|  |
| --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  **TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**  **KHOA CÔNG NGHỆ THÔNG TIN**    **LẬP TRÌNH CƠ SỞ DỮ LIỆU**  **ĐỀ TÀI: XÂY DỰNG PHẦN MỀM QUẢN LÝ**  **TIỆM GIÀY SNEAKER**  **Thành viên nhóm : Trần Nguyễn Mai Anh - 2054052005**  **Đinh Hoàng Thanh Trúc - 2055012088**  **Trần Hồ Thế Vinh - 2051010362**  **TP Hồ Chí Minh, 04/2023** |

# MỤC LỤC

[**MỤC LỤC 2**](#_j1r96ucnzhl5)

[**CHƯƠNG 1: TỔNG QUAN 3**](#_f5xxgp7e3ftl)

[1. Giới thiệu đề tài 3](#_lxw6t8b70kot)

[2. Lý do chọn đề tài 3](#_pf7p40dffdkt)

[3. Yêu cầu đề tài 4](#_yl4a7rek9ouj)

[**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 4**](#_g70jp7u7chgb)

[I. Giới thiệu về các tool được sử dụng trong bài 4](#_grien19l3k6c)

[1. Guna.UI2.WinForms 4](#_xro2yjljyhh7)

[2. Guna.Charts.WinForms 4](#_nm2c76n8k9rc)

[II. Tìm hiểu về tool được sử dụng trong bài 4](#_aztaeca49elt)

[1. Guna.UI2.WinForms 4](#_6qj6yed8399z)

[2. Guna.Charts.WinForms 5](#_j2qxpky2xsyx)

[III. Cài đặt các tool sử dụng trong bài 5](#_rv50m1mkb80k)

[1. Guna.UI2.WinForms 5](#_rokf4elel51p)

[2. Guna.Charts.WinForms 5](#_okc5l51ezc8g)

[IV. Cấu trúc của tool 5](#_tjznl5481p96)

[V. Database 5](#_qnpswtx4h9pc)

[1. Database 5](#_kzm6bx1egrde)

[2. Lược đồ quan hệ giữa các bảng 6](#_1192pvlp8vpq)

[**CHƯƠNG 3: ĐỀ TÀI ỨNG DỤNG QUẢN LÝ TIỆM GIÀY**](#_gav70h9210jb) **7**

[1. Các bước thực hiện đề tài](#_pb93lrujiite) 7

[2. Mô tả chức năng của đề tài](#_cox7me6zf9or) 7

[3. Mô tả dữ liệu 20](#_7yckg5c5sgz4)

[4. Mô tả sản phẩm](#_76jg0qfldfvv) 23

[**CHƯƠNG 4: KẾT LUẬN**](#_f86k00niunbz) **29**

[1. Kết quả đạt được](#_es4lpaq0bivn) 29

[2. Hạn chế của đề tài](#_kd57mi7sduz9) 29

[3. Phát triển đề tài trong tương lai](#_g85ft3g9hbge) **29**

[**TÀI LIỆU THAM KHẢO**](#_dvi4tbempva2) **29**

# 

# CHƯƠNG 1: TỔNG QUAN

## 1. Giới thiệu đề tài

Trong quá trình học tập và nghiên cứu tìm hiểu, nhóm chúng em đã quyết định chọn đề tài "Xây dựng phần mềm quản lý tiệm giày Sneaker" sử dụng ngôn ngữ lập trình C# trên môi trường .NET Framework, kết hợp với Microsoft SQL Server làm cơ sở dữ liệu và được thiết kế dựa trên mô hình 3 lớp - một mô hình quản lý dữ liệu phổ biến trong lập trình ứng dụng.

Với quy mô ứng dụng và mức độ phức tạp của dự án, việc áp dụng công nghệ thông tin vào quản lý giúp giảm thiểu gánh nặng về sổ sách và trí óc con người. Mục tiêu của đồ án này là xây dựng một phần mềm quản lý bán hàng áp dụng cho các công ty hay cửa hàng về mặt hàng giày sneaker, giúp dễ dàng hơn trong công việc quản lý hàng hóa của công ty hay cửa hàng.

Phần mềm bán giày sneaker được thiết kế với các chức năng chính như: quản lý sản phẩm, quản lý đơn hàng, quản lý khách hàng, quản lý nhân viên, báo cáo thống kê doanh thu, v.v. Ngoài ra, phần mềm còn được tối ưu hóa với giao diện đẹp và dễ sử dụng, giúp người dùng có thể dễ dàng tìm kiếm và mua sắm sản phẩm.

Tuy nhiên, do thời gian có hạn nên bài làm còn gặp phải một số hạn chế và việc trình bày bằng văn bản còn thiếu sót. Rất mong nhận được sự thông cảm và đóng góp ý kiến của Thầy.

## 2. Lý do chọn đề tài

* Quản lý bán hàng trong các doanh nghiệp là công việc quan trọng, đòi hỏi bộ phận quản lý phải tiến hành nhiều nghiệp vụ phức tạp.
* Trong thời đại công nghệ phát triển như hiện nay, việc sử dụng phần mềm quản lý bán hàng là rất cần thiết để giúp cho các doanh nghiệp, cửa hàng có thể quản lý hàng hóa, dịch vụ và thanh toán một cách dễ dàng và hiệu quả hơn. Đặc biệt đối với lĩnh vực bán giày sneaker, một sản phẩm được yêu thích và tiêu thụ rộng rãi, việc quản lý hàng hóa và giao dịch trở nên đặc biệt quan trọng.
* Với mục tiêu giải quyết các vấn đề trên, đề tài "Xây dựng phần mềm bán giày sneaker" ra đời. Phần mềm này hướng đến việc cung cấp một công cụ quản lý bán hàng hiệu quả cho các doanh nghiệp và cửa hàng bán giày sneaker, giúp cho việc quản lý hàng hóa và giao dịch trở nên đơn giản và thuận tiện hơn.

## 3. Yêu cầu đề tài

* Cần phải thiết kế và cài đặt một chương trình quản lý có các chức năng sau:
* Quản lý hàng hóa: thêm, sửa, xóa thông tin loại hàng và các mặt hàng.
* Quản lý nhân viên: thêm, sửa, xóa thông tin các nhân viên (chỉ có người quản trị mới có quyền này).
* Quản lý khách hàng: thêm, sửa, xóa thông tin khách hàng.
* Quản lý và sắp xếp thông tin hàng hóa, nhân viên, khách hàng.
* Thống kê hàng hóa và doanh thu theo từng tháng.
* Tự động thông báo trạng thái hết hạn sử dụng của hàng hóa cho nhân viên.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## Giới thiệu về các tool được sử dụng trong bài

### 1. Guna.UI2.WinForms

### 2. Guna.Charts.WinForms

## Tìm hiểu về tool được sử dụng trong bài

### Guna.UI2.WinForms

Guna.UI2 là một framework thiết kế giao diện người dùng (UI) cho các ứng dụng Windows Forms và WPF trong .NET. Nó cung cấp nhiều điều khiển UI tùy chỉnh và dễ sử dụng để tạo giao diện người dùng đẹp mắt, tương tác và dễ sử dụng cho ứng dụng của bạn. Guna.UI2 được phát triển bởi Guna Framework, cung cấp hỗ trợ cho các phiên bản .NET Framework từ 3.5 đến 4.8 và hỗ trợ việc tạo giao diện đồ họa cho các ứng dụng Windows Forms và WPF trên hầu hết các phiên bản của Windows.

### Guna.Charts.WinForms

Guna.Charts.Winforms là một thư viện đồ họa dành cho các ứng dụng Windows Forms và WPF trong .NET, cung cấp cho người dùng các biểu đồ tương tác và đồ họa đẹp mắt. Nó được phát triển bởi Guna Framework và được tích hợp sẵn trong Guna.UI2, cho phép người dùng tạo các biểu đồ như biểu đồ tròn, biểu đồ cột, biểu đồ đường và biểu đồ vùng. Guna.Charts cung cấp các tính năng như tùy chỉnh màu sắc, hiệu ứng động và tích hợp dễ dàng với các ứng dụng của mình.

## Cài đặt các tool sử dụng trong bài

### Guna.UI2.WinForms

* Để cài đặt Guna.UI2, bạn có thể làm theo các bước sau:
* Mở Visual Studio và tạo một dự án mới.
* Nhấp chuột phải vào thư mục dự án trong khung Solution Explorer, chọn "Manage NuGet Packages".
* Tìm kiếm "Guna.UI2" trong trình quản lý gói NuGet và nhấp vào nút "Install".
* Chờ đợi quá trình cài đặt hoàn tất, sau đó bạn có thể bắt đầu sử dụng Guna.UI2 trong dự án của mình.
* Lưu ý: Để sử dụng Guna.UI2 trong dự án của bạn, bạn cần có bản quyền của thư viện. Bạn có thể mua bản quyền trên trang web chính thức của Guna Framework hoặc sử dụng bản dùng thử trong một thời gian giới hạn.

### Guna.Charts.WinForms

* Chúng ta cài đặt tương tự như Guna.UI2

## Cấu trúc của tool

Guna.UI2 và Guna.Charts là hai thư viện khác nhau trong .NET Framework. Guna.UI2 là thư viện thiết kế giao diện người dùng cho các ứng dụng Windows Forms và WPF, còn Guna.Charts là thư viện đồ họa cung cấp các biểu đồ cho các ứng dụng Windows Forms và WPF.

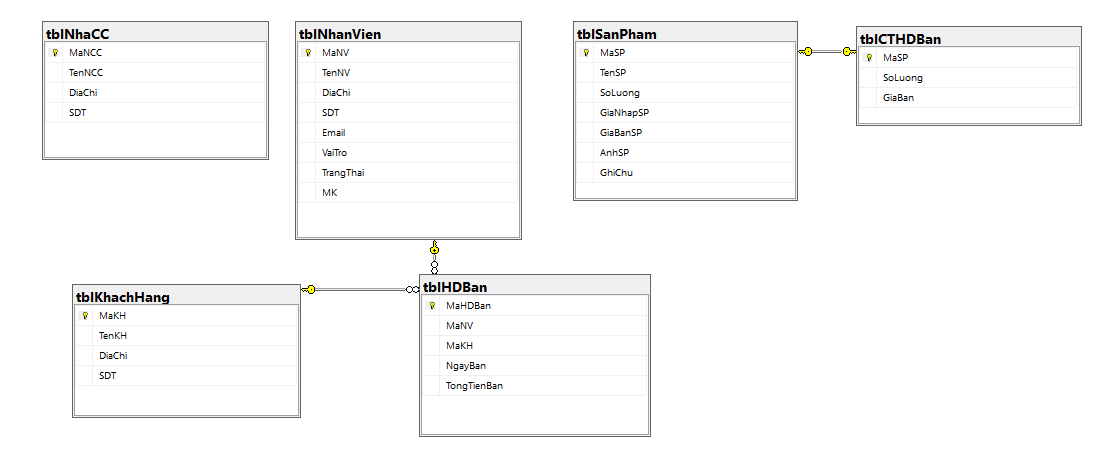
## Database

### 1. Database

* Nhóm đã sử dụng ADO.NET làm công cụ kết nối và thao tác.
* Khai báo và sử dụng biến, cấu trúc điều khiển trong ngôn ngữ T-SQL.
* Tạo và truy vấn dữ liệu.
* Lập trình Stored procedure, Trigger.
* Stored Procedure là một tập các câu lệnh: Thực hiện một nhiệm vụ cụ thể; Được đặt tên và lưu trữ dưới dạng đã biên dịch.
* Trigger là một tập các câu lệnh: Được đặt tên và lưu trữ dưới dạng đã biên dịch, chỉ được vận hành khi có một sự kiện xảy ra.

### 2. Lược đồ quan hệ giữa các bảng

* Lược đồ quan hệ là mối liên hệ giữa các bảng với nhau



# 

# CHƯƠNG 3: ĐỀ TÀI ỨNG DỤNG QUẢN LÝ TIỆM GIÀY

## 1. Các bước thực hiện đề tài

Bước 1: Thảo luận thống nhất chọn đề tài, chọn môi trường để thực hiện đề tài.

Bước 2: Xác định các đối tượng mà ứng dụng làm việc, các chức năng của ứng dụng mà nhóm hướng tới thực hiện.

Bước 3: Thiết kế cơ sở dữ liệu, thiết lập các ràng buộc.

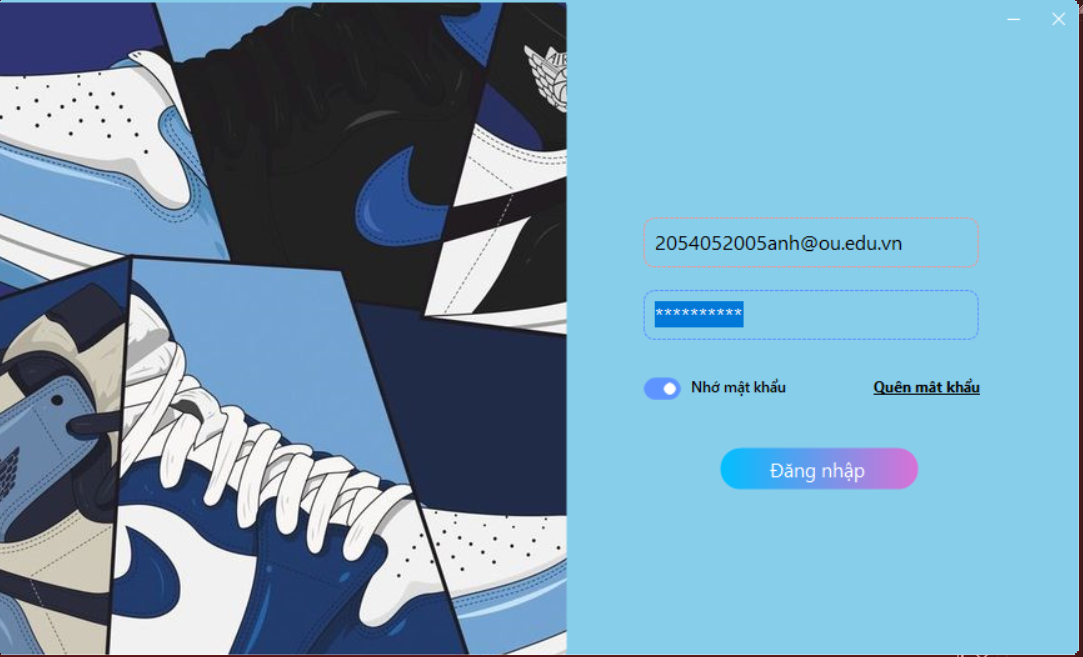
Bước 4: Thiết kế ứng dụng: các Forms, các chức năng của ứng dụng.

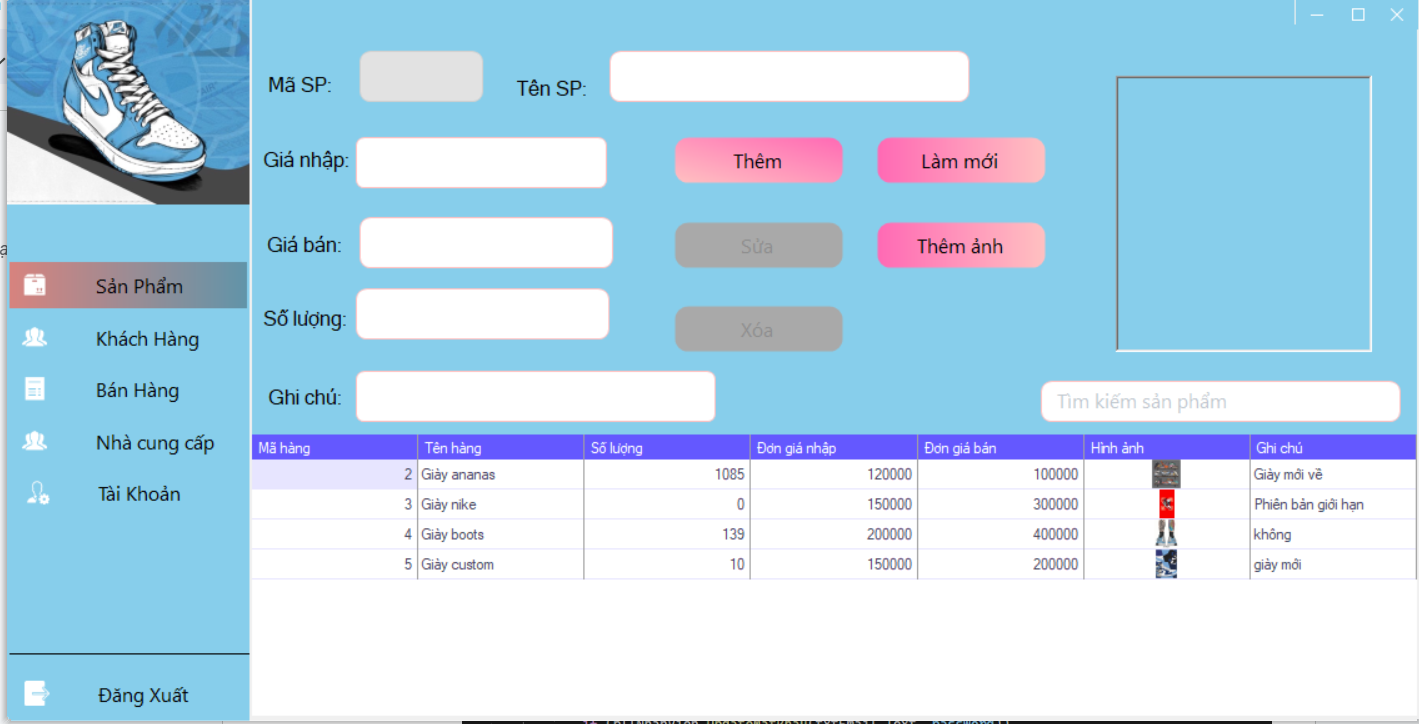
Bước 5: Cài đặt chức năng cho ứng dụng, nhập liệu và kiểm tra lỗi.

Bước 6: Viết báo cáo theo mẫu qui định.

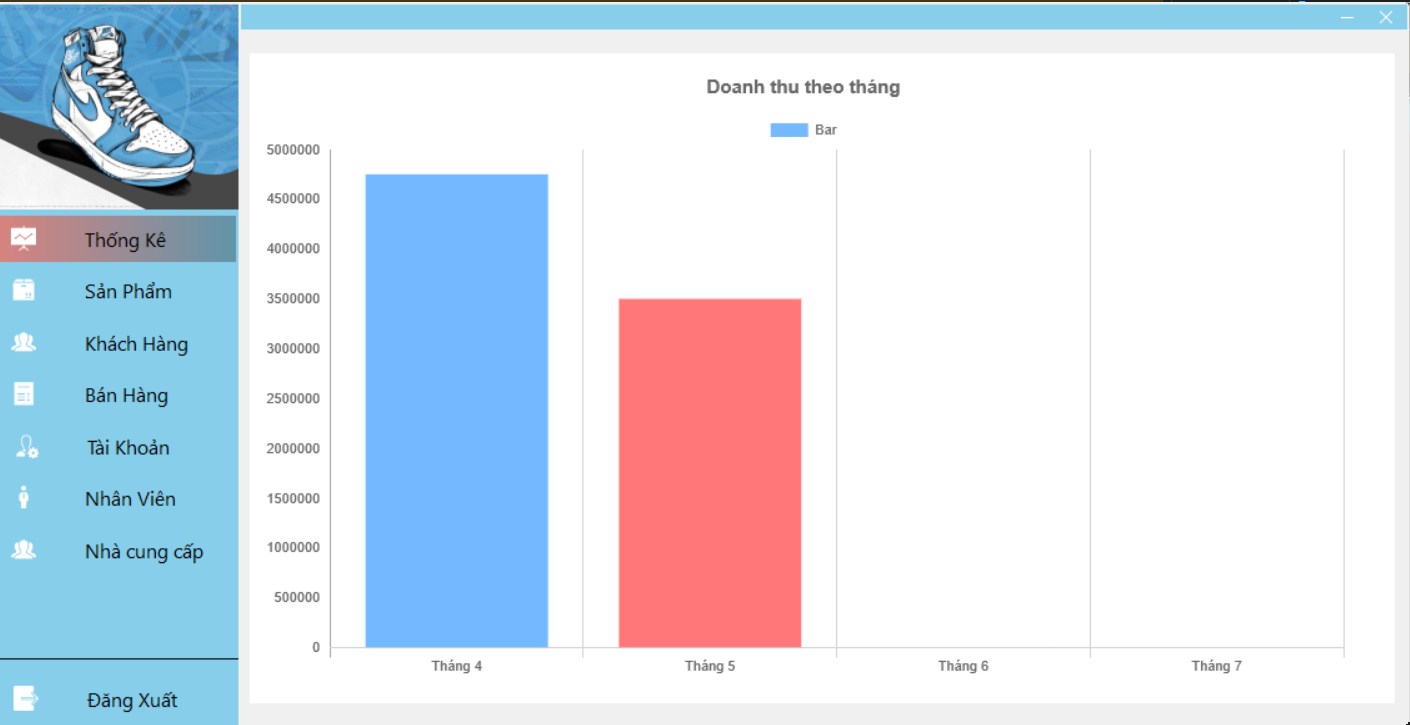
## 

## 2. Mô tả chức năng của đề tài

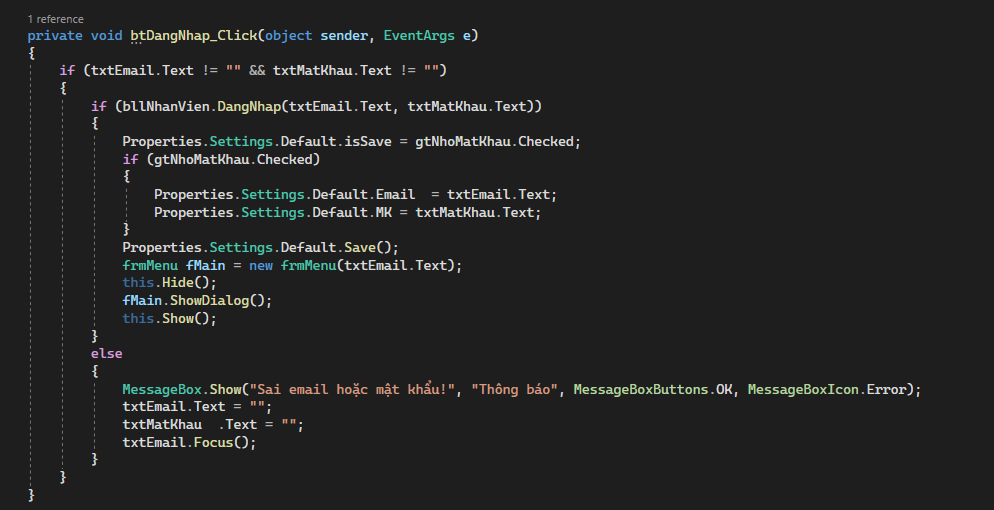
* **Quản lý đăng nhập**
* 
* Đăng nhập bằng quyền của nhân viên:



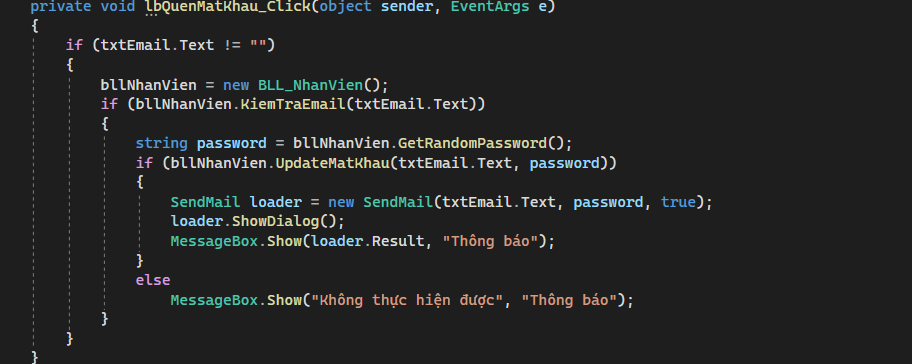
* Đăng nhập bằng quyền của quản lý:



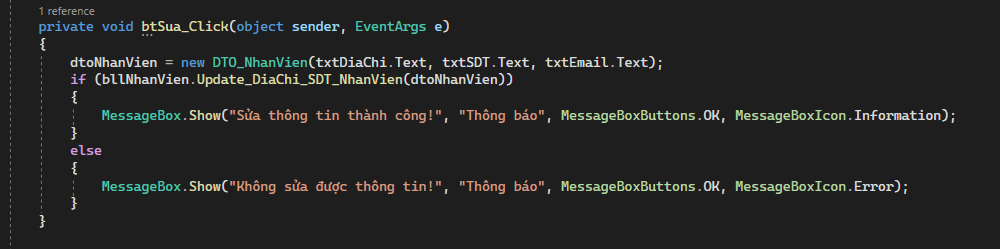
* Chức năng đăng nhập



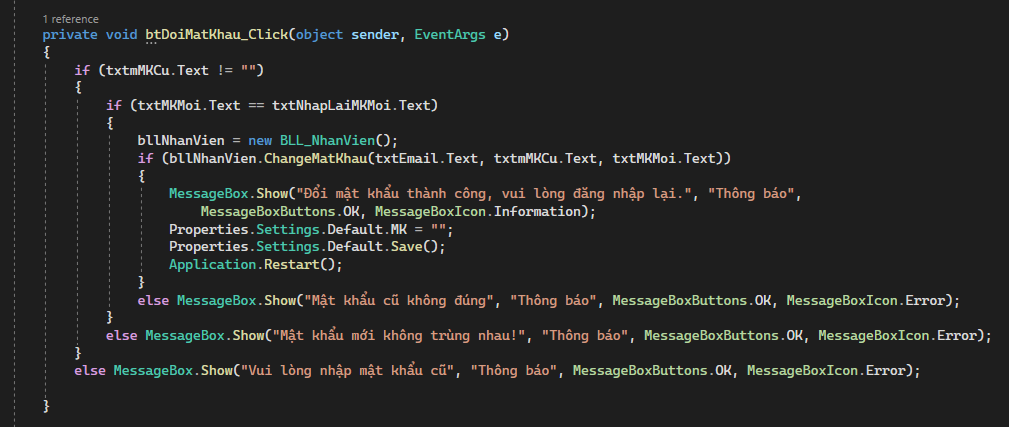
* Đầu tiên kiểm tra xem người dùng đã nhập đầy đủ thông tin email và mật khẩu chưa. Nếu cả hai trường đều được nhập đầy đủ, phương thức sẽ gọi phương thức DangNhap trong đối tượng bllNhanVien để kiểm tra thông tin đăng nhập. Nếu thông tin đăng nhập hợp lệ, phương thức sẽ lưu trạng thái checkbox "Ghi nhớ mật khẩu" vào Properties.Settings.Default.isSave và lưu lại thông tin email và mật khẩu vào Properties.Settings.Default.Email và Properties.Settings.Default.MK (nếu checkbox này được chọn). Sau đó, phương thức sẽ ẩn form hiện tại và hiển thị form frmMenu để người dùng tiếp tục sử dụng chương trình.
* Nếu thông tin đăng nhập không hợp lệ, phương thức sẽ hiển thị một thông báo lỗi và xóa thông tin email và mật khẩu đã nhập trước đó. Sau đó, phương thức sẽ đặt con trỏ vào trường email để người dùng có thể nhập lại thông tin.
* Chức năng quên mật khẩu: Khi người dùng click vào label này, phương thức sẽ kiểm tra xem người dùng đã nhập email vào trường txtEmail hay chưa. Nếu đã nhập, phương thức sẽ kiểm tra xem email này đã được đăng ký trong hệ thống hay chưa. Nếu email đã được đăng ký, phương thức sẽ cập nhật mật khẩu mới cho email này và gửi mật khẩu mới cho người dùng qua email. Nếu gửi email thành công, phương thức sẽ hiển thị một thông báo thành công, ngược lại sẽ hiển thị một thông báo lỗi. Nếu email chưa được đăng ký trong hệ thống hoặc người dùng chưa nhập email vào trường txtEmail, phương thức sẽ không thực hiện gì cả.



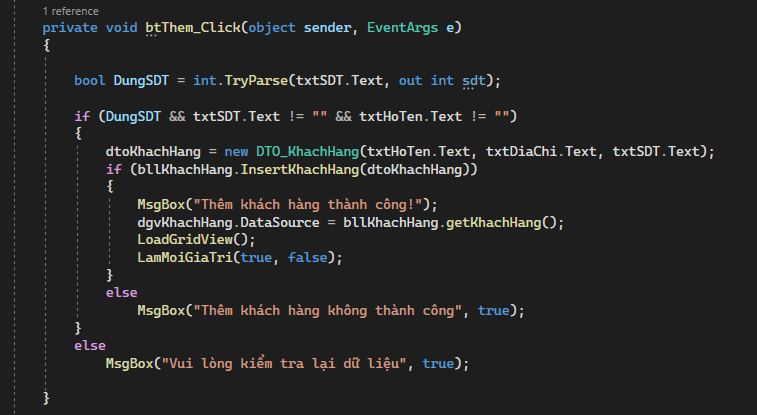
* **Quản lý tài khoản:**
* Chức năng sửa:



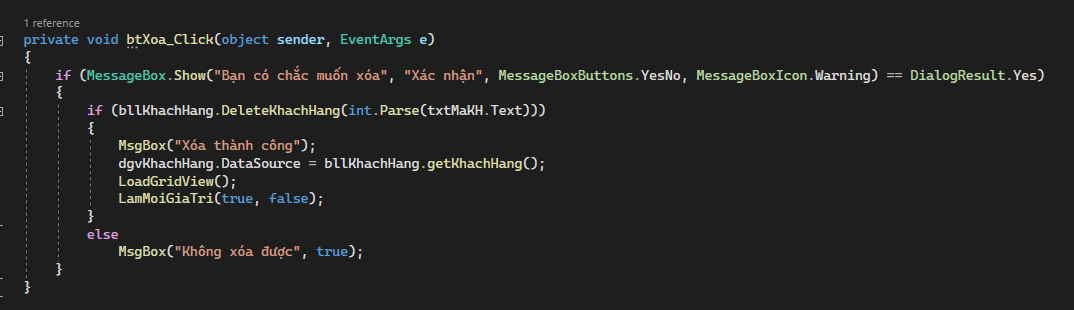
* Khi người dùng click vào nút này, phương thức sẽ tạo một đối tượng DTO\_NhanVien chứa thông tin địa chỉ, số điện thoại và email mà người dùng nhập vào các trường tương ứng trên giao diện.
* Sau đó, phương thức sẽ gọi phương thức Update\_DiaChi\_SDT\_NhanVien của đối tượng BLL\_NhanVien để cập nhật thông tin địa chỉ và số điện thoại của nhân viên trong cơ sở dữ liệu. Nếu cập nhật thành công, phương thức sẽ hiển thị một thông báo thành công, ngược lại sẽ hiển thị một thông báo lỗi.
* Chức năng đổi mật khẩu:

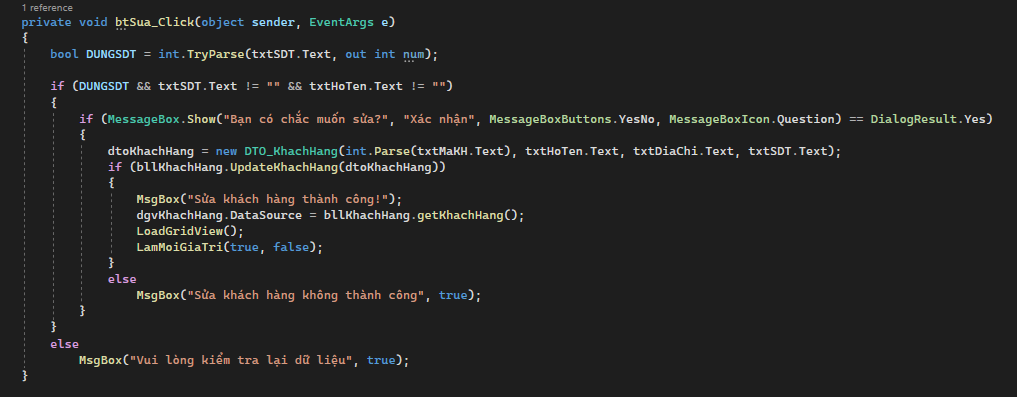


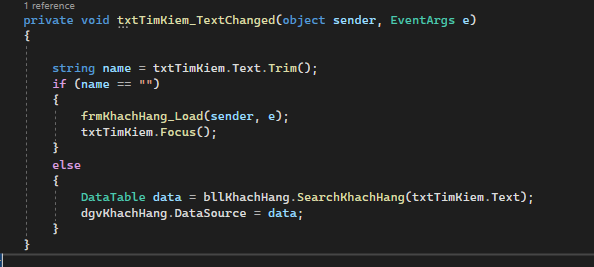
* Nếu người dùng đã nhập đúng mật khẩu cũ và mật khẩu mới được nhập lại khớp, phương thức sẽ gọi phương thức ChangeMatKhau của đối tượng BLL\_NhanVien để thay đổi mật khẩu của người dùng trong cơ sở dữ liệu.
* Nếu thay đổi mật khẩu thành công, phương thức sẽ hiển thị một thông báo thành công và yêu cầu người dùng đăng nhập lại. Sau đó, phương thức sẽ xóa mật khẩu được lưu trong cài đặt của ứng dụng và khởi động lại ứng dụng.
* Nếu có lỗi xảy ra trong quá trình thay đổi mật khẩu, phương thức sẽ hiển thị một thông báo lỗi tương ứng
* **Quản lý khách hàng**:
* Chức năng thêm: Khi người dùng nhấn nút này, đoạn code sẽ kiểm tra xem các trường thông tin "Họ tên", "Số điện thoại" và "Địa chỉ" có được nhập đầy đủ và đúng định dạng hay không. Nếu các trường thông tin này đều hợp lệ, đoạn code sẽ tạo ra một đối tượng DTO\_KhachHang chứa thông tin khách hàng và sử dụng một đối tượng BLL\_KhachHang để thêm khách hàng vào cơ sở dữ liệu. Nếu thêm khách hàng thành công, đoạn code sẽ hiển thị một thông báo thành công và cập nhật lại dữ liệu trên bảng dgvKhachHang. Nếu thêm không thành công, đoạn code sẽ hiển thị một thông báo lỗi. Nếu các trường thông tin không hợp lệ, đoạn code sẽ hiển thị một thông báo yêu cầu kiểm tra lại dữ liệu.



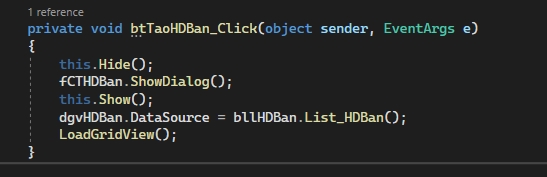
* Chức năng xóa: Kiểm tra xem người dùng đã xác nhận muốn xóa hay chưa bằng hộp thoại xác nhận. Nếu người dùng xác nhận, nó gọi phương thức DeleteKhachHang của đối tượng BLL\_KhachHang với tham số là mã khách hàng cần xóa. Nếu phương thức trả về true, nghĩa là xóa thành công, nó hiển thị một thông báo "Xóa thành công" và cập nhật lại danh sách khách hàng trong DataGridView. Nếu không, nó hiển thị một thông báo lỗi. Sau khi thực hiện xóa, đoạn code này gọi phương thức LamMoiGiaTri để làm mới giá trị các trường dữ liệu trên giao diện.



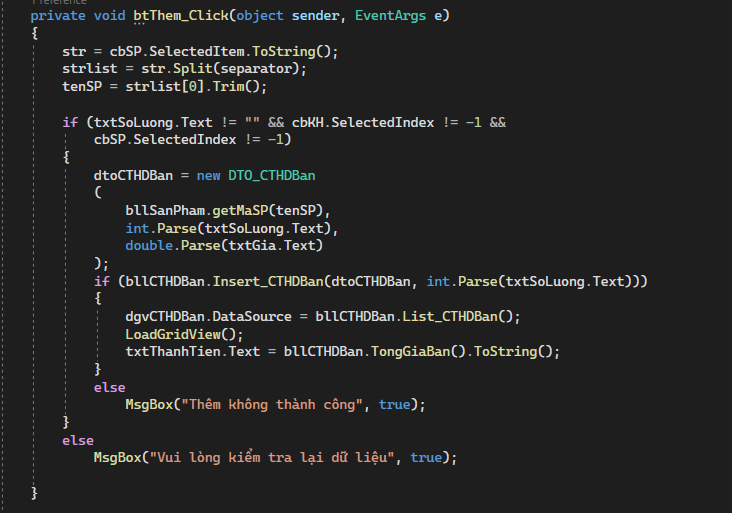
* Chức năng sửa: Kiểm tra dữ liệu đầu vào (tên khách hàng, địa chỉ, số điện thoại) có đúng định dạng hay không. Nếu dữ liệu hợp lệ, chương trình sẽ hiển thị hộp thoại xác nhận sửa thông tin khách hàng. Nếu người dùng xác nhận sửa, chương trình sẽ tạo một đối tượng DTO\_KhachHang mới và gán các giá trị thu thập được từ các trường dữ liệu trên giao diện. Sau đó, chương trình gọi phương thức UpdateKhachHang của lớp BLL\_KhachHang để cập nhật thông tin khách hàng trong cơ sở dữ liệu. Nếu việc cập nhật thành công, chương trình hiển thị thông báo thành công, tải lại danh sách khách hàng và làm mới giá trị trên giao diện để người dùng có thể nhập thông tin khách hàng mới. Nếu không thành công, chương trình hiển thị thông báo lỗi. 
* Chức năng tìm kiếm: Khi người dùng nhập vào từ khóa tìm kiếm, nếu từ khóa rỗng, nghĩa là người dùng chưa nhập gì hoặc đã xóa hết nội dung tìm kiếm, thì chương trình sẽ gọi hàm frmKhachHang\_Load để load lại toàn bộ dữ liệu khách hàng lên gridview và đặt focus vào ô tìm kiếm. Nếu từ khóa tìm kiếm không rỗng, chương trình sẽ gọi phương thức SearchKhachHang của đối tượng BLL\_KhachHang để tìm kiếm các khách hàng có tên hoặc số điện thoại chứa từ khóa tìm kiếm và hiển thị kết quả lên gridview.



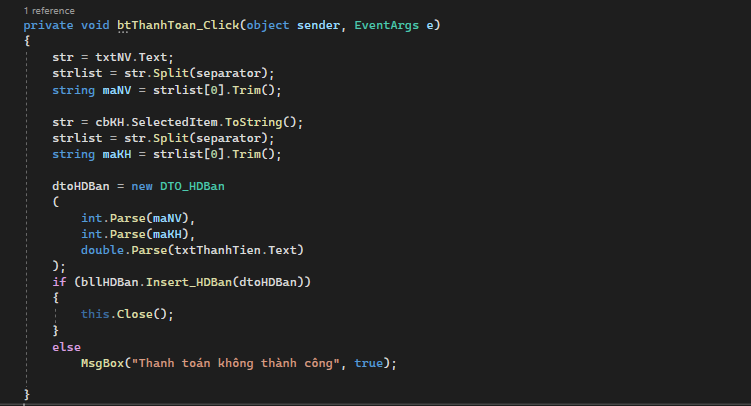
* **Quản lý hóa đơn bán:**
* Chức năng tìm kiếm: Tìm kiếm thông tin các hóa đơn đã bán theo tên khách hàng.
* Chức năng tạo hóa đơn mới: Khi người dùng nhấn nút "Tạo hóa đơn bán". Đầu tiên, nó ẩn form hiện tại, sau đó hiển thị form fCTHDBan để người dùng tạo hóa đơn bán. Khi người dùng hoàn thành tạo hóa đơn bán và đóng form fCTHDBan, form hiện tại sẽ được hiển thị trở lại và lưới dữ liệu dgvHDBan sẽ được cập nhật lại với danh sách các hóa đơn bán mới nhất. Hàm LoadGridView() cũng được gọi để cập nhật lại giao diện người dùng.

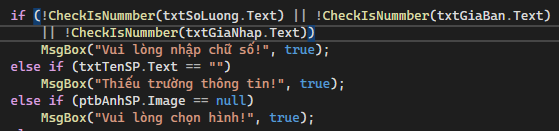


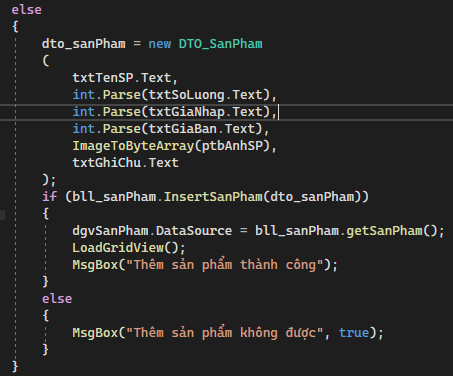
* **Quản lý chi tiết hóa đơn bán:**
* Chức năng thêm: Khi người dùng click vào nút "Thêm", hệ thống sẽ lấy thông tin sản phẩm, số lượng và giá bán được chọn trên giao diện. Sau đó, tạo một đối tượng DTO\_CTHDBan mới với các thông tin sản phẩm, số lượng và giá bán trên, và gọi phương thức Insert\_CTHDBan của lớp BLL\_CTHDBan để thêm chi tiết hóa đơn bán mới vào danh sách. Nếu thao tác thêm thành công, danh sách chi tiết hóa đơn bán trên giao diện sẽ được cập nhật, cùng với giá trị tổng tiền của hóa đơn bán. Nếu thao tác thêm không thành công, một thông báo lỗi sẽ được hiển thị.



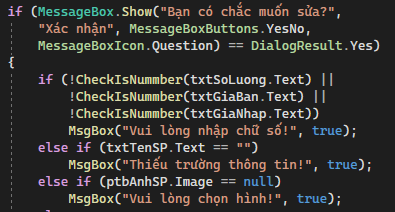
* Chức năng sửa: Khi click vào nút, đoạn code sẽ lấy mã sản phẩm và thông tin sản phẩm được chọn từ các combobox và kiểm tra tính hợp lệ của dữ liệu nhập vào. Sau đó, nó sẽ cập nhật thông tin sản phẩm trong chi tiết hóa đơn bán hàng bằng cách gọi phương thức Update\_SP\_CTHDBan của đối tượng BLL\_CTHDBan và cập nhật lại DataGridView hiển thị thông tin chi tiết hóa đơn. Nếu sửa sản phẩm thành công, chương trình sẽ hiển thị một thông báo thành công, cập nhật tổng giá bán của hóa đơn và đồng thời load lại DataGridView. Ngược lại, nếu không sửa được sản phẩm, chương trình sẽ hiển thị một thông báo lỗi.
* Chức năng xóa: Xóa thông tin hóa đơn đã bán. Nếu người dùng xác nhận muốn xóa sản phẩm, mã sản phẩm sẽ được lấy từ tên sản phẩm, sau đó sẽ gọi hàm Delete\_SP\_CTHDBan để xóa sản phẩm đó khỏi chi tiết hóa đơn bán. Nếu xóa thành công, sẽ cập nhật lại dữ liệu trên form và hiển thị thông báo "Xóa thành công". Nếu không xóa được, sẽ hiển thị thông báo "Không xóa được"
* Chức năng thanh toán: Thanh toán hóa đơn sau khi bán các mặt hàng. Khi người dùng bấm nút "Thanh toán". Đầu tiên, mã nhân viên và mã khách hàng được lấy từ các giá trị đã chọn trên giao diện. Sau đó, tạo đối tượng DTO\_HDBan với các giá trị mã nhân viên, mã khách hàng và tổng giá trị đã chọn. Nếu việc chèn đối tượng này vào cơ sở dữ liệu thành công thì sẽ đóng form hiện tại, ngược lại sẽ hiển thị một thông báo lỗi.



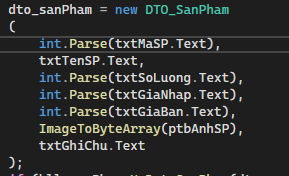
* **Quản lý nhân viên (tương tự chức năng của quản lý khách hàng)**
* Chức năng thêm: Phương thức xử lý sự kiện Click của nút "Thêm " trên giao diện người dùng. Nếu các trường thông tin đã được nhập đầy đủ và định dạng email hợp lệ, phương thức sẽ thêm nhân viên mới vào cơ sở dữ liệu và hiển thị một thông báo thành công. Sau đó, phương thức sẽ gửi một email chứa mật khẩu tạm thời cho nhân viên mới và hiển thị một hộp thoại thông báo với kết quả gửi email. Nếu không thêm nhân viên thành công hoặc định dạng email không hợp lệ, phương thức sẽ hiển thị một thông báo lỗi tương ứng.
* Chức năng sửa:
* Khi người dùng click vào nút “Sửa” này, phương thức sẽ kiểm tra xem các trường thông tin như địa chỉ, email, tên nhân viên và số điện thoại đã được nhập đầy đủ hay chưa.
* Nếu đã nhập đầy đủ, phương thức sẽ tạo một đối tượng DTO\_NhanVien chứa các thông tin được nhập và gọi phương thức UpdateNhanVien của đối tượng BLL\_NhanVien để cập nhật thông tin nhân viên trong cơ sở dữ liệu. Nếu cập nhật thành công, phương thức sẽ hiển thị một thông báo thành công và cập nhật lại bảng dữ liệu nhân viên trên giao diện người dùng.
* Nếu không cập nhật thành công hoặc các trường thông tin không được nhập đầy đủ, phương thức sẽ hiển thị một thông báo lỗi tương ứng.
* Chức năng xóa:
* Khi người dùng click vào nút “Xóa”, phương thức sẽ lấy ra mã nhân viên đang được chọn trên bảng dữ liệu (dgvNhanVien) và gọi phương thức DeleteNhanVien của đối tượng BLL\_NhanVien để xóa nhân viên tương ứng trong cơ sở dữ liệu.
* Nếu xóa thành công, phương thức sẽ hiển thị một thông báo thành công và cập nhật lại bảng dữ liệu nhân viên trên giao diện người dùng. Nếu xóa không thành công, phương thức sẽ hiển thị một thông báo lỗi tương ứng.
* Chức năng tìm kiếm:
* Khi người dùng nhập hoặc thay đổi nội dung của TextBox này, phương thức sẽ lấy ra chuỗi tìm kiếm (tenNV) từ TextBox và kiểm tra xem chuỗi này có rỗng hay không. Nếu chuỗi tìm kiếm rỗng, phương thức sẽ gọi lại phương thức frmNhanVien\_Load để tải lại toàn bộ danh sách nhân viên và đặt focus vào TextBox để người dùng nhập lại nếu cần.
* Nếu chuỗi tìm kiếm không rỗng, phương thức sẽ gọi phương thức SearchNhanVien của đối tượng BLL\_NhanVien để tìm kiếm nhân viên theo tên (hoặc một phần tên) và trả về một DataTable chứa kết quả tìm kiếm. Sau đó, phương thức sẽ cập nhật lại bảng dữ liệu nhân viên trên giao diện người dùng với kết quả tìm kiếm này.
* **Quản lý sản phẩm**
* Chức năng thêm: Thêm một sản phẩm mới vào danh sách.
* Sau khi người dùng nhập đầy đủ các thông tin lên textbox gồm tên, giá nhập, giá bán, số lượng, ghi chú (có thể không ghi chú) và thêm hình ảnh, sau khi nhập các thông tin vào ô textbox, người nhập chọn “Thêm ảnh” để lựa chọn hình ảnh, nếu không có hình ảnh mà bấm “Thêm” sẽ nhận được thông báo “Vui lòng chọn hình.”.
* Sau khi có đủ các thông tin và các thông tin được nhập đúng kiểu dữ liệu, dữ liệu sẽ được chuyển đến lớp DTO\_SanPham để tạo dto, tạo dto thành công sẽ chuyển sang lớp bll có liên kết với lớp dal để sử dụng StoredProcedure “InsertSanPham” thêm mới một sản phẩm. Thêm thành công sẽ nhận được thông báo “Thêm sản phẩm thành công”.



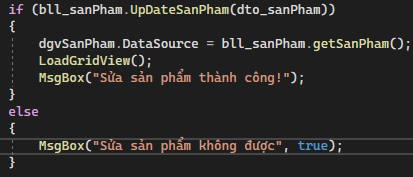
* Chức năng sửa: Sửa thông tin một sản phẩm
* Khi người dùng chọn dòng có thông tin muốn sửa, các thông tin sẽ được đổ lên các textbox, người nhập sửa thông tin muốn sửa trên textbox và click button “Sửa” sẽ nhận được thông báo “Bạn có chắc muốn sửa?”. Nếu người chọn “Có” tiến hành kiểm tra thông tin trên trên các ô textbox được nhập đầy đủ và đúng kiểu dữ liệu.



* Sau đó chuyển thông tin được nhập qua lớp DTO\_SanPham để tạo dto.



* Khi tạo dto thành công sẽ chuyển thông tin qua lớp bll có liên kết với lớp dal để sử dụng StoredProcedure “UpDateSanPham”.



* Chức năng xóa: Xóa thông tin một sản phẩm có trong sách đã lưu.
* Khi người nhập chọn dòng thông tin muốn xóa, các thông tin sẽ được đổ lên các textbox và người dùng click button “Xóa” sẽ nhận được thông báo “ Bạn có chắc chắn muốn xóa không?”.
* Nếu người dùng chọn “Có” sẽ tiến hành chuyển dữ liệu đến lớp bll có liên kết với lớp dal để sử dụng StoredProcedure “ DeleteSanPham” xóa sản phẩm đó.
* Chức năng thêm ảnh: Thêm một ảnh có trong máy. Đây là chức năng khi thêm sản phẩm, nhập các thông tin trên ô textbox, người nhập click button “Thêm ảnh” để chọn ảnh có trong máy để thêm vào thông tin sản phẩm.
* **Quản lý nhà cung cấp ( tương tự chức năng của quản lý khách hàng)**
* Chức năng thêm:
* Khi click vào button Thêm, kiểm tra người nhập đã nhập đủ các thông tin gồm tên, địa chỉ, số điện thoại, chuyển thông tin người nhập sang lớp DTO\_NhaCC.
* Sau khi tạo dtoNhaCC thành công chuyển qua lớp bllNhaCC liên kết đến lớp dalNhaCC để dùng StoredProcedure “InsertNhaCC”. Sau khi thành công tạo thông tin nhà cung cấp sẽ load lên datagridview và làm trống các ô nhập liệu. trường hợp không tạo được dtoNhaCC sẽ nhận được thông báo “Thêm nhà cung cấp không thành công.” Trường hợp không nhập đủ thông tin trên các ô nhập liệu sẽ không thêm được nhà cung cấp và nhận được thông báo “Vui lòng kiểm tra lại dữ liệu”.
* Chức năng sửa: Sửa thông tin một nhà cung cấp trong dữ liệu đang lưu.
* Khi click vào button “Sửa”, kiểm tra các thông tin hiển thị trên các ô nhập liệu không trống, hiển thị thông báo “Bạn có chắc chắn muốn sửa?”, nếu người chọn “Có”, chuyển thông tin từ các textbox qua lớp dtoNhaCC.
* Tạo dto thành công sẽ chuyển dữ liệu qua lớp bllNhaCC có liên kết với dalNhaCC để sử dụng StoredProcedure “UpdateNhaCC”. Sau khi sửa thành công dữ liệu sẽ được cập nhật trên datagridview và các ô nhập liệu sẽ được làm trống. Trường hơp không sửa được sẽ nhận được thông báo “Sửa nhà cung cấp không thành công.” Nếu khi người nhập chọn dòng dữ liệu muốn xóa, người nhập phải đảm bảo các ô textbox được nhâp đủ thông tin, không để trống, nếu có ô trống sẽ nhận thông báo “Vui lòng kiểm tra lại dữ liệu”.
* Chức năng xóa: Xóa thông tin một nhà cung cấp. Khi người nhập chọn vào dòng thông tin muốn xóa, click button xóa sẽ nhận được thông báo “Bạn chắc chắn muốn xóa.”, người dùng chọn “Có” sẽ chuyển thông tin đến bllNhaCC có liên kết với lớp dalNhaCC để sử dụng StoredProcedure “DeleteNhaCC”. Sau khi xóa thành công sẽ cập nhật dữ liệu lại trên datagridview và làm trống các ô nhập liệu.
* Chức năng tìm kiếm: Tìm kiếm thông tin nhà cung cấp đang lưu. Người dùng nhập kí tự hoặc chuỗi bất kì vào ô tìm kiếm, tiến hành tìm kiếm nhà cung cấp theo tên nhà cung cấp, thông tin có chứa ký tự hoặc chuỗi mà người dùng nhập sẽ hiện ở datagridview.

## 

## 3. Mô tả dữ liệu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên bảng:** tblKhachHang | | | | |
| **Tên cột** | **Kiểu dữ liệu** | **Cho phép null** | **Khóa** | **Mô tả** |
| MaKH | int | Not null | PK | Lưu trữ mã khách hàng, tự tăng. |
| TenKH | nvarchar(50) | Not null |  | Lưu trữ tên khách hàng |
| DiaChi | nvarchar(50) | Not null |  | Lưu trữ địa chỉ khách hàng |
| SDT | nvarchar(50) | Not null |  | Lưu trữ sđt của khách hàng |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên bảng:** tblHDBan | | | | |
| **Tên cột** | **Kiểu dữ liệu** | **Cho phép null** | **Khóa** | **Mô tả** |
| MaHDBan | int | Not null | PK | Lưu trữ mã hóa đơn bán, tự tăng. |
| MaNV | int | Not Null | FK | Lưu trữ mã nhân viên. |
| MaKH | int | Not Null | FK | Lưu trữ mã khách hàng. |
| NgayBan | datetime | Null |  | Lưu trữ ngày bán hàng |
| TongTienBan | float | Not Null |  | Lưu trữ tổng số tiền của đơn bán hàng ngày hôm đó. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên bảng:** tblCTHDBan | | | | |
| **Tên cột** | **Kiểu dữ liệu** | **Cho phép null** | **Khóa** | **Mô tả** |
| MaSP | int | Not null | PK | Lưu trữ mã sản phẩm |
| SoLuong | int | Not Null |  | Lưu số lượng sản phẩm nhập |
| GiaSP | float | Not Null |  | Lưu giá mỗi sản phẩm ( đơn giá \* số lượng) |

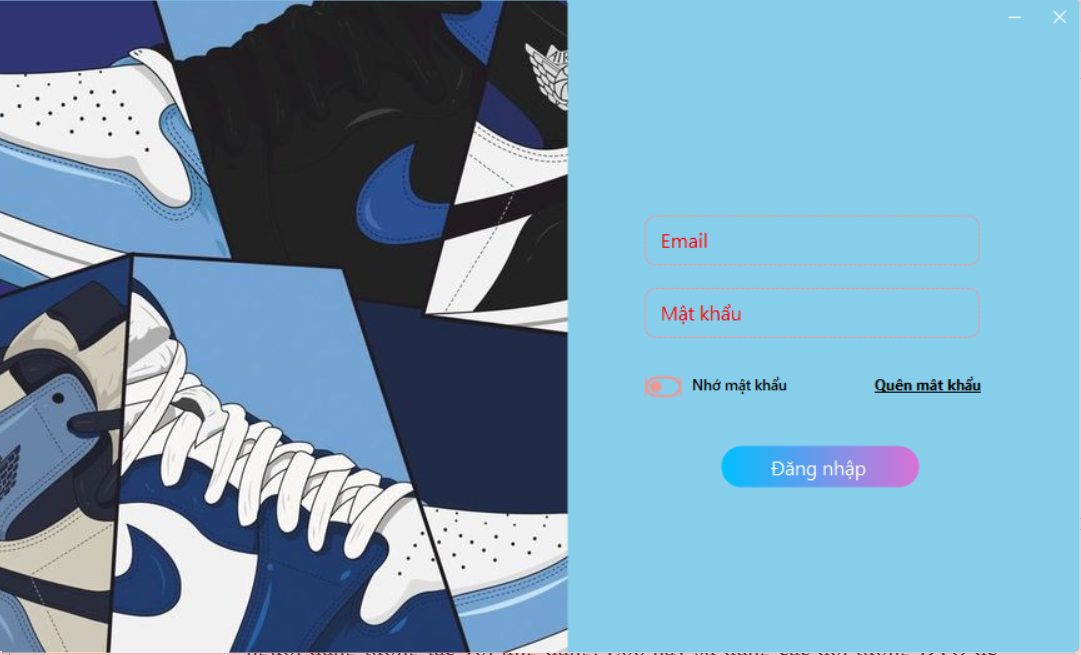
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên bảng:** tblSanPham | | | | |
| **Tên cột** | **Kiểu dữ liệu** | **Cho phép null** | **Khóa** | **Mô tả** |
| MaSP | int | Not null | PK | Lưu trữ mã sản phẩm, mã sản phẩm tự tăng(identity = yes) |
| TenSP | nvarchar(50) | Not null |  | Lưu trữ tên sản phẩm |
| SoLuong | int | Not null |  | Lưu số lượng mỗi sản phẩm |
| GiaNhapSP | float | Not null |  | Lưu giá nhập sản phẩm |
| GiaBanSP | float | Not null |  | Lưu giá bán sản phẩm |
| AnhSP | image | Not null |  | Lưu trữ ảnh mỗi sản phẩm |
| GhiChu | nvarchar(100) | Not null |  | Ghi chú cho mỗi sản phẩm |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên bảng:** tblNhaCC | | | | |
| **Tên cột** | **Kiểu dữ liệu** | **Cho phép null** | **Khóa** | **Mô tả** |
| MaNCC | int | Not null | PK | Lưu trữ mã nhà cung cấp (tự tăng) |
| TenNCC | nvarchar(50) | Not null |  | Lưu trữ tên nhà cung cấp |
| DiaChi | nvarchar(50) | Not null |  | Lưu trữ địa chỉ nhà cung cấp |
| SDT | nvarchar(50) | Not null |  | Lưu trữ sđt của NCC |

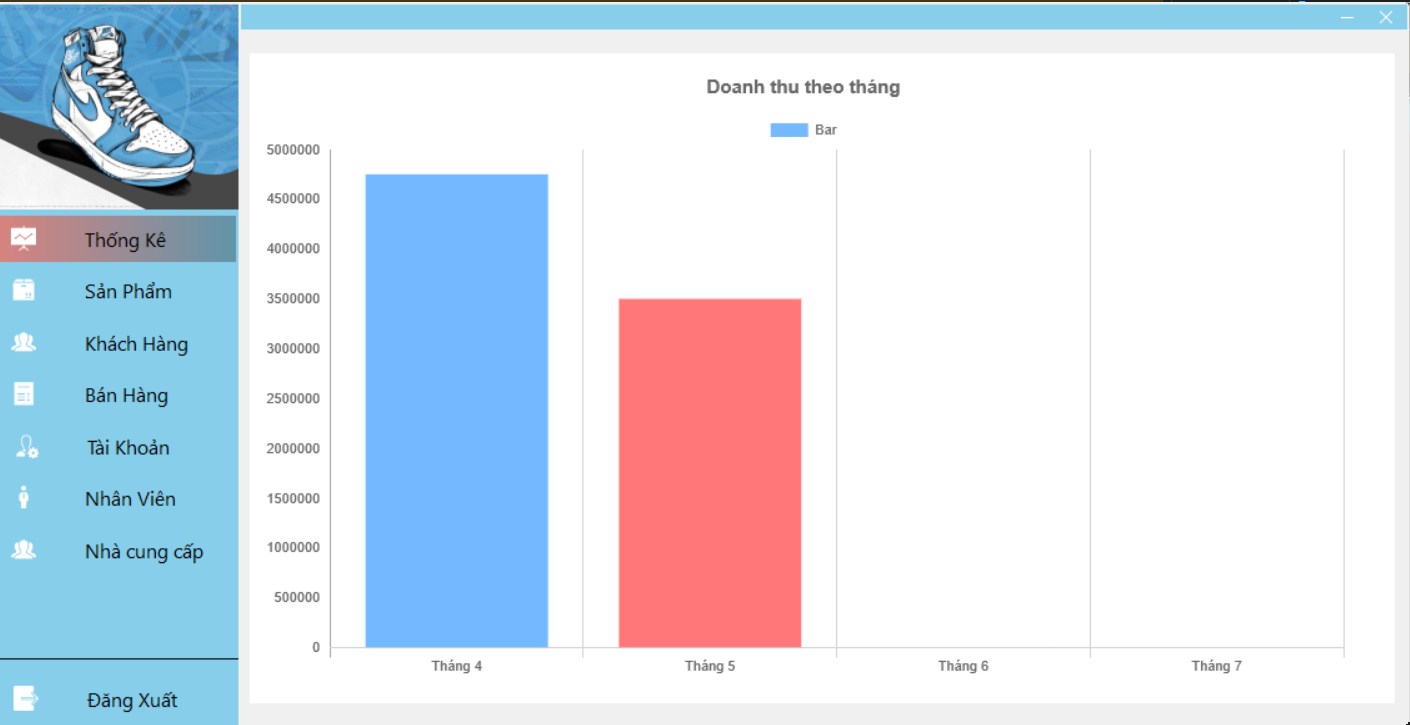
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên bảng:** tblNhanVien | | | | |
| **Tên cột** | **Kiểu dữ liệu** | **Cho phép null** | **Khóa** | **Mô tả** |
| MaNV | int | Not null | PK | Lưu trữ mã nhân viên( tự tăng) |
| TenNV | nvarchar(50) | Not null |  | Lưu trữ tên nhân viên |
| DiaChi | nvarchar(50) | Not null |  | Lưu trữ địa chỉ nhân viên |
| SDT | float | Not null |  | Lưu trữ số điện thoại nhân viên |
| Email | nvarchar(50) | Not null |  | Lưu trữ email nhân viên |
| VaiTro | bit | Not null |  | Lưu trữ vai trò nhân viên  ( bit 1 là quản trị, bit 2 là nhân viên) |
| TrangThai | bit | Not null |  | Lưu trữ trạng thái hoạt động của nhân viên( bit 1 là hoạt động, bit 2 là ngưng hoạt động |
| MK | nvarchar(50) | Not null |  | Lưu trữ mật khẩu nhân viên(đã được mã hóa) |

## 4. Mô tả sản phẩm

* Đăng nhập



* Mình sẽ nhập email và mật khẩu mình đã đăng kí vào đây, nếu như mình muốn lưu giữ lại mật khẩu cho lần đăng nhập tiếp theo thì nhấn vào nút nhớ mật khẩu. Nếu không nhớ mật khẩu thì nhấn vào nút quên mật khẩu. Mật khẩu mới sẽ được gửi vào mail mà mình đã đăng ký, nếu email chưa đăng ký thì sẽ không nhấn vào nút đó được.
* Khi mình đã nhập đủ thông tin rồi là mình nhấn vào nút đăng nhập nó sẽ chuyển mình vào giao diện người dùng(menu)
* Khi đăng nhập vào tài khoản nhân viên thì sẽ thỉ thao tác được ở các nút: sản phẩm, khách hàng, bán hàng. Ngược lại, nếu đăng nhập vào bằng tài khoản quản lý thì thao tác được trên tất cả các nút.
* Thống kê:



* Thống kê danh thu bán hàng theo 4 tháng trong năm(tháng 4,5,6,7)
* Sản phẩm:



* Nhập thông tin sản phẩm vào các ô, xong sau đó nhấn nút thêm. Thông tin sẽ được thêm vào danh sách hiển thị ở phía dưới
* Nhấn vào một dòng thông tin mà mình muốn sửa, sau đó nhập thông tin mà mình muốn sửa.
* Chọn ô muốn xóa và click button xóa, thông tin sẽ được xóa sau đó.
* Khách hàng:



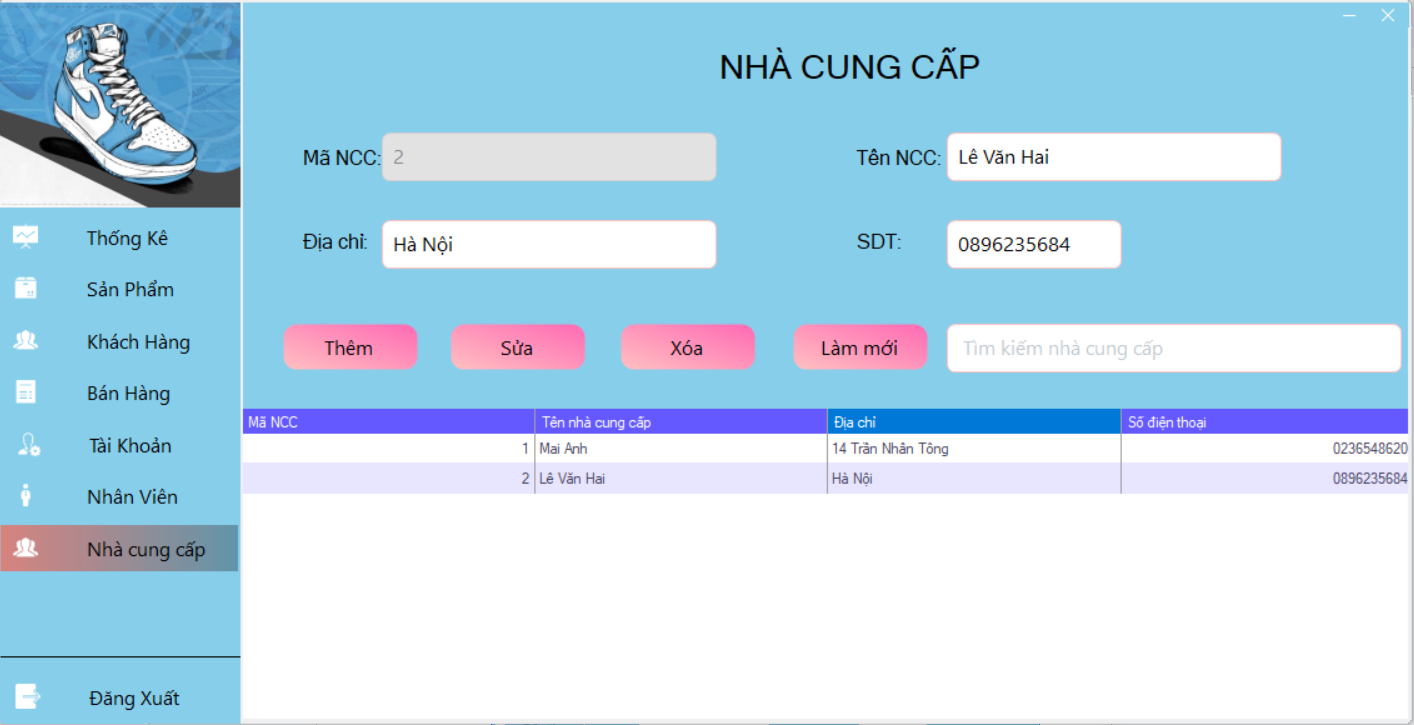
* Nhập thông tin khách hàng vào các ô, xong sau đó nhấn nút thêm. Thông tin sẽ được thêm vào danh sách hiển thị ở phía dưới
* Nhấn vào một dòng thông tin mà mình muốn sửa, sau đó nhập thông tin mà mình muốn sửa.
* Chọn ô muốn xóa và click button xóa, thông tin sẽ được xóa sau đó.
* Bán hàng:



* Khi muốn tạo một đơn hàng mới, ta bấm chọn vào nút tạo hóa đơn, sau đó nó sẽ hiện lên form chi tiết hóa đơn để mình có thể thêm được sản phẩm để thanh toán.
* Tạo chi tiết hóa đơn:



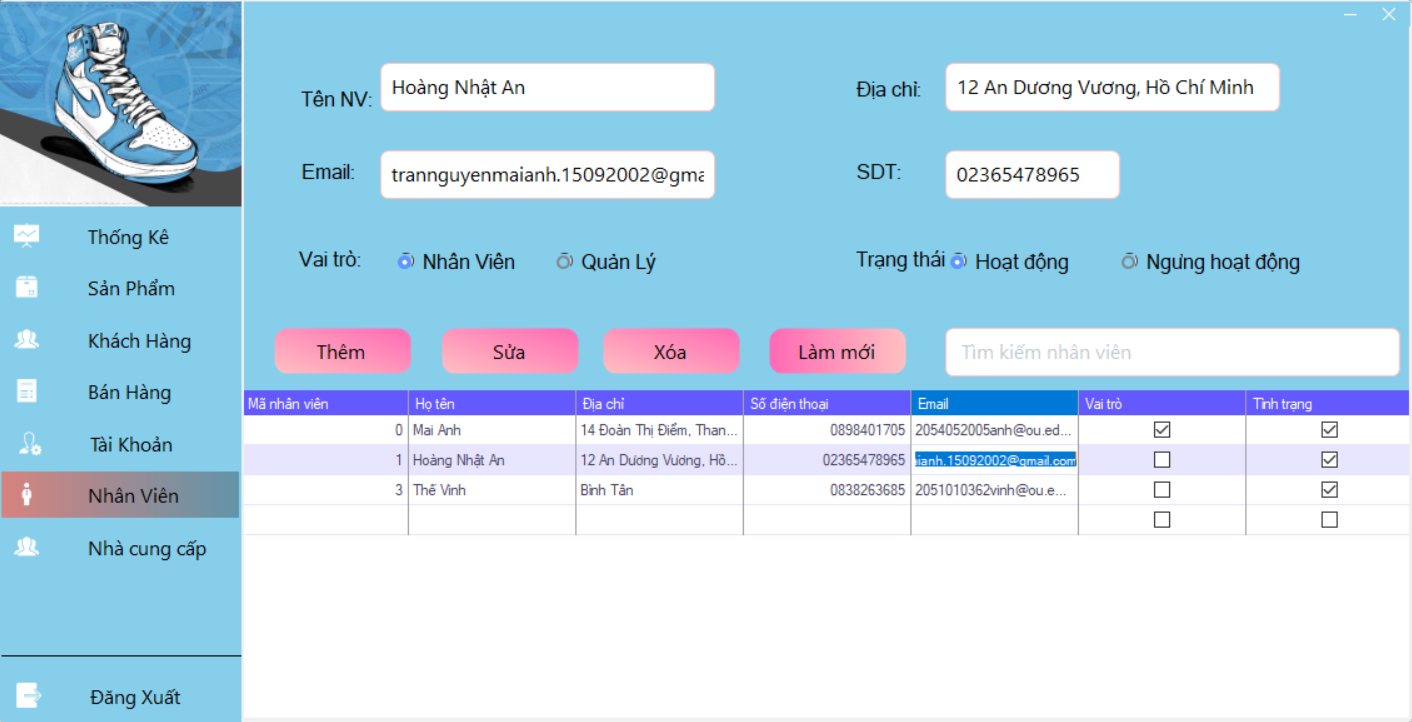
* Nhập thông tin vào các ô, xong sau đó nhấn nút thêm. Thông tin sẽ được thêm vào danh sách hiển thị ở phía dưới. Mã nv|Tên nv sẽ hiển thị khi login vào account của nhân viên đó, chế độ Readonly.
* Nhấn vào một dòng thông tin mà mình muốn sửa, sau đó nhập thông tin mà mình muốn sửa.
* Chọn ô muốn xóa và click button xóa, thông tin sẽ được xóa sau đó.
* Khi các thông tin được nhập xong, chọn thanh toán. Hóa đơn được tạo sẽ hiển thị ở form “ Danh sách hóa đơn bán”.
* Nhà cung cấp:



* Khi mình muốn thêm một nhà cung cấp, mình nhập thông tin vào các ô hiện có, ô mã sẽ không nhập được vì nó là mã tự tăng. Sau đó mình nhấn thêm thì sẽ hoàn tất việc thêm nhà cung cấp mới
* Nếu mình muốn sửa hay xóa thông tin, Mình nhấn vào một dòng hiện có ở dưới bảng, nó sẽ hiện thông tin lên và mình thao tác rồi nhấn nút “Xóa” hoặc “Sửa”.
* Khi nhấn vào nút làm mới, là tất cả các dữ liệu ở ô nhập sẽ tự động xóa.
* Tài khoản:



* Nhập thông tin mình muốn sửa lên các ô, ô email sẽ không được phép sửa. Sau khi nhập các thông tin muốn sửa, thì nhấn nút “Sửa” sẽ có một thông báo hiện lên là sửa thành công
* Nhập mật khẩu cũ, Nhập mật khẩu mới mà mình muốn đổi, xong sau đó là nhập lại một lần nữa mật khẩu. Nhấn nút “Đổi mật khẩu” mật khẩu sẽ được thay đổi
* Nhân viên:



* Nhập thông tin nhân viên vào các ô, sau đó nhấn nút thêm. Thông tin sẽ được thêm vào danh sách hiển thị ở phía dưới
* Nhấn một dòng thông tin mà mình muốn sửa, nhập thông tin cần sửa. Sau đó nhấn nút “Sửa” thông tin sẽ được sửa ngay lập tức
* Nhấn một dòng thông tin mình muốn xóa, sau đó nhấn nút “Xóa” ngay lập tức thông tin sẽ được xóa
* Muốn tìm kiếm thông tin thông tin, nhập một tên hoặc đầy đủ lên, tên nhân viên cần tìm sẽ được hiện lên trong bảng.
* Muốn xóa tất cả các ô đang có chữ cái ở trong. Nhấn nút “Làm mới” nó sẽ xóa toàn bộ.

# CHƯƠNG 4: KẾT LUẬN

## 1. Kết quả đạt được

* Ứng dụng lưu trữ được các thông tin: nhân viên, nhà cung cấp, khách hàng, sản phẩm, hóa đơn bán.
* Ứng dụng có đầy đủ các chức năng cơ bản: thêm, xóa, sửa các thông tin lưu trữ, thống kê bán hàng.
* Giao diện thân thiện, dễ dùng.

## 2. Hạn chế của đề tài

* Bắt lỗi còn hạn chế, chương trình chưa hoàn thiện.
* Hệ thống chưa gọn nhẹ.
* Chương trình chạy chưa mượt, các chức năng chưa linh động.
* Các ràng buộc về cơ sở dữ liệu còn hạn chế: chưa thể xóa nhân viên khi nhân viên đã nhập hóa đơn,...

## 3. Phát triển đề tài trong tương lai

* Hoàn thiện chương trình ở khả năng xử lý các sự kiện, các lỗi.
* Hoàn thiện tốt hơn về code lập trình C# và ràng buộc dữ liệu.
* Thêm các chức năng mới để đáp ứng điều kiện của người dùng.
* Nâng cao kỹ thuật lập trình và hoàn chỉnh chương trình theo hướng chuyên nghiệp hơn, khả năng đưa vào áp dụng thực tế khả quan.

# TÀI LIỆU THAM KHẢO

“Guna.UI2.Winforms” *YouTube*: <https://www.youtube.com/@gunaui4933>, <https://youtu.be/NReU89Npfns>

“Guna.Chart.Winform” *Youtube:* <https://github.com/sobatdata/GunaChartExamples>

“Xây dựng chương trình quản lý bán hàng.” *Tìm ở đây*,

<https://timoday.edu.vn/xay-dung-chuong-trinh-quan-ly-ban-hang-bang-c/>.

# BẢNG PHÂN CÔNG CÔNG VIỆC

|  |  |  |
| --- | --- | --- |
| Họ và tên | Công việc | Mức độ hoàn thành |
| Trần Nguyễn Mai Anh | Tạo database, thiết kế form, viết chức năng, viết báo cáo, đăng nhập, đổi mật khẩu,tạo email, thêm, sửa, xóa nhân viên, tài khoản. | 100% |
| Đinh Hoàng Thanh Trúc | Tạo database, thiết kế form,viết báo cáo ,viết chức năng, viết báo cáo, thêm, sửa, xóa, tìm kiếm nhà cung cấp và sản phẩm | 100% |
| Trần Hồ Thế Vinh | Tạo database, thiết kế form,viết báo cáo ,viết các chức năng: thêm, sửa, xóa, tìm kiếm của khách hàng, và tạo hóa đơn bán hàng, thống kê doanh thu bán hàng. | 100% |