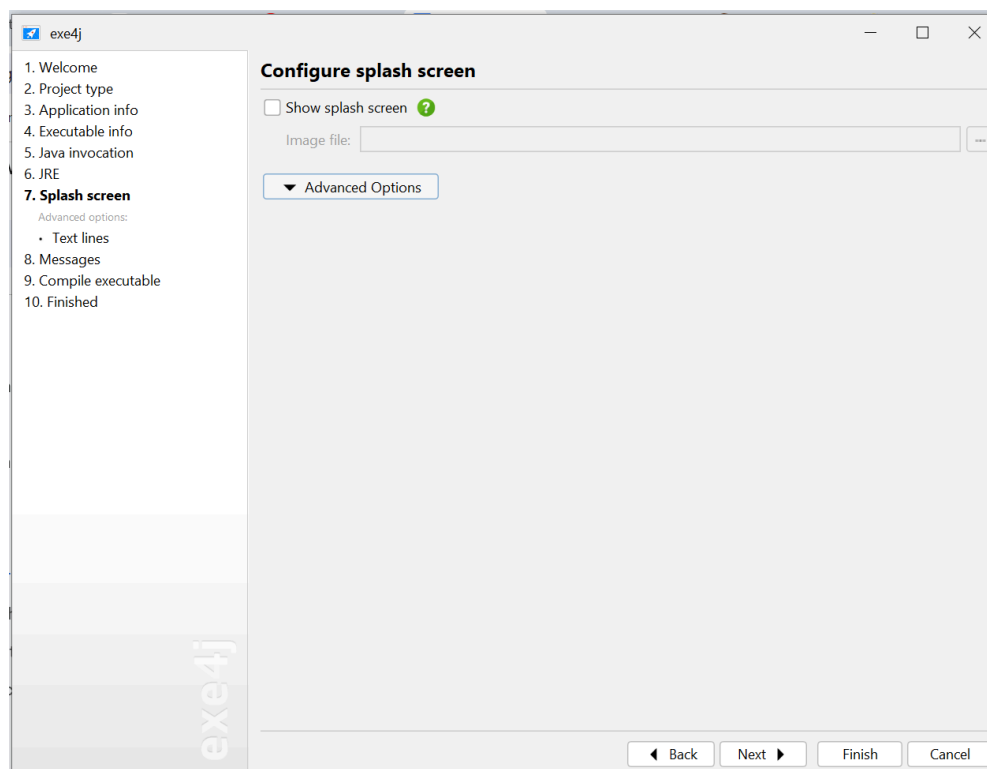
- Chọn Class chạy chính, chọn Next



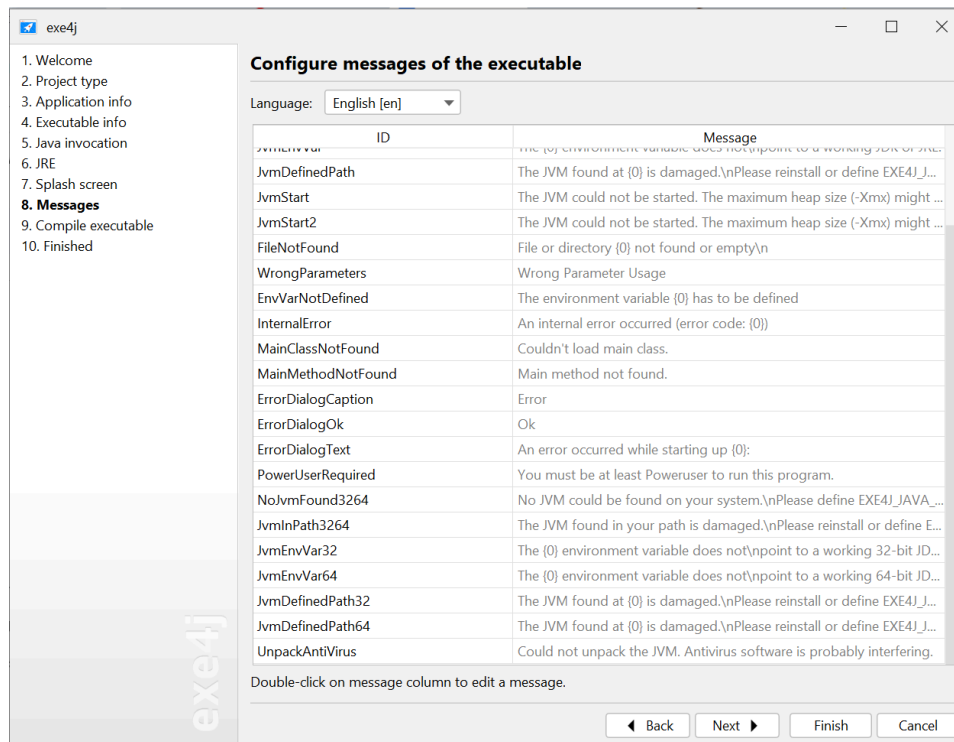
- Chọn version JRE, chọn Next



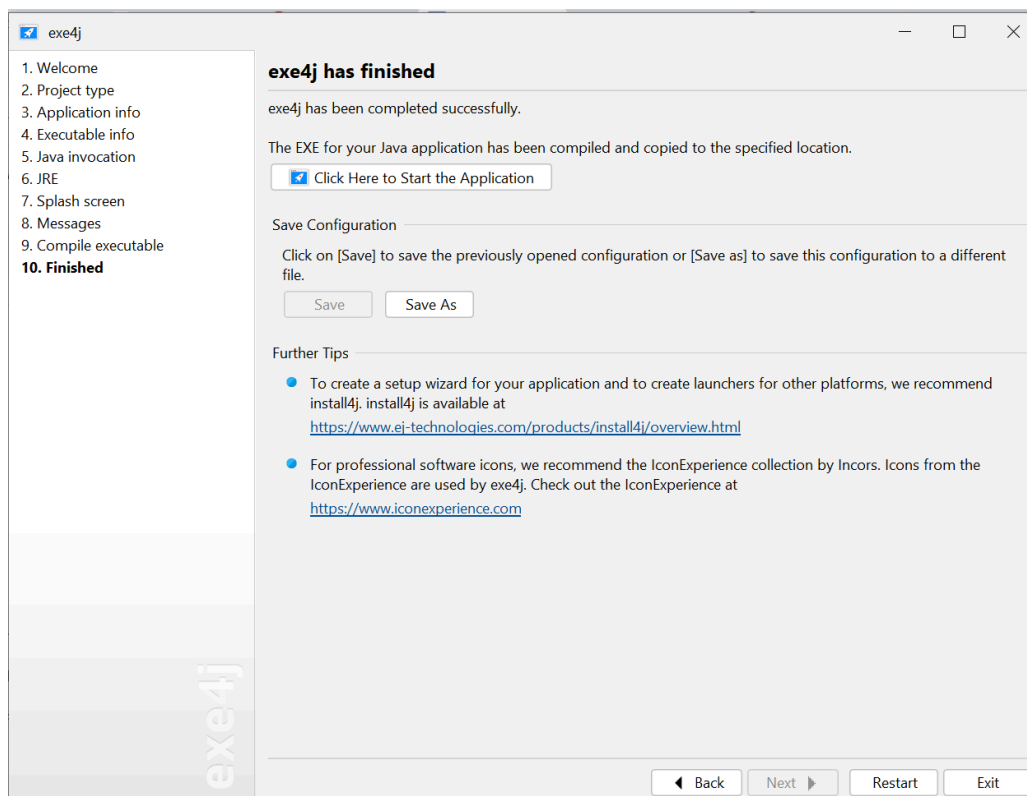
- Vì phần mềm đã có splash_screen, nên chọn Next



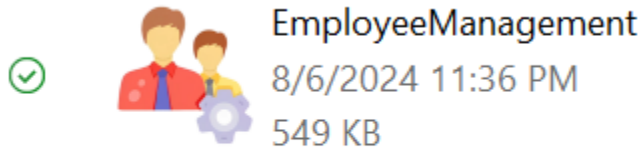
- Chọn next



- Sau khi chạy xong, nhấn Exit



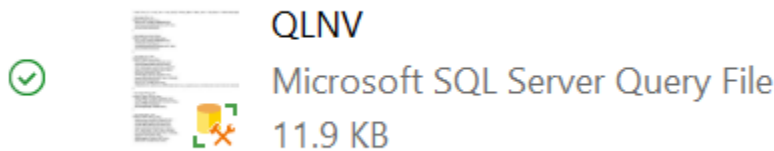
- file exe



5.2 HƯỚNG DẪN CÀI ĐẶT TRIỂN KHAI

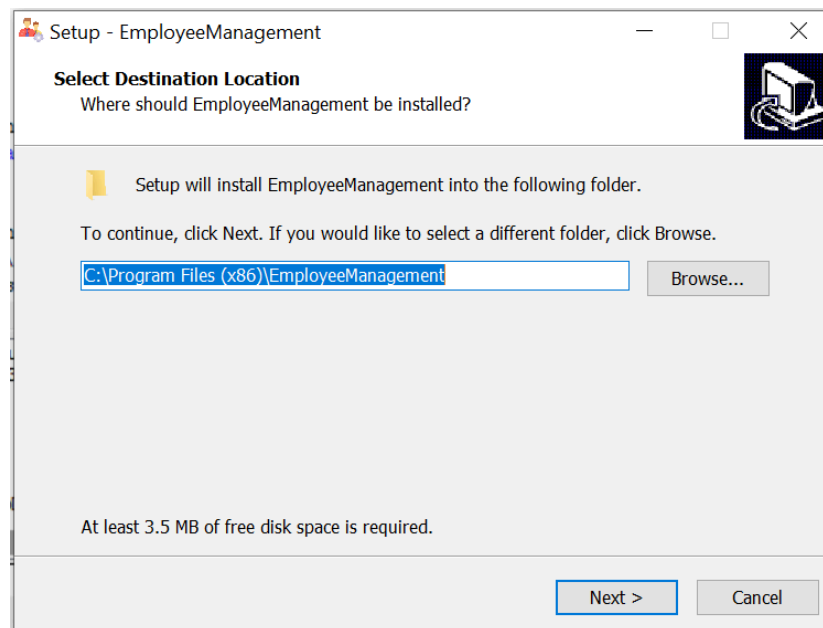
5.2.1 Server side: Restore database, cấu hình username password cho CSDL.

Tải SQL Server, tạo tài khoản đăng nhập có username là “usr01_team5” và password là “qlnhanvien”, sau đó tiến hành chạy file “QLNV.sql” để tạo database và dữ liệu.

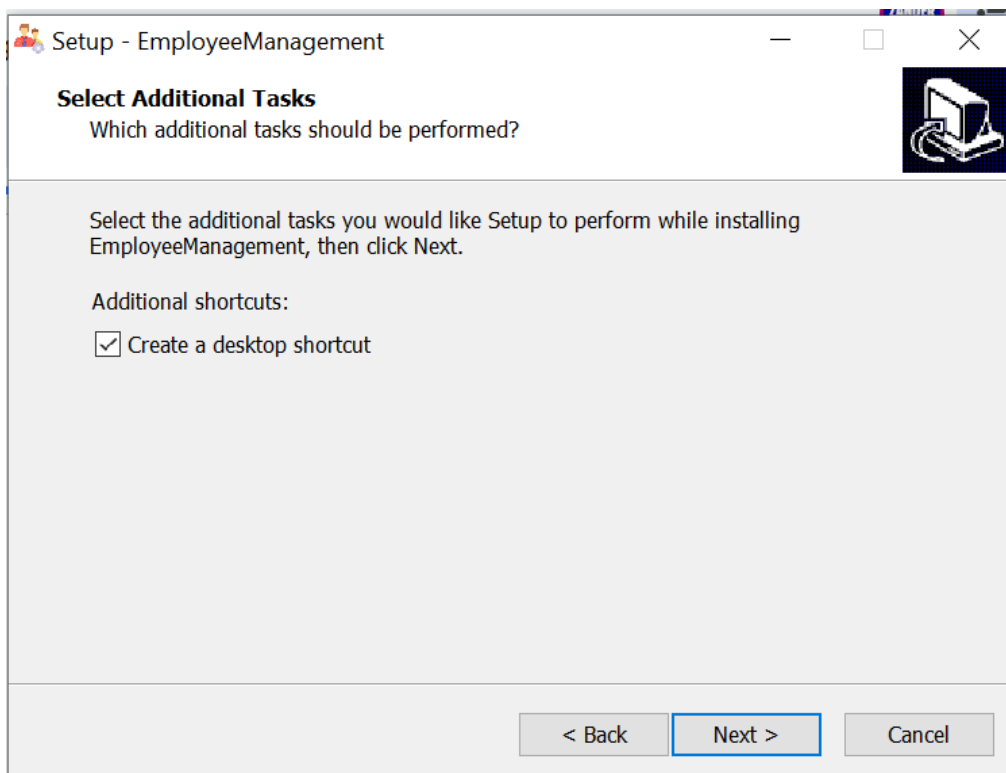


5.2.2 Client side: Các bước cài đặt phần mềm

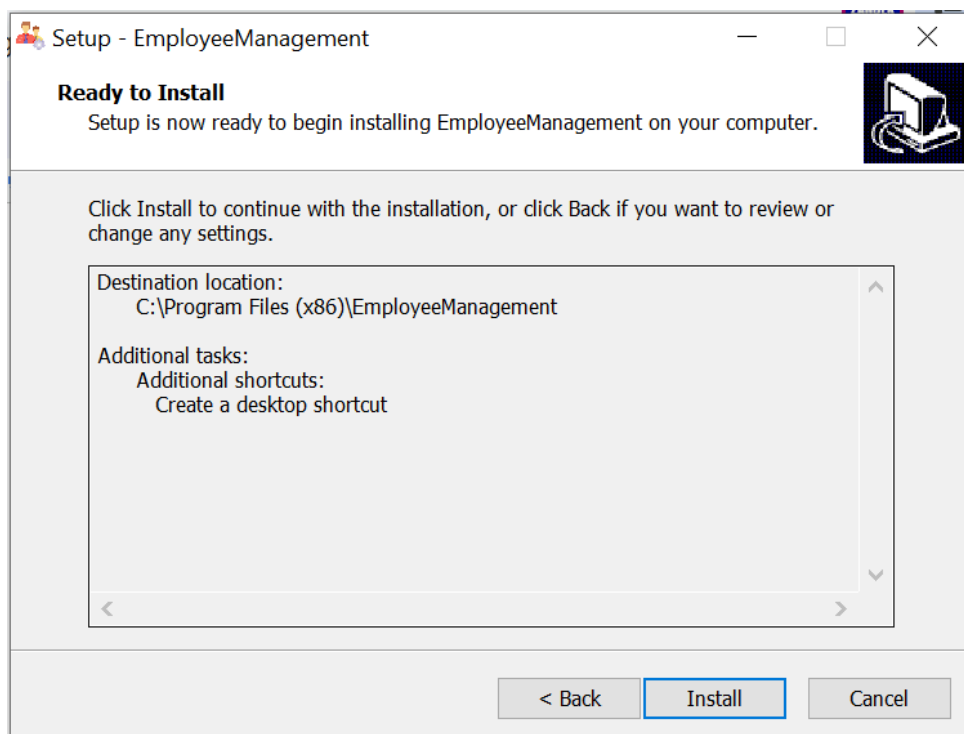
- Sau khi tạo cơ sở dữ liệu thành công, tiến hành chạy file “QuanLyNhanSu.exe” để cài đặt ứng dụng.
- Lựa chọn vị trí cài đặt ứng dụng



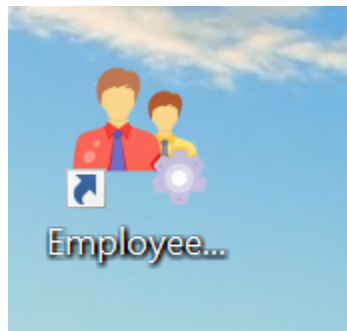
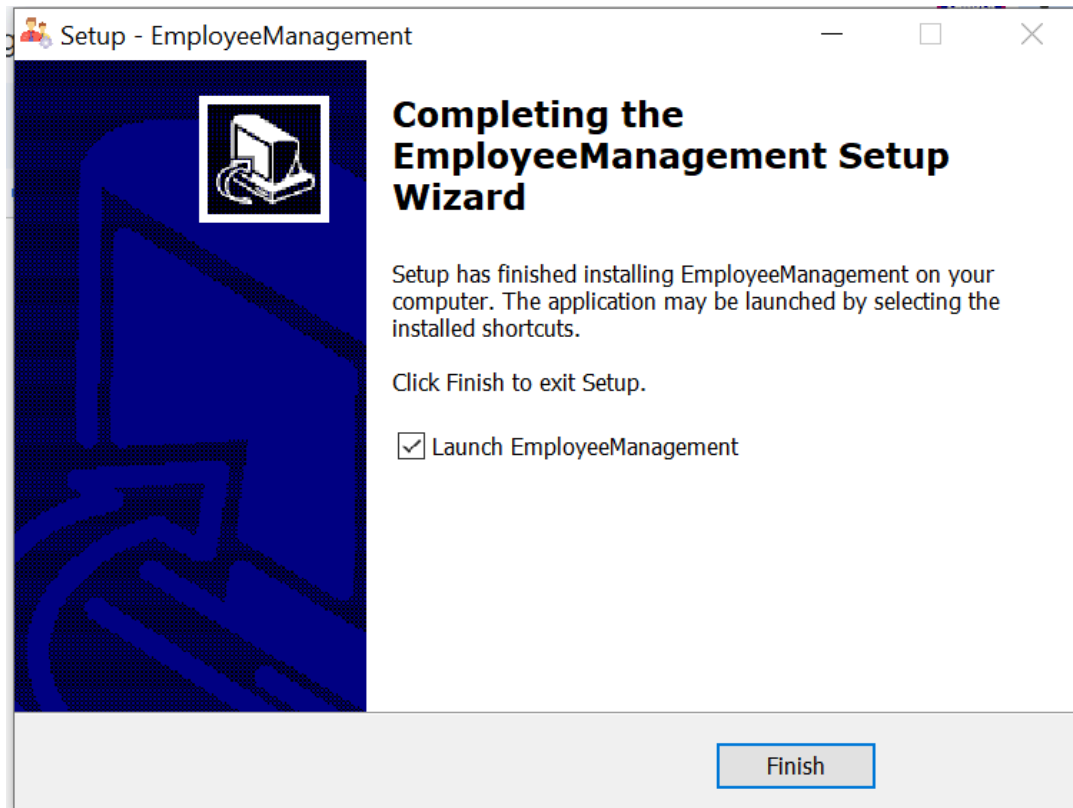
- Lựa chọn tạo shortcut trên desktop



- Chọn Install

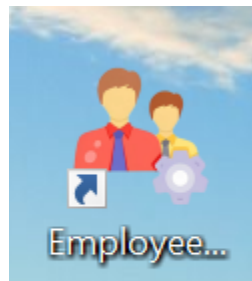


- Finish và ngoài desktop sẽ hiển thị ứng dụng



5.3 HƯỚNG DẪN SỬ DỤNG PHẦN MỀM

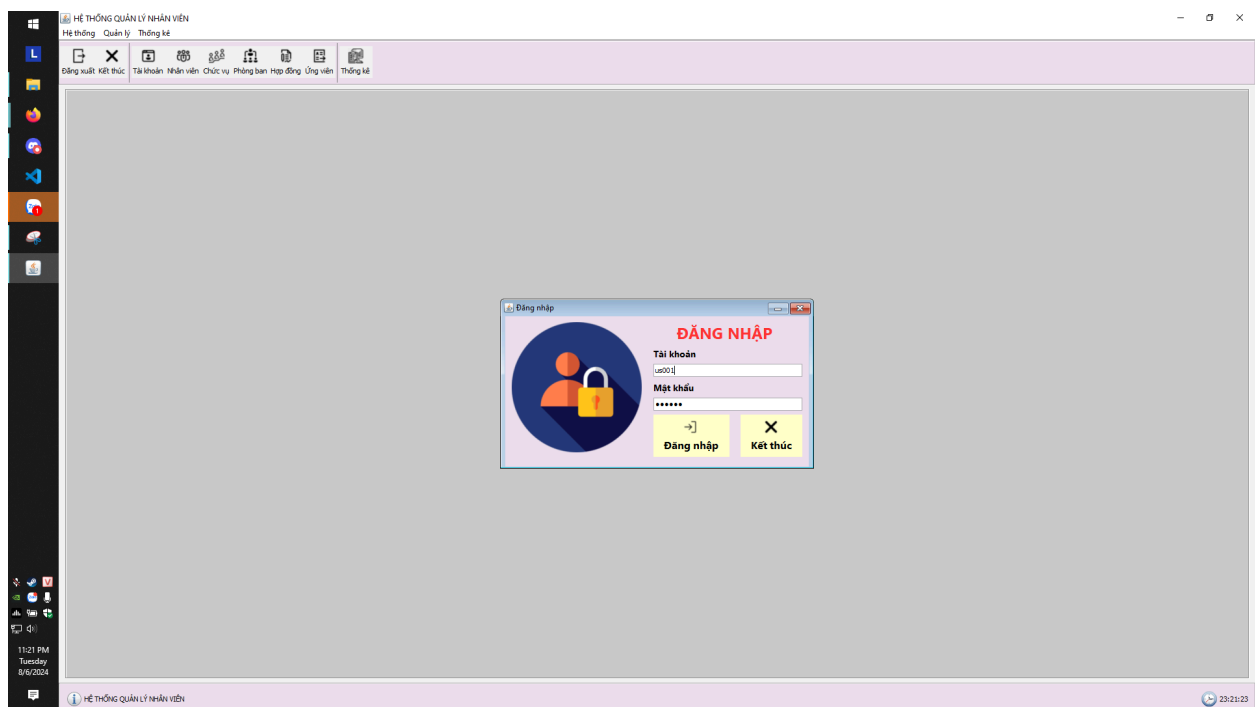
- Sau khi cài đặt ứng dụng thành công, click vào biểu tượng ứng dụng TroViet trên desktop để khởi chạy ứng dụng



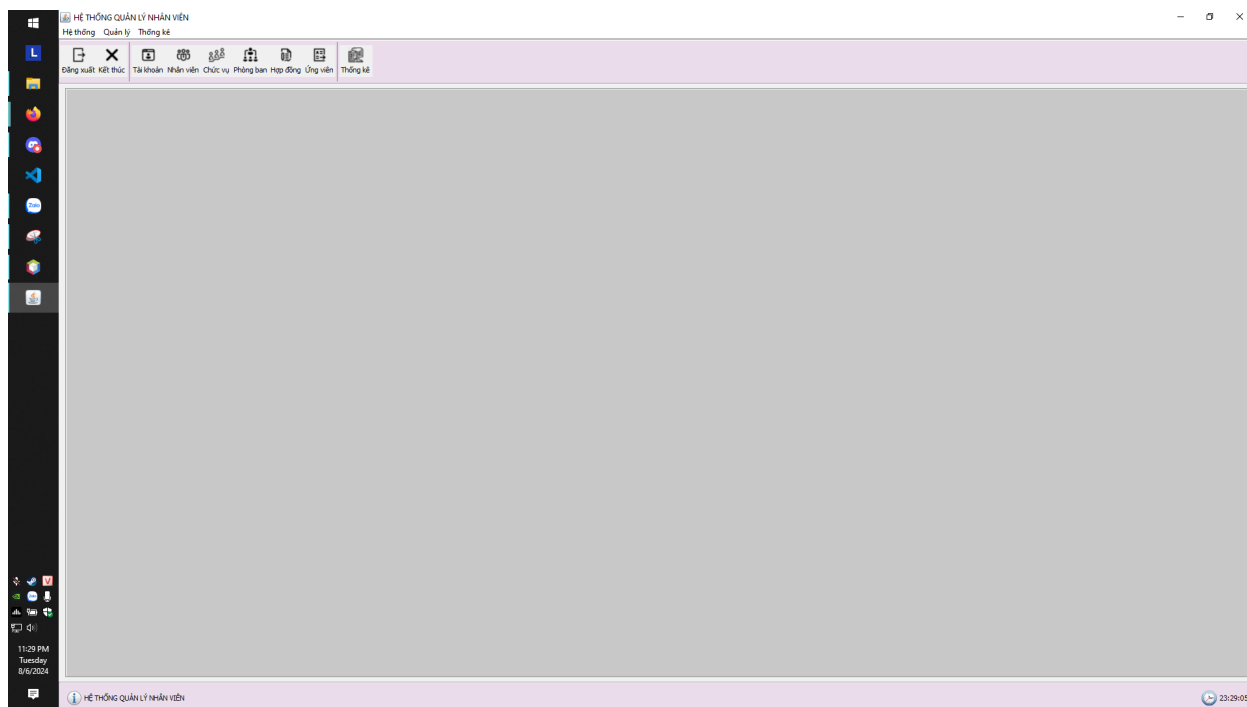
- Khởi động phần mềm, đợi phần mềm load dữ liệu từ database



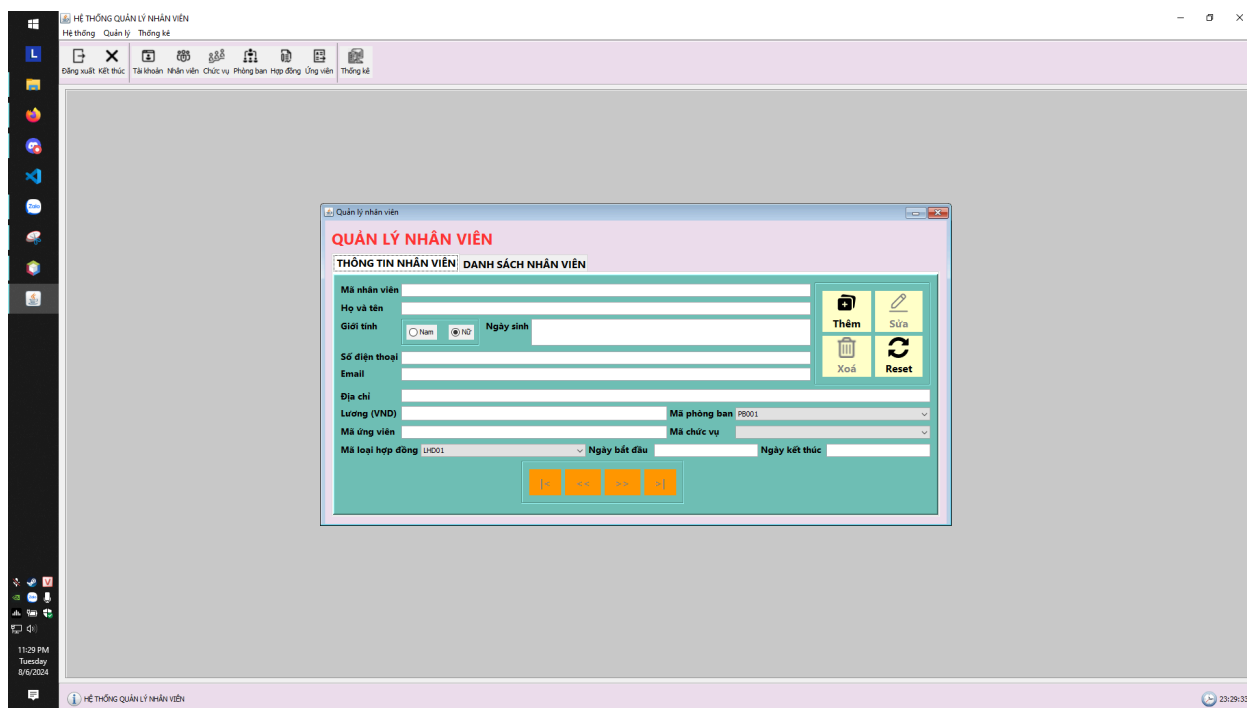
- Sau khi màn hình chính hiện lên, sẽ hiện lên form đăng nhập, đăng nhập theo tài khoản mà bạn có



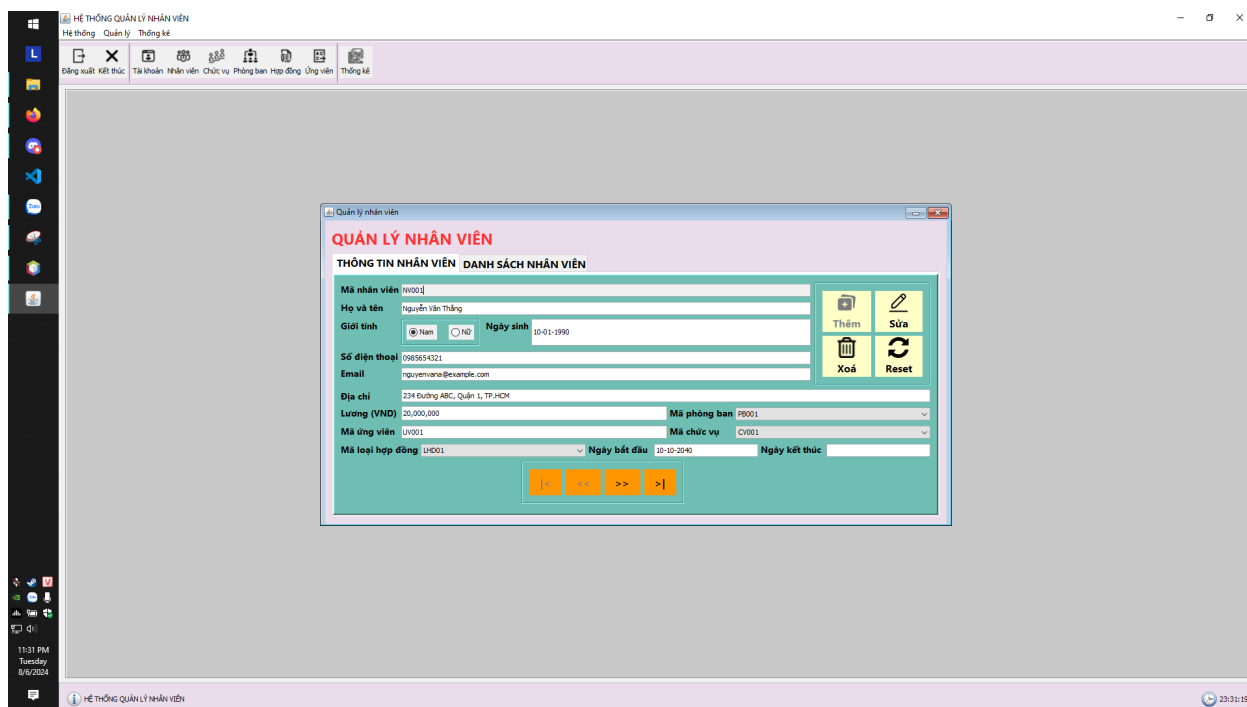
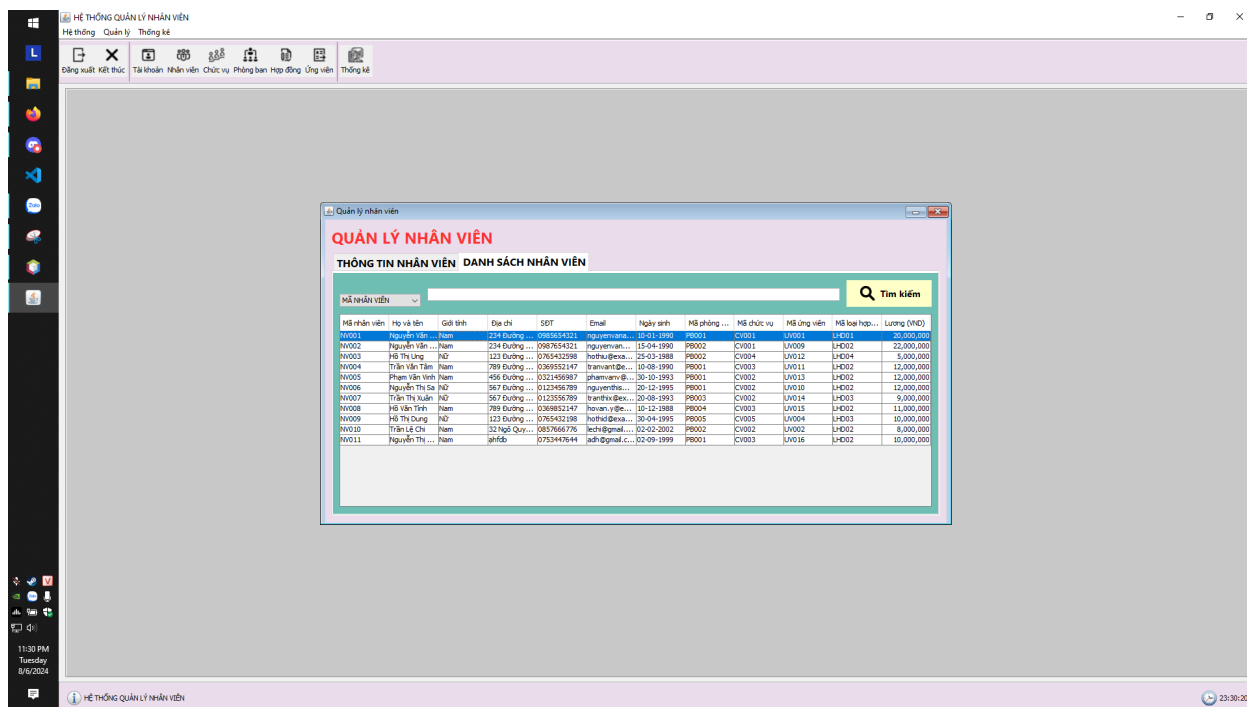
- Sau khi đã đăng nhập vào hệ thống bạn sẽ được sử dụng các chức năng chính như là quản lý nhân viên, chức vụ, thống kê,...



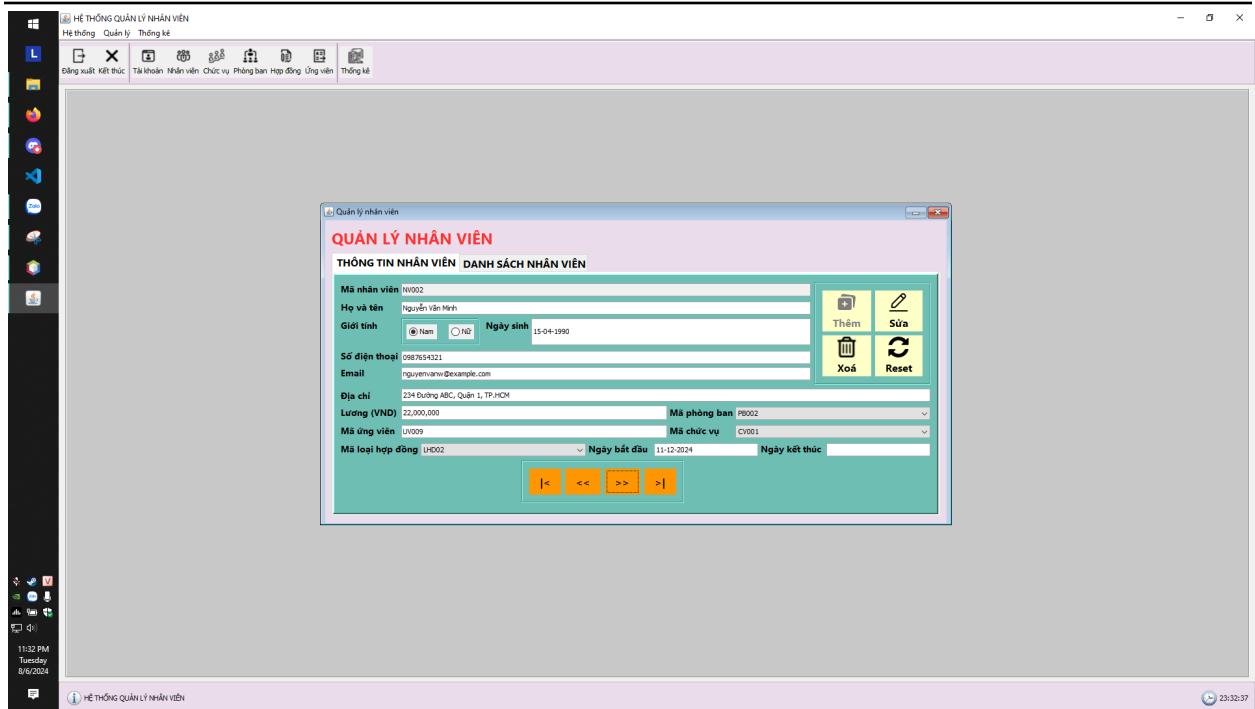
- Mỗi chức năng sẽ có 1 hoặc nhiều tab để làm việc



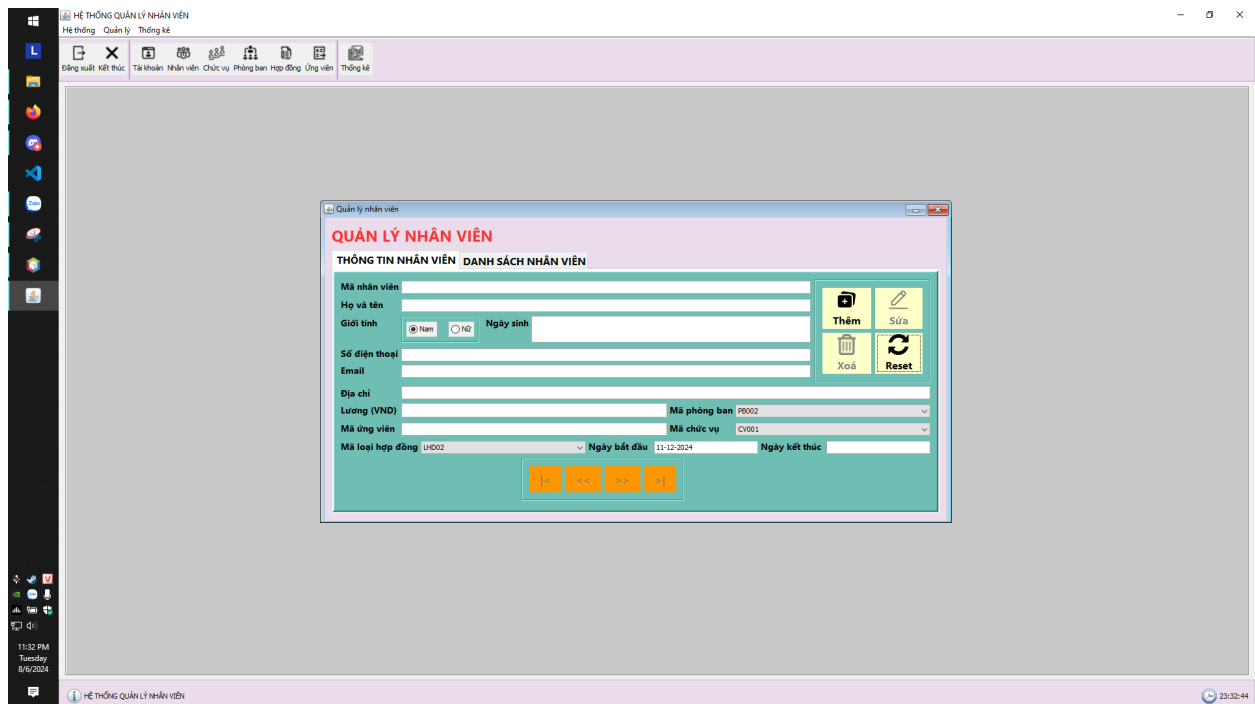
- Chuyển sang phần danh sách nhân viên, khi ta double click vào một nhân viên nào đó, thông tin của nhân viên sẽ được tự động fill lên form



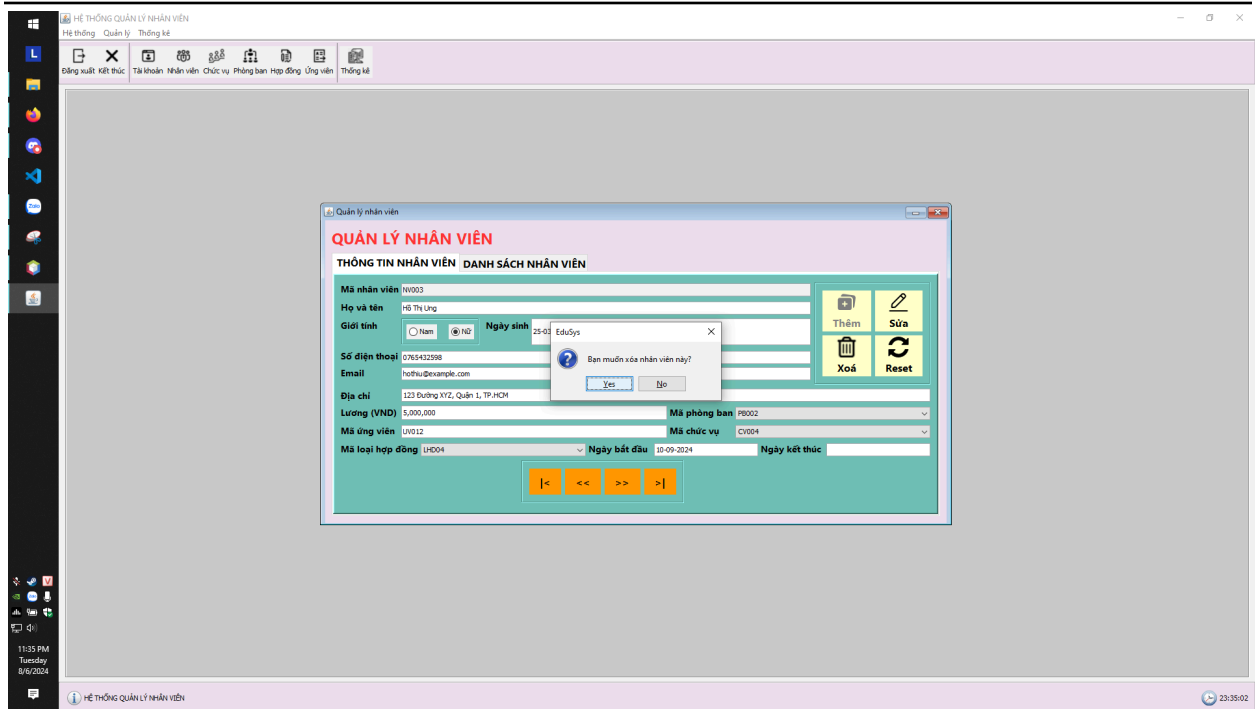
- Nhấn tiến tới để hiện thông tin nhân viên tiếp theo, lùi, đầu tiên, cuối cùng đều sẽ thực hiện tương tự theo hướng của button



- Nhấn vào nút reset sẽ clear mọi thông tin đang có trên form



- Xóa: khi đang nhân viên đang được fill lên form, nhấn nút xóa sẽ xóa đi nhân viên ấy khỏi bảng danh sách nhân viên



- Thêm: Sẽ thêm nhân viên mới vào không trùng với mã nhân viên cũ hiện đang có trong danh sách
- Sửa: Sau khi fill thông tin của nhân viên đó lên bảng, tiến hành sửa thông tin trong ô, và ấn Sửa sẽ thay đổi thông tin của nhân viên đó
- Các tính năng còn lại đều tương tự như nhau
- Đăng xuất: khi ấn vào sẽ ngay lập tức tắt ngay mọi form đang hiện và đăng xuất tài khoản ra ngoài màn hình đăng nhập

