

|  |
| --- |
| **TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**  **KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**  **BỘ MÔN HỆ THỐNG THÔNG TIN**  **----------** |
| Káº¿t quáº£ hÃ¬nh áº£nh cho hcmute**ĐỀ TÀI CUỐI KÌ MÔN HỆ QUẢN TRỊ CSDL**  **\*\*\***  **QUẢN LÝ CỬA HÀNG TIỆN LỢI**  **GVHD:** TS. Nguyễn Thành Sơn  **Lớp HP:** DBMS330284\_22\_2\_04clc  **Nhóm thực hiện:** Nhóm 04  **Học kì:** 2  **Năm học:** 2022 - 2023 |

**Thành phố Hồ Chí Minh, tháng 05, năm 2023**

**DANH SÁCH SINH VIÊN NHÓM THỰC HIỆN**

**HỌC KÌ 1 NĂM HỌC 2022-2023**

**Nhóm 04**

|  |  |  |
| --- | --- | --- |
| **STT** | **HỌ VÀ TÊN SINH VIÊN** | **MSSV** |
| 1 | Kiên Đức Trọng | 21110332 |
| 2 | Nguyễn Trần Văn Trung | 21110335 |
| 3 | Phạm Quỳnh Như | 21110843 |
| 4 | Trần Quốc Phương | 21110847 |

**Đề tài: Quản lý cửa hàng tiện lợi**

***Nhận xét của giáo viên***

………………………………………………………………………………………………..………………………………………………………………………………………………..

………………………………………………………………………………………………..

………………………………………………………………………………………………..

………………………………………………………………………………………………..

Tp. Hồ Chí Minh, ngày …, tháng …, năm 2023

Giảng viên chấm điểm

**MỤC LỤC**

[LỜI CẢM ƠN 1](#_Toc134397050)

[LỜI NÓI ĐẦU 2](#_Toc134397051)

[CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG 3](#_Toc134397052)

[1.1. Mô tả bài toán 3](#_Toc134397053)

[1.2. Nghiệp vụ 5](#_Toc134397054)

[1.2.1 Đối với nhân viên 5](#_Toc134397055)

[1.2.2 Đối với quản lý 6](#_Toc134397056)

[1.3. Mô tả các chức năng 7](#_Toc134397057)

[1.4. Cách tính lương nhân viên, doanh thu cửa hàng, điểm tích lũy 7](#_Toc134397058)

[CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 9](#_Toc134397059)

[2.1 Thiết kế cơ sở dữ liệu mức quan niệm 9](#_Toc134397060)

[2.2 Thiết kế cơ sở dữ liệu mức logic 10](#_Toc134397061)

[2.3 Các ràng buộc cần có 11](#_Toc134397062)

[2.4. Cài đặt CSDL và các ràng buộc 12](#_Toc134397063)

[2.4.1 Tạo bảng và các ràng buộc đơn giản 12](#_Toc134397064)

[2.4.2 Trigger kiểm tra các ràng buộc 15](#_Toc134397065)

[2.4.3 Các view 16](#_Toc134397066)

[CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG 18](#_Toc134397067)

[3.1 Kết nối cơ sở dữ liệu 18](#_Toc134397068)

[3.2 Các thủ tục, hàm 19](#_Toc134397069)

[3.2.1 Thông tin nhân viên 19](#_Toc134397070)

[3.2.2 Thông tin nhà sản xuất 22](#_Toc134397071)

[3.2.3 Sản phẩm 22](#_Toc134397072)

[3.2.4 Kho 23](#_Toc134397073)

[3.2.5 Cửa hàng 24](#_Toc134397074)

[3.2.6 Khách Hàng 25](#_Toc134397075)

[3.2.7 Ngày lễ 26](#_Toc134397076)

[3.2.8 Hóa đơn 26](#_Toc134397077)

[3.2.9 Thông tin chi tiết hóa đơn 27](#_Toc134397078)

[3.2.10 Thông tin phân công 28](#_Toc134397079)

[3.2.11 Hàm tính doanh thu 29](#_Toc134397080)

[3.2.12 Hàm lọc doanh thu 29](#_Toc134397081)

[3.2.13 Thủ tục Update lại cửa hàng khi thanh toán 29](#_Toc134397082)

[3.2.14 Thủ tục UpdateGioBD 30](#_Toc134397083)

[3.2.15 Thủ tục UpdateGioKT 30](#_Toc134397084)

[3.2.16 Hàm Tính Lương một Chấm Công 30](#_Toc134397085)

[3.2.17 Lương 31](#_Toc134397086)

[3.2.18 Hàm bảng chi tiết hóa đơn thanh toán 32](#_Toc134397087)

[3.2.19 Hàm tính MaxHD 32](#_Toc134397089)

[CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN 32](#_Toc134397090)

[CHƯƠNG 5: THIẾT KẾ GIAO DIỆN HỆ THỐNG 34](#_Toc134397091)

[5.1 Form đăng nhập 34](#_Toc134397092)

[5.2 Form home 35](#_Toc134397093)

[5.3 Thao tác đăng nhập 35](#_Toc134397094)

[5.4 Các hàm kết nối và gọi thực thi chung từ SQL 37](#_Toc134397095)

[5.5 Các hàm gọi thủ tục, hàm, view từ SQL trong lớp DAO (Data Access Object) 38](#_Toc134397096)

[5.5.1 Ca 38](#_Toc134397097)

[5.5.2 Chấm công 38](#_Toc134397098)

[5.5.3 Chi tiết hóa đơn 39](#_Toc134397099)

[5.5.4 Chức vụ 40](#_Toc134397100)

[5.5.5 Cửa hàng 40](#_Toc134397101)

[5.5.6 Hóa đơn 42](#_Toc134397102)

[5.5.7 Khách hàng 43](#_Toc134397103)

[5.5.8 Kho 44](#_Toc134397104)

[5.5.9 Ngày lễ 45](#_Toc134397105)

[5.5.10 Nhân viên 46](#_Toc134397106)

[5.5.11 Nhà sản xuất 48](#_Toc134397107)

[5.5.12 Phân công 49](#_Toc134397108)

[5.5.13 Phương thức thanh toán 50](#_Toc134397109)

[5.5.14 Sản phẩm 50](#_Toc134397110)

[5.5.15 Tính lương 52](#_Toc134397111)

[5.6 Các form danh mục 52](#_Toc134397112)

[5.6.1 Đăng nhập với tư cách là nhân viên 52](#_Toc134397113)

[5.6.2 Đăng nhập với tư cách là quản lý 58](#_Toc134397114)

**LỜI CẢM ƠN**

Lời đầu tiên, chúng em xin gửi lời cảm ơn sâu sắc đến thầy Nguyễn Thành Sơn vì sự hướng dẫn tận tình và giúp đỡ chúng em hoàn thành bài báo cáo cuối kì bộ môn Hệ quản trị cơ sở dữ liệu này. Nhờ những kiến thức thầy truyền đạt mà chúng em đã nắm chắc hơn được những kiến thức căn bản, nền tảng cho việc khi xây dựng một hệ thống quản trị cơ sở dữ liệu. Thông qua đồ án này, nhóm em xin trình bày lại quá trình phát triển của một hệ thống quản trị cơ sở dữ liệu, cũng như lập trình một đồ án liên quan. Trong quá trình thực hiện đồ án môn học, khó tránh khỏi những sai sót. Do đó nhóm chúng em rất mong sự góp ý đến từ thầy để bài làm của nhóm chúng em được hoàn thiện hơn, phát triển và đầy đủ hơn. Kính chúc thầy sức khỏe, hạnh phúc thành công trên con đường sự nghiệp giảng dạy. Cuối cùng chúng em xin chân thành cảm ơn các thầy và các bạn cùng tham gia học bộ môn Hệ quản trị cơ sở dữ liệu đã nhiệt tình giúp đỡ động viên chúng em trong trình thực hiện đồ án cuối kỳ.

**LỜI NÓI ĐẦU**

Ngày nay, với sự phát triển mạnh mẽ của kinh tế xã hội đi đôi với nhu cầu cấp thiết của con người thì việc xuất hiện các ứng dụng về khoa học kỹ thuật là vô cùng quan trọng. Và để giúp cho các ứng dụng khoa học kỹ thuật phát triển thì không thể phủ nhận vai trò của công nghệ thông tin. Trong những năm gần đây, ngành công nghệ thông tin ngày càng có xu hướng trở nên phổ biến và cần thiết. Bởi vì bất kể lĩnh vực hay ngành nghề nào đều có ứng dụng của công nghệ thông tin. Vì thế, công nghệ thông tin đã và đang đóng vai trò quan trọng trong đời sống kinh tế, xã hội hằng ngày của chúng ta.

Và với sự phát triển vượt bậc đó, thì không thể phủ nhận được rằng chính việc quản lý dữ liệu, thông tin của công nghệ thông tin đã giúp cho các doanh nghiệp, công ty tiết kiệm được hàng ngàn chi phí thống kê và dễ dàng kiểm soát trong kinh doanh. Đặc biệt, là cho các chủ cửa hàng có thể dễ dàng trong việc kiểm soát nhân viên, quá trình làm việc, lương,... từ xa thông qua các phần mềm quản lý trên hệ thống. Điều đó đạt hiệu quả rõ rệt khi giảm tối thiểu được chi phí nhân lực, thời gian cũng như tránh sai sót hay thiếu dữ liệu so với quản lý bằng giấy như trước đây.

Chính vì những lý do đó, ứng dụng của Công nghệ thông tin đối với công tác quản lý thông tin, nhân viên trong cửa hàng đóng vai trò hết sức quan trọng. Và để làm rõ hơn về ứng dụng của Công nghệ thông tin ứng với công tác quản lý trong thực tế thì nhóm em lựa chọn đề tài “Quản lý cửa hàng tiện lợi” để nghiên cứu và phát triển ứng dụng của việc quản lý hệ thống dữ liệu trong môn học Hệ quản trị cơ sở dữ liệu.

**CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG**

**1.1. Mô tả bài toán**

Ở bài đồ án lần này, chúng tôi lựa chọn đề tài xây dựng chương trình quản lý cửa hàng tiện lợi. Cửa hàng tiện lợi cung cấp cho mỗi nhân viên một tài khoản, mỗi tài khoản được phân quyền với các chức năng và nghiệp vụ khác nhau tùy vào chức vụ của từng nhân viên, trong đó tài khoản Admin được quản lý toàn quyền của chương trình. Các hoạt động cơ bản của chương trình ứng dụng có thể kể đến như là quản lý sản phẩm, quản lý nhân sự, thanh toán và thống kê doanh thu. Một hệ thống quản lý cửa hàng tiện lợi cần lưu trữ những dữ liệu như sau:

Cửa hàng có nhiều nhân viên, mỗi nhân viên sẽ có mã nhân viên để phân biệt, nhân viên còn có các thông tin cá nhân như họ tên, tuổi tác, giới tính, ngày sinh, số điện thoại và email. Nhân viên phân công đảm nhiệm một chức vụ và có hoạt động chấm công với mỗi ngày làm việc tùy theo ca mà mình đăng ký, nhân viên có thể đăng ký nhiều ca.

Ngoài ra mỗi nhân viên có một tài khoản bao gồm tên tài khoản và mật khẩu để đăng nhập vào hệ thống. Mỗi nhân viên sẽ có chức vụ và quyền hạn khác nhau nên sẽ có tài khoản tương ứng và phân quyền khác nhau.

Ca làm việc bao gồm năm ca là ca sáng, ca trưa, ca chiều, ca tối và ca đêm. Mỗi ca phân biệt bằng mã ca và loại ca, có giờ bắt đầu và giờ kết thúc. Các ca làm việc có thể được mô tả như sau:

a. **Ca A** (06h -10h)

b. **Ca B** (10h -14h)

c. **Ca C** (14h -18h)

d. **Ca D** (18h - 22h)

e. **Ca E** (22h - 06h)

Mỗi chức vụ có mã chức vụ và tên của chức vụ đó. Một nhân viên có thể đảm nhận các chức vụ sau: Nhân viên: Full-time, Part-time; Quản lý và Chủ cửa hàng - Người sử dụng tài khoản Admin.

Hoạt động đăng ký ca được diễn ra liên tục, mỗi nhân viên bắt buộc phải thực hiện đăng ký phần ca làm để hệ thống ghi nhận. Thông tin phân công đăng ký ca bao gồm các thông tin như mã phân công để phân biệt, mã nhân viên, mã ca và ngày đăng ký. Nhân viên chỉ được đăng ký phần ca làm vào tương lai (kể từ ngày hôm sau trở đi).

Mỗi ngày làm việc của nhân viên đều phải ghi nhận hoạt động chấm công. Thông tin của mỗi lần chấm công bao gồm mã phân công, giờ bắt đầu, giờ kết thúc và lương trong số giờ đã làm ở ca đó.

Ngoài ra, mỗi nhân viên được tính lương theo giờ của ca. Tùy vào từng chức vụ, loại ca đăng ký, số giờ đã làm trong ca đó mà mức lương của mỗi nhân viên có thể khác nhau. Sau một ngày làm việc và hoàn thành chấm công đầy đủ, lương của nhân viên sẽ được tổng kết vào cuối ngày. Tính lương bao gồm các thông tin như lương theo giờ, lương cuối ngày và lương thưởng (nếu có). Mức lương có thể được tính như sau:

a. **Ca ngày:** (ca A, B, C, D)

**+ Đối với chức vụ nhân viên part-time:** 23.000 đồng/ giờ.

**+ Đối với bảo vệ:** 25.000 đồng/ giờ.

**+ Đối với nhân viên full-time:** 30.000 đồng/ giờ.

**+ Đối với quản lý:** 40.000 đồng/ giờ.

b. **Ca đêm:** Được cộng thêm 25% so với ca ngày (ca E).

**Lương thưởng:**

**- Quyền lợi khi làm vào các dịp Lễ/ Tết:** Lương x 3.

Thông tin của ngày Lễ/ Tết bao gồm mã ngày lễ, tên sự kiện, và ngày diễn ra.

Mỗi sản phẩm có mã sản phẩm, tên sản phẩm, giá gốc, giá bán, mã nhà sản xuất, mã loại, trạng thái, hình ảnh sản phẩm. Mỗi loại sản phẩm sẽ có mã loại và tên loại sản phẩm. Mỗi sản phẩm sẽ có nhà sản xuất cung cấp cho cửa hàng đó, nhà sản xuất gồm mã nhà sản xuất, tên nhà sản xuất, địa chỉ nhà sản xuất. Mỗi sản phẩm sẽ thuộc duy nhất một loại sản phẩm trong cửa hàng và được duy nhất một nhà sản xuất cung cấp.

Kho của cửa hàng sẽ gồm có loại kho (kho lạnh và kho thường), ngày nhập kho, mã sản phẩm được lưu trữ trong kho, số lượng tồn kho của những sản phẩm đó, ngoài ra còn có các thông tin như hạn sử dụng, ngày sản xuất của những sản phẩm được lưu trữ trong đó.

Cửa hàng gồm những mã sản phẩm được đưa ra bày bán, ngày xuất kho, ngày sản xuất, hạn sử dụng và số lượng những sản phẩm đó.

Khách hàng sẽ có thể được thanh toán nhiều hóa đơn. Mỗi khách hàng có tên khách hàng, số điện thoại và điểm tích lũy. Mỗi hóa đơn gồm có mã hóa đơn, ngày thanh toán, mã nhân viên thanh toán cho khách hàng đó, số điện thoại khách hàng được thanh toán, phương thức thanh toán và tổng tiền thanh toán của hóa đơn đó . Mỗi hóa đơn có chi tiết hóa đơn, lưu số lượng, thành tiền và tiền lời tính được.

**1.2. Nghiệp vụ**

**1.2.1 Đối với nhân viên**

*Xem thông tin cá nhân*

Nhân viên có thể xem thông tin cá nhân của bản thân và có thể đăng tải và lưu ảnh đại diện của mình.

*Thanh toán*

Nghiệp vụ thanh toán bao gồm các bước sau:

**Bước 1:** Tạo hóa đơn thanh toán.

**Bước 2:** Chọn các sản phẩm có sẵn trong cửa hàng để Order cho hóa đơn, lưu lại thông tin số điện thoại của khách hàng (Nếu khách hàng đã đăng ký thành viên) để tích điểm.

**Bước 3:** Thanh toán hoặc hủy hóa đơn nếu có sai sót.

*Đăng ký công ca và chấm công*

Nhân viên chọn ca làm và ngày mà mình muốn đăng ký, có thể xóa đăng ký và đăng ký lại. Tới ngày làm việc sẽ có hoạt động chấm công. Check in và check out đúng giờ trong ca đã đăng ký sẽ hiện lương của ca làm đó.

*Quản lý khách hàng*

Nhân viên lưu trữ thông tin đăng ký thành viên của khách hàng, có các thủ tục như thêm, sửa, xóa và có thể xem các hóa đơn mà khách hàng đã đặt trước đó.

**1.2.2 Đối với quản lý**

*Quản lý nhân viên*

Quản lý quản lý toàn bộ thông tin của nhân viên và có các thủ tục cơ bản như tìm kiếm, thêm, xóa và sửa nhân viên.

*Quản lý khách hàng*

Tương tự như nghiệp vụ quản lý khách hàng của nhân viên.

*Quản lý nhập kho*

Quản lý thông tin nhà sản xuất mà cửa hàng hợp tác.

Quản lý các sản phẩm mà cửa hàng đăng ký bày bán. Có thủ tục thêm, xóa, sửa.

Thao tác nhập kho: Chọn sản phẩm muốn nhập kho từ danh sách sản phẩm và nhập số lượng cần nhập kho.

*Quản lý xuất kho*

Thao tác xuất kho: Chọn sản phẩm muốn xuất kho từ danh sách sản phẩm trong kho để xuất hàng ra cửa hàng, lựa chọn số lượng phù hợp.

*Quản lý phân công*

Tương tự như nghiệp vụ đăng ký công ca và chấm công của nhân viên. Nhưng người quản lý có thể xem và thao tác trên toàn bộ nhân viên, bên cạnh đó người quản lý còn có thể quản lý những ngày lễ và chỉnh sửa thông tin ngày lễ trong tương lai.

*Quản lý lương*

Xem và thống kê bảng lương của toàn bộ nhân viên theo ngày.

*Quản lý doanh thu*

Theo dõi doanh thu từ toàn bộ hóa đơn đã thanh toán của cửa hàng. Lọc doanh thu và xem được tổng tiền lời nhận được từ một khoảng thời gian từ ngày mấy đến ngày mấy.

**1.3. Mô tả các chức năng**

*Người quản lý - Người quản tị:*

Người quản lý có thể truy cập vào xem thông tin cá nhân của nhân viên: Toàn quyền với nhân viên và sản phẩm. Người quản trị có thể xem thông tin tài khoản và thay đổi mật khẩu.

*Nhân viên cửa hàng:*

Thêm, xóa sửa tài khoản tích lũy của khách hàng, có thể thanh toán cho từng hóa đơn, xuất hóa đơn. Phân quyền: “Chủ cửa hàng” toàn quyền với hệ thống. “Người quản lý” quản lý một phần quyền với hệ thống. “Nhân viên” không thể tham gia vào quyền của tài khoản khác.

**1.4. Cách tính lương nhân viên, doanh thu cửa hàng, điểm tích lũy**

* *Tính tiền cho hóa đơn được tính:*

Tiền hóa đơn: Tổng tiền các sản phẩm

* *Tổng doanh thu cửa hàng được tính:*

Tổng doanh thu = Tổng tiền các hóa đơn

* *Tính tổng tiền lời:*

Tổng tiền lời của một hóa đơn = Tổng tiền lời của các sản phẩm trong hóa đơn.

Tiền lời một sản phẩm sau khi bán = (Giá bán - Giá gốc) x SL.

* *Tính điểm tích lũy cho khách hàng:*

Điểm tích lũy: Khách hàng được cộng điểm tích lũy bằng 1% dựa trên tổng hóa đơn

1000 điểm = 1.000 VND

* *Tính lương nhân viên được tính:*

Lương nhân viên dựa vào ngày, mã chức vụ, mã ca, mã ngày lễ để tính.

Đối với chức vụ nhân viên part-time: 23.000 đồng/ giờ.

Đối với nhân viên full-time: 30.000 đồng/ giờ.

Đối với bảo vệ: 25.000 đồng/ giờ

Đối với quản lý: 40.000 đồng/ giờ

***Ví dụ:***

* Một nhân part-time làm 2 ca gồm 4 tiếng ca A, và 6 tiếng E trong 1 ngày bình thường (không phải ngày lễ) sẽ được tính như sau:

Lương = 4 x 23.000 + 6 x 23.000 x 1.25 = 264.500 đồng.

Nếu làm trong ngày lễ:

Lương = (4 x 23.000 + 6 x 23.000 x 1.25) x 3 = 793.500 đồng.

* Một quản lý sẽ làm full-time 1 ngày (ca A, B):

Lương = 8 \* 30.000 = 240.000 đồng

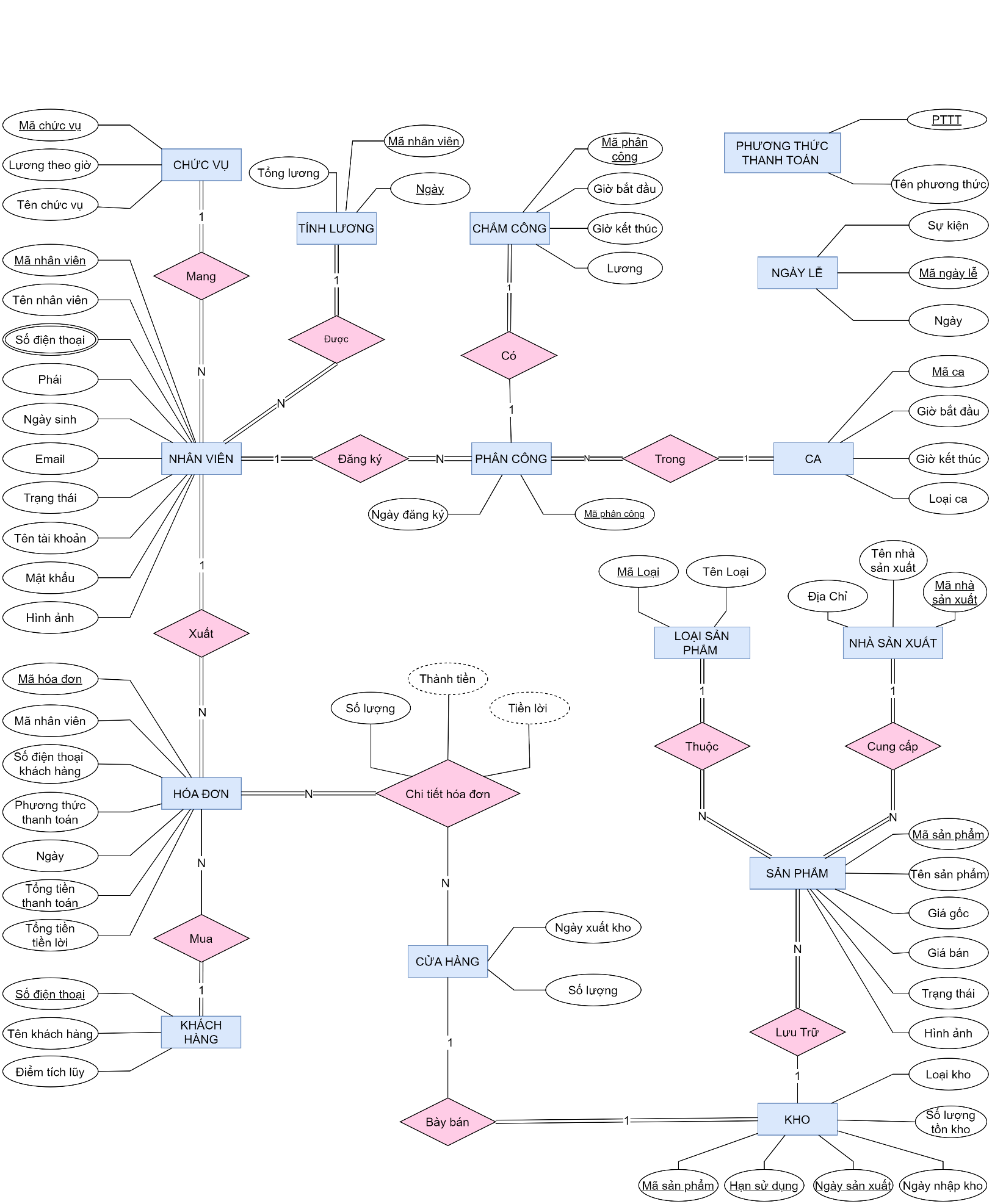
Nếu làm trong ngày lễ:

Lương = (8 \* 30.000) x3 = 720.000 đồng

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

**2.1 Thiết kế cơ sở dữ liệu mức quan niệm**

Từ mô tả về dữ liệu cần có ở phần mô tả bài toán, hình thành được sơ đồ thực thể kết hợp (ERD).



**2.2 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, LuongTheoGio)

**NHANVIEN** (MaNV, TenNV, Sdt, Phai, NgaySinh, Email, MaCV, TrangThai, TenTK, MatKhau, Hình)

**NHASANXUAT** (MaNSX, TenNSX, DiaChi)

**LOAISANPHAM** (MaLoai, TenLoai)

**SANPHAM** (MaSP, TenSP, MaNSX, MaLoai, GiaBan, GiaGoc, TrangThai, Hinh)

**KHO** (MaSP, NSX, HSD, NgayNhapKho, LoaiKho, SLTonKho)

**CUAHANG** (MaSP, NSX, HSD, NgayXuatKho, SoLuong)

**KHACHHANG** (Sdt, TenKH, DiemTichLuy)

**NGAYLE** (MaNL, Ngay, SuKien)

**HOADON** (MaHD, MaNV, SdtKH, PTTT, Ngay, TongTienTT, TongTienLoi)

**CHITIETHOADON** (MaHD, MaSP, NSX, HSD, SoLuong, GiaBan, GiaGoc, ThanhTien, TienLoi)

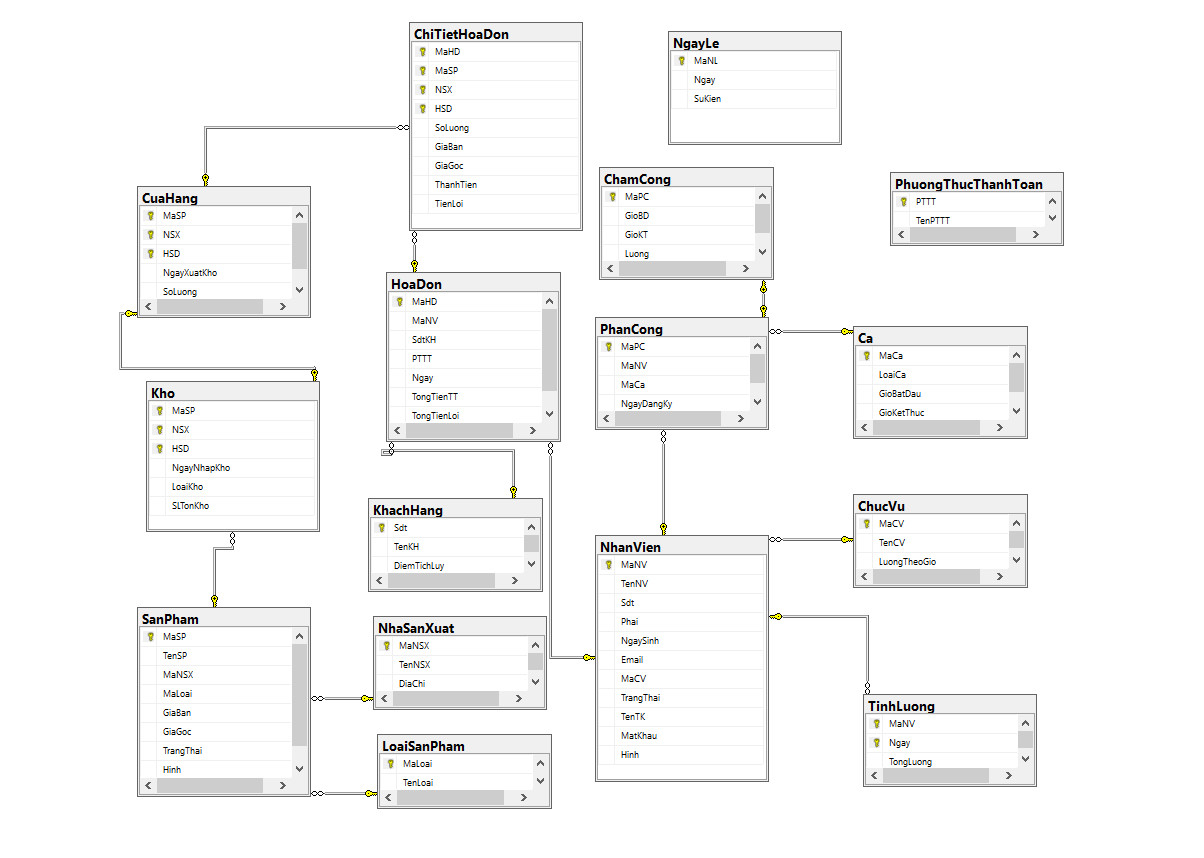
**CA** (MaCa, LoaiCa, GioBatDau, GioKetThuc)

**PHANCONG** (MaPC, MaNV, MaCa, NgayDangKy)

**CHAMCONG** (MaPC, GioBD, GioKT, Luong)

**TINHLUONG** (MaNV, Ngay, TongLuong)

**PHUONGTHUCTHANHTOAN**(PTTT, TenPTTT)

****

**2.3 Các ràng buộc cần có**

|  |  |  |
| --- | --- | --- |
| **STT** | **Bảng** | **Ràng buộc** |
| 1 | Chucvu | Khoá chính **MaCV** |
| 2 | NhanVien | Khóa chính **MaNV**  Khóa ngoại **MaCV** tham chiếu đến MaCV (ChucVu) |
| 3 | NhaSanXuat | Khóa chính **MaNSX** |
| 4 | LoaiSanPham | Khóa chính **MaLoai** |
| 5 | SanPham | Khóa chính **MaSP**  Khóa ngoại **MaNSX**, **MaLoai** tham chiếu đến MaNSX (NhaSanXuat), MaLoai (LoaiSanPham) |
| 6 | Kho | Khóa chính **MaSP, NSX, HSD**  Khóa ngoại **MaSP** tham chiếu đến MaSP (SanPham) |
| 7 | CuaHang | Khóa chính **MaSP, NSX, HSD**  Khóa ngoại **MaSP, NSX, HSD** tham chiếu đến MaSP, NSX, HSD (Kho) |
| 8 | KhachHang | Khóa chính **Sdt** |
| 9 | NgayLe | Khóa chính **MaNL** |
| 10 | HoaDon | Khóa chính **MaHD**  Khóa ngoại **MaNV**, **MaKH** tham chiếu đến MaNV(NhanVien), MaKH(KhachHang) |
| 11 | ChiTietHoaDon | Khóa chính **MaHD**, **MaSP** tham chiếu đến **MaHD(HoaDon)**, MaSP(SanPham) |
| 12 | Ca | Khóa chính **MaCa** |
| 13 | PhanCong | Khóa chính **MaPC** |
| 14 | ChamCong | Khóa chính **MaPC** tham chiếu đến **MaPC(PhanCong)** |
| 15 | TinhLuong | Khóa chính **MaNV**, **Ngay**  Khóa ngoại **MaNV** tham chiếu đến **MaNV (NhanVien)** |
| 16 | PhuongThucThanhToan | Khóa chính **PTTT** |

**2.4. Cài đặt CSDL và các ràng buộc**

**2.4.1 Tạo bảng và các ràng buộc đơn giản**

***Bảng chức vụ***

|  |
| --- |
| CREATE TABLE dbo.ChucVu  (  MaCV INT IDENTITY PRIMARY KEY,  TenCV NVARCHAR(255) NOT NULL,  LuongTheoGio FLOAT NOT NULL  ) |

***Bảng nhân viên***

|  |
| --- |
| CREATE TABLE dbo.NhanVien  (  MaNV NVARCHAR(10) PRIMARY KEY,  TenNV NVARCHAR(255) NOT NULL,  Sdt NVARCHAR(10) NOT NULL,  Phai BIT NOT NULL DEFAULT 0, -- Nam 0: | Nữ :1  NgaySinh DATE NOT NULL,  Email NVARCHAR(255) NOT NULL ,  MaCV INT REFERENCES dbo.ChucVu(MaCV) NOT NULL,  TrangThai BIT NOT NULL DEFAULT 1, -- 1: Còn làm việc | 0: Nghỉ  TenTK NVARCHAR(255) NOT NULL UNIQUE,  MatKhau NVARCHAR(255) NOT NULL,  Hinh IMAGE NULL  ) |

***Bảng nhà sản xuất***

|  |
| --- |
| CREATE TABLE dbo.NhaSanXuat  (  MaNSX INT IDENTITY PRIMARY KEY,  TenNSX NVARCHAR(255) NOT NULL,  DiaChi NVARCHAR(255) NOT NULL  ) |

***Bảng loại sản phẩm***

|  |
| --- |
| CREATE TABLE dbo.LoaiSanPham  (  MaLoai INT IDENTITY PRIMARY KEY,  TenLoai NVARCHAR(255) NOT NULL  ) |

***Bảng sản phẩm***

|  |
| --- |
| CREATE TABLE dbo.SanPham  (  MaSP INT IDENTITY PRIMARY KEY,  TenSP NVARCHAR(255) NOT NULL,  MaNSX INT REFERENCES dbo.NhaSanXuat(MaNSX) ON DELETE SET NULL ON UPDATE CASCADE NULL,  MaLoai INT REFERENCES dbo.LoaiSanPham(MaLoai) ON DELETE SET NULL ON UPDATE CASCADE NULL,  GiaBan FLOAT NOT NULL,  GiaGoc FLOAT NOT NULL,  TrangThai BIT NOT NULL DEFAULT 0, --1: Còn bán | 0: Hết bán  Hinh IMAGE NULL  ) |

***Bảng kho***

|  |
| --- |
| CREATE TABLE dbo.Kho  (  MaSP INT REFERENCES dbo.SanPham(MaSP) ON UPDATE CASCADE NOT NULL,  NSX DATE NOT NULL,  HSD DATE NOT NULL,  NgayNhapKho DATE NOT NULL,  LoaiKho BIT NOT NULL DEFAULT 1, -- 1: Thường | 0: Lạnh  SLTonKho INT NOT NULL,  PRIMARY KEY (MaSP, NSX, HSD)  ) |

***Bảng cửa hàng***

|  |
| --- |
| CREATE TABLE dbo.CuaHang  (  MaSP INT NOT NULL,  NSX DATE NOT NULL,  HSD DATE NOT NULL,  NgayXuatKho DATE NOT NULL,  SoLuong INT NOT NULL,  FOREIGN KEY (MaSP, NSX, HSD) REFERENCES dbo.Kho(MaSP,NSX, HSD) ON UPDATE CASCADE,  PRIMARY KEY (MaSP, NSX, HSD)  ) |

***Bảng khách hàng***

|  |
| --- |
| CREATE TABLE dbo.KhachHang  (  Sdt NVARCHAR(10) PRIMARY KEY,  TenKH NVARCHAR(255) NOT NULL,  DiemTichLuy FLOAT NOT NULL,  ) |

***Bảng ngày lễ***

|  |
| --- |
| CREATE TABLE dbo.NgayLe  (  MaNL INT IDENTITY PRIMARY KEY,  Ngay DATE NOT NULL,  SuKien NVARCHAR(255) NOT NULL  ) |

***Bảng hóa đơn***

|  |
| --- |
| CREATE TABLE dbo.HoaDon  (  MaHD INT IDENTITY PRIMARY KEY,  MaNV NVARCHAR(10) REFERENCES dbo.NhanVien(MaNV) ON UPDATE CASCADE NULL,  SdtKH NVARCHAR(10) REFERENCES dbo.KhachHang(Sdt) ON UPDATE CASCADE NULL,  PTTT INT NULL, -- 1: Tiền mặt | 2: Chuyển Khoản | 3: Thẻ  Ngay DATE NOT NULL,  TongTienTT FLOAT NOT NULL DEFAULT 0,  TongTienLoi FLOAT NOT NULL DEFAULT 0  ) |

***Bảng chi tiết hóa đơn***

|  |
| --- |
| CREATE TABLE dbo.ChiTietHoaDon  (  MaHD INT REFERENCES dbo.HoaDon(MaHD) ON UPDATE CASCADE NOT NULL,  MaSP INT NOT NULL,  NSX DATE NOT NULL,  HSD DATE NOT NULL,  SoLuong INT NOT NULL,  GiaBan FLOAT NOT NULL,  GiaGoc FLOAT NOT NULL,  ThanhTien FLOAT NOT NULL,  TienLoi FLOAT NOT NULL,  FOREIGN KEY (MaSP, NSX, HSD) REFERENCES dbo.CuaHang(MaSP, NSX, HSD)ON UPDATE CASCADE,  PRIMARY KEY (MaHD, MaSP, NSX, HSD)  ) |

***Bảng ca***

|  |
| --- |
| CREATE TABLE dbo.Ca  (  MaCa NVARCHAR(10) PRIMARY KEY,  LoaiCa INT NOT NULL,  GioBatDau TIME NOT NULL,  GioKetThuc TIME NOT NULL  ) |

***Bảng phân công***

|  |
| --- |
| CREATE TABLE dbo.PhanCong  (  MaPC INT IDENTITY PRIMARY KEY,  MaNV NVARCHAR(10) REFERENCES dbo.NhanVien(MaNV) ON UPDATE CASCADE NOT NULL,  MaCa NVARCHAR(10) REFERENCES dbo.Ca(MaCa) ON DELETE CASCADE ON UPDATE CASCADE NOT NULL,  NgayDangKy DATE NOT NULL,  ) |

***Bảng chấm công***

|  |
| --- |
| CREATE TABLE dbo.ChamCong  (  MaPC INT REFERENCES dbo.PhanCong(MaPC) PRIMARY KEY,  GioBD TIME NOT NULL,  GioKT TIME NOT NULL,  Luong FLOAT NULL,  ) |

***Bảng tính lương***

|  |
| --- |
| CREATE TABLE dbo.TinhLuong  (  MaNV NVARCHAR(10) REFERENCES dbo.NhanVien(MaNV) NOT NULL,  Ngay DATE NOT NULL,  TongLuong FLOAT NOT NULL,  PRIMARY KEY (MaNV, Ngay)  ) |

***Bảng phương thức thanh toán***

|  |
| --- |
| CREATE TABLE PhuongThucThanhToan  (  PTTT INT PRIMARY KEY,  TenPTTT NVARCHAR(255)  ) |

**2.4.2 Trigger kiểm tra các ràng buộc**

***Trigger kiểm tra nhân viên***

|  |
| --- |
| CREATE TRIGGER tg\_CheckNhanVien ON dbo.NhanVien  AFTER INSERT, UPDATE AS  DECLARE @sdt NVARCHAR(10), @ngaysinh DATE, @email NVARCHAR(255), @tentk NVARCHAR(255), @matkhau NVARCHAR(255)  SELECT @sdt=n.Sdt,@ngaysinh=n.NgaySinh,@email=n.Email,@tentk=n.TenTK,@matkhau=n.MatKhau  FROM inserted n  BEGIN  DECLARE @error INT  SET @error = 0  --check SO DIEN THOAI  IF @sdt NOT LIKE '[0][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'  BEGIN  RAISERROR(N'SỐ ĐIỆN THOẠI KHÔNG HỢP LỆ ', 16, 1)  SET @error = 1  END  --check NGAY SINH  IF DATEDIFF(year, @ngaysinh, GETDATE()) < 18  BEGIN  RAISERROR(N'NGÀY SINH DƯỚI 18 TUỔI ', 16, 1)  SET @error = 1  END  --check EMAIL  IF @email NOT LIKE '%\_@\_\_%.\_\_%'  BEGIN  RAISERROR(N'EMAIL KHÔNG HỢP LỆ ', 16, 1)  SET @error = 1  END  --check TEN TAI KHOAN  IF LEN(@tentk) < 5 OR @tentk LIKE '% %'  BEGIN  RAISERROR(N'TÊN TÀI KHOẢN KHÔNG HỢP LỆ', 16, 1)  SET @error = 1  END  --check MAT KHAU  IF LEN(@matkhau) < 5  BEGIN  RAISERROR(N'MẬT KHẨU KHÔNG HỢP LỆ ', 16, 1)  SET @error = 1  END  -- ROLLBACK  IF @error = 1  BEGIN  RAISERROR(N'VUI LÒNG NHẬP LẠI !', 16, 1)  ROLLBACK TRANSACTION  END  END |

***Trigger tích điểm***

|  |
| --- |
| CREATE TRIGGER TichDiem ON dbo.HoaDon  AFTER UPDATE  AS  BEGIN  DECLARE @diemtichluy FLOAT, @sdt nvarchar (10)  SELECT @diemtichluy = i.TongTienTT \* 2 / 100, @sdt = KhachHang.Sdt  FROM Inserted AS i INNER JOIN dbo.KhachHang ON KhachHang.Sdt = i.SdtKH  UPDATE dbo.KhachHang SET DiemTichLuy = @diemtichluy WHERE Sdt = @sdt  END |

**2.4.3 Các view**

***View các sản phẩm trong kho***

|  |
| --- |
| CREATE VIEW ViewKho  AS  SELECT Kho.MaSP, TenSP, NSX, HSD, NgayNhapKho, LoaiKho, SLTonKho  FROM dbo.Kho INNER JOIN dbo.SanPham ON SanPham.MaSP = Kho.MaSP |

***View các sản phẩm trong cửa hàng***

|  |
| --- |
| CREATE VIEW ViewCuaHang  AS  SELECT dbo.CuaHang.MaSP, TenSP, NSX, HSD, NgayXuatKho, SoLuong  FROM dbo.CuaHang INNER JOIN dbo.SanPham ON SanPham.MaSP = dbo.CuaHang.MaSP  GO |

***View các sản phẩm trong hóa đơn thanh toán***

|  |
| --- |
| CREATE VIEW ChiTietHoaDonThanhToan  AS  SELECT MaHD, TenSP, SoLuong, ChiTietHoaDon.GiaBan, ThanhTien  FROM dbo.ChiTietHoaDon INNER JOIN dbo.SanPham ON SanPham.MaSP = ChiTietHoaDon.MaSP  WHERE MaHD = dbo.MaxHD() |

***View xem bảng phân công của nhân viên***

|  |
| --- |
| CREATE VIEW XemBangPC\_NhanVien  AS  SELECT MaPC, PhanCong.MaNV, TenNV, Ca.MaCa, GioBatDau, GioKetThuc, NgayDangKy  FROM dbo.NhanVien, dbo.PhanCong, dbo.Ca  WHERE dbo.NhanVien.MaNV = dbo.PhanCong.MaNV AND dbo.PhanCong.MaCa = dbo.Ca.MaCa |

**CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG**

**3.1 Kết nối cơ sở dữ liệu**

|  |
| --- |
| internal class DBConnection  {  SqlConnection conn = new SqlConnection(Properties.Settings.Default.cnnStr);  public DataTable LayDanhSach(string sqlStr)  {  try  {  conn.Open();  SqlDataAdapter adapter = new SqlDataAdapter(sqlStr, conn);  DataTable data = new DataTable();  adapter.Fill(data);  return data;  }  catch (Exception ex)  {  MessageBox.Show("Lấy danh sách thất bại\n" + ex, "Thông báo");  }  finally  {  conn.Close();  }  return null;  }  public void Execute(string query)  {  try  {  conn.Open();  SqlCommand cmd = new SqlCommand(query, conn);  if (cmd.ExecuteNonQuery() > 0)  MessageBox.Show("Thao tác thành công", "Thông báo");  else MessageBox.Show("Thao tác thất bại", "Thông báo");  }  catch (Exception ex)  {  MessageBox.Show("Thất bại\n" + ex, "Thông báo");  }  finally  {  conn.Close();  }  }  public DataTable TimKiem(string sqlStr)  {  SqlDataAdapter adapter = new SqlDataAdapter(sqlStr, conn);  DataTable data = new DataTable();  adapter.Fill(data);  return data;  }  } |

**3.2 Các thủ tục, hàm**

**3.2.1 Thông tin nhân viên**

**a. Hàm**

Đặt mã nhân viên

|  |
| --- |
| CREATE FUNCTION DatMaNV()  RETURNS NVARCHAR(10)  AS  BEGIN  DECLARE @manv NVARCHAR(10)  DECLARE @so NVARCHAR(10)  IF NOT EXISTS (SELECT \* FROM dbo.NhanVien)  BEGIN  SET @so = '0';  END  ELSE  BEGIN  SELECT @so = MAX(MaNV) FROM dbo.NhanVien;  SET @so = RIGHT(@so, 5);  SET @So = CAST ((CAST(@so AS INT) +1) AS NVARCHAR(10))  END  SET @manv = 'NV' + RIGHT('00000' + @so, 5)  RETURN @manv;  END |

**b. Thủ tục**

Thêm nhân viên

|  |
| --- |
| CREATE PROCEDURE sp\_ThemNhanVien  @tennv NVARCHAR(255),  @sdt NVARCHAR(10),  @phai BIT,  @ngaysinh DATE,  @email NVARCHAR(255),  @macv INT,  @trangthai BIT,  @tentk NVARCHAR(255),  @matkhau NVARCHAR(255)  AS  BEGIN  IF EXISTS (SELECT \* FROM dbo.NhanVien WHERE TenTK = @tentk)  RAISERROR(N'LỖI TRÙNG TÊN TÀI KHOẢN', 16, 1)  ELSE  BEGIN  DECLARE @newmanv NVARCHAR(10)  SET @newmanv = dbo.DatMaNV()  INSERT INTO dbo.NhanVien  (MaNV, TenNV,Sdt,Phai,NgaySinh,Email,MaCV,TrangThai,TenTK,MatKhau)  VALUES  (@newmanv, @tennv, @sdt, @phai, @ngaysinh, @email, @macv, @trangthai, @tentk, @matkhau)    IF EXISTS (SELECT name FROM sys.sql\_logins WHERE name = @tentk)  BEGIN  DECLARE @sqltemp55 nvarchar(2000)  SET @sqltemp55= 'DROP LOGIN ' + @tentk  EXEC (@sqltemp55)  END  DECLARE @sqlString34 nvarchar(2000)  SET @sqlString34= 'CREATE LOGIN [' + @tentk +'] WITH PASSWORD ='''+ @matkhau +''',DEFAULT\_DATABASE=[QuanLyCuaHangTienLoi], CHECK\_EXPIRATION=OFF, CHECK\_POLICY=OFF'  EXEC (@sqlString34)  --Tạo tài khoản người dùng đối với nhân viên đó trên database (tên người dùng trùng với tên login)  DECLARE @sqltemp3 nvarchar(2000)  SET @sqltemp3= 'CREATE USER ' + @tentk +' FOR LOGIN '+ @tentk  EXEC (@sqltemp3)  IF @macv <> 1  BEGIN  DECLARE @sqltemp1 nvarchar(2000)  SET @sqltemp1= 'ALTER ROLE RoleNhanVien ADD MEMBER ' + @tentk  EXEC (@sqltemp1)  --ALTER ROLE RoleNhanVien ADD MEMBER @tentk  END  ELSE  BEGIN  DECLARE @sqltemp2 nvarchar(2000)  SET @sqltemp2= 'ALTER SERVER ROLE sysadmin ADD MEMBER ' + @tentk  EXEC (@sqltemp2)  --ALTER ROLE RoleQuanLy ADD MEMBER @tentk  END  END  END |

Sửa nhân viên

|  |
| --- |
| CREATE PROCEDURE sp\_SuaNhanVien  @manv NVARCHAR(10),  @tennv NVARCHAR(255),  @sdt NVARCHAR(10),  @phai BIT,  @ngaysinh DATE,  @email NVARCHAR(255),  @macv INT,  @trangthai BIT,  @tentk NVARCHAR(255),  @matkhau NVARCHAR(255)  AS  BEGIN  DECLARE @tentkcu NVARCHAR(255)  SELECT @tentkcu = TenTK FROM dbo.NhanVien WHERE MaNV = @manv  DECLARE @sqltemp99 nvarchar(2000)  SET @sqltemp99= 'Drop user ' + @tentkcu;  EXEC (@sqltemp99)  DECLARE @sqltemp100 nvarchar(2000)  SET @sqltemp100= 'Drop login ' + @tentkcu;  EXEC (@sqltemp100)  IF EXISTS (SELECT \* FROM dbo.NhanVien WHERE TenTK = @tentk AND TenTK <> @tentkcu)  RAISERROR(N'LỖI TRÙNG TÊN TÀI KHOẢN', 16, 1)  ELSE  BEGIN  UPDATE dbo.NhanVien SET TenNV = @tennv,  Sdt = @sdt,  Phai = @phai,  NgaySinh = @ngaysinh,  Email = @email,  MaCV = @macv,  TenTK = @tentk,  MatKhau = @matkhau,  TrangThai = @trangthai  WHERE MaNV = @manv    IF EXISTS (SELECT name FROM sys.sql\_logins WHERE name = @tentk)  BEGIN  DECLARE @sqltemp55 nvarchar(2000)  SET @sqltemp55= 'DROP LOGIN ' + @tentk  EXEC (@sqltemp55)  END  DECLARE @sqlString34 nvarchar(2000)  SET @sqlString34= 'CREATE LOGIN [' + @tentk +'] WITH PASSWORD ='''+ @matkhau +''', DEFAULT\_DATABASE=[QuanLyCuaHangTienLoi], CHECK\_EXPIRATION=OFF, CHECK\_POLICY=OFF'  EXEC (@sqlString34)  --Tạo tài khoản người dùng đối với nhân viên đó trên database (tên người dùng trùng với tên login)  DECLARE @sqltemp3 nvarchar(2000)  SET @sqltemp3= 'CREATE USER ' + @tentk +' FOR LOGIN '+ @tentk  EXEC (@sqltemp3)  IF @macv <> 1  BEGIN  DECLARE @sqltemp1 nvarchar(2000)  SET @sqltemp1= 'ALTER ROLE RoleNhanVien ADD MEMBER ' + @tentk  EXEC (@sqltemp1)  --ALTER ROLE RoleNhanVien ADD MEMBER @tentk  END  ELSE  BEGIN  DECLARE @sqltemp2 nvarchar(2000)  SET @sqltemp2= 'ALTER SERVER ROLE sysadmin ADD MEMBER ' + @tentk  EXEC (@sqltemp2)  --ALTER ROLE RoleQuanLy ADD MEMBER @tentk  END  END  END |

Xóa nhân viên

|  |
| --- |
| CREATE PROCEDURE sp\_XoaNhanVien  @manv NVARCHAR(10)  AS  BEGIN  IF EXISTS (SELECT \* FROM dbo.PhanCong WHERE MaNV = @maNV)  BEGIN  UPDATE NhanVien SET TrangThai = 0 WHERE MaNV = @maNV;  END  ELSE  BEGIN  DECLARE @tentk NVARCHAR(255)  SELECT @tentk = TenTK FROM dbo.NhanVien WHERE MaNV = @manv  DECLARE @sqltemp99 nvarchar(2000)  SET @sqltemp99= 'Drop user ' + @tentk;  EXEC (@sqltemp99)  DECLARE @sqltemp100 nvarchar(2000)  SET @sqltemp100= 'Drop login ' + @tentk;  EXEC (@sqltemp100)  DELETE FROM dbo.NhanVien WHERE MaNV = @manv  END  END |

**3.2.2 Thông tin nhà sản xuất**

Thủ tục thêm nhà sản xuất

|  |
| --- |
| CREATE PROC sp\_ThemNSX  @tennsx NVARCHAR(255), @diachi NVARCHAR(255)  AS  BEGIN  INSERT INTO dbo.NhaSanXuat  (  TenNSX,  DiaChi  )  VALUES  ( @tennsx,  @diachi  )  END |

Thủ tục sửa nhà sản xuất

|  |
| --- |
| CREATE PROC sp\_SuaNSX  @mansx INT, @tennsx NVARCHAR(255), @diachi NVARCHAR(255)  AS  BEGIN  UPDATE dbo.NhaSanXuat SET TenNSX = @tennsx, DiaChi= @diachi WHERE MaNSX = @mansx  END |

Thủ tục xóa nhà sản xuất

|  |
| --- |
| CREATE PROC sp\_XoaNSX  @mansx INT  AS  BEGIN  DELETE FROM dbo.NhaSanXuat WHERE MaNSX = @mansx  END |

**3.2.3 Sản phẩm**

Thủ tục hêm sản phẩm

|  |
| --- |
| CREATE PROCEDURE sp\_ThemSanPham  @tensp NVARCHAR(255),  @mansx INT,  @maloai INT,  @giaban FLOAT,  @giagoc FLOAT,  @trangthai BIT  AS  BEGIN  INSERT INTO dbo.SanPham  (TenSP,MaNSX,MaLoai,GiaBan,GiaGoc,TrangThai  )  VALUES  (@tensp, @mansx, @maloai, @giaban, @giagoc, @trangthai)  END |

Thủ tục sửa sản phẩm

|  |
| --- |
| CREATE PROCEDURE sp\_SuaSanPham  @masp INT,  @tensp NVARCHAR(255),  @mansx INT,  @maloai INT,  @giaban FLOAT,  @giagoc FLOAT,  @trangthai BIT  AS  BEGIN  UPDATE dbo.SanPham SET  TenSP = @tensp,  MaNSX = @mansx,  MaLoai = @maloai,  GiaBan = @giaban,  GiaGoc = @giagoc,  TrangThai = @trangthai  WHERE MaSP = @masp  END |

Thủ tục xóa sản phẩm

|  |
| --- |
| CREATE PROCEDURE sp\_XoaSanPham  @masp INT  AS  BEGIN  IF EXISTS (SELECT \* FROM dbo.Kho WHERE MaSP = @masp)  RAISERROR ('Bạn đã xóa sản phẩm đang có ở Kho. Không xóa được!', 16, 1)  ELSE  DELETE FROM dbo.SanPham WHERE MaSP = @masp  END |

**3.2.4 Kho**

Thủ tục nhập kho

|  |
| --- |
| CREATE PROCEDURE sp\_NhapKho  @masp INT,  @nsx DATE,  @hsd DATE,  @ngaynhapkho DATE,  @loaikho BIT,  @sltonkho INT  AS  BEGIN  IF EXISTS (SELECT \* FROM dbo.Kho WHERE MaSP = @masp AND NSX = @nsx AND HSD = @hsd)  UPDATE dbo.Kho SET SLTonKho += @sltonkho  WHERE MaSP = @masp AND NSX = @nsx AND HSD = @hsd  ELSE  INSERT INTO dbo.Kho (MaSP, NSX, HSD, NgayNhapKho,LoaiKho,SLTonKho)  VALUES (@masp, @nsx, @hsd, @ngaynhapkho, @loaikho, @sltonkho)  END |

Thủ tục xóa sản phẩm vào kho

|  |
| --- |
| CREATE PROCEDURE sp\_XoaSpVaoKho  @masp INT,  @nsx DATE,  @hsd DATE  AS  BEGIN  DELETE FROM dbo.Kho WHERE MaSP = @masp AND NSX = @nsx AND HSD = @hsd  END |

Thủ tục sửa sản phẩm vào kho

|  |
| --- |
| CREATE PROCEDURE sp\_SuaSpVaoKho  @masp INT,  @nsx DATE,  @hsd DATE,  @ngaynhapkho DATE,  @loaikho BIT,  @sltonkho INT  AS  BEGIN  UPDATE dbo.Kho SET  NgayNhapKho = @ngaynhapkho,  LoaiKho = @loaikho,  SLTonKho = @sltonkho  WHERE MaSP = @masp AND NSX = @nsx AND HSD = @hsd  END |

**3.2.5 Cửa hàng**

Thủ tục xuất kho

|  |
| --- |
| CREATE PROCEDURE sp\_XuatKho  @masp INT,  @nsx DATE,  @hsd DATE,  @ngayxuatkho DATE,  @soluong INT  AS  BEGIN  DECLARE @sltk INT  SELECT @sltk = SLTonKho FROM Kho WHERE MaSP = @masp AND NSX = @nsx AND HSD = @hsd  IF (@sltk >= @soluong)  BEGIN  UPDATE Kho SET SLTonKho -= @soluong  WHERE MaSP = @masp AND NSX = @nsx AND HSD = @hsd  IF EXISTS (SELECT \* FROM dbo.CuaHang WHERE MaSP = @masp AND NSX = @nsx AND HSD = @hsd)  UPDATE dbo.CuaHang SET SoLuong += @soluong  WHERE MaSP = @masp AND NSX = @nsx AND HSD = @hsd  ELSE  INSERT INTO dbo.CuaHang(MaSP, NSX, HSD, NgayXuatKho, SoLuong)  VALUES (@masp, @nsx, @hsd, @ngayxuatkho, @soluong)  END  ELSE  RAISERROR('Số lượng trong Kho không đủ để thêm Sản Phẩm vào Cửa Hàng', 16,1)  END |

Thủ tục trả kho

|  |
| --- |
| CREATE PROCEDURE sp\_TraKho  @masp INT,  @nsx DATE,  @hsd DATE,  @ngayxuatkho DATE,  @soluong INT  AS  BEGIN  DECLARE @sltrakho INT  SELECT @sltrakho = SoLuong FROM dbo.CuaHang WHERE MaSP = @masp AND NSX = @nsx AND HSD = @hsd  IF (@sltrakho >= @soluong)  BEGIN  UPDATE dbo.CuaHang SET SoLuong -= @soluong WHERE MaSP = @masp AND NSX = @nsx AND HSD = @hsd  UPDATE dbo.Kho SET SLTonKho += @soluong WHERE MaSP = @masp AND NSX = @nsx AND HSD = @hsd  END  ELSE  RAISERROR('Số lượng Trả Kho lớn hơn số lượng hiện còn có trong Cửa Hàng', 16,1)  END |

**3.2.6 Khách Hàng**

Thủ tục thêm khách hàng

|  |
| --- |
| CREATE PROCEDURE sp\_ThemKhachHang  @sdt NVARCHAR(10),  @tenkh NVARCHAR(255),  @diemtichluy FLOAT  AS  BEGIN  INSERT INTO dbo.KhachHang  (Sdt,TenKH,DiemTichLuy)  VALUES  (@sdt,@tenkh, @diemtichluy)  END |

Thủ tục sửa khách hàng

|  |
| --- |
| CREATE PROCEDURE sp\_SuaKhachHang  @sdt NVARCHAR(10),  @tenkh NVARCHAR(255),  @diemtichluy INT  AS  BEGIN  UPDATE dbo.KhachHang SET  TenKH = @tenkh,  DiemTichLuy = @diemtichluy  WHERE SDT = @sdt  END |

Thủ tục xóa khách hàng

|  |
| --- |
| CREATE PROCEDURE sp\_XoaKhachHang  @sdt NVARCHAR(10)  AS  BEGIN  DELETE FROM dbo.KhachHang WHERE sdt = @sdt  END |

**3.2.7 Ngày lễ**

Thủ tục thêm ngày lễ

|  |
| --- |
| CREATE PROC sp\_ThemNgayLe  @ngay DATE, @sukien NVARCHAR(255)  AS  BEGIN  INSERT INTO dbo.NgayLe  (  Ngay,  SuKien  )  VALUES  ( @ngay,  @sukien  )  END |

Thủ tục sửa ngày lễ

|  |
| --- |
| CREATE PROC sp\_SuaNgayLe  @manl INT, @ngay DATE, @sukien NVARCHAR(255)  AS  BEGIN  UPDATE dbo.NgayLe SET Ngay = @ngay, SuKien = @sukien WHERE MaNL = @manl  END |

Thủ tục xóa ngày lễ

|  |
| --- |
| CREATE PROC sp\_XoaNgayLe  @manl INT  AS  BEGIN  DELETE FROM dbo.NgayLe WHERE MaNL = @manl  END |

**3.2.8 Hóa đơn**

Thủ tục thêm hóa đơn

|  |
| --- |
| CREATE PROCEDURE sp\_ThemHoaDon  @manv NVARCHAR(10),  @sdtkh NVARCHAR(10),  @pttt INT  AS  BEGIN  INSERT INTO dbo.HoaDon(MaNV,sdtkh,PTTT,Ngay,TongTienTT, TongTienLoi)  VALUES (@manv, @sdtkh, @pttt, GETDATE(), 0.0, 0.0)  END |

Thủ tục xóa hóa đơn

|  |
| --- |
| CREATE PROCEDURE sp\_XoaHoaDon  @mahd INT  AS  BEGIN  DELETE FROM dbo.ChiTietHoaDon WHERE MaHD = @mahd  DELETE FROM dbo.HoaDon WHERE MaHD = @mahd  END |

**3.2.9 Thông tin chi tiết hóa đơn**

Thủ tục thêm chi tiết hóa đơn

|  |
| --- |
| CREATE PROCEDURE sp\_ThemChiTietHoaDon  @mahd INT,  @masp INT,  @nsx DATE,  @hsd DATE,  @sl INT  AS  BEGIN  DECLARE  @giaban FLOAT,  @giagoc FLOAT,  @thanhtien FLOAT,  @tienloi FLOAT  SET @giaban = 0  SET @giagoc = 0  SET @thanhtien = 0  SET @tienloi = 0  DECLARE @MaHDMoi INT  SET @MaHDMoi = @mahd  DECLARE @slch INT  SELECT @slch = SoLuong FROM CuaHang WHERE MaSP = @masp AND NSX = @nsx AND HSD = @hsd  IF (@slch >= @sl )  BEGIN  IF NOT EXISTS (SELECT \* FROM dbo.HoaDon WHERE MaHD = @mahd)  BEGIN  EXEC dbo.sp\_ThemHoaDon @manv = NULL, -- nvarchar(10)  @sdtkh = NULL, -- nvarchar(10)  @pttt = NULL -- nvarchar(255)  SELECT @MaHDMoi = MAX(MaHD) FROM dbo.HoaDon  END;  SELECT @giaban = GiaBan, @giagoc = GiaGoc FROM dbo.SanPham WHERE @masp = MaSP  IF EXISTS (SELECT MaHD, MaSP, NSX, HSD FROM dbo.ChiTietHoaDon WHERE MaHD = @MaHDMoi AND MaSP = @masp AND NSX = @nsx AND HSD = @hsd)  BEGIN    UPDATE dbo.ChiTietHoaDon SET SoLuong += @sl WHERE MaHD = @MaHDMoi AND MaSP = @masp AND NSX = @nsx AND HSD = @hsd  UPDATE dbo.ChiTietHoaDon SET ThanhTien = SoLuong \* GiaBan WHERE MaHD = @MaHDMoi AND MaSP = @masp AND NSX = @nsx AND HSD = @hsd  UPDATE dbo.ChiTietHoaDon SET TienLoi = ThanhTien - (GiaGoc \* SoLuong) WHERE MaHD = @MaHDMoi AND MaSP = @masp AND NSX = @nsx AND HSD = @hsd  SELECT @thanhtien = ThanhTien, @tienloi = TienLoi FROM dbo.ChiTietHoaDon WHERE MaHD = @MaHDMoi AND MaSP = @masp AND NSX = @nsx AND HSD = @hsd  END;  ELSE  BEGIN  SET @thanhtien = @sl \* @giaban  SET @tienloi = @thanhtien - (@giagoc \* @sl)  INSERT INTO dbo.ChiTietHoaDon (MaHD,MaSP, NSX, HSD ,SoLuong,GiaBan,GiaGoc,ThanhTien,TienLoi)  VALUES (@MaHDMoi, @masp, @nsx, @hsd, @sl, @giaban, @giagoc, @thanhtien, @tienloi)  END;  UPDATE dbo.HoaDon SET TongTienTT += @thanhtien WHERE MaHD = @MaHDMoi  UPDATE dbo.HoaDon SET TongTienLoi += @tienloi WHERE MaHD = @MaHDMoi  END  ELSE  BEGIN  RAISERROR('Sản Phẩm không còn đủ trong Cửa Hàng để Khách Hàng mua!', 16, 1)  END  END |

Thủ tục xóa chi tiết hóa đơn

|  |
| --- |
| CREATE PROCEDURE sp\_XoaChiTietHoaDon  @mahd INT,  @masp INT,  @nsx DATE,  @hsd DATE  AS  BEGIN  DELETE FROM dbo.ChiTietHoaDon WHERE MaHD = @mahd AND MaSP = @masp AND NSX = @nsx AND HSD = @hsd  END |

**3.2.10 Thông tin phân công**

Thủ tục thêm phân công

|  |
| --- |
| CREATE PROCEDURE sp\_ThemPhanCong  @manv NVARCHAR(10),  @maca NVARCHAR(10),  @ngaydangky DATE  AS  BEGIN  INSERT INTO dbo.PhanCong  (MaNV,MaCa,NgayDangKy)  VALUES  ( @manv, @maca, @ngaydangky)  DECLARE @maxmapc int  SELECT @maxmapc = MAX(MaPC) FROM dbo.PhanCong  INSERT dbo.ChamCong  (MaPC, GioBD,GioKT,Luong)  VALUES(@maxmapc, NULL, NULL, NULL)  END  GO |

Thủ tục xóa phân công

|  |
| --- |
| CREATE PROCEDURE sp\_XoaPhanCong  @mapc INT  AS  BEGIN  DELETE FROM dbo.ChamCong WHERE MaPC = @mapc  DELETE FROM PhanCong Where MaPC = @mapc  END |

**3.2.11 Hàm tính doanh thu**

|  |
| --- |
| CREATE FUNCTION TinhDoanhThu()  RETURNS NVARCHAR(255)  as  BEGIN  DECLARE @result FLOAT  set @RESULT = 0  SELECT @result +=TongTienTT FROM dbo.HoaDon  RETURN FORMAT(@result, 'N')  END |

**3.2.12 Hàm lọc doanh thu**

|  |
| --- |
| CREATE FUNCTION LocDoanhThu(  @ngaybatdau DATE,  @ngayketthuc DATE)  RETURNS TABLE  AS  RETURN  (  SELECT \* FROM dbo.HoaDon WHERE @ngaybatdau <= Ngay AND Ngay <= @ngayketthuc  ) |

**3.2.13 Thủ tục Update lại cửa hàng khi thanh toán**

|  |
| --- |
| CREATE PROCEDURE UpdateLaiCuaHangKhiThanhToan  AS  BEGIN    DECLARE @bangthanhtoan TABLE (masptt INT, nsxtt DATE, hsdtt DATE, sl int)  INSERT INTO @bangthanhtoan (masptt, nsxtt, hsdtt, sl)  SELECT MaSP, NSX, HSD, SoLuong  FROM dbo.BangChiTietHoaDonThanhToan()  SELECT \* FROM @bangthanhtoan  DECLARE @counter INT = 1  WHILE @counter <= (SELECT COUNT(\*) FROM @bangthanhtoan)  BEGIN  DECLARE @masptt INT, @nsxtt DATE, @hsdtt DATE, @sl INT  SELECT TOP 1 @masptt =masptt, @nsxtt = nsxtt, @hsdtt = hsdtt, @sl = sl FROM @bangthanhtoan ORDER BY masptt, nsxtt, hsdtt  DELETE FROM @bangthanhtoan WHERE @masptt =masptt AND @nsxtt = nsxtt AND @hsdtt = hsdtt  -- Thực hiện các hành động khác trên bản ghi này  UPDATE CuaHang SET SoLuong -= @sl  WHERE MaSP = @masptt AND NSX = @nsxtt AND HSD = @hsdtt  -- Tăng biến đếm để xử lý bản ghi tiếp theo  END  END |

**3.2.14 Thủ tục UpdateGioBD**

|  |
| --- |
| CREATE PROCEDURE UpDateGioBD  @mapc INT,  @giobd TIME  AS  BEGIN  DECLARE @giobatdauca TIME, @gioketthucca TIME  SELECT @giobatdauca = GioBatDau, @gioketthucca = GioBatDau  FROM dbo.PhanCong INNER JOIN Ca ON Ca.MaCa = PhanCong.MaCa  WHERE @mapc = MaPC  SET @giobatdauca = DATEADD(MINUTE, -5, @giobatdauca)  SET @gioketthucca = DATEADD(MINUTE, 5, @gioketthucca)  IF @giobatdauca <= @giobd AND @giobd <= @gioketthucca  UPDATE dbo.ChamCong SET GioBD = @giobd WHERE MaPC = @mapc  ELSE  RAISERROR('Bạn đã nhập giờ bắt đầu không nằm trong khoảng thời gian được phân công!', 16, 1)  END |

**3.2.15 Thủ tục UpdateGioKT**

|  |
| --- |
| CREATE PROCEDURE UpDateGioKT  @mapc INT,  @giokt TIME  AS  BEGIN  DECLARE @giobatdauca TIME, @gioketthucca TIME  SELECT @giobatdauca = GioKetThuc, @gioketthucca = GioKetThuc  FROM dbo.PhanCong INNER JOIN Ca ON Ca.MaCa = PhanCong.MaCa  WHERE @mapc = MaPC  SET @giobatdauca = DATEADD(MINUTE, -5, @giobatdauca)  SET @gioketthucca = DATEADD(MINUTE, 5, @gioketthucca)  IF @giobatdauca <= @giokt AND @giokt <= @gioketthucca  UPDATE dbo.ChamCong SET GioKT = @giokt WHERE MaPC = @mapc  ELSE  RAISERROR('Bạn đã nhập giờ kết thúc không nằm trong khoảng thời gian được phân công!', 16, 1)  END |

**3.2.16 Hàm Tính Lương một Chấm Công**

|  |
| --- |
| CREATE FUNCTION TinhLuongCC(@mapc INT)  RETURNS FLOAT  AS  BEGIN  DECLARE @luongtheogio FLOAT  SELECT @luongtheogio = LuongTheoGio FROM dbo.ChamCong, dbo.PhanCong, dbo.NhanVien, dbo.ChucVu  WHERE ChamCong.MaPC = dbo.PhanCong.MaPC AND dbo.PhanCong.MaNV = dbo.NhanVien.MaNV AND dbo.NhanVien.MaCV = dbo.ChucVu.MaCV AND ChamCong.MaPC = @mapc  DECLARE @maca NVARCHAR(255)  SELECT @maca = MaCa FROM dbo.ChamCong, dbo.PhanCong, dbo.NhanVien, dbo.ChucVu  WHERE ChamCong.MaPC = dbo.PhanCong.MaPC AND dbo.PhanCong.MaNV = dbo.NhanVien.MaNV AND dbo.NhanVien.MaCV = dbo.ChucVu.MaCV AND ChamCong.MaPC = @mapc  DECLARE @result FLOAT  IF @maca LIKE 'E'  BEGIN  SET @result = @luongtheogio \* 6  SET @result = @result \* 125 / 100  END  ELSE  SET @result = @luongtheogio \* 4  DECLARE @ngaychamcong DATE  SELECT @ngaychamcong = NgayDangKy FROM dbo.ChamCong, dbo.PhanCong, dbo.NhanVien, dbo.ChucVu  WHERE ChamCong.MaPC = dbo.PhanCong.MaPC AND dbo.PhanCong.MaNV = dbo.NhanVien.MaNV AND dbo.NhanVien.MaCV = dbo.ChucVu.MaCV AND ChamCong.MaPC = @mapc  IF EXISTS(SELECT \* FROM dbo.NgayLe WHERE @ngaychamcong = Ngay)  SET @result = @result \* 3  RETURN @result  END |

**3.2.17 Lương**

Thủ tục thêm tính lương

|  |
| --- |
| CREATE PROCEDURE ThemTinhLuong  @mapc INT  AS  BEGIN  DECLARE @luongca FLOAT  DECLARE @manv NVARCHAR(255)  DECLARE @ngay DATE  SELECT @luongca = Luong, @manv = NhanVien.MaNV, @ngay = NgayDangKy FROM dbo.ChamCong, dbo.PhanCong, dbo.NhanVien, dbo.ChucVu  WHERE ChamCong.MaPC = dbo.PhanCong.MaPC  AND dbo.PhanCong.MaNV = dbo.NhanVien.MaNV  AND dbo.NhanVien.MaCV = dbo.ChucVu.MaCV  AND ChamCong.MaPC = @mapc  IF EXISTS(SELECT \* FROM dbo.TinhLuong WHERE MaNV = @manv AND Ngay = @ngay)  UPDATE dbo.TinhLuong SET TongLuong += @luongca WHERE MaNV = @manv AND Ngay = @ngay  ELSE  INSERT INTO dbo.TinhLuong  (  MaNV,  Ngay,  TongLuong  )  VALUES  ( @manv, -- MaNV - nvarchar(10)  @ngay, -- Ngay - date  @luongca -- TongLuong - float  )  END |

Thủ tục sửa tính lương

|  |
| --- |
| CREATE PROCEDURE UpdateTinhLuong  @mapc INT  AS  BEGIN  UPDATE dbo.ChamCong SET Luong = dbo.TinhLuongCC(@mapc) WHERE MaPC = @mapc  END |

**3.2.18 Hàm bảng chi tiết hóa đơn thanh toán**

|  |
| --- |
| CREATE FUNCTION BangChiTietHoaDonThanhToan()  RETURNS TABLE  AS  RETURN  (  SELECT MaSP, NSX, HSD, SoLuong FROM dbo.ChiTietHoaDon WHERE MaHD = dbo.MaxHD()  ) |

**3.2.19 Hàm tính MaxHD**

|  |
| --- |
| CREATE FUNCTION MaxHD()  RETURNS int  AS  BEGIN  DECLARE @result int  SELECT @result = MAX(MaHD) FROM dbo.HoaDon  RETURN @result  END |

**CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN**

**Ứng với hai nhóm người dùng( Nhân viên và quản lý) là các role riêng**

+ Tạo role nhân viên

|  |
| --- |
| CREATE ROLE RoleNhanVien |

+ Quản lý được phân quyền với role là SYSADMIN trong thủ tục thêm nhân viên

|  |
| --- |
| DECLARE @sqltemp2 nvarchar(2000)  SET @sqltemp2= 'ALTER SERVER ROLE sysadmin ADD MEMBER ' + @tentk  EXEC (@sqltemp2) |

+ Đối với Role Nhân Viên: Tạo mới Database Role Nhân Viên và gán các quyền cần thiết cho role

|  |
| --- |
| GRANT SELECT, UPDATE ON dbo.NhanVien TO RoleNhanVien  GRANT EXECUTE ON dbo.sp\_SuaNhanVien TO RoleNhanVien  GRANT EXECUTE ON dbo.MaxHD TO RoleNhanVien  GRANT SELECT, UPDATE on dbo.CuaHang TO RoleNhanVien  GRANT SELECT ON dbo.NgayLe TO RoleNhanVien  GRANT SELECT ON f\_TimKiemKhachHang TO RoleNhanVien  GRANT SELECT ON f\_TimKiemCuaHang TO RoleNhanVien  GRANT SELECT ON ViewCuaHang TO RoleNhanVien  GRANT SELECT, UPDATE on dbo.ChucVu TO RoleNhanVien  GRANT SELECT, UPDATE, INSERT, DELETE on dbo.KhachHang TO RoleNhanVien  GRANT EXECUTE ON dbo.sp\_ThemKhachHang TO RoleNhanVien  GRANT EXECUTE ON dbo.sp\_SuaKhachHang TO RoleNhanVien  GRANT EXECUTE ON dbo.sp\_XoaKhachHang TO RoleNhanVien  GRANT SELECT, UPDATE, INSERT, DELETE on dbo.HoaDon TO RoleNhanVien  GRANT EXECUTE ON dbo.sp\_ThemHoaDon TO RoleNhanVien  GRANT EXECUTE ON dbo.sp\_XoaHoaDon TO RoleNhanVien  GRANT EXECUTE ON dbo.UpdateLaiCuaHangKhiThanhToan TO RoleNhanVien  GRANT SELECT, UPDATE, INSERT, DELETE on dbo.ChiTietHoaDon TO RoleNhanVien  GRANT EXECUTE ON dbo.sp\_ThemChiTietHoaDon TO RoleNhanVien  GRANT EXECUTE ON dbo.sp\_XoaChiTietHoaDon TO RoleNhanVien  GRANT SELECT ON dbo.ChiTietHoaDonThanhToan TO RoleNhanVien  GRANT SELECT, UPDATE, INSERT, DELETE on dbo.PhanCong TO RoleNhanVien  GRANT EXECUTE ON dbo.sp\_ThemPhanCong TO RoleNhanVien  GRANT EXECUTE ON dbo.sp\_XoaPhanCong TO RoleNhanVien  GRANT SELECT, UPDATE, INSERT, DELETE on dbo.ChamCong TO RoleNhanVien  GRANT EXECUTE ON dbo.UpDateGioBD TO RoleNhanVien  GRANT EXECUTE ON dbo.UpDateGioKT TO RoleNhanVien  GRANT EXECUTE ON dbo.UpdateTinhLuong TO RoleNhanVien  GRANT EXECUTE ON dbo.TinhLuongCC TO RoleNhanVien  GRANT SELECT, UPDATE, INSERT ON dbo.TinhLuong TO RoleNhanVien  GRANT EXECUTE ON dbo.ThemTinhLuong TO RoleNhanVien  GRANT SELECT on dbo.PhuongThucThanhToan TO RoleNhanVien  GRANT SELECT on dbo.SanPham TO RoleNhanVien  GRANT SELECT, UPDATE, INSERT, DELETE on dbo.Ca TO RoleNhanVien  GRANT SELECT, UPDATE, INSERT, DELETE on dbo.XemBangPC\_NhanVien TO RoleNhanVien |

**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 18.

- Chương trình Winform được viết bằng Microsoft Visual Studio 2022 - trên nền tảng ASP.NET Web Application (.NET Framework) C#.

**Giao diện và chức năng của chương trình:**

**5.1 Form đăng nhập**

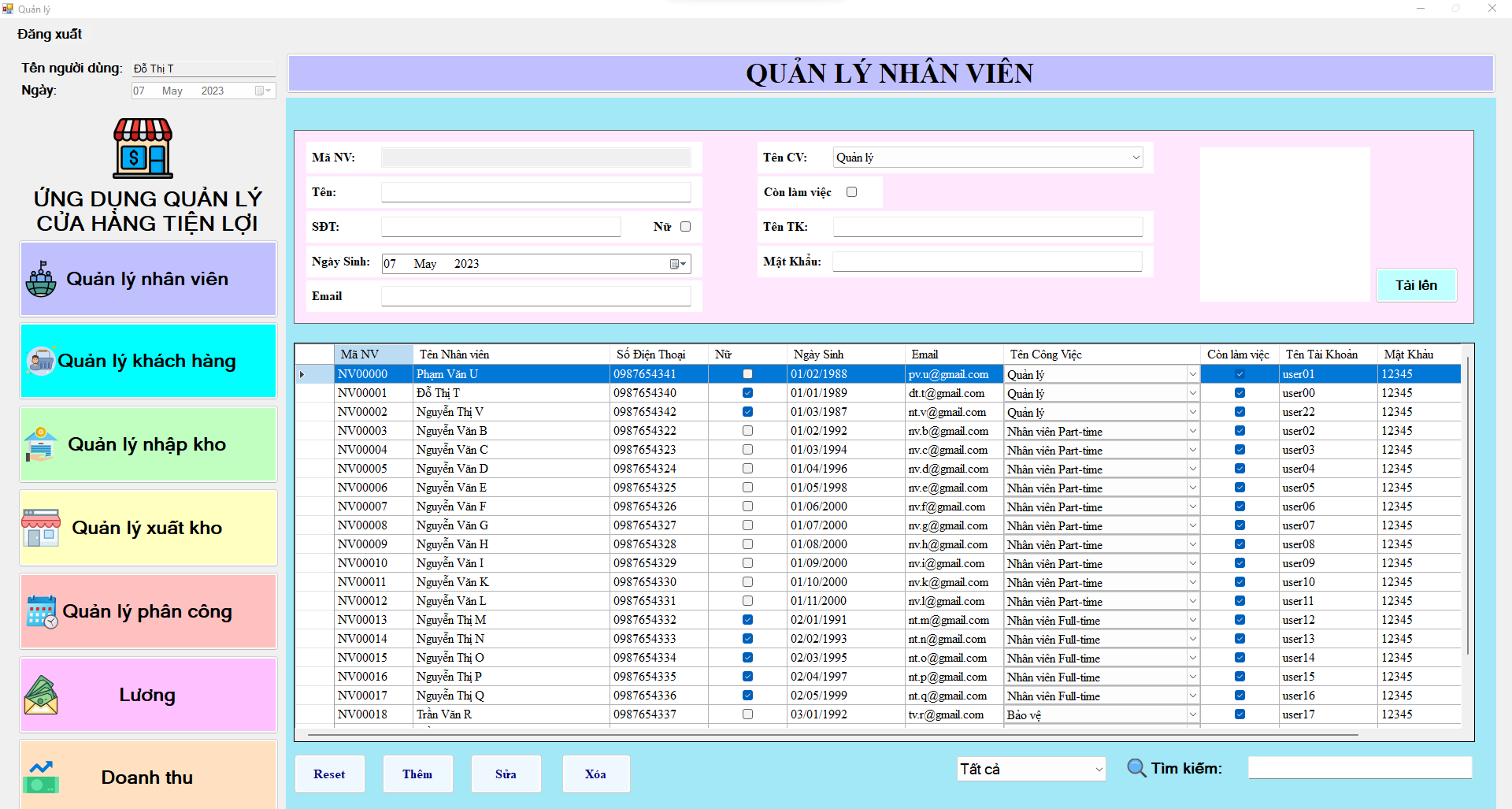
****

**5.2 Form home**

- Đối với nhân viên:

****

- Đối với quản lý

****

**5.3 Thao tác đăng nhập**

Tạo đối tượng biến toàn cục NhanVien, khi đăng nhập sẽ lưu TenTK và MatKhau nhập vào từ người dùng.

|  |
| --- |
| private void btnLogin\_Click(object sender, EventArgs e)  {  try  {  Program.nv = nvDAO.KiemTraDangNhap(tbxUserName.Text, tbxPassword.Text);  if (Program.nv != null)  {  if (rbQuanLy.Checked == true)  {  if (Program.nv.MaCV == 1)  {  fQuanLy ql = new fQuanLy(Program.nv);  this.Hide();  ql.ShowDialog();  this.Show();  }  else  MessageBox.Show("Bạn không có quyền truy cập vào quản lý!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  else  {  fNhanVien form = new fNhanVien(Program.nv);  this.Hide();  form.ShowDialog();  this.Show();  }  }  else  MessageBox.Show("Đăng nhập không thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  catch (Exception ex)  {  MessageBox.Show("Đăng nhập thất bại!\n" + ex, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

Nhận đối tượng NhanVien từ TenTK và MatKhau

|  |
| --- |
| public NhanVien KiemTraDangNhap(string tentk, string matkhau)  {  string query = string.Format("SELECT \* FROM dbo.NhanVien WHERE TenTK = N'{0}' AND MatKhau = N'{1}'", tentk, matkhau);  DataTable result = db.LayDanhSach(query);  foreach (DataRow dr in result.Rows)  {  return new NhanVien(dr);  }  return null;  } |

Kết nối Username và Password của Login SQL vào C#

|  |
| --- |
| SqlConnection conn = null;  public DBConnection(NhanVien nv)  {  string cnnStr = "Data Source = (localdb)\\mssqllocaldb;Initial Catalog = QuanLyCuaHangTienLoi; " +  $"User ID= {nv.TenTK};Password= {nv.MatKhau}";  conn = new SqlConnection(cnnStr);  } |

**5.4 Các hàm kết nối và gọi thực thi chung từ SQL**

Hàm lấy danh sách

|  |
| --- |
| public DataTable LayDanhSach(string sqlStr)  {  try  {  conn.Open();  SqlDataAdapter adapter = new SqlDataAdapter(sqlStr, conn);  DataTable data = new DataTable();  adapter.Fill(data);  return data;  }  catch (SqlException ex)  {  MessageBox.Show("Lấy danh sách thất bại\n" + ex.Message, "Thông báo");  }  finally  {  conn.Close();  }  return null;  } |

Hàm lấy giá trị string

|  |
| --- |
| public String LayGiaTri(string sqlStr)  {  try  {  conn.Open();  SqlCommand cmd = new SqlCommand(sqlStr, conn);  String data = cmd.ExecuteScalar().ToString();  return data;  }  catch (SqlException ex)  {  MessageBox.Show("Lấy giá trị thất bại\n" + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  conn.Close();  }  return null;  } |

Hàm thực thi câu lệnh

|  |
| --- |
| public void Execute(string query)  {  try  {  conn.Open();  SqlCommand cmd = new SqlCommand(query, conn);  if (cmd.ExecuteNonQuery() < 0)  MessageBox.Show("Thao tác thất bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  catch (SqlException ex)  {  MessageBox.Show("Thất bại\n" + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  conn.Close();  }  } |

**5.5 Các hàm gọi thủ tục, hàm, view từ SQL trong lớp DAO (Data Access Object)**

**5.5.1 Ca**

|  |
| --- |
| public DataTable LayDanhSach()  {  return exec.LayDanhSach("Select \* from Ca");  } |
| public Ca LayThongTinCaBangMaCa(string macv)  {  string query = string.Format($"SELECT \* FROM dbo.Ca WHERE MaCa = N'{macv}'");  DataTable result = exec.LayDanhSach(query);  foreach (DataRow dr in result.Rows)  {  return new Ca(dr);  }  return null;  } |

**5.5.2 Chấm công**

|  |
| --- |
| public DataTable LayDanhSach(int mapc)  {  return exec.LayDanhSach($"Select \* from ChamCong WHERE MaPC = {mapc}");  } |
| public void CheckIn(ChamCong cc)  {  string query = string.Format($"EXEC dbo.UpDateGioBD @mapc = {cc.MaPC}," +  $"@giobd = '{cc.GioBD}'");  exec.Execute(query);  } |
| public void CheckOut(ChamCong cc)  {  string query = string.Format($"EXEC dbo.UpDateGioKT @mapc = {cc.MaPC}," +  $"@giokt = '{cc.GioKT}'");  exec.Execute(query);  TinhLuong(cc);  ThemTinhLuong(cc);  } |
| public void TinhLuong(ChamCong cc)  {  string query = string.Format($"EXEC dbo.UpdateTinhLuong @mapc = {cc.MaPC}");  exec.Execute(query);  } |
| public void ThemTinhLuong(ChamCong cc)  {  string query = string.Format($"EXEC dbo.ThemTinhLuong @mapc = {cc.MaPC}");  exec.Execute(query);  } |
| public ChamCong LayThongTinChamCongBangMaPC(int mapc)  {  string query = string.Format("SELECT \* FROM dbo.ChamCong WHERE MaPC = {0}", mapc);  DataTable result = exec.LayDanhSach(query);  foreach (DataRow dr in result.Rows)  {  return new ChamCong(dr);  }  return null;  } |

**5.5.3 Chi tiết hóa đơn**

|  |
| --- |
| public DataTable LayDanhSach()  {  string sql = "Select \* from ChiTietHoaDon";  return db.LayDanhSach(sql);  } |
| public DataTable LayDanhSachTheoHoaDon(int mahd)  {  string sql = $"Select \* from ChiTietHoaDon Where mahd = {mahd}";  return db.LayDanhSach(sql);  } |
| public DataTable LayDanhSachThanhToan()  {  string sql = "Select \* from ChiTietHoaDonThanhToan";  return db.LayDanhSach(sql);  } |
| public void Order(ChiTietHoaDon cthd)  {  string query = String.Format($"EXEC dbo.sp\_ThemChiTietHoaDon @mahd = N'{cthd.MaHD}'," +  $"@masp= N'{cthd.MaSP}'," +  $"@nsx = N'{cthd.NSX.ToString("yyyy-MM-dd")}'," +  $"@hsd = N'{cthd.HSD.ToString("yyyy-MM-dd")}'," +  $"@sl = N'{cthd.SoLuong}'");  db.Execute(query);    } |
| public DataTable LayDanhSachThanhToanRong()  {  string sql = "Select \* from ChiTietHoaDonThanhToan Where 0=1";  return db.LayDanhSach(sql);  } |

**5.5.4 Chức vụ**

|  |
| --- |
| public DataTable LayDanhSach()  {  return exec.LayDanhSach("Select \* from ChucVu");  } |
| public ChucVu LayThongTinChucVuBangMaCV(int macv)  {  string query = string.Format($"SELECT \* FROM dbo.ChucVu WHERE MaCV = {macv}");  DataTable result = exec.LayDanhSach(query);  foreach (DataRow dr in result.Rows)  {  return new ChucVu(dr);  }  return null;  } |

**5.5.5 Cửa hàng**

|  |
| --- |
| public DataTable LayDanhSach()  {  string sql = "select \* from viewcuahang";  return db.LayDanhSach(sql);  } |
| public DataTable LayDanhSach\_ConHan()  {  string sql = "select \* from viewcuahang WHERE HSD >= GetDATE()";  return db.LayDanhSach(sql);  } |
| public DataTable LayDanhSach\_HetHan()  {  string sql = "select \* from viewcuahang WHERE HSD < GetDATE()";  return db.LayDanhSach(sql);  } |
| public void XuatKho(CuaHang ch)  {  string query = String.Format($"EXEC dbo.sp\_XuatKho @masp = N'{ch.MaSP}'," +  $"@nsx = N'{ch.NSX.ToString("yyyy-MM-dd")}'," +  $"@hsd = N'{ch.HSD.ToString("yyyy-MM-dd")}'," +  $"@ngayxuatkho = N'{ch.NgayXuatKho.ToString("yyyy-MM-dd")}'," +  $"@soluong = N'{ch.SoLuong}'");  db.Execute(query);  } |
| public void TraKho(CuaHang ch)  {  string query = String.Format($"EXEC dbo.sp\_TraKho @masp = N'{ch.MaSP}'," +  $"@nsx = N'{ch.NSX.ToString("yyyy-MM-dd")}'," +  $"@hsd = N'{ch.HSD.ToString("yyyy-MM-dd")}'," +  $"@ngayxuatkho = N'{ch.NgayXuatKho.ToString("yyyy-MM-dd")}'," +  $"@soluong = N'{ch.SoLuong}'");  db.Execute(query);  } |
| public DataTable TimKiem(string find)  {  string query = string.Format($"SELECT \* FROM dbo.f\_TimKiemCuaHang(N'{find}')");  return db.LayDanhSach(query);  } |
| public DataTable TimKiem\_ConHan(string find)  {  string query = string.Format($"SELECT \* FROM dbo.f\_TimKiemCuaHang(N'{find}') WHERE HSD >= GetDATE()");  return db.LayDanhSach(query);  } |
| public DataTable TimKiem\_HetHan(string find)  {  string query = string.Format($"SELECT \* FROM dbo.f\_TimKiemCuaHang(N'{find}') WHERE HSD < GetDATE()");  return db.LayDanhSach(query);  } |

**5.5.6 Hóa đơn**

|  |
| --- |
| public DataTable LayDanhSach()  {  string sql = "Select \* from HoaDon";  return db.LayDanhSach(sql);  } |
| public DataTable LayDanhSachTheoKhachHang(int sdtkh)  {  string sql = $"Select \* from HoaDon Where SDTKH = {sdtkh}";  return db.LayDanhSach(sql);  } |
| public void ThanhToan()  {  string sql = $"EXECUTE dbo.UpdateLaiCuaHangKhiThanhToan";  db.Execute(sql);  } |
| public void HuyThanhToan(int mahd)  {  string sql = $"EXECUTE dbo.sp\_XoaHoaDon @mahd = {mahd}";  db.Execute(sql);  } |
| public void TaoHD()  {  string sql = "EXEC dbo.sp\_ThemHoaDon @manv = NULL, @sdtkh = NULL, @pttt = NULL";  db.Execute(sql);  } |
| public int MaxHD()  {  string query = String.Format("SELECT dbo.MaxHD()");  return Convert.ToInt32(db.LayGiaTri(query));  } |
| public void TaoHD(HoaDon hd)  {  string query = String.Format($"EXEC dbo.sp\_ThemHoaDon @manv = N'{hd.MaNV}'," +  $"@sdtkh = {hd.SdtKH}," +  $"@pttt = N'{hd.PTTT}'");  db.Execute(query);  } |
| public string TinhDoanhThu()  {  string sql = "SELECT dbo.TinhDoanhThu()";  return db.LayGiaTri(sql);  } |
| public DataTable LocDoanhThu(DateTime Tu, DateTime Den)  {  string sql = $"Select \* From dbo.LocDoanhThu('{Tu.ToString("yyyy-MM-dd")}', '{Den.ToString("yyyy-MM-dd")}')";  return db.LayDanhSach(sql);  } |

**5.5.7 Khách hàng**

|  |
| --- |
| public DataTable LayDanhSach()  {  string sql = "Select \* from KhachHang";  return db.LayDanhSach(sql);  } |
| public void Them(KhachHang kh)  {  string query = String.Format($"EXEC dbo.sp\_ThemKhachHang @sdt = N'{kh.Sdt}'," +  $"@tenkh = N'{kh.TenKH}'," +  $"@diemtichluy = N'{kh.DiemTichLuy}'");    db.Execute(query);  } |
| public void Sua(KhachHang kh)  {  string query = String.Format($"EXEC dbo.sp\_SuaKhachHang @sdt = N'{kh.Sdt}'," +  $"@tenkh = N'{kh.TenKH}'," +  $"@diemtichluy = N'{kh.DiemTichLuy}'");  db.Execute(query);  } |
| public void Xoa(KhachHang kh)  {  string query = String.Format($"EXEC dbo.sp\_XoaKhachHang @sdt = N'{kh.Sdt}'");  db.Execute(query);  } |
| public DataTable TimKiem(string find)  {  string query = string.Format($"SELECT \* FROM dbo.f\_TimKiemKhachHang(N'{find}')");  return db.LayDanhSach(query);  } |

**5.5.8 Kho**

|  |
| --- |
| public DataTable LayDanhSach()  {  string sql = "Select \* from ViewKho";  return db.LayDanhSach(sql);  } |
| public DataTable LayDanhSach\_ConHan()  {  string sql = "select \* from ViewKho WHERE HSD >= GetDATE()";  return db.LayDanhSach(sql);  } |
| public DataTable LayDanhSach\_HetHan()  {  string sql = "select \* from ViewKho WHERE HSD < GetDATE()";  return db.LayDanhSach(sql);  } |
| public void NhapKho(Kho k)  {  string query = String.Format(  $"EXEC dbo.sp\_NhapKho @masp = N'{k.MaSP}'," +  $"@nsx = N'{k.NSX.ToString("yyyy-MM-dd")}'," +  $"@hsd = N'{k.HSD.ToString("yyyy-MM-dd")}'," +  $"@ngaynhapkho = N'{k.NgayNhapKho.ToString("yyyy-MM-dd")}'," +  $"@loaikho = N'{k.LoaiKho}'," +  $"@sltonkho = N'{k.SLTonKho}'"  );  db.Execute(query);  } |
| public void Sua(Kho k)  {  string query = String.Format($"EXEC dbo.sp\_SuaSPvaoKho @masp = N'{k.MaSP}'," +  $"@nsx = N'{k.NSX.ToString("yyyy-MM-dd")}'," +  $"@hsd = N'{k.HSD.ToString("yyyy-MM-dd")}'," +  $"@ngaynhapkho = N'{k.NgayNhapKho.ToString("yyyy-MM-dd")}'," +  $"@loaikho = N'{k.LoaiKho}'," +  $"@sltonkho = N'{k.SLTonKho}'");  db.Execute(query);  } |
| public void Xoa(Kho k)  {  string query = String.Format($"EXEC dbo.sp\_XoaSPvaoKho @masp = N'{k.MaSP}'," +  $"@nsx = N'{k.NSX.ToString("yyyy-MM-dd")}'," +  $"@hsd = N'{k.HSD}'");  db.Execute(query);  } |
| public DataTable TimKiem(string find)  {  string query = string.Format($"SELECT \* FROM dbo.f\_TimKiemKho(N'{find}')");  return db.LayDanhSach(query);  } |
| public DataTable TimKiem\_ConHan(string find)  {  string query = string.Format($"SELECT \* FROM dbo.f\_TimKiemKho(N'{find}') WHERE HSD >= GetDATE()");  return db.LayDanhSach(query);  } |
| public DataTable TimKiem\_HetHan(string find)  {  string query = string.Format($"SELECT \* FROM dbo.f\_TimKiemKho(N'{find}') WHERE HSD < GetDATE()");  return db.LayDanhSach(query);  } |

**5.5.9 Ngày lễ**

|  |
| --- |
| public DataTable LayDanhSach()  {  return exec.LayDanhSach("Select \* from NgayLe");  } |
| public void Them(NgayLe nl)  {  string query = string.Format($"EXECUTE dbo.sp\_ThemNgayLe @ngay = '{nl.Ngay.ToString("yyyy-MM-dd")}', " +  $"@sukien = N'{nl.SuKien}'");  exec.Execute(query);  } |
| public void Sua(NgayLe nl)  {  string query = string.Format($"EXEC dbo.sp\_SuaNgayLe @manl = {nl.MaNL}, " +  $"@ngay = '{nl.Ngay.ToString("yyyy-MM-dd")}', " +  $"@sukien = N'{nl.SuKien}'");  exec.Execute(query);  } |
| public void Xoa(NgayLe nl)  {  string query = string.Format($"EXECUTE dbo.sp\_XoaNgayLe @manl = {nl.MaNL}");  exec.Execute(query);  } |

**5.5.10 Nhân viên**

|  |
| --- |
| public DataTable LayDanhSach()  {  string sql = "Select \* from NhanVien";  return db.LayDanhSach(sql);  } |
| public DataTable LayDanhSach(int trangthai)  {  string sql = $"Select \* from NhanVien where TrangThai = {trangthai}";  return db.LayDanhSach(sql);  } |
| public void Them(NhanVien nv)  {  string query = String.Format($"EXEC dbo.sp\_ThemNhanVien @tennv = N'{nv.TenNV}'," +  $"@sdt = N'{nv.Sdt}'," +  $"@phai = {nv.Phai}," +  $"@ngaysinh = N'{nv.NgaySinh.ToString("yyyy-MM-dd")}'," +  $"@email = N'{nv.Email}'," +  $"@macv = {nv.MaCV}," +  $"@trangthai = {nv.TrangThai}," +  $"@tentk = N'{nv.TenTK}'," +  $"@matkhau = N'{nv.MatKhau}'" );  db.Execute(query);  NhanVien nNV = LayThongTinNhanVienBangTenTK(nv.TenTK);  nv.MaNV = nNV.MaNV;  if (nv.Hinh != null)  {  LuuAnh(nv, nv.Hinh);  }  } |
| public void Sua(NhanVien nv)  {  string query = String.Format($"EXEC dbo.sp\_SuaNhanVien @manv = N'{nv.MaNV}'," +  $"@tennv = N'{nv.TenNV}'," +  $"@sdt = N'{nv.Sdt}'," +  $"@phai = {nv.Phai}," +  $"@ngaysinh = N'{nv.NgaySinh.ToString("yyyy-MM-dd")}'," +  $"@email = N'{nv.Email}'," +  $"@macv = {nv.MaCV}," +  $"@trangthai = {nv.TrangThai}," +  $"@tentk = N'{nv.TenTK}'," +  $"@matkhau = N'{nv.MatKhau}'");  db.Execute(query);  if (nv.Hinh != null)  {  LuuAnh(nv, nv.Hinh);  }  } |
| public void Xoa(NhanVien nv)  {  string query = String.Format($"EXEC dbo.sp\_XoaNhanVien @manv = N'{nv.MaNV}'");  db.Execute(query);  } |
| public NhanVien LayThongTinNhanVienBangTenTK(string tentk)  {  string query = string.Format("SELECT \* FROM dbo.NhanVien WHERE TenTK = N'{0}'", tentk);  DataTable result = db.LayDanhSach(query);  foreach (DataRow dr in result.Rows)  {  return new NhanVien(dr);  }  return null;  } |
| public NhanVien LayThongTinNhanVienBangMaNV(string manv)  {  string query = string.Format("SELECT \* FROM dbo.NhanVien WHERE MaNV = N'{0}'", manv);  DataTable result = db.LayDanhSach(query);  foreach (DataRow dr in result.Rows)  {  return new NhanVien(dr);  }  return null;  } |
| public DataTable TimKiem(string find)  {  string query = string.Format($"SELECT \* FROM dbo.f\_TimKiemNhanVien(N'{find}')");  return db.LayDanhSach(query);  } |
| public DataTable TimKiem(string find, int trangthai)  {  string query = string.Format($"SELECT \* FROM dbo.f\_TimKiemNhanVien(N'{find}') where TrangThai = {trangthai}");  return db.LayDanhSach(query);  } |

**5.5.11 Nhà sản xuất**

|  |
| --- |
| public DataTable LayDanhSach()  {  return exec.LayDanhSach("Select \* from NhaSanXuat");  } |
| public void Them(NhaSanXuat nsx)  {  string query = string.Format($"EXEC dbo.sp\_ThemNSX @tennsx = N'{nsx.TenNSX}'," +  $"@diachi = N'{nsx.DiaChi}'");  exec.Execute(query);  } |
| public void Sua(NhaSanXuat nsx)  {  string query = string.Format($"EXEC dbo.sp\_SuaNSX @mansx = {nsx.MaNSX}, " +  $"@tennsx = N'{nsx.TenNSX}', " +  $"@diachi = N'{nsx.DiaChi}'");  exec.Execute(query);  } |
| public void Xoa(NhaSanXuat nsx)  {  string query = string.Format($"EXECUTE dbo.sp\_XoaNSX @mansx = {nsx.MaNSX}");  exec.Execute(query);  } |

**5.5.12 Phân công**

|  |
| --- |
| public DataTable LayDanhSach()  {  return exec.LayDanhSach("Select \* from XemBangPC\_NhanVien");  } |
| public DataTable LayDanhSach(string manv)  {  return exec.LayDanhSach($"Select \* from XemBangPC\_NhanVien WHERE MaNV = N'{manv}'");  } |
| public PhanCong LayThongTinPhanCongBangMaPC(int macv)  {  string query = string.Format($"SELECT \* FROM dbo.PhanCong WHERE MaPC = {macv}");  DataTable result = exec.LayDanhSach(query);  foreach (DataRow dr in result.Rows)  {  return new PhanCong(dr);  }  return null;  } |
| public void Them(PhanCong pc)  {  string query = string.Format($"EXEC dbo.sp\_ThemPhanCong " +  $"@manv = N'{pc.MaNV}'," +  $"@maca = N'{pc.MaCa}'," +  $"@ngaydangky = '{pc.NgayDangKy.ToString("yyyy-MM-dd")}'");  exec.Execute(query);  } |
| public void Xoa(PhanCong pc)  {  string query = string.Format($"EXEC dbo.sp\_XoaPhanCong @mapc = {pc.MaPC}");  exec.Execute(query);  } |
| public DataTable TimKiem(string find)  {  string query = string.Format($"SELECT \* FROM dbo.f\_TimKiemPhanCong(N'{find}')");  return exec.LayDanhSach(query);  } |

**5.5.13 Phương thức thanh toán**

|  |
| --- |
| public DataTable LayDanhSach()  {  string sql = "Select \* from PhuongThucThanhToan";  return db.LayDanhSach(sql);  } |

**5.5.14 Sản phẩm**

|  |
| --- |
| public DataTable LayDanhSach()  {  string sql = "Select \* from SanPham";  return db.LayDanhSach(sql);  } |
| public DataTable LayDanhSach(int trangthai)  {  string sql = $"Select \* from SanPham where TrangThai = {trangthai}";  return db.LayDanhSach(sql);  } |
| public void Them(SanPham sp)  {  string query = String.Format($"EXEC dbo.sp\_ThemSanPham @tensp = N'{sp.TenSP}'," +  $"@mansx= {sp.MaNSX}," +  $"@maloai = {sp.MaLoai}," +  $"@giaban = {sp.GiaBan}," +  $"@giagoc = {sp.GiaGoc}," +  $"@trangthai = {sp.TrangThai}");  db.Execute(query);  SanPham nSP = LayThongTinSP(sp);  sp.MaSP = nSP.MaSP;  if (sp.Hinh != null)  {  LuuAnh(sp, sp.Hinh);  }  } |
| public void Sua(SanPham sp)  {    string query = String.Format($"EXEC dbo.sp\_SuaSanPham @masp = {sp.MaSP}," +  $"@tensp = N'{sp.TenSP}'," +  $"@mansx= {sp.MaNSX}," +  $"@maloai = {sp.MaLoai}," +  $"@giaban = {sp.GiaBan}," +  $"@giagoc = {sp.GiaGoc}," +  $"@trangthai = {sp.TrangThai}");  db.Execute(query);  if (sp.Hinh != null)  {  LuuAnh(sp, sp.Hinh);  }  } |
| public void Xoa(SanPham sp)  {  string query = String.Format($"EXEC dbo.sp\_XoaSanPham @masp = {sp.MaSP}");  db.Execute(query);  } |
| public SanPham LayThongTinSP(SanPham sp)  {  string query = string.Format($"SELECT \* FROM dbo.SanPham WHERE TenSP = N'{sp.TenSP}' and " +  $"MaNSX= {sp.MaNSX} and " +  $"MaLoai = {sp.MaLoai} and " +  $"GiaBan = {sp.GiaBan} and " +  $"GiaGoc = {sp.GiaGoc} and " +  $"TrangThai = '{sp.TrangThai}'");  DataTable result = db.LayDanhSach(query);  foreach (DataRow dr in result.Rows)  {  return new SanPham(dr);  }  return null;  } |
| public SanPham LayThongTinSanPhamBangMaSP(int masp)  {  string query = string.Format($"SELECT \* FROM dbo.SanPham WHERE MaSP = {masp}");  DataTable result = db.LayDanhSach(query);  foreach (DataRow dr in result.Rows)  {  return new SanPham(dr);  }  return null;  } |
| public DataTable TimKiem(string find)  {  string query = string.Format($"SELECT \* FROM dbo.f\_TimKiemSanPham(N'{find}')");  return db.LayDanhSach(query);  } |
| public DataTable TimKiem(string find, int trangthai)  {  string query = string.Format($"SELECT \* FROM dbo.f\_TimKiemSanPham(N'{find}') where TrangThai = {trangthai}");  return db.LayDanhSach(query);  } |

**5.5.15 Tính lương**

|  |
| --- |
| public DataTable LayDanhSach()  {  return exec.LayDanhSach($"Select \* from TinhLuong");  } |
| public DataTable LayDanhSach(string manv)  {  return exec.LayDanhSach($"Select \* from TinhLuong WHERE MaNV = '{manv}'");  } |

**5.6 Các form danh mục**

**5.6.1 Đăng nhập với tư cách là nhân viên**

**5.6.1.1 Form thông tin cá nhân**

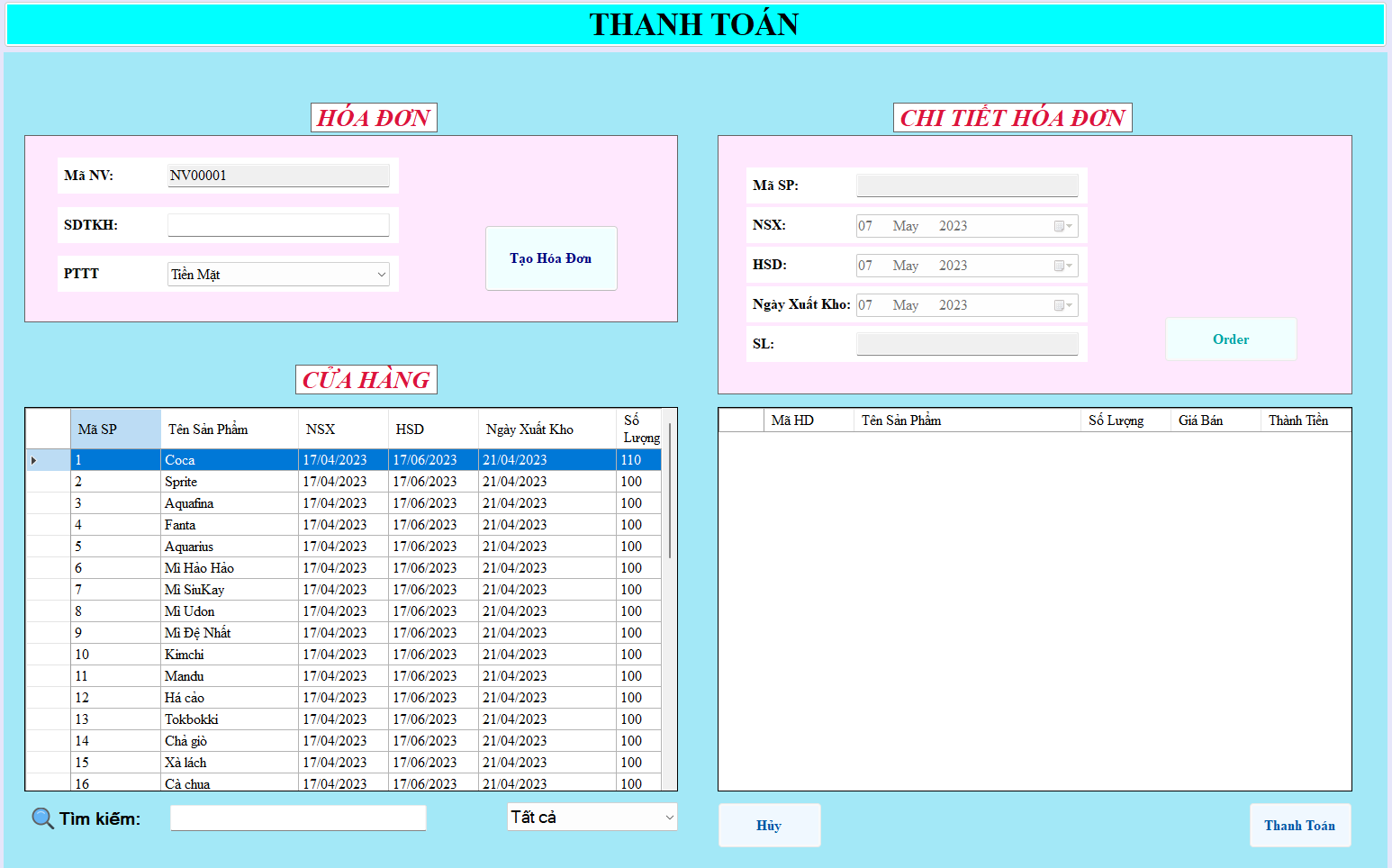


Người dùng có thể tải ảnh cá nhân và lưu ảnh

|  |
| --- |
| public void LuuAnh(NhanVien nv, byte[] anh)  {  nv.Hinh = anh;  using (SqlConnection conn = new SqlConnection(Properties.Settings.Default.cnnStr))  {  conn.Open();  using (SqlCommand command = new SqlCommand("UPDATE dbo.NhanVien SET Hinh = @anh WHERE MaNV = @id ", conn))  {  command.Parameters.AddWithValue("@id", nv.MaNV);  command.Parameters.AddWithValue("@anh", anh);  if (command.ExecuteNonQuery() > 0)  MessageBox.Show("Thao tác thành công", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  else MessageBox.Show("Thao tác thất bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  conn.Close();  }  } |

**5.6.1.2 Form thanh toán**

Trước khi thực hiện thanh toán, người dùng phải bấm vào nút tạo hóa đơn với SĐT khách hàng được ghi nhận. Người dùng chọn vào một bộ trong sản phẩm của cửa hàng để chọn món và thanh toán, ngược lại có thể hủy hóa đơn nếu thông tin hóa đơn có sai sót bằng nút hủy.



Tạo hóa đơn

|  |
| --- |
| private void btnTaoHD\_Click(object sender, EventArgs e)  {  string sdtkh = tbSDTKH.Text.ToString();  if (sdtkh == "")  sdtkh = "NULL";  else  sdtkh = "'" + sdtkh + "'";  HoaDon hd = new HoaDon(tbMaNV.Text, sdtkh, Convert.ToInt32(cbPTTT.SelectedValue.ToString()));  hdDAO.TaoHD(hd);  btnTaoHD.Enabled = false;  btnOrder.Enabled = true;  btnHuy.Enabled = true;  btnThanhToan.Enabled = true;  } |

Thao tác Order

|  |
| --- |
| private void btnOrder\_Click(object sender, EventArgs e)  {  int mahd = hdDAO.MaxHD();  ChiTietHoaDon cthd = new ChiTietHoaDon(mahd, Convert.ToInt32(tbMaSPCuaHang.Text), (DateTime)this.dtpkNSXCuaHang.Value,  (DateTime)this.dtpkHSDCuaHang.Value, Convert.ToInt32(this.tbSoLuong.Text));  cthdDAO.Order(cthd);  dgvChiTietHoaDon.DataSource = cthdDAO.LayDanhSachThanhToan();  } |

Thanh toán

|  |
| --- |
| private void btnThanhToan\_Click(object sender, EventArgs e)  {  int mahd = hdDAO.MaxHD();  hdDAO.ThanhToan();  LoadDanhSach();  dgvChiTietHoaDon.DataSource = cthdDAO.LayDanhSachThanhToanRong();  LoadData();  } |

Hủy hóa đơn thanh toán

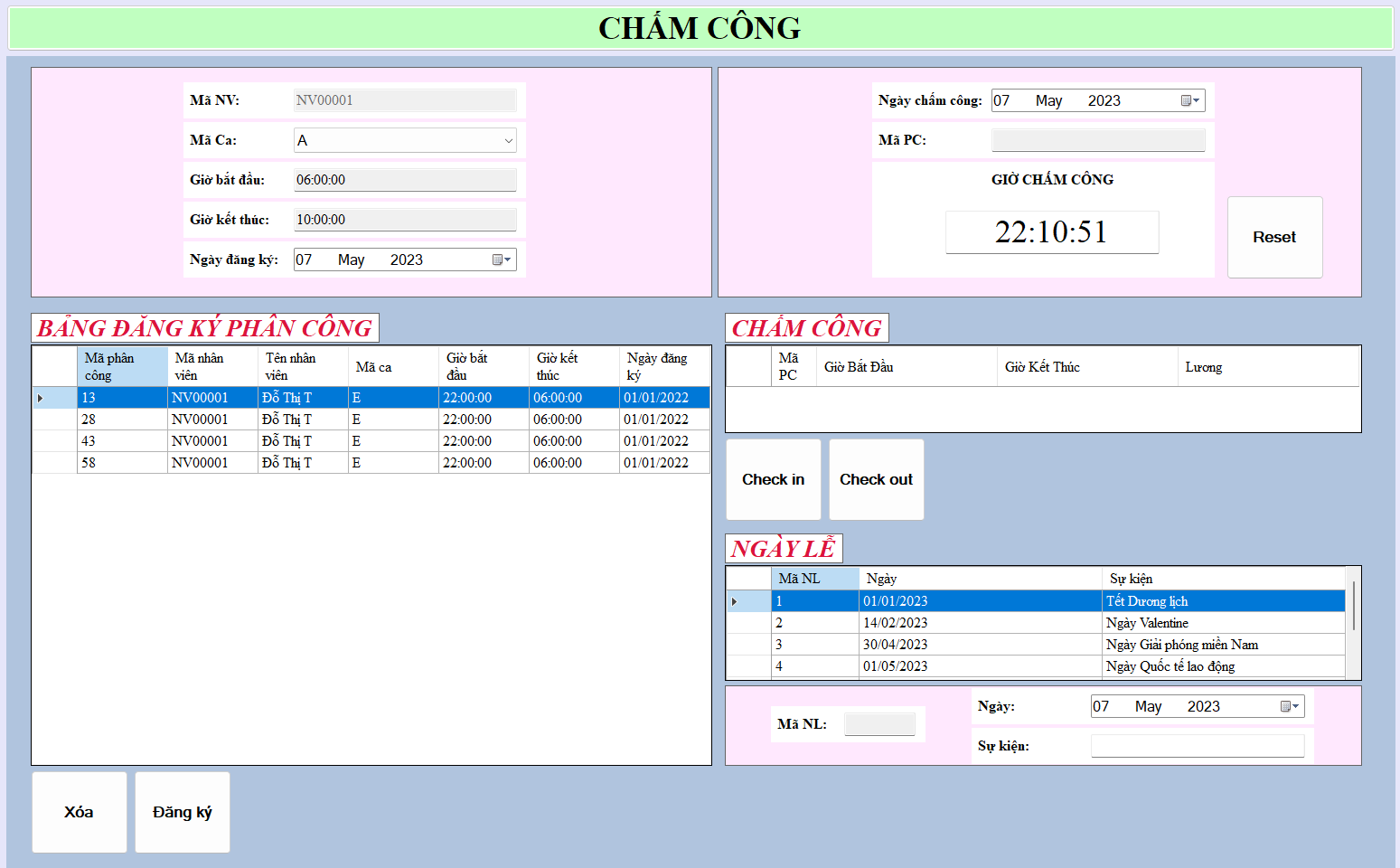
|  |
| --- |
| private void btnHuy\_Click(object sender, EventArgs e)  {  int mahd = hdDAO.MaxHD();  hdDAO.HuyThanhToan(mahd);  dgvChiTietHoaDon.DataSource = null;  LoadData();  } |

Tìm kiếm

|  |
| --- |
| void TimKiem()  {  if (cbLoaiCuaHang.Text == "Tất cả")  {  dgvCuaHang.DataSource = chDAO.TimKiem(tbTimKiem.Text);  }  else if (cbLoaiCuaHang.Text == "Còn hạn")  {  dgvCuaHang.DataSource = chDAO.TimKiem\_ConHan(tbTimKiem.Text);  }  else  dgvCuaHang.DataSource = chDAO.TimKiem\_HetHan(tbTimKiem.Text);  } |

**5.6.1.3 Form chấm công (NhanVien)**

Nhân viên chỉ được xem bảng đăng ký phân công của riêng mình. Chỉ được xem bảng NgayLe nhưng không chỉnh sửa được. Người dùng chọn vào một bộ trên bảng phân công sẽ hiện thông tin bên bảng chấm công và có thể thực hiện check in và check out theo thời gian chỉ định.



Đăng ký phân công

|  |
| --- |
| private void btDangKy\_Click(object sender, EventArgs e)  {  PhanCong pc = new PhanCong(0, tbMaNV.Text, cbMaCa.Text, dtpkNgayDK.Value);  pcDAO.Them(pc);  LoadDanhSach();  } |

Xóa đăng ký

|  |
| --- |
| private void btXoa\_Click(object sender, EventArgs e)  {  PhanCong pc = new PhanCong(Convert.ToInt32(tbMaPC.Text), tbMaNV.Text, cbMaCa.Text, DateTime.Today);  pcDAO.Xoa(pc);  LoadDanhSach();  } |

Check in

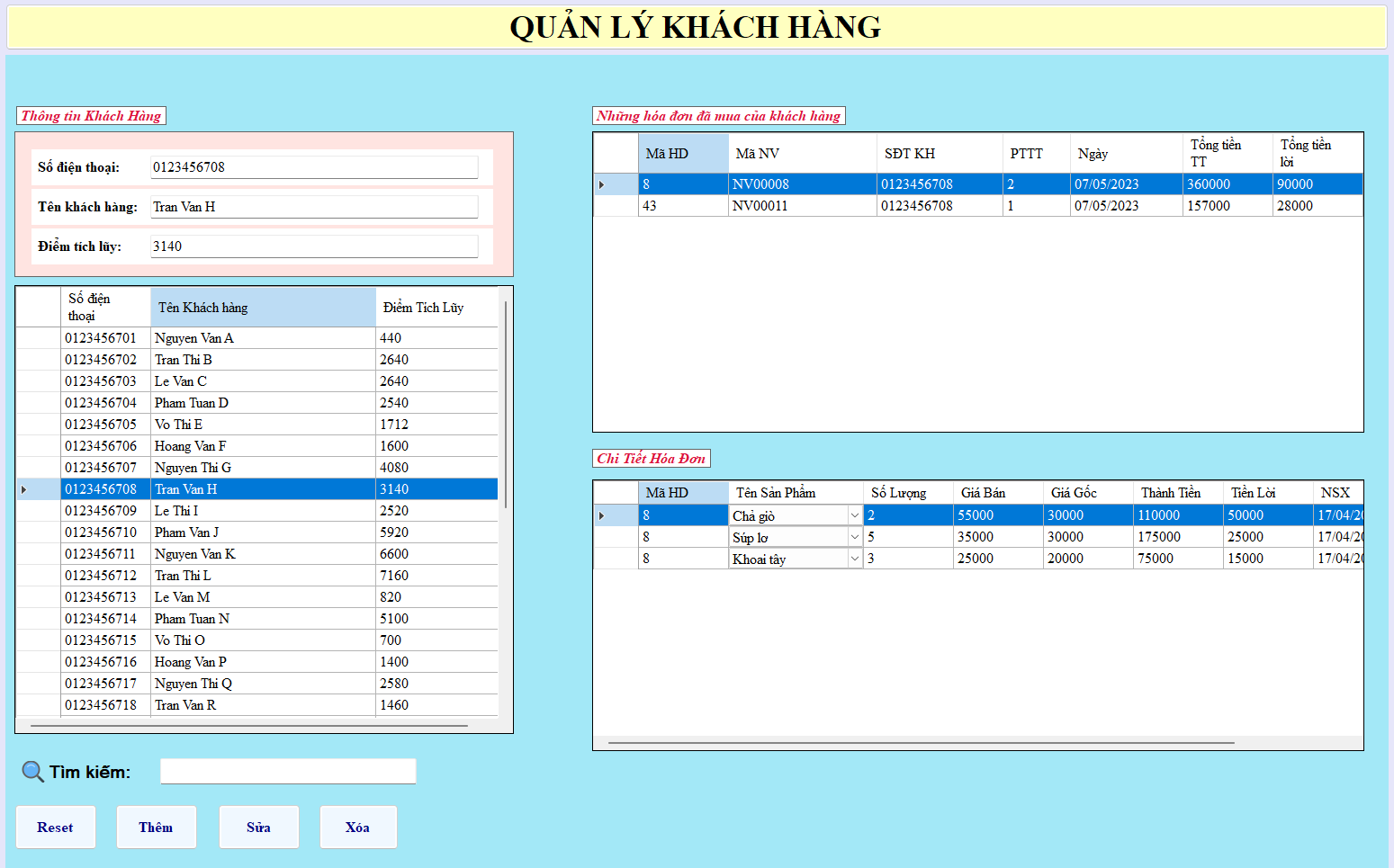
|  |
| --- |
| private void btCheckIn\_Click(object sender, EventArgs e)  {  try  {  ChamCong cc = new ChamCong(Convert.ToInt32(tbMaPC.Text), TimeSpan.Parse(tbGioChamCong.Text), null, null);  ccDAO.CheckIn(cc);  dtgvChamCong.DataSource = ccDAO.LayDanhSach(cc.MaPC);  }  catch (Exception ex)  {  MessageBox.Show("Check out thất bại!\n" + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

Check out

|  |
| --- |
| private void btCheckOut\_Click(object sender, EventArgs e)  {  try  {  ChamCong cc = new ChamCong(Convert.ToInt32(tbMaPC.Text), null, TimeSpan.Parse(tbGioChamCong.Text), null);  dtgvChamCong.CurrentRow.Selected = true;  if (dtgvChamCong.SelectedRows[0].Cells[1].Value.ToString() != "" && dtgvChamCong.SelectedRows[0].Cells[2].Value.ToString() == "")  {  ccDAO.CheckOut(cc);  dtgvChamCong.DataSource = ccDAO.LayDanhSach(cc.MaPC);  }  else  throw new Exception();  }  catch (Exception ex)  {  MessageBox.Show("Check out thất bại!\n" + ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

**5.6.1.4 Form quản lý khách hàng**

Người dùng có thể thực hiện các thao tác như thêm, xóa, sửa thông tin của khách hàng. Khi chọn một hàng trên bảng thông tin khách hàng, sẽ có thể xem được những hóa đơn và chi tiết hóa đơn mà khách hàng ấy đã mua.



Thêm khách hàng

|  |
| --- |
| private void btnThem\_Click(object sender, EventArgs e)  {  KhachHang kh = new KhachHang(tbSDT.Text, tbTenKH.Text, Convert.ToDouble(tbDTL.Text.ToString()));  khDAO.Them(kh);  LoadData();  } |

Sửa khách hàng

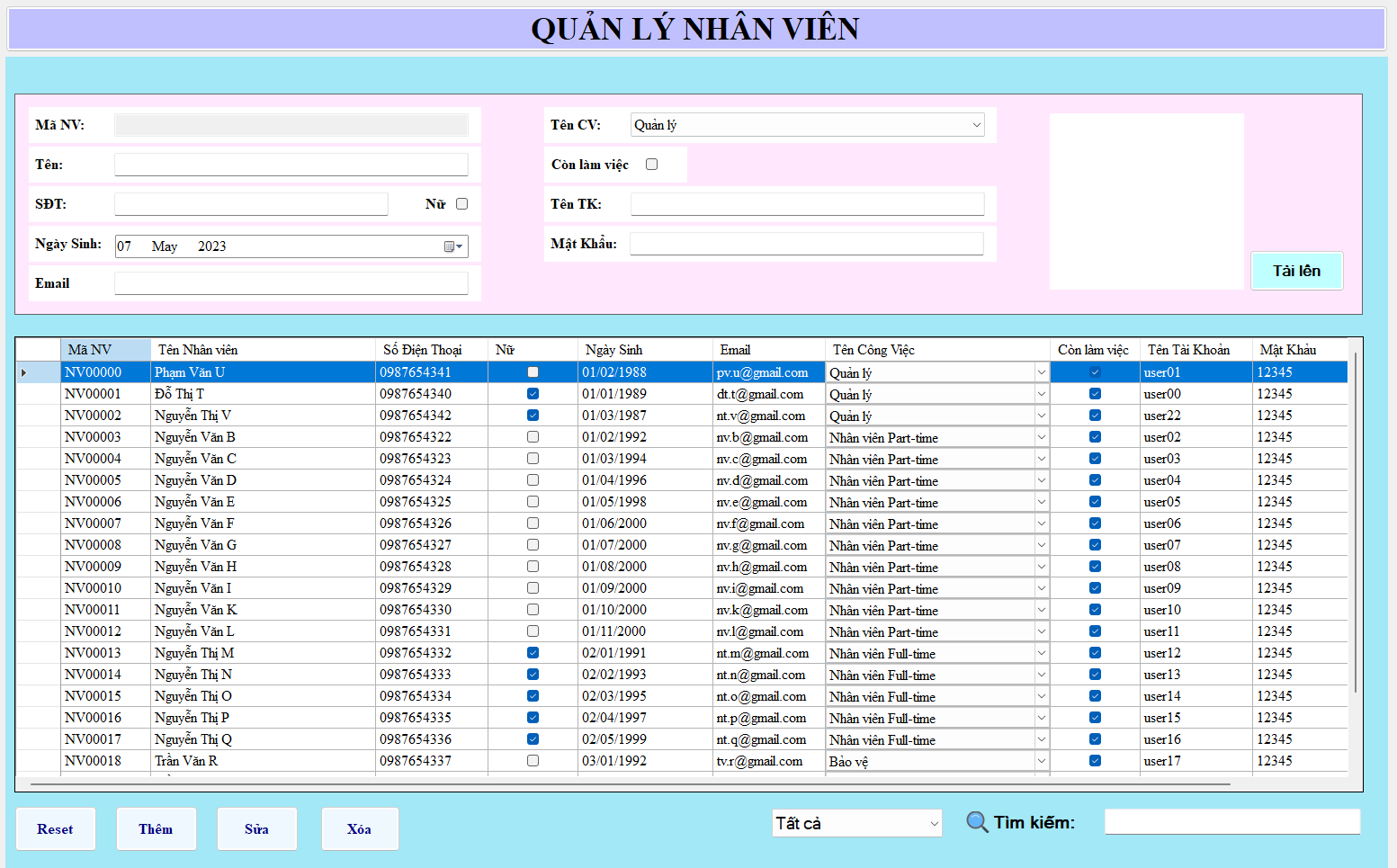
|  |
| --- |
| private void btnSua\_Click(object sender, EventArgs e)  {  KhachHang kh = new KhachHang(tbSDT.Text, tbTenKH.Text, Convert.ToInt32(tbDTL.Text.ToString()));  khDAO.Sua(kh);  LoadData();  } |

Xóa khách hàng

|  |
| --- |
| private void btnXoa\_Click(object sender, EventArgs e)  {  KhachHang kh = new KhachHang(tbSDT.Text);  khDAO.Xoa(kh);  LoadData();  } |

**5.6.2 Đăng nhập với tư cách là quản lý**

**5.6.2.1 Quản lý nhân viên**

****

Thêm nhân viên

|  |
| --- |
| private void btnThem\_Click(object sender, EventArgs e)  {  NhanVien nv = new NhanVien(tbTenNV.Text, tbSDT.Text, chbPhai.Checked, dtpkNgaySinh.Value, tbEmail.Text, cbMaCV.SelectedIndex+1, chbTrangThai.Checked, tbTenTK.Text, tbMatKhau.Text, (ptHinh.Image != null) ? nvDAO.ChuyenAnhThanhMangByte(ptHinh) : null);  nvDAO.Them(nv);  LoadDanhSach();  } |

Sửa nhân viên

|  |
| --- |
| private void btnSua\_Click(object sender, EventArgs e)  {  NhanVien nv = new NhanVien(tbMaNV.Text, tbTenNV.Text, tbSDT.Text, chbPhai.Checked, dtpkNgaySinh.Value, tbEmail.Text, cbMaCV.SelectedIndex + 1, chbTrangThai.Checked, tbTenTK.Text, tbMatKhau.Text, (ptHinh.Image != null) ? nvDAO.ChuyenAnhThanhMangByte(ptHinh) : null);  nvDAO.Sua(nv);  LoadDanhSach();  } |

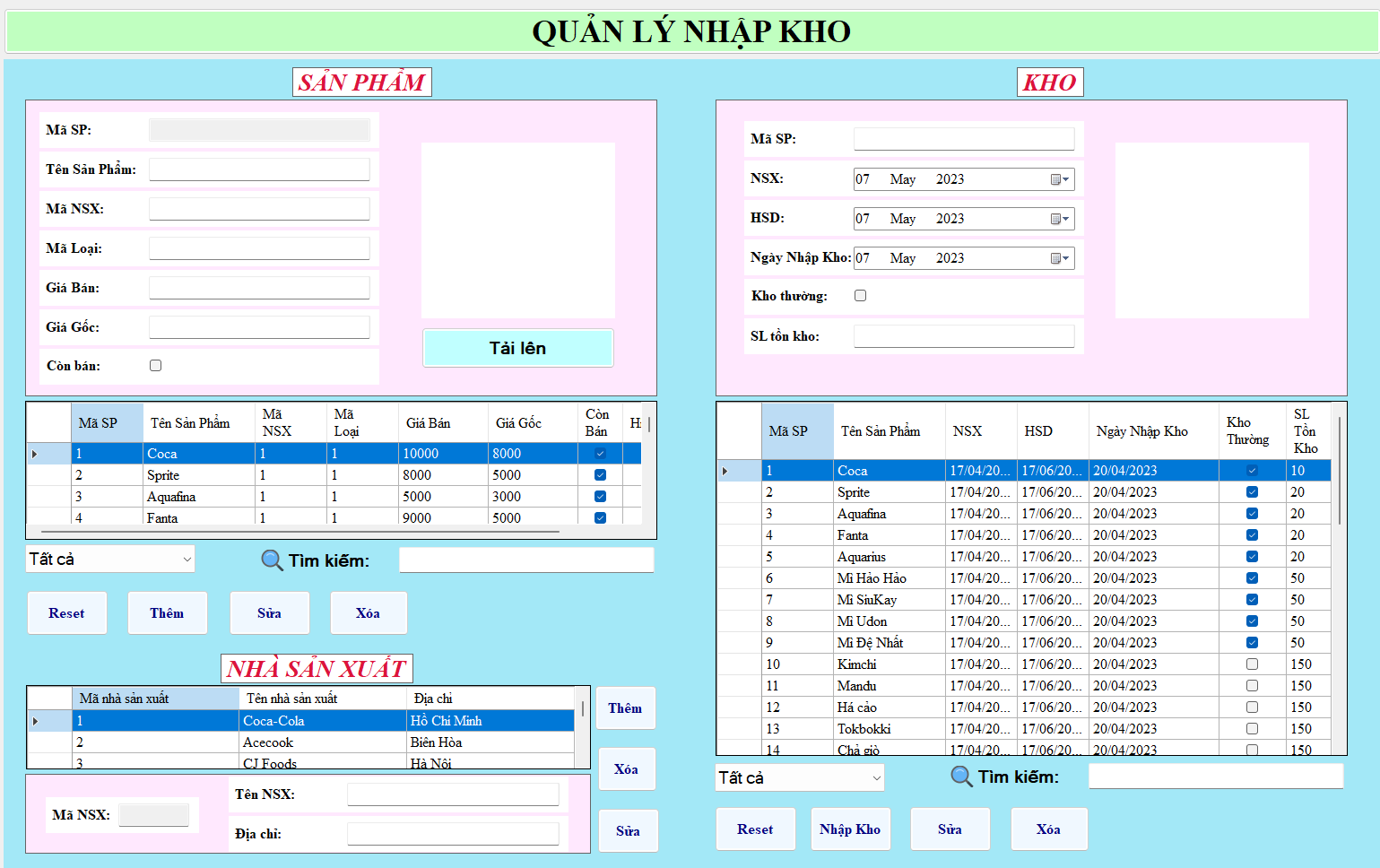
Xóa nhân viên

|  |
| --- |
| private void btnXoa\_Click(object sender, EventArgs e)  {  NhanVien nv = new NhanVien(tbMaNV.Text);  nvDAO.Xoa(nv);  LoadDanhSach();  } |

Tìm kiếm nhân viên

|  |
| --- |
| void TimKiem()  {  if (cbLoai.Text == "Tất cả")  {  dgvNhanVien.DataSource = nvDAO.TimKiem(tbTimKiem.Text);  }  else if (cbLoai.Text == "Còn làm việc")  {  dgvNhanVien.DataSource = nvDAO.TimKiem(tbTimKiem.Text, 1);  }  else  dgvNhanVien.DataSource = nvDAO.TimKiem(tbTimKiem.Text, 0);  DataTable dtcv = new DataTable();  dtcv = cvDAO.LayDanhSach();  (dgvNhanVien.Columns["MaCV"] as DataGridViewComboBoxColumn).DataSource = dtcv;  (dgvNhanVien.Columns["MaCV"] as DataGridViewComboBoxColumn).DisplayMember = "TenCV";  (dgvNhanVien.Columns["MaCV"] as DataGridViewComboBoxColumn).ValueMember = "MaCV";  } |

**5.6.2.2 Quản lý nhập kho**



1. **Sản phẩm**

Thêm

|  |
| --- |
| private void btnThem\_Click(object sender, EventArgs e)  {  SanPham sp = new SanPham(tbTenSP.Text, Convert.ToInt32(tbMaNSX.Text), Convert.ToInt32(tbMaLoai.Text),  Convert.ToInt32(tbGiaBan.Text), Convert.ToInt32(tbGiaGoc.Text), chbTrangThaiSP.Checked,  (ptHinhSP.Image != null) ? nvDAO.ChuyenAnhThanhMangByte(ptHinhSP) : null);  spDAO.Them(sp);  LoadDataSP();  } |

Xóa

|  |
| --- |
| private void btnXoa\_Click(object sender, EventArgs e)  {  SanPham sp = new SanPham(Convert.ToInt32(tbMaSP.Text));  spDAO.Xoa(sp);  LoadDataSP();  } |

Sửa

|  |
| --- |
| private void btnSua\_Click(object sender, EventArgs e)  {  SanPham sp = new SanPham(Convert.ToInt32(tbMaSP.Text), tbTenSP.Text, Convert.ToInt32(tbMaNSX.Text), Convert.ToInt32(tbMaLoai.Text),  Convert.ToInt32(tbGiaBan.Text), Convert.ToInt32(tbGiaGoc.Text), chbTrangThaiSP.Checked,  (ptHinhSP.Image != null) ? nvDAO.ChuyenAnhThanhMangByte(ptHinhSP) : null);  spDAO.Sua(sp);  LoadDataSP();  } |

Tìm kiếm

|  |
| --- |
| void TimKiemSP()  {  if (cbLoai.Text == "Tất cả")  {  dgvSanPham.DataSource = spDAO.TimKiem(tbTimKiem.Text);  }  else if (cbLoai.Text == "Còn bán")  {  dgvSanPham.DataSource = spDAO.TimKiem(tbTimKiem.Text, 1);  }  else  dgvSanPham.DataSource = spDAO.TimKiem(tbTimKiem.Text, 0);  } |

1. **Nhà sản xuất**

Thêm

|  |
| --- |
| private void btThemNSX\_Click(object sender, EventArgs e)  {  NhaSanXuat nsx = new NhaSanXuat(0, tbTenNSX.Text, tbDiaChi.Text);  nsxDAO.Them(nsx);  tbTenNSX.Text = "";  tbMaNSX1.Text = "";  tbDiaChi.Text = "";  dtgvNSX.DataSource = nsxDAO.LayDanhSach();  } |

Xóa

|  |
| --- |
| private void btXoaNSX\_Click(object sender, EventArgs e)  {  NhaSanXuat nsx = new NhaSanXuat(Convert.ToInt32(tbMaNSX1.Text), tbTenNSX.Text, tbDiaChi.Text);  nsxDAO.Xoa(nsx);  tbTenNSX.Text = "";  tbMaNSX1.Text = "";  tbDiaChi.Text = "";  dtgvNSX.DataSource = nsxDAO.LayDanhSach();  } |

Sửa

|  |
| --- |
| private void btSuaNSX\_Click(object sender, EventArgs e)  {  NhaSanXuat nsx = new NhaSanXuat(Convert.ToInt32(tbMaNSX1.Text), tbTenNSX.Text, tbDiaChi.Text);  nsxDAO.Sua(nsx);  tbTenNSX.Text = "";  tbMaNSX1.Text = "";  tbDiaChi.Text = "";  dtgvNSX.DataSource = nsxDAO.LayDanhSach();  } |

1. **Kho**

Nhập kho

|  |
| --- |
| private void btnNhapKho\_Click(object sender, EventArgs e)  {  Kho k = new Kho(Convert.ToInt32(tbMaSPKho.Text), dtpkNSXKho.Value, dtpkHSDKho.Value, dtpkNgayNhapKho.Value, chbLoaiKho.Checked, Convert.ToInt32(tbSLTonKho.Text));  khoDAO.NhapKho(k);  LoadData();  } |

Sửa

|  |
| --- |
| private void btnSuaKho\_Click(object sender, EventArgs e)  {  Kho k = new Kho(Convert.ToInt32(tbMaSPKho.Text), dtpkNSXKho.Value, dtpkHSDKho.Value, dtpkNgayNhapKho.Value, chbLoaiKho.Checked, Convert.ToInt32(tbSLTonKho.Text));  khoDAO.Sua(k);  LoadData();  } |

Xóa

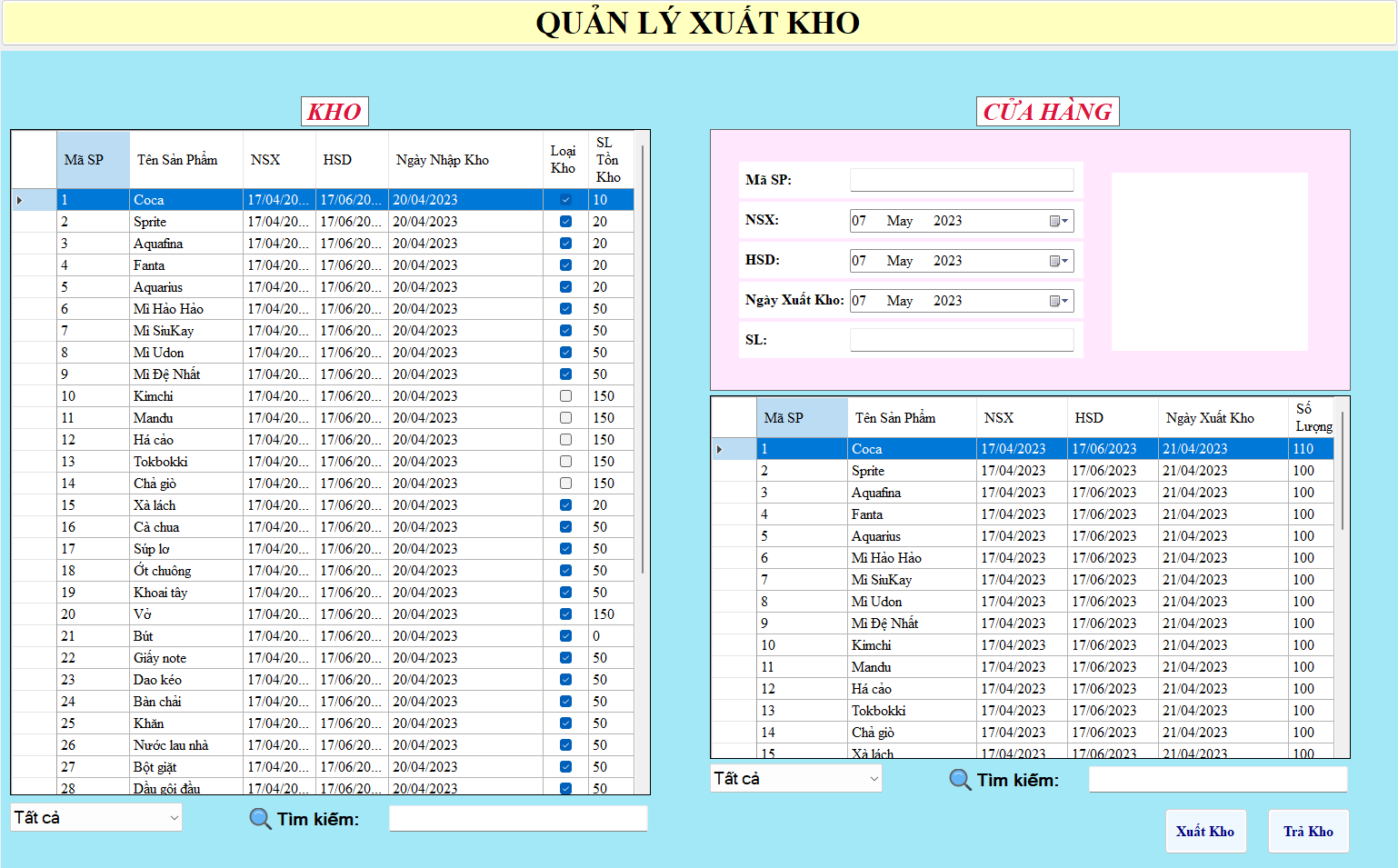
|  |
| --- |
| private void btnXoaKho\_Click(object sender, EventArgs e)  {  Kho k = new Kho(Convert.ToInt32(tbMaSPKho.Text), dtpkNSXKho.Value, dtpkHSDKho.Value);  khoDAO.Xoa(k);  LoadData();  } |

Tìm kiếm

|  |
| --- |
| void TimKiemKho()  {  if (cbLoaiKho.Text == "Tất cả")  {  dgvKho.DataSource = khoDAO.TimKiem(btTimKiemKho.Text);  }  else if (cbLoaiKho.Text == "Còn hạn")  {  dgvKho.DataSource = khoDAO.TimKiem\_ConHan(btTimKiemKho.Text);  }  else  dgvKho.DataSource = khoDAO.TimKiem\_HetHan(btTimKiemKho.Text);  } |

**5.6.2.3 Quản lý xuất kho**

Các phương thức, thủ tục của bảng kho thì giống với form quản lý nhập kho.



Xuất kho

|  |
| --- |
| private void btnXuatKho\_Click(object sender, EventArgs e)  {  CuaHang ch = new CuaHang(Convert.ToInt32(tbMaSPCuaHang.Text), this.dtpkNSXCuaHang.Value,  this.dtpkHSDCuaHang.Value, DateTime.Today, Convert.ToInt32(this.tbSoLuong.Text));  chDAO.XuatKho(ch);  LoadData();  } |

Trả kho

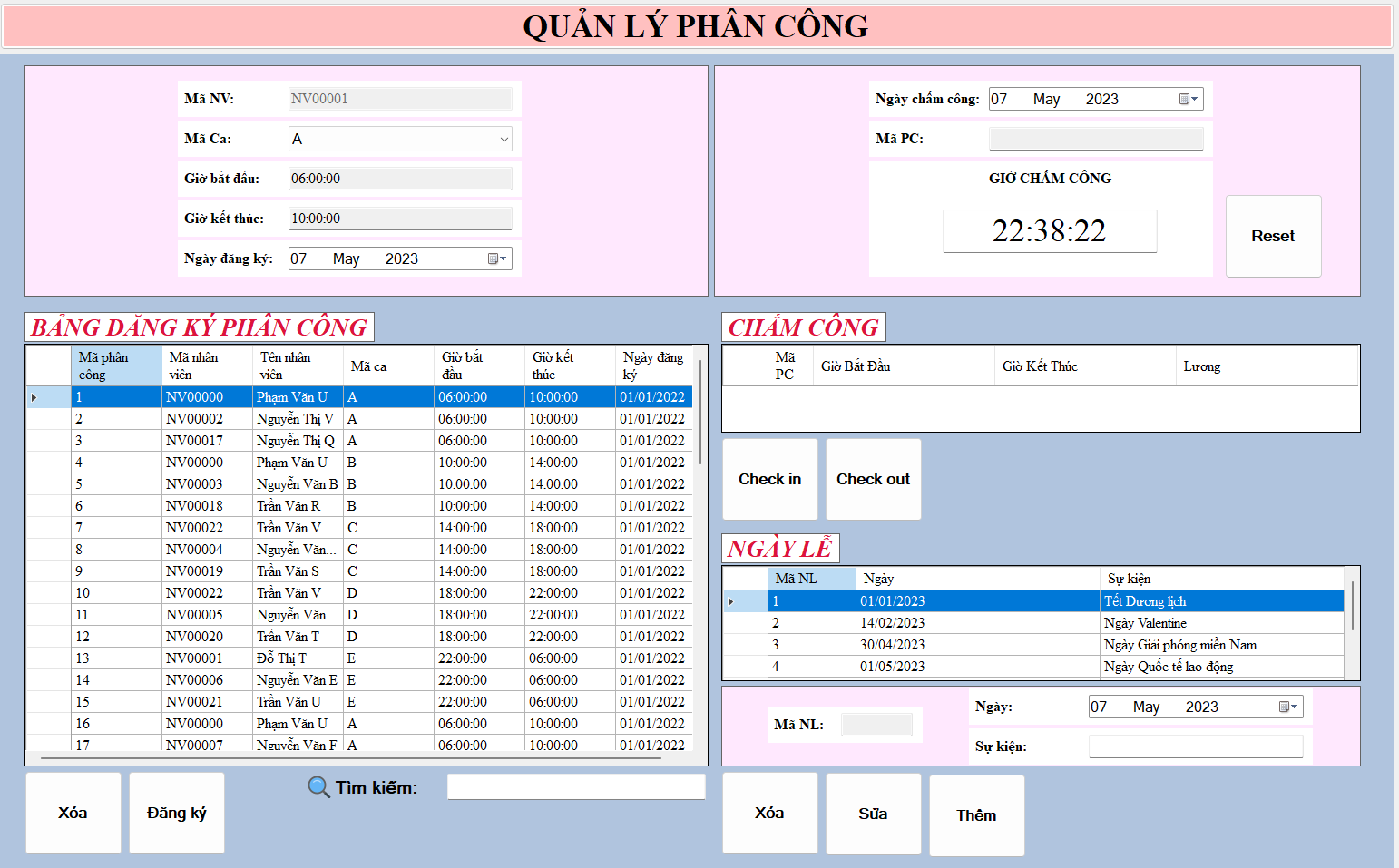
|  |
| --- |
| private void btnTraKho\_Click(object sender, EventArgs e)  {  CuaHang ch = new CuaHang(Convert.ToInt32(tbMaSPCuaHang.Text), this.dtpkNSXCuaHang.Value,  this.dtpkHSDCuaHang.Value, this.dtpkNgayXuatKho.Value, Convert.ToInt32(this.tbSoLuong.Text));  chDAO.TraKho(ch);  LoadData();  } |

Tìm kiếm cửa hàng

|  |
| --- |
| void TimKiemCuaHang()  {  if (cbLoaiCuaHang.Text == "Tất cả")  {  dgvCuaHang.DataSource = chDAO.TimKiem(tbTimKiemCuaHang.Text);  }  else if (cbLoaiCuaHang.Text == "Còn hạn")  {  dgvCuaHang.DataSource = chDAO.TimKiem\_ConHan(tbTimKiemCuaHang.Text);  }  else  dgvCuaHang.DataSource = chDAO.TimKiem\_HetHan(tbTimKiemCuaHang.Text);  } |

**5.6.2.4 Quản lý phân công**

Người quản lý thực hiện được các thao tác đăng ký ca và chấm công cho mọi nhân viên và có thể chỉnh sửa được bảng ngày lễ.



Thêm ngày lễ

|  |
| --- |
| private void btThemNL\_Click(object sender, EventArgs e)  {  NgayLe nl = new NgayLe(0, dtpkNL.Value, tbSuKien.Text);  ngayLeDAO.Them(nl);  tbSuKien.Text = "";  tbMaNL.Text = "";  dtpkNL.Value = DateTime.Today;  dtgvNgayLe.DataSource = ngayLeDAO.LayDanhSach();  } |

Sửa ngày lễ

|  |
| --- |
| private void btSuaNL\_Click(object sender, EventArgs e)  {  NgayLe nl = new NgayLe(Convert.ToInt32(tbMaNL.Text), dtpkNL.Value, tbSuKien.Text);  ngayLeDAO.Sua(nl);  tbSuKien.Text = "";  tbMaNL.Text = "";  dtpkNL.Value = DateTime.Today;  dtgvNgayLe.DataSource = ngayLeDAO.LayDanhSach();  } |

Xóa ngày lễ

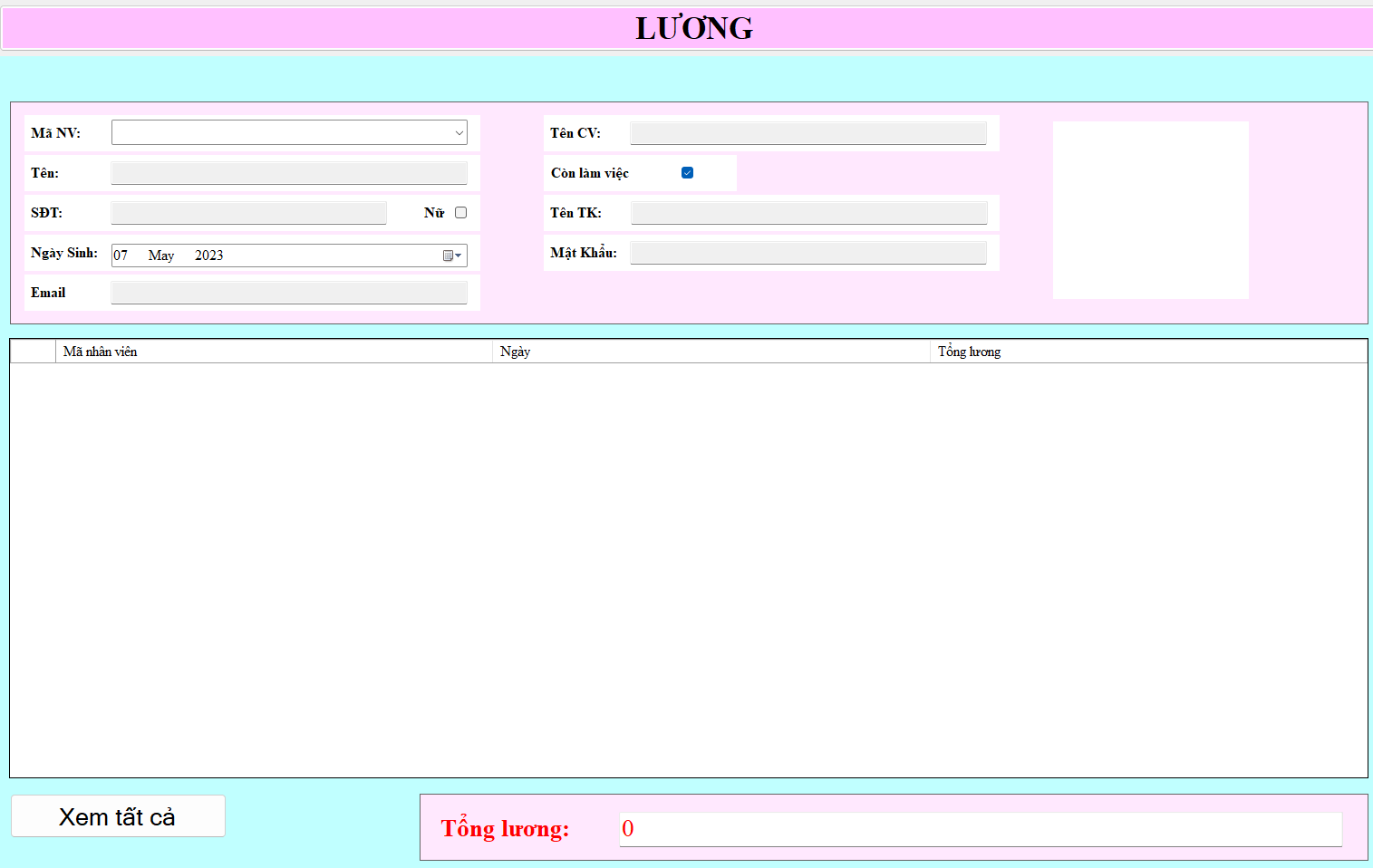
|  |
| --- |
| private void btXoaNL\_Click(object sender, EventArgs e)  {  NgayLe nl = new NgayLe(Convert.ToInt32(tbMaNL.Text), dtpkNL.Value, tbSuKien.Text);  ngayLeDAO.Xoa(nl);  tbSuKien.Text = "";  tbMaNL.Text = "";  dtpkNL.Value = DateTime.Today;  dtgvNgayLe.DataSource = ngayLeDAO.LayDanhSach();  } |

Tìm kiếm trong bảng phân công

|  |
| --- |
| private void tbTimKiem\_TextChanged(object sender, EventArgs e)  {  dtgvPhanCong.DataSource = pcDAO.TimKiem(tbTimKiem.Text);  } |

**5.6.2.5 Quản lý lương**

Người dùng có thể xem lương của từng nhân viên bằng cách chọn mã nhân viên trong Combobox MaNV hoặc có thể xem tất cả bằng cách bấm nút “Xem tất cả”. Còn thanh tổng lương ở phía dưới sẽ tính tổng lương của tất cả các bộ hiện lên trên datagridview của bảng TinhLuong.



Chọn nhân viên bằng combobox MaNV

|  |
| --- |
| private void cbMaNV\_TextChanged(object sender, EventArgs e)  {  try  {  NhanVien nv = nhanVienDAO.LayThongTinNhanVienBangMaNV(cbMaNV.Text);  if (nv != null)  {  LoadThongTinNV(nv);  dtgvTinhLuong.DataSource = tinhLuongDAO.LayDanhSach(nv.MaNV);  }  else  throw new Exception();  }  catch  {  ResetThongTinNV();  }  } |

**5.6.2.6 Quản lý doanh thu**

Lọc doanh thu từ ngày mấy đến ngày mấy. Thanh tổng doanh thu là tổng kết toàn bộ tổng tiền thanh toán thu được của cửa hàng. Thanh “Doanh thu lọc” là tính tổng tiền thanh toán của những hàng có hiện trên bảng doanh thu, còn thanh “Tiền lời lọc” là tính tổng tiền lời trên bảng doanh thu.

