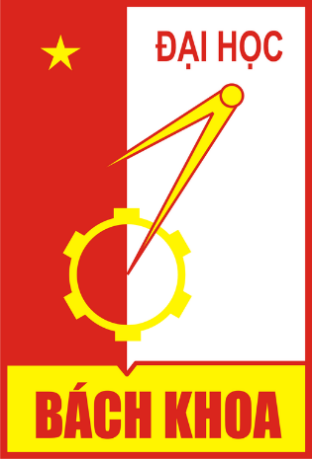
**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN ĐIỆN TỬ - VIỄN THÔNG  
\*\*\*\*\* □□ \*\*\*\*\***



**BÁO CÁO BÀI TẬP LỚN:  
*Kỹ thuật phần mềm ứng dụng***

**ĐỀ TÀI:**

**HỆ THỐNG QUẢN LÍ BÁN HÀNG CỦA HÀNG SMARTPHONE**

***Giảng viên hướng dẫn: TS. Lê Thị Lan***

|  |  |
| --- | --- |
| **Sinh viên thực hiện:** | **Kiều Trung Hiếu** *20192842* |
| **Nguyễn Quang Minh** *20193016* |
| **Phạm Viết Anh Cương** *20192733* |
| **Nguyễn Quý Hùng** *2012848* |
| **Mã lớp:** | **129212** |

***Hà Nội, tháng 2/2022***

MỤC LỤC

[DANH MỤC HÌNH VẼ 3](#_Toc74778269)

[DANH MỤC BẢNG BIỂU 3](#_Toc74778270)

[LỜI MỞ ĐẦU 4](#_Toc74778271)

[CHƯƠNG I: KHẢO SÁT HIỆN TRẠNG VÀ XÁC LẬP DỰ ÁN 5](#_Toc74778272)

[**1.1.** **Khảo sát hiện trạng** 5](#_Toc74778273)

[***1.1.1.*** ***Nghiệp vụ của hệ thống hiện tại*** 5](#_Toc74778274)

[***1.1.2.*** ***Nhược điểm của hệ thống hiện tại*** 5](#_Toc74778275)

[**1.2.** **Mục đích của dự án** 6](#_Toc74778276)

[**1.3.** **Phân tích tính khả thi** 6](#_Toc74778277)

[***1.3.1.*** ***Tính khả thi về mặt kỹ thuật*** 6](#_Toc74778278)

[***1.3.2.*** ***Tính khả thi về mặt kinh tế*** 6](#_Toc74778279)

[***1.3.3.*** ***Tính khả thi về mặt tổ chức*** 6](#_Toc74778280)

[**1.4.** **Xác lập dự án** 7](#_Toc74778281)

[CHƯƠNG II: TÌM HIỂU YÊU CẦU 10](#_Toc74778282)

[**2.1.** **Các kỹ thuật được sử dụng:** 10](#_Toc74778283)

[**2.2.** **Các yêu cầu được thu thập** 13](#_Toc74778284)

[**2.3.** **Phân loại yêu cầu** 13](#_Toc74778285)

[***2.3.1.*** ***Yêu cầu chức năng :*** 13](#_Toc74778286)

[***2.3.2.*** ***Yêu cầu phi chức năng (phần lập trình)*** 14](#_Toc74778287)

[CHƯƠNG III: PHÂN TÍCH HỆ THỐNG 15](#_Toc74778288)

[**3.1.** **Biểu đồ ca sử dụng** 15](#_Toc74778289)

[3.2. Biểu đồ hoạt động 16](#_Toc74778290)

[CHƯƠNG IV: THIẾT KẾ CƠ SỞ DỮ LIỆU 24](#_Toc74778294)

[**4.1.** **Nhận diện thực thể, thuộc tính và xây dựng từ điển dữ liệu** 24](#_Toc74778295)

[4.1.1. Các thực thể và thuộc tính 24](#_Toc74778296)

[4.1.2. Từ điển dữ liệu pha phân tích 25](#_Toc74778297)

[4.2. Xây dựng mô hình thực thể liên kết 25](#_Toc74778298)

[4.3. Chuyển mô hình thực thể liên kết sang mô hình quan hệ 25](#_Toc74778299)

[4.4. Sơ đồ quan hệ 28](#_Toc74778300)

[**4.5.** **Từ điển dữ liệu pha thiết kế** 28](#_Toc74778301)

[4.6. Thiết kế một số form đầu ra 31](#_Toc74778302)

[4.6.1. Hóa đơn 31](#_Toc74778303)

[4.6.2. Phiếu nhập 32](#_Toc74778304)

[4.6.3. Báo cáo doanh thu 32](#_Toc74778305)

[4.6.4. Thống kê hàng tồn kho 33](#_Toc74778306)

[KẾT LUẬN 34](#_Toc74778307)

PHỤ LỤC……………………………………………………………………...36

**5.1 Tổ chức dữ liệu trong SQL Server**….…………………………….…..36

**5.2 Lập trình cho hệ thống**……………………………………….……….39

**6.1 Giao diện hệ thống**…………………………………….………………64

# DANH MỤC BẢNG BIỂU

Bảng 4.1 Tên các bảng dữ liệu…………………………………………………25

Bảng 4.2 Các thuộc tính bảng Khách hàng………………………..……………28

Bảng 4.3 Các thuộc tính bảng Điện thoai...……………………………………29

Bảng 4.4 Các thuộc tính bảng Hóa đơn….………………………..……………29

# LỜI MỞ ĐẦU

Có thể nói rằng, ngày nay các công ty tư nhân hình thành và mọc lên nhiều như nấm, song song với đó là những vấn đề về quản lí nhân sự, quản lí sản phẩm, thương mại sản phẩm, … luôn là bài toán “đau đầu” với các chủ đầu tư. Với mong muốn giải quyết bài toán quản lí sản phẩm trong việc nhập, xuất một cách đơn giản, nhanh chóng, nhóm chúng em đã tìm hiểu và xây dựng một phần mềm đơn giản và có khả năng ứng dụng cao trong thực tế. Cụ thể:

* Tên phần mềm: Phần mềm quản lí cửa hàng SmartPhone.
* Cửa hàng áp dụng: Điện thoại 24H
* Địa chỉ: Chùa Bộc, Đống Đa, Hà Nội.
* Số điện thoại liên hệ: 0384880932

Đề tài này được thực hiện dưới sự hướng dẫn của giảng viên TS.Lê Thị Lan. Qua đây, chúng em xin gửi lời cảm ơn trân thành tới cô đã giúp đỡ nhóm em hoàn thành đề tài này.

Trong quá trình làm báo cáo và sản phẩm, do còn hiểu biết hạn chế nên chúng em mong cô và các nhóm khác góp ý để đề tài thêm hoàn thiện và có nhiều hướng phát triển hơn.

Chúng em xin trân thành cảm ơn!

# CHƯƠNG I: KHẢO SÁT HIỆN TRẠNG VÀ XÁC LẬP DỰ ÁN

* 1. **Khảo sát hiện trạng**
     1. ***Nghiệp vụ của hệ thống hiện tại***

- Thực tế hiện nay với cuộc cách mạng công nghiệp 4.0 thì công nghệ đang được áp dụng ngày càng nhiều vào cuộc sống và các doanh nghiệp, công ty nhỏ lẻ cũng vậy. Hầu hết trước đây mọi hoạt động quản lý sản phẩm đều do con người thực hiện bằng ghi chép thủ công trên giấy tờ và nhập liệu, quản lý trên Microsoft Excel.

- Khi khách hàng tới cửa hàng mua linh kiện, nhân viên cửa hàng sẽ đón tiếp. Sau đó nhân viên sẽ hỏi khách hàng về sản phẩm họ muốn, kiểm tra số lượng hàng hóa kho của cửa hàng. Nếu sản phẩm thỏa mãn nhu cầu của khách hàng thì nhân viên ghi vào hóa đơn và tính toán số tiền, còn không sẽ tư vấn cho khách hàng các sản phẩm của cửa hàng. Hóa đơn thanh toán gồm 2 liên, liên 1 để lưu lại, liên 2 đưa cho khách hàng. Cuối cùng, nhân viên nhập lại tổng tiền của hóa đơn vào excel để thống kê khi cần.

- Khi chủ cửa hàng muốn biết kết quả kinh doanh, số lượng xuất nhập trong kho hàng chủ cửa hàng sẽ xem thống kê từ excel theo từng ngày, tháng hay năm.

* + 1. ***Nhược điểm của hệ thống hiện tại***

Có thể thấy rõ, nhược điểm lớn nhất của hệ thống bộc lộ ở khâu quản lý, đó chính là việc chưa ứng dụng được công nghệ vào để nâng cao tính chính xác cũng như hiệu quả làm việc. Việc quản lý sản phẩm với số lượng lớn và doanh thu đều dựa trên ghi chép hóa đơn và sử dụng excel để quản lý, thống kê khó thể hiện được kết quả kinh doanh, khó nắm bắt được tình hình kinh doanh, tốn thời gian trong việc bán hàng và quản lý, cũng khó đảm bảo được sự minh bạch trong vấn đề tiền bạc. Khối lượng dữ liệu ngày càng lớn sẽ gây rắc rối, phức tạp và tốn rất nhiều thời gian cũng như nhân lực cho hoạt động nhập liệu, thống kê và báo cáo kết quả kinh doanh. Tuy nhiên viêc sử dụng phần mềm excel chưa chuyên dụng, chưa giúp cửa hàng đạt được hiệu quả tối đa.

* Gây ra rất nhiều khó khăn trong khâu quản lý cũng như vận hành.
  1. **Mục đích của dự án**

Từ những nhược điểm của hình thức quản lý hiện tại, yêu cầu đặt ra là phải sử dụng một phương thức, một công cụ quản lý mang lại tính nhanh gọn, hiệu quả và quan trọng nhất là chính xác. Phần mềm quản lý cửa hàng SmartPhone ra đời với các mục đích chính:

* Đưa ra ưu điểm của hệ thống mới: Nhanh chóng, hiệu quả, chính xác, quản lý dễ dàng, ...
* Nâng cao hiệu quả quản lý, từ đó đem lại hiệu quả kinh doanh.
* Khắc phục được hạn chế về chi phí và con người: Tiết kiệm được nguồn nhân lực, chi phí nhân công.
  1. **Phân tích tính khả thi**
     1. ***Tính khả thi về mặt kỹ thuật***

- Sản phẩm phát triển sử dụng công cụ C# và SQL Server, đây là những công cụ hỗ trợ mạnh mẽ cho lập trình phần mềm sử dụng Windows forms và cơ sở dữ liệu, tuy nhiên lại tốn rất ít tài nguyên hệ thống.

- Sản phẩm được thiết kế chạy trên nền Windows, phù hợp với đa số người dùng cũng như điều kiện về trang thiết bị của khách hàng. Dự án có quy mô nhỏ, dữ liệu không quá lớn.

- Yếu tố cản trở việc xây dựng thành công hệ thống: Bản thân sinh viên thực hiện dự án chưa có nhiều kiến thức về lập trình C# cũng như SQL Server, điều này ảnh hưởng một chút tới tính khả thi.

* + 1. ***Tính khả thi về mặt kinh tế***

- Sản phẩm nếu đi vào hoạt động sẽ giúp giảm thiểu thời gian và nâng cao hiệu quả quản lý, tiết kiệm chi phí hoạt động.

- Sản phẩm phát triển sử dụng công cụ C# và SQL Server, tiêu tốn ít tài nguyên hệ thống, chi phí sản xuất dự kiến không cao (sẽ trình bày tại mục Yêu cầu phi chức năng), yêu cầu về trang thiết bị đối với khách hàng không cao (chỉ yêu cầu 1 máy tính chạy hệ điều hành Windows).

* + 1. ***Tính khả thi về mặt tổ chức***

- Các mục tiêu của dự án hoàn toàn phù hợp với mục tiêu của nghiệp vụ mà khách hàng hướng tới, không những vậy dự án còn hướng tới phục vụ cho chiến lược phát triển lâu dài hơn của khách hàng trong tương lai.

- Sản phẩm sau khi hoàn thành sẽ được triển khai ngay lập tức tại cửa hàng trên hệ thống máy tính quản lý và do nhân viên trong cửa hàng quản lí cũng như chủ cửa hàng sử dụng.

* 1. **Xác lập dự án**

Dự án xây dựng phần mềm quản lí cửa hàng SmartPhone có quy mô nhỏ, áp dụng cho quản lý một cửa hàng.

Phân công công việc và kế hoạch thực hiện dự án được trình bày trong Bảng 1.1.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TT | Tên công việc | Bắt đầu | Kết thúc | Tài nguyên | Kết quả | Trạng thái |
| 1 | Khảo sát hiện trạng thu thập thông tin | 12/1/2022 | 12/2/2022 | MS Word | Báo cáo hiện trạng | Hoàn thành |
| 2 | Xác định mục tiêu | 12/1/2022 | 14/1/2022 | MS Word | Các mục tiêu hướng tới | Hoàn thành |
| 3 | Đánh giá khả năng thực hiện | 12/1/2022 | 14/1/2022 | MS Word | Bản phân tích tính khả thi | Hoàn thành |
| 4 | Xác lập dự án | 15/1/2022 | 15/1/2022 | MS Excel | Bản phân công công việc | Hoàn thành |
| 5 | Tìm hiểu yêu cầu khách hàng | 15/1/2022 | 15/1/2022 | MS Word | Bản liệt kê yêu cầu | Hoàn thành |
| 6 | Mô tả yêu cầu | 17/1/2021 | 17/1/2022 | MS Word | Bản mô tả yêu cầu chức năng và phi chức năng | Hoàn thành |
| 7 | Phân tích hệ thống | 16/1/2022 | 1/2/2022 | MS Visio | Sơ đồ chức năng, sơ đồ luồng dữ liệu | Hoàn thành |
| 8 | Thiết kế CSDL | 6/2/2022 | 6/2/2022 |  |  | Hoàn thành |
| 8.1 | Nhận diện thực thể của hệ thống, liệt kê thuộc tính, lập từ điển dữ liệu | 7/2/2022 | 8/2/2022 |  |  | Hoàn thành |
| 8.2 | Đưa ra các giả thiết hợp lý về các phụ thuộc hàm cần có | 12/2/2022 | 12/2/2022 |  |  | Hoàn thành |
| 8.3 | Chuẩn hóa CSDL | 14/2/2022 | 14/2/2022 |  |  | Hoàn thành |
| 8.4 | Vẽ sơ đồ thực thể liên kết của hệ thống | 15/2/2022 | 16/2/2022 |  |  | Hoàn thành |
| 8.5 | Mô tả lược đồ khái niệm của hệ thống | 17/2/2022 | 17/2/2022 |  |  | Hoàn thành |
| 8.6 | Thiết kế form báo cáo đầu ra | 15/2/2022 | 20/2/2022 |  |  | Hoàn thành |
| 9 | Lập trình giao diện cho phần mềm | 15/2/2022 | 20/2/2022 |  |  | Hoàn thành |
| 10 | Kiểm thử và sửa lỗi | 21/2/2022 | 21/2/2022 |  |  | Hoàn thành |
| 11 | Bàn giao sản phẩm | 25/2/2022 | 25/2/2022 |  |  | Hoàn thành |

*Bảng 1. 1 Bảng phân công công việc và kế hoạch thực hiện*

# CHƯƠNG II: TÌM HIỂU YÊU CẦU

* 1. **Các yêu cầu được thu thập**

- Mỗi khi có sản phẩm mới được nhập về cửa hàng, người quản lí sẽ tiến hành nhập thông tin vào hệ thống. Các thông tin bao gồm: mã sản phẩm, tên sản phẩm, loại sản phẩm, ngày nhập, giá nhập, số lượng nhập, thông số kĩ thuật, giá bán tham khảo, thời gian bảo hành. Các thông số kĩ thuật của điện thoại là mã máy, màu sắc, tình trạng, chip, bộ xử lí, quốc gia, màn hình, ổ cứng, pin, màn hình, cổng giao tiếp, cân nặng. Các thông số kĩ thuật cho sản phẩm linh kiện là một dòng ghi chú thông số máy trên linh kiện.

- Khi khách hàng mua hàng, nhân viên thu ngân sẽ nhập thông tin hàng hóa, thông tin khách hàng, lập hóa đơn, lưu vào hệ thống và in ra đưa cho khách hàng. Hóa đơn khi đã lưu có thể được chỉnh sửa hoặc xóa bởi chủ cửa hàng hoặc nhân viên trong cửa hàng. Hóa đơn bao gồm: Mã hóa đơn, ngày lập, tên khách hàng, địa chỉ khách hàng, số điện thoại khách hàng, tổng giá trị của hóa đơn, thông tin về sản phẩm (thông số đi kèm, số lượng, thời gian bảo hành, giá thành).

- Hệ thống hỗ trợ nhân viên và chủ cửa hàng có thể tìm kiếm thông tin về: Tìm kiếm điện thoại theo các tiêu chí lựa chọn về tên máy, mã máy, tìm kiếm hóa đơn.

* Hệ thống sẽ lưu lại thông tin của khách hàng bao gồm: mã khách hàng, tên khách hàng, số điện thoại liên hệ, địa chỉ.
* Hệ thống hỗ trợ chủ cửa hàng hoặc nhân viên có thể lập các báo cáo thống kê về doanh thu, số tiền lãi trong một khoảng thời gian tùy chọn (ngày, tháng, năm). Số lượng của các sản phẩm còn trong kho, sản phẩm đã hết hàng, sản phẩm đã bán được trong một khoảng thời gian tùy chọn (ngày, tháng, năm).
  1. **Phân loại yêu cầu**
     1. ***Yêu cầu chức năng:***

Phần mềm cần có các chức năng sau:

* Quản lý khách hàng: Thêm khách hàng mới, cập nhập khách hàng, xóa khách hàng.
* Quản lí sản phẩm: Thêm sản phẩm mới, cập nhật thông tin sản phẩm, xóa sản phẩm không còn được bán.
* Tạo hóa đơn mới: Thêm hóa đơn mới, cập nhập hóa đơn, xuất hóa đơn, xóa hóa đơn.
* Tìm kiếm: Tìm kiếm sản phẩm, tìm kiếm hóa đơn, tìm kiếm khách hàng.
* Báo cáo và thống kê: Chọn ngày bắt đầu và kết thúc, hiển thị chi tiết các khoản thu, chi, hàng còn trong kho, hàng đã hết. Cho phép xuất ra file excel hoặc in báo cáo.
  + 1. ***Yêu cầu phi chức năng:***

- Hoạt động trên máy tính chạy Window 7/8/8.1/10. Có ổ cứng trên 50GB, ram 4G trở lên

- Ngôn ngữ: Tiếng Việt.

- Đơn vị tiền tệ: VNĐ.

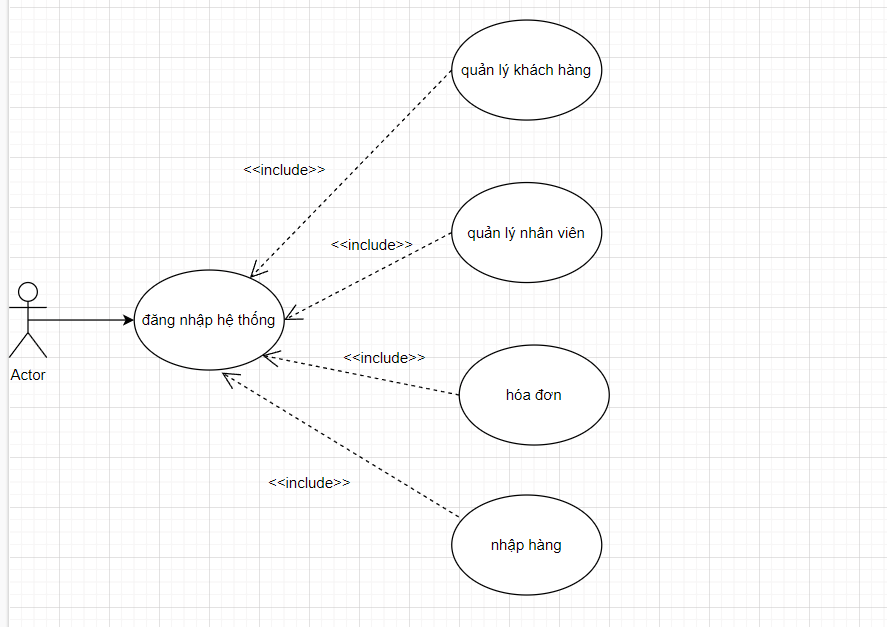
- Tính toán chính xác, dung lượng chứa thông tin đảm bảo đủ lớn để xử lí dữ liệu lớn.

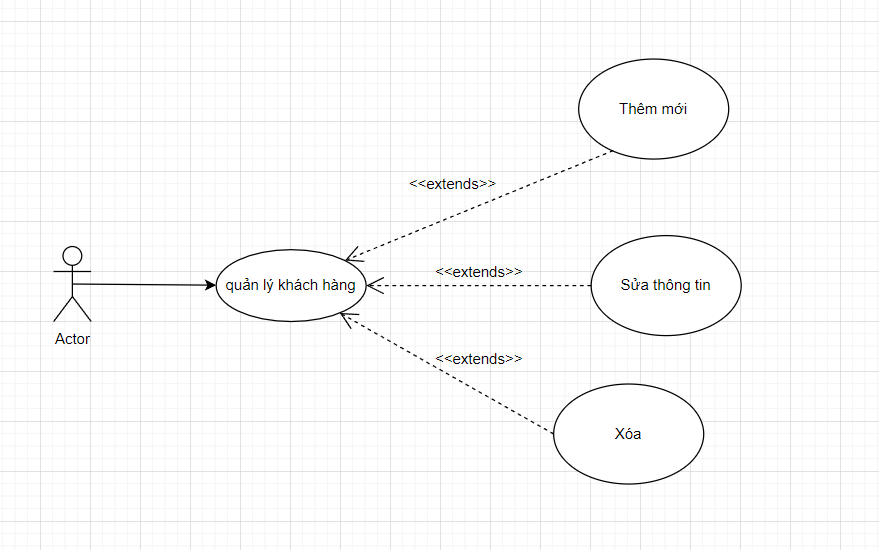
- Thời gian thực hiện: 3 tháng.

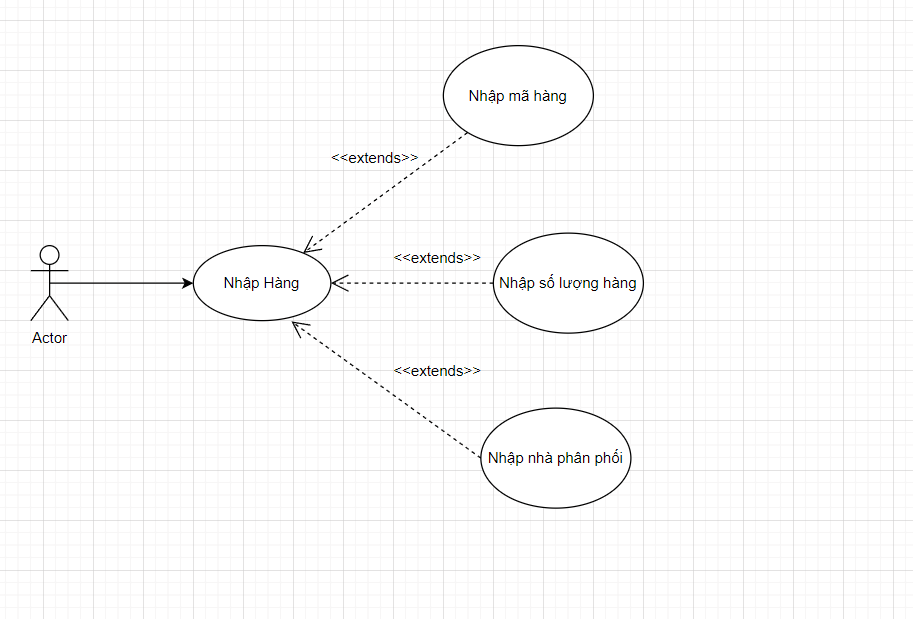
/

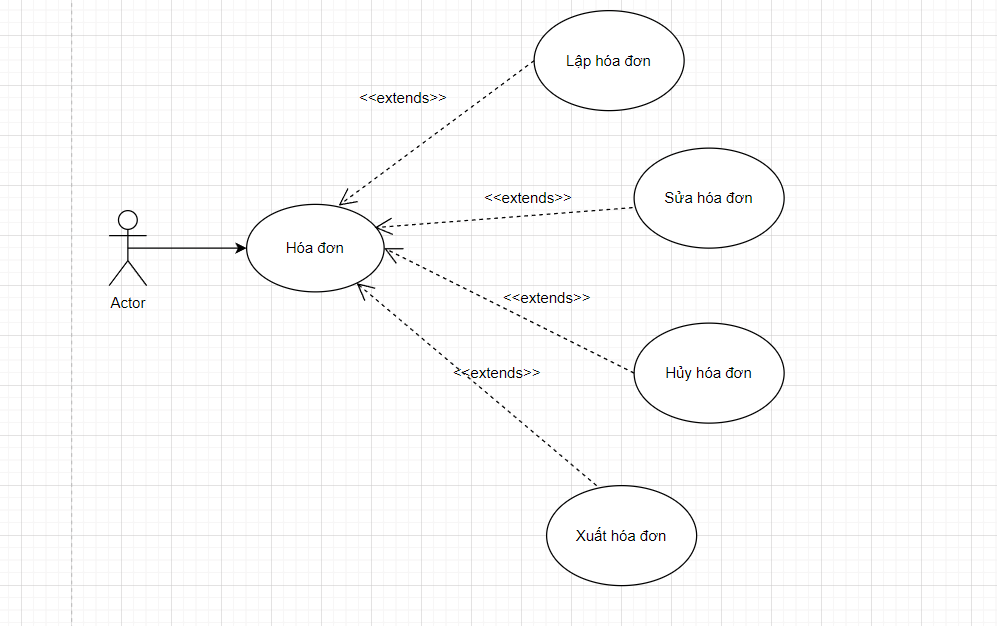
# CHƯƠNG III: PHÂN TÍCH HỆ THỐNG

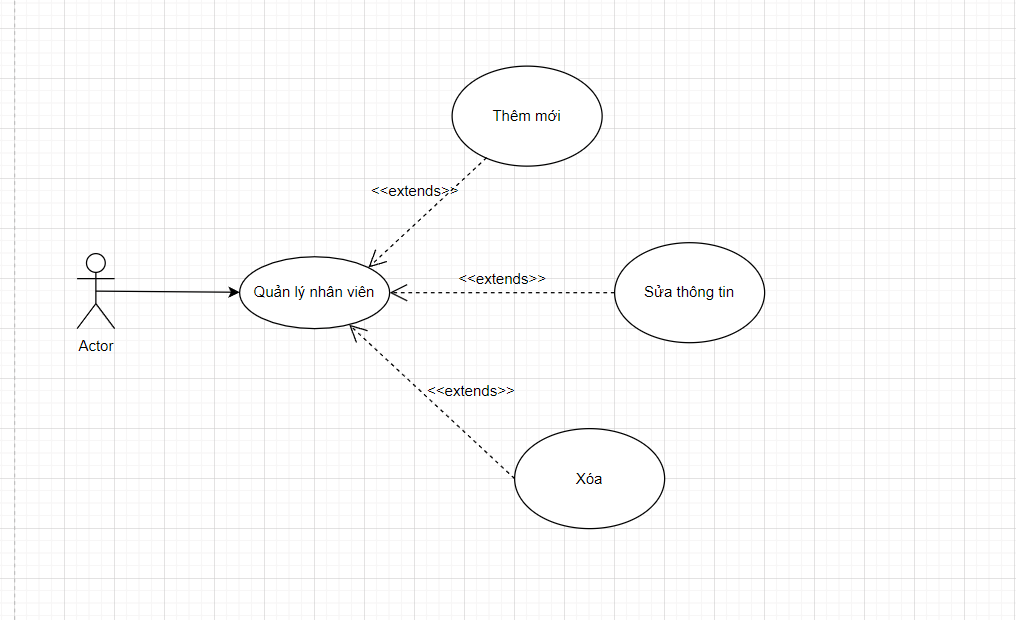
***3.1. Biểu đồ ca sử dụng:***

******

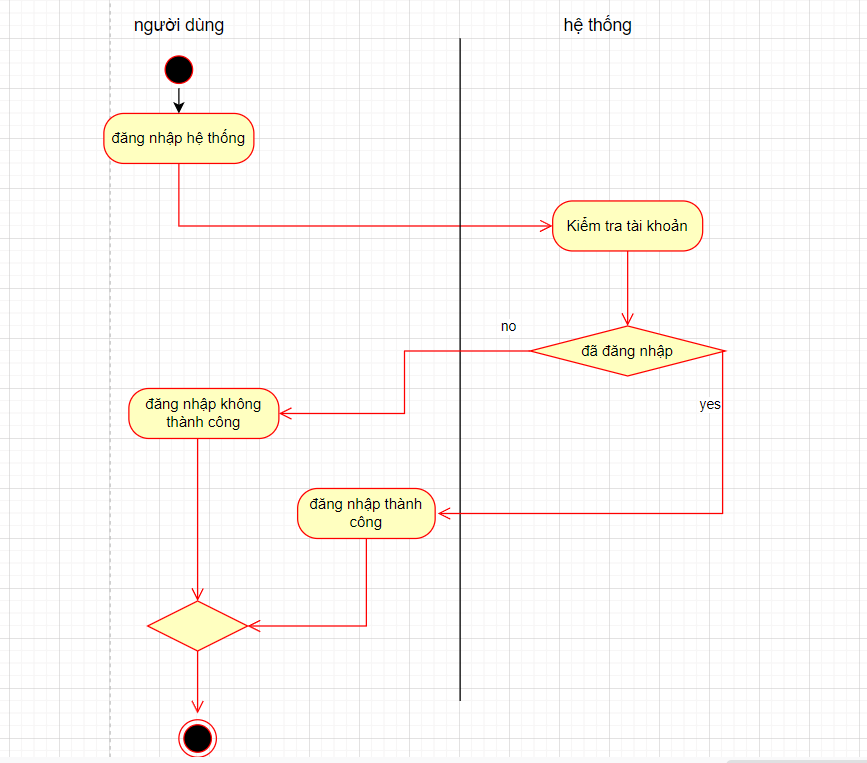
******

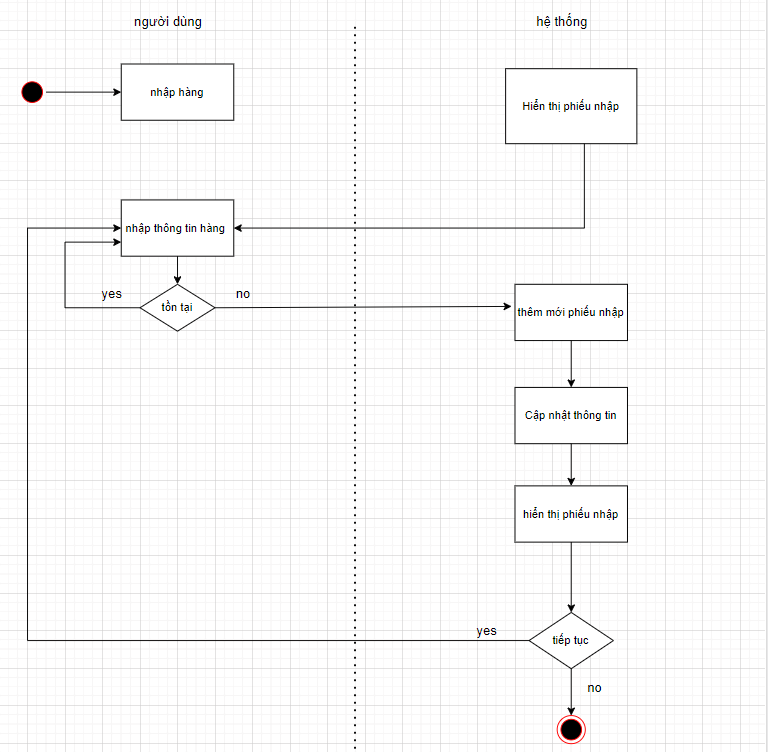
******

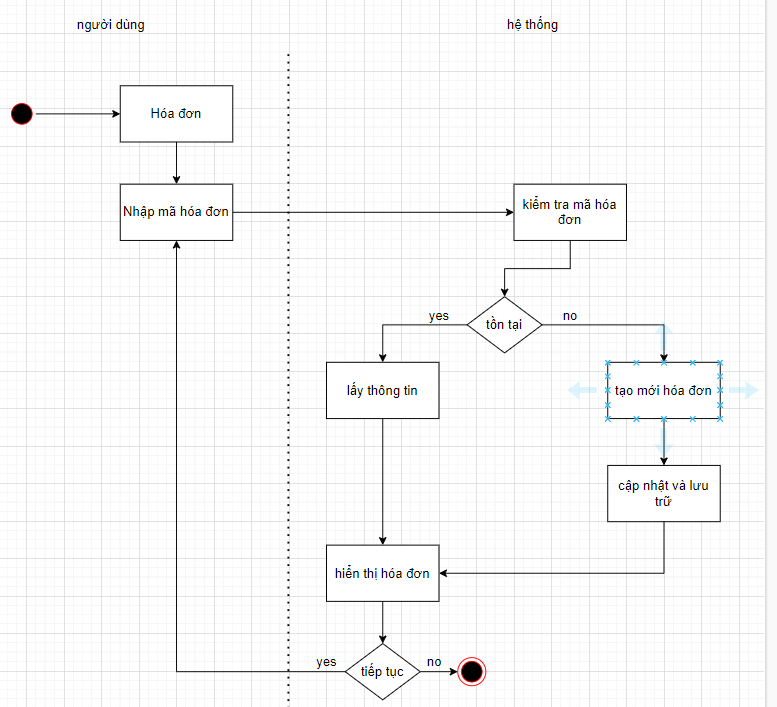
******



***3.2.Biểu đồ hoạt động:***







# CHƯƠNG IV: THIẾT KẾ CƠ SỞ DỮ LIỆU

* 1. **Nhận diện thực thể, thuộc tính và xây dựng từ điển dữ liệu**

### Các thực thể và thuộc tính

|  |
| --- |
| **Sản phẩm** |
| - Mã sản phẩm  - Loại sản phẩm  - Tên sản phẩm  - Số lượng  - Giá nhập  - Giá bán  - Ảnh  - Ghi chú |

|  |
| --- |
| **Khách hàng** |
| - Mã khách hàng  - Tên khách hàng  - Số điện thoại  - Địa chỉ |

|  |
| --- |
| **Hóa đơn** |
| - Mã hóa đơn  - Mã khách hang  - Mã nhân viên  - Ngày bán  - Tổng tiền |

|  |
| --- |
| **Chi tiết hóa đơn** |
| - Mã hóa đơn  - Mã sản phẩm  - Số lượng  - Chiết khấu  - Thành tiền |

|  |
| --- |
| **Tài Khoản** |
| - Tên đăng nhập  - Mật khẩu |

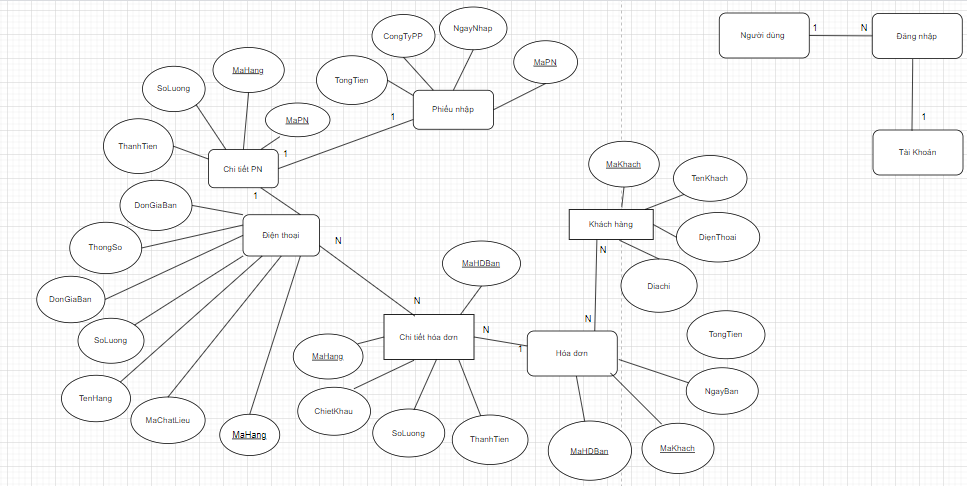
|  |
| --- |
| **Nhân viên** |
| - Mã nhân viên  - Tên nhân viên  - Giới tính  - Địa chỉ  - Điện thoại  - Ngày sinh |

### Từ điển dữ liệu pha phân tích

|  |  |  |
| --- | --- | --- |
| **Tên thực thể** | **Tên bảng** | **Ghi chú** |
| Điện thoại | Hang | = MaHang +MaChatLieu +TenHang + SoLuong + DonGiaNhap + DonGiaBan + Anh + GhiChu |
| Khách hàng | KhachHang | = MaKhach + TenKhach + DienThoai + DiaChi |
| Hóa đơn | HoaDon | = MaHDBan + MaNhanVien + MaKhach + NgayBan + TongTien |
| Hãng điện thoại | MaChatLieu | = MaChatLieu + TenChatLieu |
| Chi tiết hóa đơn | ChiTietHDBan | = MaHDBan + MaHang + ChietKhau + SoLuong + ThanhTien |
| Chi tiết phiếu nhập | CTPN | = MaPN + MaHang + SoLuong + ThanhTien |

Bảng 4. 1 Tên các bảng dữ liệu

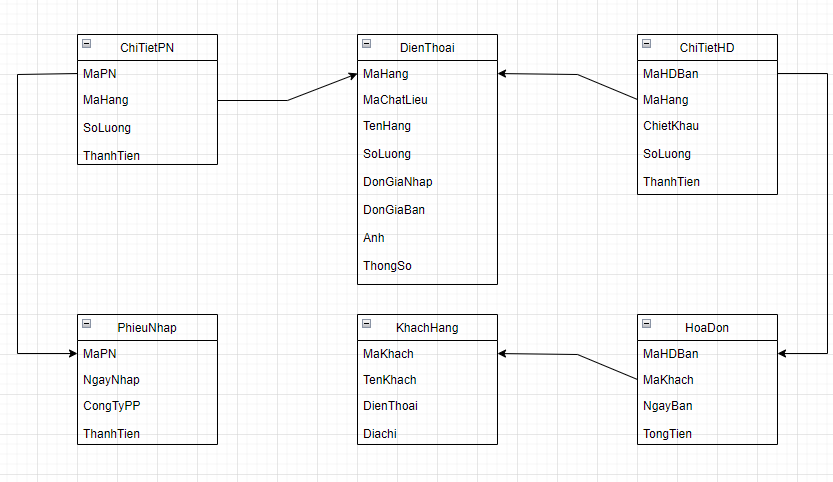
## Xây dựng mô hình thực thể liên kết



## Chuyển mô hình thực thể liên kết sang mô hình quan hệ

* + - Chuyển các thực thể và các liên kết thành các quan hệ
* Khách hàng (MaKhach, TenKhach, DienThoai, DiaChi)
* Điện Thoại (MaHang, TenHang, MaChatLieu, SoLuong, DonGiaNhap, DonGiaBan, ThongSo)
* Hóa đơn (MaHDBan, MaKhach, NgayBan, TongTien)
* Hãng điện thoại (MaChatLieu, TenChatLieu)
* Phiếu nhập (MaPN, NgayNhap, CongTyPP, TongTien)
* Chi tiết phiếu nhập (MaPN, MaSP, SoLuong, ThanhTien)
* Chỉ tiết hóa đơn (MaHDBan, MaHang, ChietKhau, SoLuong, ThanhTien)

## Sơ đồ quan hệ



* 1. **Từ điển dữ liệu pha thiết kế**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Chiều dài** | **Giá trị mẫu** | **Mô tả** | **Ghi chú** |
| MaKhach | VARCHAR | 20 | KH1 | Mã khách hàng | NOT NULL |
| TenKhach | VARCHAR | 50 | Nguyễn Xuân Đạt | Tên khách hàng | NOT NULL |
| SDT | VARCHAR | 20 | 0365147951 | Số điện thoại khách hàng | NOT NULL |
| DiaChi | VARCHAR | MAX | Phú Thọ, Hà Nội | Địa chỉ khách hàng | NOT NULL |

*Bảng 4. 2 Các thuộc tính bảng Khách hàng*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Chiều dài** | **Giá trị mẫu** | **Mô tả** | **Ghi chú** |
| MaHang | NVARCHAR | 20 | ĐT01 | Mã sản phẩm | NOT NULL |
| MaChatLieu | NVARCHAR | 50 | Iphone | Loại sản phẩm | NOT NULL |
| TenHang | NVARCHAR | 50 | Iphone X | Tên Sản phẩm | NOT NULL |
| SoLuong | INT |  | 100 | Số lượng còn trong kho | NOT NULL |
| GiaNhap | INT |  | 6000000 | Đơn giá lúc nhập | NOT NULL |
| GiaBan | INT |  | 6500000 | Giá bán ra | NOT NULL |
| Ghi chú | NVARCHAR | MAX | Hàng new like 99% | Thông số của sản phẩm | NOT NULL |

Bảng 4. 3 Các thuộc tính bảng Sản phẩm

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Chiều dài** | **Giá trị mẫu** | **Mô tả** | **Ghi chú** |
| MaHDBan | VARCHAR | 20 | HDB2232022-221613 | Mã hóa đơn | NOT NULL |
| MaKhach | VARCHAR | 20 | Nguyễn Xuân Đạt | Mã khách hàng | NOT NULL |
| NgayBan | DATETIME |  | 2/23/2022 | Ngày bán | NOT NULL |
| Tongtien | INT |  | 13000000 | Tổng tiền | NOT NULL |

Bảng 4. 4 Các thuộc tính bảng Hóa đơn

## Thiết kế một số form đầu ra

### Hóa đơn

******

# KẾT LUẬN

Sau một thời gian học tập và nghiên cứu, đề tài đã được hoàn thành đúng thời hạn. Tuy nhiên do hạn chế về kiến thức, thời gian và kỹ năng, phần mềm chưa hoàn thiện được các chức năng cần có. Một lần nữa xin cảm ơn cô Lê Thị Lan đã tận tình hướng dẫn trong suốt thời gian thực hiện đề tài này.

Xin chân thành cảm ơn cô!

**PHỤ LỤC**

Database sử dụng Microsoft SQL Server Management Sudio 18 Code

C# sử dụng MicroSoft Visual Studio 2019

**5.1. Tổ chức dữ liệu trong SQL Server**

***5.1.1. Khách Hàng (KhachHang)***

CREATE TABLE [dbo].[KhachHang] (

[MaKhach] NVARCHAR (50) NOT NULL,

[TenKhach] NVARCHAR (50) NULL,

[DiaChi] NVARCHAR (50) NULL,

[DienThoai] NVARCHAR (50) NULL,

PRIMARY KEY CLUSTERED ([MaKhach] ASC)

);

CONSTRAINT [PK\_KhachHang] PRIMARY KEY CLUSTERED

(

[MaKH] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

***5.1.2. Sản phẩm (SanPham)***

CREATE TABLE [dbo].[Hang] (

[MaHang] NVARCHAR (50) NOT NULL,

[TenHang] NVARCHAR (50) NULL,

[MaChatLieu] NVARCHAR (50) NULL,

[SoLuong] INT NULL,

[DonGiaNhap] FLOAT (53) NULL,

[DonGiaBan] FLOAT (53) NULL,

[Anh] NVARCHAR (50) NULL,

[GhiChu] NVARCHAR (50) NULL,

PRIMARY KEY CLUSTERED ([MaHang] ASC)

);

CONSTRAINT [PK\_Hang] PRIMARY KEY CLUSTERED

(

[Hang] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

***5.1.3. Hóa đơn (HoaDon)***

CREATE TABLE [dbo].[HDBan] (

[MaHDBan] NVARCHAR (50) NOT NULL,

[MaNhanVien] NVARCHAR (50) NULL,

[NgayBan] DATETIME NULL,

[MaKhach] NVARCHAR (50) NULL,

[TongTien] FLOAT (53) NULL,

PRIMARY KEY CLUSTERED ([MaHDBan] ASC)

);

CONSTRAINT [PK\_MaHDBan] PRIMARY KEY CLUSTERED

(

[MaHDBan] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[MaHDBan] WITH CHECK ADD CONSTRAINT [FK\_HoaDon\_KhachHang] FOREIGN KEY([MaKhach])

REFERENCES [dbo].[KhachHang] ([MaKH])

GO

ALTER TABLE [dbo].[MaHDBan] CHECK CONSTRAINT [FK\_HoaDon\_KhachHang]

GO

***5.1.5. Chi tiết hóa đơn (CTHD)***

CREATE TABLE [dbo].[ChiTietHDBan] (

[MaHDBan] NVARCHAR (50) NOT NULL,

[MaHang] NVARCHAR (50) NULL,

[SoLuong] INT NULL,

[DonGia] FLOAT (53) NULL,

[GiamGia] FLOAT (53) NULL,

[ThanhTien] FLOAT (53) NULL,

PRIMARY KEY CLUSTERED ([MaHDBan] ASC)

);

**5.2. Lập trình cho hệ thống**

**5.2.1.Connection**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

namespace QuanLyDT

{

class Connection

{

private static string stringConnection = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Admin- Thính\source\repos\QuanLyDT\QuanLyDT\QuanLyCuaHangDT.mdf;Integrated Security=True";

public static SqlConnection GetSqlConnection()

{

return new SqlConnection(stringConnection);

}

}

}

***5.2.2. Functions***

***using System;***

***using System.Collections.Generic;***

***using System.Linq;***

***using System.Text;***

***using System.Threading.Tasks;***

***using System.Data;***

***using System.Data.SqlClient;***

***using System.Windows.Forms; // Sử dụng đối tượng MessageBox***

***namespace QuanLyDT.Class***

***{***

***class Functions***

***{***

***public static SqlConnection Con; //Khai báo đối tượng kết nối***

***public static void Connect()***

***{***

***Con = new SqlConnection(); //Khởi tạo đối tượng***

***Con.ConnectionString = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Admin-Thính\source\repos\QuanLyDT\QuanLyDT\QuanLyCuaHangDT.mdf;Integrated Security=True";***

***Con.Open(); //Mở kết nối***

***}***

***//Lấy dữ liệu vào bảng***

***public static DataTable GetDataToTable(string sql)***

***{***

***SqlDataAdapter dap = new SqlDataAdapter(); //Định nghĩa đối tượng thuộc lớp SqlDataAdapter***

***//Tạo đối tượng thuộc lớp SqlCommand***

***dap.SelectCommand = new SqlCommand();***

***dap.SelectCommand.Connection = Functions.Con; //Kết nối cơ sở dữ liệu***

***dap.SelectCommand.CommandText = sql; //Lệnh SQL***

***//Khai báo đối tượng table thuộc lớp DataTable***

***DataTable table = new DataTable();***

***dap.Fill(table);***

***return table;***

***}***

***//Hàm kiểm tra khoá trùng***

***public static bool CheckKey(string sql)***

***{***

***SqlDataAdapter dap = new SqlDataAdapter(sql, Con);***

***DataTable table = new DataTable();***

***dap.Fill(table);***

***if (table.Rows.Count > 0)***

***return true;***

***else return false;***

***}***

***//Hàm thực hiện câu lệnh SQL***

***public static void RunSQL(string sql)***

***{***

***SqlCommand cmd; //Đối tượng thuộc lớp SqlCommand***

***cmd = new SqlCommand();***

***cmd.Connection = Con; //Gán kết nối***

***cmd.CommandText = sql; //Gán lệnh SQL***

***try***

***{***

***cmd.ExecuteNonQuery(); //Thực hiện câu lệnh SQL***

***}***

***catch (Exception ex)***

***{***

***MessageBox.Show(ex.ToString());***

***}***

***cmd.Dispose();//Giải phóng bộ nhớ***

***cmd = null;***

***}***

***public static void RunSqlDel(string sql)***

***{***

***SqlCommand cmd = new SqlCommand();***

***cmd.Connection = Functions.Con;***

***cmd.CommandText = sql;***

***try***

***{***

***cmd.ExecuteNonQuery();***

***}***

***catch (Exception ex)***

***{***

***//MessageBox.Show("Dữ liệu đang được dùng, không thể xoá...", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Stop);***

***MessageBox.Show(ex.ToString());***

***}***

***cmd.Dispose();***

***cmd = null;***

***}***

***public static void FillCombo(string sql, ComboBox cbo, string ma, string ten)***

***{***

***SqlDataAdapter dap = new SqlDataAdapter(sql, Con);***

***DataTable table = new DataTable();***

***dap.Fill(table);***

***cbo.DataSource = table;***

***cbo.ValueMember = ma; //Trường giá trị***

***cbo.DisplayMember = ten; //Trường hiển thị***

***}***

***public static string GetFieldValues(string sql)***

***{***

***string ma = "";***

***SqlCommand cmd = new SqlCommand(sql, Con);***

***SqlDataReader reader;***

***reader = cmd.ExecuteReader();***

***while (reader.Read())***

***ma = reader.GetValue(0).ToString();***

***reader.Close();***

***return ma;***

***}***

***//123 => một trăm hai ba đồng***

***//1,123,000=>một triệu một trăm hai ba nghìn đồng***

***//1,123,345,000 => một tỉ một trăm hai ba triệu ba trăm bốn lăm ngàn đồng***

***static string[] mNumText = "không;một;hai;ba;bốn;năm;sáu;bảy;tám;chín".Split(';');***

***//Viết hàm chuyển số hàng chục, giá trị truyền vào là số cần chuyển và một biến đọc phần lẻ hay không ví dụ 101 => một trăm lẻ một***

***private static string DocHangChuc(double so, bool daydu)***

***{***

***string chuoi = "";***

***//Hàm để lấy số hàng chục ví dụ 21/10 = 2***

***Int64 chuc = Convert.ToInt64(Math.Floor((double)(so / 10)));***

***//Lấy số hàng đơn vị bằng phép chia 21 % 10 = 1***

***Int64 donvi = (Int64)so % 10;***

***//Nếu số hàng chục tồn tại tức >=20***

***if (chuc > 1)***

***{***

***chuoi = " " + mNumText[chuc] + " mươi";***

***if (donvi == 1)***

***{***

***chuoi += " mốt";***

***}***

***}***

***else if (chuc == 1)***

***{//Số hàng chục từ 10-19***

***chuoi = " mười";***

***if (donvi == 1)***

***{***

***chuoi += " một";***

***}***

***}***

***else if (daydu && donvi > 0)***

***{//Nếu hàng đơn vị khác 0 và có các số hàng trăm ví dụ 101 => thì biến daydu = true => và sẽ đọc một trăm lẻ một***

***chuoi = " lẻ";***

***}***

***if (donvi == 5 && chuc >= 1)***

***{//Nếu đơn vị là số 5 và có hàng chục thì chuỗi sẽ là " lăm" chứ không phải là " năm"***

***chuoi += " lăm";***

***}***

***else if (donvi > 1 || (donvi == 1 && chuc == 0))***

***{***

***chuoi += " " + mNumText[donvi];***

***}***

***return chuoi;***

***}***

***private static string DocHangTram(double so, bool daydu)***

***{***

***string chuoi = "";***

***//Lấy số hàng trăm ví du 434 / 100 = 4 (hàm Floor sẽ làm tròn số nguyên bé nhất)***

***Int64 tram = Convert.ToInt64(Math.Floor((double)so / 100));***

***//Lấy phần còn lại của hàng trăm 434 % 100 = 34 (dư 34)***

***so = so % 100;***

***if (daydu || tram > 0)***

***{***

***chuoi = " " + mNumText[tram] + " trăm";***

***chuoi += DocHangChuc(so, true);***

***}***

***else***

***{***

***chuoi = DocHangChuc(so, false);***

***}***

***return chuoi;***

***}***

***private static string DocHangTrieu(double so, bool daydu)***

***{***

***string chuoi = "";***

***//Lấy số hàng triệu***

***Int64 trieu = Convert.ToInt64(Math.Floor((double)so / 1000000));***

***//Lấy phần dư sau số hàng triệu ví dụ 2,123,000 => so = 123,000***

***so = so % 1000000;***

***if (trieu > 0)***

***{***

***chuoi = DocHangTram(trieu, daydu) + " triệu";***

***daydu = true;***

***}***

***//Lấy số hàng nghìn***

***Int64 nghin = Convert.ToInt64(Math.Floor((double)so / 1000));***

***//Lấy phần dư sau số hàng nghin***

***so = so % 1000;***

***if (nghin > 0)***

***{***

***chuoi += DocHangTram(nghin, daydu) + " nghìn";***

***daydu = true;***

***}***

***if (so > 0)***

***{***

***chuoi += DocHangTram(so, daydu);***

***}***

***return chuoi;***

***}***

***public static string ChuyenSoSangChuoi(double so)***

***{***

***if (so == 0)***

***return mNumText[0];***

***string chuoi = "", hauto = "";***

***Int64 ty;***

***do***

***{***

***//Lấy số hàng tỷ***

***ty = Convert.ToInt64(Math.Floor((double)so / 1000000000));***

***//Lấy phần dư sau số hàng tỷ***

***so = so % 1000000000;***

***if (ty > 0)***

***{***

***chuoi = DocHangTrieu(so, true) + hauto + chuoi;***

***}***

***else***

***{***

***chuoi = DocHangTrieu(so, false) + hauto + chuoi;***

***}***

***hauto = " tỷ";***

***} while (ty > 0);***

***return chuoi + " đồng";***

***}***

***//Hàm tạo khóa có dạng: TientoNgaythangnam\_giophutgiay***

***public static string CreateKey(string tiento)***

***{***

***string key = tiento;***

***string[] partsDay;***

***partsDay = DateTime.Now.ToShortDateString().Split('/');***

***//Ví dụ 07/08/2009***

***string d = String.Format("{0}{1}{2}", partsDay[0], partsDay[1], partsDay[2]);***

***key = key + d;***

***string[] partsTime;***

***partsTime = DateTime.Now.ToLongTimeString().Split(':');***

***//Ví dụ 7:08:03 PM hoặc 7:08:03 AM***

***if (partsTime[2].Substring(3, 2) == "PM")***

***partsTime[0] = ConvertTimeTo24(partsTime[0]);***

***if (partsTime[2].Substring(3, 2) == "AM")***

***if (partsTime[0].Length == 1)***

***partsTime[0] = "0" + partsTime[0];***

***//Xóa ký tự trắng và PM hoặc AM***

***partsTime[2] = partsTime[2].Remove(2, 3);***

***string t;***

***t = String.Format("\_{0}{1}{2}", partsTime[0], partsTime[1], partsTime[2]);***

***key = key + t;***

***return key;***

***}***

***//Chuyển đổi từ PM sang dạng 24h***

***public static string ConvertTimeTo24(string hour)***

***{***

***string h = "";***

***switch (hour)***

***{***

***case "1":***

***h = "13";***

***break;***

***case "2":***

***h = "14";***

***break;***

***case "3":***

***h = "15";***

***break;***

***case "4":***

***h = "16";***

***break;***

***case "5":***

***h = "17";***

***break;***

***case "6":***

***h = "18";***

***break;***

***case "7":***

***h = "19";***

***break;***

***case "8":***

***h = "20";***

***break;***

***case "9":***

***h = "21";***

***break;***

***case "10":***

***h = "22";***

***break;***

***case "11":***

***h = "23";***

***break;***

***case "12":***

***h = "0";***

***break;***

***}***

***return h;***

***}***

***}***

***}***

**5.2.3. Modify**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

using System.Data;

namespace QuanLyDT

{

class Modify

{

public Modify()

{

}

SqlCommand sqlCommand;//dùng để truy vấn các câu lệnh insert, update, delete...

SqlDataReader dataReader; //dùng để đọc dữ liệu trong bảng

public List<TaiKhoan> TaiKhoans(string query)

{

List<TaiKhoan> taiKhoans = new List<TaiKhoan>();

using(SqlConnection sqlConnection = Connection.GetSqlConnection())

{

sqlConnection.Open();

sqlCommand = new SqlCommand(query, sqlConnection);

dataReader = sqlCommand.ExecuteReader();

while(dataReader.Read())

{

taiKhoans.Add(new TaiKhoan(dataReader.GetString(0), dataReader.GetString(1)));

}

sqlConnection.Close();

}

return taiKhoans;

}

public void Command(string query)

{

using (SqlConnection sqlConnection = Connection.GetSqlConnection())

{

sqlConnection.Open();

sqlCommand = new SqlCommand(query, sqlConnection);

sqlCommand.ExecuteNonQuery();

sqlConnection.Close();

}

}

}

}

***5.2.4. Form1.cs***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace QuanLyDT

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

Class.Functions.Connect(); //Mở kết nối

}

private Form currentFormChild;

private void OpenChildForm(Form childForm)

{

if(currentFormChild!=null)

{

currentFormChild.Close();

}

currentFormChild = childForm;

childForm.TopLevel = false;

childForm.FormBorderStyle = FormBorderStyle.None;

childForm.Dock = DockStyle.Fill;

panel\_Body.Controls.Add(childForm);

panel\_Body.Tag = childForm;

childForm.BringToFront();

childForm.Show();

}

private void button1\_Click(object sender, EventArgs e)

{

OpenChildForm(new DMChatLieu());

label1.Text = button1.Text;

}

private void button2\_Click(object sender, EventArgs e)

{

OpenChildForm(new DMHang());

label1.Text = button2.Text;

}

private void button3\_Click(object sender, EventArgs e)

{

OpenChildForm(new DMNhanVien());

label1.Text = button3.Text;

}

private void button4\_Click(object sender, EventArgs e)

{

OpenChildForm(new KhachHang());

label1.Text = button4.Text;

}

private void button5\_Click(object sender, EventArgs e)

{

OpenChildForm(new HoaDonBan());

label1.Text = button5.Text;

}

private void button7\_Click(object sender, EventArgs e)

{

OpenChildForm(new TimHDBan());

label1.Text = button7.Text;

}

private void button8\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Bạn có muốn đăng xuất không?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Information) == DialogResult.Yes)

{

this.Hide();

DangNhap dangNhap = new DangNhap();

dangNhap.ShowDialog();

}

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

if (currentFormChild != null)

{

currentFormChild.Close();

}

label1.Text = "QUẢN LÝ CỬA HÀNG ĐIỆN THOẠI 24H";

}

}

}

***5.2.5. KhachHang.cs***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using QuanLyDT.Class;

namespace QuanLyDT

{

public partial class KhachHang : Form

{

DataTable tblKH; //Bảng khách hàng

public KhachHang()

{

InitializeComponent();

}

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

{

txtMaKhach.Enabled = false;

btnLuu.Enabled = false;

btnBoQua.Enabled = false;

LoadDataGridView();

}

private void LoadDataGridView()

{

string sql;

sql = "SELECT \* from KhachHang";

tblKH = Functions.GetDataToTable(sql); //Lấy dữ liệu từ bảng

dgvKhachHang.DataSource = tblKH; //Hiển thị vào dataGridView

dgvKhachHang.Columns[0].HeaderText = "Mã khách hàng";

dgvKhachHang.Columns[1].HeaderText = "Tên khách hàng";

dgvKhachHang.Columns[2].HeaderText = "Địa chỉ";

dgvKhachHang.Columns[3].HeaderText = "Điện thoại";

dgvKhachHang.Columns[0].Width = 100;

dgvKhachHang.Columns[1].Width = 150;

dgvKhachHang.Columns[2].Width = 150;

dgvKhachHang.Columns[3].Width = 150;

dgvKhachHang.AllowUserToAddRows = false;

dgvKhachHang.EditMode = DataGridViewEditMode.EditProgrammatically;

}

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

{

if (btnThem.Enabled == false)

{

MessageBox.Show("Đang ở chế độ thêm mới!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtMaKhach.Focus();

return;

}

if (tblKH.Rows.Count == 0)

{

MessageBox.Show("Không có dữ liệu!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

txtMaKhach.Text = dgvKhachHang.CurrentRow.Cells["MaKhach"].Value.ToString();

txtTenKhach.Text = dgvKhachHang.CurrentRow.Cells["TenKhach"].Value.ToString();

txtDiaChi.Text = dgvKhachHang.CurrentRow.Cells["DiaChi"].Value.ToString();

mtbDienThoai.Text = dgvKhachHang.CurrentRow.Cells["DienThoai"].Value.ToString();

btnSua.Enabled = true;

btnXoa.Enabled = true;

btnBoQua.Enabled = true;

}

private void ResetValues()

{

txtMaKhach.Text = "";

txtTenKhach.Text = "";

txtDiaChi.Text = "";

mtbDienThoai.Text = "";

}

private void txtMaKhach\_KeyUp(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

SendKeys.Send("{TAB}");

}

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

{

btnSua.Enabled = false;

btnXoa.Enabled = false;

btnBoQua.Enabled = true;

btnLuu.Enabled = true;

btnThem.Enabled = false;

ResetValues();

txtMaKhach.Enabled = true;

txtMaKhach.Focus();

}

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

{

string sql;

if (txtMaKhach.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn phải nhập mã khách", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtMaKhach.Focus();

return;

}

if (txtTenKhach.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn phải nhập tên khách", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtTenKhach.Focus();

return;

}

if (txtDiaChi.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn phải nhập địa chỉ", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtDiaChi.Focus();

return;

}

if (mtbDienThoai.Text == "( ) -")

{

MessageBox.Show("Bạn phải nhập điện thoại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

mtbDienThoai.Focus();

return;

}

//Kiểm tra đã tồn tại mã khách chưa

sql = "SELECT MaKhach FROM KhachHang WHERE MaKhach=N'" + txtMaKhach.Text.Trim() + "'";

if (Functions.CheckKey(sql))

{

MessageBox.Show("Mã khách này đã tồn tại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtMaKhach.Focus();

return;

}

//Chèn thêm

sql = "INSERT INTO KhachHang VALUES (N'" + txtMaKhach.Text.Trim() +

"',N'" + txtTenKhach.Text.Trim() + "',N'" + txtDiaChi.Text.Trim() + "','" + mtbDienThoai.Text + "')";

Functions.RunSQL(sql);

LoadDataGridView();

ResetValues();

btnXoa.Enabled = true;

btnThem.Enabled = true;

btnSua.Enabled = true;

btnBoQua.Enabled = false;

btnLuu.Enabled = false;

txtMaKhach.Enabled = false;

}

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

{

string sql;

if (tblKH.Rows.Count == 0)

{

MessageBox.Show("Không còn dữ liệu", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

if (txtMaKhach.Text == "")

{

MessageBox.Show("Bạn phải chọn bản ghi cần sửa", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

if (txtTenKhach.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn phải nhập tên khách hàng", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtTenKhach.Focus();

return;

}

if (txtDiaChi.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn phải nhập địa chỉ", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtDiaChi.Focus();

return;

}

if (mtbDienThoai.Text == "( ) -")

{

MessageBox.Show("Bạn phải nhập điện thoại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

mtbDienThoai.Focus();

return;

}

sql = "UPDATE KhachHang SET TenKhach=N'" + txtTenKhach.Text.Trim().ToString() + "',DiaChi=N'" +

txtDiaChi.Text.Trim().ToString() + "',DienThoai='" + mtbDienThoai.Text.ToString() +

"' WHERE MaKhach=N'" + txtMaKhach.Text + "'";

Functions.RunSQL(sql);

LoadDataGridView();

ResetValues();

btnBoQua.Enabled = false;

}

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

{

string sql;

if (tblKH.Rows.Count == 0)

{

MessageBox.Show("Không còn dữ liệu", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

if (txtMaKhach.Text.Trim() == "")

{

MessageBox.Show("Bạn chưa chọn bản ghi nào", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

if (MessageBox.Show("Bạn có muốn xoá bản ghi này không?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

{

sql = "DELETE KhachHang WHERE MaKhach=N'" + txtMaKhach.Text + "'";

Functions.RunSqlDel(sql);

LoadDataGridView();

ResetValues();

}

}

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

{

ResetValues();

btnBoQua.Enabled = false;

btnThem.Enabled = true;

btnXoa.Enabled = true;

btnSua.Enabled = true;

btnLuu.Enabled = false;

txtMaKhach.Enabled = false;

}

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

{

this.Close();

}

}

}

***5.2.6. Hang.cs***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using QuanLyDT.Class;

namespace QuanLyDT

{

public partial class DMHang : Form

{

DataTable tblH; //Bảng hàng

public DMHang()

{

InitializeComponent();

}

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

{

string sql;

sql = "SELECT \* from ChatLieu";

txtMaHang.Enabled = false;

btnLuu.Enabled = false;

btnBoQua.Enabled = false;

LoadDataGridView();

Functions.FillCombo(sql, cboMaChatLieu, "MaChatLieu", "TenChatLieu");

cboMaChatLieu.SelectedIndex = -1;

ResetValues();

}

private void ResetValues()

{

txtMaHang.Text = "";

txtTenHang.Text = "";

cboMaChatLieu.Text = "";

txtSoLuong.Text = "0";

txtDonGiaNhap.Text = "0";

txtDonGiaBan.Text = "0";

txtSoLuong.Enabled = true;

txtDonGiaNhap.Enabled = false;

txtDonGiaBan.Enabled = false;

txtAnh.Text = "";

picAnh.Image = null;

txtGhiChu.Text = "";

}

private void LoadDataGridView()

{

string sql;

sql = "SELECT \* from Hang";

tblH = Functions.GetDataToTable(sql);

dgvHang.DataSource = tblH;

dgvHang.Columns[0].HeaderText = "Mã điện thoại";

dgvHang.Columns[1].HeaderText = "Tên điện thoại";

dgvHang.Columns[2].HeaderText = "Chất Hãng";

dgvHang.Columns[3].HeaderText = "Số lượng";

dgvHang.Columns[4].HeaderText = "Đơn giá nhập";

dgvHang.Columns[5].HeaderText = "Đơn giá bán";

dgvHang.Columns[6].HeaderText = "Ảnh";

dgvHang.Columns[7].HeaderText = "Ghi chú";

dgvHang.Columns[0].Width = 80;

dgvHang.Columns[1].Width = 140;

dgvHang.Columns[2].Width = 80;

dgvHang.Columns[3].Width = 80;

dgvHang.Columns[4].Width = 100;

dgvHang.Columns[5].Width = 100;

dgvHang.Columns[6].Width = 200;

dgvHang.Columns[7].Width = 300;

dgvHang.AllowUserToAddRows = false;

dgvHang.EditMode = DataGridViewEditMode.EditProgrammatically;

}

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

{

string MaChatLieu;

string sql;

if (btnThem.Enabled == false)

{

MessageBox.Show("Đang ở chế độ thêm mới!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtMaHang.Focus();

return;

}

if (tblH.Rows.Count == 0)

{

MessageBox.Show("Không có dữ liệu!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

txtMaHang.Text = dgvHang.CurrentRow.Cells["MaHang"].Value.ToString();

txtTenHang.Text = dgvHang.CurrentRow.Cells["TenHang"].Value.ToString();

MaChatLieu = dgvHang.CurrentRow.Cells["MaChatLieu"].Value.ToString();

sql = "SELECT TenChatLieu FROM ChatLieu WHERE MaChatLieu=N'" + MaChatLieu + "'";

cboMaChatLieu.Text = Functions.GetFieldValues(sql);

txtSoLuong.Text = dgvHang.CurrentRow.Cells["SoLuong"].Value.ToString();

txtDonGiaNhap.Text = dgvHang.CurrentRow.Cells["DonGiaNhap"].Value.ToString();

txtDonGiaBan.Text = dgvHang.CurrentRow.Cells["DonGiaBan"].Value.ToString();

sql = "SELECT Anh FROM Hang WHERE MaHang=N'" + txtMaHang.Text + "'";

txtAnh.Text = Functions.GetFieldValues(sql);

picAnh.Image = Image.FromFile(txtAnh.Text);

sql = "SELECT Ghichu FROM Hang WHERE MaHang = N'" + txtMaHang.Text + "'";

txtGhiChu.Text = Functions.GetFieldValues(sql);

btnSua.Enabled = true;

btnXoa.Enabled = true;

btnBoQua.Enabled = true;

}

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

{

OpenFileDialog dlgOpen = new OpenFileDialog();

dlgOpen.Filter = "Bitmap(\*.bmp)|\*.bmp|JPEG(\*.jpg)|\*.jpg|GIF(\*.gif)|\*.gif|All files(\*.\*)|\*.\*";

dlgOpen.FilterIndex = 2;

dlgOpen.Title = "Chọn ảnh minh hoạ cho sản phẩm";

if (dlgOpen.ShowDialog() == DialogResult.OK)

{

picAnh.Image = Image.FromFile(dlgOpen.FileName);

txtAnh.Text = dlgOpen.FileName;

}

}

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

{

btnSua.Enabled = false;

btnXoa.Enabled = false;

btnBoQua.Enabled = true;

btnLuu.Enabled = true;

btnThem.Enabled = false;

ResetValues();

txtMaHang.Enabled = true;

txtMaHang.Focus();

txtSoLuong.Enabled = true;

txtDonGiaNhap.Enabled = true;

txtDonGiaBan.Enabled = true;

}

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

{

string sql;

if (txtMaHang.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn phải nhập mã điện thoại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtMaHang.Focus();

return;

}

if (txtTenHang.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn phải nhập tên điện thoại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtTenHang.Focus();

return;

}

if (cboMaChatLieu.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn phải nhập hãng điện thoại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

cboMaChatLieu.Focus();

return;

}

if (txtAnh.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn phải chọn ảnh minh hoạ cho hàng", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

btnOpen.Focus();

return;

}

sql = "SELECT MaHang FROM Hang WHERE MaHang=N'" + txtMaHang.Text.Trim() + "'";

if (Functions.CheckKey(sql))

{

MessageBox.Show("Mã hàng này đã tồn tại, bạn phải chọn mã hàng khác", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtMaHang.Focus();

return;

}

sql = "INSERT INTO Hang(MaHang,TenHang,MaChatLieu,SoLuong,DonGiaNhap, DonGiaBan,Anh,Ghichu) VALUES(N'"

+ txtMaHang.Text.Trim() + "',N'" + txtTenHang.Text.Trim() +

"',N'" + cboMaChatLieu.SelectedValue.ToString() +

"'," + txtSoLuong.Text.Trim() + "," + txtDonGiaNhap.Text +

"," + txtDonGiaBan.Text + ",'" + txtAnh.Text + "',N'" + txtGhiChu.Text.Trim() + "')";

Functions.RunSQL(sql);

LoadDataGridView();

//ResetValues();

btnXoa.Enabled = true;

btnThem.Enabled = true;

btnSua.Enabled = true;

btnBoQua.Enabled = false;

btnLuu.Enabled = false;

txtMaHang.Enabled = false;

}

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

{

string sql;

if (tblH.Rows.Count == 0)

{

MessageBox.Show("Không còn dữ liệu", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

if (txtMaHang.Text == "")

{

MessageBox.Show("Bạn chưa chọn bản ghi nào", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtMaHang.Focus();

return;

}

if (txtTenHang.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn phải nhập tên hàng", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtTenHang.Focus();

return;

}

if (cboMaChatLieu.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn phải nhập hãng điện thoại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

cboMaChatLieu.Focus();

return;

}

if (txtAnh.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn phải ảnh minh hoạ cho hàng", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtAnh.Focus();

return;

}

sql = "UPDATE Hang SET TenHang=N'" + txtTenHang.Text.Trim().ToString() +

"',MaChatLieu=N'" + cboMaChatLieu.SelectedValue.ToString() +

"',SoLuong=" + txtSoLuong.Text +

",Anh='" + txtAnh.Text + "',Ghichu=N'" + txtGhiChu.Text + "' WHERE MaHang=N'" + txtMaHang.Text + "'";

Functions.RunSQL(sql);

LoadDataGridView();

ResetValues();

btnBoQua.Enabled = false;

}

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

{

string sql;

if (tblH.Rows.Count == 0)

{

MessageBox.Show("Không còn dữ liệu!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

if (txtMaHang.Text == "")

{

MessageBox.Show("Bạn chưa chọn bản ghi nào", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

if (MessageBox.Show("Bạn có muốn xoá bản ghi này không?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

{

sql = "DELETE Hang WHERE MaHang=N'" + txtMaHang.Text + "'";

Functions.RunSqlDel(sql);

LoadDataGridView();

ResetValues();

}

}

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

{

ResetValues();

btnXoa.Enabled = true;

btnSua.Enabled = true;

btnThem.Enabled = true;

btnBoQua.Enabled = false;

btnLuu.Enabled = false;

txtMaHang.Enabled = false;

}

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

{

string sql;

if ((txtMaHang.Text == "") && (txtTenHang.Text == "") && (cboMaChatLieu.Text == ""))

{

MessageBox.Show("Bạn hãy nhập điều kiện tìm kiếm", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

sql = "SELECT \* from Hang WHERE 1=1";

if (txtMaHang.Text != "")

sql += " AND MaHang LIKE N'%" + txtMaHang.Text + "%'";

if (txtTenHang.Text != "")

sql += " AND TenHang LIKE N'%" + txtTenHang.Text + "%'";

if (cboMaChatLieu.Text != "")

sql += " AND MaChatLieu LIKE N'%" + cboMaChatLieu.SelectedValue + "%'";

tblH = Functions.GetDataToTable(sql);

if (tblH.Rows.Count == 0)

MessageBox.Show("Không có bản ghi thoả mãn điều kiện tìm kiếm!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

else MessageBox.Show("Có " + tblH.Rows.Count + " bản ghi thoả mãn điều kiện!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

dgvHang.DataSource = tblH;

ResetValues();

}

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

{

string sql;

sql = "SELECT MaHang,TenHang,MaChatLieu,SoLuong,DonGiaNhap,DonGiaBan,Anh,Ghichu FROM Hang";

tblH = Functions.GetDataToTable(sql);

dgvHang.DataSource = tblH;

}

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

***5.2.7. HoaDonBan.cs***

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using QuanLyDT.Class;

using COMExcel = Microsoft.Office.Interop.Excel;

namespace QuanLyDT

{

public partial class HoaDonBan : Form

{

DataTable tblCTHDB; //Bảng chi tiết hoá đơn bán

public HoaDonBan()

{

InitializeComponent();

}

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

{

btnThem.Enabled = true;

btnLuu.Enabled = false;

btnXoa.Enabled = false;

btnInHoaDon.Enabled = false;

txtMaHoaDon.ReadOnly = true;

txtTenNhanVien.ReadOnly = true;

txtTenKhachHang.ReadOnly = true;

txtDiaChi.ReadOnly = true;

mtbDienThoai.ReadOnly = true;

txtTenSanPham.ReadOnly = true;

txtDonGia.ReadOnly = true;

txtThanhTien.ReadOnly = true;

txtTongTien.ReadOnly = true;

txtGiamGia.Text = "0";

txtTongTien.Text = "0";

Functions.FillCombo("SELECT MaKhach, TenKhach FROM KhachHang", cboMaKhachHang, "MaKhach", "MaKhach");

cboMaKhachHang.SelectedIndex = -1;

Functions.FillCombo("SELECT MaNhanVien, TenNhanVien FROM NhanVien", cboMaNhanVien, "MaNhanVien", "TenKhach");

cboMaNhanVien.SelectedIndex = -1;

Functions.FillCombo("SELECT MaHang, TenHang FROM Hang", cboMaHang, "MaHang", "MaHang");

cboMaHang.SelectedIndex = -1;

//Hiển thị thông tin của một hóa đơn được gọi từ form tìm kiếm

if (txtMaHoaDon.Text != "")

{

LoadInfoHoaDon();

btnXoa.Enabled = true;

btnInHoaDon.Enabled = true;

}

LoadDataGridView();

}

private void LoadDataGridView()

{

string sql;

sql = "SELECT a.MaHang, b.TenHang, a.SoLuong, b.DonGiaBan, a.GiamGia,a.ThanhTien FROM ChiTietHDBan AS a, Hang AS b WHERE a.MaHDBan = N'" + txtMaHoaDon.Text + "' AND a.MaHang=b.MaHang";

tblCTHDB = Functions.GetDataToTable(sql);

dgvHDBanHang.DataSource = tblCTHDB;

dgvHDBanHang.Columns[0].HeaderText = "Mã sản phẩm";

dgvHDBanHang.Columns[1].HeaderText = "Tên sản phẩm";

dgvHDBanHang.Columns[2].HeaderText = "Số lượng";

dgvHDBanHang.Columns[3].HeaderText = "Đơn giá";

dgvHDBanHang.Columns[4].HeaderText = "Giảm giá %";

dgvHDBanHang.Columns[5].HeaderText = "Thành tiền";

dgvHDBanHang.Columns[0].Width = 80;

dgvHDBanHang.Columns[1].Width = 130;

dgvHDBanHang.Columns[2].Width = 80;

dgvHDBanHang.Columns[3].Width = 90;

dgvHDBanHang.Columns[4].Width = 90;

dgvHDBanHang.Columns[5].Width = 90;

dgvHDBanHang.AllowUserToAddRows = false;

dgvHDBanHang.EditMode = DataGridViewEditMode.EditProgrammatically;

}

private void LoadInfoHoaDon()

{

string str;

str = "SELECT NgayBan FROM HDBan WHERE MaHDBan = N'" + txtMaHoaDon.Text + "'";

dtpNgayBan.Value = DateTime.Parse(Functions.GetFieldValues(str));

str = "SELECT MaNhanVien FROM HDBan WHERE MaHDBan = N'" + txtMaHoaDon.Text + "'";

cboMaNhanVien.Text = Functions.GetFieldValues(str);

str = "SELECT MaKhach FROM HDBan WHERE MaHDBan = N'" + txtMaHoaDon.Text + "'";

cboMaKhachHang.Text = Functions.GetFieldValues(str);

str = "SELECT TongTien FROM HDBan WHERE MaHDBan = N'" + txtMaHoaDon.Text + "'";

txtTongTien.Text = Functions.GetFieldValues(str);

lblBangChu.Text = "Bằng chữ: " + Functions.ChuyenSoSangChuoi(Double.Parse(txtTongTien.Text));

}

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

{

btnXoa.Enabled = false;

btnLuu.Enabled = true;

btnInHoaDon.Enabled = false;

btnThem.Enabled = false;

ResetValues();

txtMaHoaDon.Text = Functions.CreateKey("HDB");

LoadDataGridView();

}

private void ResetValues()

{

txtMaHoaDon.Text = "";

dtpNgayBan.Text = DateTime.Now.ToShortDateString();

cboMaNhanVien.Text = "";

cboMaKhachHang.Text = "";

txtTongTien.Text = "0";

lblBangChu.Text = "Bằng chữ: ";

cboMaHang.Text = "";

txtSoLuong.Text = "";

txtGiamGia.Text = "0";

txtThanhTien.Text = "0";

}

private void ResetValuesHang()

{

cboMaHang.Text = "";

txtSoLuong.Text = "";

txtGiamGia.Text = "0";

txtThanhTien.Text = "0";

}

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

{

string sql;

double sl, SLcon, tong, Tongmoi;

sql = "SELECT MaHDBan FROM HDBan WHERE MaHDBan=N'" + txtMaHoaDon.Text + "'";

if (!Functions.CheckKey(sql))

{

// Mã hóa đơn chưa có, tiến hành lưu các thông tin chung

// Mã HDBan được sinh tự động do đó không có trường hợp trùng khóa

if (dtpNgayBan.Text.Length == 0)

{

MessageBox.Show("Bạn phải nhập ngày bán", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

dtpNgayBan.Focus();

return;

}

if (cboMaNhanVien.Text.Length == 0)

{

MessageBox.Show("Bạn phải nhập nhân viên", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

cboMaNhanVien.Focus();

return;

}

if (cboMaKhachHang.Text.Length == 0)

{

MessageBox.Show("Bạn phải nhập khách hàng", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

cboMaKhachHang.Focus();

return;

}

sql = "INSERT INTO HDBan(MaHDBan, NgayBan, MaNhanVien, MaKhach, TongTien) VALUES (N'" + txtMaHoaDon.Text.Trim() + "','" +

dtpNgayBan.Value + "',N'" + cboMaNhanVien.SelectedValue + "',N'" +

cboMaKhachHang.SelectedValue + "'," + txtTongTien.Text + ")";

Functions.RunSQL(sql);

}

// Lưu thông tin của các mặt hàng

if (cboMaHang.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn phải nhập mã hàng", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

cboMaHang.Focus();

return;

}

if ((txtSoLuong.Text.Trim().Length == 0) || (txtSoLuong.Text == "0"))

{

MessageBox.Show("Bạn phải nhập số lượng", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtSoLuong.Text = "";

txtSoLuong.Focus();

return;

}

if (txtGiamGia.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn phải nhập giảm giá", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtGiamGia.Focus();

return;

}

sql = "SELECT MaHang FROM ChiTietHDBan WHERE MaHang=N'" + cboMaHang.SelectedValue + "' AND MaHDBan = N'" + txtMaHoaDon.Text.Trim() + "'";

if (Functions.CheckKey(sql))

{

MessageBox.Show("Mã hàng này đã có, bạn phải nhập mã khác", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

ResetValuesHang();

cboMaHang.Focus();

return;

}

// Kiểm tra xem số lượng hàng trong kho còn đủ để cung cấp không?

sl = Convert.ToDouble(Functions.GetFieldValues("SELECT SoLuong FROM Hang WHERE MaHang = N'" + cboMaHang.SelectedValue + "'"));

if (Convert.ToDouble(txtSoLuong.Text) > sl)

{

MessageBox.Show("Số lượng mặt hàng này chỉ còn " + sl, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtSoLuong.Text = "";

txtSoLuong.Focus();

return;

}

sql = "INSERT INTO ChiTietHDBan(MaHDBan,MaHang,SoLuong,DonGia, GiamGia,ThanhTien) VALUES(N'" + txtMaHoaDon.Text.Trim() + "',N'" + cboMaHang.SelectedValue + "'," + txtSoLuong.Text + "," + txtDonGia.Text + "," + txtGiamGia.Text + "," + txtThanhTien.Text + ")";

Functions.RunSQL(sql);

LoadDataGridView();

// Cập nhật lại số lượng của mặt hàng vào bảng tblHang

SLcon = sl - Convert.ToDouble(txtSoLuong.Text);

sql = "UPDATE Hang SET SoLuong =" + SLcon + " WHERE MaHang= N'" + cboMaHang.SelectedValue + "'";

Functions.RunSQL(sql);

// Cập nhật lại tổng tiền cho hóa đơn bán

tong = Convert.ToDouble(Functions.GetFieldValues("SELECT TongTien FROM HDBan WHERE MaHDBan = N'" + txtMaHoaDon.Text + "'"));

Tongmoi = tong + Convert.ToDouble(txtThanhTien.Text);

sql = "UPDATE HDBan SET TongTien =" + Tongmoi + " WHERE MaHDBan = N'" + txtMaHoaDon.Text + "'";

Functions.RunSQL(sql);

txtTongTien.Text = Tongmoi.ToString();

lblBangChu.Text = "Bằng chữ: " + Functions.ChuyenSoSangChuoi(Double.Parse(Tongmoi.ToString()));

ResetValuesHang();

btnXoa.Enabled = true;

btnThem.Enabled = true;

btnInHoaDon.Enabled = true;

}

private void dgvHDBanHang\_DoubleClick(object sender, EventArgs e)

{

string MaHangxoa, sql;

Double ThanhTienxoa, SoLuongxoa, sl, slcon, tong, tongmoi;

if (tblCTHDB.Rows.Count == 0)

{

MessageBox.Show("Không có dữ liệu!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

if ((MessageBox.Show("Bạn có chắc chắn muốn xóa không?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes))

{

//Xóa hàng và cập nhật lại số lượng hàng

MaHangxoa = dgvHDBanHang.CurrentRow.Cells["MaHang"].Value.ToString();

SoLuongxoa = Convert.ToDouble(dgvHDBanHang.CurrentRow.Cells["SoLuong"].Value.ToString());

ThanhTienxoa = Convert.ToDouble(dgvHDBanHang.CurrentRow.Cells["ThanhTien"].Value.ToString());

sql = "DELETE ChiTietHDBan WHERE MaHDBan=N'" + txtMaHoaDon.Text + "' AND MaHang = N'" + MaHangxoa + "'";

Functions.RunSQL(sql);

// Cập nhật lại số lượng cho các mặt hàng

sl = Convert.ToDouble(Functions.GetFieldValues("SELECT SoLuong FROM Hang WHERE MaHang = N'" + MaHangxoa + "'"));

slcon = sl + SoLuongxoa;

sql = "UPDATE Hang SET SoLuong =" + slcon + " WHERE MaHang= N'" + MaHangxoa + "'";

Functions.RunSQL(sql);

// Cập nhật lại tổng tiền cho hóa đơn bán

tong = Convert.ToDouble(Functions.GetFieldValues("SELECT TongTien FROM HDBan WHERE MaHDBan = N'" + txtMaHoaDon.Text + "'"));

tongmoi = tong - ThanhTienxoa;

sql = "UPDATE HDBan SET TongTien =" + tongmoi + " WHERE MaHDBan = N'" + txtMaHoaDon.Text + "'";

Functions.RunSQL(sql);

txtTongTien.Text = tongmoi.ToString();

lblBangChu.Text = "Bằng chữ: " + Functions.ChuyenSoSangChuoi(Double.Parse(tongmoi.ToString()));

LoadDataGridView();

}

}

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

{

double sl, slcon, slxoa;

if (MessageBox.Show("Bạn có chắc chắn muốn xóa không?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

{

string sql = "SELECT MaHang,SoLuong FROM ChiTietHDBan WHERE MaHDBan = N'" + txtMaHoaDon.Text + "'";

DataTable Hang = Functions.GetDataToTable(sql);

for (int hang = 0; hang <= Hang.Rows.Count - 1; hang++)

{

// Cập nhật lại số lượng cho các mặt hàng

sl = Convert.ToDouble(Functions.GetFieldValues("SELECT SoLuong FROM Hang WHERE MaHang = N'" + Hang.Rows[hang][0].ToString() + "'"));

slxoa = Convert.ToDouble(Hang.Rows[hang][1].ToString());

slcon = sl + slxoa;

sql = "UPDATE Hang SET SoLuong =" + slcon + " WHERE MaHang= N'" + Hang.Rows[hang][0].ToString() + "'";

Functions.RunSQL(sql);

}

//Xóa chi tiết hóa đơn

sql = "DELETE ChiTietHDBan WHERE MaHDBan=N'" + txtMaHoaDon.Text + "'";

Functions.RunSqlDel(sql);

//Xóa hóa đơn

sql = "DELETE HDBan WHERE MaHDBan=N'" + txtMaHoaDon.Text + "'";

Functions.RunSqlDel(sql);

ResetValues();

LoadDataGridView();

btnXoa.Enabled = false;

btnInHoaDon.Enabled = false;

}

}

private void cboMaNhanVien\_TextChanged(object sender, EventArgs e)

{

string str;

if (cboMaNhanVien.Text == "")

txtTenNhanVien.Text = "";

// Khi chọn Mã nhân viên thì tên nhân viên tự động hiện ra

str = "Select TenNhanVien from NhanVien where MaNhanVien =N'" + cboMaNhanVien.SelectedValue + "'";

txtTenNhanVien.Text = Functions.GetFieldValues(str);

}

private void cboMaKhachHang\_TextChanged(object sender, EventArgs e)

{

string str;

if (cboMaKhachHang.Text == "")

{

txtTenKhachHang.Text = "";

txtDiaChi.Text = "";

mtbDienThoai.Text = "";

}

//Khi chọn Mã khách hàng thì các thông tin của khách hàng sẽ hiện ra

str = "Select TenKhach from KhachHang where MaKhach = N'" + cboMaKhachHang.SelectedValue + "'";

txtTenKhachHang.Text = Functions.GetFieldValues(str);

str = "Select DiaChi from KhachHang where MaKhach = N'" + cboMaKhachHang.SelectedValue + "'";

txtDiaChi.Text = Functions.GetFieldValues(str);

str = "Select DienThoai from KhachHang where MaKhach= N'" + cboMaKhachHang.SelectedValue + "'";

mtbDienThoai.Text = Functions.GetFieldValues(str);

}

private void cboMaHang\_TextChanged(object sender, EventArgs e)

{

string str;

if (cboMaHang.Text == "")

{

txtTenSanPham.Text = "";

txtDonGia.Text = "";

}

// Khi chọn mã hàng thì các thông tin về hàng hiện ra

str = "SELECT TenHang FROM Hang WHERE MaHang =N'" + cboMaHang.SelectedValue + "'";

txtTenSanPham.Text = Functions.GetFieldValues(str);

str = "SELECT DonGiaBan FROM Hang WHERE MaHang =N'" + cboMaHang.SelectedValue + "'";

txtDonGia.Text = Functions.GetFieldValues(str);

}

private void txtSoLuong\_TextChanged(object sender, EventArgs e)

{

//Khi thay đổi số lượng thì thực hiện tính lại thành tiền

double tt, sl, dg, gg;

if (txtSoLuong.Text == "")

sl = 0;

else

sl = Convert.ToDouble(txtSoLuong.Text);

if (txtGiamGia.Text == "")

gg = 0;

else

gg = Convert.ToDouble(txtGiamGia.Text);

if (txtDonGia.Text == "")

dg = 0;

else

dg = Convert.ToDouble(txtDonGia.Text);

tt = sl \* dg - sl \* dg \* gg / 100;

txtThanhTien.Text = tt.ToString();

}

private void txtGiamGia\_TextChanged(object sender, EventArgs e)

{

//Khi thay đổi giảm giá thì tính lại thành tiền

double tt, sl, dg, gg;

if (txtSoLuong.Text == "")

sl = 0;

else

sl = Convert.ToDouble(txtSoLuong.Text);

if (txtGiamGia.Text == "")

gg = 0;

else

gg = Convert.ToDouble(txtGiamGia.Text);

if (txtDonGia.Text == "")

dg = 0;

else

dg = Convert.ToDouble(txtDonGia.Text);

tt = sl \* dg - sl \* dg \* gg / 100;

txtThanhTien.Text = tt.ToString();

}

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

{

// Khởi động chương trình Excel

COMExcel.Application exApp = new COMExcel.Application();

COMExcel.Workbook exBook; //Trong 1 chương trình Excel có nhiều Workbook

COMExcel.Worksheet exSheet; //Trong 1 Workbook có nhiều Worksheet

COMExcel.Range exRange;

string sql;

int hang = 0, cot = 0;

DataTable tblThongtinHD, tblThongtinHang;

exBook = exApp.Workbooks.Add(COMExcel.XlWBATemplate.xlWBATWorksheet);

exSheet = exBook.Worksheets[1];

// Định dạng chung

exRange = exSheet.Cells[1, 1];

exRange.Range["A1:Z300"].Font.Name = "Times new roman"; //Font chữ

exRange.Range["A1:B3"].Font.Size = 10;

exRange.Range["A1:B3"].Font.Bold = true;

exRange.Range["A1:B3"].Font.ColorIndex = 5; //Màu xanh da trời

exRange.Range["A1:A1"].ColumnWidth = 7;

exRange.Range["B1:B1"].ColumnWidth = 15;

exRange.Range["A1:B1"].MergeCells = true;

exRange.Range["A1:B1"].HorizontalAlignment = COMExcel.XlHAlign.xlHAlignCenter;

exRange.Range["A1:B1"].Value = "Shop điện thoại 24H";

exRange.Range["A2:B2"].MergeCells = true;

exRange.Range["A2:B2"].HorizontalAlignment = COMExcel.XlHAlign.xlHAlignCenter;

exRange.Range["A2:B2"].Value = "Chùa Bộc - Hà Nội";

exRange.Range["A3:B3"].MergeCells = true;

exRange.Range["A3:B3"].HorizontalAlignment = COMExcel.XlHAlign.xlHAlignCenter;

exRange.Range["A3:B3"].Value = "Điện thoại: 0384880932";

exRange.Range["C2:E2"].Font.Size = 16;

exRange.Range["C2:E2"].Font.Bold = true;

exRange.Range["C2:E2"].Font.ColorIndex = 3; //Màu đỏ

exRange.Range["C2:E2"].MergeCells = true;

exRange.Range["C2:E2"].HorizontalAlignment = COMExcel.XlHAlign.xlHAlignCenter;

exRange.Range["C2:E2"].Value = "HÓA ĐƠN BÁN";

// Biểu diễn thông tin chung của hóa đơn bán

sql = "SELECT a.MaHDBan, a.NgayBan, a.TongTien, b.TenKhach, b.DiaChi, b.DienThoai, c.TenNhanVien FROM HDBan AS a, KhachHang AS b, NhanVien AS c WHERE a.MaHDBan = N'" + txtMaHoaDon.Text + "' AND a.MaKhach = b.MaKhach AND a.MaNhanVien = c.MaNhanVien";

tblThongtinHD = Functions.GetDataToTable(sql);

exRange.Range["B6:C9"].Font.Size = 12;

exRange.Range["B6:B6"].Value = "Mã hóa đơn:";

exRange.Range["C6:E6"].MergeCells = true;

exRange.Range["C6:E6"].Value = tblThongtinHD.Rows[0][0].ToString();

exRange.Range["B7:B7"].Value = "Khách hàng:";

exRange.Range["C7:E7"].MergeCells = true;

exRange.Range["C7:E7"].Value = tblThongtinHD.Rows[0][3].ToString();

exRange.Range["B8:B8"].Value = "Địa chỉ:";

exRange.Range["C8:E8"].MergeCells = true;

exRange.Range["C8:E8"].Value = tblThongtinHD.Rows[0][4].ToString();

exRange.Range["B9:B9"].Value = "Điện thoại:";

exRange.Range["C9:E9"].MergeCells = true;

exRange.Range["C9:E9"].Value = tblThongtinHD.Rows[0][5].ToString();

//Lấy thông tin các mặt hàng

sql = "SELECT b.TenHang, a.SoLuong, b.DonGiaBan, a.GiamGia, a.ThanhTien " +

"FROM ChiTietHDBan AS a , Hang AS b WHERE a.MaHDBan = N'" +

txtMaHoaDon.Text + "' AND a.MaHang = b.MaHang";

tblThongtinHang = Functions.GetDataToTable(sql);

//Tạo dòng tiêu đề bảng

exRange.Range["A11:F11"].Font.Bold = true;

exRange.Range["A11:F11"].HorizontalAlignment = COMExcel.XlHAlign.xlHAlignCenter;

exRange.Range["C11:F11"].ColumnWidth = 12;

exRange.Range["A11:A11"].Value = "STT";

exRange.Range["B11:B11"].Value = "Tên hàng";

exRange.Range["C11:C11"].Value = "Số lượng";

exRange.Range["D11:D11"].Value = "Đơn giá";

exRange.Range["E11:E11"].Value = "Giảm giá";

exRange.Range["F11:F11"].Value = "Thành tiền";

for (hang = 0; hang < tblThongtinHang.Rows.Count; hang++)

{

//Điền số thứ tự vào cột 1 từ dòng 12

exSheet.Cells[1][hang + 12] = hang + 1;

for (cot = 0; cot < tblThongtinHang.Columns.Count; cot++)

//Điền thông tin hàng từ cột thứ 2, dòng 12

{

exSheet.Cells[cot + 2][hang + 12] = tblThongtinHang.Rows[hang][cot].ToString();

if (cot == 3) exSheet.Cells[cot + 2][hang + 12] = tblThongtinHang.Rows[hang][cot].ToString() + "%";

}

}

exRange = exSheet.Cells[cot][hang + 14];

exRange.Font.Bold = true;

exRange.Value2 = "Tổng tiền:";

exRange = exSheet.Cells[cot + 1][hang + 14];

exRange.Font.Bold = true;

exRange.Value2 = tblThongtinHD.Rows[0][2].ToString();

exRange = exSheet.Cells[1][hang + 15]; //Ô A1

exRange.Range["A1:F1"].MergeCells = true;

exRange.Range["A1:F1"].Font.Bold = true;

exRange.Range["A1:F1"].Font.Italic = true;

exRange.Range["A1:F1"].HorizontalAlignment = COMExcel.XlHAlign.xlHAlignRight;

exRange.Range["A1:F1"].Value = "Bằng chữ: " + Functions.ChuyenSoSangChuoi(Double.Parse(tblThongtinHD.Rows[0][2].ToString()));

exRange = exSheet.Cells[4][hang + 17]; //Ô A1

exRange.Range["A1:C1"].MergeCells = true;

exRange.Range["A1:C1"].Font.Italic = true;

exRange.Range["A1:C1"].HorizontalAlignment = COMExcel.XlHAlign.xlHAlignCenter;

DateTime d = Convert.ToDateTime(tblThongtinHD.Rows[0][1]);

exRange.Range["A1:C1"].Value = "Hà Nội, ngày " + d.Day + " tháng " + d.Month + " năm " + d.Year;

exRange.Range["A2:C2"].MergeCells = true;

exRange.Range["A2:C2"].Font.Italic = true;

exRange.Range["A2:C2"].HorizontalAlignment = COMExcel.XlHAlign.xlHAlignCenter;

exRange.Range["A2:C2"].Value = "Nhân viên bán hàng";

exRange.Range["A6:C6"].MergeCells = true;

exRange.Range["A6:C6"].Font.Italic = true;

exRange.Range["A6:C6"].HorizontalAlignment = COMExcel.XlHAlign.xlHAlignCenter;

exRange.Range["A6:C6"].Value = tblThongtinHD.Rows[0][6];

exSheet.Name = "Hóa đơn nhập";

exApp.Visible = true;

}

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

{

if (cboMaHoaDon.Text == "")

{

MessageBox.Show("Bạn phải chọn một mã hóa đơn để tìm", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

cboMaHoaDon.Focus();

return;

}

txtMaHoaDon.Text = cboMaHoaDon.Text;

LoadInfoHoaDon();

LoadDataGridView();

btnXoa.Enabled = true;

btnLuu.Enabled = true;

btnInHoaDon.Enabled = true;

cboMaHoaDon.SelectedIndex = -1;

}

private void txtSoLuong\_KeyPress(object sender, KeyPressEventArgs e)

{

if (((e.KeyChar >= '0') && (e.KeyChar <= '9')) || (Convert.ToInt32(e.KeyChar) == 8))

e.Handled = false;

else e.Handled = true;

}

private void txtGiamGia\_KeyPress(object sender, KeyPressEventArgs e)

{

if (((e.KeyChar >= '0') && (e.KeyChar <= '9')) || (Convert.ToInt32(e.KeyChar) == 8))

e.Handled = false;

else e.Handled = true;

}

private void cboMaHoaDon\_DropDown(object sender, EventArgs e)

{

Functions.FillCombo("SELECT MaHDBan FROM HDBan", cboMaHoaDon, "MaHDBan", "MaHDBan");

cboMaHoaDon.SelectedIndex = -1;

}

private void HoaDonBan\_FormClosing(object sender, FormClosingEventArgs e)

{

//Xóa dữ liệu trong các điều khiển trước khi đóng Form

ResetValues();

}

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

**5.2.8. NhanVien.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using QuanLyDT.Class;

namespace QuanLyDT

{

public partial class DMNhanVien : Form

{

DataTable tblNV; //Lưu dữ liệu bảng nhân viên

public DMNhanVien()

{

InitializeComponent();

}

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

{

txtMaNhanVien.Enabled = false;

btnLuu.Enabled = false;

btnBoQua.Enabled = false;

LoadDataGridView();

}

public void LoadDataGridView()

{

string sql;

sql = "SELECT MaNhanVien,TenNhanVien,GioiTinh,DiaChi,DienThoai,NgaySinh from NhanVien";

tblNV = Functions.GetDataToTable(sql); //lấy dữ liệu

dgvNhanVien.DataSource = tblNV;

dgvNhanVien.Columns[0].HeaderText = "Mã nhân viên";

dgvNhanVien.Columns[1].HeaderText = "Tên nhân viên";

dgvNhanVien.Columns[2].HeaderText = "Giới tính";

dgvNhanVien.Columns[3].HeaderText = "Địa chỉ";

dgvNhanVien.Columns[4].HeaderText = "Điện thoại";

dgvNhanVien.Columns[5].HeaderText = "Ngày sinh";

dgvNhanVien.Columns[0].Width = 100;

dgvNhanVien.Columns[1].Width = 150;

dgvNhanVien.Columns[2].Width = 100;

dgvNhanVien.Columns[3].Width = 150;

dgvNhanVien.Columns[4].Width = 100;

dgvNhanVien.Columns[5].Width = 100;

dgvNhanVien.AllowUserToAddRows = false;

dgvNhanVien.EditMode = DataGridViewEditMode.EditProgrammatically;

}

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

{

if (btnThem.Enabled == false)

{

MessageBox.Show("Đang ở chế độ thêm mới!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtMaNhanVien.Focus();

return;

}

if (tblNV.Rows.Count == 0)

{

MessageBox.Show("Không có dữ liệu!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

txtMaNhanVien.Text = dgvNhanVien.CurrentRow.Cells["MaNhanVien"].Value.ToString();

txtTenNhanVien.Text = dgvNhanVien.CurrentRow.Cells["TenNhanVien"].Value.ToString();

if (dgvNhanVien.CurrentRow.Cells["GioiTinh"].Value.ToString() == "Nam") chkGioiTinh.Checked = true;

else chkGioiTinh.Checked = false;

txtDiaChi.Text = dgvNhanVien.CurrentRow.Cells["DiaChi"].Value.ToString();

mtbDienThoai.Text = dgvNhanVien.CurrentRow.Cells["DienThoai"].Value.ToString();

dtpNgaySinh.Text = dgvNhanVien.CurrentRow.Cells["NgaySinh"].Value.ToString();

btnSua.Enabled = true;

btnXoa.Enabled = true;

btnXoa.Enabled = true;

}

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

{

btnSua.Enabled = false;

btnXoa.Enabled = false;

btnBoQua.Enabled = true;

btnLuu.Enabled = true;

btnThem.Enabled = false;

ResetValues();

txtMaNhanVien.Enabled = true;

txtMaNhanVien.Focus();

}

private void ResetValues()

{

txtMaNhanVien.Text = "";

txtTenNhanVien.Text = "";

chkGioiTinh.Checked = false;

txtDiaChi.Text = "";

dtpNgaySinh.Text = "";

mtbDienThoai.Text = "";

}

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

{

string sql, gt;

if (txtMaNhanVien.Text.Trim() == "")

{

MessageBox.Show("Bạn phải nhập mã nhân viên", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtMaNhanVien.Focus();

return;

}

if (txtTenNhanVien.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn phải nhập tên nhân viên", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtTenNhanVien.Focus();

return;

}

if (txtDiaChi.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn phải nhập địa chỉ", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtDiaChi.Focus();

return;

}

if (mtbDienThoai.Text == "( ) -")

{

MessageBox.Show("Bạn phải nhập điện thoại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

mtbDienThoai.Focus();

return;

}

if (dtpNgaySinh.Text == " / /")

{

MessageBox.Show("Bạn phải nhập ngày sinh", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

dtpNgaySinh.Focus();

return;

}

if (chkGioiTinh.Checked == true)

gt = "Nam";

else

gt = "Nữ";

sql = "SELECT MaNhanVien FROM NhanVien WHERE MaNhanVien=N'" + txtMaNhanVien.Text.Trim() + "'";

if (Functions.CheckKey(sql))

{

MessageBox.Show("Mã nhân viên này đã có, bạn phải nhập mã khác", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtMaNhanVien.Focus();

txtMaNhanVien.Text = "";

return;

}

sql = "INSERT INTO NhanVien(MaNhanVien,TenNhanVien,GioiTinh, DiaChi,DienThoai, NgaySinh) VALUES (N'" + txtMaNhanVien.Text.Trim() + "',N'" + txtTenNhanVien.Text.Trim() + "',N'" + gt + "',N'" + txtDiaChi.Text.Trim() + "','" + mtbDienThoai.Text + "','" + dtpNgaySinh.Value + "')";

Functions.RunSQL(sql);

LoadDataGridView();

ResetValues();

btnXoa.Enabled = true;

btnThem.Enabled = true;

btnSua.Enabled = true;

btnBoQua.Enabled = false;

btnLuu.Enabled = false;

txtMaNhanVien.Enabled = false;

}

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

{

string sql, gt;

if (tblNV.Rows.Count == 0)

{

MessageBox.Show("Không còn dữ liệu!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

if (txtMaNhanVien.Text == "")

{

MessageBox.Show("Bạn chưa chọn bản ghi nào", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

if (txtTenNhanVien.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn phải nhập tên nhân viên", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtTenNhanVien.Focus();

return;

}

if (txtDiaChi.Text.Trim().Length == 0)

{

MessageBox.Show("Bạn phải nhập địa chỉ", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtDiaChi.Focus();

return;

}

if (mtbDienThoai.Text == "( ) -")

{

MessageBox.Show("Bạn phải nhập điện thoại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

mtbDienThoai.Focus();

return;

}

if (chkGioiTinh.Checked == true)

gt = "Nam";

else

gt = "Nữ";

sql = "UPDATE NhanVien SET TenNhanVien=N'" + txtTenNhanVien.Text.Trim().ToString() +

"',DiaChi=N'" + txtDiaChi.Text.Trim().ToString() +

"',DienThoai='" + mtbDienThoai.Text.ToString() + "',GioiTinh=N'" + gt +

"',NgaySinh='" + dtpNgaySinh.Value +

"' WHERE MaNhanVien=N'" + txtMaNhanVien.Text + "'";

Functions.RunSQL(sql);

LoadDataGridView();

ResetValues();

btnBoQua.Enabled = false;

}

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

{

string sql;

if (tblNV.Rows.Count == 0)

{

MessageBox.Show("Không còn dữ liệu!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

if (txtMaNhanVien.Text == "")

{

MessageBox.Show("Bạn chưa chọn bản ghi nào", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

if (MessageBox.Show("Bạn có muốn xóa không?", "Thông báo", MessageBoxButtons.OKCancel, MessageBoxIcon.Question) == DialogResult.OK)

{

sql = "DELETE NhanVien WHERE MaNhanVien=N'" + txtMaNhanVien.Text + "'";

Functions.RunSqlDel(sql);

LoadDataGridView();

ResetValues();

}

}

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

{

ResetValues();

btnBoQua.Enabled = false;

btnThem.Enabled = true;

btnXoa.Enabled = true;

btnSua.Enabled = true;

btnLuu.Enabled = false;

txtMaNhanVien.Enabled = false;

}

private void txtMaNhanVien\_KeyUp(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

SendKeys.Send("{TAB}");

}

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

{

this.Close();

}

}

}

**5.2.10.DMChatLieu.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient; //Sử dụng thư viện để làm việc SQL server

using QuanLyDT.Class; //Sử dụng class Functions.cs

namespace QuanLyDT

{

public partial class DMChatLieu : Form

{

DataTable tblCL; //Chứa dữ liệu bảng Chất liệu

public DMChatLieu()

{

InitializeComponent();

}

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

{

txtMaChatLieu.Enabled = false;

btnLuu.Enabled = false;

btnBoQua.Enabled = false;

LoadDataGridView(); //Hiển thị bảng tblChatLieu

}

private void LoadDataGridView()

{

string sql;

sql = "SELECT MaChatLieu, TenChatLieu FROM ChatLieu";

tblCL = Functions.GetDataToTable(sql); //Đọc dữ liệu từ bảng

dgvChatLieu.DataSource = tblCL; //Nguồn dữ liệu

dgvChatLieu.Columns[0].HeaderText = "Mã sản phẩm";

dgvChatLieu.Columns[1].HeaderText = "Tên điện thoại";

dgvChatLieu.Columns[0].Width = 100;

dgvChatLieu.Columns[1].Width = 300;

dgvChatLieu.AllowUserToAddRows = false; //Không cho người dùng thêm dữ liệu trực tiếp

dgvChatLieu.EditMode = DataGridViewEditMode.EditProgrammatically; //Không cho sửa dữ liệu trực tiếp

}

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

{

if (btnThem.Enabled == false)

{

MessageBox.Show("Đang ở chế độ thêm mới!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtMaChatLieu.Focus();

return;

}

if (tblCL.Rows.Count == 0) //Nếu không có dữ liệu

{

MessageBox.Show("Không có dữ liệu!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

txtMaChatLieu.Text = dgvChatLieu.CurrentRow.Cells["MaChatLieu"].Value.ToString();

txtTenChatLieu.Text = dgvChatLieu.CurrentRow.Cells["TenChatLieu"].Value.ToString();

btnSua.Enabled = true;

btnXoa.Enabled = true;

btnBoQua.Enabled = true;

}

private void ResetValue()

{

txtMaChatLieu.Text = "";

txtTenChatLieu.Text = "";

}

private void txtMaChatLieu\_KeyUp(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

SendKeys.Send("{TAB}");

}

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

{

btnSua.Enabled = false;

btnXoa.Enabled = false;

btnBoQua.Enabled = true;

btnLuu.Enabled = true;

btnThem.Enabled = false;

ResetValue(); //Xoá trắng các textbox

txtMaChatLieu.Enabled = true; //cho phép nhập mới

txtMaChatLieu.Focus();

}

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

{

string sql; //Lưu lệnh sql

if (txtMaChatLieu.Text.Trim().Length == 0) //Nếu chưa nhập mã chất liệu

{

MessageBox.Show("Bạn phải nhập mã sản phẩm", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtMaChatLieu.Focus();

return;

}

if (txtTenChatLieu.Text.Trim().Length == 0) //Nếu chưa nhập tên chất liệu

{

MessageBox.Show("Bạn phải nhập tên điện thoại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

txtTenChatLieu.Focus();

return;

}

sql = "Select MaChatLieu From ChatLieu where MaChatLieu=N'" + txtMaChatLieu.Text.Trim() + "'";

if (Functions.CheckKey(sql))

{

MessageBox.Show("Mã sản phẩm này đã có, bạn phải nhập mã khác", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtMaChatLieu.Focus();

return;

}

sql = "INSERT INTO ChatLieu VALUES(N'" +

txtMaChatLieu.Text + "',N'" + txtTenChatLieu.Text + "')";

Functions.RunSQL(sql); //Thực hiện câu lệnh sql

LoadDataGridView(); //Nạp lại DataGridView

ResetValue();

btnXoa.Enabled = true;

btnThem.Enabled = true;

btnSua.Enabled = true;

btnBoQua.Enabled = false;

btnLuu.Enabled = false;

txtMaChatLieu.Enabled = false;

}

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

{

string sql; //Lưu câu lệnh sql

if (tblCL.Rows.Count == 0)

{

MessageBox.Show("Không còn dữ liệu", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

if (txtMaChatLieu.Text == "") //nếu chưa chọn bản ghi nào

{

MessageBox.Show("Bạn chưa chọn bản ghi nào", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

if (txtTenChatLieu.Text.Trim().Length == 0) //nếu chưa nhập tên chất liệu

{

MessageBox.Show("Bạn chưa nhập tên điện thoại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

sql = "UPDATE ChatLieu SET TenChatLieu=N'" +

txtTenChatLieu.Text.ToString() +

"' WHERE MaChatLieu=N'" + txtMaChatLieu.Text + "'";

Functions.RunSQL(sql);

LoadDataGridView();

ResetValue();

btnBoQua.Enabled = false;

}

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

{

string sql;

if (tblCL.Rows.Count == 0)

{

MessageBox.Show("Không còn dữ liệu", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

if (txtMaChatLieu.Text == "") //nếu chưa chọn bản ghi nào

{

MessageBox.Show("Bạn chưa chọn bản ghi nào", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

if (MessageBox.Show("Bạn có muốn xoá không?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

{

sql = "DELETE ChatLieu WHERE MaChatLieu=N'" + txtMaChatLieu.Text + "'";

Functions.RunSqlDel(sql);

LoadDataGridView();

ResetValue();

}

}

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

{

ResetValue();

btnBoQua.Enabled = false;

btnThem.Enabled = true;

btnXoa.Enabled = true;

btnSua.Enabled = true;

btnLuu.Enabled = false;

txtMaChatLieu.Enabled = false;

}

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

{

this.Close();

}

}

}

**5.2.11. TimHDBan.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using QuanLyDT.Class;

namespace QuanLyDT

{

public partial class TimHDBan : Form

{

DataTable tblHDB; //Hoá đơn bán

public TimHDBan()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

}

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

{

ResetValues();

dgvTKHoaDon.DataSource = null;

}

private void ResetValues()

{

foreach (Control Ctl in this.Controls)

if (Ctl is TextBox)

Ctl.Text = "";

txtMaHoaDon.Focus();

}

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

{

string sql;

if ((txtMaHoaDon.Text == "") && (txtThang.Text == "") && (txtNam.Text == "") &&

(txtMaNhanVien.Text == "") && (txtMaKhachHang.Text == "") &&

(txtTongTien.Text == ""))

{

MessageBox.Show("Hãy nhập một điều kiện tìm kiếm!!!", "Yêu cầu ...", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

sql = "SELECT \* FROM HDBan WHERE 1=1";

if (txtMaHoaDon.Text != "")

sql = sql + " AND MaHDBan Like N'%" + txtMaHoaDon.Text + "%'";

if (txtThang.Text != "")

sql = sql + " AND MONTH(NgayBan) =" + txtThang.Text;

if (txtNam.Text != "")

sql = sql + " AND YEAR(NgayBan) =" + txtNam.Text;

if (txtMaNhanVien.Text != "")

sql = sql + " AND MaNhanVien Like N'%" + txtMaNhanVien.Text + "%'";

if (txtMaKhachHang.Text != "")

sql = sql + " AND MaKhach Like N'%" + txtMaKhachHang.Text + "%'";

if (txtTongTien.Text != "")

sql = sql + " AND TongTien <=" + txtTongTien.Text;

tblHDB = Functions.GetDataToTable(sql);

if (tblHDB.Rows.Count == 0)

{

MessageBox.Show("Không có bản ghi thỏa mãn điều kiện!!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

MessageBox.Show("Có " + tblHDB.Rows.Count + " bản ghi thỏa mãn điều kiện!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

dgvTKHoaDon.DataSource = tblHDB;

LoadDataGridView();

}

private void LoadDataGridView()

{

dgvTKHoaDon.Columns[0].HeaderText = "Mã HĐB";

dgvTKHoaDon.Columns[1].HeaderText = "Mã nhân viên";

dgvTKHoaDon.Columns[2].HeaderText = "Ngày bán";

dgvTKHoaDon.Columns[3].HeaderText = "Mã khách";

dgvTKHoaDon.Columns[4].HeaderText = "Tổng tiền";

dgvTKHoaDon.Columns[0].Width = 150;

dgvTKHoaDon.Columns[1].Width = 100;

dgvTKHoaDon.Columns[2].Width = 80;

dgvTKHoaDon.Columns[3].Width = 80;

dgvTKHoaDon.Columns[4].Width = 80;

dgvTKHoaDon.AllowUserToAddRows = false;

dgvTKHoaDon.EditMode = DataGridViewEditMode.EditProgrammatically;

}

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

{

ResetValues();

dgvTKHoaDon.DataSource = null;

}

private void txtTongTien\_KeyPress(object sender, KeyPressEventArgs e)

{

if (((e.KeyChar >= '0') && (e.KeyChar <= '9')) || (Convert.ToInt32(e.KeyChar) == 8))

e.Handled = false;

else

e.Handled = true;

}

private void dgvTKHoaDon\_DoubleClick(object sender, EventArgs e)

{

string mahd;

if (MessageBox.Show("Bạn có muốn hiển thị thông tin chi tiết?", "Xác nhận", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

{

mahd = dgvTKHoaDon.CurrentRow.Cells["MaHDBan"].Value.ToString();

HoaDonBan frm = new HoaDonBan();

frm.txtMaHoaDon.Text = mahd;

frm.StartPosition = FormStartPosition.CenterParent;

frm.ShowDialog();

}

}

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

{

this.Close();

}

}

}

**5.2.13. DangKy.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Text.RegularExpressions;

namespace QuanLyDT

{

public partial class DangKy : Form

{

public DangKy()

{

InitializeComponent();

}

public bool checkAccount(string ac)//check mật khẩu và tên tài khoản

{

return Regex.IsMatch(ac, "^[a-zA-Z0-9]{6,24}$");

}

public bool checkEmail(string em)//check Email

{

return Regex.IsMatch(em, @"^[a-zA-Z0-9\_.]{3,20}@gmail.com(.vn|)$");

}

Modify modify = new Modify();

private void button1\_Click(object sender, EventArgs e)

{

string tentk = txtTenTaiKhoan.Text;

string matkhau = txtMatKhau.Text;

string xnmatkhau = txtXNMatKhau.Text;

string email = txtEmail.Text;

if (!checkAccount(tentk)) { MessageBox.Show("Vui lòng nhập tên tài khoản dài 6-24 ký tự, với các ký tự chữ và số, chữ hoa và chữ thường! "); return; }

if (!checkAccount(matkhau)) { MessageBox.Show("Vui lòng nhập mật khẩu dài 6-24 ký tự, với các ký tự chữ và số, chữ hoa và chữ thường! "); return; }

if (xnmatkhau != matkhau) { MessageBox.Show("Vui lòng xác nhận lại mật khẩu chính xác!"); return; }

if (!checkEmail(email)) { MessageBox.Show("Vui lòng nhập đúng định dạng email!"); return; }

if(modify.TaiKhoans("Select \* from TaiKhoan where Email= '" + email + "'").Count != 0) { MessageBox.Show("Email này đã được đăng ký, vui lòng đăng ký email khác!"); return; }

try

{

string query = "Insert into TaiKhoan values ('" + tentk + "','" + matkhau + "','" + email + "')";

modify.Command(query);

if (MessageBox.Show("Đăng ký thành công, bạn có muốn đăng nhập luôn không?","Thông báo",MessageBoxButtons.YesNo,MessageBoxIcon.Information)==DialogResult.Yes)

{

this.Close();

}

}

catch

{

MessageBox.Show("Tên tài khoản này đã được đăng ký, vui lòng đăng ký tên tài khoản khác!");

}

}

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

{

}

}

}

**5.2.14.DangNhap.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace QuanLyDT

{

public partial class DangNhap : Form

{

public DangNhap()

{

InitializeComponent();

}

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

{

}

Modify modify = new Modify();

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

{

string tentk = txtTenTaiKhoan.Text;

string matkhau = txtMatKhau.Text;

if(tentk.Trim() == "") { MessageBox.Show("Vui lòng nhập tên tài khoản!"); }

else if(matkhau.Trim() == "") { MessageBox.Show("Vui lòng nhập mật khẩu!"); }

else

{

string query = "Select \* from TaiKhoan where TaiKhoan= '"+tentk+"'and MatKhau = '"+matkhau+"'";

if(modify.TaiKhoans(query).Count!=0)

{

MessageBox.Show("Đăng nhập thành công!", "Thông báo",MessageBoxButtons.OK,MessageBoxIcon.Information);

this.Hide();

Form1 form1 = new Form1();

form1.ShowDialog();

this.Close();

}

else

{

MessageBox.Show("Tên tài khoản hoặc mật khẩu không chính xác!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

}

private void linkLabel2\_DangKy\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

DangKy dangKy = new DangKy();

dangKy.ShowDialog();

}

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

{

Application.Exit();

}

private void linkLabel1\_QuenMatKhau\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

QuenMatKhau quenMatKhau = new QuenMatKhau();

quenMatKhau.ShowDialog();

}

}

}

**5.2.15.QuenMatKhau.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace QuanLyDT

{

public partial class QuenMatKhau : Form

{

public QuenMatKhau()

{

InitializeComponent();

label3.Text = "";

}

Modify modify = new Modify();

private void button1\_Click(object sender, EventArgs e)

{

string email = textBox1.Text;

if(email.Trim() == "") { MessageBox.Show("Vui lòng nhập email đăng ký!"); }

else

{

string query = "Select \*from TaiKhoan where Email = '" + email + "'";

if(modify.TaiKhoans(query).Count!=0)

{

label3.ForeColor = Color.Blue;

label3.Text = "Mật khẩu: " + modify.TaiKhoans(query)[0].MatKhau;

}

else

{

label3.ForeColor = Color.Red;

label3.Text = "Email này chưa được đăng ký!";

}

}

}

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

{

Application.Exit();

}

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

{

}

}

}

**5.2.16.TaiKhoan.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace QuanLyDT

{

class TaiKhoan

{

private string tenTaiKhoan;

private string matKhau;

public TaiKhoan()

{

}

public TaiKhoan(string tenTaiKhoan, string matKhau)

{

this.tenTaiKhoan = tenTaiKhoan;

this.matKhau = matKhau;

}

public string TenTaiKhoan { get => tenTaiKhoan; set => tenTaiKhoan = value; }

public string MatKhau { get => matKhau; set => matKhau = value; }

}

}

**6.1. Giao diện hệ thống**



