**TRƯỜNG ĐẠI HỌC AN GIANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**A yellow circle with blue and green text

AI-generated content may be incorrect.**

**BÁO CÁO ĐỒ ÁN**

**Chuyên đề Python (COS525)**

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ CỬA HÀNG TIVI VỚI PYTHON, TKINTER VÀ SQL SERVER**

**GIẢNG VIÊN HƯỚNG DẪN**

**ThS. NGUYỄN NGỌC MINH**

**AN GIANG, THÁNG 11 NĂM 2025**

**BẢNG PHÂN CÔNG CÔNG VIỆC**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ tên thành viên** | **Mã số sinh viên** | **Công việc thực hiện** | **Tỷ lệ đóng góp (%)** |
| 1 | Phan Thanh Hoài | DTH235658 | Thiết kế giao diện, from Nhập hàng và Bán hàng, tab Nhập hàng, tab Bán hàng, form Bán Hàng và Hóa đơn, tab Bán hàng, tab Hóa đơn, kết nối CSDL, form App, kiểm thử, sửa lỗi, nội dung báo cáo, file word. | 50% |
| 2 | Nguyễn Văn Hiền | DTH235651 | Thiết kế CSDL, form Quản lý sản phẩm, tab Tivi, tab Nhà cung cấp, tab Hãng sản xuất, tab Bảo hành, form Quản lý nhân viên, from Quản lý khách hàng, form Login, form Hệ thống, form Thống kê và báo cáo, tab Báo cáo sản phẩm, tab Thống kê doanh thu. | 50% |

**LỜI NÓI ĐẦU**

Trong bối cảnh nền kinh tế số phát triển mạnh mẽ, việc chuyển đổi số và tối ưu hóa quy trình quản lý thông qua các giải pháp phần mềm là yếu tố then chốt để duy trì tính cạnh tranh, đặc biệt đối với các mô hình bán lẻ thiết bị điện tử. Điều này đặt ra nhu cầu cấp thiết về một hệ thống quản lý có khả năng tự động hóa toàn diện, bao gồm không chỉ các nghiệp vụ cơ bản (nhập, xuất, tồn kho) mà còn cả các quy trình phức tạp (bảo hành, thống kê lợi nhuận). Với mục tiêu vận dụng và tổng hợp các kiến thức chuyên ngành đã tích lũy (đặc biệt là ngôn ngữ lập trình Python, thư viện giao diện Tkinter và hệ quản trị cơ sở dữ liệu SQL Server), chúng em đã triển khai đồ án: **"XÂY DỰNG ỨNG DỤNG QUẢN LÝ CỬA HÀNG TIVI VỚI PYTHON, TKINTER VÀ SQL SERVER"**.

Đồ án không chỉ dừng lại ở một ứng dụng CRUD thông thường, mà còn đi sâu vào giải quyết các thách thức nghiệp vụ thực tế. Hệ thống được xây dựng trên kiến trúc module hóa chặt chẽ, cho phép xử lý đồng thời nhiều luồng dữ liệu, tiêu biểu như việc lưu trữ ảnh sản phẩm/nhân viên dưới dạng BLOB, tính toán lợi nhuận gộp chi tiết, và trực quan hóa doanh thu bằng biểu đồ. Đặc biệt, chúng em đã tích hợp thành công tính năng In ấn tự động ra file Word chuyên nghiệp cho hóa đơn, phiếu nhập và bảo hành, qua đó nâng cao đáng kể tính tiện dụng và khả năng ứng dụng thực tế của sản phẩm. Tất cả là kết quả của quá trình nghiên cứu, thiết kế hệ thống, và triển khai lập trình tỉ mỉ.

Chúng em xin chân thành cảm ơn ThS. Nguyễn Ngọc Minh – Giảng viên hướng dẫn, người đã cung cấp những định hướng chuyên môn sâu sắc và những góp ý kịp thời, giúp chúng em vượt qua nhiều khó khăn về kỹ thuật. Chúng em hiểu rằng, sản phẩm này là bước khởi đầu và vẫn không thể tránh khỏi những thiếu sót nhất định về mặt tối ưu hóa và trải nghiệm người dùng. Vì vậy, chúng em rất mong nhận được những ý kiến đóng góp quý báu từ thầy để đồ án được hoàn thiện hơn nữa.

**PHỤ LỤC**

[DANH SÁCH HÌNH ẢNH…………………………………………………...2](#_Toc214227793)

[I. Tính cần thiết của đề tài 4](#_Toc214227794)

[II. Mục tiêu nghiên cứu của đề tài 4](#_Toc214227795)

[CHƯƠNG 2: TỔNG QUAN VÀ CƠ SỞ LÝ THUYẾT……………………6](#_Toc214227796)

[I. Đặt vấn đề 6](#_Toc214227797)

[II. Lịch sử giải quyết vấn đề 6](#_Toc214227798)

[III. Phạm vi của đề tài 6](#_Toc214227799)

[IV. Phương pháp nghiên cứu và hướng giải quyết 7](#_Toc214227800)

[V. Cơ sở lý thuyết 7](#_Toc214227801)

[CHƯƠNG 3: TỔNG QUAN VÀ CƠ SỞ LÝ THUYẾT……………………8](#_Toc214227802)

[**A. THIẾT KẾ GIAO DIỆN QUẢN LÝ CỬA HÀNG TIVI** 8](#_Toc214227803)

[1. Giao diện dành cho người quản trị (admin) 8](#_Toc214227804)

[2. Giao diện dành cho nhân viên 17](#_Toc214227805)

[**B. CÁC BƯỚC XÂY DỰNG ỨNG DỤNG** 21](#_Toc214227806)

[1. Cài đặt thư viện cần thiết 21](#_Toc214227807)

[2. Thiết kế cơ sở dữ liệu SQL Server 22](#_Toc214227808)

[3. Xây dựng giao diện người dùng (GUI) bằng Tkinter 23](#_Toc214227809)

[4. Viết mã Python kết nối cơ sở dữ liệu 27](#_Toc214227810)

[5. Xây dựng giao diện Tkinter 32](#_Toc214227811)

[**C. HƯỚNG DẪN CÀI ĐẶT & CODE ỨNG DỤNG** 46](#_Toc214227812)

[1. Cài đặt môi trường 46](#_Toc214227813)

[2. Chuẩn bị CSDL SQL Server 48](#_Toc214227814)

[3. Viết code ứng dụng với Tkinter + SQL Server 51](#_Toc214227815)

[**DANH MỤC TÀI LIỆU THAM KHẢO** 199](#_Toc214227816)

# **DANH SÁCH HÌNH ẢNH**

[Hình 1: Giao diện đăng nhập 8](#_Toc214227457)

[Hình 2: Giao diện tổng quan 9](#_Toc214227458)

[Hình 3: Giao diện quản lý sản phẩm - tab tivi 9](#_Toc214227459)

[Hình 4: Giao diện quản lý sản phẩm – tab hãng sản xuất 10](#_Toc214227460)

[Hình 5: Giao diện quản lý sản phẩm – tab nhà cung cấp 10](#_Toc214227461)

[Hình 6: Giao diện quản lý sản phẩm – tab bảo hành 11](#_Toc214227462)

[Hình 7: Giao diện quản lý khách hàng 11](#_Toc214227463)

[Hình 8: Giao diện quản lý nhân viên 12](#_Toc214227464)

[Hình 9: Giao diện quản lý bán hàng và hóa đơn – tab bán hàng 12](#_Toc214227465)

[Hình 10: Giao diện quản lý bán hàng và hóa đơn – tab hóa đơn 13](#_Toc214227466)

[Hình 11: Giao diện quản lý bán hàng và hóa đơn – xem chi tiết hóa đơn 13](#_Toc214227467)

[Hình 12: Giao diện quản lý nhập hàng và phiếu nhập – tab nhập hàng 14](#_Toc214227468)

[Hình 13: Giao diện quản lý nhập hàng và phiếu nhập – tab phiếu nhập hàng 14](#_Toc214227469)

[Hình 14: Giao diện quản lý nhập hàng và phiếu nhập – xem chi tiết phiếu nhập hàng 15](#_Toc214227470)

[Hình 15: Giao diện quản lý thống kê và báo cáo – tab báo cáo sản phẩm 15](#_Toc214227471)

[Hình 16: Giao diện quản lý thống kê và báo cáo – tab thống kê doanh thu 16](#_Toc214227472)

[Hình 17: Giao diện quản lý tài khoản hệ thống 16](#_Toc214227473)

[Hình 18: Giao diện đăng nhập của nhân viên 17](#_Toc214227474)

[Hình 19: Giao diện quản lý nhân viên của nhân viên 17](#_Toc214227475)

[Hình 20: Giao diện quản lý sản phẩm – tab Tivi của nhân viên 18](#_Toc214227476)

[Hình 21: Giao diện quản lý sản phẩm – tab bảo hành của nhân viên 18](#_Toc214227477)

[Hình 22: Giao diện quản lý sản phẩm – tab Tivi của nhân viên 19](#_Toc214227478)

[Hình 23: Giao diện quản lý bán hàng và hóa đơn – tab bán hàng của nhân viên 19](#_Toc214227479)

[Hình 24: Giao diện quản lý bán hàng và hóa đơn – tab hóa đơn của nhân viên 20](#_Toc214227480)

[Hình 25: Giao diện quản lý bán hàng và hóa đơn – xem chi tiết hóa đơn của nhân viên ………………………………………………………………………...20](#_Toc214227481)

[Hình 26: Giao diện quản lý tài khoản hệ thống của nhân viên 21](#_Toc214227482)

[Hình 27: Diagram liên kết giữa các bảng 22](#_Toc214227483)

[Hình 28: Sơ đồ usecase – Hệ thống Quản lý cửa hành tivi 35](#_Toc214227484)

**CHƯƠNG 1: ĐẶT VẤN ĐỀ**

## **I. Tính cần thiết của đề tài**

Trong bối cảnh công nghệ thông tin phát triển mạnh mẽ, việc ứng dụng các phần mềm quản lý vào hoạt động kinh doanh đã trở thành nhu cầu thiết yếu của các cửa hàng bán lẻ, đặc biệt là các cửa hàng kinh doanh tivi và thiết bị điện tử. Tuy nhiên, phần lớn các hệ thống quản lý hiện có còn tồn tại nhiều hạn chế. Nhiều đề tài trước đây chỉ triển khai trên Excel hoặc Access, khả năng mở rộng thấp, dữ liệu không đồng bộ và thiếu tính bảo mật. Một số đề tài khác có xây dựng ứng dụng bán hàng nhưng lại chưa tích hợp đầy đủ các chức năng như quản lý sản phẩm, khách hàng, nhân viên, hóa đơn, báo cáo và tồn kho trong cùng một hệ thống. Ngoài ra, rất ít đề tài khai thác sự kết hợp giữa **Python – Tkinter – SQL Server**, dẫn đến việc thiếu tính trực quan ở giao diện và thiếu tính ổn định ở cơ sở dữ liệu.

Chính vì vậy, việc xây dựng một ứng dụng quản lý cửa hàng tivi sử dụng Python làm ngôn ngữ lập trình, Tkinter làm giao diện và SQL Server làm cơ sở dữ liệu là thật sự cần thiết. Đề tài không chỉ đáp ứng nhu cầu thực tiễn về quản lý cửa hàng mà còn mang ý nghĩa lý luận khi giúp người học củng cố kỹ năng lập trình hướng đối tượng, thiết kế giao diện và thao tác dữ liệu. Về mặt thực tiễn, hệ thống giúp tối ưu hóa quá trình quản lý sản phẩm, bán hàng, nhập hàng và thống kê với tính chính xác cao, góp phần hỗ trợ quyết định kinh doanh hiệu quả.

Tính cần thiết của đề tài cũng thể hiện qua việc nó cung cấp cơ sở để cụ thể hóa các mục tiêu phát triển phần mềm gọn nhẹ, ổn định, dễ sử dụng và phù hợp với mô hình cửa hàng quy mô nhỏ đến trung bình. Từ đó, đề tài định hướng các nội dung cần thực hiện như: thiết kế cơ sở dữ liệu hợp lý; xây dựng giao diện trực quan; triển khai các chức năng quản lý sản phẩm, nhân viên, khách hàng, hóa đơn; chức năng nhập hàng – tồn kho; và các báo cáo hỗ trợ hoạt động kinh doanh.

## **II. Mục tiêu nghiên cứu của đề tài**

Mục tiêu của đề tài là xây dựng một ứng dụng quản lý cửa hàng tivi đầy đủ chức năng, dễ sử dụng và phù hợp với nhu cầu thực tiễn của các cửa hàng điện tử. Ứng dụng phải hỗ trợ người dùng quản lý thông tin sản phẩm, khách hàng, nhân viên, hóa đơn bán hàng, nhập hàng và báo cáo thống kê một cách hiệu quả. Giao diện chương trình cần được thiết kế đơn giản, trực quan, giúp người sử dụng thao tác dễ dàng và hạn chế sai sót trong quá trình nhập liệu.

Bên cạnh mục tiêu về tính tiện dụng, đề tài còn hướng đến đảm bảo khả năng vận hành ổn định và độ chính xác trong xử lý dữ liệu. Hệ thống phải có khả năng quản lý số lượng dữ liệu nhất định, như tối thiểu 100 sản phẩm, 50 khách hàng và 10 nhân viên, mà không gây chậm trễ hoặc lỗi phát sinh. Các phép tính liên quan đến tổng tiền, số lượng bán và số lượng tồn kho phải được thực hiện chính xác, đảm bảo tính tin cậy cho quá trình kinh doanh.

Ngoài ra, đề tài đặt mục tiêu áp dụng hiệu quả các công nghệ như Python để xây dựng chức năng, Tkinter để thiết kế giao diện và SQL Server để lưu trữ cơ sở dữ liệu. Việc kết nối và xử lý dữ liệu thông qua thư viện pyodbc phải đảm bảo tính ổn định, an toàn và phù hợp với môi trường sử dụng thực tế trên hệ điều hành Windows. Những mục tiêu này đều mang tính cụ thể, đo lường được, có tính khả thi và phù hợp với phạm vi của một đồ án môn học.

# **CHƯƠNG 2: TỔNG QUAN VÀ CƠ SỞ LÝ THUYẾT**

## **I. Đặt vấn đề**

Trong hoạt động kinh doanh hiện đại, các cửa hàng bán lẻ, đặc biệt là cửa hàng tivi và thiết bị điện tử, đòi hỏi sự quản lý chặt chẽ và chính xác về hàng hóa, nhân viên, khách hàng và hóa đơn. Khi quy mô cửa hàng ngày càng mở rộng, mô hình quản lý thủ công bằng giấy tờ hoặc bảng tính Excel trở nên lỗi thời, dễ xảy ra nhầm lẫn, mất dữ liệu và khó đáp ứng nhu cầu theo dõi thường xuyên. Do đó, việc xây dựng một ứng dụng quản lý cửa hàng tivi với giao diện trực quan, đảm bảo tính ổn định, tính bảo mật và xử lý dữ liệu nhanh chóng là yêu cầu cấp thiết. Bài toán đặt ra trong đề tài là thiết kế và xây dựng một hệ thống phần mềm có khả năng hỗ trợ toàn bộ quy trình quản lý của một cửa hàng tivi, từ cập nhật thông tin sản phẩm đến lập hóa đơn, thống kê và báo cáo.

## **II. Lịch sử giải quyết vấn đề**

Trong những năm gần đây, nhiều đề tài và phần mềm đã được xây dựng để giải quyết bài toán quản lý bán hàng. Các hệ thống đơn giản thường được thiết kế bằng Access hoặc Excel, hỗ trợ một phần nhu cầu quản lý nhưng lại hạn chế về tính mở rộng và bảo mật. Một số ứng dụng khác được phát triển bằng các ngôn ngữ như C#, Java hoặc PHP, tuy có giao diện trực quan hơn nhưng đa phần chỉ tập trung vào một vài chức năng như quản lý sản phẩm hoặc hóa đơn mà chưa tích hợp đầy đủ các chức năng cần thiết cho một cửa hàng tivi.

Ngoài ra, rất ít đề tài áp dụng Python trong việc xây dựng ứng dụng quản lý bán hàng, đặc biệt là việc kết hợp **Python – Tkinter – SQL Server** để tạo ra hệ thống hoàn chỉnh. Phần lớn các ứng dụng Python chỉ dừng lại ở quản lý đơn giản hoặc thông qua giao diện console, thiếu tính trực quan như các phần mềm thương mại. Vì vậy có thể thấy rằng bài toán này đã được nghiên cứu nhưng chưa thật sự được triển khai rộng rãi với mô hình tương tự. Đây được xem là một vấn đề có tính mới, chưa có nhiều hệ thống được xây dựng hoàn chỉnh theo đúng hướng mà đề tài thực hiện.

## **III. Phạm vi của đề tài**

Đề tài tập trung xây dựng một ứng dụng quản lý cửa hàng tivi có quy mô nhỏ đến trung bình. Phạm vi nghiên cứu bao gồm các chức năng chính như quản lý sản phẩm, khách hàng, nhân viên, hóa đơn bán hàng, nhập hàng và báo cáo thống kê. Hệ thống không mở rộng đến các chức năng phức tạp như quản lý chuỗi cửa hàng, kết nối API với nhà cung cấp, phân tích dữ liệu nâng cao hoặc bán hàng trực tuyến. Bên cạnh đó, ứng dụng được thiết kế chạy trên máy tính cá nhân sử dụng hệ điều hành Windows, nên không nhằm mục tiêu triển khai đa nền tảng hoặc trên môi trường web.

## **IV. Phương pháp nghiên cứu và hướng giải quyết**

Để giải quyết bài toán, đề tài sử dụng kết hợp nhiều phương pháp nghiên cứu khác nhau. Phương pháp nghiên cứu lý thuyết được áp dụng để tìm hiểu các kiến thức về lập trình Python, thiết kế giao diện Tkinter, xử lý dữ liệu và mô hình cơ sở dữ liệu quan hệ trong SQL Server. Phương pháp nghiên cứu thực nghiệm được sử dụng để xây dựng và kiểm thử từng chức năng của hệ thống, thông qua việc lập trình, mô phỏng tình huống thực tế và xử lý dữ liệu mẫu. Ngoài ra, phương pháp phân tích – thiết kế hệ thống cũng được áp dụng để xác định yêu cầu, phân tích chức năng, thiết kế cơ sở dữ liệu và xây dựng sơ đồ hoạt động, nhằm đảm bảo hệ thống được triển khai một cách khoa học và hợp lý.

## **V. Cơ sở lý thuyết**

Đề tài dựa trên nhiều cơ sở lý thuyết quan trọng. Trước hết là nền tảng lập trình Python – một ngôn ngữ đơn giản, dễ đọc và hỗ trợ mạnh mẽ trong việc xây dựng ứng dụng. Giao diện của chương trình được phát triển bằng Tkinter, bộ thư viện tích hợp của Python cho phép thiết kế giao diện đồ họa (GUI) với các thành phần như button, label, entry và treeview. Về mặt dữ liệu, SQL Server được sử dụng để lưu trữ và truy vấn thông tin nhờ vào các đặc tính mạnh mẽ như tính bảo mật cao, tốc độ xử lý nhanh và khả năng quản lý dữ liệu ổn định. Ngoài ra, mô hình cơ sở dữ liệu quan hệ cùng các câu lệnh SQL như SELECT, INSERT, UPDATE và DELETE là nền tảng để ứng dụng hoạt động chính xác. Thư viện pyodbc đóng vai trò trung gian kết nối giữa Python và SQL Server, đảm bảo việc trao đổi dữ liệu được diễn ra liên tục và hiệu quả.

Những cơ sở lý thuyết trên được kế thừa từ nhiều tài liệu lập trình và cơ sở dữ liệu đã được nghiên cứu trước đây. Đề tài không xây dựng lý thuyết mới mà chủ yếu vận dụng và kết hợp các lý thuyết hiện có để giải quyết bài toán thực tiễn theo một cách tối ưu và phù hợp với quy mô của cửa hàng tivi.

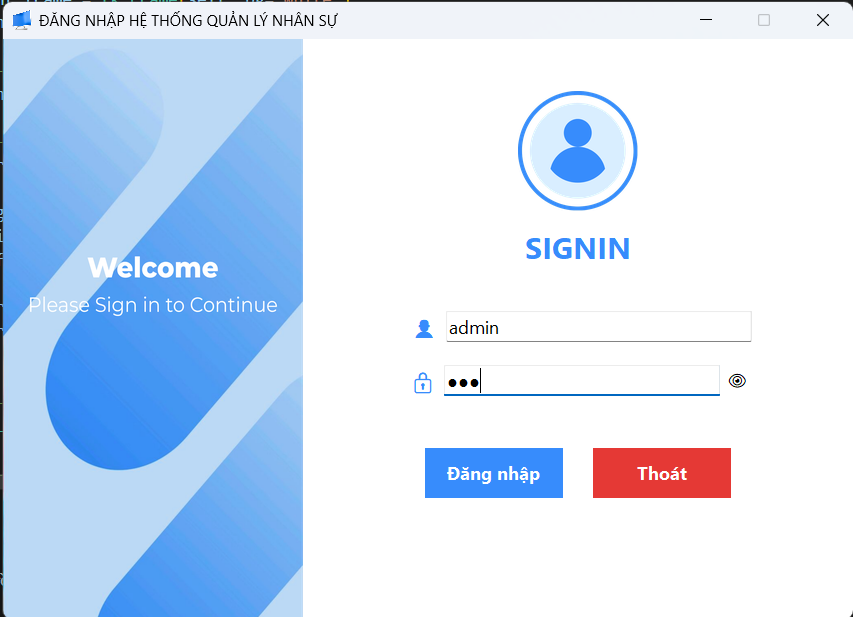
# **CHƯƠNG 3: TỔNG QUAN VÀ CƠ SỞ LÝ THUYẾT**

Trong khuôn khổ đề tài “Quản lý cửa hàng tivi”, chúng em đã xây dựng thành công một ứng dụng quản lý được triển khai bằng thư viện Tkinter của Python. Ứng dụng tập trung giải quyết các nghiệp vụ cốt lõi thường gặp trong hoạt động kinh doanh của một cửa hàng tivi, bao gồm quản lý thông tin sản phẩm, xử lý hóa đơn bán hàng, theo dõi số lượng tồn kho và hỗ trợ nhân viên thực hiện các thao tác cập nhật dữ liệu. Các chức năng như thêm, sửa và xóa thông tin tivi hoặc hóa đơn được thiết kế trực quan trên giao diện, cho phép người dùng thao tác dễ dàng và hạn chế sai sót trong quá trình nhập liệu.

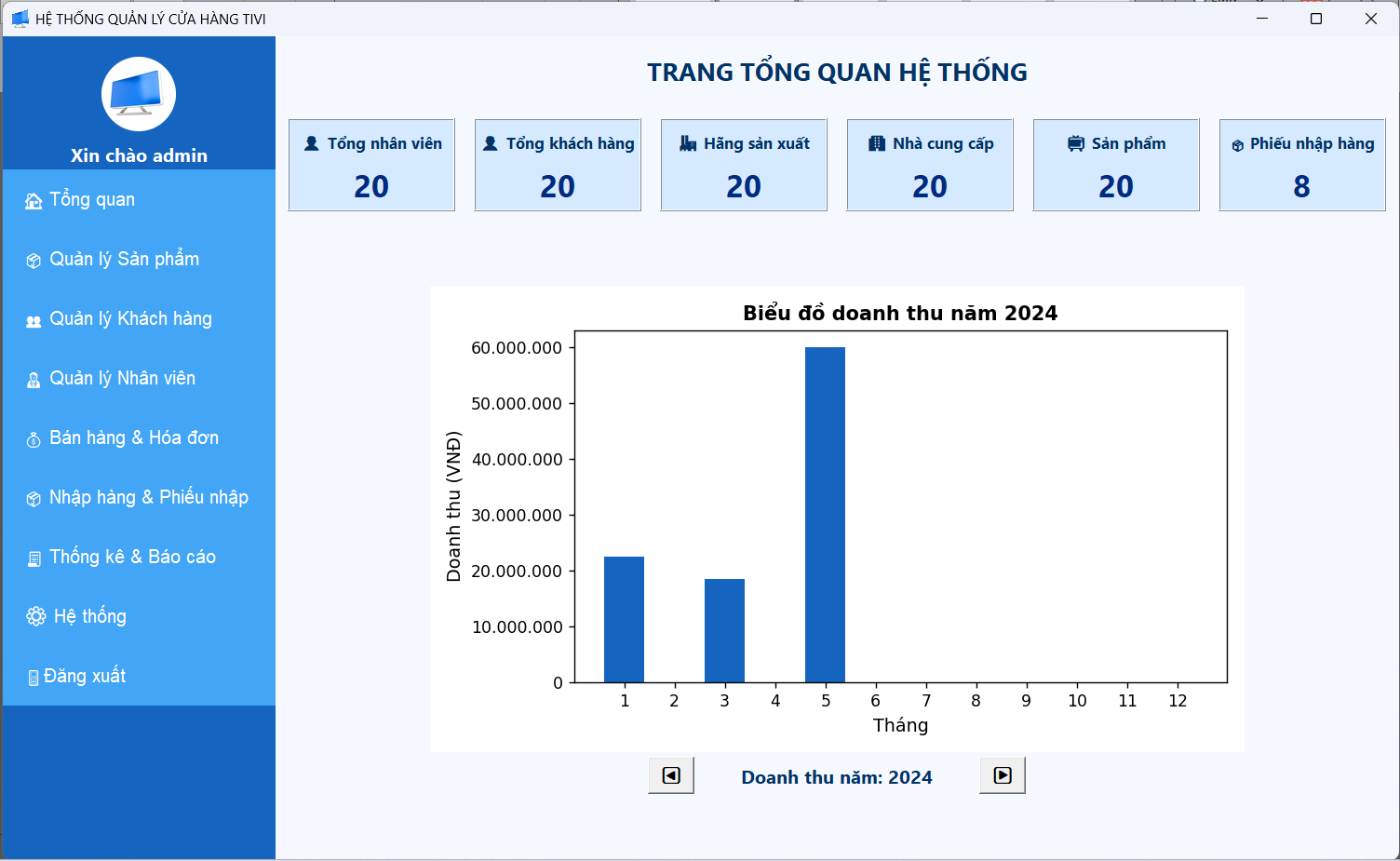
Trong quá trình phát triển ứng dụng, chúng em đã vận dụng nhiều kỹ thuật lập trình Python để mô phỏng các tình huống thực tế của hoạt động kinh doanh. Các cơ chế xử lý sự kiện, quản lý dữ liệu dạng bảng, phân tách chức năng theo từng màn hình giao diện và thao tác với cơ sở dữ liệu được áp dụng một cách hợp lý nhằm bảo đảm tính chính xác và tính ổn định của hệ thống. Việc tích hợp các chức năng tra cứu, kiểm tra dữ liệu trùng lặp hoặc ràng buộc thông tin giúp ứng dụng đáp ứng tốt các yêu cầu thực tiễn của một cửa hàng tivi, nơi số lượng sản phẩm, mẫu mã và hóa đơn thường xuyên thay đổi. Phần demo code đính kèm minh họa cụ thể cho cách triển khai từng chức năng, góp phần khẳng định tính khả thi và hiệu quả của mô hình ứng dụng mà đề tài hướng tới.

## **A. THIẾT KẾ GIAO DIỆN QUẢN LÝ CỬA HÀNG TIVI**

### **1. Giao diện dành cho người quản trị (admin)**



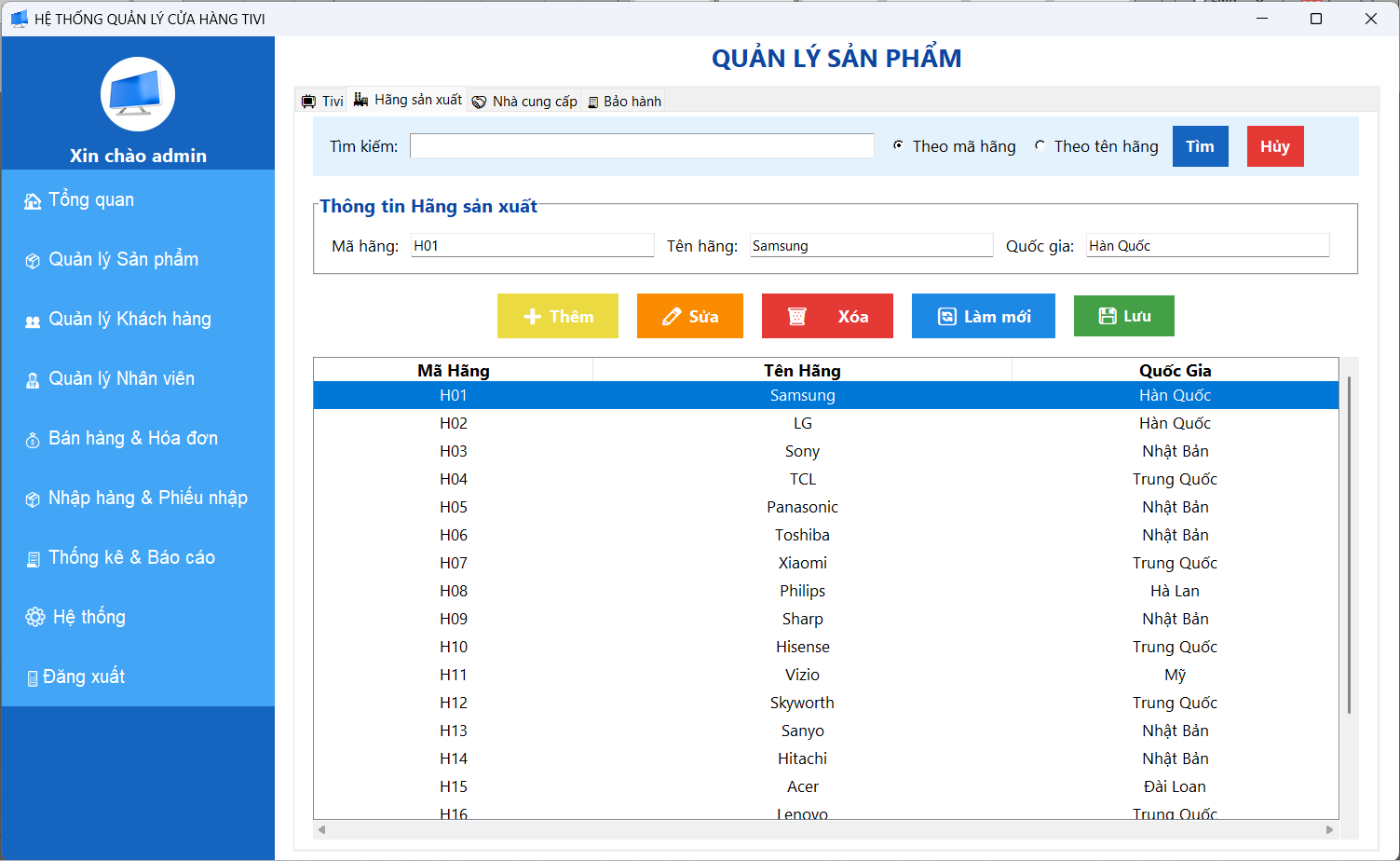
Hình 1: Giao diện đăng nhập



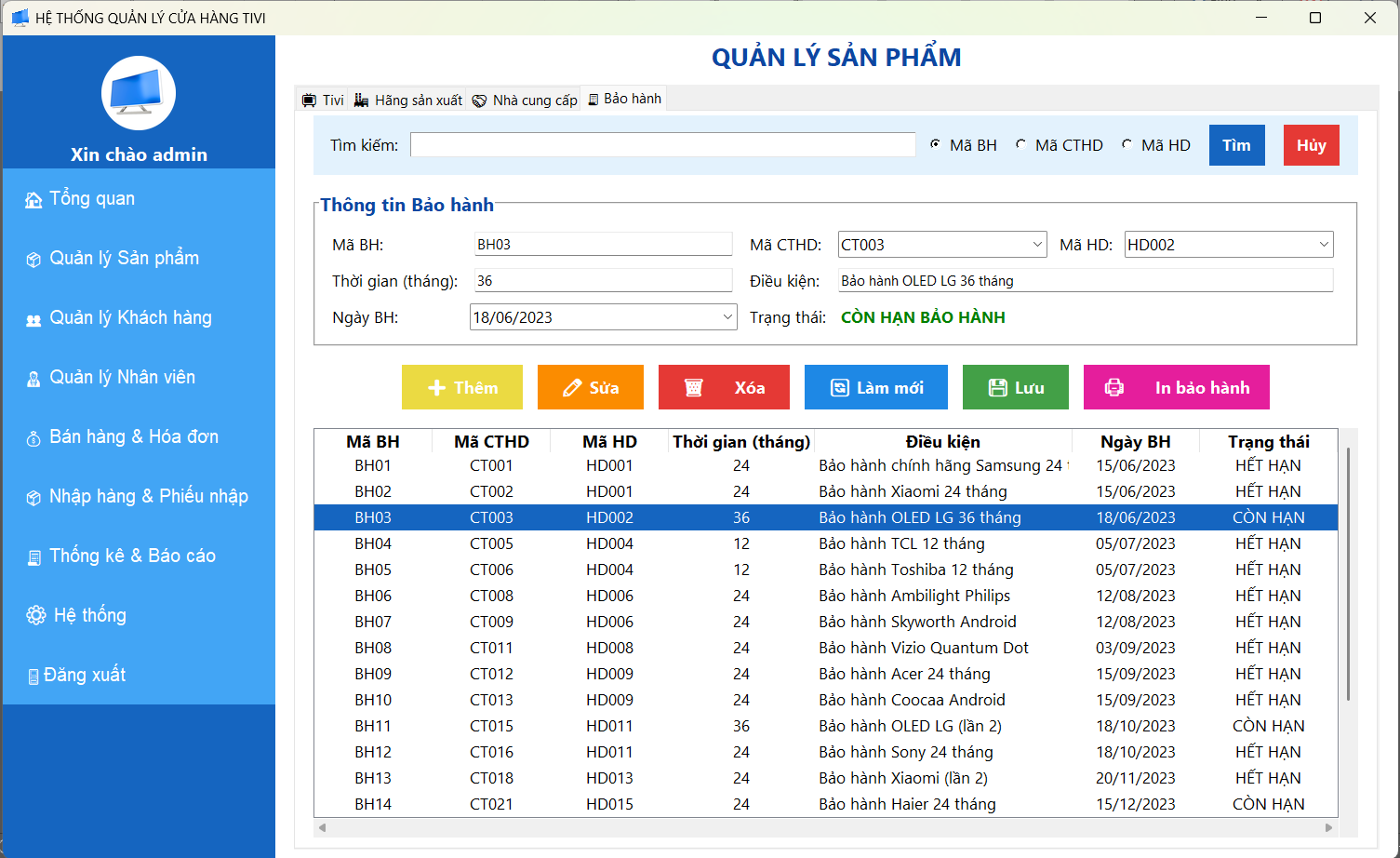
Hình 2: Giao diện tổng quan



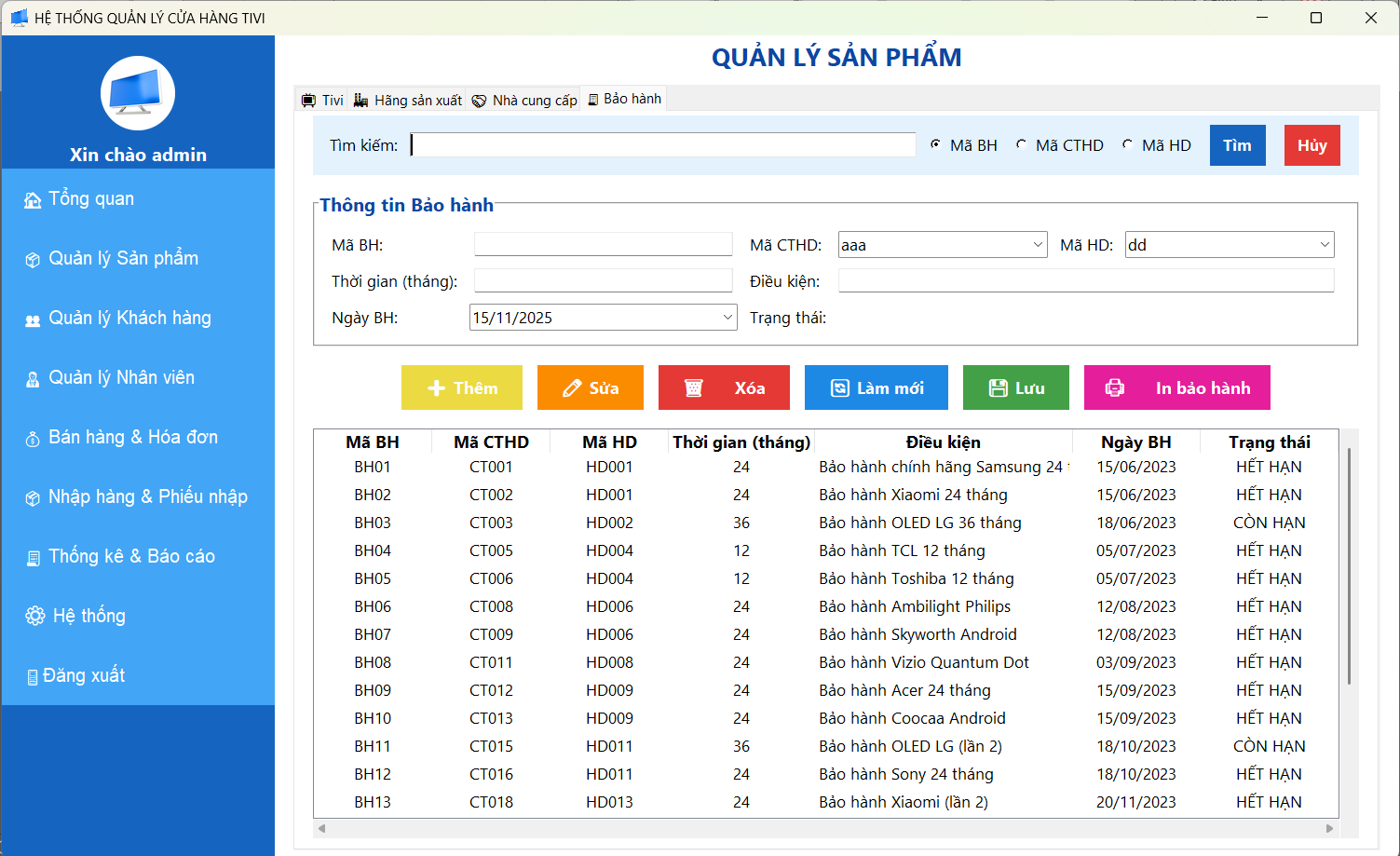
Hình 3: Giao diện quản lý sản phẩm - tab tivi



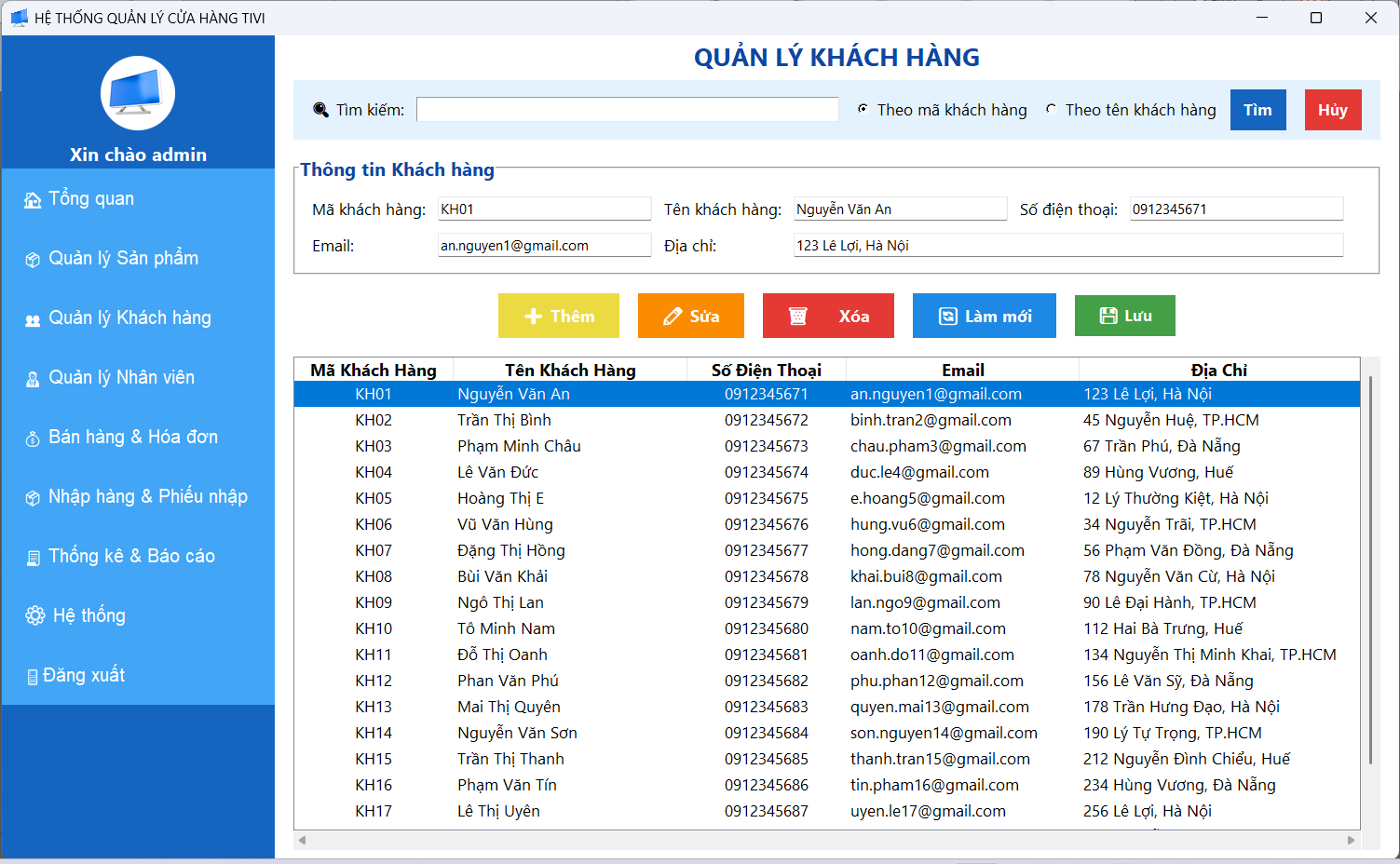
Hình 4: Giao diện quản lý sản phẩm – tab hãng sản xuất



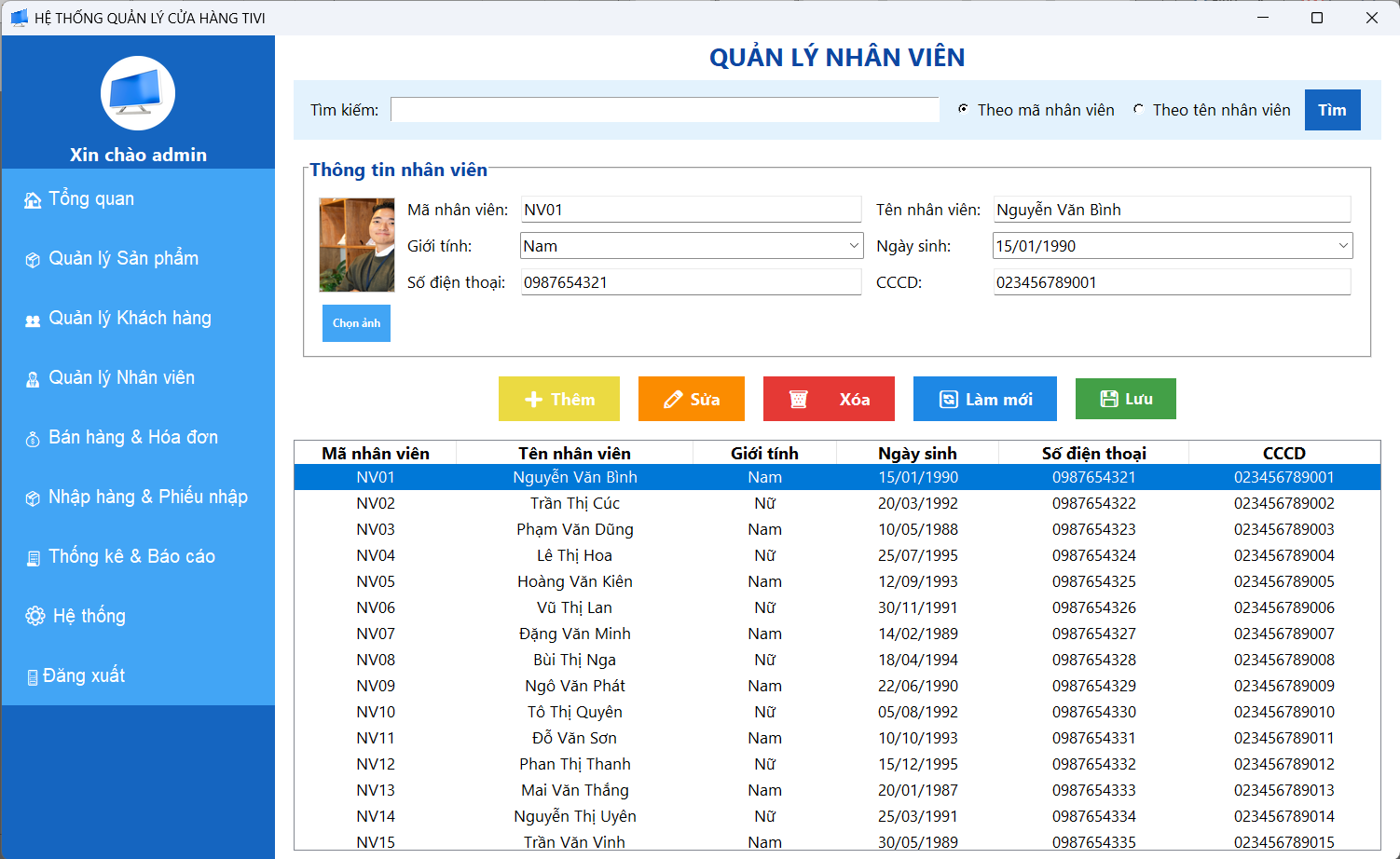
Hình 5: Giao diện quản lý sản phẩm – tab nhà cung cấp



Hình 6: Giao diện quản lý sản phẩm – tab bảo hành



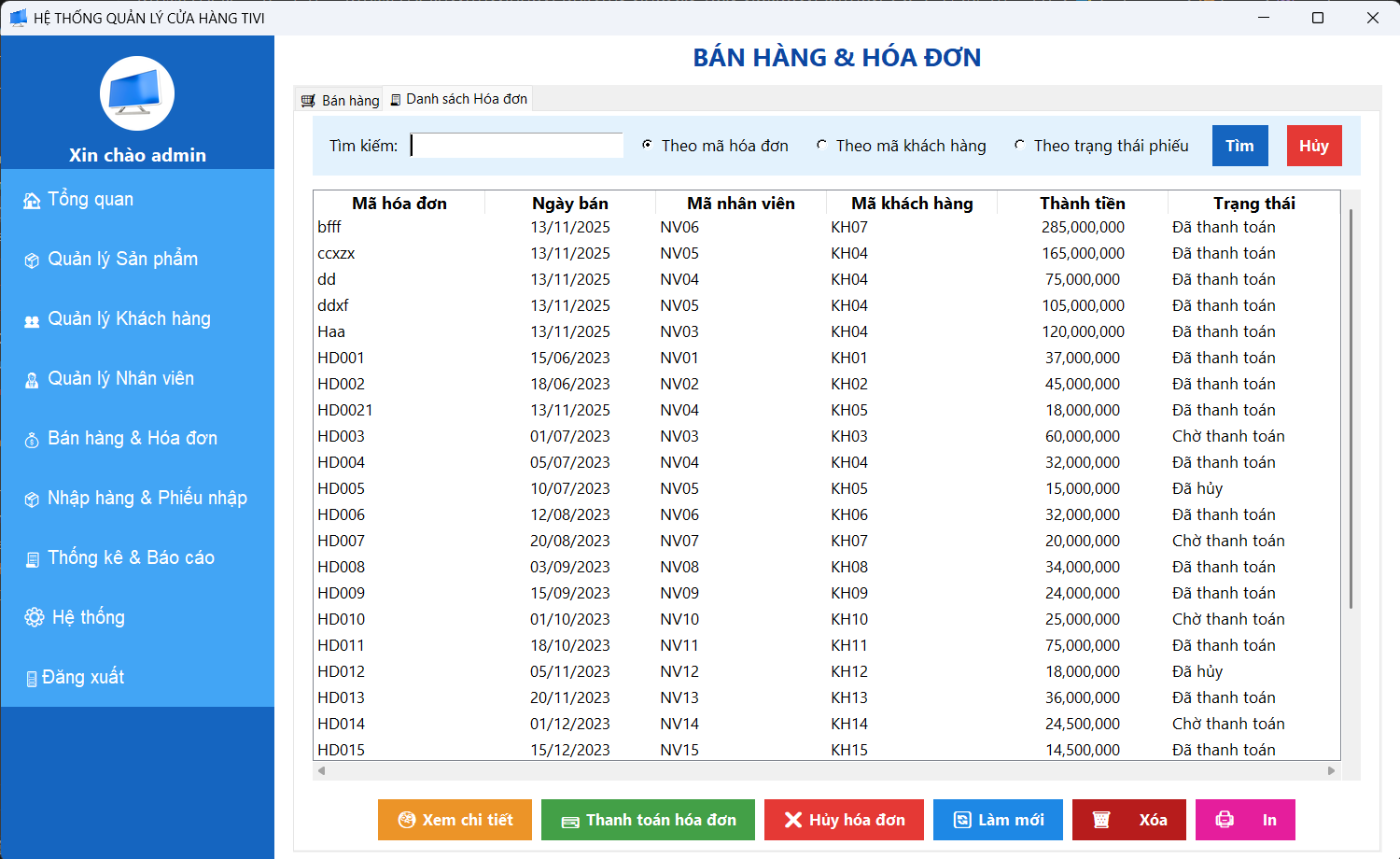
Hình 7: Giao diện quản lý khách hàng



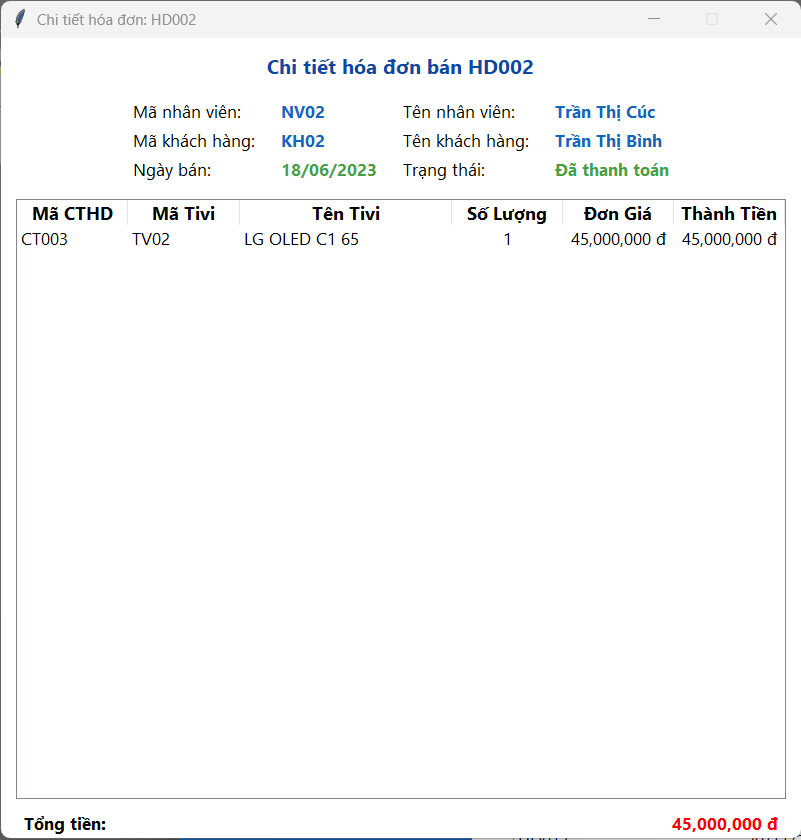
Hình 8: Giao diện quản lý nhân viên



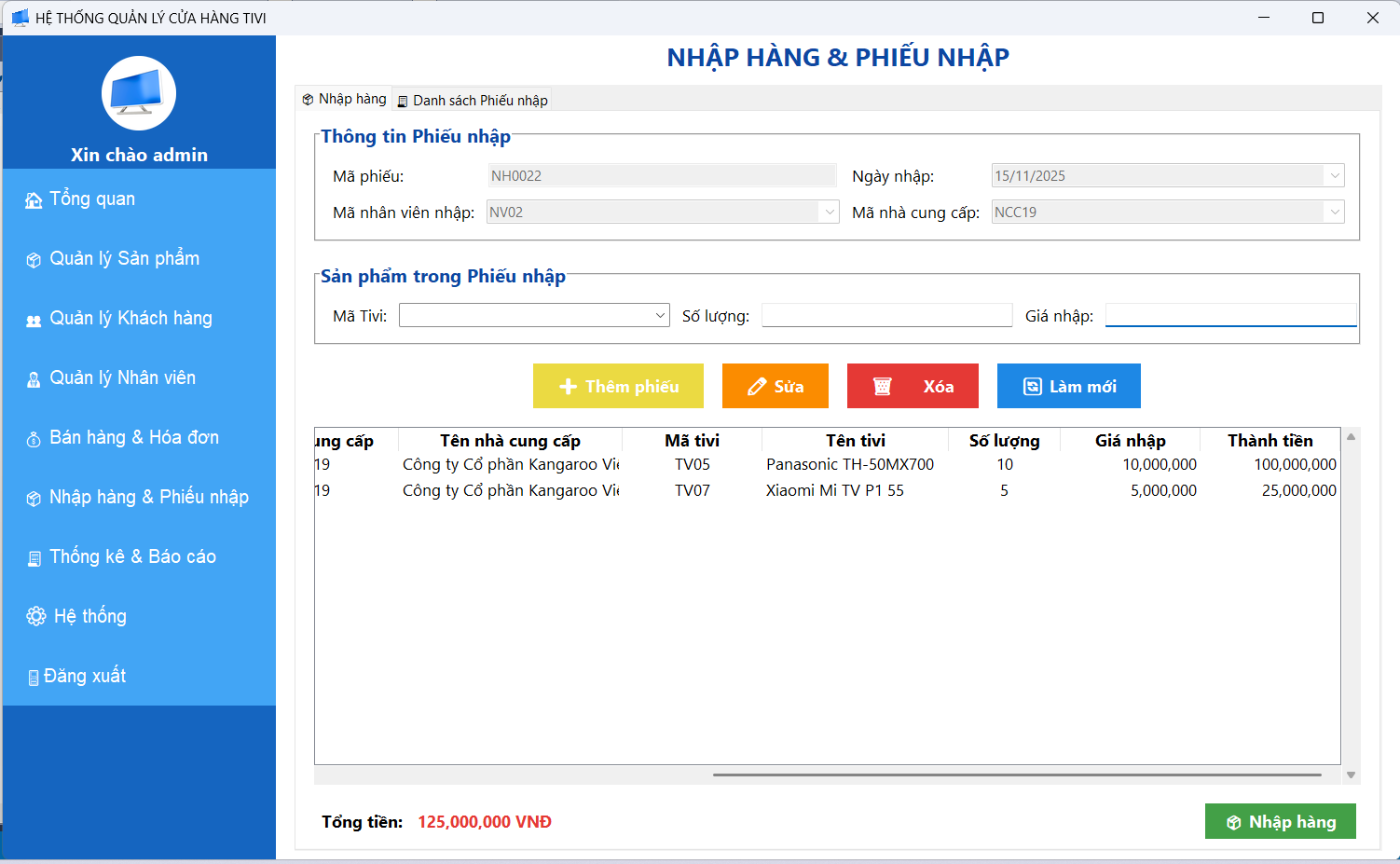
Hình 9: Giao diện quản lý bán hàng và hóa đơn – tab bán hàng



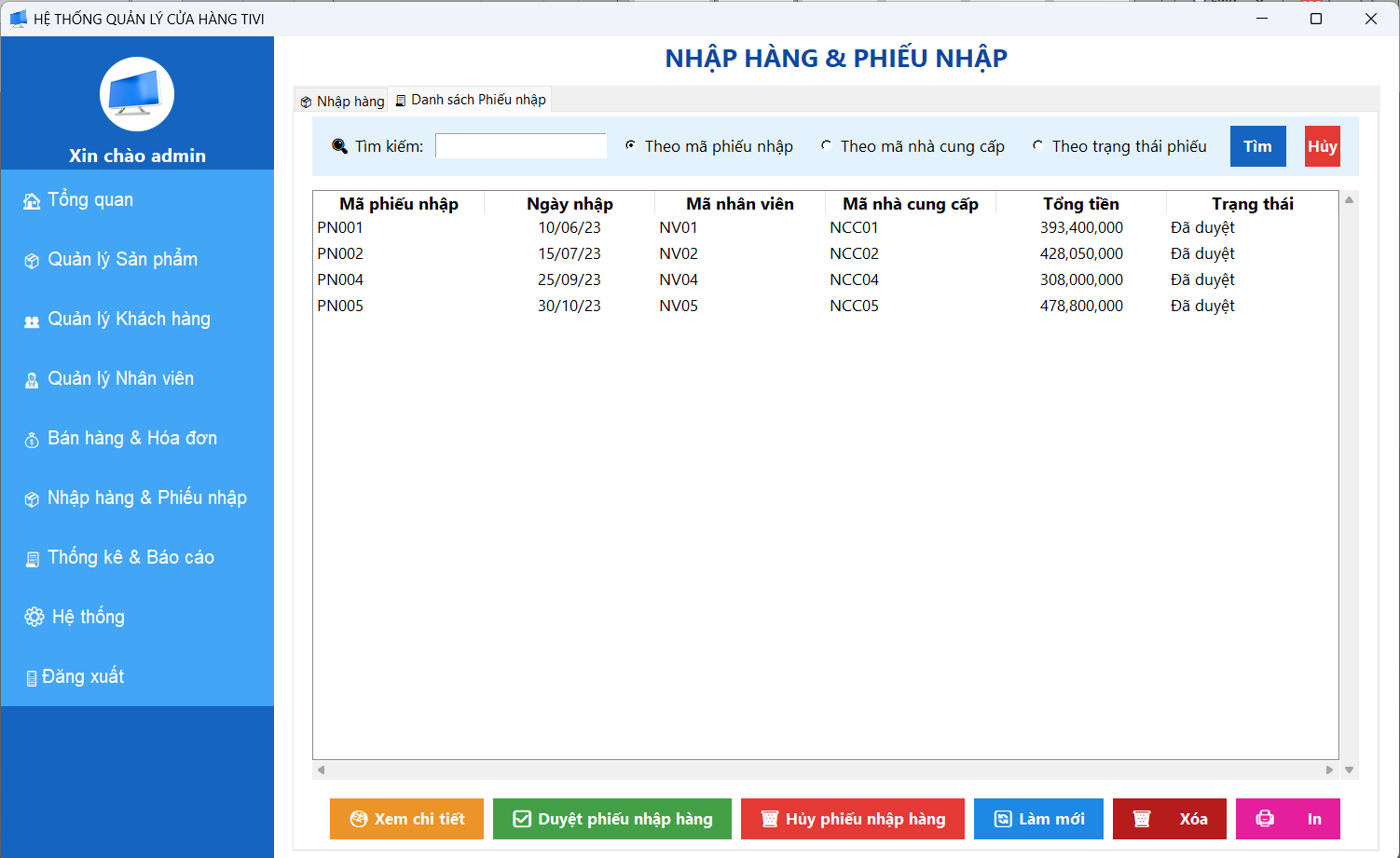
Hình 10: Giao diện quản lý bán hàng và hóa đơn – tab hóa đơn



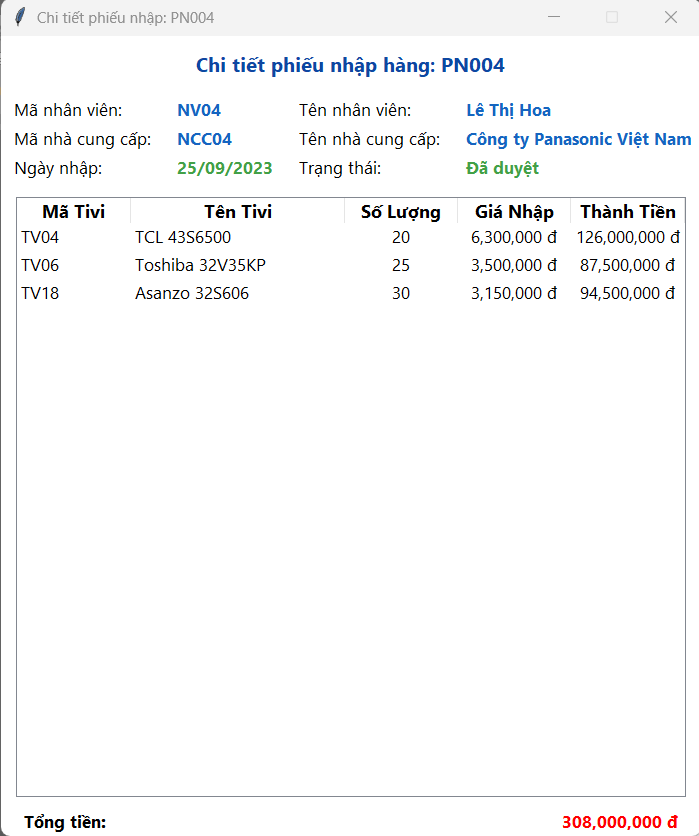
Hình 11: Giao diện quản lý bán hàng và hóa đơn – xem chi tiết hóa đơn



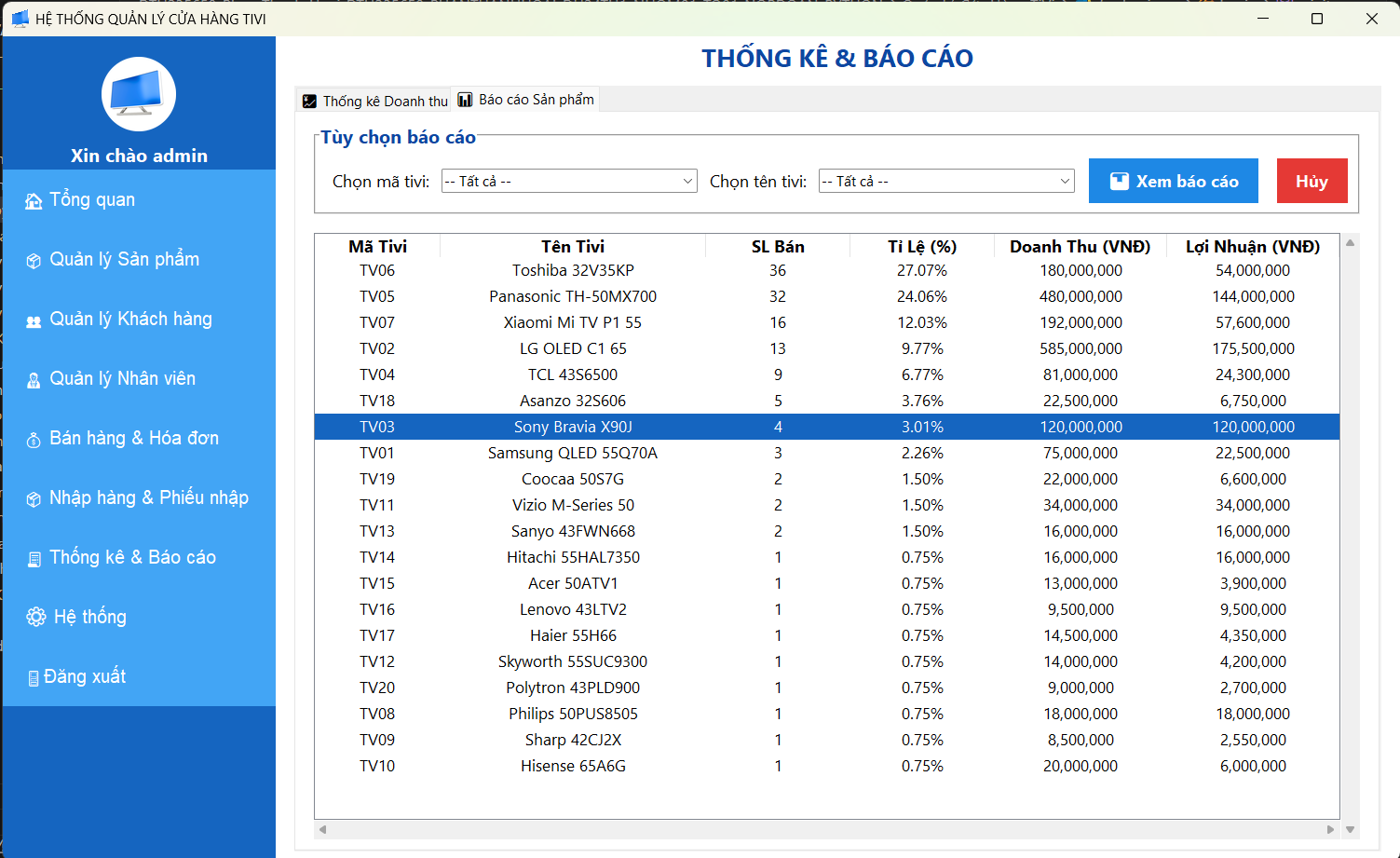
Hình 12: Giao diện quản lý nhập hàng và phiếu nhập – tab nhập hàng



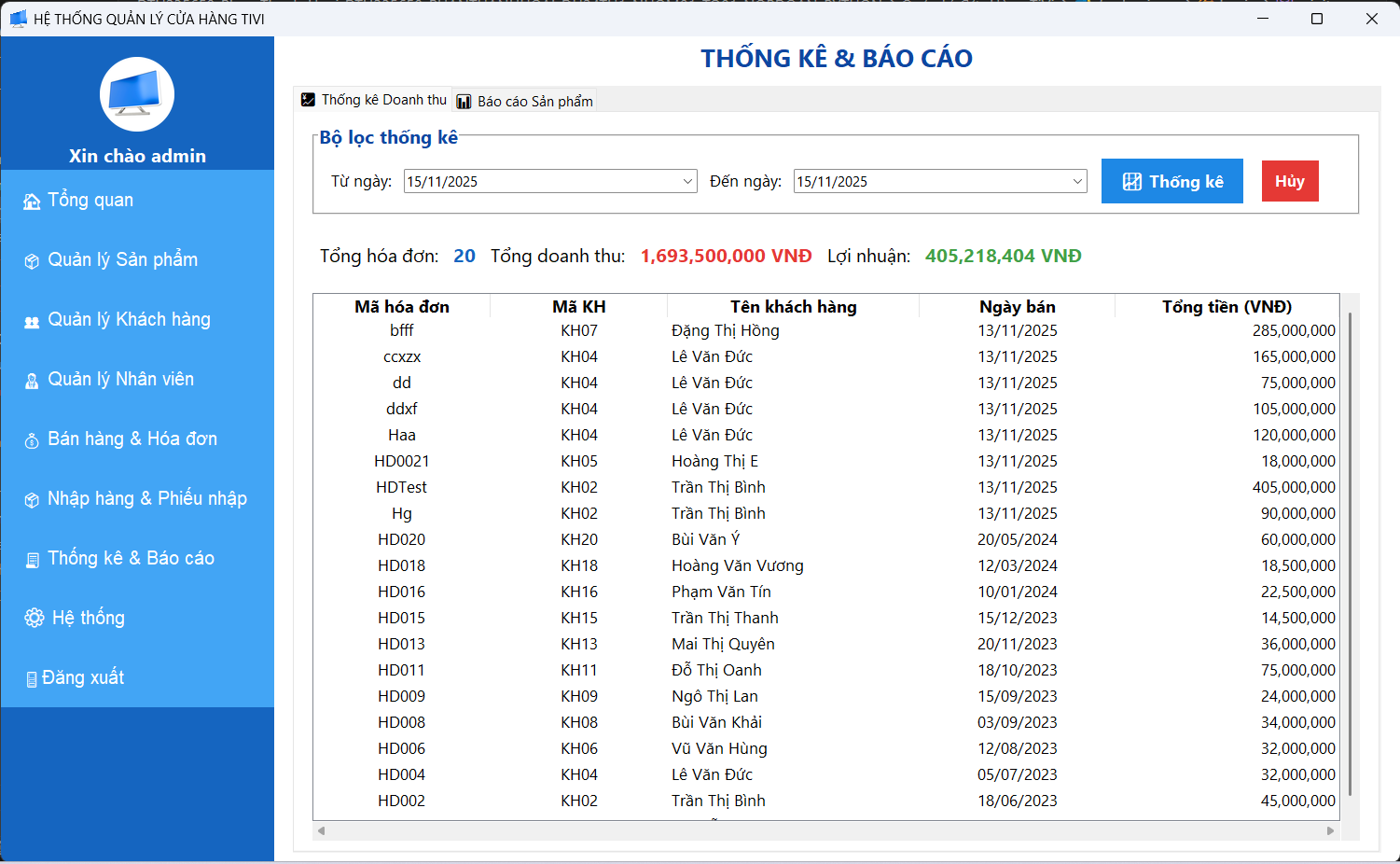
Hình 13: Giao diện quản lý nhập hàng và phiếu nhập – tab phiếu nhập hàng



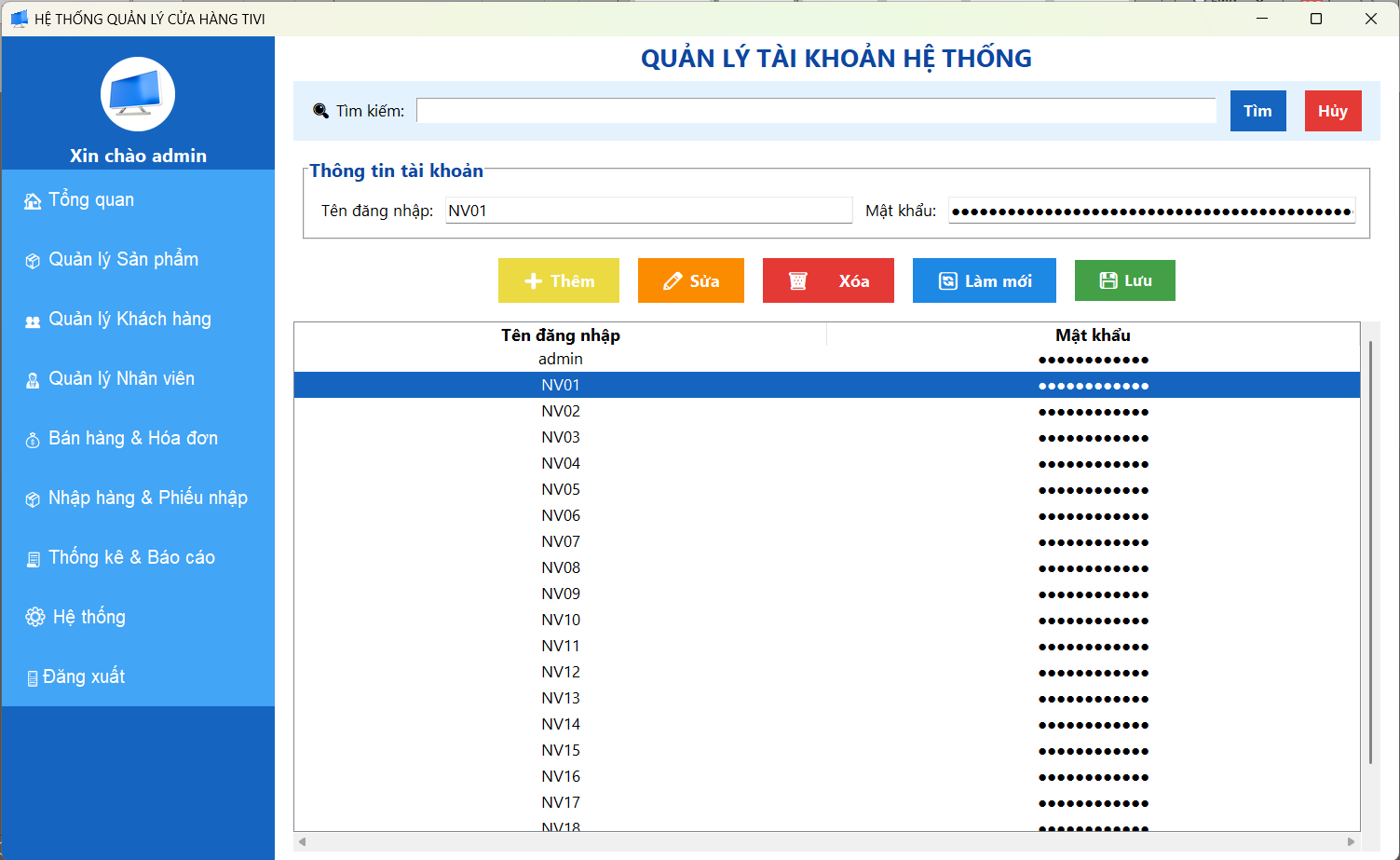
Hình 14: Giao diện quản lý nhập hàng và phiếu nhập – xem chi tiết phiếu nhập hàng



Hình 15: Giao diện quản lý thống kê và báo cáo – tab báo cáo sản phẩm

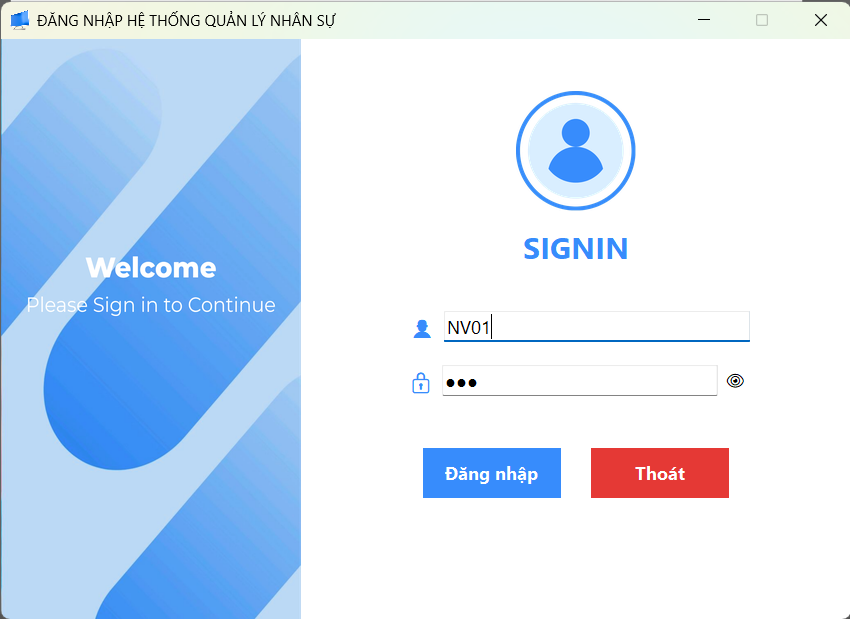


Hình 16: Giao diện quản lý thống kê và báo cáo – tab thống kê doanh thu

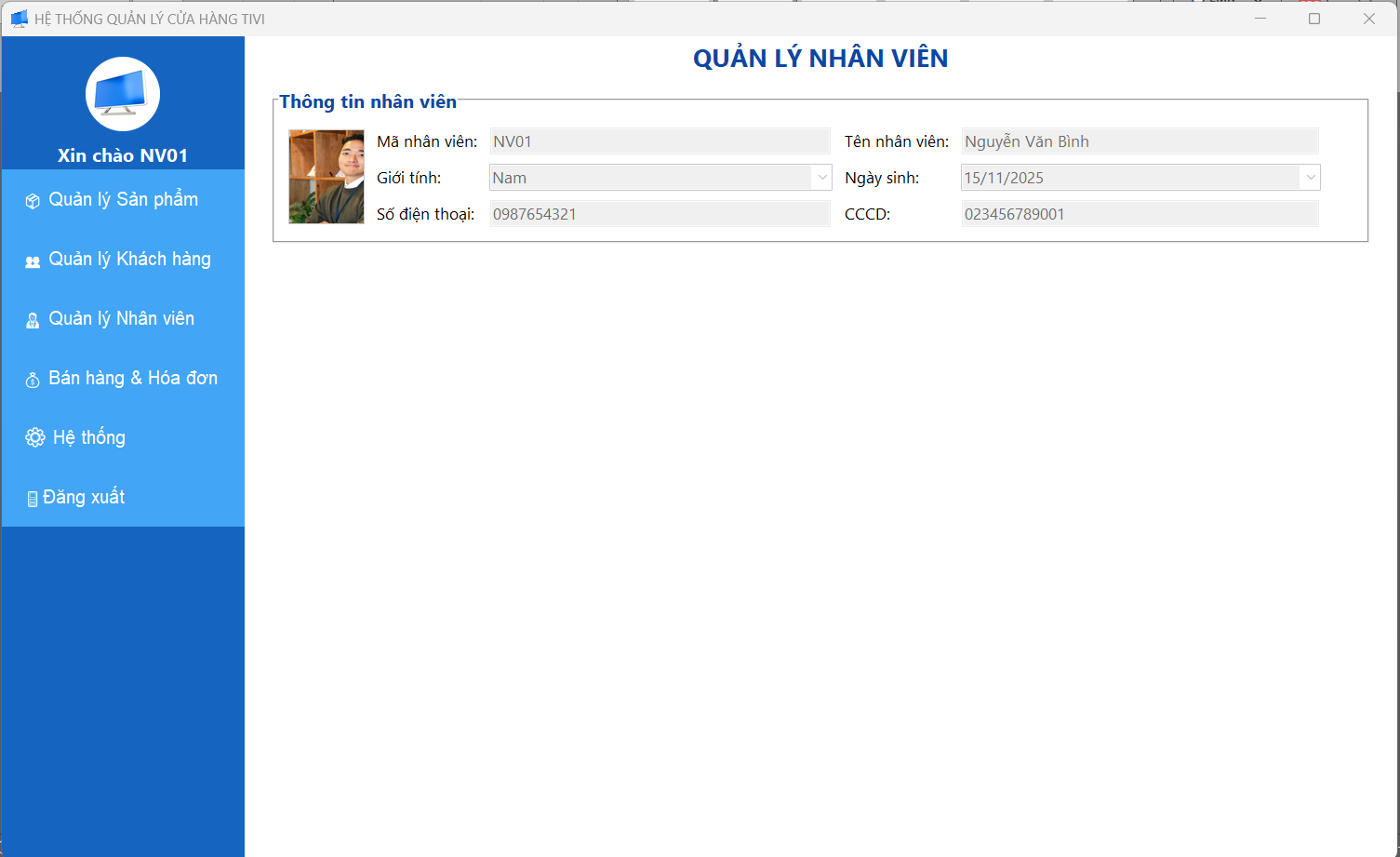


Hình 17: Giao diện quản lý tài khoản hệ thống

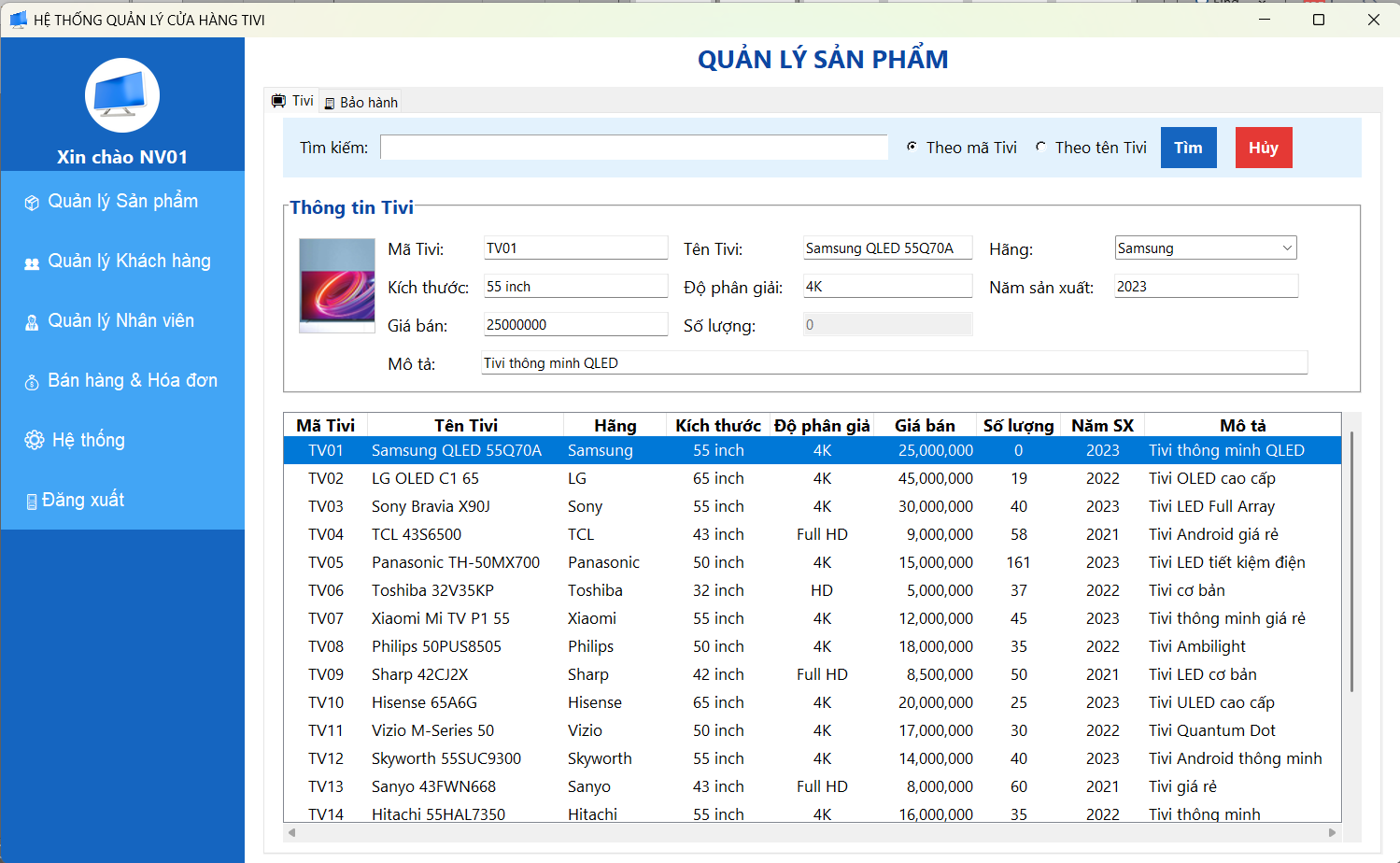
## **2. Giao diện dành cho nhân viên**

****

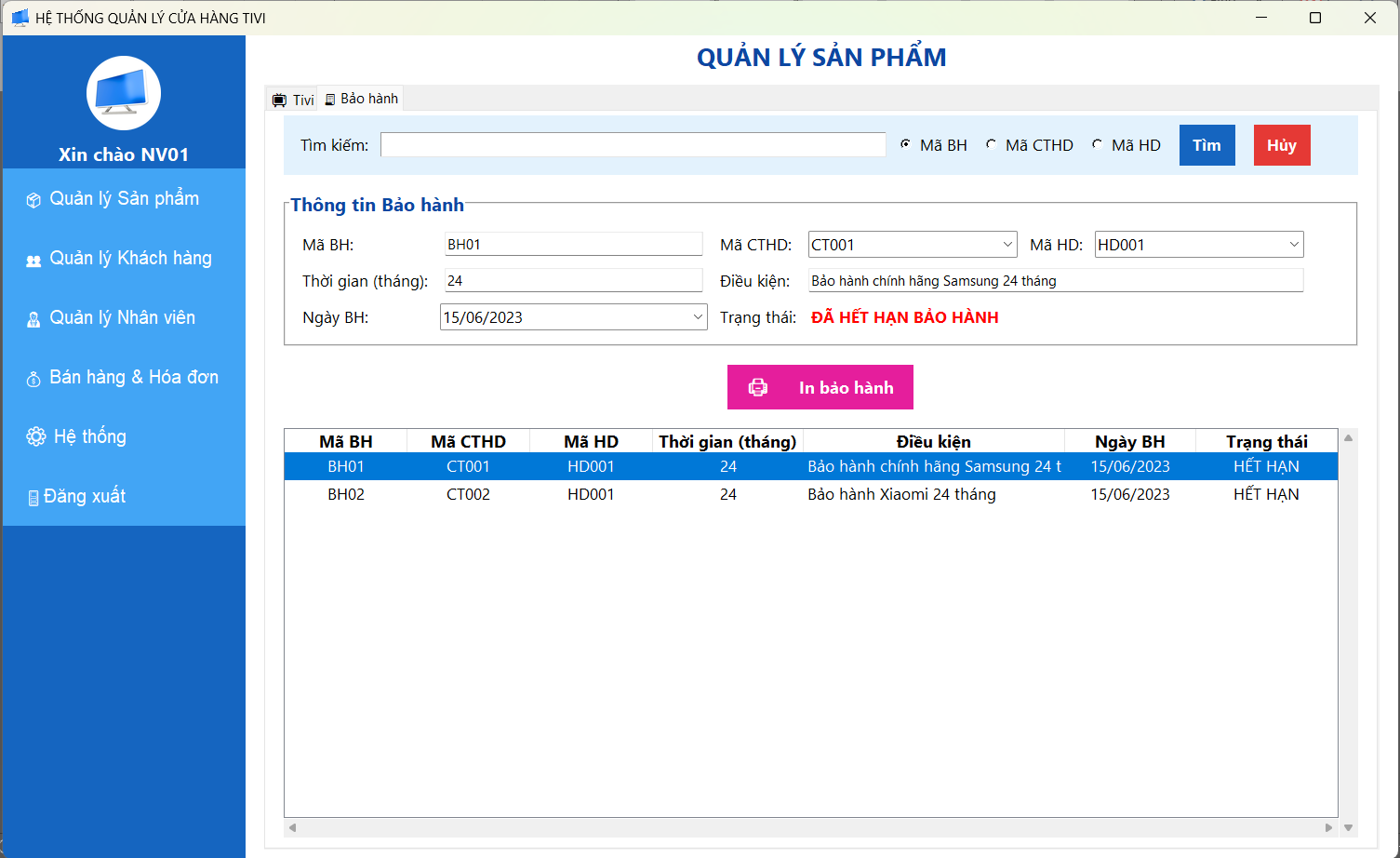
Hình 18: Giao diện đăng nhập của nhân viên



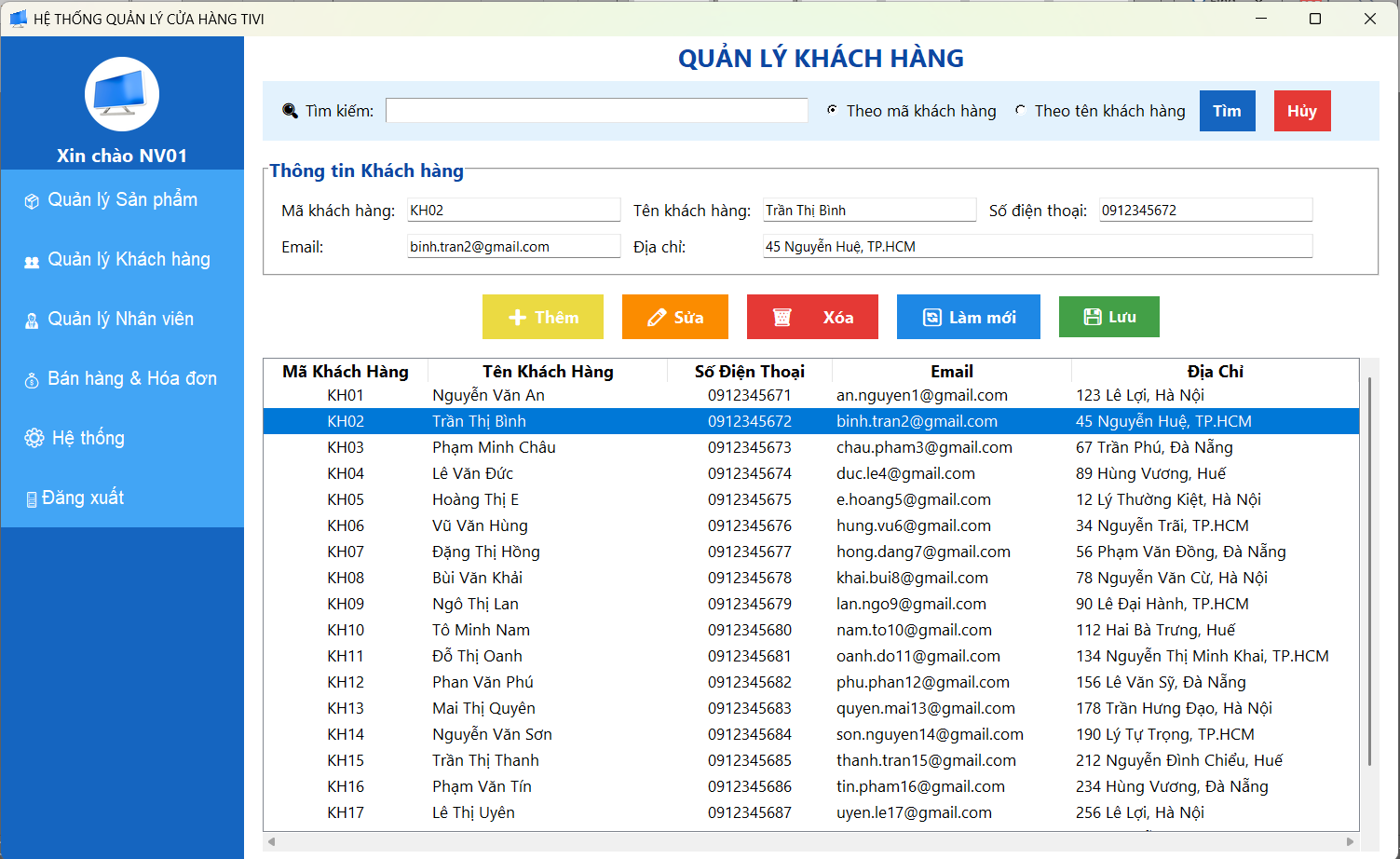
Hình 19: Giao diện quản lý nhân viên của nhân viên



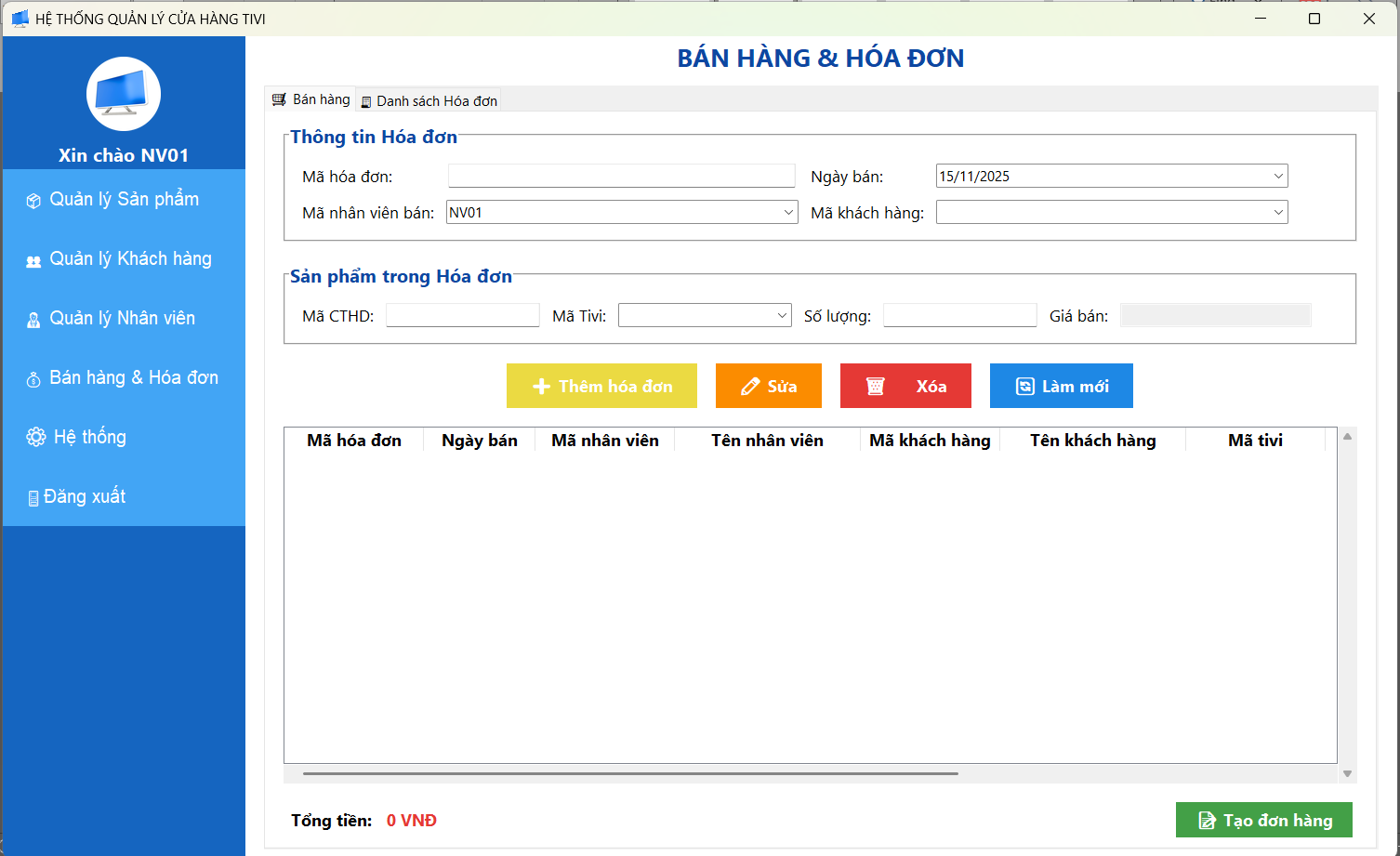
Hình 20: Giao diện quản lý sản phẩm – tab Tivi của nhân viên



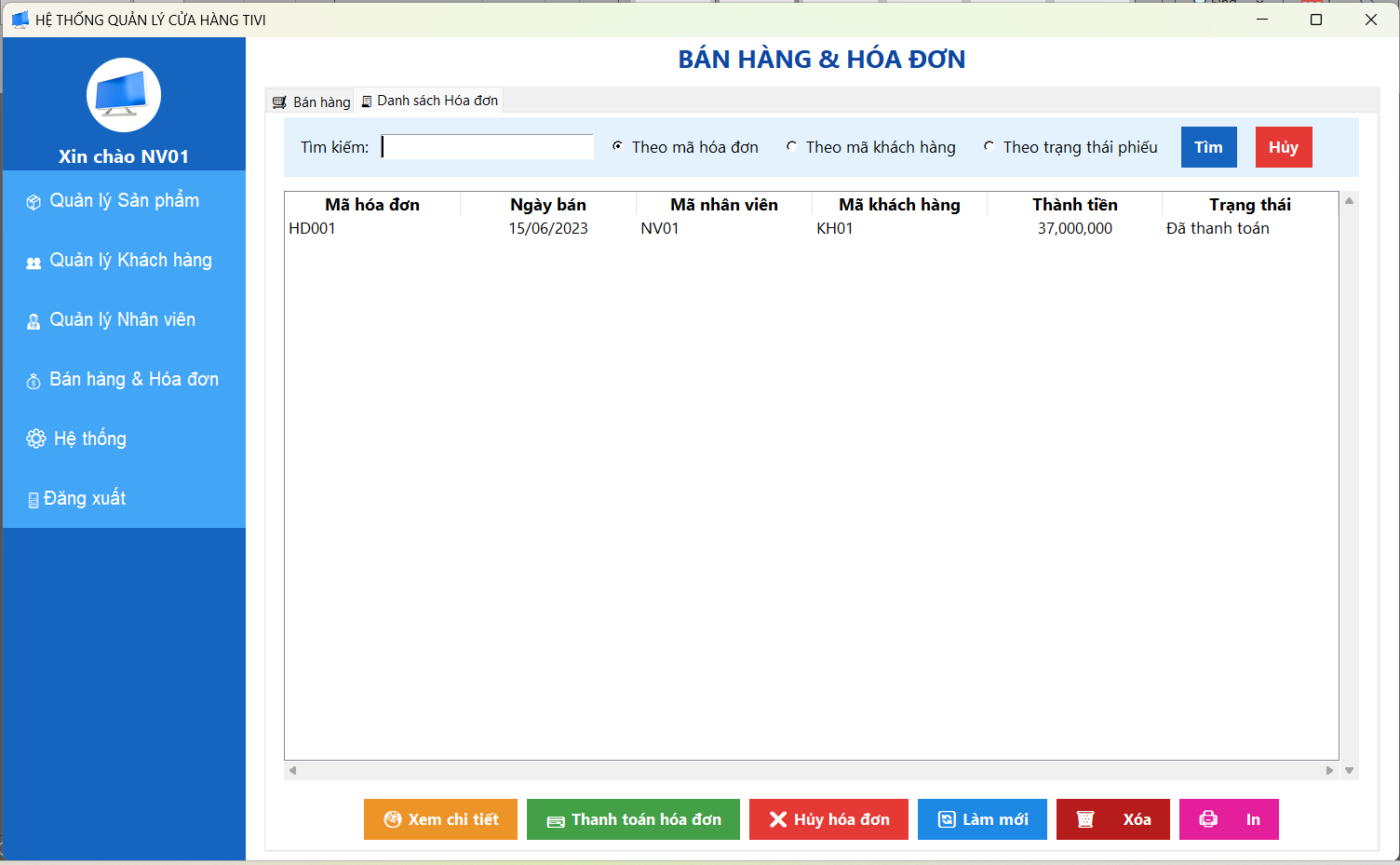
Hình 21: Giao diện quản lý sản phẩm – tab bảo hành của nhân viên



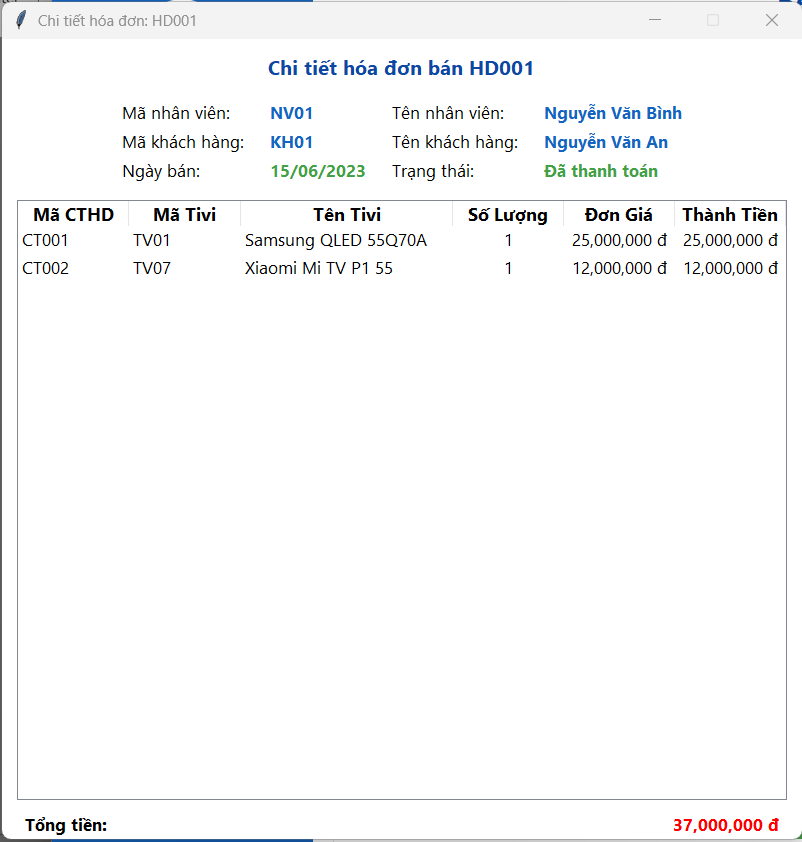
Hình 22: Giao diện quản lý sản phẩm – tab Tivi của nhân viên



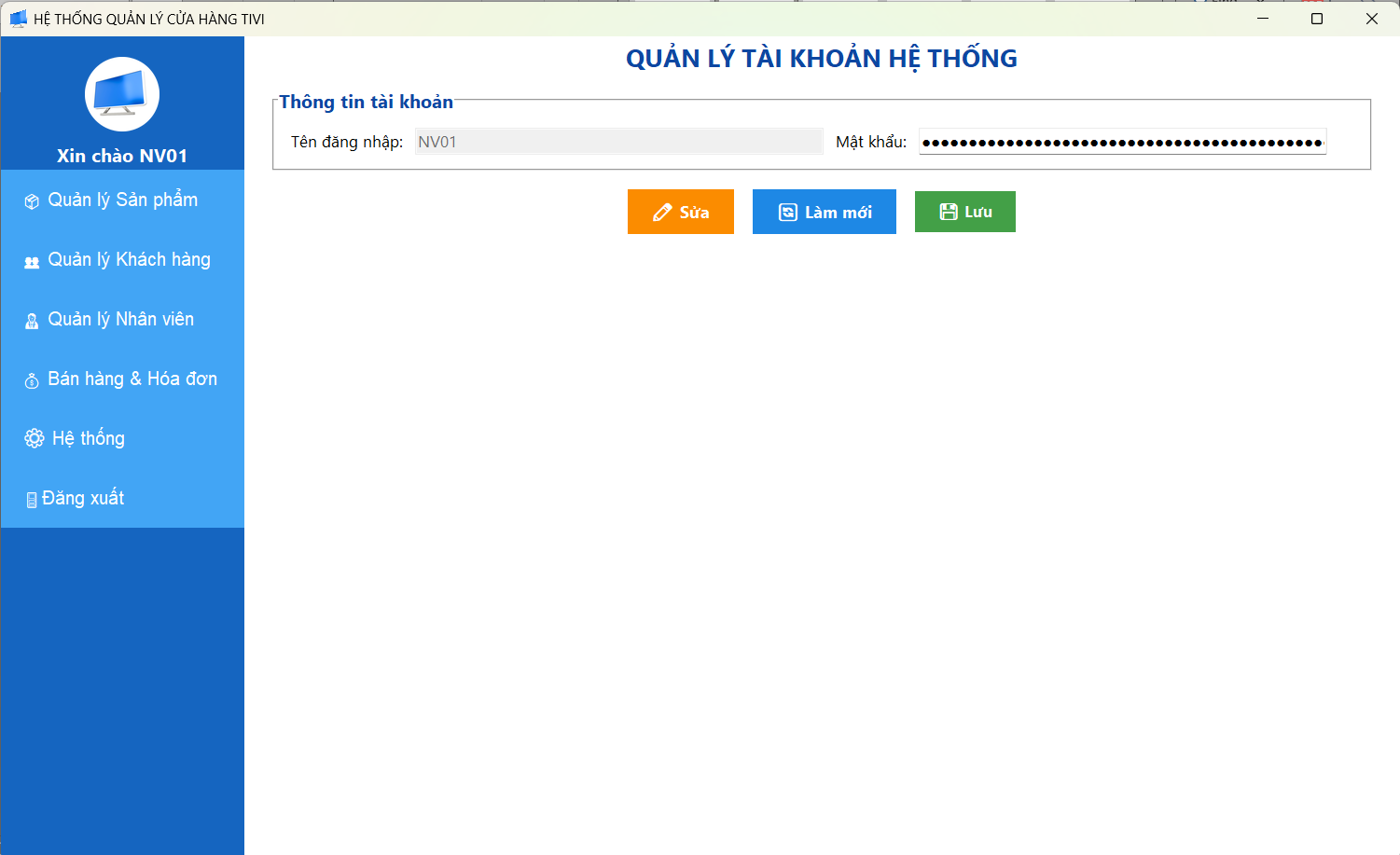
Hình 23: Giao diện quản lý bán hàng và hóa đơn – tab bán hàng của nhân viên



Hình 24: Giao diện quản lý bán hàng và hóa đơn – tab hóa đơn của nhân viên



Hình 25: Giao diện quản lý bán hàng và hóa đơn – xem chi tiết hóa đơn của nhân viên



Hình 26: Giao diện quản lý tài khoản hệ thống của nhân viên

## **B. CÁC BƯỚC XÂY DỰNG ỨNG DỤNG**

Để xây dựng ứng dụng quản lý cửa hàng Tivi bằng Python, Tkinter và SQL Server, hệ thống cần kết hợp giữa giao diện người dùng và cơ sở dữ liệu một cách chặt chẽ. Ứng dụng sử dụng pyodbc làm thư viện trung gian giúp Python kết nối với SQL Server, thực thi các câu lệnh truy vấn và xử lý dữ liệu theo thời gian thực. Bên cạnh đó, cơ sở dữ liệu QLTV được thiết kế trên SQL Server với hệ thống bảng đầy đủ như *HangSanXuat, KhachHang, NhanVien, NhaCungCap, Tivi, PhieuNhapHang, ChiTietPhieuNhap, HoaDonBan, ChiTietHoaDon và BaoHanh*. Mỗi bảng đều được xây dựng với khóa chính, khóa ngoại, các ràng buộc UNIQUE, CHECK và đi kèm các trigger tự động cập nhật tổng tiền, đảm bảo sự toàn vẹn và nhất quán của dữ liệu. Nhờ vào cấu trúc dữ liệu rõ ràng và cơ chế kết nối của pyodbc, ứng dụng có thể vận hành ổn định, hỗ trợ tốt cho các chức năng quản lý như nhập hàng, bán hàng, theo dõi tồn kho và bảo hành sản phẩm.

### **1. Cài đặt thư viện cần thiết**

★ **Python**: Đảm bảo bạn đã cài đặt Python trên máy tính của mình

★ **SQL Server**: Cài đặt máy chủ SQL Server và tạo một cơ sở dữ liệu cho ứng dụng của bạn.

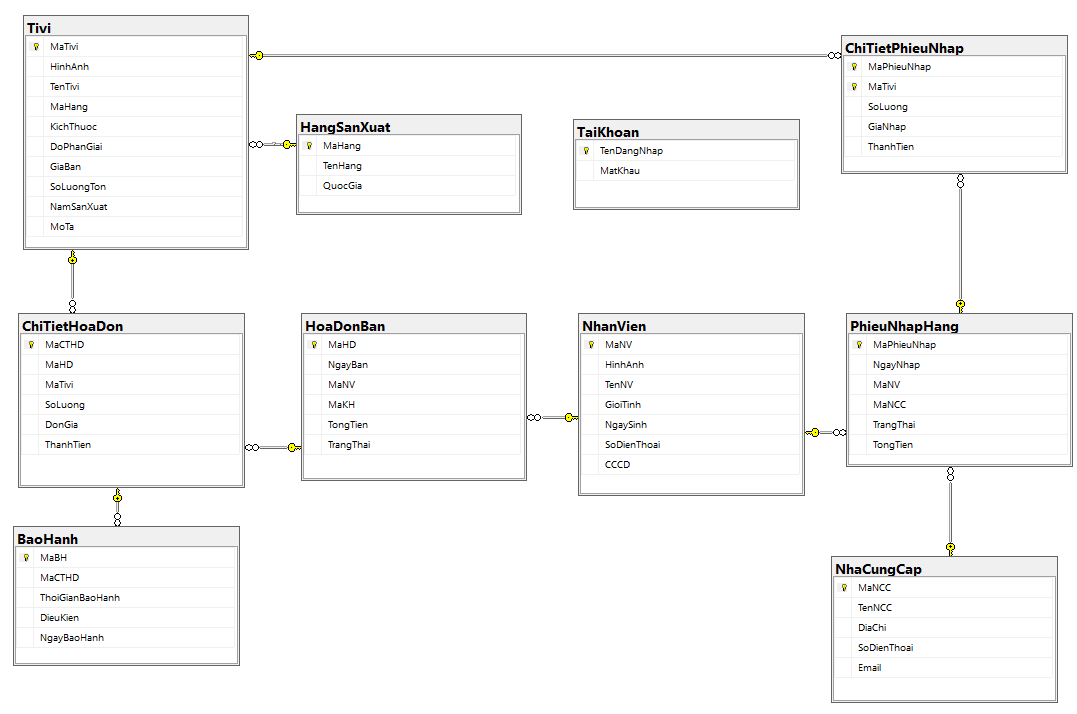
★ **ODBC Driver 17 for SQL Server**: Dùng để kết nối Python với SQL Server.

★ **pyodbc**: Cài đặt thư viện này để kết nối Python với cơ sở dữ liệu SQL Server  
(lệnh cài đặt: **pip install pyodbc**).

### **2. Thiết kế cơ sở dữ liệu SQL Server**

Tạo các bảng trong cơ sở dữ liệu SQL Server để lưu trữ thông tin quản lý cửa hàng Tivi như sau:

Hệ thống cơ sở dữ liệu được xây dựng trên SQL Server và bao gồm đầy đủ các bảng cần thiết để quản lý toàn diện hoạt động của cửa hàng Tivi. Các bảng được thiết kế theo mô hình quan hệ, có liên kết chặt chẽ với nhau thông qua khóa chính – khóa ngoại nhằm đảm bảo dữ liệu nhất quán và dễ mở rộng. Các bảng quan trọng bao gồm: HangSanXuat để lưu thông tin hãng tivi; Tivi để quản lý sản phẩm; KhachHang và NhanVien để lưu trữ thông tin người dùng và nhân viên; NhaCungCap để quản lý đối tác cung ứng; PhieuNhapHang và ChiTietPhieuNhap để xử lý hoạt động nhập kho; HoaDonBan và ChiTietHoaDon phục vụ nghiệp vụ bán hàng; cùng bảng BaoHanh để theo dõi thông tin bảo hành sau bán. Ngoài ra, bảng TaiKhoan được sử dụng để quản lý đăng nhập hệ thống. Các bảng đều áp dụng đầy đủ ràng buộc như PRIMARY KEY, UNIQUE, CHECK, DEFAULT và liên kết bằng FOREIGN KEY để đảm bảo tính toàn vẹn dữ liệu. Toàn bộ cấu trúc này tạo thành một hệ thống cơ sở dữ liệu hoàn chỉnh, hỗ trợ hiệu quả cho việc quản lý sản phẩm, khách hàng, nhân viên, nhập hàng, bán hàng và bảo hành trong cửa hàng Tivi.



Hình 27: Diagram liên kết giữa các bảng

### **3. Xây dựng giao diện người dùng (GUI) bằng Tkinter**

Sử dụng thư viện Tkinter để xây dựng các cửa sổ giao diện, khung hiển thị, ô nhập liệu, nút thao tác và bảng (Treeview) dùng để trình bày danh sách Tivi, khách hàng, nhân viên, nhà cung cấp, hóa đơn và các dữ liệu khác. Giao diện được thiết kế trực quan, dễ sử dụng và hỗ trợ đầy đủ các chức năng phục vụ hoạt động quản lý cửa hàng Tivi. Cụ thể, người dùng có thể thực hiện:

**a) Giao diện Đăng nhập(Login)**

* Đăng nhập tài khoản
* Thoát chương trình

**b) Giao diện Form chinh(app)**

* Chuyển form Tổng quan
* Chuyển form Quản lý sản phẩm
* Chuyển form Quản lý khách hàng
* Chuyển form Quản lý nhân viên
* Chuyển form Bán hàng & Hóa đơn
* Chuyển form Nhập hàng & Phiếu nhập hàng
* Chuyển form Thống kê & Báo cáo
* Chuyển form Hệ thống
* Đăng xuất tài khoản

**c) Giao diện Tổng quan**

* Xem tổng số lượng nhân viên, khách hàng, hãng sản xuất, nhà cung cấp, sản phẩm, phiếu nhập
* Xem biểu đồ doanh thu của mỗi tháng và từng năm cụ thể

**d) Giao diện Quản lý tivi**

* Thêm thông tin Tivi mới
* Lưu thông tin Tivi
* Sửa thông tin Tivi
* Làm mới thông tin Tivi
* Xóa thông tin Tivi
* Xem danh sách tất cả Tivi
* Tìm kiếm Tivi
* Chọn ảnh cho tivi

**e) Giao diện Quản lý hãng sản xuất**

* Thêm thông tin Hãng sản xuất mới
* Lưu thông tin Hãng sản xuất
* Sửa thông tin Hãng sản xuất
* Làm mới thông tin Hãng sản xuất
* Xóa thông tin Hãng sản xuất
* Xem danh sách tất cả Hãng sản xuất
* Tìm kiếm Hãng sản xuất

**f) Giao diện Quản lý nhà cung cấp**

* Thêm thông tin Nhà cung cấp mới
* Lưu thông tin Nhà cung cấp
* Sửa thông tin Nhà cung cấp
* Làm mới thông tin Nhà cung cấp
* Xóa thông tin Nhà cung cấp
* Xem danh sách tất cả Nhà cung cấp
* Tìm kiếm Nhà cung cấp

**g) Giao diện Quản lý bảo hành**

* Thêm thông tin Bảo hành mới
* Lưu thông tin Bảo hành
* Sửa thông tin Bảo hành
* Làm mới thông tin Bảo hành
* Xóa thông tin Bảo hành
* Xem danh sách tất cả Bảo hành
* Tìm kiếm Bảo hành
* In phiếu bảo hành

**h) Giao diện Quản lý khách hàng**

* Thêm thông tin Khách hàng mới
* Lưu thông tin Khách hàng
* Sửa thông tin Khách hàng
* Làm mới thông tin Khách hàng
* Xóa thông tin Khách hàng
* Xem danh sách tất cả Khách hàng
* Tìm kiếm Khách hàng

**i) Giao diện Quản lý nhân viên**

* Thêm thông tin Nhân viên mới
* Lưu thông tin Nhân viên
* Sửa thông tin Nhân viên
* Làm mới thông tin Nhân viên
* Xóa thông tin Nhân viên
* Xem danh sách tất cả Nhân viên
* Tìm kiếm Nhân viên

**k) Giao diện Quản lý bán hàng**

* Thêm thông tin Hóa đơn và Chi tiết hóa đơn mới
* Xóa thông tin Hóa đơn và Chi tiết hóa đơn
* Sửa thông tin Hóa đơn và Chi tiết hóa đơn
* Làm mới thông tin Hóa đơn và Chi tiết hóa đơn
* Tạo Hóa đơn và Chi tiết hóa đơn mới

**l) Giao diện Quản lý hóa đơn**

* Xem chi tiết Hóa đơn
* Thanh toán Hóa đơn
* Hủy Hóa đơn
* Làm mới thông tin Hóa đơn
* Xóa thông tin Hóa đơn
* Xem danh sách tất cả Hóa đơn
* Tìm kiếm Hóa đơn
* In Hóa đơn

**m) Giao diện Chi tiết hóa đơn**

* Xem thông tin Chi tiết hóa đơn
* Xem tổng tiền cần thanh toán

**n) Giao diện Quản lý nhập hàng**

* Thêm thông tin Phiếu nhập và Chi tiết phiếu nhập mới
* Xóa thông tin Phiếu nhập và Chi tiết phiếu nhập
* Sửa thông tin Phiếu nhập và Chi tiết phiếu nhập
* Làm mới thông tin Phiếu nhập và Chi tiết phiếu nhập
* Nhập hàng

**o) Giao diện Quản lý phiếu nhập**

* Xem chi tiết Phiếu nhập
* Duyệt Phiếu nhập
* Hủy Phiếu nhập
* Làm mới thông tin Phiếu nhập
* Xóa thông tin Phiếu nhập
* Xem danh sách tất cả Phiếu nhập
* Tìm kiếm Phiếu nhập
* In Phiếu nhập

**p) Giao diện Chi tiết phiếu nhập**

* Xem thông tin Chi tiết phiếu nhập
* Xem tổng tiền cần thanh toán

**q) Giao diện Thống kê doanh thu**

* Thống kê doanh thu theo ngày cụ thể
* Hủy thống kê
* Xem danh sách hóa đơn đã bán ra
* Xem tổng doanh thu và lợi nhuận

**r) Giao diện báo cáo sản phẩm**

* Xem báo cáo của tivi cụ thể đã bán ra
* Hủy báo cáo
* Xem doanh sách tỉ lệ bán chạy của tất cả tivi

**s) Giao diện Hệ thống**

* Thêm thông tin Tài khoản mới
* Lưu thông tin Tài khoản
* Sửa thông tin Tài khoản
* Làm mới thông tin Tài khoản
* Xóa thông tin Tài khoản
* Xem danh sách tất cả Tài khoản
* Tìm kiếm Tài khoản

### **4. Viết mã Python kết nối cơ sở dữ liệu**

Để các chức năng nghiệp vụ trong ứng dụng có thể hoạt động chính xác, hệ thống cần xây dựng lớp xử lý dữ liệu bằng Python kết hợp với thư viện *pyodbc* nhằm thiết lập kết nối ổn định với cơ sở dữ liệu SQL Server. Thông qua cơ chế kết nối này, chương trình có thể thực hiện các thao tác thêm, sửa, xóa và truy vấn thông tin một cách linh hoạt, phục vụ cho quá trình quản lý dữ liệu Tivi và các mục liên quan trong toàn bộ hệ thống.

**a) Form Đăng nhập (Login)**

* Đăng nhập tài khoản: Lấy dữ liệu từ các trường Tên đăng nhập và Mật khẩu, truy vấn bảng TaiKhoan để xác thực và chuyển vào giao diện chính nếu đúng.
* Thoát chương trình: Đóng toàn bộ ứng dụng và kết thúc phiên làm việc.

**b) Form chính (App)**

* Chuyển form Tổng quan: Điều hướng sang giao diện Tổng quan và tải số liệu thống kê từ cơ sở dữ liệu.
* Chuyển form Quản lý sản phẩm: Tải giao diện Tivi và truy vấn danh sách sản phẩm.  
  Chuyển form Quản lý khách hàng: Hiển thị giao diện Khách hàng và truy vấn danh sách từ bảng KhachHang.
* Chuyển form Quản lý nhân viên: Điều hướng đến giao diện Nhân viên và tải dữ liệu từ bảng NhanVien.
* Chuyển form Bán hàng & Hóa đơn: Mở giao diện tạo hóa đơn và cho phép thêm sản phẩm.
* Chuyển form Nhập hàng & Phiếu nhập hàng: Tải giao diện nhập hàng và danh sách phiếu nhập.
* Chuyển form Thống kê & Báo cáo: Điều hướng đến module thống kê, truy vấn doanh thu và dữ liệu báo cáo.
* Chuyển form Hệ thống: Hiển thị giao diện quản lý tài khoản và tải bảng TaiKhoan.  
  Đăng xuất tài khoản: Xóa dữ liệu đăng nhập và quay về form Login.

**c) Form Tổng quan**

* Xem tổng số lượng nhân viên, khách hàng, hãng, nhà cung cấp, sản phẩm, phiếu nhập: Truy vấn thống kê COUNT từ các bảng tương ứng.
* Xem biểu đồ doanh thu theo tháng và năm: Truy vấn dữ liệu hóa đơn và hiển thị biểu đồ bằng matplotlib.

**d) Form Quản lý Tivi**

* Thêm thông tin Tivi mới: Lấy dữ liệu từ các trường nhập liệu và chèn vào bảng Tivi trong cơ sở dữ liệu.
* Lưu thông tin Tivi: Ghi lại dữ liệu Tivi mới hoặc đã chỉnh sửa và cập nhật vào cơ sở dữ liệu.
* Sửa thông tin Tivi: Cập nhật thông tin Tivi dựa trên mã Tivi được chọn trong bảng.
* Làm mới thông tin Tivi: Xóa dữ liệu trong các ô nhập liệu và tải lại danh sách Tivi.
* Xóa thông tin Tivi: Xóa Tivi khỏi bảng Tivi dựa trên mã được chọn.
* Xem danh sách tất cả Tivi: Truy vấn và hiển thị toàn bộ danh sách Tivi trong Treeview.
* Tìm kiếm Tivi: Lọc Tivi theo tên, hãng hoặc mã và hiển thị kết quả trên bảng.
* Chọn ảnh cho Tivi: Mở hộp thoại chọn ảnh và lưu ảnh vào trường HinhAnh của bảng Tivi.

**e) Form Quản lý hãng sản xuất**

* Thêm thông tin Hãng sản xuất mới: Lấy dữ liệu từ các ô nhập liệu và thêm vào bảng HangSanXuat.
* Lưu thông tin Hãng sản xuất: Ghi lại toàn bộ dữ liệu đã nhập và cập nhật vào cơ sở dữ liệu.
* Sửa thông tin Hãng sản xuất: Cập nhật lại dữ liệu hãng dựa vào mã hãng được chọn.
* Làm mới thông tin Hãng sản xuất: Xóa nội dung trên các ô nhập liệu và tải lại danh sách hãng.
* Xóa thông tin Hãng sản xuất: Xóa hãng sản xuất khỏi cơ sở dữ liệu nếu không bị ràng buộc.
* Xem danh sách tất cả Hãng sản xuất: Truy vấn và hiển thị toàn bộ danh sách hãng sản xuất.
* Tìm kiếm Hãng sản xuất: Lọc hãng theo tên, mã hoặc quốc gia.

**f) From Quản lý nhà cung cấp**

* Thêm thông tin Nhà cung cấp mới: Lấy dữ liệu nhập liệu và chèn vào bảng NhaCungCap.
* Lưu thông tin Nhà cung cấp: Ghi lại thông tin nhà cung cấp vào cơ sở dữ liệu.
* Sửa thông tin Nhà cung cấp: Điều chỉnh dữ liệu của nhà cung cấp được chọn.
* Làm mới thông tin Nhà cung cấp: Xóa ô nhập liệu và tải lại danh sách nhà cung cấp.
* Xóa thông tin Nhà cung cấp: Xóa nhà cung cấp khỏi bảng dựa trên mã nhà cung cấp.
* Xem danh sách tất cả Nhà cung cấp: Truy vấn toàn bộ danh sách nhà cung cấp và hiển thị.
* Tìm kiếm Nhà cung cấp: Lọc theo tên, mã, số điện thoại hoặc email.

**g) Form Quản lý bảo hành**

* Thêm thông tin Bảo hành mới: Lấy dữ liệu nhập và thêm vào bảng BaoHanh.
* Lưu thông tin Bảo hành: Lưu thông tin bảo hành vào cơ sở dữ liệu.
* Sửa thông tin Bảo hành: Cập nhật điều kiện hoặc thời gian bảo hành theo mã bảo hành.
* Làm mới thông tin Bảo hành: Xóa ô nhập liệu và tải lại danh sách.
* Xóa thông tin Bảo hành: Xóa dòng bảo hành dựa vào mã chi tiết hóa đơn.
* Xem danh sách tất cả Bảo hành: Truy vấn bảng BaoHanh và hiển thị dữ liệu.
* Tìm kiếm Bảo hành: Lọc theo mã hóa đơn, tên sản phẩm hoặc trạng thái.
* In phiếu bảo hành: Tạo file bảo hành từ template và xuất ra dạng PDF hoặc Word.

**h) FormQuản lý khách hàng**

* Thêm thông tin Khách hàng mới: Nhập dữ liệu và chèn vào bảng KhachHang.
* Lưu thông tin Khách hàng: Lưu dữ liệu nhập liệu vào cơ sở dữ liệu.
* Sửa thông tin Khách hàng: Điều chỉnh thông tin khách hàng dựa vào mã khách hàng.
* Làm mới thông tin Khách hàng: Xóa nội dung nhập liệu và làm mới bảng danh sách.
* Xóa thông tin Khách hàng: Xóa khách hàng khỏi cơ sở dữ liệu dựa vào mã khách hàng.
* Xem danh sách tất cả Khách hàng: Truy vấn toàn bộ khách hàng và hiển thị.
* Tìm kiếm Khách hàng: Lọc theo tên, số điện thoại, email hoặc mã khách hàng.

**i) Form Quản lý nhân viên**

* Thêm thông tin Nhân viên mới: Lấy dữ liệu và chèn vào bảng NhanVien.
* Lưu thông tin Nhân viên: Ghi lại thông tin mới hoặc cập nhật dữ liệu nhân viên.
* Sửa thông tin Nhân viên: Thay đổi thông tin nhân viên dựa trên mã nhân viên.
* Làm mới thông tin Nhân viên: Xóa ô nhập liệu và cập nhật lại danh sách.
* Xóa thông tin Nhân viên: Xóa nhân viên khỏi bảng NhanVien nếu không bị liên kết.
* Xem danh sách tất cả Nhân viên: Truy vấn danh sách nhân viên và hiển thị.
* Tìm kiếm Nhân viên: Lọc theo tên, mã nhân viên, số điện thoại hoặc CCCD.

**k) Form Quản lý bán hàng**

* Thêm hóa đơn và chi tiết hóa đơn: Lấy dữ liệu và tạo bản ghi mới vào bảng
* HoaDonBan và ChiTietHoaDon.Xóa sản phẩm trong hóa đơn: Xóa dòng chi tiết ra khỏi bảng ChiTietHoaDon.
* Sửa chi tiết hóa đơn: Điều chỉnh số lượng hoặc đơn giá trong chi tiết hóa đơn.
* Làm mới hóa đơn: Xóa toàn bộ dữ liệu nhập liệu trên form.
* Tạo hóa đơn: Lưu thông tin hóa đơn mới vào cơ sở dữ liệu.

**l) Form Quản lý hóa đơn**

* Xem chi tiết Hóa đơn: Hiển thị danh sách sản phẩm trong từng hóa đơn.
* Thanh toán Hóa đơn: Cập nhật trạng thái hóa đơn sang “Đã thanh toán”.
* Hủy Hóa đơn: Hủy hóa đơn chưa thanh toán và cập nhật lại tồn kho nếu cần.
* Làm mới thông tin Hóa đơn: Xóa dữ liệu trên form và tải lại danh sách.
* Xóa thông tin Hóa đơn: Xóa hóa đơn khỏi cơ sở dữ liệu nếu hợp lệ.
* Xem danh sách tất cả Hóa đơn: Truy vấn và hiển thị toàn bộ hóa đơn.
* Tìm kiếm Hóa đơn: Lọc theo mã hóa đơn, khách hàng hoặc ngày bán.
* In Hóa đơn: Xuất hóa đơn ra file Word hoặc PDF từ template.
* Xem thông tin Chi tiết hóa đơn: Hiển thị danh sách sản phẩm trong hóa đơn được chọn.
* Xem tổng tiền cần thanh toán: Tính tổng thành tiền dựa trên số lượng và đơn giá.

**m) Form Quản lý nhập hàng**

* Thêm phiếu nhập và chi tiết phiếu nhập: Nhập dữ liệu và chèn vào bảng PhieuNhapHang và ChiTietPhieuNhap.
* Xóa phiếu nhập hoặc chi tiết: Xóa bản ghi được chọn khỏi CSDL.
* Sửa chi tiết phiếu nhập: Điều chỉnh số lượng hoặc giá nhập.
* Làm mới phiếu nhập: Xóa nhập liệu và tải lại danh sách.
* Nhập hàng: Lưu phiếu nhập và cập nhật tồn kho sản phẩm.

**n) Form Quản lý phiếu nhập**

* Xem chi tiết Phiếu nhập: Hiển thị danh sách sản phẩm của phiếu nhập.
* Duyệt Phiếu nhập: Cập nhật trạng thái sang “Đã duyệt” để chính thức nhập kho.
* Hủy Phiếu nhập: Đánh dấu phiếu nhập bị hủy và không cập nhật tồn kho.
* Làm mới thông tin Phiếu nhập: Xóa dữ liệu nhập liệu và làm mới bảng.
* Xóa thông tin Phiếu nhập: Xóa phiếu nhập nếu chưa duyệt.
* Xem danh sách tất cả Phiếu nhập: Truy vấn toàn bộ phiếu nhập hàng.
* Tìm kiếm Phiếu nhập: Lọc theo mã phiếu nhập, nhà cung cấp hoặc ngày nhập.
* In Phiếu nhập: Xuất phiếu nhập ra file Word hoặc PDF.
* Xem thông tin Chi tiết phiếu nhập: Hiển thị danh sách sản phẩm được nhập theo từng phiếu.
* Xem tổng tiền cần thanh toán: Tính tổng giá trị phiếu nhập dựa trên số lượng và giá nhập.

**o) Form Thống kê doanh thu**

* Thống kê doanh thu theo ngày cụ thể: Lọc hóa đơn bán ra theo ngày đã chọn.
* Hủy thống kê: Xóa bộ lọc và hiển thị dữ liệu mặc định.
* Xem danh sách hóa đơn đã bán ra: Hiển thị tất cả hóa đơn đã thanh toán.
* Xem tổng doanh thu và lợi nhuận: Tổng hợp doanh thu và tính lợi nhuận theo kỳ.

**p) Form báo cáo sản phẩm**

* Xem báo cáo của tivi cụ thể đã bán ra: Hiển thị lịch sử bán hàng của một sản phẩm.
* Hủy báo cáo: Làm mới dữ liệu và xóa các bộ lọc.
* Xem danh sách tỉ lệ bán chạy của tất cả tivi: Sắp xếp và hiển thị danh sách sản phẩm theo số lượng bán.

**q) Form Hệ thống**

* Thêm thông tin Tài khoản mới: Nhập dữ liệu và thêm vào bảng TaiKhoan.
* Lưu thông tin Tài khoản: Lưu dữ liệu tài khoản mới hoặc đã chỉnh sửa.
* Sửa thông tin Tài khoản: Thay đổi mật khẩu hoặc tên đăng nhập.
* Làm mới thông tin Tài khoản: Xóa dữ liệu nhập và tải lại danh sách.
* Xóa thông tin Tài khoản: Xóa tài khoản khỏi hệ thống nếu hợp lệ.
* Xem danh sách tất cả Tài khoản: Truy vấn và hiển thị toàn bộ tài khoản.
* Tìm kiếm Tài khoản: Lọc theo tên đăng nhập.

### **5. Xây dựng giao diện Tkinter**

Để xây dựng giao diện người dùng bằng Tkinter, ứng dụng được thiết kế theo mô hình hướng sự kiện, trong đó mỗi nút bấm và thao tác trên giao diện đều được liên kết trực tiếp với các hàm xử lý trong Python. Giao diện Tkinter đóng vai trò là cầu nối giữa người dùng và cơ sở dữ liệu, cho phép thực hiện các chức năng quản lý một cách trực quan và thuận tiện. Thông qua việc liên kết các widget như Buttons, Entry, Treeview với các hàm nghiệp vụ, hệ thống đảm bảo mọi thao tác thêm, sửa, xóa hay truy vấn dữ liệu đều được thực thi chính xác và đồng bộ với cơ sở dữ liệu SQL Server.

**a) Form Đăng nhập**

Form Đăng nhập (*Login*) giữ vai trò là lớp bảo vệ đầu tiên của hệ thống, đảm bảo chỉ những người dùng hợp lệ mới có thể truy cập vào các chức năng quản lý của ứng dụng. Giao diện được xây dựng bằng thư viện Tkinter, kết hợp giữa các thành phần nhập liệu cơ bản và các yếu tố đồ họa như ảnh nền (*nen\_python.png*) và biểu tượng người dùng (*user.png*), nhằm tạo ra một cửa sổ trực quan, dễ sử dụng và có tính thẩm mỹ cao. Bố cục form được tổ chức thành hai khu vực chính: khu vực hiển thị hình ảnh nhận diện và khu vực dành cho các trường nhập liệu xác thực.

Trong khu vực xác thực, giao diện cung cấp hai trường thông tin bắt buộc: “Tên đăng nhập” và “Mật khẩu”. Trường mật khẩu được thiết lập chế độ ẩn ký tự để tăng cường quyền riêng tư, đồng thời hỗ trợ chức năng hiển thị/tắt hiển thị (toggle password) giúp người dùng dễ dàng kiểm tra lại dữ liệu nhập. Các nút lệnh như “Đăng nhập” và “Thoát” được bố trí hợp lý, hỗ trợ thao tác nhanh và thân thiện. Ngoài ra, form cho phép kích hoạt đăng nhập bằng phím Enter, tạo sự thuận tiện trong quá trình sử dụng.

*Về mặt chức năng*, form thực hiện quá trình xác thực người dùng theo cơ chế bảo mật một chiều. Khi người dùng gửi yêu cầu đăng nhập, mật khẩu thô sẽ được mã hóa bằng thuật toán SHA-256 thông qua hàm *hash\_password*. Chuỗi mã hóa này được so sánh với giá trị đã được lưu trữ trong bảng *TaiKhoan* của cơ sở dữ liệu. Nhờ cơ chế so khớp hash, mật khẩu gốc hoàn toàn không bị lưu hoặc xử lý dưới dạng văn bản thuần túy, đảm bảo an toàn dữ liệu ở mức cao.

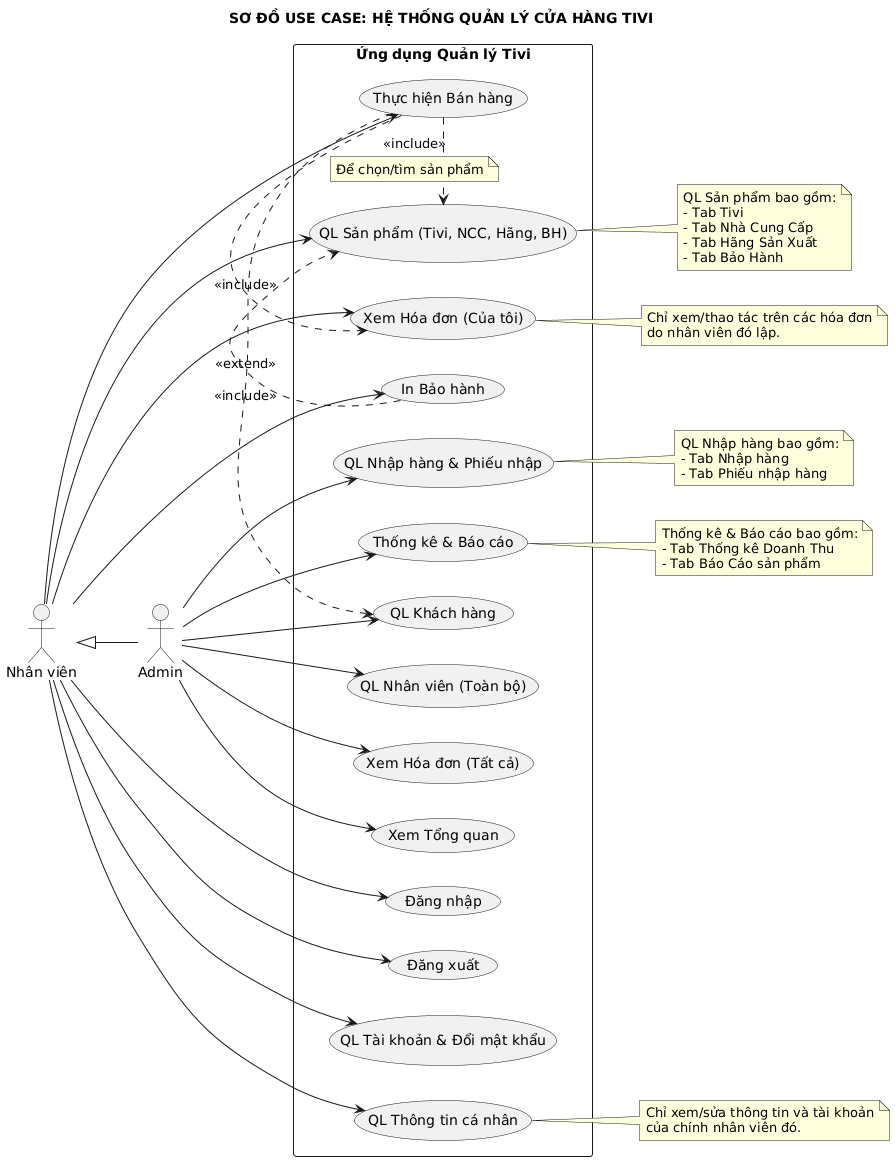
Nếu xác thực thành công, form Đăng nhập sẽ khởi tạo và chuyển quyền điều khiển sang *Form chính* (App), đồng thời truyền mã người dùng để thiết lập phiên làm việc và kích hoạt cơ chế phân quyền (Admin hoặc Nhân viên). Ngược lại, nếu thông tin đăng nhập không hợp lệ, hệ thống sẽ phản hồi bằng cảnh báo và yêu cầu người dùng thử lại. Với cách thiết kế này, form Đăng nhập không chỉ đảm bảo khả năng truy cập an toàn mà còn cung cấp trải nghiệm giao diện nhất quán, hiện đại và dễ sử dụng trong toàn bộ hệ thống.

*Về cơ chế phân quyền*, hệ thống áp dụng mô hình phân quyền hai tầng, bao gồm Admin và Nhân viên, trong đó Admin kế thừa toàn bộ quyền hạn của Nhân viên và được bổ sung thêm các chức năng quản trị nâng cao. Cơ chế phân quyền này đảm bảo mỗi người dùng chỉ có thể truy cập và thao tác trên đúng phạm vi trách nhiệm của mình, vừa tăng cường bảo mật, vừa giảm thiểu rủi ro sai sót dữ liệu trong quá trình vận hành.

Ở tầng quyền thấp hơn, Nhân viên có thể đăng nhập – đăng xuất, quản lý thông tin tài khoản cá nhân, thực hiện nghiệp vụ bán hàng và xem các hóa đơn mà chính họ đã lập. Ngoài ra, Nhân viên được phép truy cập nhóm chức năng quản lý sản phẩm (Tivi, Nhà cung cấp, Hãng sản xuất và Bảo hành), cũng như in phiếu bảo hành cho các giao dịch đã bán. Một số quan hệ include thể hiện rằng hoạt động bán hàng của Nhân viên phụ thuộc trực tiếp vào dữ liệu sản phẩm và khách hàng, đảm bảo quy trình nhập liệu được đồng bộ và thống nhất.

Trong khi đó, Admin đóng vai trò quản trị toàn hệ thống. Bên cạnh toàn bộ chức năng của Nhân viên, Admin có quyền quản lý danh sách Nhân viên, xem toàn bộ hóa đơn, duyệt hoặc hủy phiếu nhập hàng, thực hiện các nghiệp vụ nhập kho, và truy cập các công cụ thống kê – báo cáo tổng hợp. Các chức năng này giúp Admin nắm được tình hình vận hành chung, kiểm tra tồn kho, quản lý lịch sử giao dịch và giám sát hiệu suất kinh doanh của cửa hàng một cách toàn diện.

Mối quan hệ giữa các Use Case trong sơ đồ minh họa rõ sự phụ thuộc logic: chức năng xem/sửa thông tin cá nhân của Nhân viên chỉ giới hạn trong tài khoản của chính họ; hóa đơn hiển thị cho Nhân viên chỉ là hóa đơn họ lập; trong khi Admin có quyền truy cập không hạn chế. Ngoài ra, một số chức năng như in bảo hành được mở rộng từ các nhóm quản lý sản phẩm nhằm tăng tính tiện lợi. Nhờ cơ chế phân quyền rõ ràng này, hệ thống đảm bảo sự an toàn dữ liệu, phân tách trách nhiệm hợp lý và hỗ trợ vận hành cửa hàng một cách hiệu quả và nhất quán.



Hình 28: Sơ đồ usecase – Hệ thống Quản lý cửa hành tivi

**b) Form chính (App)**

Module *App* giữ vai trò là trung tâm điều phối của toàn bộ hệ thống, chịu trách nhiệm quản lý phiên làm việc của người dùng cũng như kiểm soát quá trình điều hướng giữa các module nghiệp vụ. Ngay sau khi người dùng đăng nhập thành công, lớp *App* sẽ tiếp nhận mã tài khoản và sử dụng giá trị này xuyên suốt phiên làm việc để thực hiện phân quyền truy cập và lọc dữ liệu tương ứng với từng vai trò. Đồng thời, *App* thiết lập kết nối với cơ sở dữ liệu SQL Server và đảm bảo việc đóng kết nối được thực hiện an toàn và có kiểm soát khi người dùng đăng xuất hoặc thoát khỏi ứng dụng, nhằm tránh rò rỉ tài nguyên và đảm bảo tính ổn định của hệ thống.

Về giao diện, *App* chịu trách nhiệm khởi tạo thanh điều hướng (*Sidebar*) với các nút chức năng được hiển thị theo cơ chế phân quyền động. Những module thuộc nhóm chức năng quản trị như “Tổng quan”, “Nhập hàng” hoặc “Thống kê” chỉ xuất hiện khi người đăng nhập có quyền quản trị viên. Bên cạnh đó, hệ thống triển khai cơ chế tải chậm (*Lazy Loading*), theo đó các module giao diện bên trong chỉ được khởi tạo khi người dùng truy cập lần đầu tiên. Điều này giúp giảm thời gian tải ban đầu và tối ưu hóa bộ nhớ trong quá trình vận hành.

Trong quá trình điều hướng, ứng dụng sử dụng phương thức *tkraise()* để hiển thị chính xác module đang được chọn, đồng thời tự động gọi phương thức *load\_data()* (nếu tồn tại) của module đó nhằm đảm bảo dữ liệu luôn được cập nhật mỗi khi người dùng chuyển đổi giữa các chức năng. Giao diện cũng được cá nhân hóa với ảnh đại diện và lời chào tương ứng với tài khoản đăng nhập, góp phần nâng cao trải nghiệm người dùng.

Cuối cùng, *App* tự động điều hướng đến module mặc định dựa trên vai trò người dùng. Cụ thể, tài khoản quản trị sẽ được chuyển đến giao diện Quản lý sản phẩm, trong khi nhân viên thông thường được đưa đến giao diện Quản lý nhân viên. Cơ chế này giúp người dùng nhanh chóng bắt đầu công việc với đúng phạm vi chức năng của mình, đảm bảo tính hiệu quả và nhất quán trong quá trình sử dụng hệ thống.

**c) Form Tổng quan**

Module Tổng Quan được xây dựng với mục tiêu cung cấp cho nhà quản trị một giao diện trực quan tổng hợp toàn bộ hoạt động của cửa hàng Tivi. Đây là module mang tính chất “Dashboard”, giúp Admin quan sát nhanh các chỉ số quan trọng thông qua hệ thống thẻ thống kê và biểu đồ doanh thu. Trong phần hiển thị số liệu tổng hợp, module sử dụng hàm *hienthi\_dulieu()* để thực hiện nhiều truy vấn đếm (*COUNT*) trực tiếp từ cơ sở dữ liệu SQL Server, bao gồm số lượng Nhân viên, Khách hàng, Hãng sản xuất, Nhà cung cấp, Sản phẩm Tivi và Phiếu nhập hàng. Các kết quả này được trình bày trên các thẻ thông tin (stat cards) với thiết kế nổi bật nhằm hỗ trợ người quản trị nắm bắt tức thời trạng thái hoạt động hiện tại của hệ thống.

Bên cạnh việc tổng hợp số liệu, module còn đóng vai trò phân tích tình hình doanh thu theo năm thông qua biểu đồ cột được vẽ bằng thư viện Matplotlib. Hàm *ve\_bieu\_do()* thực hiện truy vấn tổng doanh thu (SUM) của các hóa đơn đã thanh toán và nhóm theo từng tháng trong năm được chọn. Để hỗ trợ theo dõi doanh thu theo thời gian, module trang bị cơ chế điều hướng năm thông qua hai nút chuyển năm và danh sách các năm có dữ liệu được xác định bởi hàm *lay\_danh\_sach\_nam()*. Biểu đồ được định dạng với 12 cột tương ứng 12 tháng, đồng thời áp dụng bộ định dạng tiền tệ giúp hiển thị doanh thu chính xác và trực quan. Toàn bộ dữ liệu trong module được làm mới đồng bộ thông qua hàm *load\_data()*, đảm bảo rằng mỗi lần truy cập, Admin đều nhận được các thông tin cập nhật và chính xác nhất.

**d) Form Quản lý Tivi**

Module Tivi giữ vai trò trung tâm trong hệ thống quản lý cửa hàng, cho phép Admin thực hiện toàn bộ các nghiệp vụ liên quan đến danh mục và tồn kho Tivi. Giao diện của module được tổ chức theo bố cục trực quan, bao gồm vùng nhập liệu và vùng hiển thị dữ liệu. Tại khu vực nhập liệu, người dùng có thể khai báo đầy đủ các thông tin đặc tả của sản phẩm như Mã Tivi, Tên, Hãng sản xuất, Kích thước, Độ phân giải, Giá bán, Mô tả và Năm sản xuất. Hệ thống hỗ trợ hiển thị hình ảnh của Tivi thông qua Canvas và thư viện PIL, cho phép người dùng xem trước hình ảnh lưu trong cơ sở dữ liệu hoặc ảnh được chọn từ máy cục bộ. Đồng thời, combobox lựa chọn Hãng sản xuất được tải động từ cơ sở dữ liệu nhằm bảo đảm dữ liệu thống nhất và chính xác. Vùng hiển thị dữ liệu sử dụng Treeview để trình bày danh sách các sản phẩm, cho phép lựa chọn từng dòng để chỉnh sửa, xoá hoặc tìm kiếm nhanh theo nhiều tiêu chí.

Về mặt xử lý nghiệp vụ, module triển khai đầy đủ các chức năng CRUD thông qua các hàm thêm, sửa, xoá và lưu dữ liệu. Các thao tác chỉnh sửa không tác động trực tiếp lên cơ sở dữ liệu mà được lưu tạm vào các danh sách thay đổi và chỉ được ghi nhận chính thức khi người dùng xác nhận bằng chức năng lưu. Cơ chế này nâng cao tính an toàn và hạn chế sai sót khi thao tác dữ liệu. Ngoài ra, module cũng tích hợp bộ kiểm tra dữ liệu đầu vào nhằm bảo đảm tính hợp lệ trước khi ghi nhận, bao gồm kiểm tra ràng buộc bắt buộc, định dạng số, giới hạn năm sản xuất, và kiểm tra trùng lặp khóa chính để tránh phá vỡ tính toàn vẹn dữ liệu. Chức năng quản lý ảnh cũng được xử lý đầy đủ, khi ảnh sản phẩm được chuyển đổi sang dạng nhị phân và lưu trữ an toàn trong cơ sở dữ liệu SQL Server.

**e) Form Quản lý hãng sản xuất**

Module Quản lý hãng sản xuất giữ vai trò quản lý danh mục các hãng sản xuất liên quan trực tiếp đến sản phẩm Tivi, nhằm bảo đảm tính thống nhất và chuẩn hóa dữ liệu trong toàn hệ thống. Giao diện của module cung cấp các trường nhập liệu mã hãng, tên hãng và quốc gia, đồng thời hiển thị danh sách đầy đủ các hãng đã có thông qua bảng Treeview. Các thao tác thêm, sửa và xóa được thực hiện theo mô hình CRUD, với cơ chế lưu tạm thời để người dùng có thể xem xét và kiểm soát thay đổi trước khi ghi vĩnh viễn vào cơ sở dữ liệu. Đặc biệt, module tích hợp nhiều cơ chế kiểm soát dữ liệu như ràng buộc bắt buộc, kiểm tra trùng lặp mã và tên hãng, cũng như kiểm tra khóa ngoại để đảm bảo dữ liệu luôn tuân thủ cấu trúc và không gây xung đột với các bảng liên quan như bảng Tivi.

Bên cạnh khả năng chỉnh sửa dữ liệu, module còn hỗ trợ đầy đủ các chức năng hiển thị và tìm kiếm nhằm tạo thuận tiện tối đa cho người sử dụng. Danh sách hãng sản xuất được tải tự động mỗi khi mở module, và người dùng có thể chọn trực tiếp một dòng trong bảng để đưa thông tin lên form chỉnh sửa. Chức năng tìm kiếm hoạt động dựa trên truy vấn LIKE, cho phép lọc dữ liệu theo mã hoặc tên hãng một cách linh hoạt và nhanh chóng. Ngoài ra, chức năng làm mới giúp khôi phục danh sách về trạng thái ban đầu, đồng thời xóa toàn bộ dữ liệu đang hiển thị trên form, hỗ trợ người dùng thao tác thuận tiện và chính xác hơn trong quá trình quản lý danh mục hãng sản xuất.

**f) Form Quản lý nhà cung cấp**

Module Quản lý nhà cung cấp giữ vai trò quản lý danh mục các nhà cung cấp của cửa hàng – một thành phần trọng yếu trong hoạt động nhập hàng và vận hành chuỗi cung ứng. Giao diện của module được thiết kế với đầy đủ các trường thông tin cần thiết như Mã NCC, Tên NCC, Địa chỉ, Số điện thoại và Email, kết hợp với bảng Treeview để hiển thị danh sách toàn bộ nhà cung cấp hiện có. Các chức năng trong module tuân theo mô hình CRUD với cơ chế lưu tạm thời cho phép người dùng cân nhắc và điều chỉnh dữ liệu trước khi lưu vĩnh viễn vào cơ sở dữ liệu. Đặc biệt, module tích hợp hệ thống kiểm tra trùng lặp đa lớp dựa trên các trường có ràng buộc UNIQUE như mã, tên, số điện thoại và email, qua đó đảm bảo dữ liệu đầu vào luôn hợp lệ và tuân thủ các quy tắc toàn vẹn của hệ thống.

Song song với các thao tác chỉnh sửa dữ liệu, module còn cung cấp nhiều tính năng hỗ trợ giúp việc quản lý trở nên hiệu quả và minh bạch hơn. Chức năng tìm kiếm cho phép lọc danh sách theo mã hoặc tên nhà cung cấp dựa trên truy vấn LIKE, tạo điều kiện thuận lợi trong việc tra cứu thông tin nhanh chóng. Khi người dùng chọn một dòng trong bảng, dữ liệu chi tiết của nhà cung cấp sẽ được đưa lên form, hỗ trợ trực tiếp việc sửa đổi. Các cơ chế xác thực dữ liệu – như kiểm tra số điện thoại phải gồm đúng 10 chữ số, loại bỏ ký tự "0" đầu để chuẩn hóa dữ liệu so sánh, hay xử lý lỗi khóa ngoại khi xóa nhà cung cấp – đều được áp dụng nhằm đảm bảo thông tin được lưu trữ một cách chính xác, nhất quán và an toàn trong toàn bộ hệ thống.

**g) Form Quản lý bảo hành**

Module Quản lý bảo hành được xây dựng nhằm hỗ trợ việc theo dõi, cập nhật và tra cứu thông tin bảo hành của các sản phẩm đã bán trong cửa hàng. Đây là một thành phần quan trọng trong quy trình hậu mãi, góp phần đảm bảo quyền lợi của khách hàng và nâng cao chất lượng dịch vụ. Giao diện của module bao gồm các trường dữ liệu như Mã bảo hành, Mã chi tiết hóa đơn, Thời gian bảo hành, Ngày kích hoạt và Điều kiện bảo hành, kết hợp với bảng Treeview để hiển thị toàn bộ danh sách. Một trong những đặc điểm nổi bật của module là cơ chế phân quyền hiển thị và thao tác: tài khoản Admin có quyền thêm, sửa, xóa và lưu thay đổi, trong khi tài khoản Nhân viên chỉ được xem thông tin các phiếu bảo hành thuộc các hóa đơn do chính mình lập. Việc truy vấn dữ liệu được thực hiện thông qua các câu lệnh JOIN giữa nhiều bảng (BaoHanh, ChiTietHoaDon, HoaDonBan), đảm bảo việc lọc dữ liệu chính xác theo từng vai trò người dùng.

Ngoài các chức năng CRUD chuẩn hóa, module còn tích hợp hệ thống tính toán thời hạn bảo hành tự động. Dựa trên Ngày bảo hành và Thời gian bảo hành (đơn vị tháng), hệ thống xác định ngày hết hạn và hiển thị trạng thái “Còn hạn” hoặc “Hết hạn” với màu sắc trực quan, hỗ trợ người dùng nhận diện nhanh tình trạng của từng phiếu. Khi chọn một dòng trong bảng, toàn bộ dữ liệu chi tiết và trạng thái bảo hành được đổ về form để phục vụ thao tác chỉnh sửa hoặc kiểm tra. Module cũng hỗ trợ xuất phiếu bảo hành ra định dạng Word (.docx), trong đó toàn bộ thông tin liên quan đến khách hàng, sản phẩm, thời hạn bảo hành và điều kiện kèm theo được truy vấn và điền tự động vào biểu mẫu. Chức năng này hỗ trợ cả Admin và Nhân viên, đảm bảo người dùng có thể cung cấp chứng từ hợp lệ cho khách hàng một cách nhanh chóng và chính xác.

**h) Form Quản lý khách hàng**

Module Quản lý khách hàng được thiết kế để lưu trữ, cập nhật và theo dõi thông tin khách hàng, phục vụ trực tiếp cho các nghiệp vụ bán hàng, chăm sóc khách hàng và xử lý bảo hành về sau. Giao diện của module bao gồm các trường nhập liệu thông tin cá nhân và liên hệ như Mã khách hàng, Tên khách hàng, Số điện thoại, Email và Địa chỉ, cùng với bảng Treeview giúp hiển thị danh sách hiện có. Các thao tác thêm mới, chỉnh sửa, xóa và lưu đều được triển khai theo mô hình CRUD chuẩn hóa, đi kèm các cơ chế kiểm tra điều kiện đầu vào nhằm đảm bảo độ chính xác của dữ liệu. Trong quá trình thêm hoặc sửa, module tiến hành kiểm tra ràng buộc và phát hiện trùng lặp đối với những trường quan trọng mang tính duy nhất như Mã khách hàng, Số điện thoại và Email. Những bản ghi hợp lệ sẽ được lưu tạm thời vào các danh sách như ds\_them, ds\_sua hoặc ds\_xoa trước khi được ghi nhận chính thức vào cơ sở dữ liệu thông qua một giao dịch toàn vẹn, đảm bảo rằng mọi cập nhật đều nhất quán và không gây ra lỗi ràng buộc khóa ngoại.

Ngoài các chức năng quản lý dữ liệu cơ bản, module còn tích hợp hệ thống xác thực nâng cao nhằm chuẩn hóa thông tin đầu vào. Hàm kiểm tra dữ liệu buộc người dùng nhập đầy đủ các trường bắt buộc, đồng thời áp dụng quy tắc định dạng số điện thoại gồm đúng 10 chữ số. Bên cạnh đó, cơ chế chuẩn hóa số điện thoại thông qua việc loại bỏ số 0 ở đầu chuỗi được sử dụng để tăng độ chính xác khi kiểm tra trùng lặp. Trong quá trình thao tác, người dùng có thể lựa chọn một dòng trong bảng để tải dữ liệu lên form chỉnh sửa hoặc tra cứu chi tiết. Chức năng tìm kiếm hỗ trợ lọc danh sách theo Mã hoặc Tên khách hàng bằng toán tử LIKE, giúp việc tra cứu diễn ra nhanh chóng và hiệu quả. Các thao tác hủy tìm kiếm hoặc làm mới dữ liệu đều được hỗ trợ, giúp khôi phục danh sách khách hàng ban đầu một cách thuận tiện.

**i) Form Quản lý nhân viên**

Module Quản lý nhân viên được xây dựng nhằm phục vụ việc quản lý hồ sơ nhân sự của cửa hàng, bao gồm thông tin cá nhân, dữ liệu liên hệ và ảnh đại diện của nhân viên. Một đặc điểm quan trọng của module này là cơ chế phân quyền động, cho phép hệ thống tự điều chỉnh giao diện và chức năng tùy theo tài khoản đăng nhập. Khi tải dữ liệu, hệ thống sẽ xác định vai trò người dùng và quyết định phạm vi hiển thị: tài khoản Admin được phép truy cập toàn bộ dữ liệu, trong khi tài khoản Nhân viên chỉ được xem thông tin của riêng mình. Từ đó, các thành phần giao diện như bảng dữ liệu, khu vực tìm kiếm và các nút CRUD sẽ tự động ẩn hoặc vô hiệu hóa khi không phù hợp với quyền hạn của người dùng.

Về mặt nghiệp vụ, module cung cấp đầy đủ các chức năng quản lý hồ sơ theo mô hình CRUD. Người dùng (Admin) có thể nhập các thông tin cần thiết như Mã nhân viên, Họ tên, Giới tính, Ngày sinh, Số điện thoại và Số CCCD. Ngoài dữ liệu văn bản, module còn hỗ trợ quản lý ảnh đại diện thông qua khả năng tải ảnh từ tệp, chuyển đổi thành dữ liệu nhị phân và hiển thị trực tiếp trên giao diện. Các thao tác thêm mới, chỉnh sửa và xóa không được ghi vào cơ sở dữ liệu ngay lập tức mà được lưu tạm thời trong các bộ đệm (ds\_them, ds\_sua, ds\_xoa), đảm bảo người dùng có thể điều chỉnh, xem lại trước khi lưu vào CSDL thông qua một giao dịch trung tâm.

Bên cạnh đó, module tích hợp cơ chế kiểm tra dữ liệu đầu vào nghiêm ngặt nhằm đảm bảo tính chính xác và duy nhất của thông tin nhân viên. Các trường dữ liệu quan trọng như Số điện thoại và CCCD đều được kiểm tra định dạng (phải là số, đủ độ dài quy định), đồng thời được đối chiếu để tránh trùng lặp với các bản ghi hiện có. Người dùng cấp quản trị còn có thể tra cứu nhanh nhân viên thông qua chức năng tìm kiếm theo Mã hoặc Tên. Cuối cùng, hàm ghi dữ liệu chính (luu) sẽ thực hiện toàn bộ các truy vấn INSERT, UPDATE hoặc DELETE vào cơ sở dữ liệu và sử dụng cơ chế hoàn tác (rollback) nếu có lỗi xảy ra, đảm bảo tính toàn vẹn của hệ thống dữ liệu.

**k) Form Quản lý bán hàng**

Module Bán hàng được thiết kế như trung tâm xử lý giao dịch trong hệ thống quản lý cửa hàng Tivi, nơi các hóa đơn bán hàng được tạo lập, tính toán và chuẩn bị cho quy trình thanh toán. Đây là một module có độ phức tạp cao vì kết hợp đồng thời nhiều lớp nghiệp vụ: quản lý thông tin hóa đơn, kiểm soát tồn kho, tính toán chi tiết sản phẩm và phân quyền người dùng. Giao diện cho phép nhập các thông tin chung của hóa đơn như mã hóa đơn, ngày bán, nhân viên phụ trách và khách hàng. Ngoài ra, giá bán của sản phẩm được tự động truy xuất từ dữ liệu đã tải trước và hiển thị khi người dùng chọn mã Tivi tương ứng. Đối với tài khoản Nhân viên, mã người lập hóa đơn sẽ được tự động điền và khóa, đảm bảo tính toàn vẹn của thông tin giao dịch.

Về xử lý nghiệp vụ, module triển khai cơ chế kiểm soát tồn kho hai lớp nhằm đảm bảo không có sản phẩm nào được bán vượt số lượng cho phép. Khi người dùng thêm chi tiết hóa đơn, hệ thống sẽ kiểm tra số lượng yêu cầu với số lượng tồn hiện tại; nếu không đủ, thao tác sẽ bị từ chối ngay lập tức. Khi một sản phẩm đã tồn tại trong danh sách chi tiết và người dùng muốn cộng dồn số lượng, hệ thống tiếp tục tính toán số lượng mới và kiểm tra lại lần nữa để tránh vượt quá tồn kho. Mọi chi tiết sản phẩm sau khi hợp lệ sẽ được thêm hoặc sửa trong bảng Treeview, tạm thời đóng vai trò như danh sách chi tiết hóa đơn trước khi ghi xuống CSDL. Đồng thời, tổng tiền của hóa đơn được cập nhật tự động mỗi lần người dùng thêm hoặc chỉnh sửa sản phẩm.

Khi đơn hàng đã hoàn chỉnh, chức năng Tạo hóa đơn thực hiện toàn bộ giao dịch ghi dữ liệu xuống cơ sở dữ liệu SQL Server theo trình tự: tạo bản ghi hóa đơn chính và tạo các bản ghi chi tiết hóa đơn tương ứng. Giao dịch chỉ được chấp nhận khi toàn bộ dữ liệu được thêm thành công, giúp đảm bảo tính nhất quán và toàn vẹn. Bên cạnh đó, module còn cung cấp các chức năng hỗ trợ như chọn dòng để sửa, tự động hiển thị chi tiết lên form, và làm mới toàn bộ giao diện để người dùng bắt đầu một đơn hàng mới. Các thao tác này giúp quy trình bán hàng trở nên mạch lạc, kiểm soát chặt chẽ và an toàn hơn trong môi trường vận hành thực tế.

**l) Form Quản lý hóa đơn**

Module Quản lý hóa đơn được xây dựng như một trung tâm kiểm soát toàn bộ các giao dịch bán hàng của hệ thống, đảm nhiệm vai trò quản lý thông tin hóa đơn, theo dõi trạng thái thanh toán và hỗ trợ tra cứu nhanh. Đây là module có tính nhạy cảm cao vì chứa dữ liệu tài chính, do đó được tích hợp cơ chế phân quyền rõ ràng. Khi tải dữ liệu, hệ thống tự động lọc hóa đơn theo vai trò: tài khoản Admin được quyền xem toàn bộ các hóa đơn đã tạo, trong khi Nhân viên chỉ xem được những hóa đơn do chính họ lập. Bên cạnh đó, chức năng tìm kiếm linh hoạt cho phép lọc theo mã hóa đơn, mã khách hàng hoặc trạng thái giao dịch, giúp việc tra cứu trở nên thuận tiện và chính xác hơn. Người dùng cũng có thể xem chi tiết hóa đơn thông qua một cửa sổ phụ, bao gồm danh sách mặt hàng đã bán, thông tin khách hàng và nhân viên phụ trách.

Trong xử lý nghiệp vụ, module triển khai chặt chẽ quy trình kiểm soát trạng thái và kiểm tra tồn kho trước khi thực hiện các hành động quan trọng như thanh toán. Chức năng thanh toán chỉ áp dụng cho các hóa đơn đang ở trạng thái “Chờ thanh toán”, và hệ thống sẽ kiểm tra số lượng tồn kho thực tế của từng sản phẩm trước khi cho phép trừ kho và cập nhật trạng thái hóa đơn. Nếu bất kỳ mặt hàng nào không đủ số lượng, giao dịch sẽ bị từ chối nhằm đảm bảo tính trung thực của dữ liệu tồn kho. Người dùng cũng có thể hủy hóa đơn trong trường hợp cần điều chỉnh hoặc hóa đơn chưa hoàn tất; hành động này chỉ thay đổi trạng thái mà không tác động đến tồn kho. Các cơ chế này giúp đảm bảo tính nhất quán và an toàn của toàn bộ quá trình bán hàng.

Về chính sách dữ liệu, module hỗ trợ xóa hóa đơn vĩnh viễn nhưng với các ràng buộc nghiêm ngặt. Chỉ những hóa đơn đã hủy, hoặc những hóa đơn đã thanh toán nhưng đã vượt quá thời gian lưu trữ tối thiểu (5 năm), mới được phép xóa khỏi hệ thống. Khi xóa, hệ thống sẽ xử lý theo đúng thứ tự khóa ngoại, đảm bảo các bản ghi liên quan như bảo hành và chi tiết hóa đơn được xóa trước để tránh vi phạm toàn vẹn dữ liệu. Ngoài ra, module còn tích hợp chức năng xuất hóa đơn ra file Word (.docx) với định dạng chuyên nghiệp sử dụng thư viện python-docx. Phiếu in bao gồm thông tin khách hàng, thông tin giao dịch và bảng chi tiết sản phẩm, đảm bảo đáp ứng yêu cầu lưu trữ và nghiệp vụ thực tế.

**m) Form Quản lý nhập hàng**

Module quản lý nhập hàng được xây dựng nhằm hỗ trợ Admin trong việc tạo mới các phiếu nhập hàng và cập nhật thông tin nhập kho vào hệ thống. Đây là một trong những nghiệp vụ quan trọng của bài toán quản lý chuỗi cung ứng, bảo đảm dữ liệu tồn kho luôn chính xác và phản ánh kịp thời tình trạng hàng hóa. Giao diện cho phép nhập các thông tin tổng quát của phiếu nhập như Mã phiếu, Ngày nhập, Nhân viên thực hiện và Nhà cung cấp; đồng thời hỗ trợ tải dữ liệu động vào các combobox để người dùng chỉ chọn từ danh sách hợp lệ, hạn chế tối đa sai sót nhập liệu.

Trong thao tác nghiệp vụ, người dùng có thể thêm nhiều dòng chi tiết hàng hóa vào phiếu nhập thông qua bảng tạm (Treeview). Khi thêm một mặt hàng, hệ thống kiểm tra đầy đủ các ràng buộc như mã phiếu không được trùng, thông tin sản phẩm phải hợp lệ, số lượng và giá nhập phải là số dương. Nếu mặt hàng đã tồn tại trong bảng chi tiết, hệ thống cung cấp cơ chế cộng dồn số lượng, giúp người dùng thao tác linh hoạt và giảm thiểu trùng lặp. Đồng thời, sau khi thêm dòng chi tiết đầu tiên, các trường thông tin chung của phiếu sẽ bị khóa để bảo đảm rằng tất cả các mặt hàng thuộc cùng một phiếu giao dịch. Tổng tiền của phiếu được tự động tính toán dựa trên danh sách các mặt hàng và được cập nhật theo thời gian thực.

Khi người dùng hoàn tất phiếu nhập, hệ thống sẽ ghi các thông tin này vào cơ sở dữ liệu thông qua một giao dịch SQL bao gồm hai bước: tạo phiếu nhập và thêm toàn bộ chi tiết phiếu nhập. Cơ chế giao dịch đảm bảo dữ liệu được ghi đồng bộ; nếu xảy ra lỗi ở bất kỳ bước nào, toàn bộ quá trình sẽ được hoàn tác để duy trì tính toàn vẹn của cơ sở dữ liệu. Sau khi tạo phiếu thành công, module sẽ làm mới toàn bộ giao diện và cập nhật danh sách phiếu nhập ở module quản lý tương ứng, giúp Admin theo dõi dữ liệu một cách liên tục và đầy đủ.

**n) Form Quản lý phiếu nhập**

Module Quản lý phiếu nhập hàng được xây dựng nhằm hỗ trợ Admin trong việc quản lý các phiếu nhập đã được tạo trước đó, đóng vai trò như tầng kiểm soát cuối cùng của quy trình nhập kho. Đây là nơi toàn bộ các phiếu nhập được tổng hợp, phân loại theo trạng thái và cho phép nhà quản trị theo dõi lịch sử giao dịch một cách hệ thống. Giao diện của module cung cấp khả năng tải dữ liệu đầy đủ từ cơ sở dữ liệu và hỗ trợ tìm kiếm linh hoạt theo nhiều tiêu chí khác nhau như mã phiếu, mã nhà cung cấp hoặc trạng thái của phiếu. Khi cần xem chi tiết, hệ thống mở một cửa sổ riêng để hiển thị thông tin đầy đủ về nhân viên lập phiếu, nhà cung cấp, ngày lập và toàn bộ danh sách các mặt hàng được nhập trong phiếu đó.

Trong quá trình xử lý nghiệp vụ, module này chịu trách nhiệm kiểm soát trạng thái của từng phiếu nhập nhằm đảm bảo tính chính xác của dữ liệu tồn kho. Khi Admin yêu cầu duyệt phiếu, hệ thống sẽ kiểm tra trạng thái hiện tại và chỉ cho phép duyệt các phiếu đang chờ xử lý. Sau khi duyệt, số lượng tồn kho của từng sản phẩm sẽ được cộng vào bảng Tivi, hoàn tất chu trình nhập kho. Ngược lại, chức năng hủy phiếu cho phép loại bỏ các phiếu nhập không hợp lệ hoặc bị nhập sai, bằng cách chuyển trạng thái sang “Đã hủy” mà không tác động đến tồn kho. Các hành động này đều được thiết kế với cơ chế kiểm soát nhằm ngăn ngừa sai sót dữ liệu và bảo đảm sự nhất quán trong hệ thống.

Ngoài ra, module còn hỗ trợ chính sách lưu trữ và xóa phiếu nhập một cách an toàn. Hệ thống chỉ cho phép xóa vĩnh viễn các phiếu đã bị hủy hoặc những phiếu đã duyệt nhưng đã tồn tại hơn 5 năm — tuân theo quy định lưu trữ dữ liệu của doanh nghiệp. Khi thực hiện thao tác xóa, hệ thống sẽ tự động xóa trước các chi tiết phiếu nhập rồi mới xóa phiếu gốc, đảm bảo không vi phạm ràng buộc khóa ngoại. Bên cạnh đó, module cũng tích hợp chức năng in phiếu nhập dưới dạng file Word, được trình bày theo mẫu chuẩn với đầy đủ thông tin cần thiết và bố cục phù hợp cho việc lưu trữ hoặc đối chiếu chứng từ trong nghiệp vụ kho.

**o) Form Thống kê doanh thu**

Module Thống kê doanh thu được xây dựng như một công cụ phân tích tài chính chuyên sâu dành cho Admin, cho phép đánh giá toàn diện hiệu quả hoạt động kinh doanh của cửa hàng Tivi. Giao diện module cung cấp hai tùy chọn lọc thời gian (“Từ ngày” – “Đến ngày”) giúp người dùng dễ dàng xác định phạm vi thống kê theo nhu cầu. Trước khi thực hiện truy vấn, hệ thống áp dụng cơ chế kiểm tra logic thời gian nhằm đảm bảo giá trị ngày bắt đầu luôn nhỏ hơn hoặc bằng ngày kết thúc, từ đó ngăn ngừa các truy vấn sai hoặc không hợp lệ. Ngoài ra, chức năng hủy lọc giúp nhanh chóng đặt lại phạm vi thời gian và tải lại báo cáo tổng hợp của toàn bộ lịch sử giao dịch.

Trong quá trình thống kê, module triển khai các truy vấn SQL chuyên biệt để thu thập và tính toán hai chỉ số tài chính cốt lõi: Doanh thu và Lợi nhuận. Doanh thu được xác định bằng tổng giá trị của tất cả các hóa đơn ở trạng thái “Đã thanh toán”, trong khi lợi nhuận được tính dựa trên mô hình kế toán chuẩn.

|  |
| --- |
| **Lợi nhuận = Tổng doanh thu – Giá vốn hàng bán *(COGS)*** |

Giá vốn hàng bán được tính từ dữ liệu nhập kho (ChiTietPhieuNhap) bằng cách nhân số lượng đã bán với giá nhập trung bình của từng mã Tivi trong khoảng thời gian thống kê. Cách tính này đảm bảo tính chính xác và phản ánh đúng bản chất kinh tế của giao dịch.

Kết quả thống kê được hiển thị trực quan ngay trên giao diện, bao gồm tổng số hóa đơn, tổng doanh thu và lợi nhuận được trình bày với định dạng tiền tệ rõ ràng, dễ đọc. Đồng thời, bảng chi tiết (Treeview) liệt kê từng hóa đơn đã thanh toán trong phạm vi thời gian đã chọn, gồm Mã hóa đơn, Khách hàng, Ngày bán và Tổng tiền. Khi module được mở lần đầu, hệ thống tự động thực hiện thống kê toàn bộ dữ liệu để cung cấp cho Admin cái nhìn tổng thể và nhanh chóng về hiệu suất kinh doanh của cửa hàng.

**p) Form báo cáo sản phẩm**

Module Báo cáo sản phẩm được xây dựng nhằm hỗ trợ Admin phân tích sâu hiệu suất kinh doanh của từng mặt hàng Tivi, từ đó đưa ra các quyết định quản trị sản phẩm và tối ưu hóa tồn kho. Giao diện module cung cấp hai bộ lọc đồng bộ (“Mã Tivi” và “Tên Tivi”), cho phép truy xuất nhanh chóng đến một sản phẩm cụ thể. Hai Combobox này được liên kết hai chiều: khi người dùng chọn Mã Tivi, Combobox Tên Tivi sẽ tự động cập nhật giá trị tương ứng và ngược lại. Ngoài ra, hệ thống còn cung cấp chức năng tải báo cáo tổng thể gồm toàn bộ các sản phẩm (load\_baocao\_all) cũng như tùy chọn hủy lọc để nhanh chóng quay về chế độ xem toàn diện.

Về mặt xử lý dữ liệu, module sử dụng các truy vấn SQL phức hợp kết hợp với xử lý Python để tính toán các chỉ số phân tích quan trọng. Giá vốn hàng bán (COGS) của từng sản phẩm được xác định dựa trên giá nhập trung bình, được truy vấn từ bảng ChiTietPhieuNhap. Trên cơ sở đó, lợi nhuận gộp của sản phẩm được tính bằng công thức:

|  |
| --- |
| **Lợi nhuận = Số lượng bán × (Giá bán trung bình – Giá nhập trung bình)** |

Ngoài ra, hệ thống cũng tính toán tỷ lệ bán chạy (%) dựa vào số lượng bán của từng Tivi so với tổng số hàng đã bán. Những thông số này giúp Admin đánh giá độ phổ biến, hiệu suất doanh thu và lợi nhuận của từng mã sản phẩm một cách chính xác.

Kết quả phân tích được hiển thị dưới dạng bảng chi tiết (Treeview), bao gồm các cột: Số lượng bán, Tỷ lệ bán chạy, Doanh thu và Lợi nhuận. Các giá trị tiền tệ được định dạng theo chuẩn để tăng tính trực quan, trong khi các chỉ số thống kê được căn chỉnh hợp lý giúp dễ dàng so sánh. Khi Admin chọn một sản phẩm cụ thể, hàm xem\_baocao sẽ tải báo cáo đơn lẻ của sản phẩm đó, hỗ trợ phân tích chuyên sâu. Đồng thời, dữ liệu báo cáo luôn được sắp xếp giảm dần theo số lượng bán, giúp nhanh chóng nhận diện những sản phẩm bán chạy nhất và đưa ra các quyết định kinh doanh phù hợp.

**q) Giao diện Hệ thống**

Module Quản lý Tài khoản Hệ thống (frmHeThong) được thiết kế nhằm quản trị danh sách tài khoản người dùng trong toàn bộ hệ thống, đồng thời cung cấp cơ chế tự phục vụ cho Nhân viên trong việc thay đổi mật khẩu cá nhân. Hệ thống triển khai cơ chế phân quyền kép (Admin/User), nhờ đó giao diện và quyền thao tác sẽ tự động thay đổi tùy theo đối tượng đăng nhập. Ở chế độ Admin, toàn bộ Treeview hiển thị danh sách tài khoản được kích hoạt cùng các chức năng tìm kiếm và đầy đủ các nút CRUD. Ngược lại, ở chế độ Nhân viên, các thành phần quản trị (Treeview, tìm kiếm, thêm, xóa) được ẩn hoàn toàn, chỉ giữ lại khung đổi mật khẩu cá nhân với trường Tên đăng nhập ở trạng thái khóa để đảm bảo tính toàn vẹn dữ liệu.

Về mặt bảo mật, module áp dụng cơ chế mã hóa mật khẩu một chiều bằng thuật toán SHA-256 thông qua hàm hash\_password. Mật khẩu thô do Admin hoặc User nhập vào sẽ được mã hóa ngay tại thời điểm thao tác (khi thêm hoặc sửa), và chỉ chuỗi hash mới được lưu vào danh sách thay đổi tạm thời cũng như ghi vào CSDL. Trong Treeview, Admin nhìn thấy cột mật khẩu dưới dạng chuỗi che, nhưng khi chọn dòng để chỉnh sửa, hệ thống tự động tải lại hash vào ô nhập liệu nhằm hỗ trợ thay đổi mà vẫn đảm bảo không làm lộ mật khẩu gốc.

Toàn bộ quy trình quản lý đều được thực hiện theo mô hình thay đổi tạm thời (ds\_them, ds\_sua, ds\_xoa), sau đó được xác nhận bằng nút Lưu để tiến hành ghi vào cơ sở dữ liệu. Hàm luu() đóng vai trò như một giao dịch toàn diện: thực thi tuần tự các câu lệnh INSERT, UPDATE, DELETE lên bảng TaiKhoan và tự động rollback nếu phát sinh lỗi ràng buộc, đảm bảo tính nhất quán của dữ liệu hệ thống. Ngoài ra, chức năng tìm kiếm hỗ trợ Admin truy xuất nhanh tài khoản dựa trên tên đăng nhập, phục vụ nhu cầu quản trị số lượng lớn người dùng trong môi trường vận hành thực tế.

**C. HƯỚNG DẪN CÀI ĐẶT & CODE ỨNG DỤNG**

Ứng dụng được xây dựng nhằm hỗ trợ quá trình quản lý hoạt động kinh doanh của cửa hàng Tivi một cách hiệu quả, trực quan và hiện đại thông qua sự kết hợp giữa Python, thư viện giao diện Tkinter và hệ quản trị cơ sở dữ liệu SQL Server. Hệ thống cung cấp đầy đủ các giao diện chức năng như quản lý thông tin Tivi, quản lý nhân viên, khách hàng, nhà cung cấp, bán hàng, nhập hàng, hóa đơn, thống kê và báo cáo. Mỗi giao diện đều được thiết kế với các trường nhập liệu chuẩn, bảng hiển thị dữ liệu và các chức năng CRUD (Tạo, Đọc , Sửa, Xóa) giúp người dùng dễ dàng thao tác và quản lý dữ liệu.

Dữ liệu của ứng dụng được lưu trữ trực tiếp và lâu dài trong SQL Server, đảm bảo tính toàn vẹn, tính nhất quán và hạn chế tối đa thất thoát dữ liệu khi ứng dụng đóng hoặc gặp sự cố. Giao diện Tkinter được thiết kế thân thiện, dễ sử dụng, bố cục rõ ràng và tương thích với yêu cầu nghiệp vụ thực tế của cửa hàng Tivi. Thông qua việc tích hợp giữa xử lý giao diện và cơ sở dữ liệu, ứng dụng mang đến một hệ thống quản lý đồng bộ, hỗ trợ tối ưu cho công tác vận hành và quản trị cửa hàng.

**1. Cài đặt môi trường**

Ứng dụng được xây dựng bằng Python và sử dụng một số thư viện hỗ trợ để tạo giao diện, xử lý ảnh, chọn ngày và kết nối cơ sở dữ liệu SQL Server. Một số thư viện đã có sẵn trong Python, trong khi các thư viện còn lại cần được cài đặt thủ công.

**a) pyodbc**

- Lệnh cài đặt:

pig install pyodbc

- Chức năng: Thư viện dùng để kết nối ứng dụng Python với SQL Server. Toàn bộ thao tác CRUD của ứng dụng (Thêm, Sửa, Xóa, Tìm kiếm, Truy vấn thống kê) đều được thực hiện thông qua pyodbc.

- Lưu ý: Máy cần có **ODBC Driver 17 for SQL Server** để kết nối thành công.

**b) tkcalendar**

- Lệnh cài đặt:

pig install tkcalendar

- Chức năng: Cung cấp widget DateEntry cho phép chọn ngày trực quan trong các form như: ngày sinh, ngày bán, ngày nhập hàng, ngày bảo hành, v.v.

**c) Pillow (PIL)**

- Lệnh cài đặt:

pig install pillow

- Chức năng: Dùng để xử lý hình ảnh (đọc, hiển thị, thu nhỏ ảnh), hỗ trợ hiển thị ảnh Tivi, ảnh nhân viên và lưu ảnh dưới dạng nhị phân vào SQL Server.

**d) python-docx**

- Lệnh cài đặt:

pig install python-docx

- Chức năng: Đây là thư viện hỗ trợ đầy đủ các chức năng cần thiết để xuất báo cáo, in ấn hoặc sinh tài liệu tự động từ dữ liệu của chương trình.

**e) Microsoft ODBC Driver for SQL Server(nếu thiết bị chưa có sẵn)**

- Link cài đặt:

[*https://learn.microsoft.com/sql/connect/odbc/download-odbc-driver-for-sql-server*](https://learn.microsoft.com/sql/connect/odbc/download-odbc-driver-for-sql-server)

- Chức năng: Hỗ trợ pyodbc tạo kết nối và thực thi các truy vấn SQL.

Ngoài ra, cần tải thư mục images tại đường dẫn để bổ sung các ảnh cần thiết trên hệ thống

*https://drive.google.com/drive/u/1/folders/1gXUhJnLi24iNlMQq9aZ0W3zhxoOmh6Ts*

### **2. Chuẩn bị CSDL SQL Server**

Trước khi chạy ứng dụng, bạn cần tạo cơ sở dữ liệu và các bảng trên SQL Server. Người dùng có thể sử dụng SQL Server Management Studio (SSMS) hoặc bất kỳ công cụ làm việc với SQL Server nào để khởi tạo Database,Table và Trigger cần thiết.

|  |
| --- |
| --- TẠO DATABASE QLTV ---  CREATE DATABASE QLTV  ON (  NAME = 'QLTV',  FILENAME = 'D:\QLTV\_data.mdf',  SIZE = 8MB,  MAXSIZE = 80MB,  FILEGROWTH = 5MB  )  LOG ON (  NAME = 'QLTV\_log',  FILENAME = 'D:\QLTV\_log.ldf',  SIZE = 8MB,  MAXSIZE = 80MB,  FILEGROWTH = 5MB  );  GO  USE QLTV;  GO  --- TẠO CÁC TABLE ---  CREATE TABLE HangSanXuat (  MaHang NVARCHAR(10) PRIMARY KEY,  TenHang NVARCHAR(100) NOT NULL,  QuocGia NVARCHAR(50),  CONSTRAINT UQ\_HangSanXuat\_TenHang UNIQUE (TenHang)  );  GO  CREATE TABLE KhachHang (  MaKH NVARCHAR(10) PRIMARY KEY,  TenKH NVARCHAR(100) NOT NULL,  SoDienThoai NVARCHAR(10) UNIQUE,  DiaChi NVARCHAR(255),  Email NVARCHAR(100) UNIQUE  );  GO  CREATE TABLE NhanVien (  MaNV NVARCHAR(10) PRIMARY KEY,  HinhAnh VARBINARY(MAX),  TenNV NVARCHAR(100) NOT NULL,  GioiTinh NVARCHAR(10) CHECK (GioiTinh IN (N'Nam', N'Nữ')),  NgaySinh DATE,  SoDienThoai NVARCHAR(10) UNIQUE,  CCCD NVARCHAR(12) UNIQUE  );  GO  CREATE TABLE NhaCungCap (  MaNCC NVARCHAR(10) PRIMARY KEY,  TenNCC NVARCHAR(100) UNIQUE,  DiaChi NVARCHAR(255),  SoDienThoai NVARCHAR(10) UNIQUE,  Email NVARCHAR(100) UNIQUE  );  GO  CREATE TABLE TaiKhoan (  TenDangNhap NVARCHAR(10) PRIMARY KEY,  MatKhau NVARCHAR(100) DEFAULT 'a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3'  );  GO  CREATE TABLE Tivi (  MaTivi NVARCHAR(10) PRIMARY KEY,  HinhAnh VARBINARY(MAX),  TenTivi NVARCHAR(100) NOT NULL,  MaHang NVARCHAR(10) NOT NULL,  KichThuoc NVARCHAR(20),  DoPhanGiai NVARCHAR(20),  GiaBan DECIMAL(18,2) NOT NULL DEFAULT 0,  SoLuongTon INT DEFAULT 0,  NamSanXuat INT,  MoTa NVARCHAR(255),  CONSTRAINT FK\_Tivi\_Hang FOREIGN KEY (MaHang) REFERENCES HangSanXuat (MaHang)  );  GO  CREATE TABLE PhieuNhapHang (  MaPhieuNhap NVARCHAR(10) PRIMARY KEY,  NgayNhap DATE DEFAULT GETDATE(),  MaNV NVARCHAR(10) NOT NULL,  MaNCC NVARCHAR(10) NOT NULL,  TrangThai NVARCHAR(20) DEFAULT N'Đợi duyệt',  TongTien DECIMAL(18,2) DEFAULT 0,  CONSTRAINT FK\_PhieuNhap\_NhanVien FOREIGN KEY (MaNV) REFERENCES NhanVien (MaNV),  CONSTRAINT FK\_PhieuNhap\_NCC FOREIGN KEY (MaNCC) REFERENCES NhaCungCap (MaNCC)  );  GO  CREATE TABLE ChiTietPhieuNhap (  MaPhieuNhap NVARCHAR(10) NOT NULL,  MaTivi NVARCHAR(10) NOT NULL,  SoLuong INT CHECK (SoLuong > 0),  GiaNhap DECIMAL(18,2) CHECK (GiaNhap >= 0),  ThanhTien AS (SoLuong \* GiaNhap) PERSISTED,  PRIMARY KEY (MaPhieuNhap, MaTivi),  CONSTRAINT FK\_CTPN\_PhieuNhap FOREIGN KEY (MaPhieuNhap) REFERENCES PhieuNhapHang (MaPhieuNhap),  CONSTRAINT FK\_CTPN\_Tivi FOREIGN KEY (MaTivi) REFERENCES Tivi (MaTivi)  );  GO  CREATE TABLE HoaDonBan (  MaHD NVARCHAR(10) PRIMARY KEY,  NgayBan DATE DEFAULT GETDATE(),  MaNV NVARCHAR(10) NOT NULL,  MaKH NVARCHAR(10) NULL,  TongTien DECIMAL(18,2) DEFAULT 0,  TrangThai NVARCHAR(20) DEFAULT N'Chờ thanh toán',  CONSTRAINT FK\_HoaDon\_NhanVien FOREIGN KEY (MaNV) REFERENCES NhanVien (MaNV),  CONSTRAINT FK\_HoaDon\_KhachHang FOREIGN KEY (MaKH) REFERENCES KhachHang (MaKH)  );  GO  CREATE TABLE ChiTietHoaDon (  MaCTHD NVARCHAR(10) PRIMARY KEY,  MaHD NVARCHAR(10) NOT NULL,  MaTivi NVARCHAR(10) NOT NULL,  SoLuong INT CHECK (SoLuong > 0),  DonGia DECIMAL(18,2) CHECK (DonGia >= 0),  ThanhTien AS (SoLuong \* DonGia) PERSISTED,  CONSTRAINT FK\_CTHD\_HoaDon FOREIGN KEY (MaHD) REFERENCES HoaDonBan (MaHD),  CONSTRAINT FK\_CTHD\_Tivi FOREIGN KEY (MaTivi) REFERENCES Tivi (MaTivi),  CONSTRAINT UQ\_CTHD\_MaHD\_MaTivi UNIQUE (MaHD, MaTivi)  );  GO  CREATE TABLE BaoHanh (  MaBH NVARCHAR(10) PRIMARY KEY,  MaCTHD NVARCHAR(10) NOT NULL,  ThoiGianBaoHanh INT CHECK (ThoiGianBaoHanh > 0),  DieuKien NVARCHAR(255),  NgayBaoHanh DATE NOT NULL DEFAULT GETDATE(),  CONSTRAINT FK\_BaoHanh\_ChiTietHoaDon FOREIGN KEY (MaCTHD) REFERENCES ChiTietHoaDon (MaCTHD)  );  GO  --- TẠO TRIGGER CẬP NHẬT TỔNG TIỀN CHO PHIẾU NHÂP HÀNG ---  CREATE TRIGGER trg\_CapNhatTongTien\_PhieuNhap  ON ChiTietPhieuNhap AFTER INSERT, UPDATE, DELETE  AS BEGIN  SET NOCOUNT ON;  UPDATE p SET TongTien = ISNULL((  SELECT SUM(ThanhTien) FROM ChiTietPhieuNhap c  WHERE c.MaPhieuNhap = p.MaPhieuNhap  ), 0)  FROM PhieuNhapHang p  WHERE p.MaPhieuNhap IN (SELECT MaPhieuNhap FROM inserted UNION SELECT MaPhieuNhap FROM deleted);  END;  GO  --- TẠO TRIGGER CẬP NHẬT TỔNG TIỀN CHO HÓA ĐƠN ---  CREATE TRIGGER trg\_CapNhatTongTien\_HoaDon  ON ChiTietHoaDon AFTER INSERT, UPDATE, DELETE  AS BEGIN  SET NOCOUNT ON;  UPDATE p SET TongTien = ISNULL((  SELECT SUM(ThanhTien) FROM ChiTietHoaDon c  WHERE c.MaHD = p.MaHD  ), 0)  FROM HoaDonBan p  WHERE p.MaHD IN (SELECT MaHD FROM inserted UNION SELECT MaHD FROM deleted);  END;  GO |

### **3. Viết code ứng dụng với Tkinter + SQL Server**

**FORM ĐĂNG NHẬP (LOGIN)**

|  |
| --- |
| import tkinter as tk  from tkinter import ttk  import ctypes  import pyodbc  import App  import os  import sys  from PIL import Image, ImageTk  import hashlib  # === BẢNG MÀU ===  PRIMARY\_COLOR = "#378cfc"  SECONDARY\_COLOR = "#1565C0"  ACCENT\_COLOR = "#42A5F5"  HIGHLIGHT\_COLOR = "#BBDEFB"  TEXT\_COLOR = "white"  # === LÀM NÉT GIAO DIỆN ===  try:      ctypes.windll.shcore.SetProcessDpiAwareness(1)  except Exception:      pass  class Login(tk.Tk):      def \_\_init\_\_(self):          super().\_\_init\_\_()          self.title("ĐĂNG NHẬP HỆ THỐNG QUẢN LÝ NHÂN SỰ")          self.center\_window(850, 580)          self.resizable(False, False)          self.configure(bg="white")          # ==== CHUỖI KẾT NỐI SQL ====          self.conn = pyodbc.connect(              "DRIVER={ODBC Driver 17 for SQL Server};"              "SERVER=DESKTOP-LJVV0KQ;"              "DATABASE=QLTV;"              "Trusted\_Connection=yes;")            # === LẤY ĐƯỜNG DẪN CHO ẢNH ===          base\_dir = os.path.dirname(os.path.abspath(sys.argv[0]))          image\_dir = os.path.join(base\_dir, "images")          icon\_path = os.path.join(image\_dir, 'icon\_tivi.ico')          self.iconbitmap(icon\_path)          # === FRAME TRÁI (WELCOME) ===          left\_frame = tk.Frame(self, width=300, height=500)          left\_frame.pack(side="left", fill="y")            # --- XỬ LÝ VÀ ĐẶT HÌNH ẢNH NỀN ---          image\_path = os.path.join(image\_dir, "nen\_python.png")            original\_image = Image.open(image\_path)          resized\_image = original\_image.resize((300, 580), Image.Resampling.LANCZOS)          self.bg\_image = ImageTk.PhotoImage(resized\_image)            background\_label = tk.Label(left\_frame, image=self.bg\_image)          background\_label.place(x=0, y=0, relwidth=1, relheight=1)            # === FRAME PHẢI (FORM ĐĂNG NHẬP) ===          right\_frame = tk.Frame(self, bg="white")          right\_frame.pack(side="right", fill="both", expand=True, padx=20, pady=20)          # --- Căn giữa toàn bộ ---          right\_frame.columnconfigure(0, weight=1)          # --- Avatar ---          user\_icon\_path = os.path.join(image\_dir, "user.png")            original\_user\_image = Image.open(user\_icon\_path)          resized\_user\_image = original\_user\_image.resize((120, 120), Image.Resampling.LANCZOS)          self.user\_image\_tk = ImageTk.PhotoImage(resized\_user\_image)          user\_avatar\_label = tk.Label(right\_frame, image=self.user\_image\_tk, bg="white")          user\_avatar\_label.grid(row=0, column=0, pady=(30, 10))          # --- SIGNIN ---          tk.Label(right\_frame, text="SIGNIN", bg="white", fg=PRIMARY\_COLOR, font=("Segoe UI", 18, "bold")).grid(row=1, column=0, pady=(0, 30))          # --- Tên đăng nhập ---          user\_frame = tk.Frame(right\_frame, bg="white")          user\_frame.grid(row=2, column=0, pady=10)          tk.Label(user\_frame, text="👤", bg="white", fg=PRIMARY\_COLOR, font=("Segoe UI", 12)).pack(side="left", padx=5)          self.txt\_user = ttk.Entry(user\_frame, width=30, font=("Segoe UI", 11))          self.txt\_user.pack(side="left")          self.txt\_user.focus()          self.txt\_user.bind("<Return>", self.dangnhap)          # --- Mật khẩu ---          pass\_frame = tk.Frame(right\_frame, bg="white")          pass\_frame.grid(row=3, column=0, pady=10)          tk.Label(pass\_frame, text="🔒", bg="white", fg=PRIMARY\_COLOR, font=("Segoe UI", 12)).pack(side="left", padx=5)          self.txt\_password = ttk.Entry(pass\_frame, width=27, font=("Segoe UI", 11), show="●")          self.txt\_password.pack(side="left")          self.txt\_password.bind("<Return>", self.dangnhap)          # Nút hiện/ẩn mật khẩu          self.show\_pass = False          self.eye\_btn = tk.Button(pass\_frame, text="👁", bg="white", bd=0, cursor="hand2", command=self.toggle\_password)          self.eye\_btn.pack(side="left", padx=3)          # --- Nút đăng nhập & thoát ---          btn\_frame = tk.Frame(right\_frame, bg="white")          btn\_frame.grid(row=4, column=0, pady=40)          self.btn\_login = tk.Button(btn\_frame, text="Đăng nhập", bg=PRIMARY\_COLOR, fg="white", font=("Segoe UI", 11, "bold"), width=12, pady=6, bd=0, activebackground=HIGHLIGHT\_COLOR, cursor="hand2", command=self.dangnhap)          self.btn\_login.pack(side="left", padx=15)          self.btn\_exit = tk.Button(btn\_frame, text="Thoát", bg="#E53935", fg="white", font=("Segoe UI", 11, "bold"), width=12, pady=6, bd=0, activebackground="#B71C1C", cursor="hand2", command=self.destroy)          self.btn\_exit.pack(side="left", padx=15)          # --- Nhãn thông báo ---          self.lbl\_error = tk.Label(right\_frame, text="", fg="red", bg="white", font=("Segoe UI", 10))          self.lbl\_error.grid(row=5, column=0)      def center\_window(self, w=850, h=500):          ws = self.winfo\_screenwidth()          hs = self.winfo\_screenheight()          x = (ws // 2) - (w // 2)          y = (hs // 2) - (h // 2)          self.geometry(f"{w}x{h}+{x}+{y}")      def toggle\_password(self):          if self.show\_pass:              self.txt\_password.config(show="●")              self.eye\_btn.config(text="👁")          else:              self.txt\_password.config(show="")              self.eye\_btn.config(text="🚫")          self.show\_pass = not self.show\_pass      def hash\_password(self, password):          # Mã hóa mật khẩu bằng SHA-256.          return hashlib.sha256(password.encode('utf-8')).hexdigest()        def dangnhap(self, event=None):          user = self.txt\_user.get().strip()          pw = self.txt\_password.get().strip()          if not user or not pw:              self.lbl\_error.config(text="Vui lòng nhập tên đăng nhập và mật khẩu!")              return          hashed\_pw = self.hash\_password(pw)          cursor = self.conn.cursor()          cursor.execute("SELECT MatKhau FROM TaiKhoan WHERE TenDangNhap = ?", (user,))          row = cursor.fetchone()          if row:              stored\_hash = row[0]              # Mật khẩu nhập vào (đã hash) có khớp với mật khẩu đã lưu không?              if hashed\_pw == stored\_hash:                   self.mo\_form(user)              else:                   self.lbl\_error.config(text="Sai tài khoản hoặc mật khẩu!")          else:              self.lbl\_error.config(text="Sai tài khoản hoặc mật khẩu!")      def mo\_form(self, user):          self.destroy()          App.App(user)  if \_\_name\_\_ == "\_\_main\_\_":      app = Login()      app.mainloop() |

**FORM CHÍNH (APP)**

|  |
| --- |
| import tkinter as tk  from tkinter import messagebox  import ctypes  import pyodbc  import os  import sys  from PIL import Image, ImageTk  import frmTongQuan as tq  import frmBanHangVaHoaDon as bhvhd  import frmHeThong as ht  import frmQuanLyKhachHang as kh  import frmQuanLyNhanVien as nv  import frmThongKeVaBaoCao as tkvbc  import frmQuanLySanPham as sp  import frmNhapHangVaPhieuNhap as nhvpn  import frmLogin  # Hien: LAPTOP-IFECMD9V  # Hoai: DESKTOP-LJVV0KQ  # === BẢNG MÀU ===  PRIMARY\_COLOR = "#0D47A1"  SECONDARY\_COLOR = "#1565C0"  ACCENT\_COLOR = "#42A5F5"  HIGHLIGHT\_COLOR = "#BBDEFB"  TEXT\_COLOR = "white"  # === LÀM NÉT GIAO DIỆN ===  try:      ctypes.windll.shcore.SetProcessDpiAwareness(1)  except Exception:      pass  class App(tk.Tk):      # === CỬA SỐ CHÍNH ===      def \_\_init\_\_(self, user):          super().\_\_init\_\_()          self.center\_window(1500, 885)          self.resizable(True, True)          self.title("HỆ THỐNG QUẢN LÝ CỬA HÀNG TIVI")          self.user = user          # === CHUỖI KẾT NỐI ===          try:              self.conn = pyodbc.connect(                  "DRIVER={ODBC Driver 17 for SQL Server};"                  "SERVER=DESKTOP-LJVV0KQ;"                  "DATABASE=QLTV;"                  "Trusted\_Connection=yes;"              )          except Exception as e:              messagebox.showerror("Lỗi kết nối", "Không thể kết nối CSDL:")              self.destroy()              return          # === LẤY ĐƯỜNG DẪN CHO ẢNH ===          base\_dir = os.path.dirname(os.path.abspath(sys.argv[0]))          self.image\_dir = os.path.join(base\_dir, "images")          icon\_path = os.path.join(self.image\_dir, "icon\_tivi.ico")          self.iconbitmap(icon\_path)          # Tạo dictionary lưu các frame nội dung          self.frames = {}          # --- Sidebar (Menu bên trái) ---          self.TaoSidleBar()          # Vùng nội dung chính          container = tk.Frame(self, bg="white")          container.pack(side="right", fill="both", expand=True)          self.container = container          self.frame\_classes = {              "TongQuan": tq.TongQuan,              "QuanLySanPham": sp.QuanLySanPham,              "QuanLyKhachHang": kh.QuanLyKhachHang,              "QuanLyNhanVien": nv.QuanLyNhanVien,              "BanHangVaHoaDon": bhvhd.BanHangVaHoaDon,              "NhapHangVaPhieuNhap": nhvpn.NhapHangVaPhieuNhap,              "ThongKeVaBaoCao": tkvbc.ThongKeVaBaoCao,              "HeThong": ht.HeThong,          }          # Cho phép container giãn đầy vùng hiển thị          container.grid\_rowconfigure(0, weight=1)          container.grid\_columnconfigure(0, weight=1)          # Hiển thị trang đầu tiên          if self.user == "admin":              self.HienThiFrame("QuanLySanPham")          else:              self.HienThiFrame("QuanLyNhanVien")          # === Đóng kết nối khi thoát app ===          self.protocol("WM\_DELETE\_WINDOW", self.on\_close)      # ==== HÀM CANH GIỮA CỬA SỔ ====      def center\_window(self, w=1500, h=885):          ws = self.winfo\_screenwidth()          hs = self.winfo\_screenheight()          x = (ws // 2) - (w // 2)          y = (hs // 2) - (h // 2)          self.geometry(f"{w}x{h}+{x}+{y}")      def TaoSidleBar(self):          # ==== SIDEBAR BÊN TRÁI====          pnlGiaoDien = tk.Frame(self, bg=SECONDARY\_COLOR, width=250)          pnlGiaoDien.pack(side="left", fill="y")          # # Ảnh avatar (placeholder)          avatar\_path = os.path.join(self.image\_dir, "tivi\_user.png")          original\_avatar = Image.open(avatar\_path)          resized\_avatar = original\_avatar.resize((80, 80), Image.Resampling.LANCZOS)          self.avatar\_tk = ImageTk.PhotoImage(resized\_avatar)          lbl\_avatar = tk.Label(pnlGiaoDien, image=self.avatar\_tk, bg=SECONDARY\_COLOR)          lbl\_avatar.pack(pady=(20, 5))          lbl\_XinChao = tk.Label(              pnlGiaoDien,              text=f"Xin chào {self.user}",              bg=SECONDARY\_COLOR,              fg="white",              font=("Segoe UI", 12, "bold"),          ).pack()          if self.user == "admin":              # Danh mục bên trái              btn\_tongquan = tk.Button(                  pnlGiaoDien,                  text="🏠 Tổng quan",                  command=lambda: self.HienThiFrame("TongQuan"),                  \*\*self.DinhDangNut(),              )              btn\_tongquan.pack(fill="x")          btn\_quanlysanpham = tk.Button(              pnlGiaoDien,              text="📦 Quản lý Sản phẩm",              command=lambda: self.HienThiFrame("QuanLySanPham"),              \*\*self.DinhDangNut(),          )          btn\_quanlysanpham.pack(fill="x")          btn\_quanlykhachhang = tk.Button(              pnlGiaoDien,              text="👥 Quản lý Khách hàng",              command=lambda: self.HienThiFrame("QuanLyKhachHang"),              \*\*self.DinhDangNut(),          )          btn\_quanlykhachhang.pack(fill="x")          btn\_quanlynhanvien = tk.Button(              pnlGiaoDien,              text="🧑‍💼 Quản lý Nhân viên",              command=lambda: self.HienThiFrame("QuanLyNhanVien"),              \*\*self.DinhDangNut(),          )          btn\_quanlynhanvien.pack(fill="x")          btn\_banhangvahoadon = tk.Button(              pnlGiaoDien,              text="💰 Bán hàng & Hóa đơn",              command=lambda: self.HienThiFrame("BanHangVaHoaDon"),              \*\*self.DinhDangNut(),          )          btn\_banhangvahoadon.pack(fill="x")          if self.user == "admin":              btn\_nhaphangvaphieunhap = tk.Button(                  pnlGiaoDien,                  text="📦 Nhập hàng & Phiếu nhập",                  command=lambda: self.HienThiFrame("NhapHangVaPhieuNhap"),                  \*\*self.DinhDangNut(),              )              btn\_nhaphangvaphieunhap.pack(fill="x")              btn\_thongkevabaocao = tk.Button(                  pnlGiaoDien,                  text="🧾 Thống kê & Báo cáo",                  command=lambda: self.HienThiFrame("ThongKeVaBaoCao"),                  \*\*self.DinhDangNut(),              )              btn\_thongkevabaocao.pack(fill="x")          btn\_hethong = tk.Button(              pnlGiaoDien,              text="⚙️ Hệ thống",              command=lambda: self.HienThiFrame("HeThong"),              \*\*self.DinhDangNut(),          )          btn\_hethong.pack(fill="x")          btn\_dangxuat = tk.Button(              pnlGiaoDien,              text="🚪Đăng xuất",              command=self.dang\_xuat,              \*\*self.DinhDangNut(),          )          btn\_dangxuat.pack(fill="x")      def DinhDangNut(self):          return {              "bg": ACCENT\_COLOR,              "fg": "white",              "font": ("Arial", 12),              "bd": 0,              "relief": "flat",              "anchor": "w",              "padx": 20,              "pady": 15,          }      def load\_form(self, page\_name):          frame = self.frames.get(page\_name)          if frame and hasattr(frame, "load\_data"):              try:                  frame.load\_data()              except Exception as e:                  print(f"Lỗi khi load dữ liệu cho {page\_name}: {e}")        def HienThiFrame(self, page\_name):          # Tạo frame nếu nó chưa tồn tại          if page\_name not in self.frames:              FrameClass = self.frame\_classes[page\_name]              frame = FrameClass(                  parent=self.container, controller=self, conn=self.conn, user=self.user              )              self.frames[page\_name] = frame              frame.grid(row=0, column=0, sticky="nsew")          #  Lấy frame đã được tạo/lưu trữ          frame\_to\_show = self.frames[page\_name]          try:              frame\_to\_show.load\_data()          except AttributeError:              # Bỏ qua nếu frame không có hàm refresh\_data              pass          except Exception as e:              # Báo lỗi nếu hàm load\_data chạy bị lỗi              print(f"Lỗi khi làm mới {page\_name}: {e}")          #  Hiển thị frame lên trên cùng          frame\_to\_show.tkraise()      # ====== Đóng kết nối khi thoát ======      def on\_close(self):          try:              if hasattr(self, "conn") and self.conn:                  self.conn.close()                  print("Kết nối SQL đã được đóng.")          except Exception as e:              print("Lỗi khi đóng kết nối:", e)          finally:              self.destroy()      def dang\_xuat(self):          if messagebox.askyesno(              "Xác nhận đăng xuất", "Bạn có chắc chắn muốn đăng xuất?"          ):              # Đóng kết nối SQL              try:                  if hasattr(self, "conn") and self.conn:                      self.conn.close()                      print("Kết nối SQL đã được đóng (do đăng xuất).")              except Exception as e:                  print("Lỗi khi đóng kết nối:", e)              self.destroy()              login\_window = frmLogin.Login()              login\_window.mainloop() |

**FORM TỔNG QUAN**

|  |
| --- |
| import tkinter as tk  from tkinter import ttk  from tkinter import messagebox  from matplotlib.backends.backend\_tkagg import FigureCanvasTkAgg  import matplotlib.pyplot as plt  from matplotlib.ticker import FuncFormatter  from datetime import datetime  class TongQuan(tk.Frame):      def \_\_init\_\_(self, parent, controller, conn, user):          super().\_\_init\_\_(parent)          self.conn = conn          self.controller = controller  # (MỚI) Lưu controller          self.user = user          # (MỚI) Lưu user          self.configure(bg="#f5f9ff")          lbl\_title = tk.Label(self, text="TRANG TỔNG QUAN HỆ THỐNG", font=("Segoe UI", 16, "bold"), fg="#003366", bg="#f5f9ff")          lbl\_title.pack(pady=15)          frame\_cards = tk.Frame(self, bg="#f5f9ff")          frame\_cards.pack(pady=10)          self.cards = {              "nv": self.create\_card(frame\_cards, "👤 Tổng nhân viên", "0"),              "kh": self.create\_card(frame\_cards, "👤 Tổng khách hàng", "0"),              "hsx": self.create\_card(frame\_cards, "🏭 Hãng sản xuất", "0"),              "ncc": self.create\_card(frame\_cards, "🏢 Nhà cung cấp", "0"),              "sp": self.create\_card(frame\_cards, "📺 Sản phẩm", "0"),              "pn": self.create\_card(frame\_cards, "📦 Phiếu nhập hàng", "0"),          }          self.hienthi\_dulieu()          # (MỚI) Khung chính cho biểu đồ VÀ điều khiển          self.frame\_chart = tk.Frame(self, bg="#f5f9ff")          self.frame\_chart.pack(pady=20, fill="x", expand=True)          # (MỚI) Khung con 1: Chỉ để chứa biểu đồ (canvas)          self.frame\_ve = tk.Frame(self.frame\_chart, bg="#f5f9ff")          self.frame\_ve.pack()          # (MỚI) Khung con 2: Chứa các nút điều khiển          self.frame\_dieu\_khien = tk.Frame(self.frame\_chart, bg="#f5f9ff")          self.frame\_dieu\_khien.pack(pady=5)          self.btn\_truoc = tk.Button(self.frame\_dieu\_khien, text="◀", font=("Segoe UI", 10, "bold"), command=self.nam\_truoc, width=4)          self.btn\_truoc.pack(side="left", padx=10)          self.lbl\_nam = tk.Label(self.frame\_dieu\_khien, text="Năm: ...", font=("Segoe UI", 12, "bold"), bg="#f5f9ff", fg="#003366", width=20)          self.lbl\_nam.pack(side="left", padx=20)          self.btn\_sau = tk.Button(self.frame\_dieu\_khien, text="▶", font=("Segoe UI", 10, "bold"), command=self.nam\_sau, width=4)          self.btn\_sau.pack(side="left", padx=10)            # (MỚI) Khởi tạo biến năm          self.danh\_sach\_nam = []          self.vi\_tri\_nam\_hien\_tai = 0 # Index trong danh sách          self.nam\_hien\_tai = datetime.now().year # Mặc định            self.lay\_danh\_sach\_nam() # Lấy danh sách năm          self.ve\_bieu\_do()       # Vẽ biểu đồ cho năm đó      # (MỚI) Hàm lấy danh sách các năm có doanh thu      def lay\_danh\_sach\_nam(self):          try:              cursor = self.conn.cursor()              # Lấy các năm có hóa đơn đã thanh toán              cursor.execute("""                  SELECT DISTINCT YEAR(NgayBan)                  FROM HoaDonBan                  WHERE TrangThai = N'Đã thanh toán'                  ORDER BY YEAR(NgayBan) ASC              """)              rows = cursor.fetchall()              self.danh\_sach\_nam = [row[0] for row in rows if row[0] is not None]              if not self.danh\_sach\_nam:                  # Nếu không có dữ liệu, lấy năm hiện tại                  self.nam\_hien\_tai = datetime.now().year                  self.danh\_sach\_nam = [self.nam\_hien\_tai]                  self.vi\_tri\_nam\_hien\_tai = 0              else:                  # Mặc định hiển thị năm mới nhất (cuối danh sách)                  self.vi\_tri\_nam\_hien\_tai = len(self.danh\_sach\_nam) - 1                  self.nam\_hien\_tai = self.danh\_sach\_nam[self.vi\_tri\_nam\_hien\_tai]          except Exception as e:              messagebox.showerror("Lỗi", f"Không thể tải danh sách năm:\n{e}")              # Xử lý dự phòng nếu lỗi              self.nam\_hien\_tai = datetime.now().year              self.danh\_sach\_nam = [self.nam\_hien\_tai]              self.vi\_tri\_nam\_hien\_tai = 0      # (MỚI) Hàm xử lý nút "Năm Trước"      def nam\_truoc(self):          if self.vi\_tri\_nam\_hien\_tai > 0:              self.vi\_tri\_nam\_hien\_tai -= 1              self.nam\_hien\_tai = self.danh\_sach\_nam[self.vi\_tri\_nam\_hien\_tai]              self.ve\_bieu\_do() # Vẽ lại biểu đồ      # (MỚI) Hàm xử lý nút "Năm Sau"      def nam\_sau(self):          if self.vi\_tri\_nam\_hien\_tai < (len(self.danh\_sach\_nam) - 1):              self.vi\_tri\_nam\_hien\_tai += 1              self.nam\_hien\_tai = self.danh\_sach\_nam[self.vi\_tri\_nam\_hien\_tai]              self.ve\_bieu\_do() # Vẽ lại biểu đồ      # (SỬA) Hàm vẽ biểu đồ đã được nâng cấp      def ve\_bieu\_do(self):          # (MỚI) Cập nhật Label và trạng thái các nút          self.lbl\_nam.config(text=f"Doanh thu năm: {self.nam\_hien\_tai}")            # Vô hiệu hóa nút "Trước" nếu đang ở năm đầu tiên          self.btn\_truoc.config(state="disabled" if self.vi\_tri\_nam\_hien\_tai == 0 else "normal")            # Vô hiệu hóa nút "Sau" nếu đang ở năm cuối cùng          self.btn\_sau.config(state="disabled" if self.vi\_tri\_nam\_hien\_tai == (len(self.danh\_sach\_nam) - 1) else "normal")          try:              cursor = self.conn.cursor()              # (SỬA) Thêm ĐIỀU KIỆN LỌC THEO NĂM vào câu query              query = """                  SELECT                      MONTH(NgayBan) AS Thang,                      SUM(TongTien) AS DoanhThu                  FROM HoaDonBan                  WHERE TrangThai = N'Đã thanh toán' AND YEAR(NgayBan) = ?                  GROUP BY MONTH(NgayBan)                  ORDER BY Thang;"""              cursor.execute(query, (self.nam\_hien\_tai,)) # (SỬA) Truyền năm vào query              data = cursor.fetchall()              cursor.close()              # (SỬA) Xử lý dữ liệu để luôn có đủ 12 tháng              # Tạo một từ điển {tháng: doanh\_thu}              doanh\_thu\_dict = {row.Thang: float(row.DoanhThu) for row in data}                # Tạo danh sách 12 tháng              thang = list(range(1, 13))                # Lấy doanh thu cho 12 tháng, nếu tháng nào không có thì gán là 0              doanh\_thu = [doanh\_thu\_dict.get(t, 0) for t in thang]              # (Không đổi) Hàm format tiền              def format\_tien(x, pos):                  return f"{int(x):,}".replace(",", ".")              fig, ax = plt.subplots(figsize=(7, 4))              ax.bar(thang, doanh\_thu, color="#1565C0") # (SỬA) Thêm màu              ax.yaxis.set\_major\_formatter(FuncFormatter(format\_tien))              # (SỬA) Cập nhật tiêu đề biểu đồ              ax.set\_title(f"Biểu đồ doanh thu năm {self.nam\_hien\_tai}", fontsize=12, fontweight="bold")              ax.set\_xlabel("Tháng", fontsize=11)              ax.set\_ylabel("Doanh thu (VNĐ)", fontsize=11)              ax.set\_xticks(range(1, 13)) # Đảm bảo hiện đủ 12 mốc tháng              fig.tight\_layout()              # (SỬA) Chỉ xóa các widget trong frame\_ve (để giữ lại các nút)              for widget in self.frame\_ve.winfo\_children():                  widget.destroy()              # (SỬA) Vẽ canvas vào frame\_ve              canvas = FigureCanvasTkAgg(fig, master=self.frame\_ve)              canvas.draw()              canvas.get\_tk\_widget().pack()          except Exception as e:              messagebox.showerror("Lỗi", f"Không thể tải biểu đồ:\n{e}")      # (Không đổi)      def create\_card(self, parent, title, value):          card = tk.Frame(parent, bg="#d6eaff", width=180, height=100, relief="ridge", bd=2)          card.pack(side="left", padx=10, pady=5)          card.pack\_propagate(False)          lbl\_title = tk.Label(card, text=title, font=("Segoe UI", 10, "bold"), fg="#003366", bg="#d6eaff")          lbl\_title.pack(pady=(10, 0))          lbl\_value = tk.Label(card, text=value, font=("Segoe UI", 20, "bold"), fg="#002b80", bg="#d6eaff")          lbl\_value.pack(pady=(5, 0))          return lbl\_value      # (Không đổi)      def hienthi\_dulieu(self):          try:              cursor = self.conn.cursor()              queries = {                  "nv": "SELECT COUNT(\*) FROM NhanVien",                  "kh": "SELECT COUNT(\*) FROM KHACHHANG",                  "hsx": "SELECT COUNT(\*) FROM HangSanXuat",                  "ncc": "SELECT COUNT(\*) FROM NhaCungCap",                  "sp": "SELECT COUNT(\*) FROM Tivi",                  "pn": "SELECT COUNT(\*) FROM PhieuNhapHang WHERE TrangThai = N’Đã duyệt’"}              for key, query in queries.items():                  cursor.execute(query)                  count = cursor.fetchone()[0]                  self.cards[key].config(text=str(count))          except Exception as e:              messagebox.showerror("Lỗi", "Không thể tải dữ liệu tổng quan:\n" + str(e))      # Hàm làm mới tab khi click vào      def load\_data(self):          try:              self.hienthi\_dulieu()              self.lay\_danh\_sach\_nam()              self.ve\_bieu\_do()            except Exception as e:              messagebox.showerror("Lỗi khi làm mới dữ liệu: " + str(e)) |

**FORM QUẢN LÝ SẢN PHẨM**

|  |
| --- |
| import tkinter as tk  from tkinter import ttk, messagebox  import pyodbc  import tabTivi as tv  import tabHangSanXuat as hsx  import tabNhaCungCap as ncc  import tabBaoHanh as bh  # === Tạo class Quản Lý Sản Phẩm ===  class QuanLySanPham(tk.Frame):      def \_\_init\_\_(self, parent, controller, conn, user):          super().\_\_init\_\_(parent, bg="white")          # === CHUỖI KẾT NỐI  ====          self.conn = conn          self.controller = controller          self.user = user          # === TITLE ===          lbl\_title = tk.Label(self, text="QUẢN LÝ SẢN PHẨM", font=("Segoe UI", 16, "bold"), bg="white", fg="#0D47A1")          lbl\_title.pack()          # === Tạo Tab Control ===          tab\_control = ttk.Notebook(self)          tab\_control.pack(fill="both", expand=True, padx=20, pady=10)          # === Các tab con ===          self.tab\_tivi = tv.tabTivi(tab\_control, conn, self.user, self.controller)          self.tab\_hang = hsx.tabHangSanXuat(tab\_control, conn, self.load\_data)          self.tab\_nhacungcap = ncc.tabNhaCungCap(tab\_control, conn, self.controller)          self.tab\_bh = bh.tabBaoHanh(tab\_control, conn, self.user)            tab\_control.add(self.tab\_tivi, text="📺 Tivi")          # Thêm vào notebook          if(user == "admin"):              tab\_control.add(self.tab\_hang, text="🏭 Hãng sản xuất")              tab\_control.add(self.tab\_nhacungcap, text="🤝 Nhà cung cấp")            tab\_control.add(self.tab\_bh, text="🧾 Bảo hành")      # Hàm làm mới tab khi click vào      def load\_data(self):          try:              self.tab\_tivi.hienthi\_dulieu()              self.tab\_tivi.load\_hang\_san\_xuat()              self.tab\_hang.hienthi\_dulieu()              self.tab\_nhacungcap.hienthi\_dulieu()              self.tab\_bh.hienthi\_dulieu()              self.tab\_bh.hien\_thi\_du\_lieu\_cthd()            except Exception as e:              messagebox.showerror("Lỗi khi load tab con: " + str(e)) |

**FORM TIVI**

|  |
| --- |
| import tkinter as tk  from tkinter import ttk, messagebox, filedialog  import pyodbc  from PIL import Image, ImageTk  import io  # === TAB TIVI ===  class tabTivi(tk.Frame):      def \_\_init\_\_(self, parent, conn, user, controller):          super().\_\_init\_\_(parent, bg="white")          # === CHUỖI KẾT NỐI ===          self.conn = conn          self.cursor = conn.cursor()          self.user = user          self.controller = controller          self.selected\_item = None          self.image\_data = None          self.anh\_hien\_tai = None          self.ds\_them = []          self.ds\_sua = []          self.ds\_xoa = []          # === KHUNG TÌM KIẾM ===          frame\_search = tk.Frame(self, bg="#E3F2FD", padx=10, pady=10)          frame\_search.pack(fill="x", padx=20, pady=5)          tk.Label(frame\_search, text="Tìm kiếm:", font=("Segoe UI", 10), bg="#E3F2FD").pack(side="left", padx=5)          self.txt\_timkiem = tk.Entry(frame\_search, font=("Segoe UI", 10), width=60)          self.txt\_timkiem.pack(side="left", padx=5)          self.txt\_timkiem.bind("<Return>", lambda e: self.tim\_kiem())          self.search\_option = tk.StringVar(value="ma")          tk.Radiobutton(frame\_search, text="Theo mã Tivi", variable=self.search\_option, value="ma", bg="#E3F2FD", font=("Segoe UI", 10),).pack(side="left", padx=10)          tk.Radiobutton(frame\_search, text="Theo tên Tivi", variable=self.search\_option, value="ten", bg="#E3F2FD", font=("Segoe UI", 10),).pack(side="left")          tk.Button(frame\_search, text="Tìm", font=("Segoe UI", 10, "bold"), bg="#1565C0", fg="white", bd=0,  padx=10, pady=5, command=self.tim\_kiem,).pack(side="left", padx=10)          tk.Button(frame\_search,text="Hủy", font=("Segoe UI", 10, "bold"), bg="#E53935", fg="white", bd=0, padx=10, pady=5, command=self.huy,).pack(side="left", padx=10)          # ==== KHUNG THÔNG TIN ====          frame\_form = tk.LabelFrame(self, text="Thông tin Tivi", bg="white", font=("Segoe UI", 12, "bold"), fg="#0D47A1",padx=10, pady=10,)          frame\_form.pack(fill="x", padx=20, pady=10)          self.pic\_anhtivi = tk.Canvas(frame\_form, width=80, height=100, bg="#f0f0f0", highlightthickness=1, highlightbackground="#ccc", )          self.pic\_anhtivi.grid(row=0, column=0, rowspan=3, padx=5, pady=5)          self.pic\_anhtivi.create\_text(30, 40, text="Ảnh\ntivi", font=("Segoe UI", 10), fill="#888", tags="placeholder",)          self.btn\_chonanh = tk.Button(frame\_form, text="Chọn ảnh", bg="#42A5F5", fg="white", font=("Segoe UI", 7, "bold"), height=1, width=7, bd=0, padx=10, pady=10, command=self.chon\_anh, )          self.btn\_chonanh.grid(row=3, column=0, pady=5)          tk.Label(frame\_form, text="Mã Tivi:", bg="white", font=("Segoe UI", 11)).grid(row=0, column=1, sticky="w", padx=5, pady=5)          self.txt\_matv = ttk.Entry(frame\_form, width=24)          self.txt\_matv.grid(row=0, column=2, padx=5, pady=5)          tk.Label(frame\_form, text="Tên Tivi:", bg="white", font=("Segoe UI", 11)).grid(row=0, column=3, sticky="w", padx=5, pady=5)          self.txt\_tentivi = ttk.Entry(frame\_form, width=22)          self.txt\_tentivi.grid(row=0, column=4, padx=5, pady=5)          tk.Label(frame\_form, text="Hãng:", bg="white", font=("Segoe UI", 11)).grid(row=0, column=5, sticky="w", padx=5, pady=5)          self.cbo\_hang = ttk.Combobox(frame\_form, width=21, state="readonly")          self.cbo\_hang.grid(row=0, column=6, padx=5, pady=5)          tk.Label(frame\_form, text="Kích thước:", bg="white", font=("Segoe UI", 11)).grid(row=1, column=1, sticky="w", padx=5, pady=5)          self.txt\_kichthuoc = ttk.Entry(frame\_form, width=24)          self.txt\_kichthuoc.grid(row=1, column=2, padx=5, pady=5)          tk.Label(frame\_form, text="Độ phân giải:", bg="white", font=("Segoe UI", 11)).grid(row=1, column=3, sticky="w", padx=5, pady=5)          self.txt\_dophangiai = ttk.Entry(frame\_form, width=22)          self.txt\_dophangiai.grid(row=1, column=4, padx=5, pady=5)          tk.Label(frame\_form, text="Năm sản xuất:", bg="white", font=("Segoe UI", 11)).grid(row=1, column=5, sticky="w", padx=5, pady=5)          self.txt\_namsanxuat = ttk.Entry(frame\_form, width=24)          self.txt\_namsanxuat.grid(row=1, column=6, padx=5, pady=5)          tk.Label(frame\_form, text="Giá bán:", bg="white", font=("Segoe UI", 11)).grid(row=2, column=1, sticky="w", padx=5, pady=5)          self.txt\_giaban = ttk.Entry(frame\_form, width=24)          self.txt\_giaban.grid(row=2, column=2, padx=5, pady=5)          tk.Label(frame\_form, text="Số lượng:", bg="white", font=("Segoe UI", 11)).grid(row=2, column=3, sticky="w", padx=5, pady=5)          self.txt\_soluong = ttk.Entry(frame\_form, width=22, state="disabled")          self.txt\_soluong.grid(row=2, column=4, padx=5, pady=5)          tk.Label(frame\_form, text="Mô tả:", bg="white", font=("Segoe UI", 11)).grid( row=3, column=1, sticky="w", padx=5, pady=5)          self.txt\_mota = ttk.Entry(frame\_form, width=110)          self.txt\_mota.grid(row=3, column=2, columnspan=6, padx=5, pady=5)          # ==== NÚT CHỨC NĂNG ====          frame\_buttons = tk.Frame(self, bg="white")          frame\_buttons.pack(pady=10)          btn\_them = tk.Button(frame\_buttons, text="➕ Thêm", bg="#EBDA42", fg="white", font=("Segoe UI", 10, "bold"), padx=20, pady=5,  bd=0, command=self.them\_tivi,)          btn\_them.grid(row=0, column=0, padx=10)          btn\_sua = tk.Button(frame\_buttons, text="✏️ Sửa", bg="#FB8C00", fg="white", font=("Segoe UI", 10, "bold"), padx=20, pady=5, bd=0, command=self.sua\_tivi,)          btn\_sua.grid(row=0, column=1, padx=10)          btn\_xoa = tk.Button(frame\_buttons, text="🗑️ Xóa", bg="#E53935", fg="white", font=("Segoe UI", 10, "bold"), padx=20, pady=5, bd=0, command=self.xoa\_tivi,)          btn\_xoa.grid(row=0, column=2, padx=10)          btn\_lammoi = tk.Button(frame\_buttons, text="🔄 Làm mới", bg="#1E88E5", fg="white", font=("Segoe UI", 10, "bold"), padx=20, pady=5, bd=0, command=self.lam\_moi,)          btn\_lammoi.grid(row=0, column=3, padx=10)          btn\_luu = tk.Button(frame\_buttons, text="💾 Lưu", bg="#449A2D", fg="white", font=("Segoe UI", 11, "bold"), padx=20, pady=5, bd=0, command=self.luu,)          btn\_luu.grid(row=0, column=4, padx=10)          frame\_table = tk.Frame(self, bg="white")          frame\_table.pack(fill="both", expand=True, padx=20, pady=10)          scroll\_y = ttk.Scrollbar(frame\_table, orient="vertical")          scroll\_y.pack(side="right", fill="y")          scroll\_x = ttk.Scrollbar(frame\_table, orient="horizontal")          scroll\_x.pack(side="bottom", fill="x")          columns = ("MaTivi", "TenTivi", "TenHang", "KichThuoc", "DoPhanGiai", "GiaBan", "SoLuongTon", "NamSanXuat", "MoTa",)          self.trHienThi = ttk.Treeview(frame\_table, columns=columns, show="headings", height=12, yscrollcommand=scroll\_y.set, xscrollcommand=scroll\_x.set,)          scroll\_y.config(command=self.trHienThi.yview)          scroll\_x.config(command=self.trHienThi.xview)          self.trHienThi.heading("MaTivi", text="Mã Tivi")          self.trHienThi.heading("TenTivi", text="Tên Tivi")          self.trHienThi.heading("TenHang", text="Hãng")          self.trHienThi.heading("KichThuoc", text="Kích thước")          self.trHienThi.heading("DoPhanGiai", text="Độ phân giải")          self.trHienThi.heading("GiaBan", text="Giá bán")          self.trHienThi.heading("SoLuongTon", text="Số lượng")          self.trHienThi.heading("NamSanXuat", text="Năm SX")          self.trHienThi.heading("MoTa", text="Mô tả")          self.trHienThi.column("MaTivi", width=80, anchor="center")          self.trHienThi.column("TenTivi", width=200, anchor="w")          self.trHienThi.column("TenHang", width=100, anchor="w")          self.trHienThi.column("KichThuoc", width=100, anchor="center")          self.trHienThi.column("DoPhanGiai", width=100, anchor="center")          self.trHienThi.column("GiaBan", width=100, anchor="e")          self.trHienThi.column("SoLuongTon", width=80, anchor="center")          self.trHienThi.column("NamSanXuat", width=80, anchor="center")          self.trHienThi.column("MoTa", width=200, anchor="w")          self.trHienThi.pack(fill="both", expand=True)          style = ttk.Style()          style.configure("Treeview.Heading", font=("Segoe UI", 11, "bold"))          style.configure("Treeview", font=("Segoe UI", 10), rowheight=28)          if(self.user != "admin"):              self.btn\_chonanh.grid\_forget()              frame\_buttons.pack\_forget()          self.trHienThi.bind("<<TreeviewSelect>>", self.chon\_dong)          self.load\_hang\_san\_xuat()          self.hienthi\_dulieu()      def load\_hang\_san\_xuat(self):          try:              self.cursor.execute("SELECT MaHang, TenHang FROM HangSanXuat ORDER BY TenHang")              self.hang\_dict = {}              hang\_list = []              for row in self.cursor.fetchall():                  ma\_hang = row.MaHang                  ten\_hang = row.TenHang                  self.hang\_dict[ten\_hang] = ma\_hang                  hang\_list.append(ten\_hang)              self.cbo\_hang["values"] = hang\_list          except Exception as e:              messagebox.showerror("Lỗi", "Không thể load danh sách hãng: " + str(e))      def hienthi\_dulieu(self):          try:              for item in self.trHienThi.get\_children():                  self.trHienThi.delete(item)              query = """                  SELECT t.MaTivi, t.HinhAnh, t.TenTivi, h.TenHang, t.KichThuoc,                         t.DoPhanGiai, t.GiaBan, t.SoLuongTon, t.NamSanXuat, t.MoTa, h.MaHang                  FROM Tivi t                  INNER JOIN HangSanXuat h ON t.MaHang = h.MaHang                  ORDER BY t.MaTivi """              self.cursor.execute(query)              for row in self.cursor.fetchall():                  ma\_tivi = row.MaTivi                  self.trHienThi.insert("", "end", values=(ma\_tivi, row.TenTivi, row.TenHang, row.KichThuoc or "", row.DoPhanGiai or "", f"{row.GiaBan:,.0f}" if row.GiaBan else "0", row.SoLuongTon or 0, row.NamSanXuat or "", row.MoTa or "",),)          except Exception as e:              messagebox.showerror("Lỗi", "Không thể load danh sách tivi: " + str(e))      def chon\_dong(self, event):          selected = self.trHienThi.selection()          if not selected:              return          self.selected\_item = selected[0]          values = self.trHienThi.item(self.selected\_item)["values"]          self.xoa\_form()          self.txt\_matv.insert(0, values[0])          self.txt\_tentivi.insert(0, values[1])          self.cbo\_hang.set(values[2])          self.txt\_kichthuoc.insert(0, values[3])          self.txt\_dophangiai.insert(0, values[4])          gia\_ban = str(values[5]).replace(",", "")          self.txt\_giaban.insert(0, gia\_ban)            self.txt\_soluong.config(state="normal")          self.txt\_soluong.delete(0, tk.END)          self.txt\_soluong.insert(0, values[6])          self.txt\_soluong.config(state="disabled")          self.txt\_namsanxuat.insert(0, values[7])          self.txt\_mota.insert(0, values[8])          self.hienthi\_anh(values[0])      def hienthi\_anh(self, ma\_tivi):          try:              cursor = self.conn.cursor()              cursor.execute("SELECT HinhAnh FROM Tivi WHERE MaTivi = ?", (ma\_tivi,))              row = cursor.fetchone()              if row and row[0]:                  image\_data = row[0]                  anh\_pil = Image.open(io.BytesIO(image\_data))                  anh\_pil = anh\_pil.resize((120, 120))                  self.anh\_hien\_tai = ImageTk.PhotoImage(anh\_pil, master=self)                  self.pic\_anhtivi.delete("all")                  self.pic\_anhtivi.create\_image(60, 60, image=self.anh\_hien\_tai, anchor="center")                  self.pic\_anhtivi.image = self.anh\_hien\_tai              else:                  self.pic\_anhtivi.delete("all")                  self.pic\_anhtivi.create\_text(60, 60, text="(Không có ảnh)", fill="gray")          except Exception as e:              messagebox.showerror("Lỗi hiển thị ảnh: ", str(e))      def chon\_anh(self):          if not self.selected\_item:              messagebox.showerror("Lỗi", "Vui long chọn tivi trên danh sách.")              return          file\_path = filedialog.askopenfilename(title="Chọn ảnh", filetypes=[("Image files", "\*.jpg \*.jpeg \*.png \*.bmp \*.gif")],)          if file\_path:              try:                  with open(file\_path, "rb") as file:                      self.image\_data = file.read()                  anh\_pil = Image.open(io.BytesIO(self.image\_data))                  anh\_pil = anh\_pil.resize((120, 120))                  self.anh\_hien\_tai = ImageTk.PhotoImage(anh\_pil, master=self)                  self.pic\_anhtivi.delete("all")                  self.pic\_anhtivi.create\_image(60, 60, image=self.anh\_hien\_tai, anchor="center")                  self.pic\_anhtivi.image = self.anh\_hien\_tai              except Exception as e:                  messagebox.showerror("Lỗi chọn ảnh", str(e))      def them\_tivi(self):          if not self.xac\_nhan\_du\_lieu():              return          ma = self.txt\_matv.get().strip()          ten = self.txt\_tentivi.get().strip()          ten\_hang = self.cbo\_hang.get()          ma\_hang = self.hang\_dict.get(ten\_hang)          kichthuoc = self.txt\_kichthuoc.get().strip()          dophangiai = self.txt\_dophangiai.get().strip()          giaban = float(self.txt\_giaban.get()) if self.txt\_giaban.get().strip() else 0          soluong = 0          namsanxuat = (int(self.txt\_namsanxuat.get())              if self.txt\_namsanxuat.get().strip()              else None)          mota = self.txt\_mota.get().strip()          for item in self.trHienThi.get\_children():              if self.trHienThi.item(item)["values"][0] == ma:                  messagebox.showwarning("Cảnh báo", f"Mã tivi '{ma}' đã tồn tại!")                  return          self.trHienThi.insert("", "end",  values=(ma, ten, ten\_hang, kichthuoc, dophangiai, f"{giaban:,.0f}", soluong, namsanxuat if namsanxuat else "", mota,),)          self.ds\_them.append(( ma, self.image\_data, ten, ma\_hang, kichthuoc or None, dophangiai or None, giaban, soluong, namsanxuat, mota or None,))          self.xoa\_form()          messagebox.showinfo("Thành công", "Đã thêm dòng mới! Nhấn 'Lưu' để lưu vào CSDL.")      def sua\_tivi(self):          if not self.selected\_item:              messagebox.showwarning("Cảnh báo", "Vui lòng chọn dòng cần sửa!")              return          if not self.xac\_nhan\_du\_lieu():              return          ma = self.txt\_matv.get().strip()          ten = self.txt\_tentivi.get().strip()          ten\_hang = self.cbo\_hang.get()          ma\_hang = self.hang\_dict.get(ten\_hang)          kichthuoc = self.txt\_kichthuoc.get().strip()          dophangiai = self.txt\_dophangiai.get().strip()          giaban = float(self.txt\_giaban.get()) if self.txt\_giaban.get().strip() else 0          soluong = 0          namsanxuat = (int(self.txt\_namsanxuat.get())              if self.txt\_namsanxuat.get().strip()              else None)          mota = self.txt\_mota.get().strip()          item = self.trHienThi.item(self.selected\_item)          ma\_cu = item["values"][0]          self.trHienThi.item(              self.selected\_item,              values=(ma, ten, ten\_hang, kichthuoc, dophangiai, f"{giaban:,.0f}", soluong, namsanxuat if namsanxuat else "", mota,),)          is\_new = any(item[0] == ma\_cu for item in self.ds\_them)          if is\_new:              self.ds\_them = [item for item in self.ds\_them if item[0] != ma\_cu]              self.ds\_them.append((ma, self.image\_data,  ten, ma\_hang, kichthuoc or None, dophangiai or None, giaban, soluong, namsanxuat, mota or None,))          else:              self.ds\_sua = [item for item in self.ds\_sua if item[10] != ma\_cu]              self.ds\_sua.append((ma, self.image\_data, ten, ma\_hang, kichthuoc or None, dophangiai or None, giaban, soluong, namsanxuat, mota or None, ma\_cu,))          messagebox.showinfo("Thành công", "Đã cập nhật dòng! Nhấn 'Lưu' để lưu vào CSDL.")      def xoa\_tivi(self):          if not self.selected\_item:              messagebox.showwarning("Cảnh báo", "Vui lòng chọn dòng cần xóa!")              return          confirm = messagebox.askyesno("Xác nhận", "Bạn có chắc muốn xóa dòng này?")          if not confirm:              return          item = self.trHienThi.item(self.selected\_item)          ma = item["values"][0]          self.trHienThi.delete(self.selected\_item)          is\_new = any(item[0] == ma for item in self.ds\_them)          if is\_new:              self.ds\_them = [item for item in self.ds\_them if item[0] != ma]          else:              if ma not in self.ds\_xoa:                  self.ds\_xoa.append(ma)          self.xoa\_form()          self.selected\_item = None          messagebox.showinfo("Thành công", "Đã xóa dòng! Nhấn 'Lưu' để lưu vào CSDL.")      def luu(self):          try:              if not self.ds\_them and not self.ds\_sua and not self.ds\_xoa:                  messagebox.showinfo("Thông báo", "Không có thay đổi để lưu!")                  return              confirm = messagebox.askyesno("Xác nhận", "Bạn có chắc muốn lưu các thay đổi?")              if not confirm:                  return              for ma in self.ds\_xoa:                  self.cursor.execute("DELETE FROM Tivi WHERE MaTivi = ?", (ma,))              for item in self.ds\_them:                  query = """                      INSERT INTO Tivi (MaTivi, HinhAnh, TenTivi, MaHang, KichThuoc,                                        DoPhanGiai, GiaBan, SoLuongTon, NamSanXuat, MoTa)                      VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?) """                  self.cursor.execute(query, item)              for item in self.ds\_sua:                  query = """                      UPDATE Tivi                      SET MaTivi = ?, HinhAnh = ?, TenTivi = ?, MaHang = ?, KichThuoc = ?,                          DoPhanGiai = ?, GiaBan = ?, SoLuongTon = ?, NamSanXuat = ?, MoTa = ?                      WHERE MaTivi = ? """                  self.cursor.execute(query, item)              self.conn.commit()              messagebox.showinfo("Thành công", "Đã lưu thay đổi vào CSDL!")          except pyodbc.IntegrityError as e:              self.conn.rollback()              messagebox.showerror("Lỗi", "Lỗi ràng buộc dữ liệu:" + str(e))          except Exception as e:              self.conn.rollback()              messagebox.showerror("Lỗi", "Không thể lưu dữ liệu:" + str(e))          self.hienthi\_dulieu()          self.xoa\_form()          self.ds\_them.clear()          self.ds\_sua.clear()          self.ds\_xoa.clear()          self.controller.load\_form("NhapHangVaPhieuNhap")          self.controller.load\_form("BanHangVaHoaDon")      def lam\_moi(self):          confirm = messagebox.askyesno("Xác nhận", "Bạn có chắc muốn hủy các thay đổi?")          if not confirm:              return          self.ds\_them.clear()          self.ds\_sua.clear()          self.ds\_xoa.clear()          self.hienthi\_dulieu()          self.load\_hang\_san\_xuat()          self.xoa\_form()          self.txt\_timkiem.delete(0, tk.END)          messagebox.showinfo("Thông báo", "Đã làm mới dữ liệu!")      def tim\_kiem(self):          tu\_tim\_kiem = self.txt\_timkiem.get().strip()          if not tu\_tim\_kiem:              messagebox.showinfo("Thông báo", "Vui lòng nhập từ khóa tìm kiếm.")              self.hienthi\_dulieu()              return          try:              for item in self.trHienThi.get\_children():                  self.trHienThi.delete(item)              if self.search\_option.get() == "ma":                  query = """                      SELECT t.MaTivi, t.HinhAnh, t.TenTivi, h.TenHang, t.KichThuoc,                             t.DoPhanGiai, t.GiaBan, t.SoLuongTon, t.NamSanXuat, t.MoTa                      FROM Tivi t                      INNER JOIN HangSanXuat h ON t.MaHang = h.MaHang                      WHERE t.MaTivi LIKE ?                      ORDER BY t.MaTivi """              else:                  query = """                      SELECT t.MaTivi, t.HinhAnh, t.TenTivi, h.TenHang, t.KichThuoc,                             t.DoPhanGiai, t.GiaBan, t.SoLuongTon, t.NamSanXuat, t.MoTa                      FROM Tivi t                      INNER JOIN HangSanXuat h ON t.MaHang = h.MaHang                      WHERE t.TenTivi LIKE ?                      ORDER BY t.MaTivi """              self.cursor.execute(query, (f"%{tu\_tim\_kiem}%",))              rows = self.cursor.fetchall()              for row in rows:                  ma\_tivi = row.MaTivi                  self.trHienThi.insert("", "end", values=(ma\_tivi, row.TenTivi, row.TenHang, row.KichThuoc or "", row.DoPhanGiai or "", f"{row.GiaBan:,.0f}" if row.GiaBan else "0", row.SoLuongTon or 0, row.NamSanXuat or "", row.MoTa or "",),)              if not rows:                  messagebox.showinfo("Thông báo", "Không tìm thấy kết quả!")          except Exception as e:              messagebox.showerror("Lỗi", f"Không thể tìm kiếm: {str(e)}")      def xac\_nhan\_du\_lieu(self):          if not self.txt\_matv.get().strip():              messagebox.showwarning("Cảnh báo", "Vui lòng nhập mã tivi!")              self.txt\_matv.focus()              return False          if not self.txt\_tentivi.get().strip():              messagebox.showwarning("Cảnh báo", "Vui lòng nhập tên tivi!")              self.txt\_tentivi.focus()              return False          if not self.cbo\_hang.get():              messagebox.showwarning("Cảnh báo", "Vui lòng chọn hãng!")              self.cbo\_hang.focus()              return False          if self.txt\_giaban.get().strip():              try:                  gia = float(self.txt\_giaban.get())                  if gia < 0:                      messagebox.showwarning("Cảnh báo", "Giá bán phải >= 0!")                      self.txt\_giaban.focus()                      return False              except ValueError:                  messagebox.showwarning("Cảnh báo", "Giá bán phải là số!")                  self.txt\_giaban.focus()                  return False          if self.txt\_soluong.get().strip():              try:                  sl = int(self.txt\_soluong.get())                  if sl < 0:                      messagebox.showwarning("Cảnh báo", "Số lượng phải >= 0!")                      self.txt\_soluong.focus()                      return False              except ValueError:                  messagebox.showwarning("Cảnh báo", "Số lượng phải là số nguyên!")                  self.txt\_soluong.focus()                  return False          if self.txt\_namsanxuat.get().strip():              try:                  nam = int(self.txt\_namsanxuat.get())                  if nam < 1900 or nam > 2025:                      messagebox.showwarning("Cảnh báo", "Năm sản xuất không hợp lệ!")                      self.txt\_namsanxuat.focus()                      return False              except ValueError:                  messagebox.showwarning("Cảnh báo", "Năm sản xuất phải là số nguyên!")                  self.txt\_namsanxuat.focus()                  return False          return True      def xoa\_form(self):          self.txt\_matv.delete(0, tk.END)          self.txt\_tentivi.delete(0, tk.END)          self.cbo\_hang.set("")          self.txt\_kichthuoc.delete(0, tk.END)          self.txt\_dophangiai.delete(0, tk.END)          self.txt\_giaban.delete(0, tk.END)          self.txt\_soluong.delete(0, tk.END)          self.txt\_namsanxuat.delete(0, tk.END)          self.txt\_mota.delete(0, tk.END)          self.image\_data = None          self.anh\_hien\_tai = None          self.pic\_anhtivi.delete("all")          self.pic\_anhtivi.create\_text(30, 40, text="Ảnh\ntivi", font=("Segoe UI", 10), fill="#888")      def huy(self):          self.txt\_timkiem.delete(0, tk.END)          self.hienthi\_dulieu() |

**FORM HÃNG SẢN XUẤT**

|  |
| --- |
| import tkinter as tk  from tkinter import ttk, messagebox  import pyodbc  # === TAB HÃNG SẢN XUẤT ===  class tabHangSanXuat(tk.Frame):      def \_\_init\_\_(self, parent, conn, load\_data):          super().\_\_init\_\_(parent, bg="white")          self.conn = conn          self.load\_data = load\_data          self.cursor = conn.cursor()          self.ds\_them = []          self.ds\_sua = []          self.ds\_xoa = []          # === KHUNG TÌM KIẾM ===          frame\_search = tk.Frame(self, bg="#E3F2FD", padx=10, pady=10)          frame\_search.pack(fill="x", padx=20, pady=5)          tk.Label(frame\_search, text="Tìm kiếm:", font=("Segoe UI", 10), bg="#E3F2FD").pack(side="left", padx=5)          self.txt\_timkiem = tk.Entry(frame\_search, font=("Segoe UI", 10), width=55, bg="white")          self.txt\_timkiem.pack(side="left", padx=5)          self.txt\_timkiem.bind("<Return>", lambda event: self.timkiem())          self.search\_option = tk.StringVar(value="ma")          tk.Radiobutton(frame\_search, text="Theo mã hãng", variable=self.search\_option, value="ma", bg="#E3F2FD", font=("Segoe UI", 10)).pack(side="left", padx=10)          tk.Radiobutton(frame\_search, text="Theo tên hãng", variable=self.search\_option, value="ten", bg="#E3F2FD", font=("Segoe UI", 10)).pack(side="left")          tk.Button(frame\_search, text="Tìm", font=("Segoe UI", 10, "bold"), bg="#1565C0", fg="white", bd=0, padx=10, pady=5, command=self.timkiem).pack(side="left", padx=10)          tk.Button(frame\_search, text="Hủy", font=("Segoe UI", 10, "bold"), bg="#E53935", fg="white", bd=0, padx=10, pady=5, command=self.huy).pack(side="left", padx=10)          # === KHUNG THÔNG TIN ===          frame\_form = tk.LabelFrame(self, text="Thông tin Hãng sản xuất", bg="white", font=("Segoe UI", 12, "bold"), fg="#0D47A1", padx=10, pady=10)          frame\_form.pack(fill="x", padx=20, pady=10)          tk.Label(frame\_form, text="Mã hãng:", bg="white", font=("Segoe UI", 10)).grid(row=0, column=0, sticky="w", padx=5, pady=5)          self.txt\_ma = ttk.Entry(frame\_form, width=32)          self.txt\_ma.grid(row=0, column=1, padx=5, pady=5)          tk.Label(frame\_form, text="Tên hãng:", bg="white", font=("Segoe UI", 10)).grid(row=0, column=2, sticky="w", padx=5, pady=5)          self.txt\_ten = ttk.Entry(frame\_form, width=32)          self.txt\_ten.grid(row=0, column=3, padx=5, pady=5)          tk.Label(frame\_form, text="Quốc gia:", bg="white", font=("Segoe UI", 10)).grid(row=0, column=5, pady=5, padx=5, sticky="w")          self.txt\_quocgia = ttk.Entry(frame\_form, width=32)          self.txt\_quocgia.grid(row=0, column=6, padx=5, pady=5)          # ==== NÚT CHỨC NĂNG ====          frame\_buttons = tk.Frame(self, bg="white")          frame\_buttons.pack(pady=10)          btn\_them = tk.Button(frame\_buttons, text="➕ Thêm", bg="#EBDA42", fg="white", font=("Segoe UI", 11, "bold"), padx=20, pady=5, bd=0, command=self.them)          btn\_them.grid(row=0, column=0, padx=10)          btn\_sua = tk.Button(frame\_buttons, text="✏️ Sửa", bg="#FB8C00", fg="white", font=("Segoe UI", 11, "bold"), padx=20, pady=5, bd=0, command=self.sua)          btn\_sua.grid(row=0, column=1, padx=10)          btn\_xoa = tk.Button(frame\_buttons, text="🗑️ Xóa", bg="#E53935", fg="white", font=("Segoe UI", 11, "bold"), padx=20, pady=5, bd=0, command=self.xoa)          btn\_xoa.grid(row=0, column=2, padx=10)          btn\_lammoi = tk.Button(frame\_buttons, text="🔄 Làm mới", bg="#1E88E5", fg="white", font=("Segoe UI", 11, "bold"), padx=20, pady=5, bd=0, command=self.lammoi)          btn\_lammoi.grid(row=0, column=3, padx=10)          btn\_luu = tk.Button(frame\_buttons, text="💾 Lưu", bg="#43A047", fg="white", font=("Segoe UI", 11, "bold"), padx=20, pady=5, bd=0, command=self.luu)          btn\_luu.grid(row=0, column=4, padx=10)          # === BẢNG HÃNG SẢN XUẤT ===          frame\_table = tk.Frame(self, bg="white")          frame\_table.pack(fill="both", expand=True, padx=20, pady=10)          columns = ("MaHang", "TenHang", "QuocGia")          scroll\_y = ttk.Scrollbar(frame\_table, orient="vertical")          scroll\_x = ttk.Scrollbar(frame\_table, orient="horizontal")          self.trHienThi = ttk.Treeview(frame\_table, show="headings", columns=columns, height=12, yscrollcommand=scroll\_y.set, xscrollcommand=scroll\_x.set)          scroll\_y.config(command=self.trHienThi.yview)          scroll\_x.config(command=self.trHienThi.xview)          scroll\_y.pack(side="right", fill="y")          scroll\_x.pack(side="bottom", fill="x")          self.trHienThi.pack(fill="both", expand=True)          self.trHienThi.heading("MaHang", text="Mã Hãng")          self.trHienThi.heading("TenHang", text="Tên Hãng")          self.trHienThi.heading("QuocGia", text="Quốc Gia")          self.trHienThi.column("MaHang", width=150, anchor="center")          self.trHienThi.column("TenHang", width=300, anchor="center")          self.trHienThi.column("QuocGia", width=200, anchor="center")          style = ttk.Style()          style.configure("Treeview.Heading", font=("Segoe UI", 11, "bold"))          style.configure("Treeview", font=("Segoe UI", 10), rowheight=28)          self.trHienThi.bind("<<TreeviewSelect>>", self.chon\_dong)          self.hienthi\_dulieu()      def hienthi\_dulieu(self):          try:              for item in self.trHienThi.get\_children():                  self.trHienThi.delete(item)              self.cursor.execute("SELECT MaHang, TenHang, QuocGia FROM HangSanXuat ORDER BY MaHang")              rows = self.cursor.fetchall()              for row in rows:                  self.trHienThi.insert("", "end", values=(row.MaHang, row.TenHang, row.QuocGia))          except Exception as e:              messagebox.showerror("Lỗi", "Không thể tải dữ liệu: "+ str(e))      def chon\_dong(self, event):          selected = self.trHienThi.selection()          if selected:              item = self.trHienThi.item(selected[0])              values = item["values"]              self.xoa\_form()              self.txt\_ma.insert(0, values[0])              self.txt\_ten.insert(0, values[1])              self.txt\_quocgia.insert(0, values[2] if values[2] else "")      def kiemtra\_trung\_ten\_hang(self, ten\_hang, ma\_hang\_hien\_tai=None):          ten\_hang = str(ten\_hang).strip().lower()          for item in self.trHienThi.get\_children():              values = self.trHienThi.item(item)["values"]              ma\_trong\_bang = values[0]              ten\_trong\_bang = values[1].lower() if values[1] else ""              if ma\_hang\_hien\_tai and ma\_trong\_bang == ma\_hang\_hien\_tai:                  continue              if ten\_trong\_bang == ten\_hang:                  return True          return False      def them(self):          ma = self.txt\_ma.get().strip()          ten = self.txt\_ten.get().strip()          quocgia = self.txt\_quocgia.get().strip()          if not ma:              messagebox.showwarning("Cảnh báo", "Vui lòng nhập Mã hãng!")              self.txt\_ma.focus()              return          if not ten:              messagebox.showwarning("Cảnh báo", "Vui lòng nhập Tên hãng!")              self.txt\_ten.focus()              return          for item in self.trHienThi.get\_children():              if self.trHienThi.item(item)["values"][0] == ma:                  messagebox.showwarning("Cảnh báo", f"Mã hãng '{ma}' đã tồn tại!")                  return          if self.kiemtra\_trung\_ten\_hang(ten):              messagebox.showwarning("Cảnh báo", f"Tên hãng '{ten}' đã tồn tại!")              self.txt\_ten.focus()              return          self.trHienThi.insert("", "end", values=(ma, ten, quocgia))          self.ds\_them.append((ma, ten, quocgia))          self.xoa\_form()          messagebox.showinfo("Thành công", "Đã thêm dòng mới! Nhấn 'Lưu' để lưu vào CSDL.")      def sua(self):          selected = self.trHienThi.selection()          if not selected:              messagebox.showwarning("Cảnh báo", "Vui lòng chọn dòng cần sửa!")              return          ma = self.txt\_ma.get().strip()          ten = self.txt\_ten.get().strip()          quocgia = self.txt\_quocgia.get().strip()          if not ma:              messagebox.showwarning("Cảnh báo", "Vui lòng nhập Mã hãng!")              return          if not ten:              messagebox.showwarning("Cảnh báo", "Vui lòng nhập Tên hãng!")              return          item = self.trHienThi.item(selected[0])          ma\_cu = item["values"][0]          if self.kiemtra\_trung\_ten\_hang(ten, ma\_cu):              messagebox.showwarning("Cảnh báo", f"Tên hãng '{ten}' đã tồn tại!")              self.txt\_ten.focus()              return          self.trHienThi.item(selected[0], values=(ma, ten, quocgia))          is\_new = any(x[0] == ma\_cu for x in self.ds\_them)          if not is\_new:              self.ds\_sua = [x for x in self.ds\_sua if x[0] != ma\_cu]              self.ds\_sua.append((ma, ten, quocgia, ma\_cu))          else:              self.ds\_them = [                  (ma, ten, quocgia) if x[0] == ma\_cu else x for x in self.ds\_them              ]          self.xoa\_form()          messagebox.showinfo(              "Thành công", "Đã cập nhật dòng! Nhấn 'Lưu' để lưu vào CSDL."          )      def xoa(self):          selected = self.trHienThi.selection()          if not selected:              messagebox.showwarning("Cảnh báo", "Vui lòng chọn dòng cần xóa!")              return          xacnhan = messagebox.askyesno("Xác nhận", "Bạn có chắc muốn xóa dòng này?")          if not xacnhan:              return          item = self.trHienThi.item(selected[0])          ma = item["values"][0]          self.trHienThi.delete(selected[0])          is\_new = any(x[0] == ma for x in self.ds\_them)          if is\_new:              self.ds\_them = [x for x in self.ds\_them if x[0] != ma]          else:              if ma not in self.ds\_xoa:                  self.ds\_xoa.append(ma)          self.xoa\_form()          messagebox.showinfo("Thành công", "Đã xóa dòng! Nhấn 'Lưu' để lưu vào CSDL.")      def luu(self):          try:              if not self.ds\_them and not self.ds\_sua and not self.ds\_xoa:                  messagebox.showinfo("Thông báo", "Không có thay đổi để lưu!")                  return              confirm = messagebox.askyesno("Xác nhận", "Bạn có chắc muốn lưu các thay đổi?")              if not confirm:                  return              for ma in self.ds\_xoa:                  self.cursor.execute("DELETE FROM HangSanXuat WHERE MaHang = ?", (ma,))              for ma, ten, quocgia in self.ds\_them:                  self.cursor.execute("INSERT INTO HangSanXuat (MaHang, TenHang, QuocGia) VALUES (?, ?, ?)", (ma, ten, quocgia))              for ma, ten, quocgia, ma\_cu in self.ds\_sua:                  self.cursor.execute("UPDATE HangSanXuat SET MaHang = ?, TenHang = ?, QuocGia = ? WHERE MaHang = ?", (ma, ten, quocgia, ma\_cu))              self.conn.commit()              messagebox.showinfo("Thành công", "Đã lưu thay đổi vào CSDL!")          except pyodbc.IntegrityError as e:              self.conn.rollback()              messagebox.showerror("Lỗi", f"Lỗi ràng buộc dữ liệu: {str(e)}")          except Exception as e:              self.conn.rollback()              messagebox.showerror("Lỗi", f"Không thể lưu dữ liệu: {str(e)}")          finally:              self.hienthi\_dulieu()              self.xoa\_form()              self.ds\_them.clear()              self.ds\_sua.clear()              self.ds\_xoa.clear()              self.load\_data()      def lammoi(self):          confirm = messagebox.askyesno("Xác nhận", "Bạn có chắc muốn hủy các thay đổi?")          if not confirm:              return          self.ds\_them.clear()          self.ds\_sua.clear()          self.ds\_xoa.clear()          self.hienthi\_dulieu()          self.xoa\_form()          self.txt\_timkiem.delete(0, tk.END)          messagebox.showinfo("Thông báo", "Đã làm mới dữ liệu!")      def timkiem(self):          tu\_khoa\_tim = self.txt\_timkiem.get().strip()          if not tu\_khoa\_tim:              messagebox.showinfo("Thông báo", "Vui lòng nhập từ khóa tìm kiếm.")              self.hienthi\_dulieu()              return          try:              for item in self.trHienThi.get\_children():                  self.trHienThi.delete(item)              if self.search\_option.get() == "ma":                  self.cursor.execute("SELECT MaHang, TenHang, QuocGia FROM HangSanXuat WHERE MaHang LIKE ? ORDER BY MaHang", (f"%{tu\_khoa\_tim}%",))              else:                  self.cursor.execute("SELECT MaHang, TenHang, QuocGia FROM HangSanXuat WHERE TenHang LIKE ? ORDER BY MaHang", (f"%{tu\_khoa\_tim}%",))              rows = self.cursor.fetchall()              for row in rows:                  self.trHienThi.insert("", "end", values=(row.MaHang, row.TenHang, row.QuocGia))              if not rows:                  messagebox.showinfo("Thông báo", "Không tìm thấy kết quả!")          except Exception as e:              messagebox.showerror("Lỗi", "Không thể tìm kiếm: " + str(e))      def xoa\_form(self):          self.txt\_ma.delete(0, tk.END)          self.txt\_ten.delete(0, tk.END)          self.txt\_quocgia.delete(0, tk.END)      def huy(self):          self.txt\_timkiem.delete(0, tk.END)          self.hienthi\_dulieu() |

**FORM NHÀ CUNG CẤP**

|  |
| --- |
| import tkinter as tk  from tkinter import ttk, messagebox  import pyodbc  class tabNhaCungCap(tk.Frame):      def \_\_init\_\_(self, parent, conn, controller):          super().\_\_init\_\_(parent, bg="white")          self.conn = conn          self.cursor = conn.cursor()          self.ds\_them = []          self.ds\_sua = []          self.ds\_xoa = []          self.selected\_item = None          self.controller = controller          frame\_search = tk.Frame(self, bg="#E3F2FD", padx=10, pady=10)          frame\_search.pack(fill="x", padx=20, pady=5)          tk.Label(frame\_search, text="Tìm kiếm:", font=("Segoe UI", 10), bg="#E3F2FD").pack(side="left", padx=5)          self.txt\_timkiem = tk.Entry(frame\_search, font=("Segoe UI", 10), width=42)          self.txt\_timkiem.pack(side="left", padx=5)          self.search\_option = tk.StringVar(value="ma")          tk.Radiobutton(frame\_search, text="Theo mã nhà cung cấp", variable=self.search\_option, value="ma", bg="#E3F2FD", font=("Segoe UI", 10)).pack(side="left", padx=10)          tk.Radiobutton(frame\_search, text="Theo tên nhà cung cấp", variable=self.search\_option, value="ten", bg="#E3F2FD", font=("Segoe UI", 10)).pack(side="left")          tk.Button(frame\_search, text="Tìm", font=("Segoe UI", 10, "bold"), bg="#1565C0", fg="white", bd=0, padx=10, pady=5, command=self.timkiem).pack(side="left", padx=10)          tk.Button(frame\_search, text="Hủy", font=("Segoe UI", 10, "bold"), bg="#E53935", fg="white", bd=0, padx=10, pady=5, command=self.huy\_tim\_kiem).pack(side="left", padx=10)          frame\_form = tk.LabelFrame(self, text="Thông tin Nhà cung cấp", bg="white", font=("Segoe UI", 12, "bold"), fg="#0D47A1", padx=10, pady=10)          frame\_form.pack(fill="x", padx=20, pady=10)          tk.Label(frame\_form, text="Mã nhà cung cấp", bg="white", font=("Segoe UI", 10)).grid(row=0, column=0, sticky="w", padx=5, pady=5)          self.txt\_ma = ttk.Entry(frame\_form, width=24)          self.txt\_ma.grid(row=0, column=1, padx=5, pady=5)          tk.Label(frame\_form, text="Tên nhà cung cấp:", bg="white", font=("Segoe UI", 10)).grid(row=0, column=2, sticky="w", padx=5, pady=5)          self.txt\_ten = ttk.Entry(frame\_form, width=68)          self.txt\_ten.grid(row=0, column=3, columnspan=3, padx=5, pady=5)          tk.Label(frame\_form, text="Số điện thoại:", bg="white", font=("Segoe UI", 10)).grid(row=1, column=0, sticky="w", padx=5, pady=5)          self.txt\_sodienthoai = ttk.Entry(frame\_form, width=24)          self.txt\_sodienthoai.grid(row=1, column=1, padx=5, pady=5)          tk.Label(frame\_form, text="Email:", bg="white", font=("Segoe UI", 10)).grid(row=1, column=2, sticky="w", padx=5, pady=5)          self.txt\_email = ttk.Entry(frame\_form, width=28)          self.txt\_email.grid(row=1, column=3, padx=5, pady=5)          tk.Label(frame\_form, text="Địa chỉ:", bg="white", font=("Segoe UI", 10)).grid(row=1, column=4, sticky="w", padx=5, pady=5)          self.txt\_diachi = ttk.Entry(frame\_form, width=28)          self.txt\_diachi.grid(row=1, column=5, padx=5, pady=5)          frame\_buttons = tk.Frame(self, bg="white")          frame\_buttons.pack(pady=10)          tk.Button(frame\_buttons, text="➕ Thêm", bg="#EBDA42", fg="white", font=("Segoe UI", 11, "bold"), padx=20, pady=5, bd=0, command=self.them).grid(row=0, column=0, padx=10)          tk.Button(frame\_buttons, text="✏️ Sửa", bg="#FB8C00", fg="white", font=("Segoe UI", 11, "bold"), padx=20, pady=5, bd=0, command=self.sua).grid(row=0, column=1, padx=10)          tk.Button(frame\_buttons, text="🗑️ Xóa", bg="#E53935", fg="white", font=("Segoe UI", 11, "bold"), padx=20, pady=5, bd=0, command=self.xoa).grid(row=0, column=2, padx=10)          tk.Button(frame\_buttons, text="🔄 Làm mới", bg="#1E88E5", fg="white", font=("Segoe UI", 11, "bold"), padx=20, pady=5, bd=0, command=self.lammoi).grid(row=0, column=3, padx=10)          tk.Button(frame\_buttons, text="💾 Lưu", bg="#43A047", fg="white", font=("Segoe UI", 11, "bold"), padx=20, pady=5, bd=0, command=self.luu).grid(row=0, column=4, padx=10)          frame\_table = tk.Frame(self, bg="white")          frame\_table.pack(fill="both", expand=True, padx=20, pady=10)          columns = ("MaNCC", "TenNCC", "DiaChi", "SoDienThoai", "Email")          scroll\_y = ttk.Scrollbar(frame\_table, orient="vertical")          scroll\_x = ttk.Scrollbar(frame\_table, orient="horizontal")          self.trHienThi = ttk.Treeview(frame\_table, show="headings", columns=columns, height=12, yscrollcommand=scroll\_y.set, xscrollcommand=scroll\_x.set)          scroll\_y.config(command=self.trHienThi.yview)          scroll\_x.config(command=self.trHienThi.xview)          scroll\_y.pack(side="right", fill="y")          scroll\_x.pack(side="bottom", fill="x")          self.trHienThi.pack(fill="both", expand=True)          self.trHienThi.heading("MaNCC", text="Mã nhà cung cấp")          self.trHienThi.heading("TenNCC", text="Tên nhà cung cấp")          self.trHienThi.heading("DiaChi", text="Địa chỉ")          self.trHienThi.heading("SoDienThoai", text="Số điện thoại")          self.trHienThi.heading("Email", text="Email")          self.trHienThi.column("MaNCC", width=120, anchor="center")          self.trHienThi.column("TenNCC", width=250, anchor="w")          self.trHienThi.column("DiaChi", width=200, anchor="w")          self.trHienThi.column("SoDienThoai", width=150, anchor="center")          self.trHienThi.column("Email", width=200, anchor="w")          style = ttk.Style()          style.configure("Treeview.Heading", font=("Segoe UI", 11, "bold"))          style.configure("Treeview", font=("Segoe UI", 10), rowheight=28)          self.trHienThi.bind("<<TreeviewSelect>>", self.chon\_dong)          self.hienthi\_dulieu()      def hienthi\_dulieu(self):          for item in self.trHienThi.get\_children():              self.trHienThi.delete(item)          self.cursor.execute("SELECT MaNCC, TenNCC, DiaChi, SoDienThoai, Email FROM NhaCungCap ORDER BY MaNCC")          for row in self.cursor.fetchall():              self.trHienThi.insert("", "end", values=(row.MaNCC, row.TenNCC, row.DiaChi, row.SoDienThoai, row.Email))      def chon\_dong(self, event=None):          selected = self.trHienThi.selection()          if not selected:              return            self.selected\_item = selected[0]          item = self.trHienThi.item(self.selected\_item)          values = item["values"]          self.xoa\_form()          self.txt\_ma.insert(0, values[0])          self.txt\_ten.insert(0, values[1])          self.txt\_diachi.insert(0, values[2])          sdt = str(values[3])          if sdt and not sdt[0] == "0":              sdt = "0" + sdt          self.txt\_sodienthoai.insert(0, sdt)          self.txt\_email.insert(0, values[4])        def kiemtra\_trung(self, ma= "", ten= "", sdt= "", email= "", ma\_hien\_tai= ""):          for iid in self.trHienThi.get\_children():              v = self.trHienThi.item(iid)["values"]              if ma\_hien\_tai and v[0] == ma\_hien\_tai:                  continue              if ma and v[0] == ma:                  return "Mã nhà cung cấp"              if ten and v[1].strip().lower() == str(ten).strip().lower():                  return "Tên nhà cung cấp"              if self.bo\_so\_0\_dau(str(sdt)) and self.bo\_so\_0\_dau(str(v[3])) == self.bo\_so\_0\_dau(str(sdt)):                  return "Số điện thoại"              if email and v[4].strip().lower() == str(email).strip().lower():                  return "Email"          return None        def bo\_so\_0\_dau(self, chuoi):          if isinstance(chuoi, str) and chuoi[0] == "0":              return chuoi.lstrip("0") or "0"          return str(chuoi)        def kiem\_tra\_du\_lieu(self):          ma = str(self.txt\_ma.get()).strip()          ten = str(self.txt\_ten.get()).strip()          diachi = str(self.txt\_diachi.get()).strip()          sdt = str(self.txt\_sodienthoai.get()).strip()          email = str(self.txt\_email.get()).strip()          if not ma:              messagebox.showwarning("Cảnh báo", "Vui lòng nhập Mã nhà cung cấp!")              return False          if not ten:              messagebox.showwarning("Cảnh báo", "Vui lòng nhập Tên nhà cung cấp!")              return False            if not diachi:              messagebox.showwarning("Cảnh báo", "Vui lòng nhập Địa chi!")              return False            if not sdt:              messagebox.showwarning("Cảnh báo", "Vui lòng nhập Số điện thoại!")              self.txt\_sodienthoai.focus()              return False            if not sdt.isdigit():              messagebox.showwarning("Cảnh báo", "Số điện thoại chỉ được phép nhập số!")              self.txt\_sodienthoai.focus()              return False            if len(sdt) != 10:              messagebox.showwarning("Cảnh báo", "Số điện thoại phải có đúng 10 chữ số!")              self.txt\_sodienthoai.focus()              return False            if sdt[0] != '0':              messagebox.showwarning("Cảnh báo", "Số điện thoại hợp lệ phải bắt đầu bằng số 0!")              self.txt\_sodienthoai.focus()              return False            if not email:              messagebox.showwarning("Cảnh báo", "Vui lòng nhập Email!")              return False            return True      def them(self):          ma = str(self.txt\_ma.get()).strip()          ten = str(self.txt\_ten.get()).strip()          diachi = str(self.txt\_diachi.get()).strip()          sdt = str(self.txt\_sodienthoai.get()).strip()          email = str(self.txt\_email.get()).strip()          if self.kiem\_tra\_du\_lieu() == False:              messagebox.showwarning("Cảnh báo", "Vui lòng nhập dữ liệu!")              return          trung = self.kiemtra\_trung(ma=ma, ten=ten, sdt=sdt, email=email)          if trung:              messagebox.showwarning("Cảnh báo", f"{trung} đã tồn tại!")              return          self.trHienThi.insert("", "end", values=(ma, ten, diachi, sdt, email))          self.ds\_them.append((ma, ten, diachi, sdt, email))          self.xoa\_form()          messagebox.showinfo("Thêm nhà cung cấp", "Thêm nhà cung cấp thành công!")      def sua(self):          sel = self.trHienThi.selection()          if not sel:              messagebox.showwarning("Cảnh báo", "Vui lòng chọn dòng cần sửa!")              return          ma = self.txt\_ma.get().strip()          ten = self.txt\_ten.get().strip()          diachi = self.txt\_diachi.get().strip()          sdt = self.txt\_sodienthoai.get().strip()          email = self.txt\_email.get().strip()          if self.kiem\_tra\_du\_lieu() == False:              messagebox.showwarning("Cảnh báo", "Vui long nhập dữ liệu!")              return          ma\_cu = self.trHienThi.item(sel[0])["values"][0]          trung = self.kiemtra\_trung(ma=ma, ten=ten, sdt=sdt, email=email, ma\_hien\_tai=ma\_cu)          if trung:              messagebox.showwarning("Cảnh báo", f"{trung} đã tồn tại!")              return          self.trHienThi.item(sel[0], values=(ma, ten, diachi, sdt, email))          if any(x[0] == ma\_cu for x in self.ds\_them):              self.ds\_them = [(ma, ten, diachi, sdt, email) if x[0] == ma\_cu else x for x in self.ds\_them]          else:              self.ds\_sua = [x for x in self.ds\_sua if x[0] != ma\_cu]              self.ds\_sua.append((ma, ten, diachi, sdt, email, ma\_cu))          self.xoa\_form()          messagebox.showinfo("Cập nhật nhà cung cấp", "Cập nhật nhà cung cấp thành công!")      def xoa(self):          sel = self.trHienThi.selection()          if not sel:              messagebox.showwarning("Cảnh báo", "Vui lòng chọn dòng cần xóa!")              return          if not messagebox.askyesno("Xác nhận", "Bạn có chắc muốn xóa dòng này?"):              return          ma = self.trHienThi.item(sel[0])["values"][0]          self.trHienThi.delete(sel[0])          if any(x[0] == ma for x in self.ds\_them):              self.ds\_them = [x for x in self.ds\_them if x[0] != ma]          else:              if ma not in self.ds\_xoa:                  self.ds\_xoa.append(ma)          self.xoa\_form()          messagebox.showinfo("Thành công", "Đã xóa dòng! Nhấn 'Lưu' để lưu vào CSDL.")      def luu(self):          if not (self.ds\_them or self.ds\_sua or self.ds\_xoa):              messagebox.showinfo("Thông báo", "Không có thay đổi để lưu!")              return          if not messagebox.askyesno("Xác nhận", "Bạn có chắc muốn lưu các thay đổi?"):              return          try:              for ma in self.ds\_xoa:                  self.cursor.execute("DELETE FROM NhaCungCap WHERE MaNCC = ?", (ma,))              for ma, ten, diachi, sdt, email in self.ds\_them:                  self.cursor.execute("INSERT INTO NhaCungCap (MaNCC, TenNCC, DiaChi, SoDienThoai, Email) VALUES (?, ?, ?, ?, ?)", (ma, ten, diachi, sdt, email))              for ma, ten, diachi, sdt, email, ma\_cu in self.ds\_sua:                  self.cursor.execute("UPDATE NhaCungCap SET MaNCC=?, TenNCC=?, DiaChi=?, SoDienThoai=?, Email=? WHERE MaNCC=?", (ma, ten, diachi, sdt, email, ma\_cu))                self.conn.commit()              messagebox.showinfo("Thành công", "Đã lưu thay đổi vào CSDL!")          except pyodbc.IntegrityError as e:              self.conn.rollback()              messagebox.showerror("Lỗi", f"Lỗi ràng buộc dữ liệu: {str(e)}")          except Exception as e:              self.conn.rollback()              messagebox.showerror("Lỗi", f"Không thể lưu dữ liệu: {str(e)}")          finally:              self.hienthi\_dulieu()              self.xoa\_form()              self.ds\_them.clear()              self.ds\_sua.clear()              self.ds\_xoa.clear()          self.controller.load\_form("NhapHangVaPhieuNhap")      def lammoi(self):          if messagebox.askyesno("Xác nhận", "Bạn có chắc muốn hủy các thay đổi?"):              self.ds\_them.clear()              self.ds\_sua.clear()              self.ds\_xoa.clear()              self.hienthi\_dulieu()              self.xoa\_form()              self.txt\_timkiem.delete(0, tk.END)              messagebox.showinfo("Thông báo", "Đã làm mới dữ liệu!")      def timkiem(self):          kw = self.txt\_timkiem.get().strip()          if not kw:              messagebox.showinfo("Thông báo", "Vui lòng nhập từ khóa tìm kiếm.")              self.hienthi\_dulieu()              return          for item in self.trHienThi.get\_children():              self.trHienThi.delete(item)          sql = ("SELECT MaNCC, TenNCC, DiaChi, SoDienThoai, Email FROM NhaCungCap "                 "WHERE MaNCC LIKE ? ORDER BY MaNCC" if self.search\_option.get() == "ma"                 else "WHERE TenNCC LIKE ? ORDER BY MaNCC")          self.cursor.execute(sql, (f"%{kw}%",))          rows = self.cursor.fetchall()          for r in rows:              self.trHienThi.insert("", "end", values=(r.MaNCC, r.TenNCC, r.DiaChi, r.SoDienThoai, r.Email))          if not rows:              messagebox.showinfo("Kết quả", "Không tìm thấy kết quả!")      def xoa\_form(self):          self.txt\_ma.delete(0, tk.END)          self.txt\_ten.delete(0, tk.END)          self.txt\_sodienthoai.delete(0, tk.END)          self.txt\_email.delete(0, tk.END)          self.txt\_diachi.delete(0, tk.END)      def huy\_tim\_kiem(self):          self.txt\_timkiem.delete(0, tk.END)          self.hienthi\_dulieu() |

**FORM BẢO HÀNH**

|  |
| --- |
| import tkinter as tk  from tkinter import ttk, messagebox, filedialog  from tkcalendar import DateEntry  from datetime import date, datetime, timedelta  import pyodbc  import os  from docx import Document  from docx.shared import Inches, Pt, RGBColor, Cm  from docx.enum.text import WD\_ALIGN\_PARAGRAPH, WD\_TAB\_ALIGNMENT, WD\_TAB\_LEADER  class tabBaoHanh(tk.Frame):      def \_\_init\_\_(self, parent, conn, user):          super().\_\_init\_\_(parent, bg="white")          self.conn = conn          self.cursor = conn.cursor()          self.user = user          self.ds\_them = []          self.ds\_sua = []          self.ds\_xoa = []          # === KHUNG TÌM KIẾM ===          frame\_search = tk.Frame(self, bg="#E3F2FD", padx=10, pady=10)          frame\_search.pack(fill="x", padx=20, pady=5)          tk.Label(frame\_search, text="Tìm kiếm:", font=("Segoe UI", 10), bg="#E3F2FD").pack(side="left", padx=5)          self.txt\_timkiem = tk.Entry(frame\_search, font=("Segoe UI", 10), width=30, bg="white")          self.txt\_timkiem.pack(side="left", padx=5)          self.txt\_timkiem.bind("<Return>", lambda e: self.timkiem())          self.search\_option = tk.StringVar(master=self, value="mabh")          tk.Radiobutton(frame\_search, text="Mã bảo hành", variable=self.search\_option, value="mabh", bg="#E3F2FD", font=("Segoe UI", 10)).pack(side="left", padx=5)          tk.Radiobutton(frame\_search, text="Mã chi tiết hóa đơn", variable=self.search\_option, value="macthd", bg="#E3F2FD", font=("Segoe UI", 10)).pack(side="left", padx=5)          tk.Radiobutton(frame\_search, text="Mã hóa đơn", variable=self.search\_option, value="mahd", bg="#E3F2FD", font=("Segoe UI", 10)).pack(side="left", padx=5)          tk.Button(frame\_search, text="Tìm", font=("Segoe UI", 10, "bold"), bg="#1565C0", fg="white", bd=0, padx=10, pady=5, command=self.timkiem).pack(side="left", padx=10)          tk.Button(frame\_search, text="Hủy", font=("Segoe UI", 10, "bold"), bg="#E53935", fg="white", bd=0, padx=10, pady=5, command=self.huy).pack(side="left", padx=10)          # === KHUNG THÔNG TIN ===          frame\_form = tk.LabelFrame(self, text="Thông tin Bảo hành", bg="white", font=("Segoe UI", 12, "bold"), fg="#0D47A1", padx=10, pady=10)          frame\_form.pack(fill="x", padx=20, pady=10)          tk.Label(frame\_form, text="Mã BH:", bg="white", font=("Segoe UI", 10)).grid(row=0, column=0, sticky="w", padx=5, pady=5)          self.txt\_mabh = ttk.Entry(frame\_form, width=34)          self.txt\_mabh.grid(row=0, column=1, padx=5, pady=5)          tk.Label(frame\_form, text="Mã CTHD:", bg="white", font=("Segoe UI", 10)).grid(row=0, column=2, sticky="w", padx=5, pady=5)          self.cb\_macthd = ttk.Combobox(frame\_form, width=22, font=("Segoe UI", 10), state="readonly")          self.cb\_macthd.grid(row=0, column=3, padx=5, pady=5)          self.cb\_macthd.bind("<<ComboboxSelected>>", lambda e: self.capnhat\_mahd\_theo\_cthd())          tk.Label(frame\_form, text="Mã HD:", bg="white", font=("Segoe UI", 10)).grid(row=0, column=4, sticky="w", padx=5, pady=5)          self.cb\_mahd = ttk.Combobox(frame\_form, width=22, font=("Segoe UI", 10), state="readonly")          self.cb\_mahd.grid(row=0, column=5, padx=5, pady=5)          tk.Label(frame\_form, text="Thời gian (tháng):", bg="white", font=("Segoe UI", 10)).grid(row=1, column=0, sticky="w", padx=5, pady=5)          self.txt\_thoigian = ttk.Entry(frame\_form, width=34)          self.txt\_thoigian.grid(row=1, column=1, padx=5, pady=5)          tk.Label(frame\_form, text="Điều kiện:", bg="white", font=("Segoe UI", 10)).grid(row=1, column=2, sticky="w", padx=5, pady=5)          self.txt\_dieukien = ttk.Entry(frame\_form, width=45)          self.txt\_dieukien.grid(row=1, column=3, columnspan=3, padx=5, pady=5, sticky="we")          tk.Label(frame\_form, text="Ngày BH:", bg="white", font=("Segoe UI", 10)).grid(row=2, column=0, sticky="w", padx=5, pady=5)          self.date\_ngaybaohanh = DateEntry(frame\_form, width=29, font=("Segoe UI", 10), date\_pattern="dd/mm/yyyy")          self.date\_ngaybaohanh.grid(row=2, column=1, padx=5, pady=5)          tk.Label(frame\_form, text="Trạng thái:", bg="white", font=("Segoe UI", 10)).grid(row=2, column=2, sticky="w", padx=5, pady=5)          self.lbl\_trangthai = tk.Label(frame\_form, text="", bg="white", font=("Segoe UI", 10, "bold"))          self.lbl\_trangthai.grid(row=2, column=3, sticky="w", padx=5, pady=5)          # === NÚT CHỨC NĂNG ===          frame\_buttons = tk.Frame(self, bg="white")          frame\_buttons.pack(pady=10)          if self.user == "admin":              tk.Button(frame\_buttons, text="➕ Thêm", bg="#EBDA42", fg="white", font=("Segoe UI", 11, "bold"), padx=20, pady=5, bd=0, command=self.them).grid(row=0, column=0, padx=8)              tk.Button(frame\_buttons, text="✏️ Sửa", bg="#FB8C00", fg="white", font=("Segoe UI", 11, "bold"), padx=20, pady=5, bd=0, command=self.sua).grid(row=0, column=1, padx=8)              tk.Button(frame\_buttons, text="🗑️ Xóa", bg="#E53935", fg="white", font=("Segoe UI", 11, "bold"), padx=20, pady=5, bd=0, command=self.xoa).grid(row=0, column=2, padx=8)              tk.Button(frame\_buttons, text="🔄 Làm mới", bg="#1E88E5", fg="white", font=("Segoe UI", 11, "bold"), padx=20, pady=5, bd=0, command=self.lammoi).grid(row=0, column=3, padx=8)              tk.Button(frame\_buttons, text="💾 Lưu", bg="#43A047", fg="white", font=("Segoe UI", 11, "bold"), padx=20, pady=5, bd=0, command=self.luu).grid(row=0, column=4, padx=8)          tk.Button(frame\_buttons, text="🖨️ In bảo hành", bg="#E51E9C", fg="white", font=("Segoe UI", 11, "bold"), padx=15, command=self.InBaoHanh, pady=5, bd=0).grid(row=0, column=5, padx=8)          # === BẢNG HIỂN THỊ ===          frame\_table = tk.Frame(self, bg="white")          frame\_table.pack(fill="both", expand=True, padx=20, pady=10)          columns = ("MaBH", "MaCTHD", "MaHD", "ThoiGianBaoHanh", "DieuKien", "NgayBaoHanh", "TrangThai")          scroll\_y = ttk.Scrollbar(frame\_table, orient="vertical")          scroll\_x = ttk.Scrollbar(frame\_table, orient="horizontal")          self.trHienThi = ttk.Treeview(frame\_table, columns=columns, show="headings", height=14, yscrollcommand=scroll\_y.set, xscrollcommand=scroll\_x.set)          scroll\_y.pack(side="right", fill="y")          scroll\_x.pack(side="bottom", fill="x")          self.trHienThi.pack(fill="both", expand=True)          self.trHienThi.heading("MaBH", text="Mã BH")          self.trHienThi.heading("MaCTHD", text="Mã CTHD")          self.trHienThi.heading("MaHD", text="Mã HD")          self.trHienThi.heading("ThoiGianBaoHanh", text="Thời gian (tháng)")          self.trHienThi.heading("DieuKien", text="Điều kiện")          self.trHienThi.heading("NgayBaoHanh", text="Ngày BH")          self.trHienThi.heading("TrangThai", text="Trạng thái")          self.trHienThi.column("MaBH", width=100, anchor="center")          self.trHienThi.column("MaCTHD", width=100, anchor="center")          self.trHienThi.column("MaHD", width=100, anchor="center")          self.trHienThi.column("ThoiGianBaoHanh", width=130, anchor="center")          self.trHienThi.column("DieuKien", width=250, anchor="w")          self.trHienThi.column("NgayBaoHanh", width=110, anchor="center")          self.trHienThi.column("TrangThai", width=120, anchor="center")          style = ttk.Style()          style.configure("Treeview.Heading", font=("Segoe UI", 11, "bold"))          style.configure("Treeview", font=("Segoe UI", 10), rowheight=30)          self.trHienThi.bind("<<TreeviewSelect>>", self.chon\_dong)          self.hien\_thi\_du\_lieu\_cthd()          self.hienthi\_dulieu()      def hien\_thi\_du\_lieu\_cthd(self):          try:              sql\_query = """                  SELECT DISTINCT cthd.MaCTHD                  FROM ChiTietHoaDon cthd                  JOIN HoaDonBan hdb ON hdb.MaHD = cthd.MaHD                  WHERE hdb.TrangThai = N'Đã thanh toán'              """              params = []                if self.user.lower() != "admin":                  sql\_query += " AND hdb.MaNV = ?"                  params.append(self.user)                sql\_query += " ORDER BY cthd.MaCTHD"                self.cursor.execute(sql\_query, params)              rows = self.cursor.fetchall()              self.cb\_macthd["values"] = [row.MaCTHD for row in rows]              if rows:                  self.cb\_macthd.current(0)                  self.capnhat\_mahd\_theo\_cthd()          except Exception as e:              messagebox.showerror("Lỗi", f"Không tải được danh sách CTHD: {str(e)}")      def capnhat\_mahd\_theo\_cthd(self):          macthd = self.cb\_macthd.get()          if not macthd:              self.cb\_mahd["values"] = []              return          try:              self.cursor.execute(" SELECT MaHD FROM ChiTietHoaDon WHERE MaCTHD = ? ", (macthd,))              row = self.cursor.fetchone()              if row:                  self.cb\_mahd["values"] = [row.MaHD]                  self.cb\_mahd.set(row.MaHD)              else:                  self.cb\_mahd["values"] = []                  self.cb\_mahd.set("")          except Exception as e:              messagebox.showerror("Lỗi", f"Không tải được Mã HD: {str(e)}")      def hienthi\_dulieu(self):          for item in self.trHienThi.get\_children():              self.trHienThi.delete(item)          try:              sql\_query = """                  SELECT bh.MaBH, bh.MaCTHD, cthd.MaHD, hdb.MaNV, bh.ThoiGianBaoHanh, bh.DieuKien, bh.NgayBaoHanh                  FROM BaoHanh bh                  JOIN ChiTietHoaDon cthd ON bh.MaCTHD = cthd.MaCTHD                  JOIN HoaDonBan hdb ON cthd.MaHD = hdb.MaHD"""              params = []              if self.user != "admin":                  sql\_query += " WHERE hdb.MaNV = ?"                  params.append(self.user)                sql\_query += " ORDER BY bh.MaBH"                self.cursor.execute(sql\_query, params)              rows = self.cursor.fetchall()              for row in rows:                  ngay\_bh = row.NgayBaoHanh.date() if hasattr(row.NgayBaoHanh, 'date') else date.fromisoformat(str(row.NgayBaoHanh).split()[0])                    # Tính Hạn/Hết Hạn dựa trên NgayBaoHanh trong CSDL                  ngay\_het = ngay\_bh + timedelta(days=row.ThoiGianBaoHanh \* 30)                  trangthai\_hienthi = "CÒN HẠN" if ngay\_het >= date.today() else "HẾT HẠN"                    # Màu sắc hiển thị                  tag = 'con\_han' if trangthai\_hienthi == 'CÒN HẠN' else 'het\_han'                  self.trHienThi.insert("", "end",                      values=(row.MaBH, row.MaCTHD, row.MaHD, row.ThoiGianBaoHanh, row.DieuKien or "", self.chuyen\_yyyy\_sang\_dd(row.NgayBaoHanh), trangthai\_hienthi),                      tags=(tag,))            except Exception as e:              messagebox.showerror("Lỗi", "Không thể tải dữ liệu: " + str(e))      def chuyen\_yyyy\_sang\_dd(self, ngay\_db):          if ngay\_db is None:              return ""          ngay\_str = str(ngay\_db).strip().split()[0]          if "-" in ngay\_str and len(ngay\_str.split("-")) == 3:              y, m, d = ngay\_str.split("-")              return f"{d.zfill(2)}/{m.zfill(2)}/{y}"          return ngay\_str      def chon\_dong(self, event):          selected = self.trHienThi.selection()          if not selected:              return          values = self.trHienThi.item(selected[0])["values"]          self.xoa\_form()          self.txt\_mabh.insert(0, values[0])          self.cb\_macthd.set(values[1])          self.capnhat\_mahd\_theo\_cthd()          self.txt\_thoigian.insert(0, values[3])          self.txt\_dieukien.insert(0, values[4])          d, m, y = map(int, values[5].split("/"))          self.date\_ngaybaohanh.set\_date(date(y, m, d))          self.lbl\_trangthai.config(text="CÒN HẠN BẢO HÀNH" if values[6] == "CÒN HẠN" else "ĐÃ HẾT HẠN BẢO HÀNH", fg="green" if values[6] == "CÒN HẠN" else "red")      def them(self):          mabh = self.txt\_mabh.get().strip()          macthd = self.cb\_macthd.get()          mahd = self.cb\_mahd.get()          thoigian = self.txt\_thoigian.get().strip()          dieukien = self.txt\_dieukien.get().strip()          ngaybh = self.date\_ngaybaohanh.get\_date()          if not all([mabh, macthd, thoigian]):              messagebox.showwarning("Cảnh báo", "Vui lòng nhập đầy đủ thông tin bắt buộc!")              return          try:              thoigian\_int = int(thoigian)              if thoigian\_int <= 0:                  raise ValueError          except:              messagebox.showwarning("Cảnh báo", "Thời gian phải là số nguyên dương!")              return          for item in self.trHienThi.get\_children():              if self.trHienThi.item(item)["values"][0] == mabh:                  messagebox.showwarning("Cảnh báo", f"Mã BH '{mabh}' đã tồn tại!")                  return          ngay\_het = ngaybh + timedelta(days=thoigian\_int \* 30)          trangthai = "CÒN HẠN" if ngay\_het >= date.today() else "HẾT HẠN"          self.trHienThi.insert("", "end", values=(mabh, macthd, mahd, thoigian\_int, dieukien, self.chuyen\_yyyy\_sang\_dd(ngaybh), trangthai))          self.ds\_them.append((mabh, macthd, thoigian\_int, dieukien, ngaybh))          self.xoa\_form()          messagebox.showinfo("Thành công", "Đã thêm! Nhấn 'Lưu' để lưu vào CSDL.")      def sua(self):          selected = self.trHienThi.selection()          if not selected:              messagebox.showwarning("Cảnh báo", "Chọn dòng cần sửa!")              return          mabh = self.txt\_mabh.get().strip()          macthd = self.cb\_macthd.get()          thoigian = self.txt\_thoigian.get().strip()          dieukien = self.txt\_dieukien.get().strip()          ngaybh = self.date\_ngaybaohanh.get\_date()          if not all([mabh, macthd, thoigian]):              messagebox.showwarning("Cảnh báo", "Vui lòng nhập đầy đủ!")              return          try:              thoigian\_int = int(thoigian)              if thoigian\_int <= 0:                  raise ValueError          except:              messagebox.showwarning("Cảnh báo", "Thời gian phải là số nguyên dương!")              return          old\_mabh = self.trHienThi.item(selected[0])["values"][0]          ngay\_het = ngaybh + timedelta(days=thoigian\_int \* 30)          trangthai = "CÒN HẠN" if ngay\_het >= date.today() else "HẾT HẠN"          self.trHienThi.item(selected[0], values=(mabh, macthd, self.cb\_mahd.get(), thoigian\_int, dieukien, self.chuyen\_yyyy\_sang\_dd(ngaybh), trangthai))          self.ds\_sua = [x for x in self.ds\_sua if x[0] != old\_mabh]          self.ds\_sua.append((mabh, macthd, thoigian\_int, dieukien, ngaybh, old\_mabh))          self.xoa\_form()          messagebox.showinfo("Thành công", "Đã sửa! Nhấn 'Lưu' để cập nhật.")      def xoa(self):          selected = self.trHienThi.selection()          if not selected:              messagebox.showwarning("Cảnh báo", "Chọn dòng cần xóa!")              return          if not messagebox.askyesno("Xác nhận", "Xóa dòng này?"):              return          mabh = self.trHienThi.item(selected[0])["values"][0]          self.trHienThi.delete(selected[0])          self.ds\_them = [x for x in self.ds\_them if x[0] != mabh]          if mabh not in self.ds\_xoa:              self.ds\_xoa.append(mabh)          self.xoa\_form()          messagebox.showinfo("Thành công", "Đã xóa! Nhấn 'Lưu' để cập nhật CSDL.")      def luu(self):          if not (self.ds\_them or self.ds\_sua or self.ds\_xoa):              messagebox.showinfo("Thông báo", "Không có thay đổi!")              return          if not messagebox.askyesno("Xác nhận", "Lưu tất cả thay đổi?"):              return          try:              for mabh in self.ds\_xoa:                  self.cursor.execute("DELETE FROM BaoHanh WHERE MaBH = ?", (mabh,))              for mabh, macthd, thoigian, dieukien, ngaybh in self.ds\_them:                  ngaybh\_str = ngaybh.strftime('%Y-%m-%d')                  self.cursor.execute("""                      INSERT INTO BaoHanh (MaBH, MaCTHD, ThoiGianBaoHanh, DieuKien, NgayBaoHanh)                      VALUES (?, ?, ?, ?, ?)                      """,(mabh, macthd, thoigian, dieukien, ngaybh\_str),)              for mabh, macthd, thoigian, dieukien, ngaybh, old\_mabh in self.ds\_sua:                  ngaybh\_str = ngaybh.strftime('%Y-%m-%d')                  self.cursor.execute("""                      UPDATE BaoHanh SET MaBH=?, MaCTHD=?, ThoiGianBaoHanh=?, DieuKien=?, NgayBaoHanh=?                      WHERE MaBH=?                      """,(mabh, macthd, thoigian, dieukien, ngaybh\_str, old\_mabh),)              self.conn.commit()              messagebox.showinfo("Thành công", "Đã lưu tất cả thay đổi!")          except pyodbc.IntegrityError as e:              self.conn.rollback()              messagebox.showerror("Lỗi FK", f"Vi phạm ràng buộc:\n{e}")          except Exception as e:              self.conn.rollback()              messagebox.showerror("Lỗi", f"Lỗi CSDL: {e}")          self.hienthi\_dulieu()          self.xoa\_form()          self.ds\_them.clear()          self.ds\_sua.clear()          self.ds\_xoa.clear()      def lammoi(self):          if messagebox.askyesno("Xác nhận", "Hủy tất cả thay đổi?"):              self.ds\_them.clear()              self.ds\_sua.clear()              self.ds\_xoa.clear()              self.hienthi\_dulieu()              self.xoa\_form()              self.txt\_timkiem.delete(0, tk.END)      def timkiem(self):          keyword = self.txt\_timkiem.get().strip()          if not keyword:              self.hienthi\_dulieu()              return          try:              for item in self.trHienThi.get\_children():                  self.trHienThi.delete(item)              lua\_chon = self.search\_option.get()              sql = ""              param = f"%{keyword}%"              if lua\_chon == "mabh":                  sql = """                      SELECT bh.MaBH, bh.MaCTHD, cthd.MaHD, bh.ThoiGianBaoHanh, bh.DieuKien, bh.NgayBaoHanh                      FROM BaoHanh bh                      JOIN ChiTietHoaDon cthd ON bh.MaCTHD = cthd.MaCTHD                      WHERE bh.MaBH LIKE ?                  """              elif lua\_chon == "macthd":                  sql = """                      SELECT bh.MaBH, bh.MaCTHD, cthd.MaHD, bh.ThoiGianBaoHanh, bh.DieuKien, bh.NgayBaoHanh                      FROM BaoHanh bh                      JOIN ChiTietHoaDon cthd ON bh.MaCTHD = cthd.MaCTHD                      WHERE bh.MaCTHD LIKE ?                  """              elif lua\_chon == "mahd":                  sql = """                      SELECT bh.MaBH, bh.MaCTHD, cthd.MaHD, bh.ThoiGianBaoHanh, bh.DieuKien, bh.NgayBaoHanh                      FROM BaoHanh bh                      JOIN ChiTietHoaDon cthd ON bh.MaCTHD = cthd.MaCTHD                      WHERE cthd.MaHD LIKE ?                  """              self.cursor.execute(sql, (param,))              rows = self.cursor.fetchall()              for row in rows:                  ngay\_bh = row.NgayBaoHanh.date() if hasattr(row.NgayBaoHanh, 'date') else date.fromisoformat(str(row.NgayBaoHanh).split()[0])                  ngay\_het = ngay\_bh + timedelta(days=row.ThoiGianBaoHanh \* 30)                  tt = "CÒN HẠN" if ngay\_het >= date.today() else "HẾT HẠN"                  self.trHienThi.insert("", "end", values=(row.MaBH, row.MaCTHD, row.MaHD, row.ThoiGianBaoHanh, row.DieuKien or "", self.chuyen\_yyyy\_sang\_dd(row.NgayBaoHanh), tt))              if not rows:                  messagebox.showinfo("Thông báo", "Không tìm thấy!")          except Exception as e:              messagebox.showerror("Lỗi", f"Tìm kiếm lỗi: {e}")      def xoa\_form(self):          self.txt\_mabh.delete(0, tk.END)          self.cb\_macthd.set("")          self.cb\_mahd.set("")          self.txt\_thoigian.delete(0, tk.END)          self.txt\_dieukien.delete(0, tk.END)          self.date\_ngaybaohanh.set\_date(date.today())          self.lbl\_trangthai.config(text="")      def huy(self):          self.txt\_timkiem.delete(0, tk.END)          self.hienthi\_dulieu()      def InBaoHanh(self):          selected = self.trHienThi.selection()          if not selected:              messagebox.showwarning("Thông báo", "Vui lòng chọn 1 bảo hành để in.")              return          ma\_bh = self.trHienThi.item(selected[0], "values")[0]          # --- Lấy dữ liệu từ SQL ---          try:              cursor = self.conn.cursor()              cursor.execute("""                  SELECT bh.MaBH, bh.MaCTHD, bh.ThoiGianBaoHanh, bh.DieuKien, bh.NgayBaoHanh,                      cthd.MaHD, tv.MaTivi, tv.TenTivi, kh.MaKH, kh.TenKH                  FROM BaoHanh bh                  JOIN ChiTietHoaDon cthd ON bh.MaCTHD = cthd.MaCTHD                  JOIN HoaDonBan hdb ON cthd.MaHD = hdb.MaHD                  JOIN KhachHang kh ON hdb.MaKH = kh.MaKH                  JOIN Tivi tv ON cthd.MaTivi = tv.MaTivi                  WHERE bh.MaBH = ?              """, (ma\_bh,))              bh = cursor.fetchone()              cursor.close()              if not bh:                  messagebox.showerror("Lỗi", f"Không tìm thấy thông tin bảo hành {ma\_bh}")                  return          except Exception as e:              messagebox.showerror("Lỗi CSDL", f"Không thể lấy dữ liệu để in phiếu bảo hành:\n{e}")              return          # --- Tạo file Word ---          try:              document = Document()              # Cấu hình font mặc định              style = document.styles['Normal']              style.font.name = 'Times New Roman'              style.font.size = Pt(11)              # --- Tiêu đề ---              title = document.add\_paragraph()              title.alignment = WD\_ALIGN\_PARAGRAPH.CENTER              run = title.add\_run("PHIẾU BẢO HÀNH SẢN PHẨM")              run.bold = True              run.font.size = Pt(16)              document.add\_paragraph()              # --- Thông tin chung ---              TAB\_POS = Cm(8)              p1 = document.add\_paragraph()              p1.paragraph\_format.tab\_stops.add\_tab\_stop(TAB\_POS, WD\_TAB\_ALIGNMENT.LEFT, WD\_TAB\_LEADER.SPACES)              p1.add\_run("Mã bảo hành: ").bold = True              run = p1.add\_run(bh.MaBH)              run.bold = True              run.font.color.rgb = RGBColor(0x15, 0x65, 0xC0)              p1.add\_run("\t")              p1.add\_run("Mã chi tiết hóa đơn: ").bold = True              p1.add\_run(bh.MaCTHD)              p2 = document.add\_paragraph()              p2.paragraph\_format.tab\_stops.add\_tab\_stop(TAB\_POS, WD\_TAB\_ALIGNMENT.LEFT, WD\_TAB\_LEADER.SPACES)              p2.add\_run("Mã hóa đơn: ").bold = True              p2.add\_run(bh.MaHD)              p2.add\_run("\t")              p2.add\_run("Mã khách hàng: ").bold = True              p2.add\_run(bh.MaKH)              p3 = document.add\_paragraph()              p3.paragraph\_format.tab\_stops.add\_tab\_stop(TAB\_POS, WD\_TAB\_ALIGNMENT.LEFT, WD\_TAB\_LEADER.SPACES)              p3.add\_run("Tên khách hàng: ").bold = True              p3.add\_run(bh.TenKH)              p3.add\_run("\t")              p3.add\_run("Sản phẩm: ").bold = True              p3.add\_run(f"{bh.MaTivi} - {bh.TenTivi}")              document.add\_paragraph()              # --- Thông tin bảo hành ---              ngay\_bh = bh.NgayBaoHanh.date() if hasattr(bh.NgayBaoHanh, 'date') else datetime.strptime(str(bh.NgayBaoHanh).split()[0], "%Y-%m-%d").date()              ngay\_het = ngay\_bh + timedelta(days=bh.ThoiGianBaoHanh \* 30)              tt = "CÒN HẠN" if ngay\_het >= datetime.today().date() else "HẾT HẠN"              document.add\_paragraph(f"Ngày bảo hành: {ngay\_bh.strftime('%d/%m/%Y')}")              document.add\_paragraph(f"Thời gian bảo hành: {bh.ThoiGianBaoHanh} tháng")              document.add\_paragraph(f"Ngày hết hạn: {ngay\_het.strftime('%d/%m/%Y')}")              document.add\_paragraph(f"Tình trạng: {tt}")              document.add\_paragraph(f"Điều kiện bảo hành: {bh.DieuKien or 'Không có'}")              document.add\_paragraph()              document.add\_paragraph("Sản phẩm sẽ được bảo hành miễn phí nếu đáp ứng các điều kiện nêu trên.", style="Normal")              # --- Khu vực ký tên ---              document.add\_paragraph()              document.add\_paragraph()              sig\_table = document.add\_table(rows=2, cols=3)              sig\_table.style = None              sig\_table.autofit = False              headers = ["Người lập phiếu", "Kỹ thuật viên", "Khách hàng"]              for i, h in enumerate(headers):                  cell = sig\_table.cell(0, i)                  p = cell.paragraphs[0]                  p.alignment = WD\_ALIGN\_PARAGRAPH.CENTER                  run = p.add\_run(h)                  run.bold = True              for i in range(3):                  cell = sig\_table.cell(1, i)                  p = cell.paragraphs[0]                  p.alignment = WD\_ALIGN\_PARAGRAPH.CENTER                  p.add\_run("(Ký, ghi rõ họ tên)")              # --- Chọn nơi lưu file ---              default\_name = f"PhieuBaoHanh\_{ma\_bh}.docx"              file\_path = filedialog.asksaveasfilename(                  title="Chọn nơi lưu phiếu bảo hành",                  defaultextension=".docx",                  initialfile=default\_name,                  filetypes=[("Word Document", "\*.docx")]              )              if not file\_path:                  messagebox.showinfo("Thông báo", "Đã hủy lưu file.")                  return              # --- Lưu file ---              document.save(file\_path)              messagebox.showinfo("Thành công", f"Đã tạo phiếu bảo hành tại:\n{file\_path}")          except Exception as e:              messagebox.showerror("Lỗi", f"Lỗi khi tạo file Word:\n{e}") |

**FORM QUẢN LÝ KHÁCH HÀNG**

|  |
| --- |
| import tkinter as tk  from tkinter import ttk, messagebox  import pyodbc  # === TAB KHÁCH HÀNG ===  class QuanLyKhachHang(tk.Frame):      def \_\_init\_\_(self, parent, controller, conn, user):          super().\_\_init\_\_(parent, bg="white")          self.controller = controller          self.conn = conn          self.cursor = conn.cursor()          self.ds\_them = []          self.ds\_sua = []          self.ds\_xoa = []          lbl\_title = tk.Label(self, text="QUẢN LÝ KHÁCH HÀNG", font=("Segoe UI", 16, "bold"), bg="white", fg="#0D47A1",)          lbl\_title.pack()          # === KHUNG TÌM KIẾM ===          frame\_search = tk.Frame(self, bg="#E3F2FD", padx=10, pady=10)          frame\_search.pack(fill="x", padx=20, pady=5)          tk.Label(frame\_search, text="🔍 Tìm kiếm:", font=("Segoe UI", 10), bg="#E3F2FD").pack(side="left", padx=5)          self.txt\_timkiem = tk.Entry(frame\_search, font=("Segoe UI", 10), width=50, bg="white")          self.txt\_timkiem.pack(side="left", padx=5)          self.txt\_timkiem.bind("<Return>", lambda e: self.timkiem())          self.search\_option = tk.StringVar(value="ma")          tk.Radiobutton(frame\_search, text="Theo mã khách hàng", variable=self.search\_option, value="ma", bg="#E3F2FD", font=("Segoe UI", 10)).pack(side="left", padx=10)          tk.Radiobutton(frame\_search, text="Theo tên khách hàng", variable=self.search\_option, value="ten", bg="#E3F2FD", font=("Segoe UI", 10)).pack(side="left")          tk.Button(frame\_search, text="Tìm", font=("Segoe UI", 10, "bold"), bg="#1565C0", fg="white", bd=0, padx=10, pady=5, command=self.timkiem).pack(side="left", padx=10)          tk.Button(frame\_search, text="Hủy", font=("Segoe UI", 10, "bold"), bg="#E53935", fg="white", bd=0, padx=10, pady=5, command=self.huy).pack(side="left", padx=10)          # === KHUNG THÔNG TIN ===          frame\_form = tk.LabelFrame(self, text="Thông tin Khách hàng", bg="white", font=("Segoe UI", 12, "bold"), fg="#0D47A1", padx=10, pady=10)          frame\_form.pack(fill="x", padx=20, pady=10)          # Dòng 1          tk.Label(frame\_form, text="Mã khách hàng:", bg="white", font=("Segoe UI", 10)).grid(row=0, column=0, sticky="w", padx=5, pady=5)          self.txt\_ma = ttk.Entry(frame\_form, width=28)          self.txt\_ma.grid(row=0, column=1, padx=5, pady=5)          tk.Label(frame\_form, text="Tên khách hàng:", bg="white", font=("Segoe UI", 10)).grid(row=0, column=2, sticky="w", padx=5, pady=5)          self.txt\_ten = ttk.Entry(frame\_form, width=28)          self.txt\_ten.grid(row=0, column=3, padx=5, pady=5)          tk.Label(frame\_form, text="Số điện thoại:", bg="white", font=("Segoe UI", 10)).grid(row=0, column=4, sticky="w", padx=5, pady=5)          self.txt\_sdt = ttk.Entry(frame\_form, width=28)          self.txt\_sdt.grid(row=0, column=5, padx=5, pady=5)          # Dòng 2          tk.Label(frame\_form, text="Email:", bg="white", font=("Segoe UI", 10)).grid(row=1, column=0, sticky="w", padx=5, pady=5)          self.txt\_email = ttk.Entry(frame\_form, width=28)          self.txt\_email.grid(row=1, column=1, padx=5, pady=5)          tk.Label(frame\_form, text="Địa chỉ:", bg="white", font=("Segoe UI", 10)).grid(row=1, column=2, sticky="w", padx=5, pady=5)          self.txt\_diachi = ttk.Entry(frame\_form, width=64)          self.txt\_diachi.grid(row=1, column=3, columnspan=3, padx=5, pady=5, sticky="we")          # ==== NÚT CHỨC NĂNG ====          frame\_buttons = tk.Frame(self, bg="white")          frame\_buttons.pack(pady=10)          btn\_them = tk.Button(frame\_buttons, text="➕ Thêm", bg="#EBDA42", fg="white", font=("Segoe UI", 11, "bold"), padx=20, pady=5, bd=0, command=self.them)          btn\_them.grid(row=0, column=0, padx=10)          btn\_sua = tk.Button(frame\_buttons, text="✏️ Sửa", bg="#FB8C00", fg="white", font=("Segoe UI", 11, "bold"), padx=20, pady=5, bd=0, command=self.sua)          btn\_sua.grid(row=0, column=1, padx=10)          btn\_xoa = tk.Button(frame\_buttons, text="🗑️ Xóa", bg="#E53935", fg="white", font=("Segoe UI", 11, "bold"), padx=20, pady=5, bd=0, command=self.xoa)          btn\_xoa.grid(row=0, column=2, padx=10)          btn\_lammoi = tk.Button(frame\_buttons, text="🔄 Làm mới", bg="#1E88E5", fg="white", font=("Segoe UI", 11, "bold"), padx=20, pady=5, bd=0, command=self.lammoi)          btn\_lammoi.grid(row=0, column=3, padx=10)          btn\_luu = tk.Button(frame\_buttons, text="💾 Lưu", bg="#43A047", fg="white", font=("Segoe UI", 11, "bold"), padx=20, pady=5, bd=0, command=self.luu)          btn\_luu.grid(row=0, column=4, padx=10)          # === BẢNG KHÁCH HÀNG ===          frame\_table = tk.Frame(self, bg="white")          frame\_table.pack(fill="both", expand=True, padx=20, pady=10)          columns=("MaKH", "TenKH", "SoDienThoai", "Email", "DiaChi")           # --- Tạo Scrollbar ---          scroll\_y = ttk.Scrollbar(frame\_table, orient="vertical")          scroll\_x = ttk.Scrollbar(frame\_table, orient="horizontal")          self.trHienThi = ttk.Treeview( frame\_table, show="headings",  columns=columns, height=12, yscrollcommand=scroll\_y.set, xscrollcommand=scroll\_x.set)          # --- Gắn Scrollbar ---          scroll\_y.config(command=self.trHienThi.yview)          scroll\_x.config(command=self.trHienThi.xview)          # --- Bố trí Scrollbar ---          scroll\_y.pack(side="right", fill="y")          scroll\_x.pack(side="bottom", fill="x")          self.trHienThi.pack(fill="both", expand=True)          self.trHienThi.heading("MaKH", text="Mã Khách Hàng")          self.trHienThi.heading("TenKH", text="Tên Khách Hàng")          self.trHienThi.heading("SoDienThoai", text="Số Điện Thoại")          self.trHienThi.heading("Email", text="Email")          self.trHienThi.heading("DiaChi", text="Địa Chỉ")          self.trHienThi.column("MaKH", width=120, anchor="center")          self.trHienThi.column("TenKH", width=200)          self.trHienThi.column("SoDienThoai", width=120, anchor="center")          self.trHienThi.column("Email", width=200)          self.trHienThi.column("DiaChi", width=250)          style = ttk.Style()          style.configure("Treeview.Heading", font=("Segoe UI", 11, "bold"))          style.configure("Treeview", font=("Segoe UI", 10), rowheight=28)          self.trHienThi.bind("<<TreeviewSelect>>", self.chon\_dong)          self.load\_data()      def load\_data(self):          try:              for item in self.trHienThi.get\_children():                  self.trHienThi.delete(item)              self.cursor.execute("SELECT MaKH, TenKH, SoDienThoai, Email, DiaChi FROM KhachHang")              rows = self.cursor.fetchall()              for row in rows:                  self.trHienThi.insert("", "end", values=(row.MaKH, row.TenKH, row.SoDienThoai if row.SoDienThoai else "", row.Email if row.Email else "", row.DiaChi if row.DiaChi else ""))          except Exception as e:              messagebox.showerror("Lỗi", "Không thể tải dữ liệu: " + str(e))      def chon\_dong(self, event):          selected = self.trHienThi.selection()          if selected:              item = self.trHienThi.item(selected[0])              values = item["values"]              self.xoa\_form()              self.txt\_ma.insert(0, values[0])              self.txt\_ten.insert(0, values[1])              if str(values[2])[0] == "0":                  std = str(values[2])              else:                  std = "0" + str(values[2])              self.txt\_sdt.insert(0, std)              self.txt\_email.insert(0, values[3] if values[3] else "")              self.txt\_diachi.insert(0, values[4] if values[4] else "")      def bo\_so\_0\_dau(self, chuoi):          if isinstance(chuoi, str) and chuoi[0] == "0":              return chuoi.lstrip("0") or "0"          return str(chuoi)        def kiemtra\_trung(self, ma= "", sdt= "", email= "", ma\_hien\_tai= ""):          for iid in self.trHienThi.get\_children():              v = self.trHienThi.item(iid)["values"]              if ma\_hien\_tai and v[0] == ma\_hien\_tai:                  continue              if ma and v[0] == ma:                  return "Mã khách hàng"              if self.bo\_so\_0\_dau(str(sdt)) and self.bo\_so\_0\_dau(str(v[2])) == self.bo\_so\_0\_dau(str(sdt)):                  return "Số điện thoại"              if email and v[3].strip().lower() == str(email).strip().lower():                  return "Email"          return None        def kiem\_tra\_thong\_tin(self):          ma = self.txt\_ma.get().strip()          ten = self.txt\_ten.get().strip()          sdt = str(self.txt\_sdt.get()).strip()          email = self.txt\_email.get().strip()          diachi = self.txt\_diachi.get().strip()          if not ma:              messagebox.showwarning("Cảnh báo", "Vui lòng nhập Mã khách hàng!")              self.txt\_ma.focus()              return False          if not ten:              messagebox.showwarning("Cảnh báo", "Vui lòng nhập Tên khách hàng!")              self.txt\_ten.focus()              return False            if not sdt:              messagebox.showwarning("Cảnh báo", "Vui lòng nhập Số điện thoại!")              self.txt\_sodienthoai.focus()              return False            if not sdt.isdigit():              messagebox.showwarning("Cảnh báo", "Số điện thoại chỉ được phép nhập số!")              self.txt\_sodienthoai.focus()              return False            if len(sdt) != 10:              messagebox.showwarning("Cảnh báo", "Số điện thoại phải có đúng 10 chữ số!")              self.txt\_sodienthoai.focus()              return False            if sdt[0] != '0':              messagebox.showwarning("Cảnh báo", "Số điện thoại hợp lệ phải bắt đầu bằng số 0!")              self.txt\_sodienthoai.focus()              return False          if not email:              messagebox.showwarning("Cảnh báo", "Vui lòng nhập Email!")              self.txt\_email.focus()              return False            if len(sdt) != 10:              messagebox.showwarning("Cảnh báo", "Số Điện Thoại phải có 10 số!")              self.txt\_sdt.focus()              return False            if not diachi:              messagebox.showwarning("Cảnh báo", "Vui lòng nhập Địa Chị!")              self.txt\_diachi.focus()              return False            return True      def them(self):          ma = self.txt\_ma.get().strip()          ten = self.txt\_ten.get().strip()          sdt = self.txt\_sdt.get().strip()          email = self.txt\_email.get().strip()          diachi = self.txt\_diachi.get().strip()          if self.kiem\_tra\_thong\_tin() == False:              return          trung = self.kiemtra\_trung(ma=ma, sdt=sdt, email=email)          if trung:              messagebox.showwarning("Cảnh báo", f"{trung} đã tồn tại!")              return          self.trHienThi.insert("", "end", values=(ma, ten, sdt, email, diachi))          self.ds\_them.append((ma, ten, sdt, email, diachi))          self.xoa\_form()          messagebox.showinfo("Thành công", "Đã thêm dòng mới! Nhấn 'Lưu' để lưu vào CSDL.")      def sua(self):          selected = self.trHienThi.selection()          if not selected:              return          ma = self.txt\_ma.get().strip()          ten = self.txt\_ten.get().strip()          sdt = self.txt\_sdt.get().strip()          email = self.txt\_email.get().strip()          diachi = self.txt\_diachi.get().strip()          if self.kiem\_tra\_thong\_tin() == False:              messagebox.showwarning("Cảnh báo", "Vui lòng nhập thông tin!")              return          item = self.trHienThi.item(selected[0])            ma\_cu = item["values"][0]          trung = self.kiemtra\_trung(ma=ma, sdt=sdt, email=email, ma\_hien\_tai=ma\_cu)          if trung:              messagebox.showwarning("Cảnh báo", f"{trung} đã tồn tại!")              return          self.trHienThi.item(selected[0], values=(ma, ten, sdt, email, diachi))          is\_new = any(x[0] == ma\_cu for x in self.ds\_them)          if not is\_new:              self.ds\_sua = [x for x in self.ds\_sua if x[0] != ma\_cu]              self.ds\_sua.append((ma, ten, sdt, email, diachi, ma\_cu))          else:              self.ds\_them = [                  (ma, ten, sdt, email, diachi) if x[0] == ma\_cu else x                  for x in self.ds\_them]          self.xoa\_form()          messagebox.showinfo("Thành công", "Đã cập nhật dòng! Nhấn 'Lưu' để lưu vào CSDL.")      def xoa(self):          selected = self.trHienThi.selection()          if not selected:              messagebox.showwarning("Cảnh báo", "Vui lòng chọn dòng cần xóa!")              return          xacnhan = messagebox.askyesno("Xác nhận", "Bạn có chắc muốn xóa dòng này?")          if not xacnhan:              return          item = self.trHienThi.item(selected[0])          ma = item["values"][0]          self.trHienThi.delete(selected[0])          is\_new = any(x[0] == ma for x in self.ds\_them)          if is\_new:              self.ds\_them = [x for x in self.ds\_them if x[0] != ma]          else:              if ma not in self.ds\_xoa:                  self.ds\_xoa.append(ma)          self.xoa\_form()          messagebox.showinfo("Thành công", "Đã xóa dòng! Nhấn 'Lưu' để lưu vào CSDL.")      def luu(self):          try:              if not self.ds\_them and not self.ds\_sua and not self.ds\_xoa:                  messagebox.showinfo("Thông báo", "Không có thay đổi để lưu!")                  return              confirm = messagebox.askyesno("Xác nhận", "Bạn có chắc muốn lưu các thay đổi?")              if not confirm:                  return              for ma in self.ds\_xoa:                  self.cursor.execute("DELETE FROM KhachHang WHERE MaKH = ?", (ma,))              for ma, ten, sdt, email, diachi in self.ds\_them:                 self.cursor.execute("INSERT INTO KhachHang (MaKH, TenKH, SoDienThoai, Email, DiaChi) VALUES (?, ?, ?, ?, ?)", (ma, ten, sdt, email, diachi))              for ma, ten, sdt, email, diachi, ma\_cu in self.ds\_sua:                  self.cursor.execute("UPDATE KhachHang SET MaKH = ?, TenKH = ?, SoDienThoai = ?, Email = ?, DiaChi = ? WHERE MaKH = ?", (ma, ten, sdt, email, diachi, ma\_cu))              self.conn.commit()              messagebox.showinfo("Thành công", "Đã lưu thay đổi vào CSDL!")          except pyodbc.IntegrityError as e:              self.conn.rollback()              messagebox.showerror("Lỗi", f"Lỗi ràng buộc dữ liệu: {str(e)}")          except Exception as e:              self.conn.rollback()              messagebox.showerror("Lỗi", f"Không thể lưu dữ liệu: {str(e)}")          self.load\_data()          self.xoa\_form()          self.ds\_them.clear()          self.ds\_sua.clear()          self.ds\_xoa.clear()          self.controller.load\_form("BanHangVaHoaDon")      def lammoi(self):          confirm = messagebox.askyesno("Xác nhận", "Bạn có chắc muốn hủy các thay đổi?")          if not confirm:              return          self.ds\_them.clear()          self.ds\_sua.clear()          self.ds\_xoa.clear()          self.load\_data()          self.xoa\_form()          self.txt\_timkiem.delete(0, tk.END)          messagebox.showinfo("Thông báo", "Đã làm mới dữ liệu!")      def timkiem(self):          tu\_khoa\_tim = self.txt\_timkiem.get().strip()          if not tu\_khoa\_tim:              messagebox.showinfo("Thông báo", "Vui lòng nhập từ khóa tìm kiếm.")              self.load\_data()              return          try:              for item in self.trHienThi.get\_children():                  self.trHienThi.delete(item)              if self.search\_option.get() == "ma":                  self.cursor.execute(                      "SELECT MaKH, TenKH, SoDienThoai, Email, DiaChi FROM KhachHang WHERE MaKH LIKE ? ORDER BY MaKH",                      (f"%{tu\_khoa\_tim}%",),)              else:                  self.cursor.execute(                      "SELECT MaKH, TenKH, SoDienThoai, Email, DiaChi FROM KhachHang WHERE TenKH LIKE ? ORDER BY MaKH",                      (f"%{tu\_khoa\_tim}%",),)              rows = self.cursor.fetchall()              for row in rows:                  self.trHienThi.insert("", "end", values=(row.MaKH, row.TenKH, row.SoDienThoai if row.SoDienThoai else "", row.Email if row.Email else "", row.DiaChi if row.DiaChi else ""))              if not rows:                  messagebox.showinfo("Thông báo", "Không tìm thấy kết quả!")          except Exception as e:              messagebox.showerror("Lỗi", f"Không thể tìm kiếm: {str(e)}")      def xoa\_form(self):          self.txt\_ma.delete(0, tk.END)          self.txt\_ten.delete(0, tk.END)          self.txt\_sdt.delete(0, tk.END)          self.txt\_email.delete(0, tk.END)          self.txt\_diachi.delete(0, tk.END)      def huy(self):          self.txt\_timkiem.delete(0, tk.END)          self.load\_data() |

**FORM BÁN HÀNG VÀ HÓA ĐƠN**

|  |
| --- |
| import tkinter as tk  from tkinter import ttk, messagebox  import tabBanHang as bh  import tabHoaDon as hd  import pyodbc  #=== Tạo class Bán Hàng và Hóa Đơn  class BanHangVaHoaDon(tk.Frame):      def \_\_init\_\_(self, parent, controller, conn, user):          super().\_\_init\_\_(parent, bg="white")          # === CHUỖI NÉT NỐI ===          self.conn = conn          # === LẤY USER ===          self.user = user          # === LƯU THAM CHIẾU CONTROLLER ===          self.controller = controller          # === TITLE ===          lbl\_title = tk.Label(self, text="BÁN HÀNG & HÓA ĐƠN", font=("Segoe UI", 16, "bold"), bg="white", fg="#0D47A1")          lbl\_title.pack()          # === Tạo tab control ===          tab\_control = ttk.Notebook(self)          tab\_control.pack(fill="both", expand=True, padx=20, pady=10)          self.tab\_hoadon = hd.tabHoaDon(tab\_control, conn, user, self.controller)          self.tab\_banhang = bh.tabBanHang(tab\_control, conn, user, self.tab\_hoadon)          tab\_control.add(self.tab\_banhang, text="🛒 Bán hàng")          tab\_control.add(self.tab\_hoadon, text="🧾 Danh sách Hóa đơn")        # Hàm làm mới tab khi click vào      def load\_data(self):          try:              self.tab\_hoadon.load\_hoa\_don()                self.tab\_banhang.load\_Combobox()            except Exception as e:              messagebox.showerror("Lỗi khi làm mới dữ liệu: " + str(e)) |

**FORM BÁN HÀNG**

|  |
| --- |
| import tkinter as tk  from tkinter import ttk, messagebox  import pyodbc  from tkcalendar import DateEntry  from datetime import datetime  # === TAB BÁN HÀNG ===  class tabBanHang(tk.Frame):      def \_\_init\_\_(self, parent, conn, user, tab\_hoadon=None):          super().\_\_init\_\_(parent, bg="white")          # === CHUỖI KẾT NỐI ===          self.conn = conn          # === LƯU MÃ NHÂN VIÊN ===          self.user = user          # Khai báo trống cho các dict          self.dict\_nv = {}          self.dict\_kh = {}          self.dict\_tivi = {}          # === LƯU THAM CHIẾU  ===          self.tab\_hoadon= tab\_hoadon          # === KHUNG 1 – THÔNG TIN HÓA ĐƠN ===          frame\_phieu = tk.LabelFrame(self, text="Thông tin Hóa đơn", bg="white", font=("Segoe UI", 12, "bold"), fg="#0D47A1", padx=10, pady=10)          frame\_phieu.pack(fill="x", padx=20, pady=10)          tk.Label(frame\_phieu, text="Mã hóa đơn:", bg="white", font=("Segoe UI", 10)).grid(row=0, column=0, sticky="w", padx=5, pady=5)          self.txt\_mahoadonban = ttk.Entry(frame\_phieu, width=46)          self.txt\_mahoadonban.grid(row=0, column=1, padx=5, pady=5)          tk.Label(frame\_phieu, text="Ngày bán:", bg="white", font=("Segoe UI", 10)).grid(row=0, column=2, sticky="w", padx=5, pady=5)          self.dt\_ngayban = DateEntry(frame\_phieu, width=44, date\_pattern="dd/mm/yyyy", state="readonly")          self.dt\_ngayban.grid(row=0, column=3, padx=5, pady=5)          tk.Label(frame\_phieu, text="Mã nhân viên bán:", bg="white", font=("Segoe UI", 10)).grid(row=1, column=0, sticky="w", padx=5, pady=5)          self.cb\_manhanvien = ttk.Combobox(frame\_phieu, width=44, state="readonly")          self.cb\_manhanvien.grid(row=1, column=1, padx=5, pady=5)          if self.user != "admin":              self.cb\_manhanvien.config(state="readonly")          tk.Label(frame\_phieu, text="Mã khách hàng:", bg="white", font=("Segoe UI", 10)).grid(row=1, column=2, sticky="w", padx=5, pady=5)          self.cb\_makhachhang = ttk.Combobox(frame\_phieu, width=44, state="readonly")          self.cb\_makhachhang.grid(row=1, column=3, padx=5, pady=5)          # === KHUNG 2 – SẢN PHẨM BÁN ===          frame\_sanpham = tk.LabelFrame(self, text="Sản phẩm trong Hóa đơn", bg="white", font=("Segoe UI", 12, "bold"), fg="#0D47A1", padx=10, pady=10)          frame\_sanpham.pack(fill="x", padx=20, pady=10)          tk.Label(frame\_sanpham, text="Mã CTHD:", bg="white", font=("Segoe UI", 10)).grid(row=0, column=0, sticky="w", padx=5, pady=5)          self.txt\_macthd = ttk.Entry(frame\_sanpham, width=20)          self.txt\_macthd.grid(row=0, column=1, padx=5, pady=5, sticky="w")          tk.Label(frame\_sanpham, text="Mã Tivi:", bg="white", font=("Segoe UI", 10)).grid(row=0, column=2, sticky="w", padx=5, pady=5)          self.cb\_mativi = ttk.Combobox(frame\_sanpham, width=20, state="readonly")          self.cb\_mativi.grid(row=0, column=3, padx=5, pady=5)          tk.Label(frame\_sanpham, text="Số lượng:", bg="white", font=("Segoe UI", 10)).grid(row=0, column=4, sticky="w", padx=5, pady=5)          self.txt\_soluong = ttk.Entry(frame\_sanpham, width=20)          self.txt\_soluong.grid(row=0, column=5, padx=5, pady=5)          tk.Label(frame\_sanpham, text="Giá bán:", bg="white", font=("Segoe UI", 10)).grid(row=0, column=6, sticky="w", padx=5, pady=5)          self.txt\_giaban = ttk.Entry(frame\_sanpham, width=25, state="disabled")          self.txt\_giaban.grid(row=0, column=7, padx=5, pady=5)          # === NÚT CHỨC NĂNG ===          frame\_buttons = tk.Frame(self, bg="white")          frame\_buttons.pack(pady=10)          tk.Button(frame\_buttons, text="➕ Thêm hóa đơn", bg="#EBDA42", fg="white", font=("Segoe UI", 11, "bold"), command=self.ThemHoaDonChiTiet, padx=20, pady=5, bd=0).grid(row=0, column=0, padx=10)          tk.Button(frame\_buttons, text="✏️ Sửa", bg="#FB8C00", fg="white", font=("Segoe UI", 11, "bold"), command=self.SuaHoaDonChiTiet, padx=20, pady=5, bd=0).grid(row=0, column=1, padx=10)          tk.Button(frame\_buttons, text="🗑️ Xóa", bg="#E53935", fg="white", font=("Segoe UI", 11, "bold"), command=self.XoaHoaDonChiTiet, padx=20, pady=5, bd=0).grid(row=0, column=2, padx=10)          tk.Button(frame\_buttons, text="🔄 Làm mới", bg="#1E88E5", fg="white", font=("Segoe UI", 11, "bold"), command=self.LamMoi, padx=20, pady=5, bd=0).grid(row=0, column=3, padx=10)          # === BẢNG DANH SÁCH SẢN PHẨM ===          frame\_table = tk.Frame(self, bg="white")          frame\_table.pack(fill="both", expand=True, padx=20, pady=10)          columns = ("MaHD", "NgayBan", "MaNV", "TenNV", "MaKH", "TenKH", "MaTivi", "TenTivi", "SoLuong", "GiaBan", "ThanhTien")          # --- Tạo Scrollbar ---          scroll\_y = ttk.Scrollbar(frame\_table, orient="vertical")          scroll\_x = ttk.Scrollbar(frame\_table, orient="horizontal")          self.trHienThi = ttk.Treeview(frame\_table, show="headings", columns=columns, height=12, yscrollcommand=scroll\_y.set, xscrollcommand=scroll\_x.set)          # --- Gắn Scrollbar ---          scroll\_y.config(command=self.trHienThi.yview)          scroll\_x.config(command=self.trHienThi.xview)          # --- Bố trí Scrollbar ---          scroll\_y.pack(side="right", fill="y")          scroll\_x.pack(side="bottom", fill="x")          self.trHienThi.pack(fill="both", expand=True)          self.trHienThi.heading("MaHD", text="Mã hóa đơn")          self.trHienThi.heading("NgayBan", text="Ngày bán")          self.trHienThi.heading("MaNV", text="Mã nhân viên")          self.trHienThi.heading("TenNV", text="Tên nhân viên")          self.trHienThi.heading("MaKH", text="Mã khách hàng")          self.trHienThi.heading("TenKH", text="Tên khách hàng")          self.trHienThi.heading("MaTivi", text="Mã tivi")          self.trHienThi.heading("TenTivi", text="Tên tivi")          self.trHienThi.heading("SoLuong", text="Số lượng")          self.trHienThi.heading("GiaBan", text="Giá bán")          self.trHienThi.heading("ThanhTien", text="Thành tiền")          self.trHienThi.column("MaHD", anchor="center", width=150)          self.trHienThi.column("NgayBan", anchor="center", width=120)          self.trHienThi.column("MaNV", anchor="center", width=150)          self.trHienThi.column("TenNV", anchor="w", width=200)          self.trHienThi.column("MaKH", anchor="center", width=150)          self.trHienThi.column("TenKH", anchor="w", width=200)          self.trHienThi.column("MaTivi", anchor="center", width=150)          self.trHienThi.column("TenTivi", anchor="w", width=200)          self.trHienThi.column("SoLuong", anchor="center", width=150)          self.trHienThi.column("GiaBan", anchor="e", width=150)          self.trHienThi.column("ThanhTien", anchor="e", width=150)          # Style Treeview          style = ttk.Style()          style.configure("Treeview.Heading", font=("Segoe UI", 11, "bold"))          style.configure("Treeview", font=("Segoe UI", 10), rowheight=28)          # ==== TỔNG TIỀN VÀ NÚT TẠO ĐƠN HÀNG ====          frame\_bottom = tk.Frame(self, bg="white")          frame\_bottom.pack(fill="x", padx=20, pady=10)          tk.Label(frame\_bottom, text="Tổng tiền:", bg="white", font=("Segoe UI", 11, "bold")).pack(side="left", padx=5)          self.lbl\_tongtien = tk.Label(frame\_bottom, text="0 VNĐ", bg="white", font=("Segoe UI", 11, "bold"), fg="#E53935")          self.lbl\_tongtien.pack(side="left", padx=5)          tk.Button(frame\_bottom, text="📝 Tạo đơn hàng", bg="#43A047", fg="white",font=("Segoe UI", 11, "bold"), command=self.TaoHoaDon, padx=15, pady=5, bd=0).pack(side="right", padx=5)          self.load\_Combobox()          # Cập nhật đơn giá tự động          self.cb\_mativi.bind("<<ComboboxSelected>>", self.HienThi\_GiaBan)          self.trHienThi.bind("<<TreeviewSelect>>", self.HienThi\_ChiTiet)      def load\_Combobox(self):          cursor = self.conn.cursor()          # Load nhân viên          if self.user == "admin":              self.dict\_nv.clear()              cursor.execute("SELECT MaNV, TenNV FROM NHANVIEN")              for ma, ten in cursor.fetchall():                  self.dict\_nv[ma] = ten              self.cb\_manhanvien["values"] = list(self.dict\_nv.keys())            else:              self.cb\_manhanvien.config(state="normal")              self.cb\_manhanvien.set(self.user)              self.cb\_manhanvien.config(state="readonly")          # Load khách hàng          cursor.execute("SELECT MaKH, TenKH FROM KHACHHANG")          self.dict\_kh.clear()          for ma, ten in cursor.fetchall():              self.dict\_kh[ma] = ten          self.cb\_makhachhang["values"] = list(self.dict\_kh.keys())          # Load tivi + đơn giá          cursor.execute("SELECT MaTivi, TenTivi, GiaBan FROM TIVI WHERE SoLuongTon > 0 ")          self.dict\_tivi.clear()          for ma, ten, gia in cursor.fetchall():              self.dict\_tivi[ma] = {"TenTivi": ten, "GiaBan": gia}          self.cb\_mativi["values"] = list(self.dict\_tivi.keys())          cursor.close()      def HienThi\_GiaBan(self, event=None):          ma\_tivi = self.cb\_mativi.get()          if ma\_tivi in self.dict\_tivi:              gia\_ban = self.dict\_tivi[ma\_tivi]["GiaBan"]              self.txt\_giaban.config(state="normal")              self.txt\_giaban.delete(0, tk.END)              self.txt\_giaban.insert(0, float(gia\_ban))              self.txt\_giaban.config(state="disabled")      def TinhTongTien(self):          tong\_tien = 0          for item in self.trHienThi.get\_children():              values = self.trHienThi.item(item, "values")              thanh\_tien = float(values[10].replace(",", ""))              tong\_tien = tong\_tien + thanh\_tien          self.lbl\_tongtien.config(text=f"{tong\_tien:,.0f} VNĐ")      def KiemTraMaHoaDonBan(self, ma\_hd):          try:              cursor = self.conn.cursor()              cursor.execute("""                      SELECT COUNT(\*)                      FROM HOADONBAN                      WHERE MaHD = ?                             """, (ma\_hd,))                count = cursor.fetchone()[0]              cursor.close()              return count > 0          except Exception as e:              messagebox.showerror("Cảnh báo", "Có lỗi xảy ra khi kiểm tra mã hóa đơn vừa nhập đã có tồn tại không:\n" + str(e))      def KiemTraMaCTHD(self, ma\_cthd):          try:              cursor = self.conn.cursor()              cursor.execute("SELECT COUNT(\*) FROM ChiTietHoaDon WHERE MaCTHD = ?", (ma\_cthd,))              count = cursor.fetchone()[0]              cursor.close()              return count > 0          except Exception as e:              messagebox.showerror("Lỗi", f"Lỗi kiểm tra MaCTHD:\n{str(e)}")              return False      def kiemtradulieu(self):          ma\_hd = self.txt\_mahoadonban.get().strip()          ma\_nv = self.cb\_manhanvien.get()          ma\_kh = self.cb\_makhachhang.get()          if not ma\_hd:              messagebox.showwarning("Cảnh báo", "Mã hóa đơn không được để trống!")              self.txt\_mahoadonban.focus()              return False            if not ma\_nv:              messagebox.showwarning("Cảnh báo", "Mã nhân viên không được trống!")              self.cb\_manhanvien.focus()              return False            if not ma\_kh:              messagebox.showwarning("Cảnh báo", "Mã khách hàng không được trống!")              self.cb\_makhachhang.focus()              return False          ma\_cthd = self.txt\_macthd.get().strip()          ma\_tivi = self.cb\_mativi.get()          so\_luong\_str = self.txt\_soluong.get().strip()            if not ma\_cthd:              messagebox.showwarning("Cảnh báo", "Mã CTHD không được để trống!")              self.txt\_macthd.focus()              return False            if not ma\_tivi:              messagebox.showwarning("Cảnh báo", "Mã tivi không được trống!")              self.cb\_mativi.focus()              return False            if not so\_luong\_str:              messagebox.showwarning("Cảnh báo", "Số lượng không được trống!")              self.txt\_soluong.focus()              return False            try:              so\_luong = int(so\_luong\_str)              if so\_luong <= 0:                  messagebox.showwarning("Cảnh báo", "Số lượng bán phải lớn hơn 0!")                  self.txt\_soluong.focus()                  return False          except ValueError:              messagebox.showwarning("Cảnh báo", "Số lượng phải là số nguyên dương!")              self.txt\_soluong.focus()              return False          return True        def ThemHoaDonChiTiet(self):          if not self.kiemtradulieu():              return # Dừng lại nếu dữ liệu không hợp lệ          ma\_hd = self.txt\_mahoadonban.get().strip()          ma\_nv = self.cb\_manhanvien.get()          ma\_kh = self.cb\_makhachhang.get()          ngay\_ban = self.dt\_ngayban.get\_date()          ma\_cthd = self.txt\_macthd.get().strip()          ma\_tivi = self.cb\_mativi.get()          so\_luong\_str = self.txt\_soluong.get().strip()          ten\_tivi = self.dict\_tivi[ma\_tivi]["TenTivi"]          ten\_nv = self.dict\_nv.get(ma\_nv,"")          ten\_kh = self.dict\_kh.get(ma\_kh, "")            if(self.KiemTraMaHoaDonBan(ma\_hd)):              messagebox.showwarning("Cảnh báo", "Mã hóa đơn đã tồn tại!")              self.txt\_mahoadonban.delete(0, tk.END)              self.txt\_mahoadonban.focus()              return            so\_luong = int(so\_luong\_str)          gia\_ban = float(self.txt\_giaban.get().replace(",", ""))          thanh\_tien = so\_luong \* gia\_ban          # === KIỂM TRA TỒN KHO TRƯỚC KHI THỰC HIỆN ===          ton\_kho\_hien\_tai = self.KiemTraTonKho(ma\_tivi)            if so\_luong <= 0:              messagebox.showwarning("Cảnh báo", "Số lượng bán phải lớn hơn 0!")              return            if so\_luong > ton\_kho\_hien\_tai:              messagebox.showwarning("Cảnh báo", f"Số lượng tồn kho của {ma\_tivi} chỉ còn {ton\_kho\_hien\_tai}!")              return            if self.KiemTraMaCTHD(ma\_cthd):              messagebox.showwarning("Cảnh báo", f"Mã CTHD '{ma\_cthd}' đã tồn tại trong hệ thống!")              return            else:              try:                  # Kiểm tra trùng mã Tivi trong cùng phiếu                  for item in self.trHienThi.get\_children():                      values = self.trHienThi.item(item, "values")                      if values[6] == ma\_tivi:                          traloi = messagebox.askyesno("Trùng sản phẩm", "Tivi " + ma\_tivi + " dã có trong đơn này.\nBan có muốn cộng dồn số lượng không?")                          if traloi:                              # Cộng dồn số lượng và cập nhật thành tiền                              soluongcu = int(values[8])                              soluongmoi = soluongcu + so\_luong                              # === KIỂM TRA TỒN KHO CHO TRƯỜNG HỢP CỘNG DỒN ===                              if soluongmoi > ton\_kho\_hien\_tai:                                  messagebox.showwarning("Cảnh báo", f"Tổng số lượng ({soluongmoi}) vượt quá tồn kho ({ton\_kho\_hien\_tai})! Không thể cộng dồn.")                                  return                                thanhtienmoi = soluongmoi \* gia\_ban                              self.trHienThi.item(item, values=(ma\_hd, ngay\_ban.strftime("%d/%m/%Y"), ma\_nv, ten\_nv, ma\_kh, ten\_kh, ma\_tivi, ten\_tivi, soluongmoi, f"{gia\_ban:,.0f}", f"{thanhtienmoi:,.0f}"))                          return                  # Thêm hóa đơn vào trHienThi                  self.trHienThi.insert("", tk.END, text=ma\_cthd, values=(ma\_hd, ngay\_ban.strftime("%d/%m/%Y"), ma\_nv, ten\_nv, ma\_kh, ten\_kh, ma\_tivi, ten\_tivi, so\_luong, f"{gia\_ban:,.0f}", f"{thanh\_tien:,.0f}"))                  # Khóa hóa đơn bán sau khi thêm                  self.txt\_mahoadonban.config(state="disabled")                  self.cb\_makhachhang.config(state="disabled")                  self.dt\_ngayban.config(state="disabled")                  self.cb\_manhanvien.config(state="disabled")                  # Xóa dữ liệu chi tiết hóa đơn sau khi thêm                  self.txt\_macthd.delete(0, tk.END)                  self.cb\_mativi.set("")                  self.txt\_soluong.delete(0, tk.END)                    self.txt\_giaban.config(state="normal")                  self.txt\_giaban.delete(0, tk.END)                  self.txt\_giaban.config(state="disabled")                    self.TinhTongTien()              except Exception as e:                  messagebox.showerror("Lỗi", "Đã xảy ra lỗi:" + str(e))      def HienThi\_ChiTiet(self, event):          selected = self.trHienThi.selection()          if not selected:              return          item = self.trHienThi.item(selected[0], "values")          if not item:              return          # Gán giá trị lên form          self.txt\_mahoadonban.config(state="normal")          self.txt\_mahoadonban.delete(0, tk.END)          self.txt\_mahoadonban.insert(0, item[0])          self.txt\_mahoadonban.config(state="disabled")          self.dt\_ngayban.set\_date(item[1])          self.cb\_manhanvien.set(item[2])          self.cb\_makhachhang.set(item[4])          self.cb\_mativi.set(item[6])          self.txt\_soluong.delete(0, tk.END)          self.txt\_soluong.insert(0, item[8])          self.txt\_giaban.config(state="normal")          self.txt\_giaban.delete(0, tk.END)          gia\_ban = float(item[9].replace(",", ""))          self.txt\_giaban.insert(0, gia\_ban)          self.txt\_giaban.config(state="disabled")          # Hiển thị MaCTHD từ dòng được chọn (lấy từ Treeview)          ma\_cthd = self.trHienThi.item(selected[0], "text")          self.txt\_macthd.delete(0, tk.END)          self.txt\_macthd.insert(0, ma\_cthd)      def SuaHoaDonChiTiet(self):          selected = self.trHienThi.selection()          if not selected:              messagebox.showwarning("Cảnh báo", "Vui lòng chọn hóa đơn để sửa!")              return            if not self.kiemtradulieu():              return            ma\_hd = self.txt\_mahoadonban.get()          ma\_nv = self.cb\_manhanvien.get()          ten\_nv = self.dict\_nv.get(ma\_nv,"")          ma\_kh = self.cb\_makhachhang.get()          ten\_kh = self.dict\_kh.get(ma\_kh, "")          ngay\_ban = self.dt\_ngayban.get\_date()          ma\_cthd = self.txt\_macthd.get().strip()          ma\_tivi = self.cb\_mativi.get()          so\_luong\_str = self.txt\_soluong.get().strip()            ten\_tivi = self.dict\_tivi[ma\_tivi]["TenTivi"]            so\_luong = int(so\_luong\_str)          gia\_ban = float(self.txt\_giaban.get().replace(",", ""))          thanh\_tien = so\_luong \* gia\_ban          ton\_kho\_hien\_tai = self.KiemTraTonKho(ma\_tivi)            if so\_luong <= 0:              messagebox.showwarning("Cảnh báo", "Số lượng bán phải lớn hơn 0!")              return            if so\_luong > ton\_kho\_hien\_tai:              messagebox.showwarning("Cảnh báo", f"Số lượng tồn kho của {ma\_tivi} chỉ còn {ton\_kho\_hien\_tai}! Không thể sửa với số lượng này.")              return          old\_ma\_cthd = self.trHienThi.item(selected[0], "text")          if ma\_cthd != old\_ma\_cthd and self.KiemTraMaCTHD(ma\_cthd):              messagebox.showwarning("Cảnh báo", f"Mã CTHD '{ma\_cthd}' đã tồn tại!")              return          try:              for item in self.trHienThi.get\_children():                  if item == selected[0]:                      continue                  values = self.trHienThi.item(item, "values")                  if values[6] == ma\_tivi:                      traloi = messagebox.askyesno("Trùng sản phẩm", "Tivi " + ma\_tivi + " dã có trong đơn này.\nBan có muốn cộng dồn số lượng không?")                      if traloi:                          # Cộng dồn số lượng và cập nhật thành tiền                          soluongcu = int(values[8])                          soluongmoi = soluongcu + so\_luong                          thanhtienmoi = soluongmoi \* gia\_ban                          if soluongmoi > ton\_kho\_hien\_tai:                              messagebox.showwarning("Cảnh báo", f"Tổng số lượng ({soluongmoi}) vượt quá tồn kho ({ton\_kho\_hien\_tai})! Không thể gộp.")                              return                            self.trHienThi.item(item, values=(ma\_hd, ngay\_ban.strftime("%d/%m/%Y"), ma\_nv, ten\_nv, ma\_kh, ten\_kh, ma\_tivi, ten\_tivi, int(soluongmoi), f"{gia\_ban:,.0f}", f"{thanhtienmoi:,.0f}"))                          self.trHienThi.item(item, text=ma\_cthd)                          # Xóa dòng đang sửa(vì đã gộp vào dòng kia)                          self.trHienThi.delete(selected[0])                          # Cập nhật tổng tiền                          self.TinhTongTien()                      return                # Cập nhật lại hóa đơn trong trHienThi              self.trHienThi.item(selected[0], values=(ma\_hd, ngay\_ban.strftime("%d/%m/%Y"), ma\_nv, ten\_nv, ma\_kh, ten\_kh, ma\_tivi, ten\_tivi, int(so\_luong), f"{gia\_ban:,.0f}", f"{thanh\_tien:,.0f}"))              self.trHienThi.item(selected[0], text=ma\_cthd)              # Xóa dữ liệu chi tiết hóa đơn sau khi thêm              self.txt\_macthd.delete(0, tk.END)              self.cb\_mativi.set("")              self.txt\_soluong.delete(0, tk.END)              self.txt\_giaban.config(state="normal")              self.txt\_giaban.delete(0, tk.END)              self.txt\_giaban.config(state="disabled")                self.TinhTongTien()          except Exception as e:              messagebox.showerror("Lỗi", "Đã xảy ra lỗi:" + str(e))        def XoaHoaDonChiTiet(self):          selected = self.trHienThi.selection()          if not selected:              messagebox.showwarning("Cảnh báo", "Vui lòng chọn hóa đơn để xóa!")              return            traloi = messagebox.askyesno("Xác nhận", "Bạn có chắc chắn muốn xóa hóa đơn này không?")          if traloi:              self.trHienThi.delete(selected[0])              # Cập nhật lại tổng tiền              self.TinhTongTien()              # Xóa input chi tiết              self.txt\_macthd.delete(0, tk.END)              self.cb\_mativi.set("")              self.txt\_soluong.delete(0, tk.END)              self.txt\_giaban.config(state="normal")              self.txt\_giaban.delete(0, tk.END)              self.txt\_giaban.config(state="disabled")      def LamMoi(self):          self.txt\_mahoadonban.config(state="normal")          self.dt\_ngayban.config(state="normal")          self.cb\_manhanvien.config(state="readonly")          self.cb\_makhachhang.config(state="readonly")          self.txt\_mahoadonban.delete(0, tk.END)          self.dt\_ngayban.set\_date(datetime.today())          if self.user == "admin":              self.cb\_manhanvien.set("")          self.cb\_makhachhang.set("")          self.txt\_macthd.delete(0, tk.END)          self.cb\_mativi.set("")          self.txt\_soluong.delete(0, tk.END)          self.txt\_soluong.insert(0, "")          self.txt\_soluong.config(state="normal")          self.txt\_giaban.delete(0, tk.END)          self.txt\_giaban.insert(0, "")          self.txt\_giaban.config(state="disabled")          self.trHienThi.delete(\*self.trHienThi.get\_children())          self.lbl\_tongtien.config(text="0 VNĐ")      def TaoHoaDon(self):          if not self.trHienThi.get\_children():              messagebox.showwarning("Cảnh báo", "Không có hóa đơn nào để thực hiện tạo hóa đơn")              return            try:              cursor = self.conn.cursor()              ma\_hd = self.txt\_mahoadonban.get()              ngay\_ban = self.dt\_ngayban.get\_date().strftime("%Y-%m-%d")              ma\_nv = self.cb\_manhanvien.get()              ma\_kh = self.cb\_makhachhang.get()              # Thêm hóa đơn - vì 1 hóa đơn sẽ có nhiều chi tiết hóa đơn              cursor.execute("""                  INSERT INTO HOADONBAN (MaHD, NgayBan, MaNV, MaKH)                  VALUES (?, ?, ?, ?)                  """, (ma\_hd, ngay\_ban, ma\_nv, ma\_kh))                for item in self.trHienThi.get\_children():                  values = self.trHienThi.item(item, "values")                  ma\_cthd = self.trHienThi.item(item, "text")                  ma\_tivi = values[6]                  so\_luong = int(values[8])                  gia\_ban = float(values[9].replace(",", ""))                    # Thêm chi tiết hóa đơn                  cursor.execute("""                      INSERT INTO CHITIETHOADON (MaCTHD, MaHD, MaTivi, SoLuong, DonGia)                      VALUES(?, ?, ?, ?, ?)                      """, (ma\_cthd, ma\_hd, ma\_tivi, so\_luong, gia\_ban))                self.conn.commit()              cursor.close()              messagebox.showinfo("Thành công", "Tạo hóa đơn thành công!")              self.LamMoi()              if self.tab\_hoadon:                  self.tab\_hoadon.load\_hoa\_don()          except Exception as e:              self.conn.rollback()              messagebox.showerror("Lỗi", "Đã xảy ra lỗi khi tạo đơn hàng:\n" + str(e))\        def KiemTraTonKho(self, ma\_tivi):          try:              cursor = self.conn.cursor()              cursor.execute("SELECT SoLuongTon FROM TIVI WHERE MaTivi = ?", (ma\_tivi,))              ton\_kho = cursor.fetchone()              cursor.close()              return ton\_kho[0] if ton\_kho else 0          except Exception as e:              print(f"Lỗi truy vấn tồn kho: {e}")              return 0 |

**FORM HÓA ĐƠN**

|  |
| --- |
| import tkinter as tk  from tkinter import ttk, messagebox, filedialog  from datetime import datetime  import pyodbc  from docx import Document  from docx.shared import Inches, Pt, RGBColor, Cm  from docx.enum.text import WD\_ALIGN\_PARAGRAPH, WD\_TAB\_ALIGNMENT, WD\_TAB\_LEADER  import frmTongQuan  # === TAB HÓA ĐƠN ===  class tabHoaDon(tk.Frame):      def \_\_init\_\_(self, parent, conn, user, controller=None):          super().\_\_init\_\_(parent, bg="white")          # === CHUỖI KẾT NỐI ===          self.conn = conn          # === LƯU MÃ NHÂN VIÊN ===          self.user = user          # === LƯU THAM CHIẾU CONTROLLER ===          self.controller = controller          # === KHUNG TÌM KIẾM ===          frame\_search = tk.Frame(self, bg="#E3F2FD", padx=10, pady=10)          frame\_search.pack(fill="x", padx=20, pady=5)          tk.Label(frame\_search, text="Tìm kiếm:", font=("Segoe UI", 10), bg="#E3F2FD").pack(side="left", padx=5)          self.txt\_timkiem = tk.Entry(frame\_search, font=("Segoe UI", 10), width=25)          self.txt\_timkiem.pack(side="left", padx=5)          self.search\_option = tk.StringVar(value="mahd")          tk.Radiobutton(frame\_search, text="Theo mã hóa đơn", variable=self.search\_option, value="mahd", bg="#E3F2FD", font=("Segoe UI", 10)).pack(side="left", padx=10)          tk.Radiobutton(frame\_search, text="Theo mã khách hàng", variable=self.search\_option, value="makh", bg="#E3F2FD", font=("Segoe UI", 10)).pack(side="left" , padx=10)          tk.Radiobutton(frame\_search, text="Theo trạng thái phiếu", variable=self.search\_option, value="trangthai", bg="#E3F2FD", font=("Segoe UI", 10)).pack(side="left", padx=10)          tk.Button(frame\_search, text="Tìm", font=("Segoe UI", 10, "bold"), command=self.TimKiem, bg="#1565C0", fg="white", bd=0, padx=10, pady=5).pack(side="left", padx=10)          tk.Button(frame\_search, text="Hủy", font=("Segoe UI", 10, "bold"), command=self.HuyTimKiem, bg="#E53935", fg="white", bd=0, padx=10, pady=5).pack(side="left", padx=10)          # === BẢNG DANH SÁCH HÓA ĐƠN ===          frame\_table = tk.Frame(self, bg="white")          frame\_table.pack(fill="both", expand=True, padx=20, pady=10)          columns = ("MaHD", "NgayBan", "MaNV", "MaKH", "ThanhTien", "TrangThai")          # --- Tạo Scrollbar ---          scroll\_y = ttk.Scrollbar(frame\_table, orient="vertical")          scroll\_x = ttk.Scrollbar(frame\_table, orient="horizontal")          self.trHienThi = ttk.Treeview( frame\_table, show="headings",  columns=columns, height=12, yscrollcommand=scroll\_y.set, xscrollcommand=scroll\_x.set)          # --- Gắn Scrollbar ---          scroll\_y.config(command=self.trHienThi.yview)          scroll\_x.config(command=self.trHienThi.xview)          # --- Bố trí Scrollbar ---          scroll\_y.pack(side="right", fill="y")          scroll\_x.pack(side="bottom", fill="x")          self.trHienThi.pack(fill="both", expand=True)          self.trHienThi.heading("MaHD", text="Mã hóa đơn")          self.trHienThi.heading("NgayBan", text="Ngày bán")          self.trHienThi.heading("MaNV", text="Mã nhân viên")          self.trHienThi.heading("MaKH", text="Mã khách hàng")          self.trHienThi.heading("ThanhTien", text="Thành tiền")          self.trHienThi.heading("TrangThai", text="Trạng thái")          self.trHienThi.column("MaHD", width=100)          self.trHienThi.column("NgayBan", width=100, anchor="center")          self.trHienThi.column("MaNV", width=100)          self.trHienThi.column("MaKH", width=100)          self.trHienThi.column("ThanhTien", width=100, anchor="center")          self.trHienThi.column("TrangThai", width=100)          # Thêm style cho Treeview          style = ttk.Style()          style.configure("Treeview.Heading", font=("Segoe UI", 11, "bold"))          style.configure("Treeview", font=("Segoe UI", 10), rowheight=28)          # ==== Nút thao tác ====          frame\_btn = tk.Frame(self, bg="white")          frame\_btn.pack(pady=10)          tk.Button(frame\_btn, text="🧐 Xem chi tiết", bg="#EC9428", fg="white", font=("Segoe UI", 10, "bold"),command=self.XemChiTiet, padx=15, pady=5, bd=0).pack(side="left", padx=5)          tk.Button(frame\_btn, text="💳 Thanh toán hóa đơn", bg="#43A047", fg="white",  font=("Segoe UI", 10, "bold"), command=self.ThanhToanHoaDonBan, padx=15, pady=5, bd=0).pack(side="left", padx=5)          tk.Button(frame\_btn, text="❌ Hủy hóa đơn", bg="#E53935", fg="white",  font=("Segoe UI", 10, "bold"), padx=15, command=self.HuyHoaDonBan, pady=5, bd=0).pack(side="left", padx=5)          tk.Button(frame\_btn, text="🔄 Làm mới", bg="#1E88E5", fg="white", font=("Segoe UI", 10, "bold"), padx=15, command=self.load\_hoa\_don, pady=5, bd=0).pack(side="left", padx=5)          tk.Button(frame\_btn, text="🗑️ Xóa", bg="#B71C1C", fg="white", font=("Segoe UI", 10, "bold"), padx=15, command=self.XoaHoaDonVinhVien, pady=5, bd=0).pack(side="left", padx=5)          tk.Button(frame\_btn, text="🖨️ In", bg="#E51E9C", fg="white", font=("Segoe UI", 10, "bold"), padx=15, command=self.InHoaDon, pady=5, bd=0).pack(side="left", padx=5)          # === TẢI DỮ LIỆU HÓA ĐƠN ===          self.load\_hoa\_don()      def load\_hoa\_don(self):          try:              self.trHienThi.delete(\*self.trHienThi.get\_children())              cursor = self.conn.cursor()              # Khởi tạo câu truy vấn cơ bản              sql\_query = "SELECT MaHD, NgayBan, MaNV, MaKH, TongTien, TrangThai FROM HOADONBAN"              params = []                # KIỂM TRA QUYỀN TRUY CẬP              if self.user.lower() != "admin":                  # Nếu không phải admin, chỉ lấy hóa đơn của nhân viên đó                  sql\_query = sql\_query + " WHERE MaNV = ?"                  params.append(self.user)                # Thực thi truy vấn              cursor.execute(sql\_query, params)              for row in cursor.fetchall():                  ngay\_ban = datetime.strptime(str(row.NgayBan).split(" ")[0], "%Y-%m-%d")                  formatted\_row = (                      row.MaHD, ngay\_ban.strftime("%d/%m/%Y"), row.MaNV, row.MaKH,                      f"{float(row.TongTien):,.0f}" if row.TongTien else "0", row.TrangThai                  )                  self.trHienThi.insert("", tk.END, values=formatted\_row)          except Exception as e:              messagebox.showerror("Lỗi", "Không thể tải dữ liệu hóa đơn:\n" + str(e))      # === Hàm xem chi tiết phiếu nhập ===      def XemChiTiet(self):          selected = self.trHienThi.selection()          if not selected:              messagebox.showwarning("Thông báo", "Vui lòng chọn 1 hóa đơn để xem chi tiết.")              return          ma\_hd = self.trHienThi.item(selected[0], "values")[0]          try:              chitiethoadon= tk.Toplevel(self)              chitiethoadon.title("Chi tiết hóa đơn: " + ma\_hd)              chitiethoadon.geometry("800x800")              chitiethoadon.resizable(False, False)              chitiethoadon.configure(bg="white")                tk.Label(chitiethoadon, text="Chi tiết hóa đơn bán " + ma\_hd, font=("Segoe UI", 12, "bold"), bg="white", fg="#0D47A1").pack(pady=10)                cursor = self.conn.cursor()              cursor.execute("""SELECT nv.MaNV, nv.TenNV, kh.MaKH, kh.TenKH, hdb.NgayBan, hdb.TongTien, hdb.TrangThai                                FROM HOADONBAN hdb JOIN NHANVIEN nv ON hdb.MaNV = nv.MaNV                                JOIN KHACHHANG kh ON hdb.MaKH = kh.MaKH                                WHERE hdb.MaHD = ?""", (ma\_hd,))              thong\_tin = cursor.fetchone()              Frame\_thongtin = tk.Frame(chitiethoadon, bg="white")              Frame\_thongtin.pack(pady=5)              tk.Label(Frame\_thongtin, text="Mã nhân viên:", font=("Segoe UI", 10), bg="white").grid(row=0, column=0, padx=10, sticky="w")              tk.Label(Frame\_thongtin, text=thong\_tin.MaNV, font=("Segoe UI", 10, "bold"), bg="white", fg="#1565C0").grid(row=0, column=1, padx=10, sticky="w")              tk.Label(Frame\_thongtin, text="Tên nhân viên:", font=("Segoe UI", 10), bg="white").grid(row=0, column=2, padx=10, sticky="w")              tk.Label(Frame\_thongtin, text=thong\_tin.TenNV, font=("Segoe UI", 10, "bold"), bg="white", fg="#1565C0").grid(row=0, column=3, padx=10, sticky="w")              tk.Label(Frame\_thongtin, text="Mã khách hàng:", font=("Segoe UI", 10), bg="white").grid(row=1, column=0, padx=10, sticky="w")              tk.Label(Frame\_thongtin, text=thong\_tin.MaKH, font=("Segoe UI", 10, "bold"), bg="white", fg="#1565C0").grid(row=1, column=1, padx=10, sticky="w")              tk.Label(Frame\_thongtin, text="Tên khách hàng:", font=("Segoe UI", 10), bg="white").grid(row=1, column=2, padx=10, sticky="w")              tk.Label(Frame\_thongtin, text=thong\_tin.TenKH, font=("Segoe UI", 10, "bold"), bg="white", fg="#1565C0").grid(row=1, column=3, padx=10, sticky="w")              tk.Label(Frame\_thongtin, text="Ngày bán:", font=("Segoe UI", 10), bg="white").grid(row=2, column=0, padx=10, sticky="w")                ngay\_nhap = datetime.strptime(str(thong\_tin.NgayBan).split(" ")[0], "%Y-%m-%d")              tk.Label(Frame\_thongtin, text=ngay\_nhap.strftime("%d/%m/%Y"), font=("Segoe UI", 10, "bold"), bg="white", fg="#43A047").grid(row=2, column=1, padx=10, sticky="w")              tk.Label(Frame\_thongtin, text="Trạng thái:", font=("Segoe UI", 10), bg="white").grid(row=2, column=2, padx=10, sticky="w")              tk.Label(Frame\_thongtin, text=thong\_tin.TrangThai, font=("Segoe UI", 10, "bold"), bg="white", fg="#43A047").grid(row=2, column=3, padx=10, sticky="w")                columns = ("MaCTHD", "MaTivi", "TenTivi", "SoLuong", "DonGia", "ThanhTien")              tree = ttk.Treeview(chitiethoadon, columns=columns, show="headings", height=10)              tree.pack(fill="both", expand=True, padx=15, pady=10)              tree.heading("MaCTHD", text="Mã CTHD")              tree.heading("MaTivi", text="Mã Tivi")              tree.heading("TenTivi", text="Tên Tivi")              tree.heading("SoLuong", text="Số Lượng")              tree.heading("DonGia", text="Đơn Giá")              tree.heading("ThanhTien", text="Thành Tiền")              tree.column("MaCTHD", width=100)              tree.column("MaTivi", width=100)              tree.column("TenTivi", width=200)              tree.column("SoLuong", width=100, anchor="center")              tree.column("DonGia", width=100, anchor="center")              tree.column("ThanhTien", width=100, anchor="center")              tk.Label(chitiethoadon, text="Tổng tiền:", font=("Segoe UI", 10, "bold"), bg="white").pack(side="left", padx=20)              tk.Label(chitiethoadon, text=f"{float(thong\_tin.TongTien):,.0f} đ", font=("Segoe UI", 10, "bold"), bg="white", fg="red").pack(side="right", padx=20)                cursor.execute("""                  SELECT cthd.MaCTHD, cthd.MaTivi, tv.TenTivi, cthd.SoLuong, cthd.DonGia, (cthd.SoLuong \* cthd.DonGia) AS ThanhTien                  FROM CHITIETHOADON cthd                  JOIN TIVI tv ON cthd.MaTivi = tv.MaTivi                  WHERE cthd.MaHD = ?              """, (ma\_hd,))              rows = cursor.fetchall()              if not rows:                  messagebox.showinfo("Thông báo", "Không có chi tiết cho phiếu" + ma\_hd)                  chitiethoadon.destroy()                  return              for r in rows:                  tree.insert("", tk.END, values=(r.MaCTHD,r.MaTivi, r.TenTivi,r.SoLuong,f"{float(r.DonGia):,.0f} đ",f"{float(r.ThanhTien):,.0f} đ"))              cursor.close()          except Exception as e:              messagebox.showerror("Lỗi", "Không thể xem chi tiết hóa đơn:\n" + str(e))      def ThanhToanHoaDonBan(self):          selected = self.trHienThi.selection()          if not selected:              messagebox.showwarning("Thông báo", "Vui lòng chọn 1 hoa đơn để duyệt.")              return          ma\_hd= self.trHienThi.item(selected[0], "values")[0]          # Bổ sung lấy MaNV và kiểm tra quyền nếu cần thiết ở đây          trang\_thai = self.trHienThi.item(selected[0], "values")[5]          if trang\_thai != "Chờ thanh toán":              messagebox.showwarning("Thông báo", "Chỉ có thể thanh toán hóa đơn chờ thanh toán.")              return            traloi = messagebox.askyesno("Xác nhận", "Bạn có chắc chắn thanh toán hóa đơn " + ma\_hd + " không?")          if traloi:              try:                  cursor = self.conn.cursor()                  # Lấy chi tiết hóa đơn VÀ TỒN KHO hiện tại                  cursor.execute("""                      SELECT cthd.MaTivi, cthd.SoLuong, tv.SoLuongTon                      FROM CHITIETHOADON cthd                      JOIN TIVI tv ON cthd.MaTivi = tv.MaTivi                      WHERE cthd.MaHD = ?                  """, (ma\_hd,))                  chitiet = cursor.fetchall()                    # KIỂM TRA TỒN KHO TRƯỚC KHI THỰC HIỆN GIAO DỊCH                  for item in chitiet:                      ma\_tivi = item.MaTivi                      so\_luong\_ban = item.SoLuong                      ton\_kho\_hien\_tai = item.SoLuongTon                        if so\_luong\_ban > ton\_kho\_hien\_tai:                          messagebox.showerror("Lỗi tồn kho", f"Không đủ tồn kho để thanh toán! Tivi {ma\_tivi} cần {so\_luong\_ban} nhưng kho chỉ còn {ton\_kho\_hien\_tai}.")                          cursor.close()                          return # Dừng toàn bộ giao dịch ngay lập tức                  # THỰC HIỆN GIAO DỊCH (Nếu kiểm tra thành công)                    # Trừ kho trước                  for item in chitiet:                      ma\_tivi = item.MaTivi                      so\_luong = item.SoLuong                        cursor.execute("""                          UPDATE TIVI                          SET SoLuongTon = SoLuongTon - ?                          WHERE MaTivi = ? """, (so\_luong, ma\_tivi))                    # Cập nhật trạng thái sau khi trừ kho thành công                  cursor.execute("""                      UPDATE HOADONBAN                      SET TrangThai = N'Đã thanh toán'                      WHERE MaHD = ?                  """, (ma\_hd,))                    self.conn.commit()                  cursor.close()                  messagebox.showinfo("Thành công", "Hóa đơn đã thanh toán thành công!")                  # Làm mới lại danh sách hóa đơn                  self.load\_hoa\_don()                except Exception as e:                  self.conn.rollback() # Hoàn tác mọi thay đổi nếu có lỗi                  messagebox.showerror("Lỗi", "Đã xảy ra lỗi khi thanh toán hóa đơn:\n" + str(e))                self.controller.load\_form("QuanLySanPham")      def HuyHoaDonBan(self):          selected = self.trHienThi.selection()          if not selected:              messagebox.showwarning("Thông báo", "Vui lòng chọn 1 hóa đơn để hủy")              return            ma\_hd = self.trHienThi.item(selected[0], "values")[0]          trang\_thai = self.trHienThi.item(selected[0], "values")[5]          if trang\_thai != "Chờ thanh toán":              messagebox.showwarning("Thông báo", "Chỉ có thể hủy hóa đơn chưa thanh toán.")              return            traloi = messagebox.askyesno("Xác nhận", "Bạn có chắc chắn muốn hủy hóa đơn " + ma\_hd + " không?")          if traloi:              try:                  cursor = self.conn.cursor()                  cursor.execute("""                      UPDATE HOADONBAN                      SET TrangThai = N'Đã hủy'                      WHERE MaHD = ?""", (ma\_hd,))                    self.conn.commit()                  cursor.close()                  messagebox.showinfo("Thành công", "Hóa đơn đã được hủy thành công!")                  self.load\_hoa\_don()              except Exception as e:                  messagebox.showerror("Lỗi", "Đã xảy ra lỗi khi hủy hóa đơn:\n" + str(e))              self.controller.load\_form("QuanLySanPham")      def TimKiem(self):          if self.txt\_timkiem.get().strip() == "":              messagebox.showwarning("Thông báo", "Vui lòng nhập từ khóa tìm kiếm.")              return            timkkiem = self.search\_option.get().strip()          self.trHienThi.delete(\*self.trHienThi.get\_children())          try:              cursor = self.conn.cursor()              if timkkiem == "mahd":                  keyword = self.txt\_timkiem.get()                  cursor.execute("""                      SELECT MaHD, NgayBan, MaNV, MaKH, ThanhTien, TrangThai                      FROM HOADONBAN                      WHERE MaHD = ?""", (keyword,))                elif timkkiem == "makh":                  keyword = self.txt\_timkiem.get()                  cursor.execute("""                      SELECT MaHD, NgayBan, MaNV, MaKH, ThanhTien, TrangThai                      FROM HOADONBAN                      WHERE MaKH = ?""", (keyword,))                elif timkkiem == "trangthai":                  keyword = self.txt\_timkiem.get()                  cursor.execute("""                      SELECT MaHD, NgayBan, MaNV, MaKH, ThanhTien, TrangThai                      FROM HOADONBAN                      WHERE TrangThai = ?""", (keyword,))                rows = cursor.fetchall()              for row in rows:                  ngay\_ban = datetime.strptime(str(row.NgayBan).split(" ")[0], "%Y-%m-%d")                  formatted\_row = (                      row.MaHD, ngay\_ban.strftime("%d/%m/%y"), row.MaNV, row.MaKH,                      f"{float(row.ThanhTien):,.0f}" if row.ThanhTien else "0", row.TrangThai                  )                  self.trHienThi.insert("", tk.END, values=formatted\_row)              cursor.close()          except Exception as e:              messagebox.showerror("Lỗi", "Không thể tìm kiếm hóa đơn:\n" + str(e))      def HuyTimKiem(self):          self.load\_hoa\_don()          self.txt\_timkiem.delete(0, tk.END)      # === HÀM IN HÓA ĐƠN RA WORD ===      def InHoaDon(self):          selected = self.trHienThi.selection()          if not selected:              messagebox.showwarning("Thông báo", "Vui lòng chọn 1 hóa đơn để in.")              return          ma\_hd = self.trHienThi.item(selected[0], "values")[0]            # --- Lấy dữ liệu chi tiết ---          try:              cursor = self.conn.cursor()                # 1. Lấy thông tin chung của hóa đơn              cursor.execute("""SELECT hdb.MaHD, nv.MaNV, nv.TenNV, kh.MaKH, kh.TenKH, hdb.NgayBan, hdb.ThanhTien, hdb.TrangThai                                FROM HOADONBAN hdb                                JOIN NHANVIEN nv ON hdb.MaNV = nv.MaNV                                JOIN KHACHHANG kh ON hdb.MaKH = kh.MaKH                                WHERE hdb.MaHD = ?""", (ma\_hd,))              thong\_tin = cursor.fetchone()              if not thong\_tin:                  messagebox.showerror("Lỗi", f"Không tìm thấy thông tin hóa đơn {ma\_hd}")                  return              # 2. Lấy chi tiết các mặt hàng              cursor.execute("""                  SELECT cthd.MaCTHD, cthd.MaTivi, tv.TenTivi, cthd.SoLuong, cthd.DonGia, (cthd.SoLuong \* cthd.DonGia) AS ThanhTien                  FROM CHITIETHOADON cthd                  JOIN TIVI tv ON cthd.MaTivi = tv.MaTivi                  WHERE cthd.MaHD = ?              """, (ma\_hd,))              chi\_tiet\_hang = cursor.fetchall()              cursor.close()              if not chi\_tiet\_hang:                  messagebox.showinfo("Thông báo", f"Hóa đơn {ma\_hd} không có chi tiết hàng hóa.")                  return          except Exception as e:              messagebox.showerror("Lỗi CSDL", "Không thể lấy dữ liệu để in hóa đơn:\n" + str(e))              return            # --- Tạo File Word ---          try:              document = Document()                # Đặt font và size mặc định cho document              style = document.styles['Normal']              style.font.name = 'Times New Roman'              style.font.size = Pt(11)              # Dữ liệu định dạng              ngay\_ban\_date = datetime.strptime(str(thong\_tin.NgayBan).split(" ")[0], "%Y-%m-%d")              tong\_tien\_str = f"{float(thong\_tin.ThanhTien):,.0f} đ"                # Tiêu đề chính (Căn giữa)              heading = document.add\_paragraph()              heading.alignment = WD\_ALIGN\_PARAGRAPH.CENTER              run = heading.add\_run(f"HÓA ĐƠN BÁN HÀNG SỐ: {ma\_hd}")              run.bold = True              run.font.size = Pt(16)              document.add\_paragraph() # Dòng trống                # --- Bố trí Thông tin chung (Dùng Tab Stop cố định) ---                # Thiết lập Tab Stop: Căn lề trái tại vị trí 8.0 cm              TAB\_STOP\_POS = Cm(8.0)                # Thông tin Nhân viên              p1 = document.add\_paragraph()              p1.paragraph\_format.tab\_stops.add\_tab\_stop(TAB\_STOP\_POS, WD\_TAB\_ALIGNMENT.LEFT, WD\_TAB\_LEADER.SPACES)              p1.add\_run("Mã nhân viên: ").bold = False              run = p1.add\_run(thong\_tin.MaNV)              run.bold = True              run.font.color.rgb = RGBColor(0x15, 0x65, 0xC0)              p1.add\_run('\t')              p1.add\_run("Tên nhân viên: ").bold = False              run = p1.add\_run(thong\_tin.TenNV)              run.bold = True              run.font.color.rgb = RGBColor(0x15, 0x65, 0xC0)              # Thông tin Khách hàng              p2 = document.add\_paragraph()              p2.paragraph\_format.tab\_stops.add\_tab\_stop(TAB\_STOP\_POS, WD\_TAB\_ALIGNMENT.LEFT, WD\_TAB\_LEADER.SPACES)              p2.add\_run("Mã khách hàng: ").bold = False              run = p2.add\_run(thong\_tin.MaKH)              run.bold = True              run.font.color.rgb = RGBColor(0x15, 0x65, 0xC0)              p2.add\_run('\t')              p2.add\_run("Tên khách hàng: ").bold = False              run = p2.add\_run(thong\_tin.TenKH)              run.bold = True              run.font.color.rgb = RGBColor(0x15, 0x65, 0xC0)              # Thông tin Ngày bán / Trạng thái              p3 = document.add\_paragraph()              p3.paragraph\_format.tab\_stops.add\_tab\_stop(TAB\_STOP\_POS, WD\_TAB\_ALIGNMENT.LEFT, WD\_TAB\_LEADER.SPACES)              p3.add\_run("Ngày bán: ").bold = False              run = p3.add\_run(ngay\_ban\_date.strftime("%d/%m/%Y"))              run.bold = True              run.font.color.rgb = RGBColor(0x43, 0xA0, 0x47)              p3.add\_run('\t')              p3.add\_run("Trạng thái: ").bold = False              run = p3.add\_run(thong\_tin.TrangThai)              run.bold = True              run.font.color.rgb = RGBColor(0x43, 0xA0, 0x47)                document.add\_paragraph() # Dòng trống              # --- Bảng chi tiết hàng hóa ---              table = document.add\_table(rows=1, cols=6)              table.style = 'Table Grid'                hdr\_cells = table.rows[0].cells              headers = ["Mã CTHD", "Mã Tivi", "Tên Tivi", "Số Lượng", "Đơn Giá", "Thành Tiền"]              for i, header in enumerate(headers):                  hdr\_cells[i].text = header                  hdr\_cells[i].paragraphs[0].runs[0].bold = True                  hdr\_cells[i].paragraphs[0].alignment = WD\_ALIGN\_PARAGRAPH.CENTER                # Ghi dữ liệu chi tiết              for item in chi\_tiet\_hang:                  row\_cells = table.add\_row().cells                  row\_cells[0].text = item.MaCTHD                  row\_cells[1].text = item.MaTivi                  row\_cells[2].text = item.TenTivi                  row\_cells[3].text = str(item.SoLuong)                  row\_cells[4].text = f"{float(item.DonGia):,.0f} đ"                  row\_cells[5].text = f"{float(item.ThanhTien):,.0f} đ"                    # Căn giữa Số Lượng, Đơn Giá, Thành Tiền                  for i in [3, 4, 5]:                      row\_cells[i].paragraphs[0].alignment = WD\_ALIGN\_PARAGRAPH.CENTER              # --- Tổng tiền (Căn phải) ---              document.add\_paragraph()              tong\_tien\_para = document.add\_paragraph()              tong\_tien\_para.alignment = WD\_ALIGN\_PARAGRAPH.RIGHT                tong\_tien\_para.add\_run("Tổng tiền: ").bold = True              run = tong\_tien\_para.add\_run(tong\_tien\_str)              run.bold = True              run.font.size = Pt(12)              run.font.color.rgb = RGBColor(0xFF, 0x00, 0x00)              # --- KHU VỰC KÝ TÊN ---              document.add\_paragraph()              document.add\_paragraph()                # Tạo bảng 3 cột cho khu vực chữ ký              signature\_table = document.add\_table(rows=2, cols=3)              signature\_table.autofit = False              signature\_table.style = None              # Hàng 1: Chức danh              chuc\_danh = ["Người lập hóa đơn", "Kế toán bán hàng", "Khách hàng"]              for i, cd in enumerate(chuc\_danh):                  cell = signature\_table.rows[0].cells[i]                  cell.text = cd                  cell.paragraphs[0].alignment = WD\_ALIGN\_PARAGRAPH.CENTER                  cell.paragraphs[0].runs[0].bold = True                # Hàng 2: (Ghi rõ họ tên)              for i in range(3):                  cell = signature\_table.rows[1].cells[i]                  # Thêm tên nhân viên lập phiếu vào ô đầu tiên                  ten\_nguoi\_lap = thong\_tin.TenNV if i == 0 else " "                  ten\_khach\_hang = thong\_tin.TenKH if i == 2 else " "                    if i == 0:                       cell.text = f"(Ký, ghi rõ họ tên) \n{ten\_nguoi\_lap}"                  elif i == 2:                       cell.text = f"(Ký, ghi rõ họ tên) \n{ten\_khach\_hang}"                  else:                       cell.text = "(Ký, ghi rõ họ tên)"                    cell.paragraphs[0].alignment = WD\_ALIGN\_PARAGRAPH.CENTER              # Lưu file              default\_filename = f"HoaDonBanHang\_{ma\_hd}.docx"              file\_path = filedialog.asksaveasfilename(                  defaultextension=".docx",                  initialfile=default\_filename,                  filetypes=[("Word documents", "\*.docx"), ("All files", "\*.\*")]              )              if file\_path:                  document.save(file\_path)                  messagebox.showinfo("Thành công", f"Đã in hóa đơn {ma\_hd} ra file Word thành công tại:\n{file\_path}")          except Exception as e:              messagebox.showerror("Lỗi File Word", "Không thể tạo hoặc lưu file Word:\n" + str(e))      def XoaHoaDonVinhVien(self):          selected = self.trHienThi.selection()          if not selected:              messagebox.showwarning("Thông báo", "Vui lòng chọn 1 hóa đơn để xóa.")              return          ma\_hd = self.trHienThi.item(selected[0], "values")[0]          trang\_thai = self.trHienThi.item(selected[0], "values")[5]            # LẤY NGÀY BÁN CỦA HÓA ĐƠN TỪ CSDL          try:              cursor\_temp = self.conn.cursor()              cursor\_temp.execute("SELECT NgayBan FROM HOADONBAN WHERE MaHD = ?", (ma\_hd,))              ngay\_ban\_db = cursor\_temp.fetchone()[0]              cursor\_temp.close()                # Tính toán xem hóa đơn đã tồn tại trên 5 năm chưa              ngay\_gioi\_han = datetime.now().date().replace(year=datetime.now().year - 5)              # Kiểm tra: NgayBan có nhỏ hơn Ngày giới hạn 5 năm KHÔNG?              da\_hon\_5\_nam = ngay\_ban\_db < ngay\_gioi\_han            except Exception as e:              messagebox.showerror("Lỗi", f"Không thể kiểm tra ngày bán của hóa đơn:\n{str(e)}")              return            is\_allowed\_to\_delete = False          warning\_message = ""            if trang\_thai == "Đã hủy":              is\_allowed\_to\_delete = True              warning\_message = f"Bạn có chắc chắn muốn xóa vĩnh viễn hóa đơn ĐÃ HỦY {ma\_hd} không?"            elif trang\_thai == "Đã thanh toán" and da\_hon\_5\_nam:              is\_allowed\_to\_delete = True              warning\_message = f"Hóa đơn ĐÃ THANH TOÁN {ma\_hd} này đã được lập hơn 5 năm ({ngay\_ban\_db.strftime('%d/%m/%Y')}). Bạn có chắc chắn muốn XÓA VĨNH VIỄN không?"            if not is\_allowed\_to\_delete:              messagebox.showwarning("Cảnh báo", "Chỉ có thể xóa vĩnh viễn hóa đơn ĐÃ HỦY, hoặc hóa đơn ĐÃ THANH TOÁN có thời gian lưu trữ trên 5 năm.")              return          traloi = messagebox.askyesno("Xác nhận XÓA VĨNH VIỄN", warning\_message)            if traloi:              try:                  cursor = self.conn.cursor()                    # Xóa các bản ghi liên quan (Bảo Hành)                  cursor.execute("SELECT MaCTHD FROM ChiTietHoaDon WHERE MaHD = ?", (ma\_hd,))                  cthd\_list = [row[0] for row in cursor.fetchall()]                  if cthd\_list:                      placeholders = ', '.join(['?'] \* len(cthd\_list))                      cursor.execute(f"DELETE FROM BaoHanh WHERE MaCTHD IN ({placeholders})", cthd\_list)                  # Xóa Chi Tiết Hóa Đơn                  cursor.execute("DELETE FROM ChiTietHoaDon WHERE MaHD = ?", (ma\_hd,))                    # Xóa Hóa Đơn Bán                  cursor.execute("DELETE FROM HoaDonBan WHERE MaHD = ?", (ma\_hd,))                    self.conn.commit()                  cursor.close()                  messagebox.showinfo("Thành công", f"Hóa đơn {ma\_hd} đã được xóa vĩnh viễn khỏi hệ thống!")                    self.load\_hoa\_don()                except Exception as e:                  self.conn.rollback()                  messagebox.showerror("Lỗi", "Đã xảy ra lỗi khi xóa hóa đơn:\n" + str(e)) |

**FORM NHẬP HÀNG VÀ PHIẾU NHẬP**

|  |
| --- |
| import tkinter as tk  from tkinter import ttk, messagebox  import tabNhapHang as nh  import tabPhieuNhapHang as pnh  import pyodbc  # === Tạo class Bán Hàng và Hóa Đơn  class NhapHangVaPhieuNhap(tk.Frame):      def \_\_init\_\_(self, parent, controller, conn, user):          super().\_\_init\_\_(parent, bg="white")          self.conn = conn          self.controller = controller          # === TITLE ===          lbl\_title = tk.Label(              self,              text="NHẬP HÀNG & PHIẾU NHẬP",              font=("Segoe UI", 16, "bold"),              bg="white",              fg="#0D47A1",          )          lbl\_title.pack()          # === Tạo tab control ===          tab\_control = ttk.Notebook(self)          tab\_control.pack(fill="both", expand=True, padx=20, pady=10)          self.tab\_phieunhap = pnh.tabPhieuNhapHang(tab\_control, conn, self.controller)          self.tab\_nhaphang = nh.tabNhapHang(tab\_control, conn, self.tab\_phieunhap)          tab\_control.add(self.tab\_nhaphang, text="📦 Nhập hàng")          tab\_control.add(self.tab\_phieunhap, text="🧾 Danh sách Phiếu nhập")      # Hàm làm mới tab khi click vào      def load\_data(self):          try:              self.tab\_phieunhap.load\_phieu\_nhap()              self.tab\_nhaphang.load\_combobox()          except Exception as e:              messagebox.showerror("Lỗi khi làm mới dữ liệu: " + str(e)) |

**FORM NHẬP HÀNG**

|  |
| --- |
| import tkinter as tk  from tkinter import ttk, messagebox  import pyodbc  from tkcalendar import DateEntry  from datetime import datetime  class tabNhapHang(tk.Frame):      def \_\_init\_\_(self, parent, conn, tab\_phieunhap=None):          super().\_\_init\_\_(parent, bg="white")          # === CHUỖI KẾT NỐI ===          self.conn = conn          # Khai báo trống cho các dict          self.dict\_nv = {}          self.dict\_ncc = {}          self.dict\_tivi = {}          self.tab\_phieunhap = tab\_phieunhap          # === KHUNG 1 – THÔNG TIN PHIẾU NHẬP ===          frame\_phieu = tk.LabelFrame(self, text="Thông tin Phiếu nhập", bg="white", font=("Segoe UI", 12, "bold"), fg="#0D47A1", padx=10, pady=10)          frame\_phieu.pack(fill="x", padx=20, pady=10)          tk.Label(frame\_phieu, text="Mã phiếu:", bg="white", font=("Segoe UI", 10)).grid(row=0, column=0, sticky="w", padx=5, pady=5)          self.txt\_maphieunhap = ttk.Entry(frame\_phieu, width=46)          self.txt\_maphieunhap.grid(row=0, column=1, padx=5, pady=5)          tk.Label(frame\_phieu, text="Ngày nhập:", bg="white", font=("Segoe UI", 10)).grid(row=0, column=2, sticky="w", padx=5, pady=5)          self.dt\_ngaynhap = DateEntry(frame\_phieu, width=44, date\_pattern="dd/mm/yyyy", state="readonly")          self.dt\_ngaynhap.grid(row=0, column=3, padx=5, pady=5)          tk.Label(frame\_phieu, text="Mã nhân viên nhập:", bg="white", font=("Segoe UI", 10)).grid(row=1, column=0, sticky="w", padx=5, pady=5)          self.cb\_manhanvien = ttk.Combobox(frame\_phieu, width=44, state="readonly")          self.cb\_manhanvien.grid(row=1, column=1, padx=5, pady=5)          tk.Label(frame\_phieu, text="Mã nhà cung cấp:", bg="white", font=("Segoe UI", 10)).grid(row=1, column=2, sticky="w", padx=5, pady=5)          self.cb\_manhacungcap = ttk.Combobox(frame\_phieu, width=44, state="readonly")          self.cb\_manhacungcap.grid(row=1, column=3, padx=5, pady=5)          # === KHUNG 2 – SẢN PHẨM NHẬP ===          frame\_sanpham = tk.LabelFrame(self, text="Sản phẩm trong Phiếu nhập", bg="white", font=("Segoe UI", 12, "bold"), fg="#0D47A1", padx=10, pady=10)          frame\_sanpham.pack(fill="x", padx=20, pady=10)          tk.Label(frame\_sanpham, text="Mã Tivi:", bg="white", font=("Segoe UI", 10)).grid(row=0, column=0, sticky="w", padx=5, pady=5)          self.cb\_mativi = ttk.Combobox(frame\_sanpham, width=33, state="readonly")          self.cb\_mativi.grid(row=0, column=1, padx=5, pady=5)          tk.Label(frame\_sanpham, text="Số lượng:", bg="white", font=("Segoe UI", 10)).grid(row=0, column=2, sticky="w", padx=5, pady=5)          self.txt\_soluong = ttk.Entry(frame\_sanpham, width=33)          self.txt\_soluong.grid(row=0, column=3, padx=5, pady=5)          tk.Label(frame\_sanpham, text="Giá nhập:", bg="white", font=("Segoe UI", 10)).grid(row=0, column=4, sticky="w", padx=5, pady=5)          self.txt\_gianhap = ttk.Entry(frame\_sanpham, width=33)          self.txt\_gianhap.grid(row=0, column=5, padx=5, pady=5)          # === NÚT CHỨC NĂNG ===          frame\_buttons = tk.Frame(self, bg="white")          frame\_buttons.pack(pady=10)          tk.Button(frame\_buttons, text="➕ Thêm phiếu", bg="#EBDA42", fg="white", font=("Segoe UI", 11, "bold"), command=self.ThemPhieuNhapChiTiet, padx=20, pady=5, bd=0).grid(row=0, column=0, padx=10)          tk.Button(frame\_buttons, text="✏️ Sửa", bg="#FB8C00", fg="white", font=("Segoe UI", 11, "bold"), command=self.SuaPhieuNhapChiTiet, padx=20, pady=5, bd=0).grid(row=0, column=1, padx=10)          tk.Button(frame\_buttons, text="🗑️ Xóa", bg="#E53935", fg="white", font=("Segoe UI", 11, "bold"), command=self.XoaPhieuNhapChiTiet,padx=20, pady=5, bd=0).grid(row=0, column=2, padx=10)          tk.Button(frame\_buttons, text="🔄 Làm mới", bg="#1E88E5", fg="white", font=("Segoe UI", 11, "bold"), command=self.LamMoi, padx=20, pady=5, bd=0).grid(row=0, column=3, padx=10)          # === BẢNG DANH SÁCH SẢN PHẨM ===          frame\_table = tk.Frame(self, bg="white")          frame\_table.pack(fill="both", expand=True, padx=20, pady=10)          columns = ("MaPhieu", "NgayNhap", "MaNV", "TenNV", "MaNCC", "TenNCC", "MaTivi", "TenTivi", "SoLuong", "GiaNhap", "ThanhTien")          # --- Tạo Scrollbar ---          scroll\_y = ttk.Scrollbar(frame\_table, orient="vertical")          scroll\_x = ttk.Scrollbar(frame\_table, orient="horizontal")          self.trHienThi = ttk.Treeview( frame\_table, show="headings",  columns=columns, height=12, yscrollcommand=scroll\_y.set, xscrollcommand=scroll\_x.set)          # --- Gắn Scrollbar ---          scroll\_y.config(command=self.trHienThi.yview)          scroll\_x.config(command=self.trHienThi.xview)          # --- Bố trí Scrollbar ---          scroll\_y.pack(side="right", fill="y")          scroll\_x.pack(side="bottom", fill="x")          self.trHienThi.pack(fill="both", expand=True)          self.trHienThi.heading("MaPhieu", text="Mã phiếu")          self.trHienThi.heading("NgayNhap", text="Ngày nhập")          self.trHienThi.heading("MaNV", text="Mã nhân viên")          self.trHienThi.heading("TenNV", text="Tên nhân viên")          self.trHienThi.heading("MaNCC", text="Mã nhà cung cấp")          self.trHienThi.heading("TenNCC", text="Tên nhà cung cấp")          self.trHienThi.heading("MaTivi", text="Mã tivi")          self.trHienThi.heading("TenTivi", text="Tên tivi")          self.trHienThi.heading("SoLuong", text="Số lượng")          self.trHienThi.heading("GiaNhap", text="Giá nhập")          self.trHienThi.heading("ThanhTien", text="Thành tiền")          self.trHienThi.column("MaPhieu", anchor="center", width=150)          self.trHienThi.column("NgayNhap", anchor="center", width=150)          self.trHienThi.column("MaNV", anchor="center", width=150)          self.trHienThi.column("MaNCC", anchor="center", width=200)          self.trHienThi.column("MaTivi", anchor="center", width=150)          self.trHienThi.column("SoLuong", anchor="center", width=120)          self.trHienThi.column("TenNV", anchor="w", width=150)          self.trHienThi.column("TenNCC", anchor="w", width=240)          self.trHienThi.column("TenTivi", anchor="w", width=200)          self.trHienThi.column("GiaNhap", anchor="e", width=150)          self.trHienThi.column("ThanhTien", anchor="e", width=150)            # Style Treeview          style = ttk.Style()          style.configure("Treeview.Heading", font=("Segoe UI", 11, "bold"))          style.configure("Treeview", font=("Segoe UI", 10), rowheight=28)          # ==== TỔNG TIỀN VÀ NÚT NHẬP HÀNG ====          frame\_bottom = tk.Frame(self, bg="white")          frame\_bottom.pack(fill="x", padx=20, pady=10)          tk.Label(frame\_bottom, text="Tổng tiền:", bg="white", font=("Segoe UI", 11, "bold")).pack(side="left", padx=5)          self.lbl\_tongtien = tk.Label(frame\_bottom, text="0 VNĐ", bg="white", font=("Segoe UI", 11, "bold"), fg="#E53935")          self.lbl\_tongtien.pack(side="left", padx=5)          tk.Button(frame\_bottom, text="📦 Nhập hàng", bg="#43A047", fg="white",font=("Segoe UI", 11, "bold"), command=self.TaoPhieuNhap, padx=15, pady=5, bd=0).pack(side="right", padx=5)          self.trHienThi.bind("<<TreeviewSelect>>", self.HienThi\_ChiTiet)            self.load\_combobox()        def load\_combobox(self):          cursor = self.conn.cursor()          # Load nhân viên          self.dict\_nv.clear()          cursor.execute("SELECT MaNV, TenNV FROM NHANVIEN")          for ma, ten in cursor.fetchall():              self.dict\_nv[ma] = ten          self.cb\_manhanvien["values"] = list(self.dict\_nv.keys())          # Load nhà cung cấp          self.dict\_ncc.clear()          cursor.execute("SELECT MaNCC, TenNCC FROM NHACUNGCAP")          for ma, ten in cursor.fetchall():              self.dict\_ncc[ma] = ten          self.cb\_manhacungcap["values"] = list(self.dict\_ncc.keys())          # Load tivi          self.dict\_tivi.clear()          cursor.execute("SELECT MaTiVi, TenTiVi FROM TIVI")          for ma, ten in cursor.fetchall():              self.dict\_tivi[ma] = ten          self.cb\_mativi["values"] = list(self.dict\_tivi.keys())          cursor.close()      def TinhTongTien(self):          tong\_tien = 0          for item in self.trHienThi.get\_children():              values = self.trHienThi.item(item, "values")              thanh\_tien = float(values[10].replace(",", ""))              tong\_tien = tong\_tien +  thanh\_tien          self.lbl\_tongtien.config(text=f"{tong\_tien:,.0f} VNĐ")      def KiemTraMaPhieuNhapHang(self, ma\_phieu):          try:              cursor = self.conn.cursor()              cursor.execute("""                      SELECT COUNT(\*)                      FROM PHIEUNHAPHANG                      WHERE MaPhieuNhap = ?                             """, (ma\_phieu,))                count = cursor.fetchone()[0]              cursor.close()              return count > 0          except Exception as e:              messagebox.showerror("Cảnh báo", "Có lỗi xảy ra khi kiểm tra mã phiếu nhập hàng vừa nhập đã có tồn tại không:\n" + str(e))      def kiemtradulieu(self):          ma\_phieu = self.txt\_maphieunhap.get()          ma\_nv = self.cb\_manhanvien.get()          ma\_ncc = self.cb\_manhacungcap.get()          if not ma\_phieu:              messagebox.showwarning("Cảnh báo", "Mã phiếu không được để trống!")              self.txt\_maphieunhap.focus()              return False            if not ma\_nv:              messagebox.showwarning("Cảnh báo", "Mã nhân viên nhập không được để trống!")              self.cb\_manhanvien.focus()              return False            if not ma\_ncc:              messagebox.showwarning("Cảnh báo", "Mã nhà cung cấp không được để trống!")              self.cb\_manhacungcap.focus()              return False          ma\_tivi = self.cb\_mativi.get()          so\_luong\_str = self.txt\_soluong.get().strip()          gia\_nhap\_str = self.txt\_gianhap.get().strip()            if not ma\_tivi:              messagebox.showwarning("Cảnh báo", "Mã tivi không được trống!")              self.cb\_mativi.focus()              return False            if not so\_luong\_str:              messagebox.showwarning("Cảnh báo", "Số lượng không được trống!")              self.txt\_soluong.focus()              return False            try:              so\_luong = int(so\_luong\_str)              if so\_luong <= 0:                  messagebox.showwarning("Cảnh báo", "Số lượng nhập phải lớn hơn 0!")                  self.txt\_soluong.focus()                  return False          except ValueError:              messagebox.showwarning("Cảnh báo", "Số lượng phải là số nguyên dương!")              self.txt\_soluong.focus()              return False          if not gia\_nhap\_str:              messagebox.showwarning("Cảnh báo", "Giá nhập không được trống!")              self.txt\_gianhap.focus()              return False          try:              gia\_nhap = float(gia\_nhap\_str)              if gia\_nhap < 0:                  messagebox.showwarning("Cảnh báo", "Giá nhập phải lớn hơn hoặc bằng 0!")                  self.txt\_gianhap.focus()                  return False          except ValueError:              messagebox.showwarning("Cảnh báo", "Giá nhập phải là một con số!")              self.txt\_gianhap.focus()              return False          return True        def ThemPhieuNhapChiTiet(self):          if not self.kiemtradulieu():              return          ma\_phieu = self.txt\_maphieunhap.get()          ngay\_nhap = self.dt\_ngaynhap.get\_date()          ma\_nv = self.cb\_manhanvien.get()          ten\_nv = self.dict\_nv.get(ma\_nv, "")          ma\_ncc = self.cb\_manhacungcap.get()          ten\_ncc = self.dict\_ncc.get(ma\_ncc, "")          ma\_tivi = self.cb\_mativi.get()          ten\_tivi = self.dict\_tivi.get(ma\_tivi, "")            so\_luong = int(self.txt\_soluong.get())          gia\_nhap = float(self.txt\_gianhap.get())          thanh\_tien = so\_luong \* gia\_nhap          if(self.KiemTraMaPhieuNhapHang(ma\_phieu)):              messagebox.showwarning("Cảnh báo", "Mã phiếu nhập hàng đã tồn tại!")              self.txt\_maphieunhap.delete(0, tk.END)              self.txt\_maphieunhap.focus()              return            else:              try:                  # Kiểm tra trùng mã Tivi trong cùng phiếu                  for item in self.trHienThi.get\_children():                      values = self.trHienThi.item(item, "values")                      if values[6] == ma\_tivi:                          traloi = messagebox.askyesno("Trùng sản phẩm", "Tivi" + ma\_tivi+ " đã có trong phiếu này.\nBạn có muốn cộng dồn số lượng không?")                          if traloi:                              # Cộng dồn số lượng và cập nhật thành tiền                              soluongcu = int(values[8])                              soluongmoi = soluongcu + so\_luong                              thanhtienmoi = soluongmoi \* gia\_nhap                              self.trHienThi.item(item, values=( ma\_phieu, ngay\_nhap.strftime("%d/%m/%Y"), ma\_nv, ten\_nv, ma\_ncc, ten\_ncc, ma\_tivi, ten\_tivi, soluongmoi, f"{gia\_nhap:,.0f}", f"{thanhtienmoi:,.0f}"))                          return                  # Thêm phiếu nhập vào trHienThi                  self.trHienThi.insert("", tk.END, values=(ma\_phieu, ngay\_nhap.strftime("%d/%m/%Y"), ma\_nv, ten\_nv, ma\_ncc, ten\_ncc, ma\_tivi, ten\_tivi, so\_luong, f"{gia\_nhap:,.0f}", f"{thanh\_tien:,.0f}"))                  # Khóa phiếu nhập sau khi thêm                  self.txt\_maphieunhap.config(state="disabled")                  self.dt\_ngaynhap.config(state="disabled")                  self.cb\_manhanvien.config(state="disabled")                  self.cb\_manhacungcap.config(state="disabled")                  # Xóa dữ liệu chi tiết phiếu nhập sau khi thêm                  self.cb\_mativi.set("")                  self.txt\_soluong.delete(0, tk.END)                  self.txt\_gianhap.delete(0, tk.END)                  # Cập nhật tổng tiền                  self.TinhTongTien()              except Exception as e:                  messagebox.showerror("Lỗi", "Đã xảy ra lỗi:" + str(e))        def HienThi\_ChiTiet(self, event):          selected = self.trHienThi.selection()          if not selected:              return          item = self.trHienThi.item(selected[0], "values")          if not item:              return          # Gán giá trị lên form          self.txt\_maphieunhap.delete(0, tk.END)          self.txt\_maphieunhap.insert(0, item[0])          self.dt\_ngaynhap.set\_date(item[1])          self.cb\_manhanvien.set(item[2])          self.cb\_manhacungcap.set(item[4])          self.cb\_mativi.set(item[6])          self.txt\_soluong.delete(0, tk.END)          self.txt\_soluong.insert(0, item[8])          self.txt\_gianhap.delete(0, tk.END)          gia\_nhap = float(item[9].replace(",", ""))          self.txt\_gianhap.insert(0,  gia\_nhap)      def SuaPhieuNhapChiTiet(self):          selected = self.trHienThi.selection()          if not selected:              messagebox.showwarning("Cảnh báo", "Vui lòng chọn phiếu nhập để sửa!")              return            if not self.kiemtradulieu():              return            ma\_phieu = self.txt\_maphieunhap.get()          ngay\_nhap = self.dt\_ngaynhap.get\_date()          ma\_nv = self.cb\_manhanvien.get()          ten\_nv = self.dict\_nv.get(ma\_nv, "")          ma\_ncc = self.cb\_manhacungcap.get()          ten\_ncc = self.dict\_ncc.get(ma\_ncc, "")          ma\_tivi = self.cb\_mativi.get()          ten\_tivi = self.dict\_tivi.get(ma\_tivi, "")          so\_luong = int(self.txt\_soluong.get())          gia\_nhap = float(self.txt\_gianhap.get())          thanh\_tien = so\_luong \* gia\_nhap            try:              # Kiểm tra trùng mã Tivi trong cùng phiếu (trừ dòng hiện tại)              for item in self.trHienThi.get\_children():                  if item == selected[0]:                      continue                  values = self.trHienThi.item(item, "values")                  if  values[6] == ma\_tivi:                      traloi = messagebox.askyesno("Trùng sản phẩm", "Tivi" + ma\_tivi+ " đã có trong phiếu này.\nBạn có muốn cộng dồn số lượng không?")                      if traloi:                          # Cộng dồn số lượng và cập nhật thành tiền                          soluongcu = int(values[8])                          soluongmoi = soluongcu + so\_luong                          thanhtienmoi = soluongmoi \* gia\_nhap                          self.trHienThi.item(item, values=( ma\_phieu, ngay\_nhap.strftime("%d/%m/%Y"), ma\_nv, ten\_nv, ma\_ncc, ten\_ncc, ma\_tivi, ten\_tivi, soluongmoi, f"{gia\_nhap:,.0f}", f"{thanhtienmoi:,.0f}"))                          # Xóa dòng đang sửa (vì đã gộp vào dòng kia)                          self.trHienThi.delete(selected[0])                          # Cập nhật tổng tiền                          self.TinhTongTien()                      return              # Cập nhật phiếu nhập trong trHienThi              self.trHienThi.item(selected[0], values=(ma\_phieu, ngay\_nhap.strftime("%d/%m/%Y"), ma\_nv, ten\_nv, ma\_ncc, ten\_ncc, ma\_tivi, ten\_tivi, so\_luong, f"{gia\_nhap:,.0f}", f"{thanh\_tien:,.0f}"))              # Xóa dữ liệu chi tiết phiếu nhập sau khi thêm              self.cb\_mativi.set("")              self.txt\_soluong.delete(0, tk.END)              self.txt\_gianhap.delete(0, tk.END)              # Cập nhật tổng tiền              self.TinhTongTien()          except Exception as e:              messagebox.showerror("Lỗi", "Đã xảy ra lỗi:" + str(e))        def XoaPhieuNhapChiTiet(self):          selected = self.trHienThi.selection()          if not selected:              messagebox.showwarning("Cảnh báo", "Vui lòng chọn phiếu nhập để xóa!")              return            traloi = messagebox.askyesno("Xác nhận", "Bạn có chắc chắn muốn xóa phiếu nhập này không?")          if traloi:              self.trHienThi.delete(selected[0])              # Cập nhật tổng tiền              self.TinhTongTien()        def LamMoi(self):          self.txt\_maphieunhap.config(state="normal")          self.dt\_ngaynhap.config(state="normal")          self.cb\_manhanvien.config(state="readonly")          self.cb\_manhacungcap.config(state="readonly")          self.txt\_maphieunhap.delete(0, tk.END)          self.dt\_ngaynhap.set\_date(datetime.today())          self.cb\_manhanvien.set("")          self.cb\_manhacungcap.set("")          self.cb\_mativi.set("")          self.txt\_soluong.delete(0, tk.END)          self.txt\_gianhap.delete(0, tk.END)          self.trHienThi.delete(\*self.trHienThi.get\_children())          self.lbl\_tongtien.config(text="0 VNĐ")          self.load\_combobox()      def TaoPhieuNhap(self):          if not self.trHienThi.get\_children():              messagebox.showwarning("Cảnh báo", "Không có phiếu nhập nào để thực hiện tạo phiếu nhập hàng!")              return            ma\_phieu = self.txt\_maphieunhap.get()          ngay\_nhap = self.dt\_ngaynhap.get\_date().strftime("%Y-%m-%d")          ma\_nv = self.cb\_manhanvien.get()          ma\_ncc = self.cb\_manhacungcap.get()          try:              cursor = self.conn.cursor()              # Thêm phiếu nhập hàng - vì 1 phiếu nhập có thể có nhiều chi tiết nên chỉ cần thêm 1 lần              cursor.execute("""                  INSERT INTO PHIEUNHAPHANG (MaPhieuNhap, NgayNhap, MaNV, MaNCC)                  VALUES (?, ?, ?, ?)              """, (ma\_phieu, ngay\_nhap, ma\_nv, ma\_ncc))              for item in self.trHienThi.get\_children():                  values = self.trHienThi.item(item, "values")                  ma\_tivi = values[6]                  so\_luong = int(values[8])                  gia\_nhap = float(values[9].replace(",", ""))                  # Thêm chi tiết phiếu nhập hàng                  cursor.execute("""                      INSERT INTO CHITIETPHIEUNHAP (MaPhieuNhap, MaTiVi, SoLuong, GiaNhap)                      VALUES (?, ?, ?, ?)                  """, (ma\_phieu, ma\_tivi, so\_luong, gia\_nhap))              self.conn.commit()              cursor.close()              messagebox.showinfo("Thành công", "Tạo phiếu nhập hàng thành công!")              self.LamMoi()              # Cập nhật lại tab phiếu nhập hàng nếu có              if self.tab\_phieunhap:                  self.tab\_phieunhap.load\_phieu\_nhap()          except Exception as e:              messagebox.showerror("Lỗi", "Đã xảy ra lỗi khi tạo phiếu nhập hàng:\n" + str(e)) |

**FORM PHIẾU NHẬP HÀNG**

|  |
| --- |
| import tkinter as tk  from tkinter import ttk, messagebox, filedialog  from datetime import datetime  import pyodbc  from docx import Document  from docx.shared import Inches, Pt, RGBColor, Cm  from docx.enum.text import WD\_ALIGN\_PARAGRAPH, WD\_TAB\_ALIGNMENT, WD\_TAB\_LEADER  # === TAB PHIẾU NHẬP HÀNG ===  class tabPhieuNhapHang(tk.Frame):      def \_\_init\_\_(self, parent, conn, controller):          super().\_\_init\_\_(parent, bg="white")          # === CHUỖI KẾT NỐI ===          self.conn = conn          self.controller = controller          # === KHUNG TÌM KIẾM ===          frame\_search = tk.Frame(self, bg="#E3F2FD", padx=10, pady=10)          frame\_search.pack(fill="x", padx=20, pady=5)          tk.Label(frame\_search, text="🔍 Tìm kiếm:", font=("Segoe UI", 10), bg="#E3F2FD").pack(side="left", padx=5)          self.txt\_timkiem = tk.Entry(frame\_search, font=("Segoe UI", 10), width=20)          self.txt\_timkiem.pack(side="left", padx=5)          self.search\_option = tk.StringVar(value="mapn")          tk.Radiobutton(frame\_search, text="Theo mã phiếu nhập", variable=self.search\_option, value="mapn", bg="#E3F2FD", font=("Segoe UI", 10)).pack(side="left", padx=10)          tk.Radiobutton(frame\_search, text="Theo mã nhà cung cấp", variable=self.search\_option, value="mancc", bg="#E3F2FD", font=("Segoe UI", 10)).pack(side="left" , padx=10)          tk.Radiobutton(frame\_search, text="Theo trạng thái phiếu", variable=self.search\_option, value="trangthai", bg="#E3F2FD", font=("Segoe UI", 10)).pack(side="left", padx=10)          tk.Button(frame\_search, text="Tìm", font=("Segoe UI", 10, "bold"), command=self.TimKiem, bg="#1565C0", fg="white", bd=0, padx=10, pady=5).pack(side="left", padx=10)          tk.Button(frame\_search, text="Hủy", font=("Segoe UI", 10, "bold"), command=self.HuyTimKiem, bg="#E53935", fg="white", bd=0, padx=10, pady=5).pack(side="left", padx=10)          # === BẢNG DANH SÁCH PHIẾU NHẬP HÀNG ===          frame\_table = tk.Frame(self, bg="white")          frame\_table.pack(fill="both", expand=True, padx=20, pady=10)          columns = ("MaPhieuNhap", "NgayNhap", "MaNV", "MaNCC", "TongTien", "TrangThai")            # --- Tạo Scrollbar ---          scroll\_y = ttk.Scrollbar(frame\_table, orient="vertical")          scroll\_x = ttk.Scrollbar(frame\_table, orient="horizontal")          self.trHienThi = ttk.Treeview( frame\_table, show="headings",  columns=columns, height=12, yscrollcommand=scroll\_y.set, xscrollcommand=scroll\_x.set)          # --- Gắn Scrollbar ---          scroll\_y.config(command=self.trHienThi.yview)          scroll\_x.config(command=self.trHienThi.xview)          # --- Bố trí Scrollbar ---          scroll\_y.pack(side="right", fill="y")          scroll\_x.pack(side="bottom", fill="x")          self.trHienThi.pack(fill="both", expand=True)          self.trHienThi.heading("MaPhieuNhap", text="Mã phiếu nhập")          self.trHienThi.heading("NgayNhap", text="Ngày nhập")          self.trHienThi.heading("MaNV", text="Mã nhân viên")          self.trHienThi.heading("MaNCC", text="Mã nhà cung cấp")          self.trHienThi.heading("TongTien", text="Tổng tiền")          self.trHienThi.heading("TrangThai", text="Trạng thái")          self.trHienThi.column("MaPhieuNhap", width=100,)          self.trHienThi.column("NgayNhap", width=100, anchor="center")          self.trHienThi.column("MaNV", width=100,)          self.trHienThi.column("MaNCC", width=100,)          self.trHienThi.column("TongTien", width=100, anchor="center")          self.trHienThi.column("TrangThai", width=100)          # Thêm style cho Treeview          style = ttk.Style()          style.configure("Treeview.Heading", font=("Segoe UI", 11, "bold"))          style.configure("Treeview", font=("Segoe UI", 10), rowheight=28)          # ==== Nút thao tác ====          frame\_btn = tk.Frame(self, bg="white")          frame\_btn.pack(pady=10)          tk.Button(frame\_btn, text="🧐 Xem chi tiết", bg="#EC9428", fg="white", font=("Segoe UI", 10, "bold"), command=self.XemChiTiet, padx=15, pady=5, bd=0).pack(side="left", padx=5)          tk.Button(frame\_btn, text="✅ Duyệt phiếu nhập hàng", bg="#43A047", fg="white",  font=("Segoe UI", 10, "bold"), command=self.DuyetPhieuNhapHang, padx=15, pady=5, bd=0).pack(side="left", padx=5)          tk.Button(frame\_btn, text="🗑 Hủy phiếu nhập hàng", bg="#E53935", fg="white",  font=("Segoe UI", 10, "bold"), command=self.HuyPhieuNhapHang, padx=15, pady=5, bd=0).pack(side="left", padx=5)          tk.Button(frame\_btn, text="🔄 Làm mới", bg="#1E88E5", fg="white", font=("Segoe UI", 10, "bold"), command=self.load\_phieu\_nhap, padx=15, pady=5, bd=0).pack(side="left", padx=5)          tk.Button(frame\_btn, text="🗑️ Xóa", bg="#B71C1C", fg="white", font=("Segoe UI", 10, "bold"), command=self.XoaPhieuNhapVinhVien, padx=15, pady=5, bd=0).pack(side="left", padx=5)          tk.Button(frame\_btn, text="🖨️  In", bg="#E51E9C", fg="white", font=("Segoe UI", 10, "bold"), command=self.InPhieuNhapHang, padx=15, pady=5, bd=0).pack(side="left", padx=5)          # === TẢI DỮ LIỆU pHIẾU NHẬP ===          self.load\_phieu\_nhap()      def load\_phieu\_nhap(self):          try:              self.trHienThi.delete(\*self.trHienThi.get\_children())              cursor = self.conn.cursor()              cursor.execute("SELECT MaPhieuNhap, NgayNhap, MaNV, MaNCC, TongTien, TrangThai FROM PHIEUNHAPHANG")              for row in cursor.fetchall():                  ngay\_nhap = datetime.strptime(str(row.NgayNhap).split(" ")[0], "%Y-%m-%d")                  formatted\_row = (                      row.MaPhieuNhap, ngay\_nhap.strftime("%d/%m/%y"), row.MaNV, row.MaNCC,                      f"{float(row.TongTien):,.0f}" if row.TongTien else "0", row.TrangThai)                    self.trHienThi.insert("", tk.END, values=formatted\_row)            except Exception as e:              messagebox.showerror("Lỗi", "Không thể tải dữ liệu hóa đơn:\n + " + str(e))      # === Hàm xem chi tiết phiếu nhập ===      def XemChiTiet(self):          selected = self.trHienThi.selection()          if not selected:              messagebox.showwarning("Thông báo", "Vui lòng chọn 1 phiếu nhập để xem chi tiết.")              return          ma\_phieu = self.trHienThi.item(selected[0], "values")[0]          try:              chitietphieunhap= tk.Toplevel(self)              chitietphieunhap.title("Chi tiết phiếu nhập: " + ma\_phieu)              chitietphieunhap.geometry("700x800")              chitietphieunhap.resizable(False, False)              chitietphieunhap.configure(bg="white")              tk.Label(chitietphieunhap, text="Chi tiết phiếu nhập hàng: " + ma\_phieu, font=("Segoe UI", 12, "bold"), bg="white", fg="#0D47A1").pack(pady=10)                cursor = self.conn.cursor()              cursor.execute("""SELECT nv.MaNV, nv.TenNV, ncc.MaNCC, ncc.TenNCC, pnh.NgayNhap, pnh.TongTien, pnh.TrangThai                                FROM PHIEUNHAPHANG pnh JOIN NHANVIEN nv ON pnh.MaNV = nv.MaNV                                JOIN NHACUNGCAP ncc ON pnh.MaNCC = ncc.MaNCC                                WHERE pnh.MaPhieuNhap = ?""", (ma\_phieu,))              thong\_tin = cursor.fetchone()              Frame\_thongtin = tk.Frame(chitietphieunhap, bg="white")              Frame\_thongtin.pack(pady=5)              tk.Label(Frame\_thongtin, text="Mã nhân viên:", font=("Segoe UI", 10), bg="white").grid(row=0, column=0, padx=10, sticky="w")              tk.Label(Frame\_thongtin, text=thong\_tin.MaNV, font=("Segoe UI", 10, "bold"), bg="white", fg="#1565C0").grid(row=0, column=1, padx=10, sticky="w")              tk.Label(Frame\_thongtin, text="Tên nhân viên:", font=("Segoe UI", 10), bg="white").grid(row=0, column=2, padx=10, sticky="w")              tk.Label(Frame\_thongtin, text=thong\_tin.TenNV, font=("Segoe UI", 10, "bold"), bg="white", fg="#1565C0").grid(row=0, column=3, padx=10, sticky="w")              tk.Label(Frame\_thongtin, text="Mã nhà cung cấp:", font=("Segoe UI", 10), bg="white").grid(row=1, column=0, padx=10, sticky="w")              tk.Label(Frame\_thongtin, text=thong\_tin.MaNCC, font=("Segoe UI", 10, "bold"), bg="white", fg="#1565C0").grid(row=1, column=1, padx=10, sticky="w")              tk.Label(Frame\_thongtin, text="Tên nhà cung cấp:", font=("Segoe UI", 10), bg="white").grid(row=1, column=2, padx=10, sticky="w")              tk.Label(Frame\_thongtin, text=thong\_tin.TenNCC, font=("Segoe UI", 10, "bold"), bg="white", fg="#1565C0").grid(row=1, column=3, padx=10, sticky="w")              tk.Label(Frame\_thongtin, text="Ngày nhập:", font=("Segoe UI", 10), bg="white").grid(row=2, column=0, padx=10, sticky="w")                ngay\_nhap = datetime.strptime(str(thong\_tin.NgayNhap).split(" ")[0], "%Y-%m-%d")              tk.Label(Frame\_thongtin, text=ngay\_nhap.strftime("%d/%m/%Y"), font=("Segoe UI", 10, "bold"), bg="white", fg="#43A047").grid(row=2, column=1, padx=10, sticky="w")              tk.Label(Frame\_thongtin, text="Trạng thái:", font=("Segoe UI", 10), bg="white").grid(row=2, column=2, padx=10, sticky="w")              tk.Label(Frame\_thongtin, text=thong\_tin.TrangThai, font=("Segoe UI", 10, "bold"), bg="white", fg="#43A047").grid(row=2, column=3, padx=10, sticky="w")                columns = ("MaTivi", "TenTivi", "SoLuong", "GiaNhap", "ThanhTien")              tree = ttk.Treeview(chitietphieunhap, columns=columns, show="headings", height=10)              tree.pack(fill="both", expand=True, padx=15, pady=10)              tree.heading("MaTivi", text="Mã Tivi")              tree.heading("TenTivi", text="Tên Tivi")              tree.heading("SoLuong", text="Số Lượng")              tree.heading("GiaNhap", text="Giá Nhập")              tree.heading("ThanhTien", text="Thành Tiền")              tree.column("MaTivi", width=100)              tree.column("TenTivi", width=200)              tree.column("SoLuong", width=100, anchor="center")              tree.column("GiaNhap", width=100, anchor="center")              tree.column("ThanhTien", width=100, anchor="center")              tk.Label(chitietphieunhap, text="Tổng tiền:", font=("Segoe UI", 10, "bold"), bg="white").pack(side="left", padx=20)              tk.Label(chitietphieunhap, text=f"{float(thong\_tin.TongTien):,.0f} đ", font=("Segoe UI", 10, "bold"), bg="white", fg="red").pack(side="right", padx=20)                cursor.execute("""                  SELECT ctpn.MaTivi, tv.TenTivi, ctpn.SoLuong, ctpn.GiaNhap, ctpn.ThanhTien                  FROM CHITIETPHIEUNHAP ctpn                  JOIN TIVI tv ON ctpn.MaTivi = tv.MaTivi                  WHERE ctpn.MaPhieuNhap = ?              """, (ma\_phieu,))              rows = cursor.fetchall()              if not rows:                  messagebox.showinfo("Thông báo", "Không có chi tiết cho phiếu " + ma\_phieu)                  chitietphieunhap.destroy()                  return              for r in rows:                  tree.insert("", tk.END, values=(r.MaTivi, r.TenTivi,  r.SoLuong, f"{float(r.GiaNhap):,.0f} đ", f"{float(r.ThanhTien):,.0f} đ"))              cursor.close()          except Exception as e:              messagebox.showerror("Lỗi", "Không thể xem chi tiết phiếu nhập:\n" + str(e))      def DuyetPhieuNhapHang(self):          selected = self.trHienThi.selection()          if not selected:              messagebox.showwarning("Thông báo", "Vui lòng chọn 1 phiếu nhập để duyệt.")              return          ma\_phieu = self.trHienThi.item(selected[0], "values")[0]          trang\_thai = self.trHienThi.item(selected[0], "values")[5]          if trang\_thai != "Đợi duyệt":              messagebox.showwarning("Thông báo", "Chỉ có thể duyệt phiếu nhập hàng chưa được duyệt.")              return            traloi = messagebox.askyesno("Xác nhận", "Bạn có chắc chắn muốn duyệt phiếu nhập hàng " + ma\_phieu + " không?")          if traloi:              try:                  cursor = self.conn.cursor()                  # Cập nhật duyệt hàng                  cursor.execute("""                      UPDATE PHIEUNHAPHANG                      SET TrangThai = N'Đã duyệt'                      WHERE MaPhieuNhap = ?                  """, (ma\_phieu,))                  # Lấy chi tiết phiếu nhập hàng để cập nhật tồn kho                  cursor.execute("""                      SELECT MaTivi, SoLuong                      FROM CHITIETPHIEUNHAP                      WHERE MaPhieuNhap = ?                  """, (ma\_phieu,))                  chitiet = cursor.fetchall()                  for item in chitiet:                      ma\_tivi = item.MaTivi                      so\_luong = item.SoLuong                      # Cập nhật số lượng tồn kho của tivi                      cursor.execute("""                          UPDATE TIVI                          SET SoLuongTon = SoLuongTon + ?                          WHERE MaTivi = ? """, (so\_luong, ma\_tivi))                    self.conn.commit()                  cursor.close()                  messagebox.showinfo("Thành công", "Phiếu nhập hàng đã được duyệt thành công!")                  # Làm mới lại danh sách phiếu nhập hàng                  self.load\_phieu\_nhap()              except Exception as e:                  messagebox.showerror("Lỗi", "Đã xảy ra lỗi khi duyệt phiếu nhập hàng:\n" + str(e))            self.controller.load\_form("NhapHangVaPhieuNhap")      def HuyPhieuNhapHang(self):          selected = self.trHienThi.selection()          if not selected:              messagebox.showwarning("Thông báo", "Vui lòng chọn 1 phiếu nhập để hủy")              return            ma\_phieu = self.trHienThi.item(selected[0], "values")[0]          trang\_thai = self.trHienThi.item(selected[0], "values")[5]          if trang\_thai != "Đợi duyệt":              messagebox.showwarning("Thông báo", "Chỉ có thể hủy phiếu nhập hàng chưa được duyệt.")              return            traloi = messagebox.askyesno("Xác nhận", "Bạn có chắc chắn muốn hủy phiếu nhập hàng " + ma\_phieu + " không?")          if traloi:              try:                  cursor = self.conn.cursor()                  cursor.execute("""                      UPDATE PHIEUNHAPHANG                      SET TrangThai = N'Đã hủy'                      WHERE MaPhieuNhap = ?""", (ma\_phieu,))                    self.conn.commit()                  cursor.close()                  messagebox.showinfo("Thành công", "Phiếu nhập hàng đã được hủy thành công!")                  self.load\_phieu\_nhap()              except Exception as e:                  messagebox.showerror("Lỗi", "Đã xảy ra lỗi khi hủy phiếu nhập hàng:\n" + str(e))          self.controller.load\_form("NhapHangVaPhieuNhap")      def TimKiem(self):          if self.txt\_timkiem.get().strip() == "":              messagebox.showwarning("Thông báo", "Vui lòng nhập từ khóa tìm kiếm.")              return            timkkiem = self.search\_option.get().strip()          self.trHienThi.delete(\*self.trHienThi.get\_children())          try:              cursor = self.conn.cursor()              if timkkiem == "mapn":                  keyword = self.txt\_timkiem.get()                  cursor.execute("""                      SELECT MaPhieuNhap, NgayNhap, MaNV, MaNCC, TongTien, TrangThai                      FROM PHIEUNHAPHANG                      WHERE MaPhieuNhap = ?""", (keyword,))                elif timkkiem == "mancc":                  keyword = self.txt\_timkiem.get()                  cursor.execute("""                      SELECT MaPhieuNhap, NgayNhap, MaNV, MaNCC, TongTien, TrangThai                      FROM PHIEUNHAPHANG                      WHERE MaNCC = ?""", (keyword,))                elif timkkiem == "trangthai":                  keyword = self.txt\_timkiem.get()                  cursor.execute("""                      SELECT MaPhieuNhap, NgayNhap, MaNV, MaNCC, TongTien, TrangThai                      FROM PHIEUNHAPHANG                      WHERE TrangThai = ?""", (keyword,))                rows = cursor.fetchall()              for row in rows:                  ngay\_nhap = datetime.strptime(str(row.NgayNhap).split(" ")[0], "%Y-%m-%d")                  formatted\_row = (                      row.MaPhieuNhap, ngay\_nhap.strftime("%d/%m/%Y"), row.MaNV, row.MaNCC,                      f"{float(row.TongTien):,.0f}" if row.TongTien else "0", row.TrangThai                  )                  self.trHienThi.insert("", tk.END, values=formatted\_row)              cursor.close()          except Exception as e:              messagebox.showerror("Lỗi", "Không thể tìm kiếm phiếu nhập hàng:\n" + str(e))      def HuyTimKiem(self):          # Xóa txt\_timkiem nếu có          self.load\_phieu\_nhap()          self.txt\_timkiem.delete(0, tk.END)      def XoaPhieuNhapVinhVien(self):          selected = self.trHienThi.selection()          if not selected:              messagebox.showwarning("Thông báo", "Vui lòng chọn 1 phiếu nhập để xóa.")              return          ma\_phieu = self.trHienThi.item(selected[0], "values")[0]          trang\_thai = self.trHienThi.item(selected[0], "values")[5]            # LẤY NGÀY NHẬP CỦA PHIẾU ĐƯỢC CHỌN TỪ CSDL          try:              cursor\_temp = self.conn.cursor()              cursor\_temp.execute("SELECT NgayNhap FROM PHIEUNHAPHANG WHERE MaPhieuNhap = ?", (ma\_phieu,))              ngay\_nhap\_db = cursor\_temp.fetchone()[0]              cursor\_temp.close()                # Tính toán xem phiếu đã tồn tại trên 5 năm chưa              ngay\_gioi\_han = datetime.now().date().replace(year=datetime.now().year - 5)              da\_hon\_5\_nam = ngay\_nhap\_db < ngay\_gioi\_han            except Exception as e:              messagebox.showerror("Lỗi", "Không thể kiểm tra ngày nhập của phiếu: " + str(e))              return            is\_allowed\_to\_delete = False          warning\_message = ""            if trang\_thai == "Đã hủy":              is\_allowed\_to\_delete = True              warning\_message = f"Bạn có chắc chắn muốn xóa vĩnh viễn phiếu nhập ĐÃ HỦY {ma\_phieu} không?"            elif trang\_thai == "Đã duyệt" and da\_hon\_5\_nam:              is\_allowed\_to\_delete = True              warning\_message = f"Phiếu nhập ĐÃ DUYỆT {ma\_phieu} này đã được lưu trữ hơn 5 năm ({ngay\_nhap\_db.strftime('%d/%m/%Y')}). Bạn có chắc chắn muốn XÓA VĨNH VIỄN không?"            if not is\_allowed\_to\_delete:              messagebox.showwarning("Cảnh báo", "Chỉ có thể xóa vĩnh viễn phiếu nhập ĐÃ HỦY, hoặc phiếu ĐÃ DUYỆT có thời gian lưu trữ trên 5 năm.")              return          traloi = messagebox.askyesno("Xác nhận XÓA VĨNH VIỄN", warning\_message)            if traloi:              try:                  # Bắt đầu xóa CSDL                  cursor = self.conn.cursor()                    # Xóa Chi Tiết Phiếu Nhập trước                  cursor.execute("DELETE FROM ChiTietPhieuNhap WHERE MaPhieuNhap = ?", (ma\_phieu,))                    # Xóa Phiếu Nhập Hàng                  cursor.execute("DELETE FROM PhieuNhapHang WHERE MaPhieuNhap = ?", (ma\_phieu,))                    self.conn.commit()                  cursor.close()                  messagebox.showinfo("Thành công", f"Phiếu nhập hàng {ma\_phieu} đã được xóa vĩnh viễn khỏi hệ thống!")                    self.load\_phieu\_nhap()              except Exception as e:                  messagebox.showerror("Lỗi CSDL", f"Đã xảy ra lỗi khi xóa phiếu nhập: "  + str(e))            self.controller.load\_form("NhapHangVaPhieuNhap")      # === HÀM IN PHIẾU NHẬP HÀNG RA WORD ===      def InPhieuNhapHang(self):          selected = self.trHienThi.selection()          if not selected:              messagebox.showwarning("Thông báo", "Vui lòng chọn 1 phiếu nhập để in.")              return          ma\_phieu = self.trHienThi.item(selected[0], "values")[0]            # --- Lấy dữ liệu chi tiết ---          try:              cursor = self.conn.cursor()                # 1. Lấy thông tin chung của phiếu nhập (Giữ nguyên)              cursor.execute("""SELECT pnh.MaPhieuNhap, nv.MaNV, nv.TenNV, ncc.MaNCC, ncc.TenNCC, pnh.NgayNhap, pnh.TongTien, pnh.TrangThai                                FROM PHIEUNHAPHANG pnh                                JOIN NHANVIEN nv ON pnh.MaNV = nv.MaNV                                JOIN NHACUNGCAP ncc ON pnh.MaNCC = ncc.MaNCC                                WHERE pnh.MaPhieuNhap = ?""", (ma\_phieu,))              thong\_tin = cursor.fetchone()              if not thong\_tin:                  messagebox.showerror("Lỗi", "Không tìm thấy thông tin phiếu nhập " + ma\_phieu)                  return              # 2. Lấy chi tiết các mặt hàng (Giữ nguyên)              cursor.execute("""                  SELECT ctpn.MaTivi, tv.TenTivi, ctpn.SoLuong, ctpn.GiaNhap, ctpn.ThanhTien                  FROM CHITIETPHIEUNHAP ctpn                  JOIN TIVI tv ON ctpn.MaTivi = tv.MaTivi                  WHERE ctpn.MaPhieuNhap = ?              """, (ma\_phieu,))              chi\_tiet\_hang = cursor.fetchall()              cursor.close()              if not chi\_tiet\_hang:                  messagebox.showinfo("Thông báo", "Phiếu nhập " + ma\_phieu + " không có chi tiết hàng hóa.")                  return          except Exception as e:              messagebox.showerror("Lỗi CSDL", "Không thể lấy dữ liệu để in:\n" + str(e))              return            # --- Tạo File Word ---          try:              document = Document()                # Đặt font và size mặc định cho document              style = document.styles['Normal']              style.font.name = 'Times New Roman'              style.font.size = Pt(11)              # Dữ liệu định dạng              ngay\_nhap\_date = datetime.strptime(str(thong\_tin.NgayNhap).split(" ")[0], "%Y-%m-%d")              ngay\_nhap\_str = ngay\_nhap\_date.strftime("%d/%m/%Y")              tong\_tien\_str = f"{float(thong\_tin.TongTien):,.0f} đ"                # Tiêu đề chính (Căn giữa)              heading = document.add\_paragraph()              heading.alignment = WD\_ALIGN\_PARAGRAPH.CENTER              run = heading.add\_run(f"PHIẾU NHẬP HÀNG SỐ: {ma\_phieu}")              run.bold = True              run.font.size = Pt(16)              document.add\_paragraph() # Dòng trống                # --- Bố trí Thông tin chung  ---                # Thiết lập Tab Stop: Căn lề trái tại vị trí 8.0 cm              TAB\_STOP\_POS = Cm(5.0)                # Thông tin Nhân viên              p1 = document.add\_paragraph()              p1.paragraph\_format.tab\_stops.add\_tab\_stop(TAB\_STOP\_POS, WD\_TAB\_ALIGNMENT.LEFT, WD\_TAB\_LEADER.SPACES)              p1.add\_run("Mã nhân viên: ").bold = False              run = p1.add\_run(thong\_tin.MaNV)              run.bold = True              run.font.color.rgb = RGBColor(0x15, 0x65, 0xC0)              p1.add\_run('\t')              p1.add\_run("Tên nhân viên: ").bold = False              run = p1.add\_run(thong\_tin.TenNV)              run.bold = True              run.font.color.rgb = RGBColor(0x15, 0x65, 0xC0)              # Thông tin Nhà cung cấp              p2 = document.add\_paragraph()              p2.paragraph\_format.tab\_stops.add\_tab\_stop(TAB\_STOP\_POS, WD\_TAB\_ALIGNMENT.LEFT, WD\_TAB\_LEADER.SPACES)              p2.add\_run("Mã nhà cung cấp: ").bold = False              run = p2.add\_run(thong\_tin.MaNCC)              run.bold = True              run.font.color.rgb = RGBColor(0x15, 0x65, 0xC0)              p2.add\_run('\t')              p2.add\_run("Tên nhà cung cấp: ").bold = False              run = p2.add\_run(thong\_tin.TenNCC)              run.bold = True              run.font.color.rgb = RGBColor(0x15, 0x65, 0xC0)              # Thông tin Ngày nhập / Trạng thái              p3 = document.add\_paragraph()              p3.paragraph\_format.tab\_stops.add\_tab\_stop(TAB\_STOP\_POS, WD\_TAB\_ALIGNMENT.LEFT, WD\_TAB\_LEADER.SPACES)              p3.add\_run("Ngày nhập: ").bold = False              run = p3.add\_run(ngay\_nhap\_str)              run.bold = True              run.font.color.rgb = RGBColor(0x43, 0xA0, 0x47)              p3.add\_run('\t')              p3.add\_run("Trạng thái: ").bold = False              run = p3.add\_run(thong\_tin.TrangThai)              run.bold = True              run.font.color.rgb = RGBColor(0x43, 0xA0, 0x47)                document.add\_paragraph() # Dòng trống              # --- Bảng chi tiết hàng hóa (Giữ nguyên) ---              table = document.add\_table(rows=1, cols=5)              table.style = 'Table Grid'              # ... (Phần headers và ghi dữ liệu chi tiết giữ nguyên) ...                hdr\_cells = table.rows[0].cells              headers = ["Mã Tivi", "Tên Tivi", "Số Lượng", "Giá Nhập", "Thành Tiền"]              for i, header in enumerate(headers):                  hdr\_cells[i].text = header                  hdr\_cells[i].paragraphs[0].runs[0].bold = True                  hdr\_cells[i].paragraphs[0].alignment = WD\_ALIGN\_PARAGRAPH.CENTER                for item in chi\_tiet\_hang:                  row\_cells = table.add\_row().cells                  row\_cells[0].text = item.MaTivi                  row\_cells[1].text = item.TenTivi                  row\_cells[2].text = str(item.SoLuong)                  row\_cells[3].text = f"{float(item.GiaNhap):,.0f} đ"                  row\_cells[4].text = f"{float(item.ThanhTien):,.0f} đ"                  for i in [2, 3, 4]:                      row\_cells[i].paragraphs[0].alignment = WD\_ALIGN\_PARAGRAPH.CENTER              # --- Tổng tiền (Căn phải) ---              document.add\_paragraph()              tong\_tien\_para = document.add\_paragraph()              tong\_tien\_para.alignment = WD\_ALIGN\_PARAGRAPH.RIGHT                tong\_tien\_para.add\_run("Tổng tiền: ").bold = True              run = tong\_tien\_para.add\_run(tong\_tien\_str)              run.bold = True              run.font.size = Pt(12)              run.font.color.rgb = RGBColor(0xFF, 0x00, 0x00)              # --- KHU VỰC KÝ TÊN ---              document.add\_paragraph()              document.add\_paragraph()                # Tạo bảng 3 cột cho khu vực chữ ký              signature\_table = document.add\_table(rows=2, cols=3)              signature\_table.autofit = False              signature\_table.style = None # Bỏ style lưới để không có viền              # Hàng 1: Chức danh              chuc\_danh = ["Người lập phiếu", "Thủ kho", "Kế toán trưởng"]              for i, cd in enumerate(chuc\_danh):                  cell = signature\_table.rows[0].cells[i]                  cell.text = cd                  cell.paragraphs[0].alignment = WD\_ALIGN\_PARAGRAPH.CENTER                  cell.paragraphs[0].runs[0].bold = True                # Hàng 2: (Ghi rõ họ tên)              for i in range(3):                  cell = signature\_table.rows[1].cells[i]                  # Thêm tên nhân viên lập phiếu vào ô đầu tiên                  ten\_nguoi\_lap = thong\_tin.TenNV if i == 0 else " " # Thêm tên người lập vào ô đầu tiên                  cell.text = f"(Ký, ghi rõ họ tên) \n{ten\_nguoi\_lap}"                  cell.paragraphs[0].alignment = WD\_ALIGN\_PARAGRAPH.CENTER              # Lưu file              default\_filename = f"PhieuNhapHang\_{ma\_phieu}.docx"              file\_path = filedialog.asksaveasfilename(                  defaultextension=".docx",                  initialfile=default\_filename,                  filetypes=[("Word documents", "\*.docx"), ("All files", "\*.\*")]              )              if file\_path:                  document.save(file\_path)                  messagebox.showinfo("Thành công", f"Đã in phiếu nhập hàng {ma\_phieu} ra file Word thành công tại:\n{file\_path}")          except Exception as e:              messagebox.showerror("Lỗi File Word", "Không thể tạo hoặc lưu file Word:\n" + str(e)) |

**FORM THỐNG KÊ VÀ BÁO CÁO**

|  |
| --- |
| import tkinter as tk  from tkinter import ttk, messagebox  import pyodbc  import tabThongKeDoanhThu as tkdt  import tabBaoCaoSanPham as bcsp  #=== Tạo class Thống Kê và Báo Cáo  class ThongKeVaBaoCao(tk.Frame):      def \_\_init\_\_(self, parent, controller, conn, user):          super().\_\_init\_\_(parent, bg="white")          # === CHUỖI KẾT NỐI  ====          self.conn = conn          self.controller = controller          # === TITLE ===          lbl\_title = tk.Label(self, text="THỐNG KÊ & BÁO CÁO", font=("Segoe UI", 16, "bold"), bg="white", fg="#0D47A1")          lbl\_title.pack()          # === Tab control ===          tab\_control = ttk.Notebook(self)          tab\_control.pack(fill="both", expand=True, padx=20, pady=10)          #  KHỞI TẠO VÀ LƯU THAM CHIẾU VÀO self.tab\_doanhthu\_ref          self.tab\_doanhthu = tkdt.tabThongKeDoanhThu(tab\_control, conn)          self.tab\_baocao  = bcsp.tabBaoCaoSanPham(tab\_control, conn)          tab\_control.add(self.tab\_doanhthu, text="💹 Thống kê Doanh thu")          tab\_control.add(self.tab\_baocao , text="📊 Báo cáo Sản phẩm")      # Hàm làm mới tab khi click vào      def load\_data(self):          try:              self.tab\_doanhthu.thongke\_doanhthu\_tatca()                self.tab\_baocao.load\_baocao\_all()              self.tab\_baocao.load\_combo\_tivi()            except Exception as e:              messagebox.showerror("Lỗi khi làm mới tab con: " + str(e)) |

**FORM THỐNG KẾ DOANH THU**

|  |
| --- |
| import tkinter as tk  from tkinter import ttk, messagebox  import pyodbc  from tkcalendar import DateEntry  from datetime import date, datetime  class tabThongKeDoanhThu(tk.Frame):      def \_\_init\_\_(self, parent, conn):          super().\_\_init\_\_(parent, bg="white")          self.conn = conn          # === Bộ lọc thống kê ===          frame\_filter = tk.LabelFrame(self, text="Bộ lọc thống kê", bg="white", font=("Segoe UI", 12, "bold"), fg="#0D47A1", padx=10, pady=10,)          frame\_filter.pack(fill="x", padx=20, pady=10)          tk.Label(frame\_filter, text="Từ ngày:", bg="white", font=("Segoe UI", 10)).grid(row=0, column=0, padx=5, pady=5)          self.date\_tungay = DateEntry(frame\_filter, width=36, date\_pattern="dd/mm/yyyy", background="#1565C0", foreground="white", borderwidth=2,)          self.date\_tungay.grid(row=0, column=1, padx=5, pady=5)          tk.Label(frame\_filter, text="Đến ngày:", bg="white", font=("Segoe UI", 10)).grid(row=0, column=2, padx=5, pady=5)          self.date\_denngay = DateEntry(frame\_filter, width=36, date\_pattern="dd/mm/yyyy", background="#1565C0", foreground="white", borderwidth=2,)          self.date\_denngay.grid(row=0, column=3, padx=5, pady=5)          btn\_thongke = tk.Button(frame\_filter, text="📈 Thống kê", bg="#1E88E5", fg="white", font=("Segoe UI", 11, "bold"), bd=0, padx=15, pady=5, command=self.thongke\_doanhthu\_loc,)          btn\_thongke.grid(row=0, column=4, padx=10)          tk.Button(frame\_filter, text="Hủy", font=("Segoe UI", 10, "bold"), bg="#E53935", fg="white", bd=0, padx=10, pady=5, command=self.huy,).grid(row=0, column=5, padx=10)          # === Kết quả tổng hợp ===          frame\_result = tk.Frame(self, bg="white")          frame\_result.pack(fill="x", padx=20, pady=10)          tk.Label(frame\_result, text="Tổng hóa đơn:", bg="white", font=("Segoe UI", 12)).grid(row=0, column=0, padx=5, pady=5)          self.lbl\_tonghd = tk.Label(frame\_result, text="0", bg="white", font=("Segoe UI", 12, "bold"), fg="#1565C0",)          self.lbl\_tonghd.grid(row=0, column=1, padx=5, pady=5)          tk.Label(frame\_result, text="Tổng doanh thu:", bg="white", font=("Segoe UI", 12)).grid(row=0, column=2, padx=5, pady=5)          self.lbl\_doanhthu = tk.Label(frame\_result, text="0 VNĐ", bg="white", font=("Segoe UI", 12, "bold"), fg="#E53935",)          self.lbl\_doanhthu.grid(row=0, column=3, padx=5, pady=5)          tk.Label(frame\_result, text="Lợi nhuận:", bg="white", font=("Segoe UI", 12)).grid(row=0, column=4, padx=5, pady=5)          self.lbl\_loinhuan = tk.Label(frame\_result, text="0 VNĐ", bg="white", font=("Segoe UI", 12, "bold"), fg="#43A047",)          self.lbl\_loinhuan.grid(row=0, column=5, padx=5, pady=5)          # === Treeview hiển thị hóa đơn ===          frame\_table = tk.Frame(self, bg="white")          frame\_table.pack(fill="both", expand=True, padx=20, pady=10)          columns = ("MaHD", "MaKH", "TenKH", "NgayBan", "TongTien")          scroll\_y = ttk.Scrollbar(frame\_table, orient="vertical")          scroll\_x = ttk.Scrollbar(frame\_table, orient="horizontal")          self.trHienThi = ttk.Treeview(frame\_table, show="headings", columns=columns, height=12, yscrollcommand=scroll\_y.set, xscrollcommand=scroll\_x.set,)          scroll\_y.config(command=self.trHienThi.yview)          scroll\_x.config(command=self.trHienThi.xview)          scroll\_y.pack(side="right", fill="y")          scroll\_x.pack(side="bottom", fill="x")          self.trHienThi.pack(fill="both", expand=True)          self.trHienThi.heading("MaHD", text="Mã hóa đơn")          self.trHienThi.heading("MaKH", text="Mã KH")          self.trHienThi.heading("TenKH", text="Tên khách hàng")          self.trHienThi.heading("NgayBan", text="Ngày bán")          self.trHienThi.heading("TongTien", text="Tổng tiền (VNĐ)")          # Cấu hình cột          self.trHienThi.column("MaHD", width=100, anchor="center")          self.trHienThi.column("MaKH", width=100, anchor="center")          self.trHienThi.column("TenKH", width=180, anchor="w")          self.trHienThi.column("NgayBan", width=120, anchor="center")          self.trHienThi.column("TongTien", width=150, anchor="center")          # Style          style = ttk.Style()          style.configure("Treeview.Heading", font=("Segoe UI", 11, "bold"))          style.configure("Treeview", font=("Segoe UI", 10), rowheight=28)          style.map("Treeview", background=[("selected", "#1565C0")])          self.thongke\_doanhthu\_tatca()      def thongke\_doanhthu\_tatca(self):          try:              cursor = self.conn.cursor()              for item in self.trHienThi.get\_children():                  self.trHienThi.delete(item)              cursor.execute("""                  SELECT hdb.MaHD, hdb.MaKH, kh.TenKH, hdb.NgayBan, hdb.TongTien                  FROM HoaDonBan hdb                  JOIN KhachHang kh ON kh.MaKH = hdb.MaKH                  WHERE hdb.TrangThai = N'Đã thanh toán'                  ORDER BY hdb.NgayBan DESC                  """)              rows = cursor.fetchall()              tong\_hoadon = 0              for mahd, makh, tenkh, ngayban, tongtien in rows:                  ngayban\_str = self.chuyen\_yyyy\_sang\_dd(ngayban)                  tongtien\_str = f"{tongtien:,.0f}" if tongtien else "0"                  self.trHienThi.insert("", "end", values=(mahd, makh, tenkh or "", ngayban\_str, tongtien\_str))                  tong\_hoadon += 1              cursor.execute("""                  SELECT SUM(TongTien)                  FROM HoaDonBan                  WHERE TrangThai = N'Đã thanh toán'                  """)              tong\_doanhthu = cursor.fetchone()[0] or 0              cursor.execute("""                  SELECT SUM(T1.SoLuong \* T2.AvgGiaNhap)                  FROM ChiTietHoaDon T1                  JOIN HoaDonBan HDB ON T1.MaHD = HDB.MaHD                  JOIN (                      SELECT MaTivi, AVG(GiaNhap) AS AvgGiaNhap                      FROM ChiTietPhieuNhap                      GROUP BY MaTivi                  ) AS T2 ON T1.MaTivi = T2.MaTivi                  WHERE HDB.TrangThai = N'Đã thanh toán'""")              tong\_nhap = cursor.fetchone()[0] or 0              loinhuan = tong\_doanhthu - tong\_nhap              self.lbl\_tonghd.config(text=str(tong\_hoadon))              self.lbl\_doanhthu.config(text=f"{tong\_doanhthu:,.0f} VNĐ")              self.lbl\_loinhuan.config(text=f"{loinhuan:,.0f} VNĐ")              cursor.close()          except Exception as e:              messagebox.showerror("Lỗi", f"Không thể thống kê: {str(e)}")      def thongke\_doanhthu\_loc(self):          try:              cursor = self.conn.cursor()              for item in self.trHienThi.get\_children():                  self.trHienThi.delete(item)              tungay = str(self.date\_tungay.get\_date())              denngay = str(self.date\_denngay.get\_date())              if tungay > denngay:                  messagebox.showwarning("Cảnh báo", "Từ ngày phải nhỏ hơn hoặc bằng đến ngày!")                  return              cursor.execute("""                  SELECT hdb.MaHD, hdb.MaKH, kh.TenKH, hdb.NgayBan, hdb.TongTien                  FROM HoaDonBan hdb                  JOIN KhachHang kh ON kh.MaKH = hdb.MaKH                  WHERE hdb.NgayBan BETWEEN ? AND ?                  ORDER BY hdb.NgayBan DESC                  """,(tungay, denngay),)              rows = cursor.fetchall()              tong\_hoadon = 0              for mahd, makh, tenkh, ngayban, tongtien in rows:                  ngayban\_str = self.chuyen\_yyyy\_sang\_dd(ngayban)                  tongtien\_str = f"{tongtien:,.0f}" if tongtien else "0"                  self.trHienThi.insert("", "end", values=(mahd, makh, tenkh or "", ngayban\_str, tongtien\_str),)                  tong\_hoadon += 1              cursor.execute("""                  SELECT SUM(TongTien)                  FROM HoaDonBan                  WHERE TrangThai = N'Đã thanh toán'                  AND NgayBan BETWEEN ? AND ?                  """,(tungay, denngay),)              tong\_doanhthu = cursor.fetchone()[0] or 0              cursor.execute("""                      SELECT SUM(T1.SoLuong \* T2.AvgGiaNhap)                  FROM ChiTietHoaDon T1                  JOIN HoaDonBan HDB ON T1.MaHD = HDB.MaHD                  JOIN (                      SELECT MaTivi, AVG(GiaNhap) AS AvgGiaNhap                      FROM ChiTietPhieuNhap                      GROUP BY MaTivi                  ) AS T2 ON T1.MaTivi = T2.MaTivi                  WHERE HDB.TrangThai = N'Đã thanh toán'                  AND HDB.NgayBan BETWEEN ? AND ?""", (tungay, denngay))              tong\_nhap = cursor.fetchone()[0] or 0              loinhuan = tong\_doanhthu - tong\_nhap              self.lbl\_tonghd.config(text=str(tong\_hoadon))              self.lbl\_doanhthu.config(text=f"{tong\_doanhthu:,.0f} VNĐ")              self.lbl\_loinhuan.config(text=f"{loinhuan:,.0f} VNĐ")              cursor.close()              if tong\_hoadon == 0:                  messagebox.showinfo("Thông báo", "Không có hóa đơn nào trong khoảng thời gian này!")          except Exception as e:              messagebox.showerror("Lỗi", "Không thể thống kê: " + str(e))      def huy(self):          self.date\_tungay.set\_date(date.today())          self.date\_denngay.set\_date(date.today())          self.thongke\_doanhthu\_tatca()      def chuyen\_yyyy\_sang\_dd(self, ngay\_db):          if ngay\_db is None:              return ""          ngay\_str = str(ngay\_db).strip()          if "-" in ngay\_str:              parts = ngay\_str.split("-")          else:              return ngay\_str          if len(parts) != 3:              return ngay\_str          try:              y, m, d = parts              return f"{d.zfill(2)}/{m.zfill(2)}/{y}"          except:              return ngay\_str      def chuyen\_dd\_sang\_datetime(self, ngay\_entry):          if not ngay\_entry:              return date.today()          ngay\_str = str(ngay\_entry).strip()          if "-" in ngay\_str:              parts = ngay\_str.split("-")          else:              return ngay\_str          if len(parts) != 3:              return date.today()          try:              d, m, y = map(int, parts)              return datetime(y, m, d).date()          except:              return date.today() |

**FORM BÁO CÁO SẢN PHẨM**

|  |
| --- |
| import tkinter as tk  from tkinter import ttk, messagebox  import pyodbc  from datetime import datetime  class tabBaoCaoSanPham(tk.Frame):      def \_\_init\_\_(self, parent, conn):          super().\_\_init\_\_(parent, bg="white")          self.conn = conn          frame\_filter = tk.LabelFrame(self, text="Tùy chọn báo cáo", bg="white", font=("Segoe UI", 12, "bold"), fg="#0D47A1", padx=10, pady=10)          frame\_filter.pack(fill="x", padx=20, pady=10)          tk.Label(frame\_filter, text="Chọn mã tivi:", bg="white", font=("Segoe UI", 11)).grid(row=0, column=0, padx=5, pady=5)          self.cbo\_mativi = ttk.Combobox(frame\_filter, width=31, state="readonly")          self.cbo\_mativi.grid(row=0, column=1, padx=5, pady=5)          self.cbo\_mativi.bind("<<ComboboxSelected>>", self.on\_mativi\_selected)          tk.Label(frame\_filter, text="Chọn tên tivi:", bg="white", font=("Segoe UI", 11)).grid(row=0, column=2, padx=5, pady=5)          self.cbo\_tentivi = ttk.Combobox(frame\_filter, width=31, state="readonly")          self.cbo\_tentivi.grid(row=0, column=3, padx=5, pady=5)          self.cbo\_tentivi.bind("<<ComboboxSelected>>", self.on\_tentivi\_selected)          tk.Button(frame\_filter, text="📊 Xem báo cáo", bg="#1E88E5", fg="white", font=("Segoe UI", 11, "bold"), bd=0, padx=15, pady=5, command=self.xem\_baocao).grid(row=0, column=4, padx=10)          tk.Button(frame\_filter, text="Hủy", bg="#E53935", fg="white", font=("Segoe UI", 11, "bold"), bd=0, padx=15, pady=5, command=self.huy\_xem\_bao\_cao).grid(row=0, column=5, padx=10)          frame\_table = tk.Frame(self, bg="white")          frame\_table.pack(fill="both", expand=True, padx=20, pady=10)          columns = ("MaTivi", "TenTivi", "SoLuongBan", "TiLeBanChay", "DoanhThu", "LoiNhuan")          # --- Tạo Scrollbar ---          scroll\_y = ttk.Scrollbar(frame\_table, orient="vertical")          scroll\_x = ttk.Scrollbar(frame\_table, orient="horizontal")          self.trHienThi = ttk.Treeview( frame\_table, show="headings",  columns=columns, height=12, yscrollcommand=scroll\_y.set, xscrollcommand=scroll\_x.set)          # --- Gắn Scrollbar ---          scroll\_y.config(command=self.trHienThi.yview)          scroll\_x.config(command=self.trHienThi.xview)          # --- Bố trí Scrollbar ---          scroll\_y.pack(side="right", fill="y")          scroll\_x.pack(side="bottom", fill="x")          self.trHienThi.pack(fill="both", expand=True)          self.trHienThi.heading("MaTivi", text="Mã Tivi")          self.trHienThi.heading("TenTivi", text="Tên Tivi")          self.trHienThi.heading("SoLuongBan", text="SL Bán")          self.trHienThi.heading("TiLeBanChay", text="Tỉ Lệ (%)")          self.trHienThi.heading("DoanhThu", text="Doanh Thu (VNĐ)")          self.trHienThi.heading("LoiNhuan", text="Lợi Nhuận (VNĐ)")          self.trHienThi.column("MaTivi", width=100, anchor="center")          self.trHienThi.column("TenTivi", width=250, anchor="center")          self.trHienThi.column("SoLuongBan", width=120, anchor="center")          self.trHienThi.column("TiLeBanChay", width=120, anchor="center")          self.trHienThi.column("DoanhThu", width=150, anchor="center")          self.trHienThi.column("LoiNhuan", width=150, anchor="center")          style = ttk.Style()          style.configure("Treeview.Heading", font=("Segoe UI", 11, "bold"))          style.configure("Treeview", font=("Segoe UI", 10), rowheight=28)          self.tivi\_dict = {}          self.load\_combo\_tivi()          self.load\_baocao\_all()      def load\_combo\_tivi(self):          try:              cursor = self.conn.cursor()              cursor.execute("SELECT MaTivi, TenTivi FROM Tivi ORDER BY TenTivi")              rows = cursor.fetchall()              ma\_list = ["-- Tất cả --"]              ten\_list = ["-- Tất cả --"]              for row in rows:                  mativi, tentivi = row                  ma\_list.append(mativi)                  ten\_list.append(tentivi)                  self.tivi\_dict[mativi] = tentivi                  self.tivi\_dict[tentivi] = mativi              self.cbo\_mativi["values"] = ma\_list              self.cbo\_tentivi["values"] = ten\_list              self.cbo\_mativi.current(0)              self.cbo\_tentivi.current(0)              cursor.close()          except Exception as e:              messagebox.showerror("Lỗi", f"Không thể tải danh sách tivi: {str(e)}")      def on\_mativi\_selected(self, event):          mativi = self.cbo\_mativi.get()          if mativi and mativi != "-- Tất cả --":              tentivi = self.tivi\_dict.get(mativi)              if tentivi:                  idx = list(self.cbo\_tentivi["values"]).index(tentivi)                  self.cbo\_tentivi.current(idx)      def on\_tentivi\_selected(self, event):          tentivi = self.cbo\_tentivi.get()          if tentivi and tentivi != "-- Tất cả --":              mativi = self.tivi\_dict.get(tentivi)              if mativi:                  idx = list(self.cbo\_mativi["values"]).index(mativi)                  self.cbo\_mativi.current(idx)      def load\_baocao\_all(self):          try:              for item in self.trHienThi.get\_children():                  self.trHienThi.delete(item)              cursor = self.conn.cursor()              query = """                  SELECT                      t.MaTivi,                      t.TenTivi,                      ISNULL(SUM(ct.SoLuong), 0) AS TongSoLuongBan,                      ISNULL(SUM(ct.ThanhTien), 0) AS TongDoanhThu,                      ISNULL(AVG(pn.DonGia), 0) AS GiaNhap                  FROM Tivi t                  JOIN ChiTietHoaDon ct ON t.MaTivi = ct.MaTivi                  JOIN HoaDonBan hd                      ON ct.MaHD = hd.MaHD                      AND hd.TrangThai = N'Đã thanh toán'                  LEFT JOIN (                      SELECT MaTivi, AVG(GiaNhap) AS DonGia                      FROM ChiTietPhieuNhap                      GROUP BY MaTivi                  ) pn                      ON t.MaTivi = pn.MaTivi                  GROUP BY t.MaTivi, t.TenTivi                  ORDER BY TongSoLuongBan DESC;"""              cursor.execute(query)              rows = cursor.fetchall()              tong\_soluong = sum(row[2] for row in rows)              for row in rows:                  mativi, tentivi, soluongban, doanhthu, gianhap = row                  tile = (soluongban / tong\_soluong \* 100) if tong\_soluong > 0 else 0                  loinhuan = (soluongban \* (doanhthu / soluongban - gianhap)) if soluongban > 0 else 0                  self.trHienThi.insert("", "end", values=(mativi, tentivi, soluongban, f"{tile:.2f}%", f"{doanhthu:,.0f}", f"{loinhuan:,.0f}"))              cursor.close()          except Exception as e:              messagebox.showerror("Lỗi", f"Không thể tải báo cáo: {str(e)}")      def xem\_baocao(self):          mativi = self.cbo\_mativi.get()          tentivi = self.cbo\_tentivi.get()          if mativi == "-- Tất cả --" or not mativi:              self.load\_baocao\_all()              return          try:              for item in self.trHienThi.get\_children():                  self.trHienThi.delete(item)              cursor = self.conn.cursor()              query\_tong = """                  SELECT ISNULL(SUM(SoLuong), 0)                  FROM ChiTietHoaDon"""                cursor.execute(query\_tong)              tong\_soluong = cursor.fetchone()[0]              query = """                  SELECT                      t.MaTivi,                      t.TenTivi,                      ISNULL(SUM(ct.SoLuong), 0) as TongSoLuongBan,                      ISNULL(SUM(ct.ThanhTien), 0) as TongDoanhThu,                      ISNULL(AVG(pn.DonGia), 0) as GiaNhap                  FROM Tivi t                  LEFT JOIN ChiTietHoaDon ct ON t.MaTivi = ct.MaTivi                  LEFT JOIN (                      SELECT MaTivi, AVG(GiaNhap) as DonGia                      FROM ChiTietPhieuNhap                      GROUP BY MaTivi                  ) pn ON t.MaTivi = pn.MaTivi                  WHERE t.MaTivi = ?                  GROUP BY t.MaTivi, t.TenTivi"""              cursor.execute(query, (mativi,))              row = cursor.fetchone()              if row:                  mativi, tentivi, soluongban, doanhthu, gianhap = row                  tile = (soluongban / tong\_soluong \* 100) if tong\_soluong > 0 else 0                  loinhuan = (soluongban \* (doanhthu / soluongban - gianhap) if soluongban > 0 else 0)                  self.trHienThi.insert("", "end", values=(mativi, tentivi, soluongban, f"{tile:.2f}%", f"{doanhthu:,.0f}", f"{loinhuan:,.0f}"))              else:                  messagebox.showinfo("Thông báo", "Không tìm thấy dữ liệu cho sản phẩm này!")              cursor.close()          except Exception as e:              messagebox.showerror("Lỗi", "Không thể xem báo cáo: " + str(e))      def huy\_xem\_bao\_cao(self):          self.load\_baocao\_all()          self.cbo\_mativi.delete(0, tk.END)          self.cbo\_mativi.set("-- Tất cả --")          self.cbo\_tentivi.delete(0, tk.END)          self.cbo\_tentivi.set("-- Tất cả --") |

**FORM HỆ THỐNG**

|  |
| --- |
| import tkinter as tk  from tkinter import ttk, messagebox  import pyodbc  import hashlib  PRIMARY\_COLOR = "#0D47A1"  SECONDARY\_COLOR = "#1565C0"  ACCENT\_COLOR = "#42A5F5"  HIGHLIGHT\_COLOR = "#BBDEFB"  TEXT\_COLOR = "white"  class HeThong(tk.Frame):      def \_\_init\_\_(self, parent, controller, conn, user):          super().\_\_init\_\_(parent, bg="white")          self.user = user          self.conn = conn          self.cursor = conn.cursor()          self.ds\_them = []          self.ds\_sua = []          self.ds\_xoa = []          lbl\_title = tk.Label(self, text="QUẢN LÝ TÀI KHOẢN HỆ THỐNG", font=("Segoe UI", 16, "bold"), bg="white", fg="#0D47A1")          lbl\_title.pack()          self.frame\_search = tk.Frame(self, bg="#E3F2FD", padx=10, pady=10)          self.frame\_search.pack(fill="x", padx=20, pady=5)          tk.Label(self.frame\_search, text="🔍 Tìm kiếm:", font=("Segoe UI", 10), bg="#E3F2FD").pack(side="left", padx=5)          self.txt\_timkiem = tk.Entry(self.frame\_search, font=("Segoe UI", 10), width=95)          self.txt\_timkiem.pack(side="left", padx=5)          self.txt\_timkiem.bind("<Return>", self.timkiem)          tk.Button(self.frame\_search, text="Tìm", font=("Segoe UI", 10, "bold"), bg="#1565C0", fg="white", bd=0, padx=10, pady=5, command=self.timkiem,).pack(side="left", padx=10)          tk.Button(self.frame\_search, text="Hủy", font=("Segoe UI", 10, "bold"), bg="#E53935", fg="white", bd=0, padx=10, pady=5, command=self.huy,).pack(side="left", padx=10)          frame\_form = tk.LabelFrame(self, text="Thông tin tài khoản", bg="white", font=("Segoe UI", 12, "bold"), fg="#0D47A1", padx=10, pady=10,)          frame\_form.pack(fill="x", padx=30, pady=10)          tk.Label(frame\_form, text="Tên đăng nhập:", bg="white", font=("Segoe UI", 10)).grid(row=0, column=0, sticky="w", pady=5, padx=5)          self.txt\_ten = ttk.Entry(frame\_form, font=("Segoe UI", 10), width=48)          self.txt\_ten.grid(row=0, column=1, pady=5, padx=5)          tk.Label(frame\_form, text="Mật khẩu:", bg="white", font=("Segoe UI", 10)).grid(row=0, column=2, sticky="w", pady=5, padx=5)          self.txt\_mk = ttk.Entry(frame\_form, font=("Segoe UI", 10), width=48, show="●")          self.txt\_mk.grid(row=0, column=3, pady=5, padx=5)          frame\_buttons = tk.Frame(self, bg="white")          frame\_buttons.pack(pady=10)          self.btn\_them = tk.Button(frame\_buttons, text="➕ Thêm", bg="#EBDA42", fg="white", font=("Segoe UI", 11, "bold"), padx=20, pady=5, bd=0, command=self.them,)          self.btn\_them.grid(row=0, column=0, padx=10)          btn\_sua = tk.Button(frame\_buttons, text="✏️ Sửa", bg="#FB8C00", fg="white", font=("Segoe UI", 11, "bold"), padx=20, pady=5, bd=0, command=self.sua,)          btn\_sua.grid(row=0, column=1, padx=10)          self.btn\_xoa = tk.Button(frame\_buttons, text="🗑️ Xóa", bg="#E53935", fg="white", font=("Segoe UI", 11, "bold"), padx=20, pady=5, bd=0, command=self.xoa,)          self.btn\_xoa.grid(row=0, column=2, padx=10)          btn\_lammoi = tk.Button(frame\_buttons, text="🔄 Làm mới", bg="#1E88E5", fg="white", font=("Segoe UI", 11, "bold"), padx=20, pady=5, bd=0, command=self.lammoi,)          btn\_lammoi.grid(row=0, column=3, padx=10)          btn\_luu = tk.Button(frame\_buttons, text="💾 Lưu", bg="#43A047", fg="white", font=("Segoe UI", 11, "bold"), padx=20, pady=5, bd=0, command=self.luu,)          btn\_luu.grid(row=0, column=4, padx=10)          self.frame\_table = tk.Frame(self, bg="white")          self.frame\_table.pack(fill="both", expand=True, padx=20, pady=10)          columns = ("TenDangNhap", "MatKhau")          # --- Tạo Scrollbar ---          scroll\_y = ttk.Scrollbar(self.frame\_table, orient="vertical")          scroll\_x = ttk.Scrollbar(self.frame\_table, orient="horizontal")          self.trHienThi = ttk.Treeview(self.frame\_table, show="headings", columns=columns, height=12, yscrollcommand=scroll\_y.set, xscrollcommand=scroll\_x.set,)          # --- Gắn Scrollbar ---          scroll\_y.config(command=self.trHienThi.yview)          scroll\_x.config(command=self.trHienThi.xview)          # --- Bố trí Scrollbar ---          scroll\_y.pack(side="right", fill="y")          scroll\_x.pack(side="bottom", fill="x")          self.trHienThi.pack(fill="both", expand=True)          self.trHienThi.heading("TenDangNhap", text="Tên đăng nhập")          self.trHienThi.heading("MatKhau", text="Mật khẩu")          self.trHienThi.column("TenDangNhap", width=400, anchor="center")          self.trHienThi.column("MatKhau", width=400, anchor="center")          style = ttk.Style()          style.configure("Treeview.Heading", font=("Segoe UI", 11, "bold"))          style.configure("Treeview", font=("Segoe UI", 10), rowheight=28)          self.trHienThi.bind("<<TreeviewSelect>>", self.chon\_dong)          self.load\_data()      def load\_data(self):          try:              for item in self.trHienThi.get\_children():                  self.trHienThi.delete(item)              if self.user != "admin":                  self.btn\_them.grid\_forget()                  self.btn\_xoa.grid\_forget()                  self.frame\_table.pack\_forget()                  self.frame\_search.pack\_forget()                  self.cursor.execute(                      "SELECT TenDangNhap, MatKhau FROM TaiKhoan WHERE TenDangNhap = ?",                      (self.user),                  )                  row = self.cursor.fetchone()                  if row:                      self.txt\_ten.insert(0, row[0])                      self.txt\_ten.config(state="disabled")                      self.txt\_mk.delete(0, tk.END)                      self.txt\_mk.insert(0, row[1])              else:                  self.cursor.execute("SELECT TenDangNhap, MatKhau FROM TaiKhoan")                  rows = self.cursor.fetchall()                  for row in rows:                      mat\_khau\_che = '●' \* 12                        self.trHienThi.insert("", "end", values=(row[0], mat\_khau\_che)) # <-- Đã che          except Exception as e:              messagebox.showerror("Lỗi", f"Không thể tải dữ liệu: {str(e)}")      def chon\_dong(self, event=None):          selected = self.trHienThi.selection()          if selected:              item = self.trHienThi.item(selected[0])              values = item["values"]              self.xoa\_form()              self.txt\_ten.insert(0, values[0])              try:                  self.cursor.execute(                      "SELECT MatKhau FROM TaiKhoan WHERE TenDangNhap = ?", (values[0],)                  )                  row = self.cursor.fetchone()                  if row:                      self.txt\_mk.insert(0, row[0])              except Exception as e:                  print(f"Lỗi load mật khẩu: {str(e)}")      def them(self):          ten = self.txt\_ten.get().strip()          mk = self.hash\_password(self.txt\_mk.get().strip())          if not ten:              messagebox.showwarning("Cảnh báo", "Vui lòng nhập Tên đăng nhập!")              return          if not mk:              messagebox.showwarning("Cảnh báo", "Vui lòng nhập Mật khẩu!")              return          for item in self.trHienThi.get\_children():              if self.trHienThi.item(item)["values"][0] == ten:                  messagebox.showwarning("Cảnh báo", f"Tên đăng nhập '{ten}' đã tồn tại!")                  return          self.trHienThi.insert("", "end", values=(ten, mk))          self.ds\_them.append((ten, mk))          self.xoa\_form()          messagebox.showinfo("Thêm tài khoản", "Thêm tài khoản thành công!")      def sua(self):          ten = self.txt\_ten.get().strip()          mk = self.hash\_password(self.txt\_mk.get().strip())          if not mk:              messagebox.showwarning("Cảnh báo", "Vui lòng nhập Mật khẩu mới!")              return          if self.user != "admin":              if ten != self.user:                   messagebox.showwarning("Cảnh báo", "Bạn chỉ có thể sửa tài khoản của chính mình!")                   return                self.ds\_sua = [(ten, mk, ten)]                messagebox.showinfo("Cập nhật tài khoản", "Mật khẩu đã được cập nhật tạm thời. Nhấn 'Lưu' để lưu vào CSDL.")              self.xoa\_form()              return            selected = self.trHienThi.selection()          if not selected:              messagebox.showwarning("Cảnh báo", "Vui lòng chọn dòng cần sửa!")              return          ten = self.txt\_ten.get().strip()          mk = self.hash\_password(self.txt\_mk.get().strip())          if not ten:              messagebox.showwarning("Cảnh báo", "Vui lòng nhập Tên đăng nhập!")              return          if not mk:              messagebox.showwarning("Cảnh báo", "Vui lòng nhập Mật khẩu!")              return          if self.user != "admin":              self.ds\_sua = [(mk, ten)]              return          item = self.trHienThi.item(selected[0])          ten\_cu = item["values"][0]          self.trHienThi.item(selected[0], values=(ten, mk))          is\_new = any(x[0] == ten\_cu for x in self.ds\_them)          if not is\_new:              self.ds\_sua = [x for x in self.ds\_sua if x[0] != ten\_cu]              self.ds\_sua.append((ten, mk, ten\_cu))          else:              self.ds\_them = [(ten, mk) if x[0] == ten\_cu else x for x in self.ds\_them]          self.xoa\_form()          messagebox.showinfo("Cập nhật tài khoản", "Cập nhật tài khoản thành công!")      def xoa(self):          selected = self.trHienThi.selection()          if not selected:              messagebox.showwarning("Cảnh báo", "Vui lòng chọn dòng cần xóa!")              return          xacnhan = messagebox.askyesno("Xác nhận", "Bạn có chắc muốn xóa không?")          if not xacnhan:              return          item = self.trHienThi.item(selected[0])          ten = item["values"][0]          self.trHienThi.delete(selected[0])          is\_new = any(x[0] == ten for x in self.ds\_them)          if is\_new:              self.ds\_them = [x for x in self.ds\_them if x[0] != ten]          else:              if ten not in self.ds\_xoa:                  self.ds\_xoa.append(ten)          self.xoa\_form()          messagebox.showinfo("Xóa tài khoản", "Xóa tài khoản thành công!")      def luu(self):          try:              if not self.ds\_them and not self.ds\_sua and not self.ds\_xoa:                  messagebox.showinfo("Thông báo", "Không có thay đổi để lưu!")                  return              xacnhan = messagebox.askyesno("Xác nhận", "Bạn có chắc muốn lưu không?")              if not xacnhan:                  return              for ten in self.ds\_xoa:                  self.cursor.execute(                      "DELETE FROM TaiKhoan WHERE TenDangNhap = ?", (ten,)                  )              for ten, mk in self.ds\_them:                  self.cursor.execute(                      "INSERT INTO TaiKhoan (TenDangNhap, MatKhau) VALUES (?, ?)",                      (ten, mk),                  )              for ten, mk, ten\_cu in self.ds\_sua:                  if ten == ten\_cu:                      self.cursor.execute(                          "UPDATE TaiKhoan SET MatKhau=? WHERE TenDangNhap=?",                          (mk, ten\_cu),                      )                  else:                      self.cursor.execute(                          "UPDATE TaiKhoan SET TenDangNhap=?, MatKhau=? WHERE TenDangNhap=?",                          (ten, mk, ten\_cu),                      )              self.conn.commit()              messagebox.showinfo("Thành công", "Đã lưu thay đổi vào CSDL!")          except pyodbc.IntegrityError as e:              self.conn.rollback()              messagebox.showerror("Lỗi", f"Lỗi ràng buộc dữ liệu: {str(e)}")          except Exception as e:              self.conn.rollback()              messagebox.showerror("Lỗi", f"Không thể lưu dữ liệu: {str(e)}")          self.load\_data()          self.xoa\_form()          self.ds\_them.clear()          self.ds\_sua.clear()          self.ds\_xoa.clear()      def lammoi(self):          confirm = messagebox.askyesno("Xác nhận", "Bạn có chắc muốn hủy các thay đổi?")          if not confirm:              return          self.ds\_them.clear()          self.ds\_sua.clear()          self.ds\_xoa.clear()          self.load\_data()          self.xoa\_form()          messagebox.showinfo("Thông báo", "Đã làm mới dữ liệu!")      def timkiem(self):          tu\_khoa\_tim = self.txt\_timkiem.get().strip()          if not tu\_khoa\_tim:              messagebox.showinfo("Thông báo", "Vui lòng nhập từ khóa tìm kiếm.")              self.load\_data()              return          try:              for item in self.trHienThi.get\_children():                  self.trHienThi.delete(item)              self.cursor.execute(                  "SELECT TenDangNhap, MatKhau FROM TaiKhoan WHERE TenDangNhap LIKE ?",                  (f"%{tu\_khoa\_tim}%",),              )              rows = self.cursor.fetchall()              for row in rows:                  self.trHienThi.insert("", "end", values=(row[0], len(row[1])))              if not rows:                  messagebox.showinfo("Kết quả", "Không tìm thấy dữ liệu!")          except Exception as e:              messagebox.showerror("Lỗi", "Không thể tải dữ liệu: " + str(e))      def xoa\_form(self):          self.txt\_ten.delete(0, tk.END)          if self.user == "admin":              self.txt\_mk.delete(0, tk.END)      def huy(self):          self.txt\_timkiem.delete(0, tk.END)          self.load\_data()      def hash\_password(self, password):          if not password:              return ""          return hashlib.sha256(password.encode('utf-8')).hexdigest() |

## **DANH MỤC TÀI LIỆU THAM KHẢO**

Lê Quang Phúc (2021). *Hướng dẫn lập trình kết nối cơ sở dữ liệu SQL Server*. Nhà xuất bản Thông tin & Truyền thông.

Huỳnh Phước Hải, Nguyễn Văn Hòa (2022), *Giáo trình Lập trình Python cơ bản*, NXB ĐHQG TPHCM

Hoàng Văn Minh (2019). *Xây dựng ứng dụng giao diện với Tkinter trong Python*. Nhà xuất bản Đại học Quốc gia TP.HCM.

Nguyễn Thị Kim Anh (2022). *Xử lý giao diện và quản lý dữ liệu trong Python*. Nhà xuất bản Bách Khoa Hà Nội.

Phạm Minh Hoàng (2023). *Ứng dụng Python trong xây dựng phần mềm quản lý*. Nhà xuất bản Lao động – Xã hội.