

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**-----🙞🙞🙞🙞🙞-----**



**ĐỀ TÀI CUỐI KỲ MÔN LẬP TRÌNH TRÊN WINDOWS**

**TÊN ĐỀ TÀI:**

**HỆ THỐNG QUẢN LÝ NHÂN SỰ -**

**HUMAN RESOURCE MANAGEMENT SYSTEM**

**GVHD:** TS. Lê Văn Vinh

**Nhóm thực hiện:** Nhóm 16

**Học kỳ:** 1

**Năm học:** 2024 - 2025

*Thành phố Hồ Chí Minh, tháng 5, năm 2025*

**DANH SÁCH SINH VIÊN NHÓM THỰC HIỆN**

HỌC KÌ II NĂM HỌC 2024 - 2025

**Nhóm 16**

**Đề tài:** Hệ thống quản lý nhân sự

|  |  |
| --- | --- |
| **Họ tên sinh viên** | **Mã số sinh viên** |
| Trần Thị Kim Phượng | 22133044 |
| Nguyễn Thị Hồng Thơ | 22151305 |
| Nguyễn Ngọc Hiếu Hảo | 22133015 |

**NHẬN XÉT CỦA GV**

*Tp. Hồ Chí Minh, ngày …., tháng..., năm 2025*

**Giảng viên chấm điểm**

**MỤC LỤC**

[**CHƯƠNG 1: ĐẶC TẢ HỆ THỐNG QUẢN LÝ NHÂN SỰ** 6](#_Toc183175409)

[1. Mô tả bài toán 6](#_Toc183175410)

[2. Giao diện hệ thống 7](#_Toc183175411)

[2.1. Giao diện đối với nhân viên 7](#_Toc183175412)

[2.2. Giao diện đối với quản lý 7](#_Toc183175413)

[3. Tính lương nhân viên 8](#_Toc183175414)

[**CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG** 11](#_Toc183175415)

[1. Thiết kế cơ sở dữ liệu mức quan niệm 11](#_Toc183175416)

[2. Thiết kế cơ sở dữ liệu mức logic 11](#_Toc183175417)

[3. Các ràng buộc 12](#_Toc183175418)

[4. Cài đặt CSDL và các ràng buộc 15](#_Toc183175419)

[5. Các triggers 20](#_Toc183175420)

[5.1. Trigger tự động chấm công nghỉ phép có lương hoặc không lương khi nhân viên đăng ký nghỉ phép 20](#_Toc183175421)

[5.2. Trigger kiểm tra ngày bắt đầu và ngày kết thúc bảo hiểm 21](#_Toc183175422)

[5.3. Trigger kiểm tra mỗi phòng ban chỉ có một trưởng phòng, đảm bảo MaTrP là duy nhất giữa các phòng ban và đồng bộ MaTrP trong bảng PhongBan khi có nhân viên được cập nhật 21](#_Toc183175423)

[5.4. Trigger thông báo cho nhân viên nếu nhân viên có thưởng/phạt 23](#_Toc183175424)

[5.5. Trigger thông báo cho nhân viên nếu nhận được phụ cấp 24](#_Toc183175425)

[5.6. Trigger thông báo cho nhân viên nếu nhân viên đăng ký nghỉ phép thành công 25](#_Toc183175426)

[5.7. Trigger thông báo cho nhân viên nếu nhân viên chấm công thành công 25](#_Toc183175427)

[5.8. Trigger thông báo cho nhân viên nếu nhân viên đổi chức vụ 26](#_Toc183175428)

[5.9. Trigger kiểm tra đăng ký nghỉ phép nếu trùng với ngày nghỉ chính thức của công ty thì thông báo cho nhân viên 27](#_Toc183175429)

[5.10. Trigger tự động tạo tháng và tính số ngày công chuẩn nếu tháng nhân viên thực hiện chấm công chưa tồn tại 28](#_Toc183175430)

[6. Các views 30](#_Toc183175431)

[6.1. Hiển thị thông tin nhân viên 30](#_Toc183175432)

[6.2. Hiển thị thông tin các loại bảo hiểm 30](#_Toc183175433)

[6.3. Hiển thị thông tin hợp đồng 31](#_Toc183175434)

[**CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG** 32](#_Toc183175435)

[1. Kết nối cơ sở dữ liệu 32](#_Toc183175436)

[2. Xem thông tin 33](#_Toc183175437)

[2.1. Xem thông tin hợp đồng 33](#_Toc183175438)

[2.2. Xem thông tin nhân viên 34](#_Toc183175439)

[2.3. Xem lương thực lãnh 34](#_Toc183175440)

[2.4. Multi - Statement Table -Valued Function (có tham số) 35](#_Toc183175441)

[3. Quản lý nhân viên 36](#_Toc183175442)

[3.1. Thủ tục có tham số 36](#_Toc183175443)

[3.2. Inline Table – Valued Function (có tham số): Tìm kiếm nhân viên theo MaNV 41](#_Toc183175444)

[4. Quản lý phòng ban 42](#_Toc183175445)

[4.1. Thủ tục có tham số 42](#_Toc183175446)

[4.2. Multi - Statement Table -Valued Function (không tham số): Xem thông tin thống kê về phòng ban 47](#_Toc183175447)

[5. Quản lý thưởng phạt 48](#_Toc183175448)

[5.1. Thủ tục có tham số 48](#_Toc183175449)

[5.2. Inline Table – Valued Function (có tham số): Lọc chi tiết thưởng phạt theo mã nhân viên và theo loại thưởng/phạt 53](#_Toc183175450)

[6. Quản lý chức vụ 54](#_Toc183175451)

[6.1. Thêm chức vụ 54](#_Toc183175452)

[6.2. Cập nhật chức vụ 55](#_Toc183175453)

[6.3. Xóa chức vụ 56](#_Toc183175454)

[7. Quản lý phụ cấp 57](#_Toc183175455)

[7.1. Thêm phụ cấp 57](#_Toc183175456)

[7.2. Sửa phụ cấp 57](#_Toc183175457)

[7.3. Xóa phụ cấp 58](#_Toc183175458)

[7.4. Thêm chi tiết phụ cấp 59](#_Toc183175459)

[7.5. Sửa chi tiết phụ cấp 60](#_Toc183175460)

[7.6. Xóa chi tiết phụ cấp 60](#_Toc183175461)

[8. Quản lý chấm công 61](#_Toc183175462)

[8.1. Thủ tục có tham số 61](#_Toc183175463)

[8.2. Thủ tục không tham số, trả về một bảng: Xem danh sách chấm công của tất cả nhân viên 64](#_Toc183175464)

[8.3. Thủ tục có tham số, trả về một bảng: Tìm danh sách chấm công của một nhân viên 64](#_Toc183175465)

[9. Nhân viên chấm công ngày làm việc 65](#_Toc183175466)

[9.1. Chấm công 65](#_Toc183175467)

[9.2. Xem danh sách chấm công theo Mã Nhân viên 65](#_Toc183175468)

[10. Quản lý bảo hiểm 66](#_Toc183175469)

[10.1. Xem danh sách bảo hiểm 66](#_Toc183175470)

[10.2. Thủ tục có tham số, trả về một bảng: Tìm thông tin bảo hiểm của một nhân viên 66](#_Toc183175471)

[10.3. Thủ tục có tham số, không trả về kết quả 66](#_Toc183175472)

[11. Nhân viên đăng ký nghỉ phép 68](#_Toc183175473)

[11.1. Thủ tục có tham số, trả về một bảng: Xem lịch sử nghỉ phép 68](#_Toc183175474)

[11.2. Thủ tục có tham số: Xin nghỉ phép 69](#_Toc183175475)

[12. Thống kê lương cho nhân viên 69](#_Toc183175476)

[12.1. Thủ tục có tham số trả về một bảng: Tính lương nhân viên theo tháng 69](#_Toc183175477)

[12.2. Thủ tục không tham số, trả về nhiều bảng: Xem chi tiết lương 72](#_Toc183175478)

[13. Quản lý hợp đồng 76](#_Toc183175479)

[13.1. Thêm hợp đồng 76](#_Toc183175480)

[13.2. Sửa hợp đồng 77](#_Toc183175481)

[13.3. Xóa hợp đồng 77](#_Toc183175482)

[14. Quản lý người phụ thuộc 78](#_Toc183175483)

[14.1. Thêm người phụ thuộc 78](#_Toc183175484)

[14.2. Xóa người phụ thuộc 79](#_Toc183175485)

[**CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN** 80](#_Toc183175486)

[1. Tạo Role và phân quyền 80](#_Toc183175487)

[1.1. Nhân viên (Employee) 80](#_Toc183175488)

[1.2. Trưởng phòng (DepartmentHead) 80](#_Toc183175489)

[1.3. Quản lý (Admin) 80](#_Toc183175490)

[2. Tạo LOGIN và USER mẫu cho từng ROLE 80](#_Toc183175491)

[3. Viết Trigger tự động phân quyền khi thêm nhân viên mới 81](#_Toc183175492)

[3.1. Trigger tạo Nhân viên thì tạo Tài khoản 81](#_Toc183175493)

[3.2. Trigger thêm tài khoản thì thêm phân quyền 81](#_Toc183175494)

[3.3. Trigger đồng bộ mật khẩu đăng nhập 82](#_Toc183175495)

[3.4. Thủ tục xóa nhân viên 83](#_Toc183175496)

[4. Kiểm tra 83](#_Toc183175497)

[**CHƯƠNG 5: THIẾT KẾ GIAO DIỆN HỆ THỐNG** 86](#_Toc183175498)

[1. Form đăng nhập 86](#_Toc183175499)

[2. Form chính 86](#_Toc183175500)

[3. Quản lý nhân viên 88](#_Toc183175501)

[4. Quản lý phòng ban 89](#_Toc183175502)

[4.1. Xem thông tin chi tiết thống kê về các phòng ban: 90](#_Toc183175503)

[4.2. Form thông báo đến nhân viên của phòng ban 91](#_Toc183175504)

[5. Quản lý thưởng phạt 92](#_Toc183175505)

[5.1. Form thưởng phạt chính 92](#_Toc183175506)

[5.2. Form thưởng phạt sau khi lọc loại thưởng hoặc phạt theo Mã nhân viên 92](#_Toc183175507)

[6. Quản lý chức vụ 93](#_Toc183175508)

[7. Quản lý phụ cấp 93](#_Toc183175509)

[8. Quản lý chấm công 94](#_Toc183175510)

[9. Quản lý bảo hiểm 96](#_Toc183175511)

[10. Quản lý lương nhân viên 97](#_Toc183175512)

[11. Quản lý hợp đồng 98](#_Toc183175513)

[12. Quản lý người phụ thuộc 99](#_Toc183175514)

[13. Quản lý tháng và công 99](#_Toc183175515)

[14. Cập nhật mật khẩu nhân viên 100](#_Toc183175516)

[15. Nhân viên xem lương 100](#_Toc183175517)

[16. Nhân viên chấm công 101](#_Toc183175518)

[17. Nhân viên đăng ký nghỉ phép 101](#_Toc183175519)

[18. Nhân viên xem hợp đồng 102](#_Toc183175520)

[19. Xem thông báo 103](#_Toc183175521)

# **CHƯƠNG 1: ĐẶC TẢ HỆ THỐNG QUẢN LÝ NHÂN SỰ**

## 1. Mô tả bài toán

Con người là một trong những nguồn tài nguyên quan trọng nhất trong mỗi công ty. Do đó, để hoạt động hiệu quả, một công ty cần có một hệ thống quản lý nhân sự. Hệ thống này được thiết kế để phục vụ hai nhóm đối tượng là quản lý và nhân viên. Mỗi nhóm sẽ có những quyền hạn và chức năng riêng biệt, đảm bảo sự hiệu quả và bảo mật trong quản lý và vận hành.

Một hệ thống quản lý nhân sự công ty cần lưu trữ các thành phần dữ liệu quan trọng về nhân viên và các yếu tố liên quan để quản lý hiệu quả:

Mỗi nhân viên sẽ được phân biệt bằng một mã nhân viên duy nhất, nhân viên có thông tin cá nhân gồm họ tên nhân viên, giới tính, ngày sinh, địa chỉ, số điện thoại, email, số căn cước công dân, phòng ban làm việc, chức vụ, hợp đồng. Một nhân viên có thể có nhiều người phụ thuộc. Người phụ thuộc có họ tên, ngày sinh, mối quan hệ với nhân viên. Loại tài khoản có mã loại, tên. Mỗi nhân viên có một tài khoản riêng để sử dụng hệ thống, thuộc một loại tài khoản và xác định bởi tên đăng nhập và mật khẩu. Mỗi nhân viên chỉ thuộc về một phòng ban nhưng một phòng ban có một hoặc nhiều nhân viên. Mỗi phòng ban phân biệt bằng mã phòng ban, và có tên phòng ban, số điện thoại liên lạc, trưởng phòng. Mỗi nhân viên đảm nhận một chức vụ, nhưng một chức vụ có thể bị trống hoặc được đảm nhận bởi nhiều nhân viên. Chức vụ được phân biệt bằng mã chức vụ, tên chức vụ. Mỗi nhân viên ký kết một hợp đồng. Các hợp đồng phân biệt với nhau bằng mã hợp đồng, trên hợp đồng có ngày bắt đầu, ngày kết thúc, lương cơ bản. Một nhân viên có một hoặc nhiều bảo hiểm. Bảo hiểm gồm mã loại, tên bảo hiểm. Bảo hiểm cho mỗi nhân viên có mã bảo hiểm, ngày cấp bảo hiểm và ngày hết hạn.

Chấm công, nghỉ phép, thưởng phạt và phụ cấp của nhân viên được quản lý theo từng tháng. Mỗi tháng gồm mã tháng, mô tả, số ngày công chuẩn. Mỗi thưởng phạt phân biệt bằng mã thưởng phạt, có thông tin về số tiền, lý do. Một nhân viên có thể có nhiều thưởng phạt, mỗi chi tiết thưởng, phạt có thông tin về ngày thưởng phạt. Phụ cấp gồm mã phụ cấp, loại phụ cấp (vé máy bay, xăng,...). Một nhân viên có thể có nhiều phụ cấp, mỗi chi tiết phụ cấp có thông tin về số tiền, ngày phụ cấp. Một nhân viên có thể xin nghỉ phép nhiều lần, mỗi nghỉ phép có ngày nghỉ phép, ghi chú. Có nhiều loại công, được quản lý bằng bảng Chấm Công, mỗi chấm công có mã chấm công, mô tả và hệ số. Một nhân viên có thể thực hiện nhiều chấm công trong một tháng, nhưng chỉ chấm công nhiều nhất một lần trong ngày.

Bên cạnh đó, hệ thống còn có các chức năng khác nhau cho các đối tượng khác nhau. Đối với người dùng là nhân viên, sẽ có quyền xem thông tin cá nhân, xem chi tiết hợp đồng, lịch sử nghỉ phép và lương thực lãnh. Nhân viên có thể đăng nhập vào tài khoản và chấm công hay cập nhật mật khẩu cho tài khoản của mình. Nếu người dùng là quản lý thì có quyền truy cập vào các chức năng như quản lý nhân viên, quản lý phòng ban, quản lý chức vụ, quản lý hợp đồng, quản lý chấm công, quản lý thưởng/phạt, quản lý bảo hiểm và tính lương của nhân viên. Về cách tính lương nhân viên thì số tiền lương thực lãnh của một nhân viên được xác định bằng lương cơ bản o hợp đồng, số ngày công, tiền thưởng phạt, phụ cấp, bảo hiểm, thuế thu nhập cá nhân, được hệ thống tính theo công thức:

**Lương thực lãnh** = Lương cơ bản + Tiền thưởng + Phụ cấp chức vụ - Bảo hiểm - Tiền phạt - Thuế thu nhập cá nhân.

## 2. Giao diện hệ thống

### 2.1. Giao diện đối với nhân viên

**Cửa sổ Thông tin nhân viên**: gồmcác hộp văn bản chứa các thông tin cá nhân của nhân viên.

**Cửa sổ Chi tiết hợp đồng**: có hộp văn bản chứa các thông tin về hợp đồng của nhân viên.

**Cửa sổ Đăng ký nghỉ phép** có trường nhập liệu: Nhập ngày bắt đầu, ngày kết thúc, lý do nghỉ, chọn ngày nghỉ phép (nghỉ có trả lương hoặc nghỉ không trả lương), và nút “Đăng ký”. Thông báo sẽ hiện lên nếu đăng ký thành công.

**Cửa sổ Chấm công** có bảng hiện danh sách ngày công của nhân viên đó, có nút chấm công. Khi nhấn chấm công, nếu thành công thì hệ thống cập nhật lại danh sách, ngược lại thì hiện thông báo thất bại.

**Cửa sổ Thông tin lương** có tất cả thông tin chi tiết về lương của nhân viên đó, bao gồm cả bảo hiểm, và bảng chấm công trong một tháng cụ thể. Và mục Lương thực lãnh cho nhân viên biết lương được lãnh của tháng đó là bao nhiêu.

**Cửa sổ Xem thông báo** có danh sách các thông báo được gửi đến cho phòng ban của nhân viên đó, bao gồm tiêu đề, nội dung chi tiết và ngày nhận được thông báo từ trưởng phòng hay quản lý.

### 2.2. Giao diện đối với quản lý

**Cửa sổ Quản lý thông tin nhân viên** gồm hộp văn bản chứa thông tin về nhân viên. Cửa sổ quản lý thông tin nhân viên còn có một danh sách các nhân viên và dữ liệu có thể được thao tác ở phần chức năng. Phần chức năng bao gồm các nút như: "Thêm", "Sửa", "Xóa", "Xem thông tin chi tiết", "Làm mới", và "Thoát". Phần tìm kiếm có trường nhập liệu “Mã nhân viên”, và nút “Tìm kiếm”, nếu tìm thấy, thông tin nhân viên sẽ hiển thị; nếu không, thông báo không tìm thấy.

**Cửa sổ Quản lý phòng ban** có mục "Thông tin phòng ban" là phần nhập liệu nằm trên cùng, bao gồm thông tin về phòng ban. Dưới phần nhập liệu là bảng liệt kê danh sách các phòng ban hiện có. Phần chức năng bao gồm các nút như: "Thêm" , "Sửa", "Xóa", "Xem thông tin chi tiết", "Làm mới", và "Thoát". Ngoài ra, có nút "Thông báo" để điều chỉnh thông báo theo phòng ban. Khi bấm nút "Thông báo", một cửa sổ sẽ xuất hiện hiển thị các thông báo đến phòng ban và các nút như "Thêm", "Xóa", và "Sửa" thông báo. Khi bấm nút "Thêm thông báo", một form sẽ xuất hiện để nhập “Tiêu đề”, “Nội dung”, “Phòng ban gửi”, “Ngày gửi”. Để sửa, chọn thông báo đã gửi, chỉnh sửa và lưu. Nút "Xóa" loại bỏ thông báo không cần thiết.

**Cửa sổ Quản lý chức vụ** gồm: phần "Thông tin chức vụ" ở đầu giao diện, nơi nhập thông tin. Bên dưới là các nút chức năng như "Thêm", "Xóa", "Sửa" các thông tin liên quan đến chức vụ, "Làm mới" để làm mới giao diện và xóa hết các thông tin đang hiển thị, và "Thoát". Ở phần cuối giao diện, danh sách chức vụ hiện có trong hệ thống được hiển thị dưới dạng bảng. Người dùng có thể chọn một dòng trong danh sách và sử dụng các nút chức năng để thêm, xóa, sửa hoặc làm mới thông tin chức vụ.

**Cửa sổ Quản lý khen thưởng/ kỷ luật** được chia làm hai phần là Thưởng Phạt và Chi tiết Thưởng Phạt. Mỗi phần có các khung để nhập hay hiển thị thông tin chi tiết; có bảng liệt kê danh sách thưởng phạt của công ty, danh sách chi tiết thưởng phạt của nhân viên; có nút thêm, sửa, xóa, khi nút chức năng được nhấn thực hiện thì hệ thống thực hiện yêu cầu và hiện thông báo thành công hoặc thất bại. Ngoài ra, còn có checkbox và nút “Lọc loại thưởng/phạt” để lọc danh sách thưởng hay là phạt theo mã nhân viên.

**Cửa sổ Quản lý phụ cấp** chia làm hai phần là Thông Tin Loại Phụ Cấp và Thông Tin Phụ Cấp. Mỗi phần có các khung để nhập hay hiển thị thông tin chi tiết; có bảng liệt kê danh sách các loại phụ cấp, danh sách chi tiết phụ cấp của nhân viên; có nút thêm, sửa, xóa, khi nút chức năng được nhấn thực hiện thì hệ thống thực hiện yêu cầu và hiện thông báo thành công hoặc thất bại.

**Cửa sổ Quản lý hợp đồng** có các khung để nhập hay hiển thị thông tin chi tiết của một hợp đồng; có nút thêm, sửa, tìm, xem thông tin chi tiết; có danh sách hợp đồng. Khi người dùng nhấn tìm, nếu hệ thống tìm thấy hợp đồng dựa trên mã nhân viên thì hiển thị ra các khung thông tin, ngược lại hiện thông báo không tìm thấy. Khi người dùng bấm vào xem thông tin chi tiết sẽ hiện ra cửa sổ thông tin chi tiết về loại hợp đồng.

**Cửa sổ Quản lý bảo hiểm** có các khung để nhập hay hiển thị thông tin chi tiết của một bảo hiểm trong công ty; có các nút tìm, thêm, sửa, xóa; có bảng hiện danh sách bảo hiểm. Khi người dùng nhấn tìm, hệ thống sẽ hiển thị danh sách bảo hiểm tìm thấy dựa trên mã nhân viên đã chọn. Khi nhấn nếu thêm/ sửa/ xóa, nếu hệ thống thực hiện thành công thì hiện thông báo “Thành công”, ngược lại hiện thông báo “Thất bại”.

**Cửa sổ Quản lý tháng và công** chia thành hai phần: Phần Quản lý Tháng có khung mã tháng, khung mô tả, khung số ngày công chuẩn; có danh sách các tháng đã tạo, có nút thêm, sửa; sau khi nhấn thêm/ sửa, hệ thống thực hiện và hiển thị thông báo thành công hoặc thất bại. Phần Quản lý Công có khung chọn kỳ, khung chọn nhân viên, khung hiện số ngày công trong kỳ; có nút tìm; có bảng hiện danh sách công; khi nhấn tìm nếu thành công thì hệ thống cập nhật khung số ngày công và danh sách công, ngược lại hiện thông báo thất bại.

**Cửa sổ Quản lý lương nhân viên** gồm nút xem lương và khung có danh sách thông tin về lương của nhân viên bao gồm lương chịu thuế, thuế thu nhập cá nhân và lương thực lãnh. Ngoài ra, có nút xem chi tiết để xem chi tiết từng phần lương của nhân viên.

**Cửa sổ Quản lý người phụ thuộc** gồm phần nhập liệu và danh sách về thông tin người phụ thuộc của nhân viên công ty. Ngoài ra, có nút thêm, xóa, để cập nhật thông tin về người phụ thuộc của nhân viên.

**Cửa sổ Cập nhật mật khẩu** bao gồm trường chọn nhân viên và trường nhập mật khẩu mới, xác nhận mật khẩu để cập nhật mật khẩu cho tài khoản nhân viên công ty quản lý.

## 3. Tính lương nhân viên

Các yếu tố để tính lương nhân viên trong công ty bao gồm:

* **Lương cơ bản**: Đây là số tiền nhân viên nhận được cho công việc cố định mà họ đảm nhiệm theo số ngày công chuẩn được thỏa thuận. Lương cơ bản được là lương được định sẵn trong hợp đồng.
* **Tiền thưởng phạt**: Nhân viên có thể nhận thưởng hoặc phạt tùy theo quy định công ty.
* **Phụ cấp chức vụ**: Nhân viên sẽ nhận được thêm phụ cấp khi đi công tác cần hỗ trợ chi phí như đi lại, ăn uống,....
* **Tiền bảo hiểm[[1]](#footnote-1)**: Đây là khoản tiền nhân viên phải đóng cho bảo hiểm xã hội: 8%, bảo hiểm y tế: 1,5%  và bảo hiểm thất nghiệp 1% theo lương cá nhân
* **Thuế thu nhập cá nhân[[2]](#footnote-2)**: Được tính dựa trên tổng thu nhập, trừ đi cả khoản miễn giảm (như giảm trừ gia cảnh). Mức thuế suất cụ thể được áp dụng như sau:

**Biểu thuế lũy tiến từng phần**

|  |  |
| --- | --- |
| **Thu nhập tính thuế (VNĐ)** | **Thuế suất (%)** |
| Đến 5 triệu | 5% |
| Trên 5 triệu đến 10 triệu | 10% |
| Trên 10 triệu đến 18 triệu | 15% |
| Trên 18 triệu đến 32 triệu | 20% |
| Trên 32 triệu đến 52 triệu | 25% |
| Trên 52 triệu đến 80 triệu | 30% |
| Trên 80 triệu | 35% |

Công thức tổng quát:

**Lương thực lãnh** = Lương tháng + Tiền thưởng + Phụ cấp chức vụ - Bảo hiểm - Tiền phạt - Thuế thu nhập cá nhân.

Ví dụ: Giả sử một nhân viên là trưởng phòng tại công ty, mức lương cơ sở là 15.000.000 đồng, tháng này có 24 ngày công chuẩn. Nhân viên đó tháng này đi làm 23 ngày, nghỉ một ngày không lương, được thưởng 500.000 đồng do hoàn thành tốt công việc và nhận thêm phụ cấp chức vụ là 2.000.000 đồng. Ngoài ra, nhân viên này có 2 người phụ thuộc là hai người con dưới 18 tuổi.

**Tính lương cơ bản tháng**Lương cơ bản tháng = Lương thỏa thuận trong hợp đồng / Số ngày công chuẩn × Số ngày làm việc thực tế  
Lương cơ bản tháng = 15.000.000 / 24 × 23 = **14.375.000 đồng**

**Tính bảo hiểm bắt buộc**Bảo hiểm = Lương cơ bản tháng × 10.5%  
Bảo hiểm = 14.375.000 × 10.5% = **1.509.375 đồng**

**Tính giảm trừ gia cảnh**

Giảm trừ cho bản thân: 11.000.000 đồng

Giảm trừ cho 2 người phụ thuộc (4.400.000 đồng/người × 2): 8.800.000 đồng

Tổng giảm trừ gia cảnh = 11.000.000 + 8.800.000 = 19.800.000 đồng

**Tính thuế thu nhập cá nhân**

Thuế thu nhập cá nhân = Lương chịu thuế \* Mức thuế suất tương ứng

Lương chịu thuế = Lương cơ bản - Giảm trừ gia cảnh - Bảo hiểm

Lương chịu thuế = 14.375.000 - 19.800.000 - 1.509.375 = -6.934.375 đồng

Vì thu nhập tính thuế là âm, nhân viên không phải đóng thuế thu nhập cá nhân.

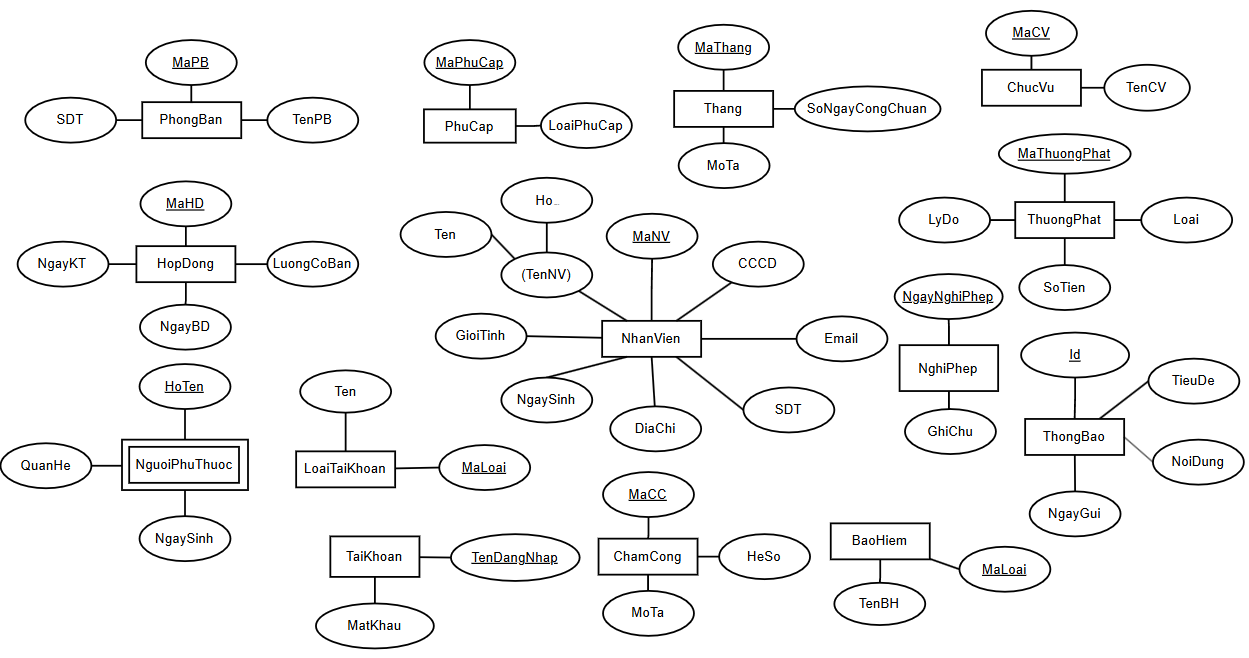
**Tính lương thực lãnh**

Lương thực lãnh = Lương tháng + Tiền thưởng + Phụ cấp chức vụ - Bảo hiểm - Thuế TNCN

Lương thực lãnh = 14.375.000 + 500.000 + 2.000.000 - 1.509.375 - 0 = 15.365.625 đồng.

# **CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

## 1. Thiết kế cơ sở dữ liệu mức quan niệm



A diagram of a diagram

Description automatically generated

## 2. Thiết kế cơ sở dữ liệu mức logic

Từ sơ đồ thực thể kết hợp (ERD), ta có các lược đồ quan hệ:

* ChucVu (MaCV, TenCV)
* NhanVien (MaNV, Ho, Ten, GioiTinh, NgaySinh, DiaChi, SDT, Email, CCCD, MaPB, MaCV, MaHD)
* HopDong (MaHD, MaNV, LuongCoBan, NgayBD, NgayKT)
* PhongBan (MaPB, TenPB, SDT, MaTrP)
* NguoiPhuThuoc (MaNV, HoTen, NgaySinh, QuanHe)
* ThongBao(Id, TieuDe, NoiDung, MaPB, NgayGui)
* Thang(MaThang, MoTa, SoNgayCongChuan)
* ThuongPhat (MaThuongPhat, Loai, SoTien, LyDo)
* ctThuongPhat (MaNV, MaThuongPhat, MaThang, NgayThuongPhat)
* ChamCong (MaCC, MoTa, HeSo)
* ctChamCong (MaNV, MaCC, MaThang, NgayChamCong)
* BaoHiem (MaLoai, TenBH)
* ctBaoHiem (MaNV, MaBH, MaLoai, NgayBD, NgayKT)
* NghiPhep(MaNV, MaThang, NgayNghiPhep, GhiChu)
* LoaiTaiKhoan(MaLoai, Ten)
* TaiKhoan (TenDangNhap, MatKhau, MaLoai)
* PhuCap(MaPhuCap, LoaiPhuCap)
* ctPhuCap(MaNV, MaPhuCap, MaThang, NgayPhuCap, SoTien)

## 3. Các ràng buộc

|  |  |  |
| --- | --- | --- |
| **STT** | **BẢNG** | **RÀNG BUỘC** |
| **1** | **ChucVu** | Khóa chính MaCV: nvarchar(10),  NOT NULL TenCV: nvarchar(50) |
| **2** | **NhanVien** | Khóa chính MaNV: nvarchar(10),  NOT NULL Ho: nvarchar(50),  NOT NULL Ten: nvarchar(50),  NOT NULL GioiTinh: nvarchar(10),  NOT NULL NgaySinh: date,  NOT NULL DiaChi: nvarchar(100),  NOT NULL SDT: nvarchar(20),  CONSTRAINT RightPhoneNum CHECK(len(SDT)=10),  NOT NULL UNIQUE Email: nvarchar(50),  NOT NULL UNIQUE CCCD: nvarchar(12),  NOT NULL MaPB: nvarchar(10).  Khóa ngoại MaPB tham chiếu đến bảng PhongBan,  NOT NULL MaCV: nvarchar(10),  Khóa ngoại MaCV tham chiếu đến bảng ChucVu,  NOT NULL MaHD: nvarchar(10),  Khóa ngoại MaHD tham chiếu đến bảng HopDong |
| **3** | **HopDong** | Khóa chính MaHD: nvarchar(10),  NOT NULL UNIQUE MaNV: nvarchar(10),  NOT NULL LuongCoBan: int,  NOT NULL NgayBD: date,  NOT NULL NgayKT: date,  Khóa ngoại MaNV tham chiếu đến bảng NhanVien |
| **4** | **PhongBan** | Khóa chính MaPB: nvarchar(10),  NOT NULL TenPB: nvarchar(50),  NOT NULL SDT: nvarchar(20),  CONSTRAINT ValidPhoneNum CHECK(len(SDT)=10),  MaTrP: nvarchar(10),  Khóa ngoại MaTrP tham chiếu đến bảng NhanVien |
| **5** | **NguoiPhuThuoc** | NOT NULL MaNV: nvarchar(10),  NOT NULL HoTen: nvarchar(50),  Khóa chính (MaNV, HoTen),  NOT NULL NgaySinh: date,  NOT NULL QuanHe: nvarchar(50),  Khóa ngoại MaNV tham chiếu đến bảng NhanVien |
| **6** | **ThongBao** | Khóa chính Id: int identity(1,1),  TieuDe: nvarchar(100),  NoiDung: nvarchar(max),  MaPB: nvarchar(10),  NgayGui: datetime,  Khóa ngoại MaPB tham chiếu đến bảng PhongBan |
| **7** | **Thang** | Khóa chính MaThang: nvarchar(6),  MoTa: nvarchar(MAX),  NOT NULL SoNgayCongChuan: int,  CHECK(SoNgayCongChuan BETWEEN 1 AND 31) |
| **8** | **ThuongPhat** | Khóa chính MaThuongPhat: nvarchar(10),  NOT NULL Loai: nvarchar(50),  NOT NULL SoTien: int,  NOT NULL LyDo: nvarchar(MAX) |
| **9** | **ctThuongPhat** | NOT NULL MaNV: nvarchar(10),  NOT NULL MaThuongPhat: nvarchar(10),  NOT NULL MaThang: nvarchar(6),  NOT NULL NgayThuongPhat: int,  Khóa chính (MaNV, MaThuongPhat, MaThang, NgayThuongPhat),  CHECK(NgayThuongPhat BETWEEN 1 AND 31),  Khóa ngoại MaNV tham chiếu đến bảng NhanVien,  Khóa ngoại MaThuongPhat tham chiếu đến bảng ThuongPhat  Khóa ngoại MaThang tham chiếu đến bảng Thang |
| **10** | **ChamCong** | Khóa chính MaCC: nvarchar(10),  MoTa: nvarchar(MAX),  NOT NULL HeSo: float |
| **11** | **ctChamCong** | NOT NULL MaNV: nvarchar(10),  NOT NULL MaCC: nvarchar(10),  NOT NULL MaThang: nvarchar(6),  NOT NULL NgayChamCong: int,  CHECK(NgayChamCong BETWEEN 1 AND 31),  Khóa chính (MaNV, MaThang, NgayChamCong),  Khóa ngoại MaNV đến bảng NhanVien,  Khóa ngoại MaCC tham chiếu đến bảng ChamCong,  Khóa ngoại MaThang tham chiếu đến bảng Thang |
| **12** | **BaoHiem** | Khóa chính MaLoai: nvarchar(10),  NOT NULL TenBH: nvarchar(50) |
| **13** | **ctBaoHiem** | NOT NULL MaNV: nvarchar(10),  Khóa chính MaBH: nvarchar(10),  NOT NULL MaLoai: nvarchar(10),  NOT NULL NgayBD: date,  NOT NULL NgayKT: date,  Khóa ngoại MaNV tham chiếu đến bảng NhanVien,  Khóa ngoại MaLoai tham chiếu đến bảng BaoHiem |
| **14** | **NghiPhep** | NOT NULL MaNV: nvarchar(10),  NOT NULL MaThang: nvarchar(6),  NOT NULL NgayNghiPhep: int,  CHECK(NgayNghiPhep BETWEEN 1 AND 31),  Khóa chính (MaNV, MaThang, NgayNghiPhep),  NOT NULL GhiChu: nvarchar(MAX),  Khóa ngoại MaNV tham chiếu đến bảng NhanVien,  Khóa ngoại MaThang tham chiếu đến bảng Thang |
| **15** | **LoaiTaiKhoan** | Khóa chính MaLoai: nvarchar(10),  NOT NULL Ten: nvarchar(50) |
| **16** | **TaiKhoan** | Khóa chính TenDangNhap: nvarchar(10),  NOT NULL MatKhau: nvarchar(255),  NOT NULL MaLoai: nvarchar(10),  Khóa ngoại TenDangNhap tham chiếu đến bảng NhanVien,  Khóa ngoại MaLoai tham chiếu đến bảng LoaiTaiKhoan |
| **17** | **PhuCap** | Khóa chính MaPhuCap: nvarchar(10),  NOT NULL LoaiPhuCap: nvarchar(10) |
| **18** | **ctPhuCap** | NOT NULL MaNV: nvarchar(10),  NOT NULL MaPhuCap: nvarchar(10),  NOT NULL MaThang: nvarchar(6),  NOT NULL NgayPhuCap: int,  CHECK(NgayPhuCap BETWEEN 1 AND 31),  Khóa chính (MaNV, MaPhuCap, MaThang, NgayPhuCap),  NOT NULL SoTien: int,  Khóa ngoại MaNV tham chiếu đến bảng NhanVien,  Khóa ngoại MaPhuCap tham chiếu đến bảng PhuCap,  Khóa ngoại MaThang tham chiếu đến bảng Thang |

## 4. Cài đặt CSDL và các ràng buộc

***Bảng Chức Vụ***

|  |
| --- |
| CREATE TABLE ChucVu (  MaCV NVARCHAR(10) PRIMARY KEY,  TenCV NVARCHAR(50) NOT NULL  ); |

***Bảng Nhân Viên***

|  |
| --- |
| CREATE TABLE NhanVien (  MaNV NVARCHAR(10) PRIMARY KEY,  Ho NVARCHAR(50) NOT NULL,  Ten NVARCHAR(50) NOT NULL,  GioiTinh NVARCHAR(10) NOT NULL,  NgaySinh DATE NOT NULL,  DiaChi NVARCHAR(100) NOT NULL,  SDT NVARCHAR(20) NOT NULL UNIQUE,  Email NVARCHAR(50) NOT NULL UNIQUE,  CCCD NVARCHAR(12) NOT NULL UNIQUE,  MaPB NVARCHAR(10),  MaCV NVARCHAR(10) NOT NULL,  MaHD NVARCHAR(10),  CONSTRAINT RightPhoneNum CHECK (LEN(SDT) = 10)  ); |

***Bảng Hợp Đồng***

|  |
| --- |
| CREATE TABLE HopDong (  MaHD NVARCHAR(10) PRIMARY KEY,  MaNV NVARCHAR(10) UNIQUE NOT NULL,  LuongCoBan INT NOT NULL,  NgayBD DATE NOT NULL,  NgayKT DATE NOT NULL  ); |

***Bảng Phòng Ban***

|  |
| --- |
| CREATE TABLE PhongBan (  MaPB NVARCHAR(10) PRIMARY KEY,  TenPB NVARCHAR(50) NOT NULL,  SDT NVARCHAR(20) NOT NULL,  MaTrP NVARCHAR(10) UNIQUE,  CONSTRAINT ValidPhoneNum CHECK (LEN(SDT) = 10)  ); |

***Khóa ngoại Bảng Phòng Ban***

|  |
| --- |
| ALTER TABLE PhongBan ADD CONSTRAINT fk\_MaTrP  FOREIGN KEY (MaTrP) REFERENCES NhanVien(MaNV)  ON UPDATE NO ACTION  ON DELETE NO ACTION; |

***Khóa ngoại Bảng Hợp đồng***

|  |
| --- |
| ALTER TABLE HopDong ADD CONSTRAINT fk\_HD\_MaNV  FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV)  ON UPDATE NO ACTION  ON DELETE CASCADE; |

***Khóa ngoại Bảng Nhân Viên***

|  |
| --- |
| ALTER TABLE NhanVien ADD CONSTRAINT fk\_NV\_MaPB  FOREIGN KEY (MaPB) REFERENCES PhongBan (MaPB)  ON UPDATE NO ACTION  ON DELETE NO ACTION;  ALTER TABLE NhanVien ADD CONSTRAINT fk\_NV\_MaCV  FOREIGN KEY (MaCV) REFERENCES ChucVu (MaCV)  ON UPDATE NO ACTION  ON DELETE NO ACTION;  ALTER TABLE NhanVien ADD CONSTRAINT fk\_NV\_MaHD  FOREIGN KEY (MaHD) REFERENCES HopDong (MaHD)  ON UPDATE NO ACTION  ON DELETE NO ACTION; |

***Bảng Người Phụ Thuộc***

|  |
| --- |
| CREATE TABLE NguoiPhuThuoc (  MaNV NVARCHAR(10) NOT NULL,  HoTen NVARCHAR(50) NOT NULL,  NgaySinh DATE NOT NULL,  QuanHe NVARCHAR(50) NOT NULL,  PRIMARY KEY (MaNV, HoTen),  CONSTRAINT fK\_NPT\_MaNV FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV)  ON UPDATE NO ACTION  ON DELETE CASCADE  ); |

***Bảng Thông báo***

CREATE TABLE ThongBao (Id INT PRIMARY KEY IDENTITY(1,1),

TieuDe NVARCHAR(100),

NoiDung NVARCHAR(MAX),

MaPB NVARCHAR(10),

NgayGui DATETIME,

CONSTRAINT fK\_TB\_MaPB FOREIGN KEY (MaPB) REFERENCES PhongBan(MaPB)

);

***Bảng Tháng***

|  |
| --- |
| CREATE TABLE Thang (  MaThang NVARCHAR(6) PRIMARY KEY,  MoTa NVARCHAR(MAX),  SoNgayCongChuan INT NOT NULL CHECK(SoNgayCongChuan BETWEEN 1 AND 31)  ); |

***Bảng Thưởng Phạt***

|  |
| --- |
| CREATE TABLE ThuongPhat (  MaThuongPhat NVARCHAR(10) PRIMARY KEY,  Loai NVARCHAR(50) NOT NULL,  SoTien INT NOT NULL,  LyDo NVARCHAR(MAX) NOT NULL  ); |

***Bảng Chi Tiết Thưởng Phạt***

|  |
| --- |
| CREATE TABLE ctThuongPhat (  MaNV NVARCHAR(10) NOT NULL,  MaThuongPhat NVARCHAR(10) NOT NULL,  MaThang NVARCHAR(6) NOT NULL,  NgayThuongPhat INT NOT NULL,  PRIMARY KEY (MaNV, MaThuongPhat, MaThang, NgayThuongPhat),  CONSTRAINT ValidNgay CHECK (NgayThuongPhat BETWEEN 1 AND 31),  CONSTRAINT fK\_ctTP\_MaNV FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV)  ON UPDATE CASCADE  ON DELETE CASCADE,  CONSTRAINT fK\_ctTP\_MaThuongPhat FOREIGN KEY (MaThuongPhat) REFERENCES ThuongPhat(MaThuongPhat)  ON UPDATE CASCADE  ON DELETE NO ACTION,  CONSTRAINT fK\_ctTP\_MaThang FOREIGN KEY (MaThang) REFERENCES Thang(MaThang)  ON UPDATE CASCADE  ON DELETE NO ACTION  ); |

***Bảng Chấm Công***

|  |
| --- |
| CREATE TABLE ChamCong (  MaCC NVARCHAR(10) PRIMARY KEY,  MoTa NVARCHAR(MAX),  HeSo FLOAT NOT NULL  ); |

***Bảng Chi Tiết Chấm Công***

|  |
| --- |
| CREATE TABLE ctChamCong (  MaNV NVARCHAR(10) NOT NULL,  MaCC NVARCHAR(10) NOT NULL,  MaThang NVARCHAR(6) NOT NULL,  NgayChamCong INT NOT NULL CHECK(NgayChamCong BETWEEN 1 AND 31),  PRIMARY KEY (MaNV, MaThang, NgayChamCong),  CONSTRAINT fK\_ctCC\_MaNV FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV)  ON UPDATE CASCADE  ON DELETE CASCADE,  CONSTRAINT fK\_ctCC\_MaCC FOREIGN KEY (MaCC) REFERENCES ChamCong(MaCC)  ON UPDATE CASCADE  ON DELETE NO ACTION,  CONSTRAINT fK\_ctCC\_MaThang FOREIGN KEY (MaThang) REFERENCES Thang(MaThang)  ON UPDATE CASCADE  ON DELETE NO ACTION  ); |

***Bảng Bảo Hiểm***

|  |
| --- |
| CREATE TABLE BaoHiem (  MaLoai NVARCHAR(10) PRIMARY KEY,  TenBH NVARCHAR(50) NOT NULL  ); |

***Bảng Chi Tiết Bảo Hiểm***

|  |
| --- |
| CREATE TABLE ctBaoHiem (  MaNV NVARCHAR(10) NOT NULL,  MaBH NVARCHAR(10) PRIMARY KEY,  MaLoai NVARCHAR(10) NOT NULL,  NgayBD DATE NOT NULL,  NgayKT DATE NOT NULL,  CONSTRAINT fK\_ctBH\_MaNV FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV)  ON UPDATE CASCADE  ON DELETE CASCADE,  CONSTRAINT fK\_ctBH\_MaLoai FOREIGN KEY (MaLoai) REFERENCES BaoHiem(MaLoai)  ON UPDATE CASCADE  ON DELETE NO ACTION  ); |

***Bảng Nghỉ Phép***

|  |
| --- |
| CREATE TABLE NghiPhep (  MaNV NVARCHAR(10) NOT NULL,  MaThang NVARCHAR(6) NOT NULL,  NgayNghiPhep INT NOT NULL CHECK(NgayNghiPhep BETWEEN 1 AND 31),  GhiChu NVARCHAR(MAX) NOT NULL,  PRIMARY KEY (MaNV, MaThang, NgayNghiPhep),  CONSTRAINT fK\_NP\_MaNV FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV)  ON UPDATE CASCADE  ON DELETE CASCADE,  CONSTRAINT fK\_NP\_MaThang FOREIGN KEY (MaThang) REFERENCES Thang(MaThang)  ON UPDATE CASCADE  ON DELETE NO ACTION  ); |

***Bảng Loại Tài Khoản***

|  |
| --- |
| CREATE TABLE LoaiTaiKhoan (  MaLoai NVARCHAR(10) PRIMARY KEY,  Ten NVARCHAR(50) NOT NULL  ); |

***Bảng Tài Khoản***

|  |
| --- |
| CREATE TABLE TaiKhoan (  TenDangNhap NVARCHAR(10) PRIMARY KEY,  MatKhau NVARCHAR(255) NOT NULL,  MaLoai NVARCHAR(10) NOT NULL,  CONSTRAINT fK\_TK\_TenDangNhap FOREIGN KEY (TenDangNhap) REFERENCES NhanVien(MaNV)  ON UPDATE CASCADE  ON DELETE CASCADE,  CONSTRAINT fK\_TK\_MaLoai FOREIGN KEY (MaLoai) REFERENCES LoaiTaiKhoan(MaLoai)  ON UPDATE CASCADE  ON DELETE NO ACTION  ); |

***Bảng Phụ Cấp***

|  |
| --- |
| CREATE TABLE PhuCap (  MaPhuCap NVARCHAR(10) PRIMARY KEY,  LoaiPhuCap NVARCHAR(10) NOT NULL  ); |

***Bảng Chi Tiết Phụ Cấp***

|  |
| --- |
| CREATE TABLE ctPhuCap (  MaNV NVARCHAR(10) NOT NULL,  MaPhuCap NVARCHAR(10) NOT NULL,  MaThang NVARCHAR(6) NOT NULL,  NgayPhuCap INT NOT NULL CHECK(NgayPhuCap BETWEEN 1 AND 31),  SoTien INT NOT NULL,  PRIMARY KEY (MaNV, MaPhuCap, MaThang, NgayPhuCap),  CONSTRAINT fK\_PC\_MaNV FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV)  ON UPDATE CASCADE  ON DELETE CASCADE,  CONSTRAINT fK\_PC\_MaPhuCap FOREIGN KEY (MaPhuCap) REFERENCES PhuCap(MaPhuCap)  ON UPDATE CASCADE  ON DELETE NO ACTION,  CONSTRAINT fK\_PC\_MaThang FOREIGN KEY (MaThang) REFERENCES Thang(MaThang)  ON UPDATE CASCADE  ON DELETE NO ACTION  ); |

**A screenshot of a computer

Description automatically generated**

## 5. Các triggers

### 5.2. Trigger kiểm tra ngày bắt đầu và ngày kết thúc bảo hiểm

|  |
| --- |
| CREATE OR ALTER TRIGGER tr\_ctBaoHiem\_KiemTraNgay  ON ctBaoHiem  AFTER INSERT, UPDATE  AS  BEGIN  DECLARE @NgayBD date,  @NgayKT date;  SELECT @NgayBD = i.NgayBD, @NgayKT = i.NgayKT  FROM inserted i;  IF @NgayBD >= @NgayKT  BEGIN  RAISERROR('Ngày bắt đầu phải trước ngày kết thúc', 16, 1);  ROLLBACK TRANSACTION;  END  END; |

### 5.1. Trigger tự động chấm công nghỉ phép có lương hoặc không lương khi nhân viên đăng ký nghỉ phép

|  |
| --- |
| CREATE OR ALTER TRIGGER tr\_NghiPhep\_ctChamCong\_CheckNghiPhep  ON NghiPhep  AFTER INSERT  AS  BEGIN  DECLARE @MaNV NVARCHAR(10),  @MaThang NVARCHAR(6),  @NgayNghiPhep INT,  @Nam NVARCHAR(4),  @SoNgayDaNghi INT,  @MaCC NVARCHAR(10);  -- Lấy thông tin từ bản ghi mới được thêm vào  SELECT @MaNV = i.MaNV,  @MaThang = i.MaThang,  @NgayNghiPhep = i.NgayNghiPhep  FROM inserted i;  -- Tách năm từ 4 ký tự cuối trong MaThang  SELECT @Nam = SUBSTRING(@MaThang, 3, 4);  -- Đếm số ngày nghỉ phép trong năm của nhân viên đó  SELECT @SoNgayDaNghi = COUNT(\*)  FROM NghiPhep  WHERE MaNV = @MaNV AND RIGHT(MaThang, 4) = @Nam;  -- Xác định MaCC dựa trên số ngày đã nghỉ phép, 13 là do tính inserted  IF (@SoNgayDaNghi >= 13)  SELECT @MaCC = MaCC FROM ChamCong WHERE MoTa LIKE N'%ghỉ không lương%';  ELSE  SELECT @MaCC = MaCC FROM ChamCong WHERE MoTa LIKE N'%ghỉ phép năm%';  -- Thêm bản ghi vào bảng ctChamCong  INSERT INTO ctChamCong (MaNV, MaCC, MaThang, NgayChamCong)  VALUES (@MaNV, @MaCC, @MaThang, @NgayNghiPhep);  END; |

### 5.3. Trigger kiểm tra mỗi phòng ban chỉ có một trưởng phòng, đảm bảo MaTrP là duy nhất giữa các phòng ban và đồng bộ MaTrP trong bảng PhongBan khi có nhân viên được cập nhật

|  |
| --- |
| CREATE OR ALTER TRIGGER tr\_NhanVien\_CapNhatThongTinTruongPhong  ON NhanVien  AFTER UPDATE  AS  BEGIN  SET NOCOUNT ON;  -- Đảm bảo mỗi phòng ban chỉ có một trưởng phòng  IF EXISTS (  SELECT nv.MaPB  FROM NhanVien nv  WHERE nv.MaCV IN (SELECT cv.MaCV FROM ChucVu cv WHERE cv.TenCV LIKE N'Trưởng Phòng')  GROUP BY nv.MaPB  HAVING COUNT(\*) > 1  )  BEGIN  RAISERROR ('Mỗi phòng ban chỉ được có duy nhất một trưởng phòng.', 16, 1);  ROLLBACK TRANSACTION;  RETURN;  END  -- Đảm bảo các trưởng phòng không giống nhau giữa các phòng ban (unique xử lý có null)  IF EXISTS (  SELECT pb.MaTrP, COUNT(\*)  FROM PhongBan pb  WHERE pb.MaTrP IS NOT NULL  GROUP BY MaTrP  HAVING COUNT(\*)>1  )  BEGIN  RAISERROR ('Trưởng phòng của các phòng ban phải khác nhau', 16, 1);  ROLLBACK TRANSACTION;  RETURN;  END  -- Cập nhật thông tin trưởng phòng cho phòng ban mới  IF EXISTS (  SELECT 1  FROM inserted i  JOIN PhongBan pb ON i.MaPB = pb.MaPB  )  BEGIN  -- Đặt MaTrP thành NULL nếu phòng ban không có nhân viên  UPDATE PhongBan  SET MaTrP = NULL  WHERE MaPB IN (  SELECT pb.MaPB  FROM PhongBan pb  LEFT JOIN NhanVien nv ON pb.MaPB = nv.MaPB  WHERE nv.MaPB IS NULL  );  -- Cập nhật MaTrP nếu phòng ban có nhân viên  UPDATE PhongBan  SET MaTrP = (  SELECT TOP 1 nv.MaNV  FROM NhanVien nv  JOIN ChucVu cv ON nv.MaCV = cv.MaCV  WHERE nv.MaPB = PhongBan.MaPB  AND cv.TenCV LIKE N'Trưởng Phòng'  )  WHERE MaPB IN (SELECT i.MaPB FROM inserted i)  AND EXISTS (  SELECT 1  FROM NhanVien nv  WHERE nv.MaPB = PhongBan.MaPB  );  END  -- Kiểm tra và cập nhật thông tin trưởng phòng nếu nhân viên không còn thuộc phòng ban đó  IF EXISTS (  SELECT 1  FROM deleted d  JOIN PhongBan pb ON d.MaPB = pb.MaPB  )  BEGIN  -- Đặt MaTrP thành NULL nếu phòng ban không có nhân viên  UPDATE PhongBan  SET MaTrP = NULL  WHERE MaPB IN (  SELECT pb.MaPB  FROM PhongBan pb  LEFT JOIN NhanVien nv ON pb.MaPB = nv.MaPB  WHERE nv.MaPB IS NULL  );  -- Cập nhật MaTrP nếu phòng ban có nhân viên  UPDATE PhongBan  SET MaTrP = (  SELECT TOP 1 nv.MaNV  FROM NhanVien nv  JOIN ChucVu cv ON nv.MaCV = cv.MaCV  WHERE nv.MaPB = PhongBan.MaPB  AND cv.TenCV LIKE N'Trưởng Phòng'  )  WHERE MaPB IN (SELECT d.MaPB FROM deleted d)  AND EXISTS (  SELECT 1  FROM NhanVien nv  WHERE nv.MaPB = PhongBan.MaPB  );  END  END; |

### 5.4. Trigger thông báo cho nhân viên nếu nhân viên có thưởng/phạt

CREATE OR ALTER TRIGGER tr\_ctThuongPhat\_ThemThongBaoThuongPhat

ON ctThuongPhat

AFTER INSERT

AS

BEGIN

DECLARE @MaNV NVARCHAR(10);

DECLARE @MaPB NVARCHAR(10);

DECLARE @NgayGui DATETIME;

DECLARE @TieuDe NVARCHAR(100);

DECLARE @NoiDung NVARCHAR(MAX);

DECLARE @Loai NVARCHAR(50);

DECLARE @Ho NVARCHAR(50);

DECLARE @Ten NVARCHAR(50);

-- Lấy thông tin từ bản ghi vừa được thêm

SELECT @MaNV = inserted.MaNV, @Loai = tp.Loai, @Ho = nv.Ho, @Ten = nv.Ten

FROM inserted

JOIN ThuongPhat tp ON inserted.MaThuongPhat = tp.MaThuongPhat

JOIN NhanVien nv ON nv.MaNV = inserted.MaNV

-- Lấy mã phòng ban

SELECT @MaPB = MaPB

FROM NhanVien

WHERE MaNV = @MaNV;

-- Tạo thông báo

SET @NgayGui = GETDATE();

IF @Loai = N'Phạt'

BEGIN

SET @TieuDe = N'Có hình phạt';

SET @NoiDung = @Ho + ' ' + @Ten + N' đã nhận được một hình phạt.';

END

ELSE IF @Loai = N'Thưởng'

BEGIN

SET @TieuDe = N'Có thưởng';

SET @NoiDung = @Ho + ' ' + @Ten + N' đã nhận được thưởng.';

END

-- Gọi thủ tục lưu trữ

EXEC sp\_ThemThongBao @TieuDe, @NoiDung, @MaPB, @NgayGui;

END;

### 5.5. Trigger thông báo cho nhân viên nếu nhận được phụ cấp

CREATE OR ALTER TRIGGER tr\_ctPhuCap\_ThemThongBaoPhuCap

ON ctPhuCap

AFTER INSERT

AS

BEGIN

DECLARE @MaNV NVARCHAR(10),

@MaPhuCap NVARCHAR(10),

@LoaiPhuCap NVARCHAR(50),

@TieuDe NVARCHAR(100),

@NgayPhuCap INT,

@MaPB NVARCHAR(10),

@NgayGui DATE,

@NoiDung NVARCHAR(MAX),

@Ho NVARCHAR(50),

@Ten NVARCHAR(50);

SELECT @MaNV = i.MaNV, @MaPhuCap = i.MaPhuCap, @NgayPhuCap = i.NgayPhuCap, @Ho = nv.Ho, @Ten = nv.Ten

FROM inserted i JOIN NhanVien nv ON i.MaNV = nv.MaNV;

SELECT @LoaiPhuCap = p.LoaiPhuCap

FROM PhuCap p

WHERE p.MaPhuCap = @MaPhuCap;

-- Lấy mã phòng ban

SELECT @MaPB = MaPB

FROM NhanVien

WHERE MaNV = @MaNV;

-- Tạo thông báo

SET @NgayGui = GETDATE();

SET @TieuDe =N'Thông báo phụ cấp';

SET @NoiDung = @Ho + ' ' + @Ten + N' đã nhận được phụ cấp: ' + @LoaiPhuCap;

-- Gọi thủ tục lưu trữ

EXEC sp\_ThemThongBao @TieuDe, @NoiDung, @MaPB, @NgayGui;

END;

### 5.6. Trigger thông báo cho nhân viên nếu nhân viên đăng ký nghỉ phép thành công

CREATE OR ALTER TRIGGER tr\_NghiPhep\_ThongBao

ON NghiPhep

AFTER INSERT

AS

BEGIN

DECLARE @MaNV NVARCHAR(10), @Ho NVARCHAR(50), @Ten NVARCHAR(50), @NgayNghiPhep INT, @MoTa NVARCHAR(MAX), @MaPB NVARCHAR(10);

DECLARE @TieuDe NVARCHAR(100), @NoiDung NVARCHAR(MAX);

-- Lấy thông tin nhân viên từ bảng NhanVien và thông tin ngày nghỉ phép

SELECT

@MaNV = MaNV,

@NgayNghiPhep = NgayNghiPhep

FROM inserted;

-- Lấy tên, phòng ban và các thông tin khác của nhân viên

SELECT

@Ho = Ho,

@Ten = Ten,

@MoTa = MoTa,

@MaPB = MaPB

FROM NhanVien nv

JOIN NghiPhep np ON nv.MaNV = np.MaNV

LEFT JOIN Thang t ON np.MaThang = t.MaThang

WHERE nv.MaNV = @MaNV;

-- Tạo tiêu đề và nội dung thông báo

SET @TieuDe = N'Thông Báo Nghỉ Phép: ' + @Ho + ' ' + @Ten;

SET @NoiDung = @Ho + ' ' + @Ten + N' đã đăng ký nghỉ phép vào ngày ' + CAST(@NgayNghiPhep AS NVARCHAR(10)) + ' ' + @MoTa + '.';

-- Chèn thông báo vào bảng ThongBao

INSERT INTO ThongBao (TieuDe, NoiDung, MaPB, NgayGui)

VALUES (@TieuDe, @NoiDung, @MaPB, GETDATE());

END;

### 5.7. Trigger thông báo cho nhân viên nếu nhân viên chấm công thành công

CREATE OR ALTER TRIGGER tr\_ctChamCong\_ThongBao

ON ctChamCong

AFTER INSERT

AS

BEGIN

SET NOCOUNT ON;

-- Tạo bảng tạm để xử lý tập hợp

DECLARE @ThongBao TABLE (

TieuDe NVARCHAR(100),

NoiDung NVARCHAR(MAX),

MaPB NVARCHAR(10),

NgayGui DATETIME

);

-- Lấy thông tin từ bảng inserted và liên kết với bảng NhanVien

INSERT INTO @ThongBao (TieuDe, NoiDung, MaPB, NgayGui)

SELECT

N'Thông Báo Chấm Công: ' + nv.Ho + ' ' + nv.Ten AS TieuDe,

nv.Ho + ' ' + nv.Ten + N' đã chấm công thành công vào Ngày ' +

CAST(i.NgayChamCong AS NVARCHAR(10)) + ' ' + ISNULL(t.MoTa, '') + '.' AS NoiDung,

nv.MaPB,

GETDATE()

FROM

inserted i

LEFT JOIN NhanVien nv ON i.MaNV = nv.MaNV

LEFT JOIN Thang t ON i.MaThang = t.MaThang;

-- Chèn thông báo vào bảng ThongBao

INSERT INTO ThongBao (TieuDe, NoiDung, MaPB, NgayGui)

SELECT TieuDe, NoiDung, MaPB, NgayGui

FROM @ThongBao;

SET NOCOUNT OFF;

END;

### 5.8. Trigger thông báo cho nhân viên nếu nhân viên đổi chức vụ

CREATE OR ALTER TRIGGER tr\_NhanVien\_DoiChucVu

ON NhanVien

AFTER UPDATE

AS

BEGIN

DECLARE @MaNV NVARCHAR(10), @Ho NVARCHAR(50), @Ten NVARCHAR(50), @MaCVOld NVARCHAR(10), @MaCVNew NVARCHAR(10), @TieuDe NVARCHAR(100), @NoiDung NVARCHAR(MAX), @MaPB NVARCHAR(10);

-- Lấy thông tin từ bảng inserted (cập nhật mới)

SELECT

@MaNV = MaNV,

@MaCVOld = (SELECT MaCV FROM deleted WHERE MaNV = inserted.MaNV), -- Lấy MaCV cũ từ bảng deleted

@MaCVNew = MaCV

FROM inserted

WHERE MaNV IN (SELECT MaNV FROM inserted);

-- Kiểm tra nếu MaCV có thay đổi (thăng chức)

IF @MaCVOld <> @MaCVNew

BEGIN

-- Lấy thông tin nhân viên như Họ, Tên, Phòng ban

SELECT

@Ho = Ho,

@Ten = Ten,

@MaPB = MaPB

FROM NhanVien

WHERE MaNV = @MaNV;

-- Tạo tiêu đề và nội dung thông báo

SET @TieuDe = N'Thông Báo Chuyển Chức Vụ: ' + @Ho + ' ' + @Ten;

SET @NoiDung = @Ho + ' ' + @Ten + N' đã được chuyển từ chức vụ ' +

(SELECT TenCV FROM ChucVu WHERE MaCV = @MaCVOld) + N' sang ' +

(SELECT TenCV FROM ChucVu WHERE MaCV = @MaCVNew) + '.';

-- Chèn thông báo vào bảng ThongBao

INSERT INTO ThongBao (TieuDe, NoiDung, MaPB, NgayGui)

VALUES (@TieuDe, @NoiDung, @MaPB, GETDATE());

END

END;

CREATE OR ALTER TRIGGER tr\_UpdateThongBaoId

ON ThongBao

AFTER INSERT, DELETE

AS

BEGIN

SET NOCOUNT ON;

-- Tạo bảng tạm để lưu trữ các thông báo còn lại

DECLARE @ThongBaoTemp TABLE (

NewId INT IDENTITY(1,1),

TieuDe NVARCHAR(100),

NoiDung NVARCHAR(MAX),

MaPB NVARCHAR(10),

NgayGui DATETIME

);

-- Chèn các thông báo còn lại vào bảng tạm

INSERT INTO @ThongBaoTemp (TieuDe, NoiDung, MaPB, NgayGui)

SELECT TieuDe, NoiDung, MaPB, NgayGui

FROM ThongBao

ORDER BY Id;

-- Xóa tất cả các thông báo trong bảng ThongBao

DELETE FROM ThongBao;

-- Bật IDENTITY\_INSERT để chèn giá trị cụ thể vào cột IDENTITY

SET IDENTITY\_INSERT ThongBao ON;

-- Chèn lại các thông báo từ bảng tạm vào bảng ThongBao với Id mới

INSERT INTO ThongBao (Id, TieuDe, NoiDung, MaPB, NgayGui)

SELECT NewId, TieuDe, NoiDung, MaPB, NgayGui

FROM @ThongBaoTemp;

-- Tắt IDENTITY\_INSERT sau khi chèn xong

SET IDENTITY\_INSERT ThongBao OFF;

END;

### 5.9. Trigger kiểm tra đăng ký nghỉ phép nếu trùng với ngày nghỉ chính thức của công ty thì thông báo cho nhân viên

|  |
| --- |
| CREATE OR ALTER TRIGGER tr\_NghiPhep\_CheckNgayLe  ON NghiPhep  INSTEAD OF INSERT  AS  BEGIN  DECLARE @MaNV NVARCHAR(10), @MaThang NVARCHAR(6), @NgayNghiPhep INT, @NgayPhepFull DATE;  SELECT @MaNV = i.MaNV, @MaThang = i.MaThang, @NgayNghiPhep = i.NgayNghiPhep  FROM inserted i;  -- Chuyển đổi ngày và tháng thành kiểu DATE để kiểm tra với ngày nghỉ lễ  SET @NgayPhepFull = DATEFROMPARTS(CONVERT(INT, RIGHT(@MaThang, 4)), CONVERT(INT, LEFT(@MaThang, 2)), @NgayNghiPhep);  -- Danh sách các ngày lễ cố định  IF @NgayPhepFull IN (  '2024-01-01', -- Tết Dương lịch  '2024-04-30', -- Ngày Giải phóng miền Nam  '2024-05-01', -- Ngày Quốc tế Lao động  '2024-09-02', -- Quốc khánh Việt Nam  '2024-12-31' -- Ngày cuối năm  -- Thêm các ngày lễ khác vào đây nếu cần  )  BEGIN  RAISERROR('Ngày này là ngày lễ chính thức của công ty, không cần đăng ký nghỉ phép.', 16, 1);  RETURN;  END  ELSE  BEGIN  -- Chèn dữ liệu nếu ngày không trùng với ngày lễ  INSERT INTO NghiPhep (MaNV, MaThang, NgayNghiPhep, GhiChu)  SELECT MaNV, MaThang, NgayNghiPhep, GhiChu  FROM inserted;  END  END; |

### 5.10. Trigger tự động tạo tháng và tính số ngày công chuẩn nếu tháng nhân viên thực hiện chấm công chưa tồn tại

|  |
| --- |
| CREATE OR ALTER TRIGGER [dbo].[tr\_AddctChamCong]  ON [dbo].[ctChamCong]  INSTEAD OF INSERT  AS  BEGIN  DECLARE @MaNV VARCHAR(10),  @MaCC VARCHAR(10),  @MaThang VARCHAR(6),  @NgayChamCong INT,  @SoNgayCongChuan INT = 0;  BEGIN TRANSACTION;  BEGIN TRY  SELECT @MaNV = inserted.MaNV,  @MaCC = inserted.MaCC,  @MaThang = inserted.MaThang,  @NgayChamCong = inserted.NgayChamCong  FROM inserted;  -- Kiểm tra Thang Cham Cong đã tồn tại trong bảng tháng chưa  IF NOT EXISTS (SELECT 1 FROM Thang WHERE MaThang = @MaThang)  BEGIN  DECLARE @Thang NVARCHAR(2) = LEFT(@MaThang, 2);  DECLARE @Nam NVARCHAR(4) = RIGHT(@MaThang, 4);  SET @SoNgayCongChuan = dbo.ft\_SoNgayCongChuan(@MaThang);  INSERT INTO Thang (MaThang, MoTa, SoNgayCongChuan)  VALUES (@MaThang, N'Tháng ' + @Thang + N' năm ' + @Nam, @SoNgayCongChuan);  END  DECLARE @NgayHienTai DATE = GETDATE(), @NgayCCFull DATE  SET @NgayCCFull = CONVERT(DATE, RIGHT(@MaThang, 4) + LEFT(@MaThang, 2) + RIGHT('00' + CAST(@NgayChamCong AS VARCHAR), 2), 112)  IF @NgayCCFull > @NgayHienTai and ( @MaCC = 'CC01' or @MaCC = 'CC03')  BEGIN  RAISERROR('Không thể chấm công đi làm cho ngày ở tương lai', 16, 1);  ROLLBACK TRANSACTION;  RETURN;  END  ELSE  BEGIN  -- Kiểm tra và Thêm record vào bảng ctChamCong  IF NOT EXISTS (SELECT 1 FROM ctChamCong WHERE MaNV = @MaNV  AND MaThang = @MaThang  AND NgayChamCong = @NgayChamCong)  BEGIN  INSERT INTO ctChamCong (MaNV, MaCC, MaThang, NgayChamCong)  VALUES (@MaNV, @MaCC, @MaThang, @NgayChamCong);  END  ELSE  BEGIN  RAISERROR('Đã chấm công cho ngày này', 16, 1);  ROLLBACK TRANSACTION;  RETURN;  END  COMMIT TRANSACTION;  END  END TRY  BEGIN CATCH  DECLARE @ErrorMessage NVARCHAR(4000);  SET @ErrorMessage = ERROR\_MESSAGE();  PRINT 'Error Message: ' + @ErrorMessage;  ROLLBACK TRANSACTION;  END CATCH;  END;  GO |

**Hàm tính số ngày công chuẩn trong trigger tr\_AddctChamCong (số ngày làm việc không tính thứ bảy, chủ nhật):**

|  |
| --- |
| CREATE FUNCTION dbo.ft\_SoNgayCongChuan (@MaThang VARCHAR(6))  RETURNS INT  AS  BEGIN  DECLARE @SoNgayLamViec INT = 0;  DECLARE @NgayBatDau DATE;  DECLARE @NgayKetThuc DATE;  DECLARE @NgayHienTai DATE;  DECLARE @Thang INT;  DECLARE @Nam INT;  -- Lấy tháng và năm từ MaThang  SET @Thang = CAST(SUBSTRING(@MaThang, 1, 2) AS INT);  SET @Nam = CAST(SUBSTRING(@MaThang, 3, 4) AS INT);  -- Xác định ngày bắt đầu và ngày kết thúc của tháng  SET @NgayBatDau = DATEFROMPARTS(@Nam, @Thang, 1);  SET @NgayKetThuc = EOMONTH(@NgayBatDau);  -- Duyệt qua từng ngày trong tháng  SET @NgayHienTai = @NgayBatDau;  WHILE @NgayHienTai <= @NgayKetThuc  BEGIN  -- Nếu là thứ Hai đến thứ Sáu tăng biến đếm số ngày làm việc  IF DATEPART(WEEKDAY, @NgayHienTai) BETWEEN 2 AND 6  BEGIN  SET @SoNgayLamViec = @SoNgayLamViec + 1;  END  -- Chuyển sang ngày tiếp theo  SET @NgayHienTai = DATEADD(DAY, 1, @NgayHienTai);  END  RETURN @SoNgayLamViec;  END; |

## 6. Các views

### 6.1. Hiển thị thông tin nhân viên

|  |
| --- |
| CREATE OR ALTER VIEW vw\_ThongTinNhanVien AS  SELECT nv.MaNV, nv.Ho, nv.Ten, nv.GioiTinh, nv.NgaySinh, nv.DiaChi, nv.SDT, nv.Email, nv.CCCD, pb.TenPB AS TenPhongBan, cv.TenCV AS TenChucVu  FROM NhanVien nv JOIN PhongBan pb ON nv.MaPB = pb.MaPB JOIN ChucVu cv ON nv.MaCV = cv.MaCV;  GO |

### 6.2. Hiển thị thông tin các loại bảo hiểm

|  |
| --- |
| CREATE OR ALTER VIEW vw\_QuanLyBaoHiem AS  SELECT nv.MaNV, nv.Ho, nv.Ten, bh.TenBH, ctbh.MaBH, ctbh.NgayBD, ctbh.NgayKT  FROM NhanVien nv JOIN ctBaoHiem ctbh ON nv.MaNV = ctbh.MaNV JOIN BaoHiem bh ON ctbh.MaLoai = bh.MaLoai; |

### 6.3. Hiển thị thông tin hợp đồng

|  |
| --- |
| CREATE OR ALTER VIEW vw\_ThongTinHopDong AS  SELECT nv.MaNV,hd.MaHD, hd.LuongCoBan, hd.NgayBD AS NgayBatDauHopDong, hd.NgayKT AS NgayKetThucHopDong, pb.TenPB AS TenPhongBan, cv.TenCV AS TenChucVu  FROM NhanVien nv  JOIN HopDong hd ON nv.MaHD = hd.MaHD  JOIN ChucVu cv ON nv.MaCV = cv.MaCV  JOIN PhongBan pb ON pb.MaPB = nv.MaPB; |

# **CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG**

## 1. Kết nối cơ sở dữ liệu

|  |
| --- |
| using System;  using System.Data;  using System.Data.SqlClient;  using System.Windows.Forms;  namespace DoAnNhom21  {  internal class Connection  {  public static string user;  public static string pass;  public static string connectString = Properties.Settings.Default.ConnectionString;  public static string GetConnectionString(string user, string pass)  {  return connectString + ";User Id=" + user + ";Password=" + pass;  }  public static SqlConnection getConnection()  {  return new SqlConnection(connectString);  }  public static DataTable LoadDataTable(string query)  {  connectString = GetConnectionString(user, pass);  SqlConnection conn = new SqlConnection(connectString);  DataTable dataTable = new DataTable();  try  {  conn.Open();  SqlDataAdapter adapter = new SqlDataAdapter(query, conn);  adapter.Fill(dataTable);  }  catch (Exception exc)  {  MessageBox.Show(exc.Message);  }  finally  {  conn.Close();  }  return dataTable;  }  public static DataTable LoadDataTable(SqlCommand command)  {  connectString = GetConnectionString(user, pass);  SqlConnection conn = new SqlConnection(connectString);  DataTable dataTable = new DataTable();  try  {  command.Connection = conn;  conn.Open();  SqlDataAdapter adapter = new SqlDataAdapter(command);  adapter.Fill(dataTable);  }  catch (Exception exc)  {  MessageBox.Show(exc.Message);  }  finally  {  conn.Close();  }  return dataTable;  }  public static void ExecuteCommand(SqlCommand command)  {  connectString = GetConnectionString(user, pass);  using (SqlConnection connection = new SqlConnection(connectString))  {  command.Connection = connection;  connection.Open();  command.ExecuteNonQuery();  connection.Close();  }  }  public static void ExecuteCommandAdmin(SqlCommand command)  {  using (SqlConnection connection = new SqlConnection(connectString))  {  command.Connection = connection;  connection.Open();  command.ExecuteNonQuery();  connection.Close();  }  }  }  } |

## 2. Xem thông tin

### 2.1. Xem thông tin hợp đồng

private void FormHopDongNhanVien\_Load(object sender, EventArgs e)

{

try

{

using (conn = Connection.getConnection())

{

string query = "select \* from vw\_ThongTinHopDong HD where HD.MaNV = @MaNV";

conn.Open();

SqlCommand cmd = new SqlCommand(query, conn);

cmd.Parameters.AddWithValue("@MaNV", this.maNV);

SqlDataReader reader = cmd.ExecuteReader();

if (reader.Read())

{

txtMaNV.Text = reader["MaNV"].ToString();

txtLuongCoBan.Text = reader["LuongCoBan"].ToString();

dateTimePickerNgayBD.Text = reader["NgayBatDauHopDong"].ToString();

dateTimePickerNgayKT.Text = reader["NgayKetThucHopDong"].ToString();

txtChucVu.Text = reader["TenChucVu"].ToString();

txtPhongBan.Text = reader["TenPhongBan"].ToString();

txtMaHD.Text = reader["MaHD"].ToString();

}

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

### 2.2. Xem thông tin nhân viên

private void FormThongTinNhanVien\_Load(object sender, EventArgs e)

{

conn = Connection.getConnection();

conn.Open();

string data = "Select \* from vw\_ThongTinNhanVien where MaNV = '" + txtMaNV.Text + "'";

SqlCommand cmd = new SqlCommand(data, conn);

SqlDataReader reader = cmd.ExecuteReader();

if (reader.Read())

{

txtMaNV.Text = reader["MaNV"].ToString();

txtEmail.Text = reader["Email"].ToString();

txtChucVu.Text = reader["TenChucVu"].ToString();

txtDiaChi.Text = reader["DiaChi"].ToString();

txtGioiTinh.Text = reader["GioiTinh"].ToString();

//dateTimePickerNgaySinh.Text = reader["NamSinh"].ToString();

txtHo.Text = reader["Ho"].ToString();

txtTen.Text = reader["Ten"].ToString();

txtPhongBan.Text = reader["TenPhongBan"].ToString();

txtCCCD.Text = reader["CCCD"].ToString();

txtSDT.Text = reader["SDT"].ToString();

}

reader.Close();

conn.Close();

}

### 2.3. Xem lương thực lãnh

|  |
| --- |
| private void btnTinhLuong\_Click(object sender, EventArgs e)  {  SqlCommand cmd = new SqlCommand("sp\_TinhLuongTheoThang");  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaNV", SessionInfo.MaNV);  cmd.Parameters.AddWithValue("@MaThang", this.guna2DateTimePicker1.Value.ToString("MMyyyy"));  DataTable dt = Connection.LoadDataTable(cmd);  DataRow reader = dt.Rows[0];  txtMaNV.Text = reader["MaNV"].ToString();  txtLuongCoBan.Text = reader["LuongCoBan"].ToString();  txtBHTN.Text = reader["BH03"].ToString();  txtBHXH.Text = reader["BH02"].ToString();  txtBHYT.Text = reader["BH01"].ToString();  txtPhuCap.Text = reader["TongPhuCap"].ToString();  txtGiamTruGiaCanh.Text = reader["GiamTruGiaCanh"].ToString();  txtLuongThucTe.Text = reader["LuongChiuThue"].ToString();  txtThucLanh.Text = reader["LuongThucLanh"].ToString();  txtSoNgayCongChuan.Text = reader["SoNgayCongChuan"].ToString();  txtSoNgayLamViec.Text = reader["SoNgayCong"].ToString();  txtTongBaoHiem.Text = reader["TongTienBaoHiem"].ToString();  txtThue.Text = reader["Thue"].ToString();  txtThuongPhat.Text = reader["TongThuongPhat"].ToString();  } |

### 2.4. Multi - Statement Table -Valued Function (có tham số)

#### 2.4.1. Nhân viên xem thông báo theo phòng ban

CREATE OR ALTER FUNCTION dbo.ft\_NhanVienNhanThongBao(@MaNV nvarchar(10))

RETURNS @ThongBaoNhanVien TABLE (

TieuDe nvarchar(100),

NoiDung nvarchar(MAX),

NgayGui DATE

)

AS

BEGIN

INSERT INTO @ThongBaoNhanVien (TieuDe, NoiDung, NgayGui)

SELECT tb.TieuDe, tb.NoiDung, tb.NgayGui

FROM

PhongBan pb

LEFT JOIN

ThongBao tb ON tb.MaPB = pb.MaPB

JOIN

NhanVien nv ON pb.MaPB = nv.MaPB

WHERE

nv.MaNV = @MaNV

RETURN;

END;

Thực thi trên C#

private void btnXemThongBao\_Click(object sender, EventArgs e)

{

try

{

using (SqlCommand command = new SqlCommand("SELECT \* FROM dbo.ft\_NhanVienNhanThongBao(@MaNV)"))

{

command.CommandType = CommandType.Text;

command.Parameters.AddWithValue("@MaNV", this.maNv);

Connection.ExecuteCommand(command);

guna2DataGridViewThongBao.Visible = true;

guna2DataGridViewThongBao.DataSource = Connection.LoadDataTable(command);

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

#### 2.4.2. Trưởng phòng xem thông báo của tất cả nhân viên

CREATE OR ALTER FUNCTION dbo.ft\_TruongPhongNhanThongBao(@MaNV NVARCHAR(10))

RETURNS @ThongBao TABLE (

TieuDe NVARCHAR(100),

NoiDung NVARCHAR(MAX),

NgayGui DATETIME

)

AS

BEGIN

-- Kiểm tra xem người dùng có phải là trưởng phòng không

IF EXISTS (

SELECT 1

FROM NhanVien nv

JOIN PhongBan pb ON nv.MaNV = pb.MaTrP

WHERE nv.MaNV = @MaNV

)

BEGIN

-- Nếu là trưởng phòng, lấy thông báo của tất cả phòng ban

INSERT INTO @ThongBao (TieuDe, NoiDung, NgayGui)

SELECT DISTINCT

tb.TieuDe,

tb.NoiDung,

tb.NgayGui

FROM ThongBao tb

ORDER BY tb.NgayGui DESC;

END

RETURN;

END;

## 3. Quản lý nhân viên

### 3.1. Thủ tục có tham số

#### 3.1.1.Thêm nhân viên

|  |
| --- |
| CREATE OR ALTER PROCEDURE sp\_AddNhanVien  @MaNV NVARCHAR(10),  @Ho NVARCHAR(50),  @Ten NVARCHAR(50),  @GioiTinh NVARCHAR(10),  @NgaySinh DATE,  @DiaChi NVARCHAR(100),  @SDT NVARCHAR(20),  @Email NVARCHAR(50),  @CCCD NVARCHAR(12),  @MaPB NVARCHAR(10) = NULL,  @MaCV NVARCHAR(10),  @MaHD NVARCHAR(10) = NULL  AS  BEGIN  -- Kiểm tra xem mã nhân viên đã tồn tại hay chưa  IF EXISTS (SELECT 1 FROM NhanVien WHERE MaNV = @MaNV)  BEGIN  THROW 50000, 'Đã tồn tại mã nhân viên', 1;  RETURN;  END  -- Thực hiện việc chèn dữ liệu  INSERT INTO NhanVien(MaNV, Ho, Ten, GioiTinh, NgaySinh, DiaChi, SDT, Email, CCCD, MaPB, MaCV, MaHD)  VALUES (@MaNV, @Ho, @Ten, @GioiTinh, @NgaySinh, @DiaChi, @SDT, @Email, @CCCD, @MaPB, @MaCV, @MaHD);  END; |

Thực hiện trên C#

|  |
| --- |
| private void btnThem\_Click(object sender, EventArgs e)  {  try  {  using (SqlConnection conn = Connection.getConnection())  {  conn.Open();  SqlCommand cmd = new SqlCommand("sp\_AddNhanVien");  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaNV", this.txtMaNhanVien.Text);  cmd.Parameters.AddWithValue("@Ho", this.txtHoVaTenLot.Text);  cmd.Parameters.AddWithValue("@Ten", this.txtTen.Text);  cmd.Parameters.AddWithValue("@GioiTinh", this.cbbGioiTinh.Text);  cmd.Parameters.AddWithValue("@NgaySinh", this.dateTimePickerNgaySinh.Value);  cmd.Parameters.AddWithValue("@DiaChi", this.txtDiaChi.Text);  cmd.Parameters.AddWithValue("@CCCD", this.txtCCCD.Text);  cmd.Parameters.AddWithValue("@Email", this.txtEmail.Text);  cmd.Parameters.AddWithValue("@SDT", this.txtSDT.Text);  cmd.Parameters.AddWithValue("@MaPB", this.cbbPhongBan.SelectedValue.ToString());  cmd.Parameters.AddWithValue("@MaCV", this.cbbChucVu.SelectedValue.ToString());  cmd.Connection = conn;  cmd.ExecuteNonQuery();  MessageBox.Show("Thêm thành công");  load();  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

#### 3.1.2. Cập nhật thông tin nhân viên

|  |
| --- |
| CREATE OR ALTER PROCEDURE sp\_UpdateNhanVien  @MaNV NVARCHAR(10),  @Ho NVARCHAR(50),  @Ten NVARCHAR(50),  @GioiTinh NVARCHAR(10),  @NgaySinh DATE,  @DiaChi NVARCHAR(100),  @CCCD NVARCHAR(12),  @Email NVARCHAR(50),  @SDT NVARCHAR(10),  @MaPB NVARCHAR(10),  @MaCV NVARCHAR(10)  AS  BEGIN  -- Kiểm tra dữ liệu không được bỏ trống  IF  @Ho IS NULL OR  @Ten IS NULL OR  @GioiTinh IS NULL OR  @NgaySinh IS NULL OR  @DiaChi IS NULL OR  @CCCD IS NULL OR  @Email IS NULL OR  @SDT IS NULL OR  @MaCV IS NULL  BEGIN  -- Trả về một lỗi hoặc thông báo tùy thuộc vào yêu cầu  RAISERROR('Vui lòng điền đầy đủ thông tin cho các trường bắt buộc', 16, 1);  RETURN;  END  -- Kiểm tra độ dài của SDT  IF LEN(@SDT) <> 10  BEGIN  RAISERROR('Số điện thoại phải có độ dài 10 ký tự', 16, 1);  RETURN;  END  -- Kiểm tra tính duy nhất của CCCD, Email và SDT  IF EXISTS (SELECT 1 FROM NhanVien WHERE CCCD = @CCCD AND MaNV <> @MaNV)  BEGIN  RAISERROR('CCCD đã tồn tại', 16, 1);  RETURN;  END  IF EXISTS (SELECT 1 FROM NhanVien WHERE Email = @Email AND MaNV <> @MaNV)  BEGIN  RAISERROR('Email đã tồn tại', 16, 1);  RETURN;  END  IF EXISTS (SELECT 1 FROM NhanVien WHERE SDT = @SDT AND MaNV <> @MaNV)  BEGIN  RAISERROR('Số điện thoại đã tồn tại', 16, 1);  RETURN;  END  -- Thực hiện cập nhật thông tin nhân viên trong bảng NhanVien dựa trên MaNV  UPDATE NhanVien  SET  Ho = @Ho,  Ten = @Ten,  GioiTinh = @GioiTinh,  NgaySinh = @NgaySinh,  DiaChi = @DiaChi,  CCCD = @CCCD,  Email = @Email,  SDT = @SDT,  MaPB = @MaPB,  MaCV = @MaCV  WHERE MaNV = @MaNV;  END; |

Thực hiện trên C#

|  |
| --- |
| private void btnSua\_Click\_1(object sender, EventArgs e)  {  try  {  using (SqlCommand command = new SqlCommand("sp\_UpdateNhanVien"))  {  command.CommandType = CommandType.StoredProcedure;  command.Parameters.AddWithValue("@MaNV", this.txtMaNhanVien.Text);  command.Parameters.AddWithValue("@Ho", this.txtHoVaTenLot.Text);  command.Parameters.AddWithValue("@Ten", this.txtTen.Text);  command.Parameters.AddWithValue("@GioiTinh", this.cbbGioiTinh.Text);  command.Parameters.AddWithValue("@NgaySinh", this.dateTimePickerNgaySinh.Value);  command.Parameters.AddWithValue("@DiaChi", this.txtDiaChi.Text);  command.Parameters.AddWithValue("@CCCD", this.txtCCCD.Text);  command.Parameters.AddWithValue("@Email", this.txtEmail.Text);  command.Parameters.AddWithValue("@SDT", this.txtSDT.Text);  command.Parameters.AddWithValue("@MaPB", this.cbbPhongBan.SelectedValue.ToString());  command.Parameters.AddWithValue("@MaCV", this.cbbChucVu.SelectedValue.ToString());  Connection.ExecuteCommand(command);  MessageBox.Show("Cập nhật thành công");  load();  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

#### 3.1.3. Xóa nhân viên

|  |
| --- |
| CREATE OR ALTER PROCEDURE sp\_DeleteNhanVien  @MaNV nvarchar(10)  AS  BEGIN  SET NOCOUNT ON;  DECLARE @TenTaiKhoan varchar(10);  SELECT @TenTaiKhoan=TenDangNhap FROM TaiKhoan WHERE TenDangNhap=@MaNV  DECLARE @sql varchar(100)  DECLARE @SessionID INT;  SELECT @SessionID = session\_id  FROM sys.dm\_exec\_sessions  WHERE login\_name = @TenTaiKhoan;  IF @SessionID IS NOT NULL  BEGIN  SET @sql = 'kill ' + Convert(NVARCHAR(20), @SessionID)  exec(@sql)  END  BEGIN TRANSACTION;  BEGIN TRY  DELETE FROM NhanVien WHERE MaNV = @MaNV;  --  SET @sql = 'DROP USER '+ @TenTaiKhoan  exec (@sql)  --  SET @sql = 'DROP LOGIN '+ @TenTaiKhoan  exec (@sql)  END TRY  BEGIN CATCH  DECLARE @err NVARCHAR(MAX)  SELECT @err = N'Lỗi ' + ERROR\_MESSAGE()  RAISERROR(@err, 16, 1)  ROLLBACK TRANSACTION;  THROW;  END CATCH  COMMIT TRANSACTION;  END;  GO |

Thực hiện trên C#

|  |
| --- |
| private void btnXoa\_Click\_1(object sender, EventArgs e)  {  try  {  using (SqlCommand command = new SqlCommand("sp\_DeleteNhanVien"))  {  command.CommandType = CommandType.StoredProcedure;  command.Parameters.AddWithValue("@MaNV", this.txtMaNhanVien.Text);  Connection.ExecuteCommand(command);  MessageBox.Show("Xoá thành công");  load();  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

#### 3.1.4. Cập nhật mật khẩu nhân viên

CREATE OR ALTER PROCEDURE [dbo].[sp\_UpdateTaiKhoan]

@TenDangNhap NVARCHAR(10),

@MatKhau NVARCHAR(255),

@NhapLaiMatKhau NVARCHAR(255)

AS

BEGIN

-- Kiểm tra nếu @MatKhau hoặc @NhapLaiMatKhau bị trống

IF @MatKhau IS NULL OR @MatKhau = '' OR @NhapLaiMatKhau IS NULL OR @NhapLaiMatKhau = ''

BEGIN

RAISERROR('Dữ liệu không được bỏ trống', 16, 1)

RETURN;

END

-- Kiểm tra nếu @MatKhau và @NhapLaiMatKhau không khớp

IF @MatKhau <> @NhapLaiMatKhau

BEGIN

RAISERROR('Mật khẩu và Nhập lại mật khẩu không khớp', 16, 1)

RETURN;

END

-- Cập nhật thông tin trong bảng TaiKhoan nếu dữ liệu hợp lệ

UPDATE TaiKhoan

SET

MatKhau = @MatKhau

WHERE TenDangNhap = @TenDangNhap;

PRINT 'Cập nhật mật khẩu thành công';

END;

Thực hiện trên C#

private void btnCapNhat\_Click(object sender, EventArgs e)

{

try

{

using (SqlCommand command = new SqlCommand("sp\_UpdateTaiKhoan"))

{

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@TenDangNhap", this.cbbMaNV.SelectedValue.ToString());

command.Parameters.AddWithValue("@MatKhau", this.txtMatKhauMoi.Text);

command.Parameters.AddWithValue("@NhapLaiMatKhau ", this.txtNhapLaiMatkhau.Text);

Connection.ExecuteCommand(command);

Connection.pass = txtMatKhauMoi.Text;

MessageBox.Show("Cập Nhật Mật Khẩu Thành Công");

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

### 3.2. Inline Table – Valued Function (có tham số): Tìm kiếm nhân viên theo MaNV

CREATE OR ALTER FUNCTION [dbo].[TimKiemNhanVien]

(

@MaNV nvarchar(10)

)

RETURNS TABLE

AS

RETURN

(

SELECT \*

FROM NhanVien

WHERE MaNV = @MaNV

);

Thực thi trên C#

private void btnTimKiem\_Click\_2(object sender, EventArgs e)

{

try

{

using (SqlConnection connection = new SqlConnection(Connection.connectString))

{

string TimKiemNV = "SELECT \* FROM TimKiemNhanVien(@MaNV)";

using (SqlCommand command = new SqlCommand(TimKiemNV, connection))

{

command.Parameters.AddWithValue("@MaNV", txtTimKiemMaNV.Text);

connection.Open();

using (SqlDataReader reader = command.ExecuteReader())

{

if (reader.Read())

{

txtMaNhanVien.Text = reader["MaNV"].ToString();

txtHoVaTenLot.Text = reader["Ho"].ToString();

txtTen.Text = reader["Ten"].ToString();

cbbGioiTinh.Text = reader["GioiTinh"].ToString();

dateTimePickerNgaySinh.Value = Convert.ToDateTime(reader["NgaySinh"]);

txtDiaChi.Text = reader["DiaChi"].ToString();

txtCCCD.Text = reader["CCCD"].ToString();

txtEmail.Text = reader["Email"].ToString();

txtSDT.Text = reader["SDT"].ToString();

cbbPhongBan.Text = reader["MaPB"].ToString();

cbbChucVu.Text = reader["MaCV"].ToString();

txtHopDong.Text = reader["MaHD"].ToString();

tempMaNV = txtMaNhanVien.Text;

}

else

{

MessageBox.Show("Không tìm thấy mã nhân viên này");

return;

}

}

using (SqlDataAdapter adapter = new SqlDataAdapter(command))

{

DataTable dt = new DataTable();

adapter.Fill(dt);

this.dataGridViewNhanVien.DataSource = dt;

}

}

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

## 4. Quản lý phòng ban

### 4.1. Thủ tục có tham số

#### 4.1.1. Thêm phòng ban

|  |
| --- |
| CREATE OR ALTER PROCEDURE sp\_ThemPhongBan  @MaPB nvarchar(10),  @TenPB nvarchar(50),  @SDT nvarchar(20),  @MaTrP nvarchar(10)  AS  BEGIN  INSERT INTO PhongBan (MaPB, TenPB, MaTrP, SDT)  VALUES (@MaPB, @TenPB, @MaTrP, @SDT);  END |

Thực hiện trên C#

private void btnThem\_Click(object sender, EventArgs e)

{

try

{

using (SqlCommand command = new SqlCommand("ThemPhongBan"))

{

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@MaPB", this.txtMaPB.Text);

command.Parameters.AddWithValue("@TenPB", this.txtTenPB.Text);

command.Parameters.AddWithValue("@SDT", this.txtSDT.Text);

command.Parameters.AddWithValue("@MaTrP", this.cbbMaTrP.Text);

Connection.ExecuteCommand(command);

MessageBox.Show("Thêm thành công!");

load();

}

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi:" + ex.Message, "Thông báo lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

#### 4.1.2. Cập nhật phòng ban

|  |
| --- |
| CREATE OR ALTER PROCEDURE sp\_CapNhatPhongBan  @MaPB nvarchar(10),  @TenPB nvarchar(50),  @SDT nvarchar(20),  @MaTrP nvarchar(10)  AS  BEGIN  IF @MaPB IS NULL OR @MaPB = '' OR @TenPB IS NULL OR @TenPB = '' OR @MaTrP IS NULL OR @MaTrP ='' or  @SDT IS NULL OR @SDT = ''  BEGIN  RAISERROR('Vui lòng điền đầy đủ thông tin', 16, 1)  RETURN;  END  ELSE  UPDATE PhongBan  SET TenPB = @TenPB, MaTrP = @MaTrP, SDT = @SDT  WHERE MaPB = @MaPB;  END |

Thực hiện trên C#

private void btnSua\_Click(object sender, EventArgs e)

{

try

{

using (SqlCommand command = new SqlCommand("CapNhatPhongBan"))

{

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@MaPB", this.txtMaPB.Text);

command.Parameters.AddWithValue("@TenPB", this.txtTenPB.Text);

command.Parameters.AddWithValue("@SDT", this.txtSDT.Text);

command.Parameters.AddWithValue("@MaTrP", this.cbbMaTrP.Text);

Connection.ExecuteCommand(command);

MessageBox.Show("Sửa thành công!");

}

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi:" + ex.Message, "Thông báo lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

#### 4.1.3. Xóa phòng ban

CREATE OR ALTER PROCEDURE sp\_XoaPhongBan

@MaPB nvarchar(10)

AS

BEGIN

DELETE FROM PhongBan

WHERE MaPB = @MaPB;

END

Thực hiện trên C#

private void btnXoa\_Click(object sender, EventArgs e)

{

try

{

using (SqlCommand command = new SqlCommand("sp\_XoaPhongBan"))

{

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@MaPB", this.txtMaPB.Text);

Connection.ExecuteCommand(command);

MessageBox.Show("Xoá thành công");

load();

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

#### 4.1.4. Gửi thông báo đến nhân viên của phòng ban

CREATE OR ALTER PROCEDURE sp\_ThemThongBao

@TieuDe NVARCHAR(100),

@NoiDung NVARCHAR(MAX),

@MaPB nvarchar(10),

@NgayGui DATETIME

AS

BEGIN

INSERT INTO ThongBao (TieuDe, NoiDung, MaPB, NgayGui)

VALUES (@TieuDe, @NoiDung, @MaPB, @NgayGui)

WHERE @MaPB = MaPB;

END;

Thực hiện trên C#

private void btnThem\_Click(object sender, EventArgs e)

{

try

{

SqlCommand cmd = new SqlCommand("sp\_ThemThongBao");

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@TieuDe", txtTieuDe.Text);

cmd.Parameters.AddWithValue("@NoiDung", txtNoiDung.Text);

cmd.Parameters.AddWithValue("@MaPB", cbbMaPB.Text);

cmd.Parameters.AddWithValue("@NgayGui", dtpNgayGui.Value);

Connection.ExecuteCommand(cmd);

load();

MessageBox.Show("Thêm thành công!");

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi:" + ex.Message, "Thông báo lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

#### 4.1.5. Cập nhật thông báo

CREATE OR ALTER PROCEDURE sp\_CapNhatThongBao

@Id INT,

@TieuDe NVARCHAR(100),

@NoiDung NVARCHAR(MAX),

@MaPB nvarchar(10),

@NgayGui DATETIME

AS

BEGIN

UPDATE ThongBao

SET TieuDe = @TieuDe, NoiDung = @NoiDung, MaPB = @MaPB, NgayGui = @NgayGui

WHERE Id = @Id;

END;

Thực hiện trên C#:

private void btnSua\_Click(object sender, EventArgs e)

{

try

{

SqlCommand cmd = new SqlCommand("sp\_CapNhatThongBao");

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@Id", cbbId.Text);

cmd.Parameters.AddWithValue("@TieuDe", txtTieuDe.Text);

cmd.Parameters.AddWithValue("@NoiDung", txtNoiDung.Text);

cmd.Parameters.AddWithValue("@MaPB", cbbMaPB.Text);

cmd.Parameters.AddWithValue("@NgayGui", dtpNgayGui.Value);

Connection.ExecuteCommand(cmd);

load();

MessageBox.Show("Sửa thành công!");

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi:" + ex.Message, "Thông báo lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

#### 4.1.6. Xóa thông báo

CREATE OR ALTER PROCEDURE sp\_XoaThongBao

@Id INT

AS

BEGIN

-- Xóa thông báo dựa trên Id

DELETE FROM ThongBao WHERE Id = @Id;

-- Tạo bảng tạm để lưu trữ các thông báo còn lại

DECLARE @ThongBaoTemp TABLE (

NewId INT IDENTITY(1,1),

TieuDe NVARCHAR(100),

NoiDung NVARCHAR(MAX),

MaPB NVARCHAR(10),

NgayGui DATETIME

);

-- Chèn các thông báo còn lại vào bảng tạm

INSERT INTO @ThongBaoTemp (TieuDe, NoiDung, MaPB, NgayGui)

SELECT TieuDe, NoiDung, MaPB, NgayGui

FROM ThongBao

ORDER BY Id;

-- Xóa tất cả các thông báo trong bảng ThongBao

DELETE FROM ThongBao;

-- Bật IDENTITY\_INSERT để chèn giá trị cụ thể vào cột IDENTITY

SET IDENTITY\_INSERT ThongBao ON;

-- Chèn lại các thông báo từ bảng tạm vào bảng ThongBao với Id mới

INSERT INTO ThongBao (Id, TieuDe, NoiDung, MaPB, NgayGui)

SELECT NewId, TieuDe, NoiDung, MaPB, NgayGui

FROM @ThongBaoTemp;

-- Tắt IDENTITY\_INSERT sau khi chèn xong

SET IDENTITY\_INSERT ThongBao OFF;

END;

Thực hiện trên C#:

private void btnXoa\_Click(object sender, EventArgs e)

{

try

{

SqlCommand cmd = new SqlCommand("sp\_XoaThongBao");

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@Id", cbbId.Text);

Connection.ExecuteCommand(cmd);

load();

MessageBox.Show("Xóa thành công!");

}

catch (SqlException ex)

{

MessageBox.Show("Lỗi:" + ex.Message, "Thông báo lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

### 4.2. Multi - Statement Table -Valued Function (không tham số): Xem thông tin thống kê về phòng ban

CREATE OR ALTER FUNCTION dbo.ft\_XemChiTietPhongBan()

RETURNS @PhongBanChiTiet TABLE (

MaPB nvarchar(10),

MaTrP nvarchar(10),

SoNV INT,

SoNVNam INT,

SoNVNu INT,

SoThongBao INT

)

AS

BEGIN

INSERT INTO @PhongBanChiTiet (MaPB, MaTrP, SoNV, SoNVNam, SoNVNu, SoThongBao)

SELECT

t.MaPB,

t.MaTrP,

COUNT(nv.MaNV) AS SoNV,

COUNT(CASE WHEN nv.GioiTinh LIKE N'Nam' THEN 1 END) AS SoNV\_Nam,

COUNT(CASE WHEN nv.GioiTinh LIKE N'Nữ' THEN 1 END) AS SoNV\_Nu,

t.SoThongBao

FROM

(SELECT pb.MaPB, pb.MaTrP, COUNT (tb.Id) as SoThongBao

FROM

PhongBan pb

LEFT JOIN

ThongBao tb ON tb.MaPB = pb.MaPB

GROUP BY pb.MaPB, pb.MaTrP) as t

LEFT JOIN

NhanVien nv ON t.MaPB = nv.MaPB

GROUP BY t.MaPB, t.SoThongBao, t.MaTrP

RETURN;

END;

GO

SELECT \* FROM dbo.ft\_XemChiTietPhongBan()

Thực hiện trên C#:

private void btnXemThongTin\_Click(object sender, EventArgs e)

{

try

{

using (SqlCommand command = new SqlCommand("SELECT \* FROM dbo.ft\_XemChiTietPhongBan()"))

{

command.CommandType = CommandType.Text;

Connection.ExecuteCommand(command);

FormXemChiTietPhongBan formXemChiTietPhongBan = new FormXemChiTietPhongBan();

formXemChiTietPhongBan.dataGridViewTTPB.DataSource = Connection.LoadDataTable(command);

formXemChiTietPhongBan.Show();

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

## 5. Quản lý thưởng phạt

### 5.1. Thủ tục có tham số

#### 5.1.1. Thêm thưởng phạt

CREATE OR ALTER PROCEDURE sp\_ThemThuongPhat

@MaThuongPhat nvarchar(10),

@Loai nvarchar(50),

@SoTien int,

@LyDo nvarchar(MAX)

AS

BEGIN

-- Kiểm tra nếu một hoặc nhiều tham số là NULL hoặc rỗng

IF @MaThuongPhat IS NULL OR @MaThuongPhat = '' OR @Loai IS NULL OR @Loai = '' OR

@SoTien IS NULL OR @SoTien ='' OR @LyDo ='' OR @LyDo IS NULL

BEGIN

RAISERROR('Dữ liệu không được bỏ trống', 16, 1)

RETURN;

END

ELSE

IF EXISTS (SELECT 1 FROM ThuongPhat WHERE MaThuongPhat = @MaThuongPhat)

BEGIN

RAISERROR('Mã Thưởng phạt đã tồn tại', 16, 1)

RETURN;

END

ELSE

-- Nếu không trùng mã và không có dữ liệu rỗng, thực hiện INSERT

INSERT INTO ThuongPhat(MaThuongPhat, Loai, SoTien, LyDo)

VALUES (@MaThuongPhat, @Loai, @SoTien, @LyDo);

END

CREATE OR ALTER PROCEDURE ThemctThuongPhat

@MaNV nvarchar(10),

@NgayThuongPhat date,

@MaThuongPhat nvarchar(10)

AS

BEGIN

IF EXISTS (SELECT 1 FROM ctThuongPhat WHERE MaNV = @MaNV and NgayThuongPhat = @NgayThuongPhat and MaThuongPhat = @MaThuongPhat)

BEGIN

RAISERROR('Dữ liệu đã tồn tại', 16, 1)

RETURN;

END

ELSE

INSERT INTO ctThuongPhat(MaNV, NgayThuongPhat, MaThuongPhat)

VALUES (@MaNV, @NgayThuongPhat, @MaThuongPhat);

END

Thực thi trên C#

private void btnThem\_Click(object sender, EventArgs e)

{

try

{

using (SqlCommand command = new SqlCommand("sp\_ThemThuongPhat"))

{

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@MaThuongPhat", this.txtMaTP.Text);

command.Parameters.AddWithValue("@Loai", this.cbbLoai.Text);

command.Parameters.AddWithValue("@SoTien", this.txtTien.Text);

command.Parameters.AddWithValue("@LyDo", this.txtLyDo.Text);

Connection.ExecuteCommand(command);

MessageBox.Show("Thêm thành công");

load();

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

#### 5.1.2 Thêm chi tiết thưởng phạt

CREATE OR ALTER PROCEDURE sp\_ThemctThuongPhat

@MaNV nvarchar(10),

@MaThuongPhat nvarchar(10),

@MaThang nvarchar(6),

@NgayThuongPhat int

AS

BEGIN

IF EXISTS (SELECT 1 FROM ctThuongPhat WHERE MaNV = @MaNV and MaThang = @MaThang and NgayThuongPhat = @NgayThuongPhat and MaThuongPhat = @MaThuongPhat)

BEGIN

RAISERROR('Dữ liệu đã tồn tại', 16, 1)

RETURN;

END

ELSE

INSERT INTO ctThuongPhat(MaNV, MaThuongPhat, MaThang, NgayThuongPhat)

VALUES (@MaNV, @MaThuongPhat, @MaThang, @NgayThuongPhat);

END;

Thực hiện trên C#:

private void btnThemTPNV\_Click(object sender, EventArgs e)

{

try

{

using (SqlCommand command = new SqlCommand("sp\_ThemctThuongPhat"))

{

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@MaThuongPhat", this.cbbMaTP.Text);

command.Parameters.AddWithValue("@MaNV", this.cbbMaNV.Text);

command.Parameters.AddWithValue("@MaThang", this.cbbMaThang.Text);

command.Parameters.AddWithValue("@NgayThuongPhat", this.txtNgayTP.Text);

Connection.ExecuteCommand(command);

MessageBox.Show("Thêm thành công");

load();

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

#### 5.1.3. Cập nhật thưởng phạt

CREATE OR ALTER PROCEDURE sp\_CapNhatThuongPhat

@MaThuongPhat nvarchar(10),

@Loai nvarchar(20),

@SoTien int,

@LyDo nvarchar(30)

AS

BEGIN

IF @MaThuongPhat IS NULL OR @MaThuongPhat = '' OR @Loai IS NULL OR @Loai = '' OR

@SoTien IS NULL OR @SoTien ='' OR @LyDo ='' OR @LyDo IS NULL

BEGIN

RAISERROR('Dữ liệu không được bỏ trống', 16, 1)

RETURN;

END

ELSE

-- Thực hiện cập nhật thông tin nhân viên trong bảng NhanVien dựa trên MaNV

UPDATE ThuongPhat

SET

MaThuongPhat = @MaThuongPhat,

Loai = @Loai,

SoTien = @SoTien,

LyDo = @LyDo

WHERE MaThuongPhat = @MaThuongPhat

END

Thực hiện trên C#

private void btnSua\_Click(object sender, EventArgs e)

{

try

{

using (SqlCommand command = new SqlCommand("sp\_CapNhatThuongPhat"))

{

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@MaThuongPhat", this.txtMaTP.Text);

command.Parameters.AddWithValue("@Loai", this.cbbLoai.Text);

command.Parameters.AddWithValue("@SoTien", this.txtTien.Text);

command.Parameters.AddWithValue("@LyDo", this.txtLyDo.Text);

Connection.ExecuteCommand(command);

MessageBox.Show("Cập nhật thành công");

load();

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

#### 5.1.4. Sửa ngày tháng của chi tiết thưởng phạt (có transaction)

CREATE OR ALTER PROCEDURE sp\_CapNhatNgayThangThuongPhat

@MaNV NVARCHAR(10),

@MaThuongPhat NVARCHAR(10),

@MaThang NVARCHAR(6),

@NgayThuongPhat INT

AS

BEGIN

BEGIN TRANSACTION;

DECLARE @OldMaThang NVARCHAR(6);

DECLARE @OldNgayThuongPhat INT;

SELECT @OldMaThang = MaThang, @OldNgayThuongPhat = NgayThuongPhat

FROM ctThuongPhat

WHERE MaThuongPhat = @MaThuongPhat AND MaNV = @MaNV;

-- Kiểm tra xem có dữ liệu cũ hay không

IF @OldMaThang IS NULL OR @OldNgayThuongPhat IS NULL

BEGIN

-- Nếu không có dữ liệu cũ, rollback giao dịch và trả về lỗi

ROLLBACK TRANSACTION;

RAISERROR('Không tìm thấy thông tin thưởng phạt để cập nhật.', 16, 1);

RETURN;

END

-- Cập nhật thông tin

UPDATE ctThuongPhat

SET

MaThang = @MaThang,

NgayThuongPhat = @NgayThuongPhat

WHERE MaThuongPhat = @MaThuongPhat AND MaNV = @MaNV;

-- Commit giao dịch nếu cập nhật thành công

COMMIT TRANSACTION;

END

Thực thi trên C#

private void btnSuaTPNV\_Click(object sender, EventArgs e)

{

try

{

using (SqlCommand command = new SqlCommand("sp\_CapNhatNgayThangThuongPhat"))

{

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@MaNV", this.cbbMaNV.Text);

command.Parameters.AddWithValue("MaThuongPhat", this.cbbMaTP.Text);

command.Parameters.AddWithValue("@MaThang", this.cbbMaThang.Text);

command.Parameters.AddWithValue("@NgayThuongPhat", this.txtNgayTP.Text);

Connection.ExecuteCommand(command);

MessageBox.Show("Cập nhật thành công");

load();

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

#### 5.1.5. Xóa thưởng phạt

CREATE OR ALTER PROCEDURE sp\_XoaThuongPhat

@MaThuongPhat nvarchar(10)

AS

BEGIN

DELETE FROM ThuongPhat

WHERE MaThuongPhat = @MaThuongPhat;

END;

Thực thi trên C#

private void btnXoa\_Click(object sender, EventArgs e)

{

try

{

using (SqlCommand command = new SqlCommand("sp\_XoaThuongPhat"))

{

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@MaThuongPhat", this.txtMaTP.Text);

Connection.ExecuteCommand(command);

MessageBox.Show("Xoá thành công");

load();

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

#### 5.1.6. Xóa chi tiết thưởng phạt

CREATE OR ALTER PROCEDURE sp\_XoactThuongPhat

@MaNV nvarchar(10),

@NgayThuongPhat int,

@MaTP nvarchar (10),

@MaThang nvarchar(6)

AS

BEGIN

DELETE FROM ctThuongPhat

WHERE MaNV=@MaNV and NgayThuongPhat = @NgayThuongPhat and MaThuongPhat = @MaTP and MaThang = @MaThang

END

Thực thi trên C#

private void btnXoaTPNV\_Click(object sender, EventArgs e)

{

try

{

using (SqlCommand command = new SqlCommand("sp\_XoactThuongPhat"))

{

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@MaNV", this.cbbMaNV.Text);

command.Parameters.AddWithValue("@MaThang", this.cbbMaThang.Text);

command.Parameters.AddWithValue("@NgayThuongPhat", this.txtNgayTP.Text);

command.Parameters.AddWithValue("@MaTP", this.cbbMaTP.Text);

Connection.ExecuteCommand(command);

MessageBox.Show("Xoá thành công");

load();

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

### 5.2. Inline Table – Valued Function (có tham số): Lọc chi tiết thưởng phạt theo mã nhân viên và theo loại thưởng/phạt

CREATE OR ALTER FUNCTION dbo.ft\_LocThuongPhatNhanVien(@MaNV NVARCHAR(10), @Loai NVARCHAR(50))

RETURNS TABLE

AS

RETURN

(

SELECT

nv.MaNV AS MaNhanVien,

nv.Ho AS Ho,

nv.Ten AS Ten,

tp.Loai AS Loai,

tp.LyDo AS LyDo,

tp.SoTien AS TienThuongPhat,

cttp.MaThang AS MaThang,

cttp.NgayThuongPhat AS NgayThuongPhat,

cv.TenCV AS TenChucVu,

pb.TenPB AS TenPhongBan

FROM

ctThuongPhat cttp

JOIN

ThuongPhat tp ON cttp.MaThuongPhat = tp.MaThuongPhat

JOIN

NhanVien nv ON cttp.MaNV = nv.MaNV

LEFT JOIN

ChucVu cv ON nv.MaCV = cv.MaCV

LEFT JOIN

PhongBan pb ON nv.MaPB = pb.MaPB

WHERE

tp.Loai = @Loai AND nv.MaNV = @MaNV

);

GO

SELECT \* FROM dbo.ft\_LocThuongPhatNhanVien('NV01', N'Thưởng')

Thực thi trên C#

private void btnXemThuongHayPhat\_Click(object sender, EventArgs e)

{

try

{

string loai = string.Empty;

if (rbThuong.Checked)

{

loai = "Thưởng";

}

else if (rbPhat.Checked)

{

loai = "Phạt";

}

using (SqlCommand cmd = new SqlCommand("SELECT \* FROM dbo.ft\_LocThuongPhatNhanVien(@MaNV, @Loai)"))

{

cmd.CommandType = CommandType.Text;

cmd.Parameters.AddWithValue("@MaNV", cbbMaNV.Text);

cmd.Parameters.AddWithValue("@Loai", loai);

// Hiển thị FormCTThuongPhat trong Panel

this.panelFormCTThuongPhat.Visible = true;

FormCTThuongPhat formCTThuongPhat = new FormCTThuongPhat();

formCTThuongPhat.TopLevel = false;

formCTThuongPhat.FormBorderStyle = FormBorderStyle.None;

formCTThuongPhat.Dock = DockStyle.Fill;

this.panelFormCTThuongPhat.Controls.Clear();

this.panelFormCTThuongPhat.Controls.Add(formCTThuongPhat);

formCTThuongPhat.dataGridView1.DataSource = Connection.LoadDataTable(cmd);

formCTThuongPhat.Show();

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

## 6. Quản lý chức vụ

### 6.1. Thêm chức vụ

|  |
| --- |
| CREATE OR ALTER PROCEDURE sp\_AddChucVu  @MaCV nvarchar(10),  @TenCV nvarchar(50)  AS  BEGIN  -- Kiểm tra nếu một hoặc nhiều tham số là NULL hoặc rỗng  IF @MaCV IS NULL OR @MaCV = '' OR @TenCV IS NULL OR @TenCV = ''  BEGIN  RAISERROR('Dữ liệu không được bỏ trống', 16, 1)  RETURN;  END  ELSE  INSERT INTO ChucVu(MaCV, TenCV)  VALUES (@MaCV, @TenCV);  END |

Thực thi trên C#

|  |
| --- |
| private void btnThem\_Click(object sender, EventArgs e)  {  try  {  using (SqlCommand command = new SqlCommand("sp\_AddChucVu"))  {  command.CommandType = CommandType.StoredProcedure;  command.Parameters.AddWithValue("@MaCV", this.txtMaCV.Text);  command.Parameters.AddWithValue("@TenCV", this.txtTenCV.Text);    Connection.ExecuteCommand(command);  MessageBox.Show("Thêm thành công");  load();  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

### 6.2. Cập nhật chức vụ

|  |
| --- |
| CREATE OR ALTER PROCEDURE sp\_UpdateChucVu  @MaCV nvarchar(10),  @TenCV nvarchar(50)  AS  BEGIN  IF @MaCV IS NULL OR @MaCV = '' OR @TenCV IS NULL OR @TenCV = ''  BEGIN  RAISERROR('Dữ liệu không được bỏ trống', 16, 1)  RETURN;  END  ELSE  UPDATE ChucVu  SET  MaCV = @MaCV,  TenCV = @TenCV  WHERE MaCV = @MaCV;  END; |

Thực thi trên C#

|  |
| --- |
| private void btnSua\_Click(object sender, EventArgs e)  {  try  {  using (SqlCommand command = new SqlCommand("sp\_UpdateChucVu"))  {  command.CommandType = CommandType.StoredProcedure;  command.Parameters.AddWithValue("@MaCV", this.txtMaCV.Text);  command.Parameters.AddWithValue("@TenCV", this.txtTenCV.Text);  Connection.ExecuteCommand(command);  MessageBox.Show("Cập nhật thành công");  load();  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

### 6.3. Xóa chức vụ

|  |
| --- |
| CREATE OR ALTER PROCEDURE sp\_DeleteChucVu  @MaCV nvarchar(10)  AS  BEGIN  DELETE FROM ChucVu  WHERE MaCV = @MaCV;  END; |

Thực thi trên C#

|  |
| --- |
| private void btnXoa\_Click(object sender, EventArgs e)  {  try  {  using (SqlCommand command = new SqlCommand("sp\_DeleteChucVu"))  {  command.CommandType = CommandType.StoredProcedure;  command.Parameters.AddWithValue("@MaCV", this.txtMaCV.Text);  Connection.ExecuteCommand(command);  MessageBox.Show("Xoá thành công");  load();  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

## 7. Quản lý phụ cấp

### 7.1. Thêm phụ cấp

|  |
| --- |
| CREATE OR ALTER PROCEDURE sp\_AddPhuCap  @MaPhuCap NVARCHAR(10),  @LoaiPhuCap NVARCHAR(10)  AS  BEGIN  INSERT INTO PhuCap (MaPhuCap, LoaiPhuCap)  VALUES (@MaPhuCap, @LoaiPhuCap);    PRINT 'Thêm thành công!';  END; |

Thực thi trên C#

|  |
| --- |
| private void btnThem\_Click(object sender, EventArgs e)  {  try  {  using (SqlCommand command = new SqlCommand("sp\_AddPhuCap"))  {  command.CommandType = CommandType.StoredProcedure;  command.Parameters.AddWithValue("@MaPhuCap", this.txtMaPC.Text);  command.Parameters.AddWithValue("@LoaiPhuCap", this.txtLoaiPC.Text);  Connection.ExecuteCommand(command);  MessageBox.Show("Thêm phụ cấp thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  load();  }  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

### 7.2. Sửa phụ cấp

|  |
| --- |
| CREATE OR ALTER PROCEDURE sp\_UpdatePhuCap  @MaPhuCap NVARCHAR(10),  @LoaiPhuCap NVARCHAR(10)  AS  BEGIN  UPDATE PhuCap  SET LoaiPhuCap = @LoaiPhuCap  WHERE MaPhuCap = @MaPhuCap;    PRINT 'Cập nhật thành công!';  END; |

Thực thi trên C#

|  |
| --- |
| private void btnThem\_Click(object sender, EventArgs e)  {  try  {  using (SqlCommand command = new SqlCommand("sp\_AddPhuCap"))  {  command.CommandType = CommandType.StoredProcedure;  command.Parameters.AddWithValue("@MaPhuCap", this.txtMaPC.Text);  command.Parameters.AddWithValue("@LoaiPhuCap", this.txtLoaiPC.Text);  Connection.ExecuteCommand(command);  MessageBox.Show("Thêm phụ cấp thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  load();  }  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

### 7.3. Xóa phụ cấp

|  |
| --- |
| CREATE OR ALTER PROCEDURE sp\_DeletePhuCap  @MaPhuCap NVARCHAR(10)  AS  BEGIN  -- Xóa bản ghi phụ cấp  DELETE FROM PhuCap  WHERE MaPhuCap = @MaPhuCap;  PRINT 'Xóa thành công!';  END; |

Thực thi trên C#

|  |
| --- |
| private void btnXoa\_Click(object sender, EventArgs e)  {  try  {  using (SqlCommand command = new SqlCommand("sp\_DeletePhuCap"))  {  command.CommandType = CommandType.StoredProcedure;  command.Parameters.AddWithValue("@MaPhuCap", this.txtMaPC.Text);  Connection.ExecuteCommand(command);  MessageBox.Show("Xóa phụ cấp thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  load();  }  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

### 7.4. Thêm chi tiết phụ cấp

|  |
| --- |
| CREATE OR ALTER PROCEDURE sp\_AddctPhuCap  @MaNV NVARCHAR(10),  @MaPhuCap NVARCHAR(10),  @MaThang NVARCHAR(6),  @NgayPhuCap INT,  @SoTien INT  AS  BEGIN  INSERT INTO ctPhuCap (MaNV, MaPhuCap, MaThang, NgayPhuCap, SoTien)  VALUES (@MaNV, @MaPhuCap, @MaThang, @NgayPhuCap, @SoTien);  PRINT 'Thêm thành công!';  END; |

Thực thi trên C#

|  |
| --- |
| private void btnThemct\_Click(object sender, EventArgs e)  {  try  {  using (SqlCommand command = new SqlCommand("sp\_AddctPhuCap"))  {  command.CommandType = CommandType.StoredProcedure;  command.Parameters.AddWithValue("@MaNV", this.cbbMaNV.Text);  command.Parameters.AddWithValue("@MaPhuCap", this.cbbMaPhuCapctPhuCap.Text);  command.Parameters.AddWithValue("@SoTien", this.txtSoTien.Text);  string maThang = dtpNgayPhuCap.Value.ToString("MMyyyy");  string ngayPhuCap = dtpNgayPhuCap.Value.ToString("dd");  command.Parameters.AddWithValue("@MaThang", maThang);  command.Parameters.AddWithValue("@NgayPhuCap", ngayPhuCap);  Connection.ExecuteCommand(command);  MessageBox.Show("Thêm phụ cấp thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  load();  }  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

### 7.5. Sửa chi tiết phụ cấp

|  |
| --- |
| CREATE OR ALTER PROCEDURE sp\_UpdatectPhuCap  @MaNV NVARCHAR(10),  @MaPhuCap NVARCHAR(10),  @MaThang NVARCHAR(6),  @NgayPhuCap INT,  @SoTien INT  AS  BEGIN  UPDATE ctPhuCap  SET MaNV = @MaNV,  MaPhuCap=@MaPhuCap,  MaThang=@MaThang,  NgayPhuCap=@NgayPhuCap,  SoTien=@SoTien  WHERE MaNV = @MaNV AND MaPhuCap = @MaPhuCap AND MaThang=@MaThang AND NgayPhuCap=@NgayPhuCap;    PRINT 'Cập nhật thành công!';  END; |

Thực thi trên C#

|  |
| --- |
| private void btnSuact\_Click(object sender, EventArgs e)  {  try  {  using (SqlCommand command = new SqlCommand("sp\_UpdatectPhuCap"))  {  command.CommandType = CommandType.StoredProcedure;  command.Parameters.AddWithValue("@MaNV", this.cbbMaNV.Text);  command.Parameters.AddWithValue("@MaPhuCap", this.cbbMaPhuCapctPhuCap.Text);  string maThang = dtpNgayPhuCap.Value.ToString("MMyyyy");  string ngayPhuCap = dtpNgayPhuCap.Value.ToString("dd");  command.Parameters.AddWithValue("@MaThang", maThang);  command.Parameters.AddWithValue("@NgayPhuCap", ngayPhuCap);  command.Parameters.AddWithValue("@SoTien", this.txtSoTien.Text);  Connection.ExecuteCommand(command);  MessageBox.Show("Sửa phụ cấp thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  load();  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

### 7.6. Xóa chi tiết phụ cấp

|  |
| --- |
| CREATE OR ALTER PROCEDURE sp\_DeletectPhuCap  @MaNV NVARCHAR(10),  @MaPhuCap NVARCHAR(10),  @MaThang NVARCHAR(6),  @NgayPhuCap INT  AS  BEGIN  DELETE FROM ctPhuCap  WHERE MaNV = @MaNV AND MaPhuCap=@MaPhuCap AND MaThang=@MaThang AND NgayPhuCap=@NgayPhuCap;  PRINT 'Xóa thành công!';  END; |

Thực thi trên C#

|  |
| --- |
| private void btnXoact\_Click(object sender, EventArgs e)  {  try  {  using (SqlCommand command = new SqlCommand("sp\_DeletectPhuCap"))  {  command.CommandType = CommandType.StoredProcedure;  command.Parameters.AddWithValue("@MaNV", this.cbbMaNV.Text);  command.Parameters.AddWithValue("@MaPhuCap", this.cbbMaPhuCapctPhuCap.Text);  string maThang = dtpNgayPhuCap.Value.ToString("MMyyyy");  string ngayPhuCap = dtpNgayPhuCap.Value.ToString("dd");  command.Parameters.AddWithValue("@MaThang", maThang);  command.Parameters.AddWithValue("@NgayPhuCap", ngayPhuCap);  Connection.ExecuteCommand(command);  MessageBox.Show("Xóa phụ cấp thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  load();  }  }  catch (SqlException ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

## 8. Quản lý chấm công

### 8.1. Thủ tục có tham số

#### 8.1.1. Thêm chấm công

CREATE OR ALTER PROCEDURE sp\_AddctChamCong

@MaNV VARCHAR(10),

@MaCC VARCHAR(10),

@MaThang VARCHAR(6),

@NgayChamCong int

AS

BEGIN

IF EXISTS (SELECT 1 FROM ctChamCong WHERE MaNV = @MaNV AND MaThang = @MaThang AND NgayChamCong = @NgayChamCong)

BEGIN

RAISERROR('Đã chấm công cho ngày này', 16, 1);

END

ELSE

BEGIN

INSERT INTO ctChamCong (MaNV, MaCC, MaThang, NgayChamCong)

VALUES (@MaNV, @MaCC, @MaThang, @NgayChamCong);

END

END;

Thực thi trên C#

private void btnThem\_Click(object sender, EventArgs e)

{

try

{

using (SqlCommand command = new SqlCommand("sp\_AddctChamCong"))

{

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@MaNV", cbbMaNV.Text);

command.Parameters.AddWithValue("@MaThang", cbbThang.SelectedValue.ToString());

command.Parameters.AddWithValue("@NgayChamCong", txtNgayCC.Text);

command.Parameters.AddWithValue("@MaCC", cbbLoaiCong.SelectedValue);

Connection.ExecuteCommand(command);

MessageBox.Show("Chấm công thành công");

load();

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

#### 8.1.2. Cập nhật chấm công

CREATE OR ALTER PROCEDURE sp\_UpdatectChamCong

@MaNV VARCHAR(10),

@MaCC VARCHAR(10),

@MaThang VARCHAR(6),

@NgayChamCong int

AS

BEGIN

-- Kiểm tra xem bản ghi có tồn tại trước khi cập nhật

IF EXISTS (SELECT 1 FROM ctChamCong WHERE MaNV = @MaNV AND MaThang = @MaThang AND NgayChamCong = @NgayChamCong)

BEGIN

UPDATE ctChamCong

SET MaCC = @MaCC

WHERE MaNV = @MaNV AND MaThang = @MaThang AND NgayChamCong = @NgayChamCong;

END

ELSE

BEGIN

RAISERROR('Không tìm thấy chi tiết chấm công theo yêu cầu.', 16, 1);

END

END;

Thực thi trên C#

private void btnSua\_Click(object sender, EventArgs e)

{

try

{

using (SqlCommand command = new SqlCommand("sp\_UpdatectChamCong"))

{

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@MaNV", cbbMaNV.Text);

command.Parameters.AddWithValue("@MaThang", cbbThang.SelectedValue.ToString());

command.Parameters.AddWithValue("@NgayChamCong", txtNgayCC.Text);

command.Parameters.AddWithValue("@MaCC", cbbLoaiCong.SelectedValue);

Connection.ExecuteCommand(command);

MessageBox.Show("Cập nhật thành công");

load();

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

#### 8.1.3. Xóa chấm công

CREATE OR ALTER PROCEDURE sp\_DeletectChamCong

@MaNV VARCHAR(10),

@MaThang VARCHAR(6),

@NgayChamCong int

AS

BEGIN

DELETE FROM ctChamCong WHERE MaNV = @MaNV AND MaThang = @MaThang AND NgayChamCong = @NgayChamCong;

END;

Thực thi trên C#

private void btnXoa\_Click(object sender, EventArgs e)

{

try

{

using (SqlCommand command = new SqlCommand("sp\_DeletectChamCong"))

{

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@MaNV", cbbMaNV.Text);

command.Parameters.AddWithValue("@MaThang", cbbThang.SelectedValue);

command.Parameters.AddWithValue("@NgayChamCong", txtNgayCC.Text);

Connection.ExecuteCommand(command);

MessageBox.Show("Xóa thành công");

clearFields();

load();

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

### 8.2. Thủ tục không tham số, trả về một bảng: Xem danh sách chấm công của tất cả nhân viên

|  |
| --- |
| CREATE OR ALTER PROCEDURE sp\_GetAllctChamCong  AS  BEGIN  SELECT  ct.MaNV,  ct.NgayChamCong,  thg.MoTa AS Thang,  cong.MoTa AS MoTa,  cong.HeSo  FROM  ctChamCong ct  JOIN  ChamCong cong ON cong.MaCC = ct.MaCC  JOIN  Thang thg ON thg.MaThang = ct.MaThang  END; |

Thực thi trên C#

|  |
| --- |
| private void load()  {  SqlCommand cmd = new SqlCommand("sp\_GetAllctChamCong");  cmd.CommandType = CommandType.StoredProcedure;  dataGVctChamCong.DataSource = Connection.LoadDataTable(cmd);  } |

### 8.3. Thủ tục có tham số, trả về một bảng: Tìm danh sách chấm công của một nhân viên

|  |
| --- |
| CREATE OR ALTER PROCEDURE sp\_GetctChamCong  @MaNV VARCHAR(10),  @MaThang VARCHAR(6)  AS  BEGIN  IF (@MaThang IS NULL)  BEGIN  SELECT  ct.MaNV,  ct.NgayChamCong,  thg.MoTa AS Thang,  cong.MoTa AS MoTa,  cong.HeSo  FROM  ctChamCong ct  JOIN  ChamCong cong ON cong.MaCC = ct.MaCC  JOIN  Thang thg ON thg.MaThang = ct.MaThang  WHERE  ct.MaNV = @MaNV  ORDER BY thg.MaThang, ct.NgayChamCong DESC;  END  ELSE  BEGIN  SELECT  ct.MaNV,  ct.NgayChamCong,  thg.MoTa AS Thang,  cong.MoTa AS MoTa,  cong.HeSo  FROM  ctChamCong ct  JOIN  ChamCong cong ON cong.MaCC = ct.MaCC  JOIN  Thang thg ON thg.MaThang = ct.MaThang  WHERE  ct.MaNV = @MaNV and ct.MaThang = @MaThang  ORDER BY thg.MaThang, ct.NgayChamCong DESC;  END  END; |

Thực thi trên C#

|  |
| --- |
| private void btnTim\_Click(object sender, EventArgs e)  {  SqlCommand cmd = new SqlCommand("sp\_GetctChamCong");  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaNV", cbbMaNV.Text);  cmd.Parameters.AddWithValue("@MaThang", cbbThang.SelectedValue ?? DBNull.Value);  dataGVctChamCong.DataSource = Connection.LoadDataTable(cmd);  } |

## 9. Nhân viên chấm công ngày làm việc

### 9.1. Chấm công

private void btnChamCong\_Click(object sender, EventArgs e)

{

try

{

DateTime today = DateTime.Now;

string month = today.Month.ToString();

if (month.Length == 1)

{

month = "0" + month;

}

string year = today.Year.ToString();

using (SqlCommand command = new SqlCommand("sp\_AddctChamCong"))

{

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@MaNV", SessionInfo.MaNV);

command.Parameters.AddWithValue("@MaThang", month + year);

command.Parameters.AddWithValue("@NgayChamCong", today.Day);

command.Parameters.AddWithValue("@MaCC", "CC01");

Connection.ExecuteCommand(command);

MessageBox.Show("Chấm công thành công");

load();

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

### 9.2. Xem danh sách chấm công theo Mã Nhân viên

Thủ tục có tham số, trả về một bảng:

CREATE OR ALTER PROCEDURE sp\_GetChamCongByMaNV

@MaNV VARCHAR(10)

AS

BEGIN

SELECT

ct.NgayChamCong,

thg.MoTa AS Thang,

cong.MoTa AS MoTa,

cong.HeSo

FROM

ctChamCong ct

JOIN

ChamCong cong ON cong.MaCC = ct.MaCC

JOIN

Thang thg ON thg.MaThang = ct.MaThang

WHERE

ct.MaNV = @MaNV;

END;

Thực thi trên C#

private void load()

{

SqlCommand cmd = new SqlCommand("sp\_GetChamCongByMaNV");

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaNV", SessionInfo.MaNV);

dataGridViewChamCong.DataSource = Connection.LoadDataTable(cmd);

txtMaNV.Text = SessionInfo.MaNV;

}

## 10. Quản lý bảo hiểm

### 10.1. Xem danh sách bảo hiểm

|  |
| --- |
| private void load()  {  string queryAllBH = "Select \* from vw\_QuanLyBaoHiem";  dataGridViewBaoHiemNV.DataSource = Connection.LoadDataTable(queryAllBH);  } |

### 10.2. Thủ tục có tham số, trả về một bảng: Tìm thông tin bảo hiểm của một nhân viên

|  |
| --- |
| CREATE OR ALTER PROCEDURE sp\_GetctBaoHiemByMaNV  @MaNV VARCHAR(10)  AS  BEGIN  SELECT \* FROM vw\_QuanLyBaoHiem WHERE MaNV = @MaNV;  END; |

Thực thi trên C#

|  |
| --- |
| private void btnTim\_Click(object sender, EventArgs e)  {  SqlCommand cmd = new SqlCommand("sp\_GetctBaoHiemByMaNV");  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaNV", this.cbbMaNV.Text);  dataGridViewBaoHiemNV.DataSource = Connection.LoadDataTable(cmd);  } |

### 10.3. Thủ tục có tham số, không trả về kết quả

#### 10.3.1. Tạo bảo hiểm mới cho một nhân viên

|  |
| --- |
| CREATE OR ALTER PROCEDURE sp\_AddctBaoHiem  @MaNV VARCHAR(10),  @MaLoai VARCHAR(10),  @MaBH VARCHAR(10),  @NgayBD DATE,  @NgayKT DATE  AS  BEGIN  INSERT INTO ctBaoHiem (MaNV, MaBH, MaLoai, NgayBD, NgayKT) VALUES (@MaNV, @MaBH, @MaLoai, @NgayBD, @NgayKT);  END; |

Thực thi trên C#

|  |
| --- |
| private void btnThem\_Click(object sender, EventArgs e)  {  try  {  using (SqlCommand command = new SqlCommand("sp\_AddctBaoHiem"))  {  command.CommandType = CommandType.StoredProcedure;  command.Parameters.AddWithValue("@MaNV", this.cbbMaNV.Text);  command.Parameters.AddWithValue("@MaLoai", this.cbbLoaiBH.SelectedValue.ToString());  command.Parameters.AddWithValue("@MaBH", this.txtMaBH.Text);  command.Parameters.AddWithValue("@NgayBD", this.dateTimePickerNgayBD.Value);  command.Parameters.AddWithValue("@NgayKT", this.dateTimePickerNgayKT.Value);  Connection.ExecuteCommand(command);  MessageBox.Show("Thêm thành công");  load();  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

#### 10.3.2. Sửa thông tin bảo hiểm

|  |
| --- |
| CREATE OR ALTER PROCEDURE sp\_UpdatectBaoHiem  @LoaiBH VARCHAR(10),  @MaBH VARCHAR(10),  @NgayBD DATE,  @NgayKT DATE  AS  BEGIN  UPDATE ctBaoHiem SET MaLoai=@LoaiBH, NgayBD=@NgayBD, NgayKT=@NgayKT WHERE MaBH=@MaBH;  END; |

Thực thi trên C#

private void btnSua\_Click(object sender, EventArgs e)

{

try

{

using (SqlCommand command = new SqlCommand("UpdatectBaoHiem"))

{

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@MaNV", this.cbbMaNV.Text);

command.Parameters.AddWithValue("@LoaiBH", this.cbbLoaiBH.Text);

command.Parameters.AddWithValue("@MaBH", this.txtMaBH.Text);

command.Parameters.AddWithValue("@NgayBD", this.dateTimePickerNgayBD.Value);

command.Parameters.AddWithValue("@NgayKT", this.dateTimePickerNgayKT.Value);

Connection.ExecuteCommand(command);

MessageBox.Show("Cập nhật thành công");

load();

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

#### 10.3.3. Xóa chi tiết bảo hiểm

|  |
| --- |
| CREATE OR ALTER PROCEDURE sp\_DeletectBaoHiem  @MaBH VARCHAR(10)  AS  BEGIN  DELETE FROM ctBaoHiem WHERE MaBH=@MaBH;  END; |

Thực thi trên C#

|  |
| --- |
| private void btnXoa\_Click(object sender, EventArgs e)  {  try  {  using (SqlCommand command = new SqlCommand("sp\_DeletectBaoHiem"))  {  command.CommandType = CommandType.StoredProcedure;  command.Parameters.AddWithValue("@MaBH", this.txtMaBH.Text);  Connection.ExecuteCommand(command);  MessageBox.Show("Xóa thành công");  load();  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

## 11. Nhân viên đăng ký nghỉ phép

### 11.1. Thủ tục có tham số, trả về một bảng: Xem lịch sử nghỉ phép

|  |
| --- |
| CREATE OR ALTER PROCEDURE sp\_GetNghiPhepByMaNV  @MaNV VARCHAR(10)  AS  BEGIN  SELECT  NgayNghiPhep,  thg.MoTa AS Thang,  GhiChu  FROM  NghiPhep np  JOIN  Thang thg ON thg.MaThang = np.MaThang  WHERE  MaNV = @MaNV;  END; |

Thực thi trên C#

|  |
| --- |
| private void load()  {  SqlCommand cmd = new SqlCommand("sp\_GetNghiPhepByMaNV");  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaNV", SessionInfo.MaNV);  dataGVoffHistory.DataSource = Connection.LoadDataTable(cmd);  } |

### 11.2. Thủ tục có tham số: Xin nghỉ phép

CREATE OR ALTER PROCEDURE sp\_AddNghiPhep

@MaNV VARCHAR(10),

@MaThang VARCHAR(6),

@NgayNghiPhep int,

@GhiChu NVARCHAR(MAX)

AS

BEGIN

INSERT INTO NghiPhep (MaNV, MaThang, NgayNghiPhep, GhiChu) VALUES (@MaNV, @MaThang, @NgayNghiPhep, @GhiChu);

END;

Thực thi trên C#

private void btnNghiPhep\_Click(object sender, System.EventArgs e)

{

try

{

using (SqlCommand command = new SqlCommand("sp\_AddNghiPhep"))

{

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@MaNV", this.txtMaNV.Text);

command.Parameters.AddWithValue("@MaThang", this.cbbThang.SelectedValue.ToString());

command.Parameters.AddWithValue("@NgayNghiPhep", this.txtNgayNghi.Text);

command.Parameters.AddWithValue("@GhiChu", this.txtLyDo.Text);

Connection.ExecuteCommand(command);

MessageBox.Show("Đăng ký nghỉ thành công");

txtNgayNghi.Text = txtLyDo.Text = "";

load();

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

## 12. Thống kê lương cho nhân viên

### 12.1. Thủ tục có tham số trả về một bảng: Tính lương nhân viên theo tháng

|  |
| --- |
| CREATE OR ALTER PROCEDURE sp\_TinhLuongTheoThang  @MaThang VARCHAR(6),  @MaNV VARCHAR(10) = NULL -- Thêm MaNV để chỉ tính lương cho nhân viên cụ thể  AS  BEGIN  -- Bảng tạm chứa số ngày công  DECLARE @ChamCongThang TABLE (  MaNV VARCHAR(10),  SoNgayCong FLOAT  );  INSERT INTO @ChamCongThang (MaNV, SoNgayCong)  SELECT MaNV, SUM(HeSo)  FROM ctChamCong ct  JOIN ChamCong cc ON ct.MaCC = cc.MaCC  WHERE MaThang = @MaThang  AND (@MaNV IS NULL OR ct.MaNV = @MaNV) -- Lọc theo MaNV nếu có  GROUP BY MaNV, MaThang;  -- Bảng tạm chứa thông tin người phụ thuộc  DECLARE @NguoiPhuThuoc TABLE (  MaNV VARCHAR(10),  SoNguoiPhuThuoc INT,  GiamTruGiaCanh INT  );  INSERT INTO @NguoiPhuThuoc (MaNV, SoNguoiPhuThuoc, GiamTruGiaCanh)  SELECT MaNV, COUNT(\*), ISNULL(COUNT(\*), 0) \* 4400000  FROM NguoiPhuThuoc  WHERE (@MaNV IS NULL OR MaNV = @MaNV) -- Lọc theo MaNV nếu có  GROUP BY MaNV;  -- Bảng tạm chứa phụ cấp tháng  DECLARE @PhuCapThang TABLE (  MaNV VARCHAR(10),  TongPhuCap INT DEFAULT 0  );  INSERT INTO @PhuCapThang (MaNV, TongPhuCap)  SELECT MaNV, ISNULL(SUM(SoTien), 0)  FROM ctPhuCap  WHERE MaThang = @MaThang  AND (@MaNV IS NULL OR MaNV = @MaNV) -- Lọc theo MaNV nếu có  GROUP BY MaNV, MaThang;  -- Bảng tạm chứa thưởng phạt tháng  DECLARE @ThuongPhatThang TABLE (  MaNV VARCHAR(10),  TongThuongPhat INT DEFAULT 0  );  INSERT INTO @ThuongPhatThang (MaNV, TongThuongPhat)  SELECT MaNV, ISNULL(SUM(SoTien), 0)  FROM ctThuongPhat ctp  JOIN ThuongPhat tp ON ctp.MaThuongPhat = tp.MaThuongPhat  WHERE MaThang = @MaThang  AND (@MaNV IS NULL OR ctp.MaNV = @MaNV) -- Lọc theo MaNV nếu có  GROUP BY MaNV, MaThang;  -- Bảng tạm chứa bảo hiểm  DECLARE @BaoHiem TABLE (  MaNV VARCHAR(10),  BH01 FLOAT DEFAULT 0,  BH02 FLOAT DEFAULT 0,  BH03 FLOAT DEFAULT 0  );  INSERT INTO @BaoHiem (MaNV, BH01, BH02, BH03)  SELECT bh.MaNV,  SUM(CASE WHEN bh.MaLoai = 'BH01' THEN 0.015 \* hd.LuongCoBan ELSE 0 END),  SUM(CASE WHEN bh.MaLoai = 'BH02' THEN 0.01 \* hd.LuongCoBan ELSE 0 END),  SUM(CASE WHEN bh.MaLoai = 'BH03' THEN 0.08 \* hd.LuongCoBan ELSE 0 END)  FROM ctBaoHiem bh  JOIN HopDong hd ON bh.MaNV = hd.MaNV  WHERE (@MaNV IS NULL OR bh.MaNV = @MaNV) -- Lọc theo MaNV nếu có  GROUP BY bh.MaNV;  -- Số ngày công chuẩn  DECLARE @SoNgayCongChuan INT;  SELECT @SoNgayCongChuan = SoNgayCongChuan  FROM Thang WHERE MaThang = @MaThang;  -- Tính lương chịu thuế  DECLARE @LuongChiuThue TABLE (  MaNV VARCHAR(10),  LuongCoBan INT,  GiamTruGiaCanh INT,  TongPhuCap INT,  TongThuongPhat INT,  BH01 INT, BH02 INT, BH03 INT,  TongTienBaoHiem INT,  SoNgayCong INT,  LuongChiuThue INT  );  INSERT INTO @LuongChiuThue (MaNV, LuongCoBan, GiamTruGiaCanh, TongPhuCap, TongThuongPhat, BH01, BH02, BH03, TongTienBaoHiem, SoNgayCong, LuongChiuThue)  SELECT  nv.MaNV, LuongCoBan, ISNULL(npt.GiamTruGiaCanh, 0), ISNULL(pc.TongPhuCap, 0), ISNULL(tp.TongThuongPhat, 0),  BH01, BH02, BH03, ISNULL(bhct.BH01, 0) + ISNULL(bhct.BH02, 0) + ISNULL(bhct.BH03, 0), ISNULL(SoNgayCong, 0),  ( ( (hd.LuongCoBan / @SoNgayCongChuan \* ISNULL(cc.SoNgayCong, 0)) + ISNULL(pc.TongPhuCap, 0) + ISNULL(tp.TongThuongPhat, 0) )  - (ISNULL(bhct.BH01, 0) + ISNULL(bhct.BH02, 0) + ISNULL(bhct.BH03, 0)) - ISNULL(npt.GiamTruGiaCanh, 0))  FROM NhanVien nv  JOIN HopDong hd ON nv.MaNV = hd.MaNV  LEFT JOIN @ChamCongThang cc ON nv.MaNV = cc.MaNV  LEFT JOIN @NguoiPhuThuoc npt ON nv.MaNV = npt.MaNV  LEFT JOIN @PhuCapThang pc ON nv.MaNV = pc.MaNV  LEFT JOIN @ThuongPhatThang tp ON nv.MaNV = tp.MaNV  LEFT JOIN @BaoHiem bhct ON nv.MaNV = bhct.MaNV  WHERE @MaNV IS NULL OR nv.MaNV = @MaNV; -- Lọc theo MaNV nếu có  -- Tính thuế thu nhập cá nhân  DECLARE @ThueThuNhapCaNhan TABLE (  MaNV VARCHAR(10),  Thue INT  );  INSERT INTO @ThueThuNhapCaNhan (MaNV, Thue)  SELECT nv.MaNV,  CASE  WHEN LuongChiuThue <= 5000000  THEN LuongChiuThue \* 5 / 100  WHEN LuongChiuThue <= 10000000  THEN LuongChiuThue \* 10 / 100  WHEN LuongChiuThue <= 18000000  THEN LuongChiuThue \* 15 / 100  WHEN LuongChiuThue <= 32000000  THEN LuongChiuThue \* 20 / 100  WHEN LuongChiuThue <= 52000000  THEN LuongChiuThue \* 25 / 100  WHEN LuongChiuThue <= 80000000  THEN LuongChiuThue \* 30 / 100  ELSE LuongChiuThue \* 35 / 100  END AS ThueThuNhapCaNhan  FROM NhanVien nv  LEFT JOIN @LuongChiuThue lct ON nv.MaNV = lct.MaNV  WHERE @MaNV IS NULL OR nv.MaNV = @MaNV; -- Lọc theo MaNV nếu có  -- Kết quả lương của nhân viên  SELECT nv.MaNV, Ho, Ten, LuongCoBan, BH01, BH02, BH03,  (BH01 + BH02 + BH03) AS TongTienBaoHiem,  @SoNgayCongChuan AS SoNgayCongChuan,  TongPhuCap, GiamTruGiaCanh, LuongChiuThue,  SoNgayCong, Thue, TongThuongPhat, (LuongChiuThue - Thue) AS LuongThucLanh  FROM NhanVien nv  LEFT JOIN @LuongChiuThue lct ON nv.MaNV = lct.MaNV  LEFT JOIN @ThueThuNhapCaNhan thue ON nv.MaNV = thue.MaNV  WHERE @MaNV IS NULL OR nv.MaNV = @MaNV; -- Lọc theo MaNV nếu có  END; |

Thực thi trên C#

|  |
| --- |
| private void btnXemLuong\_Click(object sender, EventArgs e)  {  SqlCommand cmd = new SqlCommand("sp\_TinhLuongTheoThang");  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@MaThang", cbbMaThang.SelectedValue);  dgvLuong.DataSource = Connection.LoadDataTable(cmd);  } |

### 12.2. Thủ tục không tham số, trả về nhiều bảng: Xem chi tiết lương

|  |
| --- |
| CREATE OR ALTER PROCEDURE sp\_TinhLuongChiTietTatCaNhanVien  AS  BEGIN  -- Bảng phụ tính toán chấm công  DECLARE @ChamCong TABLE (  MaNV VARCHAR(10),  MaThang VARCHAR(6),  SoNgayLamViec INT  );  INSERT INTO @ChamCong (MaNV, MaThang, SoNgayLamViec)  SELECT MaNV, MaThang, COUNT(\*) AS SoNgayLamViec  FROM ctChamCong  WHERE MaCC = 'CC01'  GROUP BY MaNV, MaThang;  -- Bảng phụ người phụ thuộc  DECLARE @NguoiPhuThuoc TABLE (  MaNV VARCHAR(10),  SoNguoiPhuThuoc INT  );  INSERT INTO @NguoiPhuThuoc (MaNV, SoNguoiPhuThuoc)  SELECT MaNV, COUNT(\*) AS SoNguoiPhuThuoc  FROM NguoiPhuThuoc  GROUP BY MaNV;  -- Bảng phụ phụ cấp  DECLARE @PhuCap TABLE (  MaNV VARCHAR(10),  MaThang VARCHAR(6),  TongPhuCap INT  );  INSERT INTO @PhuCap (MaNV, MaThang, TongPhuCap)  SELECT MaNV, MaThang, ISNULL(SUM(SoTien), 0)  FROM ctPhuCap  GROUP BY MaNV, MaThang;  -- Bảng phụ thưởng phạt  DECLARE @ThuongPhat TABLE (  MaNV VARCHAR(10),  MaThang VARCHAR(6),  TongThuongPhat INT  );  INSERT INTO @ThuongPhat (MaNV, MaThang, TongThuongPhat)  SELECT MaNV, MaThang, ISNULL(SUM(tp.SoTien), 0)  FROM ctThuongPhat ctp  JOIN ThuongPhat tp ON ctp.MaThuongPhat = tp.MaThuongPhat  GROUP BY MaNV, MaThang;  -- Bảng phụ bảo hiểm  DECLARE @BaoHiem TABLE (  MaNV VARCHAR(10),  BH01 FLOAT,  BH02 FLOAT,  BH03 FLOAT  );  INSERT INTO @BaoHiem (MaNV, BH01, BH02, BH03)  SELECT bh.MaNV,  SUM(CASE WHEN bh.MaLoai = 'BH01' THEN 0.015 \* hd.LuongCoBan ELSE 0 END),  SUM(CASE WHEN bh.MaLoai = 'BH02' THEN 0.01 \* hd.LuongCoBan ELSE 0 END),  SUM(CASE WHEN bh.MaLoai = 'BH03' THEN 0.08 \* hd.LuongCoBan ELSE 0 END)  FROM ctBaoHiem bh  JOIN HopDong hd ON bh.MaNV = hd.MaNV  GROUP BY bh.MaNV;  -- Bảng phụ lương chịu thuế  DECLARE @LuongChiuThue TABLE (  MaNV VARCHAR(10),  MaThang VARCHAR(6),  LuongCoBan INT,  TongPhuCap INT,  TongThuongPhat INT,  TongTienBaoHiem INT,  SoNgayLamViec INT,  GiamTruGiaCanh INT,  ThuNhapChiuThue INT  );  INSERT INTO @LuongChiuThue (MaNV, MaThang, LuongCoBan, TongPhuCap, TongThuongPhat, TongTienBaoHiem, SoNgayLamViec, GiamTruGiaCanh, ThuNhapChiuThue)  SELECT  nv.MaNV, th.MaThang, hd.LuongCoBan,  ISNULL(pc.TongPhuCap, 0), ISNULL(tp.TongThuongPhat, 0),  ISNULL(bhct.BH01, 0) + ISNULL(bhct.BH02, 0) + ISNULL(bhct.BH03, 0) AS TongTienBaoHiem,  ISNULL(cc.SoNgayLamViec, 0),  ISNULL(npt.SoNguoiPhuThuoc, 0) \* 4400000 AS GiamTruGiaCanh,  ( ( (hd.LuongCoBan / th.SoNgayCongChuan \* ISNULL(cc.SoNgayLamViec, 0))  + ISNULL(pc.TongPhuCap, 0) + ISNULL(tp.TongThuongPhat, 0))  - (ISNULL(bhct.BH01, 0) + ISNULL(bhct.BH02, 0) + ISNULL(bhct.BH03, 0))  - ISNULL(npt.SoNguoiPhuThuoc, 0) \* 4400000  ) AS ThuNhapChiuThue  FROM NhanVien nv  JOIN HopDong hd ON nv.MaNV = hd.MaNV  JOIN Thang th ON 1 = 1  LEFT JOIN @ChamCong cc ON nv.MaNV = cc.MaNV AND cc.MaThang = th.MaThang  LEFT JOIN @NguoiPhuThuoc npt ON nv.MaNV = npt.MaNV  LEFT JOIN @PhuCap pc ON nv.MaNV = pc.MaNV AND pc.MaThang = th.MaThang  LEFT JOIN @ThuongPhat tp ON nv.MaNV = tp.MaNV AND tp.MaThang = th.MaThang  LEFT JOIN @BaoHiem bhct ON nv.MaNV = bhct.MaNV;  -- Bảng phụ tính lương thực lãnh  DECLARE @LuongThucLanh TABLE (  MaNV VARCHAR(10),  MaThang VARCHAR(6),  LuongThucTe INT,  ThueThuNhapCaNhan INT,  LuongThucLanh INT  );  INSERT INTO @LuongThucLanh (MaNV, MaThang, LuongThucTe, ThueThuNhapCaNhan, LuongThucLanh)  SELECT  lct.MaNV, lct.MaThang,  ( (lct.LuongCoBan / th.SoNgayCongChuan \* lct.SoNgayLamViec)  + lct.TongPhuCap + lct.TongThuongPhat  ) AS LuongThucTe,  CASE  WHEN lct.ThuNhapChiuThue <= 5000000 THEN lct.ThuNhapChiuThue \* 5 / 100  WHEN lct.ThuNhapChiuThue <= 10000000 THEN lct.ThuNhapChiuThue \* 10 / 100  WHEN lct.ThuNhapChiuThue <= 18000000 THEN lct.ThuNhapChiuThue \* 15 / 100  WHEN lct.ThuNhapChiuThue <= 32000000 THEN lct.ThuNhapChiuThue \* 20 / 100  WHEN lct.ThuNhapChiuThue <= 52000000 THEN lct.ThuNhapChiuThue \* 25 / 100  WHEN lct.ThuNhapChiuThue <= 80000000 THEN lct.ThuNhapChiuThue \* 30 / 100  ELSE lct.ThuNhapChiuThue \* 35 / 100  END AS ThueThuNhapCaNhan,  ( lct.ThuNhapChiuThue - CASE  WHEN lct.ThuNhapChiuThue <= 5000000 THEN lct.ThuNhapChiuThue \* 5 / 100  WHEN lct.ThuNhapChiuThue <= 10000000 THEN lct.ThuNhapChiuThue \* 10 / 100  WHEN lct.ThuNhapChiuThue <= 18000000 THEN lct.ThuNhapChiuThue \* 15 / 100  WHEN lct.ThuNhapChiuThue <= 32000000 THEN lct.ThuNhapChiuThue \* 20 / 100  WHEN lct.ThuNhapChiuThue <= 52000000 THEN lct.ThuNhapChiuThue \* 25 / 100  WHEN lct.ThuNhapChiuThue <= 80000000 THEN lct.ThuNhapChiuThue \* 30 / 100  ELSE lct.ThuNhapChiuThue \* 35 / 100  END  ) AS LuongThucLanh  FROM @LuongChiuThue lct  JOIN Thang th ON lct.MaThang = th.MaThang;  -- Trả về các bảng phụ  SELECT \* FROM @LuongChiuThue;  SELECT \* FROM @LuongThucLanh;  SELECT \* FROM @ChamCong;  SELECT \* FROM @PhuCap;  SELECT \* FROM @ThuongPhat;  SELECT \* FROM @BaoHiem;  END; |

Thực thi trên C#

|  |  |
| --- | --- |
| void load()  {  SqlCommand cmd = new SqlCommand("sp\_TinhLuongChiTietTatCaNhanVien");  DataSet dataSet = new DataSet();  using (SqlConnection connection = new SqlConnection(Connection.connectString))  {  using (SqlCommand command = new SqlCommand("sp\_TinhLuongChiTietTatCaNhanVien", connection))  {  command.CommandType = CommandType.StoredProcedure;  SqlDataAdapter adapter = new SqlDataAdapter(command);  try  {  connection.Open();  adapter.Fill(dataSet);  }  catch (Exception ex)  {  MessageBox.Show("Có lỗi xảy ra: " + ex.Message);  }  }  }  if (dataSet.Tables.Count > 0)  {  dataGridViewLuongChiuThue.DataSource = dataSet.Tables[0];  dataGridViewLuongThucLanh.DataSource = dataSet.Tables[1];  dataGridViewChamCong.DataSource = dataSet.Tables[2];  dataGridViewPhuCap.DataSource = dataSet.Tables[3];  dataGridViewThuongPhat.DataSource = dataSet.Tables[4];  dataGridViewBaoHiem.DataSource = dataSet.Tables[5];  }  else  {  MessageBox.Show("Không có dữ liệu trả về.");  }  } |  |

## 13. Quản lý hợp đồng

### 13.1. Thêm hợp đồng

CREATE OR ALTER PROCEDURE sp\_AddHopDong

@MaHD NVARCHAR(10),

@MaNV NVARCHAR(10),

@LuongCoBan INT,

@NgayBD DATE,

@NgayKT DATE

AS

BEGIN

-- Kiểm tra các tham số có trống hoặc NULL không

IF (@MaHD = '' OR @MaNV = '' OR @LuongCoBan IS NULL OR @NgayBD IS NULL OR @NgayKT IS NULL)

BEGIN

THROW 50000, 'Các thuộc tính bắt buộc không được để trống', 1;

RETURN;

END

-- Chèn dữ liệu vào bảng HopDong

INSERT INTO HopDong (MaHD, MaNV, LuongCoBan, NgayBD, NgayKT)

VALUES (@MaHD, @MaNV, @LuongCoBan, @NgayBD, @NgayKT);

END;

Thực thi trên C#

private void btnThem\_Click\_1(object sender, EventArgs e)

{

try

{

using (SqlConnection conn = Connection.getConnection())

{

conn.Open();

SqlCommand cmd = new SqlCommand("sp\_AddHopDong");

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@MaNV", this.txtMaNhanVien.Text);

cmd.Parameters.AddWithValue("@MaHD", this.txtMaHopDong.Text);

cmd.Parameters.AddWithValue("@LuongCoBan", this.txtLuongCoBan.Text);

cmd.Parameters.AddWithValue("@NgayBD", this.dtpNgayBd.Value);

cmd.Parameters.AddWithValue("@NgayKT", this.dtpNgayKt.Value);

cmd.Connection = conn;

cmd.ExecuteNonQuery();

MessageBox.Show("Thêm thành công");

load();

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

### 13.2. Sửa hợp đồng

CREATE OR ALTER PROCEDURE sp\_UpdateHopDong

@MaHD nvarchar(10),

@NgayBD DATE,

@NgayKT DATE,

@LuongCoBan int

AS

BEGIN

IF @NgayBD IS NULL OR @NgayBD = '' OR @NgayKT IS NULL OR @NgayKT = ''

BEGIN

RAISERROR('Vui lòng điền đầy đủ thông tin cho tất cả các trường', 16, 1)

RETURN;

END

ELSE

UPDATE HopDong

SET NgayBD = @NgayBD, NgayKT = @NgayKT, LuongCoBan = @LuongCoBan

WHERE MaHD = @MaHD;

END

Thực thi trên C#

private void btnSua\_Click\_1(object sender, EventArgs e)

{

try

{

using (SqlCommand command = new SqlCommand("sp\_UpdateHopDong"))

{

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@MaHD", this.txtMaHopDong.Text);

command.Parameters.AddWithValue("@NgayBD", this.dtpNgayBd.Value);

command.Parameters.AddWithValue("@NgayKT", this.dtpNgayKt.Value);

command.Parameters.AddWithValue("@LuongCoBan", this.txtLuongCoBan.Text);

Connection.ExecuteCommand(command);

MessageBox.Show("Cập nhật thành công");

load();

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

### 13.3. Xóa hợp đồng

CREATE OR ALTER PROCEDURE sp\_DeleteHopDong

@MaHD nvarchar(10)

AS

BEGIN

DELETE FROM HopDong

WHERE MaHD = @MaHD;

END;

Thực thi trên C#

private void btnXoa\_Click\_1(object sender, EventArgs e)

{

try

{

using (SqlCommand command = new SqlCommand("sp\_DeleteHopDong"))

{

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@MaHD", this.txtMaHopDong.Text);

Connection.ExecuteCommand(command);

MessageBox.Show("Xoá thành công");

load();

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

## 14. Quản lý người phụ thuộc

### 14.1. Thêm người phụ thuộc

CREATE OR ALTER PROCEDURE sp\_AddNguoiPhuThuoc

@MaNV NVARCHAR(10),

@HoTen NVARCHAR(50),

@NgaySinh DATE,

@QuanHe NVARCHAR(50)

AS

BEGIN

-- Thêm thông tin người phụ thuộc mới

INSERT INTO NguoiPhuThuoc (MaNV, HoTen, NgaySinh, QuanHe)

VALUES (@MaNV, @HoTen, @NgaySinh, @QuanHe);

END;

GO

Thực thi trên C#

private void btnThem\_Click(object sender, EventArgs e)

{

try

{

using (SqlCommand command = new SqlCommand("sp\_AddNguoiPhuThuoc"))

{

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@MaNV", this.cbbMaNV.Text);

command.Parameters.AddWithValue("@HoTen", this.txtHoTen.Text);

command.Parameters.AddWithValue("@NgaySinh", this.dateTimePickerNgaySinh.Text);

command.Parameters.AddWithValue("@QuanHe", this.cbbQuanHe.Text);

Connection.ExecuteCommand(command);

MessageBox.Show("Thêm thành công");

load();

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

### 14.2. Xóa người phụ thuộc

CREATE OR ALTER PROCEDURE sp\_DeleteNguoiPhuThuoc

@MaNV NVARCHAR(10),

@HoTen NVARCHAR(50)

AS

BEGIN

DELETE FROM NguoiPhuThuoc

WHERE MaNV = @MaNV AND HoTen = @HoTen;

END;

GO

Thực thi trên C#

private void btnXoa\_Click(object sender, EventArgs e)

{

try

{

using (SqlCommand command = new SqlCommand("sp\_DeleteNguoiPhuThuoc"))

{

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@MaNV", this.cbbMaNV.Text);

command.Parameters.AddWithValue("@HoTen", this.txtHoTen.Text);

Connection.ExecuteCommand(command);

MessageBox.Show("Cập nhật thành công");

load();

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

# **CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN**

Hệ thống có ba nhóm người dùng là: Nhân viên, Trưởng phòng và Quản lý

## 1. Tạo Role và phân quyền

### 1.1. Nhân viên (Employee)

CREATE ROLE Employee

GO

GRANT SELECT, REFERENCES ON NhanVien TO Employee

GRANT SELECT, REFERENCES ON ctChamCong TO Employee

GRANT SELECT, REFERENCES ON ctBaoHiem TO Employee

GRANT SELECT, INSERT, REFERENCES ON NghiPhep TO Employee

GRANT SELECT, UPDATE, REFERENCES ON TaiKhoan TO Employee

GRANT SELECT, INSERT, UPDATE, DELETE, REFERENCES ON ThongBao TO Employee

GRANT SELECT ON vw\_ThongTinNhanVien TOEmployee

GRANT SELECT ON vw\_ThongTinHopDong TO Employee

DENY SELECT ON ft\_TruongPhongNhanThongBao TO Employee;

GRANT SELECT ON ft\_NhanVienNhanThongBao TO Employee;

GRANT EXECUTE ON sp\_GetNghiPhepByMaNV TO Employee;

GRANT EXECUTE ON sp\_GetChamCongByMaNV TO Employee;

GRANT EXECUTE ON sp\_AddctChamCong TO Employee;

GRANT EXECUTE ON sp\_TinhLuongTheoThang TO Employee;

GRANT EXECUTE ON sp\_GetThang TO Employee;

### 1.2. Trưởng phòng (DepartmentHead)

CREATE ROLE DepartmentHead

GO

GRANT SELECT, REFERENCES ON NhanVien TO DepartmentHead

GRANT SELECT, REFERENCES ON ctChamCong TO DepartmentHead

GRANT SELECT, REFERENCES ON ctBaoHiem TO DepartmentHead

GRANT SELECT, INSERT, REFERENCES ON NghiPhep TO DepartmentHead

GRANT SELECT, UPDATE, REFERENCES ON TaiKhoan TO DepartmentHead

GRANT SELECT, INSERT, UPDATE, DELETE, REFERENCES ON ThongBao TO DepartmentHead

GRANT SELECT ON vw\_ThongTinNhanVien to DepartmentHead

GRANT SELECT ON vw\_ThongTinHopDong to DepartmentHead

GRANT SELECT ON ft\_TruongPhongNhanThongBao to DepartmentHead;

GRANT SELECT ON ft\_NhanVienNhanThongBao TO DepartmentHead;

GRANT EXECUTE ON sp\_GetNghiPhepByMaNV to DepartmentHead;

GRANT EXECUTE ON sp\_GetChamCongByMaNV to DepartmentHead;

GRANT EXECUTE ON sp\_AddctChamCong to DepartmentHead;

GRANT EXECUTE ON sp\_TinhLuongTheoThang to DepartmentHead;

GRANT EXECUTE ON sp\_GetThang to DepartmentHead;

### 1.3. Quản lý (Admin)

Đối với quản lý ta sử dụng server role sysadmin.

## 2. Tạo LOGIN và USER mẫu cho từng ROLE

-- Quan ly

CREATE LOGIN NV01 WITH PASSWORD = 'password1', DEFAULT\_DATABASE = [QLNSG21], CHECK\_EXPIRATION = OFF, CHECK\_POLICY = OFF;

CREATE USER NV01 FOR LOGIN NV01;

ALTER SERVER ROLE sysadmin ADD MEMBER NV01

-- Truong Phong

CREATE LOGIN NV02 WITH PASSWORD = 'password2', DEFAULT\_DATABASE = [QLNSG21], CHECK\_EXPIRATION = OFF, CHECK\_POLICY = OFF;

CREATE USER NV02 FOR LOGIN NV02;

ALTER ROLE DepartmentHead ADD MEMBER NV02

-- NhanVien

CREATE LOGIN NV03 WITH PASSWORD = 'password3', DEFAULT\_DATABASE = [QLNSG21], CHECK\_EXPIRATION = OFF, CHECK\_POLICY = OFF;

CREATE USER NV03 FOR LOGIN NV03;

ALTER ROLE Employee ADD MEMBER NV03

## 3. Viết Trigger tự động phân quyền khi thêm nhân viên mới

Để phân quyền cho nhân viên mới, khi quản lý tạo nhân viên, trigger tr\_NhanVien\_ThemTaiKhoan sẽ được kích hoạt để tự động thêm tài khoản đăng nhập hệ thống cho nhân viên mới tương ứng với chức vụ của họ. Sau đó trigger tr\_TaiKhoan\_ThemQuyen được kích hoạt để phân quyền cho nhân viên đó.

Bên cạnh đó, để mật khẩu tài khoản đăng nhập hệ thống Quản lý nhân sự đồng bộ với mật khẩu user đăng nhập vào cơ sở dữ liệu thì cần có trigger tr\_TaiKhoan\_UpdateMatKhauLogin. Và khi xóa nhân viên trên hệ thống, Login và User của nhân viên đó cũng sẽ bị xóa để đảm bảo hệ thống không bị rò rỉ thông tin cho đối tượng không có quyền.

### 3.1. Trigger tạo Nhân viên thì tạo Tài khoản

CREATE OR ALTER TRIGGER tr\_NhanVien\_ThemTaiKhoan

ON NhanVien

AFTER INSERT

AS

BEGIN

INSERT INTO TaiKhoan(TenDangNhap, MatKhau, MaLoai)

SELECT

inserted.MaNV,'123',

CASE

WHEN NhanVien.MaCV = 'CV01' THEN 'LTK01'

WHEN NhanVien.MaCV = 'CV02' THEN 'LTK02'

ELSE 'LTK03'

END AS MaLoai

FROM inserted

JOIN NhanVien ON inserted.MaNV = NhanVien.MaNV;

END;

GO

### 3.2. Trigger thêm tài khoản thì thêm phân quyền

CREATE OR ALTER TRIGGER tr\_TaiKhoan\_ThemQuyen

ON TaiKhoan

AFTER INSERT

AS

BEGIN

DECLARE @tendangnhap NVARCHAR(10),

@matkhau NVARCHAR(255),

@maloai NVARCHAR(10),

@sqlString NVARCHAR(2000);

SELECT @tendangnhap = TenDangNhap,

@matkhau = MatKhau,

@maloai = MaLoai

FROM inserted;

BEGIN TRY

-- Create Login

SET @sqlString = 'CREATE LOGIN ' + QUOTENAME(@tendangnhap) +

' WITH PASSWORD = ''' + @matkhau + ''', DEFAULT\_DATABASE = [QLNSG21], CHECK\_EXPIRATION = OFF, CHECK\_POLICY = OFF';

EXEC sp\_executesql @sqlString;

-- Create User

SET @sqlString = 'CREATE USER ' + QUOTENAME(@tendangnhap) + ' FOR LOGIN ' + QUOTENAME(@tendangnhap);

EXEC sp\_executesql @sqlString;

-- Assign Role

IF @maloai = 'LTK01'

BEGIN

SET @sqlString = ALTER SERVER ROLE sysadmin ADD MEMBER ' + QUOTENAME(@tendangnhap);

EXEC sp\_executesql @sqlString;

END

IF @maloai = 'LTK02'

BEGIN

SET @sqlString = 'ALTER ROLE DepartmentHead ADD MEMBER ' + QUOTENAME(@tendangnhap);

EXEC sp\_executesql @sqlString;

END

ELSE

BEGIN

SET @sqlString = 'ALTER ROLE Employee ADD MEMBER ' + QUOTENAME(@tendangnhap);

EXEC sp\_executesql @sqlString;

END

END TRY

BEGIN CATCH

PRINT 'Loi khi them quyen cho tai khoan';

END CATCH

END;

GO

### 3.3. Trigger đồng bộ mật khẩu đăng nhập

CREATE OR ALTER TRIGGER tr\_TaiKhoan\_UpdateMatKhauLogin

ON TaiKhoan

AFTER UPDATE

AS

BEGIN

SET NOCOUNT ON;

IF UPDATE(MatKhau)

BEGIN

DECLARE @TenDangNhap nvarchar(10);

DECLARE @MatKhau nvarchar(255);

SELECT @TenDangNhap = i.TenDangNhap, @MatKhau = i.MatKhau

FROM inserted i;

DECLARE @sqlString nvarchar(2000);

SET @sqlString = 'ALTER LOGIN [' + @TenDangNhap + '] WITH PASSWORD=''' + @MatKhau + '''';

EXEC (@sqlString);

END

END;

GO

### 3.4. Thủ tục xóa nhân viên

CREATE OR ALTER PROCEDURE sp\_DeleteNhanVien

@MaNV nvarchar(10)

AS

BEGIN

SET NOCOUNT ON;

DECLARE @TenTaiKhoan varchar(10);

SELECT @TenTaiKhoan=TenDangNhap FROM TaiKhoan WHERE TenDangNhap=@MaNV

DECLARE @sql varchar(100)

DECLARE @SessionID INT;

SELECT @SessionID = session\_id

FROM sys.dm\_exec\_sessions

WHERE login\_name = @TenTaiKhoan;

IF @SessionID IS NOT NULL

BEGIN

SET @sql = 'kill ' + Convert(NVARCHAR(20), @SessionID)

exec(@sql)

END

BEGIN TRANSACTION;

BEGIN TRY

DELETE FROM NhanVien WHERE MaNV = @MaNV;

--

SET @sql = 'DROP USER '+ @TenTaiKhoan

exec (@sql)

--

SET @sql = 'DROP LOGIN '+ @TenTaiKhoan

exec (@sql)

END TRY

BEGIN CATCH

DECLARE @err NVARCHAR(MAX)

SELECT @err = N'Lỗi ' + ERROR\_MESSAGE()

RAISERROR(@err, 16, 1)

ROLLBACK TRANSACTION;

THROW;

END CATCH

COMMIT TRANSACTION;

END;

GO

## 4. Kiểm tra

Hai nhóm người dùng quản lý và nhân viên đã được phân quyền trên giao diện hệ thống, nghĩa là các nút chức năng riêng biệt. Riêng nhân viên và trưởng phòng có cùng giao diện nên ta thực hiện kiểm tra với chức năng xem thông báo mà trưởng phòng có quyền xem nhưng nhân viên thì không.

- Trưởng phòng có thể xem thông báo của tất cả các phòng ban do có quyền trên thủ tục xem thông báo phòng ban: ft\_TruongPhongNhanThongBao() và thông báo phòng ban của trưởng phòng đó:

Thông báo của tất cả phòng ban:



Thông báo theo phòng ban:



- Nhân viên không thể xem thông báo của tất cả phòng ban do không được cấp quyền trên thủ tục xem thông báo: ft\_TruongPhongNhanThongBao()

****

Nhưng nhân viên vẫn có thể nhận thông báo của phòng ban mình làm việc:



# **CHƯƠNG 5: THIẾT KẾ GIAO DIỆN HỆ THỐNG**

Môi trường cài đặt và công nghệ sử dụng

- Hệ cơ sở dữ liệu Microsoft SQL Server Management Studio 20.

- Chương trình Winform được viết bằng Microsoft Visual Studio 2022 - trên nền tảng

ASP.NET Web Application (.NET Framework) C#.

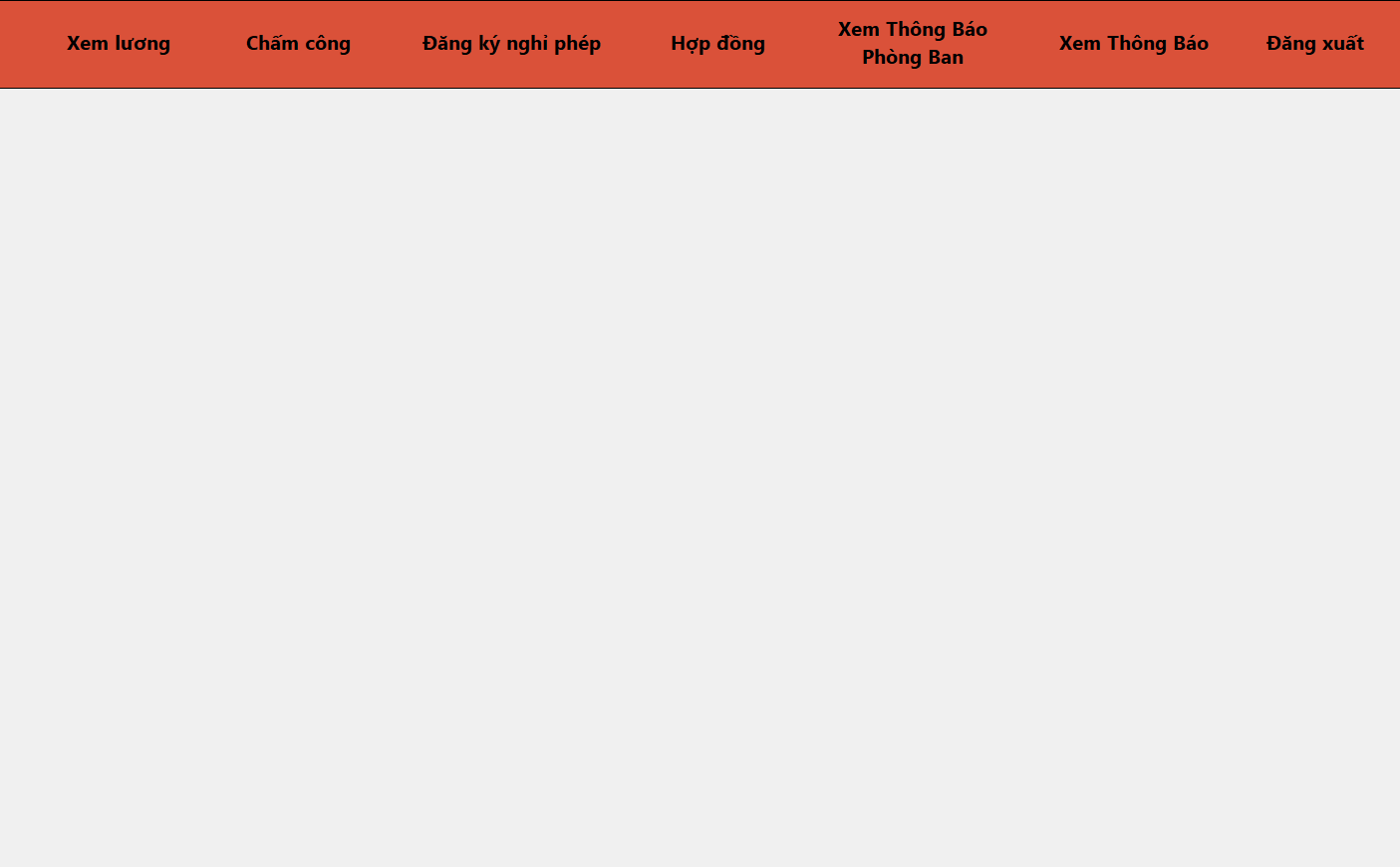
Giao diện của chương trình:

## 1. Form đăng nhập

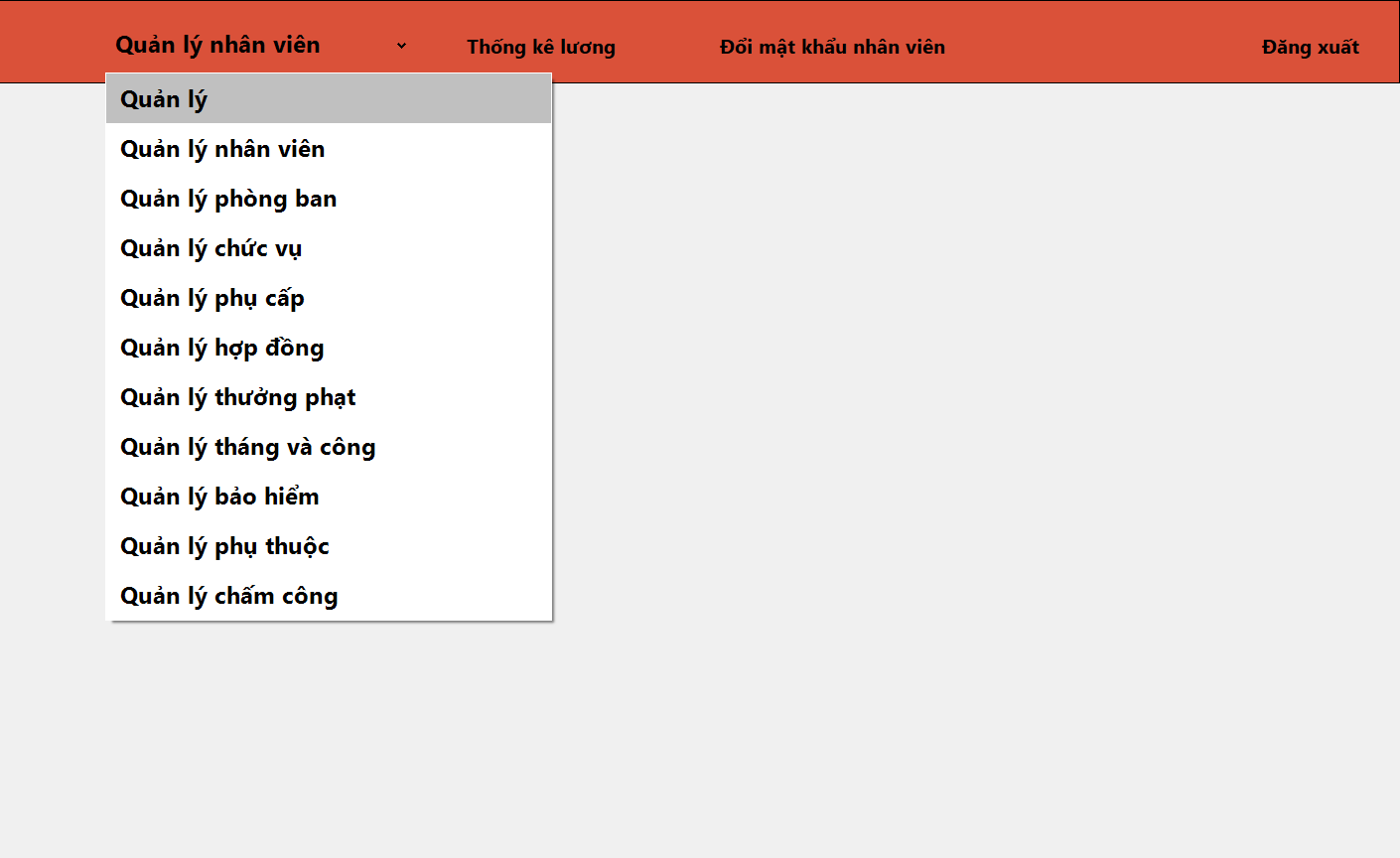


## 2. Form chính

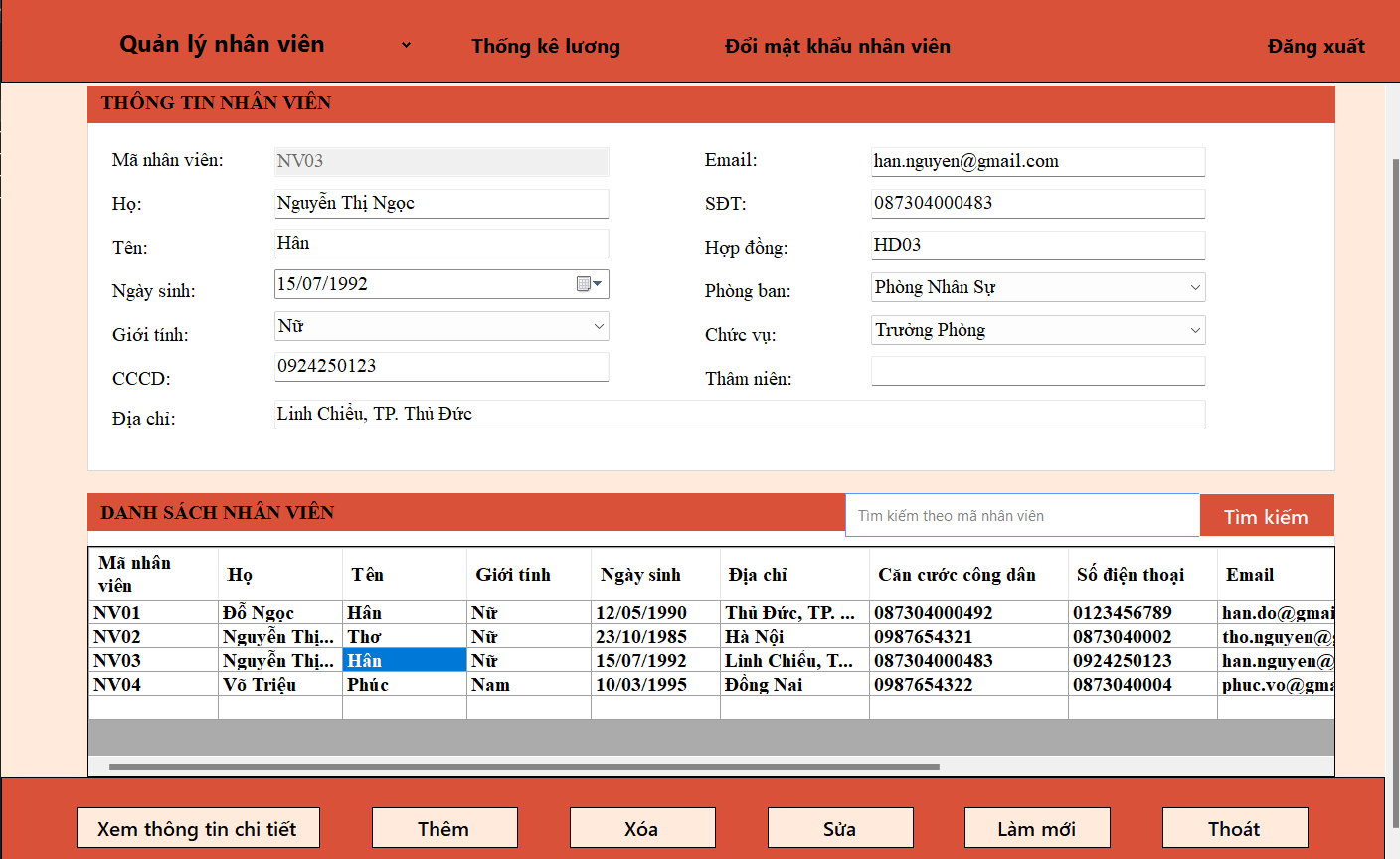
- Đối với Nhân viên và Trưởng phòng:



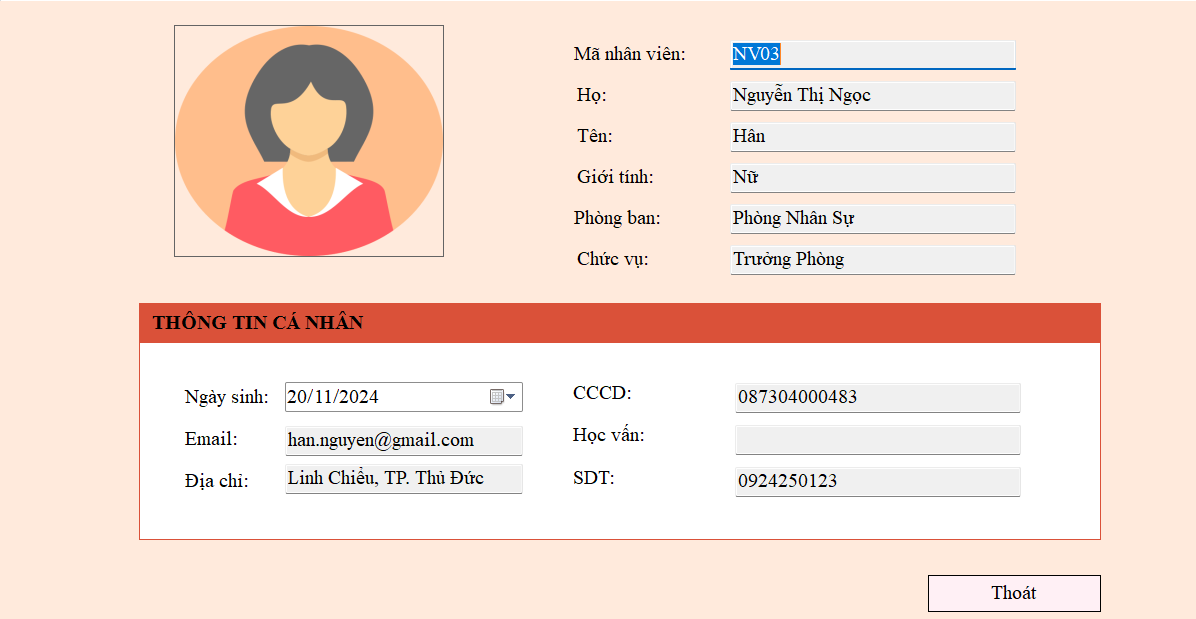
- Đối với Quản lý:



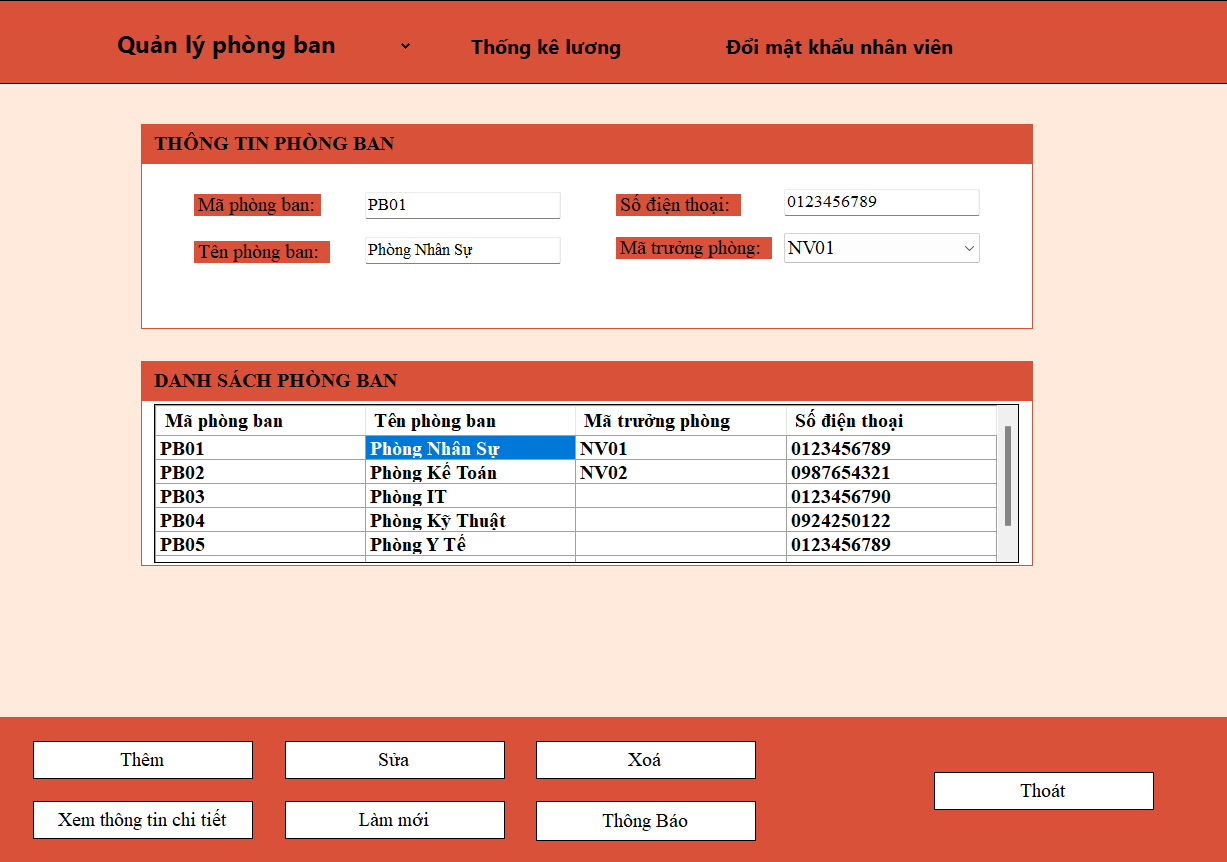
## 3. Quản lý nhân viên



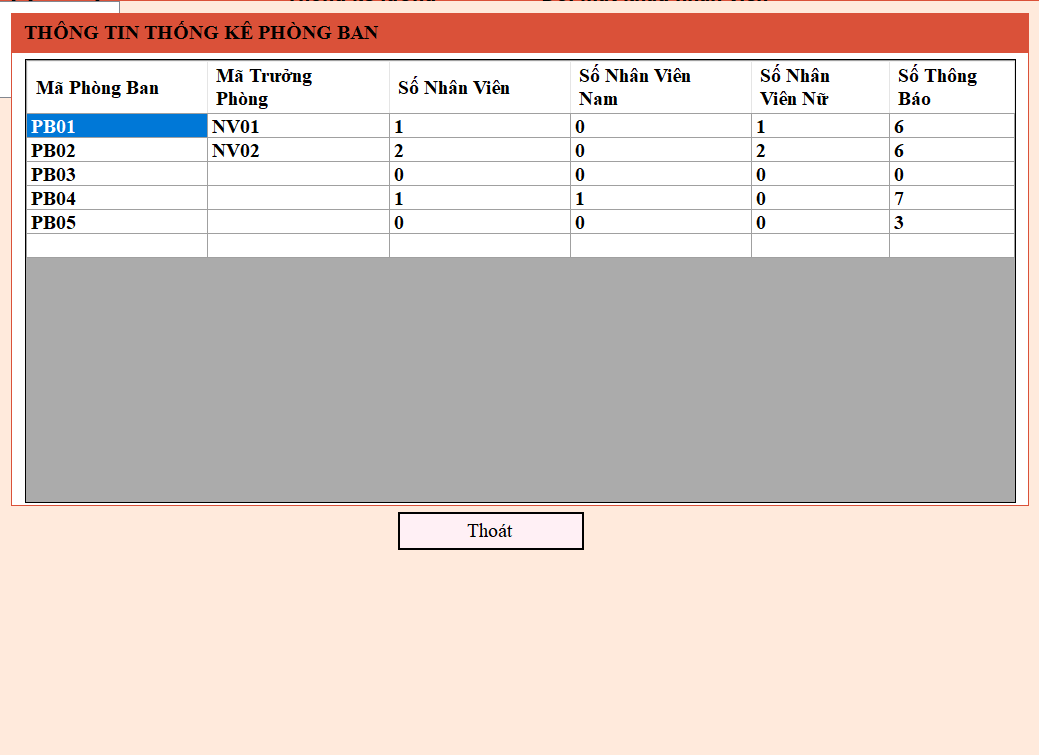
Xem thông tin chi tiết của nhân viên:



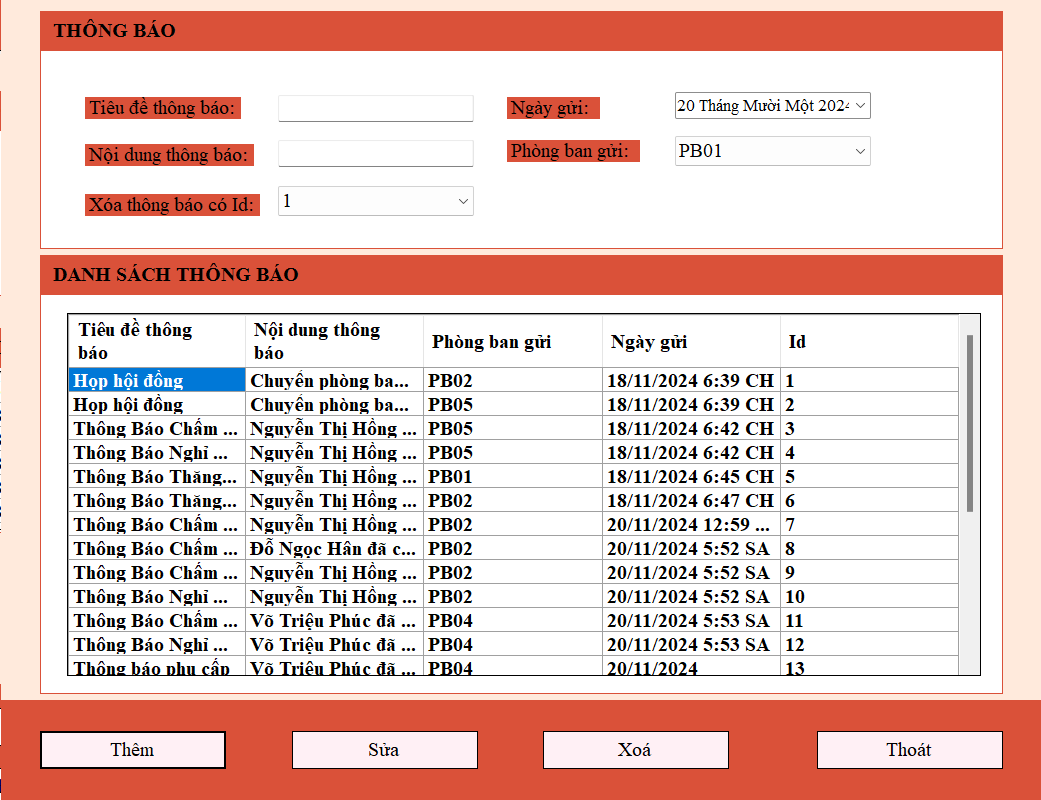
## 4. Quản lý phòng ban



### 4.1. Xem thông tin chi tiết thống kê về các phòng ban:

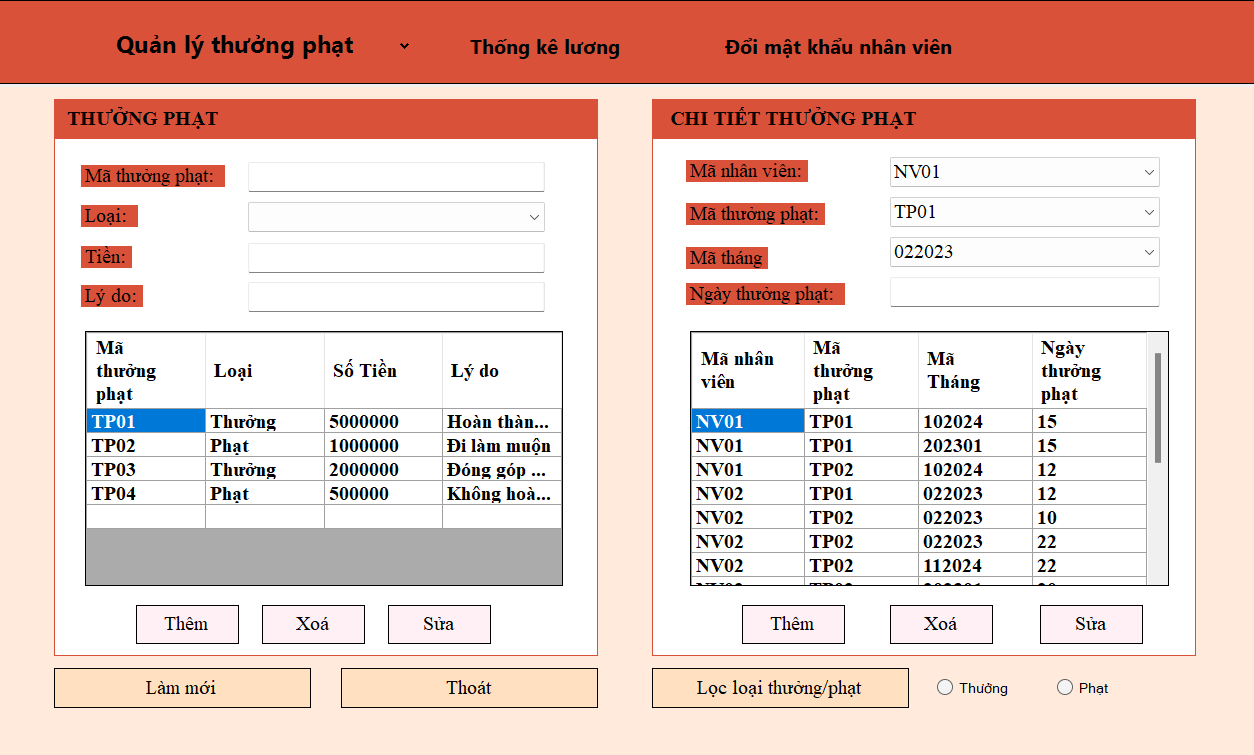


### 4.2. Form thông báo đến nhân viên của phòng ban

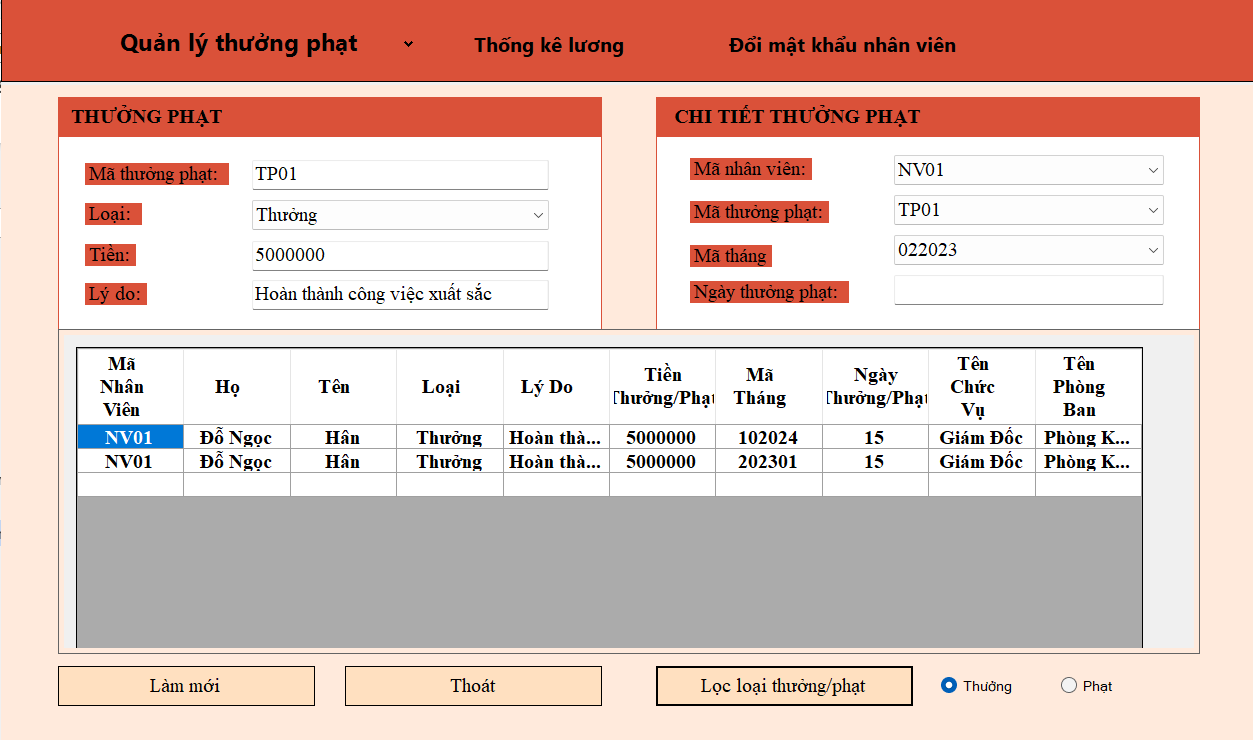


## 5. Quản lý thưởng phạt

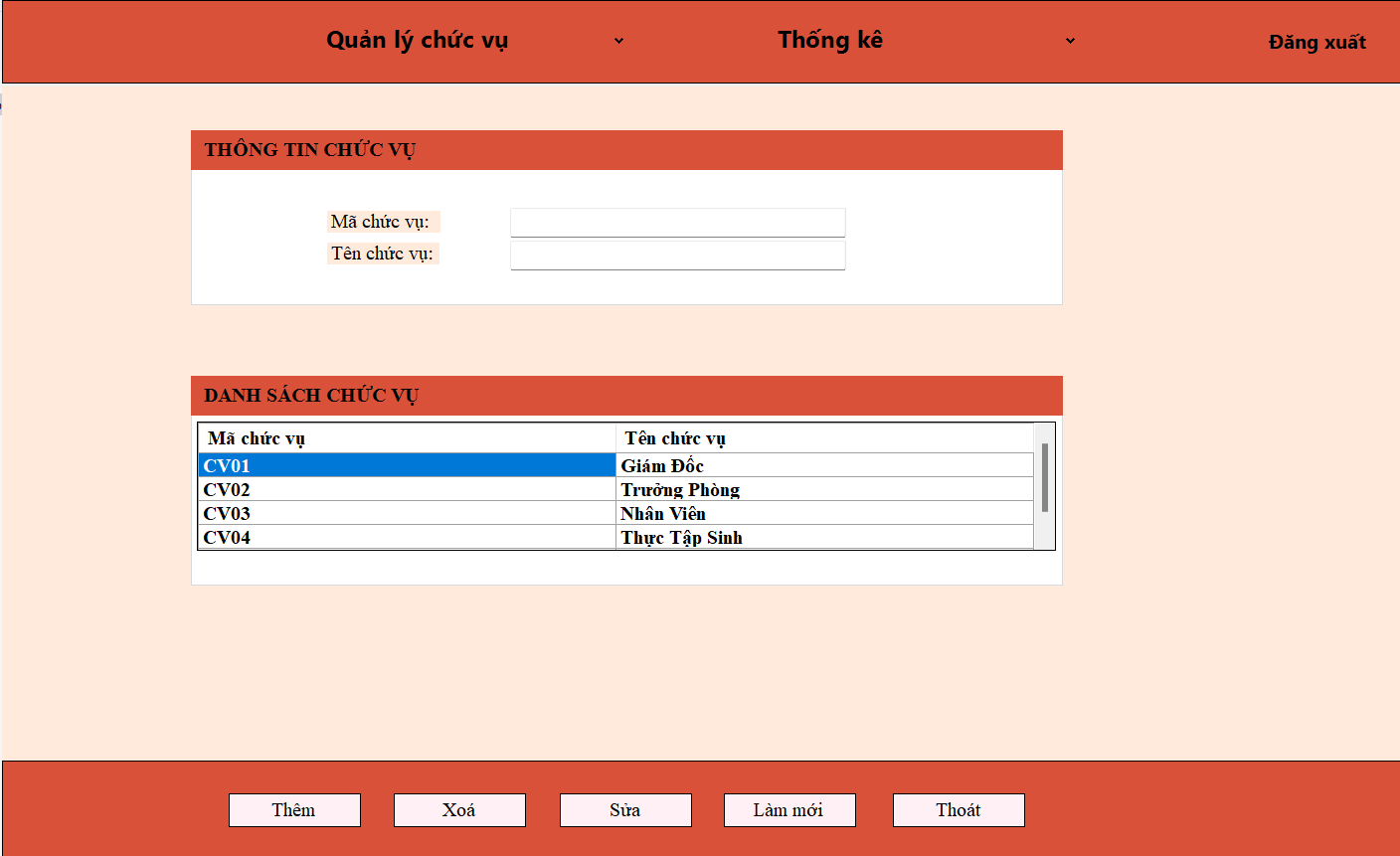
### 5.1. Form thưởng phạt chính



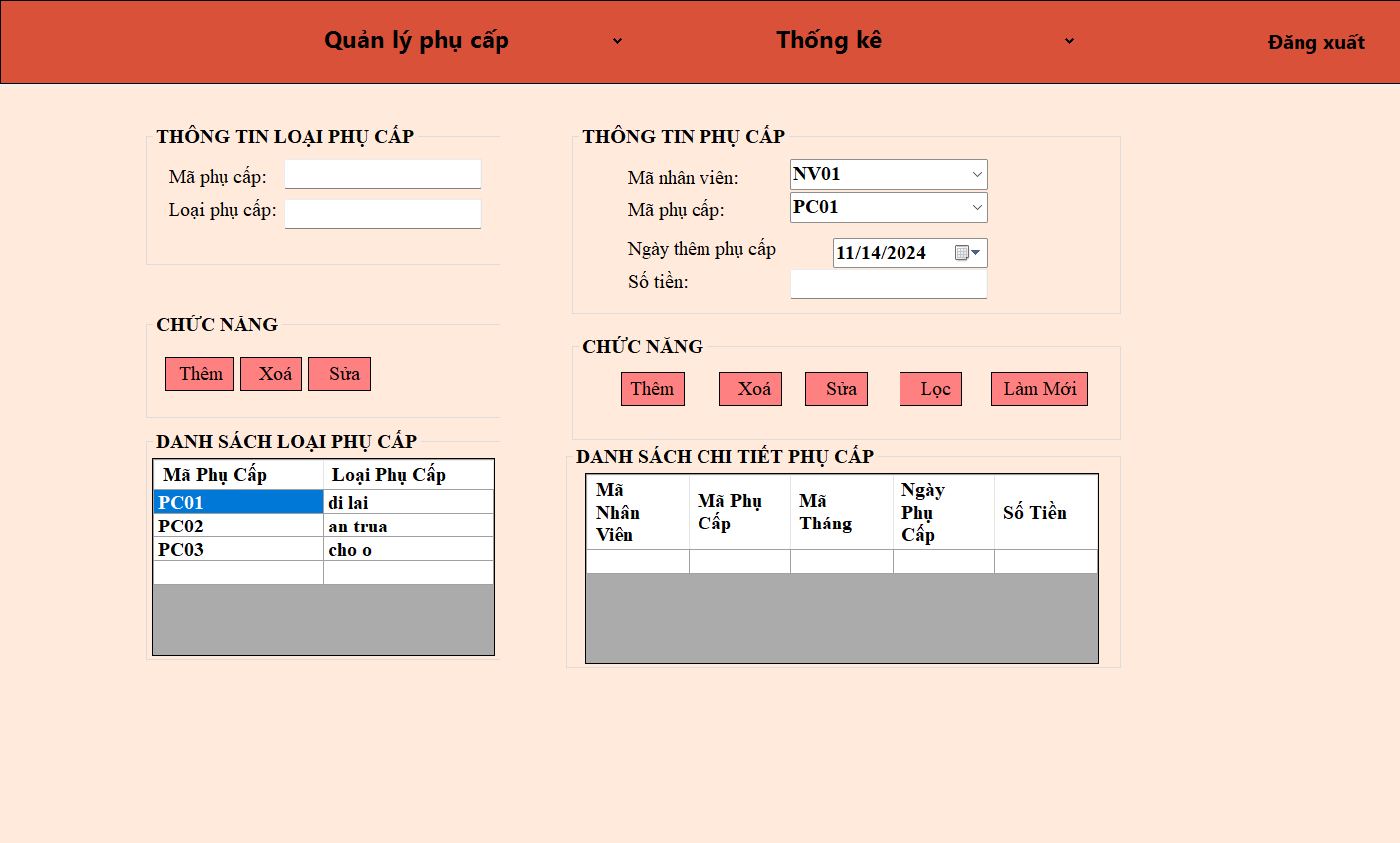
### 5.2. Form thưởng phạt sau khi lọc loại thưởng hoặc phạt theo Mã nhân viên



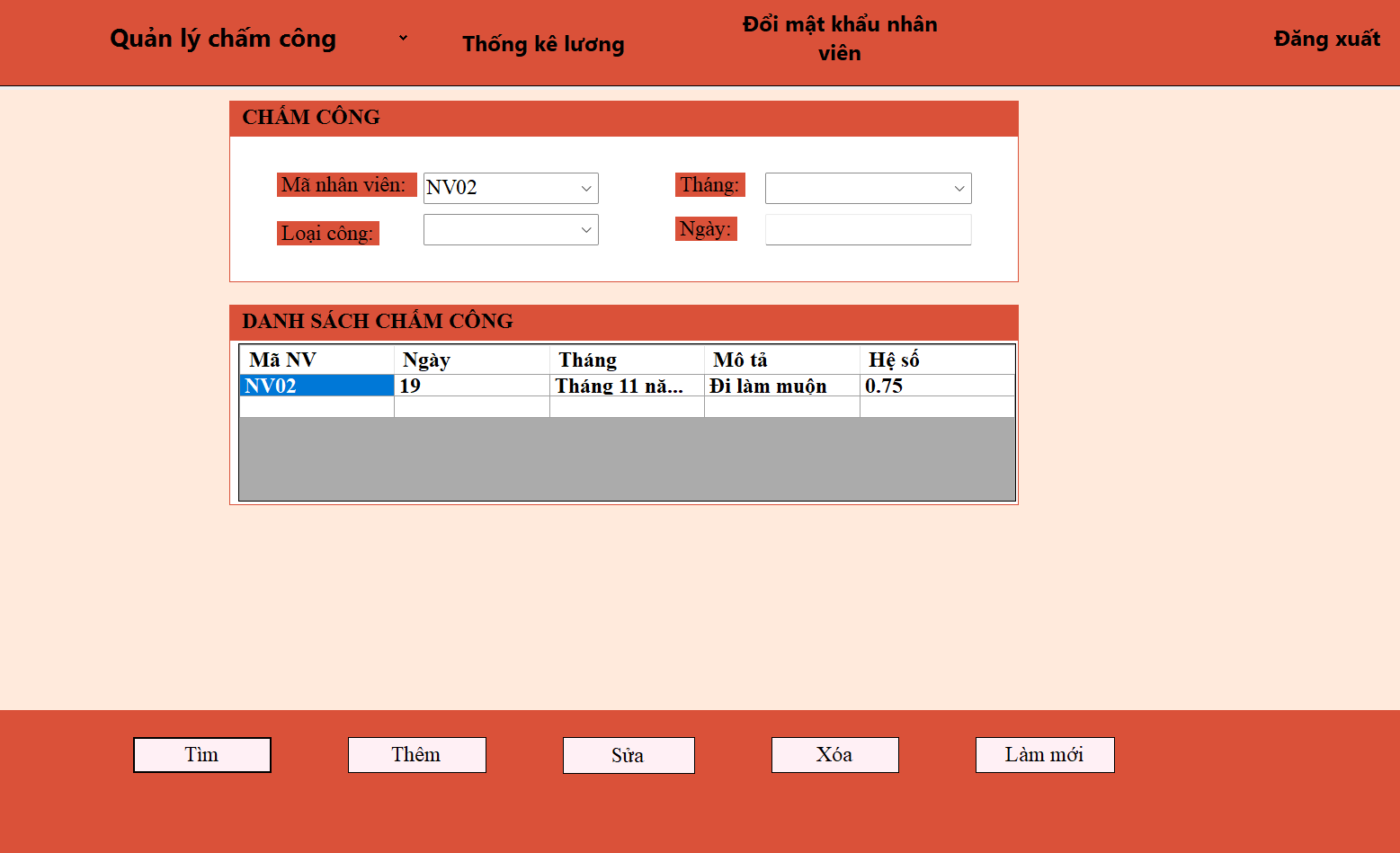
## 6. Quản lý chức vụ

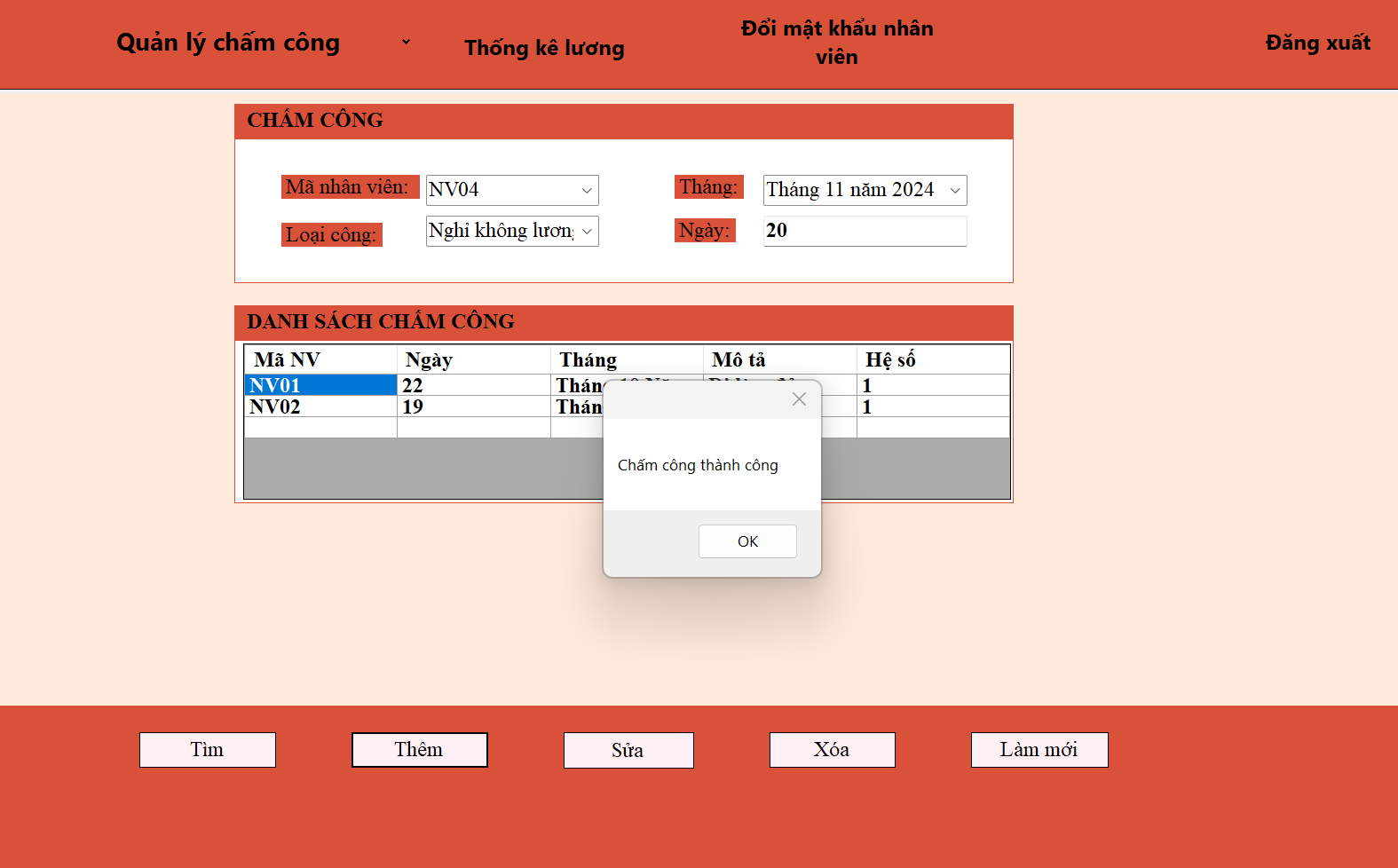


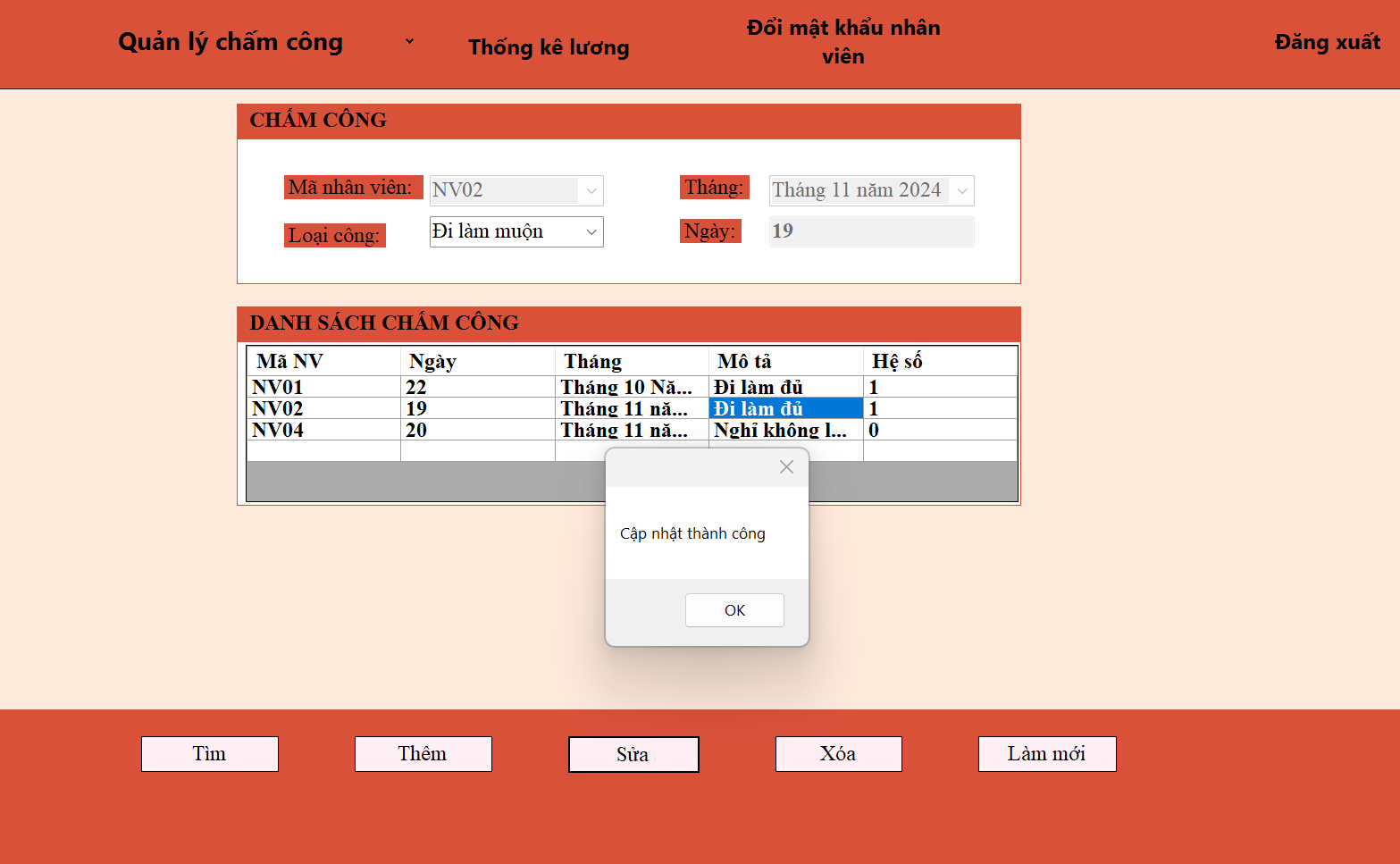
## 7. Quản lý phụ cấp

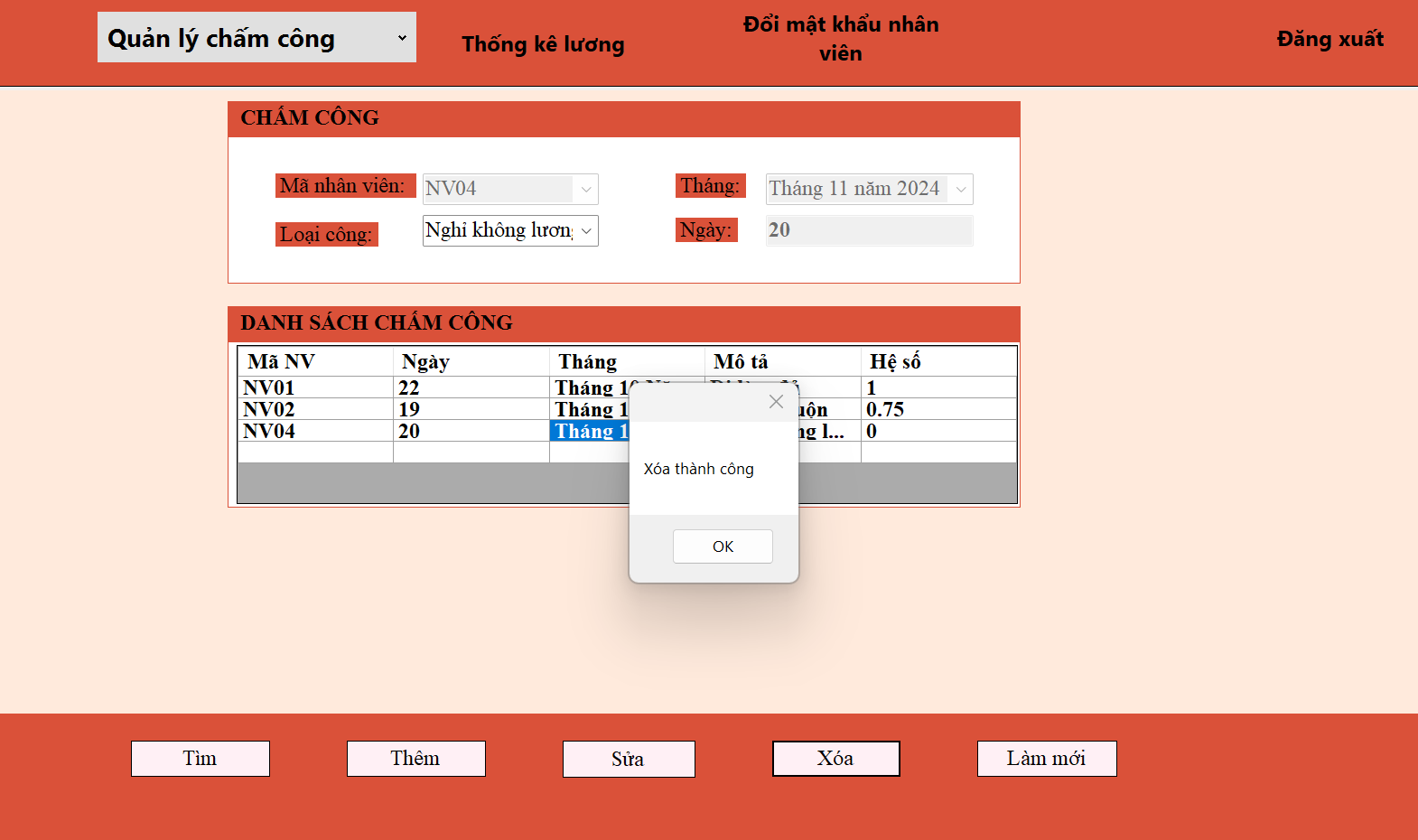


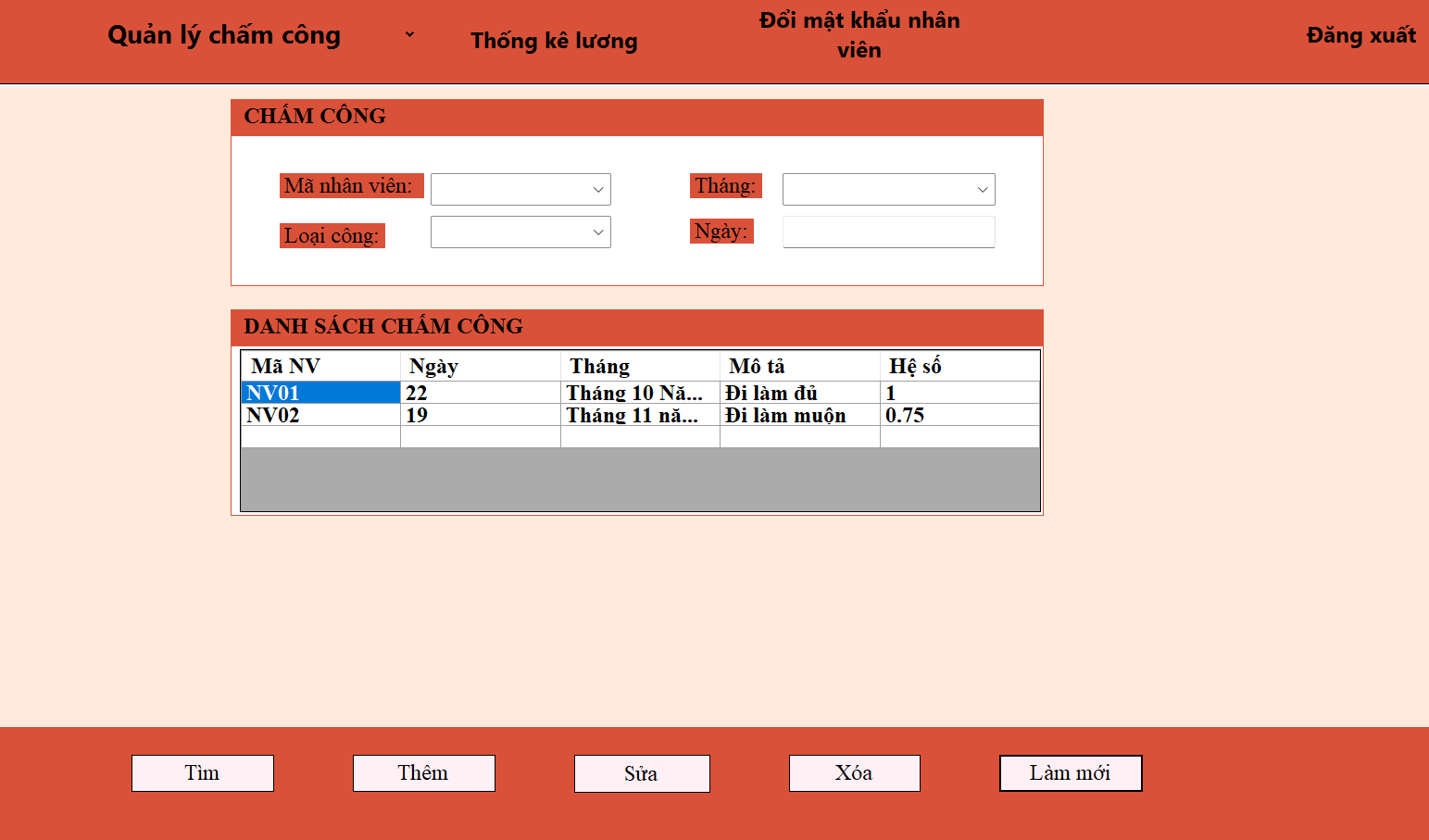
## 8. Quản lý chấm công



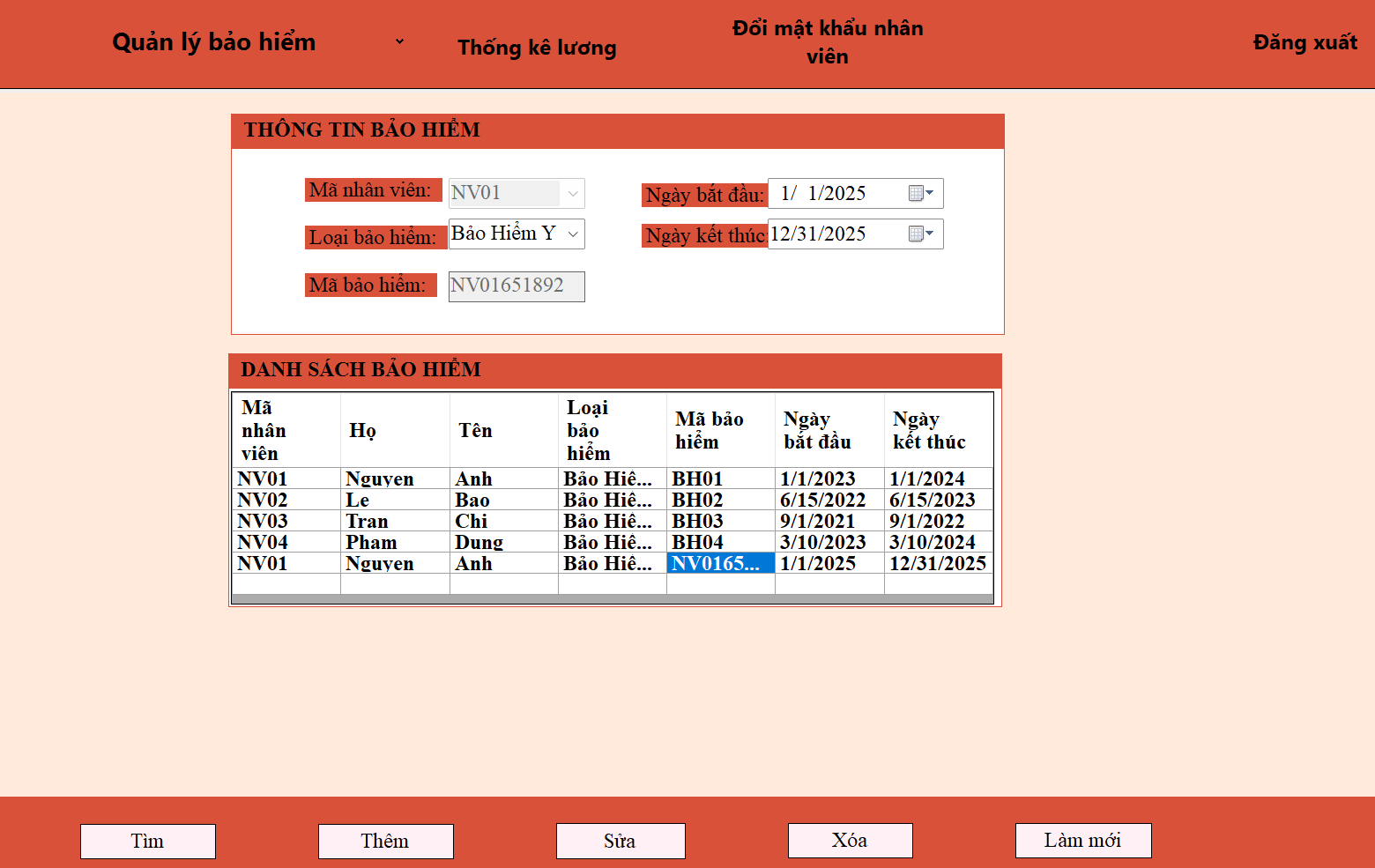


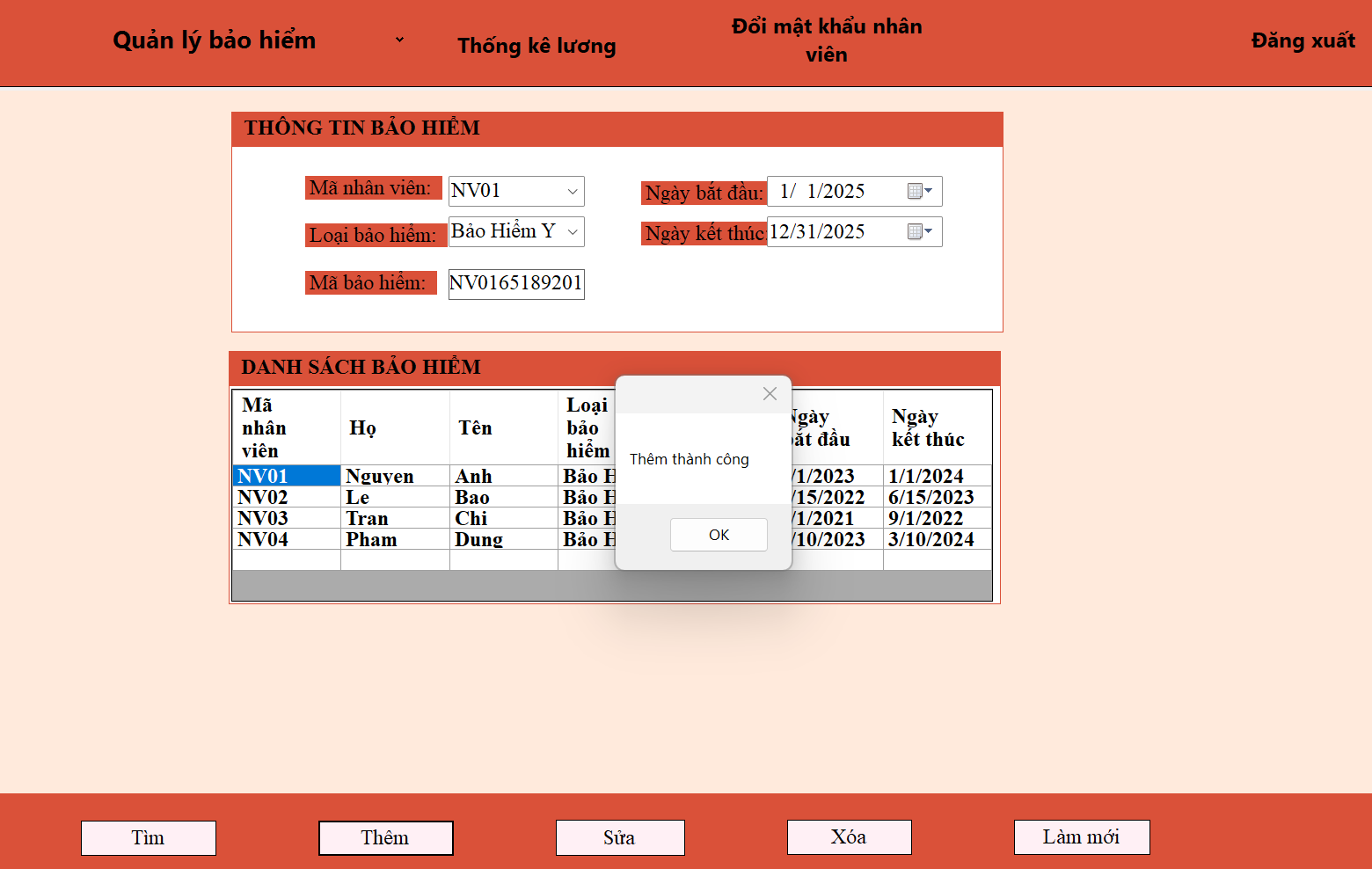




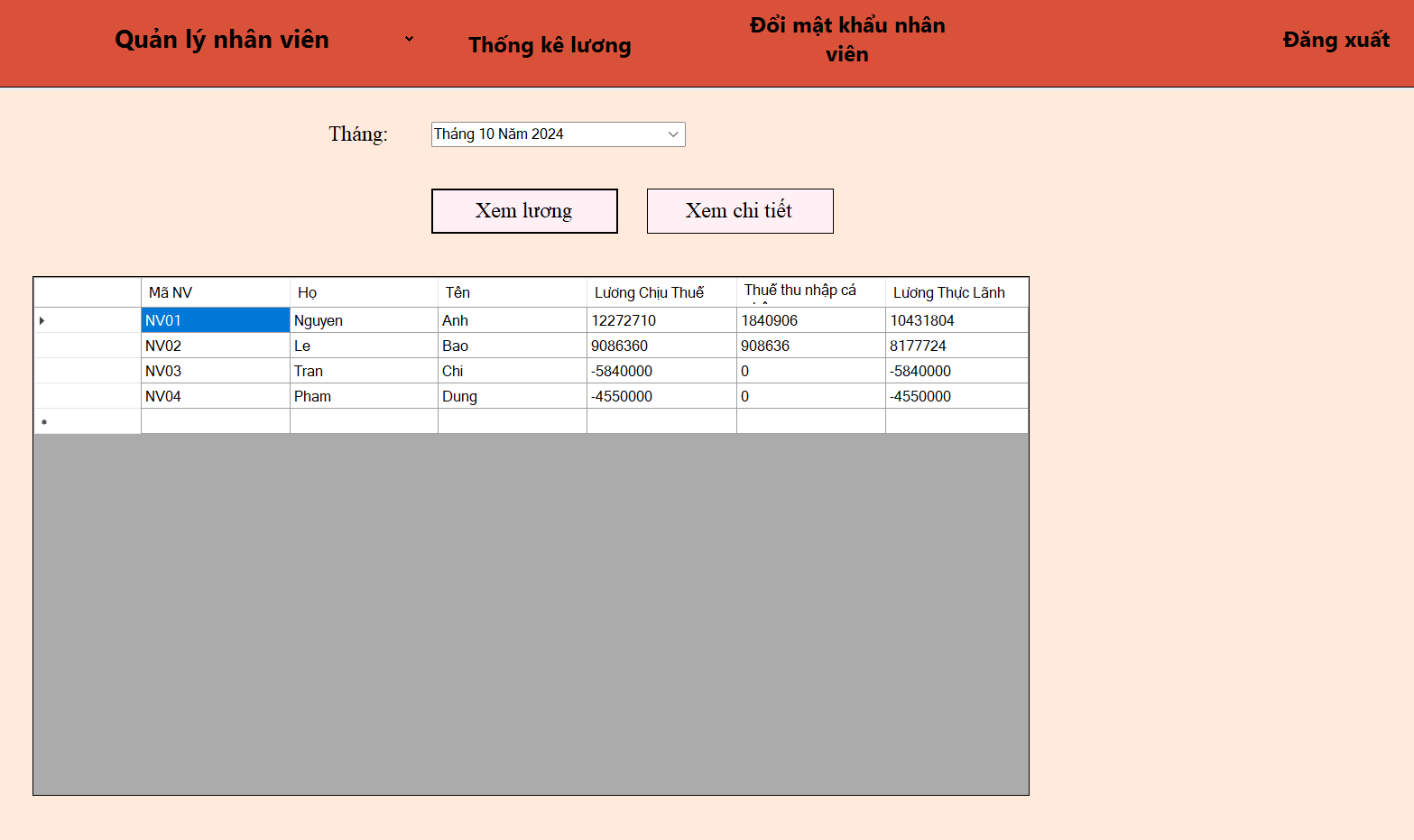


## 9. Quản lý bảo hiểm

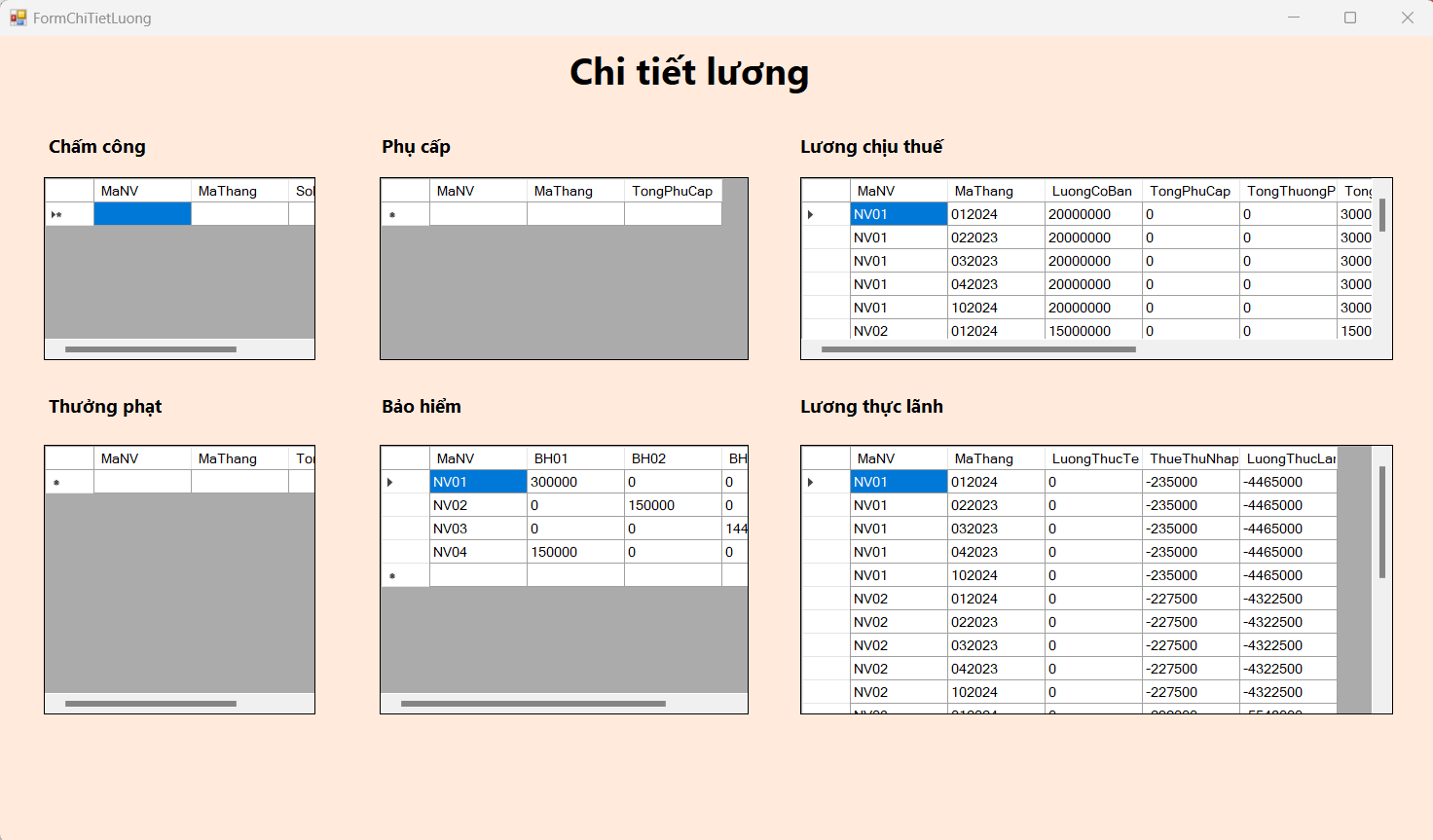




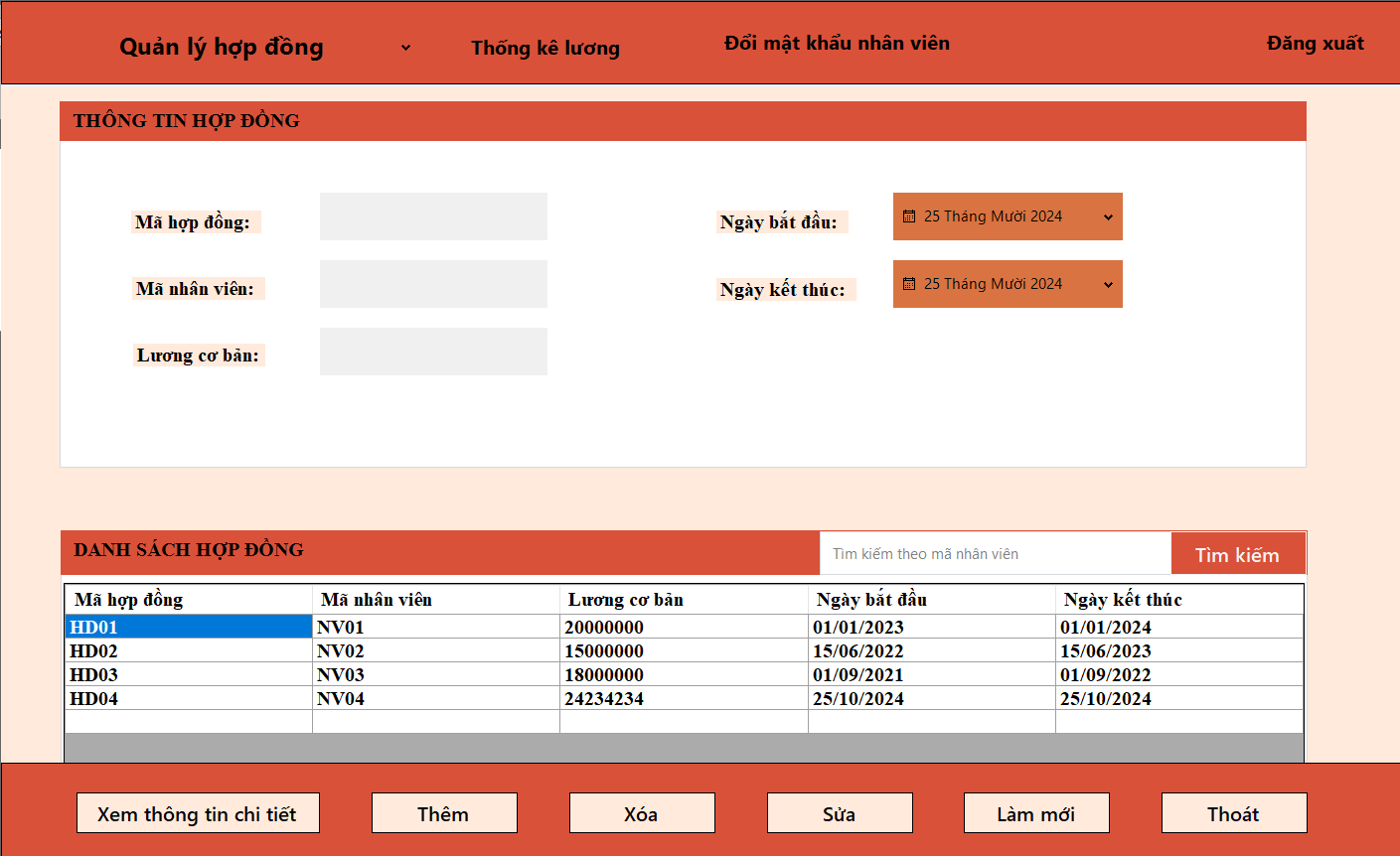
## 10. Quản lý lương nhân viên



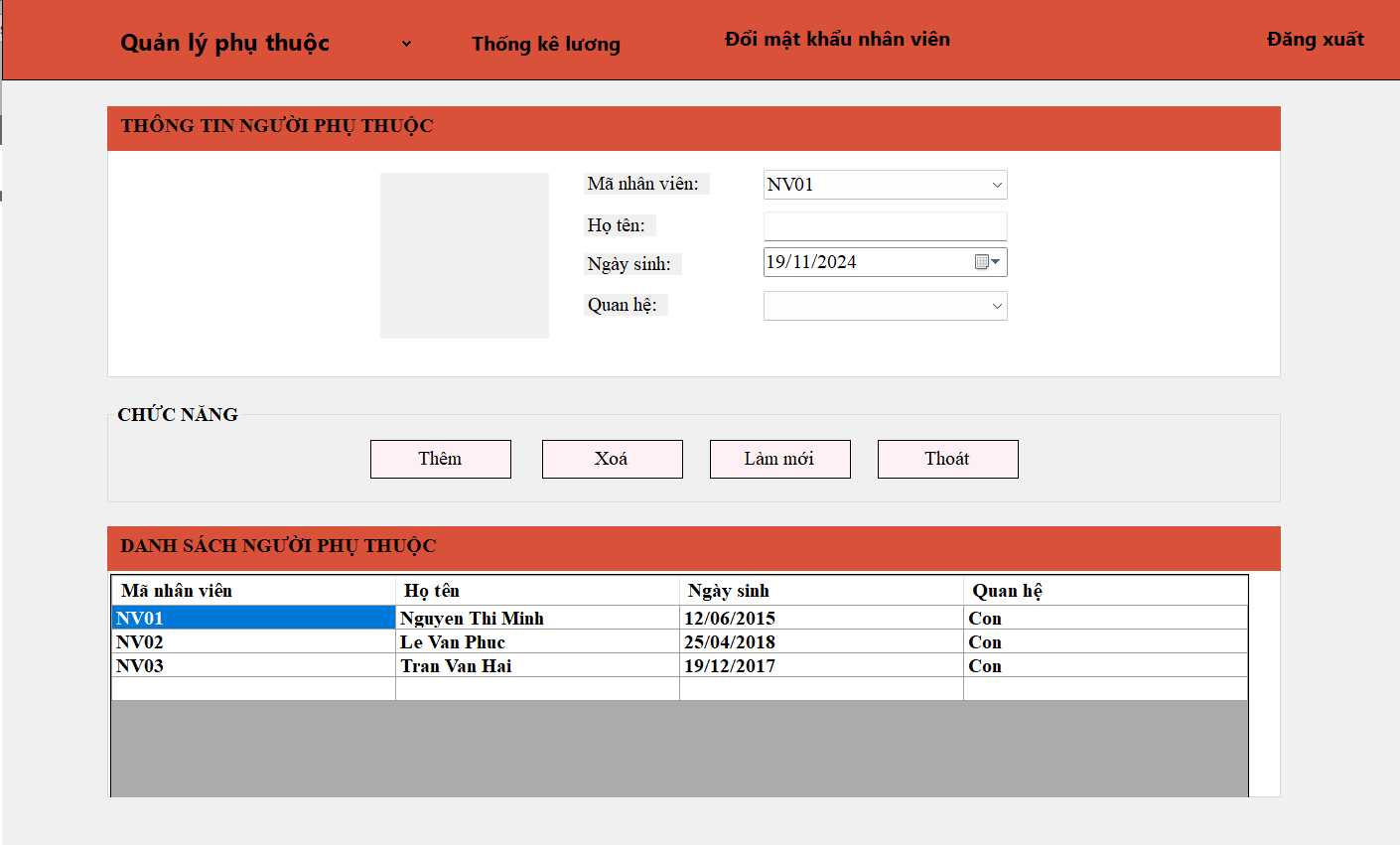
**Quản lý xem chi tiết lương**



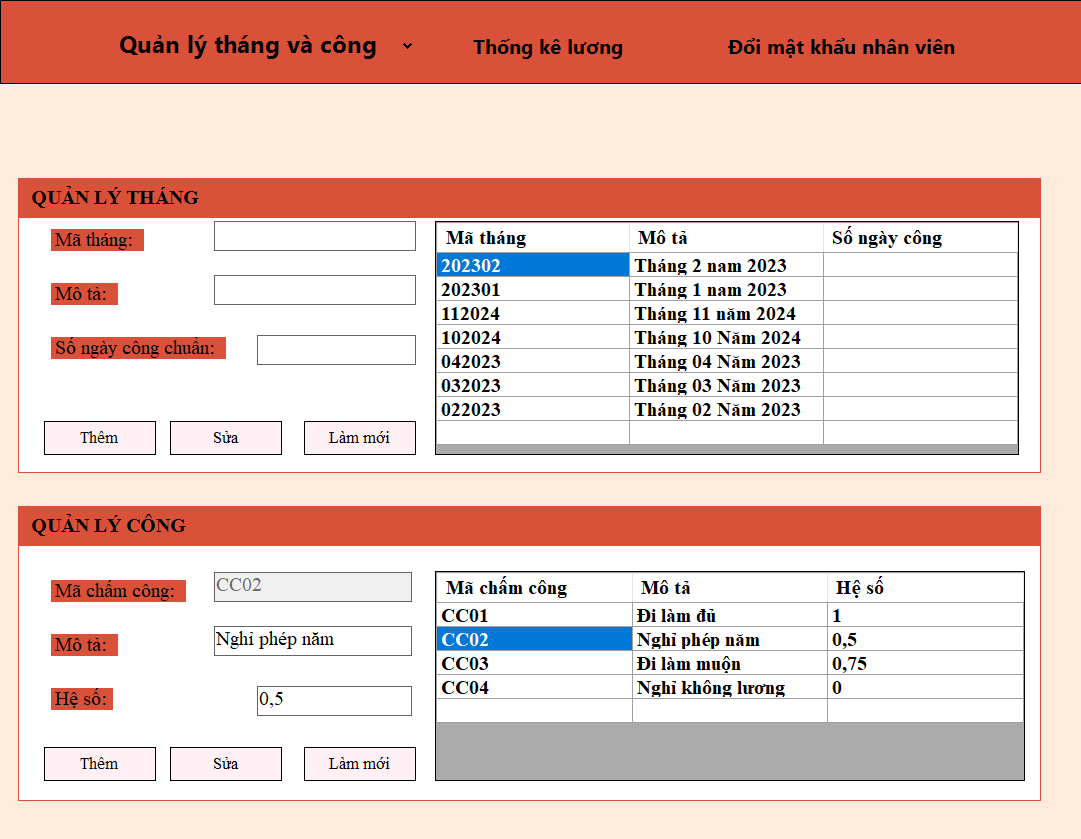
## 11. Quản lý hợp đồng



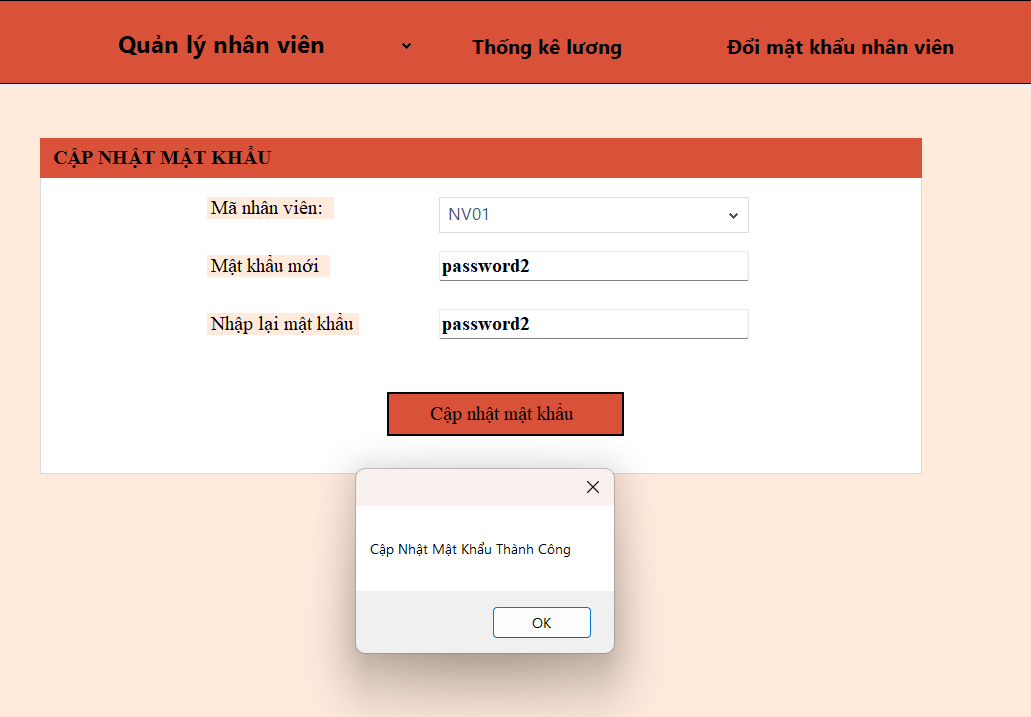
## 12. Quản lý người phụ thuộc



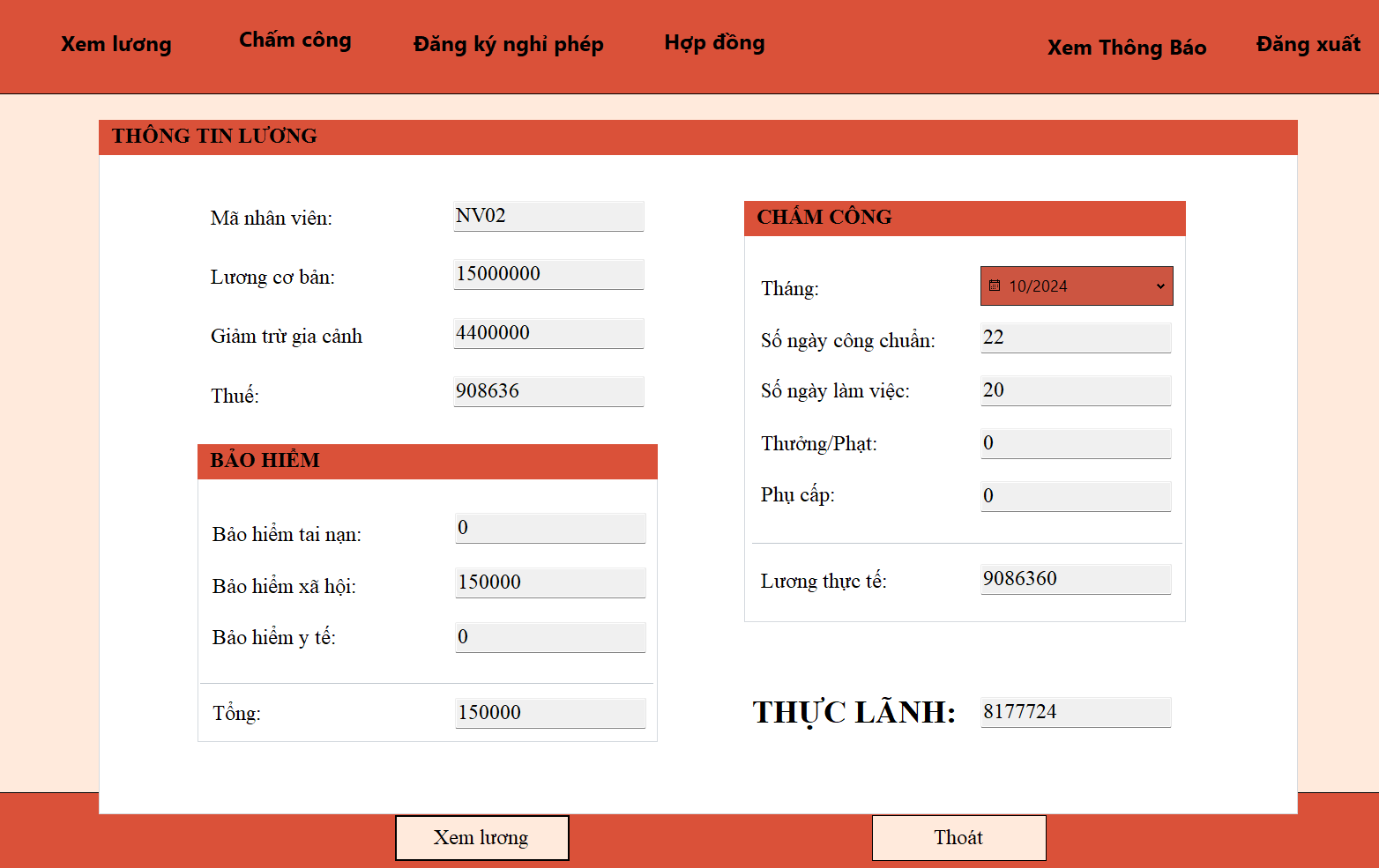
## 13. Quản lý tháng và công



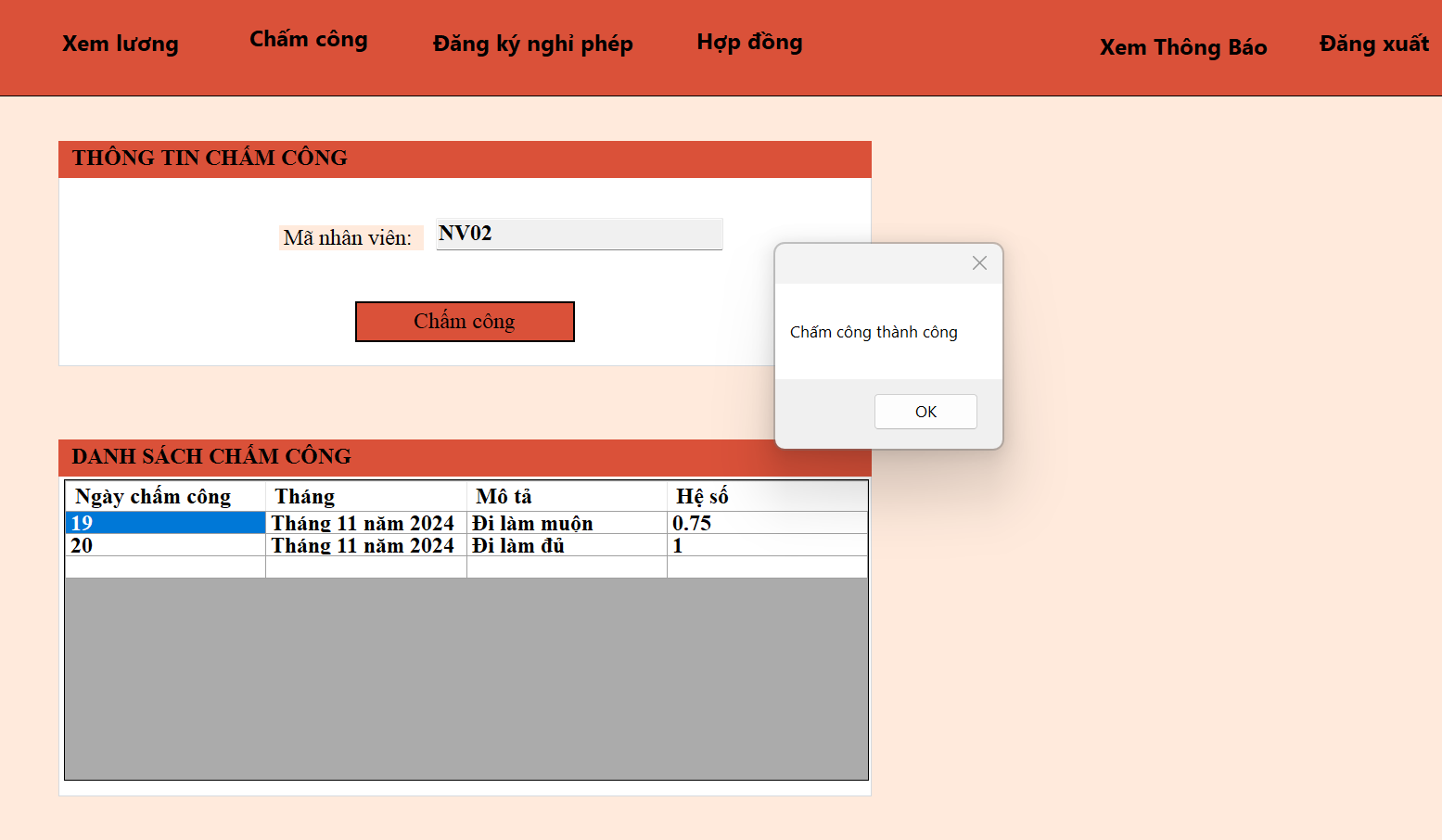
## 14. Cập nhật mật khẩu nhân viên



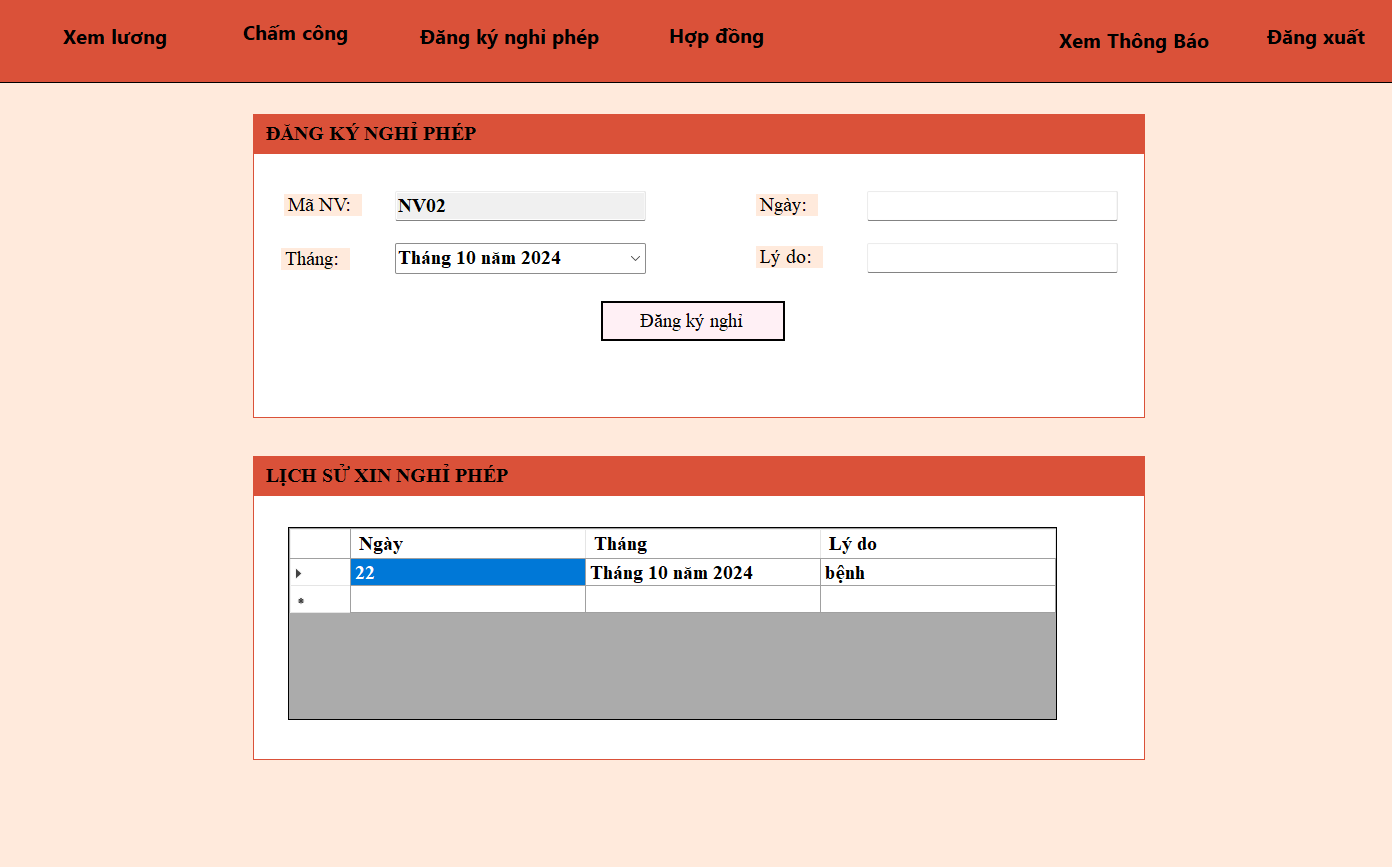
## 15. Nhân viên xem lương

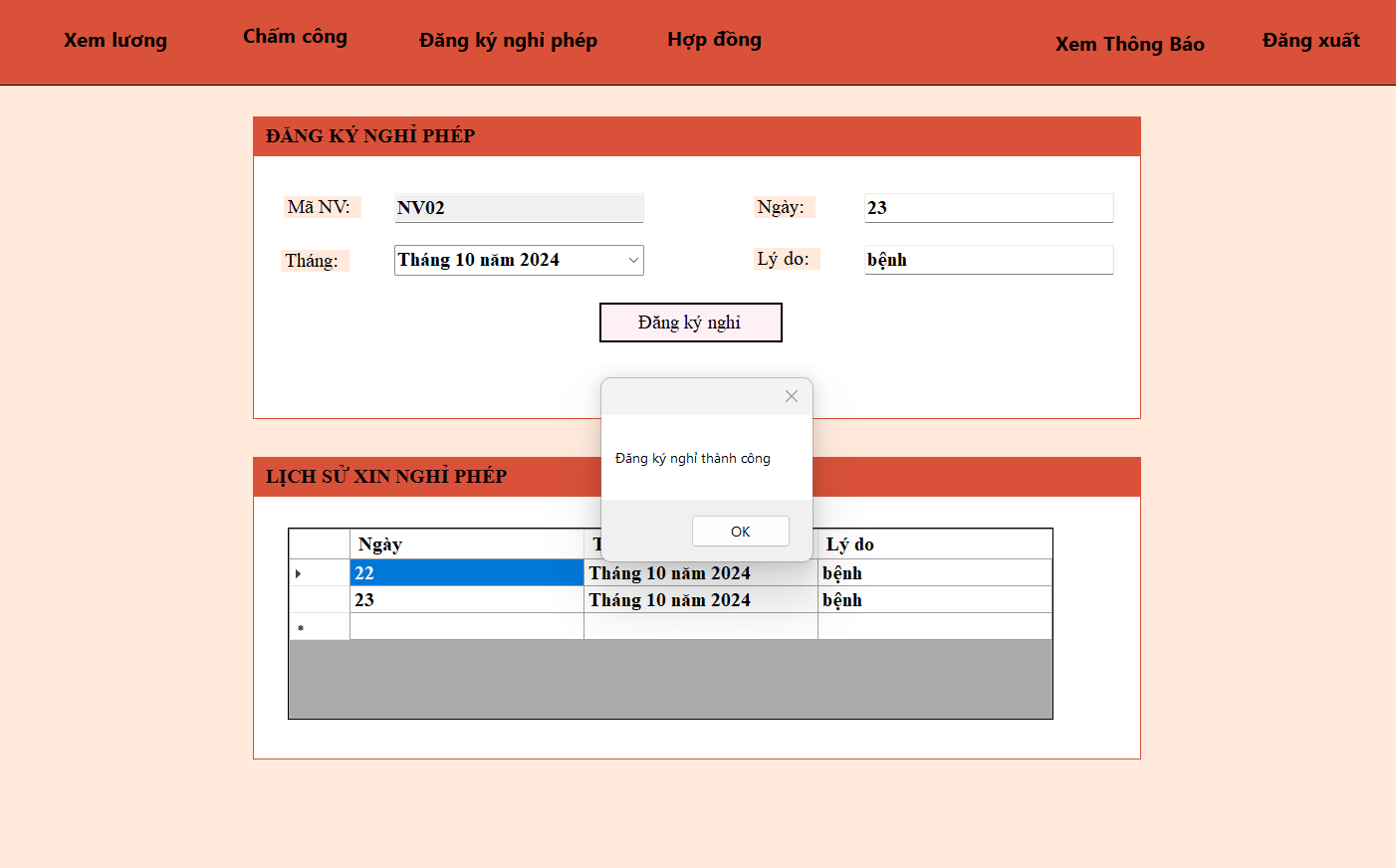


## 16. Nhân viên chấm công

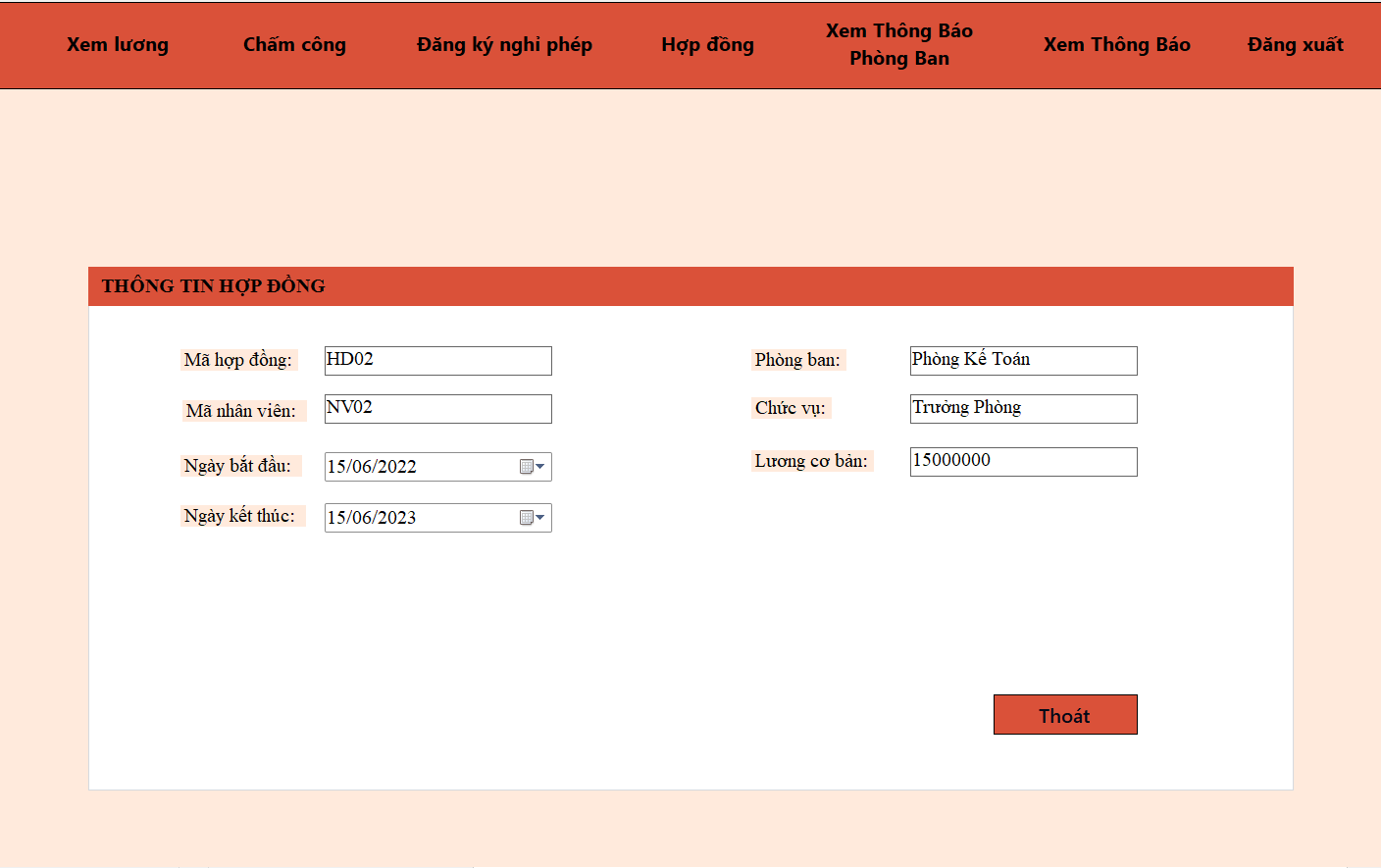


## 17. Nhân viên đăng ký nghỉ phép

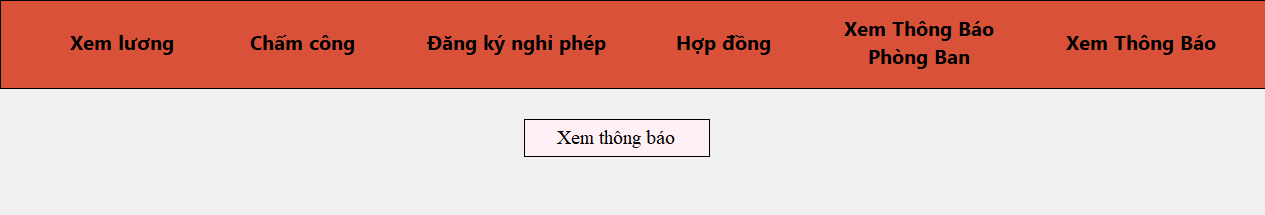




## 18. Nhân viên xem hợp đồng



## 19. Xem thông báo



Nhân viên xem thông báo:



Trưởng phòng xem tất cả thông báo:



1. Bảo hiểm xã hội Việt Nam, Tư vấn hỏi đáp, truy cập ngày 25/10/2024, [Mức đóng bảo hiểm người lao động](https://baohiemxahoi.gov.vn/chuyen-trang-bhxh-bhyt/tu-van-hoi-dap/Pages/default.aspx?ItemID=1835) [↑](#footnote-ref-1)
2. Báo Người Lao Động, Giải đáp pháp luật (3/4/2024), truy cập ngày 25/10/2024, Cách tính thuế thu nhập cá nhân năm 2024 [↑](#footnote-ref-2)