

**TRƯỜNG ĐẠI HỌC AN GIANG**  
**KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO ĐỒ ÁN**

**Chuyên đề Python (COS525)**

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ QUÁN  
CAFE VỚI PYTHON, TKINTER VÀ SQL  
SERVER**

Giảng viên hướng dẫn: Ths. Nguyễn Ngọc Minh

Sinh viên thực hiện :DTH235833\_Trần Quang Toàn.DH24TH3\_Nhóm TH2\_Tổ TH1

AN GIANG, 11-2025

# Mục lục

I.	ĐẶT VẤN ĐỀ.....	2
II.	TỔNG QUAN VỀ CƠ SỞ LÝ THUYẾT .....	2
1.	Đặt vấn đề. ....	2
2.	Phạm vi đề tài. ....	3
2.1.	Phạm vi nghiệp vụ.....	3
2.2.	Phạm vi kĩ thuật.....	3
2.3.	Phạm vi nghiên cứu.....	3
3.	Cơ sở lý thuyết. ....	3
3.1.	Kiến thức lập trình bằng Python.....	3
3.2.	Khái niệm về hệ thống quản lý.....	3
3.3.	Cơ sở dữ liệu. ....	3
3.4.	Mô hình quản lý quán Cafe Hệ thống được thiết kế để quản lý các đối tượng:	4
III.	Demo code.....	4
1.	Cơ sở dữ liệu SQL server.....	4
2.	Giao diện. ....	4
2.1.	Giao diện đăng nhập.....	4
2.2.	Giao diện chức năng.....	5
2.3.	Giao diện bán hàng.....	5
2.4.	Giao diện kiểm tra kho.....	6
2.5.	Giao diện nhập nguyên liệu. ....	6
2.6.	Giao diện kiểm tra hóa đơn. ....	6
2.6.1.	Chức năng thống kê doanh thu. ....	7
2.7.	Giao diện quản lý.....	7
2.8.	Giao diện khách thành viên. ....	8
2.9.	Giao diện danh sách sản phẩm. ....	8
3.	Code. ....	9
3.1.	Code tạo database (Sử dụng SQL Server Mangagement Studio 20). ....	9
3.2.	Code dữ liệu chung (lưu các giá trị cần tham chiếu cho mỗi phiên đăng nhập)	11
3.3.	Code form Đăng nhập.....	12
3.4.	Code giao diện chức năng. ....	15
3.5.	Code giao diện bán hàng.....	18
3.6.	Code giao diện quản lý kho. ....	25
3.7.	Code giao diện nhập nguyên liệu.....	26
3.8.	Code giao diện quản lý hóa đơn. ....	30
3.9.	Code giao diện quản lý. ....	36

## I. ĐẶT VẤN ĐỀ.

- Hiện nay, việc quản lý một hệ thống bằng phần mềm là một xu hướng tất yếu vì khả năng lưu trữ dữ liệu, tốc độ và độ chính xác, Đặc biệt với các loại cửa hàng cần ưu tiên tốc độ phục vụ như quán cafe. Việc quản lý và sử dụng phương pháp thủ công đã không còn đủ hiệu quả trong việc đảm bảo độ chính xác, minh bạch và khả năng mở rộng. Do đó việc xây dựng phần mềm quản lý quán cafe bằng ngôn ngữ Python là rất quan trọng nhờ những ưu điểm sau:
  - Giúp tự động hóa các thao tác thêm, xóa sửa các thông tin về đơn hàng, sản phẩm, tồn kho vào trong cơ sở dữ liệu.
  - Giảm thiểu khả năng xảy ra sai sót do ghi chép thủ công.
  - Dễ dàng tra cứu thông tin.
  - Tăng hiệu quả hoạt động do giảm được công sức ghi chép.
  - Tạo phong cách chuyên nghiệp, triển khai dễ dàng đến các cửa hàng tiếp theo.
  - Khả năng nâng cấp, mở rộng quy mô và phạm vi hoạt động của phần mềm khi cần thiết.
- Việc lập đồ án quản lý quán cafe này giúp cho sinh viên cải thiện khả năng lập kế hoạch, viết phần mềm, tư duy thiết kế một hệ thống và triển khai phần mềm. Trên hết là hiểu và đáp ứng được nhu cầu thực tế của các quán cafe.

## II. TỔNG QUAN VỀ CƠ SỞ LÝ THUYẾT

### 1. Đặt vấn đề.

- Trong những năm gần đây, thị trường F&B (thực phẩm và đồ uống), đặc biệt là kinh doanh quán cafe đang chuyển mình từ các quán cafe nhỏ lẻ, sử dụng phương pháp ghi chép thủ công sang các quán cafe hiện đại, áp dụng các phương pháp quản lý sử dụng công nghệ thông tin vào các nhiệm vụ tạo, lưu trữ các thông tin.
- Nếu sử dụng các phương pháp quản lý thủ công như ghi chép bằng tay thì sẽ có nhiều hạn chế như:
  - Khó tra cứu thông tin.
  - Dễ mất mát, sai lệch dữ liệu.
  - Không có phương pháp tính doanh thu nhanh chóng.
  - Gây khó khăn trong việc triển khai khi cần mở rộng hay mở thêm cửa hàng.

- Trong khi đó, Python là một ngôn ngữ lập trình phổ biến, dễ học, với nhiều thư viện mạnh mẽ hỗ trợ xây dựng ứng dụng quản lý và tích hợp cơ sở dữ liệu. Vì vậy, xây dựng phần mềm quản lý quán cafe bằng Python là rất cần thiết để tối ưu hóa tốc độ phục vụ, giảm thiểu sai sót, và hỗ trợ nhà quản lý đưa ra quyết định kinh doanh nhanh chóng, chính xác dựa trên dữ liệu.
2. Phạm vi đề tài.

Đề tài tập trung xây dựng một phần mềm quản lý với các chức năng cơ bản cần thiết để quản lý một quán cafe gồm:

2.1. Phạm vi nghiệp vụ.

- Tạo hóa đơn bán hàng.
- Quản lý thông tin khách hàng thành viên.
- Quản lý kho (Xem tồn kho, Tạo phiếu nhập).
- Quản lý sản phẩm (Thêm, sửa sản phẩm).
- Thông báo doanh thu theo ngày, tháng, năm.

2.2. Phạm vi kĩ thuật.

- Ngôn ngữ: Python.
- Cơ sở dữ liệu: SQL server.
- Giao diện sử dụng: Tkinter.

2.3. Phạm vi nghiên cứu.

- Đề tài mang tính mô phỏng ở mức cơ bản, chưa phù hợp để triển khai thực tế tại doanh nghiệp.
- Đề tài tập trung vào tính dễ sử dụng và nâng cấp.

3. Cơ sở lý thuyết.

3.1. Kiến thức lập trình bằng Python.

- Biến, kiểu dữ liệu và cấu trúc điều khiển.
- Kỹ thuật Transaction: Áp dụng trong SQL để đảm bảo tính toàn vẹn dữ liệu khi thực hiện nhiều thao tác cùng lúc.
- Thư viện: Tkinter, datetime, os, pyodbc.
- Cơ sở dữ liệu: SQL server.
- Ngôn ngữ: Python (có sử dụng cấu trúc hướng đối tượng).

3.2. Khái niệm về hệ thống quản lý.

- Hệ thống quản lý là tập hợp các chức năng dùng cho việc nhập, lưu trữ, xử lý và xuất dữ liệu.
- Các module thường có: Quản lý hóa đơn, quản lý kho, quản lý nhân viên,...

3.3. Cơ sở dữ liệu.

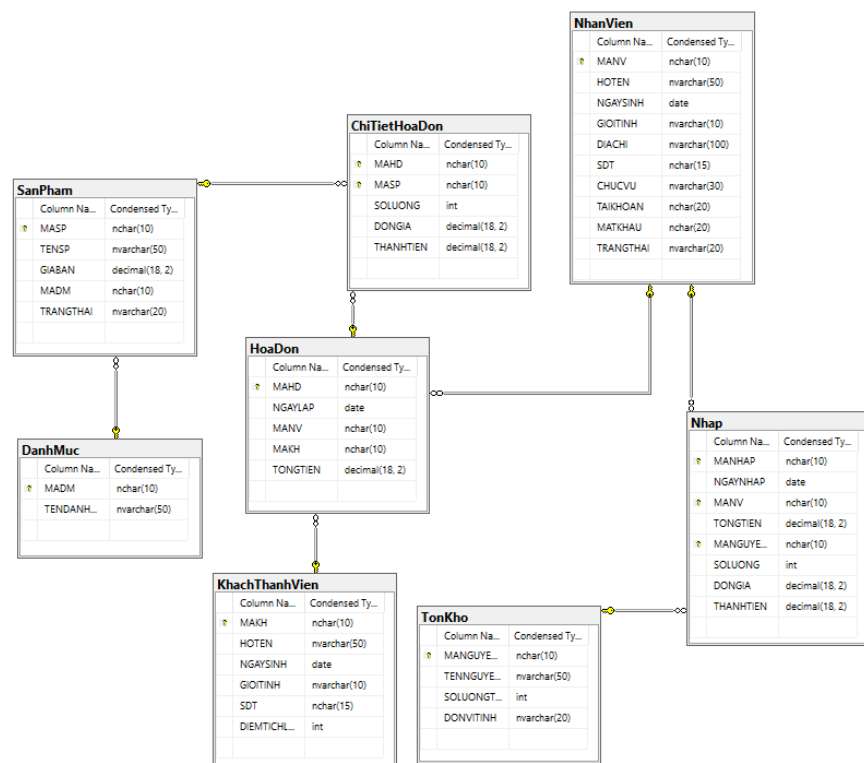
- Sử dụng Microsoft SQL Server (mô hình quan hệ) để tổ chức dữ liệu thành các bảng có liên kết chặt chẽ.
- Lựa chọn SQL Server thay vì SQLite/JSON để đảm bảo tính bảo mật, khả năng mở rộng và hỗ trợ giao dịch (Transaction) tốt hơn cho nghiệp vụ bán hàng.

3.4. Mô hình quản lý quán Cafe Hệ thống được thiết kế để quản lý các đối tượng:

- Sản phẩm & Nguyên liệu: Quản lý thực đơn và tồn kho.
- Nhân viên: Quản lý thông tin và phân quyền truy cập (Quản lý/Nhân viên).
- Khách hàng: Quản lý khách thành viên và điểm tích lũy.
- Hóa đơn & Doanh thu: Xử lý thanh toán và thống kê báo cáo theo thời gian.

### III. Demo code.

#### 1. Cơ sở dữ liệu SQL server.



#### 2. Giao diện.

##### 2.1. Giao diện đăng nhập.

The screenshot shows a login window with the title 'ĐĂNG NHẬP'. It contains two input fields: 'Tên đăng nhập' (Username) and 'Mật khẩu' (Password). Below these fields are two buttons: 'Đăng nhập' (Login) and 'Thoát' (Exit).

- Người dùng cung cấp tên đăng nhập và mật khẩu của mình.

## 2.2. Giao diện chức năng.

The screenshot shows a function interface window titled 'CHỨC NĂNG'. It displays a grid of buttons for various functions: 'Bán hàng' (Sales), 'Nhập Nguyên liệu' (Input Raw Materials), 'Quản lý' (Management), 'Danh Sách sản phẩm' (Product List), 'Kiểm tra kho' (Check Warehouse), 'Kiểm tra Hóa Đơn' (Check Invoice), 'Khách Thành Viên' (Member Customer), and 'Thoát' (Exit). The window also shows the user's role: 'chức vụ của bạn: Quản lý' (Your position: Manager).

- Tùy thuộc vào chức vụ của mình, nhân viên có thể truy cập vào các chức năng của phần mềm dựa trên các nút hiển thị trên giao diện này.

## 2.3. Giao diện bán hàng.

The screenshot shows a sales interface window titled 'BÁN HÀNG'. It includes fields for 'Mã nhân viên' (Employee ID) with value 'NV001', 'Tên nhân viên' (Employee Name) with value 'Nguyễn Văn An', and 'Chức vụ' (Position) with value 'Quản lý'. There are buttons for 'Tạo hóa đơn' (Create Invoice), 'Reset', and 'Hủy' (Cancel). Below these is a section for 'Chọn sản phẩm thêm vào giỏ' (Select product to add to cart) with a dropdown for 'Sản phẩm' (Product), 'Số lượng' (Quantity), 'Đơn giá' (Unit Price) with value '0', and buttons for 'Thêm / Cập nhật' (Add / Update) and 'Xóa dòng chọn' (Delete selected row). A table below shows columns for 'Tên Sản Phẩm' (Product Name), 'Số Lượng' (Quantity), 'Đơn Giá' (Unit Price), and 'Thành Tiền' (Amount). At the bottom, there is a field for 'Mã khách hàng' (Customer ID) with value '-- Khách lẻ --' and a 'Tổng tiền' (Total amount) field with value '0'.

- Ở giao diện này người dùng có thể tạo hóa đơn, chỉnh sửa hóa đơn đang được tạo, tính tổng giá trị hóa đơn.

#### 2.4. Giao diện kiểm tra kho.

Mã nguyên liệu	Tên nguyên liệu	Đơn vị tính	Số lượng
NL001	Hạt cà phê	1500	g
NL002	Sữa đặc	5	lít
NL003	Trái cây tươi	35	kg
NL004	Trà xanh	2500	g
NL005	đường mía	1500	kg

- Giao diện này cho phép người dùng xem số lượng hàng tồn kho.

#### 2.5. Giao diện nhập nguyên liệu.

Thông tin nguyên liệu nhập

Mã NL:  Tên NL:

Đơn vị:  Số lượng:

Mã nguyên liệu	Tên nguyên liệu	Đơn vị tính	Số lượng
----------------	-----------------	-------------	----------

- Giao diện này có chức năng giúp người dùng nhập thông tin của các nguyên liệu.

#### 2.6. Giao diện kiểm tra hóa đơn.

Quản Lý Hóa Đơn

**DANH SÁCH HÓA ĐƠN**

Xóa hóa đơn | **Thống kê DT** | Thoát

Mã hóa đơn	Ngày lập	Mã nhân viên	Mã khách hàng	Tổng tiền
HD010	17/11/2025	NV001	None	940,000
HD009	16/11/2025	NV001	TV005	1,634,000
<b>HD007</b>	<b>15/11/2025</b>	<b>NV001</b>	<b>TV001</b>	<b>11,235,000</b>
HD004	10/11/2025	NV001	TV003	300,000
HD003	03/11/2025	NV002	TV003	70,000
HD002	02/11/2025	NV003	TV002	65,000
HD001	01/11/2025	NV002	TV001	95,000

**CHI TIẾT HÓA ĐƠN**

Tên Sản Phẩm	Số Lượng	Đơn Giá	Thành Tiền
Sinh tố bơ	321	35,000	11,235,000

- Giao diện này có chức năng hiển thị thông tin hóa đơn và chi tiết cho từng hóa đơn và xóa hóa đơn (chức năng xóa hóa đơn chỉ có tài khoản có chức vụ quản lý sử dụng được).

#### 2.6.1. Chức năng thống kê doanh thu.

Thống Kê Doanh Thu

**Chọn kiểu thống kê:**

☐ Theo Ngày ☒ Theo Tháng ☐ Theo Năm

Tháng:  Năm:

**Doanh thu: 0 VNĐ**

[Xem Doanh Thu](#)

- Giao diện này có chức năng thông báo doanh thu theo ngày, tháng, năm.

#### 2.7. Giao diện quản lý.



Quản lý Nhân Viên

## NHÂN VIÊN

Mã nhân viên	Họ tên	Ngày sinh	Giới tính	Địa chỉ	Số điện thoại	Chức vụ	Tài khoản	Mật khẩu	Trạng thái
NV001	Nguyễn Văn An	15/03/1998	Nam	12 Lý Tự Trọng, TP.HCM	912345678	Quản lý	1	1	Đang làm
NV002	Lê Thị Bình	10/07/2000	Nữ	45 Hai Bà Trưng, TP.HCM	967854321	Thu ngân	binh_lt	654321	Đang làm
NV003	Trần Quốc Huy	01/12/2001	Nam	22 Pasteur, TP.HCM	909123123	Pha chế	huy_tq	123123	Đang làm
NV004	Phạm Thị Hoa	20/05/1999	Nữ	101 CMT8, TP.HCM	909999999	Phục vụ	hoa_pt	999999	Đang làm
NV005	Ngô Đức Tín	09/11/1997	Nam	56 Lê Lợi, TP.HCM	911111222	Pha chế	tin_nd	111222	Đang làm
NV006	Nguyễn Hoàng Uy	12/12/2005	Nữ	Lơng Xuyên	3123124341	Quản lý	NT_LUY	nn1	Đã nghỉ
NV007	testThuNgan	22/01/1999	Nam	nn	31231232323	Nhân Viên	nv1	nv1	Đang lm

Chi tiết nhân viên (Sửa/Thêm)

Mã NV:  Họ Tên:  Ngày Sinh:

Giới Tính:  Địa Chỉ:  SĐT:

Chức Vụ:  Tài Khoản:  Mật Khẩu:

Trạng Thái:

(Sửa xong nhớ bấm 'Cập nhật danh sách' ở trên để lưu DB)

- Giao diện này cho phép tài khoản có chức vụ quản lý quyền thêm, chỉnh sửa thông tin của nhân viên.

## 2.8. Giao diện khách thành viên.

Khách Thành Viên

## KHÁCH HÀNG

Mã khách hàng	Tên khách hàng	Ngày sinh	Giới tính	Số điện thoại	Điểm tích lũy
TV001	Trần Hải Nam	22/01/2005	Nam	908000111	255
TV002	Võ Thị Yến	09/09/1999	Nữ	933000444	120
TV003	Phạm Minh Khang	10/10/2002	Nam	912349999	400
TV005	Trần Quang Toàn	22/01/2005	Nam	3123123123	1602
TV006	Tần Thủy		Nữ	12323123	12

Thông tin chi tiết (Sửa/Thêm)

Mã KH:  Họ Tên:  Ngày Sinh (dd/MM/yyyy):

Giới Tính:  SĐT:  Điểm:

(Sau khi sửa xong bảng, bấm 'Cập nhật danh sách' ở trên để lưu DB)

- Giao diện này cho phép người dùng xem, thêm và chỉnh sửa thông tin của khách hàng thành viên.

## 2.9. Giao diện danh sách sản phẩm.

formSanPham

## SẢN PHẨM

Danh mục:

Tên sản phẩm:

Giá bán:

Trạng thái:

☒ Còn bán
 ☐ Hết

Mã SP	Tên SP	Giá Bán	Mã danh mục	Trạng Thái
SP004	Nước ép cam	32,000	DM03	Còn bán
SP006	Nước ép táo	20,000	DM03	Còn bán

- Giao diện này có chức năng xem, cập nhật và thêm sản phẩm mới

### 3. Code.

#### 3.1. Code tạo database (Sử dụng SQL Server Mangagement Studio 20).

```

CREATE TABLE NhanVien
(
    MANV NCHAR(10) PRIMARY KEY,
    HOTEN NVARCHAR(50),
    NGAYSINH DATE,
    GIOITINH NVARCHAR(10),
    DIACHI NVARCHAR(100),
    SDT NCHAR(15),
    CHUCVU NVARCHAR(30),
    TAIKHOAN NCHAR(20),
    MATKHAU NCHAR(20),
    TRANGTHAI NVARCHAR(20)
);
CREATE TABLE KhachThanhVien
(
    MAKH NCHAR(10) PRIMARY KEY,
    HOTEN NVARCHAR(50),
    NGAYSINH DATE,
    GIOITINH NVARCHAR(10),
    SDT NCHAR(15),
    DIEMTICHLUY INT
);
CREATE TABLE SanPham
(
    MASP NCHAR(10) PRIMARY KEY,
    TENS P NVARCHAR(50),
    GIABAN DECIMAL(18,2),
    MADM NCHAR(10),
    TRANGTHAI NVARCHAR(20),
    FOREIGN KEY (MADM) REFERENCES DanhMucMonAn(MADM)
);

```

```

CREATE TABLE HoaDon
(
    MAHD NCHAR(10) PRIMARY KEY,
    NGAYLAP DATE,
    MANV NCHAR(10),
    MAKH NCHAR(10),
    TONGTIEN DECIMAL(18,2),
    FOREIGN KEY (MANV) REFERENCES NhanVien(MANV),
    FOREIGN KEY (MAKH) REFERENCES KhachThanhVien(MAKH),
);

CREATE TABLE ChiTietHoaDon
(
    MAHD NCHAR(10),
    MASP NCHAR(10),
    SOLUONG INT,
    DONGIA DECIMAL(18,2),
    THANHTIEN DECIMAL(18,2),
    PRIMARY KEY (MAHD, MASP),
    FOREIGN KEY (MAHD) REFERENCES HoaDon(MAHD),
    FOREIGN KEY (MASP) REFERENCES SanPham(MASP)
);

CREATE TABLE TonKho
(
    MANGUYENLIEU NCHAR(10) PRIMARY KEY,
    TENNGUYENLIEU NVARCHAR(50),
    SOLUONGTON INT,
    DONVITINH NVARCHAR(20)
);

CREATE TABLE Nhap
(
    MANHAP NCHAR(10),
    NGAYNHAP DATE,
    MANV NCHAR(10),
    TONGTIEN DECIMAL(18,2),
    MANGUYENLIEU NCHAR(10),
    SOLUONG INT,
    DONGIA DECIMAL(18,2),
    THANHTIEN DECIMAL(18,2),
    PRIMARY KEY (MANHAP, MANV, MANGUYENLIEU),
    FOREIGN KEY (MANV) REFERENCES NhanVien(MANV),
    FOREIGN KEY (MANGUYENLIEU) REFERENCES TonKho(MANGUYENLIEU)
);

CREATE TABLE DanhMuc
(
    MADM NCHAR(10) PRIMARY KEY,
    TENDANH MUC NVARCHAR(50),
);

```

3.2. Code dữ liệu chung (lưu các giá trị cần tham chiếu cho mỗi phiên đăng nhập)

```

import os

class SharedVariables:
    MaNV = ""

    # Đường dẫn kết nối SQL
    connectionString = (
        "Driver={ODBC Driver 17 for SQL Server};"
        "Server=PERSONAL-01;"
        "Database=PY_CAFE;"
        "Trusted_Connection=yes;"
        "TrustServerCertificate=yes;"
    )

    # Tự động lấy đường dẫn thư mục chứa file code hiện tại
    base_dir = os.path.dirname(os.path.abspath(__file__))
    # Nối thêm folder Icon và tên file
    icon_path = os.path.join(base_dir, "Icon", "cafe_icon.ico")

    @staticmethod
    def set_icon(window): # Đặt icon cho cửa sổ, không cần truyền
        # đường dẫn mỗi lần
        try:
            if os.path.exists(SharedVariables.icon_path):
                window.iconbitmap(SharedVariables.icon_path)
            else:
                print(f"Không tìm thấy icon tại: {SharedVariables.icon_path}") # không có thì in log ở console
        except Exception as ex:
            print(f"Lỗi set icon: {ex}")

```

### 3.3. Code form Đăng nhập.

```

import tkinter as tk # Thư viện giao diện đồ họa
from tkinter import messagebox # Thư viện hiển thị hộp thoại
import pyodbc # Thư viện kết nối cơ sở dữ liệu
from DuLieuChung import SharedVariables

class Form1(tk.Tk): # Form chính kế thừa từ tk.Tk là cửa sổ gốc main
    def __init__(self):
        super().__init__()
        self.InitializeComponent()
        self.Form1_Load()

    def InitializeComponent(self):
        # Cấu hình Form
        self.title("Login")
        SharedVariables.set_icon(self)
        self.geometry("599x299")
        self.resizable(False, False)

```

```

        # Label 1: ĐĂNG NHẬP
        self.label1 = tk.Label(self, text="ĐĂNG NHẬP", font=("Segoe
UI", 17))
        self.label1.place(x=184, y=9, width=213, height=46)

        # Label 2: Tên đăng nhập
        self.label2 = tk.Label(self, text="Tên đăng nhập",
font=("Segoe UI", 12))
        self.label2.place(x=21, y=83, width=174, height=32)

        # txtTenDangNhap
        self.txtTenDangNhap = tk.Entry(self, font=("Segoe UI", 12))
        self.txtTenDangNhap.place(x=218, y=80, width=322, height=39)

        # Label 3: Mật khẩu
        self.label3 = tk.Label(self, text="Mật khẩu", font=("Segoe
UI", 12))
        self.label3.place(x=80, y=138, width=115, height=32)

        # txtMatKhau (UseSystemPasswordChar -> show="*")
        self.txtMatKhau = tk.Entry(self, font=("Segoe UI", 12),
show="*")
        self.txtMatKhau.place(x=218, y=135, width=322, height=39)

        # btnDangNhap
        self.btnDangNhap = tk.Button(self, text="Đăng nhập",
font=("Segoe UI", 12), command=self.btnDangNhap_Click)
        self.btnDangNhap.place(x=73, y=208, width=153, height=63)

        # btnThoat
        self.btnThoat = tk.Button(self, text="Thoát", font=("Segoe
UI", 12), command=self.btnThoat_Click)
        self.btnThoat.place(x=365, y=208, width=153, height=63)

    def Form1_Load(self):
        pass

    def KiemTraDangNhap(self, taiKhoan, matKhau):
        ketQua = False
        try:
            conn = pyodbc.connect(SharedVariables.connectionString)
            cursor = conn.cursor()

            # Câu lệnh SQL để kiểm tra tài khoản và mật khẩu
            query = "SELECT COUNT(*) FROM NhanVien WHERE TAIKHOAN
= ? AND MATKHAU = ?"
            cursor.execute(query, (taiKhoan, matKhau))

```

```

        count = cursor.fetchone()[0]

        if count > 0: # Nếu có bản ghi khớp
            ketQua = True
            # Câu lệnh SQL để lấy thông tin nhân viên
            queryMANV = "SELECT MANV FROM NhanVien WHERE
TAIKHOAN = ? AND MATKHAU = ?"
            cursor.execute(queryMANV, (taiKhoan.strip(),
matKhai.strip()))
            row = cursor.fetchone()
            if row:
                SharedVariables.MaNV = str(row[0]) # Lấy giá trị
MANV lưu vào biến chung

            conn.close()
        except Exception as ex:
            messagebox.showerror("Lỗi", "Lỗi kết nối cơ sở dữ liệu:
" + str(ex))

        return ketQua

def btnDangNhap_Click(self):
    taiKhoan = self.txtTenDangNhap.get().strip()
    matKhai = self.txtMatKhai.get().strip()

    if not taiKhoan or not matKhai:
        messagebox.showwarning("Thông báo", "Vui lòng nhập đầy
đủ thông tin đăng nhập.")
        return

    if self.KiemTraDangNhap(taiKhoan, matKhai):
        messagebox.showinfo("Thông báo", "Đăng nhập thành
công!")

    # Logic chuyển Form: Hide Login -> Show ChucNang
    self.withdraw() # Hide form hiện tại

    # Import ở đây để tránh lỗi circular import ban đầu
    from formChucNang import formChucNang
    f2 = formChucNang(self) # Truyền form cha vào để quản lý
    f2.wait_window() # Chờ form con đóng

    # Khi formChucNang đóng, code sẽ chạy tiếp ở đây (nếu
cần xử lý gì thêm)
    self.destroy() # Đóng luôn Login khi thoát
    else:
        messagebox.showerror("Lỗi đăng nhập", "Tên đăng nhập
hoặc mật khẩu không đúng.")

```

```
def btnThoat_Click(self):  
    self.destroy()
```

### 3.4. Code giao diện chức năng.

```
import tkinter as tk  
from tkinter import messagebox  
import pyodbc  
from DuLieuChung import SharedVariables  
  
class formChucNang(tk.Toplevel):  
    def __init__(self, parent=None):  
        super().__init__(parent)  
        self.parent = parent  
        self.InitializeComponent()  
        self.Form2_Load()  
  
    def InitializeComponent(self):  
        self.title("Chức năng")  
        SharedVariables.set_icon(self)  
        self.geometry("955x502")  
        self.configure(bg="Beige")  
  
        # Label Title  
        self.label1 = tk.Label(self, text="CHỨC NĂNG", font=("Segoe  
UI", 15), bg="Beige")  
        self.label1.place(x=378, y=9)  
  
        # Label Chuc Vu  
        self.label2 = tk.Label(self, text="chức vụ của bạn là: ...",  
font=("Segoe UI", 12), bg="Beige")  
        self.label2.place(x=637, y=42)  
  
        # Buttons style  
        btn_style = {"bg": "Linen", "font": ("Segoe UI", 12)}  
  
        self.btnBanHang = tk.Button(self, text="Bán hàng",  
**btn_style, command=self.btnBanHang_Click)  
        self.btnBanHang.place(x=105, y=77, width=167, height=74)  
  
        self.btnNhapNguyenLieu = tk.Button(self, text="Nhập Nguyên  
liệu", **btn_style, command=self.btnNhapNguyenLieu_Click)  
        self.btnNhapNguyenLieu.place(x=278, y=77, width=167,  
height=74)  
  
        self.btnQuanLyNhanVien = tk.Button(self, text="Quản lý",  
**btn_style, command=self.btnQuanLyNhanVien_Click)  
        self.btnQuanLyNhanVien.place(x=451, y=77, width=167,  
height=74)
```



```

        self.btnDanhSachSanPham = tk.Button(self, text="Danh Sách
sản phẩm", **btn_style, command=self.btnDanhSachSanPham_Click)
        self.btnDanhSachSanPham.place(x=624, y=77, width=167,
height=74)

        self.btnKiemTraKho = tk.Button(self, text="Kiểm tra kho",
**btn_style, command=self.btnKiemTraKho_Click)
        self.btnKiemTraKho.place(x=105, y=157, width=167, height=74)

        self.btnKiemTraHoaDon = tk.Button(self, text="Kiểm tra Hóa
Đơn", **btn_style, command=self.btnKiemTraHoaDon_Click)
        self.btnKiemTraHoaDon.place(x=278, y=157, width=167,
height=74)

        self.btnKhachThanhVien = tk.Button(self, text="Khách Thành
Viên", **btn_style, command=self.btnKhachThanhVien_Click)
        self.btnKhachThanhVien.place(x=451, y=157, width=167,
height=74)

        self.btnThoat = tk.Button(self, text="Thoát", **btn_style,
command=self.button7_Click)
        self.btnThoat.place(x=624, y=157, width=167, height=74)

    def Form2_Load(self):
        chucVu = ""
        queryCHUCVU = "SELECT CHUCVU FROM NhanVien WHERE MANV= ?"

        try:
            conn = pyodbc.connect(SharedVariables.connectionString)
            cursor = conn.cursor()
            cursor.execute(queryCHUCVU, (SharedVariables.MaNV,))
            row = cursor.fetchone()

            if row:
                chucVu = row[0].strip()
                self.label2.config(text="chức vụ của bạn: " +
chucVu)

            conn.close()
        except Exception as ex:
            # Nếu lỗi kết nối thì thôi, không làm gì để tránh crash
lúc load
            print(f"Lỗi load chức vụ: {ex}")

        # Phân quyền
        if chucVu == "Quản lý" or chucVu == "Quản Lý": # Python phân
biệt hoa thường nên check cả 2 (để phòng)
            self.btnQuanLyNhanVien['state'] = 'normal'
        else:

```

```

        self.btnQuanLyNhanVien['state'] = 'disabled'

#gọi form khác
def btnBanHang_Click(self):
    try:
        from formBanHang import Form3
        self.withdraw()
        f = Form3(self)
        f.wait_window()
        self.deiconify()
    except Exception as ex: messagebox.showerror("Lỗi", str(ex))

def btnKiemTraKho_Click(self):
    try:
        from formQuanLyKho import formQuanLyKho
        self.withdraw()
        f = formQuanLyKho(self)
        f.wait_window()
        self.deiconify()
    except Exception as ex: messagebox.showerror("Lỗi", str(ex))

def btnNhapNguyenLieu_Click(self):
    try:
        from formNhapNguyenLieu import formNhapNguyenLieu
        self.withdraw()
        f = formNhapNguyenLieu(self)
        f.wait_window()
        self.deiconify()
    except Exception as ex: messagebox.showerror("Lỗi", str(ex))

def btnKiemTraHoaDon_Click(self):
    try:
        from formQuanLyHoaDon import formQuanLyHoaDon
        self.withdraw()
        f = formQuanLyHoaDon(self)
        f.wait_window()
        self.deiconify()
    except Exception as ex: messagebox.showerror("Lỗi", str(ex))

def btnKhachThanhVien_Click(self):
    try:
        from formKhachThanhVien import formKhachThanhVien
        self.withdraw()
        f = formKhachThanhVien(self)
        f.wait_window()
        self.deiconify()
    except Exception as ex: messagebox.showerror("Lỗi", str(ex))

def btnQuanLyNhanVien_Click(self):

```

```

try:
    from formQuanLyNhanVien import formQuanLyNhanVien
    self.withdraw()
    f = formQuanLyNhanVien(self)
    f.wait_window()
    self.deiconify()
except Exception as ex: messagebox.showerror("Lỗi", str(ex))

def btnDanhSachSanPham_Click(self):
    try:
        from formSanPham import formSanPham
        self.withdraw()
        f = formSanPham(self)
        f.wait_window()
        self.deiconify()
    except Exception as ex: messagebox.showerror("Lỗi", str(ex))

def button7_Click(self):
    self.destroy()

```

### 3.5. Code giao diện bán hàng.

```

import tkinter as tk
from tkinter import ttk, messagebox
import pyodbc
from datetime import datetime
from DuLieuChung import SharedVariables

class Form3(tk.Toplevel):
    def __init__(self, parent=None):
        super().__init__(parent)
        self.parent = parent
        self.InitializeComponent()

        # Data Tables (bảng dữ liệu ảo)
        self._dtSanPham = []
        self._dtKhachHang = []

        self.Form3_Load()

    def InitializeComponent(self):
        self.title("Bán Hàng")
        SharedVariables.set_icon(self)
        self.geometry("1156x720")

        # --- Labels & Entries Thông tin nhân viên ---
        tk.Label(self, text="Mã nhân viên:").place(x=16, y=75)
        self.txtMaNV = tk.Entry(self, width=25);
        self.txtMaNV.place(x=188, y=68)

```

```

        tk.Label(self, text="Tên nhân viên:").place(x=12, y=120)
        self.txtTenNV = tk.Entry(self, width=25);
self.txtTenNV.place(x=188, y=117)

        tk.Label(self, text="Chức vụ:").place(x=16, y=165)
        self.txtChucVu = tk.Entry(self, width=25);
self.txtChucVu.place(x=188, y=162)

        tk.Label(self, text="BÁN HÀNG", font=("Segoe UI",
15)).place(x=452, y=9)

        # --- Buttons ---
        self.btnTao = tk.Button(self, text="Tạo hóa đơn", width=20,
height=2, command=self.btnTao_Click)
        self.btnTao.place(x=784, y=68)

        self.btnReset = tk.Button(self, text="Reset", width=20,
height=2, command=self.btnReset_Click)
        self.btnReset.place(x=535, y=147)

        self.btnThoat = tk.Button(self, text="Hủy", width=20,
height=2, command=self.btnHuy_Click)
        self.btnThoat.place(x=784, y=147)

        #vùng nhập dữ liệu
        input_frame = tk.LabelFrame(self, text="Chọn sản phẩm thêm
vào giỏ")
        input_frame.place(x=12, y=218, width=1132, height=80)

        tk.Label(input_frame, text="Sản phẩm:").place(x=10, y=20)
        self.cboSanPham_Input = ttk.Combobox(input_frame, width=35)
        self.cboSanPham_Input.place(x=80, y=20)
        # Event khi chọn SP để cập nhật giá
        self.cboSanPham_Input.bind("<<ComboboxSelected>>",
self.on_product_select)

        tk.Label(input_frame, text="Số lượng:").place(x=350, y=20)
        self.txtSoLuong_Input = tk.Entry(input_frame, width=10)
        self.txtSoLuong_Input.place(x=420, y=20)

        tk.Label(input_frame, text="Đơn giá:").place(x=520, y=20)
        self.lblDonGia_Display = tk.Label(input_frame, text="0",
fg="blue")
        self.lblDonGia_Display.place(x=580, y=20)

        tk.Button(input_frame, text="Thêm / Cập nhật",
command=self.add_to_grid).place(x=700, y=15)

```

```

        tk.Button(input_frame, text="Xóa dòng chọn",
command=self.remove_from_grid).place(x=820, y=15)

        # Grid1 (Treeview)
        columns = ("TenSP", "SoLuong", "DonGia", "ThanhTien")
        self.Grid1 = ttk.Treeview(self, columns=columns,
show='headings', height=10)
        self.Grid1.heading("TenSP", text="Tên Sản Phẩm")
        self.Grid1.heading("SoLuong", text="Số Lượng")
        self.Grid1.heading("DonGia", text="Đơn Giá")
        self.Grid1.heading("ThanhTien", text="Thành Tiền")

        self.Grid1.column("TenSP", width=400)
        self.Grid1.column("SoLuong", width=100)
        self.Grid1.column("DonGia", width=300)
        self.Grid1.column("ThanhTien", width=300)

        self.Grid1.place(x=12, y=300, width=1132, height=171)

        # --- Footer ---
        tk.Label(self, text="Mã khách hàng").place(x=6, y=506)
        self.cboMaKH = ttk.Combobox(self, width=25)
        self.cboMaKH.place(x=188, y=503)

        tk.Label(self, text="Tổng tiền:").place(x=720, y=491) # Thêm
label cho rõ
        self.txtThanhTien = tk.Entry(self, width=25, font=("Arial",
12, "bold"))
        self.txtThanhTien.place(x=822, y=491)

    def Form3_Load(self):
        self.txtMaNV['state'] = 'disabled'
        self.txtTenNV['state'] = 'disabled'
        self.txtChucVu['state'] = 'disabled'
        self.txtThanhTien.insert(0, "0")

        self.LayThongTinNV()
        self.LoadSanPhamvaKH()

    def LayThongTinNV(self):
        query = "SELECT MANV, HOTEN, CHUCVU FROM NhanVien WHERE MANV
= ?"
        try:
            conn = pyodbc.connect(SharedVariables.connectionString)
            cursor = conn.cursor()
            cursor.execute(query, (SharedVariables.MaNV,))
            row = cursor.fetchone()
            if row:

```

```

        self.txtMaNV['state'] = 'normal';
self.txtMaNV.delete(0, tk.END); self.txtMaNV.insert(0, row[0]);
self.txtMaNV['state'] = 'disabled'
        self.txtTenNV['state'] = 'normal';
self.txtTenNV.delete(0, tk.END); self.txtTenNV.insert(0, row[1]);
self.txtTenNV['state'] = 'disabled'
        self.txtChucVu['state'] = 'normal';
self.txtChucVu.delete(0, tk.END); self.txtChucVu.insert(0, row[2]);
self.txtChucVu['state'] = 'disabled'
        conn.close()
    except Exception as ex:
        print("Lỗi load NV: " + str(ex))

def LoadSanPhamvaKH(self):
    # Load San Pham
    self._dtSanPham = []
    try:
        conn = pyodbc.connect(SharedVariables.connectionString)
        cursor = conn.cursor()
        cursor.execute("SELECT MASP, TENSP, GIABAN FROM SanPham
ORDER BY TENSP")
        for row in cursor.fetchall():
            self._dtSanPham.append({"MASP": row[0], "TENSP":
row[1], "GIABAN": float(row[2])})

        # Đổ dữ liệu vào ComboBox Input
        sp_names = [x["TENSP"] for x in self._dtSanPham]
        self.cboSanPham_Input['values'] = sp_names

        # Load Khách Hang
        self._dtKhachHang = [{"MAKH": None, "HOTEN": "-- Khách
lẻ --"}]
        cursor.execute("SELECT MAKH, HOTEN FROM KhachThanhVien
ORDER BY HOTEN")
        for row in cursor.fetchall():
            self._dtKhachHang.append({"MAKH": row[0], "HOTEN":
row[1]})

        kh_names = [x["HOTEN"] for x in self._dtKhachHang]
        self.cboMaKH['values'] = kh_names
        self.cboMaKH.current(0)

        conn.close()
    except Exception as ex:
        messagebox.showerror("Lỗi", "Lỗi tải dữ liệu: " +
str(ex))

    # --- Logic xử lý Grid ( khi chọn sản phẩm, thay đổi số lượng )
    ---

```

```

def on_product_select(self, event):
    ten_sp = self.cboSanPham_Input.get()
    sp = next((s for s in self._dtSanPham if s["TENSP"] ==
ten_sp), None)
    if sp:
        self.lblDonGia_Display.config(text=f"{sp['GIABAN']:, .0f}
")
        self.txtSoLuong_Input.focus()

def add_to_grid(self):
    ten_sp = self.cboSanPham_Input.get()
    sl_str = self.txtSoLuong_Input.get()

    if not ten_sp or not sl_str.isdigit():
        messagebox.showwarning("Lỗi", "Vui lòng chọn SP và nhập
số lượng hợp lệ (số nguyên).")
        return

    sl = int(sl_str)
    sp = next((s for s in self._dtSanPham if s["TENSP"] ==
ten_sp), None)

    if sp:
        gia = sp['GIABAN']
        thanh_tien = sl * gia

        # Kiểm tra trùng (Logic Grid1_ChongChonTrung)
        existing_item = None
        for item in self.Grid1.get_children():
            if self.Grid1.item(item)['values'][0] == ten_sp:
                existing_item = item
                break

        if existing_item:
            # Nếu trùng thì update số lượng và thành tiền
            self.Grid1.item(existing_item, values=(ten_sp, sl,
f"{gia:, .0f}", f"{thanh_tien:, .0f}"))
        else:
            self.Grid1.insert("", tk.END, values=(ten_sp, sl,
f"{gia:, .0f}", f"{thanh_tien:, .0f}"))

        self.CalculateTotal()
        # Reset input sau khi thêm
        self.cboSanPham_Input.set("")
        self.txtSoLuong_Input.delete(0, tk.END)
        self.lblDonGia_Display.config(text="0")

def remove_from_grid(self):
    selected = self.Grid1.selection()

```

```

        if selected:
            self.Grid1.delete(selected[0])
            self.CalculateTotal()

    def CalculateTotal(self):
        total = 0
        for item in self.Grid1.get_children():
            tt_str = self.Grid1.item(item)['values'][3]
            tt = float(str(tt_str).replace(",", ""))
            total += tt

        self.txtThanhTien.delete(0, tk.END)
        self.txtThanhTien.insert(0, f"{total:,.0f}")

    def btnTao_Click(self):
        self.TaoHoaDon()

    def TaoHoaDon(self):
        # Kiểm tra Grid có dữ liệu không
        if not self.Grid1.get_children():
            messagebox.showerror("Lỗi", "Chưa có sản phẩm nào.")
            return

        maHD = self.TaoMaHD()
        tong_tien_str = self.txtThanhTien.get().replace(",", "")
        tong_tien = float(tong_tien_str) if tong_tien_str else 0
        diemThuong = int(tong_tien / 10000) # Logic TínhDiem

        # Lấy MAKH
        kh_name = self.cboMaKH.get()
        kh_obj = next((k for k in self._dtKhachHang if k["HOTEN"] == kh_name), None)
        ma_kh = kh_obj["MAKH"] if kh_obj else None

        conn = None
        try:
            conn = pyodbc.connect(SharedVariables.connectionString)
            cursor = conn.cursor() # Mặc định pyodbc dùng
            transaction, phải commit mới lưu vào DB

            # 1. Insert HoaDon
            sql_hd = "INSERT INTO HoaDon (MAHD, MANV, MAKH, NGAYLAP, TONGTIEN) VALUES (?, ?, ?, ?, ?)"
            cursor.execute(sql_hd, (maHD, self.txtMaNV.get(), ma_kh, datetime.now(), tong_tien))

            # 2. Insert ChiTiet
            sql_ct = "INSERT INTO ChiTietHoaDon (MAHD, MASP, SOLUONG, DONGIA, THANHtien) VALUES (?, ?, ?, ?, ?)"

```



```

        for item in self.Grid1.get_children():
            vals = self.Grid1.item(item)['values']
            ten_sp = vals[0]
            sl = int(vals[1])
            dg = float(str(vals[2]).replace(",", ""))
            tt = float(str(vals[3]).replace(",", ""))

            # Lấy MASP từ tên
            sp_obj = next((s for s in self._dtSanPham if
s["TENSP"] == ten_sp), None)
            masp = sp_obj["MASP"]

            cursor.execute(sql_ct, (maHD, masp, sl, dg, tt))

        # 3. Update DiemTichLuy
        if ma_kh: # Nếu là khách thành viên mới cập nhật điểm
            sql_diem = "UPDATE KhachThanhVien SET DIEMTICHLUY =
DIEMTICHLUY + ? WHERE MAKH = ?"
            cursor.execute(sql_diem, (diemThuong, ma_kh))

        conn.commit() # Transaction Commit
        messagebox.showinfo("Thông báo", "Tạo hóa đơn thành
công!")

        # Reset Form
        self.Grid1.delete(*self.Grid1.get_children())
        self.txtThanhTien.delete(0, tk.END);
        self.txtThanhTien.insert(0, "0")

    except Exception as ex:
        if conn: conn.rollback() # Rollback nếu lỗi xảy ra
        messagebox.showerror("Lỗi DB", str(ex)) # Hiển thị lỗi
do CSDL
    finally:
        if conn: conn.close()

def TaoMaHD(self):
    maHD = "HD001"
    try:
        conn = pyodbc.connect(SharedVariables.connectionString)
        cursor = conn.cursor()
        cursor.execute("SELECT TOP 1 MAHD FROM HoaDon ORDER BY
MAHD DESC")
        row = cursor.fetchone()
        if row:
            last_ma = row[0] # VD: HD005
            num = int(last_ma[2:]) + 1 # Cắt chuỗi từ vị trí 2
            maHD = "HD" + f"{num:03d}" # Format 3 số

```

```

        conn.close()
    except: pass
    return maHD

    def btnReset_Click(self):
        self.LayThongTinNV()
        self.Grid1.delete(*self.Grid1.get_children())
        self.txtThanhTien.delete(0, tk.END);
self.txtThanhTien.insert(0, "0")

    def btnHuy_Click(self):
        self.destroy()

```

### 3.6. Code giao diện quản lý kho.

```

import tkinter as tk
from tkinter import ttk, messagebox
import pyodbc
from DuLieuChung import SharedVariables

class formQuanLyKho(tk.Toplevel):
    def __init__(self, parent=None):
        super().__init__(parent)
        self.InitializeComponent()
        self.formQuanLyKho_Load()

    def InitializeComponent(self):
        self.title("formQuanLyKho")
        SharedVariables.set_icon(self)
        self.geometry("831x425")

        # Label Title
        self.label1 = tk.Label(self, text="KHO HÀNG", font=("Segoe
UI", 25))
        self.label1.place(x=50, y=22)

        # Button Thoat
        self.btnThoat = tk.Button(self, text="Thoát", width=20,
height=2, command=self.btnThoat_Click)
        self.btnThoat.place(x=569, y=41)

        # GridKho (Treeview)
        cols = ("MaNL", "TenNL", "DonViTinh", "SoLuong")
        self.GridKho = ttk.Treeview(self, columns=cols,
show='headings')

        self.GridKho.heading("MaNL", text="Mã nguyên liệu")
        self.GridKho.heading("TenNL", text="Tên nguyên liệu")
        self.GridKho.heading("DonViTinh", text="Đơn vị tính")

```

```

        self.GridKho.heading("SoLuong", text="Số lượng")

        self.GridKho.column("MaNL", width=250)
        self.GridKho.column("TenNL", width=250)
        self.GridKho.column("DonViTinh", width=150)
        self.GridKho.column("SoLuong", width=150)

        self.GridKho.place(x=12, y=118, width=800, height=253)

    def formQuanLyKho_Load(self):
        # Trong C# có code set icon, Python bỏ qua
        self.LayThongTinKho()
        # GridKho.ReadOnly = True (Tkinter Treeview mặc định là
        # ReadOnly)

    def LayThongTinKho(self):
        query = "SELECT * FROM TonKho"
        try:
            conn = pyodbc.connect(SharedVariables.connectionString)
            cursor = conn.cursor()
            cursor.execute(query)

            self.GridKho.delete(*self.GridKho.get_children())

            for row in cursor.fetchall():
                # thứ tự cột: MANGUYENLIEU, TENNGUYENLIEU,
                # DONVITINH, SOLUONGTON
                self.GridKho.insert("", tk.END, values=(row[0],
                row[1], row[2], row[3]))

            conn.close()
        except Exception as ex:
            messagebox.showerror("Lỗi", "Lỗi kết nối cơ sở dữ liệu:
            " + str(ex))
        def btnThoat_Click(self):
            self.destroy()

```

### 3.7. Code giao diện nhập nguyên liệu.

```

import tkinter as tk
from tkinter import ttk, messagebox
import pyodbc
from DuLieuChung import SharedVariables

class formNhapNguyenLieu(tk.Toplevel):
    def __init__(self, parent=None):
        super().__init__(parent)
        self.InitializeComponent()

    def InitializeComponent(self):

```

```

self.title("Nhập Nguyên Liệu")
SharedVariables.set_icon(self)
self.geometry("1180x550")

# Label Title
self.label1 = tk.Label(self, text="NHẬP NGUYÊN LIỆU",
font=("Segoe UI", 25))
self.label1.place(x=51, y=9)

# Buttons
self.btnNhap = tk.Button(self, text="NHẬP KHO (Lưu)",
width=20, height=2, command=self.btnNhap_Click, bg="lightgreen")
self.btnNhap.place(x=639, y=26)

self.btnHuy = tk.Button(self, text="Xóa danh sách",
width=20, height=2, command=self.btnHuy_Click)
self.btnHuy.place(x=906, y=82)

self.btnThoat = tk.Button(self, text="Thoát", width=20,
height=2, command=self.btnThoat_Click)
self.btnThoat.place(x=906, y=26)

#Vùng Nhập liệu (Thêm dòng vào Grid)
input_frame = tk.LabelFrame(self, text="Thông tin nguyên
liệu nhập")
input_frame.place(x=17, y=80, width=600, height=150)

tk.Label(input_frame, text="Mã NL:").grid(row=0, column=0,
pady=5)
self.txtMaNL = tk.Entry(input_frame);
self.txtMaNL.grid(row=0, column=1)

tk.Label(input_frame, text="Tên NL:").grid(row=0, column=2,
pady=5)
self.txtTenNL = tk.Entry(input_frame, width=30);
self.txtTenNL.grid(row=0, column=3)

tk.Label(input_frame, text="Đơn vị:").grid(row=1, column=0,
pady=5)
self.txtDVT = tk.Entry(input_frame); self.txtDVT.grid(row=1,
column=1)

tk.Label(input_frame, text="Số lượng:").grid(row=1,
column=2, pady=5)
self.txtSoLuong = tk.Entry(input_frame);
self.txtSoLuong.grid(row=1, column=3)

```

```

        tk.Button(input_frame, text="Thêm vào bảng bên dưới",
command=self.add_to_grid).grid(row=2, column=1, columnspan=3,
pady=10)

        # GridNhap (Treeview)
        cols = ("MaNL", "TenNL", "DonViTinh", "SoLuong")
        self.GridNhap = ttk.Treeview(self, columns=cols,
show='headings')

        self.GridNhap.heading("MaNL", text="Mã nguyên liệu")
        self.GridNhap.heading("TenNL", text="Tên nguyên liệu")
        self.GridNhap.heading("DonViTinh", text="Đơn vị tính")
        self.GridNhap.heading("SoLuong", text="Số lượng")

        self.GridNhap.column("MaNL", width=250)
        self.GridNhap.column("TenNL", width=350)
        self.GridNhap.column("DonViTinh", width=150)
        self.GridNhap.column("SoLuong", width=150)

        self.GridNhap.place(x=17, y=240, width=1132, height=253)

    def add_to_grid(self):
        # Lấy dữ liệu từ ô nhập
        ma = self.txtMaNL.get().strip()
        ten = self.txtTenNL.get().strip()
        dvt = self.txtDVT.get().strip()
        sl_str = self.txtSoLuong.get().strip()

        if not ma or not ten or not dvt or not sl_str:
            messagebox.showwarning("Lỗi", "Vui lòng nhập đầy đủ
thông tin")
            return

        if not sl_str.isdigit() or int(sl_str) <= 0:
            messagebox.showwarning("Lỗi", "Số lượng phải là số
dương")
            return

        # Thêm vào Treeview
        self.GridNhap.insert("", tk.END, values=(ma, ten, dvt,
sl_str))

        # Clear ô nhập để nhập tiếp
        self.txtMaNL.delete(0, tk.END)
        self.txtTenNL.delete(0, tk.END)
        self.txtDVT.delete(0, tk.END)
        self.txtSoLuong.delete(0, tk.END)
        self.txtMaNL.focus()

```

```

def btnHuy_Click(self):
    # Xóa hết dòng trong Grid
    self.GridNhap.delete(*self.GridNhap.get_children())

def btnNhap_Click(self):
    self.CapNhatCSDL()

def CapNhatCSDL(self):
    # Kiểm tra Grid có dữ liệu không
    if not self.GridNhap.get_children():
        messagebox.showwarning("Lỗi", "Danh sách nhập đang trống")
        return

    try:
        conn = pyodbc.connect(SharedVariables.connectionString)
        cursor = conn.cursor()

        # Duyệt từng dòng trong Grid
        for item in self.GridNhap.get_children():
            vals = self.GridNhap.item(item)['values']
            maNL = str(vals[0])
            tenNL = str(vals[1])
            donViTinh = str(vals[2])
            soLuong = int(vals[3])

            # 1. Kiểm tra tồn tại trong kho
            check_sql = "SELECT TENNGUYENLIEU, SOLUONGTON FROM TonKho WHERE MANGUYENLIEU = ? OR TENNGUYENLIEU = ?"
            cursor.execute(check_sql, (maNL, tenNL))
            rows = cursor.fetchall() # Lấy tất cả các dòng khớp

            if len(rows) > 0:
                # Logic check trùng tên/mã
                maTenTrung = False
                # Nếu tìm thấy dòng khớp Mã và Tên (trong DB, row[0] là TENNGUYENLIEU)
                match_exact = False
                for r in rows:
                    if r[0] == tenNL: # Tên khớp (và Mã đã khớp hoặc Tên đã khớp do câu Query)
                        match_exact = True
                        break

                if match_exact:
                    # UPDATE cộng dồn số lượng
                    up_sql = "UPDATE TonKho SET SOLUONGTON = SOLUONGTON + ? WHERE MANGUYENLIEU = ? AND TENNGUYENLIEU = ?"

```

```

        cursor.execute(up_sql, (soLuong, maNL,
tenNL))
    else:
        # Tồn tại Mã hoặc Tên nhưng không khớp cả
hai -> Lỗi
        raise Exception(f"Dữ liệu không hợp lệ: Mã
'{maNL}' hoặc Tên '{tenNL}' bị trùng lệch với dữ liệu kho.")
    else:
        # INSERT Mới
        in_sql = "INSERT INTO TonKho (MANGUYENLIEU,
TENNGUYENLIEU, DONVITINH, SOLUONGTON) VALUES (?, ?, ?, ?)"
        cursor.execute(in_sql, (maNL, tenNL, donViTinh,
soLuong))

        conn.commit() # Xác nhận transaction
        conn.close()

        messagebox.showinfo("Thành công", "Cập nhật kho thành
công!")

        self.GridNhap.delete(*self.GridNhap.get_children()) #
Clear lưới sau khi nhập xong

except Exception as ex:
    messagebox.showerror("Lỗi CSDL", str(ex))

def btnThoat_Click(self):
    self.destroy()

```

### 3.8. Code giao diện quản lý hóa đơn.

```

import tkinter as tk
from tkinter import ttk, messagebox
import pyodbc
from datetime import datetime
from DuLieuChung import SharedVariables

class formQuanLyHoaDon(tk.Toplevel):
    def __init__(self, parent=None):
        super().__init__(parent)
        self.InitializeComponent()
        self.formQuanLyHoaDon_Load()

    def InitializeComponent(self):
        self.title("Quản Lý Hóa Đơn")
        self.geometry("1156x700")
        SharedVariables.set_icon(self)

        self.label1 = tk.Label(self, text="DANH SÁCH HÓA ĐƠN",
font=("Segoe UI", 20, "bold"), fg="blue")

```

```

self.label1.place(x=35, y=10)

# Buttons
self.btnXoa = tk.Button(self, text="Xóa hóa đơn", width=15,
command=self.btnXoa_Click)
self.btnXoa.place(x=500, y=21)

self.btnThongKe = tk.Button(self, text="Thống kê DT",
width=15, command=self.btnThongKe_Click, bg="lightyellow")
self.btnThongKe.place(x=650, y=21)

self.btnThoat = tk.Button(self, text="Thoát", width=15,
command=self.btnThoat_Click)
self.btnThoat.place(x=890, y=21)

# GridHoaDon (Treeview)
cols = ("MaHD", "NgayLap", "MaNV", "MaKH", "TongTien")
self.GridHoaDon = ttk.Treeview(self, columns=cols,
show='headings', height=8)

self.GridHoaDon.heading("MaHD", text="Mã hóa đơn")
self.GridHoaDon.heading("NgayLap", text="Ngày lập")
self.GridHoaDon.heading("MaNV", text="Mã nhân viên")
self.GridHoaDon.heading("MaKH", text="Mã khách hàng")
self.GridHoaDon.heading("TongTien", text="Tổng tiền")

self.GridHoaDon.column("MaHD", width=100)
self.GridHoaDon.column("NgayLap", width=200)
self.GridHoaDon.column("MaNV", width=100)
self.GridHoaDon.column("MaKH", width=150)
self.GridHoaDon.column("TongTien", width=200)

self.GridHoaDon.place(x=12, y=70, width=1132, height=250)

# BIND EVENT: Khi chọn dòng ở bảng trên -> Load bảng dưới
self.GridHoaDon.bind("<<TreeviewSelect>>",
self.on_invoice_select)

tk.Label(self, text="CHI TIẾT HÓA ĐƠN", font=("Segoe UI",
16, "bold"), fg="green").place(x=35, y=340)

# GridChiTiet (Treeview)
cols_ct = ("TenSP", "SoLuong", "DonGia", "ThanhTien")
self.GridChiTiet = ttk.Treeview(self, columns=cols_ct,
show='headings', height=8)

self.GridChiTiet.heading("TenSP", text="Tên Sản Phẩm")
self.GridChiTiet.heading("SoLuong", text="Số Lượng")
self.GridChiTiet.heading("DonGia", text="Đơn Giá")

```



```

        self.GridChiTiet.heading("ThanhTien", text="Thành Tiền")

        self.GridChiTiet.column("TenSP", width=400)
        self.GridChiTiet.column("SoLuong", width=100)
        self.GridChiTiet.column("DonGia", width=200)
        self.GridChiTiet.column("ThanhTien", width=200)

        self.GridChiTiet.place(x=12, y=380, width=1132, height=250)

    def formQuanLyHoaDon_Load(self):
        self.XemQuyen()
        self.NapDuLieu()

    def XemQuyen(self):
        query = "SELECT CHUCVU FROM NhanVien WHERE MANV= ?"
        try:
            conn = pyodbc.connect(SharedVariables.connectionString)
            cursor = conn.cursor()
            cursor.execute(query, (SharedVariables.MaNV,))
            row = cursor.fetchone()
            if row:
                chucVu = row[0].strip()
                if chucVu.lower() != "quản lý":
                    self.btnXoa['state'] = 'disabled'
            conn.close()
        except Exception as ex: pass

    def NapDuLieu(self):
        query = "SELECT * FROM HoaDon ORDER BY NGAYLAP DESC"
        try:
            self.GridHoaDon.delete(*self.GridHoaDon.get_children())
            self.GridChiTiet.delete(*self.GridChiTiet.get_children())
        )

        conn = pyodbc.connect(SharedVariables.connectionString)
        cursor = conn.cursor()
        cursor.execute(query)
        for row in cursor.fetchall():

            # row[1] là NGAYLAP
            ngayLap = ""
            if row[1]:
                try: ngayLap = row[1].strftime("%d/%m/%Y")
                except: ngayLap = str(row[1])

            # row[4] là TONGTIEN
            tongTien = f"{row[4]:,.0f}" if row[4] else "0"

```

```

        self.GridHoaDon.insert("", tk.END, values=(row[0],
ngayLap, row[2], row[3], tongTien))

        conn.close()
    except Exception as ex:
        messagebox.showerror("Lỗi", "Lỗi tải dữ liệu: " +
str(ex))

# --- LOGIC HIỂN THỊ CHI TIẾT ---
def on_invoice_select(self, event):
    # Lấy dòng đang chọn
    selected = self.GridHoaDon.selection()
    if not selected: return

    # Lấy MaHD (cột đầu tiên)
    item = self.GridHoaDon.item(selected[0])
    maHD = item['values'][0]

    self.LoadChiTiet(maHD)

def LoadChiTiet(self, maHD):
    # Query JOIN để lấy Tên SP thay vì Mã SP
    query = """
SELECT B.TENSP, A.SOLUONG, A.DONGIA, A.THANHTIEN
FROM ChiTietHoaDon A
JOIN SanPham B ON A.MASP = B.MASP
WHERE A.MAHD = ?
"""
    try:
        self.GridChiTiet.delete(*self.GridChiTiet.get_children()
)

        conn = pyodbc.connect(SharedVariables.connectionString)
        cursor = conn.cursor()
        cursor.execute(query, (maHD,))

        for row in cursor.fetchall():
            # row[0]=TenSP, row[1]=SoLuong, row[2]=DonGia,
row[3]=ThanhTien
            dg = f"{row[2]:,.0f}" if row[2] else "0"
            tt = f"{row[3]:,.0f}" if row[3] else "0"

            self.GridChiTiet.insert("", tk.END, values=(row[0],
row[1], dg, tt))

        conn.close()
    except Exception as ex:
        print("Lỗi tải chi tiết: " + str(ex))

def btnXoa_Click(self):
    selected = self.GridHoaDon.selection()

```

```

        if not selected:
            messagebox.showwarning("Chú ý", "Vui lòng chọn hóa đơn cần xóa")
            return
        confirm = messagebox.askyesno("Xác nhận", "Bạn có chắc muốn xóa hóa đơn này không?")
        if not confirm: return
        item = self.GridHoaDon.item(selected[0])
        maHD = item['values'][0]
        self.XoaHoaDon(maHD)

    def XoaHoaDon(self, maHD):
        sql_details = "DELETE FROM ChiTietHoaDon WHERE MAHD = ?"
        sql_header = "DELETE FROM HoaDon WHERE MAHD = ?"
        try:
            conn = pyodbc.connect(SharedVariables.connectionString)
            cursor = conn.cursor()
            cursor.execute(sql_details, (maHD,))
            cursor.execute(sql_header, (maHD,))
            conn.commit()
            conn.close()
            messagebox.showinfo("Thông báo", "Xóa hóa đơn thành công!")
            self.NapDuLieu()
        except Exception as ex:
            messagebox.showerror("Lỗi DB", str(ex))

    def btnThongKe_Click(self):
        self.hien_thi_popup_thong_ke()

    def hien_thi_popup_thong_ke(self):
        top = tk.Toplevel(self)
        top.title("Thống Kê Doanh Thu")
        top.geometry("400x300")
        SharedVariables.set_icon(top)

        tk.Label(top, text="Chọn kiểu thống kê:", font=("Arial", 12, "bold")).pack(pady=10)
        self.var_mode = tk.StringVar(value="Tháng")

        frame_opt = tk.Frame(top); frame_opt.pack()
        tk.Radiobutton(frame_opt, text="Theo Ngày", variable=self.var_mode, value="Ngày", command=lambda: self.update_inputs(frame_input)).pack(side=tk.LEFT, padx=10)
        tk.Radiobutton(frame_opt, text="Theo Tháng", variable=self.var_mode, value="Tháng", command=lambda: self.update_inputs(frame_input)).pack(side=tk.LEFT, padx=10)

```

```

        tk.Radiobutton(frame_opt, text="Theo Năm",
variable=self.var_mode, value="Năm", command=lambda:
self.update_inputs(frame_input)).pack(side=tk.LEFT, padx=10)

        frame_input = tk.Frame(top); frame_input.pack(pady=15)
        self.lbl_ketqua = tk.Label(top, text="Doanh thu: 0 VNĐ",
font=("Arial", 14, "bold"), fg="red"); self.lbl_ketqua.pack(pady=20)

        self.cbo_day = ttk.Combobox(frame_input, width=5,
values=[str(i) for i in range(1, 32)])
        self.cbo_month = ttk.Combobox(frame_input, width=5,
values=[str(i) for i in range(1, 13)])
        self.cbo_year = ttk.Combobox(frame_input, width=8,
values=[str(i) for i in range(2020, 2030)])

        tk.Button(top, text="Xem Doanh Thu", bg="lightblue",
command=self.tinh_doanh_thu).pack()
        self.update_inputs(frame_input)
        now = datetime.now()
        self.cbo_day.set(now.day); self.cbo_month.set(now.month);
self.cbo_year.set(now.year)

    def update_inputs(self, parent_frame):
        mode = self.var_mode.get()
        for widget in parent_frame.winfo_children():
            widget.pack_forget()
        if mode == "Ngày":
            tk.Label(parent_frame, text="Ngày:").pack(side=tk.LEFT);
self.cbo_day.pack(side=tk.LEFT)
            tk.Label(parent_frame,
text="Tháng:").pack(side=tk.LEFT); self.cbo_month.pack(side=tk.LEFT)
            tk.Label(parent_frame, text="Năm:").pack(side=tk.LEFT);
self.cbo_year.pack(side=tk.LEFT)
        elif mode == "Tháng":
            tk.Label(parent_frame,
text="Tháng:").pack(side=tk.LEFT); self.cbo_month.pack(side=tk.LEFT)
            tk.Label(parent_frame, text="Năm:").pack(side=tk.LEFT);
self.cbo_year.pack(side=tk.LEFT)
        elif mode == "Năm":
            tk.Label(parent_frame, text="Năm:").pack(side=tk.LEFT);
self.cbo_year.pack(side=tk.LEFT)

    def tinh_doanh_thu(self):
        mode = self.var_mode.get()
        try:
            d, m, y = self.cbo_day.get(), self.cbo_month.get(),
self.cbo_year.get()
            conn = pyodbc.connect(SharedVariables.connectionString);
            cursor = conn.cursor()

```

```

        sql = ""
        params = ()
        if mode == "Ngày": sql = "SELECT SUM(TONGTIEN) FROM
HoaDon WHERE DAY(NGAYLAP)=? AND MONTH(NGAYLAP)=? AND
YEAR(NGAYLAP)=?"; params = (d, m, y)
        elif mode == "Tháng": sql = "SELECT SUM(TONGTIEN) FROM
HoaDon WHERE MONTH(NGAYLAP)=? AND YEAR(NGAYLAP)=?"; params = (m, y)
        elif mode == "Năm": sql = "SELECT SUM(TONGTIEN) FROM
HoaDon WHERE YEAR(NGAYLAP)=?"; params = (y,)

        cursor.execute(sql, params)
        row = cursor.fetchone()
        total = float(row[0]) if row and row[0] else 0
        self.lbl_ketqua.config(text=f"Doanh thu: {total:,.0f}
VNĐ")

        conn.close()
    except Exception as ex: messagebox.showerror("Lỗi", str(ex))

def btnThoat_Click(self):
    self.destroy()

```

### 3.9. Code giao diện quản lý.

```

import tkinter as tk
from tkinter import ttk, messagebox
import pyodbc
from datetime import datetime
from DuLieuChung import SharedVariables

class formQuanLyNhanVien(tk.Toplevel):
    def __init__(self, parent=None):
        super().__init__(parent)
        self.InitializeComponent()
        self.formQuanLyNhanVien_Load()

    def InitializeComponent(self):
        self.title("QuanLyNhanVien")
        SharedVariables.set_icon(self)
        self.geometry("1156x720")

        # Label Title
        self.label1 = tk.Label(self, text="NHÂN VIÊN", font=("Segoe
UI", 25))
        self.label1.place(x=50, y=3)

        # Buttons
        self.btnReset = tk.Button(self, text="Reset", width=20,
height=2, command=self.btnReset_Click)
        self.btnReset.place(x=407, y=22)

```

```

        self.btnCapNhat = tk.Button(self, text="Cập nhật danh sách",
width=20, height=2, command=self.btnCapNhat_Click)
        self.btnCapNhat.place(x=656, y=20)

        self.btnThoat = tk.Button(self, text="Thoát", width=20,
height=2, command=self.btnThoat_Click)
        self.btnThoat.place(x=905, y=20)

        # GridNhanVien (Treeview)
        cols = ("MaNV", "HoTen", "NgaySinh", "GioiTinh", "DiaChi",
"SDT", "ChucVu", "TaiKhoan", "MatKhu", "TrangThai")
        self.GridNhanVien = ttk.Treeview(self, columns=cols,
show='headings')

        self.GridNhanVien.heading("MaNV", text="Mã nhân viên")
        self.GridNhanVien.heading("HoTen", text="Họ tên")
        self.GridNhanVien.heading("NgaySinh", text="Ngày sinh")
        self.GridNhanVien.heading("GioiTinh", text="Giới tính")
        self.GridNhanVien.heading("DiaChi", text="Địa chỉ")
        self.GridNhanVien.heading("SDT", text="Số điện thoại")
        self.GridNhanVien.heading("ChucVu", text="Chức vụ")
        self.GridNhanVien.heading("TaiKhoan", text="Tài khoản")
        self.GridNhanVien.heading("MatKhu", text="Mật khẩu")
        self.GridNhanVien.heading("TrangThai", text="Trạng thái")

        # Set width nhỏ lại xíu để vừa màn hình
        self.GridNhanVien.column("MaNV", width=80);
self.GridNhanVien.column("HoTen", width=150)
        self.GridNhanVien.column("NgaySinh", width=100);
self.GridNhanVien.column("GioiTinh", width=60)
        self.GridNhanVien.column("DiaChi", width=150);
self.GridNhanVien.column("SDT", width=100)
        self.GridNhanVien.column("ChucVu", width=100);
self.GridNhanVien.column("TaiKhoan", width=100)
        self.GridNhanVien.column("MatKhu", width=100);
self.GridNhanVien.column("TrangThai", width=100)

        self.GridNhanVien.place(x=12, y=99, width=1132, height=320)
        self.GridNhanVien.bind("<<TreeviewSelect>>",
self.on_row_select)

        # --- Vùng Nhập Liệu (Thay thế Editing Control của GridView)
        ---
        edit_frame = tk.LabelFrame(self, text="Chi tiết nhân viên
(Sửa/Thêm)")
        edit_frame.place(x=12, y=430, width=1132, height=250)

        # Tạo các ô nhập liệu

```

```

        tk.Label(edit_frame, text="Mã NV:").grid(row=0, column=0,
padx=5, pady=5)
        self.txtMaNV = tk.Entry(edit_frame);
self.txtMaNV.grid(row=0, column=1)

        tk.Label(edit_frame, text="Họ Tên:").grid(row=0, column=2,
padx=5)
        self.txtHoTen = tk.Entry(edit_frame, width=30);
self.txtHoTen.grid(row=0, column=3)

        tk.Label(edit_frame, text="Ngày Sinh:").grid(row=0,
column=4, padx=5)
        self.txtNgaySinh = tk.Entry(edit_frame);
self.txtNgaySinh.grid(row=0, column=5) # dd/mm/yyyy

        tk.Label(edit_frame, text="Giới Tính:").grid(row=1,
column=0, padx=5, pady=5)
        self.txtGioiTinh = ttk.Combobox(edit_frame, values=["Nam",
"Nữ"], width=17); self.txtGioiTinh.grid(row=1, column=1)

        tk.Label(edit_frame, text="Địa Chỉ:").grid(row=1, column=2,
padx=5)
        self.txtDiaChi = tk.Entry(edit_frame, width=30);
self.txtDiaChi.grid(row=1, column=3)

        tk.Label(edit_frame, text="SĐT:").grid(row=1, column=4,
padx=5)
        self.txtSDT = tk.Entry(edit_frame); self.txtSDT.grid(row=1,
column=5)

        tk.Label(edit_frame, text="Chức Vụ:").grid(row=2, column=0,
padx=5, pady=5)
        self.txtChucVu = ttk.Combobox(edit_frame, values=["Quản Lý",
"Nhân Viên"], width=17); self.txtChucVu.grid(row=2, column=1)

        tk.Label(edit_frame, text="Tài Khoản:").grid(row=2,
column=2, padx=5)
        self.txtTaiKhoan = tk.Entry(edit_frame, width=30);
self.txtTaiKhoan.grid(row=2, column=3)

        tk.Label(edit_frame, text="Mật Khẩu:").grid(row=2, column=4,
padx=5)
        self.txtMatKhau = tk.Entry(edit_frame);
self.txtMatKhau.grid(row=2, column=5)

        tk.Label(edit_frame, text="Trạng Thái:").grid(row=3,
column=0, padx=5, pady=5)
        self.txtTrangThai = tk.Entry(edit_frame);
self.txtTrangThai.grid(row=3, column=1)

```

```

        # Nút thao tác cục bộ trên Grid
        btn_frame = tk.Frame(edit_frame)
        btn_frame.grid(row=4, column=0, columnspan=6, pady=10)

        tk.Button(btn_frame, text="Lưu vào bảng (Tạm)",
command=self.update_grid_row, bg="lightblue",
width=20).pack(side=tk.LEFT, padx=10)
        tk.Button(btn_frame, text="Thêm dòng mới",
command=self.add_new_row, bg="lightgreen",
width=20).pack(side=tk.LEFT, padx=10)
        tk.Label(btn_frame, text="(Sửa xong nhớ bấm 'Cập nhật danh
sách' ở trên để lưu DB)", fg="red").pack(side=tk.LEFT)

    def formQuanLyNhanVien_Load(self):
        self.NapDuLieu()

    def NapDuLieu(self):
        query = "SELECT * FROM NhanVien"
        try:
            conn = pyodbc.connect(SharedVariables.connectionString)
            cursor = conn.cursor()
            cursor.execute(query)

            self.GridNhanVien.delete(*self.GridNhanVien.get_children
())

            for row in cursor.fetchall():
                ns_str = ""
                if row[2]:
                    ns_str = row[2].strftime("%d/%m/%Y")
                    # Map columns: MANV, HOTEN, NGAYSINH, GIOITINH,
DIACHI, SDT, CHUCVU, TAIKHOAN, MATKHAU, TRANGTHAI
                    self.GridNhanVien.insert("", tk.END, values=(row[0],
row[1], ns_str, row[3], row[4], row[5], row[6], row[7], row[8],
row[9]))

            conn.close()
        except Exception as ex:
            messagebox.showerror("Lỗi", "Lỗi kết nối CSDL: " +
str(ex))

    def on_row_select(self, event):
        # Đổ dữ liệu từ dòng chọn xuống ô nhập
        selected = self.GridNhanVien.selection()
        if selected:
            vals = self.GridNhanVien.item(selected[0])['values']
            self.txtMaNV.delete(0, tk.END); self.txtMaNV.insert(0,
vals[0])

```



```

        self.txtHoTen.delete(0, tk.END); self.txtHoTen.insert(0,
vals[1])
        self.txtNgaySinh.delete(0, tk.END);
self.txtNgaySinh.insert(0, vals[2])
        self.txtGioiTinh.set(vals[3])
        self.txtDiaChi.delete(0, tk.END);
self.txtDiaChi.insert(0, vals[4])
        self.txtSDT.delete(0, tk.END); self.txtSDT.insert(0,
vals[5])
        self.txtChucVu.set(vals[6])
        self.txtTaiKhoan.delete(0, tk.END);
self.txtTaiKhoan.insert(0, vals[7])
        self.txtMatKhau.delete(0, tk.END);
self.txtMatKhau.insert(0, vals[8])
        self.txtTrangThai.delete(0, tk.END);
self.txtTrangThai.insert(0, vals[9])

    def get_input_values(self):
        return (
            self.txtMaNV.get(), self.txtHoTen.get(),
self.txtNgaySinh.get(),
            self.txtGioiTinh.get(), self.txtDiaChi.get(),
self.txtSDT.get(),
            self.txtChucVu.get(), self.txtTaiKhoan.get(),
self.txtMatKhau.get(),
            self.txtTrangThai.get()
        )

    def update_grid_row(self):
        selected = self.GridNhanVien.selection()
        if selected:
            self.GridNhanVien.item(selected[0],
values=self.get_input_values())
        else:
            messagebox.showwarning("Chú ý", "Chưa chọn dòng để sửa")

    def add_new_row(self):
        self.GridNhanVien.insert("", tk.END,
values=self.get_input_values())

    def btnCapNhat_Click(self):
        confirm = messagebox.askyesno("Xác nhận", "Bạn có chắc muốn
cập nhật dữ liệu nhân viên không?")
        if confirm:
            self.CapNhatDuLieu()

    def CapNhatDuLieu(self):
        try:
            conn = pyodbc.connect(SharedVariables.connectionString)

```

```

        cursor = conn.cursor()

        for item in self.GridNhanVien.get_children():
            vals = self.GridNhanVien.item(item)['values']

            # Kiểm tra null (giả lập logic đếm cell null)
            if not vals[0] or not vals[1]: # Check cơ bản MaNV,
            TenNV là ko dc null
                messagebox.showerror("Lỗi", "Dữ liệu nhân viên
không được để trống!")
                return

            manv = str(vals[0])
            hoten = str(vals[1])
            ngaysinh_str = str(vals[2])
            gioitinh = str(vals[3])
            diachi = str(vals[4])
            sdt = str(vals[5])
            chucvu = str(vals[6])
            taikhoan = str(vals[7])
            matkhou = str(vals[8])
            trangthai = str(vals[9])

            # Parse Date
            ngaysinh = None
            try:
                ngaysinh = datetime.strptime(ngaysinh_str,
"%d/%m/%Y")
            except: pass

            # Check Exist
            cursor.execute("SELECT COUNT(*) FROM NhanVien WHERE
MANV = ?", (manv,))
            count = cursor.fetchone()[0]

            if count == 0:
                # INSERT
                sql = """INSERT INTO NhanVien (MANV, HOTEN,
NGAYSINH, GIOITINH, DIACHI, SDT, CHUCVU, TAIKHOAN, MATKHAU,
TRANGTHAI)
                VALUES
                (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)"""
                cursor.execute(sql, (manv, hoten, ngaysinh,
gioitinh, diachi, sdt, chucvu, taikhoan, matkhou, trangthai))
            else:
                # UPDATE
                sql = """UPDATE NhanVien
                SET HOTEN=?, NGAYSINH=?, GIOITINH=?,
DIACHI=?, SDT=?, CHUCVU=?, TAIKHOAN=?, MATKHAU=?, TRANGTHAI=?

```

```

WHERE MANV=?"""
        cursor.execute(sql, (hoten, ngaysinh, gioitinh,
diachi, sdt, chucvu, taikhoan, matkhou, trangthai, manv))

        conn.commit()
        conn.close()
        messagebox.showinfo("Thông báo", "Cập nhật dữ liệu nhân
viên thành công!")

    except Exception as ex:
        messagebox.showerror("Lỗi", "Lỗi cập nhật DB: " +
str(ex))

def btnReset_Click(self):
    self.NapDuLieu()

def btnThoat_Click(self):
    self.destroy()

```

### 3.10. Giao diện khách hàng thành viên.

```

import tkinter as tk
from tkinter import ttk, messagebox
import pyodbc
from datetime import datetime
from DuLieuChung import SharedVariables

class formKhachThanhVien(tk.Toplevel):
    def __init__(self, parent=None):
        super().__init__(parent)
        self.InitializeComponent()
        self.formKhachThanhVien_Load()

    def InitializeComponent(self):
        self.title("Khách Thành Viên")
        SharedVariables.set_icon(self)
        self.geometry("1162x650")

        # Label Title
        tk.Label(self, text="KHÁCH HÀNG", font=("Segoe UI",
25)).place(x=50, y=17)

        # Buttons
        self.btnReset = tk.Button(self, text="Reset", width=20,
height=2, command=self.btnReset_Click)
        self.btnReset.place(x=401, y=34)

        self.btnCapNhat = tk.Button(self, text="Cập nhật danh sách",
width=20, height=2, command=self.btnCapNhat_Click)

```

```

        self.btnCapNhat.place(x=656, y=34)

        self.btnThoat = tk.Button(self, text="Thoát", width=20,
height=2, command=self.btnThoat_Click)
        self.btnThoat.place(x=905, y=34)

        # GridKhachHang (Treeview)
        cols = ("MaKH", "HoTen", "NgaySinh", "GioiTinh", "SDT",
"DiemTichLuy")
        self.GridKhachHang = ttk.Treeview(self, columns=cols,
show='headings')

        self.GridKhachHang.heading("MaKH", text="Mã khách hàng")
        self.GridKhachHang.heading("HoTen", text="Tên khách hàng")
        self.GridKhachHang.heading("NgaySinh", text="Ngày sinh")
        self.GridKhachHang.heading("GioiTinh", text="Giới tính")
        self.GridKhachHang.heading("SDT", text="Số điện thoại")
        self.GridKhachHang.heading("DiemTichLuy", text="Điểm tích
lũy")

        self.GridKhachHang.column("MaKH", width=150)
        self.GridKhachHang.column("HoTen", width=300)
        self.GridKhachHang.column("NgaySinh", width=150)

        self.GridKhachHang.place(x=12, y=113, width=1132,
height=320)

        # Bind sự kiện chọn dòng
        self.GridKhachHang.bind("<<TreeviewSelect>>",
self.on_row_select)

        # --- Vùng Edit
        edit_frame = tk.LabelFrame(self, text="Thông tin chi tiết
(Sửa/Thêm)")
        edit_frame.place(x=12, y=450, width=1132, height=150)

        # Dòng 1
        tk.Label(edit_frame, text="Mã KH:").grid(row=0, column=0,
padx=5, pady=5)
        self.txtMaKH = tk.Entry(edit_frame);
        self.txtMaKH.grid(row=0, column=1)

        tk.Label(edit_frame, text="Họ Tên:").grid(row=0, column=2,
padx=5)
        self.txtHoTen = tk.Entry(edit_frame, width=30);
        self.txtHoTen.grid(row=0, column=3)

        tk.Label(edit_frame, text="Ngày Sinh
(dd/MM/yyyy):").grid(row=0, column=4, padx=5)

```

```

        self.txtNgaySinh = tk.Entry(edit_frame);
self.txtNgaySinh.grid(row=0, column=5)

        # Dòng 2
        tk.Label(edit_frame, text="Giới Tính:").grid(row=1,
column=0, padx=5, pady=5)
        self.txtGioiTinh = tk.Entry(edit_frame);
self.txtGioiTinh.grid(row=1, column=1)

        tk.Label(edit_frame, text="SĐT:").grid(row=1, column=2,
padx=5)
        self.txtSDT = tk.Entry(edit_frame); self.txtSDT.grid(row=1,
column=3)

        tk.Label(edit_frame, text="Điểm:").grid(row=1, column=4,
padx=5)
        self.txtDiem = tk.Entry(edit_frame);
self.txtDiem.grid(row=1, column=5)

        # Nút hỗ trợ cập nhật vào Grid (chưa lưu DB)
        tk.Button(edit_frame, text="Lưu vào bảng (Tạm)",
command=self.update_grid_row, bg="lightblue").grid(row=2, column=1,
columnspan=2, pady=10)
        tk.Button(edit_frame, text="Thêm mới vào bảng",
command=self.add_new_to_grid, bg="lightgreen").grid(row=2, column=3,
columnspan=2)
        tk.Label(edit_frame, text="(Sau khi sửa xong bảng, bấm 'Cập
nhật danh sách' ở trên để lưu DB)", fg="red").grid(row=2, column=5)

    def formKhachThanhVien_Load(self):
        self.LayThongTinKH()

    def LayThongTinKH(self):
        self.GridKhachHang.delete(*self.GridKhachHang.get_children()
)

        query = "SELECT * FROM KhachThanhVien"
        try:
            conn = pyodbc.connect(SharedVariables.connectionString)
            cursor = conn.cursor()
            cursor.execute(query)
            for row in cursor.fetchall():
                # Xử lý ngày sinh
                ns_str = ""
                if row[2]: # row[2] là NGAYSINH
                    ns_str = row[2].strftime("%d/%m/%Y")

                self.GridKhachHang.insert("", tk.END,
values=(row[0], row[1], ns_str, row[3], row[4], row[5]))
            conn.close()

```

```

except Exception as ex:
    messagebox.showerror("Lỗi", str(ex))

def on_row_select(self, event):
    selected = self.GridKhachHang.selection()
    if selected:
        vals = self.GridKhachHang.item(selected[0])['values']
        # Đổ dữ liệu xuống các ô text
        self.txtMaKH.delete(0, tk.END); self.txtMaKH.insert(0,
vals[0])
        self.txtHoTen.delete(0, tk.END); self.txtHoTen.insert(0,
vals[1])
        self.txtNgaySinh.delete(0, tk.END);
self.txtNgaySinh.insert(0, vals[2])
        self.txtGioiTinh.delete(0, tk.END);
self.txtGioiTinh.insert(0, vals[3])
        self.txtSDT.delete(0, tk.END); self.txtSDT.insert(0,
vals[4])
        self.txtDiem.delete(0, tk.END); self.txtDiem.insert(0,
vals[5])

    def update_grid_row(self):
        # Cập nhật dòng đang chọn
        selected = self.GridKhachHang.selection()
        if selected:
            self.GridKhachHang.item(selected[0], values=(
                self.txtMaKH.get(), self.txtHoTen.get(),
self.txtNgaySinh.get(),
                self.txtGioiTinh.get(), self.txtSDT.get(),
self.txtDiem.get()
            ))
        else:
            messagebox.showwarning("Chú ý", "Chưa chọn dòng nào để
sửa")

    def add_new_to_grid(self):
        # Thêm dòng mới vào bảng
        self.GridKhachHang.insert("", tk.END, values=(
            self.txtMaKH.get(), self.txtHoTen.get(),
self.txtNgaySinh.get(),
            self.txtGioiTinh.get(), self.txtSDT.get(),
self.txtDiem.get()
        ))

    def btnCapNhat_Click(self):
        self.CapNhatThongTinKH()

    def CapNhatThongTinKH(self):
        try:

```

```

        conn = pyodbc.connect(SharedVariables.connectionString)
        cursor = conn.cursor()

        for item in self.GridKhachHang.get_children():
            vals = self.GridKhachHang.item(item)['values']
            maKH = str(vals[0])
            hoTen = str(vals[1])
            ngaySinhStr = str(vals[2])
            gioiTinh = str(vals[3])
            sdt = str(vals[4])
            diem = int(vals[5]) if vals[5] else 0

            # Parse Date
            ngaySinh = None
            if ngaySinhStr and ngaySinhStr.strip():
                try:
                    ngaySinh = datetime.strptime(ngaySinhStr,
"%d/%m/%Y")
                except: pass

            # Check exist logic
            cursor.execute("SELECT COUNT(*) FROM KhachThanhVien
WHERE MAKH = ?", (maKH,))
            count = cursor.fetchone()[0]

            if count == 0:
                # Insert
                sql = "INSERT INTO KhachThanhVien (MAKH, HOTEN,
NGAYSINH, GIOITINH, SDT, DIEMTICHLUY) VALUES (?, ?, ?, ?, ?, ?)"
                cursor.execute(sql, (maKH, hoTen, ngaySinh,
gioiTinh, sdt, diem))
            else:
                # Update
                sql = "UPDATE KhachThanhVien SET HOTEN=?,
NGAYSINH=?, GIOITINH=?, SDT=?, DIEMTICHLUY=? WHERE MAKH=?"
                cursor.execute(sql, (hoTen, ngaySinh, gioiTinh,
sdt, diem, maKH))

            conn.commit()
            conn.close()
            messagebox.showinfo("Thông báo", "Cập nhật thành công!")
        except Exception as ex:
            messagebox.showerror("Lỗi DB", str(ex))

    def btnReset_Click(self):
        self.LayThongTinKH()

    def btnThoat_Click(self):
        self.destroy()

```

### 3.11. Giao diện danh sách sản phẩm.

```
import tkinter as tk
from tkinter import ttk, messagebox
import pyodbc
from DuLieuChung import SharedVariables

class formSanPham(tk.Toplevel):
    def __init__(self, parent=None):
        super().__init__(parent)
        self.parent = parent
        # List lưu trữ mapping giữa Tên DM (hiển thị) và Mã DM (giá trị)
        # Cấu trúc: [{"id": "DM01", "name": "Cafe"}, ...]
        self.categories_data = []

        self.InitializeComponent()
        self.formSanPham_Load()

    def InitializeComponent(self):
        self.title("formSanPham")
        SharedVariables.set_icon(self)
        self.geometry("1170x520")

        # Label Title
        self.label1 = tk.Label(self, text="SẢN PHẨM", font=("Segoe UI", 25))
        self.label1.place(x=50, y=8)

        # Inputs
        tk.Label(self, text="Danh mục:").place(x=56, y=90)
        self.cboDanhMuc = ttk.Combobox(self, width=30, state="readonly")
        self.cboDanhMuc.place(x=259, y=90)
        # Bind event chọn danh mục
        self.cboDanhMuc.bind("<<ComboboxSelected>>", self.CboDanhMuc_ThayDoiLuaChon)

        tk.Label(self, text="Tên sản phẩm:").place(x=56, y=139)
        self.txtTen = tk.Entry(self, width=30)
        self.txtTen.place(x=259, y=136)

        tk.Label(self, text="Giá bán:").place(x=56, y=188)
        self.txtGiaBan = tk.Entry(self, width=30)
        self.txtGiaBan.place(x=259, y=181)

        # GroupBox Trạng thái (Radio Buttons)
        self.gr1 = tk.LabelFrame(self, text="Trạng thái")
```



```

        self.gr1.place(x=574, y=90, width=326, height=100)

        self.var_status = tk.StringVar(value="Còn bán") # Default
value
        self.rdoCon = tk.Radiobutton(self.gr1, text="Còn bán",
variable=self.var_status, value="Còn bán")
        self.rdoCon.place(x=23, y=30)

        self.rdoHet = tk.Radiobutton(self.gr1, text="Hết",
variable=self.var_status, value="Ngừng bán")
        self.rdoHet.place(x=207, y=30)

        # Buttons
        self.btnThem = tk.Button(self, text="Thêm sản phẩm",
width=20, height=2, command=self.btnThem_Click)
        self.btnThem.place(x=919, y=24)

        self.btnCapNhat = tk.Button(self, text="Cập nhật sản phẩm",
width=20, height=2, command=self.btnCapNhat_Click)
        self.btnCapNhat.place(x=670, y=24)

        self.btnHuy = tk.Button(self, text="Hủy (Reset)", width=20,
height=2, command=self.btnHuy_Click)
        self.btnHuy.place(x=919, y=95)

        self.btnThoat = tk.Button(self, text="Thoát", width=20,
height=2, command=self.btnThoat_Click)
        self.btnThoat.place(x=919, y=167)

        # GridSanPham (Treeview)
        cols = ("MASP", "TENSP", "GIABAN", "MADM", "TRANGTHAI")
        self.GridSanPham = ttk.Treeview(self, columns=cols,
show='headings')

        self.GridSanPham.heading("MASP", text="Mã SP")
        self.GridSanPham.heading("TENSP", text="Tên SP")
        self.GridSanPham.heading("GIABAN", text="Giá Bán")
        self.GridSanPham.heading("MADM", text="Mã danh mục")
        self.GridSanPham.heading("TRANGTHAI", text="Trạng Thái")

        self.GridSanPham.column("MASP", width=150)
        self.GridSanPham.column("TENSP", width=300)
        self.GridSanPham.column("GIABAN", width=200) # Rộng hơn xít
để hiển thị số tiền
        self.GridSanPham.column("MADM", width=150)
        self.GridSanPham.column("TRANGTHAI", width=200)

        self.GridSanPham.place(x=12, y=242, width=1150, height=253)

```

```

        # Bind sự kiện chọn dòng
        self.GridSanPham.bind("<<TreeviewSelect>>",
self.GridSanPham_CellClick)
        # Bind click vùng trống để bỏ chọn
        self.GridSanPham.bind("<Button-1>", self.check_empty_click)

    def formSanPham_Load(self):
        self.btnCapNhat['state'] = 'disabled'
        self.btnThem['state'] = 'normal'
        self.SetDanhMuc()
        # Mặc định chọn danh mục đầu tiên nếu có
        if self.categories_data:
            self.cboDanhMuc.current(0)
            self.NapDuLieu()

    def SetDanhMuc(self):
        query = "SELECT * FROM DANHMUC"
        self.categories_data = []
        try:
            conn = pyodbc.connect(SharedVariables.connectionString)
            cursor = conn.cursor()
            cursor.execute(query)
            for row in cursor.fetchall():
                # row[0]: MADM, row[1]: TENDANHMUC
                self.categories_data.append({"id": row[0], "name":
row[1]})

            # Set values hiển thị cho ComboBox
            display_names = [item["name"] for item in
self.categories_data]
            self.cboDanhMuc['values'] = display_names

            conn.close()
        except Exception as ex:
            messagebox.showerror("Lỗi", "Lỗi load danh mục: " +
str(ex))

    def get_selected_madm(self):
        # Helper: Lấy MADM từ tên đang chọn trong ComboBox
        selected_name = self.cboDanhMuc.get()
        for item in self.categories_data:
            if item["name"] == selected_name:
                return item["id"]
        return None

    def NapDuLieu(self): # Nạp dữ liệu sản phẩm theo danh mục đã
chọn
        madm = self.get_selected_madm()
        if not madm: return

```

```

        query = "SELECT * FROM SANPHAM WHERE MADM = ?"
        try:
            self.GridSanPham.delete(*self.GridSanPham.get_children()
)

            conn = pyodbc.connect(SharedVariables.connectionString)
            cursor = conn.cursor()
            cursor.execute(query, (madm,))

            for row in cursor.fetchall():
                # row indices: 0=MASP, 1=TENSP, 2=GIABAN, 3=MADM,
                4=TRANGTHAI (Check DB thực tế)
                # Format Giá bán
                gia = f"{row[2]:,.0f}" if row[2] else "0"
                self.GridSanPham.insert("", tk.END, values=(row[0],
row[1], gia, row[3], row[4]))

            conn.close()
        except Exception as ex:
            messagebox.showerror("Lỗi", "Lỗi kết nối CSDL: " +
str(ex))

    def CboDanhMuc_ThayDoiLuaChon(self, event):
        self.NapDuLieu()

    def check_empty_click(self, event):
        # Nếu click vào vùng không có item, Treeview trả về item
        rỗng
        item = self.GridSanPham.identify_row(event.y)
        if not item:
            self.reset_form_state()

    def reset_form_state(self):
        # Hủy chọn, reset inputs
        if self.GridSanPham.selection():
            self.GridSanPham.selection_remove(self.GridSanPham.selec
tion())

        self.txtTen.delete(0, tk.END)
        self.txtGiaBan.delete(0, tk.END)
        self.var_status.set("Còn bán") # Reset radio về mặc định

        self.btnThem['state'] = 'normal'
        self.btnCapNhat['state'] = 'disabled'

    def GridSanPham_CellClick(self, event):
        selected = self.GridSanPham.selection()
        if not selected: return

```

```

        item = self.GridSanPham.item(selected[0])
        vals = item['values']

        # Đổ dữ liệu vào Inputs
        self.txtTen.delete(0, tk.END); self.txtTen.insert(0,
vals[1])

        # Xử lý giá bán (xóa dấu phẩy để hiển thị raw)
        gia_clean = str(vals[2]).replace(",", "")
        self.txtGiaBan.delete(0, tk.END); self.txtGiaBan.insert(0,
gia_clean)

        # Radio Button
        trang_thai = vals[4]
        # Logic check chuỗi "Còn bán" hay "Hết"
        # (C# logic: rdoCon.Checked = true if "Còn bán", else
rdoHet)
        if "Còn bán" in str(trang_thai):
            self.var_status.set("Còn bán")
        else:
            self.var_status.set("Hết")

        self.btnThem['state'] = 'disabled'
        self.btnCapNhat['state'] = 'normal'

    def TaoMASP(self):
        masp = "SP001"
        query = "SELECT TOP 1 MASP FROM SANPHAM ORDER BY MASP DESC"
        try:
            conn = pyodbc.connect(SharedVariables.connectionString)
            cursor = conn.cursor()
            cursor.execute(query)
            row = cursor.fetchone()
            if row:
                last_id = row[0] # VD: SP005
                num = int(last_id[2:]) + 1
                masp = "SP" + f"{num:03d}"
            conn.close()
        except: pass
        return masp

    def btnThem_Click(self):
        self.ThemSanPham()
        self.txtGiaBan.delete(0, tk.END)
        self.txtTen.delete(0, tk.END)

    def ThemSanPham(self):
        masp = self.TaoMASP()
        tensp = self.txtTen.get()

```

```

giaban_str = self.txtGiaBan.get()

# Validate Giá bán
try:
    giaban = float(giaban_str)
except:
    messagebox.showerror("Lỗi", "Giá bán không hợp lệ.")
    return

madm = self.get_selected_madm()
trangthai = self.var_status.get()

sql = "INSERT INTO SANPHAM (MASP, TENS, GIABAN, MADM,
TRANGTHAI) VALUES (?, ?, ?, ?, ?)"

try:
    conn = pyodbc.connect(SharedVariables.connectionString)
    cursor = conn.cursor()
    cursor.execute(sql, (masp, tensp, giaban, madm,
trangthai))
    conn.commit()
    conn.close()

    messagebox.showinfo("Thông báo", "Thêm sản phẩm thành
công!")

    self.NapDuLieu()
except Exception as ex:
    messagebox.showerror("Lỗi", "Lỗi thêm SP: " + str(ex))

def btnCapNhat_Click(self):
    self.CapNhatSanPham()
    self.txtGiaBan.delete(0, tk.END)
    self.txtTen.delete(0, tk.END)
    self.reset_form_state() # Reset về trạng thái thêm

def CapNhatSanPham(self): #Cập nhật sản phẩm
    selected = self.GridSanPham.selection()
    if not selected: return # Không có dòng nào được chọn

    # Lấy MASP từ dòng đang chọn (không cho sửa MASP)
    masp = self.GridSanPham.item(selected[0])['values'][0]

    tensp = self.txtTen.get()
    giaban_str = self.txtGiaBan.get()
    try:
        giaban = float(giaban_str)
    except:
        messagebox.showerror("Lỗi", "Giá bán không hợp lệ.")
    return

```

```

        madm = self.get_selected_madm()
        trangthai = self.var_status.get()

        sql = "UPDATE SANPHAM SET TENSP=?, GIABAN=?, MADM=?,
TRANGTHAI=? WHERE MASP=?"

        try:
            conn = pyodbc.connect(SharedVariables.connectionString)
            cursor = conn.cursor()
            cursor.execute(sql, (tensp, giaban, madm, trangthai,
masp))

            conn.commit()
            conn.close()

            messagebox.showinfo("Thông báo", "Cập nhật sản phẩm
thành công!")
            self.NapDuLieu()
        except Exception as ex:
            messagebox.showerror("Lỗi", "Lỗi cập nhật SP: " +
str(ex))

    def btnHuy_Click(self):
        self.reset_form_state()

    def btnThoat_Click(self):
        self.destroy()

```