**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**



**ĐỀ TÀI CUỐI KỲ MÔN HỆ QUẢN TRỊ CSDL**

**TÊN ĐỀ TÀI: QUẢN LÝ CỬA HÀNG BÁN QUẦN ÁO**

**( QUẢN LÝ NHÂN VIÊN )**

**GVHD:** TS. Nguyễn Thành Sơn

**Lớp HP:** DBMS330284\_02

**Sinh viên thực hiện:**

Đào Minh Nhựt – 23110282

**Học kỳ**: 1

**Năm học**: 2025 – 2026

*Thành phố Hồ Chí Minh, tháng 9, năm 2025*

DANH SÁCH SINH VIÊN THỰC HIỆN

HỌC KÌ 1 NĂM HỌC 2025 – 2026

Nhóm 1

Đề tài: QUẢN LÝ CỬA HÀNG BÁN QUẦN ÁO ( QUẢN LÝ NHÂN VIÊN )

|  |  |
| --- | --- |
| 23110282 | Đào Minh Nhựt |

**NHẬN XÉT CỦA GIÁO VIÊN**

Tp. Hồ Chí Minh, ngày 27 tháng 9 năm 2025

Giảng viên chấm điểm

**MỤC LỤC**

[CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG 1](#_Toc209905348)

[1. Đặc tả đề tài 1](#_Toc209905349)

[1.1 Mô tả bài toán 1](#_Toc209905350)

[1.2 Mô tả dữ liệu 1](#_Toc209905351)

[1.2.1 Mô tả các tập thực thể 1](#_Toc209905352)

[1.2.2 Mô tả các mối quan hệ 2](#_Toc209905353)

[1.3 Mô tả chức năng 4](#_Toc209905354)

[1.3.1. Chức năng hệ thống 4](#_Toc209905355)

[1.3.2 Chức năng chung 4](#_Toc209905356)

[1.3.3 Chức năng dành cho Quản lý 4](#_Toc209905357)

[1.3.4 Chức năng dành cho Nhân viên 5](#_Toc209905358)

[1.4 Về phần giao diện 6](#_Toc209905359)

[1.4.1 Phần giao diện chung 6](#_Toc209905360)

[1.4.2 Phần giao diện riêng 6](#_Toc209905361)

[CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 8](#_Toc209905362)

[1. Thiết kế cơ sở dữ liệu mức quan niệm 8](#_Toc209905363)

[2. Thiết kế cơ sở dữ liệu mức logic 9](#_Toc209905364)

[3. Các ràng buộc cần có 9](#_Toc209905365)

[4. Cài đặt SQL và các ràng buộc 11](#_Toc209905366)

[5. Lược đồ quan hệ sau khi cài đặt SQL 14](#_Toc209905367)

[6. Các view 14](#_Toc209905368)

[7. Các trigger 15](#_Toc209905369)

[CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG 18](#_Toc209905370)

[1. Kết nối cơ sở dữ liệu 18](#_Toc209905371)

[1.1 Class Database 18](#_Toc209905372)

[1.2 Class Global State 20](#_Toc209905373)

[2. Chức năng của Quản lý 21](#_Toc209905374)

[2.1 Thêm tài khoản 21](#_Toc209905375)

[2.2 Xóa tài khoản 23](#_Toc209905376)

[2.3 Quên mật khẩu 25](#_Toc209905377)

[2.4 Tìm kiếm tài khoản 27](#_Toc209905378)

[2.5 Thêm nhân viên 28](#_Toc209905379)

[2.6 Xóa nhân viên 29](#_Toc209905380)

[2.7 Cập nhật nhân viên 31](#_Toc209905381)

[2.8 Tìm kiếm nhân viên 33](#_Toc209905382)

[2.9 Thêm công việc 34](#_Toc209905383)

[2.10 Xóa công việc 35](#_Toc209905384)

[2.11 Cập nhật công việc 37](#_Toc209905385)

[2.12 Tìm kiếm công việc 38](#_Toc209905386)

[2.13 Chấm công 39](#_Toc209905387)

[2.14 Xóa bảng ghi chấm công 40](#_Toc209905388)

[2.15 Xóa tất cả chấm công 41](#_Toc209905389)

[2.16 Tính lương 42](#_Toc209905390)

[2.17 Xóa lương 44](#_Toc209905391)

[2.18 Tính tổng lương phải trả cho tất cả nhân viên 46](#_Toc209905392)

[2.19 Xem số nhân viên theo công việc 47](#_Toc209905393)

[2.20 Xem danh sách nhân viên theo tên công việc 47](#_Toc209905394)

[2 Chức năng của Nhân viên 48](#_Toc209905395)

[3.1 Quên mật khẩu 48](#_Toc209905396)

[3.2 Xem thông tin lương của mình 50](#_Toc209905397)

[3.3 Xem hồ sơ nhân viên của mình 52](#_Toc209905398)

[CHƯƠNG 4: PHÂN QUYỀN 55](#_Toc209905399)

[1. Tạo Role và gán quyền 55](#_Toc209905400)

[1.1 Quản lý 55](#_Toc209905401)

[1.2 Nhân viên 56](#_Toc209905402)

[1.3 Login bên C# 56](#_Toc209905403)

[CHƯƠNG 5: TRANSACTION 58](#_Toc209905404)

[1. Transaction trong thủ tục sp\_ThemTaiKhoan 58](#_Toc209905405)

[2. Transaction trong thủ tục sp\_XoaTaiKhoan 59](#_Toc209905406)

[3. Transaction trong thủ tục sp\_QuenMatKhau 60](#_Toc209905407)

[4. Transaction trong thủ tục sp\_TinhLuong 61](#_Toc209905408)

[CHƯƠNG 6: THIẾT KẾ GIAO DIỆN 64](#_Toc209905409)

[1. Môi trường cài đặt và công nghệ sử dụng 64](#_Toc209905410)

[2. Giao diện chung 64](#_Toc209905411)

[2.1 Form đăng nhập (frmLogin) 64](#_Toc209905412)

[2.2 Form đổi mật khẩu (frmForgotPassword) 65](#_Toc209905413)

[2.3 Form Menu chính (frmMenu) 65](#_Toc209905414)

[3. Giao diện riêng 66](#_Toc209905415)

[3.1 Giao diện Quản lý 66](#_Toc209905416)

[3.1.1 User Control Quản lý tài khoản (UC\_Quanlytaikhoan) 66](#_Toc209905417)

[3.1.2 User Control Quản lý nhân viên (UC\_Quanlynhanvien) 66](#_Toc209905418)

[3.1.3 User Control Quản lý công việc (UC\_Quanlycongviec) 67](#_Toc209905419)

[3.1.4 User Control Chấm Công (UC\_ChamCong) 67](#_Toc209905420)

[3.1.5 User Control Tính Lương (UC\_TinhLuong) 67](#_Toc209905421)

[3.2 Giao diện Nhân viên 68](#_Toc209905422)

[3.2.1 User Control Xem bảng lương (UC\_Xembangluong) 68](#_Toc209905423)

[3.2.2 User Control Hosonhanvien (UC\_Hosonhanvien) 68](#_Toc209905424)

**DANH MỤC HÌNH ẢNH**

[Hình 2.1 Form đăng nhập (frmLogin) 64](#_Toc209905425)

[Hình 2.2 Form đổi mật khẩu (frmForgotPassword) 65](#_Toc209905426)

[Hình 2.3 Form Menu chính (frmMenu) 65](#_Toc209905427)

[Hình 3.1 User Control Quản lý tài khoản (UC\_Quanlytaikhoan) 66](#_Toc209905428)

[Hình 3.2 User Control Quản lý nhân viên (UC\_Quanlynhanvien) 66](#_Toc209905429)

[Hình 3.3 User Control Quản lý công việc (UC\_Quanlycongviec) 67](#_Toc209905430)

[Hình 3.4 User Control Chấm Công (UC\_ChamCong) 67](#_Toc209905431)

[Hình 3.5 User Control Tính Lương (UC\_TinhLuong) 68](#_Toc209905432)

[Hình 3.6 User Control Xem bảng lương (UC\_Xembangluong) 68](#_Toc209905433)

[Hình 3.7 User Control Hosonhanvien (UC\_Hosonhanvien) 69](#_Toc209905434)

**LỜI CẢM ƠN**

Kính thưa thầy Nguyễn Thành Sơn,

Em xin gửi lời cảm ơn sâu sắc và lòng biết ơn chân thành đến thầy vì sự hướng dẫn tận tâm và những hỗ trợ quý báu trong suốt quá trình thực hiện đồ án môn Hệ quản trị cơ sở dữ liệu.

Nhờ vào những chỉ dẫn tận tình, những bài giảng đầy tâm huyết và sự chia sẻ kinh nghiệm quý giá từ thầy, Em đã nỗ lực cố gắng để hoàn thiện đồ án một cách hiệu quả và đạt được nhiều kết quả. Thầy không chỉ truyền đạt kiến thức chuyên môn mà còn giúp em xây dựng nền tảng vững chắc về tư duy thiết kế và quản lý cơ sở dữ liệu, những kỹ năng mà chúng em tin rằng sẽ là hành trang giá trị cho hành trình học tập và sự nghiệp sau này.

Một lần nữa, Em xin bày tỏ lòng tri ân sâu sắc đến thầy. Kính chúc thầy luôn dồi dào sức khỏe, hạnh phúc và tiếp tục truyền cảm hứng cho nhiều thế hệ sinh viên.

Trân trọng,

Đào Minh Nhựt

**LỜI NÓI ĐẦU**

Kính gửi quý thầy cô và các bạn sinh viên,

Đồ án “Quản lý cửa hàng bán quần áo (Quản lý nhân viên)” là kết quả của quá trình học tập và thực hành môn Hệ quản trị cơ sở dữ liệu của em. Đây là một dự án đầy thử thách nhưng cũng mang lại nhiều kiến thức và kinh nghiệm quý báu trong việc áp dụng lý thuyết vào thực tiễn.

Trong bối cảnh ngành bán lẻ thời trang tại Việt Nam ngày càng phát triển, việc quản lý nhân viên hiệu quả là yếu tố then chốt để đảm bảo hoạt động kinh doanh của các cửa hàng quần áo diễn ra suôn sẻ. Đề tài ra đời nhằm đáp ứng nhu cầu tổ chức và quản lý thông tin nhân sự một cách khoa học, giúp tối ưu hóa quy trình làm việc và nâng cao hiệu quả quản lý của doanh nghiệp.

Đồ án này là một ứng dụng thực tiễn, được xây dựng để hỗ trợ các cửa hàng bán quần áo quản lý thông tin nhân viên, lịch làm việc, chấm công, tính lương và các hoạt động liên quan một cách dễ dàng và hiệu quả. Bằng việc áp dụng kiến thức về hệ quản trị cơ sở dữ liệu, em đã thiết kế một hệ thống quản lý hoàn chỉnh với các chức năng như quản lý tài khoản, quản lý nhân viên, công việc, chấm công, tính lương và nhiều thứ khác.

Em hy vọng rằng đồ án này không chỉ thể hiện được những kiến thức đã học mà còn mang lại giá trị tham khảo hữu ích cho quý thầy cô và các bạn sinh viên. Đồng thời, em mong rằng dự án sẽ khơi gợi cảm hứng và ý tưởng sáng tạo cho các bạn sinh viên khác trong việc ứng dụng hệ quản trị cơ sở dữ liệu vào thực tế.

Em xin chân thành cảm ơn quý thầy cô và các bạn sinh viên đã dành thời gian theo dõi và đóng góp ý kiến quý báu để đồ án của em được hoàn thiện hơn.

# CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG

# 1. Đặc tả đề tài

## 1.1 Mô tả bài toán

Hệ thống quản lý cửa hàng bán quần áo được phát triển nhằm tối ưu hóa các hoạt động kinh doanh, trong đó phân hệ quản lý nhân viên đóng vai trò then chốt, hỗ trợ cửa hàng tổ chức và giám sát hiệu quả công việc của đội ngũ nhân sự.

Nhân viên là lực lượng cốt lõi, trực tiếp thực hiện các nhiệm vụ như bán hàng, kế toán, quản lý kho và giải quyết các vấn đề phát sinh. Tuy nhiên, việc quản lý thủ công thường dẫn đến sai sót trong chấm công, tính lương và đánh giá hiệu suất, gây tốn thời gian và thiếu minh bạch. Hệ thống quản lý nhân viên, được xây dựng trên nền tảng WinForms C# và tích hợp với hệ quản trị cơ sở dữ liệu (DBMS) như SQL Server, hướng tới tự động hóa các tác vụ này. Mục tiêu chính bao gồm lưu trữ và quản lý thông tin nhân viên, chấm công chính xác, tính toán lương thưởng minh bạch dựa trên giờ làm và hiệu suất, đồng thời cung cấp các báo cáo thống kê về hiệu quả lao động. Giao diện WinForms thân thiện, dễ sử dụng cùng cơ sở dữ liệu mạnh mẽ giúp giảm thiểu sai sót, tiết kiệm thời gian và hỗ trợ nhà quản lý đưa ra quyết định chiến lược.

Hệ thống không chỉ nâng cao hiệu suất quản lý nhân sự mà còn đảm bảo tính bảo mật thông qua phân quyền tài khoản, đồng thời tạo nền tảng để mở rộng các chức năng như quản lý kho hoặc bán hàng trực tuyến. Với những lợi ích này, hệ thống góp phần giúp cửa hàng vận hành hiệu quả, minh bạch và đáp ứng tốt hơn nhu cầu kinh doanh trong bối cảnh cạnh tranh ngày càng cao.

## 1.2 Mô tả dữ liệu

### *1.2.1 Mô tả các tập thực thể*

Trong hệ thống quản lý nhân viên cửa hàng bán quần áo, tập thực thể TaiKhoan dùng để lưu trữ thông tin đăng nhập của người dùng hệ thống, bao gồm các thuộc tính như: MaTK là mã tài khoản tự tăng, dùng để định danh duy nhất cho mỗi tài khoản; TenTK là tên đăng nhập của người dùng, thường là duy nhất; MatKhau là mật khẩu đã được mã hóa hoặc lưu ở dạng chuỗi; và VaiTro thể hiện quyền hạn sử dụng, có thể là "QuanLy" hoặc "NhanVien", được dùng để phân quyền chức năng trong hệ thống.

Tập thực thể NhanVien thể hiện toàn bộ thông tin cá nhân của các nhân viên đang làm việc tại cửa hàng. Mỗi nhân viên có một MaNV là khóa chính, được dùng để liên kết với các bảng khác. Các thuộc tính khác bao gồm HoTen là họ và tên đầy đủ của nhân viên, DiaChi là nơi cư trú, NgaySinh ghi ngày sinh để dễ quản lý độ tuổi, Email và SoDT dùng để liên hệ. Đặc biệt, MaCV liên kết tới công việc hiện tại của nhân viên trong bảng CongViec, và MaTK liên kết với bảng TaiKhoan để xác định tài khoản tương ứng.

Tập thực thể CongViec mô tả các loại công việc có trong cửa hàng như bán hàng, thu ngân, quản lý kho, v.v... Mỗi công việc có một mã MaCV là khóa chính, TenCV là tên của công việc và LuongCoBan thể hiện mức lương tính theo giờ làm việc cho công việc đó. Dữ liệu trong bảng này giúp hệ thống phân loại công việc và tính lương phù hợp.

Tập thực thể ChamCong ghi nhận lịch sử làm việc của từng nhân viên theo từng ngày. Mỗi bản ghi có một MaCC là mã chấm công, MaNV xác định nhân viên được ghi nhận, Ngay là ngày làm việc, GioVao và GioRa ghi thời gian bắt đầu và kết thúc ca làm. Hệ thống tính số giờ làm việc qua SoGioLam, và nếu có làm ngoài giờ thì được ghi lại ở SoGioTangCa. Bảng này là cơ sở để tính lương hằng ngày cho nhân viên.

Tập thực thể Luong dùng để tính lương chi tiết cho từng nhân viên trong từng ngày. MaLuong là mã lương, MaNV liên kết với nhân viên được tính lương, Thuong là khoản tiền thưởng nếu có, PhuCap là phụ cấp thêm theo chính sách của cửa hàng, LuongTangCa là số tiền được tính riêng cho giờ làm thêm, và cuối cùng TongLuong là tổng tiền thực lãnh, được tính toán dựa trên số giờ làm, mức lương cơ bản, thưởng, phụ cấp và tăng ca.

### *1.2.2 Mô tả các mối quan hệ*

**Tài khoản (TaiKhoan) - Nhân viên (NhanVien)**:

Là mối quan hệ một – một (1:1). Mỗi nhân viên khi được tuyển dụng và đưa vào hệ thống sẽ được cấp phát một tài khoản duy nhất để có thể đăng nhập vào phần mềm và thực hiện các chức năng phù hợp với vai trò của mình. Ngược lại, mỗi tài khoản cũng chỉ được gán cho một nhân viên duy nhất, đảm bảo tính bảo mật và phân quyền rõ ràng cho từng cá nhân. Tài khoản bao gồm thông tin tên đăng nhập, mật khẩu và vai trò hệ thống, trong khi bảng nhân viên lưu giữ thông tin cá nhân và công việc. Sự ràng buộc 1:1 này là cốt lõi để đảm bảo rằng không có trường hợp một tài khoản dùng chung cho nhiều người, hay một người dùng có nhiều tài khoản gây ra sai lệch dữ liệu và mất an toàn hệ thống.

**Nhân viên (NhanVien) - Công việc (CongViec)**:

Là mối quan hệ một – nhiều (1:N). Trong thực tế hoạt động của cửa hàng, mỗi nhân viên chỉ được phân công một chức vụ duy nhất, chẳng hạn như quản lý, nhân viên thu ngân, nhân viên bán hàng, hoặc nhân viên kho. Tuy nhiên, một công việc cụ thể như "Nhân viên bán hàng" lại có thể được đảm nhiệm bởi nhiều người cùng lúc, tùy theo quy mô hoạt động và số lượng ca làm việc. Điều đó có nghĩa là, mỗi bản ghi trong bảng CongViec sẽ có thể liên kết với nhiều bản ghi trong bảng NhanVien, nhưng ngược lại, mỗi nhân viên chỉ liên kết với duy nhất một công việc tại một thời điểm. Việc thiết kế quan hệ theo hướng này giúp hệ thống quản lý tốt việc phân bổ công việc, hỗ trợ thống kê và tính toán lương chính xác dựa trên vị trí làm việc.

**Nhân viên (NhanVien) - Chấm công (ChamCong)**:

Là mối quan hệ một – nhiều (1:N). Trong quá trình làm việc, mỗi nhân viên sẽ được ghi nhận thời gian vào ca và ra ca hàng ngày nhằm quản lý thời gian làm việc, tính lương và đánh giá hiệu suất công việc. Vì thế, mỗi nhân viên sẽ có thể có nhiều bảng ghi chấm công tương ứng với từng ngày đi làm khác nhau. Bảng ChamCong sẽ ghi nhận thông tin mã nhân viên, ngày làm việc, giờ vào – giờ ra, số giờ làm và giờ tăng ca (nếu có). Thiết kế này phản ánh đúng bản chất dữ liệu thực tế: mỗi nhân viên có thể làm nhiều ca, nhiều ngày, do đó tạo ra nhiều bản ghi chấm công, nhưng mỗi bản ghi chỉ thuộc về duy nhất một nhân viên.

**Nhân viên (NhanVien) - Lương (Luong)**:

Là mối quan hệ một – một (1:1) trong bối cảnh mô hình đơn giản hóa. Mỗi nhân viên trong hệ thống được tính lương dựa trên thời gian làm việc, số giờ tăng ca, mức lương cơ bản theo công việc đang đảm nhận, cộng với các khoản phụ cấp, thưởng nếu có. Hệ thống được thiết kế để mỗi nhân viên chỉ có một bản ghi lương tổng hợp tương ứng với mỗi kỳ tính lương (theo tháng, theo ca,... tùy cách tổ chức), nên tại thời điểm hiện tại trong hệ thống, mỗi nhân viên chỉ có một bản ghi lương duy nhất. Bảng Luong sẽ chứa các thông tin như mã nhân viên, lương tăng ca, phụ cấp, thưởng và tổng lương. Sự ràng buộc 1:1 trong quan hệ này giúp hệ thống quản lý lương gọn gàng, đảm bảo tính duy nhất và rõ ràng trong báo cáo, thống kê tài chính.

## 1.3 Mô tả chức năng

### *1.3.1. Chức năng hệ thống*

Hệ thống quản lý nhân viên cho cửa hàng bán quần áo được xây dựng nhằm phục vụ 2 nhóm người dùng chính bao gồm: Quản lý và Nhân viên. Mỗi nhóm người dùng có vai trò và chức năng khác nhau trong hệ thống, tùy thuộc vào quyền hạn được cấp phát thông qua tài khoản đăng nhập.

### *1.3.2 Chức năng chung*

Tất cả người dùng khi tham gia sử dụng hệ thống đều có thể truy cập một số chức năng cơ bản như:

Đăng nhập: Mỗi người dùng được cung cấp một tài khoản cá nhân gồm tên đăng nhập và mật khẩu. Sau khi đăng nhập thành công, hệ thống sẽ tự động phân quyền để truy cập vào menu giao diện chính, ở đó hiển thị các chức năng tương ứng với vai trò của từng người dùng.

Quên mật khẩu: Mỗi người dùng khi quên mật khẩu có thể tiến hành tạo mật khẩu mới để tiếp tục vào hệ thống.

### *1.3.3 Chức năng dành cho Quản lý*

Quản lý là người có quyền cao nhất trong hệ thống, có thể truy cập toàn bộ dữ liệu và thao tác quản trị. Các chức năng chính bao gồm:

Quản lý tài khoản và phân quyền: Quản lý có thể tạo mới hoặc xóa tài khoản đăng nhập của nhân viên. Hay tìm kiếm thông tin 1 tài khoản nhân viên khi có việc cần đến. Khi tạo tài khoản mới cho nhân viên, hệ thống sẽ tự động tạo tài khoản user cho nhân viên, cấp quyền phù hợp với role bên SQL.

Quản lý nhân viên: Quản lý có thể thêm, chỉnh sửa hoặc xóa thông tin nhân viên trong hệ thống, đồng thời có thể tìm kiếm theo tên nhân viên một cách nhanh chóng. Các thông tin bao gồm mã nhân viên, họ tên, giới tính, ngày sinh, địa chỉ, số điện thoại, email, mã công việc đang đảm nhận,...

Quản lý công việc: Có thể thêm hoặc điều chỉnh danh sách các công việc như “Nhân viên bán hàng”, “Nhân viên kho”, “Nhân viên kế toán”,... Mỗi công việc sẽ có mức lương theo giờ tương ứng.

Chấm công nhân viên:Quản lý chấm công cho nhân viên bằng cách nhập đầy đủ thông tin chấm công. Thông tin chấm công bao gồm thời gian vào – ra, tổng số giờ làm việc, số giờ tăng ca. Đồng thời có thể thực hiện chức năng xóa 1 bảng ghi chấm công nếu nhập sai, hay việc xóa tất cả các bảng ghi chấm công trong ngày để tiếp tục chấm công cho ngày mới.

Tính lương nhân viên: Hệ thống hỗ trợ tự động tính lương dựa trên giờ công, giờ tăng ca, thưởng và phụ cấp (nếu có), quản lý có thể nhập thêm những thuộc tính như Thưởng, Phụ cấp nếu cần thiết. Hiển thị tổng lương phải trả cho nhân viên giúp người quản lý tính được số tiền phải trả theo từng ngày.

### *1.3.4 Chức năng dành cho Nhân viên*

Nhân viên là những người thực hiện công việc tại cửa hàng như bán hàng, kho, thu ngân,... Họ có tài khoản cá nhân và quyền truy cập hạn chế để thực hiện các chức năng sau:

Xem bảng lương: Nhân viên có thể xem được thông tin chấm công của bản thân, đồng thời xem thông tin lương dựa trên đó, đảm bảo quyền nhân viên và tính minh bạch của hệ thống

Hồ sơ nhân viên: Nhân viên có thể xem được thông tin chính mình để báo cáo lại Quản lý việc chỉnh sửa, cập nhật hồ sơ tránh xảy ra những vấn đề khác.

## 1.4 Về phần giao diện

### *1.4.1 Phần giao diện chung*

Các giao diện trong hệ thống quản lý nhân viên cho cửa hàng bán quần áo được thiết kế để hỗ trợ 2 nhóm người dùng chính, bao gồm quản lý và nhân viên. Mỗi người dùng sẽ có quyền truy cập vào các chức năng khác nhau tùy thuộc vào vai trò của mình trong hệ thống. Giao diện của hệ thống được tối ưu hóa để đảm bảo sự dễ sử dụng, trực quan và thân thiện với người dùng.

*Giao diện đăng nhập (frmLogin):* Đây là giao diện đầu tiên người dùng gặp khi truy cập vào hệ thống. Giao diện này có các thành phần như: nhãn tên tài khoản, nhãn mật khẩu, ô nhập tên tài khoản, ô nhập mật khẩu (mật khẩu được ẩn đi), nút đăng nhập và liên kết quên mật khẩu. Giao diện này cho phép người dùng nhập tên tài khoản và mật khẩu để truy cập vào hệ thống. Nếu thông tin đăng nhập không đúng, hệ thống sẽ thông báo lỗi và yêu cầu người dùng thử lại.

*Giao diện Quên mật khẩu (frmForgotPassword):* Phần giao diện này bao gồm các textbox để người dùng nhập thông tin về tài khoản, mật khẩu mới và xác nhận mật khẩu mới, button Xác nhận để lưu những thông tin đã thay đổi, giúp người khi cần có việc vào hệ thống nhưng lại quên mật khẩu có thể vào ngay để xử lý công việc của mình.

*Giao diện Menu (frmMenu):* Phần giao diện này bao gồm các UserControl (UC) được quản lý trong một giao diện chính để tối ưu trải nghiệm người dùng, phù hợp với các vai trò khác nhau. Sau khi đăng nhập thành công, người dùng sẽ được đưa đến giao diện chính, nơi có các nút chức năng ở thanh bên trái. Các chức năng này sẽ tùy thuộc vào vai trò của người dùng (quản lý hay nhân viên), và khi nhấn vào mỗi nút, hệ thống sẽ load UserControl tương ứng vào Panel chính

### *1.4.2 Phần giao diện riêng*

*Phần giao diện của Quản lý sẽ bao gồm các form/chức năng:*

Quản lý tài khoản (UC\_Quanlytaikhoan): Dành cho quản lý thực hiện các chức năng thêm, xóa tài khoản, tìm kiếm 1 tài khoản nhanh chóng, thuận tiện.

Quản lý nhân viên (UC\_Quanlynhanvien): Dành cho quản lý để thêm, sửa, xóa, tìm kiếm thông tin nhân viên và theo dõi hoạt động của nhân viên trong hệ thống. Hiển thị danh sách nhân viên theo từng công việc để nắm được việc phân bổ nhân sự trong cửa hàng.

Quản lý công việc(UC\_Quanlycongviec): Quản lý các công việc (chức vụ) mà nhân viên đang đảm nhận trong cửa hàng. Thực hiện thêm, cập nhật và xóa công việc, đồng thời xem thống kê số nhân viên theo từng vị trí công việc để có chiến lược điều chỉnh nhân sự phù hợp.

Chấm công(UC\_ChamCong): Dành cho quản lý để ghi lại thời gian làm việc hàng ngày và chấm công cho nhân viên. Thực hiện xóa bảng ghi chấm công hoặc xóa toàn bộ bảng ghi khi đã hoàn thành 1 ngày làm việc.

Tính lương(UC\_TinhLuong): Dành cho quản lý để tính lương cho nhân viên dựa trên giờ làm việc, tăng ca và các khoản phụ cấp. Xem tổng lương phải trả trong nhân viên trong 1 ngày.

*Phần giao diện của Nhân viên sẽ bao gồm các form/chức năng:*

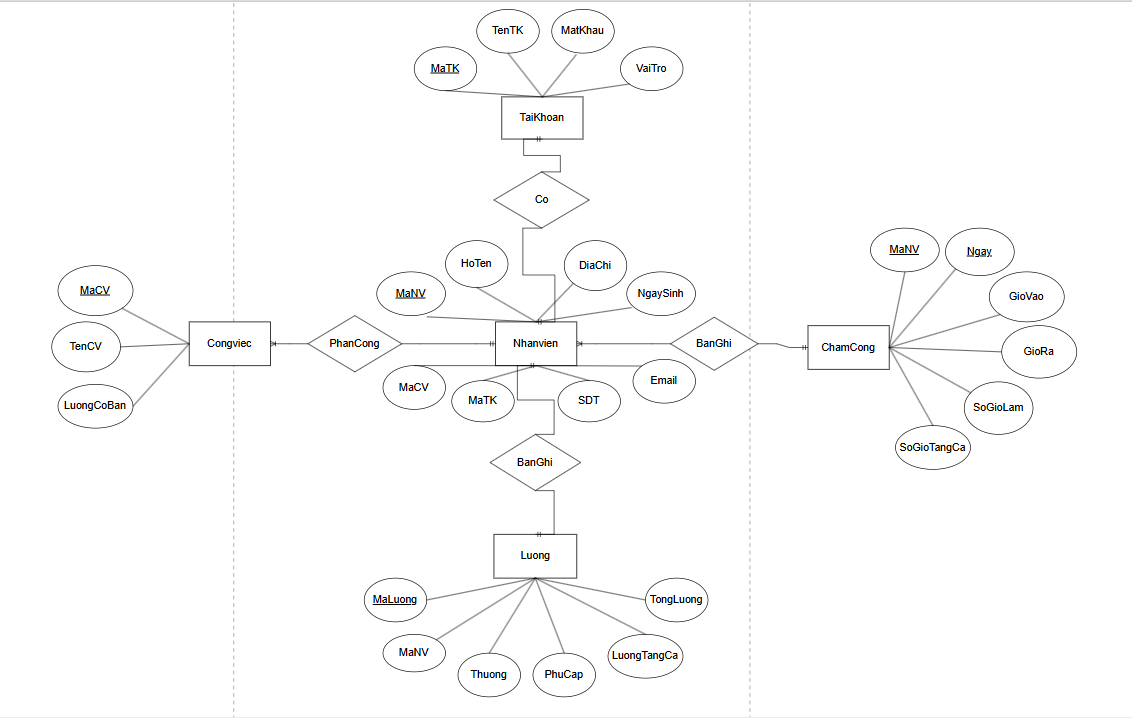
Xem bảng lương(UC\_Xembangluong): Nhân viên có thể xem bảng lương của mình.

Hồ sơ nhân viên(UC\_Hosonhanvien): Nhân viên có thể xem hồ sơ cá nhân của mình, kịp thời báo cáo cho Quản lý có những thay đổi phù hợp.

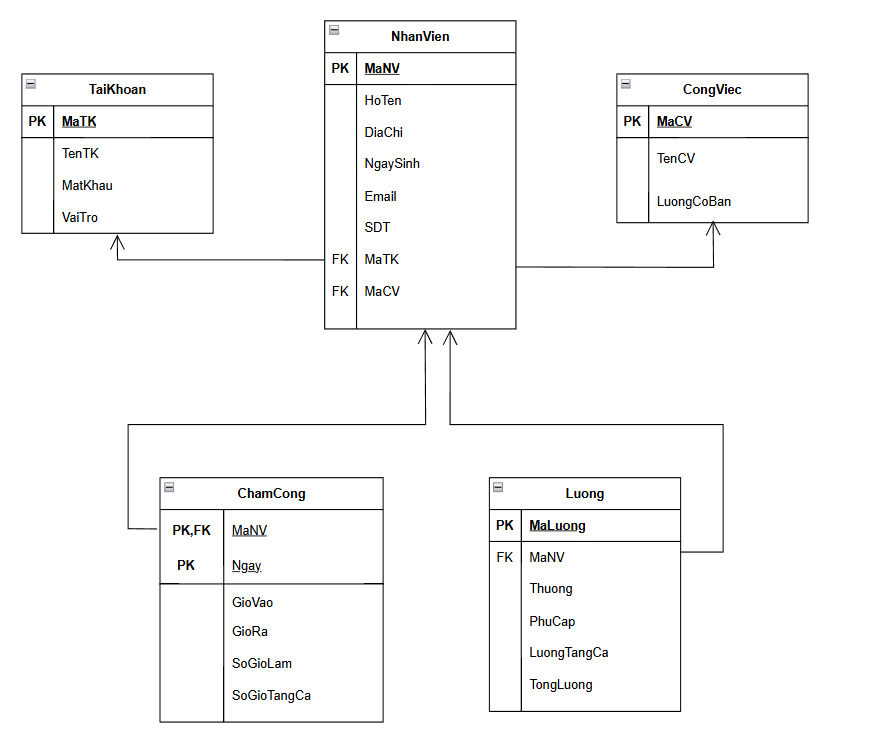
# CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

# Thiết kế cơ sở dữ liệu mức quan niệm

Sơ đồ thực thể kết hợp ERD:

****

# Thiết kế cơ sở dữ liệu mức logic



*Lược đồ quan hệ*

TaiKhoan (MaTK, TenTK, MatKhau, VaiTro)

NhanVien (MaNV, HoTen, DiaChi, NgaySinh, Email, SDT, MaTK, MaCV)

CongViec (MaCV, TenCV,LuongCoBan)

ChamCong(MaNV, Ngay, GioVao, GioRa, SoGioLam, SoGioTangCa)

Luong(MaLuong, MaNV, Thuong, PhuCap, LuongTangCa, TongLuong)

# Các ràng buộc cần có

|  |  |  |
| --- | --- | --- |
| **STT** | **Bảng** | **Ràng buộc** |
| 1 | **TaiKhoan**  MaTK INT  TenTK NVARCHAR(50)  MatKhau NVARCHAR(255)  VaiTro NVARCHAR(20) | Khóa chính MaTK (tự tăng)  NOT NULL TenTK  NOT NULL MatKhau  NOT NULL VaiTro CHECK VaiTro |
| 2 | **CongViec**  MaCV INT,  TenCV NVARCHAR(50),  LuongCoBan DECIMAL(18,2) | Khóa chính MaCV (tự tăng)  NOT NULL TenCV  NOT NULL LuongCoBan  UNIQUE TenCV  CHECK LuongCoBan |
| 3 | **NhanVien**  MaNV INT ,  HoTen NVARCHAR(100),  DiaChi NVARCHAR(200),  NgaySinh DATE,  Email NVARCHAR(100),  SDT NVARCHAR(15),  MaTK INT,  MaCV INT, | Khóa chính MaNV (tự tăng)  Khóa ngoại MaTK tham chiếu đến MaTK trong bảng TaiKhoan, ON DELETE SET NULL (Xóa tài khoản trong bảng TaiKhoan thì MaTK trong bảng NhanVien sẽ được set thành NULL)  Khóa ngoại MaCV tham chiếu đến MaCV trong bảng CongViec (Xóa 1 công việc trong bảng CongViec thì nhân viên liên quan tới công việc đó cũng bị xóa)  NOT NULL HoTen  NOT NULL MaCV  UNIQUE Email  UNIQUE SDT  UNIQUE MaTK  CHECK NgaySinh |
| 4 | **ChamCong**  MaNV INT,  Ngay DATE,  GioVao TIME,  GioRa TIME ,  SoGioLam AS DATEDIFF(HOUR, GioVao, GioRa)  SoGioTangCa INT | Khóa chính (MaNV Ngay)  Khóa ngoại MaNV tham chiếu đến MaNV trong bảng NhanVien, ON DELETE CASCADE (Khi xóa nhân viên thì bảng chấm công tương ứng với nhân viên đó cũng được xóa)  NOT NULL MaNV  NOT NULL Ngay  NOT NULL GioVao  NOT NULL GioRa  CHECK SoGioTangCa |
| 5 | **Luong**  MaLuong INT,  MaNV INT NOT NULL,  Thuong DECIMAL(18,2),  PhuCap DECIMAL(18,2),  LuongTangCa DECIMAL(18,2),  TongLuong DECIMAL(18,2) | Khóa chính MaLuong (tự tăng)  Khóa ngoại MaNV tham chiếu đến MaNV trong bảng NhanVien, khi xóa nhân viên thì tương ứng với bảng ghi lương của nhân viên đó cũng được xóa.  NOT NULL MaNV  CHECK Thuong  CHECK PhuCap  CHECK LuongTangCa |

# Cài đặt SQL và các ràng buộc

**Bảng Tài khoản**

|  |
| --- |
| CREATE TABLE TaiKhoan (  MaTK INT IDENTITY(1,1) PRIMARY KEY,  TenTK NVARCHAR(50) NOT NULL,  MatKhau NVARCHAR(255) NOT NULL,  VaiTro NVARCHAR(20) NOT NULL CHECK (VaiTro IN ('QuanLy', 'NhanVien'))  ); |

**Bảng Công Việc**

|  |
| --- |
| CREATE TABLE CongViec (  MaCV INT IDENTITY(1,1) PRIMARY KEY,  TenCV NVARCHAR(50) NOT NULL UNIQUE,  LuongCoBan DECIMAL(18,2) NOT NULL CHECK (LuongCoBan >= 0)  ); |

**Bảng Nhân Viên**

|  |
| --- |
| CREATE TABLE NhanVien (  MaNV INT IDENTITY(1,1) PRIMARY KEY,  HoTen NVARCHAR(100) NOT NULL,  DiaChi NVARCHAR(200),  NgaySinh DATE CHECK (NgaySinh <= GETDATE()),  Email NVARCHAR(100) UNIQUE,  SDT NVARCHAR(15) UNIQUE,  MaTK INT NULL UNIQUE,  MaCV INT NOT NULL,  FOREIGN KEY (MaTK) REFERENCES TaiKhoan(MaTK) ON DELETE SET NULL,  FOREIGN KEY (MaCV) REFERENCES CongViec(MaCV) ON DELETE CASCADE  ); |

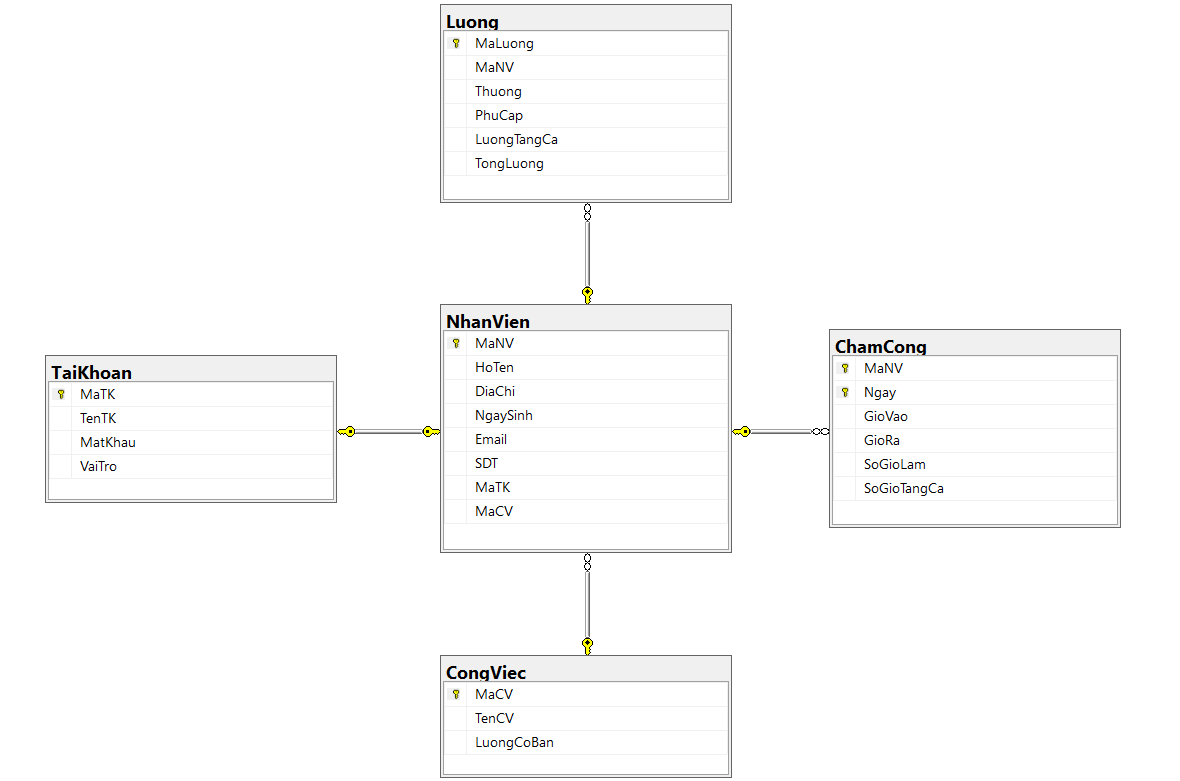
**Bảng Chấm Công**

|  |
| --- |
| CREATE TABLE ChamCong (  MaNV INT NOT NULL,  Ngay DATE NOT NULL,  GioVao TIME NOT NULL,  GioRa TIME NOT NULL,  SoGioLam AS DATEDIFF(HOUR, GioVao, GioRa) PERSISTED,  SoGioTangCa INT DEFAULT 0 CHECK (SoGioTangCa >= 0),  PRIMARY KEY (MaNV, Ngay),  FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV) ON DELETE CASCADE  ); |

**Bảng Lương**

|  |
| --- |
| CREATE TABLE Luong (  MaLuong INT IDENTITY(1,1) PRIMARY KEY,  MaNV INT NOT NULL,  Thuong DECIMAL(18,2) DEFAULT 0 CHECK (Thuong >= 0),  PhuCap DECIMAL(18,2) DEFAULT 0 CHECK (PhuCap >= 0),  LuongTangCa DECIMAL(18,2) DEFAULT 0 CHECK (LuongTangCa >= 0),  TongLuong DECIMAL(18,2),  FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV) ON DELETE CASCADE  ); |

# Lược đồ quan hệ sau khi cài đặt SQL

****

# Các view

1. **View xem Tài khoản**

|  |
| --- |
| CREATE VIEW vw\_NhanVien AS  SELECT \*  FROM NhanVien; |

1. **View xem Nhân viên**

|  |
| --- |
| CREATE VIEW vw\_NhanVien AS  SELECT \*  FROM NhanVien; |

1. **View xem Công việc**

|  |
| --- |
| CREATE VIEW vw\_CongViec AS  SELECT \*  FROM CongViec; |

1. **View xem Chấm Công**

|  |
| --- |
| CREATE VIEW vw\_ChamCong AS  SELECT \*  FROM ChamCong; |

1. **View xem Lương**

|  |
| --- |
| CREATE VIEW vw\_Luong AS  SELECT \*  FROM Luong; |

# Các trigger

1. **Trigger xóa bảng ghi Chấm Công khi xóa bảng ghi Lương của nhân viên**

|  |
| --- |
| CREATE TRIGGER trg\_Delete\_ChamCong\_Khi\_Xoa\_Luong  ON Luong  AFTER DELETE  AS  BEGIN  -- Kiểm tra xem có bản ghi nào bị xóa không  IF EXISTS (SELECT 1 FROM deleted)  BEGIN  -- Lấy mã nhân viên từ bảng deleted  DECLARE @MaNV INT;  SELECT @MaNV = MaNV FROM deleted;  -- Xóa các bản ghi trong bảng ChamCong tương ứng với MaNV của nhân viên  DELETE FROM ChamCong  WHERE MaNV = @MaNV;  END  END; |

1. **Trigger Cập nhật Lương khi thay đổi Lương Cơ Bản**

|  |
| --- |
| CREATE TRIGGER trg\_UpdateLuongKhiThayDoiLuongCoBan  ON CongViec  AFTER UPDATE  AS  BEGIN  -- Kiểm tra nếu cột LuongCoBan có thay đổi  IF UPDATE(LuongCoBan)  BEGIN  DECLARE @MaCV INT, @LuongCoBanMoi DECIMAL(18, 2);  -- Lấy thông tin MaCV và Lương cơ bản mới từ bản ghi đã cập nhật  SELECT @MaCV = MaCV, @LuongCoBanMoi = LuongCoBan  FROM INSERTED; -- INSERTED chứa giá trị của bản ghi sau khi cập nhật  -- Cập nhật lại Lương tăng ca cho tất cả nhân viên có MaCV tương ứng  -- Tính lại lương tăng ca dựa trên số giờ tăng ca và lương cơ bản mới  WITH UpdatedLuongTangCa AS (  SELECT MaNV,  SUM(SoGioTangCa \* 1.5 \* @LuongCoBanMoi) AS NewLuongTangCa  FROM ChamCong  WHERE MaNV IN (SELECT MaNV FROM NhanVien WHERE MaCV = @MaCV)  GROUP BY MaNV  )  -- Cập nhật lại Lương tăng ca và Tổng lương cho nhân viên có MaCV tương ứng  UPDATE Luong  SET  LuongTangCa = ULT.NewLuongTangCa,  -- Tính lại tổng lương sau khi đã cập nhật Lương tăng ca  TongLuong = (@LuongCoBanMoi \* CL.SoGioLam) + Thuong + PhuCap + ISNULL(ULT.NewLuongTangCa, 0)  FROM Luong L  JOIN UpdatedLuongTangCa ULT ON L.MaNV = ULT.MaNV  JOIN ChamCong CL ON CL.MaNV = L.MaNV -- Join với ChamCong để lấy SoGioLam  WHERE L.MaNV IN (SELECT MaNV FROM NhanVien WHERE MaCV = @MaCV); -- Cập nhật cho tất cả nhân viên có MaCV đã thay đổi  END  END; |

1. **Trigger kiểm tra giờ vào, giờ ra trong bảng Chấm Công**

|  |
| --- |
| CREATE TRIGGER trg\_KiemTraGioRa  ON ChamCong  INSTEAD OF INSERT, UPDATE  AS  BEGIN  IF EXISTS (  SELECT 1 FROM inserted WHERE GioRa <= GioVao  )  BEGIN  RAISERROR(N'Giờ ra phải lớn hơn giờ vào!', 16, 1);  RETURN;  END  MERGE ChamCong AS target  USING inserted AS source  ON target.MaNV = source.MaNV AND target.Ngay = source.Ngay  WHEN MATCHED THEN  UPDATE SET GioVao = source.GioVao, GioRa = source.GioRa, SoGioTangCa = source.SoGioTangCa  WHEN NOT MATCHED THEN  INSERT (MaNV, Ngay, GioVao, GioRa, SoGioTangCa)  VALUES (source.MaNV, source.Ngay, source.GioVao, source.GioRa, source.SoGioTangCa);  END; |

# CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG

# Kết nối cơ sở dữ liệu

## 1.1 Class Database

|  |
| --- |
| using System;  using System.Data;  using System.Data.SqlClient;  namespace QL\_NhanVien  {  internal class Database  {  private SqlConnection conn;  public Database()  {  conn = new SqlConnection(GlobalState.ConnectionString);  }  private void Open()  {  if (conn.State == ConnectionState.Closed)  conn.Open();  }  private void Close()  {  if (conn.State == ConnectionState.Open)  conn.Close();  }  public DataTable ExecuteQuery(string sql, SqlParameter[] param = null)  {  DataTable dt = new DataTable();  try  {  Open();  SqlCommand cmd = new SqlCommand(sql, conn);  if (param != null)  cmd.Parameters.AddRange(param);  SqlDataAdapter da = new SqlDataAdapter(cmd);  da.Fill(dt);  }  finally  {  Close();  }  return dt;  }  public int ExecuteNonQuery(string sql, SqlParameter[] param = null)  {  int rows = 0;  try  {  Open();  SqlCommand cmd = new SqlCommand(sql, conn);  if (param != null)  cmd.Parameters.AddRange(param);  rows = cmd.ExecuteNonQuery();  }  finally  {  Close();  }  return rows;  }  public int ExecuteStoredProc(string procName, SqlParameter[] param = null)  {  int rows = 0;  try  {  Open();  SqlCommand cmd = new SqlCommand(procName, conn);  cmd.CommandType = CommandType.StoredProcedure;  if (param != null)  cmd.Parameters.AddRange(param);  rows = cmd.ExecuteNonQuery();  }  finally  {  Close();  }  return rows;  }  public DataTable ExecuteProcToTable(string procName, SqlParameter[] param = null)  {  DataTable dt = new DataTable();  try  {  Open();  SqlCommand cmd = new SqlCommand(procName, conn);  cmd.CommandType = CommandType.StoredProcedure;  if (param != null)  cmd.Parameters.AddRange(param);  SqlDataAdapter da = new SqlDataAdapter(cmd);  da.Fill(dt);  }  finally  {  Close();  }  return dt;  }  }  } |

## 1.2 Class Global State

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace QL\_NhanVien  {  public static class GlobalState  {  public static int MaTK { get; set; }  public static string TenTaiKhoan { get; set; }  public static string VaiTro { get; set; }  public static string ConnectionString { get; set; }  }  } |

Lớp GlobalState giúp lưu trữ các giá trị toàn cục mà nhiều lớp hoặc phương thức trong ứng dụng có thể cần truy cập. Các giá trị này có thể là thông tin liên quan đến người dùng (như mã tài khoản, tên tài khoản, vai trò) và thông tin kết nối cơ sở dữ liệu. Sử dụng lớp tĩnh này giúp giảm thiểu việc phải truyền thông tin giữa các đối tượng và cho phép dễ dàng truy cập các giá trị toàn cục trong suốt ứng dụng.

# Chức năng của Quản lý

## 2.1 Thêm tài khoản

|  |
| --- |
| CREATE PROCEDURE sp\_ThemTaiKhoan  @TenTK NVARCHAR(50),  @MatKhau NVARCHAR(255),  @VaiTro NVARCHAR(20)  AS  BEGIN  SET NOCOUNT ON;  -- Bắt đầu giao dịch  BEGIN TRANSACTION;  BEGIN TRY  -- 1. Thêm vào bảng TaiKhoan  INSERT INTO TaiKhoan (TenTK, MatKhau, VaiTro)  VALUES (@TenTK, @MatKhau, @VaiTro);  -- 2. Tạo LOGIN nếu chưa có  IF NOT EXISTS (SELECT \* FROM sys.server\_principals WHERE name = @TenTK)  BEGIN  DECLARE @sqlLogin NVARCHAR(MAX);  SET @sqlLogin = 'CREATE LOGIN [' + @TenTK + '] WITH PASSWORD = ''' + @MatKhau + ''';';  EXEC(@sqlLogin);  END  -- 3. Tạo USER trong database nếu chưa có  IF NOT EXISTS (SELECT \* FROM sys.database\_principals WHERE name = @TenTK)  BEGIN  DECLARE @sqlUser NVARCHAR(MAX);  SET @sqlUser = 'CREATE USER [' + @TenTK + '] FOR LOGIN [' + @TenTK + '];';  EXEC(@sqlUser);  END  -- 4. Gán vào role tương ứng  IF @VaiTro = 'QuanLy'  BEGIN  EXEC sp\_addrolemember 'role\_QuanLy', @TenTK;  END  ELSE IF @VaiTro = 'NhanVien'  BEGIN  EXEC sp\_addrolemember 'role\_NhanVien', @TenTK;  END  -- Nếu tất cả các bước thành công, commit giao dịch  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  -- Nếu có lỗi xảy ra, rollback giao dịch  ROLLBACK TRANSACTION;  -- Thông báo lỗi  PRINT 'Lỗi: ' + ERROR\_MESSAGE();  THROW; -- Ném lại lỗi để có thể xử lý thêm ngoài thủ tục này nếu cần  END CATCH  END; |

**Xử lý bên C#:**

|  |
| --- |
| private void guna2Button1\_Click(object sender, EventArgs e)  {  if (txtTenTK.Text == "" || txtMatKhau.Text == "")  {  MessageBox.Show(" Vui lòng nhập đầy đủ thông tin!");  return;  }  SqlParameter[] parameters = {  new SqlParameter("@TenTK", txtTenTK.Text.Trim()),  new SqlParameter("@MatKhau", txtMatKhau.Text.Trim()),  new SqlParameter("@VaiTro", cboVaiTro.SelectedItem.ToString())  };  try  {  int rows = db.ExecuteStoredProc("sp\_ThemTaiKhoan", parameters);  MessageBox.Show("Thêm tài khoản thành công.");  LoadDanhSachTaiKhoan();  ClearForm();    }  catch (Exception ex)  {  MessageBox.Show(" Lỗi: " + ex.Message);  }  } |

## 2.2 Xóa tài khoản

|  |
| --- |
| CREATE PROCEDURE sp\_XoaTaiKhoan  @MaTK INT  AS  BEGIN  SET NOCOUNT ON;  -- Bắt đầu giao dịch  BEGIN TRANSACTION;  DECLARE @TenTK NVARCHAR(50);  -- Lấy tên tài khoản từ MaTK  SELECT @TenTK = TenTK FROM TaiKhoan WHERE MaTK = @MaTK;  IF @TenTK IS NULL  BEGIN  -- Nếu không tìm thấy tài khoản, ném lỗi và rollback giao dịch  RAISERROR(N'Tài khoản không tồn tại!', 16, 1);  ROLLBACK TRANSACTION;  RETURN;  END  BEGIN TRY  -- 1. Xóa tài khoản trong bảng TaiKhoan  DELETE FROM TaiKhoan WHERE MaTK = @MaTK;  -- 2. Xóa USER trong database nếu tồn tại  IF EXISTS (SELECT 1 FROM sys.database\_principals WHERE name = @TenTK)  BEGIN  DECLARE @sqlDropUser NVARCHAR(MAX);  SET @sqlDropUser = 'DROP USER [' + @TenTK + '];';  EXEC(@sqlDropUser);  END  -- 3. Xóa LOGIN trong server nếu tồn tại  IF EXISTS (SELECT 1 FROM sys.server\_principals WHERE name = @TenTK)  BEGIN  DECLARE @sqlDropLogin NVARCHAR(MAX);  SET @sqlDropLogin = 'DROP LOGIN [' + @TenTK + '];';  EXEC(@sqlDropLogin);  END  -- Nếu mọi thao tác thành công, commit giao dịch  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  -- Nếu có lỗi xảy ra, rollback giao dịch  ROLLBACK TRANSACTION;  -- Thông báo lỗi  PRINT 'Lỗi: ' + ERROR\_MESSAGE();  THROW; -- Ném lại lỗi để có thể xử lý thêm ngoài thủ tục này nếu cần  END CATCH  END;  GO |

**Xử lý bên C#:**

|  |
| --- |
| private void guna2Button2\_Click(object sender, EventArgs e)  {  if (dgvTaiKhoan.SelectedRows.Count == 0)  {  MessageBox.Show("Vui lòng chọn tài khoản cần xóa.");  return;  }  int maTK = Convert.ToInt32(dgvTaiKhoan.SelectedRows[0].Cells["MaTK"].Value);  DialogResult r = MessageBox.Show("Bạn có chắc muốn xóa tài khoản này?", "Xác nhận", MessageBoxButtons.YesNo);  if (r == DialogResult.Yes)  {  SqlParameter[] parameters = {  new SqlParameter("@MaTK", maTK)  };  try  {  int rows = db.ExecuteStoredProc("sp\_XoaTaiKhoan", parameters);  MessageBox.Show("Tài khoản đã được xóa.");  LoadDanhSachTaiKhoan();  ClearForm();  }  catch (Exception ex)  {  MessageBox.Show(" Lỗi: " + ex.Message);  }  }  } |

## 2.3 Quên mật khẩu

|  |
| --- |
| CREATE PROCEDURE sp\_QuenMatKhau  @TenTK NVARCHAR(50),  @MatKhauMoi NVARCHAR(255)  AS  BEGIN  SET NOCOUNT ON;  -- Bắt đầu giao dịch  BEGIN TRANSACTION;  BEGIN TRY  -- 1. Cập nhật mật khẩu trong bảng TaiKhoan  UPDATE TaiKhoan  SET MatKhau = @MatKhauMoi  WHERE TenTK = @TenTK;  -- 2. Cập nhật mật khẩu trong SQL Server Login  IF EXISTS (SELECT \* FROM sys.server\_principals WHERE name = @TenTK)  BEGIN  -- Cập nhật mật khẩu cho Login trong SQL Server  DECLARE @sqlLogin NVARCHAR(MAX);  SET @sqlLogin = 'ALTER LOGIN [' + @TenTK + '] WITH PASSWORD = ''' + @MatKhauMoi + ''';';  EXEC(@sqlLogin);  END  ELSE  BEGIN  -- Nếu Login không tồn tại trong SQL Server, ném lỗi  RAISERROR('Login không tồn tại trong SQL Server', 16, 1);  END  -- Nếu tất cả thành công, commit giao dịch  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  -- Nếu có lỗi xảy ra, rollback giao dịch  ROLLBACK TRANSACTION;  -- Xử lý lỗi  PRINT 'Lỗi: ' + ERROR\_MESSAGE();  -- Thông báo lỗi cho người dùng hoặc ghi log lỗi tùy theo nhu cầu  THROW;  END CATCH  END; |

**Xử lý bên C#:**

|  |
| --- |
| private void btnXacnhan\_Click(object sender, EventArgs e)  {  string tenTK = txtTentaikhoan.Text.Trim();  string matKhauMoi = txtMatkhaumoi.Text;  string xacNhan = txtXacnhanmatkhau.Text;  // Kiểm tra nhập liệu  if (string.IsNullOrEmpty(tenTK) || string.IsNullOrEmpty(matKhauMoi) || string.IsNullOrEmpty(xacNhan))  {  MessageBox.Show("Vui lòng nhập đầy đủ thông tin!");  return;  }  if (matKhauMoi != xacNhan)  {  MessageBox.Show("Mật khẩu xác nhận không khớp!");  return;  }  try  {    Database db = new Database();  SqlParameter[] parameters = new SqlParameter[]  {  new SqlParameter("@TenTK", tenTK),  new SqlParameter("@MatKhauMoi", matKhauMoi)  };  db.ExecuteStoredProc("sp\_QuenMatKhau", parameters);  MessageBox.Show("Đổi mật khẩu thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  this.Close();    }  catch (Exception ex)  {  MessageBox.Show(" Lỗi đăng nhập: " + ex.Message);  }    } |

## 2.4 Tìm kiếm tài khoản

|  |
| --- |
| CREATE PROCEDURE sp\_TimKiemTaiKhoan  @Search NVARCHAR(100)  AS  BEGIN  SELECT \*  FROM TaiKhoan  WHERE TenTK LIKE '%' + @Search + '%'  END |

**Xử lý bên C#**

|  |
| --- |
| private void txtSearch\_TextChanged(object sender, EventArgs e)  {  string searchText = txtSearch.Text.Trim();  // Tạo tham số truyền vào thủ tục  SqlParameter[] parameters = {  new SqlParameter("@Search", searchText)  };  Database db = new Database();  dgvTaiKhoan.DataSource = db.ExecuteProcToTable("sp\_TimKiemTaiKhoan", parameters);  } |

## 2.5 Thêm nhân viên

|  |
| --- |
| CREATE PROCEDURE sp\_ThemNhanVien  @HoTen NVARCHAR(100),  @DiaChi NVARCHAR(200),  @NgaySinh DATE,  @Email NVARCHAR(100),  @SDT NVARCHAR(15),  @MaTK INT,  @MaCV INT  AS  BEGIN  IF NOT EXISTS (SELECT 1 FROM TaiKhoan WHERE MaTK = @MaTK)  RAISERROR(N'Tài khoản không tồn tại', 16, 1);  ELSE IF NOT EXISTS (SELECT 1 FROM CongViec WHERE MaCV = @MaCV)  RAISERROR(N'Công việc không tồn tại', 16, 1);  ELSE  INSERT INTO NhanVien(HoTen, DiaChi, NgaySinh, Email, SDT, MaTK, MaCV)  VALUES (@HoTen, @DiaChi, @NgaySinh, @Email, @SDT, @MaTK, @MaCV);  END; |

**Xử lý bên C#:**

|  |
| --- |
| private void btnThem\_Click(object sender, EventArgs e)  {  // Kiểm tra các trường nhập liệu  if (string.IsNullOrEmpty(txtHoten.Text) || string.IsNullOrEmpty(txtDiachi.Text) || string.IsNullOrEmpty(txtEmail.Text))  {  MessageBox.Show("Vui lòng nhập đầy đủ thông tin!");  return;  }  SqlParameter[] parameters = {  new SqlParameter("@HoTen", txtHoten.Text),  new SqlParameter("@DiaChi", txtDiachi.Text),  new SqlParameter("@NgaySinh", dtpNgaysinh.Value),  new SqlParameter("@Email", txtEmail.Text),  new SqlParameter("@SDT", txtSoDT.Text),  new SqlParameter("@MaTK", txtMaTK.Text),  new SqlParameter("@MaCV", txtMaCV.Text)  };  try  {  Database db = new Database();  int rows = db.ExecuteStoredProc("sp\_ThemNhanVien", parameters);  if (rows > 0)  {  MessageBox.Show(" Thêm nhân viên thành công!");  LoadDanhSachSinhVien(); // Tải lại danh sách nhân viên  LoadDanhSachNhanVienTheoCongViec(cboTencongviec.SelectedItem.ToString());  ClearForm();  }  else  {  MessageBox.Show(" Thêm nhân viên thất bại.");  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message);  }  } |

## 2.6 Xóa nhân viên

|  |
| --- |
| CREATE PROCEDURE sp\_XoaNhanVien  @MaNV INT  AS  BEGIN  -- Kiểm tra xem nhân viên có tồn tại không  IF EXISTS (SELECT 1 FROM NhanVien WHERE MaNV = @MaNV)  BEGIN  -- Xóa thông tin chấm công của nhân viên  DELETE FROM ChamCong WHERE MaNV = @MaNV;  -- Xóa thông tin lương của nhân viên  DELETE FROM Luong WHERE MaNV = @MaNV;  -- Xóa nhân viên  DELETE FROM NhanVien WHERE MaNV = @MaNV;  END  ELSE  BEGIN  RAISERROR(N'Nhân viên không tồn tại!', 16, 1);  END  END; |

**Xử lý bên C#:**

|  |
| --- |
| private void btnXoa\_Click(object sender, EventArgs e)  {  if (dgvNhanvien.SelectedRows.Count == 0)  {  MessageBox.Show(" Vui lòng chọn nhân viên cần xóa.");  return;  }  int maNV = Convert.ToInt32(dgvNhanvien.SelectedRows[0].Cells["MaNV"].Value);  DialogResult result = MessageBox.Show("Bạn có chắc muốn xóa nhân viên này?", "Xác nhận", MessageBoxButtons.YesNo);  if (result == DialogResult.Yes)  {  SqlParameter[] parameters = {  new SqlParameter("@MaNV", maNV)  };  try  {  Database db = new Database();  int rows = db.ExecuteStoredProc("sp\_XoaNhanVien", parameters);  if (rows > 0)  {  MessageBox.Show(" Xóa nhân viên thành công!");  LoadDanhSachSinhVien(); // Tải lại danh sách nhân viên  LoadDanhSachNhanVienTheoCongViec(cboTencongviec.SelectedItem.ToString());  ClearForm();  }  else  {  MessageBox.Show(" Xóa nhân viên thất bại.");  }  }  catch (Exception ex )  {  MessageBox.Show("Lỗi: " + ex.Message);  }  }  } |

## 2.7 Cập nhật nhân viên

|  |
| --- |
| CREATE PROCEDURE sp\_CapNhatNhanVien  @MaNV INT,  @HoTen NVARCHAR(100),  @DiaChi NVARCHAR(200),  @NgaySinh DATE,  @Email NVARCHAR(100),  @SDT NVARCHAR(15),  @MaCV INT  AS  BEGIN  -- Kiểm tra xem nhân viên có tồn tại không  IF EXISTS (SELECT 1 FROM NhanVien WHERE MaNV = @MaNV)  BEGIN  -- Kiểm tra công việc có tồn tại không  IF NOT EXISTS (SELECT 1 FROM CongViec WHERE MaCV = @MaCV)  BEGIN  RAISERROR(N'Công việc không tồn tại!', 16, 1);  END  ELSE  BEGIN  -- Cập nhật thông tin nhân viên  UPDATE NhanVien  SET HoTen = @HoTen,  DiaChi = @DiaChi,  NgaySinh = @NgaySinh,  Email = @Email,  SDT = @SDT,  MaCV = @MaCV  WHERE MaNV = @MaNV;  END  END  ELSE  BEGIN  RAISERROR(N'Nhân viên không tồn tại!', 16, 1);  END  END; |

**Xử lý bên C#:**

|  |
| --- |
| private void btnSua\_Click(object sender, EventArgs e)  {  // Kiểm tra các trường nhập liệu  if (string.IsNullOrEmpty(txtHoten.Text) || string.IsNullOrEmpty(txtDiachi.Text) || string.IsNullOrEmpty(txtEmail.Text))  {  MessageBox.Show("Vui lòng nhập đầy đủ thông tin!");  return;  }  SqlParameter[] parameters = {  new SqlParameter("@MaNV", dgvNhanvien.SelectedRows[0].Cells["MaNV"].Value),  new SqlParameter("@HoTen", txtHoten.Text),  new SqlParameter("@DiaChi", txtDiachi.Text),  new SqlParameter("@NgaySinh", dtpNgaysinh.Value),  new SqlParameter("@Email", txtEmail.Text),  new SqlParameter("@SDT", txtSoDT.Text),  new SqlParameter("@MaCV", txtMaCV.Text)  };  try  {  Database db = new Database();  int rows = db.ExecuteStoredProc("sp\_CapNhatNhanVien", parameters);  if (rows > 0)  {  MessageBox.Show(" Cập nhật thông tin nhân viên thành công!");  LoadDanhSachSinhVien(); // Tải lại danh sách nhân viên  LoadDanhSachNhanVienTheoCongViec(cboTencongviec.SelectedItem.ToString());  ClearForm();  }  else  {  MessageBox.Show(" Cập nhật thất bại.");  }  }  catch (Exception ex )  {  MessageBox.Show("Lỗi: " + ex.Message);  }  } |

## 2.8 Tìm kiếm nhân viên

|  |
| --- |
| CREATE PROCEDURE sp\_TimKiemNhanVien  @Search NVARCHAR(100)  AS  BEGIN  SELECT \* FROM vw\_NhanVien  WHERE HoTen LIKE '%' + @Search + '%'  OR Email LIKE '%' + @Search + '%'  OR SDT LIKE '%' + @Search + '%'  END |

**Xử lý bên C#**

|  |
| --- |
| private void txtSearch\_TextChanged(object sender, EventArgs e)  {  string searchText = txtSearch.Text.Trim();  SqlParameter[] parameters = {  new SqlParameter("@Search", searchText)  };  Database db = new Database();  dgvNhanvien.DataSource = db.ExecuteProcToTable("sp\_TimKiemNhanVien", parameters);  } |

## 2.9 Thêm công việc

|  |
| --- |
| CREATE PROCEDURE sp\_ThemCongViec  @TenCV NVARCHAR(50),  @LuongCoBan DECIMAL(18,2)  AS  BEGIN  INSERT INTO CongViec (TenCV, LuongCoBan)  VALUES (@TenCV, @LuongCoBan);  END; |

**Xử lý bên C#:**

|  |
| --- |
| private void btnThemcongviec\_Click(object sender, EventArgs e)  {  if (string.IsNullOrEmpty(txtTencongviec.Text) || string.IsNullOrEmpty(txtLuongcoban.Text))  {  MessageBox.Show("Vui lòng nhập đầy đủ thông tin công việc!");  return;  }  string sql = "INSERT INTO CongViec (TenCV, LuongCoBan) VALUES (@TenCV, @LuongCoBan)";  SqlParameter[] parameters = {  new SqlParameter("@TenCV", txtTencongviec.Text),  new SqlParameter("@LuongCoBan", Convert.ToDecimal(txtLuongcoban.Text))  };  try  {  Database db = new Database();  int rows = db.ExecuteNonQuery(sql, parameters);  if (rows > 0)  {  MessageBox.Show("Thêm công việc thành công!");  LoadDanhSachCongViec(); // Tải lại danh sách công việc  ClearForm();  }  else  {  MessageBox.Show("Thêm công việc thất bại.");  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message);  }  } |

## 2.10 Xóa công việc

|  |
| --- |
| CREATE PROCEDURE sp\_XoaCongViec  @MaCV INT  AS  BEGIN  -- Kiểm tra xem công việc có đang được sử dụng không  IF EXISTS (SELECT 1 FROM NhanVien WHERE MaCV = @MaCV)  BEGIN  RAISERROR(N'Không thể xóa công việc vì có nhân viên đang làm việc!', 16, 1);  END  ELSE  BEGIN  DELETE FROM CongViec WHERE MaCV = @MaCV;  END  END; |

**Xử lý bên C#:**

|  |
| --- |
| private void btnXoacongviec\_Click(object sender, EventArgs e)  {  if (dgvDanhsachcongviec.SelectedRows.Count == 0)  {  MessageBox.Show(" Vui lòng chọn công việc cần xóa.");  return;  }  int maCV = Convert.ToInt32(dgvDanhsachcongviec.SelectedRows[0].Cells["MaCV"].Value);  DialogResult result = MessageBox.Show("Bạn có chắc muốn xóa công việc này?", "Xác nhận", MessageBoxButtons.YesNo);  if (result == DialogResult.Yes)  {  SqlParameter[] parameters = {  new SqlParameter("@MaCV", maCV)  };  try  {  Database db = new Database();  int rows = db.ExecuteStoredProc("sp\_XoaCongViec", parameters);  if (rows > 0)  {  MessageBox.Show(" Xóa công việc thành công!");  LoadDanhSachCongViec(); // Tải lại danh sách công việc  ClearForm();  }  else  {  MessageBox.Show(" Xóa công việc thất bại.");  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message);  }  }  } |

## 2.11 Cập nhật công việc

|  |
| --- |
| CREATE PROCEDURE sp\_CapNhatCongViec  @MaCV INT,  @TenCV NVARCHAR(50),  @LuongCoBan DECIMAL(18,2)  AS  BEGIN  UPDATE CongViec  SET TenCV = @TenCV, LuongCoBan = @LuongCoBan  WHERE MaCV = @MaCV;  END; |

**Xử lý bên C#:**

|  |
| --- |
| private void btnCapnhatcongviec\_Click(object sender, EventArgs e)  {  if (string.IsNullOrEmpty(txtTencongviec.Text) || string.IsNullOrEmpty(txtLuongcoban.Text))  {  MessageBox.Show("Vui lòng nhập đầy đủ thông tin công việc!");  return;  }  SqlParameter[] parameters = {  new SqlParameter("@MaCV", dgvDanhsachcongviec.SelectedRows[0].Cells["MaCV"].Value),  new SqlParameter("@TenCV", txtTencongviec.Text),  new SqlParameter("@LuongCoBan", Convert.ToDecimal(txtLuongcoban.Text))  };  try  {  Database db = new Database();  int rows = db.ExecuteStoredProc("sp\_CapNhatCongViec", parameters);  if (rows > 0)  {  MessageBox.Show("Cập nhật công việc thành công!");  LoadDanhSachCongViec(); // Tải lại danh sách công việc  ClearForm();  }  else  {  MessageBox.Show("Cập nhật công việc thất bại.");  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message);  }  } |

## 2.12 Tìm kiếm công việc

|  |
| --- |
| CREATE PROCEDURE sp\_TimKiemCongViec  @Search NVARCHAR(100)  AS  BEGIN  SELECT \*  FROM CongViec  WHERE TenCV LIKE '%' + @Search + '%'  END |

**Xử lý bên C#**

|  |
| --- |
| private void txtSearch\_TextChanged(object sender, EventArgs e)  {  string searchText = txtSearch.Text.Trim();  // Tạo tham số truyền vào thủ tục  SqlParameter[] parameters = {  new SqlParameter("@Search", searchText)  };  Database db = new Database();  dgvDanhsachcongviec.DataSource = db.ExecuteProcToTable("sp\_TimKiemCongViec", parameters);  } |

## 2.13 Chấm công

|  |
| --- |
| CREATE PROCEDURE sp\_ChamCong  @MaNV INT,  @Ngay DATE,  @GioVao TIME,  @GioRa TIME,  @SoGioTangCa INT  AS  BEGIN  -- Chèn thông tin chấm công vào bảng ChamCong  INSERT INTO ChamCong (MaNV, Ngay, GioVao, GioRa, SoGioTangCa)  VALUES (@MaNV, @Ngay, @GioVao, @GioRa, @SoGioTangCa);  END; |

**Xử lý bên C#:**

|  |
| --- |
| private void btnChamCong\_Click(object sender, EventArgs e)  {  // Lấy giá trị từ các điều khiển trên form  int maNV = Convert.ToInt32(txtManhanvien.Text); // Mã nhân viên  DateTime ngay = dtpNgay.Value; // Ngày  TimeSpan gioVao = dtpGiovao.Value.TimeOfDay; // Giờ vào  TimeSpan gioRa = dtpGiora.Value.TimeOfDay; // Giờ ra  int soGioTangCa = Convert.ToInt32(txtSogiotangca.Text); // Số giờ tăng ca  // Gọi thủ tục Chấm công  SqlParameter[] parameters = {  new SqlParameter("@MaNV", maNV),  new SqlParameter("@Ngay", ngay),  new SqlParameter("@GioVao", gioVao),  new SqlParameter("@GioRa", gioRa),  new SqlParameter("@SoGioTangCa", soGioTangCa)  };  try  {  Database db = new Database();  db.ExecuteStoredProc("sp\_ChamCong", parameters);  MessageBox.Show("Chấm công thành công!");  // Làm mới lại bảng chấm công sau khi chấm công thành công  LoadChamCong();  ClearForm();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi chấm công: " + ex.Message);  }  } |

## 2.14 Xóa bảng ghi chấm công

|  |
| --- |
| CREATE PROCEDURE sp\_XoaBangGhiChamCong  @MaNV INT,  @Ngay DATE  AS  BEGIN  DELETE FROM ChamCong  WHERE MaNV = @MaNV AND Ngay = @Ngay;  END; |

**Xử lý bên C#**

|  |
| --- |
| private void btnXoabangghi\_Click(object sender, EventArgs e)  {    int maNV = Convert.ToInt32(txtManhanvien.Text);  DateTime ngay = dtpNgay.Value; // Ngày    DialogResult dialogResult = MessageBox.Show("Bạn có chắc chắn muốn xóa bản ghi này?", "Xác nhận", MessageBoxButtons.YesNo);  if (dialogResult == DialogResult.Yes)  {  // Thực thi thủ tục xóa bản ghi chấm công  SqlParameter[] parameters = {  new SqlParameter("@MaNV", maNV),  new SqlParameter("@Ngay", ngay)  };  try  {  Database db = new Database();  db.ExecuteStoredProc("sp\_XoaBangGhiChamCong", parameters);  MessageBox.Show("Xóa bảng ghi thành công!");    LoadChamCong();  ClearForm();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi xóa bảng ghi: " + ex.Message);  }  }  } |

## 2.15 Xóa tất cả chấm công

|  |
| --- |
| private void btnClear\_Click(object sender, EventArgs e)  {    DialogResult dialogResult = MessageBox.Show("Bạn có chắc chắn muốn xóa toàn bộ bảng ghi chấm công?", "Xác nhận", MessageBoxButtons.YesNo);  if (dialogResult == DialogResult.Yes)  {  try  {    Database db = new Database();  db.ExecuteStoredProc("sp\_XoaTatCaChamCong", null);  MessageBox.Show("Xóa toàn bộ bảng ghi thành công!");    LoadChamCong();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi xóa toàn bộ bảng ghi: " + ex.Message);  }  }  } |

## 2.16 Tính lương

|  |
| --- |
| CREATE PROCEDURE sp\_TinhLuong  @MaNV INT,  @Thuong DECIMAL(18, 2),  @PhuCap DECIMAL(18, 2)  AS  BEGIN  DECLARE @LuongCoBan DECIMAL(18, 2), @SoGioLam INT, @LuongTangCa DECIMAL(18, 2), @TongLuong DECIMAL(18, 2);  -- Bắt đầu transaction  BEGIN TRY  BEGIN TRANSACTION;  -- Lấy Lương cơ bản từ bảng CongViec (lương theo giờ của công việc)  SELECT @LuongCoBan = CV.LuongCoBan  FROM CongViec CV  JOIN NhanVien NV ON NV.MaCV = CV.MaCV  WHERE NV.MaNV = @MaNV;  -- Lấy số giờ làm việc thực tế của nhân viên trong tháng từ bảng ChamCong  SELECT @SoGioLam = SUM(DATEDIFF(HOUR, GioVao, GioRa))  FROM ChamCong  WHERE MaNV = @MaNV;  -- Tính Lương tăng ca (Số giờ tăng ca \* tỷ lệ 1.5 \* lương theo giờ)  SELECT @LuongTangCa = SUM(SoGioTangCa \* 1.5 \* @LuongCoBan)  FROM ChamCong  WHERE MaNV = @MaNV;  -- Tính Tổng lương (Lương cơ bản + Thưởng + Phụ cấp + Lương tăng ca)  SET @TongLuong = (@LuongCoBan \* @SoGioLam) + @Thuong + @PhuCap + ISNULL(@LuongTangCa, 0);  -- Kiểm tra xem bản ghi lương đã có chưa, nếu chưa thì thêm mới  IF NOT EXISTS (SELECT 1 FROM Luong WHERE MaNV = @MaNV)  BEGIN  INSERT INTO Luong (MaNV, Thuong, PhuCap, LuongTangCa, TongLuong)  VALUES (@MaNV, @Thuong, @PhuCap, @LuongTangCa, @TongLuong);  END  ELSE  BEGIN  -- Nếu đã có bản ghi, cập nhật thông tin  UPDATE Luong  SET Thuong = @Thuong, PhuCap = @PhuCap, LuongTangCa = @LuongTangCa, TongLuong = @TongLuong  WHERE MaNV = @MaNV;  END  -- Commit transaction nếu tất cả các thao tác thành công  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  -- Nếu có lỗi xảy ra, rollback transaction  ROLLBACK TRANSACTION;  -- Thông báo lỗi  DECLARE @ErrorMessage NVARCHAR(4000), @ErrorSeverity INT, @ErrorState INT;  SELECT @ErrorMessage = ERROR\_MESSAGE(),  @ErrorSeverity = ERROR\_SEVERITY(),  @ErrorState = ERROR\_STATE();  RAISERROR(@ErrorMessage, @ErrorSeverity, @ErrorState);  END CATCH  END; |

**Xử lý bên C#:**

|  |
| --- |
| private void btnthemcapnhatluong\_Click(object sender, EventArgs e)  {  try  {  // Lấy giá trị từ các TextBox  int maNV = Convert.ToInt32(txtManhanvien.Text); // Mã nhân viên  decimal thuong = Convert.ToDecimal(txtThuong.Text); // Thưởng  decimal phuCap = Convert.ToDecimal(txtPhucap.Text); // Phụ cấp  // Gọi thủ tục TinhLuong từ cơ sở dữ liệu  SqlParameter[] parameters = {  new SqlParameter("@MaNV", maNV),  new SqlParameter("@Thuong", thuong),  new SqlParameter("@PhuCap", phuCap)  };  Database db = new Database();  db.ExecuteStoredProc("sp\_TinhLuong", parameters);  // Làm mới bảng lương sau khi thêm hoặc cập nhật lương  LoadLuong();  TinhTongLuong();  Clear();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi thêm/cập nhật lương: " + ex.Message);  }  } |

## 2.17 Xóa lương

|  |
| --- |
| CREATE PROCEDURE sp\_XoaLuong  @MaNV INT  AS  BEGIN  -- Xóa bản ghi trong bảng Luong dựa trên MaNV  DELETE FROM Luong  WHERE MaNV = @MaNV;    -- Trả về thông báo hoặc số lượng bản ghi bị xóa (tùy theo yêu cầu)  SELECT 'Xóa lương thành công' AS Message;  END; |

**Xử lý bên C#:**

|  |
| --- |
| private void btnXoaluong\_Click(object sender, EventArgs e)  {  try  {  // Lấy Mã nhân viên từ TextBox  int maNV = Convert.ToInt32(txtManhanvien.Text);  // Gọi thủ tục XoaLuong từ cơ sở dữ liệu  SqlParameter[] parameters = {  new SqlParameter("@MaNV", maNV)  };  Database db = new Database();  db.ExecuteStoredProc("sp\_XoaLuong", parameters); // Thực thi thủ tục xóa lương  // Làm mới bảng lương sau khi xóa  LoadLuong();  TinhTongLuong();  Clear();  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi xóa lương: " + ex.Message);  }  } |

## 2.18 Tính tổng lương phải trả cho tất cả nhân viên

|  |
| --- |
| CREATE FUNCTION fn\_TinhTongLuong()  RETURNS DECIMAL(18, 2)  AS  BEGIN  DECLARE @TongLuongToanBo DECIMAL(18, 2);  -- Tính tổng tất cả các tổng lương trong bảng Luong  SELECT @TongLuongToanBo = SUM(TongLuong)  FROM Luong;  -- Trả về tổng lương  RETURN @TongLuongToanBo;  END; |

**Xử lý bên C#:**

|  |
| --- |
| private void TinhTongLuong()  {  try  {  // Gọi hàm TinhTongLuong từ cơ sở dữ liệu để tính tổng lương  string sql = "SELECT dbo.fn\_TinhTongLuong() AS TongLuong";  Database db = new Database();  var result = db.ExecuteQuery(sql); // Thực thi câu lệnh SQL gọi hàm  // Lấy tổng lương từ kết quả trả về  decimal tongLuong = Convert.ToDecimal(result.Rows[0]["TongLuong"]);  // Hiển thị tổng lương trong TextBox  txtTongluongphaitra.Text = tongLuong.ToString("C");  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi tính tổng lương: " + ex.Message);  }  } |

## 2.19 Xem số nhân viên theo công việc

|  |
| --- |
| CREATE FUNCTION fn\_ThongKeSoNhanVienTheoCongViec()  RETURNS TABLE  AS  RETURN  (  SELECT  CV.MaCV,  CV.TenCV,  COUNT(NV.MaNV) AS SoNhanVien  FROM CongViec CV  LEFT JOIN NhanVien NV ON NV.MaCV = CV.MaCV  GROUP BY CV.MaCV, CV.TenCV  ); |

**Xử lý bên C#:**

|  |
| --- |
| private void LoadThongKeSoNhanVienTheoCongViec()  {  string sql = "SELECT \* FROM dbo.fn\_ThongKeSoNhanVienTheoCongViec()";  Database db = new Database();  dgvSonhanvientheocongviec.DataSource = db.ExecuteQuery(sql);  } |

## 2.20 Xem danh sách nhân viên theo tên công việc

|  |
| --- |
| CREATE FUNCTION fn\_DanhSachNhanVienTheoTenCongViec(@TenCV NVARCHAR(50))  RETURNS TABLE  AS  RETURN  (  SELECT NV.MaNV, NV.HoTen, NV.Email, CV.TenCV, CV.LuongCoBan  FROM NhanVien NV  JOIN CongViec CV ON NV.MaCV = CV.MaCV  WHERE CV.TenCV = @TenCV  ); |

**Xử lý bên C#:**

|  |
| --- |
| private void LoadDanhSachNhanVienTheoCongViec(string tenCV)  {  string sql = "SELECT \* FROM dbo.fn\_DanhSachNhanVienTheoTenCongViec(@TenCV)";  SqlParameter[] parameters = { new SqlParameter("@TenCV", tenCV) };  Database db = new Database();  dgvNhanvientheocongviec.DataSource = db.ExecuteQuery(sql, parameters);  } |

# Chức năng của Nhân viên

## 3.1 Quên mật khẩu

*Chức năng này tương tự với chức năng của Quản lý*

|  |
| --- |
| CREATE PROCEDURE sp\_QuenMatKhau  @TenTK NVARCHAR(50),  @MatKhauMoi NVARCHAR(255)  AS  BEGIN  SET NOCOUNT ON;  -- Bắt đầu giao dịch  BEGIN TRANSACTION;  BEGIN TRY  -- 1. Cập nhật mật khẩu trong bảng TaiKhoan  UPDATE TaiKhoan  SET MatKhau = @MatKhauMoi  WHERE TenTK = @TenTK;  -- 2. Cập nhật mật khẩu trong SQL Server Login  IF EXISTS (SELECT \* FROM sys.server\_principals WHERE name = @TenTK)  BEGIN  -- Cập nhật mật khẩu cho Login trong SQL Server  DECLARE @sqlLogin NVARCHAR(MAX);  SET @sqlLogin = 'ALTER LOGIN [' + @TenTK + '] WITH PASSWORD = ''' + @MatKhauMoi + ''';';  EXEC(@sqlLogin);  END  ELSE  BEGIN  -- Nếu Login không tồn tại trong SQL Server, ném lỗi  RAISERROR('Login không tồn tại trong SQL Server', 16, 1);  END  -- Nếu tất cả thành công, commit giao dịch  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  -- Nếu có lỗi xảy ra, rollback giao dịch  ROLLBACK TRANSACTION;  -- Xử lý lỗi  PRINT 'Lỗi: ' + ERROR\_MESSAGE();  -- Thông báo lỗi cho người dùng hoặc ghi log lỗi tùy theo nhu cầu  THROW;  END CATCH  END; |

**Xử lý bên C#:**

|  |
| --- |
| private void btnXacnhan\_Click(object sender, EventArgs e)  {  string tenTK = txtTentaikhoan.Text.Trim();  string matKhauMoi = txtMatkhaumoi.Text;  string xacNhan = txtXacnhanmatkhau.Text;  // Kiểm tra nhập liệu  if (string.IsNullOrEmpty(tenTK) || string.IsNullOrEmpty(matKhauMoi) || string.IsNullOrEmpty(xacNhan))  {  MessageBox.Show("Vui lòng nhập đầy đủ thông tin!");  return;  }  if (matKhauMoi != xacNhan)  {  MessageBox.Show("Mật khẩu xác nhận không khớp!");  return;  }  try  {    Database db = new Database();  SqlParameter[] parameters = new SqlParameter[]  {  new SqlParameter("@TenTK", tenTK),  new SqlParameter("@MatKhauMoi", matKhauMoi)  };  db.ExecuteStoredProc("sp\_QuenMatKhau", parameters);  MessageBox.Show("Đổi mật khẩu thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  this.Close();    }  catch (Exception ex)  {  MessageBox.Show("⚠ Lỗi đăng nhập: " + ex.Message);  }    } |

## 3.2 Xem thông tin lương của mình

|  |
| --- |
| CREATE FUNCTION dbo.fn\_HienThiThongTinLuong  (@MaNV INT)  RETURNS @LuongTable TABLE  (  [Mã nhân viên] INT,  [Họ tên] NVARCHAR(100),  [Ngày chấm công] DATE,  [Giờ vào] TIME,  [Giờ ra] TIME,  [Số giờ làm] INT,  [Số giờ tăng ca] INT,  [Lương thưởng] DECIMAL(18, 2),  [Lương phụ cấp] DECIMAL(18, 2),  [Lương tăng ca] DECIMAL(18, 2),  [Tổng lương] DECIMAL(18, 2)  )  AS  BEGIN  INSERT INTO @LuongTable  SELECT  NV.MaNV,  NV.HoTen,  CC.Ngay,  CC.GioVao,  CC.GioRa,  DATEDIFF(HOUR, CC.GioVao, CC.GioRa) AS SoGioLam,  CC.SoGioTangCa,  L.Thuong,  L.PhuCap,  L.LuongTangCa,  L.TongLuong  FROM  NhanVien NV  JOIN  ChamCong CC ON NV.MaNV = CC.MaNV  JOIN  Luong L ON NV.MaNV = L.MaNV  WHERE  NV.MaNV = @MaNV;  RETURN;  END; |

**Xử lý bên C#:**

|  |
| --- |
| private void UC\_Xembangluong\_Load(object sender, EventArgs e)  {  // Lấy MaNV từ bảng NhanVien dựa trên MaTK (tài khoản đăng nhập)  int maNV = GetMaNVByMaTK(GlobalState.MaTK);  // Truy vấn bảng lương với MaNV đã lấy được  string sql = "SELECT \* FROM dbo.fn\_HienThiThongTinLuong(@MaNV)";  SqlParameter[] parameters = { new SqlParameter("@MaNV", maNV) };  Database db = new Database();  dgvXembangluong.DataSource = db.ExecuteQuery(sql, parameters);  } |

## 3.3 Xem hồ sơ nhân viên của mình

|  |
| --- |
| CREATE FUNCTION dbo.fn\_Hosonhanvien (@MaTK INT)  RETURNS @HoSoTable TABLE  (  [Tên Tài Khoản] NVARCHAR(100),  [Mật khẩu] NVARCHAR(100),  [Họ tên] NVARCHAR(100),  [Địa chỉ] NVARCHAR(255),  [Ngày sinh] DATE,  [Email] NVARCHAR(100),  [Số điện thoại] NVARCHAR(20),  [Tên Công Việc] NVARCHAR(100)  )  AS  BEGIN  INSERT INTO @HoSoTable  SELECT  TK.TenTK,  TK.MatKhau,  NV.HoTen,  NV.DiaChi,  NV.NgaySinh,  NV.Email,  NV.SDT,  CV.TenCV  FROM  TaiKhoan TK  JOIN  NhanVien NV ON TK.MaTK = NV.MaTK  JOIN  CongViec CV ON NV.MaCV = CV.MaCV  WHERE  TK.MaTK = @MaTK;  RETURN;  END; |

**Xử lý bên C#:**

|  |
| --- |
| private void UC\_Hosonhanvien\_Load(object sender, EventArgs e)  {  try  {  // Lấy MaNV từ MaTK của tài khoản đang đăng nhập  int maNV = GetMaNVByMaTK(GlobalState.MaTK);  // Nếu không tìm thấy MaNV, thông báo lỗi  if (maNV == 0)  {  MessageBox.Show("Không tìm thấy thông tin nhân viên");  return;  }  // Gọi hàm Hosocanhan để lấy dữ liệu  string sql = "SELECT \* FROM dbo.fn\_Hosonhanvien(@MaTK)";  SqlParameter[] parameters = { new SqlParameter("@MaTK", GlobalState.MaTK) };  Database db = new Database();  DataTable result = db.ExecuteQuery(sql, parameters);  // Kiểm tra nếu có kết quả trả về  if (result.Rows.Count > 0)  {  // Lấy thông tin từ DataTable và gán vào các Label  DataRow row = result.Rows[0]; // Dữ liệu trả về từ hàm Hosocanhan  lblTentaikhoan.Text = row["Tên Tài Khoản"].ToString();  lblMatkhau.Text = row["Mật khẩu"].ToString();  lblHoten.Text = row["Họ tên"].ToString();  lblDiachi.Text = row["Địa chỉ"].ToString();  lblNgaysinh.Text = Convert.ToDateTime(row["Ngày sinh"]).ToString("dd/MM/yyyy");  lblEmail.Text = row["Email"].ToString();  lblSodienthoai.Text = row["Số điện thoại"].ToString();  lblTencongviec.Text = row["Tên Công Việc"].ToString();  }  else  {  MessageBox.Show("Không có thông tin lương của nhân viên");  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi: " + ex.Message);  }  } |

# CHƯƠNG 4: PHÂN QUYỀN

# Tạo Role và gán quyền

* Hệ thống quản lý nhân viên cửa hàng bán quần áo có 2 role chính là role Quản lý và role Nhân viên
* Bên SQL tiến hành tạo role và phân quyền tương ứng.

## 1.1 Quản lý

|  |
| --- |
| CREATE ROLE role\_QuanLy;  -- PHÂN QUYỀN CHO ROLE: QuanLy  -- Toàn quyền  GRANT SELECT, INSERT, UPDATE, DELETE ON NhanVien TO role\_QuanLy;  GRANT SELECT, INSERT, UPDATE, DELETE ON CongViec TO role\_QuanLy;  GRANT SELECT, INSERT, UPDATE, DELETE ON ChamCong TO role\_QuanLy;  GRANT SELECT, INSERT, UPDATE, DELETE ON TaiKhoan TO role\_QuanLy;  GRANT SELECT, INSERT, UPDATE, DELETE ON Luong TO role\_QuanLy;  -- Các thủ tục  GRANT EXECUTE ON sp\_ThemTaiKhoan TO role\_QuanLy;  GRANT EXECUTE ON sp\_XoaTaiKhoan TO role\_QuanLy;  GRANT EXECUTE ON sp\_QuenMatKhau TO role\_QuanLy;  GRANT EXECUTE ON sp\_TimKiemTaiKhoan TO role\_QuanLy;  GRANT EXECUTE ON sp\_ThemNhanVien TO role\_QuanLy;  GRANT EXECUTE ON sp\_XoaNhanVien TO role\_QuanLy;  GRANT EXECUTE ON sp\_CapNhatNhanVien TO role\_QuanLy;  GRANT EXECUTE ON sp\_TimKiemNhanVien TO role\_QuanLy;  GRANT EXECUTE ON sp\_ThemCongViec TO role\_QuanLy;  GRANT EXECUTE ON sp\_XoaCongViec TO role\_QuanLy;  GRANT EXECUTE ON sp\_CapNhatCongViec TO role\_QuanLy;  GRANT EXECUTE ON sp\_TimKiemCongViec TO role\_QuanLy;  GRANT EXECUTE ON sp\_ChamCong TO role\_QuanLy;  GRANT EXECUTE ON sp\_XoaLuong TO role\_QuanLy;  GRANT EXECUTE ON sp\_TinhLuong TO role\_QuanLy;  -- Các hàm trả về giá trị  GRANT EXECUTE ON fn\_TinhTongLuong TO role\_QuanLy;  GRANT EXECUTE ON fn\_TinhTongLuongTheoMaNV TO role\_QuanLy;  -- Các hàm trả về bảng  GRANT SELECT ON fn\_ThongKeSoNhanVienTheoCongViec TO role\_QuanLy;  GRANT SELECT ON fn\_DanhSachNhanVienTheoTenCongViec TO role\_QuanLy; |

## 1.2 Nhân viên

|  |
| --- |
| CREATE ROLE role\_NhanVien;  -- PHÂN QUYỀN CHO ROLE: NhanVien  GRANT SELECT ON ChamCong TO role\_NhanVien;  GRANT SELECT ON NhanVien TO role\_NhanVien;  GRANT SELECT ON Luong TO role\_NhanVien;  GRANT UPDATE ON TaiKhoan TO role\_NhanVien;  --Thủ tục  GRANT EXECUTE ON sp\_QuenMatKhau TO role\_NhanVien;  -- Các hàm  GRANT SELECT ON fn\_HienThiThongTinLuong TO role\_NhanVien;  GRANT SELECT ON fn\_Hosonhanvien TO role\_NhanVien; |

## Login bên C#

|  |
| --- |
| string tenTK = txtTenTK.Text.Trim();  string matKhau = txtMatKhau.Text.Trim();  if (string.IsNullOrEmpty(tenTK) || string.IsNullOrEmpty(matKhau))  {  MessageBox.Show(" Vui lòng nhập đầy đủ tên tài khoản và mật khẩu!");  return;  }  string connDefault = @"Data Source=MINH\_NHUT\MINH\_NHUT;Initial Catalog=QL\_NhanVien;Integrated Security=True";  try  {  using (SqlConnection conn = new SqlConnection(connDefault))  {  string sql = "SELECT \* FROM TaiKhoan WHERE TenTK = @TenTK AND MatKhau = @MatKhau";  SqlCommand cmd = new SqlCommand(sql, conn);  cmd.Parameters.AddWithValue("@TenTK", tenTK);  cmd.Parameters.AddWithValue("@MatKhau", matKhau);  SqlDataAdapter da = new SqlDataAdapter(cmd);  DataTable dt = new DataTable();  da.Fill(dt);  if (dt.Rows.Count > 0)  {  // Đăng nhập thành công  GlobalState.TenTaiKhoan = tenTK;  GlobalState.VaiTro = dt.Rows[0]["VaiTro"].ToString();  GlobalState.MaTK = Convert.ToInt32(dt.Rows[0]["MaTK"]);  // Gán ConnectionString trực tiếp bằng tài khoản SQL login vừa nhập  GlobalState.ConnectionString = $@"Server=MINH\_NHUT\MINH\_NHUT;Database=QL\_NhanVien;User Id={tenTK};Password={matKhau}";  frmMenu main = new frmMenu();  this.Hide();  main.ShowDialog();  this.Show();  }  else  {  MessageBox.Show(" Tài khoản hoặc mật khẩu không đúng!",  "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  }  }  catch (Exception ex)  {  MessageBox.Show(" Lỗi đăng nhập: " + ex.Message);  } |

# CHƯƠNG 5: TRANSACTION

# Transaction trong thủ tục sp\_ThemTaiKhoan

|  |
| --- |
| CREATE PROCEDURE sp\_ThemTaiKhoan  @TenTK NVARCHAR(50),  @MatKhau NVARCHAR(255),  @VaiTro NVARCHAR(20)  AS  BEGIN  SET NOCOUNT ON;  -- Bắt đầu giao dịch  BEGIN TRANSACTION;  BEGIN TRY  -- 1. Thêm vào bảng TaiKhoan  INSERT INTO TaiKhoan (TenTK, MatKhau, VaiTro)  VALUES (@TenTK, @MatKhau, @VaiTro);  -- 2. Tạo LOGIN nếu chưa có  IF NOT EXISTS (SELECT \* FROM sys.server\_principals WHERE name = @TenTK)  BEGIN  DECLARE @sqlLogin NVARCHAR(MAX);  SET @sqlLogin = 'CREATE LOGIN [' + @TenTK + '] WITH PASSWORD = ''' + @MatKhau + ''';';  EXEC(@sqlLogin);  END  -- 3. Tạo USER trong database nếu chưa có  IF NOT EXISTS (SELECT \* FROM sys.database\_principals WHERE name = @TenTK)  BEGIN  DECLARE @sqlUser NVARCHAR(MAX);  SET @sqlUser = 'CREATE USER [' + @TenTK + '] FOR LOGIN [' + @TenTK + '];';  EXEC(@sqlUser);  END  -- 4. Gán vào role tương ứng  IF @VaiTro = 'QuanLy'  BEGIN  EXEC sp\_addrolemember 'role\_QuanLy', @TenTK;  END  ELSE IF @VaiTro = 'NhanVien'  BEGIN  EXEC sp\_addrolemember 'role\_NhanVien', @TenTK;  END  -- Nếu tất cả các bước thành công, commit giao dịch  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  -- Nếu có lỗi xảy ra, rollback giao dịch  ROLLBACK TRANSACTION;  -- Thông báo lỗi  PRINT 'Lỗi: ' + ERROR\_MESSAGE();  THROW; -- Ném lại lỗi để có thể xử lý thêm ngoài thủ tục này nếu cần  END CATCH  END; |

# Transaction trong thủ tục sp\_XoaTaiKhoan

|  |
| --- |
| CREATE PROCEDURE sp\_XoaTaiKhoan  @MaTK INT  AS  BEGIN  SET NOCOUNT ON;  -- Bắt đầu giao dịch  BEGIN TRANSACTION;  DECLARE @TenTK NVARCHAR(50);  -- Lấy tên tài khoản từ MaTK  SELECT @TenTK = TenTK FROM TaiKhoan WHERE MaTK = @MaTK;  IF @TenTK IS NULL  BEGIN  -- Nếu không tìm thấy tài khoản, ném lỗi và rollback giao dịch  RAISERROR(N'Tài khoản không tồn tại!', 16, 1);  ROLLBACK TRANSACTION;  RETURN;  END  BEGIN TRY  -- 1. Xóa tài khoản trong bảng TaiKhoan  DELETE FROM TaiKhoan WHERE MaTK = @MaTK;  -- 2. Xóa USER trong database nếu tồn tại  IF EXISTS (SELECT 1 FROM sys.database\_principals WHERE name = @TenTK)  BEGIN  DECLARE @sqlDropUser NVARCHAR(MAX);  SET @sqlDropUser = 'DROP USER [' + @TenTK + '];';  EXEC(@sqlDropUser);  END  -- 3. Xóa LOGIN trong server nếu tồn tại  IF EXISTS (SELECT 1 FROM sys.server\_principals WHERE name = @TenTK)  BEGIN  DECLARE @sqlDropLogin NVARCHAR(MAX);  SET @sqlDropLogin = 'DROP LOGIN [' + @TenTK + '];';  EXEC(@sqlDropLogin);  END  -- Nếu mọi thao tác thành công, commit giao dịch  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  -- Nếu có lỗi xảy ra, rollback giao dịch  ROLLBACK TRANSACTION;  -- Thông báo lỗi  PRINT 'Lỗi: ' + ERROR\_MESSAGE();  THROW; -- Ném lại lỗi để có thể xử lý thêm ngoài thủ tục này nếu cần  END CATCH  END; |

# Transaction trong thủ tục sp\_QuenMatKhau

|  |
| --- |
| CREATE PROCEDURE sp\_QuenMatKhau  @TenTK NVARCHAR(50),  @MatKhauMoi NVARCHAR(255)  AS  BEGIN  SET NOCOUNT ON;  -- Bắt đầu giao dịch  BEGIN TRANSACTION;  BEGIN TRY  -- 1. Cập nhật mật khẩu trong bảng TaiKhoan  UPDATE TaiKhoan  SET MatKhau = @MatKhauMoi  WHERE TenTK = @TenTK;  -- 2. Cập nhật mật khẩu trong SQL Server Login  IF EXISTS (SELECT \* FROM sys.server\_principals WHERE name = @TenTK)  BEGIN  -- Cập nhật mật khẩu cho Login trong SQL Server  DECLARE @sqlLogin NVARCHAR(MAX);  SET @sqlLogin = 'ALTER LOGIN [' + @TenTK + '] WITH PASSWORD = ''' + @MatKhauMoi + ''';';  EXEC(@sqlLogin);  END  ELSE  BEGIN  -- Nếu Login không tồn tại trong SQL Server, ném lỗi  RAISERROR('Login không tồn tại trong SQL Server', 16, 1);  END  -- Nếu tất cả thành công, commit giao dịch  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  -- Nếu có lỗi xảy ra, rollback giao dịch  ROLLBACK TRANSACTION;  -- Xử lý lỗi  PRINT 'Lỗi: ' + ERROR\_MESSAGE();  -- Thông báo lỗi cho người dùng hoặc ghi log lỗi tùy theo nhu cầu  THROW;  END CATCH  END; |

# Transaction trong thủ tục sp\_TinhLuong

|  |
| --- |
| CREATE PROCEDURE sp\_TinhLuong  @MaNV INT,  @Thuong DECIMAL(18, 2),  @PhuCap DECIMAL(18, 2)  AS  BEGIN  DECLARE @LuongCoBan DECIMAL(18, 2), @SoGioLam INT, @LuongTangCa DECIMAL(18, 2), @TongLuong DECIMAL(18, 2);  -- Bắt đầu transaction  BEGIN TRY  BEGIN TRANSACTION;  -- Lấy Lương cơ bản từ bảng CongViec (lương theo giờ của công việc)  SELECT @LuongCoBan = CV.LuongCoBan  FROM CongViec CV  JOIN NhanVien NV ON NV.MaCV = CV.MaCV  WHERE NV.MaNV = @MaNV;  -- Lấy số giờ làm việc thực tế của nhân viên trong tháng từ bảng ChamCong  SELECT @SoGioLam = SUM(DATEDIFF(HOUR, GioVao, GioRa))  FROM ChamCong  WHERE MaNV = @MaNV;  -- Tính Lương tăng ca (Số giờ tăng ca \* tỷ lệ 1.5 \* lương theo giờ)  SELECT @LuongTangCa = SUM(SoGioTangCa \* 1.5 \* @LuongCoBan)  FROM ChamCong  WHERE MaNV = @MaNV;  -- Tính Tổng lương (Lương cơ bản + Thưởng + Phụ cấp + Lương tăng ca)  SET @TongLuong = (@LuongCoBan \* @SoGioLam) + @Thuong + @PhuCap + ISNULL(@LuongTangCa, 0);  -- Kiểm tra xem bản ghi lương đã có chưa, nếu chưa thì thêm mới  IF NOT EXISTS (SELECT 1 FROM Luong WHERE MaNV = @MaNV)  BEGIN  INSERT INTO Luong (MaNV, Thuong, PhuCap, LuongTangCa, TongLuong)  VALUES (@MaNV, @Thuong, @PhuCap, @LuongTangCa, @TongLuong);  END  ELSE  BEGIN  -- Nếu đã có bản ghi, cập nhật thông tin  UPDATE Luong  SET Thuong = @Thuong, PhuCap = @PhuCap, LuongTangCa = @LuongTangCa, TongLuong = @TongLuong  WHERE MaNV = @MaNV;  END  -- Commit transaction nếu tất cả các thao tác thành công  COMMIT TRANSACTION;  END TRY  BEGIN CATCH  -- Nếu có lỗi xảy ra, rollback transaction  ROLLBACK TRANSACTION;  -- Thông báo lỗi  DECLARE @ErrorMessage NVARCHAR(4000), @ErrorSeverity INT, @ErrorState INT;  SELECT @ErrorMessage = ERROR\_MESSAGE(),  @ErrorSeverity = ERROR\_SEVERITY(),  @ErrorState = ERROR\_STATE();  RAISERROR(@ErrorMessage, @ErrorSeverity, @ErrorState);  END CATCH  END; |

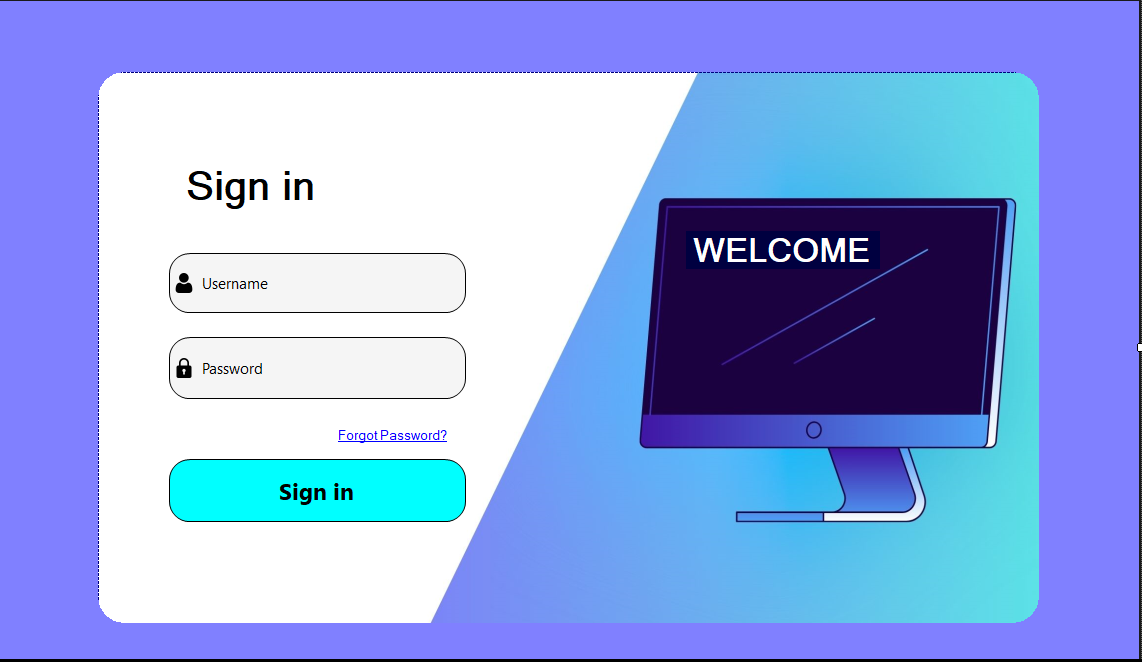
# CHƯƠNG 6: THIẾT KẾ GIAO DIỆN

# Môi trường cài đặt và công nghệ sử dụng

* Hệ cơ sở dữ liệu: Microsoft SQL Server Management Studio 18.
* Chương trình WinForm: Được viết bằng Microsoft Visual Studio 2022, trên nền tảng ASP.NET Web Application (.NET Framework) C#.
* Giao diện người dùng: Sử dụng Guna UI 2 tạo giao diện thân thiện đẹp mắt.

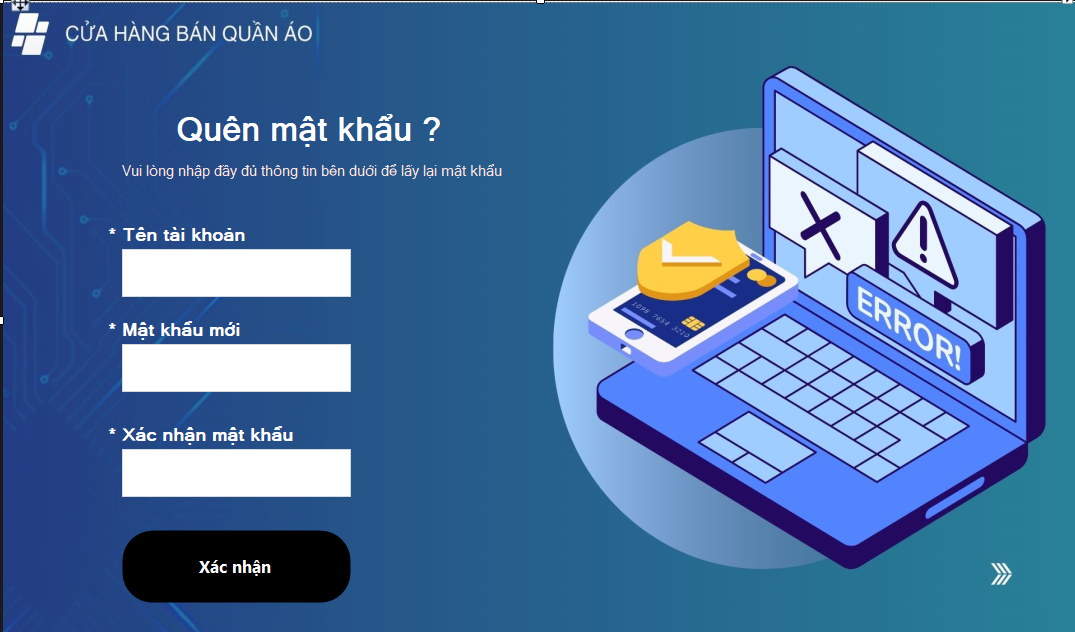
# Giao diện chung

## 2.1 Form đăng nhập (frmLogin)



Hình 2.1 Form đăng nhập (frmLogin)

## 2.2 Form đổi mật khẩu (frmForgotPassword)



Hình 2.2 Form đổi mật khẩu (frmForgotPassword)

## 2.3 Form Menu chính (frmMenu)

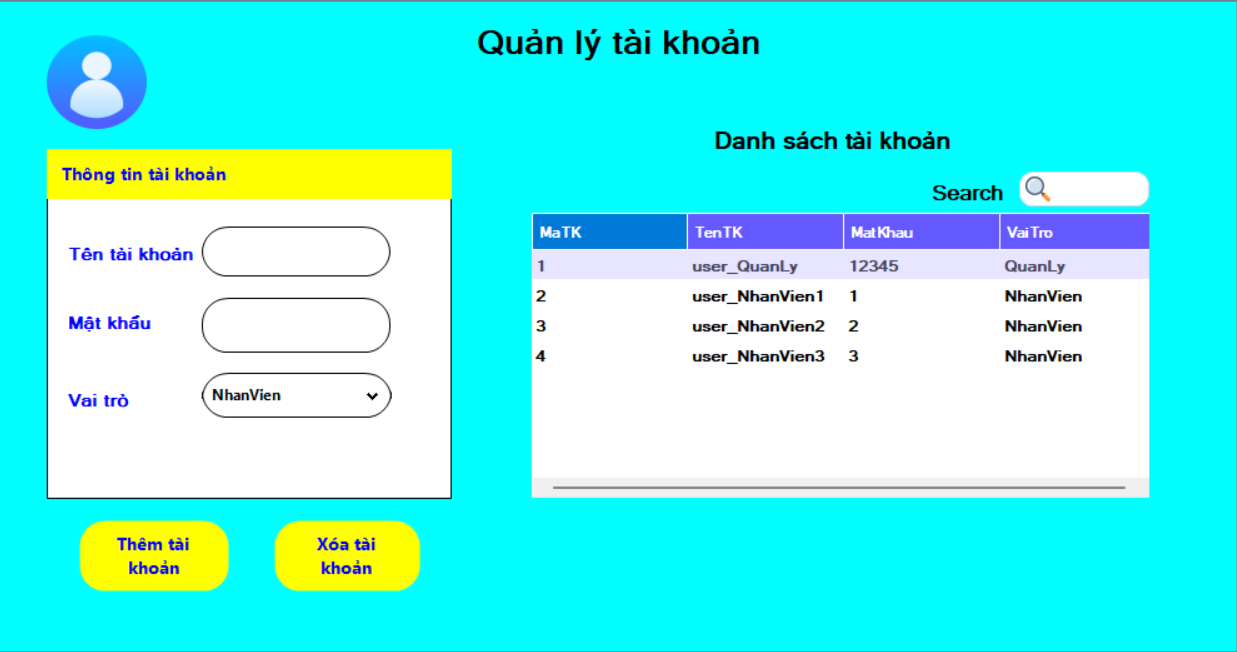


Hình 2.3 Form Menu chính (frmMenu)

# Giao diện riêng

## Giao diện Quản lý

### *3.1.1 User Control Quản lý tài khoản (UC\_Quanlytaikhoan)*



Hình 3.1 User Control Quản lý tài khoản (UC\_Quanlytaikhoan)

### *3.1.2 User Control Quản lý nhân viên (UC\_Quanlynhanvien)*



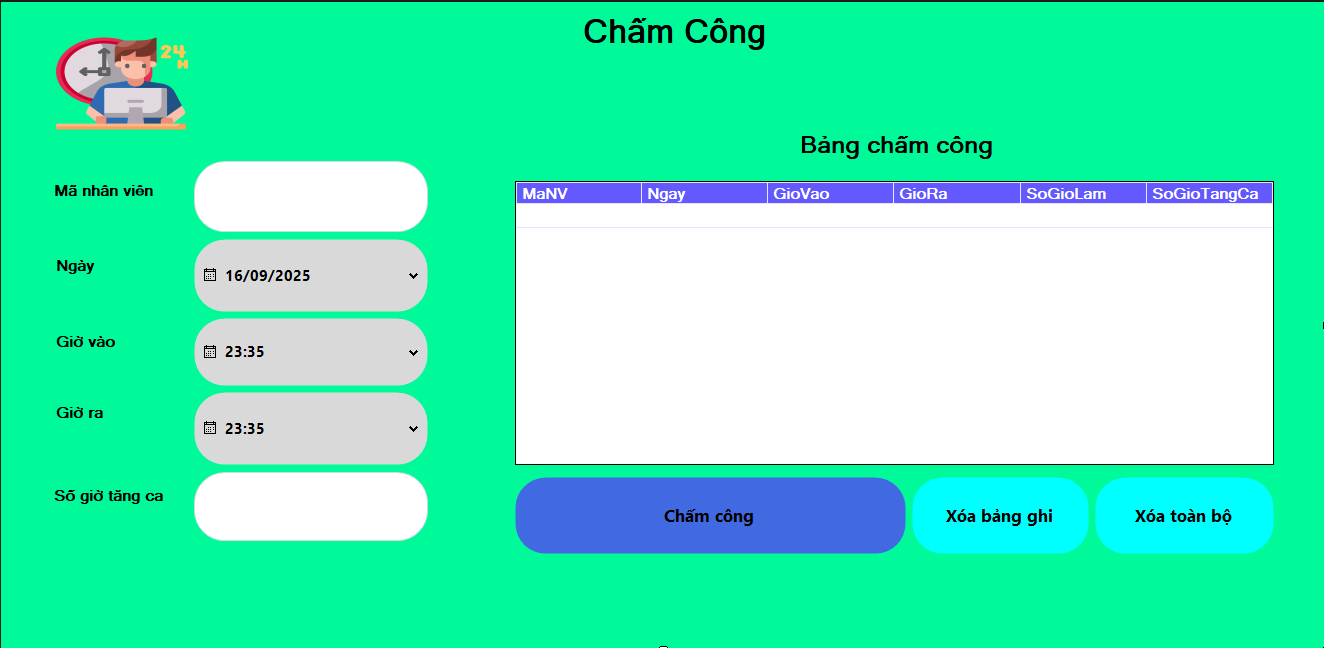
Hình 3.2 User Control Quản lý nhân viên (UC\_Quanlynhanvien)

### *User Control Quản lý công việc (UC\_Quanlycongviec)*



Hình 3.3 User Control Quản lý công việc (UC\_Quanlycongviec)

### *3.1.4 User Control Chấm Công (UC\_ChamCong)*



Hình 3.4 User Control Chấm Công (UC\_ChamCong)

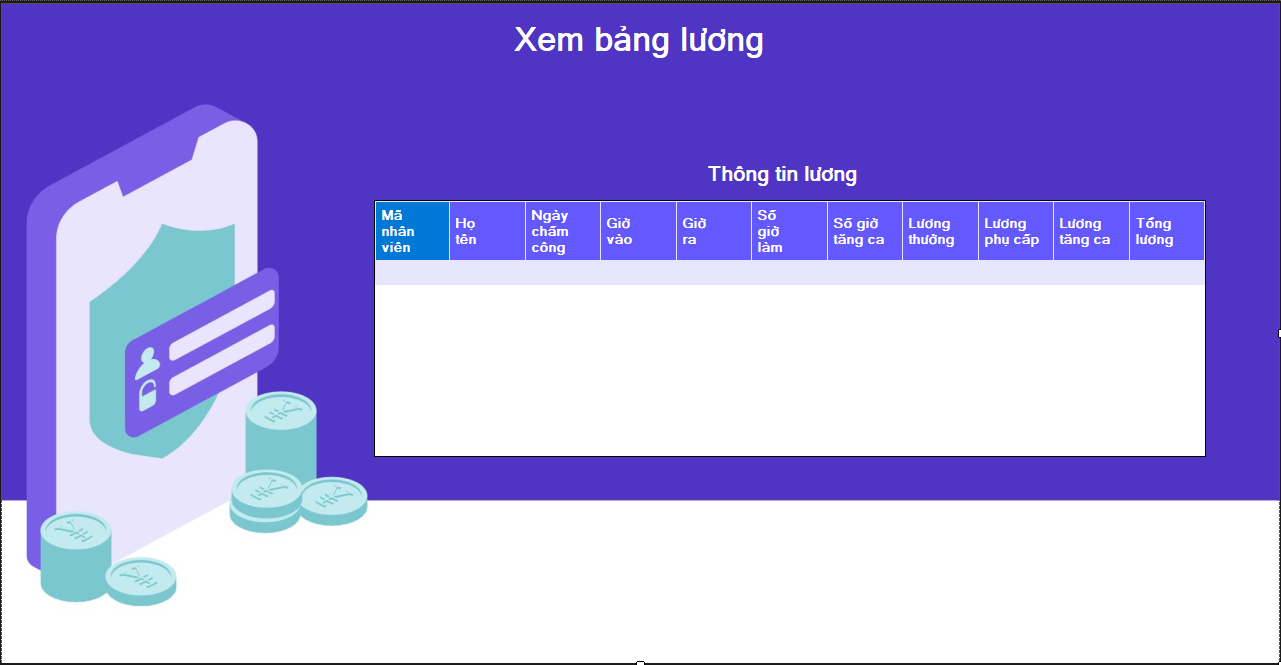
### *3.1.5 User Control Tính Lương (UC\_TinhLuong)*



Hình 3.5 User Control Tính Lương (UC\_TinhLuong)

## 3.2 Giao diện Nhân viên

### 3.2.1 User Control Xem bảng lương (UC\_Xembangluong)



Hình 3.6 User Control Xem bảng lương (UC\_Xembangluong)

### 3.2.2 User Control Hosonhanvien (UC\_Hosonhanvien)



Hình 3.7 User Control Hosonhanvien (UC\_Hosonhanvien)