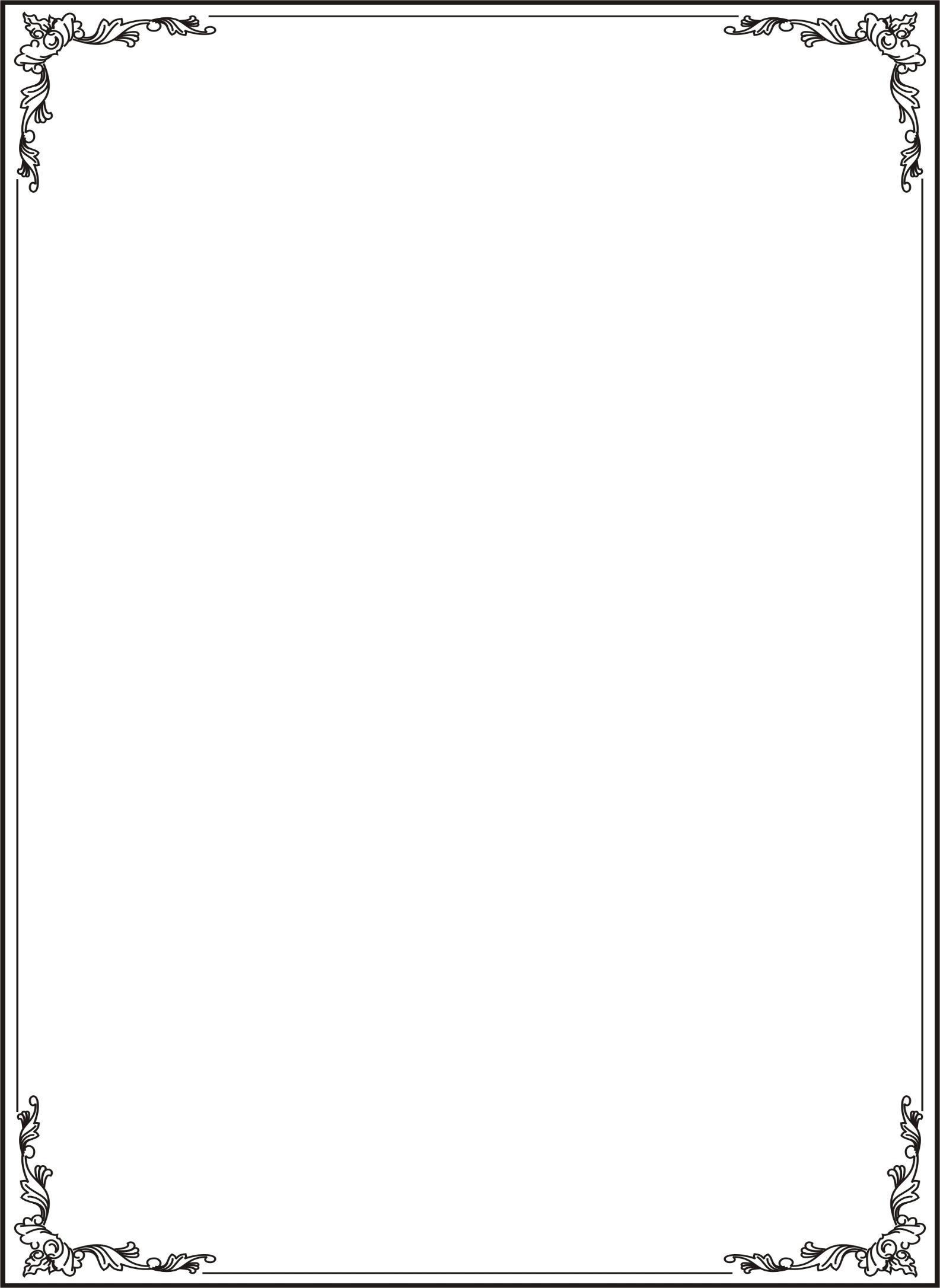
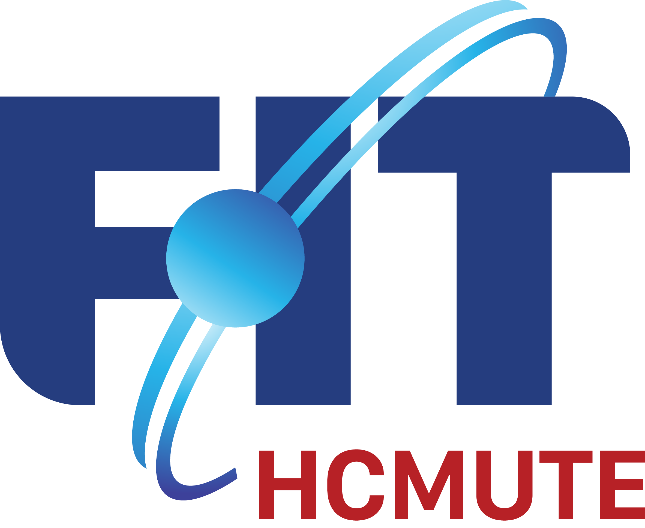
**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỀ TÀI CUỐI KÌ**

**HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

Mã học phần: DBMS330284

**ĐỀ TÀI: HỆ THỐNG QUẢN LÝ DỊCH VỤ QUÁN NET**

Giảng viên hướng dẫn: **TS. Nguyễn Thành Sơn**

**Danh sách sinh viên thực hiện:**

1. Đỗ Phú Luân – 22110372
2. Huỳnh Minh Mẫn – 22110377
3. Đặng Minh Nhật – 22110389
4. Vương Lập Quế - 22110402

*TP. Hồ Chí Minh, tháng 09 năm 2024*

**LỜI CẢM ƠN**

Kính thưa thầy Nguyễn Thành Sơn,

Nhóm em xin bày tỏ lòng biết ơn sâu sắc đến thầy vì sự tận tình hướng dẫn và hỗ trợ trong suốt quá trình thực hiện đồ án môn Hệ quản trị cơ sở dữ liệu. Sự quan tâm và chỉ dạy của thầy đã giúp chúng em không chỉ hoàn thành tốt đồ án mà còn học hỏi được nhiều kinh nghiệm quý báu trong lĩnh vực cơ sở dữ liệu.

Nhờ có thầy, nhóm em đã có cơ hội tiếp thu những kiến thức bổ ích và ứng dụng vào thực tiễn một cách hiệu quả. Qua từng bài học, thầy đã giúp chúng em nắm vững những kỹ năng quan trọng, tạo tiền đề vững chắc cho những công việc và thách thức trong tương lai.

Chúng em vô cùng biết ơn thầy vì đã dành nhiều thời gian, tâm huyết để giúp đỡ và đồng hành cùng nhóm trong quá trình học tập. Một lần nữa, nhóm em xin gửi lời cảm ơn chân thành nhất đến thầy Nguyễn Thành Sơn. Chúng em sẽ mãi trân trọng và ghi nhớ những gì thầy đã truyền đạt.

**Trân trọng!**

**MỤC LỤC**

[CHƯƠNG 1: ĐẶC TẢ BÀI TOÁN 1](#_Toc182540919)

[CHƯƠNG 2: THIẾT KẾ CƠ SỞ DỮ LIỆU 8](#_Toc182540920)

[2.1. Thiết kế cơ sở dữ liệu mức quan niệm 8](#_Toc182540921)

[2.2. Thiết kế cơ sở dữ liệu mức logic 9](#_Toc182540922)

[2.3. Thiết kế cơ sở dữ liệu mức vật lý 9](#_Toc182540923)

[2.3.1 Constraints 9](#_Toc182540924)

[2.3.1.1 Bảng TaiKhoan 9](#_Toc182540925)

[2.3.1.2 Bảng NguoiQuanLy 9](#_Toc182540926)

[2.3.1.3 Bảng KhachHang 10](#_Toc182540927)

[2.3.1.4 Bảng PhienDangNhap 11](#_Toc182540928)

[2.3.1.5 Bảng UuDai 11](#_Toc182540929)

[2.3.1.6 Bảng HoaDon 12](#_Toc182540930)

[2.3.1.7 Bảng DichVu 13](#_Toc182540931)

[2.3.1.8 Bảng ChiTietHoaDon 14](#_Toc182540932)

[2.3.1.9 Bảng NapTien 14](#_Toc182540933)

[2.3.1.10 Bảng MayTinh 15](#_Toc182540934)

[2.3.2 Relation Diagram 16](#_Toc182540935)

[CHƯƠNG III: CÀI ĐẶT CHỨC NĂNG 17](#_Toc182540936)

[3.1 Cài đặt CSDL và các ràng buộc: 17](#_Toc182540937)

[3.2 Cài đặt Trigger: 20](#_Toc182540938)

[3.2.1 Trigger trong KhachHang 20](#_Toc182540939)

[3.2.1.1 UpdateThoiGianConLai\_PhienDangNhap 20](#_Toc182540940)

[3.2.1.2 Update\_LoaiKhachHang 20](#_Toc182540941)

[3.2.2 Trigger trong PhienDangNhap 21](#_Toc182540942)

[3.2.2.1 trg\_UpdateTinhTrangMayTinh\_AfterInsertPhienDangNhap 21](#_Toc182540943)

[3.2.2.2 trg\_UpdateTinhTrangMayTinh\_AfterDeletePhienDangNhap 21](#_Toc182540944)

[3.2.3 Trigger trong UuDai 22](#_Toc182540945)

[3.2.3.1 tr\_CheckUuDai 22](#_Toc182540946)

[3.2.4 Trigger trong HoaDon 23](#_Toc182540947)

[3.2.4.1 trg\_RecalculateTriGia\_AfterUpdateHoaDon 23](#_Toc182540948)

[3.2.5 Trigger trong DichVu 24](#_Toc182540949)

[3.2.5.1 trg\_AfterDelete\_DichVu 24](#_Toc182540950)

[3.2.5.2 tr\_CheckDichVu 24](#_Toc182540951)

[3.2.6 Trigger trong ChiTietHoaDon 25](#_Toc182540952)

[3.2.6.1 trg\_CalculateTongGiaTien\_AfterInsertChiTietHoaDon 25](#_Toc182540953)

[3.2.7 Trigger trong NapTien 25](#_Toc182540954)

[3.2.7.1 trg\_UpdateThoiGianConLaiVaDiemTichLuy\_AfterNapTien 25](#_Toc182540955)

[3.3 Cài đặt View: 27](#_Toc182540956)

[3.4 Cài đặt Function 30](#_Toc182540957)

[3.4.1 dbo.LayPhienDangNhap 30](#_Toc182540958)

[3.4.2 fn\_LayMaKhachHang 31](#_Toc182540959)

[3.4.3 GetTenTaiKhoan 32](#_Toc182540960)

[3.4.4 KiemTraKhachHangDangNhap 33](#_Toc182540961)

[3.4.5 KiemTraNguoiQuanLyDangNhap 35](#_Toc182540962)

[3.4.6 LayLoaiKhachHang 37](#_Toc182540963)

[3.4.7 fn\_TinhDoanhThuDichVuTheoNgay 38](#_Toc182540964)

[3.4.8 fn\_TinhDoanhThuNapTienTheoNgay 38](#_Toc182540965)

[3.5 Cài đặt Procedure 39](#_Toc182540966)

[3.5 Cài đặt Procedure 44](#_Toc182540967)

[3.5.1 sp\_AddQuanLy 44](#_Toc182540968)

[3.5.2 DoiMatKhau 45](#_Toc182540969)

[3.5.3 NapTienProcedure 47](#_Toc182540970)

[3.5.4 sp\_AddKhachHang 48](#_Toc182540971)

[3.5.5 sp\_CapNhatTinhTrangHoaDon 50](#_Toc182540972)

[3.5.6 sp\_ChuyenDoiMaKhachHangSangMaTaiKhoan 52](#_Toc182540973)

[3.5.8 sp\_SearchKhachHangByTaiKhoan 53](#_Toc182540974)

[3.5.9 sp\_ThemDichVuVaoChiTietHoaDon 54](#_Toc182540975)

[3.5.10 sp\_UpdateTinhTrangMayTinh 57](#_Toc182540976)

[3.5.11 sp\_XemHoaDon\_KhachHang 59](#_Toc182540977)

[3.5.12 SuaDichVuProcedure 61](#_Toc182540978)

[3.5.13 SuaUuDaiProcedure 62](#_Toc182540979)

[3.5.14 ThemDichVuProcedure 64](#_Toc182540980)

[3.5.15 ThemMaUuDaiVaoHoaDon 65](#_Toc182540981)

[3.5.16 ThemPhienDangNhap 67](#_Toc182540982)

[3.5.17 ThemUuDaiProcedure 69](#_Toc182540983)

[3.5.18 TimKiemDichVuProcedure 71](#_Toc182540984)

[3.5.19 TimKiemUuDaiThuongProcedure 72](#_Toc182540985)

[3.5.20 TimKiemUuDaiVipProcedure 73](#_Toc182540986)

[3.5.21 XemChiTietHoaDonProcedure 75](#_Toc182540987)

[3.5.22 XoaDichVuProcedure 77](#_Toc182540988)

[3.5.23 dbo.XoaPhienDangNhap 78](#_Toc182540989)

[3.5.24 XoaUuDaiProcedure 79](#_Toc182540990)

[CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN 81](#_Toc182540991)

[CHƯƠNG 5: THIẾT KẾ GIAO DIỆN HỆ THỐNG 83](#_Toc182540992)

[5.1 Form màn hình kết nối database 84](#_Toc182540993)

[5.2 Form đăng nhập: 85](#_Toc182540994)

[5.3 Các Form dành cho tài khoản người quản lý 86](#_Toc182540995)

[5.3.1 Form màn hình chính 86](#_Toc182540996)

[5.3.1.1 Form Quản lý thành viên 86](#_Toc182540997)

[5.3.1.2 Form Đăng ký thành viên 86](#_Toc182540998)

[5.3.1.3 Form Đăng ký quản lý 87](#_Toc182540999)

[5.3.1.4 Form Quản lý máy 87](#_Toc182541000)

[5.3.1.5 Form Quản lý gọi dịch vụ 87](#_Toc182541001)

[5.3.1.6 Form Quản lý ưu đãi 89](#_Toc182541002)

[5.3.1.7 Form Thống kê trong tuần 89](#_Toc182541003)

[5.3.1.8 Form Nạp tiền 90](#_Toc182541004)

[5.3.2 Form Quản lý danh mục 92](#_Toc182541005)

[5.3.2.1 Form Đồ ăn 93](#_Toc182541006)

[5.3.2.2 Form Thức uống 93](#_Toc182541007)

[5.3.2.3 Form Thẻ cào 94](#_Toc182541008)

[5.4 Form Khách hang 95](#_Toc182541009)

[5.4.1 Form Dịch vụ 96](#_Toc182541010)

[5.4.2 Form Hóa đơn 97](#_Toc182541011)

[5.4.3 Form Đổi mật khẩu 98](#_Toc182541012)

**LỜI MỞ ĐẦU**

CHƯƠNG 1: ĐẶC TẢ BÀI TOÁN

Hệ thống quản lý quán net được xây dựng xoay quanh hai nhóm đối tượng thao tác chính: người quản lý và khách hàng.

Để sử dụng hệ thống, các đối tượng sử dụng sẽ được cấp một tài khoản mang một mã tài khoản, mỗi tài khoản sẽ bao gồm tên tài khoản, mật khẩu để truy cập hệ thống.

Tài khoản người quản lý sẽ có mã người quản lý riêng biệt.

Đối với tài khoản khách hàng gồm có mã khách hàng, thời gian thuê máy còn lại, loại khách hàng và điểm tích lũy.

Khi khách hàng đăng nhập, hệ thống sẽ tạo mới một phiên đăng nhập để khách hàng truy cập vào máy tính, phiên đăng nhập gồm có thời gian bắt đầu, thời gian sử dụng, thời gian còn lại.

Mỗi máy tính sẽ có một mã số riêng để phân biệt với các máy khác. Mỗi máy sẽ lưu trữ các thông tin quan trọng bao gồm mã máy, tình trạng hiện tại của máy tính (có thể là trống, đang sử dụng hoặc đang bảo trì)

Khi máy tính gặp sự cố, người quản lý sẽ tiến hành bảo trì máy tính. Một máy tính chỉ có một người quản lý bảo trì, một người quản lý có bảo trì nhiều máy tính.

Một tài khoản khách hàng chỉ có thể đăng nhập vào một phiên đăng nhập duy nhất, phiên đăng nhập này sẽ được máy tính ghi nhận để thay đổi tình trạng hoạt động của máy tính, cho phép khách hàng đăng nhập vào máy tính. Trong một thời điểm, mỗi máy chỉ có thể có một phiên đăng nhập do khách hàng đăng nhập.

Hệ thống cung cấp nhiều dịch vụ đa dạng. Các dịch vụ này đều được quản lý theo các thông tin chung như mã dịch vụ, tên dịch vụ, loại dịch vụ, đơn giá, số lượng.

Quán nét có nhiều ưu đãi cho khách hàng và được đặc trưng với nhau bằng mã ưu đãi, tên ưu đãi, giá trị, thời gian bắt đầu, thời gian kết thúc, điều kiện, số lượng và tình trạng. Mỗi ưu đãi sẽ được quản lý bởi một người quản lý, một người quản lý có thể quản lý nhiều ưu đãi hoặc không quản lý ưu đãi nào.

Hóa đơn được quản lý với nhiều thông tin như mã hóa đơn, ngày tạo hóa đơn, trạng thái, trị giá. Mỗi hóa đơn chỉ được áp dụng một ưu đãi. Số lượng ưu đãi có giới hạn nên chỉ có thể áp dụng được cho một số hóa đơn.

Mỗi hóa đơn bao gồm các dịch vụ mà khách hàng đã yêu cầu, được lưu trong chi tiết hóa đơn. Chi tiết hóa đơn sẽ có một mã chi tiết hóa đơn và ghi nhận thông tin về số lượng từng dịch vụ và tổng tiền của các dịch vụ đó. Một hóa đơn có thể chứa nhiều dịch vụ khác nhau và mỗi dịch vụ cũng có thể xuất hiện trong nhiều hóa đơn khác nhau. Quản lý có thể xuất hóa đơn cho khách hàng.

Mỗi khách hàng sẽ thanh toán một hóa đơn tương ứng của mình (nếu có). Hóa đơn này tham chiếu đến máy tính hiện tại mà khách hàng đang sử dụng

Tài khoản người quản lý có thể quản lý các loại dịch vụ mà quán net cung cấp.

Tài khoản khách hàng được nạp tiền bởi người quản lý. Tài khoản khách hàng có thể được nạp tiền nhiều lần từ người quản lý. Người quản lý có thể nạp tiền cho nhiều khách hàng. Mỗi lần nạp tiền sẽ có thông tin về thời gian nạp tiền, giá trị nạp và thời gian quy đổi. Thời gian quy đổi được tính toán dựa trên giá trị nạp. Điểm tích lũy trong mỗi tài khoản khách hàng sẽ được cộng dồn thời gian quy đổi.

Quán net cần một hệ thống quản lý dịch vụ để tối ưu hóa việc kinh doanh và quản lý tài nguyên. Hệ thống này sẽ quản lý tài khoản, các loại dịch vụ, tình trạng máy móc và các khoản thanh toán của khách hàng.

Về quản lý tài khoản, người quản lý sẽ tiến hành khởi tạo tài khoản gồm tên tài khoản và mật khẩu cho khách hàng. Người quản lý có thể tạo tài khoản cho nhiều khách hàng, mỗi khách hàng sẽ được cung cấp một tài khoản duy nhất cho phép đăng nhập vào hệ thống dịch vụ của quán net. Trong đó, các thông tin của tài khoản khách hàng như thời gian còn lại, điểm tích lũy là 0 và loại khách hàng là thường.

Bên cạnh việc khởi tạo tài khoản khách hàng, người quản lý cũng có thể khởi tạo tài khoản cho người quản lý khác bao gồm tên tài khoản và mật khẩu. Mỗi người quản lý sẽ được cung cấp một tài khoản được cấp quyền quản trị cho phép đăng nhập vào hệ thống và quản lý dịch vụ của quán net.

Người quản lý sẽ tiến hành nạp tiền vào tài khoản khách với số tiền mà khách hàng cung cấp, số tiền sẽ được cộng vào tài khoản của khách hàng và quy ra thời gian chơi tương ứng. Khi khách hàng nạp tiền, ứng với giá trị nạp điểm tích lũy của khách hàng sẽ được tăng lên, khi điểm tích lũy đến mức nhất định tài khoản khách hàng sẽ được nâng lên loại tài khoản cao cấp hơn.

Khách hàng có thể đăng nhập vào hệ thống bằng tài khoản đã cấp, hệ thống kiểm tra thông tin đăng nhập và cho phép truy cập vào máy tính khi hợp lệ. Sau khi đăng nhập, hệ thống ghi nhận thời gian bắt đầu sử dụng máy và tiếp tục theo dõi các hoạt động khác của khách hàng như sử dụng dịch vụ.

Để theo dõi trạng thái máy tính khi khách hàng đăng nhập vào hệ thống, máy tính mà khách hàng sử dụng sẽ tự động được đánh dấu là "đang sử dụng" và hệ thống ghi nhận thời gian bắt đầu phiên đăng nhập. Khi đó thời gian còn lại sẽ được đếm ngược về giá trị 0. Khi khách hàng đăng xuất, hệ thống sẽ dừng bộ đếm thời gian thuê máy còn lại và trạng thái máy sẽ được đánh dấu là “trống”. Nếu chơi đến hết thời gian, hệ thống sẽ tạm thời đăng xuất và xoá phiên đăng nhập, nếu khách hàng muốn tiếp tục sử dụng máy thì phải nạp thêm tiền và đăng nhập lại, còn nếu khách hàng đăng xuất thì hệ thống sẽ xoá phiên đăng nhập đi. Nếu máy tính gặp sự cố, người quản lý thay đổi trạng thái máy tính thành "đang bảo trì", khách hàng không thể đăng nhập vào máy đang bảo trì.

Hệ thống sẽ lưu trữ danh sách các dịch vụ như đồ ăn, thức uống, thẻ cào,... vào menu dịch vụ. Hệ thống cho phép người quản lý có thể cập nhật menu, thêm mới hoặc xóa bỏ các dịch vụ tùy vào nhu cầu kinh doanh của quán net.

Ngoài ra, quán net còn có chức năng quản lý doanh số bán hàng thông qua việc thống kê các khoảng tiền từ việc nạp tiền, gọi dịch vụ theo biểu đồ tròn và biểu đồ cột, biểu đồ cột được thống kê theo ngày từ thứ hai đến chủ nhật giúp người quản lý dễ dàng theo dõi nhu cầu sử dụng dịch vụ.

Thông qua giao diện menu trên máy tính, khách hàng có thể gọi các dịch vụ đồ ăn, thức uống, thẻ cào,.... Khi đó, hệ thống sẽ quản lý việc sử dụng dịch vụ bằng cách tạo hóa đơn tạm thời với trạng thái của hóa đơn là “chưa thanh toán”. Các dịch vụ này sẽ được thêm vào hóa đơn tạm thời của khách hàng. Sau khi gọi dịch vụ, hệ thống sẽ gửi hóa đơn đó đến quản lý, từ đó người quản lý có thể đáp ứng được nhu cầu dịch vụ của khách hàng. Hệ thống tự động cập nhật chi phí dịch vụ vào hóa đơn tạm thời của khách hàng, giúp khách hàng dễ dàng kiểm soát số tiền mình cần thanh toán. Mỗi khách hàng có thể kiểm tra dịch vụ đã gọi trong hóa đơn.

Để phát triển doanh thu, quán net có thể áp dụng các chương trình ưu đãi lên hóa đơn, mỗi ưu đãi sẽ có điều kiện kích hoạt riêng dựa vào cấp độ thành viên của khách hàng. Người quản lý có thể tạo, xóa, sửa ưu đãi tùy vào mục tiêu kinh doanh.

Sau khi khách hàng đã hoàn tất sử dụng dịch vụ, khách hàng sẽ tiến hành thanh toán các dịch vụ đã sử dụng trước khi đăng xuất khỏi máy, và hệ thống sẽ tự động tiến hành kết cập nhật trạng thái hóa đơn là “đã thanh toán”.  Khách hàng có thể nhận hóa đơn từ người quản lý bảng cách yêu cầu người quản lý xuất hóa đơn

Trường hợp hết thời gian sử dụng nhưng khách hàng vẫn chưa thanh toán được hóa đơn, khách hàng sẽ thanh toán hóa đơn trực tiếp với người quản lý, người quản lý sẽ xác nhận điều này lên hệ thống.

Để có cái nhìn toàn diện và chi tiết hơn về hệ thống quản lý dịch vụ quán net, chúng ta cần tập trung đi sâu vào việc mô tả giao diện của hệ thống. Cũng giống như chức năng, giao diện được phân chia thành hai phần riêng biệt, đáp ứng nhu cầu của hai đối tượng: người quản lý và khách hàng.

Khi người quản lý truy cập vào hệ thống quản lý quán net, họ sẽ thấy màn hình đăng nhập. Màn hình với hai ô nhập thông tin tài khoản mật khẩu. Ngay bên dưới là nút “Đăng nhập”, khi nhấn vào đó nếu thông tin đúng thì màn hình máy sẽ được chuyển đến giao diện quản lý dịch vụ.

Sau khi đăng nhập thành công, màn hình hiện ra với hai nút lựa chọn cho người quản lý: màn hình chính, quản lý danh mục. Khi người quản lý nhấn vào một trong các lựa chọn này, hệ thống sẽ chuyển sang màn hình tương ứng.

Nếu người quản lý chọn vào nút màn hình chính sẽ xuất hiện màn hình bao gồm các tùy chọn khác như quản lý thành viên, đăng ký thành viên, quản lý máy, quản lý gọi dịch vụ, quản lý chương trình ưu đãi và quay lại trang chủ. Đây là màn hình bao gồm các chức năng hoạt động cơ bản của quán, phía bên trái trên cùng là một thanh công cụ ngang là nơi điều khiển cho phép người quản lý có thể di chuyển sang các tùy chọn khác như đã liệt kê ở trên, điều này cho phép người quản lý dễ dàng điều hướng đến các phần cụ thể mà họ cần thao tác.

Quản lý thành viên cũng là một chức năng không thể thiếu. Khi vào giao diện này sẽ thấy ở phía trên cùng là ô để nhập vào từ khóa cần tìm và bên cạnh là nút tìm kiếm, ở bên dưới là một bảng danh sách các khách hàng của quán được hiện ra theo từng ô, mỗi ô là một thành viên với các thông tin chi tiết bên trong như mã khách hàng, tên tài khoản, loại thành viên, thời gian còn lại trong tài khoản và điểm tích lũy. Bên dưới góc phải của mỗi ô là một dấu cộng, khi nhấn vào đó, một cửa sổ gia hạn thời gian chơi sẽ bật lên, màn hình nhỏ hiện lên có ba ô gồm các ô tên tài khoản không cho phép sửa và được nhập sẵn, bên dưới là ô để nhập số tiền khách hàng muốn nạp thêm, ô kế tiếp bên dưới là ô sẽ hiện lên thời gian được cộng thêm vào tài khoản ứng với số tiền nhập vào ô bên trên, dưới cùng là nút nạp tiền và nút thoát.

Khi muốn đăng ký thành viên, nhân viên sẽ chọn vào mục đăng ký thành viên. Giao diện sẽ hiện lên có ba ô nhập liệu thông tin tài khoản, mật khẩu và xác nhận mật khẩu. Bên dưới là nút "Đăng ký" và “Thoát”.

Tương tự như khi tạo tại khoản khách hàng, tạo người quản lý cũng tương tự như vậy. Khi nhân viên chọn vào mục đăng ký người quản lý, giao diện sẽ hiện lên có ba ô nhập liệu thông tin tài khoản, mật khẩu và xác nhận mật khẩu. Bên dưới là nút "Đăng ký" và Thoát.

Khi nhấn vào màn hình quản lý máy, sẽ có một loạt các ô, mỗi ô đại diện cho một máy tính trong quán. Mỗi ô sẽ hiện một hình ảnh đại diện cho máy, cùng với mã máy. Bên dưới hình ảnh máy là sẽ là thông tin chi tiết về trạng thái của máy như “đang sử dụng”, “trống”, hoặc “đang bảo trì”.

Nếu máy đang được sử dụng, hệ thống cũng sẽ hiển thị thời gian khách hàng đã sử dụng máy và thời gian còn lại trong tài khoản của họ. Khi đó, ở góc phải của mỗi ô là một nút nhỏ có biểu tượng dấu cộng. Khi nhấn vào đây, một cửa sổ gia hạn thời gian chơi sẽ bật lên, giao diện cửa sổ  này giống với giao diện nạp tiền ở màn hình quản lý thành viên.

Trường hợp máy tính gặp sự cố, nút “Bảo trì” sẽ được hiển thị ngay bên dưới trong mỗi ô giúp người quản lý dễ dàng cập nhật trạng thái máy tính thành đang bảo trì để tránh khách hàng đăng nhập vào máy đó.

Một trong những tính năng hữu ích khác là quản lý gọi dịch vụ. Màn hình này có hai nút phía trên cùng, nút "Chưa thanh toán" và nút "Đã thanh toán". Nếu ấn vào nút "Chưa thanh toán", màn hình sẽ hiển thị giao diện hiện danh sách các hóa đơn dịch vụ mà khách hàng đã gọi nhưng chưa thanh toán. Mỗi hóa đơn đều có nút "Chi tiết", khi ấn vào sẽ hiện lên màn hình chứa thông tin cụ thể về các dịch vụ đã gọi. Trong chi tiết hóa đơn của hóa đơn chưa thanh toán, người quản lý có thể nhấn vào nút "Đã nhận thanh toán" hóa đơn đó sẽ chuyển sang trạng thái "Đã thanh toán" và xuất hiện bên giao diện hóa đơn đã thanh toán. Tại giao diện hóa đơn đã thanh toán, hóa đơn khi ấn vào nút chi tiết sẽ hiện chi tiết hóa đơn và có thêm nút "Xuất hóa đơn" bên dưới để in hóa đơn (nếu cần).

Hệ thống còn cung cấp giao diện quản lý chương trình ưu đãi, nơi người quản lý có thể tạo, chỉnh sửa hoặc xóa các chương trình khuyến mãi cho khách hàng. Màn hình này được chia thành hai khu vực: khu vực trình bày và khu vực điều khiển. Khu vực trình bày liệt kê danh sách các ưu đãi theo từng ô hiện các thông tin cơ bản như tên ưu đãi, ngày bắt đầu, ngày kết thúc, điều kiện và giá trị. Trong khu vực điều khiển, chứa các ô có thể nhập thông tin mới hoặc chứa các thông tin cụ thể khi nhấn vào một ưu đãi bất kỳ ở khu vực trình bày, các nút thêm, xóa và sửa cho phép người quản lý tùy chỉnh các ưu đãi hiện có. Khi ấn vào nút thêm thì sẽ tạo thêm ưu đãi dựa vào thông tin những ô ở khu vực này gồm có các ô cho phép nhập tên ưu đãi, ô nhập ngày bắt đầu và hết hạn, điều kiện và ô giá trị ưu đãi và bên dưới là nút thêm, khi ấn vào sẽ thêm thành công, bên cạnh là nút sửa sẽ chỉnh sửa thông tin ưu đãi và nút xóa dùng để xóa ưu đãi khỏi khu vực trình bày.

Cuối cùng, Khi nhấn vào trang thống kê trong tuần, màn hình sẽ hiển thị hai biểu đồ cột bên trái và biểu đồ tròn bên phải. Biểu đồ bên trái sẽ hiển thị doanh thu trong các ngày trong tuần dựa trên bốn loại dịch vụ đồ ăn, thức uống, thẻ cào và nạp tiền. Bên phải là biểu đồ tròn, biểu đồ tròn sẽ hiển thị phần trăm doanh thu dựa trên bốn loại dịch vụ đó.

Khi nhấn vào tùy chọn quay lại trang chủ, màn hình sẽ được chuyển lại trang chủ có hai nút lớn để thao tác với nút còn lại. Ấn vào nút quản lý danh mục sẽ hiện lên các trang quản lý các danh mục như: đồ ăn, thức uống, thẻ cào,...

 Khi ấn vào từng trang cụ thể, trang màn hình hiện lên sẽ có bố cục 2 phần. Phần trên sẽ là ô tìm kiếm, các ô thông tin và các nút thêm, xóa, sửa. Phần bên dưới sẽ là bảng hiển thị thông tin của dịch vụ ứng với danh mục, mỗi dòng là một loại dịch vụ thuộc danh mục dịch vụ tương ứng. Khi người dùng chọn vào từng dịch vụ, thông tin của dịch vụ đó sẽ hiển thị cụ thể lên các ô thông tin phía bên trên.

Về phía màn hình khách hàng, màn hình sẽ được chia bố cục làm 2 phần,

Về màn hình phía khách hàng, góc ngoài cùng bên phải sẽ có một cửa sổ hình chữ nhật hiển thị thông tin cơ bản trong các ô như tên tài khoản, thời gian bắt đầu, thời gian sử dụng, thời gian còn lại, chi phí dịch vụ bên dưới là các nút dịch vụ, xem hoá đơn, đổi mật khẩu, đăng xuất mà khách hàng có thể bấm vào.

Khi khách hàng ấn vào nút dịch vụ sẽ hiện lên một màn hình. Phía trên cùng có thanh tìm kiếm. Phía dưới thanh tìm kiếm có các ô chọn loại dịch vụ được hiển thị. Khi khách muốn gọi đồ ăn khách hàng có thể nhấn vào ô đồ ăn, khi đó bên dưới sẽ hiện lên danh sách các món ăn của quán được hiển thị theo từng ô, mỗi ô chứa hình ảnh món, tên và giá được hiển thị, phía bên phải mỗi món có một ô và 2 mũi tên lên, xuống để chọn số lượng gọi, có thể điền trực tiếp hoặc dùng nút, giá trị mặc định ban đầu là 0. Ở phía dưới sẽ có nút gửi yêu cầu. Khách hàng còn có thể truy cập vào cửa sổ hiển thị hóa đơn thông qua nút hóa đơn ở bên phải nút dịch vụ.

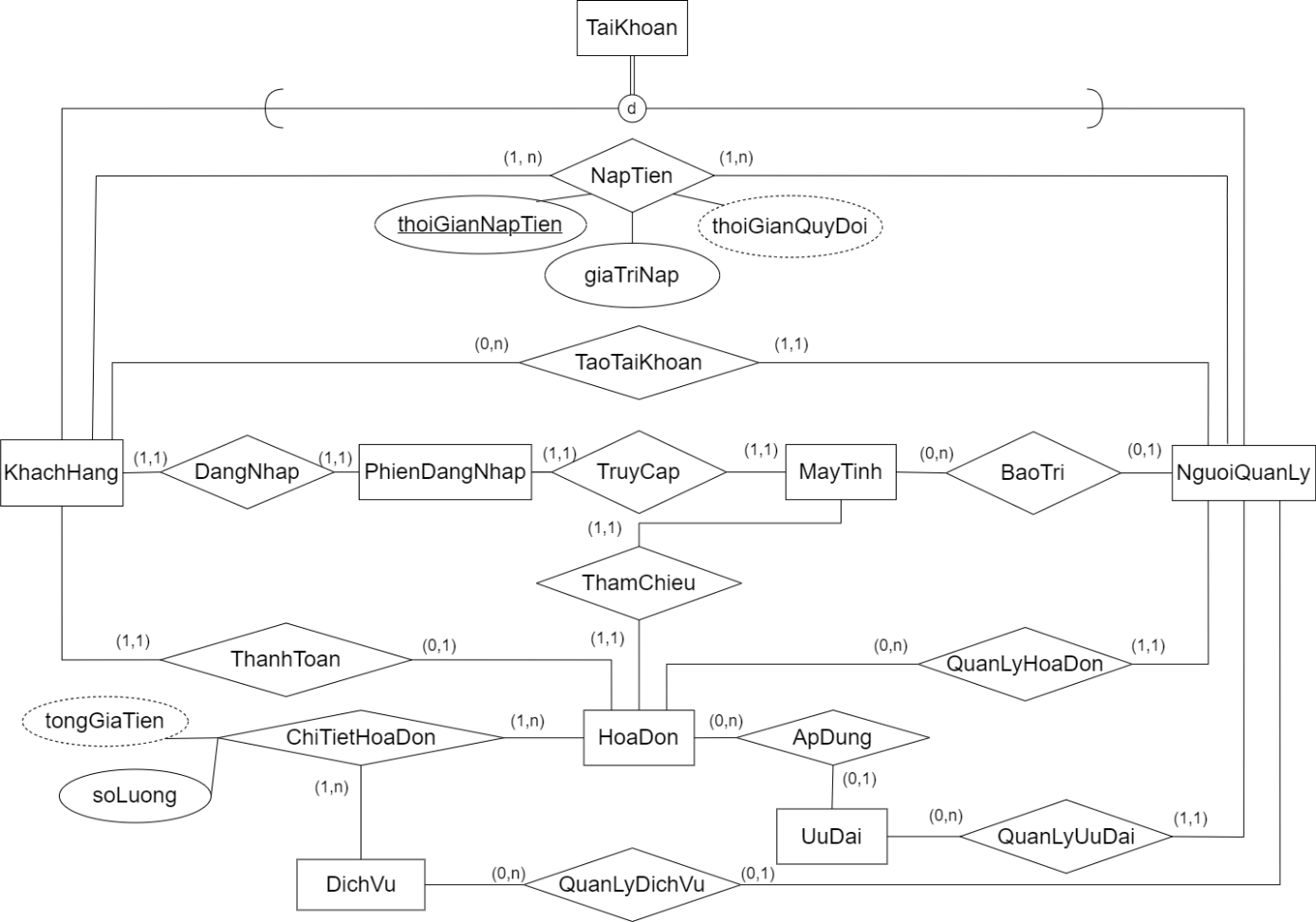
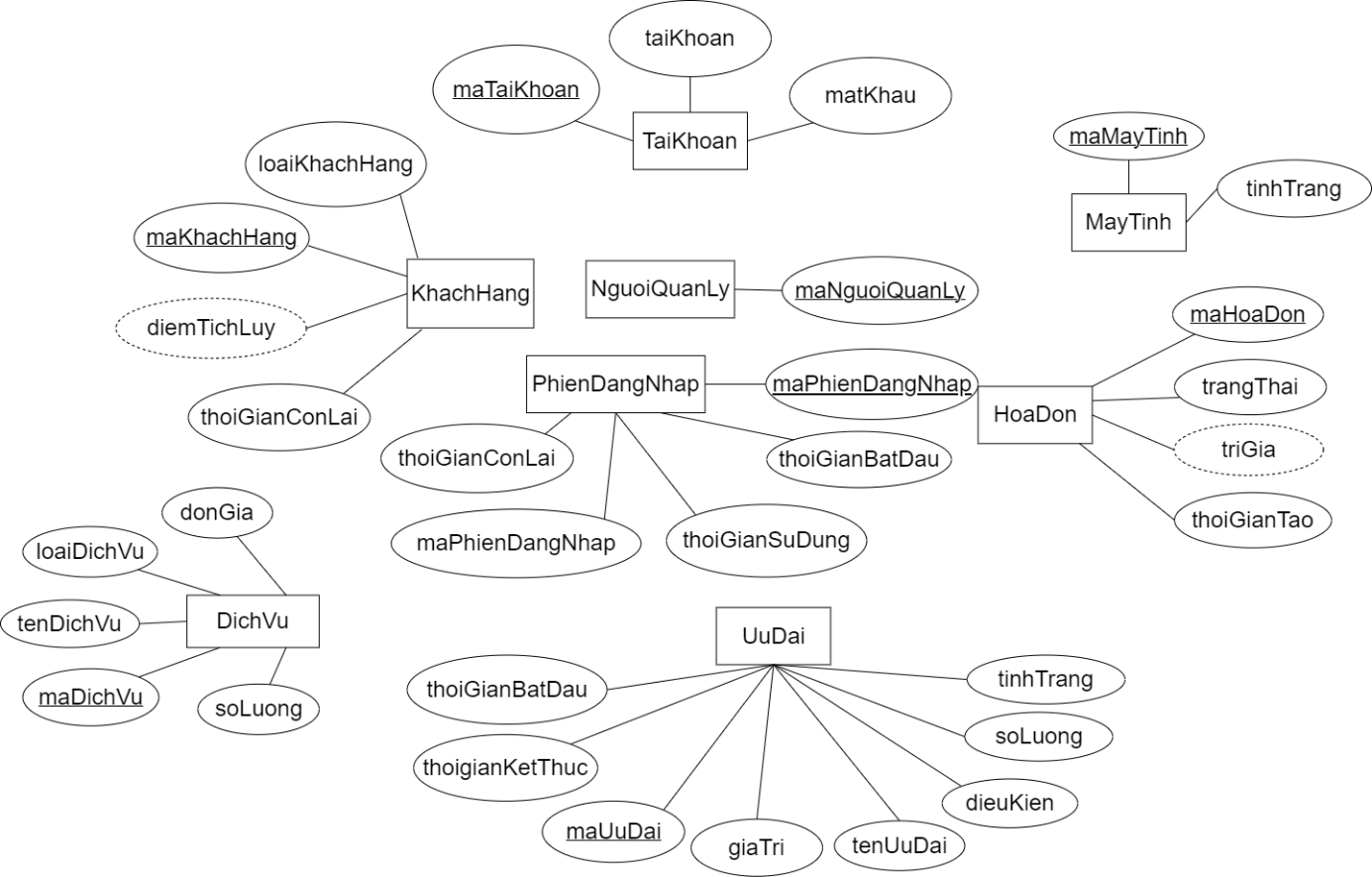
Màn hình hóa đơn được chia bố cục thành 3 phần gồm thông tin hóa đơn, thông tin chi tiết hóa đơn và tổng giá trị hóa đơn. Phần thông tin hóa đơn được hiển thị phía trên, sẽ hiển thị các thông tin của hóa đơn như mã hóa đơn, ngày tạo hóa đơn, tài khoản khách hàng, mã máy tính. Phần thông tin chi tiết hóa đơn được hiển thị ở phía bên dưới dạng bảng với mỗi dòng là thông tin như mã dịch vụ, số lượng, ưu đãi, tổng giá trị của từng dịch vụ mà đã yêu cầu. Phần tổng giá trị hóa đơn là một dòng cuối cùng bên dưới bảng thông tin chi tiết hóa đơn. Bên trên nút thanh toán hóa đơn sẽ có nút chọn ưu đãi, khi nhấn vào sẽ hiện ra một màn hình hiển thị trên cùng là thanh tìm kiếm theo tên, danh sách các ưu đãi và bên phải mỗi ưu đãi là một nút chọn ưu đãi.

Ấn vào nút đổi mật khẩu sẽ hiện lên màn hình cho phép người dùng đổi mật khẩu. Có các ô nhập tên tài khoản, mật khẩu và mật khẩu mới. Bên dưới là nút đổi và thoát khỏi màn hình.

Tổng hợp lại, hệ thống quản lý dịch vụ quán net là một công cụ mạnh mẽ giúp tối ưu hóa việc quản lý tài nguyên và cung cấp dịch vụ hiệu quả. Với các chức năng đa dạng từ quản lý khách hàng, quản lý máy tính, dịch vụ ăn uống, đến quản lý kho và doanh thu, hệ thống không chỉ giúp quán net hoạt động trơn tru mà còn tạo ra trải nghiệm dịch vụ tốt nhất cho khách hàng, đồng thời đảm bảo tăng trưởng doanh thu cho quán.

CHƯƠNG 2: THIẾT KẾ CƠ SỞ DỮ LIỆU

2.1. Thiết kế cơ sở dữ liệu mức quan niệm

****

2.2. Thiết kế cơ sở dữ liệu mức logic

Từ sơ đồ thực thể kết hợp (ERD), ta có các lược đồ quan hệ:

* TaiKhoan (maTaiKhoan, taiKhoan, matKhau)
* NguoiQuanLy (maNguoiQuanLy, maTaiKhoan)
* KhachHang (maKhachHang, maTaiKhoan, maNguoiQuanLy , thoiGianConLai, loaiKhachHang, diemTichLuy)
* PhienDangNhap (maPhienDangNhap, maKhachHang, maMayTinh, thoiGianBatDau, thoiGianSuDung, thoiGianConLai)
* UuDai (maUuDai, maNguoiQuanLy, tenUuDai, giaTri, thoiGianBatDau, thoiGianKetThuc, dieuKien, soLuong, tinhTrang)
* HoaDon (maHoaDon, maNguoiQuanLy, thoiGianTao, maKhachHang, maMayTinh, trangThai, triGia, maUuDai)
* DichVu (maDichVu, maNguoiQuanLy, tenDichVu, loaiDichVu, donGia, soLuong)
* ChiTietHoaDon (maHoaDon, maDichVu, soLuong, tongGiaTien)
* NapTien (maNguoiQuanLy, maKhachHang, thoiGianNapTien, giaTriNap, thoiGianQuyDoi)
* MayTinh (maMayTinh, maNguoiQuanLy, tinhTrang)

2.3. Thiết kế cơ sở dữ liệu mức vật lý

2.3.1 Constraints

2.3.1.1 Bảng TaiKhoan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| **1** | maTaiKhoan | INT | Mã tài khoản | PRIMARY KEY |
| **2** | taiKhoan | VARCHAR(50) | tên tài khoản | NOT NULL UNIQUE |
| **3** | matKhau | VARCHAR(50) | mật khẩu | NOT NULL |

2.3.1.2 Bảng NguoiQuanLy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| **1** | maNguoiQuanLy | INT | Mã người quản lý | PRIMARY KEY |
| **2** | maTaiKhoan | INT | Mã tài khoản | FOREIGN KEY (maTaiKhoan) REFERENCES TaiKhoan(maTaiKhoan ) |

2.3.1.3 Bảng KhachHang

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| **1** | maKhachHang | INT | Mã khách hàng | PRIMARY KEY |
| **2** | maTaiKhoan | INT | Mã tài khoản | FOREIGN KEY (maTaiKhoan) REFERENCES TaiKhoan(maTaiKhoan) |
| **3** | thoiGianConLai | TIME | Thời gian còn lại trong tài khoản | NOT NULL |
| **4** | loaiKhachHang | NVARCHAR(50) | Loại khách hàng | CHECK (loaiKhachHang = ‘Thuong’ OR loaiKhachHang = ‘VIP’) |
| **5** | diemTichLuy | INT | Điểm tích lũy trong tài khoản |  |
| **6** | maNguoiQuanLy | INT | Người tạo tài khoản khách hàng | FOREIGN KEY (maNguoiQuanLy) REFERENCES NguoiQuanLy(maNguoiQuanLy) |

2.3.1.4 Bảng PhienDangNhap

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| **1** | maPhienDangNhap | INT | Mã phiên đăng nhập | PRIMARY KEY |
| **2** | maKhachHang | INT | Mã khách hàng | FOREIGN KEY (maKhachHang) REFERENCES KhachHang(maKhachHang) |
| **3** | maMayTinh | INT | Mã máy tính | FOREIGN KEY (maMayTinh) REFERENCES MayTinh(maMayTinh) |
| **4** | thoiGianBatDau | TIME | Mốc thời gian bắt đầu phiên đăng nhập |  |
| **5** | thoiGianSuDung | TIME | Tổng thời gian sử dụng |  |
| **6** | thoiGianConLai | TIME | Thời gian còn lại trong tài khoản |  |

2.3.1.5 Bảng UuDai

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| **1** | maUuDai | INT | Mã ưu đãi | PRIMARY KEY |
| **2** | tenUuDai | NVARCHAR(100) | Tên ưu đãi |  |
| **3** | giaTri | DECIMAL(10, 2) | Giá trị ưu đãi |  |
| **4** | thoiGianBatDau | DATETIME | Thời gian bắt đầu | CHECK (‘thoiGianKetThuc’ > ‘thoiGianBatDau’) |
| **5** | thoiGianKetThuc | DATETIME | Thời gian kết thúc |  |
| **6** | dieuKien | NVARCHAR(255) | Điều kiện ưu đãi |  |
| **7** | soLuong | INT | Số lượng ưu đãi |  |
| **8** | tinhTrang | NVARCHAR(50) | Tình trạng uu đãi | CHECK (tinhTrang  = ‘HieuLuc' OR tinhTrang = 'HetHieuLuc') |
| **9** | maNguoiQuanLy | INT | Người tạo tài khoản khách hàng | FOREIGN KEY (maNguoiQuanLy) REFERENCES NguoiQuanLy(maNguoiQuanLy) |

2.3.1.6 Bảng HoaDon

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| **1** | maHoaDon | INT | Mã hóa đơn | PRIMARY KEY |
| **2** | thoiGianTao | DATETIME | Thời gian tạo hóa đơn |  |
| **3** | maKhachHang | INT | Mã khách hàng | FOREIGN KEY (maKhachHang) REFERENCES KhachHang(maKhachHang) |
| **4** | maMayTinh | INT | Mã máy tính | FOREIGN KEY (maMayTinh) REFERENCES MayTinh(maMayTinh) |
| **5** | trangThai | NVARCHAR(50) | Trạng thái hóa đơn | CHECK (trangThai = 'ChuaThanhToan' OR trangThai = 'DaThanhToan') |
| **6** | triGia | DECIMAL(10, 2) | Trị giá hóa đơn |  |
| **7** | maUuDai INT |  | Mã ưu đãi cho hóa đơn | FOREIGN KEY (maUuDai) REFERENCES UuDai(maUuDai)  ON DELETE SET NULL  ON UPDATE CASCADE, |
| **8** | maNguoiQuanLy | INT | Người tạo tài khoản khách hàng | FOREIGN KEY (maNguoiQuanLy) REFERENCES NguoiQuanLy(maNguoiQuanLy) |

2.3.1.7 Bảng DichVu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| **1** | maDichVu | INT | Mã dịch vụ | PRIMARY KEY |
| **2** | tenDichVu | NVARCHAR(100) | Tên dịch vụ |  |
| **3** | loaiDichVu | NVARCHAR(50) | Loại dịch vụ |  |
| **4** | donGia | DECIMAL(10, 2) | Đơn giá dịch vụ |  |
| **5** | soLuong | INT | Số lượng dịch vụ |  |
| **6** | maNguoiQuanLy | INT | Người tạo dịch vụ | FOREIGN KEY (maNguoiQuanLy) REFERENCES NguoiQuanLy(maNguoiQuanLy) |

2.3.1.8 Bảng ChiTietHoaDon

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| **1** | maHoaDon | INT | Mã hóa đơn | PRIMARY KEY (maHoaDon, maDichVu),  FOREIGN KEY (maHoaDon) REFERENCES HoaDon(maHoaDon) |
| **2** | maDichVu | INT | Mã dịch vụ | PRIMARY KEY (maHoaDon, maDichVu), FOREIGN KEY (maDichVu) REFERENCES DichVu(maDichVu) ON DELETE CASCADE |
| **3** | soLuong | INT | Số lượng dịch vụ |  |
| **4** | tongGiaTien | DECIMAL(10, 2) |  |  |

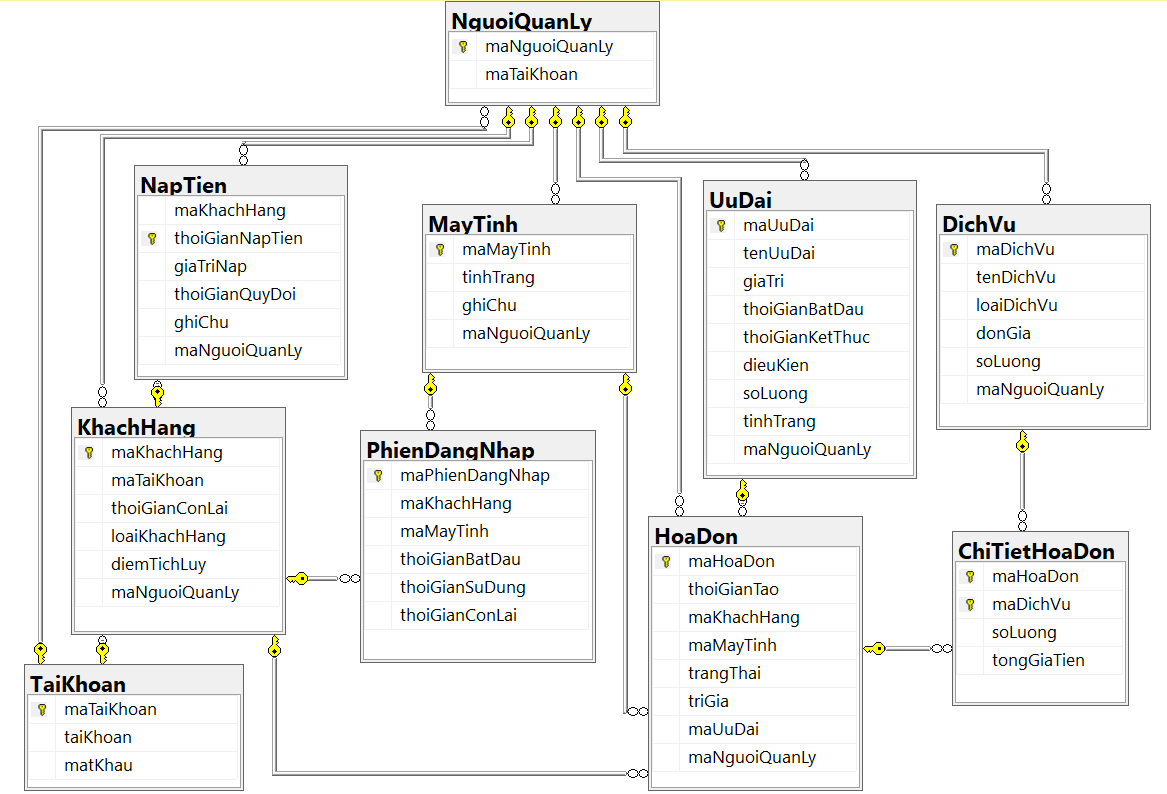
2.3.1.9 Bảng NapTien

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| **1** | maKhachHang | INT | Mã khách hàng | PRIMARY KEY  FOREIGN KEY (maKhachHang) REFERENCES KhachHang(maKhachHang) |
| **2** | thoiGianNapTien | DATETIME | Thời gian nạp tiền | PRIMARY KEY |
| **3** | giaTriNap | DECIMAL(10, 2) | Giá trị nạp tiền |  |
| **4** | thoiGianQuyDoi | TIME | Thời gian quy đổi |  |
| **5** | maNguoiQuanLy | INT | Người nạp tiền | PRIMARY KEY  FOREIGN KEY (maNguoiQuanLy) REFERENCES NguoiQuanLy(maNguoiQuanLy) |

2.3.1.10 Bảng MayTinh

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ràng buộc** |
| **1** | maMayTinh | INT | Mã máy tính | PRIMARY KEY |
| **2** | tinhTrang | NVARCHAR(50) | Tình trạng máy tính | CHECK(tinhTrang  = 'Trong' OR tinhTrang = 'DangSuDung' OR tinhTrang = 'DangBaoTri')) |
| **3** | maNguoiQuanLy | INT | Người tạo tài khoản khách hàng | FOREIGN KEY (maNguoiQuanLy) REFERENCES NguoiQuanLy(maNguoiQuanLy) |

2.3.2 Relation Diagram

****

CHƯƠNG III: CÀI ĐẶT CHỨC NĂNG

3.1 Cài đặt CSDL và các ràng buộc:

-- Bảng TaiKhoan (lớp cha)

CREATE TABLE TaiKhoan (

maTaiKhoan INT PRIMARY KEY,

taiKhoan VARCHAR(50) NOT NULL UNIQUE,

matKhau VARCHAR(50) NOT NULL,

);

-- Bảng NguoiQuanLy

CREATE TABLE NguoiQuanLy (

maNguoiQuanLy INT PRIMARY KEY,

maTaiKhoan INT,

FOREIGN KEY (maTaiKhoan) REFERENCES TaiKhoan(maTaiKhoan )

);

-- Bảng KhachHang

CREATE TABLE KhachHang (

maKhachHang INT PRIMARY KEY,

maTaiKhoan INT,

thoiGianConLai TIME NOT NULL,

loaiKhachHang NVARCHAR(50),

diemTichLuy INT,

maNguoiQuanLy INT,

FOREIGN KEY (maTaiKhoan) REFERENCES TaiKhoan(maTaiKhoan),

FOREIGN KEY (maNguoiQuanLy) REFERENCES NguoiQuanLy(maNguoiQuanLy)

);

-- Bảng MayTinh

CREATE TABLE MayTinh (

maMayTinh INT PRIMARY KEY ,

tinhTrang NVARCHAR(50),

ghiChu NVARCHAR(255),

maNguoiQuanLy INT,

CONSTRAINT tinhTrangHienTaiCuaMayTinh CHECK(tinhTrang = 'Trong' OR tinhTrang = 'DangSuDung' OR tinhTrang = 'DangBaoTri')

FOREIGN KEY (maNguoiQuanLy) REFERENCES NguoiQuanLy(maNguoiQuanLy)

);

-- Bảng PhienDangNhap

CREATE TABLE PhienDangNhap(

maPhienDangNhap INT PRIMARY KEY

maKhachHang INT,

maMayTinh INT,

thoiGianBatDau TIME,

thoiGianSuDung TIME,

thoiGianConLai TIME,

FOREIGN KEY (maKhachHang) REFERENCES KhachHang(maKhachHang),

FOREIGN KEY (maMayTinh) REFERENCES MayTinh(maMayTinh)

);

-- Bảng UuDai

CREATE TABLE UuDai (

maUuDai INT PRIMARY KEY ,

tenUuDai NVARCHAR(100),

giaTri DECIMAL(10, 2),

thoiGianBatDau DATETIME,

thoiGianKetThuc DATETIME,

dieuKien NVARCHAR(255),

soLuong INT,

tinhTrang NVARCHAR(50),

maNguoiQuanLy INT,

FOREIGN KEY (maNguoiQuanLy) REFERENCES NguoiQuanLy(maNguoiQuanLy),

CONSTRAINT chk\_ThoiGian CHECK (thoiGianBatDau < thoiGianKetThuc)

);

-- Bảng DichVu

CREATE TABLE DichVu (

maDichVu INT PRIMARY KEY,

tenDichVu NVARCHAR(100),

loaiDichVu NVARCHAR(50),

donGia DECIMAL(10, 2),

soLuong INT,

maNguoiQuanLy INT,

FOREIGN KEY (maNguoiQuanLy) REFERENCES NguoiQuanLy(maNguoiQuanLy)

);

-- Bảng HoaDon

CREATE TABLE HoaDon (

maHoaDon INT PRIMARY KEY,

thoiGianTao DATETIME,

maKhachHang INT,

trangThai NVARCHAR(50),

triGia DECIMAL(10, 2),

maUuDai INT,

maNguoiQuanLy INT,

FOREIGN KEY (maNguoiQuanLy) REFERENCES NguoiQuanLy(maNguoiQuanLy),

FOREIGN KEY (maKhachHang) REFERENCES KhachHang(maKhachHang),

FOREIGN KEY (maUuDai) REFERENCES UuDai(maUuDai)

ON DELETE SET NULL

ON UPDATE CASCADE,

CONSTRAINT trangThaiHienTaiCuaHoaDon CHECK (trangThai = 'ChuaThanhToan' OR trangThai = 'DaThanhToan')

);

-- Bảng ChiTietHoaDon

CREATE TABLE ChiTietHoaDon (

maHoaDon INT,

maDichVu INT NULL,

soLuong INT,

tongGiaTien DECIMAL(10, 2),

PRIMARY KEY (maHoaDon, maDichVu),

FOREIGN KEY (maHoaDon) REFERENCES HoaDon(maHoaDon),

FOREIGN KEY (maDichVu) REFERENCES DichVu(maDichVu) ON DELETE CASCADE

);

-- Bảng NapTien

CREATE TABLE NapTien (

maKhachHang INT,

thoiGianNapTien DATETIME PRIMARY KEY,

giaTriNap DECIMAL(10, 2),

thoiGianQuyDoi TIME,

ghiChu NVARCHAR(255),

maNguoiQuanLy INT,

FOREIGN KEY (maKhachHang) REFERENCES KhachHang(maKhachHang),

FOREIGN KEY (maNguoiQuanLy) REFERENCES NguoiQuanLy(maNguoiQuanLy)

);

3.2 Cài đặt Trigger:

3.2.1 Trigger trong KhachHang

3.2.1.1 UpdateThoiGianConLai\_PhienDangNhap

Create TRIGGER UpdateThoiGianConLai\_PhienDangNhap

ON KhachHang

AFTER UPDATE

AS

BEGIN

-- Kiểm tra nếu cột thoiGianConLai được cập nhật

IF UPDATE(thoiGianConLai)

BEGIN

-- Cập nhật thoiGianConLai trong bảng KhachHang dựa trên thay đổi ở bảng PhienDangNhap

UPDATE PhienDangNhap

SET PhienDangNhap.thoiGianConLai = inserted.thoiGianConLai

FROM PhienDangNhap

JOIN inserted ON PhienDangNhap.maKhachHang = inserted.maKhachHang

END

END;

3.2.1.2 Update\_LoaiKhachHang

|  |
| --- |
| CREATE TRIGGER Update\_LoaiKhachHang  ON [dbo].[KhachHang]  AFTER UPDATE  AS  BEGIN  -- Khai báo các biến để lưu mã khách hàng, điểm tích lũy và loại khách hàng  DECLARE @maKhachHang INT, @diemTichLuy INT, @loaiKhachHang NVARCHAR(50);  -- Lấy thông tin mã khách hàng và điểm tích lũy từ bản ghi vừa được cập nhật  SELECT @maKhachHang = inserted.maKhachHang, @diemTichLuy = inserted.diemTichLuy  FROM inserted;  -- Xác định loại khách hàng dựa trên điểm tích lũy  IF @diemTichLuy >= 100  BEGIN  SET @loaiKhachHang = 'Vip';  END  ELSE  BEGIN  SET @loaiKhachHang = N'Thường';  END  -- Cập nhật loại khách hàng trong bảng KhachHang  UPDATE KhachHang  SET loaiKhachHang = @loaiKhachHang  WHERE maKhachHang = @maKhachHang;  END; |

3.2.2 Trigger trong PhienDangNhap

3.2.2.1 trg\_UpdateTinhTrangMayTinh\_AfterInsertPhienDangNhap

|  |
| --- |
| -- Trigger này sẽ kích hoạt sau khi có một bản ghi mới được thêm vào bảng PhienDangNhap  -- Khi khách hàng bắt đầu phiên đăng nhập, trạng thái máy tính sẽ được cập nhật thành "ĐangSuDung".  CREATE TRIGGER trg\_UpdateTinhTrangMayTinh\_AfterInsertPhienDangNhap  ON PhienDangNhap  AFTER INSERT  AS  BEGIN  -- Khai báo biến để lưu mã máy tính  DECLARE @maMayTinh INT;  -- Lấy thông tin mã máy tính từ bản ghi vừa được thêm vào  SELECT @maMayTinh = inserted.maMayTinh  FROM inserted;  -- Cập nhật trạng thái máy tính thành "ĐangSuDung"  UPDATE MayTinh  SET tinhTrang = 'DangSuDung'  WHERE maMayTinh = @maMayTinh;  END; |

3.2.2.2 trg\_UpdateTinhTrangMayTinh\_AfterDeletePhienDangNhap

|  |
| --- |
| -- Trigger này sẽ kích hoạt sau khi một bản ghi bị xóa khỏi bảng PhienDangNhap  -- Khi khách hàng kết thúc phiên đăng nhập, trạng thái máy tính sẽ được đặt lại thành "Trống".  CREATE TRIGGER trg\_UpdateTinhTrangMayTinh\_AfterDeletePhienDangNhap  ON PhienDangNhap  AFTER DELETE  AS  BEGIN  -- Khai báo biến để lưu mã máy tính  DECLARE @maMayTinh INT;  -- Lấy thông tin mã máy tính từ bản ghi vừa bị xóa  SELECT @maMayTinh = deleted.maMayTinh  FROM deleted;  -- Cập nhật trạng thái máy tính thành "Trong"  UPDATE MayTinh  SET tinhTrang = 'Trong'  WHERE maMayTinh = @maMayTinh;  END; |

3.2.3 Trigger trong UuDai

3.2.3.1 tr\_CheckUuDai

|  |
| --- |
| -- Kiểm tra xem ưu đãi đã tồn tại chưa, nếu có thì hủy bỏ Insert, chưa có thì insert vào bảng UuDai  CREATE TRIGGER tr\_CheckUuDai  ON UuDai  INSTEAD OF INSERT  AS  BEGIN  IF EXISTS (SELECT 1  FROM UuDai dv  JOIN inserted i ON dv.tenUuDai = i.tenUuDai)  BEGIN  -- Nếu ưu đãi đã tồn tại, hủy bỏ giao dịch  RAISERROR ('Ưu đãi đã tồn tại', 16, 1);  ROLLBACK TRANSACTION;  END  ELSE  BEGIN  -- Thêm ưu đãi nếu không tồn tại  INSERT INTO UuDai (maUuDai,tenUuDai, giaTri, thoiGianBatDau, thoiGianKetThuc, dieuKien, soLuong, tinhTrang, maNguoiQuanLy)  SELECT maUuDai,tenUuDai, giaTri, thoiGianBatDau, thoiGianKetThuc, dieuKien, soLuong, tinhTrang, maNguoiQuanLy FROM inserted;  END  END; |

3.2.4 Trigger trong HoaDon

3.2.4.1 trg\_RecalculateTriGia\_AfterUpdateHoaDon

|  |
| --- |
| CREATE TRIGGER trg\_RecalculateTriGia\_AfterUpdateHoaDon  ON HoaDon  AFTER INSERT, UPDATE  AS  BEGIN  -- Khai báo các biến cần thiết  DECLARE @maHoaDon INT, @maUuDai INT, @tongGiaTien DECIMAL(10, 2), @giaTriUuDai DECIMAL(10, 2);  -- Lấy mã hóa đơn và mã ưu đãi từ bản ghi vừa thêm hoặc cập nhật  SELECT @maHoaDon = maHoaDon, @maUuDai = maUuDai  FROM inserted;  -- Tính tổng giá tiền của các dịch vụ trong ChiTietHoaDon cho hóa đơn hiện tại  SELECT @tongGiaTien = SUM(tongGiaTien)  FROM ChiTietHoaDon  WHERE maHoaDon = @maHoaDon;  -- Kiểm tra nếu maUuDai có trong HoaDon, lấy giaTri của UuDai  IF @maUuDai IS NOT NULL  BEGIN  SELECT @giaTriUuDai = giaTri  FROM UuDai  WHERE maUuDai = @maUuDai;  -- Tính lại trị giá hóa đơn với ưu đãi (giảm giá theo phần trăm)  UPDATE HoaDon  SET triGia = @tongGiaTien \* (1 - @giaTriUuDai / 100)  WHERE maHoaDon = @maHoaDon;  END  ELSE  BEGIN  -- Nếu không có ưu đãi, trị giá hóa đơn chỉ là tổng giá tiền  UPDATE HoaDon  SET triGia = @tongGiaTien  WHERE maHoaDon = @maHoaDon;  END  END; |

3.2.5 Trigger trong DichVu

3.2.5.1 trg\_AfterDelete\_DichVu

|  |
| --- |
| CREATE TRIGGER trg\_AfterDelete\_DichVu  ON DichVu  AFTER DELETE  AS  BEGIN  -- Xóa những hóa đơn không còn tồn tại trong bảng ChiTietHoaDon  DELETE FROM HoaDon  WHERE maHoaDon NOT IN (  SELECT DISTINCT maHoaDon  FROM ChiTietHoaDon  );  END; |

3.2.5.2 tr\_CheckDichVu

-- Kiểm tra xem dịch vụ đã tồn tại chưa, nếu có thì hủy bỏ Insert, chưa có thì insert vào bảng DichVu

CREATE TRIGGER tr\_CheckDichVu

ON DichVu

INSTEAD OF INSERT

AS

BEGIN

IF EXISTS (SELECT 1

FROM DichVu dv

JOIN inserted i ON dv.tenDichVu = i.tenDichVu AND dv.loaiDichVu = i.loaiDichVu)

BEGIN

-- Nếu dịch vụ đã tồn tại, hủy bỏ giao dịch

RAISERROR ('Dịch vụ đã tồn tại', 16, 1);

ROLLBACK TRANSACTION;

END

ELSE

BEGIN

-- Thêm dịch vụ nếu không tồn tại

INSERT INTO DichVu(maDichVu, tenDichVu, loaiDichVu, donGia, soLuong, maNguoiQuanLy)

SELECT maDichVu, tenDichVu, loaiDichVu, donGia, soLuong, maNguoiQuanLy FROM inserted;

END

END;

3.2.6 Trigger trong ChiTietHoaDon

3.2.6.1 trg\_CalculateTongGiaTien\_AfterInsertChiTietHoaDon

|  |
| --- |
| -- Trigger này sẽ kích hoạt sau khi có một bản ghi mới được thêm vào bảng ChiTietHoaDon  -- Khi thêm dịch vụ vào chi tiết hóa đơn, tổng giá tiền của dịch vụ sẽ được tính tự động dựa trên đơn giá và số lượng.  CREATE TRIGGER trg\_CalculateTongGiaTien\_AfterInsertChiTietHoaDon  ON ChiTietHoaDon  AFTER INSERT  AS  BEGIN  -- Khai báo biến để lưu thông tin mã dịch vụ, mã hóa đơn, số lượng và đơn giá  DECLARE @maDichVu INT, @maHoaDon INT, @soLuong INT, @donGia DECIMAL(10,2);  -- Lấy thông tin mã dịch vụ, mã hóa đơn, và số lượng từ bản ghi vừa được thêm vào  SELECT @maDichVu = inserted.maDichVu, @maHoaDon = inserted.maHoaDon, @soLuong = inserted.soLuong  FROM inserted;  -- Lấy đơn giá của dịch vụ từ bảng DichVu  SELECT @donGia = donGia  FROM DichVu  WHERE maDichVu = @maDichVu;  -- Cập nhật tổng giá tiền cho chi tiết hóa đơn bằng cách nhân số lượng với đơn giá  UPDATE ChiTietHoaDon  SET tongGiaTien = @soLuong \* @donGia  WHERE maDichVu = @maDichVu AND maHoaDon = @maHoaDon;  END; |

3.2.7 Trigger trong NapTien

3.2.7.1 trg\_UpdateThoiGianConLaiVaDiemTichLuy\_AfterNapTien

|  |
| --- |
| CREATE TRIGGER trg\_UpdateThoiGianConLaiVaDiemTichLuy\_AfterNapTien  ON NapTien  AFTER INSERT  AS  BEGIN  -- Khai báo các biến để lưu mã khách hàng, thời gian quy đổi và giá trị nạp  DECLARE @maKhachHang INT, @thoiGianQuyDoi TIME, @giaTriNap DECIMAL(10,2);  -- Lấy thông tin mã khách hàng, thời gian quy đổi và giá trị nạp từ bản ghi vừa được thêm vào  SELECT @maKhachHang = inserted.maKhachHang, @thoiGianQuyDoi = inserted.thoiGianQuyDoi, @giaTriNap = inserted.giaTriNap  FROM inserted;  -- Cập nhật thời gian còn lại của khách hàng bằng cách cộng thêm thời gian quy đổi  UPDATE KhachHang  SET thoiGianConLai = DATEADD(MINUTE, DATEDIFF(MINUTE, '00:00:00', @thoiGianQuyDoi), thoiGianConLai)  WHERE maKhachHang = @maKhachHang;  -- Cộng thêm điểm tích lũy cho khách hàng dựa trên giá trị nạp (mỗi 1 tiếng được 1 điểm)  UPDATE KhachHang  SET diemTichLuy = diemTichLuy + FLOOR(@giaTriNap / 5000)  WHERE maKhachHang = @maKhachHang;  END; |

3.3 Cài đặt View:

|  |
| --- |
| -- View Danh sách máy tính (sau khi nhấn vào "quản lý máy" từ "Màn hình chính" của Quản lý)  CREATE VIEW QuanLyMayTinhView AS  SELECT mt.maMayTinh,  pdn.maKhachHang,  mt.tinhTrang,  COALESCE(pdn.thoiGianBatDau, '00:00:00') AS thoiGianBatDau,  COALESCE(pdn.thoiGianSuDung, '00:00:00') AS thoiGianSuDung,  COALESCE(kh.thoiGianConLai, '00:00:00') AS thoiGianConLai -- Lấy thoiGianConLai từ KhachHang  FROM MayTinh mt  LEFT JOIN PhienDangNhap pdn ON mt.maMayTinh = pdn.maMayTinh  LEFT JOIN KhachHang kh ON pdn.maKhachHang = kh.maKhachHang; -- Thêm JOIN với bảng KhachHang |
| -- View Danh sách dịch vụ chưa thanh toán (sau khi nhấn vào "quản lý gọi dịch vụ" từ "Màn hình chính" của Quản lý)  CREATE VIEW DichVuChuaThanhToanView AS  SELECT hd.maHoaDon,  hd.thoiGianTao,  hd.trangThai,  tk.taiKhoan AS tenTaiKhoan,  hd.maUuDai,  hd.triGia  FROM HoaDon hd  JOIN KhachHang kh ON hd.maKhachHang = kh.maKhachHang  JOIN TaiKhoan tk ON kh.maTaiKhoan = tk.maTaiKhoan  WHERE hd.trangThai = 'ChuaThanhToan'; |
| -- View Danh sách dịch vụ chưa thanh toán (sau khi nhấn vào "quản lý gọi dịch vụ" từ "Màn hình chính" của Quản lý)  CREATE VIEW DichVuDaThanhToanView AS  SELECT hd.maHoaDon,  hd.thoiGianTao,  hd.trangThai,  tk.taiKhoan AS tenTaiKhoan,  hd.maUuDai,  hd.triGia  FROM HoaDon hd  JOIN KhachHang kh ON hd.maKhachHang = kh.maKhachHang  JOIN TaiKhoan tk ON kh.maTaiKhoan = tk.maTaiKhoan  WHERE hd.trangThai = 'DaThanhToan'; |
| -- View xem dịch vụ "Thức uống"  CREATE VIEW DichVuThucUongView AS  SELECT maDichVu, tenDichVu, donGia, loaiDichVu, soLuong  FROM DichVu  WHERE loaiDichVu = N'Thức uống'; |
| -- View xem dịch vụ "Thẻ cào"  CREATE VIEW DichVuTheCaoView AS  SELECT maDichVu, tenDichVu, donGia, loaiDichVu, soLuong  FROM DichVu  WHERE loaiDichVu = N'Thẻ cào'; |
| -- View xem tất cả dịch vụ  CREATE VIEW DichVuView AS  SELECT maDichVu, tenDichVu, loaiDichVu, donGia, soLuong  FROM DichVu; |
| -- View xem danh sách khách hàng  CREATE VIEW QuanLyKhachHangView AS  SELECT  maKhachHang,  maTaiKhoan,  thoiGianConLai,  loaiKhachHang,  diemTichLuy  FROM KhachHang; |
| -- View xem đanh sách máy tính  CREATE VIEW QuanLyMayTinhView AS  SELECT mt.maMayTinh,  pdn.maKhachHang,  mt.tinhTrang,  COALESCE(pdn.thoiGianBatDau, '00:00:00') AS thoiGianBatDau,  COALESCE(pdn.thoiGianSuDung, '00:00:00') AS thoiGianSuDung,  COALESCE(kh.thoiGianConLai, '00:00:00') AS thoiGianConLai -- Lấy thoiGianConLai từ KhachHang  FROM MayTinh mt  LEFT JOIN PhienDangNhap pdn ON mt.maMayTinh = pdn.maMayTinh  LEFT JOIN KhachHang kh ON pdn.maKhachHang = kh.maKhachHang; -- Thêm JOIN với bảng KhachHang |
| -- View xem lịch sử nạp tiền  CREATE VIEW ViewNapTien AS  SELECT  maKhachHang,  thoiGianNapTien,  giaTriNap,  thoiGianQuyDoi  FROM NapTien; |
| -- View xem thông tin phiên đăng nhập  CREATE VIEW ViewPhienDangNhap AS  SELECT \*  FROM PhienDangNhap; |
| -- View xem danh sách ưu đãi dành cho khách hàng thường  CREATE VIEW ViewUuDaiThuong AS  SELECT  maUuDai,  tenUuDai,  giaTri,  thoiGianBatDau,  thoiGianKetThuc,  dieuKien,  soLuong,  tinhTrang  FROM UuDai  WHERE dieuKien = N'Tất cả khách hàng'; |
| -- View xem danh sách ưu đãi dành cho khách hàng vip  CREATE VIEW ViewUuDaiVIP AS  SELECT  maUuDai,  tenUuDai,  giaTri,  thoiGianBatDau,  thoiGianKetThuc,  dieuKien,  soLuong,  tinhTrang  FROM UuDai; |
| CREATE VIEW ChiTietHoaDonView AS  SELECT  cthd.maHoaDon,  cthd.maDichVu,  dv.tenDichVu,  cthd.soLuong,  cthd.tongGiaTien  FROM  ChiTietHoaDon cthd  JOIN  DichVu dv ON cthd.maDichVu = dv.maDichVu; |

3.4 Cài đặt Function

3.4.1 dbo.LayPhienDangNhap

**-- Lấy thông tin về phiên đăng nhập với đầu vào là maKhachHang**

|  |
| --- |
| CREATE FUNCTION dbo.LayPhienDangNhap (@maKhachHang INT)  RETURNS TABLE  AS  RETURN  (  SELECT maKhachHang, maMayTinh, thoiGianBatDau, thoiGianSuDung, thoiGianConLai  FROM ViewPhienDangNhap  WHERE maKhachHang = @maKhachHang  ); |

**-- Code c#**

|  |
| --- |
| public ClassPhienDangNhap LapPhienDangNhap(int maTaiKhoan)  {  try  {  dBConnection.openConnectionAdmin();  SqlCommand cmd = new SqlCommand("SELECT \* FROM dbo.LayPhienDangNhap(@maTaiKhoan)", dBConnection.getConnectionAdmin);  cmd.Parameters.AddWithValue("@maTaiKhoan", maTaiKhoan);  SqlDataReader reader = cmd.ExecuteReader();  reader.Read();  ClassPhienDangNhap classPhienDangNhap = new ClassPhienDangNhap  (Convert.ToInt32(reader["maKhachHang"]), Convert.ToInt32(reader["maMayTinh"]),  TimeSpan.Parse(reader["thoiGianBatDau"].ToString()), TimeSpan.Parse(reader["thoiGianSuDung"].ToString()),  TimeSpan.Parse(reader["thoiGianConLai"].ToString())  );  return classPhienDangNhap;  }  catch (Exception ex)  {  MessageBox.Show("Đã xảy ra lỗi: " + ex.Message + "\n" + ex.StackTrace, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  return null;  }  finally  {  dBConnection.closeConnectionAdmin();  }  } |

3.4.2 fn\_LayMaKhachHang

**-- Lấy mã khách hàng dựa vào mã tài khoản**

|  |
| --- |
| CREATE FUNCTION fn\_LayMaKhachHang(@maTaiKhoan INT)  RETURNS INT  AS  BEGIN  DECLARE @maKhachHang INT;  SELECT @maKhachHang = maKhachHang  FROM KhachHang  WHERE maTaiKhoan = @maTaiKhoan;  RETURN @maKhachHang;  END; |

**-- Code c#**

|  |
| --- |
| public int LayMaKhachHang(int maTaiKhoan)  {  try  {  object maKhachHang = -1;  dBConnection.openConnectionAdmin();  using (SqlCommand cmd = new SqlCommand("SELECT dbo.fn\_LayMaKhachHang(@maTaiKhoan)", dBConnection.getConnectionAdmin))  {  cmd.Parameters.AddWithValue("@maTaiKhoan", maTaiKhoan);  maKhachHang = cmd.ExecuteScalar();  }  return Convert.ToInt32(maKhachHang);  }  catch (Exception ex)  {  MessageBox.Show("Thất bại (LayMaKhachHang): " + ex);  return -1;  }  finally  {  dBConnection.closeConnectionAdmin();  }  } |

3.4.3 GetTenTaiKhoan

**-- Lấy tên tài khoản dựa vào mã tài khoản**

|  |
| --- |
| CREATE FUNCTION GetTenTaiKhoan (  @maTaiKhoan INT  )  RETURNS NVARCHAR(255)  AS  BEGIN  DECLARE @tenTaiKhoan NVARCHAR(255);    -- Truy vấn tên tài khoản dựa trên mã tài khoản  SELECT @tenTaiKhoan = taiKhoan  FROM TaiKhoan  WHERE maTaiKhoan = @maTaiKhoan;    -- Trả về tên tài khoản  RETURN @tenTaiKhoan;  END; |

**-- Code c#**

|  |
| --- |
| public string getTenTaiKhoan(int maTaiKhoan)  {  try  {  SqlCommand cmd = new SqlCommand("SELECT dbo.GetTenTaiKhoan(@maTaiKhoan)", dBConnection.getConnectionAdmin);  cmd.Parameters.AddWithValue("@maTaiKhoan", maTaiKhoan);  dBConnection.openConnectionAdmin();  object result = cmd.ExecuteScalar();  if (result != null)  {  return result.ToString();  }  else  {  MessageBox.Show("Không tìm thấy tài khoản với mã này.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  return null;  }  }  catch (Exception ex)  {  MessageBox.Show("Đã xảy ra lỗi: " + ex.Message + "\n" + ex.StackTrace, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  return null;  }  finally  {  dBConnection.closeConnectionAdmin();  }  } |

3.4.4 KiemTraKhachHangDangNhap

**-- Kiểm tra tài khoản và mật khẩu nhập khách hàng vào có tồn tại và hợp lệ không**

|  |
| --- |
| CREATE FUNCTION KiemTraKhachHangDangNhap (  @taiKhoan VARCHAR(50),  @matKhau VARCHAR(50)  )  RETURNS INT  AS  BEGIN  DECLARE @maTaiKhoan INT;  -- Kiểm tra tài khoản và mật khẩu, join với KhachHang  SELECT @maTaiKhoan = tk.maTaiKhoan  FROM TaiKhoan tk  JOIN KhachHang kh ON tk.maTaiKhoan = kh.maTaiKhoan  WHERE tk.taiKhoan = @taiKhoan AND tk.matKhau = @matKhau;  -- Trả về mã tài khoản nếu hợp lệ, ngược lại trả về NULL  RETURN @maTaiKhoan;  END; |

**-- Code c#**

|  |
| --- |
| public int KiemTraKhachHangDangNhap(ClassTaiKhoan classTaiKhoan)  {    string SQLFunctionName = "KiemTraKhachHangDangNhap";  return KiemTraDangNhap(classTaiKhoan, SQLFunctionName); ;  }  public int KiemTraDangNhap(ClassTaiKhoan classTaiKhoan, string SQLFunctionName)  {  try  {  dBConnection.openConnection();  SqlCommand cmd = new SqlCommand($"SELECT dbo.{SQLFunctionName}(@taiKhoan, @matKhau)", dBConnection.getConnection);    // Thêm tham số cho hàm  cmd.Parameters.AddWithValue("@taiKhoan", classTaiKhoan.TenTaiKhoan);  cmd.Parameters.AddWithValue("@matKhau", classTaiKhoan.MatKhau);  // Thực thi câu lệnh và lấy giá trị trả về  object result = cmd.ExecuteScalar();  if (result != null && result != DBNull.Value)  {  // Chuyển đổi kết quả trả về thành int (maTaiKhoan)  return Convert.ToInt32(result.ToString());  }  else  {  return -1;  }  }  catch (Exception ex)  {  MessageBox.Show($"Lỗi: {ex.Message}");  return -1; // Trả về -1 nếu có lỗi xảy ra  }  finally  {  dBConnection.closeConnection();  }  } |

3.4.5 KiemTraNguoiQuanLyDangNhap

**-- Kiểm tra tài khoản và mật khẩu người quản lý nhập vào có tồn tại và hợp lệ không**

|  |
| --- |
| CREATE FUNCTION KiemTraNguoiQuanLyDangNhap (  @taiKhoan VARCHAR(50),  @matKhau VARCHAR(50)  )  RETURNS INT  AS  BEGIN  DECLARE @maNguoiQuanLy INT;  -- Kiểm tra tài khoản và mật khẩu, join với NguoiQuanLy  SELECT @maNguoiQuanLy = ql.maNguoiQuanLy  FROM TaiKhoan tk  JOIN NguoiQuanLy ql ON tk.maTaiKhoan = ql.maTaiKhoan  WHERE tk.taiKhoan = @taiKhoan AND tk.matKhau = @matKhau;  -- Trả về mã tài khoản nếu hợp lệ, ngược lại trả về NULL  RETURN @maNguoiQuanLy;  END; |

**-- Code c#**

|  |
| --- |
| public int KiemTraNguoiQuanLyDangNhap(ClassTaiKhoan classTaiKhoan)  {    string SQLFunctionName = "KiemTraNguoiQuanLyDangNhap";  return KiemTraDangNhap(classTaiKhoan, SQLFunctionName);  } |
| public int KiemTraDangNhap(ClassTaiKhoan classTaiKhoan, string SQLFunctionName)  {  try  {  dBConnection.openConnection();  SqlCommand cmd = new SqlCommand($"SELECT dbo.{SQLFunctionName}(@taiKhoan, @matKhau)", dBConnection.getConnection);    // Thêm tham số cho hàm  cmd.Parameters.AddWithValue("@taiKhoan", classTaiKhoan.TenTaiKhoan);  cmd.Parameters.AddWithValue("@matKhau", classTaiKhoan.MatKhau);  // Thực thi câu lệnh và lấy giá trị trả về  object result = cmd.ExecuteScalar();  if (result != null && result != DBNull.Value)  {  // Chuyển đổi kết quả trả về thành int (maTaiKhoan)  return Convert.ToInt32(result.ToString());  }  else  {  return -1;  }  }  catch (Exception ex)  {  MessageBox.Show($"Lỗi: {ex.Message}");  return -1; // Trả về -1 nếu có lỗi xảy ra  }  finally  {  dBConnection.closeConnection();  }  } |

3.4.6 LayLoaiKhachHang

**-- Lấy loại khách hàng của khách hàng với đầu vào là mã khách hàng**

|  |
| --- |
| CREATE FUNCTION LayLoaiKhachHang (@maKhachHang INT)  RETURNS NVARCHAR(50)  AS  BEGIN  DECLARE @loaiKhachHang NVARCHAR(50);  -- Truy vấn loại khách hàng dựa vào mã khách hàng  SELECT @loaiKhachHang = loaiKhachHang  FROM KhachHang  WHERE maKhachHang = @maKhachHang;  RETURN @loaiKhachHang;  END; |

**-- Code c#**

|  |
| --- |
| public string GetLoaiKhachHang(int maKhachHang)  {  try  {  dBConnection.openConnection();  // Tạo một đối tượng SqlCommand để gọi hàm SQL  string SQL = "SELECT dbo.LayLoaiKhachHang(@maKhachHang)";  SqlCommand cmd = new SqlCommand(SQL, dBConnection.getConnection);  // Thêm tham số cho hàm  cmd.Parameters.AddWithValue("@maKhachHang", maKhachHang);  // Thực thi câu lệnh và lấy giá trị trả về  object result = cmd.ExecuteScalar();  return result.ToString();  }  catch (Exception ex)  {  MessageBox.Show($"Lỗi: {ex.Message}");  return "";  }  finally  {  dBConnection.closeConnection();  }  } |

3.4.7 fn\_TinhDoanhThuDichVuTheoNgay

|  |
| --- |
| CREATE FUNCTION fn\_TinhDoanhThuDichVuTheoNgay (@ngayTrongTuan INT, @loaiDichVu NVARCHAR(50))  RETURNS DECIMAL(18, 2)  AS  BEGIN  DECLARE @tongDoanhThu DECIMAL(18, 2);  -- Tính tổng doanh thu từ loại dịch vụ được truyền vào theo ngày trong tuần  SELECT @tongDoanhThu = SUM(cthd.tongGiaTien)  FROM ChiTietHoaDon cthd  JOIN DichVu dv ON cthd.maDichVu = dv.maDichVu  JOIN HoaDon hd ON cthd.maHoaDon = hd.maHoaDon  WHERE dv.loaiDichVu = @loaiDichVu  AND hd.trangThai = 'DaThanhToan'  AND DATEPART(WEEKDAY, hd.thoiGianTao) = @ngayTrongTuan;  RETURN ISNULL(@tongDoanhThu, 0);  END; |

3.4.8 fn\_TinhDoanhThuNapTienTheoNgay

|  |
| --- |
| CREATE FUNCTION fn\_TinhDoanhThuNapTienTheoNgay (@ngayTrongTuan INT)  RETURNS DECIMAL(18, 2)  AS  BEGIN  DECLARE @tongDoanhThu DECIMAL(18, 2);  -- Tính tổng doanh thu từ nạp tiền theo ngày trong tuần  SELECT @tongDoanhThu = SUM(giaTriNap)  FROM NapTien  WHERE DATEPART(WEEKDAY, thoiGianNapTien) = @ngayTrongTuan;  RETURN ISNULL(@tongDoanhThu, 0);  END; |

3.5 Cài đặt Procedure

|  |
| --- |
| CREATE PROCEDURE sp\_ThongKeDoanhThuTheoNgayTrongTuan  AS  BEGIN  -- Bắt đầu giao dịch  BEGIN TRANSACTION;  BEGIN TRY  DECLARE @ngayTrongTuan INT = 1;  DECLARE @doanhThuDoAn DECIMAL(18, 2);  DECLARE @doanhThuThucUong DECIMAL(18, 2);  DECLARE @doanhThuTheCao DECIMAL(18, 2);  DECLARE @doanhThuNapTien DECIMAL(18, 2);  -- Tạo bảng tạm để lưu kết quả  CREATE TABLE #ThongKeDoanhThu (  NgayTrongTuan NVARCHAR(50),  DoAn DECIMAL(18, 2),  ThucUong DECIMAL(18, 2),  TheCao DECIMAL(18, 2),  NapTien DECIMAL(18, 2)  );  -- Vòng lặp từ Thứ Hai (1) đến Chủ Nhật (7)  WHILE @ngayTrongTuan <= 7  BEGIN  -- Gọi hàm tính doanh thu cho từng loại dịch vụ  SET @doanhThuDoAn = dbo.fn\_TinhDoanhThuDichVuTheoNgay(@ngayTrongTuan, N'Đồ ăn');  SET @doanhThuThucUong = dbo.fn\_TinhDoanhThuDichVuTheoNgay(@ngayTrongTuan, N'Thức uống');  SET @doanhThuTheCao = dbo.fn\_TinhDoanhThuDichVuTheoNgay(@ngayTrongTuan, N'Thẻ cào');  -- Gọi hàm tính doanh thu từ nạp tiền  SET @doanhThuNapTien = dbo.fn\_TinhDoanhThuNapTienTheoNgay(@ngayTrongTuan);  -- Chèn kết quả vào bảng tạm  INSERT INTO #ThongKeDoanhThu (NgayTrongTuan, DoAn, ThucUong, TheCao, NapTien)  VALUES (  CASE @ngayTrongTuan  WHEN 1 THEN N'Thứ Hai'  WHEN 2 THEN N'Thứ Ba'  WHEN 3 THEN N'Thứ Tư'  WHEN 4 THEN N'Thứ Năm'  WHEN 5 THEN N'Thứ Sáu'  WHEN 6 THEN N'Thứ Bảy'  WHEN 7 THEN N'Chủ Nhật'  END,  ISNULL(@doanhThuDoAn, 0),  ISNULL(@doanhThuThucUong, 0),  ISNULL(@doanhThuTheCao, 0),  ISNULL(@doanhThuNapTien, 0)  );  -- Tăng biến đếm ngày  SET @ngayTrongTuan = @ngayTrongTuan + 1;  END  -- Commit giao dịch nếu không có lỗi  COMMIT TRANSACTION;  -- Trả về kết quả  SELECT \* FROM #ThongKeDoanhThu;  -- Xóa bảng tạm  DROP TABLE #ThongKeDoanhThu;  END TRY  BEGIN CATCH  -- Rollback giao dịch nếu có lỗi  ROLLBACK TRANSACTION;  -- In ra thông báo lỗi  DECLARE @ErrorMessage NVARCHAR(4000) = ERROR\_MESSAGE();  RAISERROR(@ErrorMessage, 16, 1);  END CATCH  END; |

|  |
| --- |
| public DataTable GetThongKeDoanhThu()  {  try  {  dBConnection.openConnection();  DataTable dataTable = new DataTable();  SqlCommand cmd = new SqlCommand("sp\_ThongKeDoanhThuTheoNgayTrongTuan", dBConnection.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  adapter.Fill(dataTable);  return dataTable;  }  catch (Exception exc)  {  MessageBox.Show("that bai (GetThongKeDoanhThu)" + exc);  return null;  }  finally  {  dBConnection.closeConnection();  }  } |

**-- Truy thông tin chi tiết của một hóa đơn dựa vào mã hóa đơn**

|  |
| --- |
| CREATE PROCEDURE sp\_GetChiTietHoaDon  @maHoaDon INT  AS  BEGIN  SELECT \*  FROM ChiTietHoaDonView  WHERE maHoaDon = @maHoaDon;  END; |

**-- Code c#**

|  |
| --- |
| public DataTable LayChiTietHoaDon(int maHoaDon)  {  try  {  dBConnection.openConnection();  SqlCommand cmd = new SqlCommand("sp\_GetChiTietHoaDon", dBConnection.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@maHoaDon", maHoaDon);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dataTable = new DataTable();  adapter.Fill(dataTable);  return dataTable;  }  catch (Exception ex)  {  MessageBox.Show("Đã xảy ra lỗi: LayChiTietHoaDon " + ex.Message);  return null;  }  finally  {  dBConnection.closeConnection();  }  } |

**-- Kiểm tra thời gian còn lại trong tài khoản trước khi khách hàng đăng nhập**

|  |
| --- |
| CREATE PROCEDURE sp\_KiemTraThoiGianConLai  @maKhachHang INT,  @result BIT OUTPUT -- Tham số OUTPUT kiểu BIT để trả về kết quả  AS  BEGIN  DECLARE @thoiGianConLai TIME;  -- Lấy thời gian còn lại từ bảng KhachHang  SELECT @thoiGianConLai = thoiGianConLai  FROM KhachHang  WHERE maKhachHang = @maKhachHang;  -- Kiểm tra nếu thoiGianConLai > '00:00:00' thì trả về 1 (true), ngược lại trả về 0 (false)  IF @thoiGianConLai > '00:00:00'  SET @result = 1;  ELSE  SET @result = 0;  END; |

**-- Code c#**

|  |
| --- |
| public bool KiemTraThoiGianConLai(int maKhachHang)  {  bool result = false;  SqlConnection connection = null;  SqlCommand cmd = null;  try  {  connection = dBConnection.getConnectionAdmin;  dBConnection.openConnectionAdmin();  cmd = new SqlCommand("sp\_KiemTraThoiGianConLai", connection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@maKhachHang", maKhachHang);  SqlParameter outputParam = new SqlParameter("@result", SqlDbType.Bit)  {  Direction = ParameterDirection.Output  };  cmd.Parameters.Add(outputParam);  cmd.ExecuteNonQuery();  result = Convert.ToBoolean(outputParam.Value);  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi kiểm tra thời gian còn lại: " + ex.Message);  }  finally  {  if (cmd != null)  {  cmd.Dispose();  }  dBConnection.closeConnectionAdmin();  }  return result;  } |

3.5 Cài đặt Procedure

3.5.1 sp\_AddQuanLy

|  |
| --- |
| CREATE PROCEDURE sp\_AddQuanLy  @taiKhoan VARCHAR(50),  @matKhau VARCHAR(50)  AS  BEGIN  -- Kiểm tra xem tài khoản đã tồn tại hay chưa  IF EXISTS (SELECT 1 FROM TaiKhoan WHERE taiKhoan = @taiKhoan)  BEGIN  -- Nếu tài khoản đã tồn tại, sử dụng RAISERROR để trả về lỗi cho C#  RAISERROR('Tài khoản đã tồn tại', 16, 1);  RETURN;  END  -- Nếu tài khoản chưa tồn tại, tiến hành thêm mới  BEGIN  -- Bước 1: Tìm giá trị maTaiKhoan lớn nhất hiện tại và cộng thêm 1  DECLARE @maTaiKhoan INT;  SELECT @maTaiKhoan = ISNULL(MAX(maTaiKhoan), 0) + 1 FROM TaiKhoan;  -- Bước 2: Thêm vào bảng TaiKhoan  INSERT INTO TaiKhoan (maTaiKhoan, taiKhoan, matKhau)  VALUES (@maTaiKhoan, @taiKhoan, @matKhau);  -- Bước 3: Tìm giá trị maNguoiQuanLy lớn nhất hiện tại và cộng thêm 1  DECLARE @maNguoiQuanLy INT;  SELECT @maNguoiQuanLy = ISNULL(MAX(maNguoiQuanLy), 0) + 1 FROM NguoiQuanLy;  -- Bước 4: Thêm khách hàng mới vào bảng NguoiQuanLy  INSERT INTO NguoiQuanLy (maNguoiQuanLy, maTaiKhoan)  VALUES (@maNguoiQuanLy, @maTaiKhoan);  -- Thông báo thành công  PRINT 'Thêm quản lý thành công';  END  END; |

|  |
| --- |
| public string AddQuanLy(string taiKhoan, string matKhau)  {  try  {  dBConnection.openConnection();  SqlCommand cmd = new SqlCommand("sp\_AddQuanLy", dBConnection.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@taiKhoan", taiKhoan);  cmd.Parameters.AddWithValue("@matKhau", matKhau);  dBConnection.openConnection();  // Thực thi stored procedure  cmd.ExecuteNonQuery();  return "Thêm quản lý thành công";  }  catch (SqlException ex)  {  return "Lỗi: " + ex.Message;  }  finally  {  dBConnection.closeConnection();  }  } |

3.5.2 DoiMatKhau

**– Đổi mật khẩu khách hàng**

|  |
| --- |
| CREATE PROC DoiMatKhau(  @taiKhoan NVARCHAR(50),  @matKhau NVARCHAR(50),  @matKhauMoi NVARCHAR(50)  )  AS  BEGIN  -- Kiểm tra mật khẩu nhập đã đúng chưa  IF NOT EXISTS (SELECT 1 FROM TaiKhoan WHERE taiKhoan = @taiKhoan AND matKhau = @matKhau)  BEGIN  -- Nếu sai mật khẩu, sử dụng RAISERROR để trả về lỗi cho C#  RAISERROR('Sai mật khẩu', 16, 1);  RETURN;  END  -- Thực hiện cập nhật  UPDATE TaiKhoan  SET matKhau = @matKhauMoi  WHERE taiKhoan = @taiKhoan;  END; |

**-- Code c#**

|  |
| --- |
| public void DoiMatKhau(ClassTaiKhoan classTaiKhoan, string matKhauMoi)  {    try  {  dBConnection.openConnection();  // Tạo một đối tượng SqlCommand để gọi hàm SQL  using (SqlCommand command = new SqlCommand("DoiMatKhau", dBConnection.getConnection))  {  command.CommandType = CommandType.StoredProcedure;  // Thêm các tham số vào command  command.Parameters.Add(new SqlParameter("@taiKhoan", classTaiKhoan.TenTaiKhoan));  command.Parameters.Add(new SqlParameter("@matKhau", classTaiKhoan.MatKhau));  command.Parameters.Add(new SqlParameter("@matKhauMoi", matKhauMoi));  // Thực thi thủ tục  command.ExecuteNonQuery();  MessageBox.Show("Đổi mật khẩu thành công!");  }  }  catch (Exception ex)  {  MessageBox.Show($"Lỗi: {ex.Message}");  }  finally  {  dBConnection.closeConnection();  }    } |

3.5.3 NapTienProcedure

**-- Nạp tiền vào tài khoản Khách hàng**

|  |
| --- |
| CREATE PROCEDURE NapTienProcedure  @maKhachHang INT,  @thoiGianNapTien DateTime,  @giaTriNap decimal(10, 2),  @thoiGianQuyDoi time(7),  @maNguoiQuanLy INT  As  BEGIN  -- Thêm bản ghi nạp tiền vào bảng NapTien  INSERT INTO NapTien (maKhachHang, thoiGianNapTien, giaTriNap, thoiGianQuyDoi, maNguoiQuanLy)  VALUES (@maKhachHang, @thoiGianNapTien, @giaTriNap, @thoiGianQuyDoi, @maNguoiQuanLy);  END; |

**-- Code c#**

|  |
| --- |
| public void napTien(ClassNapTien classNapTien, int maNguoiQuanLy)  {  try  {  dBConnection.openConnection();  SqlCommand cmd = new SqlCommand("NapTienProcedure", dBConnection.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@maKhachHang", classNapTien.MaKhachHang);  cmd.Parameters.AddWithValue("@thoiGianNapTien", classNapTien.ThoiGianNapTien);  cmd.Parameters.AddWithValue("@giaTriNap", classNapTien.GiaTriNap);  cmd.Parameters.AddWithValue("@thoiGianQuyDoi", classNapTien.ThoiGianQuyDoi);  cmd.Parameters.AddWithValue("@maNguoiQuanLy", maNguoiQuanLy);  cmd.ExecuteNonQuery();  MessageBox.Show("Nạp tiền thành công");  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi nạp tiền vào tài khoản khách hàng: " + ex.Message);  }  } |

3.5.4 sp\_AddKhachHang

**-- Thêm tài khoản khách hàng**

|  |
| --- |
| CREATE PROCEDURE sp\_AddKhachHang  @taiKhoan VARCHAR(50),  @matKhau VARCHAR(50),  @maNguoiQuanLy INT  AS  BEGIN  -- Kiểm tra xem tài khoản đã tồn tại hay chưa  IF EXISTS (SELECT 1 FROM TaiKhoan WHERE taiKhoan = @taiKhoan)  BEGIN  -- Nếu tài khoản đã tồn tại, sử dụng RAISERROR để trả về lỗi cho C#  RAISERROR('Tài khoản đã tồn tại', 16, 1);  RETURN;  END  -- Nếu tài khoản chưa tồn tại, tiến hành thêm mới  BEGIN  -- Bước 1: Tìm giá trị maTaiKhoan lớn nhất hiện tại và cộng thêm 1  DECLARE @maTaiKhoan INT;  SELECT @maTaiKhoan = ISNULL(MAX(maTaiKhoan), 0) + 1 FROM TaiKhoan;  -- Bước 2: Thêm vào bảng TaiKhoan  INSERT INTO TaiKhoan (maTaiKhoan, taiKhoan, matKhau)  VALUES (@maTaiKhoan, @taiKhoan, @matKhau);  -- Bước 3: Tìm giá trị maKhachHang lớn nhất hiện tại và cộng thêm 1  DECLARE @maKhachHang INT;  SELECT @maKhachHang = ISNULL(MAX(maKhachHang), 0) + 1 FROM KhachHang;  -- Bước 4: Thêm khách hàng mới vào bảng KhachHang  INSERT INTO KhachHang (maKhachHang, maTaiKhoan, thoiGianConLai, loaiKhachHang, diemTichLuy, maNguoiQuanLy)  VALUES (@maKhachHang, @maTaiKhoan, '00:00:00', N'Thường', 0, @maNguoiQuanLy);  -- Thông báo thành công  PRINT 'Thêm khách hàng thành công';  END  END; |

**-- Code c#**

|  |
| --- |
| public string AddKhachHang(string taiKhoan, string matKhau, int maTaiKhoanNguoiQuanLy)  {  try  {  dBConnection.openConnection();  SqlCommand cmd = new SqlCommand("sp\_AddKhachHang", dBConnection.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@taiKhoan", taiKhoan);  cmd.Parameters.AddWithValue("@matKhau", matKhau);  cmd.Parameters.AddWithValue("@maNguoiQuanLy", maTaiKhoanNguoiQuanLy);  dBConnection.openConnection();  // Thực thi stored procedure  cmd.ExecuteNonQuery();  return "Thêm khách hàng thành công";  }  catch (SqlException ex)  {  return "Lỗi: " + ex.Message;  }  finally  {  dBConnection.closeConnection();  }  } |

3.5.5 sp\_CapNhatTinhTrangHoaDon

**-- Cập nhật hóa đơn thành đã thanh toán**

|  |
| --- |
| CREATE PROCEDURE sp\_CapNhatTinhTrangHoaDon  @maHoaDon INT  AS  BEGIN  DECLARE @maUuDai INT;  -- Kiểm tra nếu hóa đơn tồn tại và có trạng thái 'ChuaThanhToan'  IF EXISTS (SELECT 1 FROM HoaDon WHERE maHoaDon = @maHoaDon AND trangThai = 'ChuaThanhToan')  BEGIN  -- Lấy mã ưu đãi từ hóa đơn  SELECT @maUuDai = maUuDai  FROM HoaDon  WHERE maHoaDon = @maHoaDon;  -- Cập nhật trạng thái hóa đơn thành 'DaThanhToan'  UPDATE HoaDon  SET trangThai = 'DaThanhToan'  WHERE maHoaDon = @maHoaDon;  -- Nếu hóa đơn có mã ưu đãi, trừ số lượng ưu đãi đi một đơn vị  IF @maUuDai IS NOT NULL  BEGIN  UPDATE UuDai  SET soLuong = soLuong - 1  WHERE maUuDai = @maUuDai AND soLuong > 0;  END;  -- Thông báo thành công  PRINT 'Hóa đơn đã được Thanh Toán';  END  ELSE  BEGIN  -- Thông báo nếu hóa đơn không tồn tại hoặc không ở trạng thái 'ChuaThanhToan'  PRINT 'Hóa đơn không tìm thấy.';  END  END; |

**-- Code c#**

|  |
| --- |
| public void CapNhatTinhTrangHoaDon(int maHoaDon)  {  try  {  dBConnection.openConnection();  using (SqlCommand cmd = new SqlCommand("sp\_CapNhatTinhTrangHoaDon", dBConnection.getConnection))  {  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@maHoaDon", maHoaDon);  cmd.ExecuteNonQuery();  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi CapNhatTinhTrangHoaDon " + ex.Message);  }  } |

3.5.6 sp\_ChuyenDoiMaKhachHangSangMaTaiKhoan

**-- Chuyển mã khách hàng sang mã tài khoản**

|  |
| --- |
| CREATE PROCEDURE sp\_ChuyenDoiMaKhachHangSangMaTaiKhoan  @maKhachHang INT,  @maTaiKhoan INT OUTPUT -- Sử dụng tham số OUTPUT để trả về mã tài khoản  AS  BEGIN  -- Truy vấn để lấy mã tài khoản từ mã khách hàng  SELECT @maTaiKhoan = kh.maTaiKhoan  FROM KhachHang kh  WHERE kh.maKhachHang = @maKhachHang;  END; |

**-- Code c#**

|  |
| --- |
| public int ChuyenDoiMaKhachHangSangMaTaiKhoan(int maKhachHang)  {      try  {  dBConnection.openConnectionAdmin();  using (SqlCommand command = new SqlCommand("sp\_ChuyenDoiMaKhachHangSangMaTaiKhoan", dBConnection.getConnectionAdmin))  {  command.CommandType = CommandType.StoredProcedure;  command.Parameters.Add(new SqlParameter("@maKhachHang", maKhachHang));  SqlParameter outputParam = new SqlParameter("@maTaiKhoan", SqlDbType.Int)  {  Direction = ParameterDirection.Output  };  command.Parameters.Add(outputParam);  command.ExecuteNonQuery();  int maTaiKhoan = (int)command.Parameters["@maTaiKhoan"].Value;  return maTaiKhoan;  }  }  catch (Exception ex)  {  MessageBox.Show($"Lỗi: {ex.Message}");  return -1;  }  finally  {  dBConnection.closeConnectionAdmin();  }    } |

3.5.8 sp\_SearchKhachHangByTaiKhoan

**-- Tìm kiếm tài khoản khách hàng**

|  |
| --- |
| CREATE PROCEDURE sp\_SearchKhachHangByTaiKhoan  @taiKhoan VARCHAR(50)  AS  BEGIN  SELECT kh.maKhachHang, kh.maTaiKhoan, tk.taiKhoan, kh.thoiGianConLai, kh.loaiKhachHang, kh.diemTichLuy  FROM KhachHang kh  JOIN TaiKhoan tk ON kh.maTaiKhoan = tk.maTaiKhoan  WHERE tk.taiKhoan LIKE '%' + @taiKhoan + '%';  END;  **-- Code c#**  public List<ClassKhachHang> searchKhachHang(string taiKhoan)  {  try  {  dBConnection.openConnection();  SqlCommand cmd = new SqlCommand("sp\_SearchKhachHangByTaiKhoan", dBConnection.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@taiKhoan", taiKhoan);  SqlDataAdapter adapter = new SqlDataAdapter(cmd);  DataTable dataTable = new DataTable();  adapter.Fill(dataTable);  List<ClassKhachHang> khachHangs = new List<ClassKhachHang>();  foreach (DataRow row in dataTable.Rows)  {  ClassKhachHang khachHang = new ClassKhachHang(  maKhachHang: Convert.ToInt32(row["maKhachHang"]),  maTaiKhoan: Convert.ToInt32(row["maTaiKhoan"]),  thoiGianConLai: TimeSpan.Parse(row["thoiGianConLai"].ToString()),  loaiKhachHang: row["loaiKhachHang"].ToString(),  diemTichLuy: Convert.ToInt32(row["diemTichLuy"])  );  khachHangs.Add(khachHang);  }  return khachHangs;  }  catch (Exception ex)  {  MessageBox.Show("Đã xảy ra lỗi: " + ex.Message + "\n" + ex.StackTrace,  "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  return null;  }  finally  {  dBConnection.closeConnection();  }  } |

3.5.9 sp\_ThemDichVuVaoChiTietHoaDon

**-- Thêm dịch vụ vào hóa đơn**

|  |
| --- |
| CREATE PROCEDURE sp\_ThemDichVuVaoChiTietHoaDon  @maKhachHang INT,  @maMayTinh INT,  @maDichVu INT,  @soLuong INT  AS  BEGIN  -- Bắt đầu transaction  BEGIN TRANSACTION;    DECLARE @maHoaDon INT;  DECLARE @donGia DECIMAL(10, 2);  DECLARE @tongGiaTien DECIMAL(10, 2);    -- Kiểm tra hóa đơn chưa thanh toán của khách hàng  SELECT @maHoaDon = maHoaDon  FROM HoaDon  WHERE maKhachHang = @maKhachHang  AND maMayTinh = @maMayTinh  AND trangThai = 'ChuaThanhToan';  -- Nếu hóa đơn chưa tồn tại, tạo hóa đơn mới  IF @maHoaDon IS NULL  BEGIN  SET @maHoaDon = (SELECT ISNULL(MAX(maHoaDon), 0) + 1 FROM HoaDon);  -- Tạo hóa đơn mới  INSERT INTO HoaDon(maHoaDon, thoiGianTao, maKhachHang, maMayTinh, trangThai, triGia)  VALUES (@maHoaDon, GETDATE(), @maKhachHang, @maMayTinh, 'ChuaThanhToan', 0);    END  -- Kiểm tra số lượng dịch vụ còn đủ không  DECLARE @soLuongCon INT;  SELECT @soLuongCon = soLuong FROM DichVu WHERE maDichVu = @maDichVu;  IF @soLuong > @soLuongCon  BEGIN  RAISERROR('Số lượng dịch vụ không đủ', 16, 1);  ROLLBACK TRANSACTION;  RETURN;  END  -- Kiểm tra dịch vụ có tồn tại trong chi tiết hóa đơn chưa  IF EXISTS (SELECT 1 FROM ChiTietHoaDon WHERE maHoaDon = @maHoaDon AND maDichVu = @maDichVu)  BEGIN  -- Lấy đơn giá của dịch vụ  SELECT @donGia = donGia FROM DichVu WHERE maDichVu = @maDichVu;  -- Cập nhật số lượng và tổng giá tiền  UPDATE ChiTietHoaDon  SET soLuong = soLuong + @soLuong,  tongGiaTien = tongGiaTien + (@donGia \* @soLuong)  WHERE maHoaDon = @maHoaDon AND maDichVu = @maDichVu;  END  ELSE  BEGIN  -- Lấy đơn giá của dịch vụ  SELECT @donGia = donGia FROM DichVu WHERE maDichVu = @maDichVu;  -- Tính tổng giá tiền cho dịch vụ này  SET @tongGiaTien = @donGia \* @soLuong;  -- Thêm vào ChiTietHoaDon  INSERT INTO ChiTietHoaDon (maHoaDon, maDichVu, soLuong, tongGiaTien)  VALUES (@maHoaDon, @maDichVu, @soLuong, @tongGiaTien);  END  -- Cập nhật số lượng dịch vụ trong bảng DichVu  UPDATE DichVu  SET soLuong = soLuong - @soLuong  WHERE maDichVu = @maDichVu;  -- Cập nhật tổng giá trị hóa đơn  UPDATE HoaDon  SET triGia = triGia + @tongGiaTien  WHERE maHoaDon = @maHoaDon;  -- Commit transaction  COMMIT TRANSACTION;  END; |

**-- Code c#**

|  |
| --- |
| public bool ThemDichVu\_ChiTietHoaDon(int maTaiKhoan, int maMayTinh, int maDichVu, int soLuong)  {  int maKhachHang = LayMaKhachHang(maTaiKhoan);    try  {  dBConnection.openConnection();  SqlCommand cmd = new SqlCommand("sp\_ThemDichVuVaoChiTietHoaDon", dBConnection.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@maKhachHang", maKhachHang);  cmd.Parameters.AddWithValue("@maMayTinh", maMayTinh);  cmd.Parameters.AddWithValue("@maDichVu", maDichVu);  cmd.Parameters.AddWithValue("@soLuong", soLuong);  cmd.ExecuteNonQuery();  return true;  }  catch (Exception ex)  {  MessageBox.Show("that bai (ThemDichVu\_ChiTietHoaDon)" + ex);  return false;  }  finally  {  dBConnection.closeConnection();  }  } |

3.5.10 sp\_UpdateTinhTrangMayTinh

**-- Cập nhật trạng thái máy tính**

|  |
| --- |
| CREATE PROCEDURE sp\_UpdateTinhTrangMayTinh  @maMayTinh INT,  @tinhTrang NVARCHAR(50),  @maNguoiQuanLy INT  AS  BEGIN  UPDATE MayTinh  SET tinhTrang = @tinhTrang,  maNguoiQuanLy = @maNguoiQuanLy  WHERE maMayTinh = @maMayTinh;  END; |

**-- Code c#**

|  |
| --- |
| public void CapNhatTinhTrangMayTinh(int maMayTinh, string tinhTrang, int maTaiKhoanNguoiQuanLy)  {  try  {  using (SqlConnection conn = dBConnection.getConnection)  {  conn.Open();  using (SqlCommand cmd = new SqlCommand("sp\_UpdateTinhTrangMayTinh", conn))  {  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@maMayTinh", maMayTinh);  cmd.Parameters.AddWithValue("@tinhTrang", tinhTrang);  cmd.Parameters.AddWithValue("@maNguoiQuanLy", maTaiKhoanNguoiQuanLy);  cmd.ExecuteNonQuery();  }  }  }  catch (Exception ex)  {  MessageBox.Show("Lỗi khi cập nhật trạng thái máy tính: " + ex.Message);  }  } |

3.5.11 sp\_XemHoaDon\_KhachHang

**-- Khách hàng xem thông tin hóa đơn chưa thanh toán**

|  |
| --- |
| CREATE PROCEDURE sp\_XemHoaDon\_KhachHang  @maKhachHang INT  AS  BEGIN  -- Khai báo biến để lưu thông tin hóa đơn  DECLARE @maHoaDon INT;  -- Lấy mã hóa đơn lớn nhất của khách hàng  SELECT TOP 1 @maHoaDon = maHoaDon  FROM HoaDon  WHERE maKhachHang = @maKhachHang AND trangThai = 'ChuaThanhToan'  ORDER BY maHoaDon DESC;  -- Trả về thông tin hóa đơn nếu tồn tại  IF @maHoaDon IS NOT NULL  BEGIN  SELECT hd.maHoaDon,  hd.maUuDai,  hd.thoiGianTao,  kh.maKhachHang,  mt.maMayTinh,  dv.maDichVu,  cthd.soLuong,  cthd.tongGiaTien,  hd.triGia,  hd.trangThai  FROM HoaDon hd  JOIN KhachHang kh ON hd.maKhachHang = kh.maKhachHang  JOIN TaiKhoan tk ON kh.maTaiKhoan = tk.maTaiKhoan  JOIN MayTinh mt ON hd.maMayTinh = mt.maMayTinh  JOIN ChiTietHoaDon cthd ON hd.maHoaDon = cthd.maHoaDon  JOIN DichVu dv ON cthd.maDichVu = dv.maDichVu  WHERE hd.maHoaDon = @maHoaDon;  END  ELSE  BEGIN  PRINT 'Không tìm thấy hóa đơn cho khách hàng';  END  END; |

-- **Code c#**

|  |
| --- |
| public ClassHoaDon LayHoaDon\_KhachHang(int maKhachHang)  {  try  {  dBConnection.openConnection();  SqlCommand cmd = new SqlCommand("sp\_XemHoaDon\_KhachHang", dBConnection.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@maKhachHang", maKhachHang);  SqlDataReader reader = cmd.ExecuteReader();  if (reader.Read())  {  ClassHoaDon classHoaDon = new ClassHoaDon(Convert.ToInt32(reader["maHoaDon"]), Convert.ToInt32(reader["maKhachHang"]), Convert.ToInt32(reader["maMayTinh"]),  DateTime.Parse(reader["thoiGianTao"].ToString()), Convert.ToString(reader["trangThai"]), Convert.ToDecimal(reader["triGia"]),  reader.IsDBNull(reader.GetOrdinal("maUuDai")) ? (int?)null : Convert.ToInt32(reader["maUuDai"]));  reader.Close();  return classHoaDon;  }  else  {  reader.Close();  MessageBox.Show("Không tìm thấy hóa đơn với mã khách hàng: " + maKhachHang);  return null;  }  }  catch (Exception ex)  {  MessageBox.Show("Đã xảy ra lỗi: LayHoaDon\_KhachHang " + ex.Message);  return null;  }  finally  {  dBConnection.closeConnection();  }  } |

3.5.12 SuaDichVuProcedure

-- **Sửa thông tin dịch vụ**

|  |
| --- |
| CREATE PROCEDURE SuaDichVuProcedure  @maDichVu INT,  @tenDichVu NVARCHAR(100),  @loaiDichVu NVARCHAR(50),  @donGia DECIMAL(10, 2),  @soLuong INT  AS  BEGIN  -- Kiểm tra nếu dịch vụ tồn tại  IF NOT EXISTS (SELECT \* FROM DichVu WHERE maDichVu = @maDichVu)  BEGIN  PRINT 'Dịch vụ không tồn tại';  RETURN;  END  -- Cập nhật thông tin dịch vụ  UPDATE DichVu  SET  tenDichVu = @tenDichVu,  loaiDichVu = @loaiDichVu,  donGia = @donGia,  soLuong = @soLuong  WHERE maDichVu = @maDichVu;  END; |

**-- Code c#**

|  |
| --- |
| public void SuaDichVu(ClassDichVu dichvu)  {  try  {  dBConnection.openConnection();  SqlCommand cmd = new SqlCommand("SuaDichVuProcedure", dBConnection.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@maDichVu", dichvu.MaDichVu);  cmd.Parameters.AddWithValue("@tenDichVu", dichvu.TenDichVu);  cmd.Parameters.AddWithValue("@loaiDichVu", dichvu.LoaiDichVu);  cmd.Parameters.AddWithValue("@donGia", dichvu.DonGia);  cmd.Parameters.AddWithValue("@soLuong", dichvu.SoLuong);  cmd.ExecuteNonQuery();  MessageBox.Show("Sửa dịch vụ thành công");  }  catch (Exception exc)  {  MessageBox.Show("that bai (SuaDichVu)" + exc);  }  finally  {  dBConnection.closeConnection();  }  } |

3.5.13 SuaUuDaiProcedure

**-- Sửa thông tin ưu đãi**

|  |
| --- |
| CREATE PROCEDURE SuaUuDaiProcedure  @maUuDai INT,  @tenUuDai NVARCHAR(100),  @giaTri DECIMAL(10, 2),  @thoiGianBatDau DATETIME,  @thoiGianKetThuc DATETIME,  @dieuKien NVARCHAR(255),  @soLuong INT,  @tinhTrang NVARCHAR(50)  AS  BEGIN  -- Kiểm tra nếu ưu đãi tồn tại  IF NOT EXISTS (SELECT \* FROM UuDai WHERE maUuDai = @maUuDai)  BEGIN  PRINT 'Ưu đãi không tồn tại';  RETURN;  END  -- Cập nhật thông tin ưu đãi  UPDATE UuDai  SET  tenUuDai = @tenUuDai,  giaTri = @giaTri,  thoiGianBatDau = @thoiGianBatDau,  thoiGianKetThuc = @thoiGianKetThuc,  dieuKien = @dieuKien,  soLuong = @soLuong,  tinhTrang = @tinhTrang  WHERE maUuDai = @maUuDai;  END; |

**-- Code c#**

|  |
| --- |
| public void SuaUuDai(ClassUuDai UuDai)  {  try  {  dBConnection.openConnection();  SqlCommand cmd = new SqlCommand("SuaUuDaiProcedure", dBConnection.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@maUuDai", UuDai.MaUuDai);  cmd.Parameters.AddWithValue("@tenUuDai", UuDai.TenUuDai);  cmd.Parameters.AddWithValue("@giaTri", UuDai.GiaTri);  cmd.Parameters.AddWithValue("@thoiGianBatDau", UuDai.ThoiGianBatDau);  cmd.Parameters.AddWithValue("@thoiGianKetThuc", UuDai.ThoiGianKetThuc);  cmd.Parameters.AddWithValue("@dieuKien", UuDai.DieuKien);  cmd.Parameters.AddWithValue("@soLuong", UuDai.SoLuong);  cmd.Parameters.AddWithValue("@tinhTrang", UuDai.TinhTrang);  cmd.ExecuteNonQuery();  MessageBox.Show("Sửa ưu đãi thành công");  }  catch (Exception exc)  {  MessageBox.Show("that bai (SuaUuDai)" + exc);  }  finally  {  dBConnection.closeConnection();  }  } |

3.5.14 ThemDichVuProcedure

**-- Thêm dịch vụ**

|  |
| --- |
| CREATE PROCEDURE ThemDichVuProcedure  @tenDichVu NVARCHAR(100),  @loaiDichVu NVARCHAR(50),  @donGia DECIMAL(10, 2),  @soLuong INT,  @maNguoiQuanLy INT  AS  BEGIN  -- Tìm giá trị maDichVu lớn nhất hiện có và tăng thêm 1  DECLARE @maDichVu INT;  SELECT @maDichVu = ISNULL(MAX(maDichVu), 0) + 1 FROM DichVu;  -- Thêm dịch vụ mới với maDichVu được tính toán  INSERT INTO DichVu (maDichVu, tenDichVu, loaiDichVu, donGia, soLuong, maNguoiQuanLy)  VALUES (@maDichVu, @tenDichVu, @loaiDichVu, @donGia, @soLuong, @maNguoiQuanLy);  PRINT 'Dịch vụ đã được thêm thành công';  END; |

-- **Code c#**

|  |
| --- |
| public void ThemDichVu(ClassDichVu dichvu, int maTaiKhoanNguoiQuanLy)  {  try  {  dBConnection.openConnection();  SqlCommand cmd = new SqlCommand("ThemDichVuProcedure", dBConnection.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@tenDichVu", dichvu.TenDichVu);  cmd.Parameters.AddWithValue("@loaiDichVu", dichvu.LoaiDichVu);  cmd.Parameters.AddWithValue("@donGia", dichvu.DonGia);  cmd.Parameters.AddWithValue("@soLuong", dichvu.SoLuong);  cmd.Parameters.AddWithValue("@maNguoiQuanLy", maTaiKhoanNguoiQuanLy);  cmd.ExecuteNonQuery();  MessageBox.Show("Thêm dịch vụ thành công");  }  catch (Exception exc)  {  MessageBox.Show("that bai (ThemDichVu)" + exc);  }  finally  {  dBConnection.closeConnection();  }  } |

3.5.15 ThemMaUuDaiVaoHoaDon

**-- Thêm ưu đãi cho hóa đơn**

|  |
| --- |
| CREATE PROCEDURE ThemMaUuDaiVaoHoaDon  @maHoaDon INT,  @maUuDai INT  AS  BEGIN  DECLARE @soLuong INT, @thoiGianBatDau DATE, @thoiGianKetThuc DATE, @thoiGianHienTai DATE;  SET @thoiGianHienTai = GETDATE();  -- Kiểm tra xem mã hóa đơn có tồn tại hay không  IF NOT EXISTS (SELECT 1 FROM HoaDon WHERE maHoaDon = @maHoaDon)  BEGIN  RAISERROR(N'Mã hóa đơn không tồn tại.', 16, 1);  RETURN;  END;  -- Lấy thông tin ưu đãi: số lượng, ngày bắt đầu và ngày kết thúc  SELECT @soLuong = soLuong, @thoiGianBatDau = thoiGianBatDau, @thoiGianKetThuc = thoiGianKetThuc  FROM UuDai  WHERE maUuDai = @maUuDai;  -- Kiểm tra xem mã ưu đãi có tồn tại và số lượng còn lại có lớn hơn 0 hay không  IF @soLuong IS NULL  BEGIN  RAISERROR(N'Mã ưu đãi không tồn tại.', 16, 1);  RETURN;  END;  ELSE IF @soLuong <= 0  BEGIN  RAISERROR(N'Ưu đãi đã hết số lượng.', 16, 1);  RETURN;  END;  -- Kiểm tra ngày hiện tại có nằm trong khoảng ngày bắt đầu và ngày kết thúc của ưu đãi hay không  IF @thoiGianHienTai < @thoiGianBatDau OR @thoiGianHienTai > @thoiGianKetThuc  BEGIN  RAISERROR(N'Ưu đãi không hợp lệ cho ngày hiện tại.', 16, 1);  RETURN;  END;  -- Cập nhật mã ưu đãi vào hóa đơn nếu tất cả điều kiện trên đều thỏa  UPDATE HoaDon  SET maUuDai = @maUuDai  WHERE maHoaDon = @maHoaDon;  PRINT 'Cập nhật mã ưu đãi thành công.';  END; |

**-- Code c#**

|  |
| --- |
| public bool ThemUuDai\_HoaDon(int maHoaDon, int maUuDai)  {  try  {  dBConnection.openConnection();  SqlCommand cmd = new SqlCommand("ThemMaUuDaiVaoHoaDon", dBConnection.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@maHoaDon", maHoaDon);  cmd.Parameters.AddWithValue("@maUuDai", maUuDai);  cmd.ExecuteNonQuery();  MessageBox.Show("Thêm ưu đãi thành công");  return true;  }  catch (Exception ex)  {  MessageBox.Show("that bai (ThemUuDai\_HoaDon)" + ex);  return false;  }  finally  {  dBConnection.closeConnection();  }  } |

3.5.16 ThemPhienDangNhap

**-- Thêm phiên đăng nhập khi khách hàng đăng nhập**

|  |
| --- |
| CREATE PROCEDURE ThemPhienDangNhap  @maKhachHang INT,  @maMayTinh INT  AS  BEGIN  -- Tìm giá trị maPhienDangNhap lớn nhất hiện có và tăng thêm 1  DECLARE @maPhienDangNhap INT;  SELECT @maPhienDangNhap = ISNULL(MAX(maPhienDangNhap), 0) + 1 FROM PhienDangNhap;  -- Cài đặt thời gian bắt đầu sử dụng  DECLARE @thoiGianBatDau TIME;  SET @thoiGianBatDau= CONVERT(TIME, GETDATE());  -- Tính toán thời gian đã sử dụng  DECLARE @thoiGianSuDung TIME;  SET @thoiGianSuDung = '00:00:00';  -- Lấy thoiGianConLai của khách hàng  DECLARE @thoiGianConLai TIME;  SELECT @thoiGianConLai = thoiGianConLai FROM KhachHang WHERE maKhachHang = @maKhachHang;  -- Thêm phiên đăng nhập  INSERT INTO PhienDangNhap (maPhienDangNhap, maKhachHang, maMayTinh, thoiGianBatDau, thoiGianSuDung, thoiGianConLai)  VALUES (@maPhienDangNhap, @maKhachHang, @maMayTinh, @thoiGianBatDau, @thoiGianSuDung, @thoiGianConLai);  END; |

**-- Code c#**

|  |
| --- |
| public void ThemPhienDangNhap(int maKhachHang)  {  try  {  dBConnection.openConnectionAdmin();  int maMayTinh = 0;  SqlCommand cmd = new SqlCommand("ThemPhienDangNhap", dBConnection.getConnectionAdmin);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@maKhachHang", maKhachHang);  cmd.Parameters.AddWithValue("@maMayTinh", maMayTinh);  cmd.ExecuteNonQuery();  MessageBox.Show("Thêm phiên đăng nhập thành công vào máy: " + maMayTinh, "Thông báo");  }  catch (Exception ex)  {  MessageBox.Show("Đã xảy ra lỗi: " + ex.Message, "Lỗi");  }  finally  {  dBConnection.closeConnectionAdmin();  }  } |

3.5.17 ThemUuDaiProcedure

**-- Thêm ưu đãi**

|  |
| --- |
| CREATE PROCEDURE ThemUuDaiProcedure  @tenUuDai NVARCHAR(100),  @giaTri DECIMAL(10, 2),  @thoiGianBatDau DATETIME,  @thoiGianKetThuc DATETIME,  @dieuKien NVARCHAR(255),  @soLuong INT,  @tinhTrang NVARCHAR(50),  @maNguoiQuanLy INT  AS  BEGIN  -- Tìm giá trị maUuDai lớn nhất hiện có và tăng thêm 1  DECLARE @maUuDai INT;  SELECT @maUuDai = ISNULL(MAX(maUuDai), 0) + 1 FROM UuDai;  -- Thêm ưu đãi mới với maUuDai được tính toán  INSERT INTO UuDai (maUuDai, tenUuDai, giaTri, thoiGianBatDau, thoiGianKetThuc, dieuKien, soLuong, tinhTrang, maNguoiQuanLy)  VALUES (@maUuDai, @tenUuDai, @giaTri, @thoiGianBatDau, @thoiGianKetThuc, @dieuKien, @soLuong, @tinhTrang, @maNguoiQuanLy);  END; |

**-- Code c#**

|  |
| --- |
| public void ThemUuDai(ClassUuDai UuDai, int maTaiKhoanNguoiQuanLy)  {  try  {  dBConnection.openConnection();  SqlCommand cmd = new SqlCommand("ThemUuDaiProcedure", dBConnection.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@tenUuDai", UuDai.TenUuDai);  cmd.Parameters.AddWithValue("@giaTri", UuDai.GiaTri);  cmd.Parameters.AddWithValue("@thoiGianBatDau", UuDai.ThoiGianBatDau);  cmd.Parameters.AddWithValue("@thoiGianKetThuc", UuDai.ThoiGianKetThuc);  cmd.Parameters.AddWithValue("@dieuKien", UuDai.DieuKien);  cmd.Parameters.AddWithValue("@soLuong", UuDai.SoLuong);  cmd.Parameters.AddWithValue("@tinhTrang", UuDai.TinhTrang);  cmd.Parameters.AddWithValue("@maNguoiQuanLy", maTaiKhoanNguoiQuanLy);  cmd.ExecuteNonQuery();  MessageBox.Show("Thêm ưu đãi thành công");  }  catch (Exception exc)  {  MessageBox.Show("that bai (ThemUuDai)" + exc);  }  finally  {  dBConnection.closeConnection();  }  } |

3.5.18 TimKiemDichVuProcedure

**-- Tìm kiếm dịch vụ**

|  |
| --- |
| CREATE PROCEDURE TimKiemDichVuProcedure  @tenDichVu NVARCHAR(100),  @loaiDichVu NVARCHAR(50)  AS  BEGIN  -- Tìm các dịch vụ có tên chứa chuỗi @tenDichVu  SELECT \*  FROM DichVu  WHERE tenDichVu LIKE N'%' + @tenDichVu + N'%'  AND loaiDichVu = @loaiDichVu;  END; |

**-- Code c#**

|  |
| --- |
| public List<ClassDichVu> TimKiemDichVu(string timkiem, string loaiDichVu)  {  try  {  dBConnection.openConnection();  SqlCommand cmd = new SqlCommand("TimKiemDichVuProcedure", dBConnection.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@tenDichVu", timkiem);  cmd.Parameters.AddWithValue("@loaiDichVu", loaiDichVu);  SqlDataReader reader = cmd.ExecuteReader();  List<ClassDichVu> listdv = new List<ClassDichVu>();  while (reader.Read())  {  ClassDichVu dv = new ClassDichVu(Convert.ToInt32(reader["maDichVu"]),  reader["tenDichVu"].ToString(), reader["loaiDichVu"].ToString(),  Convert.ToDouble(reader["donGia"]), Convert.ToInt32(reader["soLuong"]));  listdv.Add(dv);  }  reader.Close();  return listdv;  }  catch (Exception exc)  {  MessageBox.Show("that bai (TimKiemDichVu)" + exc);  return null;  }  finally  {  dBConnection.closeConnection();  }  } |

3.5.19 TimKiemUuDaiThuongProcedure

**-- Tìm kiếm ưu đãi dành cho khách hàng thường**

|  |
| --- |
| CREATE PROCEDURE TimKiemUuDaiThuongProcedure  @tenUuDai NVARCHAR(100)  AS  BEGIN  SELECT \*  FROM UuDai  WHERE tenUuDai LIKE N'%' + @tenUuDai + '%' AND dieuKien = N'Tất cả khách hàng';  END; |

**-- Code c#**

|  |
| --- |
| public List<ClassUuDai> TimKiemUuDaiThuong(string timkiem)  {  try  {  dBConnection.openConnection();  SqlCommand cmd = new SqlCommand("TimKiemUuDaiThuongProcedure", dBConnection.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@tenUuDai", timkiem);  SqlDataReader reader = cmd.ExecuteReader();  List<ClassUuDai> listud = new List<ClassUuDai>();  while (reader.Read())  {  ClassUuDai dv = new ClassUuDai(Convert.ToInt32(reader["maUuDai"]),  reader["tenUuDai"].ToString(), Convert.ToDouble(reader["giaTri"]),  Convert.ToDateTime(reader["thoiGianBatDau"]), Convert.ToDateTime(reader["thoiGianKetThuc"]),  reader["dieuKien"].ToString(), Convert.ToInt32(reader["soLuong"]), reader["tinhTrang"].ToString());  listud.Add(dv);  }  reader.Close();  return listud;  }  catch (Exception exc)  {  MessageBox.Show("that bai (TimKiemUuDaiThuong)" + exc);  return null;  }  finally  {  dBConnection.closeConnection();  }  } |

3.5.20 TimKiemUuDaiVipProcedure

**-- Tìm kiếm ưu đãi dành cho khách hàng Vip**

|  |
| --- |
| CREATE PROCEDURE TimKiemUuDaiVipProcedure  @tenUuDai NVARCHAR(100)  AS  BEGIN  SELECT \*  FROM UuDai  WHERE tenUuDai LIKE N'%' + @tenUuDai + '%';  END; |

**-- Code c#**

|  |
| --- |
| public List<ClassUuDai> TimKiemUuDaiVip(string timkiem)  {  try  {  dBConnection.openConnection();  SqlCommand cmd = new SqlCommand("TimKiemUuDaiVipProcedure", dBConnection.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@tenUuDai", timkiem);  SqlDataReader reader = cmd.ExecuteReader();  List<ClassUuDai> listud = new List<ClassUuDai>();  while (reader.Read())  {  ClassUuDai dv = new ClassUuDai(Convert.ToInt32(reader["maUuDai"]),  reader["tenUuDai"].ToString(), Convert.ToDouble(reader["giaTri"]),  Convert.ToDateTime(reader["thoiGianBatDau"]), Convert.ToDateTime(reader["thoiGianKetThuc"]),  reader["dieuKien"].ToString(), Convert.ToInt32(reader["soLuong"]), reader["tinhTrang"].ToString());  listud.Add(dv);  }  reader.Close();  return listud;  }  catch (Exception exc)  {  MessageBox.Show("that bai (TimKiemUuDaiVip)" + exc);  return null;  }  finally  {  dBConnection.closeConnection();  }  } |

3.5.21 XemChiTietHoaDonProcedure

**-- Xem chi tiết hóa đơn thông qua mã hóa đơn**

|  |
| --- |
| CREATE PROCEDURE XemChiTietHoaDonProcedure  @maHoaDon INT  AS  BEGIN  SELECT hd.maHoaDon,  hd.maUuDai,  hd.thoiGianTao,  kh.maKhachHang,  mt.maMayTinh,  dv.maDichVu,  cthd.soLuong,  cthd.tongGiaTien,  hd.triGia,  hd.trangThai  FROM HoaDon hd  JOIN KhachHang kh ON hd.maKhachHang = kh.maKhachHang  JOIN TaiKhoan tk ON kh.maTaiKhoan = tk.maTaiKhoan  JOIN MayTinh mt ON hd.maMayTinh = mt.maMayTinh  JOIN ChiTietHoaDon cthd ON hd.maHoaDon = cthd.maHoaDon  JOIN DichVu dv ON cthd.maDichVu = dv.maDichVu  WHERE hd.maHoaDon = @maHoaDon;  END; |

**-- Code c#**

|  |
| --- |
| public ClassHoaDon LayHoaDon(int maHoaDon)  {    try  {  string sqlStr = string.Format("XemChiTietHoaDonProcedure");  dBConnection.openConnection();  SqlCommand cmd = new SqlCommand(sqlStr, dBConnection.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@maHoaDon", maHoaDon);  SqlDataReader reader = cmd.ExecuteReader();  if (reader.Read())  {  ClassHoaDon classHoaDon = new ClassHoaDon(maHoaDon, Convert.ToInt32(reader["maKhachHang"]), Convert.ToInt32(reader["maMayTinh"]),  DateTime.Parse(reader["thoiGianTao"].ToString()), Convert.ToString(reader["trangThai"]), Convert.ToDecimal(reader["triGia"]),  reader.IsDBNull(reader.GetOrdinal("maUuDai")) ? (int?)null : Convert.ToInt32(reader["maUuDai"]));  reader.Close();  return classHoaDon;  }  else  {  reader.Close();  MessageBox.Show("Không tìm thấy hóa đơn với mã: " + maHoaDon);  return null;  }  }  catch (Exception ex)  {  MessageBox.Show("Đã xảy ra lỗi: LayHoaDon " + ex.Message);  return null;  }  finally  {  dBConnection.closeConnection();  }  } |

3.5.22 XoaDichVuProcedure

**-- Xóa dịch vụ**

|  |
| --- |
| CREATE PROCEDURE XoaDichVuProcedure  @maDichVu INT  AS  BEGIN  -- Kiểm tra nếu dịch vụ tồn tại  IF NOT EXISTS (SELECT \* FROM DichVu WHERE maDichVu = @maDichVu)  BEGIN  PRINT 'Dịch vụ không tồn tại';  RETURN;  END  -- Thực hiện xóa dịch vụ  DELETE FROM DichVu WHERE maDichVu = @maDichVu;  END; |

**-- Code c#**

|  |
| --- |
| public void XoaDichVu(int madichvu)  {  try  {  dBConnection.openConnection();  SqlCommand cmd = new SqlCommand("XoaDichVuProcedure", dBConnection.getConnection);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.Add(new SqlParameter("@maDichVu", madichvu));  cmd.ExecuteNonQuery();  MessageBox.Show("Xóa dịch vụ thành công");  }  catch (Exception exc)  {  MessageBox.Show("that bai (XoadichVu)" + exc);  }  finally  {  dBConnection.closeConnection();  }  } |

3.5.23 dbo.XoaPhienDangNhap

**-- Xóa phiên đăng nhập**

|  |
| --- |
| CREATE PROCEDURE dbo.XoaPhienDangNhap  @maKhachHang INT,  @thoiGianConLai Time  AS  BEGIN  -- Cập nhật thời gian còn lại cho khách hàng  UPDATE KhachHang  SET thoiGianConLai = @thoiGianConLai  WHERE maKhachHang = @maKhachHang;  -- Xoá phiên đăng nhập  DELETE FROM PhienDangNhap  WHERE maKhachHang = @maKhachHang;    END; |

**-- Code c#**

|  |
| --- |
| public void XoaPhienDangNhap(int maKhachHang, TimeSpan thoiGianConLai)  {  try  {  dBConnection.openConnectionAdmin();  SqlCommand cmd = new SqlCommand("XoaPhienDangNhap", dBConnection.getConnectionAdmin);  cmd.CommandType = CommandType.StoredProcedure;  cmd.Parameters.AddWithValue("@maKhachHang", maKhachHang);  cmd.Parameters.AddWithValue("@thoiGianConLai", thoiGianConLai);  cmd.ExecuteNonQuery();  MessageBox.Show("Bạn đã đăng xuất thành công!");    }  catch (Exception ex)  {  MessageBox.Show("Đã xảy ra lỗi khi đăng xuất: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally { dBConnection.closeConnectionAdmin();}  } |

3.5.24 XoaUuDaiProcedure

**-- Xóa ưu đãi**

|  |
| --- |
| CREATE PROCEDURE XoaUuDaiProcedure  @maUuDai INT  AS  BEGIN  -- Kiểm tra nếu ưu đãi tồn tại  IF NOT EXISTS (SELECT \* FROM UuDai WHERE maUuDai = @maUuDai)  BEGIN  PRINT 'Ưu đãi không tồn tại';  RETURN;  END  DELETE FROM UuDai WHERE maUuDai = @maUuDai;  END; |

**-- Code c#**

|  |
| --- |
| CREATE PROCEDURE XoaUuDaiProcedure  @maUuDai INT  AS  BEGIN  -- Kiểm tra nếu ưu đãi tồn tại  IF NOT EXISTS (SELECT \* FROM UuDai WHERE maUuDai = @maUuDai)  BEGIN  PRINT 'Ưu đãi không tồn tại';  RETURN;  END  DELETE FROM UuDai WHERE maUuDai = @maUuDai;  END; |

CHƯƠNG 4: TẠO USER VÀ PHÂN QUYỀN

|  |
| --- |
| --Khởi tạo role NguoiQuanLy và role KhachHang  CREATE ROLE NguoiQuanLy  CREATE ROLE KhachHang  --Gán các quyền truy cập view cho role KhachHang  GRANT SELECT ON ChiTietHoaDonView TO KhachHang;  GRANT SELECT ON DichVuChuaThanhToanView TO KhachHang;  GRANT SELECT ON DichVuDoAnView TO KhachHang;  GRANT SELECT ON DichVuTheCaoView TO KhachHang;  GRANT SELECT ON DichVuThucUongView TO KhachHang;  GRANT SELECT ON ViewUuDaiThuong TO KhachHang;  GRANT SELECT ON ViewUuDaiVIP TO KhachHang;  GRANT EXECUTE ON sp\_XemHoaDon\_KhachHang TO KhachHang;  --Gán các quyền thủ tục và hàm cho role KhachHang  GRANT EXECUTE ON DoiMatKhau TO KhachHang;  GRANT EXECUTE ON TimKiemDichVuProcedure TO KhachHang;  GRANT EXECUTE ON TimKiemUuDaiThuongProcedure TO KhachHang;  GRANT EXECUTE ON TimKiemUuDaiVipProcedure TO KhachHang;  GRANT EXECUTE ON sp\_ThemDichVuVaoChiTietHoaDon TO KhachHang;  GRANT EXECUTE ON ThemMaUuDaiVaoHoaDon TO KhachHang;  GRANT EXECUTE ON sp\_CapNhatTinhTrangHoaDon TO KhachHang;  GRANT EXECUTE ON sp\_GetChiTietHoaDon TO KhachHang;  GRANT EXECUTE ON KiemTraKhachHangDangNhap TO KhachHang;  GRANT EXECUTE ON fn\_LayMaKhachHang TO KhachHang;  GRANT EXECUTE ON layLoaiKhachHang TO KhachHang; |

|  |
| --- |
| CREATE TRIGGER CreateSQLAccountKhachHang  ON KhachHang  AFTER INSERT  AS  BEGIN  DECLARE  @maTaiKhoan INT,  @taiKhoan VARCHAR(50),  @matKhau VARCHAR(50),  @sqlString NVARCHAR(2000);  -- Lấy giá trị từ bảng inserted (giả định chỉ có một hàng được chèn mỗi lần)  SELECT @maTaiKhoan = i.maTaiKhoan,  @taiKhoan = tk.taiKhoan,  @matKhau = tk.matKhau  FROM inserted i JOIN TaiKhoan tk ON  i.maTaiKhoan = tk.maTaiKhoan;  -- Tạo tài khoản đăng nhập (LOGIN)  SET @sqlString = 'CREATE LOGIN [' + @taiKhoan + '] WITH PASSWORD = ''' + @matKhau + ''', DEFAULT\_DATABASE=[QuanLyDichVuQuanNet], CHECK\_EXPIRATION=OFF, CHECK\_POLICY=OFF';  EXEC (@sqlString);  -- Tạo người dùng (USER) liên kết với tài khoản đăng nhập  SET @sqlString = 'CREATE USER [' + @taiKhoan + '] FOR LOGIN [' + @taiKhoan + ']';  EXEC (@sqlString);  SET @sqlString = ' ALTER ROLE KhachHang ADD MEMBER '+ @taiKhoan;  EXEC (@sqlString);    END;​ |

|  |
| --- |
| CREATE TRIGGER CreateSQLAccountQuanLy  ON NguoiQuanLy  AFTER INSERT  AS  BEGIN  DECLARE  @maTaiKhoan INT,  @taiKhoan VARCHAR(50),  @matKhau VARCHAR(50),  @sqlString NVARCHAR(2000);  -- Lấy giá trị từ bảng inserted (giả định chỉ có một hàng được chèn mỗi lần)  SELECT @maTaiKhoan = i.maTaiKhoan,  @taiKhoan = tk.taiKhoan,  @matKhau = tk.matKhau  FROM inserted i JOIN TaiKhoan tk ON  i.maTaiKhoan = tk.maTaiKhoan;  -- Tạo tài khoản đăng nhập (LOGIN)  SET @sqlString = 'CREATE LOGIN [' + @taiKhoan + '] WITH PASSWORD = ''' + @matKhau + ''', DEFAULT\_DATABASE=[QuanLyDichVuQuanNet], CHECK\_EXPIRATION=OFF, CHECK\_POLICY=OFF';  EXEC (@sqlString);  -- Tạo người dùng (USER) liên kết với tài khoản đăng nhập  SET @sqlString = 'CREATE USER [' + @taiKhoan + '] FOR LOGIN [' + @taiKhoan + ']';  EXEC (@sqlString);  SET @sqlString = 'ALTER server ROLE sysadmin ADD MEMBER '+ @taiKhoan;  EXEC (@sqlString);    END;​ |

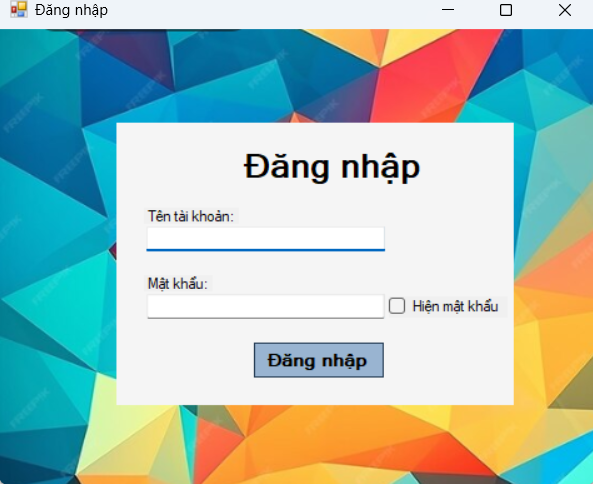
CHƯƠNG 5: THIẾT KẾ GIAO DIỆN HỆ THỐNG

**Môi trường cài đặt và công nghệ sử dụng**- Hệ cơ sở dữ liệu Microsoft SQL Server Management Studio 19.  
- Chương trình Winform được viết bằng Microsoft Visual Studio 2022 - trên nền tảng  
ASP.NET Web Application (.NET Framework) C#.  
**Giao diện và chức năng của chương trình:**

5.1 Form màn hình kết nối database

**  
Form sẽ duyệt các server có trên máy đồng thời lấy ra các database có trong server đó. Có 2 phương thức đăng nhập: bằng Window Authentication hoặc SQL Server Authentication**

5.2 Form đăng nhập:

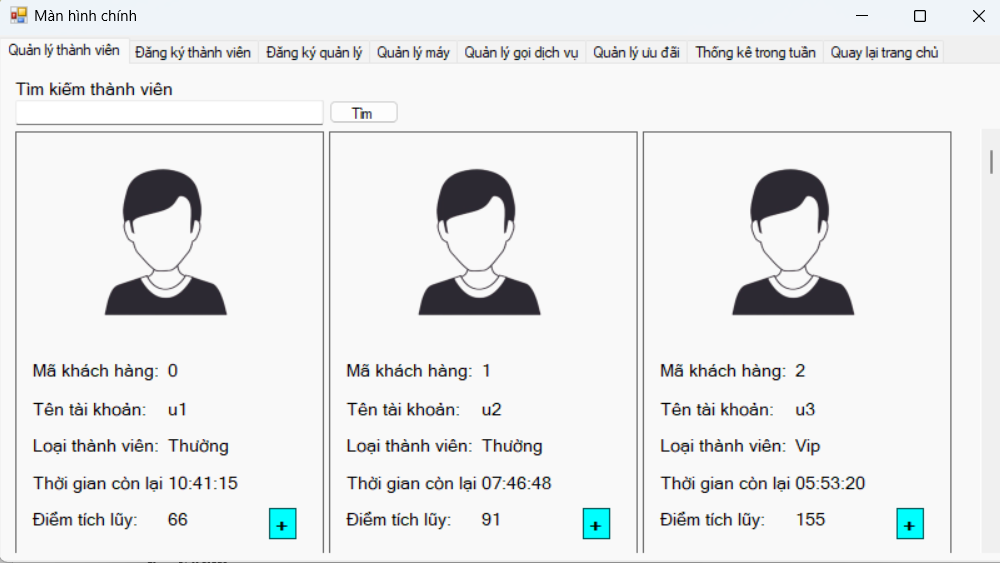


Đăng nhập bằng tài khoản khách hang hoặc tài khoản người quản lý:

5.3 Các Form dành cho tài khoản người quản lý

5.3.1 Form màn hình chính

5.3.1.1 Form Quản lý thành viên

****

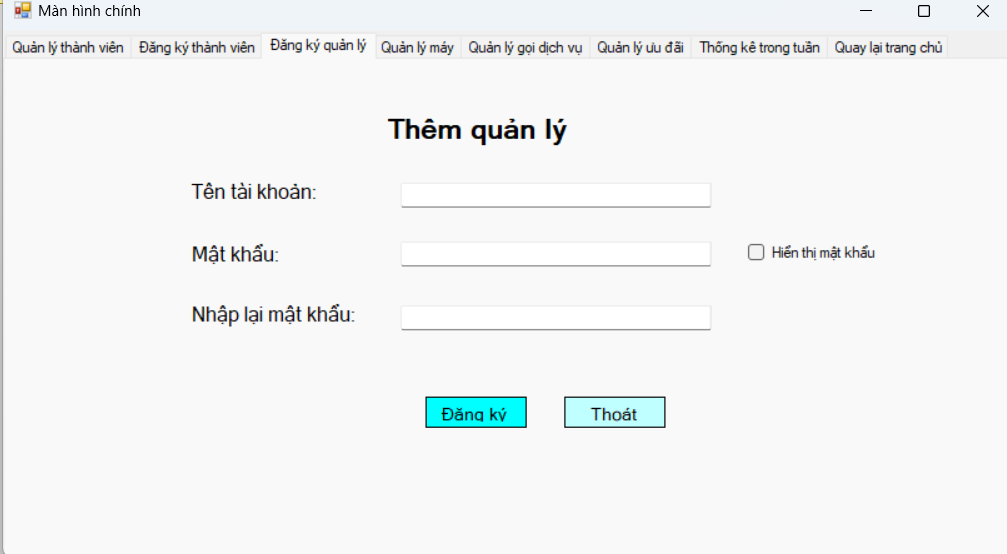
**Quản lý các tài khoản thành viên**

5.3.1.2 Form Đăng ký thành viên

****

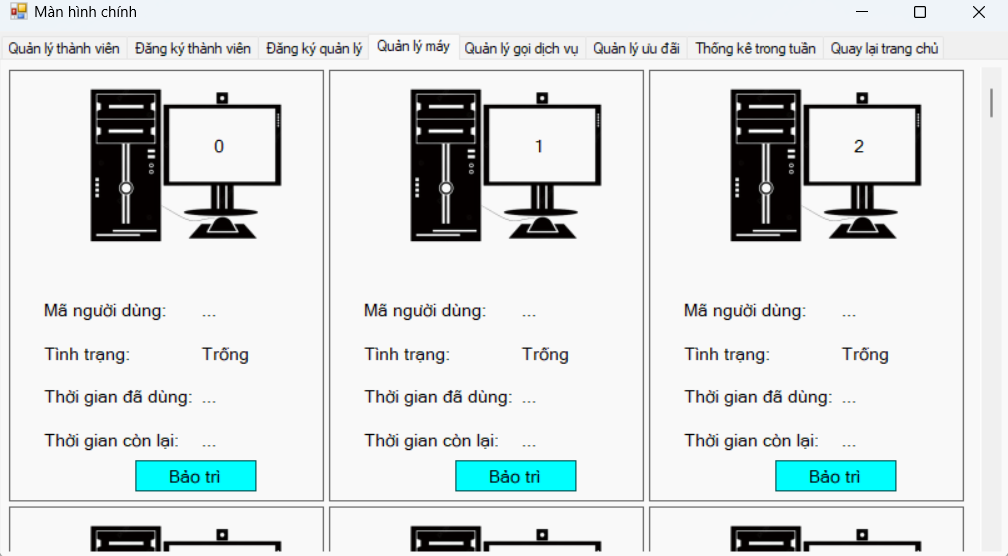
**Đăng ký tài khoản thành viên mới**

5.3.1.3 Form Đăng ký quản lý

****

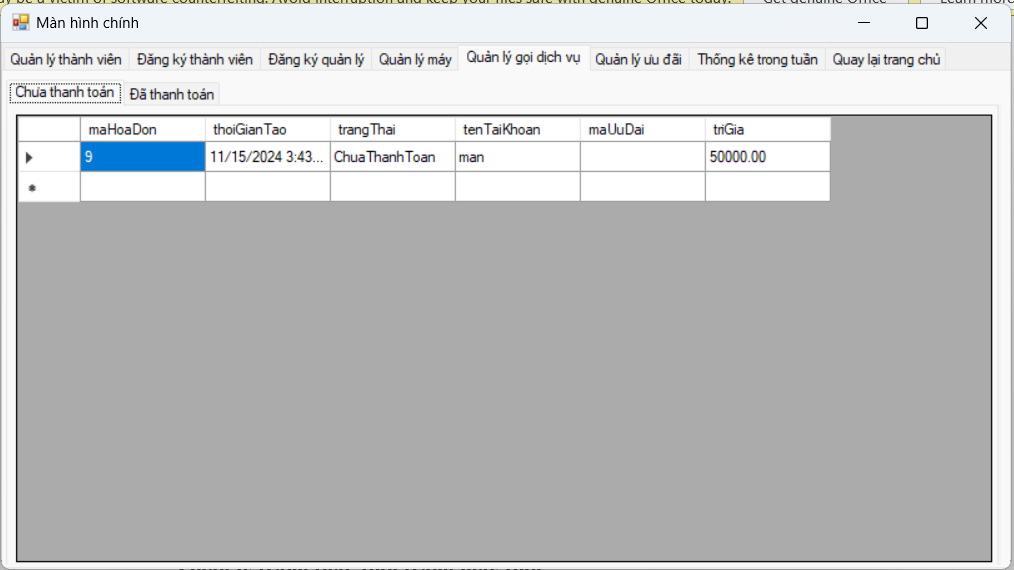
**Đăng ký tài khoản người quản lý**

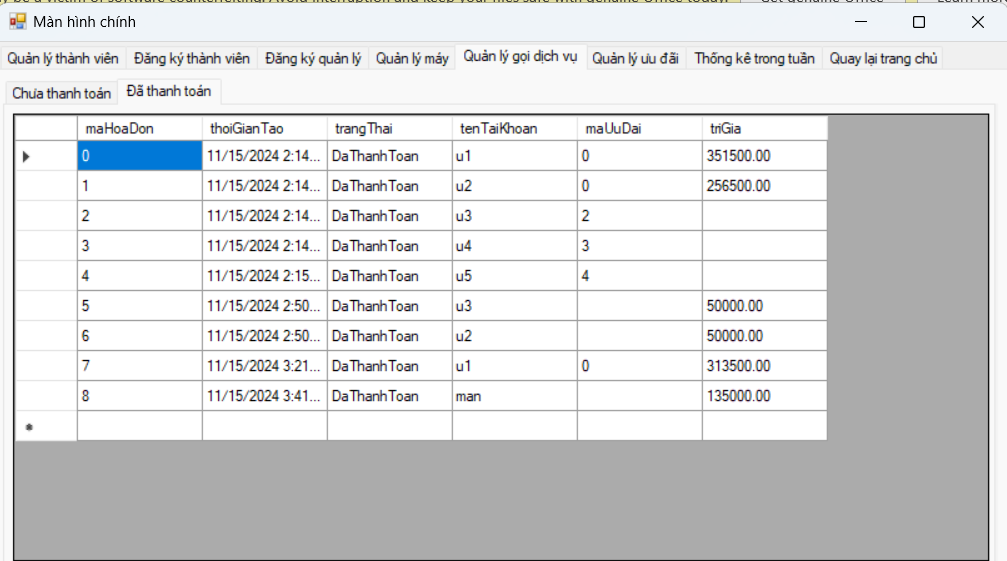
5.3.1.4 Form Quản lý máy

**  
Quản lý trạng thái, tình trạng máy tính**

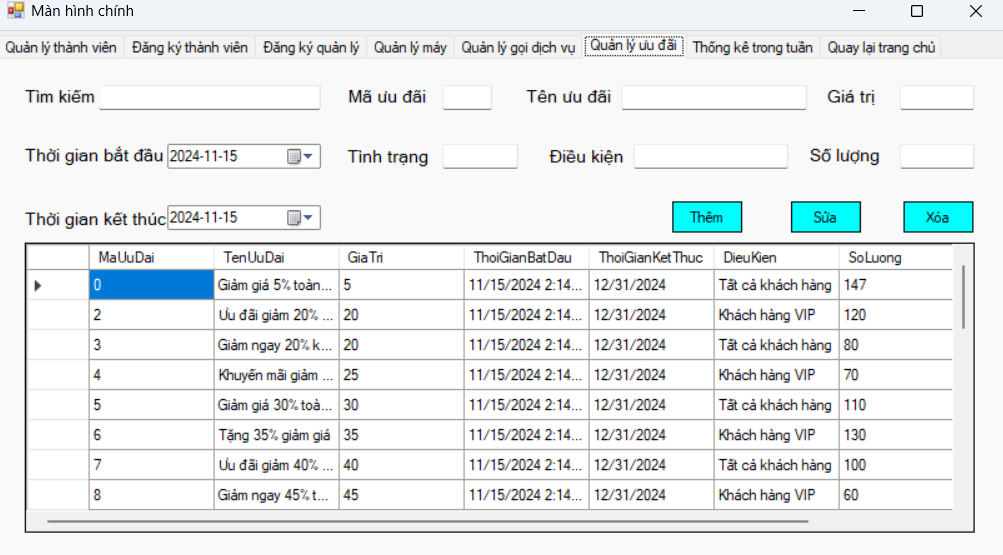
5.3.1.5 Form Quản lý gọi dịch vụ

**Quản lý các hóa đơn dịch vụ có 2 Form: chưa thanh toán và đã thanh toán**

****

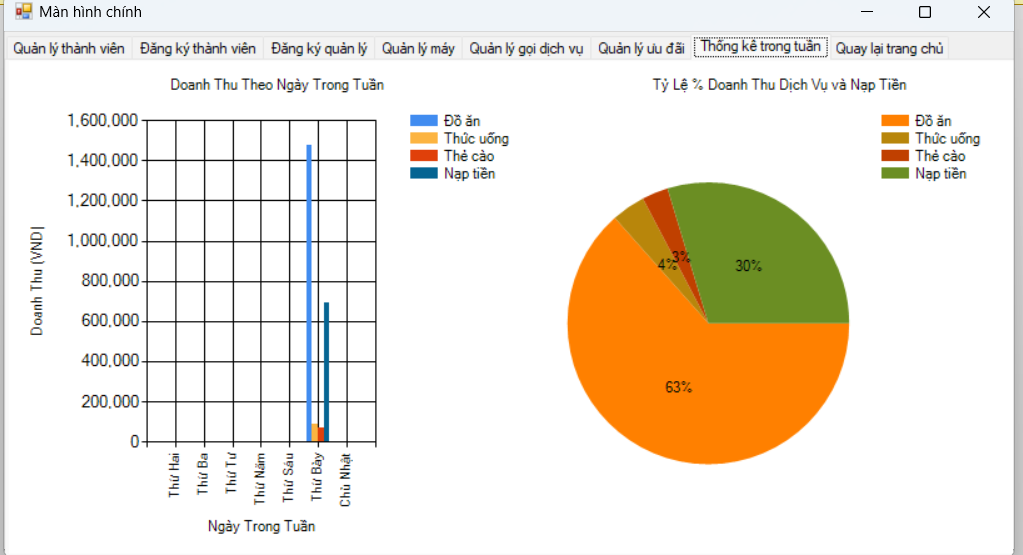
****

5.3.1.6 Form Quản lý ưu đãi

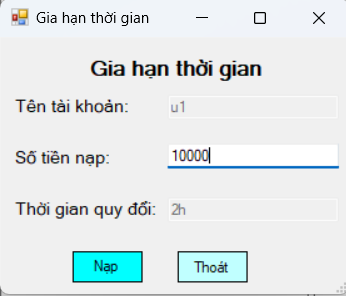
****

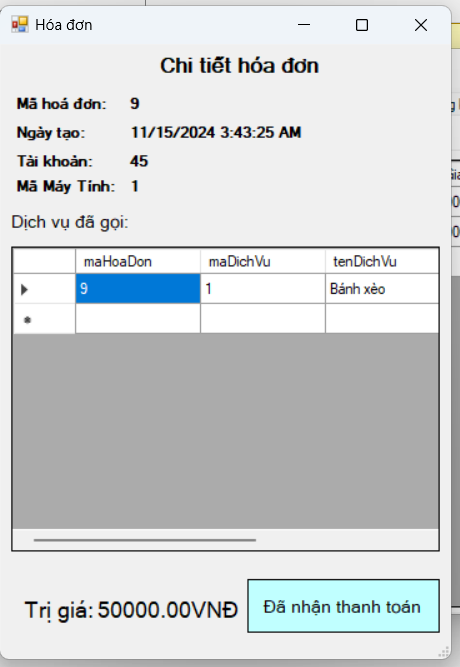
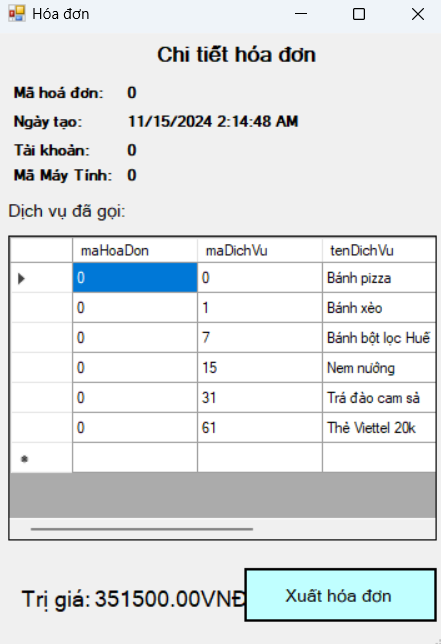
**Quản lý them xóa sửa các ưu đãi**

5.3.1.7 Form Thống kê trong tuần

**  
Thống kê doanh thu trong tuần**

5.3.1.8 Form Nạp tiền

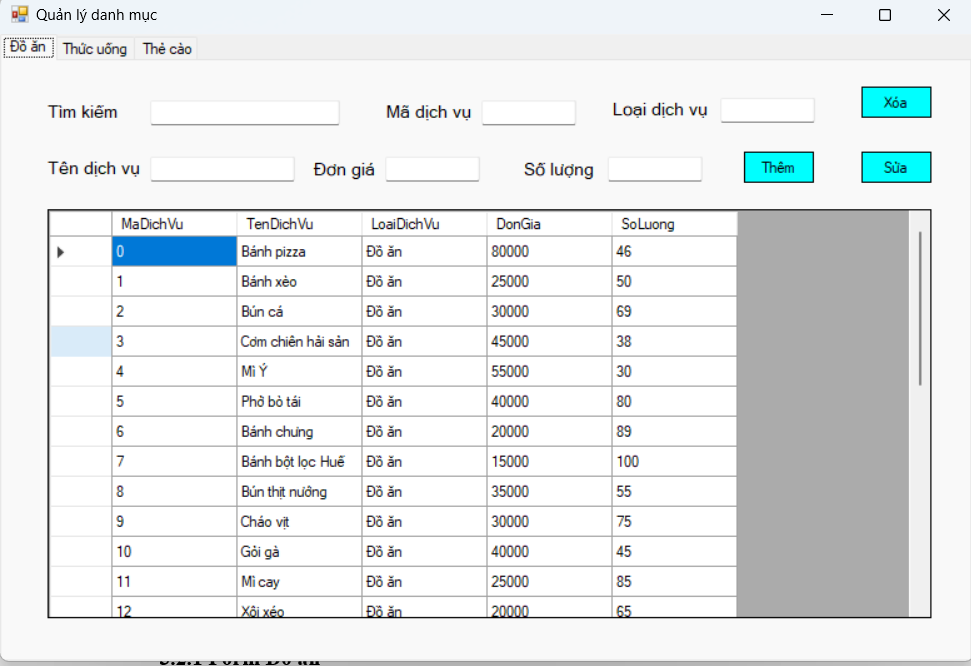
**  
Khi ấn dấu cộng ở tài khoản, hoặc máy tính đang có tài khoản đăng nhập sẽ hiện form này**

**5.3.1.9 Form Hóa đơn**   
**Hóa đơn chưa thanh toán và đã thanh toán**

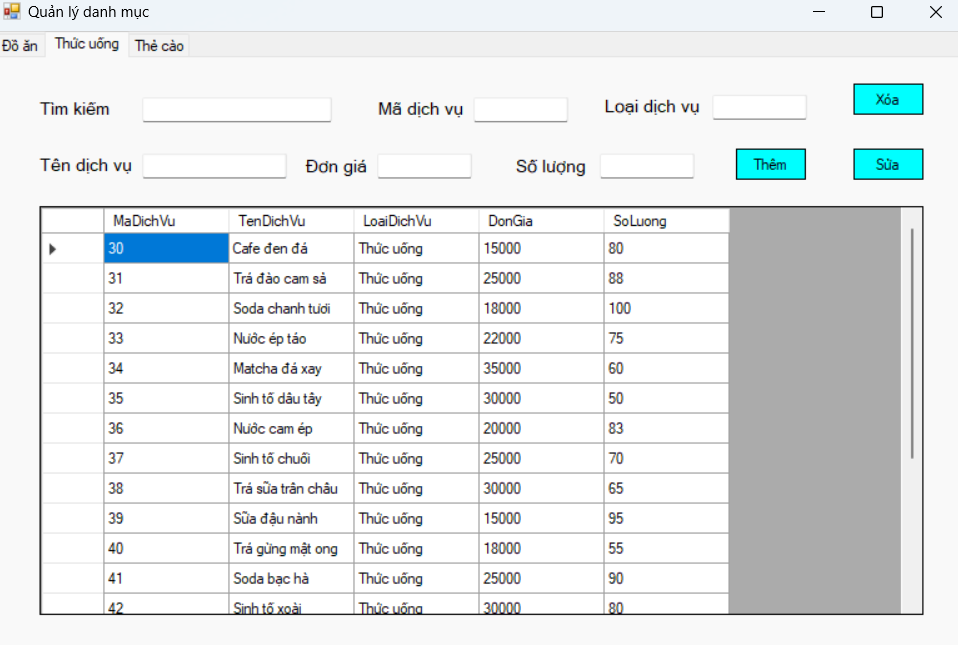
5.3.2 Form Quản lý danh mục

**Thêm xóa sửa tìm kiếm các loại dịch vụ**

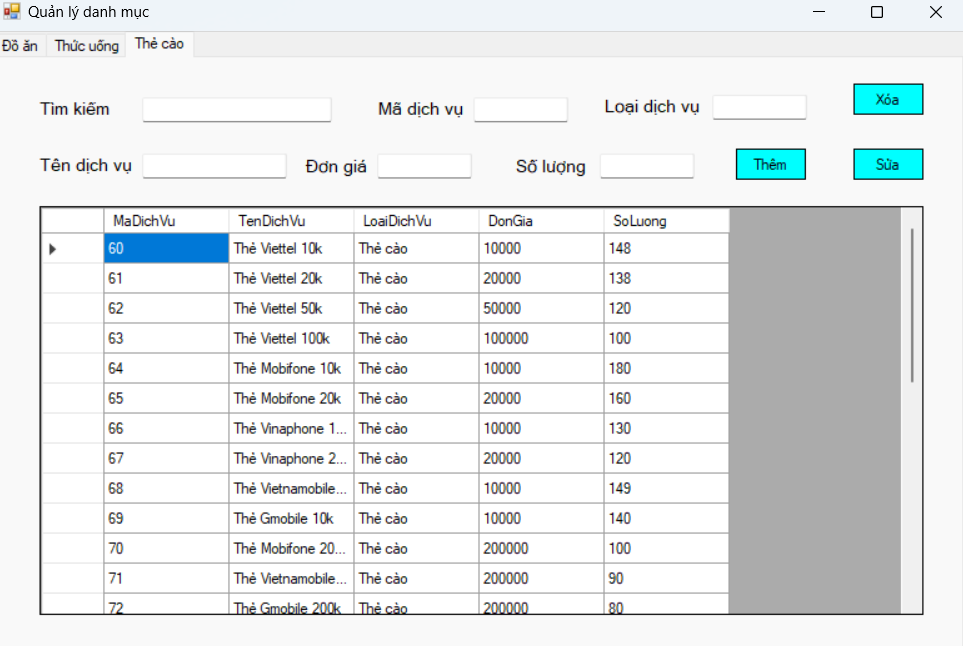
5.3.2.1 Form Đồ ăn



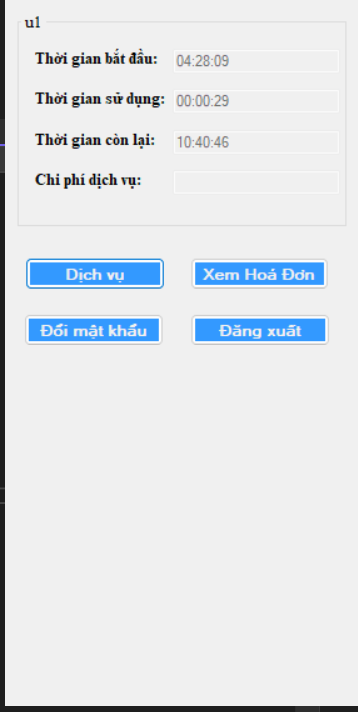
5.3.2.2 Form Thức uống



5.3.2.3 Form Thẻ cào

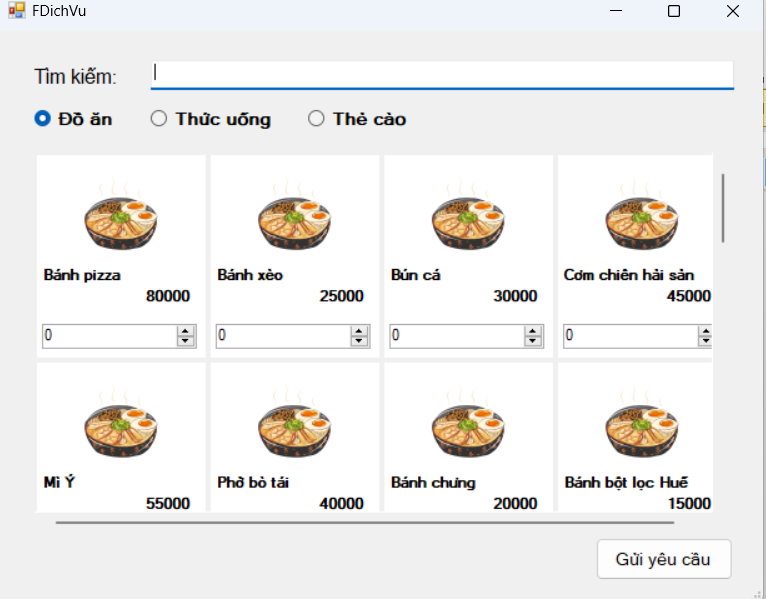


5.4 Form Khách hang

****

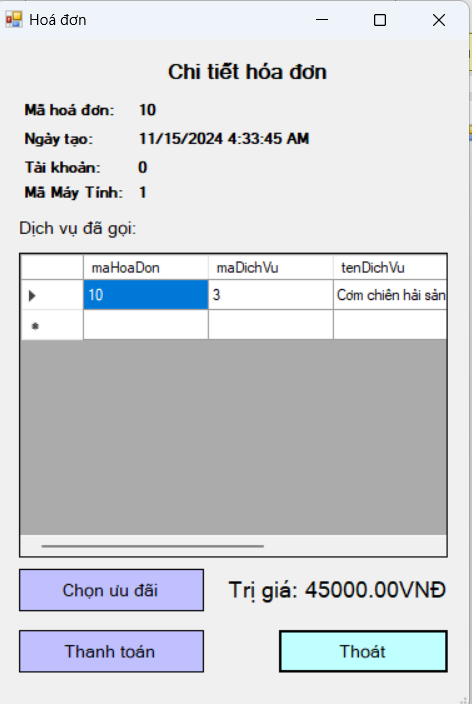
**Xuất hiện góc phải trên cùng bên phải**

5.4.1 Form Dịch vụ

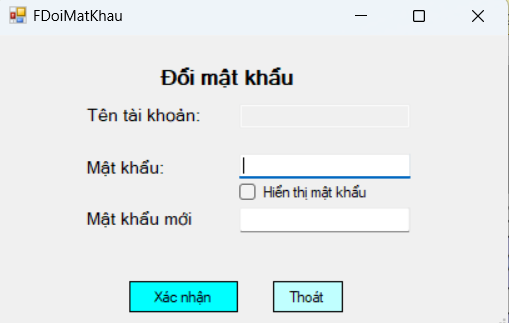
****

**Chọn loại dịch vụ và số lượng dịch vụ, bấm gửi yêu cầu máy chủ sẽ nhận được thông tin.**

5.4.2 Form Hóa đơn

**  
Khi gửi yêu cầu dịch vụ, hóa đơn sẽ được tạo. Được chọn các loại ưu đãi tùy vào loại tài khoản, có thể thanh toán trên hóa đơn hoặc ra quầy lễ tân**

5.4.3 Form Đổi mật khẩu

**  
Đổi mật khẩu tài khoản**