**BỘ** **CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

---------------------------



**XÂY DỰNG PHẦN MỀM QUẢN LÝ PHÒNG KHÁM TIÊM CHỦNG**

**SINH VIÊN THỰC HIỆN**

1. 2001224963, Nguyễn Hoàng Thịnh, Lớp 13DHTH05

2. 2001220573, Võ Trường Danh, Lớp 13DHTH01

3. 2001224207, Trần Tấn Tài, Lớp 13DHTH05

**GVHD:** ThS. Nguyễn Thị Thanh Thủy

**BÁO CÁO BÀI TẬP LỚN**

Môn học: Hệ quản trị cơ sở dữ liệu

**BẢNG PHÂN CHIA CÔNG VIỆC CHO TỪNG THÀNH VIÊN NHÓM**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên Sinh Viên** | **Mã Số Sinh Viên** | **Nội Dung Thực Hiện** | **Đánh Giá** |
| Nguyễn Hoàng Thịnh | 2001224963 | Xây dựng ,phát triển chức năng xử lý dữ liệu và giao diện người dùng cho các chức năng liên quan đến các bảng Đăng nhập, Đăng ký, Quản lý Bệnh nhân, Quản lý Nhân viên và Quản lý Tài khoản,Quản lý hoá đơn , Quản lý lịch tiêm . Thiết kế và triển khai giao diện đăng nhập và đăng ký | 100% |
| Võ Trường Danh | 2001220573 | Xây dựng , phát triển chức năng xử lý dữ liệu cho các chức năng liên quan đến các bảng Vaccine, Loại Vaccine, Nhà cung cấp, Nhập vaccine và Chi tiết nhập vaccine , Quản lý Bệnh nhân, Quản lý Nhân viên. Thiết kế và xây dựng cơ sở dữ liệu. Phụ trách thiết kế giao diện chung của hệ thống. Hỗ trợ nội dung word | 100% |
| Trần Tấn Tài | 2001224207 | Nghiên cứu phạm vi đề tài và khảo sát nghiệp vụ. Phát triển chức năng xử lý dữ liệu cho các chức năng liên quan đến các bảng Hóa đơn, Chi tiết hóa đơn, Lịch tiêm, Chi tiết lịch tiêm và Ghi nhận tiêm chủng,Quản lý Bệnh nhân, | 100% |

MỤC LỤC

[LỜI CẢM ƠN iv](#_Toc184619027)

[MỞ ĐẦU 1](#_Toc184619028)

[CHƯƠNG 1. GIỚI THIỆU 2](#_Toc184619029)

[I. Tổng quan 2](#_Toc184619030)

[II. Phạm vi đề tài 2](#_Toc184619031)

[III. Mục tiêu, sự cần thiết của đề tài, tại sao lại chọn đề tài này 3](#_Toc184619032)

[1. Mục tiêu 3](#_Toc184619033)

[2. Sự cần thiết 3](#_Toc184619034)

[3. Tại sao lại chọn đề tài này 3](#_Toc184619035)

[CHƯƠNG 2. THIẾT KẾ VÀ CÀI ĐẶT CƠ SỞ DỮ LIỆU 4](#_Toc184619036)

[2.1. Xây dựng cơ sở dữ liệu 4](#_Toc184619037)

[2.2. Cài đặt yêu cầu xử lý 6](#_Toc184619038)

[2.2.1. Trigger 6](#_Toc184619039)

[2.2.2. Procedure 9](#_Toc184619040)

[2.2.3. Function 18](#_Toc184619041)

[2.3. Quản trị người dùng 19](#_Toc184619042)

[CHƯƠNG 3. XÂY DỰNG ỨNG DỤNG 20](#_Toc184619043)

[1. Chức năng danh mục 20](#_Toc184619044)

[2. Chức năng thống kê 22](#_Toc184619045)

[3. Chức năng tìm kiếm 22](#_Toc184619046)

[4. Chức năng nghiệp vụ 23](#_Toc184619047)

[5. Cài đặt chức năng phân quyền người dùng 25](#_Toc184619048)

[TÀI LIỆU THAM KHẢO 26](#_Toc184619049)

**MỤC LỤC HÌNH ẢNH**

[Hình 1 Mô hình quan hệ 15](#_Toc184180429)

[Hình 2 Giao diện chính 16](#_Toc184180430)

[Hình 3 Quản lý bệnh nhân 18](#_Toc184180431)

[Hình 4 Đăng ký chích Vaccine 19](#_Toc184180432)

[Hình 5 Quản lý Vaccine 20](#_Toc184180433)

[Hình 6 Nhập Hàng 21](#_Toc184180434)

[Hình 7 Quản lý loại Vaccine vs Nhà Cung Cấp 22](#_Toc184180435)

[Hình 8 Quản Lý Nhân Viên 23](#_Toc184180436)

[Hình 9 Thống kê 24](#_Toc184180437)

# LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành đến thầy đã tận tình giảng dạy, truyền đạt kiến thức và hướng dẫn chúng em trong suốt quá trình học tập và thực hiện tiểu luận này. Đặc biệt, chúng em xin bày tỏ lòng biết ơn sâu sắc đến thầy người đã trực tiếp hướng dẫn, chỉ bảo và cung cấp những kiến thức quý báu giúp chúng em hoàn thành bài tiểu luận " Xây dựng phần mềm quản lý phòng khám tiêm chủng ". Sự nhiệt tình, tâm huyết và những đóng góp quý giá ấy đã giúp chúng em vượt qua những khó khăn, đồng thời tạo nên động lực lớn để chúng em hoàn thiện nghiên cứu của mình.

Thành phố Hồ Chí Minh, ngày tháng năm 2024

Sinh viên thực hiện

(Ký và ghi rõ họ tên)

# 

# MỞ ĐẦU

Trong bối cảnh hiện nay, việc tiêm chủng đóng vai trò quan trọng trong việc bảo vệ sức khỏe cộng đồng, đặc biệt trong việc phòng chống các dịch bệnh. Đồ án "Quản lý phòng khám tiêm chủng" được phát triển nhằm cung cấp một hệ thống quản lý toàn diện, giúp các phòng khám tiêm chủng nâng cao hiệu quả hoạt động và cải thiện chất lượng dịch vụ chăm sóc sức khỏe.

Với sự gia tăng nhu cầu tiêm chủng trong cộng đồng, hệ thống này không chỉ giúp tổ chức quy trình tiêm chủng một cách khoa học và hiệu quả mà còn đảm bảo thông tin được cập nhật và lưu trữ một cách chính xác. Hệ thống sẽ bao gồm các chức năng quan trọng như quản lý thông tin bệnh nhân, nhân viên, vaccine, theo dõi lịch tiêm và ghi nhận quá trình tiêm chủng.

Mục tiêu chính của đồ án là xây dựng một cơ sở dữ liệu mạnh mẽ, hỗ trợ các nghiệp vụ quản lý liên quan đến tiêm chủng, từ việc quản lý kho vaccine, theo dõi lịch sử tiêm chủng cho đến việc lập hóa đơn và ghi nhận thông tin y tế của bệnh nhân.

Ngoài ra, hệ thống còn đáp ứng yêu cầu về tự động hóa quy trình, đảm bảo tính nhất quán của dữ liệu thông qua việc sử dụng các triggers và stored procedures trong cơ sở dữ liệu. Việc này không chỉ giúp tiết kiệm thời gian mà còn giảm thiểu sai sót trong quá trình quản lý.

Chúng tôi tin rằng đồ án này sẽ góp phần quan trọng trong việc nâng cao chất lượng dịch vụ y tế tại các phòng khám tiêm chủng, đồng thời tạo điều kiện thuận lợi cho bệnh nhân trong việc tiếp cận dịch vụ y tế cần thiết. Hệ thống quản lý phòng khám tiêm chủng không chỉ đơn thuần là một công cụ mà còn là một giải pháp toàn diện để tối ưu hóa quy trình làm việc và nâng cao trải nghiệm của bệnh nhân trong dịch vụ chăm sóc sức khỏe.

# CHƯƠNG 1. GIỚI THIỆU

## Tổng quan

Hiện nay, nhu cầu quản lý thông tin tiêm chủng tại các phòng khám ngày càng trở nên cấp thiết trong bối cảnh dịch bệnh và sự gia tăng nhu cầu tiêm phòng. Các phòng khám đối mặt với nhiều thách thức trong việc duy trì quy trình tổ chức, đảm bảo tính chính xác, và tối ưu hóa hiệu suất hoạt động. Sự thiếu hụt một hệ thống quản lý chuyên nghiệp dễ dẫn đến các vấn đề như sai sót trong quản lý hồ sơ, khó khăn trong theo dõi lịch tiêm, và việc kiểm soát kho vaccine kém hiệu quả.

Để giải quyết vấn đề này, đồ án "Quản lý phòng khám tiêm chủng" được xây dựng nhằm cung cấp một giải pháp toàn diện, đáp ứng nhu cầu quản lý thông tin bệnh nhân, nhân viên, vaccine, lịch tiêm, và các dịch vụ tiêm chủng. Hệ thống còn hỗ trợ quản lý kho vaccine, kiểm soát phiếu nhập, hóa đơn, và ghi nhận đầy đủ quá trình tiêm chủng, qua đó nâng cao hiệu quả vận hành và chất lượng dịch vụ của phòng khám.

Với mục tiêu cải thiện trải nghiệm cho bệnh nhân và tăng cường hiệu quả hoạt động của phòng khám, hệ thống này mang lại một phương thức quản lý thông minh, góp phần vào việc nâng cao chất lượng chăm sóc sức khỏe cộng đồng.

## Phạm vi đề tài

Quản lý thông tin bệnh nhân: Hệ thống cho phép đăng ký, cập nhật thông tin và theo dõi lịch sử tiêm chủng của bệnh nhân, đảm bảo quản lý dữ liệu bệnh nhân chính xác và dễ dàng tra cứu.

Quản lý nhân viên: Lưu trữ thông tin cá nhân của nhân viên phòng khám, đồng thời phân quyền truy cập hệ thống dựa trên vai trò và trách nhiệm của từng nhân viên, giúp đảm bảo tính bảo mật và phân công công việc hiệu quả.

Quản lý vaccine: Theo dõi số lượng tồn kho, ngày sản xuất, hạn sử dụng và các thông tin liên quan đến vaccine, giúp phòng khám kiểm soát kho vaccine một cách hiệu quả và tránh lãng phí.

Quản lý lịch tiêm: Hệ thống hỗ trợ việc đặt lịch hẹn, theo dõi trạng thái tiêm chủng và ghi nhận chi tiết quá trình tiêm chủng của bệnh nhân, từ đó nâng cao hiệu quả công tác tiêm phòng.

Quản lý hóa đơn và thanh toán: Tạo, lưu trữ hóa đơn và thực hiện các quy trình thanh toán cho bệnh nhân, giúp phòng khám kiểm soát tài chính và hỗ trợ bệnh nhân thanh toán nhanh chóng, dễ dàng.

Báo cáo và thống kê: Cung cấp các báo cáo chi tiết về hoạt động tiêm chủng và tình hình kho vaccine, giúp quản lý phòng khám nắm bắt được tình hình và đưa ra quyết định kịp thời, chính xác.

Phần mềm sẽ không bao gồm các chức năng liên quan đến marketing, quản lý kho hay tích hợp với các hệ thống bên ngoài. Phạm vi phát triển chỉ tập trung vào việc xây dựng một công cụ quản lý cho các khách sạn vừa và nhỏ, giúp tối ưu hóa quy trình làm việc và nâng cao hiệu quả công tác quản lý.

## Mục tiêu, sự cần thiết của đề tài, tại sao lại chọn đề tài này

### 1. Mục tiêu

Xây dựng một hệ thống quản lý tiêm chủng điện tử giúp tự động hóa các quy trình quản lý thông tin tiêm chủng, từ đăng ký, lịch tiêm, theo dõi đến báo cáo thống kê.

Cung cấp một giải pháp tiện lợi và hiệu quả cho việc quản lý thông tin bệnh nhân, nhân viên, vaccine và các quy trình liên quan đến tiêm chủng.

Nâng cao hiệu suất công việc, giảm thiểu sai sót và tối ưu hóa quy trình tiêm chủng.

Đảm bảo tính bảo mật và toàn vẹn dữ liệu tiêm chủng.

### 2. Sự cần thiết

Đáp ứng nhu cầu thực tế: Trong bối cảnh dịch bệnh, nhu cầu tiêm phòng gia tăng, hệ thống quản lý thông tin sẽ giúp các phòng khám hoạt động hiệu quả hơn.

Đảm bảo an toàn cho bệnh nhân: Hệ thống giúp ghi nhận chi tiết quá trình tiêm chủng, đảm bảo bệnh nhân được theo dõi sức khỏe đầy đủ.

Tối ưu hóa quy trình: Giảm thiểu thủ tục giấy tờ, giúp nhân viên và bệnh nhân tiết kiệm thời gian và công sức.

### 3. Tại sao lại chọn đề tài này

Tính ứng dụng cao: Đề tài hướng tới giải quyết một vấn đề thực tiễn, có tính ứng dụng cao trong lĩnh vực y tế.

Phù hợp với xu hướng chuyển đổi số: Xây dựng hệ thống quản lý tiêm chủng điện tử góp phần vào quá trình chuyển đổi số trong ngành y tế.

Tính khả thi: Đề tài có thể được thực hiện trong phạm vi thời gian và nguồn lực cho phép.

Mở rộng kiến thức và kỹ năng: Thực hiện đề tài giúp sinh viên áp dụng kiến thức đã học vào thực tế, phát triển kỹ năng lập trình, quản lý dự án và làm việc nhóm.

# CHƯƠNG 2. THIẾT KẾ VÀ CÀI ĐẶT CƠ SỞ DỮ LIỆU

## 2.1. Xây dựng cơ sở dữ liệu

Bảng `BENHNHAN`

- MaBN: Mã bệnh nhân (khóa chính).

- HoTen: Họ và tên bệnh nhân.

- NgaySinh: Ngày sinh.

- GioiTinh: Giới tính (Nam/Nữ).

- DiaChi: Địa chỉ.

- SoDienThoai: Số điện thoại (ràng buộc check 10 ký tự, chỉ chứa số).

Bảng `NHANVIEN`

- MaNV: Mã nhân viên (khóa chính).

- HoTen: Họ và tên nhân viên.

- GioiTinh: Giới tính (Nam/Nữ).

- ChucVu: Chức vụ.

- DiaChi: Địa chỉ.

- SoDienThoai: Số điện thoại (ràng buộc check 10 ký tự, chỉ chứa số).

- Pass: Mật khẩu (ràng buộc default: `123`).

- Quyen: Quyền truy cập (kiểu int).

Bảng `NHACUNGCAP`

- MaNCC: Mã nhà cung cấp (khóa chính).

- TenNCC: Tên nhà cung cấp.

- DiaChi: Địa chỉ nhà cung cấp.

- SoDienThoai: Số điện thoại (ràng buộc check 10 ký tự, chỉ chứa số).

Bảng `LOAIBENH`

- MaLoai: Mã loại bệnh (khóa chính).

- TenLoai: Tên loại bệnh.

- SoMui: Số mũi tiêm (ràng buộc default: `1`).

Bảng `VACCINE`

- MaVC: Mã vaccine (khóa chính).

- MaLoai: Mã loại bệnh (khóa ngoại liên kết tới bảng `LOAIBENH`).

- TenVC: Tên vaccine.

- XuatXu: Xuất xứ (khóa ngoại liên kết tới bảng `Chưa Xác Định`).

- Gia: Giá vaccine (khóa ngoại liên kết tới bảng `0`).

- SoLuongTon: Số lượng tồn (ràng buộc check >= 0, khóa ngoại liên kết tới bảng `0`).

Bảng `LOVACCINE`

- MaLo: Mã lô (khóa chính).

- MaVC: Mã vaccine (khóa ngoại liên kết tới bảng `VACCINE`).

- NgaySX: Ngày sản xuất.

- HanSuDung: Hạn sử dụng.

- SoLuong: Số lượng trong lô (ràng buộc check >= 0).

Bảng `PHIEUNHAP`

- MaPN: Mã phiếu nhập (khóa chính).

- MaNV: Mã nhân viên (khóa ngoại liên kết tới bảng `NHANVIEN`).

- NgayNhap: Ngày nhập (khóa ngoại liên kết tới bảng ngày hiện tại).

- MaNCC: Mã nhà cung cấp (khóa ngoại liên kết tới bảng `NHACUNGCAP`).

- TongTien: Tổng tiền (khóa ngoại liên kết tới bảng `0`).

Bảng `CHITIETPHIEUNHAP`

- MaPN: Mã phiếu nhập (khóa ngoại liên kết tới bảng `PHIEUNHAP`).

- MaVC: Mã vaccine (khóa ngoại liên kết tới bảng `VACCINE`).

- MaLo: Mã lô (khóa ngoại liên kết tới bảng `LOVACCINE`).

- SoLuong: Số lượng nhập.

- DonGia: Đơn giá nhập.

Bảng `HOADON`

- MaHD: Mã hóa đơn (khóa chính).

- NgayLap: Ngày lập hóa đơn (khóa ngoại liên kết tới bảng ngày hiện tại).

- MaBN: Mã bệnh nhân (khóa ngoại liên kết tới bảng `BENHNHAN`).

- MaNV: Mã nhân viên (khóa ngoại liên kết tới bảng `NHANVIEN`).

- TongTien: Tổng tiền (ràng buộc check >= 0, khóa ngoại liên kết tới bảng `0`).

Bảng `CHITIETHOADON`

- MaHD: Mã hóa đơn (khóa ngoại liên kết tới bảng `HOADON`).

- MaVC: Mã vaccine (khóa ngoại liên kết tới bảng `VACCINE`).

- SoLuong: Số lượng vaccine.

- DonGia: Đơn giá.

Bảng `LICHTIEM`

- MaLT: Mã lịch tiêm (khóa chính).

- MaHD: Mã hóa đơn (khóa ngoại liên kết tới bảng `HOADON`).

- MaBN: Mã bệnh nhân (khóa ngoại liên kết tới bảng `BENHNHAN`).

- MaVC: Mã vaccine (khóa ngoại liên kết tới bảng `VACCINE`).

- NgayHenTiem: Ngày hẹn tiêm.

- TrangThai: Trạng thái.

Bảng `GHINHANTIEMCHUNG`

- MaBN: Mã bệnh nhân (khóa ngoại liên kết tới bảng `BENHNHAN`).

- MaVC: Mã vaccine (khóa ngoại liên kết tới bảng `VACCINE`).

- MaNV: Mã nhân viên (khóa ngoại liên kết tới bảng `NHANVIEN`).

- MaLT: Mã lịch tiêm (khóa ngoại liên kết tới bảng `LICHTIEM`).

- NgayTiem: Ngày tiêm.

- TinhTrangSucKhoe: Tình trạng sức khỏe sau tiêm.

- MaLo: Mã lô vaccine (khóa ngoại liên kết tới bảng `LOVACCINE`).

A computer screen shot of a computer

Description automatically generated

Hình 1.1. Lược đồ diagram

## 2.2. Cài đặt yêu cầu xử lý

### 2.2.1. Trigger

**a. Phần bài làm của Nguyễn Hoàng Thịnh**

* ***Tự động cập nhật giá tiền khi thêm trong CHITIETHOADON từ VACCINE***

GO

CREATE TRIGGER TG\_LAYDONGIACHOCTHD

ON CHITIETHOADON

FOR INSERT,UPDATE

AS

BEGIN

UPDATE CHITIETHOADON

SET CHITIETHOADON.DONGIA = VC.Gia

FROM VACCINE VC

WHERE VC.MaVC = CHITIETHOADON.MaVC

END

* ***Tự tính tổng tiền hoá đơn khi thêm CHITIETHOADON***

GO

CREATE TRIGGER TG\_TINHTONGTIEN

ON CHITIETHOADON

FOR INSERT, UPDATE

AS

BEGIN

UPDATE HOADON

SET TongTien = (

SELECT SUM(SOLUONG \* DONGIA)

FROM CHITIETHOADON

WHERE MaHD IN (SELECT MaHD FROM inserted)

)

WHERE MaHD IN (SELECT MaHD FROM inserted);

END

GO

* ***Tự động thêm LICHTIEM khi thêm CHITIETHOADON***

CREATE TRIGGER TG\_THEMLICHTIEM

ON CHITIETHOADON

FOR INSERT

AS

BEGIN

DECLARE @NewMaLT CHAR(5);

-- Tạo mã mới LT

SELECT @NewMaLT = 'LT' + RIGHT('000' + CAST(ISNULL(MAX(CAST(SUBSTRING(MaLT, 3, 3) AS INT)), 0) + 1 AS VARCHAR(3)), 3)

FROM LICHTIEM;

-- Tạo lịch tiêm cho từng chi tiết hóa đơn

INSERT INTO LICHTIEM (MaLT, MaHD, MaBN, MaVC, NgayHenTiem, TrangThai)

SELECT

@NewMaLT,

i.MaHD,

h.MaBN,

i.MaVC,

GETDATE(),

N'Chưa tiêm'

FROM

INSERTED i

JOIN

HOADON h ON i.MaHD = h.MaHD;

END;

GO

* ***Trừ tồn kho khi thêm CHITIETHOADON***

CREATE TRIGGER TG\_TRUTONKHOCHITIETHOADON

ON CHITIETHOADON

AFTER INSERT

AS

BEGIN

UPDATE V

SET SoLuongTon = V.SoLuongTon - I.SoLuong

FROM VACCINE V

JOIN inserted I ON V.MaVC = I.MaVC;

END;

**b. Phần bài làm của Võ Trường Danh**

* ***Chuyển trạng thái lịch tiêm thành đã tiêm khi thêm 1 dòng trong*** ***GHINHANTIENCHUNG***

GO

CREATE TRIGGER TG\_CAPNHAPTRANGTHAILICHTIEM

ON GHINHANTIEMCHUNG

FOR INSERT

AS

BEGIN

IF NOT EXISTS (SELECT \* FROM inserted i JOIN LICHTIEM LT ON LT.MaLT = I.MaLT WHERE i.MaVC = LT.MaVC AND LT.MaBN =i.MaBN AND LT.MaVC =I.MaVC)

BEGIN

PRINT N'Dữ liệu không hợp lệ! Không khớp với lịch tiêm.';

rollback tran

END;

UPDATE LICHTIEM

SET TrangThai = N'Đã tiêm'

WHERE MaLT = (SELECT MaLT FROM inserted) AND MaVC = (SELECT MaVC FROM inserted ) AND MaBN = (SELECT MaBN FROM inserted)

END

* ***Tính tổng tiền cho phiếu nhập***

GO

CREATE TRIGGER TG\_TinhTongTienPN

ON CHITIETPHIEUNHAP

FOR INSERT,UPDATE

AS

BEGIN

UPDATE PHIEUNHAP

SET TongTien = (

SELECT SUM(SOLUONG \* DONGIA)

FROM CHITIETPHIEUNHAP

WHERE MaPN IN (SELECT MaPN FROM inserted)

)

WHERE MaPN IN (SELECT MaPN FROM inserted);

END

* ***Cộng số lượng tồn trong VACCINE***

CREATE TRIGGER TG\_CONGSOLUONGTONCUAVACCINE

ON CHITIETPHIEUNHAP

FOR INSERT

AS

BEGIN

UPDATE V

SET SoLuongTon = ISNULL((SELECT SUM(CT.SoLuong)

FROM CHITIETPHIEUNHAP CT

WHERE CT.MaVC = I.MaVC), 0)

FROM VACCINE V

JOIN inserted I ON V.MaVC = I.MaVC;

END

GO

**c. Phần bài làm của Trần Tấn Tài**

### 2.2.2. Procedure

**a. Phần bài làm của Nguyễn Hoàng Thịnh**

* ***Thêm BENHNHAN***

CREATE PROCEDURE SP\_ThemBenhNhan

@HoTen NVARCHAR(50),

@NgaySinh DATE,

@GioiTinh NVARCHAR(3),

@DiaChi NVARCHAR(90),

@SoDienThoai CHAR(10)

AS

BEGIN

BEGIN TRY

-- Tự động tạo mã bệnh nhân

DECLARE @MaBN CHAR(5);

SELECT @MaBN = 'BN' + RIGHT('000' + CAST(ISNULL(MAX(CAST(SUBSTRING(MaBN, 3, 3) AS INT)), 0) + 1 AS NVARCHAR), 3)

FROM BENHNHAN;

INSERT INTO BENHNHAN (MaBN, HoTen, NgaySinh, GioiTinh, DiaChi, SoDienThoai)

VALUES (@MaBN, @HoTen, @NgaySinh, @GioiTinh, @DiaChi, @SoDienThoai);

PRINT N'Thêm bệnh nhân thành công với mã ' + @MaBN;

END TRY

BEGIN CATCH

PRINT N'Có lỗi xảy ra khi thêm bệnh nhân: ' + ERROR\_MESSAGE();

END CATCH

END;

* ***Xóa BENHNHAN***

CREATE PROCEDURE SP\_XoaBenhNhan

@MaBN CHAR(5)

AS

BEGIN

BEGIN TRANSACTION;

IF EXISTS (SELECT 1 FROM GHINHANTIEMCHUNG WHERE MaBN = @MaBN)

BEGIN

DELETE FROM GHINHANTIEMCHUNG WHERE MaBN = @MaBN;

END

IF EXISTS (SELECT 1 FROM LICHTIEM WHERE MaBN = @MaBN)

BEGIN

DELETE FROM LICHTIEM WHERE MaBN = @MaBN;

END

IF EXISTS (SELECT 1 FROM HOADON WHERE MaBN = @MaBN)

BEGIN

DELETE FROM CHITIETHOADON

WHERE MaHD IN (SELECT MaHD FROM HOADON WHERE MaBN = @MaBN);

END

IF EXISTS (SELECT 1 FROM HOADON WHERE MaBN = @MaBN)

BEGIN

DELETE FROM HOADON WHERE MaBN = @MaBN;

END

DELETE FROM BENHNHAN WHERE MaBN = @MaBN;

COMMIT TRANSACTION;

END;

GO

* ***Cập nhật thông tin BENHNHAN***

CREATE PROCEDURE SP\_SuaBenhNhan

@MaBN CHAR(5),

@HoTen NVARCHAR(50) = NULL,

@NgaySinh DATE = NULL,

@GioiTinh NVARCHAR(3) = NULL,

@DiaChi NVARCHAR(90) = NULL,

@SoDienThoai CHAR(10) = NULL

AS

BEGIN

BEGIN TRY

IF EXISTS (SELECT 1 FROM BENHNHAN WHERE MaBN = @MaBN)

BEGIN

IF @HoTen IS NOT NULL

UPDATE BENHNHAN SET HoTen = @HoTen WHERE MaBN = @MaBN;

IF @NgaySinh IS NOT NULL

UPDATE BENHNHAN SET NgaySinh = @NgaySinh WHERE MaBN = @MaBN;

IF @GioiTinh IS NOT NULL

UPDATE BENHNHAN SET GioiTinh = @GioiTinh WHERE MaBN = @MaBN;

IF @DiaChi IS NOT NULL

UPDATE BENHNHAN SET DiaChi = @DiaChi WHERE MaBN = @MaBN;

IF @SoDienThoai IS NOT NULL

UPDATE BENHNHAN SET SoDienThoai = @SoDienThoai WHERE MaBN = @MaBN;

PRINT N'Sửa thông tin bệnh nhân thành công!';

END

ELSE

BEGIN

PRINT N'Mã bệnh nhân không tồn tại!';

END

END TRY

BEGIN CATCH

PRINT N'Có lỗi xảy ra khi sửa thông tin bệnh nhân: ' + ERROR\_MESSAGE();

END CATCH

END;

**b. Phần bài làm của Võ Trường Danh**

- **Cập nhật lại dử liệu số lượng tồn**

GO

CREATE PROCEDURE UpdateSoLuongTonVaccine

AS

BEGIN

DECLARE @MaVC NVARCHAR(50);

DECLARE CURSOR\_VACCINE CURSOR FOR

SELECT MaVC FROM VACCINE;

OPEN CURSOR\_VACCINE;

FETCH NEXT FROM CURSOR\_VACCINE INTO @MaVC;

WHILE @@FETCH\_STATUS = 0

BEGIN

UPDATE VACCINE

SET SoLuongTon =

ISNULL((SELECT SUM(CT.SoLuong)

FROM CHITIETPHIEUNHAP CT

WHERE CT.MaVC = @MaVC), 0) -

ISNULL((SELECT SUM(CT.SoLuong)

FROM CHITIETHOADON CT

WHERE CT.MaVC = @MaVC), 0)

WHERE MaVC = @MaVC;

FETCH NEXT FROM CURSOR\_VACCINE INTO @MaVC;

END

CLOSE CURSOR\_VACCINE;

DEALLOCATE CURSOR\_VACCINE;

END;

GO

* ***Thống kê số tiền đã nhập VACCINE theo tháng***

CREATE PROC pro\_thongkenhaphang AS

BEGIN

SELECT CONCAT(MONTH(NGAYNHAP), '/', YEAR(NGAYNHAP)) AS THANGNAM, SUM(TONGTIEN) AS TONGTIEN

FROM PHIEUNHAP

GROUP BY YEAR(NGAYNHAP), MONTH(NGAYNHAP)

ORDER BY YEAR(NGAYNHAP), MONTH(NGAYNHAP)

END;

* ***Thống kê số lượng VACCINE đã được tiêm theo loại***

CREATE PROC pro\_thongkesoluongtheoloai AS

BEGIN

SELECT LVC.TENLOAI , SUM(CTHD.SOLUONG) AS SOLUONG FROM VACCINE

JOIN CHITIETHOADON CTHD ON CTHD.MAVC = VACCINE.MAVC

JOIN LOAIBENH LVC ON LVC.MALOAI = VACCINE.MALOAI

GROUP BY LVC.TENLOAI

END;

* ***Kiểm tra tình trạng tồn kho***

CREATE PROC pro\_tinhtrangvacine @MAVC CHAR(6) AS

BEGIN

DECLARE @SOLUONGTON INT;

SELECT @SOLUONGTON = ISNULL(SOLUONGTON,0)

FROM VACCINE

WHERE MAVC = @MAVC

IF (@SOLUONGTON < 2)

BEGIN

PRINT CONCAT(N'VACCINE CỦA BẠN GẦN NHƯ ĐÃ HẾT, HÃY NHẬP HÀNG ĐI NHÉ! NÓ CHỈ CÒN: ', @SOLUONGTON);

END

ELSE IF (@SOLUONGTON < 10)

BEGIN

PRINT CONCAT(N'VACCINE CỦA BẠN SẮP HẾT, HÃY NHẬP HÀNG ĐI NHÉ! NÓ CHỈ CÒN: ', @SOLUONGTON);

END

ELSE

BEGIN

PRINT CONCAT(N'VACCINE CỦA BẠN VẪN CÒN ĐỦ! NÓ CÒN: ', @SOLUONGTON);

END

END;

* ***Thêm VACCINE***

CREATE PROCEDURE pro\_them\_vaccine

@MaVC CHAR(6),

@MaLoai CHAR(6),

@TenVC NVARCHAR(100),

@Gia INT,

@XuatXu NVARCHAR(100)

AS

BEGIN

BEGIN TRY

INSERT INTO VACCINE (MaVC, MaLoai, TenVC, Gia, XuatXu)

VALUES (@MaVC, @MaLoai, @TenVC, @Gia, @XuatXu)

END TRY

BEGIN CATCH

ROLLBACK TRAN

END CATCH

END;

GO

* ***Xoá VACCINE***

CREATE PROCEDURE pro\_xoa\_vaccine

@MaVC CHAR(6)

AS

BEGIN

BEGIN TRY

DELETE FROM VACCINE

WHERE MaVC = @MaVC;

END TRY

BEGIN CATCH

rollback tran

END CATCH;

END;

* ***Cập nhật thông tin VACCINE***

CREATE PROCEDURE pro\_capnhat\_vaccine

@MaVC CHAR(6),

@MaLoai CHAR(6),

@TenVC NVARCHAR(100),

@Gia INT,

@XuatXu NVARCHAR(100)

AS

BEGIN

BEGIN TRY

UPDATE VACCINE

SET MaLoai = @MaLoai,

TenVC = @TenVC,

Gia = @Gia,

XuatXu = @XuatXu

WHERE MaVC = @MaVC;

END TRY

BEGIN CATCH

rollback tran

END CATCH;

END;

GO

* ***Thêm LOAIBENH***

CREATE PROCEDURE pro\_them\_LOAIBENH

@MaLoai CHAR(6),

@TenLoai NVARCHAR(100),

@SoMui INT

AS

BEGIN

begin try

INSERT INTO LOAIBENH (MaLoai, TenLoai, SoMui)

VALUES (@MaLoai, @TenLoai, @SoMui);

end try

BEGIN CATCH

rollback tran -- c3 bắt lỗi thêm không thành công -- c# báo khi thêm khôgn thành công

END CATCH

END;

GO

* ***Cập nhật thông tin LOAIBENH***

CREATE PROCEDURE pro\_capnhat\_LOAIBENH

@MaLoai CHAR(6),

@TenLoai NVARCHAR(100),

@SoMui INT

AS

BEGIN

begin try

update LOAIBENH

set MaLoai = @MaLoai,

TenLoai = @TenLoai,

SoMui = @SoMui

where MaLoai = @MaLoai

end try

BEGIN CATCH

rollback tran

END CATCH

END;

GO

* ***Xoá LOAIBENH***

CREATE PROCEDURE pro\_xoa\_LOAIBENH

@MaLoai CHAR(6)

AS

BEGIN

BEGIN TRY

DELETE FROM LOAIBENH

WHERE MaLoai = @MaLoai;

END TRY

BEGIN CATCH

rollback tran

END CATCH;

END;

* ***Thêm PHIEUNHAP***

GO

CREATE PROCEDURE pro\_them\_phieunhap

@MaPN CHAR(6),

@MaNV CHAR(6),

@NgayNhap DATE,

@MaNCC CHAR(6)

AS

BEGIN

BEGIN TRY

INSERT INTO PHIEUNHAP (MaPN, MaNV, NgayNhap, MaNCC)

VALUES (@MaPN, @MaNV, @NgayNhap, @MaNCC);

END TRY

BEGIN CATCH

rollback tran

END CATCH

END;

* ***Thêm CHITIETPHIEUNHAP***

GO

CREATE PROCEDURE pro\_them\_chitietphieunhap

@MaPN CHAR(6),

@MaVC CHAR(6),

@MaLo CHAR(6),

@SoLuong INT,

@DonGia INT

AS

BEGIN

BEGIN TRY

INSERT INTO CHITIETPHIEUNHAP ([MaPN], [MaVC], [MaLo], [SoLuong], [DonGia])

VALUES (@MaPN, @MaVC,@MaLo, @SoLuong, @DonGia);

END TRY

BEGIN CATCH

rollback tran -- c3 bắt lỗi thêm không thành công

END CATCH

END;

GO

* ***Thêm LOVACCINE (Lô Vaccine)***

CREATE PROCEDURE pro\_them\_lovacine

@MaLo CHAR(6),

@MaVC CHAR(6),

@nsx date,

@hsd date,

@soluong INT

AS

BEGIN

BEGIN TRY

INSERT INTO LOVACCINE([MaLo], [MaVC], [NgaySX], [HanSuDung], [SoLuong])

VALUES (@MaLo,@MaVC,@nsx,@hsd,@soluong);

END TRY

BEGIN CATCH

rollback tran

END CATCH

END;

* ***Thêm NHACUNGCAP***

CREATE PROc pro\_them\_nhacungcap

@MaNCC CHAR(6),

@TenNCC NVARCHAR(200),

@DiaChi NVARCHAR(200),

@SoDienThoai CHAR(10)

AS

BEGIN

BEGIN TRY

INSERT INTO NHACUNGCAP (MaNCC, TenNCC, DiaChi, SoDienThoai)

VALUES (@MaNCC, @TenNCC, @DiaChi, @SoDienThoai);

END TRY

BEGIN CATCH

rollback tran -- c3 bắt lỗi thêm không thành công

END CATCH

END;

GO

* ***Cập nhật thông tin NHACUNGCAP***

CREATE PROc pro\_caonhat\_nhacungcap

@MaNCC CHAR(6),

@TenNCC NVARCHAR(200),

@DiaChi NVARCHAR(200),

@SoDienThoai CHAR(10)

AS

BEGIN

BEGIN TRY

UPDATE NHACUNGCAP

SET MaNCC = @MaNCC,

TenNCC = @TenNCC,

DiaChi = @DiaChi,

SoDienThoai = @SoDienThoai

where MaNCC = @MaNCC

END TRY

BEGIN CATCH

rollback tran -- c3 bắt lỗi thêm không thành công

END CATCH

END;

* ***Xoá NHACUNGCAP***

CREATE PROCEDURE pro\_xoa\_nhacungcap

@MaNCC CHAR(6)

AS

BEGIN

BEGIN TRY

DELETE FROM NHACUNGCAP

WHERE MaNCC = @MaNCC;

END TRY

BEGIN CATCH

rollback tran -- c3 bắt lỗi thêm không thành công

END CATCH;

END;

**c. Phần bài làm của Trần Tấn Tài**

* ***Thêm HOADON***

CREATE PROCEDURE SP\_THEM\_HOADON

@MaHD CHAR(5),

@MaBN CHAR(5),

@MaNV CHAR(5),

@TongTien FLOAT = 0

AS

BEGIN

BEGIN TRY

INSERT INTO HOADON (MaHD, NgayLap, MaBN, MaNV, TongTien)

VALUES (@MaHD, GETDATE(), @MaBN, @MaNV, @TongTien);

PRINT 'Thêm hóa đơn thành công!';

END TRY

BEGIN CATCH

PRINT 'Lỗi: Không thể thêm hóa đơn!';

THROW;

END CATCH

END;

* ***Xóa HOADON***

CREATE PROCEDURE SP\_XOA\_HOADON

@MaHD CHAR(5)

AS

BEGIN

BEGIN TRY

DELETE FROM CHITIETHOADON WHERE MaHD = @MaHD; -- Xóa chi tiết hóa đơn trước

DELETE FROM HOADON WHERE MaHD = @MaHD; -- Sau đó xóa hóa đơn

PRINT 'Xóa hóa đơn thành công!';

END TRY

BEGIN CATCH

PRINT 'Lỗi: Không thể xóa hóa đơn!';

THROW;

END CATCH

END;

* ***Cập nhật thông tin HOADON***

CREATE PROCEDURE SP\_SUA\_HOADON

@MaHD CHAR(5),

@MaBN CHAR(5),

@MaNV CHAR(5),

@TongTien FLOAT

AS

BEGIN

BEGIN TRY

UPDATE HOADON

SET MaBN = @MaBN,

MaNV = @MaNV,

TongTien = @TongTien

WHERE MaHD = @MaHD;

PRINT 'Cập nhật hóa đơn thành công!';

END TRY

BEGIN CATCH

PRINT 'Lỗi: Không thể cập nhật hóa đơn!';

THROW;

END CATCH

END;

* ***Thêm CHITIETHOADON***

CREATE PROCEDURE SP\_THEM\_CHITIETHOADON

@MaHD CHAR(5),

@MaVC CHAR(5),

@SoLuong INT,

@DonGia FLOAT

AS

BEGIN

BEGIN TRY

INSERT INTO CHITIETHOADON (MaHD, MaVC, SoLuong, DonGia)

VALUES (@MaHD, @MaVC, @SoLuong, @DonGia);

PRINT 'Thêm chi tiết hóa đơn thành công!';

END TRY

BEGIN CATCH

PRINT 'Lỗi: Không thể thêm chi tiết hóa đơn!';

THROW;

END CATCH

END;

* ***Xóa CHITIETHOADON***

CREATE PROCEDURE SP\_XOA\_CHITIETHOADON

@MaHD CHAR(5),

@MaVC CHAR(5)

AS

BEGIN

BEGIN TRY

DELETE FROM CHITIETHOADON

WHERE MaHD = @MaHD AND MaVC = @MaVC;

PRINT 'Xóa chi tiết hóa đơn thành công!';

END TRY

BEGIN CATCH

PRINT 'Lỗi: Không thể xóa chi tiết hóa đơn!';

THROW;

END CATCH

END;

* ***Sửa CHITIETHOADON***

CREATE PROCEDURE SP\_SUA\_CHITIETHOADON

@MaHD CHAR(5),

@MaVC CHAR(5),

@SoLuong INT,

@DonGia FLOAT

AS

BEGIN

BEGIN TRY

UPDATE CHITIETHOADON

SET SoLuong = @SoLuong,

DonGia = @DonGia

WHERE MaHD = @MaHD AND MaVC = @MaVC;

PRINT 'Cập nhật chi tiết hóa đơn thành công!';

END TRY

BEGIN CATCH

PRINT 'Lỗi: Không thể cập nhật chi tiết hóa đơn!';

THROW;

END CATCH

END;

### 2.2.3. Function

**a. Phần bài làm của Nguyễn Hoàng Thịnh**

* ***Tạo mã HOADON tự động***

CREATE FUNCTION FN\_TaoMaHDMoi()

RETURNS CHAR(5)

AS

BEGIN

DECLARE @MaHDMax CHAR(5);

DECLARE @SoMoi INT;

DECLARE @MaHDMoi CHAR(5);

SELECT @MaHDMax = MAX(MaHD)

FROM HOADON;

IF @MaHDMax IS NULL

BEGIN

SET @MaHDMoi = 'HD001';

END

ELSE

BEGIN

SET @SoMoi = CAST(SUBSTRING(@MaHDMax, 3, 3) AS INT) + 1;

SET @MaHDMoi = 'HD' + RIGHT('000' + CAST(@SoMoi AS VARCHAR(3)), 3);

END

RETURN @MaHDMoi;

END;

GO

* ***Tìm kiếm theo tên BENHNHAN***

CREATE FUNCTION FN\_TimKiemBN

(

@HoTen NVARCHAR(50)

)

RETURNS TABLE

AS

RETURN

(

SELECT \*

FROM BENHNHAN

WHERE HoTen LIKE '%' + @HoTen + '%'

);

**b. Phần bài làm của Võ Trường Danh**

**c. Phần bài làm của Trần Tấn Tài**

## 2.3. Quản trị người dùng

- Mô tả giải pháp quản trị người dùng của nhóm, chọn lựa cơ chế xác thực. Tạo các tài khoản login, user, nhóm quyền role, cấp, hủy quyền người dùng.

- Lập lịch trình sao lưu định kỳ cho cơ sở dữ liệu và thiết lập sao lưu tự động.

# CHƯƠNG 3. XÂY DỰNG ỨNG DỤNG

## 1. Chức năng danh mục

- Tạo form danh mục để hiển thị danh mục (ví dụ: danh mục hàng hóa, danh mục nhân viên,…), hiển thị kết quả trên datagridview trong c# hoặc table trong java,…

- Cài đặt các chức năng thêm, xóa, sửa dữ liệu, xuất report

* Danh sách các bệnh nhân **(Trần Tấn Tài)**

A screenshot of a computer

Description automatically generated

Hình 3.1. Danh sách bệnh nhân

+ Hàm Load ở lớp DAL

public DataTable **GetFullData()**

{

try

{

return dbHelper.ExecuteQuery("SELECT \* FROM BENHNHAN");

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

return null;

}

+ Gọi hàm load

public void LoadListViewDSBN()

{

lstvDSBN.Items.Clear();

foreach (DataRow row in **bnBLL.GetFullData().**Rows)

{

if (row.RowState != DataRowState.Deleted)

{

ListViewItem item1 = new ListViewItem(row["MaBN"].ToString());

item1.SubItems.Add(row["HoTen"].ToString());

item1.SubItems.Add(Convert.ToDateTime(row["NgaySinh"]).ToString("dd/MM/yyyy"));

item1.SubItems.Add(row["GioiTinh"].ToString());

item1.SubItems.Add(row["DiaChi"].ToString());

item1.SubItems.Add(row["SoDienThoai"].ToString());

lstvDSBN.Items.Add(item1);

}

}

}

* Chức năng xóa và sửa bệnh nhân **(Nguyễn Hoàng Thịnh)**

A screenshot of a computer

Description automatically generated

Hình 3.2. Giao diện xóa bệnh nhân

+ Hàm xóa bệnh nhân ở lớp DAL

public bool Delete(string MaBN)

{

string query = "EXEC SP\_XoaBenhNhan @MaBN";

SqlParameter[] parameters = {

new SqlParameter("@MaBN", MaBN)

};

dbHelper.ExecuteNonQuery(query, parameters);

Load();

return true;

}

A screenshot of a computer

Description automatically generated

Hình 3.3. Giao diện cập nhật thông tin bệnh nhân

+ Hàm cập nhật bệnh nhân

public bool Edit(string MaBNCanSua, BenhNhanDTO bnDTONew)

{

string query = "EXEC SP\_SuaBenhNhan @MaBN, @HoTen, @NgaySinh, @GioiTinh, @DiaChi, @SoDienThoai";

SqlParameter[] parameters = {

new SqlParameter("@MaBN", bnDTONew.MaBN),

new SqlParameter("@HoTen", bnDTONew.HoTen),

new SqlParameter("@NgaySinh", bnDTONew.NgaySinh),

new SqlParameter("@GioiTinh", bnDTONew.GioiTinh),

new SqlParameter("@DiaChi", bnDTONew.DiaChi),

new SqlParameter("@SoDienThoai", bnDTONew.SoDienThoai)

};

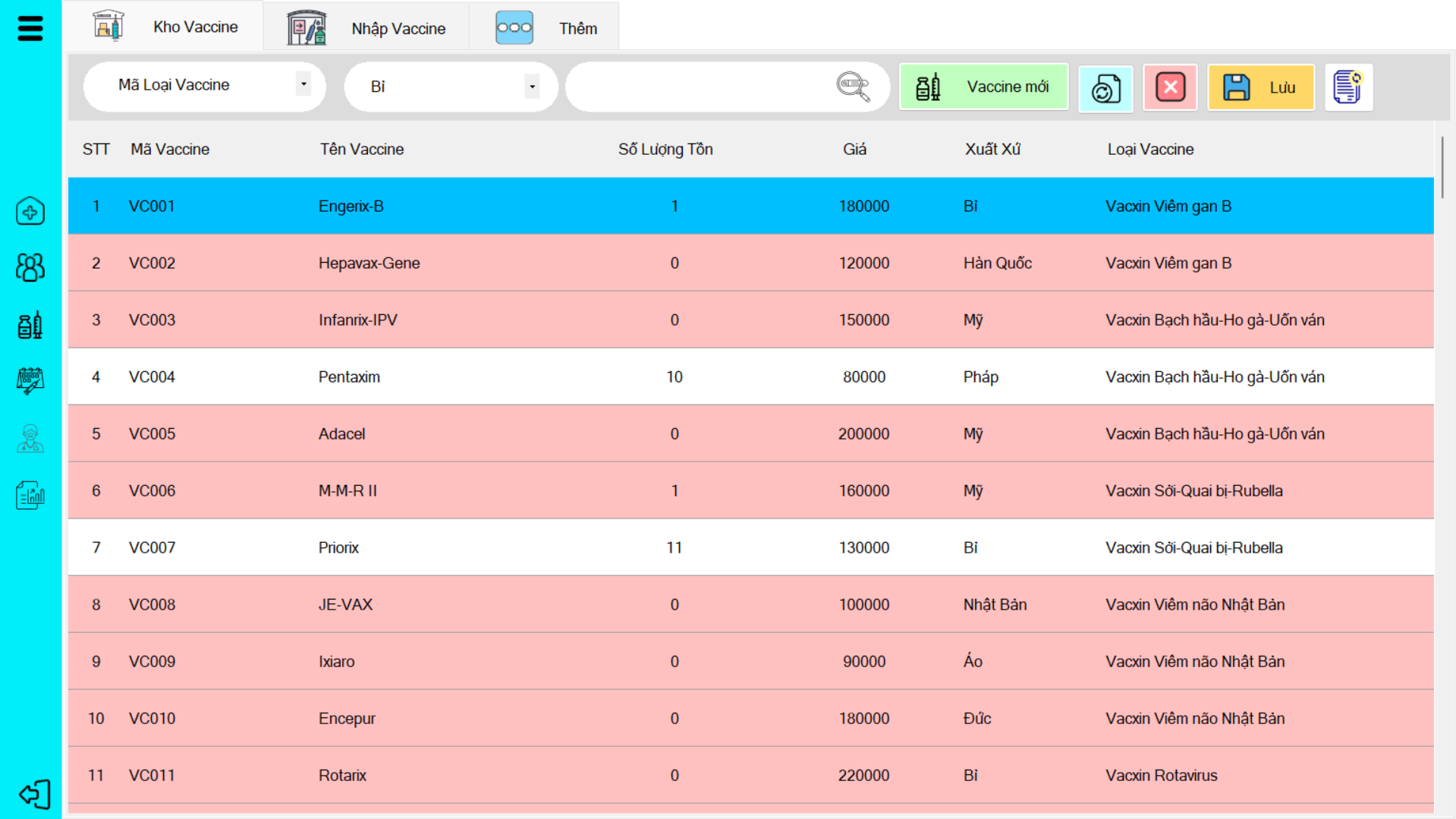
dbHelper.ExecuteNonQuery(query, parameters);

Load();

return true;

}

* Danh muc VACCINE **(Võ Trường Danh)**



+ Load Thông Tin ( **DAL** )

public DataTable LayTTVC()

{

return SqlCMDLayBang("select \* from Vaccine");

}

private DataTable SqlCMDLayBang(string truyxuat)

{

DataTable tmp = new DataTable();

try

{

if (conn.State == System.Data.ConnectionState.Closed)

conn.Open();

using (SqlCommand cmd = new SqlCommand(truyxuat, conn))

{

using (SqlDataReader reader = cmd.ExecuteReader())

{

tmp.Load(reader);

}

}

if (conn.State == System.Data.ConnectionState.Open)

conn.Close();

}

catch

{

return tmp = null;

}

return tmp;

}

* Load Thông Tin (**BLL**)

public DataTable LayTTVC()

{

return vcbll.LayTTVC();

}

* Load Thông Tin (**GUI**)

CreateDTGV(vaccineBLL.LayTTVC()); # Gọi hàm để view vào datagridview

private void CreateDTGV(DataTable dttb)

{

dgvVaccine.Columns.Clear(); // Xóa tất cả các cột

dgvVaccine.Rows.Clear(); // Xóa tất cả các hàng

DataGridViewTextBoxColumn stt = new DataGridViewTextBoxColumn

{

Name = "STT",

HeaderText = "STT"

};

DataGridViewTextBoxColumn mavc = new DataGridViewTextBoxColumn

{

Name = "MaVC",

HeaderText = "Mã Vaccine"

};

DataGridViewTextBoxColumn tenvc = new DataGridViewTextBoxColumn

{

Name = "TenVC",

HeaderText = "Tên Vaccine"

};

DataGridViewTextBoxColumn soluongton = new DataGridViewTextBoxColumn

{

Name = "SoLuongTon",

HeaderText = "Số Lượng Tồn"

};

DataGridViewTextBoxColumn gia = new DataGridViewTextBoxColumn

{

Name = "Gia",

HeaderText = "Giá"

};

DataGridViewTextBoxColumn xuatxu = new DataGridViewTextBoxColumn

{

Name = "XuatXu",

HeaderText = "Xuất Xứ"

};

DataGridViewTextBoxColumn loai = new DataGridViewTextBoxColumn

{

Name = "Loai",

HeaderText = "Loại Vaccine"

};

//Thêm vào dtg

dgvVaccine.Columns.Add(stt);

dgvVaccine.Columns.Add(mavc);

dgvVaccine.Columns.Add(tenvc);

dgvVaccine.Columns.Add(soluongton);

dgvVaccine.Columns.Add(gia);

dgvVaccine.Columns.Add(xuatxu);

dgvVaccine.Columns.Add(loai);

int tmp = 1;

DataTable gan = loaivcbll.GetData();

if (dttb != null)

{

foreach (DataRow row in dttb.Rows)

{

if (row.RowState != DataRowState.Deleted)

{

DataRow[] dr = gan.Select("MaLoai = '" + row["MaLoai"] + "'");

// Lấy giá trị từ DataRow

string tenLoai = dr[0]["TenLoai"].ToString();

string maLoai = row["MaLoai"].ToString();

dgvVaccine.Rows.Add(tmp++, row["MaVC"], row["TenVC"], row["SoLuongTon"], row["Gia"], row["XuatXu"], tenLoai);

}

}

}

CustomSizeCol();

foreach (DataGridViewRow row in dgvVaccine.Rows)

{

if (row.Cells["SoLuongTon"].Value != null)

{

int soLuongTon;

if (int.TryParse(row.Cells["SoLuongTon"].Value.ToString(), out soLuongTon))

{

if (soLuongTon < 10)

{

row.DefaultCellStyle.BackColor = Color.LightYellow;

}

if (soLuongTon < 2)

{

row.DefaultCellStyle.BackColor = Color.FromArgb(255, 192, 192);

}

}

}

}

}

* Thêm Vaccine (**GUI**)

VaccineBLL vcbll = new VaccineBLL();

int sl = vcbll.LayTTVC().Rows.Count + 1;

VaccineDTO vcdto = new VaccineDTO()

{

Mavc = "VC" + sl.ToString("D3") + "",

Maloai = cboLoaiVC.SelectedValue.ToString(),

Tenvc = txtTenVC.Text,

Gia = int.Parse(txtGia.Text),

Xuatxu = txtXuatXu.Text,

};

vcbll.Insert(vcdto);

* Thêm Vaccine (BLL)

public bool Insert(VaccineDTO vaccineDTO)

{

return vcbll.Insert(vaccineDTO);

}

* Thêm Vaccine (**DAL**)

private bool SqlCMD(string truyxuat)

{

try

{

if (conn.State == System.Data.ConnectionState.Closed)

conn.Open();

using (SqlCommand cmd = new SqlCommand(truyxuat, conn))

{

cmd.ExecuteNonQuery();

}

if (conn.State == System.Data.ConnectionState.Open)

conn.Close();

}

catch

{

return false;

}

return true;

}

public bool Insert(VaccineDTO vcDTO)

{

return SqlCMD("EXEC pro\_them\_vaccine @MaVC = '"+vcDTO.Mavc+"',@MaLoai = '"+vcDTO.Maloai+"',@TenVC = N'"+vcDTO.Tenvc+"',@Gia = "+vcDTO.Gia+", @XuatXu = N'"+vcDTO.Xuatxu+"';");

}

* Cập nhật VACCINE Vaccine (**GUI**)

VaccineBLL vcbll = new VaccineBLL();

VaccineDTO vcdto = new VaccineDTO()

{

Mavc = mavcdto,

Maloai = cboLoaiVC.SelectedValue.ToString(),

Tenvc = txtTenVC.Text,

Gia = int.Parse(txtGia.Text),

Xuatxu = txtXuatXu.Text,

};

vcbll.Update(vcdto);

* Cập nhật VACCINE Vaccine (**BLL**)

public bool Update(VaccineDTO vaccineDTO)

{

return vcbll.Update(vaccineDTO);

}

* Cập nhật VACCINE Vaccine (**DAL**)

#Gọi hàm SQLCMD ở trên để thực hiện thủ tục

public bool Update(VaccineDTO vcDTO)

{

return SqlCMD("EXEC pro\_capnhat\_vaccine @MaVC = '" + vcDTO.Mavc + "',@MaLoai = '" + vcDTO.Maloai + "',@TenVC = N'" + vcDTO.Tenvc + "',@Gia = " + vcDTO.Gia + ", @XuatXu = N'" + vcDTO.Xuatxu + "';");

}

* Xoá VACCINE Vaccine (**GUI**)

string ma = dgvVaccine.SelectedRows[0].Cells[1].Value.ToString();

string ten = dgvVaccine.SelectedRows[0].Cells[2].Value.ToString();

DataRow[] drcheckctpn = ctpnbll.GetData().Select("MaVC = '" + ma + "'");

if (drcheckctpn.Length == 0)

{

DialogResult t = MessageBox.Show("Bạn có chắc chắn muốn xóa Vaccine '" + ten + "' này không?",

"Xác nhận",

MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (t == DialogResult.Yes)

{

bool ck = vaccineBLL.Delete(ma);

if (ck)

{

ThongBao("Xoá thành công Vaccine " + ten + "", 1);

MessageBox.Show("Vaccine đã được xoá");

ReLoadFRM();

}

else

{

ThongBao("Xoá thất lại Vaccine " + ten + "", 2);

MessageBox.Show("Vaccine đã được sử dung thông tin không thể xoá");

ReLoadFRM();

}

}

}

else

{

ThongBao("Vaccine " + ten + " đã được sử dụng không thể xoá thông tin!", 2);

MessageBox.Show("Vaccine đã được sử dung thông tin không thể xoá");

}

* Xoá VACCINE Vaccine (**BLL**)

public bool Delete(string mavc)

{

return vcbll.Delete(mavc);

}

* Xoá VACCINE Vaccine (**DAL**)

#Gọi hàm SQLCMD ở trên để thực hiện thủ tục

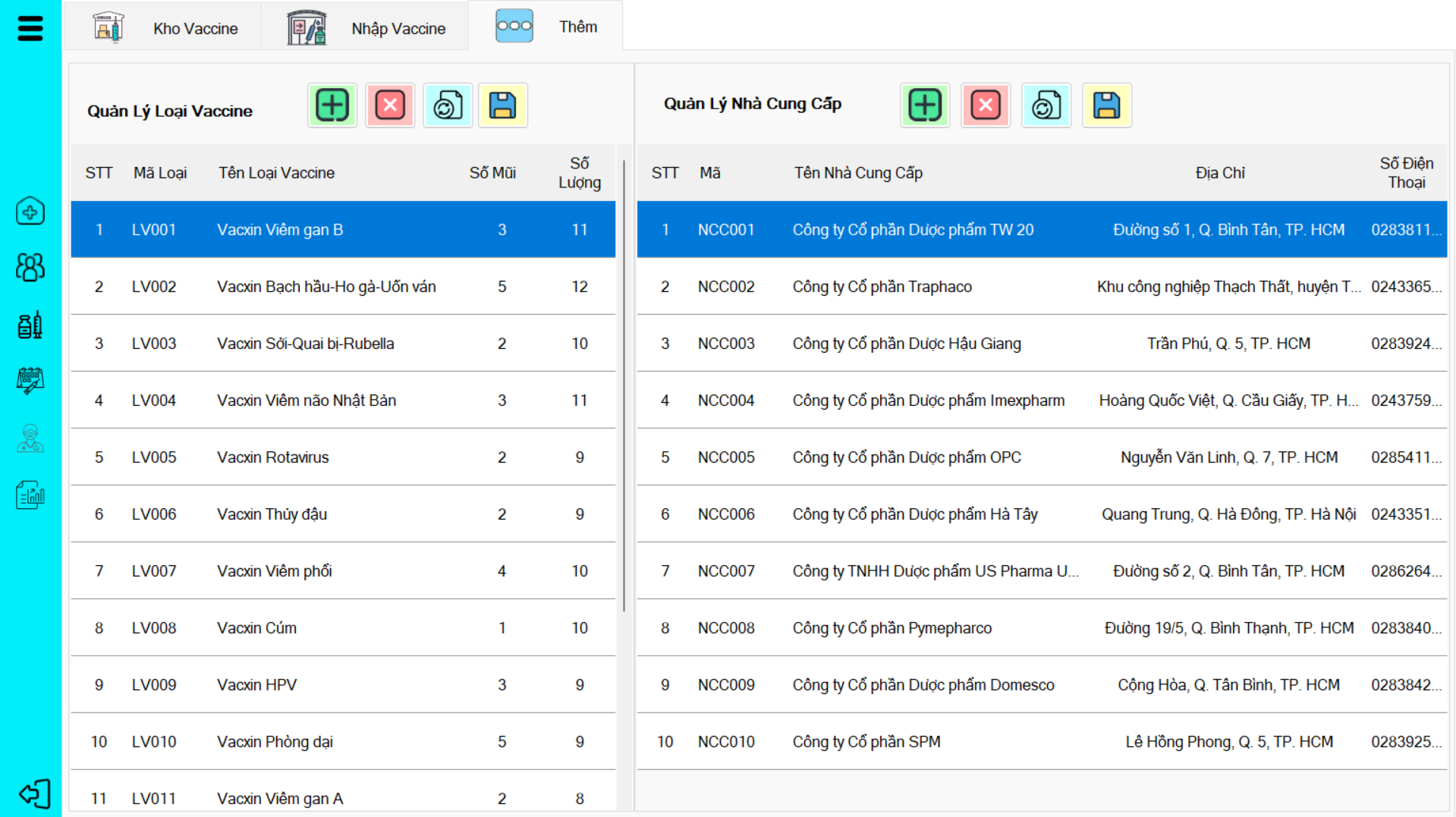
public bool Delete(string maVC)

{

return SqlCMD("EXEC pro\_xoa\_vaccine '"+maVC+"'");

}

* Danh mục Loại VACCINE và Nhà Cung Cấp (**Võ Trường Danh**)



* Load loại VACCINE (**DAL**)

public DataTable GetData()

{

return SqlCMDLayBang("select \* from LOAIBENH");

}

private DataTable SqlCMDLayBang(string truyxuat)

{

DataTable tmp = new DataTable();

try

{

if (conn.State == System.Data.ConnectionState.Closed)

conn.Open();

using (SqlCommand cmd = new SqlCommand(truyxuat, conn))

{

using (SqlDataReader reader = cmd.ExecuteReader())

{

tmp.Load(reader);

}

}

if (conn.State == System.Data.ConnectionState.Open)

conn.Close();

}

catch

{

return tmp = null;

}

return tmp;

}

* Load loại VACCINE (**BLL**)

LoaiVaccineDAL loaiVaccineDAL = new LoaiVaccineDAL();

public DataTable GetData()

{

return loaiVaccineDAL.GetData();

}

* Load loại VACCINE (**GUI**)

CreateDgvLoaiVaccine(loaivcbll.GetData()); # gọi hàm view

private void CreateDgvLoaiVaccine(DataTable dttb)

{

dgvLoaiVC.Columns.Clear(); // Xóa tất cả các cột

dgvLoaiVC.Rows.Clear(); // Xóa tất cả các hàng

DataGridViewTextBoxColumn stt = new DataGridViewTextBoxColumn

{

Name = "STT",

HeaderText = "STT"

};

DataGridViewTextBoxColumn maloai = new DataGridViewTextBoxColumn

{

Name = "MaLoai",

HeaderText = "Mã Loại"

};

DataGridViewTextBoxColumn tenloai = new DataGridViewTextBoxColumn

{

Name = "TenLoai",

HeaderText = "Tên Loại Vaccine"

};

DataGridViewTextBoxColumn somui = new DataGridViewTextBoxColumn

{

Name = "SoMui",

HeaderText = "Số Mũi"

};

DataGridViewTextBoxColumn soluong = new DataGridViewTextBoxColumn

{

Name = "SoLuong",

HeaderText = "Số Lượng"

};

//Thêm vào dtg

dgvLoaiVC.Columns.Add(stt);

dgvLoaiVC.Columns.Add(maloai);

dgvLoaiVC.Columns.Add(tenloai);

dgvLoaiVC.Columns.Add(somui);

dgvLoaiVC.Columns.Add(soluong);

int tmp = 1;

DataTable dttmp = vaccineBLL.LayTTVC();

if (dttb != null)

{

foreach (DataRow row in dttb.Rows)

{

if (row.RowState != DataRowState.Deleted)

{

string maloaivc = row["MaLoai"].ToString();

DataRow[] sltmp = dttmp.Select("MaLoai = '" + maloaivc + "'");

dgvLoaiVC.Rows.Add(tmp++, row["MaLoai"], row["TenLoai"], row["SoMui"], sltmp.Count());

}

}

}

CustomSizeColLoaiVacine();

}

* Thêm loại VACCINE (**GUI**)

LoaiVaccineBLL lvcbll = new LoaiVaccineBLL();

int sl = lvcbll.GetData().Rows.Count + 1;

LoaiVaccineDTO lvcdto = new LoaiVaccineDTO()

{

Maloai = "LV" + sl.ToString("D3") + "",

Tenloai = txtTenLoaiVC.Text,

Somui = int.Parse(txtSoMui.Text),

};

if(lvcbll.Insert(lvcdto))

{ MessageBox.Show("Thêm thành công Loại Vaccine");

VaccineDTO.CheckTB = true;

}

else

{

MessageBox.Show("Thêm không thành công vui lòng kiểm tra lại dử liệu ");

return;

}

* Thêm loại VACCINE (**BLL**)

public bool Insert(LoaiVaccineDTO tmp)

{

return loaiVaccineDAL.Insert(tmp);

}

* Thêm loại VACCINE (**DAL**)

private bool SqlCMD(string truyxuat)

{

try

{

if (conn.State == System.Data.ConnectionState.Closed)

conn.Open();

using (SqlCommand cmd = new SqlCommand(truyxuat, conn))

{

cmd.ExecuteNonQuery();

}

if (conn.State == System.Data.ConnectionState.Open)

conn.Close();

}

catch

{

return false;

}

return true;

}

public bool Insert(LoaiVaccineDTO lvcDTO)

{

return SqlCMD("EXEC pro\_them\_loaivaccine '"+lvcDTO.Maloai+"', N'"+lvcDTO.Tenloai+"','"+lvcDTO.Somui+"' ");

}

* Cập nhật loại VACCINE (**GUI**)

LoaiVaccineDTO tmp = new LoaiVaccineDTO()

{

Maloai = malvc.ToString(),

Tenloai = txtTenLoaiVC.Text,

Somui = int.Parse(txtSoMui.Text),

};

if (lvcbll.Update(tmp))

{

MessageBox.Show("Cập nhật thành công Loại Vaccine");

VaccineDTO.CheckTB = true;

}

else

{

MessageBox.Show("Cập nhật không thành công vui lòng kiểm tra lại dử liệu ");

return;

}

* Cập nhật loại VACCINE (**BLL**)

public bool Update(LoaiVaccineDTO loaiVaccineDTO)

{

return loaiVaccineDAL.Update(loaiVaccineDTO);

}

* Cập nhật loại VACCINE **(DAL**)

#gọi hàm SQLCMD như trên phần thêm để thực hiện thủ tục

public bool Update(LoaiVaccineDTO loaiVaccineDTO)

{

return SqlCMD("EXEC pro\_capnhat\_loaivaccine '" + loaiVaccineDTO.Maloai + "', N'" + loaiVaccineDTO.Tenloai + "','" + loaiVaccineDTO.Somui + "' ");

}

* Xoá loại VACCINE (**GUI**)

string ma = dgvLoaiVC.SelectedRows[0].Cells[1].Value.ToString();

string ten = dgvLoaiVC.SelectedRows[0].Cells[2].Value.ToString();

DataRow[] drcheckctpn = vaccineBLL.LayTTVC().Select("MaLoai = '" + ma + "'");

if (drcheckctpn.Length == 0)

{

DialogResult t = MessageBox.Show("Bạn có chắc chắn muốn xóa loại Vaccine '" + ten + "' này không?",

"Xác nhận",

MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (t == DialogResult.Yes)

{

bool ck = loaivcbll.Delete(ma);

if (ck)

{

ThongBaoTab3("Xoá thành công loại Vaccine " + ten + "", 1);

CreateDgvLoaiVaccine(loaivcbll.GetData());

}

else

{

ThongBaoTab3("Xoá thất lại loại Vaccine " + ten + "", 2);

ReLoadFRM();

}

}

}

else

{

ThongBaoTab3("Loại Vaccine " + ten + " đã được sử dụng không thể xoá thông tin!", 2);

}

* Xoá loại VACCINE (**BLL**)

public bool Delete(string ndxoa)

{

return loaiVaccineDAL.Delete(ndxoa);

}

* Xoá loại VACCINE (**DAL**)

#gọi hàm SQLCMD như trên phần thêm để thực hiện thủ tục

public bool Delete(string maLoai)

{

return SqlCMD("EXEC pro\_xoa\_loaivaccine '" + maLoai+"' ");

}

* Load Nhà Cung Cấp (**DAL**)

public DataTable GetData()

{

return SqlCMDLayBang("select \* from NhaCungCap");

}

private DataTable SqlCMDLayBang(string truyxuat)

{

DataTable tmp = new DataTable();

try

{

if (conn.State == System.Data.ConnectionState.Closed)

conn.Open();

using (SqlCommand cmd = new SqlCommand(truyxuat, conn))

{

using (SqlDataReader reader = cmd.ExecuteReader())

{

tmp.Load(reader);

}

}

if (conn.State == System.Data.ConnectionState.Open)

conn.Close();

}

catch

{

return tmp = null;

}

return tmp;

}

* Load Nhà Cung Cấp (**BLL**)

NhaCungCapDAL NhaCcDAL = new NhaCungCapDAL();

public DataTable GetData()

{

return NhaCcDAL.GetData();

}

* Load Nhà Cung Cấp (**GUI**)

CreateDgvNCC(nccbll.GetData());

private void CreateDgvNCC(DataTable dttb)

{

dgvNCC.Columns.Clear(); // Xóa tất cả các cột

dgvNCC.Rows.Clear(); // Xóa tất cả các hàng

DataGridViewTextBoxColumn stt = new DataGridViewTextBoxColumn

{

Name = "STT",

HeaderText = "STT"

};

DataGridViewTextBoxColumn mancc = new DataGridViewTextBoxColumn

{

Name = "MaNCC",

HeaderText = "Mã"

};

DataGridViewTextBoxColumn tenncc = new DataGridViewTextBoxColumn

{

Name = "TenNCC",

HeaderText = "Tên Nhà Cung Cấp"

};

DataGridViewTextBoxColumn diachi = new DataGridViewTextBoxColumn

{

Name = "DiaChi",

HeaderText = "Địa Chỉ"

};

DataGridViewTextBoxColumn sdt = new DataGridViewTextBoxColumn

{

Name = "SoDienThoai",

HeaderText = "Số Điện Thoại"

};

//Thêm vào dtg

dgvNCC.Columns.Add(stt);

dgvNCC.Columns.Add(mancc);

dgvNCC.Columns.Add(tenncc);

dgvNCC.Columns.Add(diachi);

dgvNCC.Columns.Add(sdt);

int tmp = 1;

if (dttb != null)

{

foreach (DataRow row in dttb.Rows)

{

if (row.RowState != DataRowState.Deleted)

{

dgvNCC.Rows.Add(tmp++, row["MaNCC"], row["TenNCC"], row["DiaChi"], row["SoDienThoai"]);

}

}

}

CustomSizeColNCC();

}

* Thêm Nhà Cung Cấp (**GUI**)

NhaCungCapBLL nccbll = new NhaCungCapBLL();

int sl = nccbll.GetData().Rows.Count + 1;

NhaCungCapDTO tmp = new NhaCungCapDTO()

{

Mancc = "NCC" + sl.ToString("D3") + "",

Tenncc = txtTenNCC.Text,

Diachi = txtDiaChi.Text,

Sodienthoai = txtSoDienThoai.Text,

};

if (nccbll.Insert(tmp))

MessageBox.Show("Thêm thành công Nhà Cung Cấp mới");

else

{

MessageBox.Show("Thêm không thành công vui lòng kiểm tra lại dữ liệu !");

return;

}

* Thêm Nhà Cung Cấp (**BLL**)

public bool Insert(NhaCungCapDTO tmp)

{

return NhaCcDAL.Insert(tmp);

}

* Thêm Nhà Cung Cấp (**DAL**)

public bool Insert(NhaCungCapDTO tmp)

{

return SqlCMD("EXEC pro\_them\_nhacungcap '"+tmp.Mancc+"',N'"+tmp.Tenncc+"',N'"+tmp.Diachi+"','"+tmp.Sodienthoai+"'");

}

private bool SqlCMD(string truyxuat)

{

try

{

if (conn.State == System.Data.ConnectionState.Closed)

conn.Open();

using (SqlCommand cmd = new SqlCommand(truyxuat, conn))

{

cmd.ExecuteNonQuery();

}

if (conn.State == System.Data.ConnectionState.Open)

conn.Close();

}

catch

{

return false;

}

return true;

}

* Cập nhật Nhà Cung Cấp (**GUI**)

NhaCungCapDTO tmp = new NhaCungCapDTO()

{

Mancc = ma,

Tenncc = txtTenNCC.Text,

Diachi = txtDiaChi.Text,

Sodienthoai = txtSoDienThoai.Text,

};

if (nccbll.Update(tmp))

MessageBox.Show("Cập nhật thành công Nhà Cung Cấp");

else

{

MessageBox.Show("Cập nhật không thành công vui lòng kiểm tra lại dữ liệu !");

return;

}

* Cập nhật Cung Cấp (**BLL**)

public bool Update(NhaCungCapDTO tmp)

{

return NhaCcDAL.Update(tmp);

}

* Cập nhật Nhà Cung Cấp (**DAL**)

#gọi hàm SQLCMD như trên phần thêm để thực hiện thủ tục

public bool Update(NhaCungCapDTO tmp)

{

return SqlCMD("EXEC pro\_caonhat\_nhacungcap '" + tmp.Mancc + "',N'" + tmp.Tenncc + "',N'" + tmp.Diachi + "','" + tmp.Sodienthoai + "'");

}

* Xoá Nhà Cung Cấp (**GUI**)

string ma = dgvNCC.SelectedRows[0].Cells[1].Value.ToString();

string ten = dgvNCC.SelectedRows[0].Cells[2].Value.ToString();

DataRow[] drcheck = pnbll.GetData().Select("MaNCC = '" + ma + "'");

if (drcheck.Length == 0)

{

DialogResult t = MessageBox.Show("Bạn có chắc chắn muốn xóa nhà cung cấp Vaccine '" + ten + "' này không?",

"Xác nhận",

MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (t == DialogResult.Yes)

{

bool ck = nccbll.Delete(ma);

if (ck)

{

ThongBaoTab3("Xoá thành công nhà cung cấp Vaccine " + ten + "", 1);

CreateDgvNCC(nccbll.GetData());

}

else

{

ThongBaoTab3("Xoá thất lại nhà cung cấp Vaccine " + ten + "", 2);

ReLoadFRM();

}

}

}

else

{

ThongBaoTab3("Nhà cung cấp Vaccine " + ten + " đã được sử dụng không thể xoá thông tin!", 2);

}

* Xoá Nhà Cung Cấp (**BLL**)

public bool Delete(string tmp)

{

return NhaCcDAL.Delete(tmp);

}

* Xoá Nhà Cung Cấp (**DAL**)

#gọi hàm SQLCMD như trên phần thêm để thực hiện thủ tục

public bool Delete(string tmp)

{

return SqlCMD("EXEC pro\_xoa\_nhacungcap '" + tmp+ "' ");

}

## 2. Chức năng thống kê

- Xây dựng form thống kê để thực hiện một dạng thống kê (ví dụ: thống kê doanh thu theo mặt hàng, theo nhân viên bán hàng, theo loại hàng,…). hiển thị kết quả thống kê trên datagridview trong c# hoặc table trong java,…Khuyến khích hiển thị kết quả thống kê trên report.

* Thống kê nhập hàng (**GUI**):  
  PhieuNhapBLL pnbll = new PhieuNhapBLL();

chartPhieuNhap.DataSource = pnbll.ThongKe();

chartPhieuNhap.Series.Clear();

var series = chartPhieuNhap.Series.Add("Tổng Tiền");

foreach (Series t in chartPhieuNhap.Series)

{

t.IsValueShownAsLabel = true;

t.LabelFormat = "{0,0}₫";

}

series.XValueMember = "ThangNam";

series.YValueMembers = "TongTien";

series.ChartType = SeriesChartType.Column;

* Thống kê nhập hàng **(BLL)**

PhieuNhapDAL pn = new PhieuNhapDAL();

public DataTable ThongKe()

{

return pn.ThongKe();

}

* Thống kê nhập hàng **(DAL)**

private bool SqlCMD(string truyxuat)

{

try

{

if (conn.State == System.Data.ConnectionState.Closed)

conn.Open();

using (SqlCommand cmd = new SqlCommand(truyxuat, conn))

{

cmd.ExecuteNonQuery();

}

if (conn.State == System.Data.ConnectionState.Open)

conn.Close();

}

catch

{

return false;

}

return true;

}

public DataTable ThongKe()

{

return SqlCMDLayBang("EXEC pro\_thongkenhaphang");

}

* Thống kê bán hàng **(GUI)**

chartHoaDon.DataSource = pnbll.ThongKeHD();

chartHoaDon.Series.Clear();

chartHoaDon.ChartAreas[0].AxisX.Interval = 1;

chartHoaDon.ChartAreas[0].AxisX.LabelStyle.Angle = -45;

chartHoaDon.ChartAreas[0].AxisX.LabelStyle.Font = new Font("Arial", 8f);

var seriess = chartHoaDon.Series.Add("Tổng Tiền");

foreach (Series t in chartHoaDon.Series)

{

t.IsValueShownAsLabel = true;

t.LabelFormat = "{0,0}₫";

}

seriess.XValueMember = "ThangNam";

seriess.YValueMembers = "TongTien";

seriess.ChartType = SeriesChartType.Column;

chartHoaDon.DataBind();

* Thống kê bán hàng (**BLL)**

HoaDonDAL pn = new HoaDonDAL();

public DataTable ThongKeHD()

{

return pn.ThongKeHD();

}

* Thống kê bán hàng **(DAL)**

private DataTable SqlCMDLayBang(string truyxuat)

{

DataTable tmp = new DataTable();

try

{

if (conn.State == System.Data.ConnectionState.Closed)

conn.Open();

using (SqlCommand cmd = new SqlCommand(truyxuat, conn))

{

using (SqlDataReader reader = cmd.ExecuteReader())

{

tmp.Load(reader);

}

}

if (conn.State == System.Data.ConnectionState.Open)

conn.Close();

}

catch

{

return tmp = null;

}

return tmp;

}

public DataTable ThongKeHD()

{

return SqlCMDLayBang("EXEC pro\_thongkebanvaccine");

}

## 3. Chức năng tìm kiếm

- Xây dựng form tìm kiếm để tìm kiếm tương đối dựa vào các giá trị nhập vào như: mã nhân viên, họ tên, điện thoại, …lưu ý: các giá trị này chỉ nhập vào 1 ô tìm kiếm duy nhất. Kết quả hiện thị thể hiện ở đối tượng bảng (ví dụ datagridview trong c#, table trong java,…)

* Chức năng tìm kiếm bệnh nhân **(Nguyễn Hoàng Thịnh)**

A screenshot of a computer

Description automatically generated

Hình 3.4. Giao diện tìm kiếm

+ Hàm tìm kiếm bệnh nhân trong lớp DAL

public DataView GetDataViewFromTimKiem(string searchStr)

{

string query = "SELECT \* FROM dbo.FN\_TimKiemBN(@HoTen)";

SqlParameter[] parameters = new SqlParameter[]

{

new SqlParameter("@HoTen", SqlDbType.NVarChar) { Value = searchStr }

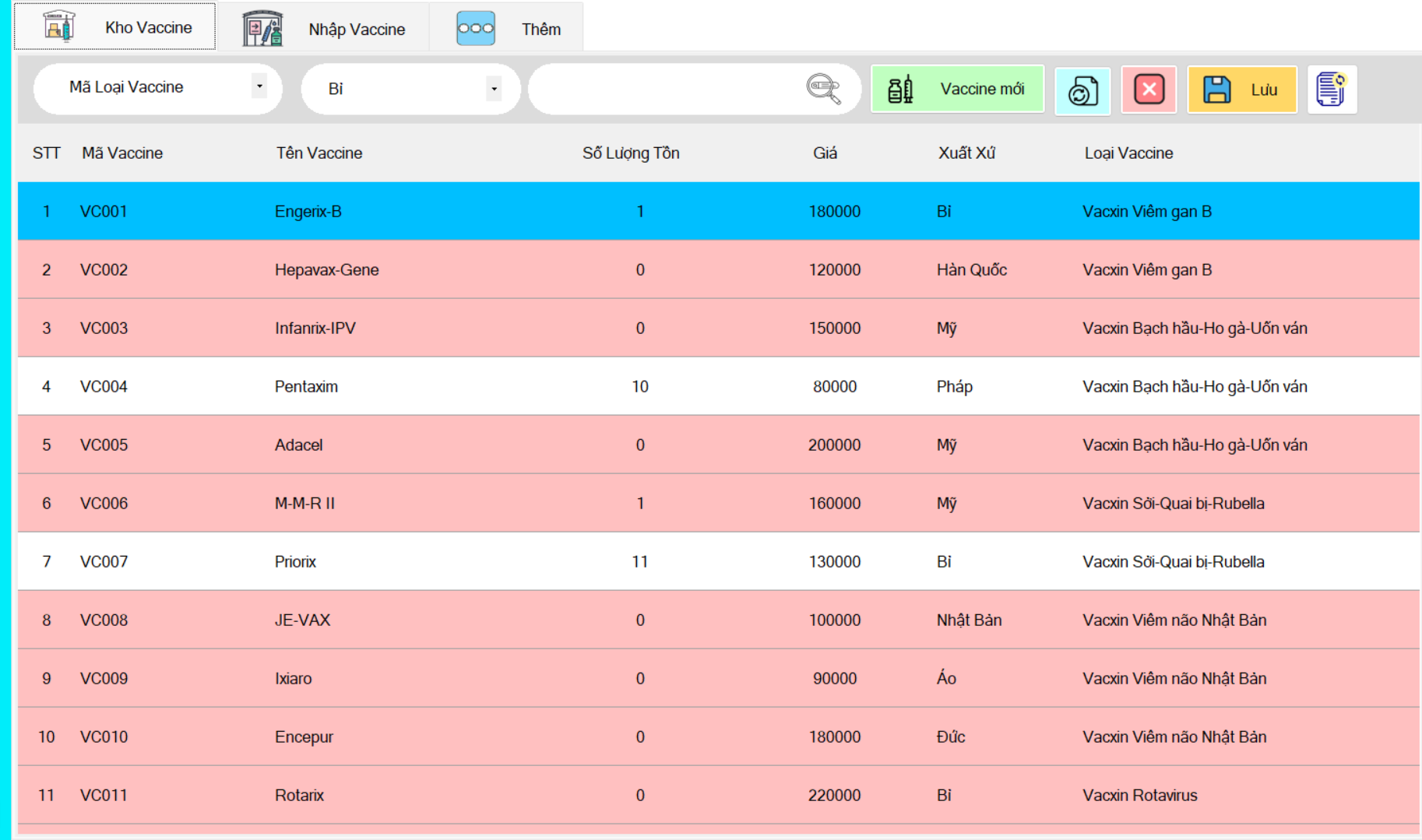
};

DataTable dataTable = dbHelper.ExecuteQuery(query, parameters);

return dataTable.DefaultView;

}

Chức năng tìm kiếm Vaccine **(Võ Trường Danh)**

****

* **Tìm Kiếm (GUI)**

string ndtimkiem = txtSearch.Text;

DataTable DTTMP = vaccineBLL.SearchTen(ndtimkiem);

if (DTTMP.Rows.Count > 0)

{

CreateDTGV(DTTMP);

ThongBao("Nội dung đã được tìm thấy ", 1);

return;

}

else

CreateDTGV(DTTMP = null);

ThongBao("Không tìm thấy nội dung nào phù hợp", 2);

* **Tìm Kiếm (BLL)**

public DataTable SearchTen(string nd)

{

return vcbll.SearchTen(nd);

}

* **Tìm Kiếm (DAL)**

public DataTable SearchTen(string ndtimkiem)

{

return SqlCMDLayBang("exec pro\_timkiemtheoten '" + ndtimkiem + "'");

}

private DataTable SqlCMDLayBang(string truyxuat)

{

DataTable tmp = new DataTable();

try

{

if (conn.State == System.Data.ConnectionState.Closed)

conn.Open();

using (SqlCommand cmd = new SqlCommand(truyxuat, conn))

{

using (SqlDataReader reader = cmd.ExecuteReader())

{

tmp.Load(reader);

}

}

if (conn.State == System.Data.ConnectionState.Open)

conn.Close();

}

catch

{

return tmp = null;

}

return tmp;

}

* **Lọc Theo Mã Hoặc Xuất Xứ (GUI)**

private void cboLoaiVC\_SelectionChangeCommitted(object sender, EventArgs e)

{

string maloai = cboLoaiVC.SelectedValue.ToString();

dgvVaccine.Columns.Clear(); // Xóa tất cả các cột

dgvVaccine.Rows.Clear(); // Xóa tất cả các hàng

CreateDTGV(vaccineBLL.Search(maloai));

}

private void cboXuatXu\_SelectionChangeCommitted(object sender, EventArgs e)

{

string maloai = cboXuatXu.SelectedValue.ToString();

dgvVaccine.Columns.Clear(); // Xóa tất cả các cột

dgvVaccine.Rows.Clear(); // Xóa tất cả các hàng

CreateDTGV(vaccineBLL.Search(maloai));

}

* **Lọc Theo Mã Hoặc Xuất Xứ (BLL)**

public DataTable Search(string ndtimkiem)

{

return vcbll.Search(ndtimkiem);

}

* **Lọc Theo Mã Hoặc Xuất Xứ (DAL)**

#Gọi hàm SLQLaynBang như ở SearchTen để thực hiện thủ thục

public DataTable Search(string ndtimkiem)

{

return SqlCMDLayBang("exec pro\_loctheoloaihoacxuatxu N'"+ndtimkiem+"'");

}

## 4. Chức năng nghiệp vụ

- Xây dựng form thực hiện chức năng nghiệp vụ ví dụ: lập hóa đơn bán hàng/nhập hàng/đăng ký phòng ở/cho mượn sách,…

- Cài đặt chức năng thêm, xóa, sửa dữ liệu.

* Nghiệp vụ đăng ký tiêm vaccine cho bệnh nhân **(Nguyễn Hoàng Thịnh)**

A screenshot of a computer

Description automatically generated

Hình 3.5. Giao diện đăng ký vaccine

Khi điền đầy đủ thông tin và click vào button “Thêm mới bệnh nhân” thì sẽ chạy 3 thủ tục đó là:

+ Thêm bệnh nhân

public bool Insert(BenhNhanDTO bnDTO)

{

string query = "EXEC SP\_ThemBenhNhan @HoTen, @NgaySinh, @GioiTinh, @DiaChi, @SoDienThoai";

SqlParameter[] parameters = {

new SqlParameter("@HoTen", bnDTO.HoTen),

new SqlParameter("@NgaySinh", bnDTO.NgaySinh),

new SqlParameter("@GioiTinh", bnDTO.GioiTinh),

new SqlParameter("@DiaChi", bnDTO.DiaChi),

new SqlParameter("@SoDienThoai", bnDTO.SoDienThoai)

};

dbHelper.ExecuteNonQuery(query, parameters);

Load();

return true;

}

+ Thêm hóa đơn

public bool InsertHD(HoaDonDTO hoaDonDTO)

{

string query = "EXEC SP\_THEM\_HOADON @MaHD, @MaBN, @MaNV, @TongTien";

SqlParameter[] parameters = {

new SqlParameter("@MaHD", hoaDonDTO.MaHD),

new SqlParameter("@MaBN", hoaDonDTO.MaBN),

new SqlParameter("@MaNV", hoaDonDTO.MaNV),

new SqlParameter("@TongTien", hoaDonDTO.TongTien)

};

dbHelper.ExecuteNonQuery(query, parameters);

Load();

return true;

}

+ Thêm chi tiết hóa đơn

public bool InsertCTHD(ChiTietHoaDonDTO chiTietHoaDonDTO)

{

string query = "EXEC SP\_THEM\_CHITIETHOADON @MaHD, @MaVC, @SoLuong, @DonGia";

SqlParameter[] parameters = {

new SqlParameter("@MaHD", chiTietHoaDonDTO.MaHD),

new SqlParameter("@MaVC", chiTietHoaDonDTO.MaVC),

new SqlParameter("@SoLuong", chiTietHoaDonDTO.SoLuong),

new SqlParameter("@DonGia", chiTietHoaDonDTO.DonGia)

};

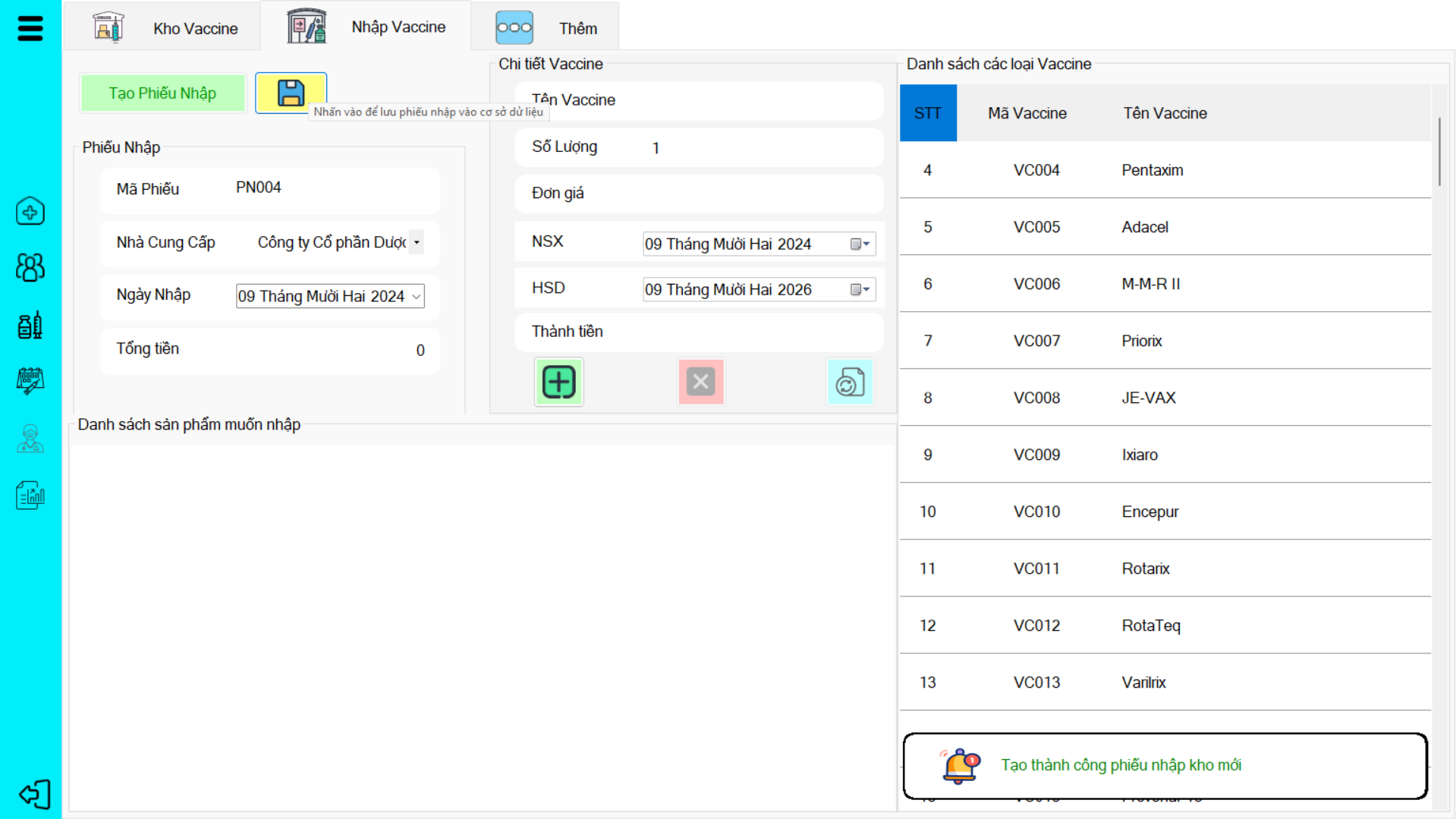
dbHelper.ExecuteNonQuery(query, parameters);

Load();

return true;

}

* Nghiệp vụ nhập hàng **(Võ Trường Danh)**

**s**

## 5. Cài đặt chức năng phân quyền người dùng

- Bổ sung thêm một số bảng dữ liệu để lưu trữ thông tin người dùng, quyền, nhóm quyền

- Xây dựng form thực hiện chức năng tạo, chỉnh sửa thông tin người dùng

- Tạo nhóm quyền

- Cấp quyền cho người dùng theo nhóm.

# TÀI LIỆU THAM KHẢO

[1] Phần Mền Quản Lý Phòng Mạch - Medict - <https://medict.mobi/phan-mem-phong-mach-bang-gia.html>

[2] Phần mền Quản Lý Phòng Khám – VietMis - https://www.vietmis.com/ehealth-phong-kham.html