

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TPHCM

KHOA CÔNG NGHỆ THÔNG TIN

TP.Hồ Chí Minh, ngày 24 tháng 11 năm 2021

GVHD: TS. NGUYỄN THÀNH SƠN

SVTH: HUỲNH THIÊN LONG 18133025

**MÔN: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI:**

**PHẦN MỀM QUẢN LÝ MÓN ĂN**

MỤC LỤC

[CHƯƠNG 1: ĐẶC TẢ HỆ THỐNG 4](#_Toc88681578)

[**I.** Mô tả hệ thống 4](#_Toc88681579)

[**1.** Thức ăn quan trọng như thế nào ? 4](#_Toc88681580)

[**2.** Mô tả dữ liệu phần mềm 4](#_Toc88681581)

[**II.** Quy trình xử lý 4](#_Toc88681582)

[**III.** Mô hình Diagram 4](#_Toc88681583)

[CHƯƠNG 2: CẤU HÌNH VÀ CÀI ĐẶT HỆ THỐNG 4](#_Toc88681584)

[**I.** **Tạo cơ sở dữ liệu và ràng buộc** 4](#_Toc88681585)

[**II.** **Tạo các câu lệnh thêm, xoá, sửa bằng Stored Procedure** 4](#_Toc88681586)

[**1.** **Tài khoản** 4](#_Toc88681587)

[**2.** **Nhân viên** 4](#_Toc88681588)

[**3.** **Khách hàng** 4](#_Toc88681589)

[**4.** **Nguyên liệu** 4](#_Toc88681590)

[**5.** **Công thức** 4](#_Toc88681591)

[**6.** **Món ăn** 4](#_Toc88681592)

[**7.** **Loại món ăn** 4](#_Toc88681593)

[**8.** **Đặt món** 4](#_Toc88681594)

[**III.** KẾT NỐI CSDL 4](#_Toc88681595)

[**IV.** PHÂN QUYỀN 4](#_Toc88681596)

[**V.** TRIGGER VÀ TRANSACTION 5](#_Toc88681597)

[**1.** Kiểm tra gmail tài khoản có hợp lệ không 5](#_Toc88681598)

[**2.** Kiểm tra gmail nhân viên có hợp lệ không 5](#_Toc88681599)

[**3.** Kiểm tra tên nhân viên không trùng nhau 5](#_Toc88681600)

[**4.** Kiểm tra số CMND không vượt qua 9 số 5](#_Toc88681601)

[**5.** Kiểm tra số điện thoại phải đủ 10 số 5](#_Toc88681602)

[**6.** Kiểm tra tên khách hàng không được trùng 5](#_Toc88681603)

[**7.** Kiểm tra gmail khách hàng có hợp lệ không 5](#_Toc88681604)

[**8.** Kiểm tra SDT khách hàng đúng 10 số 5](#_Toc88681605)

[**9.** Kiểm tra tên món ăn không được giống nhau 5](#_Toc88681606)

[**10.** Kiểm tra không cho nhập giá trị vào thứ 7 và chủ nhật 5](#_Toc88681607)

[**11.** Kiểm tra giá của món ăn phải giống nhau 5](#_Toc88681608)

[**12.** Kiểm tra loại món ăn không được giống nhau 5](#_Toc88681609)

[**13.** Tự thanh toán tiền 5](#_Toc88681610)

[**VI.** FUNCTION 5](#_Toc88681611)

[1. Trả về kết quả của 1 bảng để cho biết được KPI của nhân viên 5](#_Toc88681612)

[2. Trả về kết quả là 1 bảng công thức món ăn 5](#_Toc88681613)

[3. Trả về 1 bảng về nguồn gốc của món ăn 5](#_Toc88681614)

[4. Trả về 1 giá trị thể hiện giá của món ăn 5](#_Toc88681615)

[5. Trả về 1 giá trị thể hiện tổng số tiền của khách hàng 5](#_Toc88681616)

[6. Trả về 1 giá trị tính điểm của món ăn 5](#_Toc88681617)

[**VII.** VIEW 5](#_Toc88681618)

[1. Menu món ăn 5](#_Toc88681619)

[2. Tính tiền 5](#_Toc88681620)

[3. Thống kê 5](#_Toc88681621)

[CHƯƠNG 3: HƯỚNG DẪN SỬ DỤNG PHẦN MỀM 5](#_Toc88681622)

[**I.** Màn hình trang chủ 5](#_Toc88681623)

[**II.** Đăng nhập 6](#_Toc88681624)

[**III.** Đổi mật khẩu 7](#_Toc88681625)

[**IV.** Trang quản lý 8](#_Toc88681626)

[**V.** Trang tài khoản 9](#_Toc88681627)

[**VI.** Trang khách hàng 10](#_Toc88681628)

[**VII.** Trang nhân viên 12](#_Toc88681629)

[**VIII.** Trang món ăn 13](#_Toc88681630)

[**IX.** Trang nguyên liệu 13](#_Toc88681631)

[**X.** Trang loại món ăn 14](#_Toc88681632)

[**XI.** Trang công thức 15](#_Toc88681633)

[**XII.** Trang đặt món 17](#_Toc88681634)

[**XIII.** Trang đánh giá 17](#_Toc88681635)

[**XIV.** Thống kê 18](#_Toc88681636)

[**XV.** Địa điểm 19](#_Toc88681637)

# CHƯƠNG 1: ĐẶC TẢ HỆ THỐNG

1. **Mô tả hệ thống**
   1. **Thức ăn quan trọng như thế nào ?**

Thực phẩm là thứ nhu cầu không thể thiếu trong mỗi bên cạnh chúng ta, bên cạnh nước uống, oxi thì thực phẩm nó đóng vai trò không kém. Trên thị trường thực phẩm cung cấp cho con người nguồn năng lượng, dưỡng chất để duy trì sự sống và phát triển cơ thể một cách toàn diện từ thể lực cho tới trí tuệ. Từ đó mà nhiều món ăn ngon được ra đời và chính món ăn đã đưa con người chúng ta tới gần nhau hơn. Câu cửa miệng chúng ta thường hay nói rằng là muốn chinh phục trái tim của người mình yêu thì phải đi qua day dày. Quả thật câu nói này không sai.

Việc học nấu ăn đối với nhiều người quả thật không dễ dàng gì. Ngay từ việc chọn nguyên liệu, rồi cách nấu cũng là một bức tường thành dày và cao rất khó vượt qua. Việc ra đời của một phần mềm quản lý món ăn là rất cần thiết và tiện tích hơn nữa là hệ thống có tích sẵn việc đặt món ăn

* 1. **Mô tả dữ liệu phần mềm**

Để học nấu ăn hiệu quả, trước hết chúng ta cần có những kĩ năng về nhà bếp hay là kĩ năng về chọn nguyên liệu và công thức nấu món ăn đó sao cho hợp với đa số khẩu vị của mỗi người. Chính vì tầm quạ trọng như vậy mà phần mềm của em sẽ tập trung việc lưu trữ các công thức, món ăn, nguyên liệu , giá cả để mọi người có thể cùng học hỏi cũng như đóng góp xây dựng vào món ăn để món ăn ngày càng hoàn thiện hơn.

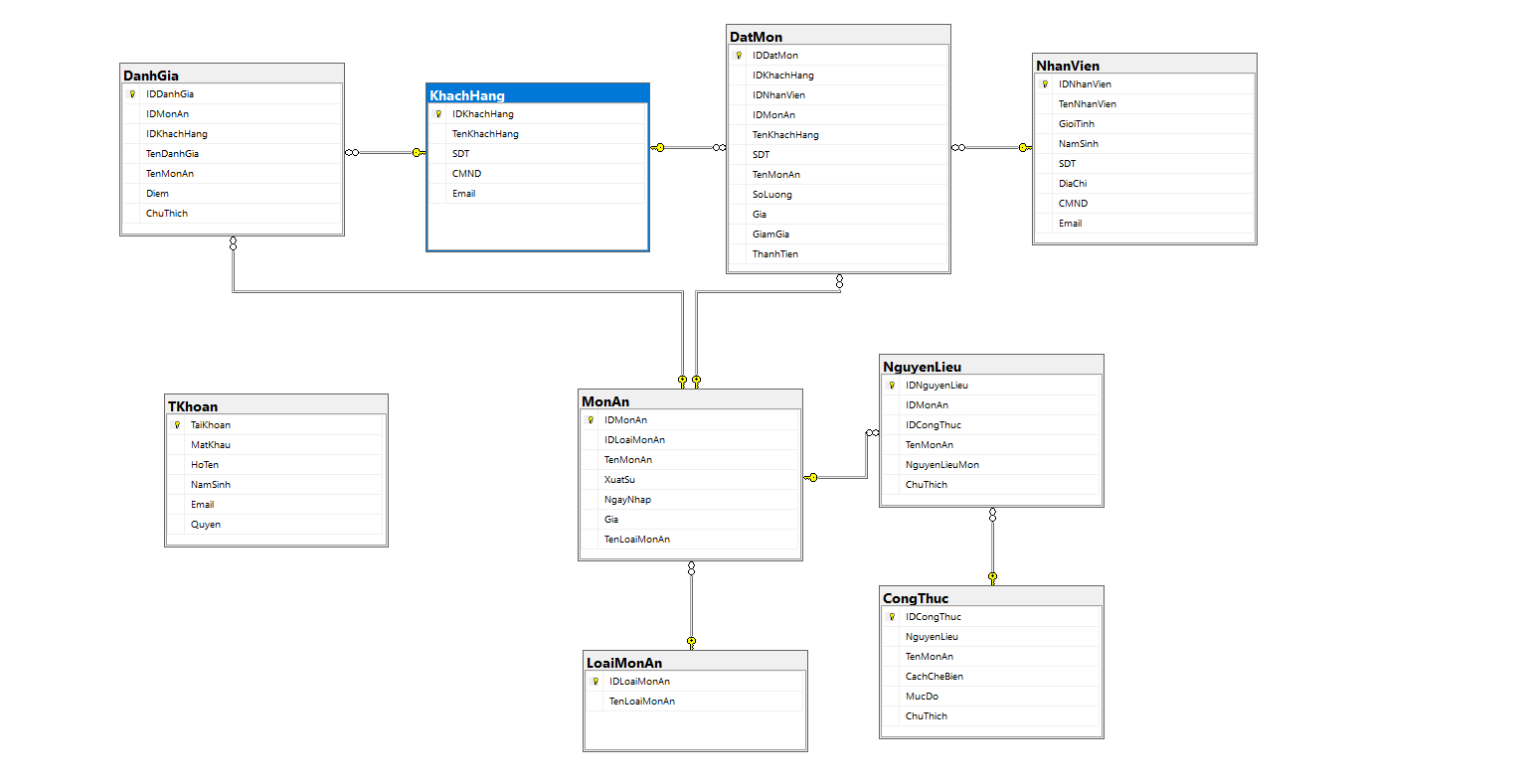
Phần mềm của em gồm có các thành phần chính:

* Quản lý thông tin khách hàng, nhân viên, quyền đăng nhập hệ thống: Lưu thông tin nhân viên, khách hàng, thông tin cũng như quyền hạn của mỗi người.
* Quản lý nguyên liệu: Đây là nơi cho phép người dùng luu trữ các nguyên liệu nấu ăn (có thể là nhiều nguyên liệu món ăn ngon)
* Quản lý món ăn: Chứa bao gồm các thông tin về món ăn, nguồn gốc, giá thành
* Quản lý loại món ăn: Món ăn này là thuộc thể loại món ăn gì (truyền thống, đường phố, ăn nhanh, v.v…)
* Quản lý công thức: Mấu chốt nằm ở đây, nơi chứa những công thức gia truyền của mỗi người, nơi mà mọi người cùng chia sẻ học hỏi
* Quản lý đặt món: Điểm đạc biệt là quán có thể sử dụng phần mềm cho việc kinh doanh như quản lý co việc đặt món
* Quản lý đánh giá: Dùng để mọi người cùng góp ý, giúp cho món ăn ngày càng hoàn thiện hơn
* Thống kê: Thống kê doanh thu và các chỉ số doanh thu
* Địa điểm: Nơi tìm kiếm các món ăn ở nhiều địa điểm khác nhau để mọi người thưởng thức

1. **Quy trình xử lý**

* Để có thể sử dụng phần mềm thì yêu cầu người dùng phải đăng nhập vào hệ thống, Trong phần mềm có quyền ADMIN cho phép sử dụng hầu hết tất cả các chức năng quản lý. Quyền còn lại ngoại trừ Quản lý thông tin khách hàng, nhân viên, quyền đăng nhập hệ thống thì sử dụng được hết các chức năng còn lại
* Tới đây khi mà muốn thêm, sửa hay xoá thì ADMIN có thể truy cập vào phần mềm và thực hiện các thao tác trên
* Các món ăn được bày bán cho khách hàng sẽ được lưu vào trong **Quản lý món ăn**, và món ăn đó có thể sẽ chứa trong phần **Quản lý công thức, Quản lý loại món ăn và Quản lý nguyên liệu** món ăn để cho nhân viên, đầu bếp mới vô làm có thể học hỏi cách nấu để phục vụ cho khách hàng, điều đặc biệt là khách hàng có thể học cách nấu món ăn nếu như khách hàng có thể nấu thông qua mục **Quản lý đánh giá**
* Khi mà khách hàng tới mua món ăn thì nhân viên sẽ đặt món dựa trên yêu cầu của khách hàng thông qua **Quản lý đặt món**, khi khách thanh toán thì ta sẽ xuất ra bill tính tiền và bên cạnh đó là sẽ cho qua mục **Thống kê** để biết được hôm nay việc kinh doanh như thế nào
* Khách có thể tra cứu các địa điểm quán ăn hoặc chi nhánh có món ăn tương tự vậy để thưởng thức

1. **Mô hình Diagram**



# CHƯƠNG 2: CẤU HÌNH VÀ CÀI ĐẶT HỆ THỐNG

1. **Tạo cơ sở dữ liệu và ràng buộc**

CREATE TABLE TKhoan(

TaiKhoan varchar(50) NOT NULL,

MatKhau nvarchar(50) NULL,

HoTen nvarchar(50) NULL,

NamSinh date NULL,

Email nvarchar(50) NULL,

Quyen varchar(15) NULL,

CONSTRAINT pk\_TKhoan PRIMARY KEY(TaiKhoan)

)

CREATE TABLE LoaiMonAn(

IDLoaiMonAn int NOT NULL,

TenLoaiMonAn nvarchar(50) NULL,

CONSTRAINT pk\_LoaiMonAn PRIMARY KEY(IDLoaiMonAn)

)

CREATE TABLE MonAn(

IDMonAn int NOT NULL,

IDLoaiMonAn int NOT NULL,

TenMonAn nvarchar(50) NULL,

XuatSu nvarchar(50) NULL,

NgayNhap date NULL,

Gia money NULL,

TenMonAn nvarchar(50) NULL,

CONSTRAINT pk\_MonAn PRIMARY KEY(IDMonAn)

)

CREATE TABLE NguyenLieu(

IDNguyenLieu int NOT NULL,

IDMonAn int NOT NULL,

IDCongThuc int NOT NULL,

TenMonAn nvarchar(50) NULL,

NguyenLieuMon ntext NULL,

ChuThich nvarchar(300) NULL,

CONSTRAINT pk\_NguyenLieu PRIMARY KEY(IDNguyenLieu)

)

CREATE TABLE CongThuc(

IDCongThuc int NOT NULL,

NguyenLieu ntext NOT NULL,

TenMonAn nvarchar(50) NULL,

CachCheBien ntext NULL,

MucDo nvarchar(50) NULL,

ChuThich nvarchar(300) NULL,

CONSTRAINT pk\_CongThuc PRIMARY KEY(IDCongThuc)

)

CREATE TABLE DanhGia(

IDDanhGia int NOT NULL,

IDMonAn int NOT NULL,

IDKhachHang int NOT NULL,

TenDanhGia nvarchar(50) NULL,

TenMonAn nvarchar(50) NULL,

Diem float NULL,

ChuThich nvarchar(300) NULL,

CONSTRAINT pk\_DanhGia PRIMARY KEY(IDDanhGia)

)

CREATE TABLE KhachHang(

IDKhachHang int NOT NULL,

TenKhachHang nvarchar(50) NULL,

SDT int NULL,

CMND int NULL,

Email nvarchar(50) NULL,

CONSTRAINT pk\_KhachHang PRIMARY KEY(IDKhachHang)

)

CREATE TABLE DatMon(

IDDatMon int NOT NULL,

IDKhachHang int NOT NULL,

IDNhanVien int NOT NULL,

IDMonAn int NOT NULL,

TenKhachHang nvarchar(50) NULL,

SDT int NULL,

TenMonAn nvarchar(50) NULL,

SoLuong int NULL,

Gia float NULL,

GiamGia float NULL,

ThanhTien float NULL,

CONSTRAINT pk\_DatMon PRIMARY KEY(IDDatMon)

)

CREATE TABLE NhanVien(

IDNhanVien int NOT NULL,

TenNhanVien nvarchar(50) NULL,

GioiTinh nvarchar(50) NULL,

NamSinh date NULL,

SDT int NULL,

DiaChi nvarchar(500) NULL,

CMND int NULL,

Email nvarchar(50) NULL,

CONSTRAINT pk\_NhanVien PRIMARY KEY(IDNhanVien)

)

* + - * Ràng buộc khóa ngoại:

ALTER TABLE dbo.NguyenLieu ADD CONSTRAINT FK\_NguyenLieu\_CongThuc FOREIGN KEY(IDCongThuc) REFERENCES dbo.CongThuc(IDCongThuc)

ALTER TABLE dbo.NguyenLieu ADD CONSTRAINT FK\_NguyenLieu\_MonAn FOREIGN KEY(IDMonAn) REFERENCES dbo.MonAn(IDMonAn)

ALTER TABLE dbo.MonAn ADD CONSTRAINT FK\_MonAn\_LoaiMonAn FOREIGN KEY(IDLoaiMonAn) REFERENCES dbo.LoaiMonAn(IDLoaiMonAn)

ALTER TABLE dbo.DanhGia ADD CONSTRAINT FK\_DanhGia\_MonAn FOREIGN KEY(IDMonAn) REFERENCES dbo.MonAn(IDMonAn)

ALTER TABLE dbo.DatMon ADD CONSTRAINT FK\_DatMon\_NhanVien FOREIGN KEY(IDNhanvien) REFERENCES dbo.NhanVien(IDNhanVien)

ALTER TABLE dbo.DatMon ADD CONSTRAINT FK\_DatMon\_KhachHang FOREIGN KEY(IDKhachHang) REFERENCES dbo.KhachHang(IDKhachHang)

ALTER TABLE dbo.DatMon ADD CONSTRAINT FK\_DatMon\_MonAn FOREIGN KEY(IDMonAn) REFERENCES dbo.MonAn(IDMonAn)

ALTER TABLE dbo.DanhGia ADD CONSTRAINT FK\_DanhGia\_KhachHang FOREIGN KEY(IDKhachHang) REFERENCES dbo.KhachHang(IDKhachHang)

1. **STORE PROCEDURE**
2. **Tài khoản**

CREATE PROCEDURE USP\_Login

@taikhoan nvarchar(50),

@matkhau nvarchar(50)

AS

BEGIN

SELECT Quyen, HoTen FROM TKhoan WHERE TaiKhoan = @taikhoan AND MatKhau = @matkhau;

END

CREATE PROCEDURE XoaTaiKhoan

@taikhoan nvarchar(50)

AS

BEGIN

DELETE FROM TKhoan WHERE TaiKhoan = @taikhoan;

IF @@REMSERVER > 0 RETURN 1

ELSE RETURN 0;

END

CREATE PROCEDURE ThemTaiKhoan

@taikhoan nvarchar(100),

@matkhau nvarchar(100),

@hoten nvarchar(100),

@namsinh date,

@email nvarchar(100),

@quyen nvarchar(100)

AS

BEGIN

INSERT INTO TKhoan

(

TaiKhoan,

MatKhau,

HoTen,

NamSinh,

Email,

Quyen

)

VALUES

(

@taikhoan, @matkhau, @hoten, @namsinh, @email, @quyen

);

IF @@ROWCOUNT > 0 RETURN 1

ELSE RETURN 0;

END

CREATE PROCEDURE dbo.ChangePassword

@taikhoan nvarchar(50),

@matkhaucu nvarchar(100),

@matkhaumoi nvarchar(10)

AS

BEGIN

UPDATE TKhoan

SET MatKhau = @matkhaumoi

WHERE TaiKhoan = @taikhoan

AND MatKhau = @matkhaucu

IF @@ROWCOUNT > 0

RETURN 1

RETURN 0

END

1. **Nhân viên**

CREATE PROCEDURE ThemNhanVien

@idnhanvien int,

@tennhanvien nvarchar(100),

@gioitinh nvarchar(100),

@namsinh date,

@sdt int,

@diachi nvarchar(100),

@cmnd int,

@email nvarchar(100)

AS

BEGIN

INSERT INTO NhanVien

(

IDNhanVien,

TenNhanVien,

GioiTinh,

NamSinh,

SDT,

DiaChi,

CMND,

Email

)

VALUES

(

@idnhanvien,@tennhanvien,@gioitinh,@namsinh,@sdt,@diachi,@cmnd,@email

);

IF @@ROWCOUNT > 0 RETURN 1

ELSE RETURN 0;

END

CREATE PROCEDURE SuaNhanVien

@idnhanvien int,

@tennhanvien nvarchar(100),

@gioitinh nvarchar(100),

@namsinh date,

@sdt int,

@diachi nvarchar(100),

@cmnd int,

@email nvarchar(100)

AS

BEGIN

UPDATE NhanVien

SET

TenNhanVien = @tennhanvien,

GioiTinh = @gioitinh,

NamSinh = @namsinh,

SDT = @sdt,

DiaChi = @diachi,

CMND = @cmnd,

Email = @email

WHERE IDNhanVien = @idnhanvien;

IF @@ROWCOUNT > 0 RETURN 1

ELSE RETURN 0;

END

CREATE PROCEDURE XoaNhanVien

@idnhanvien int

AS

BEGIN

DELETE FROM NhanVien WHERE IDNhanVien = @idnhanvien;

IF @@REMSERVER > 0 RETURN 1

ELSE RETURN 0;

END

CREATE PROCEDURE TimKiemNhanVien

@searchTen nvarchar(100)

AS

BEGIN

SELECT \*

FROM NhanVien

WHERE TenNhanVien LIKE @searchTen+'%' or Email LIKE @searchTen+'%' or IDNhanVien LIKE @searchTen+'%' or GioiTinh LIKE @searchTen+'%'

END

CREATE PROCEDURE LayNhanVien

AS

BEGIN

SELECT \* FROM NhanVien

END

1. **Khách hàng**

CREATE PROCEDURE ThemKhachHang

@idkhachhang int,

@tenkhachhang nvarchar(100),

@sdt int,

@cmnd int,

@email nvarchar(100)

AS

BEGIN

INSERT INTO KhachHang

(

IDKhachHang,

TenKhachHang,

SDT,

CMND,

Email

)

VALUES

(

@idkhachhang,@tenkhachhang,@sdt,@cmnd,@email

);

IF @@ROWCOUNT > 0 RETURN 1

ELSE RETURN 0;

END

CREATE PROCEDURE SuaKhachHang

@idkhachhang int,

@tenkhachhang nvarchar(100),

@sdt int,

@cmnd int,

@email nvarchar(100)

AS

BEGIN

UPDATE KhachHang

SET

TenKhachHang = @tenkhachhang,

SDT = @sdt,

CMND = @cmnd,

Email = @email

WHERE IDKhachHang = @idkhachhang;

IF @@ROWCOUNT > 0 RETURN 1

ELSE RETURN 0;

END

CREATE PROCEDURE XoaKhachHang

@idkhachhang int

AS

BEGIN

DELETE FROM KhachHang WHERE IDKhachHang = @idkhachhang;

IF @@REMSERVER > 0 RETURN 1

ELSE RETURN 0;

END

CREATE PROCEDURE TimKiemKhachHang

@searchTen nvarchar(100)

AS

BEGIN

SELECT \*

FROM KhachHang

WHERE TenKhachHang LIKE @searchTen+'%' or Email LIKE @searchTen+'%'

END

CREATE PROCEDURE LayKhachHang

AS

BEGIN

SELECT \* FROM KhachHang

END

1. **Nguyên liệu**

CREATE PROCEDURE ThemNguyenLieu

@idnguyenlieu int,

@idmonan int,

@idcongthuc int,

@tenmonan nvarchar(100),

@nguyenlieumon nvarchar(4000),

@chuthich nvarchar(300)

AS

BEGIN

INSERT INTO NguyenLieu

(

IDNguyenLieu,

IDMonAn,

IDCongThuc,

TenMonAn,

NguyenLieuMon,

ChuThich

)

VALUES

(

@idnguyenlieu,@idmonan,@idcongthuc,@tenmonan,@nguyenlieumon,@chuthich

);

IF @@ROWCOUNT > 0 RETURN 1

ELSE RETURN 0;

END

CREATE PROCEDURE SuaNguyenLieu

@idnguyenlieu int,

@idmonan int,

@idcongthuc int,

@tenmonan nvarchar(100),

@nguyenlieumon nvarchar(4000),

@chuthich nvarchar(300)

AS

BEGIN

UPDATE NguyenLieu

SET

IDMonAn = @idmonan,

IDCongThuc = @idcongthuc,

TenMonAn = @tenmonan,

NguyenLieuMon = @nguyenlieumon,

ChuThich = @chuthich

WHERE IDNguyenLieu = @idnguyenlieu;

IF @@ROWCOUNT > 0 RETURN 1

ELSE RETURN 0;

END

CREATE PROCEDURE XoaNguyenLieu

@idnguyenlieu int

AS

BEGIN

DELETE FROM NguyenLieu WHERE IDNguyenLieu = @idnguyenlieu;

IF @@REMSERVER > 0 RETURN 1

ELSE RETURN 0;

END

CREATE PROCEDURE TimKiemNguyenLieu

@searchTen nvarchar(100)

AS

BEGIN

SELECT \*

FROM NguyenLieu

WHERE TenMonAn LIKE @searchTen+'%'

END

CREATE PROCEDURE LayNguyenLieu

AS

BEGIN

SELECT \* FROM NguyenLieu

END

1. **Công thức**

CREATE PROCEDURE ThemCongThuc

@idcongthuc int,

@nguyenlieu nvarchar(4000),

@tenmonan nvarchar(100),

@cachchebien nvarchar(4000),

@mucdo nvarchar(50),

@chuthich nvarchar(50)

AS

BEGIN

INSERT INTO CongThuc

(

IDCongThuc,

NguyenLieu,

TenMonAn,

CachCheBien,

MucDo,

ChuThich

)

VALUES

(

@idcongthuc, @nguyenlieu, @tenmonan, @cachchebien, @mucdo, @chuthich

);

IF @@ROWCOUNT > 0 RETURN 1

ELSE RETURN 0;

END

CREATE PROCEDURE SuaCongThuc

@idcongthuc int,

@nguyenlieu nvarchar(4000),

@tenmonan nvarchar(100),

@cachchebien nvarchar(4000),

@mucdo nvarchar(50),

@chuthich nvarchar(50)

AS

BEGIN

UPDATE CongThuc

SET

NguyenLieu = @nguyenlieu,

TenMonAn = @tenmonan,

CachCheBien = @cachchebien,

MucDo = @mucdo,

ChuThich = @chuthich

WHERE IDCongThuc = @idcongthuc;

IF @@ROWCOUNT > 0 RETURN 1

ELSE RETURN 0;

END

CREATE PROCEDURE XoaCongThuc

@idcongthuc int

AS

BEGIN

DELETE FROM CongThuc WHERE IDCongThuc = @idcongthuc;

IF @@REMSERVER > 0 RETURN 1

ELSE RETURN 0;

END

CREATE PROCEDURE LayCongThuc

AS

BEGIN

SELECT \* FROM CongThuc

END

CREATE PROCEDURE TimKiemCongThuc

@searchTen nvarchar(100)

AS

BEGIN

SELECT \*

FROM CongThuc

WHERE TenMonAn LIKE @searchTen+'%' or MucDo LIKE @searchTen+'%' or CachCheBien LIKE @searchTen+'%'

END

1. **Món ăn**

CREATE PROCEDURE ThemMonAn

@idmonan int,

@idloaimonan int,

@tenmonan nvarchar(100),

@tenloaimonan nvarchar(100),

@xuatsu nvarchar(100),

@ngaynhap date,

@gia money

AS

BEGIN

INSERT INTO MonAn

(

IDMonAn,

IDLoaiMonAn,

TenMonAn,

TenLoaiMonAn,

XuatSu,

NgayNhap,

Gia

)

VALUES

(

@idmonan ,@idloaimonan ,@tenmonan, @tenloaimonan, @xuatsu, @ngaynhap, @gia

);

IF @@ROWCOUNT > 0 RETURN 1

ELSE RETURN 0;

END

CREATE PROCEDURE SuaMonAn

@idmonan int,

@idloaimonan int,

@tenmonan nvarchar(100),

@tenloaimonan nvarchar(100),

@xuatsu nvarchar(100),

@ngaynhap date,

@gia money

AS

BEGIN

UPDATE MonAn

SET

IDLoaiMonAn = @idloaimonan,

TenMonAn = @tenmonan,

TenLoaiMonAn = @tenloaimonan,

XuatSu = @xuatsu,

NgayNhap = @ngaynhap,

Gia = @gia

WHERE IDMonAn = @idmonan;

IF @@ROWCOUNT > 0 RETURN 1

ELSE RETURN 0;

END

CREATE PROCEDURE XoaMonAn

@idmonan int

AS

BEGIN

DELETE FROM MonAn WHERE IDMonAn = @idmonan;

IF @@REMSERVER > 0 RETURN 1

ELSE RETURN 0;

END

CREATE PROCEDURE TimKiemMonAn

@searchTen nvarchar(100)

AS

BEGIN

SELECT \*

FROM MonAn

WHERE TenMonAn LIKE @searchTen+'%' or XuatSu LIKE @searchTen+'%' or TenLoaiMonAn LIKE @searchTen+'%'

END

CREATE PROCEDURE LayMonAn

AS

BEGIN

SELECT \* FROM MonAn

END

1. **Loại món ăn**

CREATE PROCEDURE ThemLoaiMonAn

@idloaimonan int,

@tenloaimonan nvarchar(100)

AS

BEGIN

INSERT INTO LoaiMonAn

(

IDLoaiMonAn,

TenLoaiMonAn

)

VALUES

(

@idloaimonan ,@tenloaimonan

);

IF @@ROWCOUNT > 0 RETURN 1

ELSE RETURN 0;

END

CREATE PROCEDURE SuaLoaiMonAn

@idloaimonan int,

@tenloaimonan nvarchar(100)

AS

BEGIN

UPDATE LoaiMonAn

SET

TenLoaiMonAn = @tenloaimonan

WHERE IDLoaiMonAn = @idloaimonan;

IF @@ROWCOUNT > 0 RETURN 1

ELSE RETURN 0;

END

CREATE PROCEDURE XoaLoaiMonAn

@idloaimonan int

AS

BEGIN

DELETE FROM LoaiMonAn WHERE IDLoaiMonAn = @idloaimonan;

IF @@REMSERVER > 0 RETURN 1

ELSE RETURN 0;

END

CREATE PROCEDURE LayLoaiMonAn

AS

BEGIN

SELECT \* FROM LoaiMonAn

END

1. **Đặt món**

CREATE PROCEDURE ThemDatMon

@iddatmon int,

@idkhachhang int,

@idnhanvien int,

@idmonan int,

@tenkhachhang nvarchar(100),

@sdt int,

@tenmonan nvarchar(100),

@soluong int,

@gia float,

@giamgia float

AS

BEGIN

INSERT INTO DatMon

(

IDDatMon,

IDKhachHang,

IDNhanVien,

IDMonAn,

TenKhachHang,

SDT,

TenMonAn,

SoLuong,

Gia,

GiamGia

)

VALUES

(

@iddatmon, @idkhachhang, @idnhanvien, @idmonan, @tenkhachhang, @sdt, @tenmonan, @soluong, @gia, @giamgia

);

IF @@ROWCOUNT > 0 RETURN 1

ELSE RETURN 0;

END

CREATE PROCEDURE SuaDatMon

@iddatmon int,

@idkhachhang int,

@idnhanvien int,

@idmonan int,

@tenkhachhang nvarchar(100),

@sdt int,

@tenmonan nvarchar(100),

@soluong int,

@gia float,

@giamgia float,

@thanhtien float

AS

BEGIN

UPDATE dbo.DatMon

SET

IDKhachHang = @idkhachhang,

IDNhanVien = @idnhanvien,

IDMonAn = @idmonan,

TenKhachHang = @tenkhachhang,

SDT = @sdt,

TenMonAn = @tenmonan,

SoLuong = @soluong,

Gia = @gia,

GiamGia = @giamgia,

ThanhTien = @thanhtien

WHERE IDDatMon = @iddatmon;

IF @@ROWCOUNT > 0 RETURN 1

ELSE RETURN 0;

END

CREATE PROCEDURE XoaDatMon

@iddatmon int

AS

BEGIN

DELETE FROM dbo.DatMon WHERE IDDatMon = @iddatmon;

IF @@REMSERVER > 0 RETURN 1

ELSE RETURN 0;

END

CREATE PROCEDURE TimKiemDatMon

@searchTen nvarchar(100)

AS

BEGIN

SELECT \*

FROM dbo.DatMon

WHERE TenMonAn LIKE @searchTen+'%' or TenKhachHang LIKE @searchTen+'%'

END

CREATE PROCEDURE LayDatMon

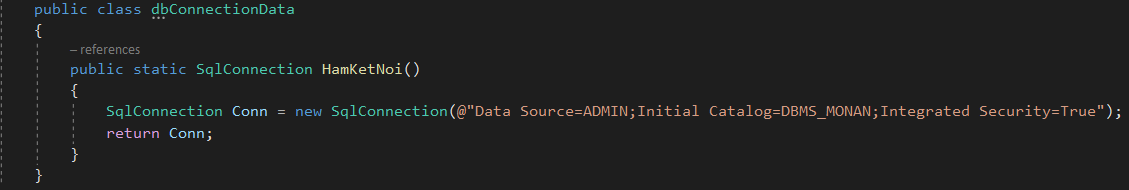
AS

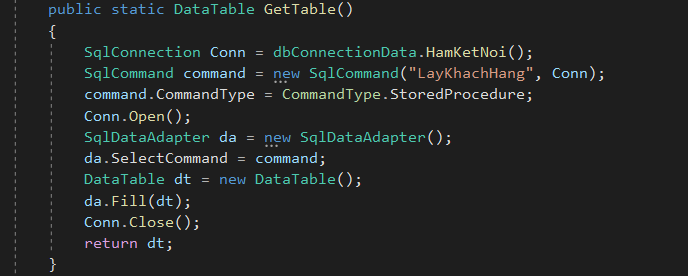
BEGIN

SELECT \* FROM dbo.BILL

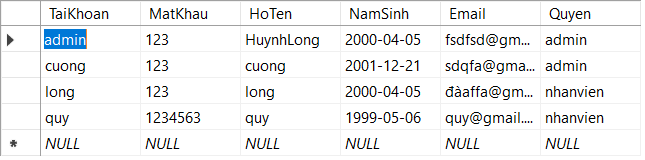
END

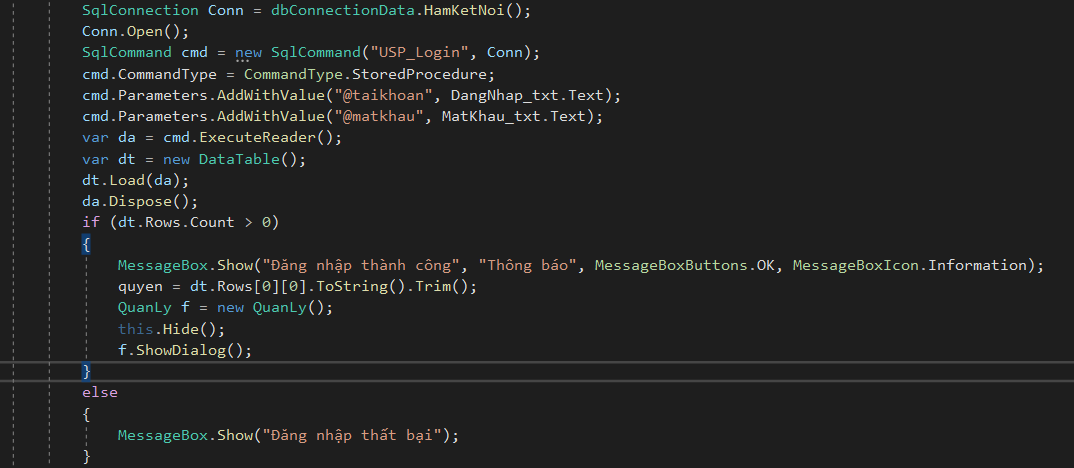
1. **KẾT NỐI CSDL**

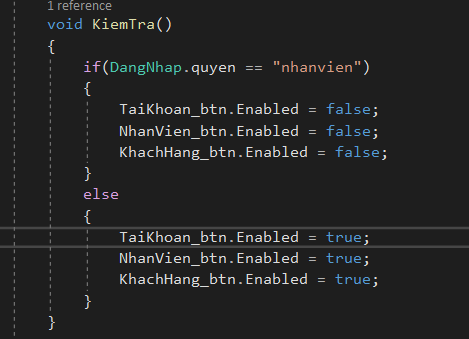




1. **PHÂN QUYỀN**

****





1. **TRIGGER VÀ TRANSACTION**
   1. Kiểm tra gmail tài khoản có hợp lệ không

CREATE TRIGGER KiemTra\_EmailTK

ON dbo.TKhoan

FOR UPDATE,INSERT

AS

BEGIN

DECLARE @email varchar(100)

SELECT @email=inserted.Email FROM inserted

IF (@email NOT LIKE '%\_@%\_.\_\_%')

BEGIN

PRINT N'Email không hợp lệ'

ROLLBACK TRANSACTION

END

END

* 1. Kiểm tra gmail nhân viên có hợp lệ không

CREATE TRIGGER KiemTra\_Email

ON dbo.NhanVien

FOR UPDATE,INSERT

AS

BEGIN

DECLARE @email varchar(100)

SELECT @email=inserted.Email FROM inserted

IF (@email NOT LIKE '%\_@%\_.\_\_%')

BEGIN

PRINT N'Email không hợp lệ'

ROLLBACK TRANSACTION

END

END

* 1. Kiểm tra tên nhân viên không trùng nhau

CREATE TRIGGER dbo.KiemTra\_TenNhanVien

ON dbo.NhanVien

FOR UPDATE,INSERT

AS

BEGIN

DECLARE @NAME AS NCHAR(50), @TEMP AS INT

SELECT @NAME=inserted.TenNhanVien FROM inserted

SELECT @TEMP=COUNT(\*) FROM dbo.NhanVien

WHERE TenNhanVien=@NAME

IF (@TEMP>1)

BEGIN

PRINT N'Tên nhân viên đã tồn tại'

ROLLBACK TRANSACTION

END

END

* 1. Kiểm tra số CMND không vượt qua 9 số

CREATE TRIGGER dbo.KiemTra\_CMND

ON dbo.NhanVien

FOR UPDATE,INSERT

AS

BEGIN

DECLARE @cmnd AS nvarchar(50)

SELECT @cmnd=inserted.CMND FROM inserted

IF LEN(@cmnd) < 9

BEGIN

PRINT N'CMND phải đủ 9 số'

ROLLBACK TRANSACTION

END

END

* 1. Kiểm tra số điện thoại phải đủ 10 số

CREATE TRIGGER dbo.KiemTra\_SDT

ON dbo.NhanVien

FOR UPDATE,INSERT

AS

BEGIN

DECLARE @sdt AS nvarchar(50)

SELECT @sdt=inserted.SDT FROM inserted

IF LEN(@sdt) < 10

BEGIN

PRINT N'SDT phải đủ 10 số'

ROLLBACK TRANSACTION

END

END

* 1. Kiểm tra tên khách hàng không được trùng

CREATE TRIGGER dbo.KiemTra\_TenKhachHang

ON dbo.KhachHang

FOR UPDATE,INSERT

AS

BEGIN

DECLARE @NAME AS NCHAR(50), @TEMP AS INT

SELECT @NAME=inserted.TenKhachHang FROM inserted

SELECT @TEMP=COUNT(\*) FROM dbo.KhachHang

WHERE TenKhachHang=@NAME

IF (@TEMP>1)

BEGIN

PRINT N'Tên khách hàng đã tồn tại'

ROLLBACK TRANSACTION

END

END

* 1. Kiểm tra gmail khách hàng có hợp lệ không

CREATE TRIGGER KiemTra\_EmailKH

ON dbo.KhachHang

FOR UPDATE,INSERT

AS

BEGIN

DECLARE @email varchar(100)

SELECT @email=inserted.Email FROM inserted

IF (@email NOT LIKE '%\_@%\_.\_\_%')

BEGIN

PRINT N'Email không hợp lệ'

ROLLBACK TRANSACTION

END

END

* 1. Kiểm tra SDT khách hàng đúng 10 số

CREATE TRIGGER dbo.KiemTra\_SDTKH

ON dbo.KhachHang

FOR UPDATE,INSERT

AS

BEGIN

DECLARE @sdt AS nvarchar(50)

SELECT @sdt=inserted.SDT FROM inserted

IF LEN(@sdt) < 10

BEGIN

PRINT N'SDT phải đủ 10 số'

ROLLBACK TRANSACTION

END

END

* 1. Kiểm tra tên món ăn không được giống nhau

CREATE TRIGGER dbo.KiemTra\_TenMonAn

ON dbo.NguyenLieu

FOR UPDATE,INSERT

AS

BEGIN

DECLARE @NAME AS NCHAR(50), @TEMP AS INT

SELECT @NAME=inserted.TenMonAn FROM inserted

SELECT @TEMP=COUNT(\*) FROM dbo.NguyenLieu

WHERE TenMonAn=@NAME

IF (@TEMP>1)

BEGIN

PRINT N'Tên Món ăn đã tồn tại'

ROLLBACK TRANSACTION

END

END

* 1. Kiểm tra không cho nhập giá trị vào thứ 7 và chủ nhật

CREATE TRIGGER ThoiGian

ON dbo.MonAn

AFTER INSERT, UPDATE, DELETE

AS

IF DATEPART(DW, GETDATE()) = 7 OR DATEPART(DW, GETDATE()) = 1

BEGIN

ROLLBACK

RAISERROR(N'Không được phép cập nhật dữ liệu!', 16, 1)

RETURN

END

* 1. Kiểm tra giá của món ăn phải giống nhau

CREATE TRIGGER dbo.KiemTra\_Gia

ON dbo.MonAn

FOR UPDATE,INSERT

AS

IF (SELECT MonAn.Gia FROM MonAn inner join inserted on MonAn.Gia = inserted.Gia) != (SELECT DatMon.Gia FROM DatMon inner join inserted on DatMon.Gia = inserted.Gia)

BEGIN

Print N'Giá của món ăn và đặt món phải giống nhau'

RollBack Transaction

END

* 1. Kiểm tra loại món ăn không được giống nhau

CREATE TRIGGER dbo.KiemTra\_TenLoaiMonAn

ON dbo.LoaiMonAn

FOR UPDATE,INSERT

AS

BEGIN

DECLARE @NAME AS NCHAR(50), @TEMP AS INT

SELECT @NAME=inserted.TenLoaiMonAn FROM inserted

SELECT @TEMP=COUNT(\*) FROM dbo.LoaiMonAn

WHERE TenLoaiMonAn=@NAME

IF (@TEMP>1)

BEGIN

PRINT N'Loại món ăn đã tồn tại'

ROLLBACK TRANSACTION

END

END

* 1. Tự thanh toán tiền

CREATE TRIGGER [dbo].[TuDongTinhTien]

ON [dbo].[DatMon]

AFTER INSERT, UPDATE

AS

BEGIN

UPDATE dbo.DatMon set ThanhTien = (dbo.DatMon.SoLuong \* dbo.DatMon.Gia - (dbo.DatMon.SoLuong \* dbo.DatMon.Gia \* dbo.DatMon.GiamGia)) from dbo.DatMon, inserted where dbo.DatMon.IDDatMon = inserted.IDDatMon

END

1. **FUNCTION**
2. Trả về kết quả của 1 bảng để cho biết được KPI của nhân viên

CREATE FUNCTION [dbo].[NhanVienCham]

(

@idnhanvien int

)

RETURNS TABLE

AS

RETURN

(

(select NhanVien.TenNhanVien, DatMon.ThanhTien from NhanVien, DatMon where NhanVien.IDNhanVien = @idnhanvien and NhanVien.IDNhanVien = DatMon.IDNhanVien)

)

1. Trả về kết quả là 1 bảng công thức món ăn

CREATE function [dbo].[XemCongThucMonAn](@idcongthuc nvarchar(50))

RETURNS TABLE

AS

RETURN

( select NguyenLieu.NguyenLieuMon, CongThuc.CachCheBien, CongThuc.MucDo, NguyenLieu.TenMonAn

from NguyenLieu, CongThuc

where NguyenLieu.IDCongThuc = @idcongthuc and NguyenLieu.IDCongThuc = CongThuc.IDCongThuc)

1. Trả về 1 bảng về nguồn gốc của món ăn

CREATE function [dbo].[XuatSuMonAn](@idmonan nvarchar(50))

RETURNS TABLE

AS

RETURN

( select MonAn.TenMonAn, MonAn.XuatSu

from MonAn

where MonAn.IDMonAn = @idmonan)

1. Trả về 1 giá trị thể hiện giá của món ăn

CREATE FUNCTION [dbo].[GiaMonAn](@tenmonan nvarchar(50))

RETURNS float

AS

BEGIN

RETURN(select MonAn.Gia as TienMonAn

FROM MonAn

WHERE MonAn.TenMonAn = @tenmonan)

END

1. Trả về 1 giá trị thể hiện tổng số tiền của khách hàng

create function dbo.TongTienKhach(@tenkhachhang nvarchar(50))

returns float

as

begin

return(select SUM(DatMon.ThanhTien) as TongTien

from DatMon

where DatMon.TenKhachHang = @tenkhachhang)

end

1. Trả về 1 giá trị tính điểm của món ăn

CREATE FUNCTION [dbo].[MaxDiem](@tendanhgia nvarchar(50))

RETURNS float

AS

BEGIN

RETURN(select MAX(DanhGia.Diem) as MaxDiem

FROM DanhGia

WHERE DanhGia.TenDanhGia = @tendanhgia)

END

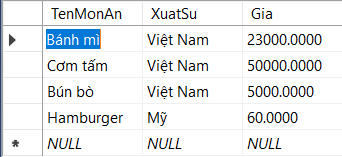
1. **VIEW**
2. Menu món ăn

CREATE VIEW MenuMonAn

AS

SELECT TenMonAn, XuatSu, Gia

FROM dbo.MonAn



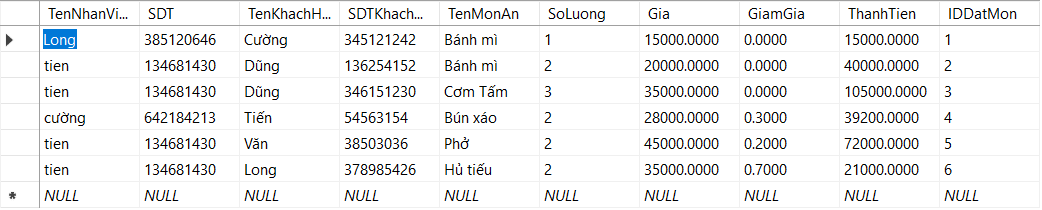
1. Tính tiền

CREATE VIEW TinhTien

AS

SELECT dbo.NhanVien.TenNhanVien, dbo.NhanVien.SDT, dbo.DatMon.TenKhachHang, dbo.DatMon.SDT AS SDTKhachHang, dbo.DatMon.TenMonAn, dbo.DatMon.SoLuong, dbo.DatMon.Gia, dbo.DatMon.GiamGia, dbo.DatMon.ThanhTien, dbo.DatMon.IDDatMon

FROM dbo.NhanVien INNER JOIN dbo.DatMon ON dbo.NhanVien.IDNhanVien = dbo.DatMon.IDNhanVien



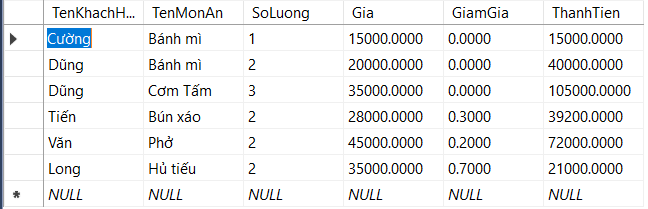
1. Thống kê

CREATE VIEW BILL

AS

SELECT TenKhachHang, TenMonAn, SoLuong, Gia, GiamGia, ThanhTien

FROM dbo.DatMon



# CHƯƠNG 3: HƯỚNG DẪN SỬ DỤNG PHẦN MỀM

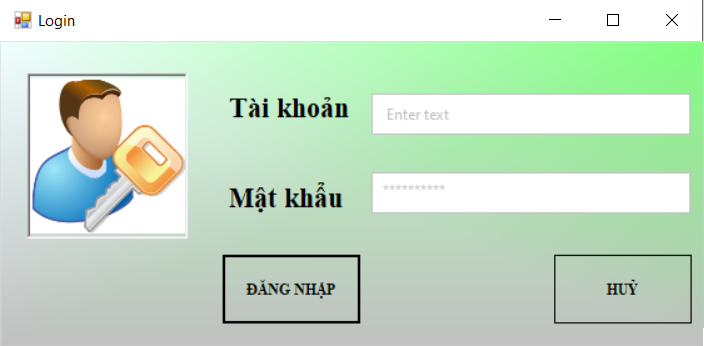
1. Màn hình trang chủ

* Khi mở phần mềm thì sẽ hiện ra trang chủ như hình bên dưới:
* Ở đây có 3 nút là Đăng nhập, đổi mật khẩu, thoát



1. Đăng nhập

* Sau khi chọn đăng nhập thì trang đăng nhập hiện ra
* Tại đây, có hai loại tài khoản là admin và nhân viên
* Khi đăng nhập vào sẽ chạy ra trang quản lý các chức năng chương trình



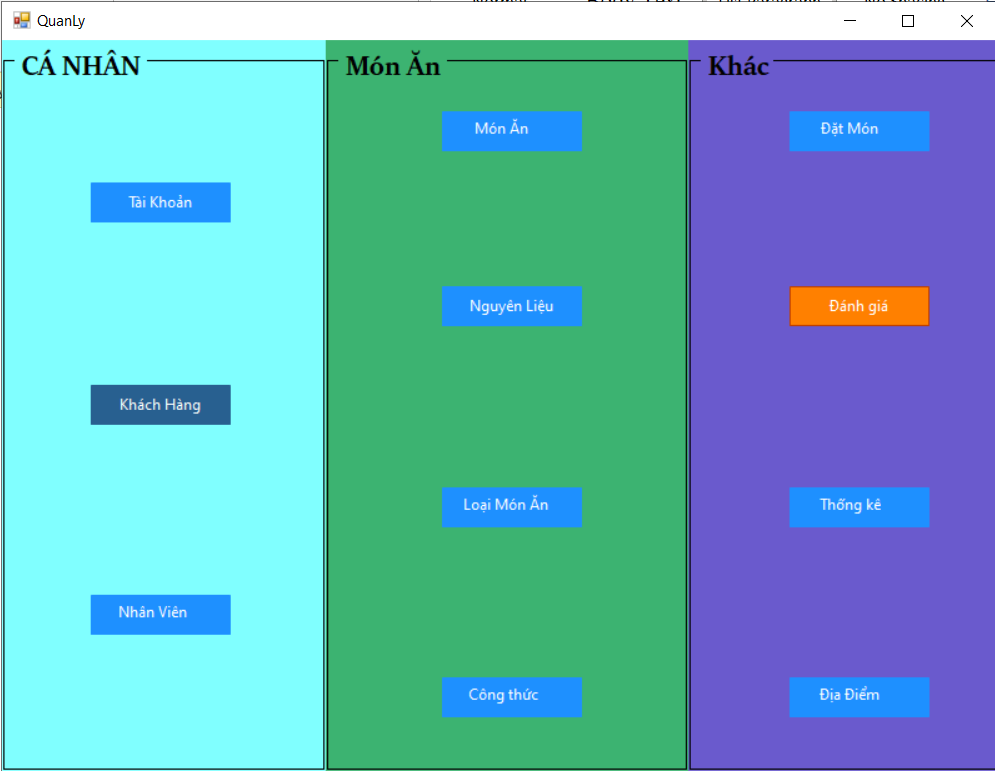
1. Đổi mật khẩu

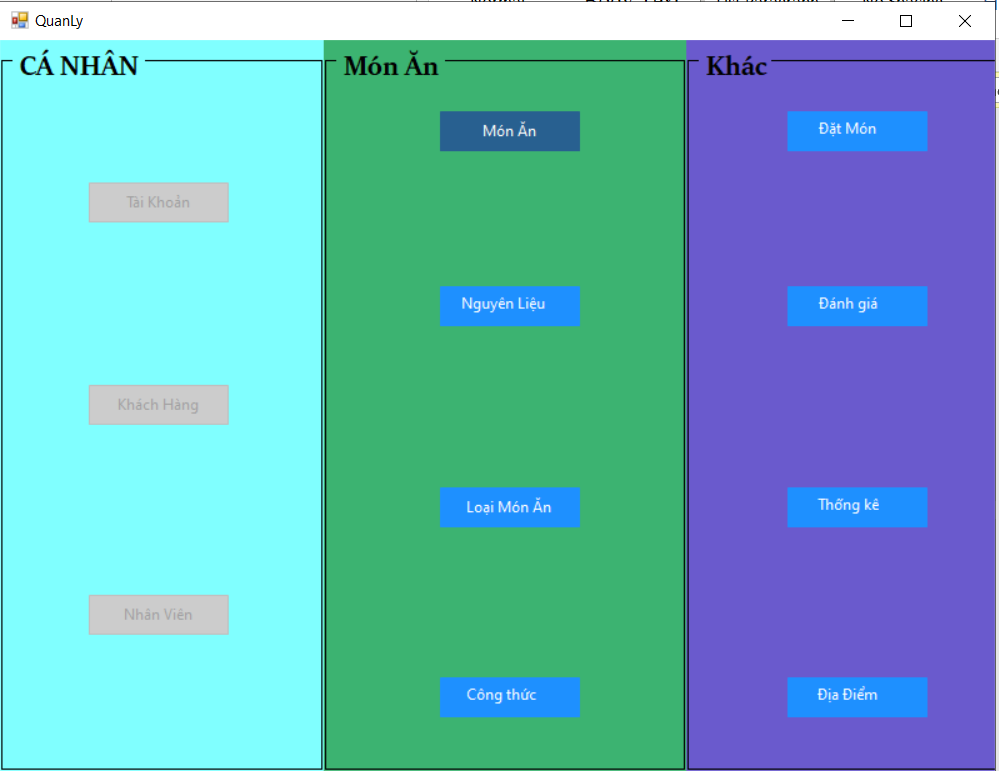
* Ở mục này tài khoản được cấp thì mỗi người có thể được phép đổi mật khảu theo ý mình muốn



1. Trang quản lý

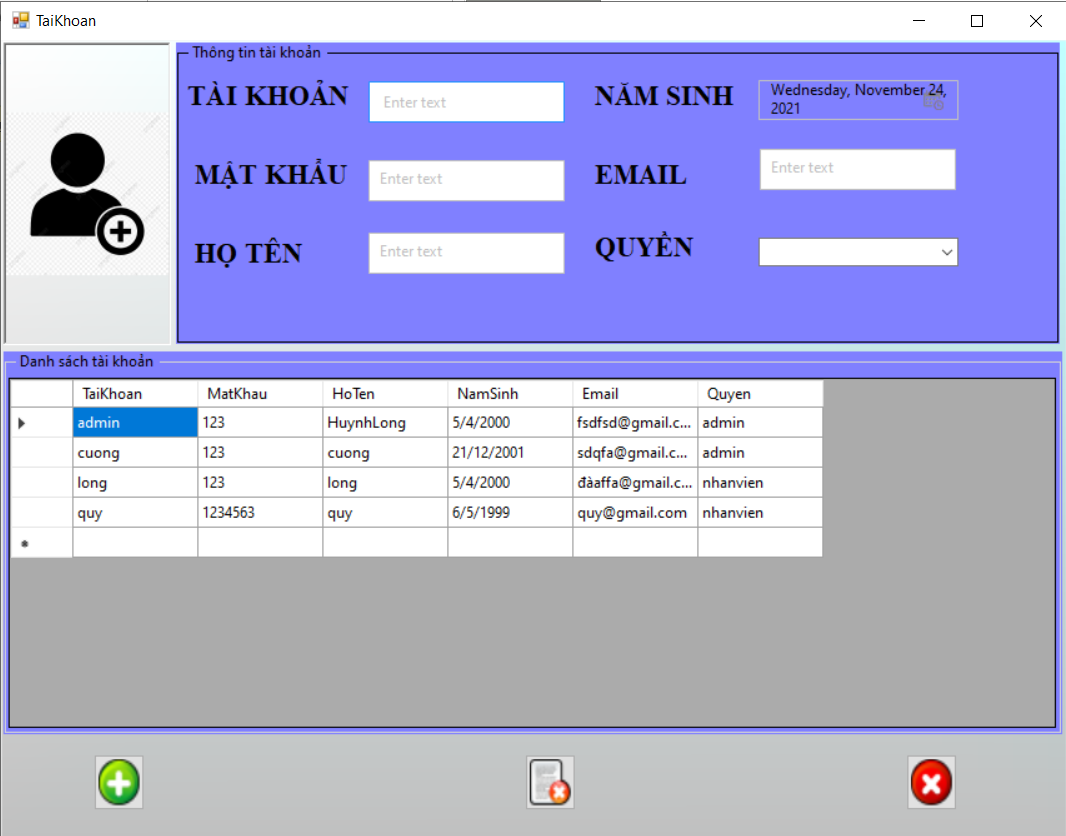
* Sau khi đăng nhập thành công thì ta có thể lựa chọn các chương trình cho sản phẩm
* Nếu đăng nhập bằng tài khoản admin thì phần cá nhân sử dụng được, ngược lại sẽ bị ẩn





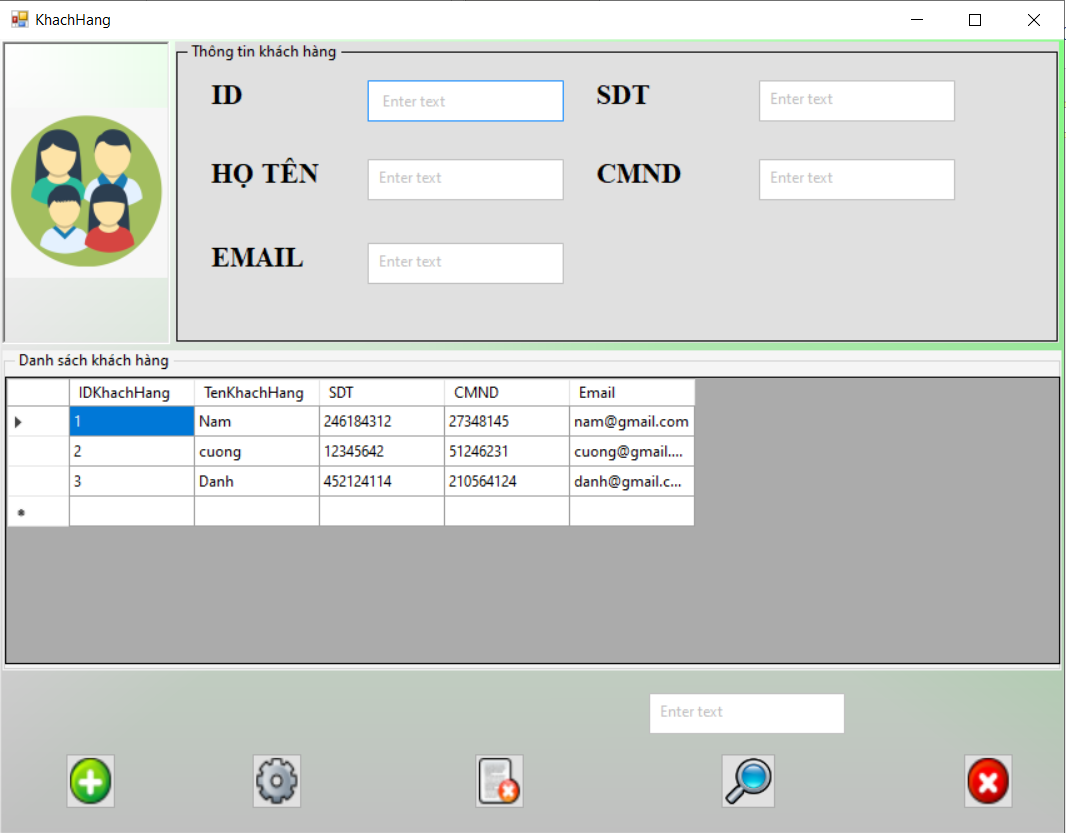
1. Trang tài khoản

* Tại trang tài khoản là nơi admin sẽ có quyền được tạo, xoá tài khoản theo yêu cầu



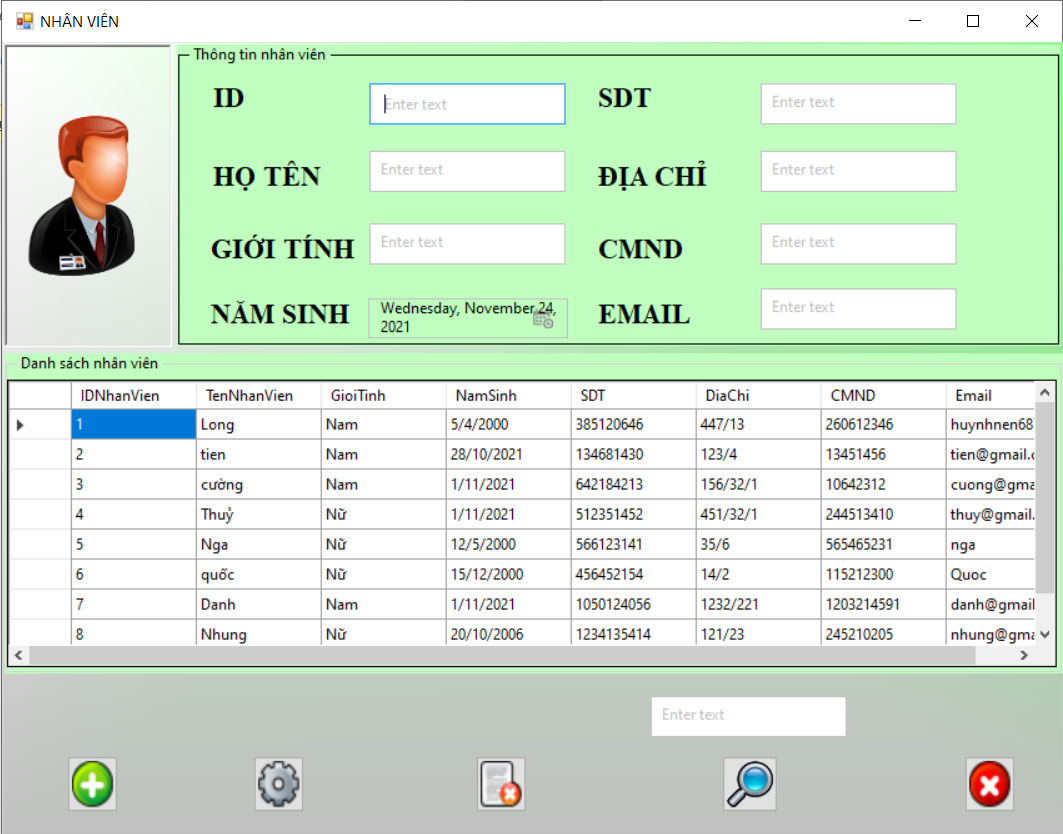
1. Trang khách hàng

* Trang này dùng để quản lý khách hàng với mục đích để xem khách hàng nào là khách hàng quý của cửa hàng thông qua việc tới quán ăn nhiều sẽ được lưu lại



1. Trang nhân viên

* Trang này dùng với mục đích để kiểm soát nhân viên cnf đi làm hay nghỉ



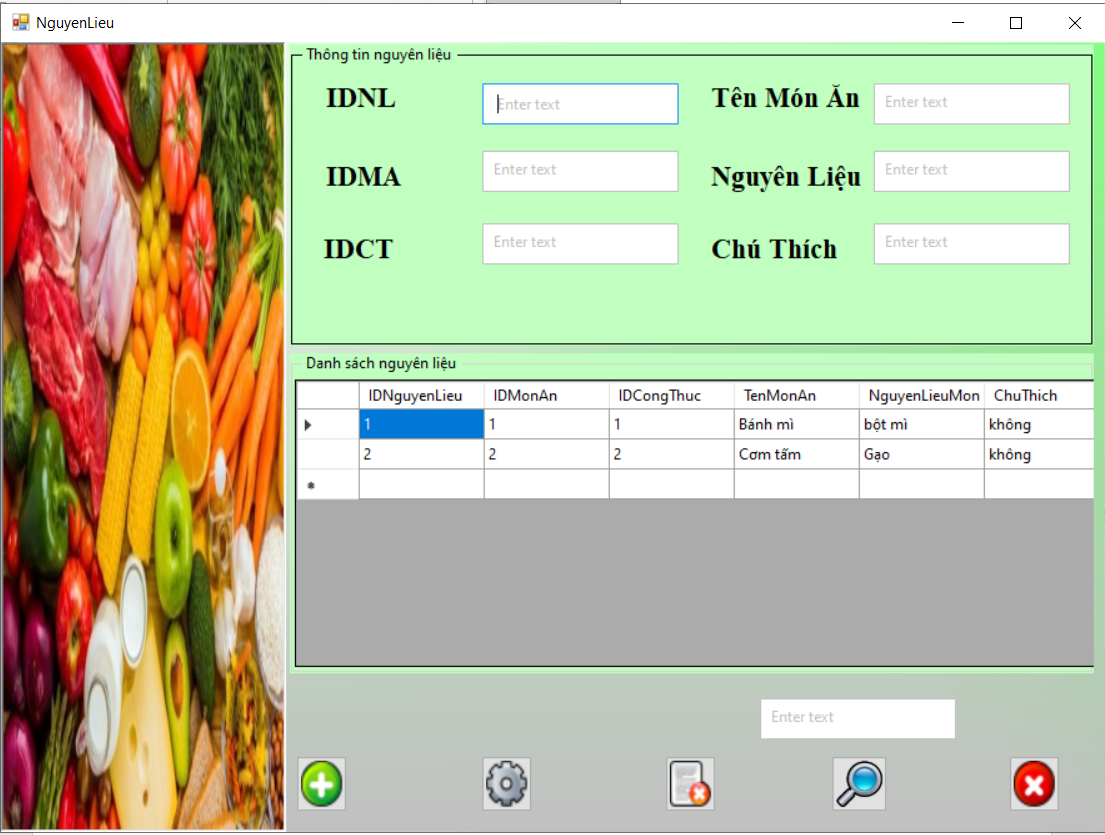
1. Trang món ăn

* Dùng để tra cứu món ăn, nguồn gốc và giá cả các món ăn của quán



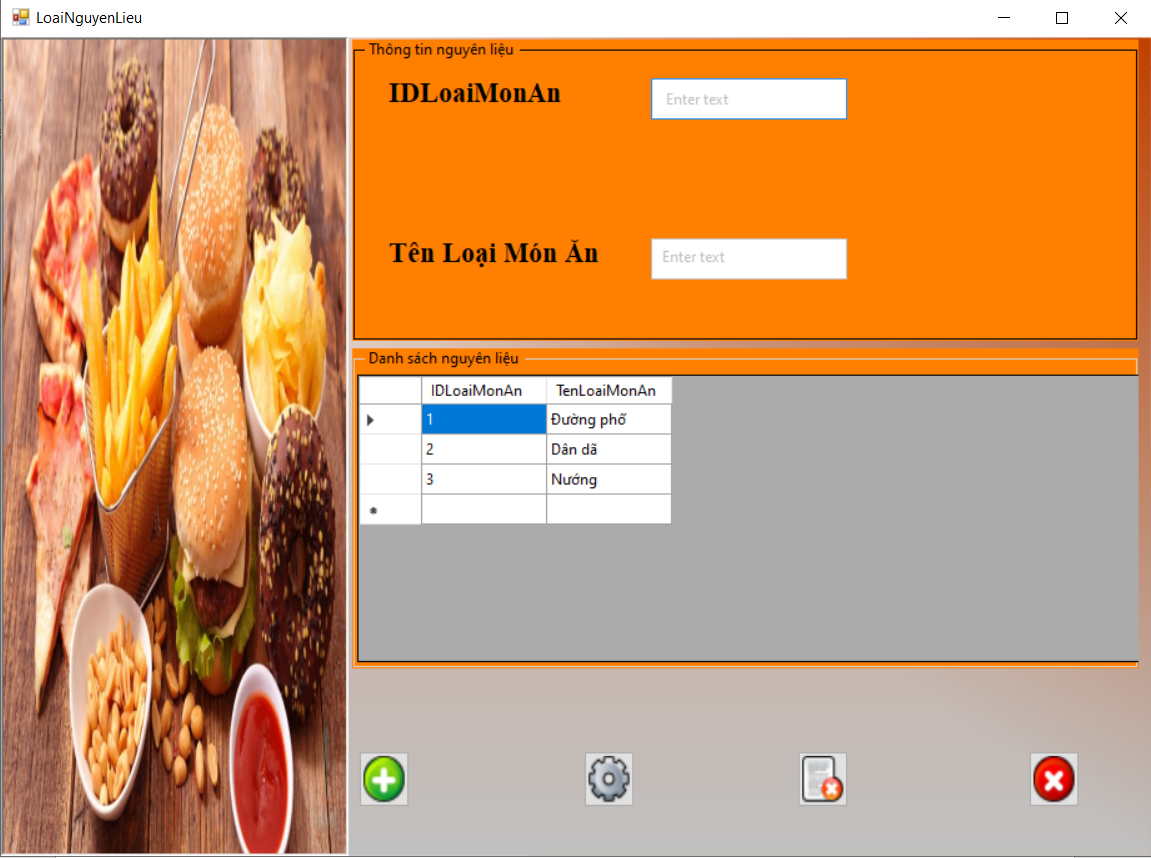
1. Trang nguyên liệu

* Trang này cho biết được nguyên liệu của món ăn



1. Trang loại món ăn

* Món ăn thuộc loại ẩm thực này



1. Trang công thức

* Tại đây nhân viên quán hay đầu bếp có thể tự mình thêm công thức món ăn để cùng nhau trao đổi món, hoặc là nơi để các đầu bếp mới cơ thể học nấu những món ăn của quán



1. Trang đặt món

* Khi khách hàng vào thì nhân viên sẽ chọn mục đặt món để có thể đặt món cho khách hàng
* Khi khách hàng ra về thì thanh toán có phần nút in sẽ tự động in ra hoá đơn



1. Trang đánh giá

* Phần này để xem được nhân viên nào làm việc hiệu quả, hay xem cụ thể công thức nấu, nguồn gốc món ăn, hay là số tiền của khách hàng



1. Thống kê

* Nơi xem những thống kê món ăn theo ngày và có trực quan hoá dữ liệu để xem thử món ăn nào bán chạy



1. Địa điểm

* Nơi tra cứu địa điểm của các món ăn đó (bánh mì sẽ có ở đâu bán)

