

**ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA ĐÀ NẴNG
KHOA ĐIỆN TỬ- VIỄN THÔNG**



THUYẾT MINH

MÔN: PBL1-LẬP TRÌNH

ĐỀ TÀI: QUẢN LÝ NHÂN SỰ

Giảng viên: TS. NGUYỄN DUY NHẬT VIỄN

Sinh viên thực hiện (Nhóm 7-Lớp học phần:22.44A):

1.Nguyễn Xuân Trường Lớp sinh hoạt: 22KTMT2 MSSV:106220275

2.Bùi Trịnh Thế Viên Lớp sinh hoạt: 22KTMT2 MSSV:106220276

ĐÀ NẴNG, NGÀY THÁNG NĂM 2023

MỤC LỤC

Lời nói đầu.....	3
Phân công thành viên trong nhóm.....	4
Khảo sát yêu cầu của bài toán.....	5
Chương 1. Khảo sát, đặc tả yêu cầu dự án.....	6
1.1. Mô tả yêu cầu bài toán.....	6
1.2. Biểu đồ Use-case.....	6
1.2.1. Biểu đồ Use-case tổng quan.....	6
1.2.1.1. Biểu đồ Use-case đăng nhập.....	7
1.2.1.2. Biểu đồ Use-case quản lý nhân sự.....	7
1.2.1.3. Biểu đồ Use-case quản lý phòng ban.....	8
1.2.1.4. Biểu đồ Use-case quản lý task.....	8
1.2.2. Đặc tả biểu đồ Use-case.....	9
1.2.2.1. Đặc tả biểu đồ Use-case đăng nhập.....	9
1.2.2.2. Đặc tả biểu đồ Use-case quản lý nhân sự.....	10
1.2.2.3. Đặc tả biểu đồ Use-case quản lý phòng ban.....	11
1.2.2.4. Đặc tả biểu đồ Use-case quản lý task.....	13
Chương 2. Phân tích thiết kế bài toán.....	14
2.1. Thiết kế cơ sở dữ liệu.....	14
2.2. Biểu đồ lớp.....	15
2.3. Thiết kế chi tiết lớp.....	15
Chương 3. Công nghệ và thuật toán sử dụng.....	19
3.1. Công nghệ sử dụng.....	19
3.2. Thuật toán sử dụng.....	19
Chương 4. Xây dựng chương trình.....	19
4.1. Kết quả xây dựng chương trình.....	19
4.2. Giao diện chương trình.....	20
4.2.1. Giao diện đăng nhập.....	20
4.2.2. Giao diện quản lý nhân sự.....	23

4.2.3. Giao diện quản lý phòng ban.....	25
4.2.4. Giao diện quản lý task.....	27
4.3. Kiểm thử chương trình	30
TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN.....	35
TÀI LIỆU THAM KHẢO.....	36

Lời nói đầu

Số doanh nghiệp quay lại hoạt động một số năm gần đây gần như liên tục tăng lên. Theo Tổng cục Thống kê, số doanh nghiệp năm 2018 là 34.010, năm 2019 là 39.421, năm 2020 là 44.096, năm 2021 là 43.116 và ước năm 2022 là 59.835 doanh nghiệp. Như vậy, ngay cả năm 2020 khi đại dịch Covid-19 xảy ra và năm 2021 khi đại dịch bùng phát, số lượng doanh nghiệp quay trở lại hoạt động vẫn khá nhiều, nhiều hơn cả những năm trước đại dịch. Trong thời đại chuyển đổi số 4.0 ngày nay, cách quản trị nhân sự có những thay đổi vô cùng mạnh mẽ và nhanh chóng. Tuy nhiên, đi cùng với những thay đổi tích cực, đó cũng là những khó khăn mà doanh nghiệp gặp phải. Chính vì điều đó chúng em đã đưa ra ý tưởng xây dựng một phần mềm quản lý nhân sự cho các công ty và doanh nghiệp.

Nhân sự hay còn gọi là nhân viên của doanh nghiệp, đây là thành phần cốt lõi cho sự phát triển bền vững. Tuy nhiên, việc quản lý một cách hiệu quả không phải điều mà các doanh nghiệp có thể đảm bảo là tốt nhất. Chính vì điều đó, xây dựng phần mềm quản lý nhân sự cung cấp công cụ hiệu quả cho nhà quản trị. Nhờ phần mềm quản lý nhân sự thì việc quản lý được dễ dàng hơn rất nhiều so với phương pháp truyền thống.

Mục tiêu của dự án vừa có thể áp dụng vào thực tế hiện nay vừa có thể thực hành về việc ứng dụng những kiến thức đã học về môn “Lập trình hướng đối tượng”, sử dụng ngôn ngữ C++ (ngôn ngữ dùng để lập trình hướng đối tượng). Ngoài ra thông qua việc xây dựng và phát triển dự án chúng em sẽ có thêm kinh nghiệm làm việc nhóm, chia sẻ và giúp đỡ giữa các thành viên giúp hoàn thành tốt dự án.

PHÂN CÔNG THÀNH VIÊN TRONG NHÓM

Thành viên	Công việc thực hiện	Đánh giá
Nguyễn Xuân Trường	<ul style="list-style-type: none"> -Lên ý tưởng và chiến lược phát triển cho dự án -Xây dựng cơ sở dữ liệu cho dự án -Xây dựng code khung sườn và thuật toán chi tiết cho hệ thống -Xây dựng và thiết kế biểu đồ UML -Xây dựng các class cho dự án -Kiểm thử chức năng của dự án -Xây dựng file word và báo cáo 	Hoàn thành đúng tiến độ
Bùi Trịnh Thế Viên	<ul style="list-style-type: none"> -Lên ý tưởng và chiến lược phát triển cho dự án -Xây dựng cơ sở dữ liệu cho dự án -Xây dựng code khung sườn và thuật toán chi tiết cho hệ thống -Lên ý tưởng cho các use-case -Xây dựng các class cho dự án -Kiểm thử chức năng của dự án -Xây dựng slide và báo cáo 	Hoàn thành đúng tiến độ

CHƯƠNG 1. KHẢO SÁT, ĐẶC TẢ YÊU CẦU DỰ ÁN

1.1. Mô tả yêu cầu bài toán

Sử dụng ứng dụng quản lý nhân sự vào việc quản lý trong các công ty và doanh nghiệp. Ứng dụng quản lý nhân sự cần những tiêu chuẩn sau:

1. Giao diện: đơn giản, trực quan và dễ dàng sử dụng cho người mới bắt đầu.

Ứng dụng có độ chính xác cao, dữ liệu được bảo mật kỹ càng.

2. Thông tin về nhân sự:

Quản lý được số lượng nhân viên, trưởng phòng, giám đốc, phòng ban, công việc.

Có thể thêm, xóa, xuất, lưu trữ dữ liệu.

Tính toán được tiền lương của nhân sự.

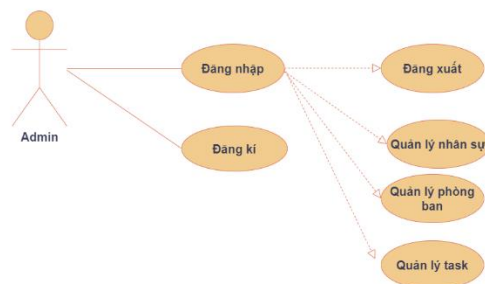
3. Bảo mật thông tin:

Cần có tài khoản mới có thể đăng nhập và sử dụng được ứng dụng.

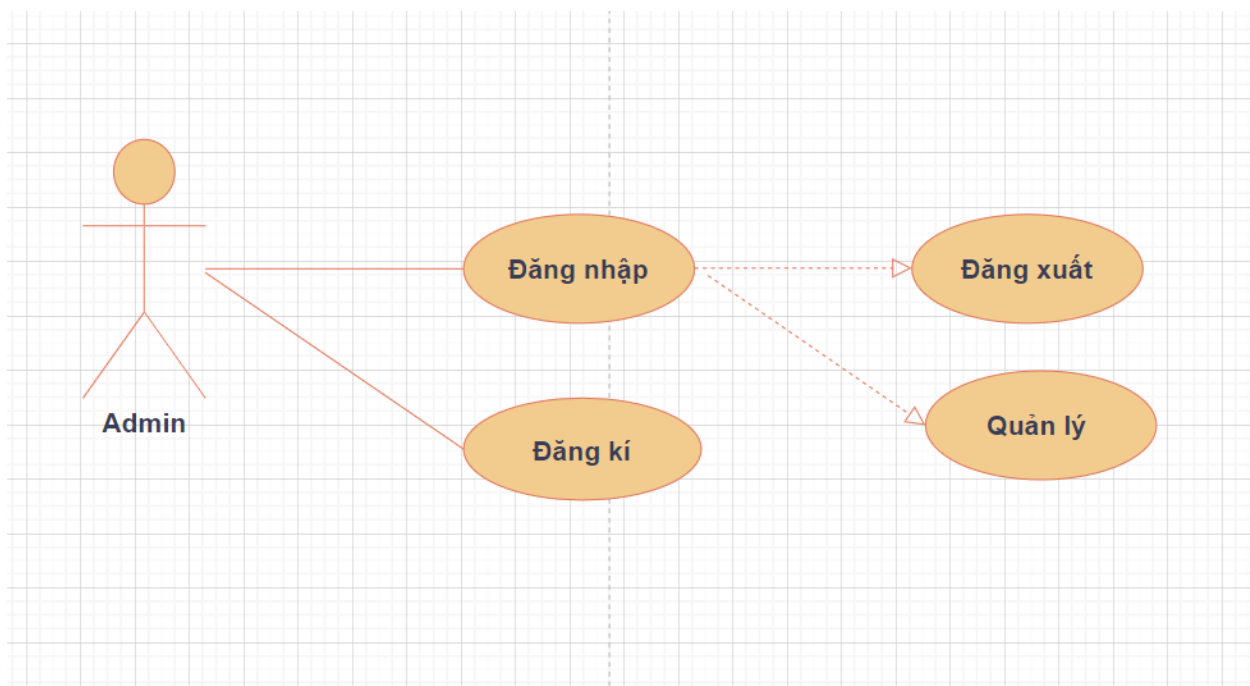
Admin có thể tạo thêm tài khoản.

1.2. Biểu đồ Use-case

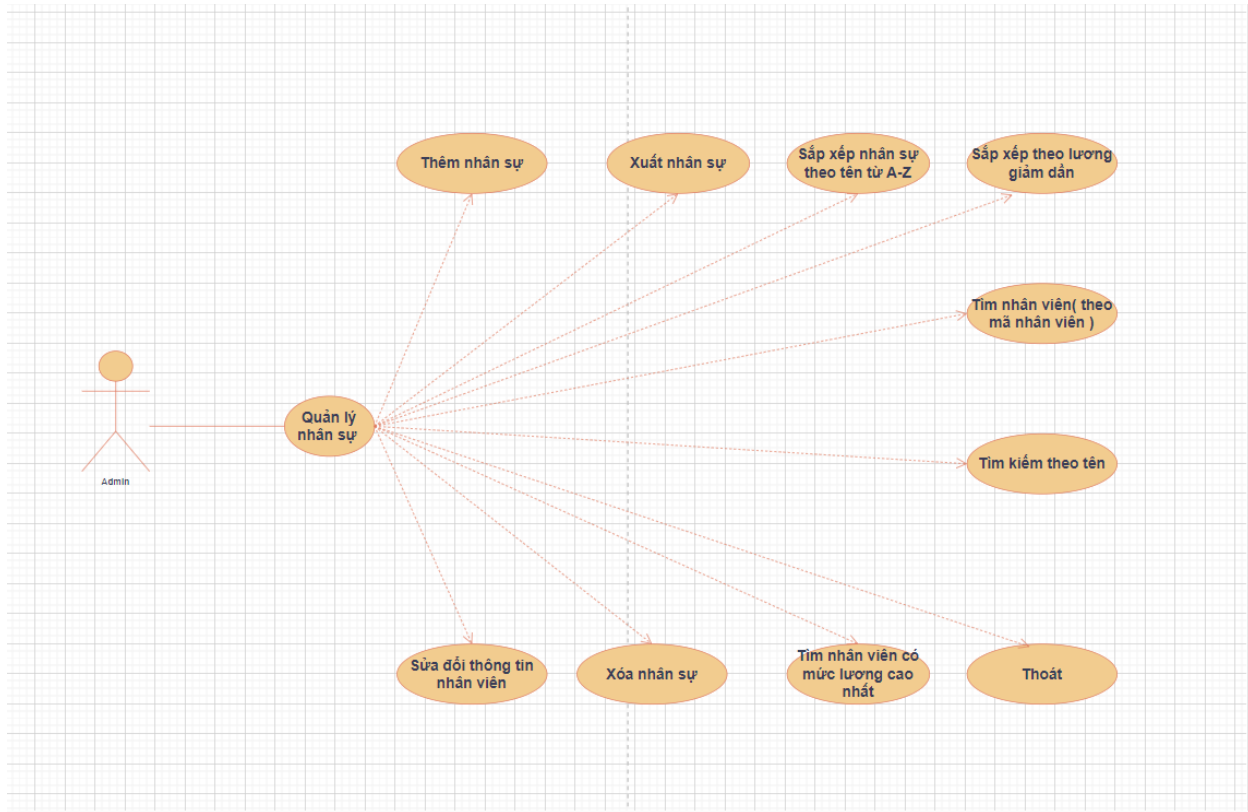
1.2.1. Biểu đồ Use-case tổng quan



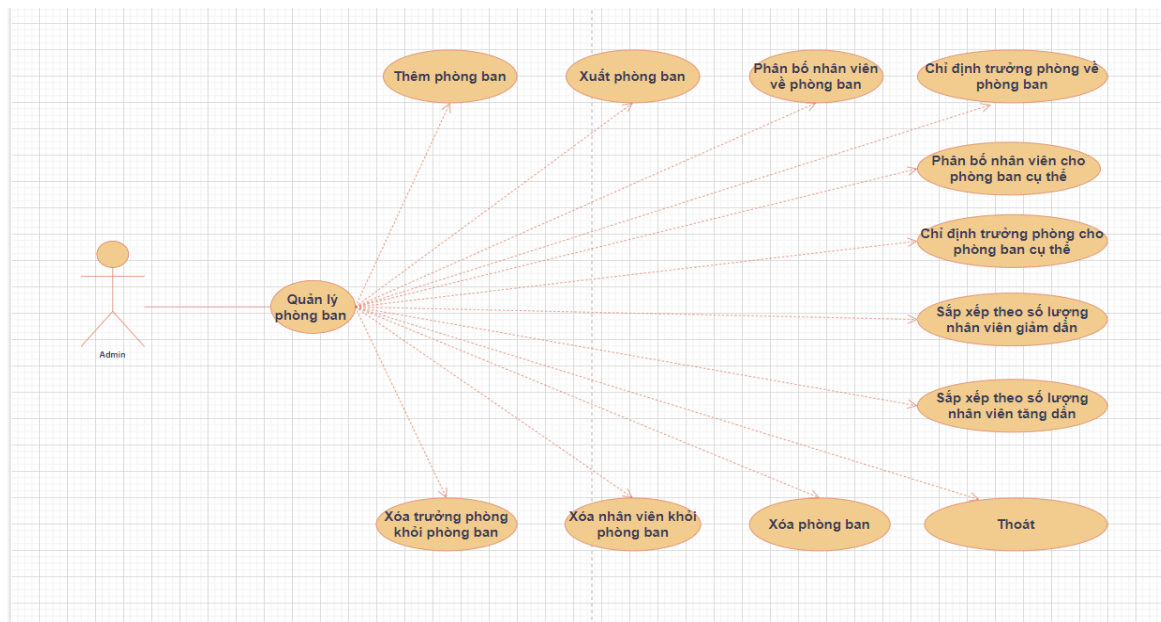
1.2.1.1. Biểu đồ Use-case đăng nhập



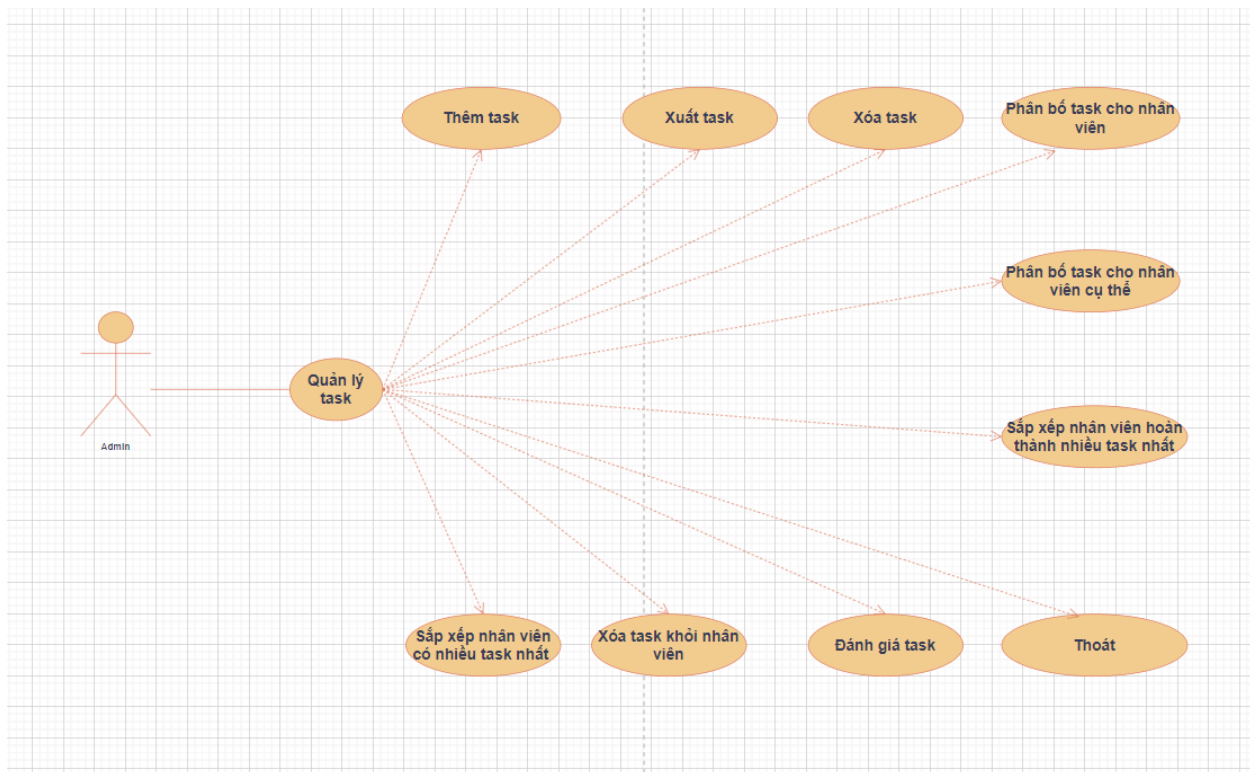
1.2.1.2. Biểu đồ Use-case quản lý nhân sự



1.2.1.3. Biểu đồ Use-case quản lý phòng ban



1.2.1.4. Biểu đồ Use-case quản lý task



1.2.2. Đặc tả biểu đồ Use-case

1.2.2.1. Đặc tả biểu đồ Use-case đăng nhập

Mã UC	UC01	
Tác nhân (Actor)	Admin	
Điều kiện trước(Pre-condition)	Chương trình đang ở menu đăng nhập	
Luồng chính	Tác nhân	Hệ thống
	Chọn đăng nhập và nhập tài khoản mật khẩu đúng	Đăng nhập và cho phép sử dụng chương trình
	Chọn đăng kí và đăng kí	Lưu thông tin tài khoản và về menu đăng nhập
Luồng thay thế	Nhập sai mật khẩu	Hiển thị thông báo nhập lại
	Chọn đăng kí nhưng nhập sai mật khẩu lần 2	Nhập lại mật khẩu lần 2

	Chọn đăng kí nhưng đăng kí tài khoản đã tồn tại	Yêu cầu đăng kí tài khoản khác
Điều kiện sau		

1.2.2.2. Đặc tả Use-case quản lý nhân sự

Mã UC	UC02	
Tác nhân(Actor)	Admin	
Điều kiện trước(Pre-condition)	Chương trình đang ở menu quản lý nhân sự	
Luồng chính	Tác nhân	Hệ thống
	Chọn thêm nhân sự	Hiện thị menu để thêm nhân sự
	Chọn xuất nhân sự	Hiện thị thông tin toàn bộ nhân sự
	Chọn sắp xếp nhân sự theo tên từ A-Z	Hiện thị thông tin nhân sự sau khi đã sắp xếp
	Chọn sắp xếp theo lương giảm dần	Hiện thị thông tin nhân sự sau khi đã sắp xếp
	Chọn tìm nhân viên	Hiện thị ô nhập mã số nhân viên=> Hiện thị thông tin nhân viên , thoát nếu nhấn 0
	Chọn tìm kiếm nhân viên theo tên	Hiện thị ô nhập kí tự có thể có trong tên nhân sự=> Hiện thị thông tin các nhân sự có chứa kí tự đó trong tên
	Chọn tìm kiếm nhân viên có mức lương cao nhất	Hiện thị thông tin nhân viên có mức lương cao nhất
	Chọn sửa đổi thông tin nhân viên	Hiện thị danh sách nhân viên=> Nhập mã nhân viên cần sửa=> Thay đổi thông tin nhân viên và lưu
	Chọn xóa nhân sự	Hiện thị ô để nhập mã nhân viên cần xóa=> Xóa nhân viên được nhập mã

	Chọn thoát khỏi menu quản lý nhân sự	Thoát về menu chính
Luồng thay thế	Chọn thêm nhân sự=> nhập sai dạng	Chương trình không lưu
	Chọn tìm nhân viên=> Nhập mã nhân viên không tồn tại	Yêu cầu nhập lại
	Chọn tìm kiếm nhân viên theo tên=> Kí tự được nhập vào không có chứa trong tên	Thông báo ra console không có nhân viên phù hợp
	Chọn sửa đổi thông tin nhân viên=> Nhập mã nhân viên không tồn tại	Thoát về menu quản lý nhân sự
	Chọn xóa nhân viên=> Nhập mã nhân viên không tồn tại	Thông báo không tìm thấy và yêu cầu nhập lại
Điều kiện sau		

1.2.2.3. Đặc tả Use-case quản lý phòng ban

Mã UC	UC03	
Tác nhân(Actor)	Admin	
Điều kiện trước(Pre-condition)	Chương trình đang ở menu quản lý phòng ban	
Luồng chính	Tác nhân	Hệ thống
	Chọn thêm phòng ban	Hiện thị ô nhập số phòng ban=> lưu thông tin phòng ban đã nhập
	Chọn xuất phòng ban	Hiện thị thông tin toàn bộ phòng ban
	Chọn phân bổ nhân viên về phòng ban	Chuyển qua nhân viên khác nếu chọn 2, phân bổ nhân viên về phòng ban nếu chọn 1
	Chọn chỉ định trưởng phòng về phòng ban	Chuyển qua trưởng phòng khác nếu chọn 2, phân bổ nhân viên về phòng ban nếu chọn 1

	Chọn phân bố nhân viên về phòng ban cụ thể	Hiện thị thông tin nhân viên và phòng ban để phân bố=> Phân bố và lưu sau khi nhập mã phòng ban và mã nhân viên
	Chọn chỉ định trưởng phòng về phòng ban cụ thể	Hiện thị thông tin trưởng phòng và phòng ban để chỉ định=> Chỉ định và lưu, nhập 0 để thoát
	Chọn xóa trưởng phòng ra khỏi phòng ban	Hiện thông tin phòng ban và trưởng phòng=> Xóa trưởng phòng khỏi phòng ban và thoát về menu
	Chọn xóa nhân viên ra khỏi phòng ban	Hiện thị thông tin phòng ban=> Xóa nhân viên khỏi phòng ban và thoát hoặc thoát khi ấn 0
	Chọn xóa phòng ban	Hiện thị thông tin phòng ban=> Xóa phòng ban và về menu hoặc thoát khi ấn 0
	Chọn sắp xếp số lượng nhân sự theo thứ tự giảm dần	Hiện thị thông tin sau khi đã sắp xếp
	Chọn sắp xếp số lượng nhân sự theo chế độ tăng dần	Hiện thị thông tin sau khi đã sắp xếp
	Chọn thoát khỏi menu quản lý phòng ban	Thoát khỏi menu quản lý phòng ban
Luồng thay thế	Chọn thêm phòng ban=> Nhập sai định dạng	Chương trình không lưu
	Chọn phân bố nhân viên về phòng ban=> Nhập kí tự khác ngoài 1 và 2	Thông báo không được nhập kí tự khác ngoài 1 và 2
	Chọn chỉ định trưởng phòng về phòng ban=> Nhập kí tự khác ngoài 1 và 2	Thông báo không được nhập kí tự khác ngoài 1 và 2

	Chọn phân bố nhân viên về phòng ban cụ thể => Nhập sai mã nhân viên hoặc mã phòng ban	Thông báo lỗi và mời nhập lại
	Chọn chỉ định trưởng phòng về phòng ban cụ thể=>Nhập sai mã phòng ban hoặc sai mã trưởng phòng	Thông báo lỗi và nhập lại
	Chọn xóa trưởng phòng khỏi phòng ban	Nếu chưa có trưởng phòng thì sẽ không thực hiện được chức năng
	Chọn xóa nhân viên khỏi phòng ban	Nếu chưa có nhân viên thì sẽ không thực hiện được chức năng
	Chọn xóa phòng ban=> Nhập sai mã phòng ban	Thông báo lỗi và yêu cầu nhập lại
Điều kiện sau		

1.2.2.4. Đặc tả Use-case quản lý task

Mã UC	UC04	
Tác nhân(Actor)	Admin	
Điều kiện trước(Pre-condition)	Chương trình đang ở menu quản lý task	
Luồng chính	Tác nhân	Hệ thống
	Chọn thêm task=> Nhập mã task và mã nhân viên	Hiển thị thông tin task và thông tin nhân viên
	Chọn xuất task	Hiển thị thông tin các task
	Chọn xóa task=>Nhập mã task cần xóa	Hiển thị thông tin task=> Xóa task được nhập mã
	Chọn phân bố task cho nhân viên=> Chọn 1 để phân bố, 2 để chuyển qua nhân viên khác	Chọn 1: hiển thị danh sách task=> Nhập mã task để phân bố hoặc 0 để thoát
	Chọn phân bố task cho nhân viên cụ thể=> Nhập mã nhân viên và mã task	Hiển thị thông tin nhân viên và task=> Phân bố task

	Chọn sắp xếp nhân viên hoàn thành nhiều task nhất	Hiển thị thông tin sau khi sắp xếp
	Chọn sắp xếp nhân viên có nhiều task nhất	Hiển thị thông tin sau khi sắp xếp
	Xóa task ra khỏi nhân viên	Hiển thị danh sách nhân viên=>Nhập mã nhân viên, mã task=> Xóa
	Chọn đánh giá task	Hiện thị thông tin các task đã được phân bổ=> Nhập mã task cần đánh giá hoặc ấn 0 để thoát=> Đánh giá task
	Chọn thoát ra khỏi menu quản lý task	Thoát về menu chính
Luồng thay thế	Chọn xóa task=>Nhập sai mã task	Thông báo nhập sai và yêu cầu nhập lại
	Chọn phân bổ task cho nhân viên=> Nhập kí tự khác 1 và 2, nhập sai mã task hoặc nhập mã task đã được phân	Yêu cầu chỉ được nhập 1 hoặc 2 Thông báo mã task không tồn tại hoặc đã phân=> Yêu cầu nhập lại
	Chọn phân bổ task cho nhân viên cụ thể=> Nhập sai mã task hoặc mã nhân viên	Thông báo không tìm thấy task phù hợp và yêu cầu nhập lại
	Chọn xóa task ra khỏi nhân viên=> Nhập sai mã nhân viên	Thông báo nhập sai và yêu cầu nhập lại
	Chọn đánh giá task=> Nhập sai mã nhân viên hoặc mã task	Thông báo không tìm thấy và yêu cầu nhập lại
Điều kiện sau		

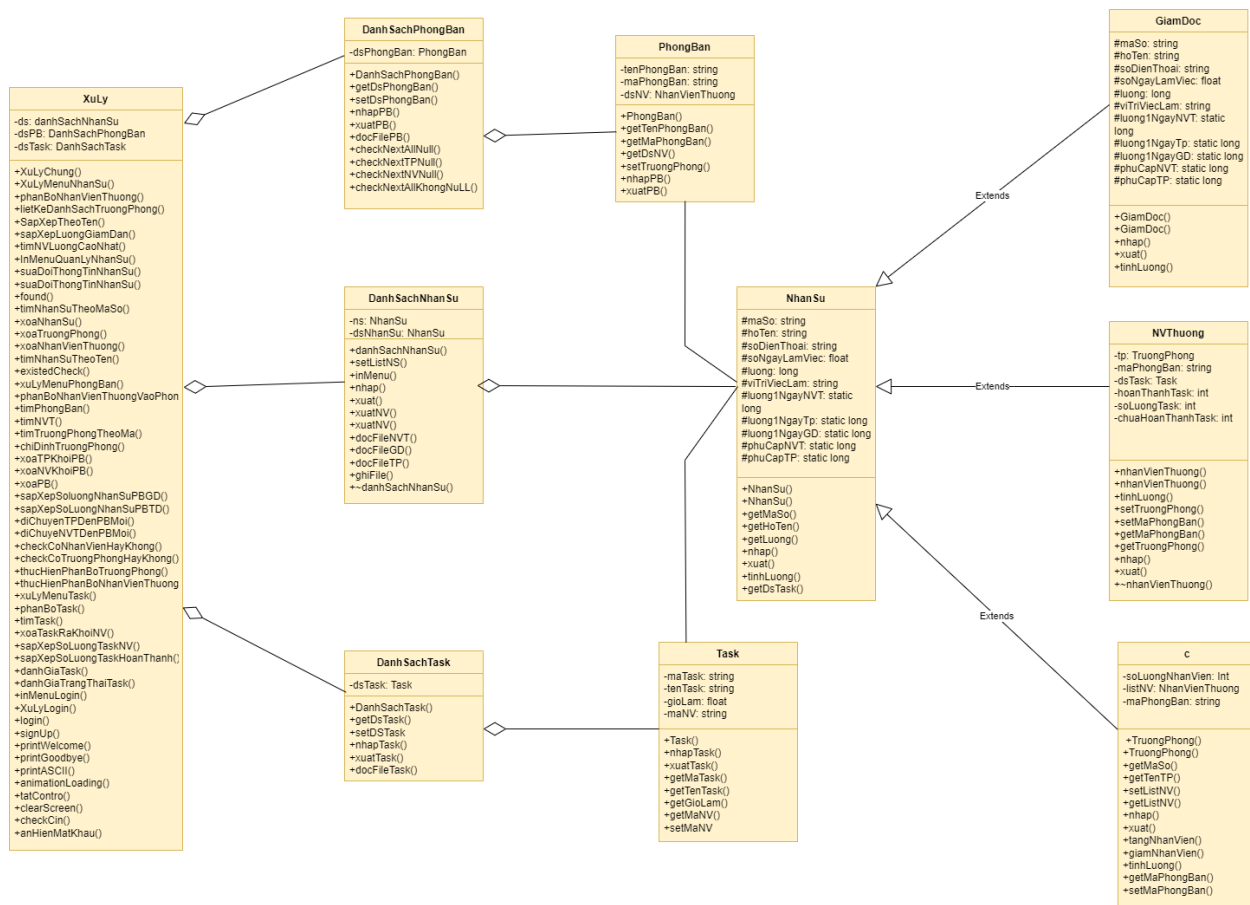
CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ BÀI TOÁN

2.1. Thiết kế cơ sở dữ liệu

Cấu trúc dữ liệu gồm các phần sau đây:

DanhSachNhanSu(NhanVienThuong, TruongPhong, GiamDoc)	Thông tin về nhân sự trong công ty
DanhSachPhongBan	Thông tin về các phòng ban trong công ty
DanhSachTask	Thông tin về các tác vụ, công việc trong công ty
XuLy(hàm login(), XuLyLogin(), signUp())	Thông tin và xử lý về tài khoản đăng nhập

2.2. Biểu đồ lớp



2.3. Thiết kế lớp chi tiết

Class DanhSachPhongBan

Class DanhSachNhanSu

Danh SachPhongBan
-dsPhongBan: PhongBan
+DanhSachPhongBan() +getDsPhongBan() +setDsPhongBan() +nhapPB() +xuatPB() +docFilePB() +checkNextAllNull() +checkNextTPNull() +checkNextNVNull() +checkNextAllKhongNuLL()

Danh SachNhan Su
-ns: NhanSu -dsNhanSu: NhanSu
+danhSachNhanSu() +setListNS() +inMenu() +nhap() +xuat() +xuatNV() +xuatNV() +docFileNVT() +docFileGD() +docFileTP() +ghiFile() +~danhSachNhanSu()

Class DanhSachTask

Danh SachTask
-dsTask: Task
+DanhSachTask() +getDsTask() +setDsTask +nhapTask() +xuatTask() +docFileTask()

Class PhongBan

PhongBan
-tenPhongBan: string -maPhongBan: string -dsNV: NhanVienThuong
+PhongBan() +getTenPhongBan() +getMaPhongBan() +getDsNV() +setTruongPhong() +nhapPB() +xuatPB()

Class NhanSu

Class Task

Nhan Su
#maSo: string #hoTen: string #soDienThoai: string #soNgayLamViec: float #luong: long #viTriViecLam: string #luong1NgayNVT: static long #luong1NgayTp: static long #luong1NgayGD: static long #phuCapNVT: static long #phuCapTP: static long
+NhanSu() +NhanSu() +getMaSo() +getHoTen() +getLuong() +nhap() +xuat() + tinhLuong() +getDsTask()

Class GiamDoc

Task
-maTask: string -tenTask: string -gioLam: float -maNv: string
+Task() +nhapTask() +xuatTask() +getMaTask() +getTenTask() +getGioLam() +getMaNV() +setMaNV

Class NVThuong

GiamDoc
#maSo: string #hoTen: string #soDienThoai: string #soNgayLamViec: float #luong: long #viTriViecLam: string #luong1NgayNVT: static long #luong1NgayTp: static long #luong1NgayGD: static long #phuCapNVT: static long #phuCapTP: static long
+GiamDoc() +GiamDoc() +nhap() +xuat() + tinhLuong()

NVThuong
-tp: TruongPhong -maPhongBan: string -dsTask: Task -hoanThanhTask: int -soLuongTask: int -chuaHoanThanhTask: int
+nhanVienThuong() +nhanVienThuong() + tinhLuong() +setTruongPhong() +setMaPhongBan() +getMaPhongBan() +getTruongPhong() +nhap() +xuat() +~nhanVienThuong()

Class Xuly

XuLy			
-ds: danhSachNhanSu -dsPB: DanhSachPhongBan -dsTask: DanhSachTask			
+XuLyChung() +XuLyMenuNhanSu() +phanBoNhanVienThuong() +lietKeDanhSachTruongPhong() +SapXepTheoTen() +sapXepLuongGiamDan() +timNVLuongCaoNhat() +InMenuQuanLyNhanSu() +suaDoiThongTinNhanSu() +suaDoiThongTinNhanSu() +found() +timNhanSuTheoMaSo() +xoaNhanSu() +xoaTruongPhong() +xoaNhanVienThuong() +timNhanSuTheoTen()	+checkCin() +animationLoading() +printWelcome() +XuLyLogin() +danhGiaTask() +sapXepSoLuongTaskNV() +phanBoTask() +thucHienPhanBoTruongPhong() +checkCoNhanVienHayKhong() +diChuyenTPDenPBMoi() +sapXepSoLuongNhanSuPBGD() +xoaNVKhoiPB() +chiDinhTruongPhong() +timNVT() +phanBoNhanVienThuongVaoPhongBan() +xuLyMenuPhongBan()	+clearScreen() +printASCII() +signUp() +inMenuLogin() +sapXepSoLuongTaskHoanThanh() +xoaTaskRaKhoiNV() +xuLyMenuTask() +checkCoTruongPhongHayKhong() +diChuyeNVTDenPBMoi() +sapXepSoLuongNhanSuPBTD() +xoaPB() +xoaTPKhoiPB() +timTruongPhongTheoMa() +anHienMatKhau() +existedCheck()	+tatContro() +printGoodbye() +login() +danhGiaTrangThaiTask() +timTask() +thucHienPhanBoNhanVienThuong()

CHƯƠNG 3. CÔNG NGHỆ VÀ THUẬT TOÁN SỬ DỤNG

3.1. Công nghệ sử dụng

Để xây dựng và phát triển chương trình, nhóm đã sử dụng ngôn ngữ lập trình C++ và các thư viện STL C++.

Nhóm sử dụng trình soạn thảo mã nguồn là Visual Studio Code và Github dùng để các thành viên trong nhóm có thể xây dựng và trao đổi source code với nhau.

3.2. Thuật toán sử dụng

Nhóm sử dụng các thuật toán tính toán số học để tính toán lương, thời gian làm việc của nhân sự. Bên cạnh đó là sử dụng thuật toán sắp xếp Lambda C++ và thuật toán kiểm tra điều kiện để thực hiện các tác vụ của chương trình.

CHƯƠNG 4. XÂY DỰNG CHƯƠNG TRÌNH

4.1. Kết quả xây dựng chương trình

Tổng quan kết quả chạy chương trình với các chức năng đã được thực hiện bao gồm:

1. Đăng nhập/ Đăng kí

2. Quản lý nhân sự:

- Phân bổ nhân viên thường vào các phòng ban.
- Liệt kê danh sách các trưởng phòng.
- Sắp xếp nhân sự theo tên.
- Sắp xếp nhân sự theo lương giảm dần.
- Tìm nhân viên có mức lương cao nhất.
- Sửa đổi thông tin nhân sự.
- Tìm kiếm nhân sự theo mã số.
- Xóa nhân sự.

3. Quản lý phòng ban:

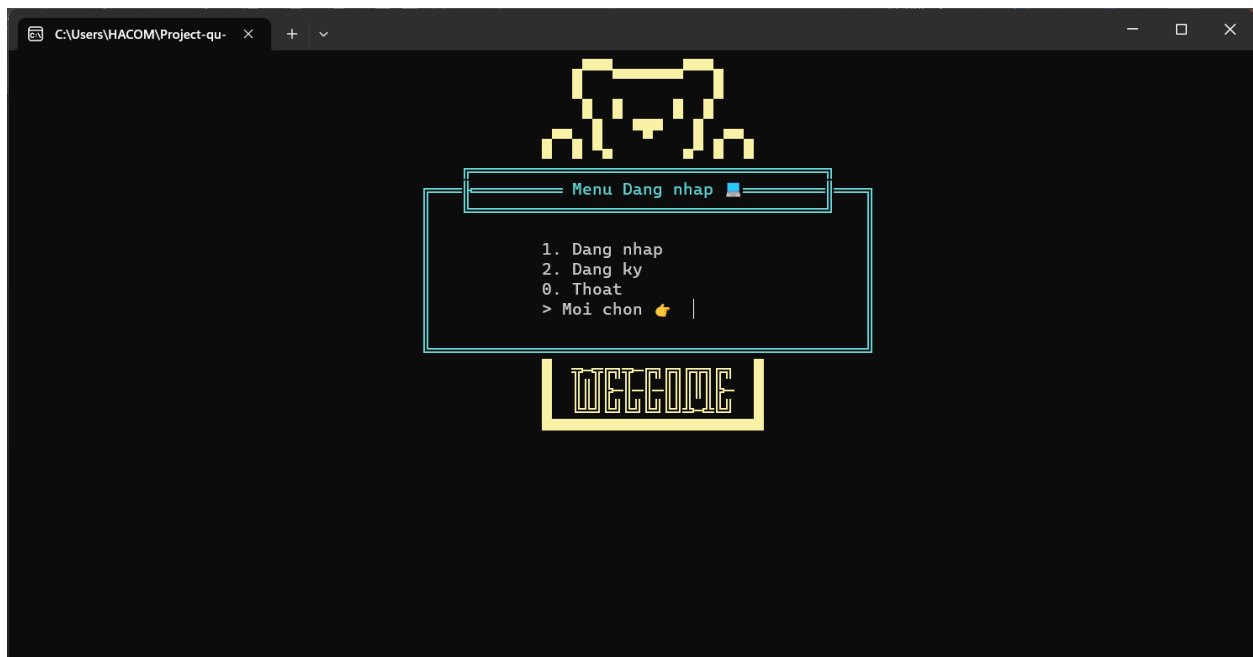
- Phân bổ nhân viên thường vào các phòng ban.
- Tìm phòng ban theo mã.
- Tìm nhân viên thường trong một phòng ban.
- Tìm trưởng phòng của một phòng ban.
- Xóa phòng ban.
- Sắp xếp phòng ban theo số lượng nhân sự (phòng ban chung và phòng ban công việc).

4. Quản lý task:

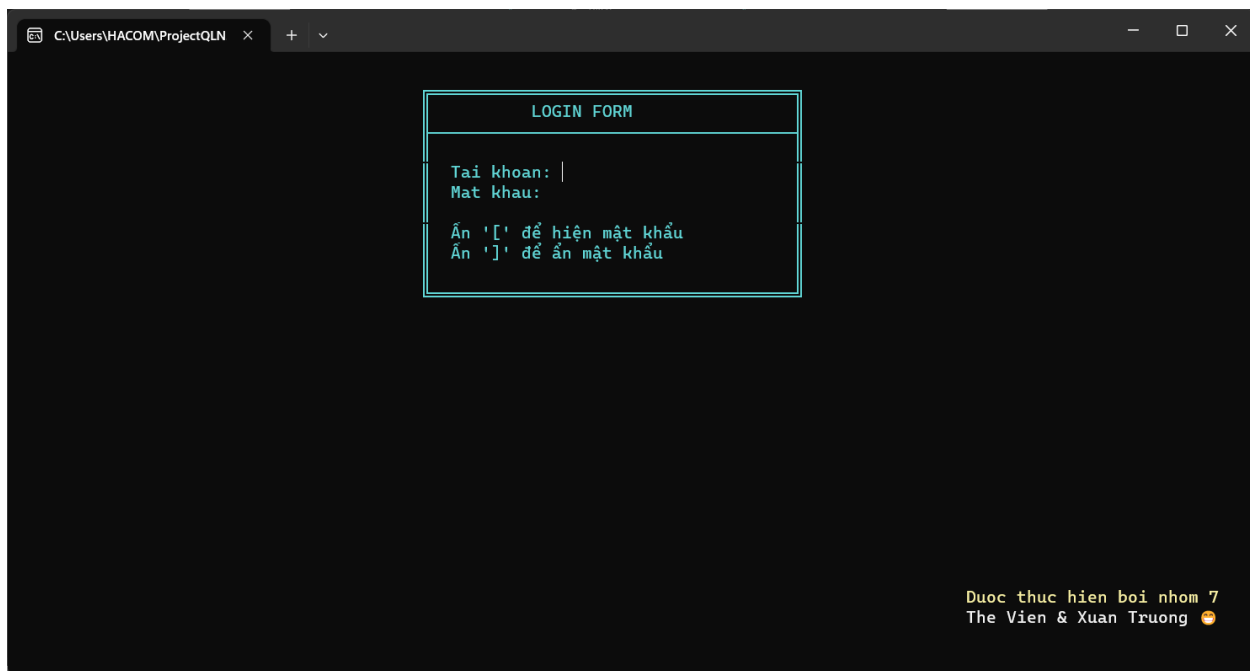
- Phân bổ công việc cho nhân viên.
- Tìm công việc theo mã.
- Xóa công việc.
- Xóa công việc đã được phân cho nhân viên.
- Sắp xếp công việc theo số lượng nhân viên được phân công.
- Sắp xếp công việc theo trạng thái hoàn thành.
- Đánh giá trạng thái công việc.

4.2. Giao diện chương trình

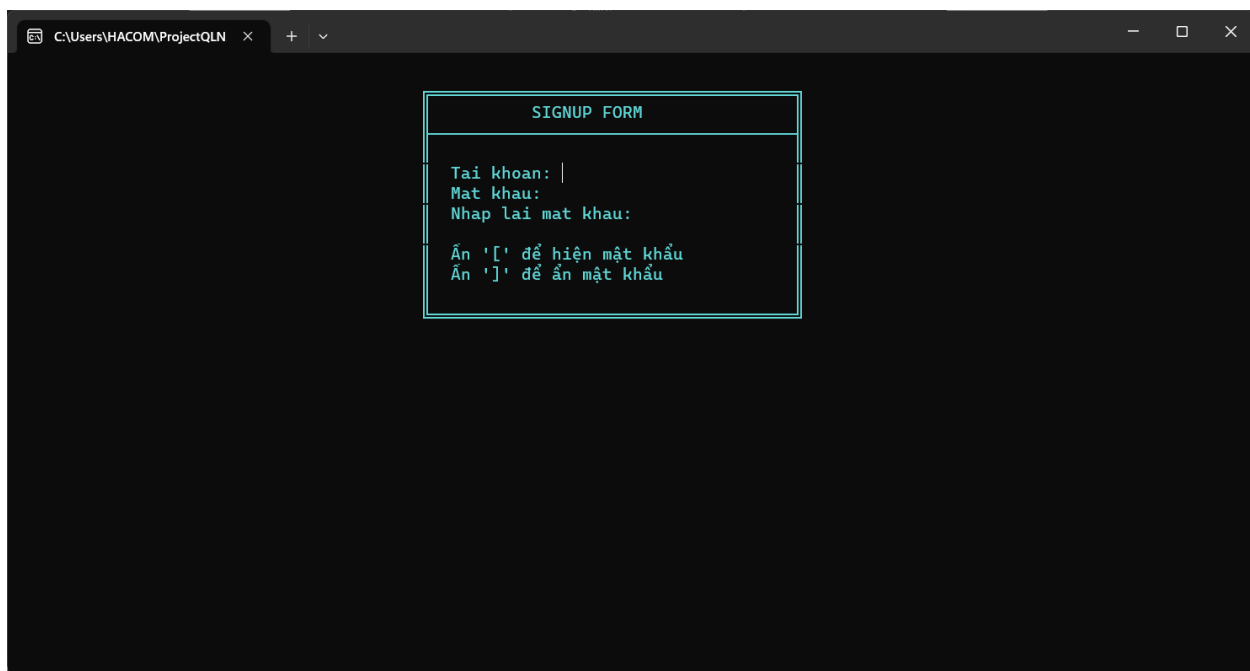
4.2.1. Giao diện đăng nhập



Hình 1. Giao diện màn hình chính



Hình 2. Giao diện màn hình đăng nhập



Hình 3. Giao diện màn hình đăng kí

The screenshot shows a web browser window with the address bar displaying 'C:\Users\HACOM\ProjectQLN'. The main content area features a 'SIGNUP FORM' box with the following text:

```
Tai khoan: admin
Mat khau: *****
Nhap lai mat khau: *****
Ấn '[' để hiện mật khẩu
Ấn '[' để ẩn mật khẩu
```

Below the form, a red error message is displayed: 'Tai khoan ban dang ky da ton tai!'.

Hình 4. Màn hình đăng kí tài khoản đã tồn tại

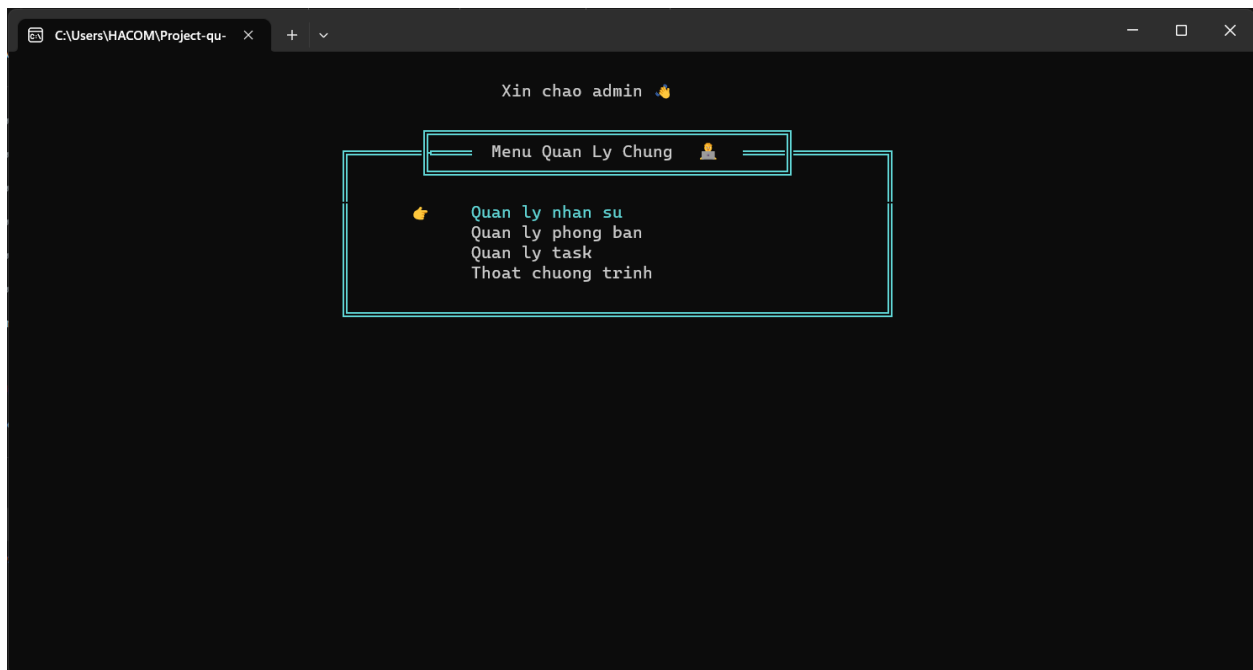
The screenshot shows the same 'SIGNUP FORM' box as in the previous image, but with updated text:

```
Tai khoan: taikhoanmoi
Mat khau: *****
Nhap lai mat khau: *****
Ấn '[' để hiện mật khẩu
Ấn '[' để ẩn mật khẩu
```

Below the form, a green success message is displayed: 'Dang ky thanh cong' followed by a green checkmark icon.

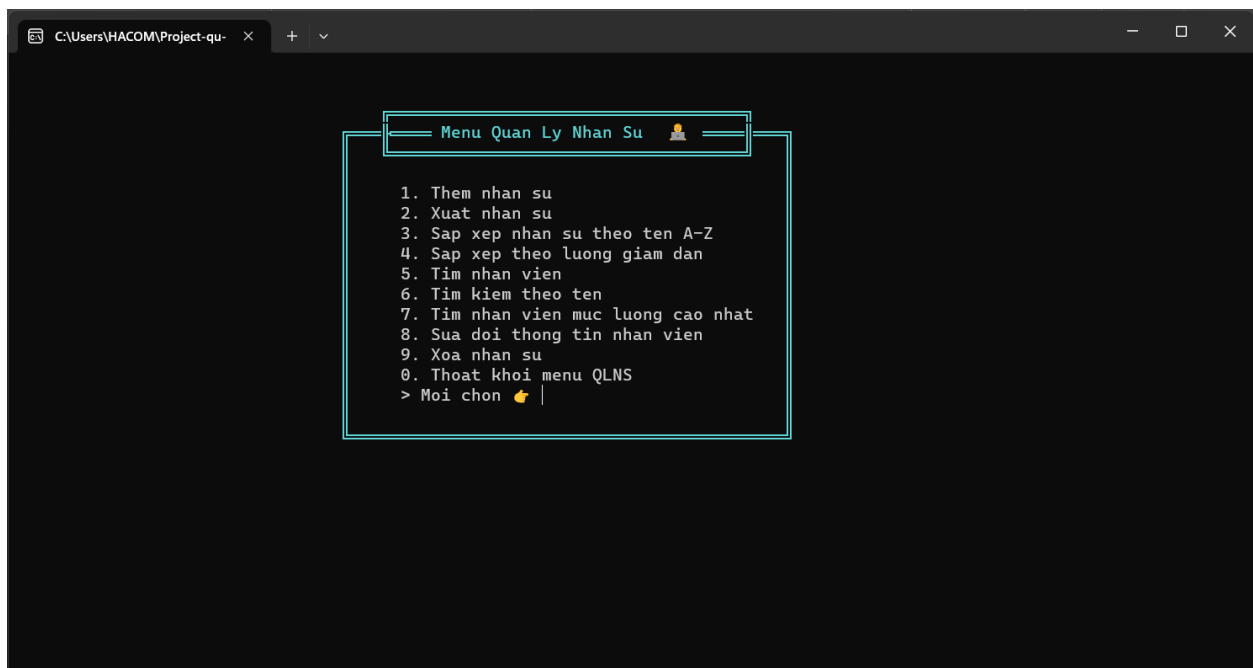
Hình 5. Đăng kí tài khoản thành công

Sau khi đăng nhập thì admin có thể sử dụng các chức năng của chương trình



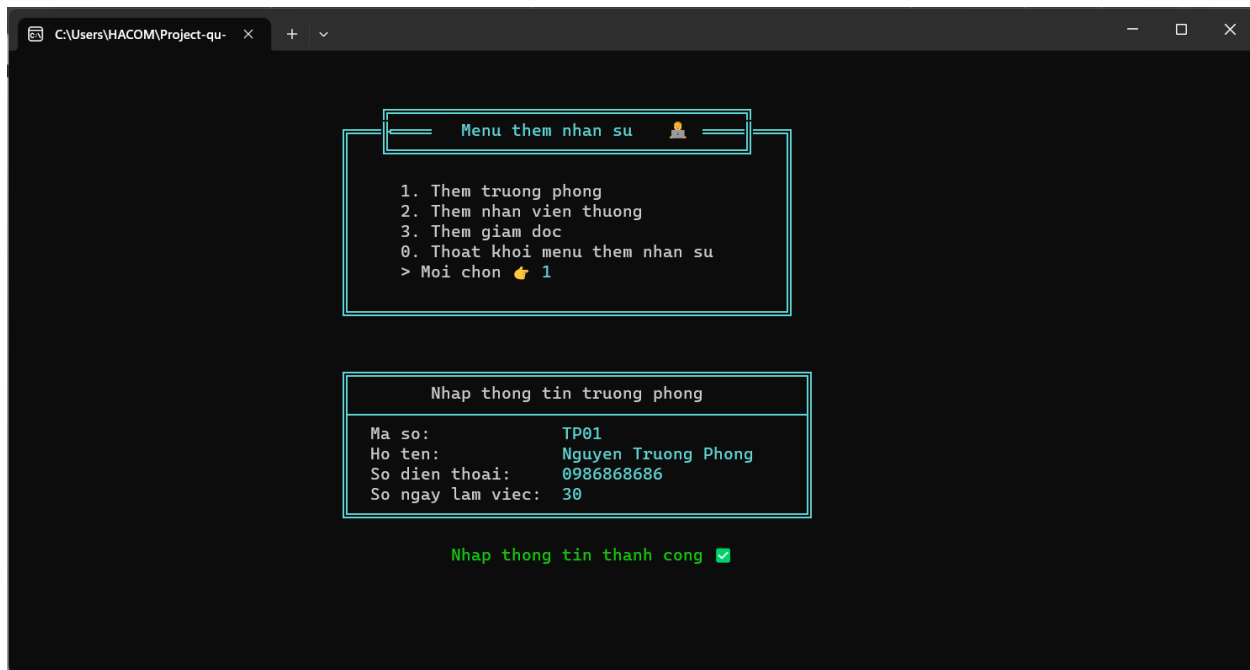
Hình 6. Menu quản lý chung

4.2.2. Giao diện menu quản lý nhân sự

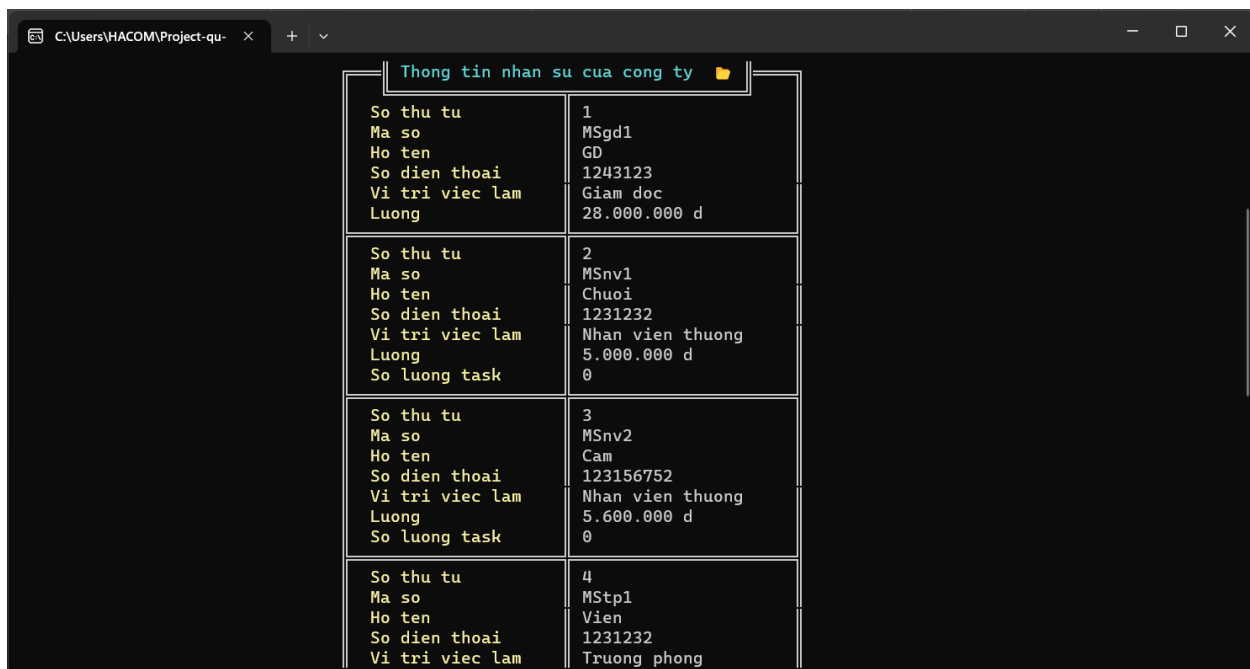


Hình 7. Menu quản lý nhân sự

Dưới đây là một số chức năng trong menu quản lý nhân sự:



Hình 8. Ví dụ cho việc thêm nhân sự

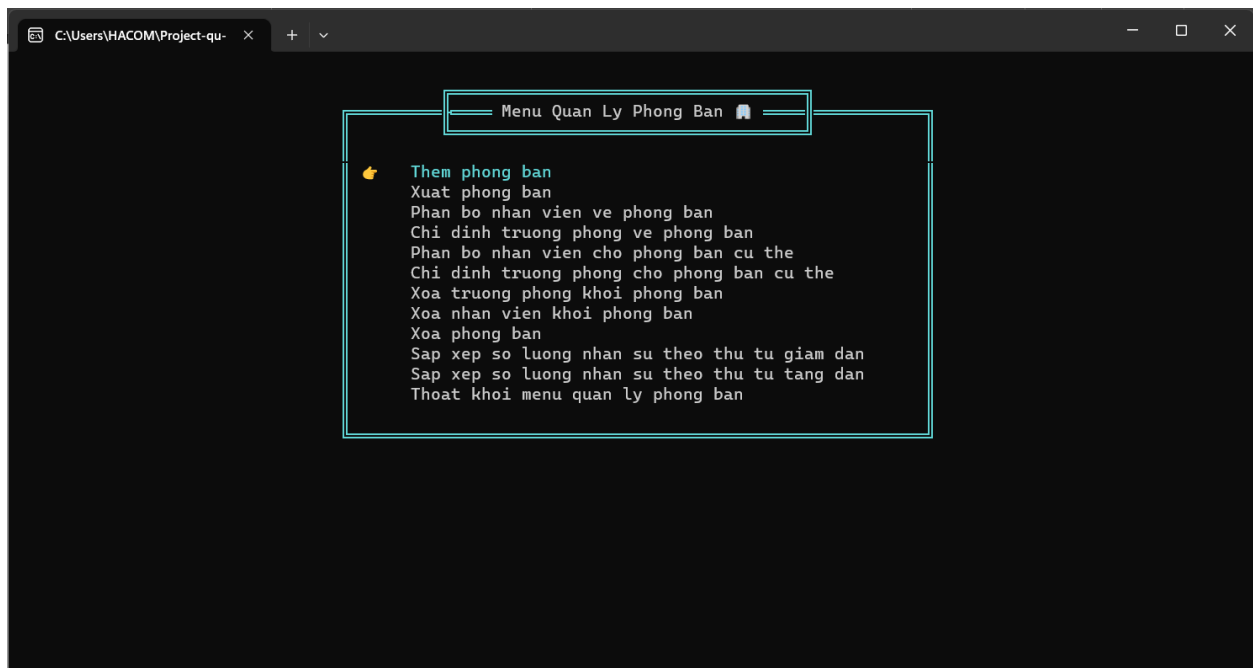


Hình 9. Thông tin được xuất ra khi thực hiện chức năng “Xuất nhân sự”



Hình 10. Kết quả chương trình khi thực hiện chức năng “Tìm kiếm theo tên”

4.2.3. Giao diện menu quản lý phòng ban

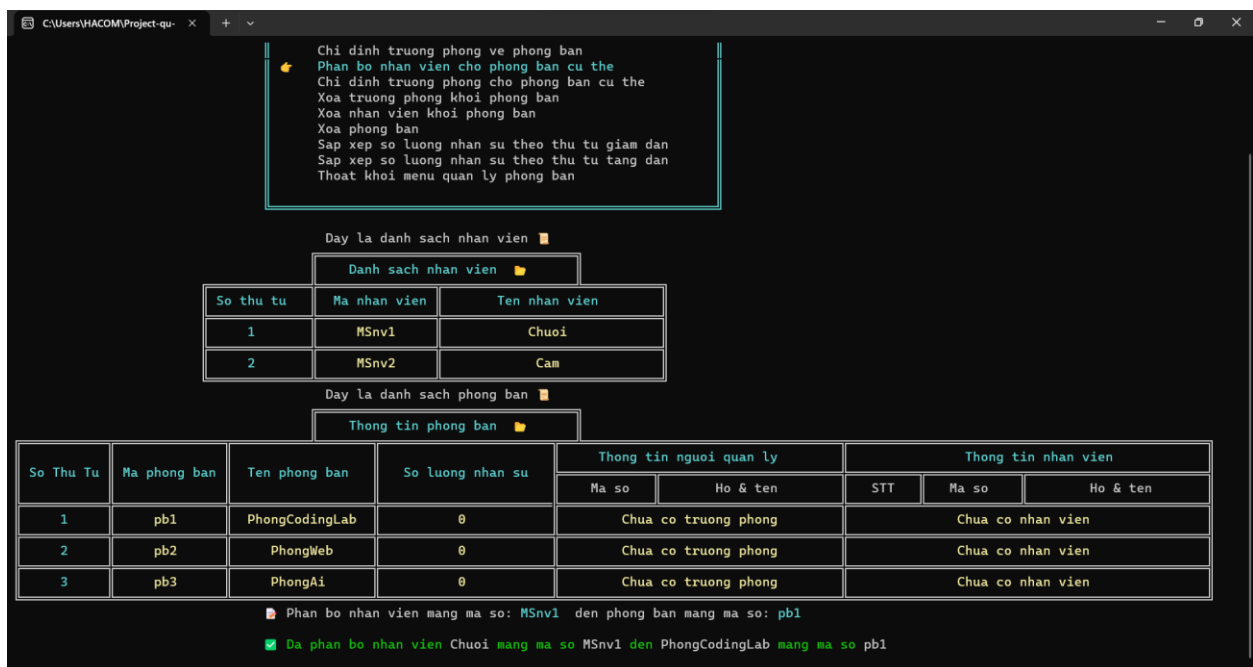


Hình 11. Menu quản lý phòng ban

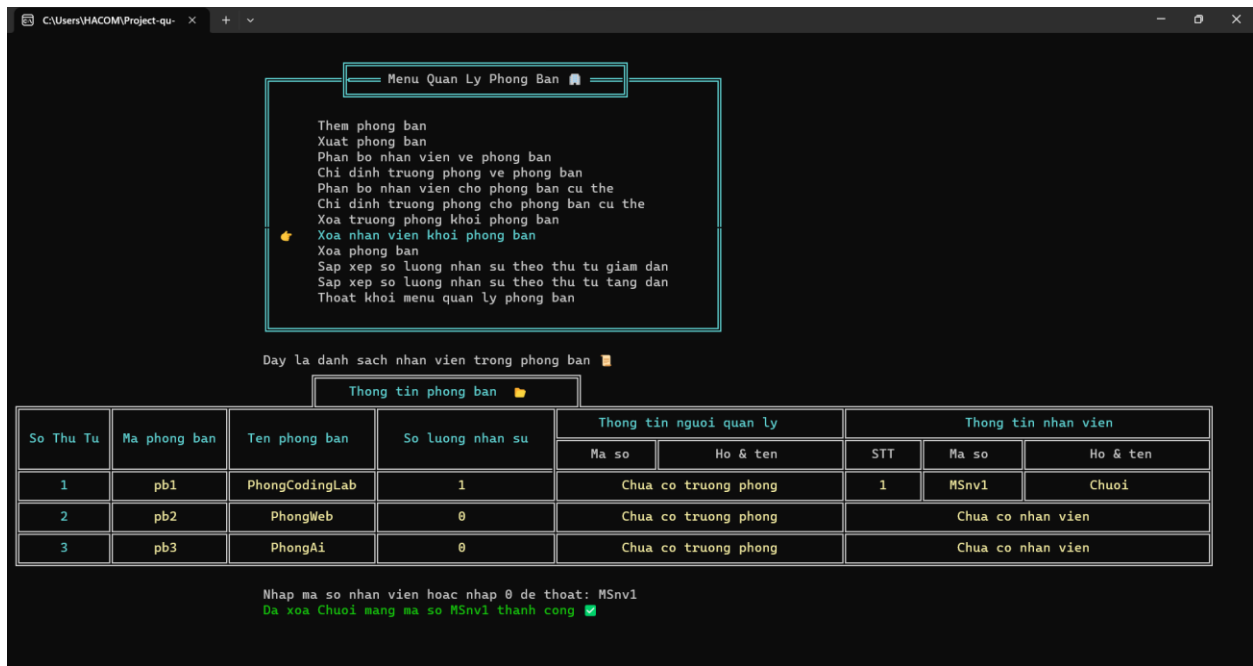
Một số chức năng của menu quản lý phòng ban:



Hình 12. Chương trình thực hiện chức năng thêm phòng ban thành công

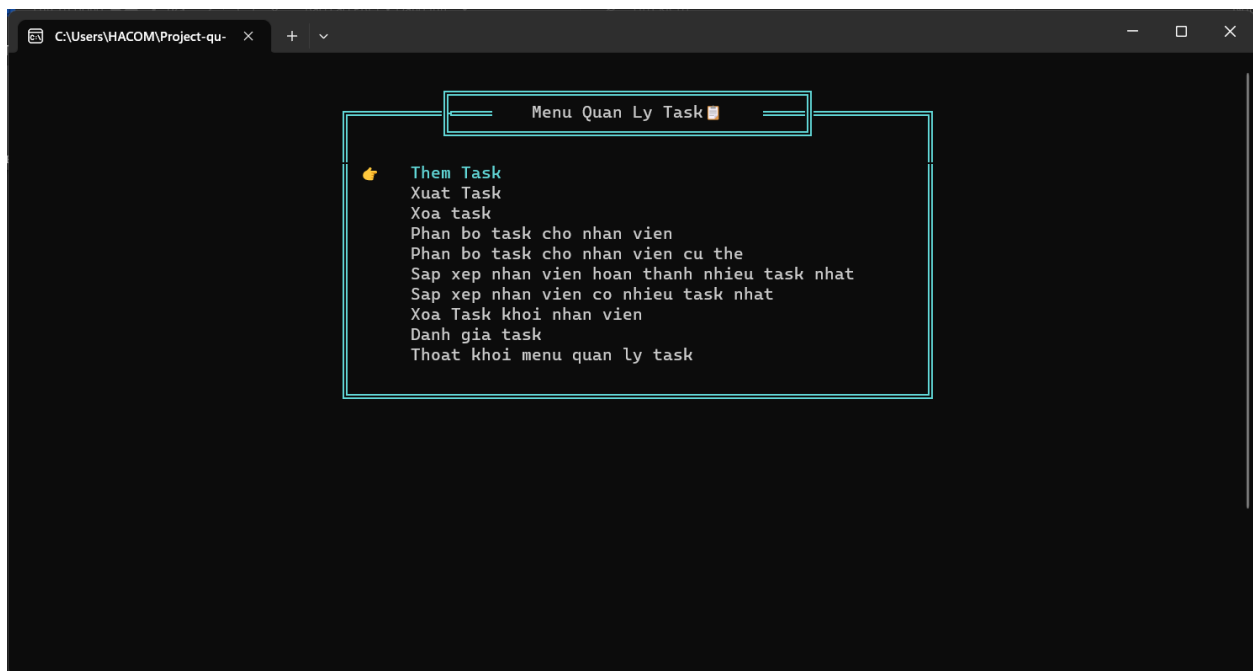


Hình 13. Chương trình thực hiện chức năng phân bổ thành công



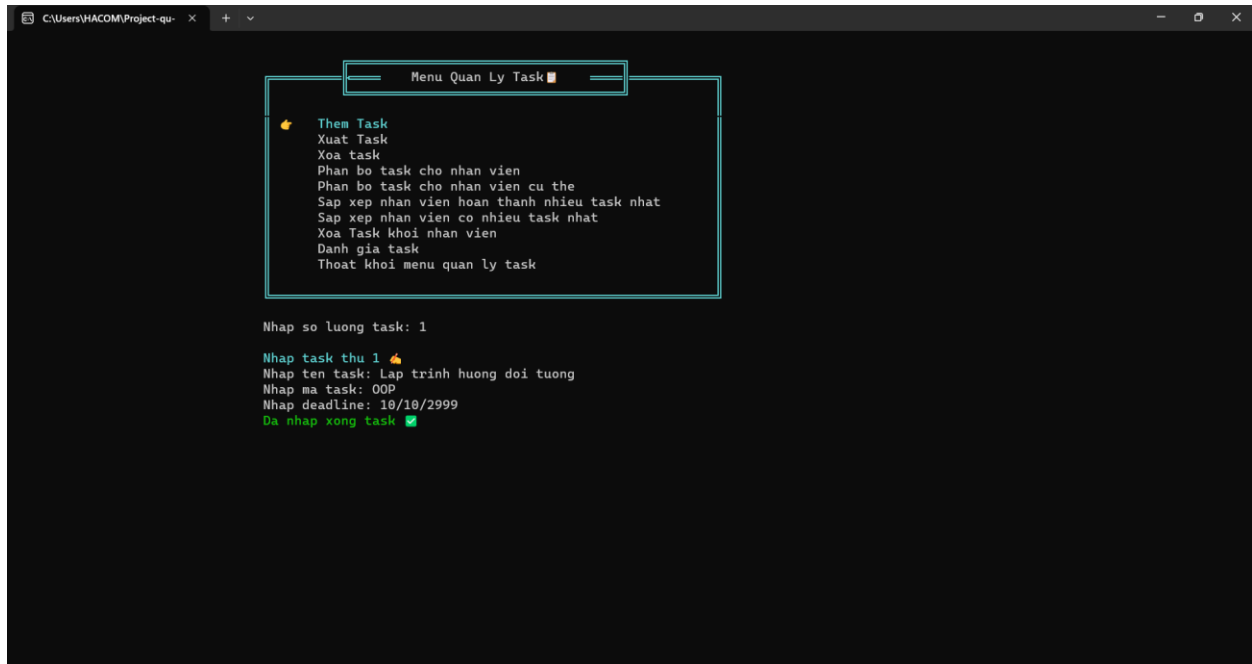
Hình 14. Chương trình thực hiện chức năng xóa nhân viên khỏi phòng ban thành công

4.2.4. Giao diện quản lý task

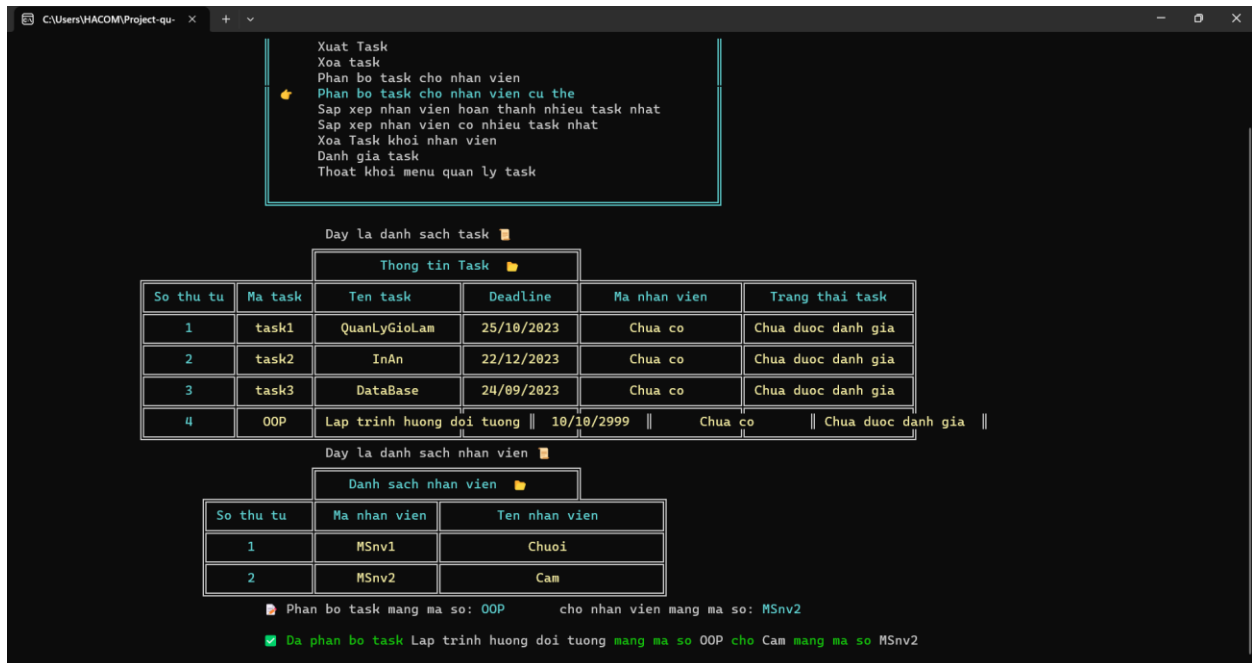


Hình 15. Menu quản lý task

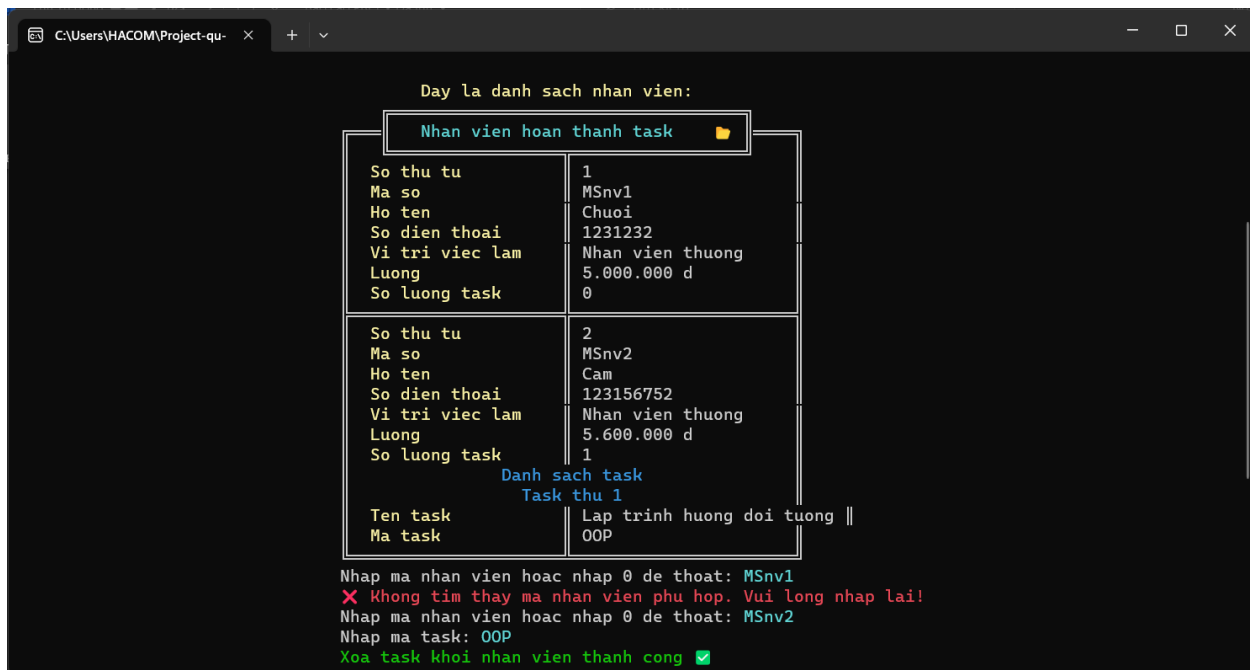
Dưới đây là một số chức năng của menu quản lý task:



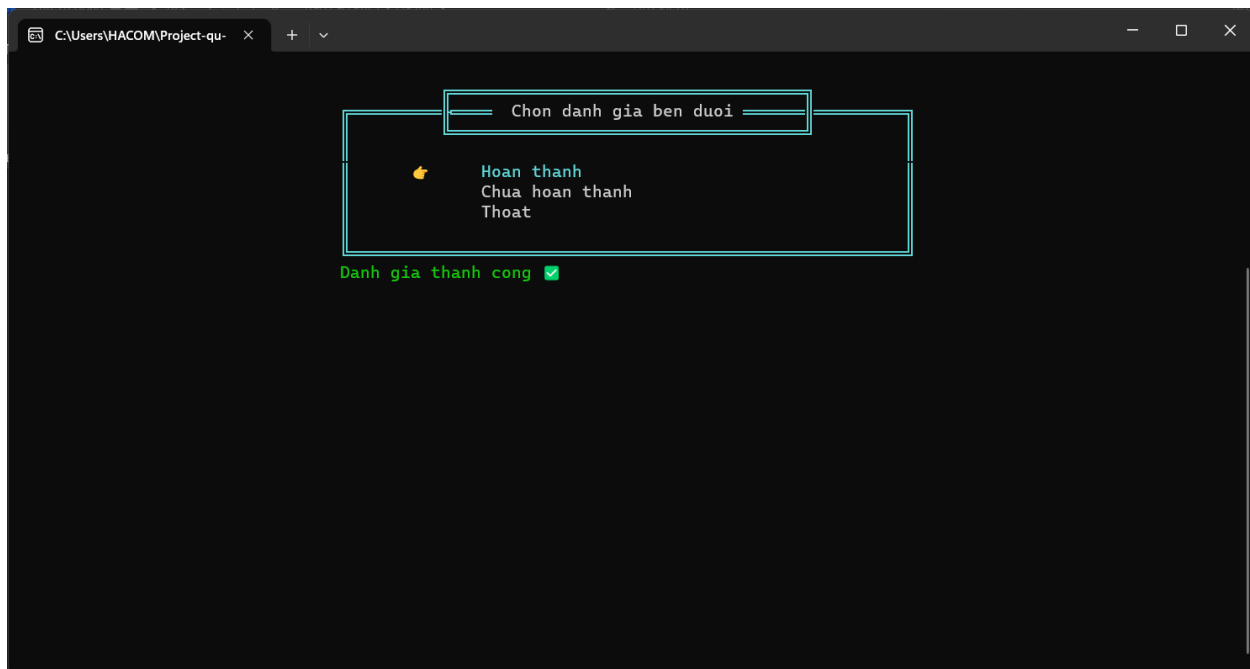
Hình 16. Chương trình thực hiện chức năng “Them task” thành công



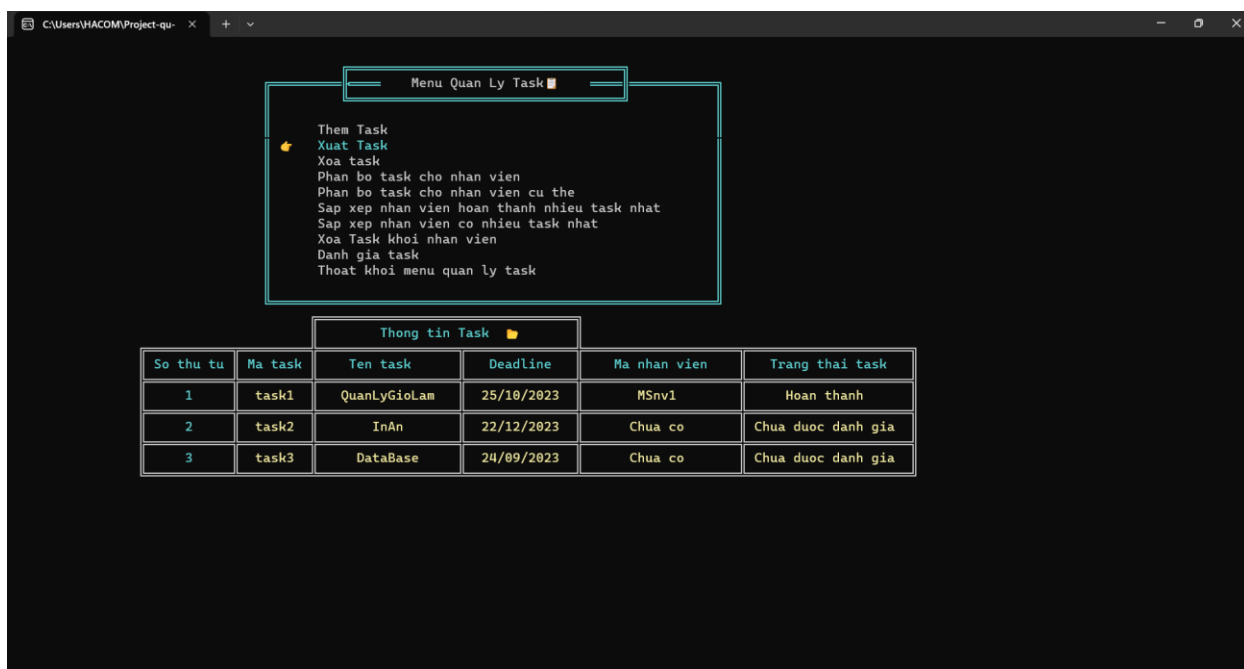
Hình 17. Chương trình thực hiện chức năng phân bổ thành công



Hình 18. Chương trình thực hiện chức năng xóa task khỏi nhân viên



Hình 19. Chương trình thực hiện chức năng đánh giá task



Hình 20. Task mang mã “task1” được đánh giá hoàn thành

4.3. Kiểm thử chương trình

1. Kiểm thử các chức năng đã thực hiện

STT	Tên chức năng kiểm thử
1	Chức năng đăng nhập / đăng kí
2	Chức năng quản lý nhân sự
3	Chức năng quản lý phòng ban
4	Chức năng quản lý task

2. Kiểm thử chức năng đăng nhập/ đăng kí

STT	Input	Output	Kết quả
-----	-------	--------	---------

1	Nhập đúng tên đăng nhập và mật khẩu	Đăng nhập vào chương trình	Pass
2	Nhập sai tên đăng nhập/ mật khẩu	Báo lỗi sai tên đăng nhập/mật khẩu và yêu cầu nhập lại	Pass
3	Nhập tên đăng nhập/ mật khẩu để đăng kí tài khoản	Tạo tài khoản thành công và thông báo	Pass
4	Thoát	Thoát chương trình	Pass

3. Kiểm thử chức năng quản lý nhân sự

STT	Tên chức năng	Input	Output	Kết quả
1	Thêm nhân sự	Nhập thông tin nhân sự muốn thêm	Thêm nhân sự và thông báo thêm thông tin thành công	Pass
2	Xuất nhân sự		In ra toàn bộ thông tin nhân sự của công ti	Pass
3	Sắp xếp nhân sự theo tên từ A-Z		Sắp xếp lại tên nhân sự theo thứ tự alphabet	Pass
4	Sắp xếp nhân sự theo lương giảm dần		Sắp xếp lại tên nhân sự theo lương giảm dần	Pass
5	Tìm nhân viên	Nhập mã nhân sự	In ra thông tin nhân sự cần tìm	Pass
6	Tìm kiếm theo tên	Nhập một hay một chuỗi kí tự	In ra thông tin của nhân viên có tên chứa kí tự đó	Pass

7	Tìm nhân viên có lương cao nhất		In ra thông tin nhân viên có lương cao nhất	Pass
8	Sửa đổi thông tin nhân viên	Nhập mã nhân viên muốn sửa thông tin	Nhập thông tin muốn sửa đổi và thông báo sửa đổi thông tin thành công	Pass
9	Xóa nhân sự	Nhập mã số nhân sự muốn xóa	Xóa nhân sự và thông báo xóa thành công	Pass
10	Thoát menu quản lý nhân sự		Thoát về menu chính	Pass

4. Kiểm thử chức năng quản lý phòng ban

STT	Tên chức năng	Input	Output	Kết quả
1	Thêm phòng ban	Nhập số lượng phòng ban, nhập thông tin phòng ban muốn thêm	Thêm phòng ban và thông báo thành công	Pass
2	Xuất phòng ban		In ra toàn bộ thông tin phòng ban	Pass
3	Phân bố nhân viên về phòng ban	Chọn nhân viên muốn chỉ định, nhập mã phòng ban để thêm / nhấn 0 để hủy	Phân bố nhân viên về phòng ban và thông báo thành công	Pass
4	Chỉ định trưởng phòng về phòng ban	Chọn trưởng phòng muốn chỉ định, nhập mã phòng ban	Phân bố trưởng phòng về phòng ban và thông báo thành công	Pass

		/ nhấn 0 để hủy		
5	Phân bố nhân viên về phòng ban cụ thể	Nhập mã phòng ban và mã nhân viên	Phân bố nhân viên về phòng ban	Pass
6	Chỉ định trưởng phòng về phòng ban cụ thể	Nhập mã phòng ban và mã nhân viên	Chỉ định trưởng phòng về phòng ban cụ thể	Pass
7	Xóa trưởng phòng khỏi phòng ban	Nhập mã số trưởng phòng / nhấn 0 để thoát	Xóa và thông báo thành công (Nếu chưa có trưởng phòng nào được phân bố về phòng ban thông báo)	Pass
8	Xóa nhân viên khỏi phòng ban	Nhập mã số nhân viên / nhấn 0 để thoát	Xóa và thông báo thành công (Nếu chưa có nhân viên nào được phân bố về phòng ban thông báo)	Pass
9	Xóa phòng ban	Nhập mã phòng ban	Xóa và thông báo thành công	Pass
10	Sắp xếp theo thứ tự số lượng nhân viên giảm dần		In ra màn hình danh sách phòng ban đã sắp xếp	Pass
11	Sắp xếp theo thứ tự số lượng nhân viên tăng dần		In ra màn hình danh sách phòng ban đã sắp xếp	Pass
12	Thoát menu quản lý phòng ban		Thoát về menu chính	Pass

5.Kiểm thử chức năng quản lý task

STT	Tên chức năng	Input	Output	Kết quả
1	Thêm task	Nhập thông tin task	Thêm và thông báo thành công	Pass
2	Xuất task		In ra màn hình toàn bộ task	Pass
3	Xóa task	Nhập mã task	Xóa và thông báo thành công	Pass
4	Phân bổ task cho nhân viên	Chọn nhân viên muốn phân bổ, nhập mã task	Phân bổ và thông báo thành công	Pass
5	Phân bổ task cho nhân viên cụ thể	Nhập mã nhân viên và mã task	Phân bổ và thông báo thành công	Pass
6	Sắp xếp nhân viên theo số lượng task hoàn thành(nhiều nhất)		Sắp xếp và in ra màn hình danh sách nhân viên	Pass
7	Sắp xếp nhân viên theo số lượng task phải làm(nhiều nhất)		Sắp xếp và in ra màn hình danh sách nhân viên	Pass
8	Xóa task ra khỏi nhân viên	Chọn nhân viên, nhập mã task	Xóa và thông báo thành công	Pass
9	Đánh giá task	Nhập mã task	Hiện menu để thực hiện đánh giá	Pass
10	Thoát khỏi menu quản lý task		Thoát ra menu chính	Pass

6.Kết luận

Qua những kiểm thử trên, chương trình hoạt động chuẩn và ổn định.

TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN

1.Tổng kết

- Nâng cao được kỹ năng làm việc nhóm: làm việc nhóm yêu cầu tất cả các thành viên phải cùng nhau góp sức – lên ý tưởng và ý kiến cho dự án. Bên cạnh đó còn nâng cao tinh thần đoàn kết hợp tác giữa các thành viên trong nhóm. Tổ chức và phân bố công việc hợp lý cho từng thành viên trong nhóm.
- Viết báo cáo dự án và làm slide thuyết trình: xác định đúng yêu cầu dự án, định hướng cấu trúc của một bài báo cáo. Kiểm tra lỗi về phong chữ, hình ảnh, số liệu,.. trong bản word và powerpoint.
- Áp dụng được những kiến thức đã học về lập trình hướng đối tượng và ngôn ngữ lập trình C++.
- Hiểu hơn về cách xây dựng UML.

2.Hướng phát triển

Sản phẩm hiện đã hoàn thiện được những chức năng cơ bản cần thiết cho việc quản lý nhân sự trong doanh nghiệp. Trong tương lai nhóm muốn định hướng phát triển thêm như:

- Nâng cao bảo mật dữ liệu cho hệ thống.
- Xây dựng giao diện thân thiện với người dùng hơn.
- Thêm chi tiết thông tin lịch sử sử dụng ứng dụng.
- Hệ thống làm việc trên thời gian thực.

TÀI LIỆU THAM KHẢO

[1] Slide bài giảng lập trình hướng đối tượng-TS. Nguyễn Duy Nhật Viễn

<https://nhatvien.com/courses/lap-trinh-huong-doi-tuong/>

[2] Slide Github trong dự án lập trình, Slide Phân tích thiết kế hướng đối tượng- TS. Nguyễn Duy Nhật Viễn

<https://nhatvien.com/courses/pbl1-lap-trinh/>

[3] ClearConsole() in C++ -function with clears console

[https://stackoverflow.com/questions/34653442/clearconsole-in-c-c-function-which-](https://stackoverflow.com/questions/34653442/clearconsole-in-c-c-function-which-clears-)

[clears-](https://stackoverflow.com/questions/34653442/clearconsole-in-c-c-function-which-clears-console?fbclid=IwAR1WBvu5XaHgvMJ5OQ48JJ6aiLKG4K0JHMrItJpOVk9DjPO-JvimppqaivU)
[console?fbclid=IwAR1WBvu5XaHgvMJ5OQ48JJ6aiLKG4K0JHMrItJpOVk9DjP](https://stackoverflow.com/questions/34653442/clearconsole-in-c-c-function-which-clears-console?fbclid=IwAR1WBvu5XaHgvMJ5OQ48JJ6aiLKG4K0JHMrItJpOVk9DjPO-JvimppqaivU)
[O-JvimppqaivU](https://stackoverflow.com/questions/34653442/clearconsole-in-c-c-function-which-clears-console?fbclid=IwAR1WBvu5XaHgvMJ5OQ48JJ6aiLKG4K0JHMrItJpOVk9DjPO-JvimppqaivU)

- Code tham khảo:

```
#include <windows.h>
```

```
void ClearScreen()
```

```
{  
    HANDLE          hStdOut;  
    CONSOLE_SCREEN_BUFFER_INFO csbi;  
    DWORD           count;  
    DWORD           cellCount;  
    COORD            homeCoords = { 0, 0 };  
    hStdOut = GetStdHandle( STD_OUTPUT_HANDLE );  
    if (hStdOut == INVALID_HANDLE_VALUE) return;  
    if (!GetConsoleScreenBufferInfo( hStdOut, &csbi )) return;  
    cellCount = csbi.dwSize.X *csbi.dwSize.Y;  
    if (!FillConsoleOutputCharacter(  
        hStdOut,  
        (TCHAR) ' ',  
        cellCount,  
        homeCoords,  
        &count  
    )) return;  
    if (!FillConsoleOutputAttribute(  
        hStdOut,  
        csbi.wAttributes,
```

```

    cellCount,
    homeCoords,
    &count
    )) return;
SetConsoleCursorPosition( hStdOut, homeCoords );
}

```

- Code của nhóm:

```

void XuLy::clearScreen()
{
    HANDLE hConsole = GetStdHandle(STD_OUTPUT_HANDLE);
    COORD coordScreen = {0, 0};
    DWORD cCharsWritten;
    CONSOLE_SCREEN_BUFFER_INFO csbi;
    DWORD dwConSize;

    if (!GetConsoleScreenBufferInfo(hConsole, &csbi))
    {
        throw runtime_error("Loi: GetConsoleScreenBufferInfo");
        return;
    }
    dwConSize = csbi.dwSize.X * csbi.dwSize.Y;

    if (!FillConsoleOutputCharacter(hConsole, (TCHAR)' ', dwConSize,
    coordScreen, &cCharsWritten))
    {
        throw runtime_error("Loi: FillConsoleOutputCharacter");
        return;
    }
    if (!FillConsoleOutputAttribute(hConsole, csbi.wAttributes, dwConSize,
    coordScreen, &cCharsWritten))
    {
        throw runtime_error("Loi: FillConsoleOutputAttribute");
        return;
    }
    SetConsoleCursorPosition(hConsole, coordScreen);
}

```

- Phần mới trong Code của nhóm: báo ngay lỗi tới terminal để dễ dàng thực hiện fixbug hơn.

[4] Sort lambda expresstion C++ by two conditions

https://stackoverflow.com/questions/43557771/sort-lambda-expression-c-by-two-condition?fbclid=IwAR02DrRlVbnqORovZqU08_X60gP_gs9IvvjoAHlEP2kWgVrXbk-NqQsUvhE

- Code tham khảo:

```
l_name->sort([](type*& s1, type*& s2)
{ if (s1->a() > s2->b())
return s1->a() > s2->b()
else if(s1->a() == s2->b())
return s1->c() > s2->d(); })
```

- Code của nhóm:

```
void XuLy::sapXepSoLuongNhanSuPBGD()
{
    HANDLE color = GetStdHandle(STD_OUTPUT_HANDLE);
    if (checkCoNhanSuHayKhong() == false)
    {
        return;
    }

    vector<PhongBan *> sortedPB = dsPB.getDsPhongBan();




    if (dsPB.getDsPhongBan().size() > 0)
    {
        sort(sortedPB.begin(), sortedPB.end(), [](PhongBan *pb1, PhongBan *pb2)
            { return pb1->getSoluongNhanSu() > pb2->getSoluongNhanSu(); });

        dsPB.setDsPhongBan(sortedPB);

        bool isSorted = true;
        int n = sortedPB.size();
        for (int i = 0; i < n - 1; i++)
        {
            if (sortedPB.front()->getSoluongNhanSu() <= sortedPB.back()->getSoluongNhanSu())
            {
                isSorted = false;
            }
        }
    }
}
```

```

        break;
    }
    if (sortedPB[i]->getSoluongNhanSu() > sortedPB[i + 1]-
>getSoluongNhanSu())
    {
        isSorted = true;
    }
    else if (sortedPB[i]->getSoluongNhanSu() < sortedPB[i + 1]-
>getSoluongNhanSu())
    {
        isSorted = false;
        break;
    }
}

if (isSorted)
{
    SetConsoleTextAttribute(color, 10);
    cout << endl;
    cout << "\t\t\tDa sap xep theo thu tu giam dan  " << endl;
    SetConsoleTextAttribute(color, 7);
    dsPB.xuatPB();
}
else
{
    SetConsoleTextAttribute(color, 12);
    cout << endl;
    cout << "\t\t\tDanh sach chua duoc sap xep  " << endl;
    SetConsoleTextAttribute(color, 7);
}
}
else
{
    SetConsoleTextAttribute(color, 12);
    cout << "\t\t\tDanh sach phong ban rong  " << endl;
    SetConsoleTextAttribute(color, 7);
}
}
}

```

- Phần mới trong code của nhóm:

+check xem trong list có nhân sự hay không rồi mới thực hiện sắp xếp.

+ check xem list đã sắp xếp đúng thứ tự chưa bằng cách so sánh đầu cuối, so sánh với các đối tượng tiếp theo.

+ Nếu thỏa mãn điều kiện thông báo sắp xếp thành công.

+ Nếu không thỏa mãn điều kiện thông báo sắp xếp không thành công.

[5] Check if a string contains a substring in C++

<https://stackoverflow.com/questions/2340281/check-if-a-string-contains-a-string-in-c>

- Code tham khảo:

```
if (s1.find(s2) != std::string::npos) {
    std::cout << "found!" << '\n';
}
```

- Code của nhóm:

bool XuLy::existedCheck(string &kitu) // Kiểm tra xem có tồn tại kí tự đó trong danh sách tên nhân sự

```
{
    for (NhanSu *ns : ds.getListNS())
    {
        if (ns->getHoTen().find(kitu) != string::npos)
        {
            return true;
        }
    }
}
```



```

    }
    return false;
}

void XuLy::timNhanSuTheoTen()
{
    HANDLE color = GetStdHandle(STD_OUTPUT_HANDLE);
    if (checkCoNhanSuHayKhong() == false)
    {
        return;
    }

    string kitu;
    vector<NhanSu *> listNSFound;
    cout << "\t\t\t\t ➡ Nhap ten muon tim kiem: ";

    SetConsoleTextAttribute(color, 11);
    cin.ignore();
    getline(cin, kitu);

    bool existed = existedCheck(kitu);
    if (existed == true)
    {
        cout << endl;
        SetConsoleTextAttribute(color, 10);
    }
}

```



```

        cout <<
"\t\t\t\t||=====||=====
==||\n";
    }
    else
    {
        cout <<
"\t\t\t\t||=====||=====
==||\n";
    }
}
else
{
    SetConsoleTextAttribute(color, 12);

    cout << "\t\t\t\tOOPS! Không tìm thấy nhân su voi ten " << kitu << " □" <<
endl;

    SetConsoleTextAttribute(color, 7);
}
}

```

- Phần mới trong code của nhóm: kiểm tra điều kiện được nhiều chuỗi bằng cách duyệt các chuỗi trong file

