



CÔNG NGHỆ PHẦN MỀM

Desktop Application for Billiard Club Management

NHÓM: 1 | 02/12/2024

THÀNH VIÊN

2001223947

Hồ Đức Hoàng Quân

2001224551

Phạm Minh Tuấn

2001223660

Hà Huy Phong

2001224717

Lê Văn Thành

Mục lục

01

Phần mở đầu

- Lý do chọn đề tài
- Mục tiêu nghiên cứu
- Đối tượng nghiên cứu
- Phạm vi nghiên cứu
- Ý nghĩa khoa học
- Ý nghĩa thực tiễn

02

Nội dung

- Tổng quan
- Phân tích hệ thống

03

Phân tích thiết kế

- CSDL
- Biểu đồ mô hình hóa
- Giao diện hệ thống

MỞ ĐẦU



Lý do chọn đề tài

Quản lý quán bida nhằm tối ưu hóa kinh doanh. Hệ thống hỗ trợ kiểm soát hiệu quả, cải thiện trải nghiệm khách hàng qua khuyến mãi và tích điểm.



Đối tượng & phạm vi nghiên cứu

Đối tượng: Quy trình và yêu cầu quản lý quán bida.

Phạm vi: Phát triển hệ thống quản lý đặt bàn, khách hàng, nhân viên, và doanh thu.



Mục tiêu nghiên cứu

- Thiết kế hệ thống quản lý quán bida với các chức năng: quản lý bàn, nhân viên, khách hàng, thanh toán, tồn kho, báo cáo.
- Tối ưu quy trình, nâng cao hiệu suất, đảm bảo bảo mật dữ liệu.



Ý nghĩa khoa học & thực tiễn

- Nâng cao kiến thức về phân tích, thiết kế hệ thống quản lý, góp phần phát triển giải pháp công nghệ cho các dịch vụ tương tự.
- Tối ưu quy trình, tiết kiệm thời gian, nhân lực, giảm sai sót, nâng cao hiệu quả, hỗ trợ mở rộng kinh doanh và cải thiện phục vụ.

TỔNG QUAN

01.

QUY MÔ VÀ LOẠI HÌNH KINH DOANH

Quán bida cung cấp dịch vụ cho thuê bàn chơi, bán đồ ăn/uống, với quy mô từ vài đến hàng chục bàn. Hệ thống quản lý cần theo dõi tình trạng bàn và dịch vụ khách hàng.

02.

QUẢN LÝ NHÂN SỰ VÀ KHÁCH HÀNG

Bao gồm lưu trữ thông tin nhân viên, phân ca làm việc và quản lý thông tin khách hàng.

03.

QUẢN LÝ TÀI CHÍNH VÀ HOẠT ĐỘNG

Tự động cập nhật doanh thu từ các dịch vụ, theo dõi chi phí vận hành, và tạo báo cáo chi tiết về doanh thu, chi phí, hỗ trợ phân tích tài chính.

04.

YÊU CẦU BẢO MẬT VÀ PHÂN QUYỀN

Hệ thống đảm bảo bảo mật thông tin nhạy cảm và phân quyền vai trò, giới hạn truy cập theo quyền hạn.

PHÂN TÍCH HỆ THỐNG

01

QUY TRÌNH NGHIỆP VỤ

- Quản lý đặt bàn
- Quản lý đặt món
- Quản lý thanh toán

02

CHỨC NĂNG HỆ THỐNG

- Quản lý bàn
- Quản lý khách hàng và tích điểm
- Quản lý dịch vụ kèm theo
- Quản lý nhân viên
- Quản lý kho hàng
- Báo cáo và phân tích doanh thu

QUY TRÌNH NGHIỆP VỤ

QUẢN LÝ THANH TOÁN

Tính tổng tiền dựa trên thời gian chơi, dịch vụ và khuyến mãi.



QUẢN LÝ ĐẶT BÀN

Hiển thị trạng thái bàn (trống, đang sử dụng) và hỗ trợ đổi loại bàn (bida phẳng, bida lỗ).

QUẢN LÝ ĐẶT MÓN

Hệ thống tính số lượng món đã thêm vào bàn và kiểm tra kho để đảm bảo đủ cung ứng.



CHỨC NĂNG HỆ THỐNG

01

QUẢN LÝ DỊCH VỤ

Xử lý yêu cầu đồ ăn, thức uống và tự động cập nhật kho hàng.

03

QUẢN LÝ KHÁCH HÀNG VÀ TÍCH ĐIỂM

Lưu trữ thông tin khách hàng, tích điểm và áp dụng khuyến mãi tự động.

05

QUẢN LÝ KHO HÀNG

Admin có thể quản lý kho: xem tồn kho, thêm, xóa, sửa sản phẩm và nhập hàng khi cần.

02

QUẢN LÝ BÀN

Hiển thị trạng thái bàn, hỗ trợ đặt trước, và tự động cập nhật thời gian sử dụng.

04

QUẢN LÝ NHÂN VIÊN

Quản lý thông tin nhân viên và phân ca lịch làm việc.

CHỨC NĂNG HỆ THỐNG

02

QUẢN LÝ BÀN

Hiển thị trạng thái bàn, hỗ trợ đặt trước, và tự động cập nhật thời gian sử dụng.

04

QUẢN LÝ NHÂN VIÊN

Quản lý thông tin nhân viên và phân ca lịch làm việc.

06

BÁO CÁO VÀ PHÂN TÍCH DOANH THU

Tạo báo cáo chi tiết về doanh thu, chi phí và lợi nhuận theo giai đoạn và loại dịch vụ.

03

QUẢN LÝ KHÁCH HÀNG VÀ TÍCH ĐIỂM

Lưu trữ thông tin khách hàng, tích điểm và áp dụng khuyến mãi tự động.

05

QUẢN LÝ KHO HÀNG

Admin có thể quản lý kho: xem tồn kho, thêm, xóa, sửa sản phẩm và nhập hàng khi cần.



KIẾN TRÚC HỆ THỐNG VÀ CÔNG NGHỆ ĐƯỢC SỬ DỤNG

KIẾN TRÚC HỆ THỐNG

BUSINESS LOGIC LAYER



PRESENTATION LAYER



DATA ACCESS LAYER

PRESENTATION LAYER

01

THÀNH PHẦN GIAO DIỆN

Sử dụng Windows Forms để hiển thị trực quan.

02

CHỨC NĂNG CHÍNH

- Quản lý danh sách bàn bida và trạng thái.
- Hiển thị thông tin khách hàng, nhân viên, kho hàng, và báo cáo.
- Tương tác qua nút bấm, textbox, combo box, và grid view.

BUSINESS LOGIC LAYER

01

CHỨC NĂNG XỬ LÝ NGHIỆP VỤ CHÍNH

Tính toán điểm thưởng, xử lý khuyến mãi, quản lý trạng thái bàn, và tính toán hóa đơn.

02

CÁC MODULE QUẢN LÝ

- Khách hàng: Thêm, sửa, xóa, xử lý điểm thưởng.
- Nhân viên: Lập lịch làm việc, phân quyền truy cập.
- Kho hàng: Theo dõi tồn kho, cập nhật số lượng.
- Doanh thu: Tổng hợp dữ liệu, tạo báo cáo chi tiết.

DATA ACCESS LAYER

01

VAI TRÒ

Tương tác trực tiếp với cơ sở dữ liệu SQL Server.

02

CHỨC NĂNG CHÍNH

- Kết nối và thực hiện truy vấn SQL.
- Trả dữ liệu cho lớp Business Logic xử lý.
- Sử dụng LINQ to SQL để thao tác với các bảng dữ liệu.

CÔNG NGHỆ SỬ DỤNG

01.

NGÔN NGỮ LẬP TRÌNH

C# (.NET Framework) được chọn vì tích hợp tốt với Windows Forms, LINQ, và SQL Server.

02.

GIAO DIỆN NGƯỜI DÙNG

Windows Forms được chọn nhờ khả năng kéo thả trực quan và thư viện phong phú.

03.

CƠ SỞ DỮ LIỆU

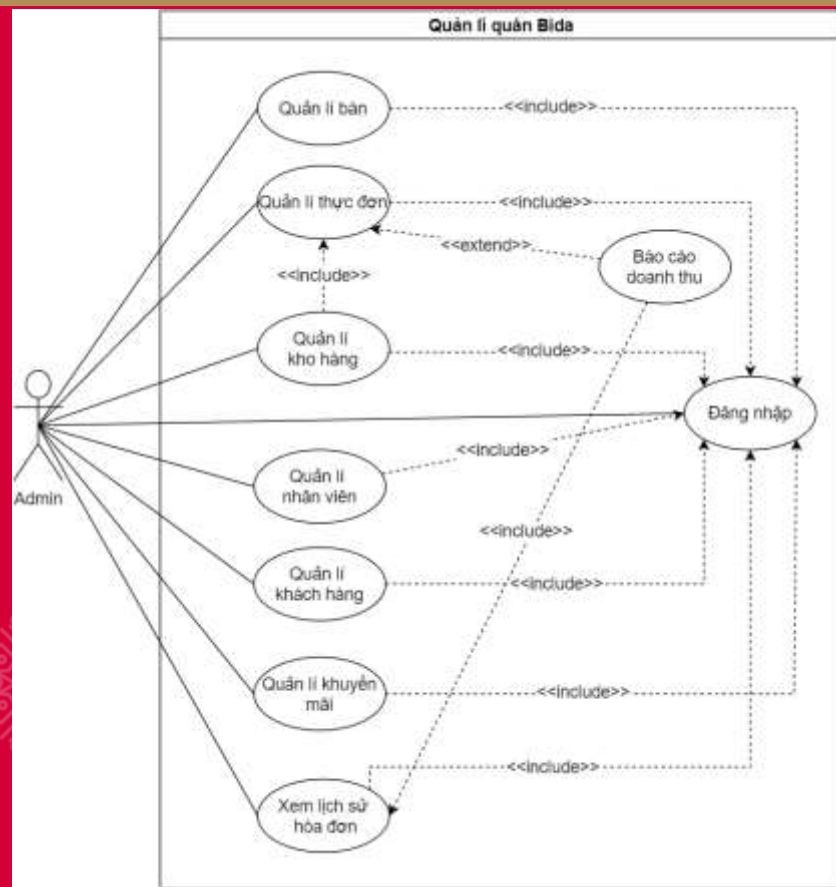
SQL Server được sử dụng để quản lý dữ liệu khách hàng, nhân viên, bàn bida, và kho hàng, với khả năng xử lý lớn và tích hợp dễ dàng với .NET.

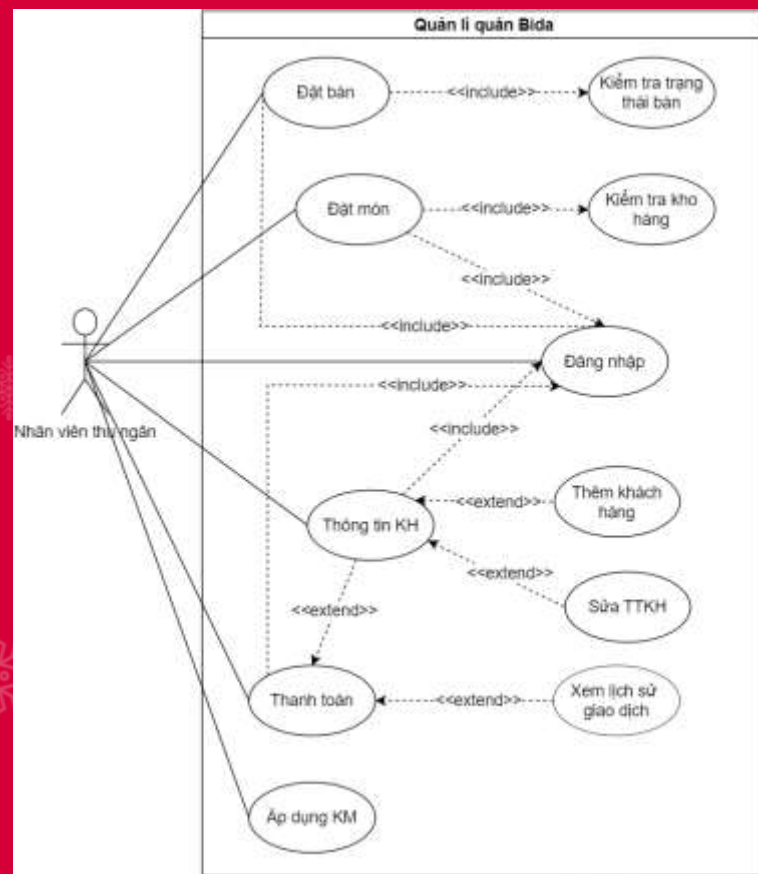
04.

TRUY CẬP CƠ SỞ DỮ LIỆU

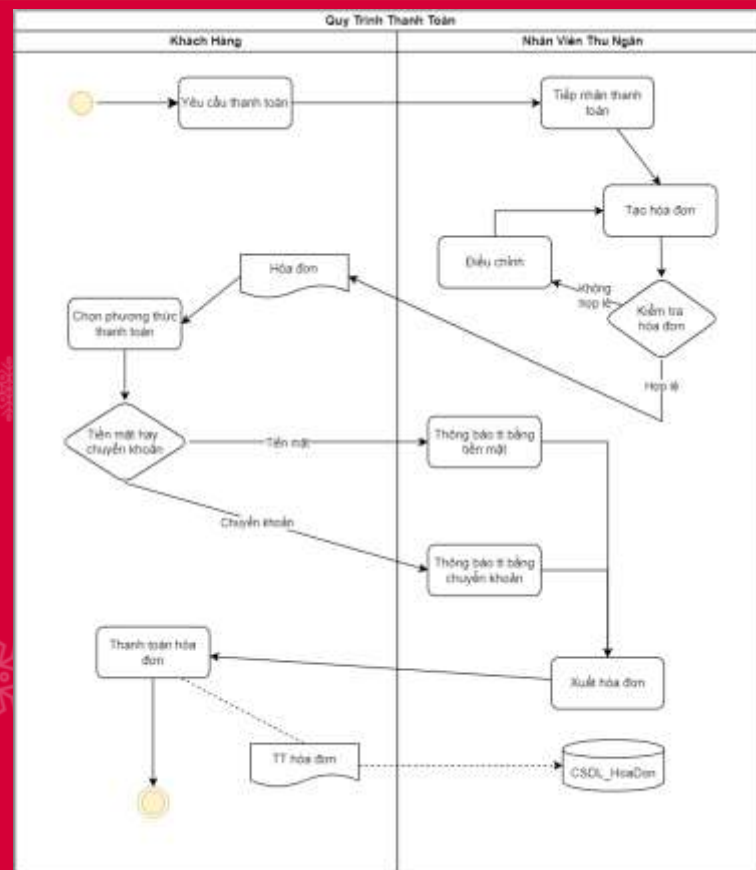
LINQ to SQL được dùng để viết và thao tác trực tiếp các truy vấn SQL trong C#.

BIỂU ĐỒ MÔ HÌNH HÓA HỆ THỐNG

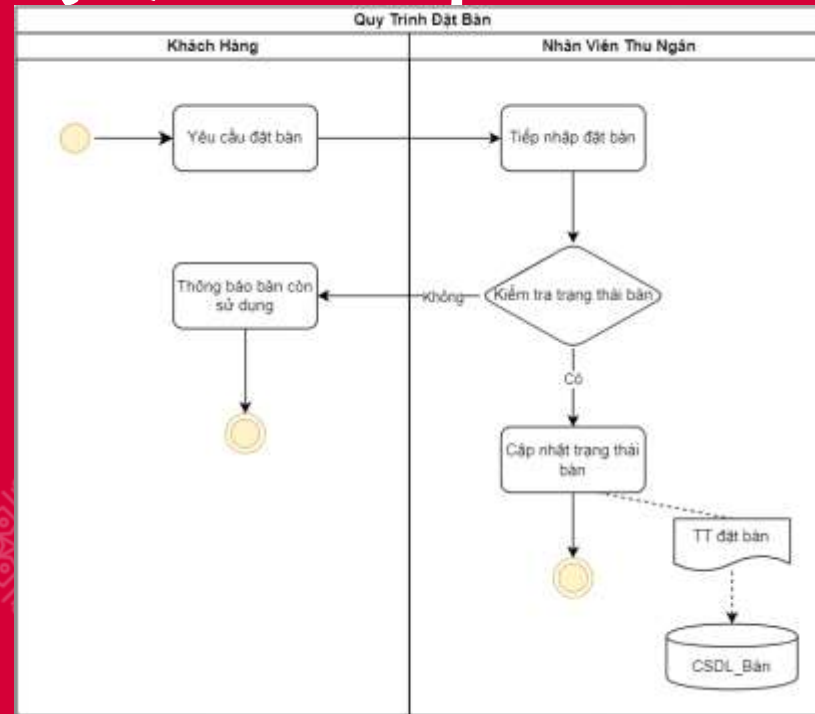




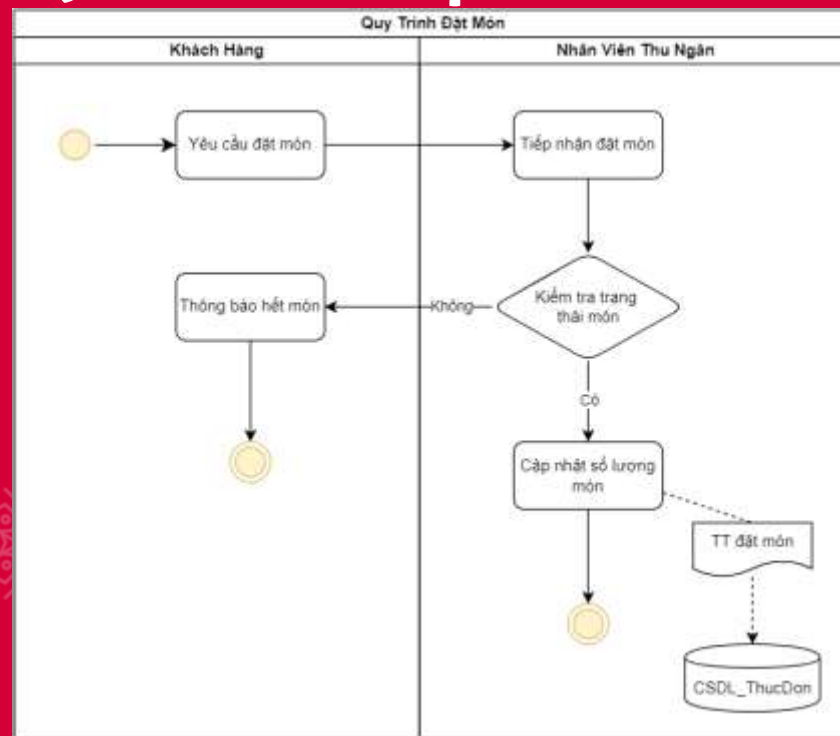
```
graph TD
    subgraph "Quy Trình Quản Lý Nhân Viên"
        direction TB
        subgraph "Nhân Viên"
            Start(( )) --> HN1[Hồ sơ nhân viên]
            HN1 --> Admin
            Admin --> CN[Cập nhật hồ sơ]
            CN --> HN1
            HN2[Hồ sơ nhân viên] --> Admin
            Admin --> TK[Tài khoản nhân viên]
            TK --> End(( ))
        end
        subgraph "Admin"
            HN1 --> TN[Tiếp nhận hồ sơ]
            TN --> KT{Kiểm tra hồ sơ}
            KT -- "Không hợp lệ" --> HN2
            KT -- "Hợp lệ" --> NTV[Nhập thông tin NV]
            NTV --> PQT[Phân quyền truy cập]
            PQT --> End2(( ))
            NTV -.-> HN3[Hồ sơ nhân viên]
            HN3 -.-> CSDL[(CSDL_NV)]
        end
    end
```



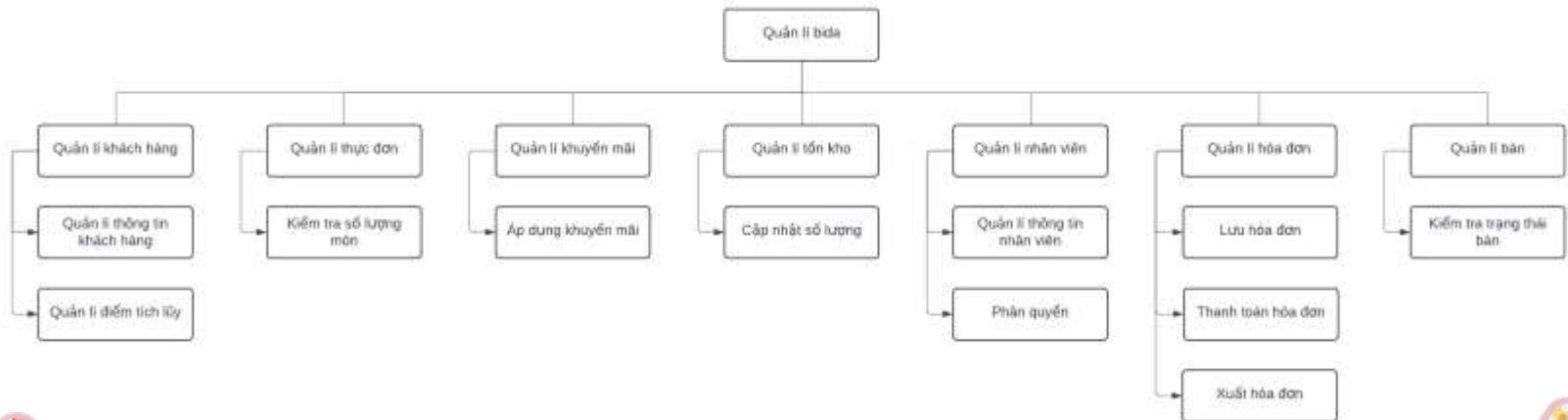
QUY TRÌNH ĐẶT BÀN BPM



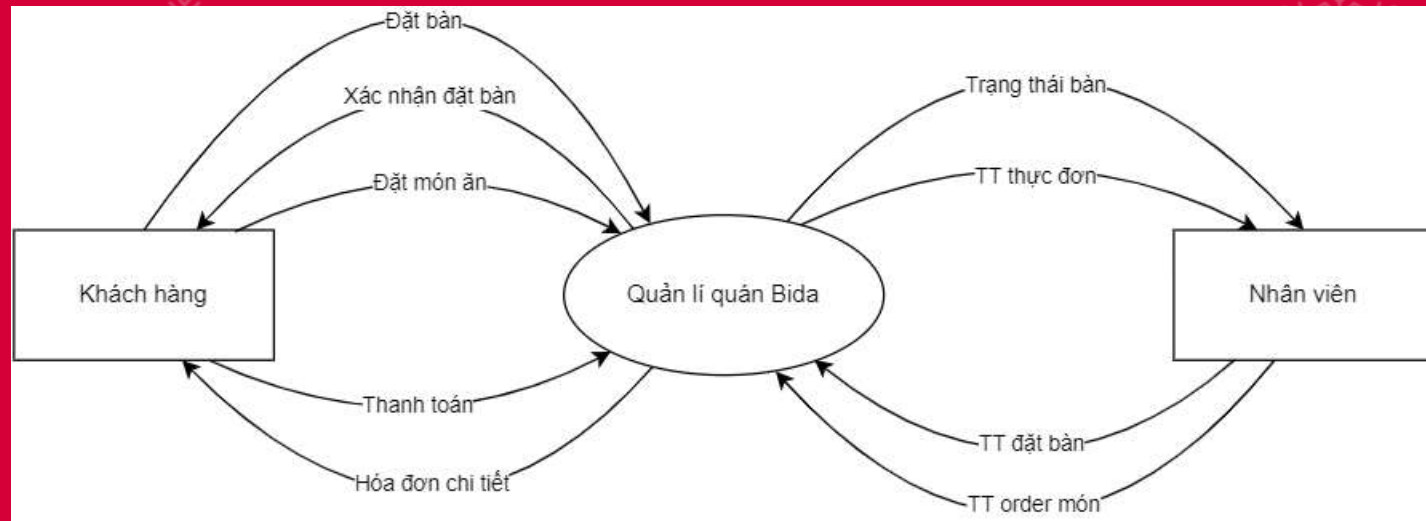
QUY TRÌNH ĐẶT MÓN BPM



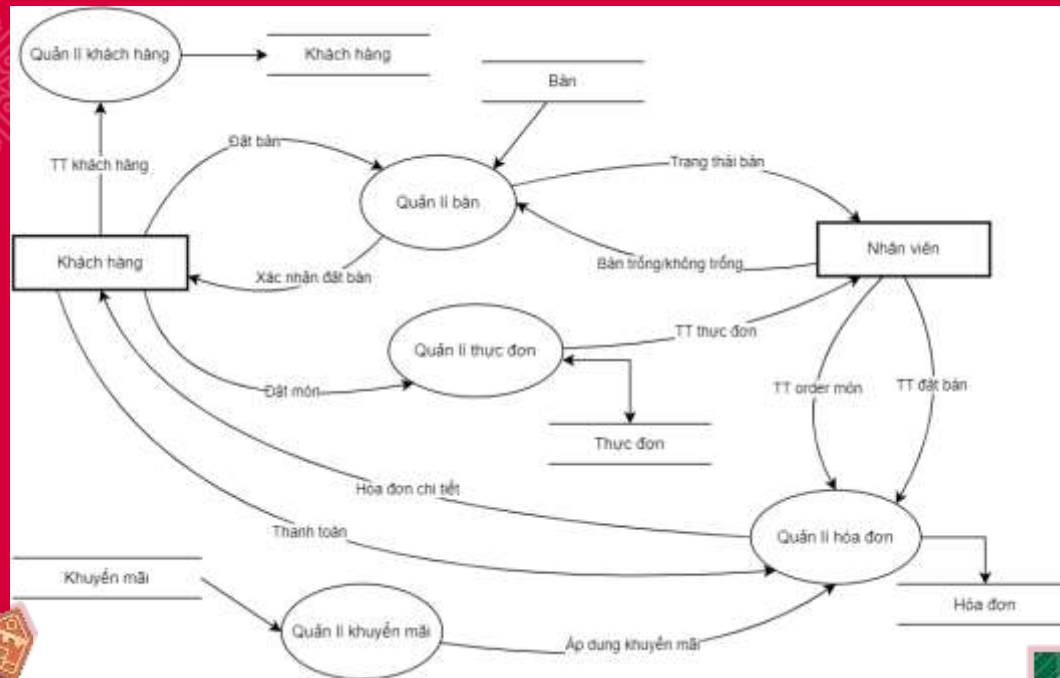
SƠ ĐỒ BFD

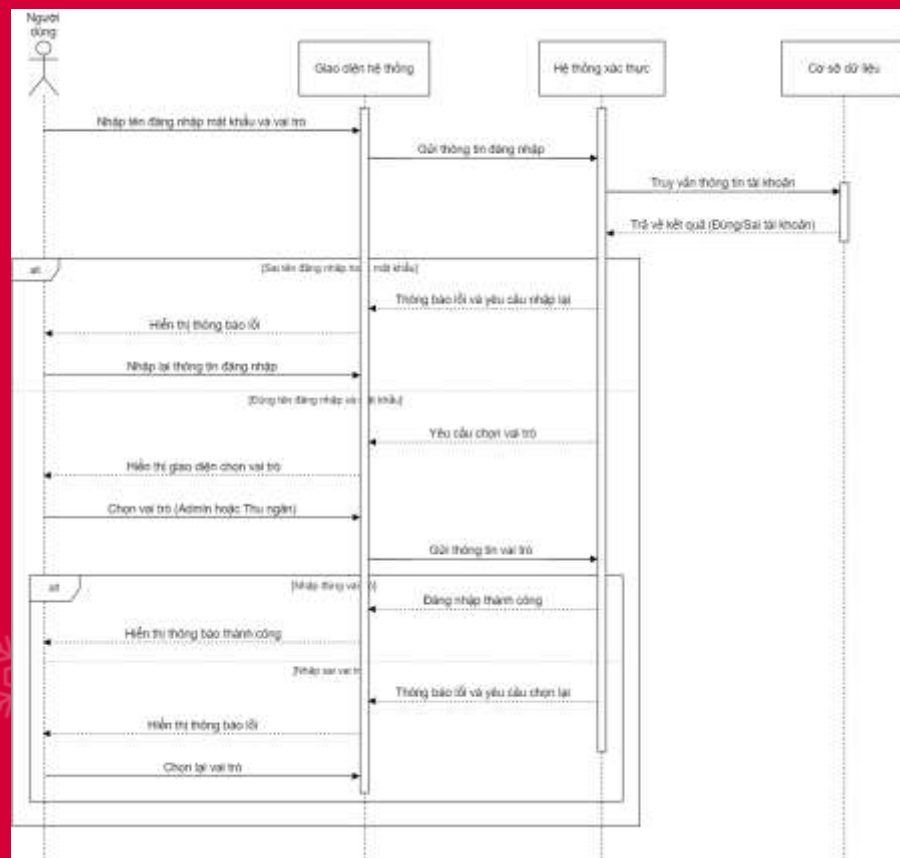


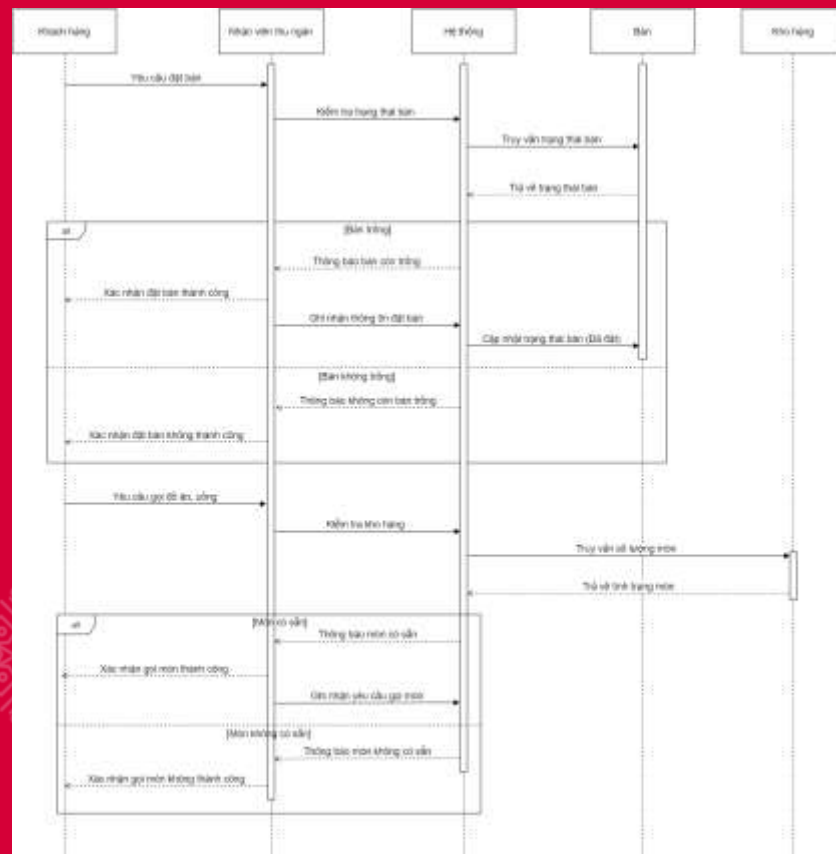
DFD MỨC 0

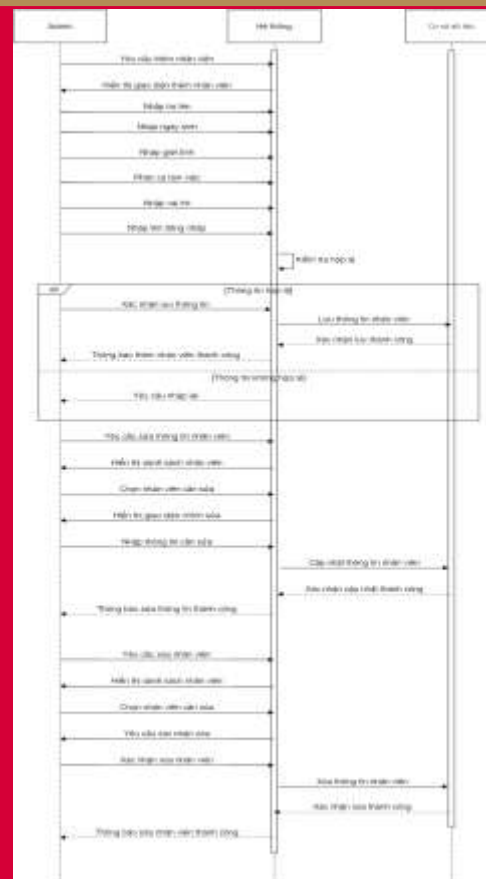


DFD MỨC 1

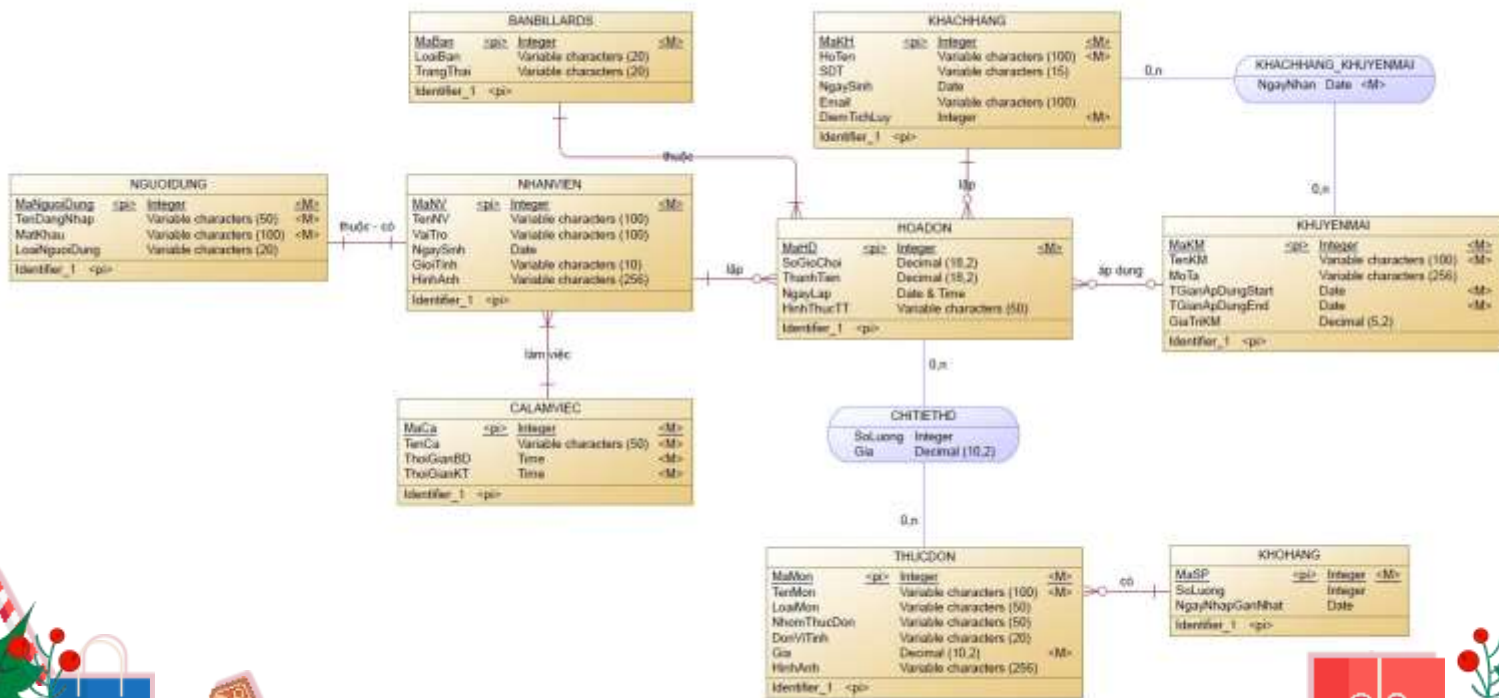


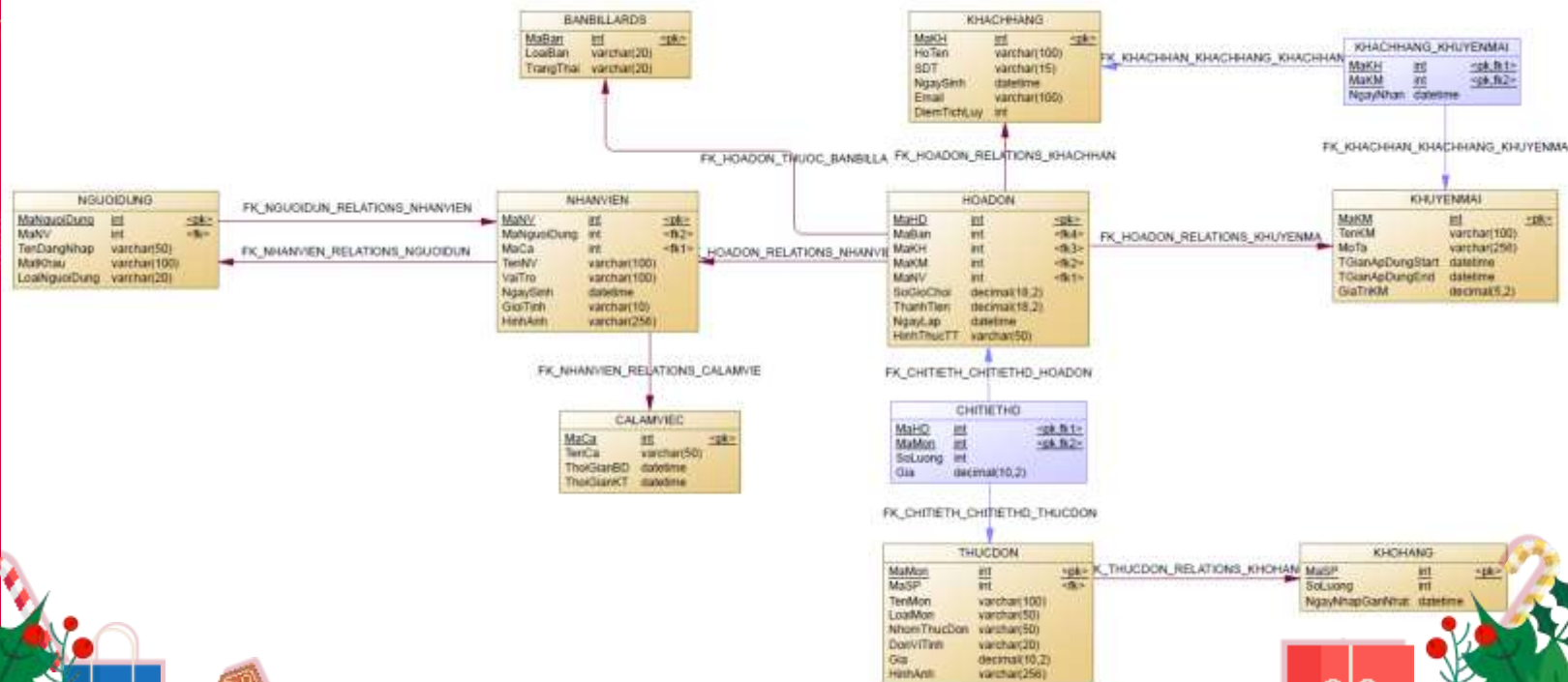














THỰC THI CSDL

BƯỚC 1

Loại Backup	Thời gian thực hiện
Full Backup	23:00 hàng ngày
Differential Backup	02:00, 08:00, 14:00, 20:00
Transaction Log Backup	Mỗi 15 phút

TẠO JOB VÀ LÊN LỊCH BACK-UP

BƯỚC 2

```
-- 3. CẤP QUYỀN CHO ADMIN, THU NGÂN, KỸ THUẬT
EXEC sp_addrolemember 'db_owner', 'role_admin';
EXEC sp_addrolemember 'role_admin', 'thanh';

GRANT SELECT, INSERT, UPDATE ON BanBilliards TO role_thungan;
GRANT SELECT ON ThucDon TO role_thungan;
GRANT SELECT, INSERT, UPDATE ON KhachHang TO role_thungan;
GRANT SELECT, UPDATE ON HoaDon TO role_thungan;
EXEC sp_addrolemember 'role_thungan', 'phong';
EXEC sp_addrolemember 'role_thungan', 'tuan';

GRANT SELECT, UPDATE ON BanBilliards TO role_kythuat;
GRANT SELECT, UPDATE ON ThucDon TO role_kythuat;
GRANT SELECT ON KhachHang TO role_kythuat;
GRANT SELECT ON HoaDon TO role_kythuat;
EXEC sp_addrolemember 'role_kythuat', 'quan';
```

PHÂN QUYỀN NGƯỜI DÙNG

BƯỚC 3

**TAO BẢNG VÀ
XÂY DỰNG CÁC THỦ TỤC**



XÂY DỰNG MÔ HÌNH QUAN HỆ



ThucDon (**MaMon**, TenMon, LoaiMon, NhomThucDon, DonViTinh, Gia, HinhAnh)

BanBillards (**MaBan**, LoaiBan, TrangThai)

NhanVien (**MaNV**, TenNV, VaiTro, NgaySinh, GioiTinh, MaCa, HinhAnh)

CaLamViec (**MaCa**, TenCa, ThoiGianBatDau, ThoiGianKetThuc)

NguaiDung (**MaNguoiDung**, MaNV, TenDangNhap, MatKhau, LoaiNguoiDung)

KhachHang (**MaKH**, HoTen, SDT, NgaySinh, Email, DiemTichLuy)



HoaDon (**SoHoaDon**, MaKH, MaBan, MaNV, SoGioChoi, ThanhTien, NgayLap, HinhThucThanhToan, MaKM)

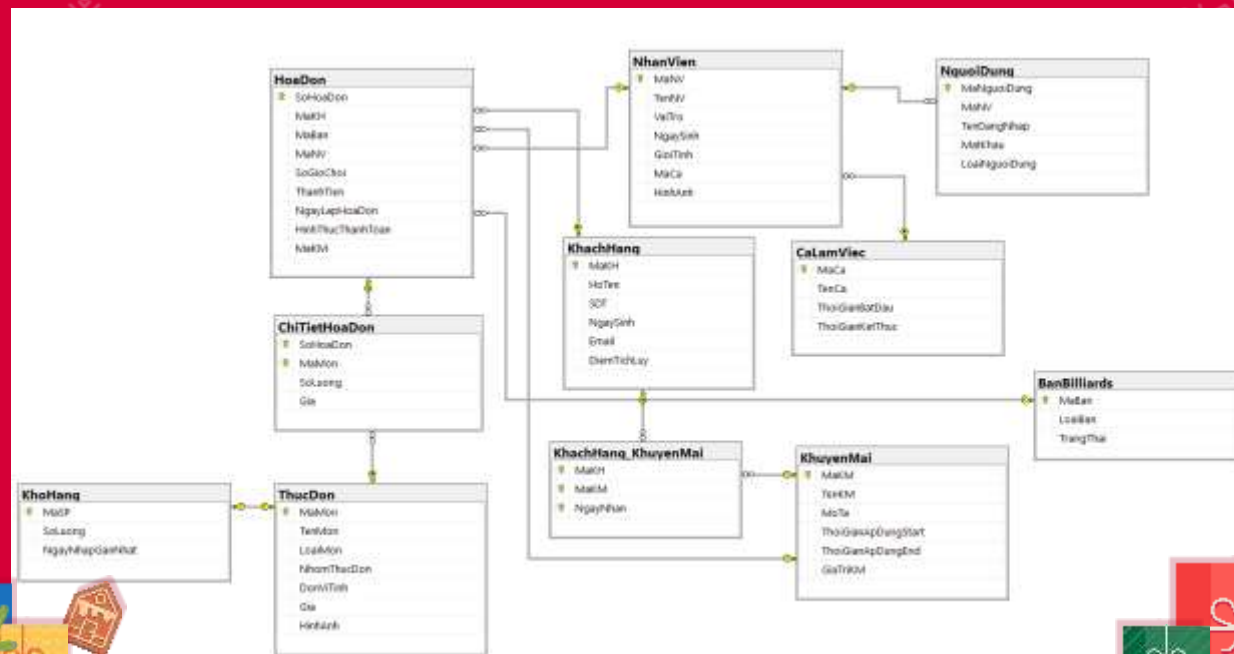
ChiTietHoaDon (**SoHoaDon**, **MaMon**, SoLuong, Gia)

KhoHang (**MaSP**, SoLuong, NgayNhapGanNhat)

KhuyenMai (**MaKM**, TenKM, MoTa, ThoiGianApDungStart, ThoiGianApDungEnd, GiaTriKM)

KhachHang_KhuyenMai (**MaKH**, **MaKM**, **NgayNhan**)

SƠ ĐỒ DIAGRAM



XÂY DỰNG THỦ TỤC KHO HÀNG

```
CREATE PROCEDURE sp_GetKhoList
AS
BEGIN
    SET NOCOUNT ON;

    SELECT
        KhoHang.MaSP,
        ThucDon.MaMon,
        ThucDon.TenMon,
        KhoHang.SoLuong,
        ThucDon.DonViTinh,
        ThucDon.Gia,
        KhoHang.NgayNhapGanNhat
    FROM
        KhoHang
    JOIN
        ThucDon ON KhoHang.MaSP = ThucDon.MaMon;
END
```

THỦ TỤC LẤY DANH SÁCH THỰC ĐƠN TRONG KHO

XÂY DỰNG THỦ TỤC KHO HÀNG

```
CREATE PROCEDURE sp_UpdateKhoHang
    @MaMon INT,
    @SoLuong INT,
    @NgayNhapGanNhat DATETIME
AS
BEGIN
    SET NOCOUNT ON;

    UPDATE KhoHang
    SET SoLuong = @SoLuong,
        NgayNhapGanNhat = @NgayNhapGanNhat
    WHERE MaSP = @MaMon;
END
```

THỦ TỤC CẬP NHẬT KHO HÀNG

XÂY DỰNG THỦ TỤC KHO HÀNG

```
CREATE PROCEDURE sp_DeleteKhoHangByMaMon
    @MaMon INT
AS
BEGIN
    SET NOCOUNT ON;

    DELETE FROM KhoHang WHERE MaSP = @MaMon;
END
```

THỦ TỤC XÓA MÓN TRONG KHO HÀNG

XÂY DỰNG THỦ TỤC BÀN BIDA

```
CREATE PROCEDURE sp_GetAllBanBilliards  
AS  
BEGIN  
    SET NOCOUNT ON; -- Tắt thông báo về số dòng bị ảnh hưởng  
    SELECT * FROM BanBilliards;  
END
```

go

THỦ TỤC TẮT CẢ CÁC BÀN

XÂY DỰNG THỦ TỤC BÀN BIDA

```
CREATE PROCEDURE sp_ThemBan
    @LoaiBan NVARCHAR(50)
AS
BEGIN
    SET NOCOUNT ON; -- Tắt thông báo về số dòng bị ảnh hưởng
    INSERT INTO BanBilliards (LoaiBan, TrangThai)
    VALUES (@LoaiBan, 'Trống');
END
go
```

THỦ TỤC THÊM BÀN MỚI

XÂY DỰNG THỦ TỤC BÀN BIDA

```
CREATE PROCEDURE sp_XoaBan
    @MaBan INT
AS
BEGIN
    DELETE FROM BanBilliards WHERE MaBan = @MaBan;
END
```

THỦ TỤC XÓA BÀN



XÂY DỰNG THỦ TỤC THỰC ĐƠN

```
CREATE PROCEDURE sp_GetNhomThucDon
AS
BEGIN
    SET NOCOUNT ON; -- Tắt thông báo về số hàng bị ảnh hưởng

    -- Lấy danh sách các nhóm thực đơn khác nhau
    SELECT DISTINCT NhomThucDon FROM ThucDon;
END
```



THỦ TỤC LẤY DANH SÁCH CÁC NHÓM THỰC ĐƠN

2



XÂY DỰNG THỦ TỤC THỰC ĐƠN

```
CREATE PROCEDURE sp_GetAllThucDon
AS
BEGIN
    SET NOCOUNT ON; -- Tắt thông báo ảnh hưởng của các câu lệnh
    SELECT * FROM ThucDon;
END
go
```

THỦ TỤC LẤY DANH SÁCH THỰC ĐƠN

XÂY DỰNG THỦ TỤC THỰC ĐƠN

```
CREATE PROCEDURE sp_UpdateThucDon
@MaMon INT,
@TenMon NVARCHAR(100),
@LoaiMon NVARCHAR(50),
@NhomThucDon NVARCHAR(50),
@DonViTinh NVARCHAR(20),
@Gia DECIMAL(10, 2),
@HinhAnh NVARCHAR(255)
AS
BEGIN
    SET NOCOUNT ON;

    UPDATE ThucDon
    SET TenMon = @TenMon,
        LoaiMon = @LoaiMon,
        NhomThucDon = @NhomThucDon,
        DonViTinh = @DonViTinh,
        Gia = @Gia,
        HinhAnh = @HinhAnh
    WHERE MaMon = @MaMon;
END
```

THỦ TỤC CẬP NHẬT THỰC ĐƠN

XÂY DỰNG THỦ TỤC THỰC ĐƠN

```
CREATE PROCEDURE sp_DeleteThucDon  
    @MaMon INT  
AS  
BEGIN  
    SET NOCOUNT ON; -- Tắt thông báo về số hàng bị ảnh hưởng  
    DELETE FROM ThucDon WHERE MaMon = @MaMon;  
END  
go
```

THỦ TỤC XÓA THỰC ĐƠN

XÂY DỰNG HÀM

HÀM LẤY MÃ NHÂN VIÊN

```
CREATE FUNCTION dbo.fn_LayMaNV (@TenDangNhap NVARCHAR(100))
RETURNS INT
AS
BEGIN
    DECLARE @MaNV INT;

    -- Lấy MaNV từ bảng NguoiDung dựa trên TenDangNhap
    SELECT @MaNV = MaNV
    FROM NguoiDung
    WHERE TenDangNhap = @TenDangNhap;

    -- Trả về kết quả (Nếu không tìm thấy, trả về NULL)
    RETURN ISNULL(@MaNV, 0);
END;
```

XÂY DỰNG HÀM

HÀM LẤY DỮ LIỆU TỪ BẢNG THỰC ĐƠN

```
CREATE FUNCTION dbo.fn_GetMenuItems()  
RETURNS @MenuTable TABLE  
(  
    MaMon INT,  
    TenMon NVARCHAR(100),  
    LoaiMon NVARCHAR(50),  
    NhomThucDon NVARCHAR(50),  
    Gia DECIMAL(10, 2),  
    HinhAnh NVARCHAR(255)  
)  
AS  
BEGIN  
    -- Thêm dữ liệu vào bảng tạm @MenuTable từ bảng ThucDon  
    INSERT INTO @MenuTable  
    SELECT  
        MaMon,  
        TenMon,  
        LoaiMon,  
        NhomThucDon,  
        Gia,  
        HinhAnh  
    FROM ThucDon;  
  
    RETURN;  
END;
```

XÂY DỰNG HÀM

HÀM LẤY LẤY DỮ LIỆU TỪ BẢNG KHÁCH HÀNG

```
CREATE FUNCTION dbo.fn_GetKhachHangList()
RETURNS @KhachHangTable TABLE
(
    MaKH INT,
    HoTen NVARCHAR(100),
    SDT NVARCHAR(20),
    NgaySinh DATE,
    Email NVARCHAR(100),
    DiemTichLuy INT
)
AS
BEGIN
    -- Thêm dữ liệu vào bảng tạm @KhachHangTable từ bảng KháchHang
    INSERT INTO @KhachHangTable
    SELECT
        MaKH,
        HoTen,
        SDT,
        NgaySinh,
        Email,
        DiemTichLuy
    FROM KháchHang;

    RETURN;
END;
```


XÂY DỰNG HÀM

HÀM TÍNH THỜI GIAN SỬ DỤNG BÀN

```
CREATE FUNCTION dbo.fn_GetHoaDonList()  
RETURNS @HoaDonTable TABLE  
(  
    SoHoaDon INT,  
    MaKH INT NULL,  
    MaBan INT,  
    MaNV INT,  
    SoGioChoi DECIMAL(10, 2),  
    ThanhTien DECIMAL(18, 2) NULL,  
    NgayLapHoaDon DATETIME,  
    HinhThucThanhToan NVARCHAR(50),  
    MaKM INT NULL  
)  
AS  
BEGIN  
    -- Thêm dữ liệu vào bảng tạm @HoaDonTable từ bảng HoaDon  
    INSERT INTO @HoaDonTable  
    SELECT  
        SoHoaDon,  
        MaKH,  
        MaBan,  
        MaNV,  
        SoGioChoi,  
        ThanhTien,  
        NgayLapHoaDon,  
        HinhThucThanhToan,  
        MaKM  
    FROM HoaDon;  
  
    RETURN;  
END;
```

XÂY DỰNG TRIGGER

**TRIGGER TỰ ĐỘNG CẬP NHẬT
NGÀY XUẤT HÓA ĐƠN LÀ NGÀY
HIỆN TẠI**

```
-- Trigger tự động cập nhật ngày xuất hoá đơn là ngày hiện tại
GO
CREATE TRIGGER trg_UpdateNgayLapHoaDon
ON HoaDon
AFTER INSERT, UPDATE
AS
BEGIN
    DECLARE @SoHoaDon INT, @NgayLapHoaDon DATETIME;

    SELECT @SoHoaDon = SoHoaDon FROM INSERTED;

    -- Nếu chưa có giá trị NgayLapHoaDon thì cập nhật
    SELECT @NgayLapHoaDon = NgayLapHoaDon FROM HoaDon WHERE SoHoaDon = @SoHoaDon;

    IF @NgayLapHoaDon IS NULL
    BEGIN
        UPDATE HoaDon
        SET NgayLapHoaDon = GETDATE()
        WHERE SoHoaDon = @SoHoaDon;
    END;
END;
```

XÂY DỰNG TRIGGER

TRIGGER KIỂM TRA SỐ LƯỢNG TRONG KHO TRƯỚC KHI BÁN

```
--Kiểm tra số lượng trong kho trước khi bán
GO
CREATE TRIGGER trg_CheckKhoHang
ON ChiTietHoaDon
AFTER INSERT
AS
BEGIN
    DECLARE @MaMon INT, @SoLuong INT, @SoLuongKho INT;
    SELECT @MaMon = MaMon, @SoLuong = SoLuong
    FROM INSERTED;
    SELECT @SoLuongKho = SoLuong
    FROM KhoHang
    WHERE MaSP = @MaMon;
    IF @SoLuong > @SoLuongKho
    BEGIN
        PRINT 'Lỗi: Không đủ số lượng sản phẩm trong kho!';
        ROLLBACK TRANSACTION;
    END
END;
END;
```

XÂY DỰNG TRIGGER

TRIGGER CẬP NHẬT ĐIỂM TÍCH LŨY KHI KHÁCH HÀNG THANH TOÁN HÓA ĐƠN

```
--Trigger cập nhật giá trị DiemTichLuy của khách hàng khi thanh toán  
go  
CREATE TRIGGER trg_UpdateDiemTichLuy  
ON ChiTietHoaDon  
AFTER INSERT  
AS  
BEGIN  
    DECLARE @MaKH INT, @ThanhTien DECIMAL(18, 2);  
    SELECT @MaKH = H.MaKH  
    FROM HoaDon H  
    JOIN INSERTED I ON H.SoHoaDon = I.SoHoaDon;  
    SELECT @ThanhTien = SUM(I.Gia * I.SoLuong)  
    FROM INSERTED I  
    WHERE I.SoHoaDon = (SELECT SoHoaDon FROM INSERTED);  
    UPDATE KháchHàng  
    SET DiemTichLuy = DiemTichLuy + FLOOR(@ThanhTien / 1000)  
    WHERE MaKH = @MaKH;  
END;
```

XÂY DỰNG TRIGGER

TRIGGER TỰ ĐỘNG CẬP NHẬT THỜI GIAN GẦN NHẤT TRONG KHO HÀNG

```
-- Tự động cập nhật lại thời gian gần nhất của trong kho hàng  
GO  
CREATE TRIGGER trg_UpdateNgayNhapKho  
ON KhoHang  
AFTER INSERT  
AS  
BEGIN  
    DECLARE @MaSP INT;  
    SELECT @MaSP = MaSP FROM INSERTED;  
    UPDATE KhoHang  
    SET NgayNhapGanNhat = GETDATE()  
    WHERE MaSP = @MaSP;  
END;
```


XÂY DỰNG TRIGGER

TRIGGER KIỂM TRA KHUYẾN MÃI PHẢI LỚN HƠN BẮT ĐẦU

```
-- Trigger kiểm tra khuyến mãi phải lớn hơn ngày bắt đầu
GO
CREATE TRIGGER trg_CheckKhuyenMai
ON KhuyenMai
AFTER INSERT
AS
BEGIN
    DECLARE @ThoiGianApDungStart DATE, @ThoiGianApDungEnd DATE;
    SELECT @ThoiGianApDungStart = ThoiGianApDungStart, @ThoiGianApDungEnd = ThoiGianApDungEnd
    FROM INSERTED;
    IF @ThoiGianApDungEnd <= @ThoiGianApDungStart
    BEGIN
        PRINT 'Lỗi: Ngày kết thúc khuyến mãi phải lớn hơn ngày bắt đầu!';
        ROLLBACK TRANSACTION;
    END;
END;
END;
```

XÂY DỰNG VIEW

VIEW THÔNG TIN NHÂN VIÊN

```
-- Tạo View để hiển thị thông tin nhân viên
CREATE VIEW V_NhanVien AS
SELECT
    NV.MaNV,
    NV.TenNV,
    NV.VaiTro,
    ND.TenDangNhap,
    ND.MatKhau,
    NV.NgaySinh,
    NV.GioiTinh,
    NV.MaCa,
    NV.HinhAnh
FROM
    NhanVien NV
JOIN
    NguoiDung ND ON NV.MaNV = ND.MaNV;
GO
```

XÂY DỰNG VIEW

VIEW THÔNG TIN KHUYẾN MÃI

```
-- Tạo View để hiển thị thông tin khuyến mãi
CREATE VIEW V_KhuyenMai AS
SELECT
    KM.MaKM,
    KM.TenKM AS TenLM,
    KM.MoTa,
    KM.ThoiGianApDungStart,
    KM.ThoiGianApDungEnd,
    KM.GiaTriKM,
    CONCAT(KM.TenKM, ' - ', KM.GiaTriKM, '%') AS TenKMWithValue
FROM
    KhuyenMai KM;
GO
```

XÂY DỰNG VIEW

VIEW THÔNG TIN KHO

```
-- Tạo View để hiển thị thông tin kho
CREATE VIEW V_Kho AS
SELECT
    K.MaSP,
    T.MaMon,
    T.TenMon,
    K.SoLuong,
    T.DonViTinh AS DonVi,
    T.Gia AS GiaBan,
    K.NgayNhapGanNhat
FROM
    KhoHang K
JOIN
    ThucDon T ON K.MaSP = T.MaMon;
GO
```

XÂY DỰNG VIEW

VIEW THÔNG TIN LỊCH SỬ HÓA ĐƠN

```
-- Tạo View để hiển thị thông tin lịch sử hóa đơn
CREATE VIEW V_LichSu AS
SELECT
    SoHoaDon,
    MaKH,
    MaBan,
    MaNV,
    SoGioChoi,
    ThanhTien,
    NgayLapHoaDon,
    HinhThucThanhToan,
    MaKM
FROM
    HoaDon;
GO
```


XÂY DỰNG VIEW

VIEW THÔNG TIN KHÁCH HÀNG

```
-- Tạo View để hiển thị thông tin khách hàng
CREATE VIEW V_KhachHang AS
SELECT
    MaKH,
    HoTen,
    SĐT,
    NgaySinh,
    Email,
    DiemTichLuy,
    -- Tạo trường TenHienThi bằng cách kết hợp HoTen và SĐT
    CONCAT(HoTen, ' - ', SĐT) AS TenHienThi
FROM
    KhachHang;
GO
```

XÂY DỰNG VIEW

VIEW THÔNG TIN THỰC ĐƠN

```
-- Tạo View để hiển thị thông tin các món trong thực đơn
CREATE VIEW V_ThucDon AS
SELECT
    MaMon,
    TenMon,
    LoaiMon,
    NhomThucDon,
    DonViTinh,
    Gia,
    HinhAnh
FROM
    ThucDon;
GO
```

XÂY DỰNG CURSOR

CURSOR CẬP NHẬT TRẠNG THÁI CÁC BÀN

```
--Cập nhật trạng thái thành trống khi khách hàng thanh toán xong
DECLARE @MaBan INT, @TrangThai NVARCHAR(20);
DECLARE BanBilliardsCursor CURSOR FOR
SELECT MaBan, TrangThai FROM BanBilliards WHERE TrangThai = N'Đang Sử Dụng';
OPEN BanBilliardsCursor;
FETCH NEXT FROM BanBilliardsCursor INTO @MaBan, @TrangThai;
WHILE @@FETCH_STATUS = 0
BEGIN

    UPDATE BanBilliards
    SET TrangThai = N'Trống'
    WHERE MaBan = @MaBan;
    FETCH NEXT FROM BanBilliardsCursor INTO @MaBan, @TrangThai;
END;
CLOSE BanBilliardsCursor;
DEALLOCATE BanBilliardsCursor;

go

--Duyệt qua tất cả các Khách Hàng và In Thông Tin
DECLARE @HoTen NVARCHAR(100), @SDT NVARCHAR(15);
DECLARE KháchHangCursor CURSOR FOR
SELECT HoTen, SDT FROM KháchHang;
OPEN KháchHangCursor;
FETCH NEXT FROM KháchHangCursor INTO @HoTen, @SDT;
WHILE @@FETCH_STATUS = 0
BEGIN

    PRINT 'Khách hàng: ' + @HoTen + N', Số điện thoại: ' + @SDT;
    FETCH NEXT FROM KháchHangCursor INTO @HoTen, @SDT;
END;

CLOSE KháchHangCursor;
DEALLOCATE KháchHangCursor;
```

XÂY DỰNG CURSOR

CURSOR CẬP NHẬT THÔNG TIN KHÁCH HÀNG

```
-- Cập nhật thông tin khách hàng
DECLARE @MaBan INT, @TrangThai NVARCHAR(20);

-- Khai báo cursor để duyệt qua các bản trong bảng BanBilliards
DECLARE CS_capNhatTrangThaiBan CURSOR FOR
    SELECT MaBan, TrangThai
    FROM BanBilliards;

-- Mở cursor
OPEN CS_capNhatTrangThaiBan;

-- Lấy qua từng bản
FETCH NEXT FROM CS_capNhatTrangThaiBan INTO @MaBan, @TrangThai;

-- Cập nhật trạng thái bàn
WHILE @@FETCH_STATUS = 0
BEGIN
    -- Kiểm tra nếu bàn đang trống và có yêu cầu sử dụng thì cập nhật trạng thái
    IF @TrangThai = N'Trống'
    BEGIN
        -- Giả sử có yêu cầu đặt bàn (ví dụ qua PhiếuĐặtBàn) thì cập nhật trạng thái sang "Đang Sử Dụng"
        UPDATE BanBilliards
        SET TrangThai = N'Đang Sử Dụng'
        WHERE MaBan = @MaBan;
    END
    -- Nếu bàn đang sử dụng và không có yêu cầu đặt lại bàn, cập nhật lại trạng thái là "Trống"
    ELSE IF @TrangThai = N'Đang Sử Dụng'
    BEGIN
        -- Giả sử bàn đã được giải phóng (ví dụ qua HoáĐơn hoặc PhiếuĐặtBàn)
        UPDATE BanBilliards
        SET TrangThai = N'Trống'
        WHERE MaBan = @MaBan;
    END
    -- Lấy bản tiếp theo trong cursor
    FETCH NEXT FROM CS_capNhatTrangThaiBan INTO @MaBan, @TrangThai;
END;

-- Đóng cursor
CLOSE CS_capNhatTrangThaiBan;
DEALLOCATE CS_capNhatTrangThaiBan;
```

XÂY DỰNG CURSOR

CURSOR TÍNH TỔNG TIỀN CỦA TẤT CẢ HÓA ĐƠN

```
--Tính Tổng Tiền của Tất Cả Hóa Đơn
DECLARE @SoHoaDon INT, @ThanhTien DECIMAL(18, 2), @TongTien DECIMAL(18, 2);
SET @TongTien = 0;
DECLARE HoaDonCursor CURSOR FOR
SELECT SoHoaDon, ThanhTien FROM HoaDon;
OPEN HoaDonCursor;
FETCH NEXT FROM HoaDonCursor INTO @SoHoaDon, @ThanhTien;
WHILE @@FETCH_STATUS = 0
BEGIN

    SET @TongTien = @TongTien + ISNULL(@ThanhTien, 0);
    FETCH NEXT FROM HoaDonCursor INTO @SoHoaDon, @ThanhTien;
END;
PRINT N'Tổng tiền tất cả hóa đơn: ' + CAST(@TongTien AS NVARCHAR(18));

CLOSE HoaDonCursor;
DEALLOCATE HoaDonCursor;
```


XÂY DỰNG CURSOR

CURSOR HIỂN THỊ CÁC NHÂN VIÊN THEO CA

```
-- Hiển thị các nhân viên theo ca
DECLARE @MaCa INT, @MaNV INT, @TenNV NVARCHAR(100);
DECLARE CaLamViecCursor CURSOR FOR
SELECT DISTINCT MaCa
FROM NhanVien;
OPEN CaLamViecCursor;
FETCH NEXT FROM CaLamViecCursor INTO @MaCa;
WHILE @@FETCH_STATUS = 0
BEGIN
    PRINT N'Ca làm việc: ' + CAST(@MaCa AS NVARCHAR(10));
    DECLARE NhanVienInCaCursor CURSOR FOR
    SELECT MaNV, TenNV
    FROM NhanVien
    WHERE MaCa = @MaCa;
    OPEN NhanVienInCaCursor;
    FETCH NEXT FROM NhanVienInCaCursor INTO @MaNV, @TenNV;
    WHILE @@FETCH_STATUS = 0
    BEGIN
        PRINT N' Nhân viên: ' + @TenNV;
        FETCH NEXT FROM NhanVienInCaCursor INTO @MaNV, @TenNV;
    END;
    CLOSE NhanVienInCaCursor;
    DEALLOCATE NhanVienInCaCursor;
    FETCH NEXT FROM CaLamViecCursor INTO @MaCa;
END;
```

XÂY DỰNG CURSOR

CURSOR KIỂM TRA CÁC SẢN PHẨM SỐ LƯỢNG SẮP HẾT


```
--Kiểm tra các sản phẩm số lượng sắp hết
DECLARE @MaSP INT, @SoLuong INT, @TenMon NVARCHAR(100);
DECLARE KhoHangCursor CURSOR FOR
SELECT KH.MaSP, KH.SoLuong, TD.TenMon
FROM KhoHang KH
JOIN ThucDon TD ON KH.MaSP = TD.MaMon
WHERE KH.SoLuong < 10;
OPEN KhoHangCursor;
FETCH NEXT FROM KhoHangCursor INTO @MaSP, @SoLuong, @TenMon;
WHILE @@FETCH_STATUS = 0
BEGIN

    PRINT N'Sản phẩm: ' + @TenMon + N' còn lại: ' + CAST(@SoLuong AS NVARCHAR(10)) + N' sản phẩm';
    FETCH NEXT FROM KhoHangCursor INTO @MaSP, @SoLuong, @TenMon;
END;

CLOSE KhoHangCursor;
DEALLOCATE KhoHangCursor;
```



THIẾT KẾ GIAO DIỆN



DEMO HỆ THỐNG