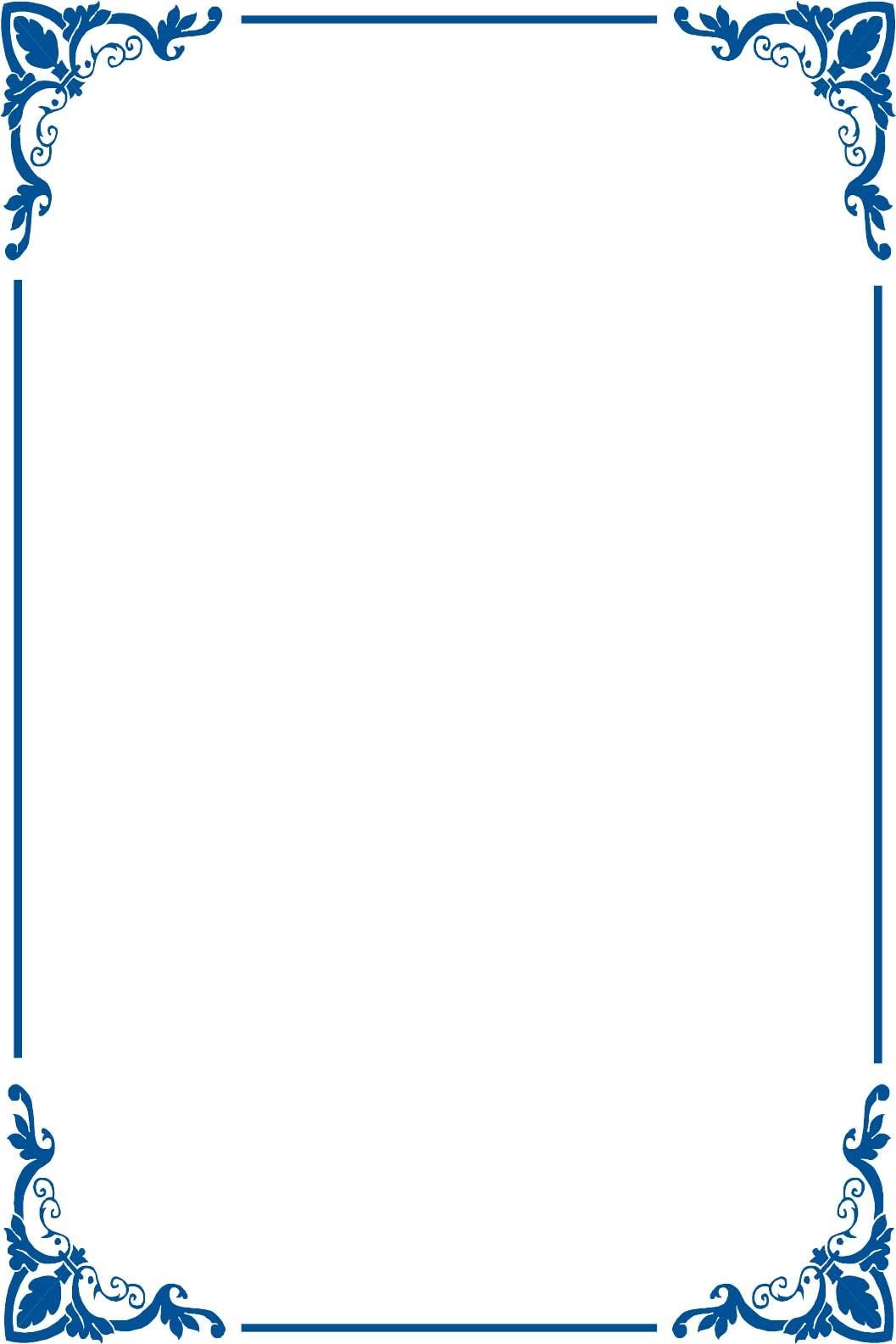
**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH**



**KHOA CÔNG NGHỆ THÔNG TIN**

**TIỂU LUẬN MÔN HỌC**

**HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

Đề tài:

**QUẢN LÝ CHO THUÊ PHÒNG   
VÀ DỊCH VỤ PHÒNG TRỌ**

**GVHD**: TS. Nguyễn Thành Sơn

**Mã LHP**: DBMS330284\_23\_2\_05CLC

**Nhóm thực hiện**: Nhóm 07

**Năm học**: 2023-2024

**Học kỳ: 2**

Thành phố Hồ Chí Minh, Tháng 11 năm 2023

**DANH SÁCH SINH VIÊN THAM GIA**

**Mã học phần:** DBMS330284\_23\_2\_05CLC.

Nhóm: 07.

Tên đề tài: *Quản lý cho thuê phòng và dịch vụ phòng trọ.*

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Tỉ lệ hoàn thành** |
| 1 | Hà Đăng Quang | 22110210 |  |
| 2 | Vũ Xuân Quang | 22110212 |  |
| 3 | Phan Hoàng An | 22110095 |  |
| 4 | Trần Đinh Gia Bảo | 22110111 |  |

Ghi chú:

Tỷ lệ %: Mức độ phần trăm hoàn thành của từng sinh viên tham gia.

**Trưởng nhóm:** Hà Đăng Quang

Nhận xét của giáo viên

...………………………………………………………………………………………………………………………………………………………………………....………………………………………………………………………………………………………………………………………………………………………

Tp. Hồ Chí Minh - Tháng … năm 2023

**Mục lục**

[PHẦN MỞ ĐẦU 11](#_Toc165293559)

[1. Lời mở đầu 11](#_Toc165293560)

[2. Mục đích, nhiệm vụ của đề tài 11](#_Toc165293561)

[3. Phạm vi thực hiện đề tài 12](#_Toc165293562)

[CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG 13](#_Toc165293563)

[1.1. Đặc tả đề tài 13](#_Toc165293564)

[1.2. Nghiệp vụ của bài toán 14](#_Toc165293565)

[1.2.1. Quá trình cho thuê phòng 14](#_Toc165293566)

[1.2.2. Quản lý thành viên và người đại diện trong phòng 15](#_Toc165293567)

[1.2.3. Quản lý nội thất mỗi phòng 15](#_Toc165293568)

[1.2.4. Quản lý dịch vụ phòng trọ 16](#_Toc165293569)

[1.2.5. Thu tiền hàng tháng 16](#_Toc165293570)

[1.2.5. Quản lý đăng ký tạm trú, tạm vắng và lưu trú 17](#_Toc165293571)

[1.2.6. Quá trình trả phòng 18](#_Toc165293572)

[1.3. Mô tả chức năng của bài toán 19](#_Toc165293573)

[CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 21](#_Toc165293574)

[2.1. Thiết kế cơ sở dữ liệu mức quan niệm 21](#_Toc165293575)

[2.2. Thiết kế cơ sở dữ liệu mức logic 22](#_Toc165293576)

[2.3. Các ràng buộc cần có 22](#_Toc165293577)

[2.4. Cài đặt các CSDL và các ràng buộc 28](#_Toc165293578)

[2.4.1. Bảng NguoiThueTro 28](#_Toc165293579)

[2.4.2. Bảng LoaiNoiThat 29](#_Toc165293580)

[2.4.3. Bảng DichVu 29](#_Toc165293581)

[2.4.4. Bảng LoaiPhongTro 29](#_Toc165293582)

[2.4.5. Bảng PhongTro 30](#_Toc165293583)

[2.4.6. Bảng DienNuoc 30](#_Toc165293584)

[2.4.7. Bảng HopDong 30](#_Toc165293585)

[2.4.8. Bảng NoiThat 31](#_Toc165293586)

[2.4.9. Bảng ChiTietDichVu 31](#_Toc165293587)

[2.4.10. Bảng HoaDon 31](#_Toc165293588)

[2.5. Các view 34](#_Toc165293589)

[2.5.1. View người thuê trọ 34](#_Toc165293590)

[2.5.2. View dịch vụ 34](#_Toc165293591)

[2.5.3. View phòng trọ 35](#_Toc165293592)

[2.5.4. View nội thất 36](#_Toc165293593)

[2.5.5. View chi tiết dịch vụ 37](#_Toc165293594)

[2.6. Các Trigger 37](#_Toc165293595)

[2.6.1. Trigger số người tối đa ở một phòng là 4 37](#_Toc165293596)

[2.6.2. Trigger tính tiền điện nước thông qua số điện nước số nước, 1 số điện 3000 VNĐ, 1 khối nước 6000 VNĐ 38](#_Toc165293597)

[2.6.3. Trigger nếu mã phòng trọ null thì trạng thái nội thất là 0 39](#_Toc165293598)

[2.6.4. Trigger tính tổng tiền 39](#_Toc165293599)

[2.6.5. Trigger kiểm tra dịch vụ đã tồn tại thì không được insert vào: 39](#_Toc165293600)

[2.6.6. Trigger kiểm tra thời điểm đăng ký phải lớn hơn hoặc bằng thời điểm người thuê dọn vào 40](#_Toc165293601)

[2.6.7. Trigger cập nhật lại trạng thái phòng khi điều chỉnh trong bảng người thuê trọ 40](#_Toc165293602)

[2.6.8. Trigger kiểm tra ngày dọn vào và ngày tạo hợp đồng 41](#_Toc165293603)

[2.6.9. Trigger khi insert ngày tạo 41](#_Toc165293604)

[2.6.10. Trigger tạo hóa đơn 42](#_Toc165293605)

[2.6.11. Trigger xóa thông tin người thuê khi xóa hợp đồng 43](#_Toc165293606)

[CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG 43](#_Toc165293607)

[3.1. Kết nối cơ sở dữ liệu 43](#_Toc165293608)

[Lớp DAL 43](#_Toc165293609)

[Lớp BAL 43](#_Toc165293610)

[Lớp Presentation 44](#_Toc165293611)

[3.2. Các thủ tục về hoá đơn 45](#_Toc165293612)

[3.2.1. Tạo hoá đơn 45](#_Toc165293613)

[3.2.2. Thêm hoá đơn 45](#_Toc165293614)

[3.2.3. Xóa hoá đơn 46](#_Toc165293615)

[3.2.4. Cập nhật hoá đơn 46](#_Toc165293616)

[3.3. Các thủ tục về loại phòng trọ 47](#_Toc165293617)

[3.3.1. Show loại phòng trọ 47](#_Toc165293618)

[3.3.2. Thêm loại phòng trọ 47](#_Toc165293619)

[3.3.3. Xóa loại phòng trọ 48](#_Toc165293620)

[3.3.4. Cập nhật loại phòng trọ 48](#_Toc165293621)

[3.4. Các thủ tục về phòng trọ 50](#_Toc165293622)

[3.4.1. Show phòng trọ 50](#_Toc165293623)

[3.4.2. Thêm phòng trọ 50](#_Toc165293624)

[3.4.3. Xóa phòng trọ 51](#_Toc165293625)

[3.4.4. Cập nhật phòng trọ 51](#_Toc165293626)

[3.5. Các thủ tục về điện nước 52](#_Toc165293627)

[3.5.1. Show điện nước 52](#_Toc165293628)

[3.5.2. Thêm điện nước 53](#_Toc165293629)

[3.5.3. Xóa điện nước 54](#_Toc165293630)

[3.5.4. Cập nhật điện nước 54](#_Toc165293631)

[3.6. Các thủ tục về loại nội thất 56](#_Toc165293632)

[3.6.1. Show loại nội thất 56](#_Toc165293633)

[3.6.2. Thêm loại nội thất 56](#_Toc165293634)

[3.6.3. Xóa loại nội thất 56](#_Toc165293635)

[3.6.4. Cập nhật loại nội thất 57](#_Toc165293636)

[3.7. Các hàm, thủ tục về nội thất 58](#_Toc165293637)

[3.7.1. Show nội thất 58](#_Toc165293638)

[3.7.2. Thêm nội thất 58](#_Toc165293639)

[3.7.3. Xóa nội thất 59](#_Toc165293640)

[3.7.4. Cập nhật nội thất 59](#_Toc165293641)

[3.8. Các hàm, thủ tục về dịch vụ 61](#_Toc165293642)

[3.8.1. Show dịch vụ 61](#_Toc165293643)

[3.8.2. Thêm dịch vụ 61](#_Toc165293644)

[3.8.3. Xóa dịch vụ 61](#_Toc165293645)

[3.8.4. Cập nhật dịch vụ 62](#_Toc165293646)

[3.9. Các thủ tục về người thuê trọ 63](#_Toc165293647)

[3.9.1. Show người thuê trọ 63](#_Toc165293648)

[3.9.2. Tìm kiếm theo mã người thuê trọ 63](#_Toc165293649)

[3.9.3. Show chi tiết người thuê trọ theo mã 64](#_Toc165293650)

[3.9.4. Thêm người thuê trọ 64](#_Toc165293651)

[3.9.5. Xoá người thuê trọ 65](#_Toc165293652)

[3.9.6. Cập nhật người thuê trọ 65](#_Toc165293653)

[3.10. Các thủ tục về hợp đồng 66](#_Toc165293654)

[3.10.1. Show hợp đồng 66](#_Toc165293655)

[3.10.2. Thêm hợp đồng 67](#_Toc165293656)

[3.10.3. Xóa hợp đồng 67](#_Toc165293657)

[3.10.4. Cập nhật hợp đồng 68](#_Toc165293658)

[3.11. Các hàm, thủ tục về chi tiết dịch vụ 69](#_Toc165293659)

[3.11.1. Show chi tiết dịch vụ 69](#_Toc165293660)

[3.11.2. Thêm chi tiết dịch vụ 69](#_Toc165293661)

[3.12. Các hàm về chi tiết dịch vụ 73](#_Toc165293662)

[3.12.1. Kiểm tra dịch vụ phòng đang có hay không (Sử dụng trong trigger) 73](#_Toc165293663)

[3.12.2. Tính tổng tiền dịch vụ của phòng tại một thời điểm nào đó 73](#_Toc165293664)

[3.13. Các hàm về loại phòng trọ 74](#_Toc165293665)

[3.13.1. Lấy mã loại phòng trọ lớn nhất đang có (Sử dụng trong proc) 74](#_Toc165293666)

[3.14. Các hàm về người thuê trọ 74](#_Toc165293667)

[3.14.1. Lấy mã người thuê lớn nhất đang có (Sử dụng trong proc) 74](#_Toc165293668)

[3.15. Các hàm về hợp đồng 75](#_Toc165293669)

[3.15.1. Số người đang ở phòng hiện tại (Sử dụng trong trigger giới hạn người thuê) 75](#_Toc165293670)

[3.15.2. Kiểm tra có tồn tại hợp đồng của phòng nào đó (Sử dụng trong trigger cập nhật trạng thái phòng trọ) 75](#_Toc165293671)

[3.15.3. Ngày tạo hợp đồng nhỏ nhất của phòng đó (Sử dụng trong trigger) 75](#_Toc165293672)

[3.15.4. Kiểm tra ở một thời điểm phòng trọ có đang được thuê (Sử dụng trong trigger tính tổng tiền dịch vụ) 75](#_Toc165293673)

[3.16. Các hàm về điện nước 76](#_Toc165293674)

[3.16.1. Hàm đếm số lượng phòng trong bảng điện nước (Sử dụng cho trigger cập nhật điện nước và trong các function khác) 76](#_Toc165293675)

[3.16.2. Hàm lấy thời điểm mới nhất theo mã phòng của điện nước (Sử dụng cho procedure lấy thời điểm tiếp theo) 76](#_Toc165293676)

[3.16.3. Lấy tiền điện 77](#_Toc165293677)

[3.16.4. Lấy tiền nước 77](#_Toc165293678)

[3.16.5. Lấy số điện cũ (Sử dụng trong trigger cập nhật điện nước) 78](#_Toc165293679)

[3.16.6. Lấy số nước cũ (Sử dụng trong trigger cập nhật điện nước) 78](#_Toc165293680)

[3.17. Thống kê chi phí phòng trọ theo tháng 79](#_Toc165293681)

[CHƯƠNG 4. TẠO USER VÀ PHÂN QUYỀN 79](#_Toc165293682)

[CHƯƠNG 5: THIẾT KẾ GIAO DIỆN HỆ THỐNG 83](#_Toc165293683)

[5.1. Giao điện đăng nhập 83](#_Toc165293684)

[5.2. Giao điện của tài khoản Admin 85](#_Toc165293685)

[5.2.1. Giao diện trang chủ 85](#_Toc165293686)

[5.2.2. Giao diện cài đặt hệ thống 86](#_Toc165293687)

[5.2.3. Giao diện quản lý phòng trọ 87](#_Toc165293688)

[5.2.4. Giao diện quản lý loại phòng trọ 88](#_Toc165293689)

[5.2.5. Giao diện quản lý nội thất 89](#_Toc165293690)

[5.2.6. Giao diện quản lý loại nội thất 89](#_Toc165293691)

[5.2.7. Giao diện quản lý người thuê trọ 90](#_Toc165293692)

[5.2.8. Giao diện quản lý điện nước 91](#_Toc165293693)

[5.2.9. Giao diện quản lý dịch vụ 91](#_Toc165293694)

[5.2.10. Giao diện quản lý chi tiết dịch vụ 92](#_Toc165293695)

[5.2.11. Giao diện quản lý hợp đồng 93](#_Toc165293696)

[5.2.12. Giao diện quản lý hóa đơn 93](#_Toc165293697)

**LỜI CẢM ƠN**

Kính gửi thầy Nguyễn Thành Sơn!

Nhóm chúng em xin gửi lời cảm ơn chân thành đến thầy về sự hướng dẫn và sự hỗ trợ mà thầy đã dành cho chúng em trong quá trình hoàn thiện đồ án môn Hệ quản trị cơ sở dữ liệu. Được thầy hướng dẫn và hỗ trợ, chúng em đã có thể hoàn thành đồ án một cách hiệu quả và đạt được kết quả mà chúng tôi mong đợi. Chúng em rất biết ơn thời gian và sự tâm huyết mà thầy đã dành để truyền đạt kiến thức và kinh nghiệm cho chúng em.

Nhờ sự chỉ dẫn của thầy, chúng em đã tích luỹ được những kiến thức và kỹ năng quan trọng để làm việc với cơ sở dữ liệu một cách hiệu quả. Chúng em cam kết sử dụng những kiến thức này để phát triển bản thân và góp phần vào công việc và học tập trong tương lai.

Một lần nữa, chúng em xin bày tỏ lòng biết ơn sâu sắc đối với sự giúp đỡ và hướng dẫn tận tâm của thầy Nguyễn Thành Sơn trong quá trình học tập của chúng em. Trân trọng và cảm ơn thầy rất nhiều.

# PHẦN MỞ ĐẦU

## 1. Lời mở đầu

Trong bối cảnh xã hội hiện nay, vấn đề về nhu cầu thuê trọ trở ngày càng trở nên phổ biến và cấp thiết hơn bao giờ hết. Với sự gia tăng đáng kể về số lượng người thuê trọ, đặc biệt là ở những đô thị lớn những người chủ phải đối mặt với thách thức của việc thay đổi cách quản lý để có thể giảm thiểu chi phí, quản lý hiệu quả và chính xác. Sự phát triển mạnh mẽ của công nghệ thông tin đã cung cấp những phần mềm giúp cho việc quản lý khu trọ một cách hiệu quả và chính xác hơn so với quản lý bằng sổ sách.

Nắm bắt được nhu cầu của người quản lý trọ và được sự cho phép và hướng dẫn của thầy Nguyễn Thành Sơn, nhóm chúng em đã quyết định chọn đề tài "Phần mềm quản lý phòng trọ" cho đề tài cuối kỳ môn Hệ quản trị cơ sở dữ liệu.

## 2. Mục đích, nhiệm vụ của đề tài

Quản lý cho thuê phòng và dịch vụ của phòng trọ đòi hỏi một hệ thống quản lý hiệu quả, thuận tiện và chặt chẽ. Đề tài này không chỉ có giá trị trong thực tế mà còn đóng góp vào quá trình học thêm được các kiến thức nghiệp vụ của sinh viên chúng em. Sau khi hoàn thành dự án, chúng em đã củng cố kiến thức cơ bản của môn học và cũng học hỏi thêm về nghiệp vụ quản lý phòng trọ trong môi trường doanh nghiệp.

Trong phạm vi đề tài của môn học này, mục tiêu chính của chúng em là phát triển một hệ thống quản lý có khả năng thực hiện các nhiệm vụ cơ bản, bao gồm quản lý người thuê, quản lý việc cho thuê và trả phòng,quản lý thu chi hàng tháng của phòng, cung cấp và quản lý việc cho thuê các dịch vụ theo yêu cầu của người thuê.

## 3. Phạm vi thực hiện đề tài

Đề tài tập trung vào việc quản lý những khu trọ vừa và nhỏ, vì vậy những tính năng của phần mềm sẽ tập trung vào những tính năng hỗ trợ cho những nghiệp vụ chính trong việc quản lý phòng trọ mà không đi sâu vào những tính năng phụ khác.

# CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG

## 1.1. Đặc tả đề tài

Một khu trọ cần một hệ thống quản lý. Hệ thống quản lý dành cho hai đối tượng người dùng: Người quản lý khu trọ và người ở mỗi phòng trọ. Với người quản lý khu trọ, người quản lý sẽ có các chức năng: Quản lý thông tin của người thuê phòng, quản lý dịch vụ mỗi phòng, quản lý nội thất mỗi phòng, quản lý việc cho thuê và trả phòng, quản lý thu tiền hàng tháng, thống kê tiền thu mỗi tháng. Với người đại diện mỗi phòng sẽ có các chức năng: Xem tiền phải đóng hàng tháng, xem thông tin thành viên mỗi phòng, Xem nội thất được trang bị cho phòng

Khu trọ sẽ có nhiều **phòng trọ**, mỗi phòng trọ sẽ được phân biệt bằng **Mã phòng trọ**. Phòng trọ sẽ có hai **trạng thái** là đã thuê và chưa thuê. **Mỗi phòng trọ** sẽ thuộc về **một** **loại phòng trọ** và mỗi loại phòng trọ sẽ có **nhiều phòng trọ**.Mỗi loại phòng trọ sẽ được phân biệt bằng **mã loại phòng trọ** và sẽ gồm những thông tin: diện tích phòng, giá tiền, tiền cọc.

Mỗi phòng trọ thì sẽ được trang bị nhiều **nội thất** và mỗi nội thất chỉ có thể được trang bị cho một phòng trọ tại một thời điểm. Mỗi nội thất sẽ được phân biệt qua **mã nội thất**, và sẽ có những thông tin: Tên nội thất, NSX, giá tiền. Trong quá trình sử dụng, nội thất có thể bị hư hỏng, vì vậy người quản lý cần biết **trạng thái** của nội thất trong quá trình người thuê sử dụng.

Để dễ dàng quản lý các nội thất, người quản lý cần lưu trữ thông tin của **loại nội thất**, mỗi loại nội thất sẽ có nhiều nội thất, mỗi nội thất chỉ thuộc về một loại nội thất. Mỗi loại nội thất sẽ được phân biệt qua **mã loại nội thất** và bao gồm thông tin tên loại nội thất.

Mỗi **người thuê** có thể thuê được nhiều phòng và mỗi phòng có thể được nhiều người thuê. Để quản lý được người thuê phòng, người quản lý cần những thông tin: **Mã người thuê,** họ, tên lót, tên, số CCCD, quê quán, số điện thoại, nghề nghiệp. Mỗi người thuê sẽ có thể vào ở và dọn đi ở mỗi thời điểm khác nhau, vì vậy người quản lý cần lưu thông tin **ngày dọn đi** và **ngày vào ở** của người thuê. Mỗi phòng sẽ bao gồm một người đại diện, để phân biệt được người chủ cần thông tin **vai trò** của mỗi người thuê.

Người đại diện mỗi phòng sẽ đại diện để ký **hợp đồng thuê trọ**. Hợp đồng sẽ gồm những thông tin: **Ngày tạo hợp đồng**, tiền cọc.

Người thuê khi thuê phòng sẽ có thể chọn thêm **dịch vụ** cho phòng và mỗi dịch vụ có thể cung cấp cho nhiều phòng. Mỗi dịch vụ bao gồm những thông tin: **Mã dịch vụ**, tên dịch vụ, chi phí dịch vụ.

Khi người thuê lựa chọn những dịch vụ cho phòng sẽ có **chi tiết dịch vụ.** Chi tiết dịch gồm những thông tin: **Thời điểm dăng ký** và **thời điểm ngưng sử dụng**.

Hàng tháng, mỗi phòng sẽ phát sinh tiền **điện nước.** Điện nước bao gồm những thông tin: Số điện hiện tại, số nước hiện tại, thời điểm phát sinh, tiền nước, tiền điện.

## 1.2. Nghiệp vụ của bài toán

Việc cho thuê phòng trọ hiện nay đang rất phổ biến, đặc biệt là ở các thành phố lớn, việc quản lý một khu trọ cần phải được chia nhỏ thành nhiều phần: **Quá trình người thuê tiến hành tiếp cận phòng trọ và thuê phòng**, **quản lý phòng trọ trong lúc thuê phòng, Quá trình trả phòng**.

### 1.2.1. Quá trình cho thuê phòng

**Thời điểm trọ trống:** Người quản lý sẽ treo bảng trọ trống ở trước khu trọ hoặc đăng bài cho thuê trên các nền tảng mạng xã hội.

**Thời điểm người thuê trọ tiếp cận được thông tin trọ trống:** Người thuê trọ sau khi tiếp cận được thông tin khu trọ đang có phòng trống thì sẽ liên lạc với người quản lý ở khu đó để trao đổi những thông tin về phòng trọ.

**Người thuê muốn xem phòng trực tiếp:** Người thuê trọ muốn xem phòng trực tiếp thì sẽ đến khu trọ và gặp người quản lý, người quản lý sẽ cho xem phòng. Trong quá trình xem người thuê sẽ có thể trao đổi trực tiếp với người chủ những thông tin xoay quanh phòng trọ.

**Người thuê muốn thuê trọ:** Người thuê muốn thuê trọ thì sẽ tiến hành xem hợp đồng mà người quản lý cung cấp. Hợp đồng sẽ bao gồm thông tin của bên thuê và bên cho thuê, các ràng buộc và điều kiện trong lúc thuê trọ. Nếu cả hai bên đều thống nhất với những điều kiện trong hợp đồng trong lúc trao đổi thì sẽ tiến hành kí vào hợp đồng.

**Tiến hành đặt cọc:** Sau khi kí hợp đồng người thuê tiến hành đặt cọc cho phòng, tiền cọc có giá tương ứng với 1 tháng tiền phòng.

### 1.2.2. Quản lý thành viên và người đại diện trong phòng

Mỗi phòng cần có một người đại diện, ban đầu, người ký hợp đồng cũng sẽ là người đại diện, trường hợp người đại diện dọn đi thì phòng phải bầu lại người đại diện mới.

**Mỗi phòng** chỉ gồm tối đa **4 ngườ**i ở.

### 1.2.3. Quản lý nội thất mỗi phòng

Với nội thất, người thuê trong quá trình thuê sẽ có **hai lựa chọn:** Chọn đầy đủ nội thất cho phòng hoặc phòng trống. Với phòng đầy đủ nội thất, những nội thất trong phòng bao gồm: **Máy lạnh, tủ lạnh, bếp từ, giường, điều hòa.** Mỗi loại nội thất sẽ được người chủ nhập từ ngoài về và **đánh mã phân biệt** khi trang bị cho mỗi phòng, mỗi loại nội thất sẽ có giá tiền riêng. Với phòng đầy đủ nội thất, người thuê phải đóng thêm 500.000 VNĐ / tháng tiền phòng. Với phòng trống, giá tiền phòng được giữ như cũ.

Trong quá trình thuê, nếu người thuê gặp tình trạng hư hỏng đồ nội thất thì người thuê sẽ báo cho người quản lý. Nếu việc hư hỏng không do người thuê gây ra thì người quản lý sẽ tiến hành bảo trì, nếu việc hư hỏng là lỗi do người thuê thì người thuê phải đền bù.

### 1.2.4. Quản lý dịch vụ phòng trọ

**Bảng giá dịch vụ của khu trọ**

|  |  |
| --- | --- |
| Tên dịch vụ | Giá tiền / Tháng |
| Wifi | 100.000 VNĐ |
| Giặc sấy | 50.000 VNĐ |
| Giữ xe: | 50.000 VNĐ |
| Vệ sinh phòng | 50.000 VNĐ |
| Rác | 20.000 VNĐ |

Người thuê sẽ được lựa chọn các loại dịch vụ trên trong quá trình thuê trọ. Tiền dịch vụ sẽ được tính theo hàng tháng.

### 1.2.5. Thu tiền hàng tháng

**Bảng giá các khoảng tiền của khu trọ**

|  |  |
| --- | --- |
| **Tên khoảng tiền** | **Giá tiền** |
| Tiền phòng | 18m^2: 1.800.000 VNĐ / tháng  25m^2: 2.500.500 VNĐ/ tháng |
| Tiền điện | 3.000 VNĐ / Kwh |
| Tiền nước | 6.000 VNĐ / m^3 |

Hàng tháng, người quản lý sẽ đi qua các phòng trong khu trọ để xem số điện, số nước, đối chiếu với số điện, nước cũ. Sau đó sẽ tiến hành tính các khoảng tiền.

**Tính các khoảng tiền:**

* Tiền điện = (Số điện hiện tại – số điện tháng trước) \* 3.000 (VNĐ)
* Tiền nước = (Số nước hiện tại – số nước tháng trước) \* 6.000 (VNĐ)
* Tiền phòng = tiền phòng cố định + 500.000 (Nếu trang bị đầy đủ nội thất) (VNĐ)
* Tổng tiền hàng tháng = Tiền điện + tiền nước + tiền dịch vụ + tiền phòng (VNĐ)

Sau đó người chủ sẽ gửi phiếu thu tiền của tháng đó cho người thuê phòng qua các hình thức: Trực tiếp, nhắn tin zalo, messenger, …

### 1.2.5. Quản lý đăng ký tạm trú, tạm vắng và lưu trú

* Đăng ký tạm trú:

Việc đăng ký tạm trú sẽ được chính quyền địa phương cung cấp một phiếu đăng ký cho mỗi phòng và người thuê sẽ điền thông tin cá nhân của mình vào phiếu.

* Đăng ký tạm vắng:

Người quản lý mỗi ngày sẽ đi kiểm tra mỗi khu trọ 2 lần sáng và tối. Nếu trong quá trình kiểm tra thấy phòng có người đi vắng và đúng điều kiện của việc đăng ký tạm vắng thì người quản lý sẽ làm hồ sơ đăng ký tạm vắng cho người ở phòng.

* Đăng ký lưu trú:

Việc đăng ký lưu trú sẽ diễn tra khi người thuê có cho người quen ở lại qua đêm hoặc một vài ngày.

Người đang thuê sẽ phải cung cấp những thông tin của người ở lại qua đêm: số CCCD, hình ảnh mặt trước/sau CCCD. Người quản lý sẽ dùng những thông tin này để đăng ký lưu trú online.

### 1.2.6. Quá trình trả phòng

Khi trả phòng thì sẽ có 2 trường hợp:

- Trường hợp tất cả các thành viên trong phòng đều trả phòng (sau khi trả phòng thì phòng trống), lúc này các thành viên báo cho người quản lý, người quản lý sẽ xem xét thời gian, nếu người trả phòng vào giữa tháng thì phải chấp nhận đóng tiền cho tháng đó, tiền phòng lúc này sẽ được tính: Tiền điện nước (đầu tháng tời thời điểm hiện tại), tiền phòng, tiền dịch vụ. Người chủ sẽ trả lại tiền cọc ban đầu. Nếu các quá trình thanh toán hoàn thành thì sẽ chính thức trả phòng. Người quản lý sẽ tiến hành cho thuê lại phòng.

- Trường hợp nhỏ hơn hoặc bằng 3 thành viên trả phòng (sau khi trả thì còn lại người ở phòng), lúc này các thành viên chỉ cần báo cho người quản lý, các khoảng tiền sẽ do các thành viên trong phòng tự trao đổi với nhau. Sau khi chính thức trả phòng, người quản lý sẽ xóa đi thông tin người rời đi, nếu người đi bao gồm cả người đại diện thì người quản lý phải bầu lại người đại diện cho phòng.

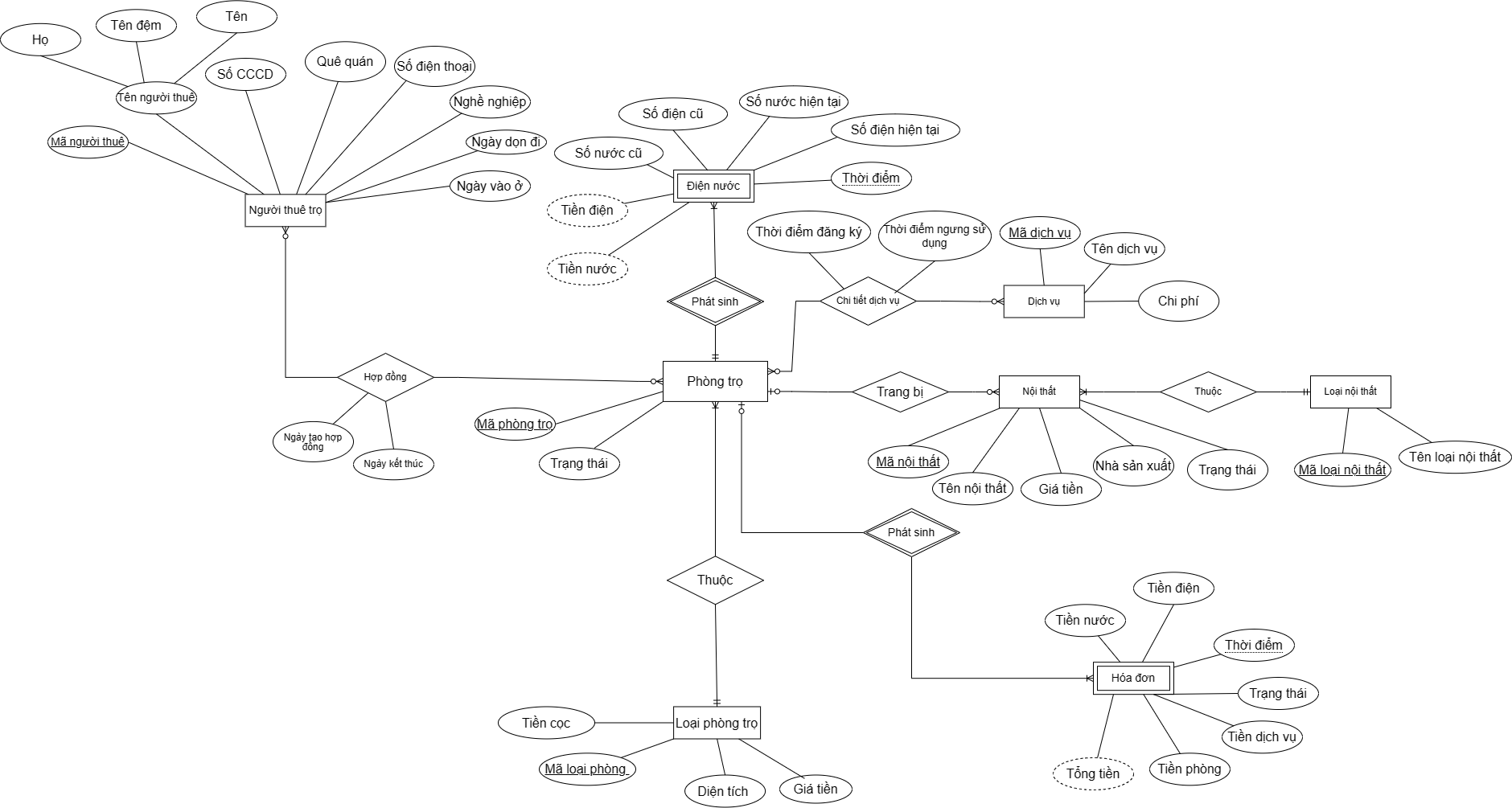
## 1.3. Mô tả chức năng của bài toán

Đối với người quản lý

* **Quản lý thông tin người thuê:**
* Thêm, sửa đổi, và xóa thông tin người thuê.
* Tạo thông tin người thuê mỗi phòng.
* Tìm kiếm người thuê theo mã NT/CCCD,…
* **Quản lý thông tin phòng và loại phòng:** Thêm, sửa đổi, và xóa thông tin loại phòng và phòng cụ thể.
* **Quản lý nội thất:** 
  + Thêm, sửa đổi, và xóa thông tin nội thất.
  + Cập nhật tiền trang bị nội thất vào tổng tiền hàng tháng
* **Quản lý thông tin dịch vụ:** Thêm, sửa đổi, và xóa thông tin dịch vụ.
* **Quản lý đặt dịch vụ:**
* Ghi nhận thông tin sử dụng dịch vụ theo yêu cầu của khách hàng.
* Tra cứu thông tin đặt dịch vụ.
* Cập nhật tiền sử dụng dịch vụ vào tổng tiền hàng tháng.
* **Quản lý điện nước**
  + Thêm,sửa,xóa thông tin điện nước.
  + Cập nhật tiền điện nước vào tổng tiền hàng tháng.
* **Quản lý thu tiền hàng tháng:**
* Tính tổng tiền qua mỗi tháng mà khách hàng phải trả
* Xuất hóa đơn trình bày chi tiết các chi phí dịch vụ, tiền phòng, điện nước mà người thuê cần thanh toán
* Tra cứu thông tin tiền khách hàng cần trả qua mỗi tháng
* **Báo cáo và thống kê:**
* Tạo báo cáo về doanh thu của phòng trọ qua mỗi tháng, quý, năm.

# CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## 2.1. Thiết kế cơ sở dữ liệu mức quan niệm

****Từ mô tả về dữ liệu cần có ở phần mô tả của bài toán ta hình thành được sơ đồ thực thể kết hợp (ERD):

## 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:

1. NguoiThueTro(Manguoithuetro,Vaitro, Ngaydondi, Ngayvaoo, Ho, Tenlot, Ten, SoCCCD, Quequan, Sodt, Nghenghiep, Manguoidaidien)
2. LoaiNoiThat(Maloainoithat, Tenloainoithat)
3. DichVu(Madichvu,Tendichvu,Chiphi)
4. LoaiPhongTro(Maloaiphong,Dientich,Giatien)
5. PhongTro(Maphongtro,Maloaiphong, Trangthai,Songuoio)
6. DienNuoc(Thoidiem,Maphongtro,Sodienhientai,Sonuochientai,Tiendien,Tiennuoc)
7. HopDong(Manguoidaidien,Maphongtro, NgaytaoHD,Tiencoc)
8. NoiThat(Manoithat,Maphongtro,Maloainoithat, Giatien, Nhasanxuat,Trangthai)
9. ChiTietDichVu(Maphongtro,Madichvu,Thoidiemdangky,Thoidiemngungsudung)

## 2.3. Các ràng buộc cần có

|  |  |  |
| --- | --- | --- |
| STT | TABLE | RÀNG BUỘC |
| 1 | NguoiThueTro | * Ràng buộc khóa chính: Manguoithue * Ràng buộc not null: * Ho * Tenlot * Ten * SoCCCD * Sodt * Ràng buộc unique: * SoCCCD * Sodt * Ràng buộc check: * SoCCCD = 12 * Sodt = 10, không chứa kí tự đặc biệt (@#$,…) * Manguoithue > 0 * Ngaydondi is null hoặc Ngaydondi > Ngayvaoo * Ràng buộc default: * Nghenghiep = N’Không có' |
| 2 | LoaiNoiThat | * Ràng buộc khóa chính: Maloainoithat * Ràng buộc not null: * Tenloainoithat * Ràng buộc check: * Maloainoithat > 0 |
| 3 | DichVu | * Ràng buộc khóa chính: Madichvu * Ràng buộc not null: * Tendichvu * Chiphi * Ràng buộc unique: * Tendichvu * Ràng buộc check: * Chiphi >=0 và Madichvu > 0 |
| 4 | LoaiPhongTro | * Ràng buộc khóa chính: Maloaiphong * Ràng buộc not null: * Dientich * Giatien * Tiencoc * Ràng buộc check * Dientich > 0 * Giatien > 0 * Tiencoc > 0 |
| 5 | PhongTro | * Ràng buộc khóa chính: Maphongtro * Ràng buộc khóa ngoại: Maloaiphongtro tham chiếu đến bảng **LoaiPhongTro** * Ràng buộc check: * Maphongtro > 0 |
| 6 | DienNuoc | * Ràng buộc khóa chính: * Maphongtro * Thoidiem * Ràng buộc khoá ngoại: Maphongtro tham chiếu đến bảng **PhongTro** * Ràng buộc not null: * Sodienhientai * Sonuochientai * Ràng buộc check: * Thoidiem <= thời điểm hiện tại * Sodienhientai > 0 * Sonuochientai > 0 * Tiendien >= 0 * Tiennuoc >= 0 * Sodiencu <= Sodienhientai * Sonuoccu <= Sonuochientai |
| 7 | HopDong | * Ràng buộc khóa chính: * Manguoithue * Maphongtro * Ràng buộc khóa ngoại: * Manguoithue tham chiếu đến bảng **NguoiThueTro** * Maphongtro tham chiếu đến bảng **PhongTro** * Ràng buộc not null: * Ngaytaohd * Ngayketthuchopdong * Ràng buộc check: * Ngayketthuchopdong > Ngaytaohd * Ngaytaohopdong phải là ngày đầu tiên của tháng |
| 8 | NoiThat | * Ràng buộc khóa chính: Manoithat * Ràng buộc khóa ngoại: * Maphongtro tham chiếu đến bảng **PhongTro** * Maloainoithat tham chiếu đến bảng **LoaiNoiThat** * Ràng buộc not null: * Giatien * Ràng buộc check: * Giatien > 0 * Ràng buộc default: * Tennoithat = ‘No name’ |
| 9 | ChiTietDichVu | * Ràng buộc khóa chính: * Maphongtro * Madichvu * Thoidiemdangky * Ràng buộc khóa ngoại: * Maphongtro tham chiếu đến bảng **PhongTro** * Madichvu tham chiếu đến bảng **DichVu** * Ràng buộc check: * Thoidiemdangky phải là ngày đầu tiên của tháng |
| 10 | HoaDon | * Ràng buộc khóa chính: * Maphongtro * Thoidiem * Ràng buộc khóa ngoại: * Maphongtro tham chiếu đến bảng **PhongTro** * Ràng buộc not null: * Thoidiem * Tiendien * Tiennuoc * Tiendichvu * Tienphong * Ràng buộc check: * Tiendien >= 0 * Tiennuoc >= 0 * Tienphong >= 0 * Tongtien >= 0 * Ràng buộc default: * Trangthai = 0 |

## 2.4. Cài đặt các CSDL và các ràng buộc

### 2.4.1. Bảng NguoiThueTro

create table NguoiThueTro(

Manguoithue int primary key,

Ngayvaoo Date null, -- Nhập ngày tạo hợp đồng trước, Ngày vào ở sau, Ngày vào ở phải sau ngày tạo hợp đồng

Ngaydondi Date , --ngaydondi null

Ho Nvarchar(50) not null,

Tenlot Nvarchar(50) not null,

Ten Nvarchar(50) not null,

SoCCCD char(12) not null unique,

Quequan nvarchar(50), --Có thể not null được

Sodt char(10) not null unique,

Nghenghiep nvarchar(50) default(N'Không có'),

Constraint CHECK\_SoCCCD Check(

Len(SoCCCD) = 12

AND ISNUMERIC(SoCCCD) = 1

AND SoCCCD Not like '%[.]%' --Không chứa kí tự .

),

Constraint CHECK\_Sodt Check(

Len(Sodt) = 10

AND Sodt Not like '%[.]%'

),

Constraint CHECK\_Manguoithue Check(

Manguoithue > 0

),

-- Thêm: Check ngày dọn đi > ngày vào ở

Constraint CHECK\_Ngaydondi Check(

Ngaydondi is null --KIEM TRA

OR Ngaydondi > Ngayvaoo

)

--trigger moi phong chi co mot nguoi dai dien  
);

### 2.4.2. Bảng LoaiNoiThat

Create table LoaiNoiThat(

Maloainoithat int primary key,

Tenloainoithat nvarchar(50) not null,

Constraint CHECK\_Maloainoithat Check(

Maloainoithat > 0

)

)

### 2.4.3. Bảng DichVu

Create table DichVu(

Madichvu int primary key,

Tendichvu nvarchar(50) not null unique,

Chiphi int not null,

Constraint CHECK\_Value\_DichVu Check(

Chiphi >= 0 and Madichvu > 0

)

)

### 2.4.4. Bảng LoaiPhongTro

Create table LoaiPhongTro(

Maloaiphong int primary key,

Dientich int not null,

Giatien int not null,

Tiencoc int not null,

constraint CHECK\_Dientichduong Check ( Dientich > 0),

constraint CHECK\_Giatienduong Check (Giatien > 0),

constraint CHECK\_Tiencocduong Check(Tiencoc > 0)

)

### 2.4.5. Bảng PhongTro

Create table PhongTro(

Maphongtro int primary key check( Maphongtro > 0),

Maloaiphongtro int foreign key references LoaiPhongTro(MaloaiPhong) On update cascade On delete set null,

Trangthai bit ,

)

### 2.4.6. Bảng DienNuoc

create table DienNuoc(

Maphongtro int foreign key references PhongTro(Maphongtro) on update cascade,

Thoidiem date,

Sodienhientai int not null,

Sonuochientai int not null,

Sodiencu int ,

Sonuoccu int ,

Tiendien int ,

Tiennuoc int ,

constraint prim\_diennuoc primary key(Maphongtro,Thoidiem),

constraint check\_value\_diennuoc check(Thoidiem <= GETDATE() and Sodienhientai > 0 and Sonuochientai > 0 and Tiendien >= 0 and Tiennuoc >= 0 and Sodiencu <= Sodienhientai and Sonuoccu <= Sonuochientai), -- Thoi diem xuat bill phai truoc hoac la ngay hom nay

-- trigger sodien nuuoc thang sau phai nho hon so dien nuoc thang dau

-- trigger tính số điện nước

)

### 2.4.7. Bảng HopDong

create table HopDong(

Manguoithue int references NguoiThueTro(Manguoithue) on update cascade, --foreign key on update cascade

Maphongtro int references PhongTro(Maphongtro) on update cascade, --foreign key on update cascade

Ngaytaohd date not null, -- trigger: datediff(day, Ngaydonvao --cua nguoidai dien, Ngaytaohd) <= 7

Ngayketthuchopdong date null, -- quy định Ngaydondi < Ngayketthuchopdong: matcoc

constraint prim\_hopdong primary key(Manguoithue,Maphongtro),

constraint check\_day\_Hopdong check(Ngayketthuchopdong > Ngaytaohd),

constraint check\_ngay\_tao\_hopdong check( datepart(day,Ngaytaohd) = 1 )

--procedure: khi nhap phong tro thi update trang thai phong  
)

### 2.4.8. Bảng NoiThat

create table NoiThat

(

Manoithat int primary key,

Tennoithat nvarchar(50) default('No name'),

Maphongtro int references PhongTro(Maphongtro) on update cascade on delete set null , -- foreign key on update cascade,set null

Maloainoithat int references LoaiNoiThat(Maloainoithat) on update cascade on delete set null, -- foreign key on update cascade,set null

Giatien int not null,

Nhasanxuat nvarchar(50),

Trangthai bit , -- trigger if(Maphongtro = NULL) set 0

constraint check\_giatien check(Giatien > 0)

)

### 2.4.9. Bảng ChiTietDichVu

create table ChiTietDichVu(

Maphongtro int references PhongTro(Maphongtro) on update cascade, -- foreign key on update cascade

Madichvu int references DichVu(Madichvu) on update cascade, -- foreign key on update cascade

Thoidiemdangky date,

Thoidiemngungsudung date, --

constraint prim\_chitietdichvu primary key(Maphongtro,Madichvu,Thoidiemdangky),

constraint check\_ngaydangkydichvu check (datepart(day,Thoidiemdangky) = 1),

constraint check\_ngayhuydichvu check ((datepart(day,Thoidiemngungsudung) = 1 and Thoidiemngungsudung > Thoidiemdangky)or Thoidiemngungsudung = null)  
)

### 2.4.10. Bảng HoaDon

create table HoaDon

(

Thoidiem date not null,

Maphongtro int references PhongTro(Maphongtro) on update cascade,

Tiendien int NOT NULL, -- trigger

Tiennuoc int NOT NULL, -- trigger tinh

Trangthai bit default(0), --Khi tao hoa don mac dinh nguoi thue chua thanh toan

Tiendichvu int NOT NULL, -- trigger check mien gia tri

Tienphong int NOT NULL, -- trigger check mien gia tri

Tongtien int , --trigger tinh tong tien tu cac khoang khac

constraint CHECK\_Value\_HoaDon CHECK(Tiendien >= 0 and Tiennuoc >= 0 and Tiendichvu >= 0 and Tienphong >= 0 and Tongtien >= 0),

primary key(Thoidiem,Maphongtro)

-- trigger tự động cập nhật số tiền điện, nước, số điện, số nước, khi nhập vào bảng điện nước, tiền dịch vụ, tiền phòng, thì bảng hóa đơn tự động cập nhật

-- trigger tính tổng tiền thông qua các khoảng tiền   
)

**DIAGRAM**

****

## 2.5. Các view

### 2.5.1. View người thuê trọ

**Xem danh sách người thuê trọ hiện tại**

Create View View\_Ds\_NguoiThueTro\_HienTai As

Select \* From NguoiThueTro Where Ngaydondi is null Or Ngaydondi >= GETDATE();

**Thực hiện trên C#**

public DataSet ShowNguoiThueHienTaiTheoMa(int maPhongTro)

{

return db.ExecuteQueryDataSet("select \* from View\_Ds\_NguoiThueTro\_HienTai where Maphongtro=@maPhongTro", CommandType.Text, new SqlParameter("@maPhongTro", maPhongTro));

}

### 2.5.2. View dịch vụ

**Xem các dịch vụ đang được trang bị trong phòng trọ**

create view [dbo].[View\_Ds\_DichVuDangTrangBi]

as

SELECT dbo.ChiTietDichVu.Maphongtro, dbo.ChiTietDichVu.Madichvu, dbo.ChiTietDichVu.Thoidiemdangky, dbo.DichVu.Tendichvu, dbo.DichVu.Chiphi

FROM dbo.ChiTietDichVu INNER JOIN

dbo.DichVu ON dbo.ChiTietDichVu.Madichvu = dbo.DichVu.Madichvu

WHERE (dbo.ChiTietDichVu.Thoidiemngungsudung IS NULL or (datepart(year, dbo.ChiTietDichVu.Thoidiemngungsudung) <= datepart(year, GETDATE()) and datepart(month, dbo.ChiTietDichVu.Thoidiemngungsudung) <= datepart(month, GETDATE())))

**Thực hiện trên C#**

public DataSet DanhSachDichVuDangTrangBiChoTro(int maPhongTro)

{

return db.ExecuteQueryDataSet("select \* from View\_Ds\_DichVuDangTrangBi where Maphongtro = @Maphongtro", CommandType.Text, new SqlParameter("@Maphongtro", maPhongTro));

}

### 2.5.3. View phòng trọ

**Xem chi tiết phòng trọ**

create view View\_ChiTietPhongTro as

select PhongTro.Maphongtro, LoaiPhongTro.Dientich, LoaiPhongTro.Giatien, PhongTro.Trangthai

from PhongTro

inner join LoaiPhongTro on PhongTro.Maloaiphongtro = LoaiPhongTro.Maloaiphong;

**Thực hiện trên C#**

public DataSet LayChiTietPhongTro(int maPhongTro)

{

return db.ExecuteQueryDataSet("select \* from View\_ChiTietPhongTro where Maphongtro = @Maphongtro", CommandType.Text, new SqlParameter("@Maphongtro",maPhongTro));

}

**Xem trọ chưa được thuê**

create view View\_TroChuaThue

as

select Maphongtro,Dientich,Giatien from PhongTro inner join LoaiPhongTro on PhongTro.Maloaiphongtro = LoaiPhongTro.Maloaiphong

where Phongtro.TrangThai = 0

**Thực hiện trên C#**

public DataSet ShowPhongTroDangTrongTheoMa(int maPhongTro)

{

return db.ExecuteQueryDataSet("select \* from View\_TroChuaThue where maPhongTro = @Maphongtro", CommandType.Text,new SqlParameter("@Maphongtro",maPhongTro));

}

### 2.5.4. View nội thất

**Xem nội thất đang được trang bị cho trọ**

create view View\_XemNoiThatDangDuocTrangBiChoTro

as

select Maphongtro,Manoithat,Tennoithat,Nhasanxuat,Giatien,Trangthai --0 1 2: 0 chua trang bi 1 da trang bi 2 dang sua chua

from NoiThat

where Trangthai != 0

**Thực hiện trên C#**

public DataSet ShowAllNoiThatTrangBiChoTro(int maPhongTro)

{

return db.ExecuteQueryDataSet("select\* from View\_XemNoiThatDangDuocTrangBiChoTro where Maphongtro = @Maphongtro", CommandType.Text,

new SqlParameter("@Maphongtro",maPhongTro));

}

### 2.5.5. View chi tiết dịch vụ

**Xem danh sách dịch vụ đang được trang bị**

create view [dbo].[View\_Ds\_DichVuDangTrangBi]

as

SELECT dbo.ChiTietDichVu.Maphongtro, dbo.ChiTietDichVu.Madichvu, dbo.ChiTietDichVu.Thoidiemdangky, dbo.DichVu.Tendichvu, dbo.DichVu.Chiphi

FROM dbo.ChiTietDichVu INNER JOIN

dbo.DichVu ON dbo.ChiTietDichVu.Madichvu = dbo.DichVu.Madichvu

WHERE (dbo.ChiTietDichVu.Thoidiemngungsudung IS NULL or (datepart(year, dbo.ChiTietDichVu.Thoidiemngungsudung) <= datepart(year, GETDATE()) and datepart(month, dbo.ChiTietDichVu.Thoidiemngungsudung) <= datepart(month, GETDATE())))

GO

public DataSet DanhSachDichVuDangTrangBiChoTro(int maPhongTro)

{

return db.ExecuteQueryDataSet("select \* from View\_Ds\_DichVuDangTrangBi where Maphongtro =@Maphongtro", CommandType.Text, new SqlParameter("@Maphongtro", maPhongTro));

}

## 2.6. Các Trigger

### 2.6.1. Trigger số người tối đa ở một phòng là 4

create trigger trig\_LimitStayers on HopDong after insert

as

begin

declare @roomID int, @NoS int

select @roomID = i.Maphongtro from inserted i

select @NoS = dbo.func\_SoNguoiTrongPhongHienTai(@roomID)

if(@NoS > 4)

begin

raiserror(N'Quá số người quy định', 16, -1)

rollback

end

end

### 2.6.2. Trigger tính tiền điện nước thông qua số điện nước số nước, 1 số điện 3000 VNĐ, 1 khối nước 6000 VNĐ

create trigger Trigger\_CapNhatDienNuoc on DienNuoc

for insert, update

as

begin

declare @tienDien int, @tienNuoc int, @giaDien int, @giaNuoc int, @soKw int, @soKhoi int,@soNuocCu int,@soDienCu int, @thoidiem date

set @giaDien = 3000

set @giaNuoc = 6000

declare @maPhong int

select @maPhong = i.Maphongtro, @thoidiem = Thoidiem from inserted i

if(dbo.func\_DemSoLuongPhongTrongBangDienNuoc(@maPhong) = 1) -- đếm số lượng phòng trong bảng điện nước

begin

update DienNuoc

set Tiendien = 0, Tiennuoc = 0,Sodiencu = 0,Sonuoccu = 0

where Maphongtro = @maPhong

end

else

begin

select @soKw = Sodienhientai -Sodiencu, @soKhoi = Sonuochientai - Sonuoccu

from inserted

set @tienDien = @soKw \* @giaDien

set @tienNuoc = @soKhoi \* @giaNuoc

set @soDienCu = dbo.func\_LaySoDienCu(@maPhong,@thoiDiem)

set @soNuocCu = dbo.func\_LaySoNuocCu(@maPhong,@thoiDiem)

update DienNuoc

set Tiendien = @tienDien, Tiennuoc = @tienNuoc

from DienNuoc d inner join inserted i on d.Maphongtro = i.Maphongtro and d.Thoidiem = i.Thoidiem

end

end

### 2.6.3. Trigger nếu mã phòng trọ null thì trạng thái nội thất là 0

create trigger trig\_trangthainoithat on NoiThat

after insert, update as

begin

declare @room\_id int

select @room\_id = i.Maphongtro from inserted i

if(@room\_id is null)

begin

update NoiThat

set Trangthai = 0

end

else

begin

update NoiThat

set Trangthai = 1

end

end

### 2.6.4. Trigger tính tổng tiền

create trigger trig\_TinhTongTien on HoaDon

after insert, update

as begin

update HoaDon

set Tongtien = Tiendien + Tiennuoc + Tiendichvu + Tienphong;

end

### 2.6.5. Trigger kiểm tra dịch vụ đã tồn tại thì không được insert vào:

create trigger dbo.trig\_Kiemtradichvudatontai on ChiTietDichVu after insert

as

begin

declare @room\_id int, @ma\_dichvu int

select @room\_id = i.Maphongtro, @ma\_dichvu = i.Madichvu from inserted i

if(dbo.func\_kiemtradichvuphongdangco(@room\_id,@ma\_dichvu) = 0)

begin

raiserror(N'Dịch vụ đã tồn tại', 16, -1)

rollback;

end

end

### 2.6.6. Trigger kiểm tra thời điểm đăng ký phải lớn hơn hoặc bằng thời điểm người thuê dọn vào

create trigger trig\_Kiemtrathoidiemdangky on ChiTietDichVu after update

as

begin

declare @min\_ngaytaohopdong date, @room\_id int, @Ngaydangkydichvu date, @ma\_dichvu int

select @room\_id = i.Maphongtro, @Ngaydangkydichvu = i.Thoidiemdangky,@ma\_dichvu = i.Madichvu from inserted i;

select @min\_ngaytaohopdong = dbo.func\_MinNgayTaoHopDong(@room\_id)

begin

if(@min\_ngaytaohopdong < @Ngaydangkydichvu and @min\_ngaytaohopdong is not null)

begin

raiserror(N'Ngày đăng ký phải lớn hơn hoặc bằng thời điểm người thuê dọn vào', 16, -1)

rollback;

end

end

end

### 2.6.7. Trigger cập nhật lại trạng thái phòng khi điều chỉnh trong bảng người thuê trọ

create trigger trig\_Capnhattrangthaiphong on NguoiThueTro after insert,update

as

begin

declare @Maphong int,@Trangthaiphong bit

select @Maphong = HopDong.Maphongtro

from inserted i inner join HopDong on i.Manguoithue = HopDong.Manguoithue

if exists (select dbo.func\_KiemTraTonTaiHopDong(@Maphong))

begin

set @Trangthaiphong = 1

end

else

begin

set @Trangthaiphong = 0

end

update PhongTro

set Trangthai = @Trangthaiphong

where Maphongtro = @Maphong

end

### 2.6.8. Trigger kiểm tra ngày dọn vào và ngày tạo hợp đồng

create trigger trig\_Kiemtrangaydonvao on NguoiThueTro after update

as

begin

declare @Ngaydonvao date,@Ngaytaohopdong date, @Manguoithue int

select @Manguoithue = i.Manguoithue from inserted i

select @Ngaydonvao = i.Ngayvaoo, @Ngaytaohopdong = HopDong.Ngaytaohd

from inserted i inner join HopDong on i.Manguoithue = HopDong.Manguoithue

where HopDong.Manguoithue = @Manguoithue

if(@Ngaydonvao < @Ngaytaohopdong)

begin

raiserror(N'Ngày dọn vào phải sau hoặc bằng ngày tạo hợp đồng', 16, -1)

rollback

end

end

### 2.6.9. Trigger khi insert ngày tạo

create trigger trig\_capnhatngayketthuchopdong on HopDong after insert

as

begin

declare @ngayTaoHopDong date,@maNguoiThue int,@maPhong int

select @ngayTaoHopDong = i.Ngaytaohd,@maNguoiThue = i.Manguoithue,@maPhong = i.Maphongtro from inserted i

declare @ngayKetThucHopDong date

set @ngayKetThucHopDong = dateadd(mm,6,@ngayTaoHopDong)

update HopDong

set Ngayketthuchopdong = @ngayKetThucHopDong

where Manguoithue = @maNguoiThue and Maphongtro = @maPhong

end

### 2.6.10. Trigger tạo hóa đơn

create or alter trigger trig\_TaoHoaDon on HoaDon

after insert

as

begin

declare @mapt int, @tiendien int, @tiennuoc int, @tiendv int, @tienphong int, @thoidiem date

select @mapt = i.Maphongtro, @thoidiem = Thoidiem from inserted i

select @tiendien = Tiendien, @tiennuoc = Tiennuoc from DienNuoc where Maphongtro = @mapt and DienNuoc.Thoidiem = @thoidiem

select @tiendv = sum(dv.Chiphi) from ChiTietDichVu ctdv

join DichVu dv on ctdv.Madichvu = dv.Madichvu

where DATEDIFF(month, ctdv.Thoidiemdangky, @thoidiem) = 1 and (@thoidiem < ctdv.Thoidiemngungsudung or ctdv.Thoidiemngungsudung = null)

and ctdv.Maphongtro = @mapt

if(@tiendv is null)

begin

select @tiendv = 0

end

select @tienphong = lpt.Giatien from PhongTro pt

join LoaiPhongTro lpt on pt.Maloaiphongtro = lpt.Maloaiphong where pt.Maphongtro = @mapt

update Hoadon

set Tiendien = @tiendien,

Tiennuoc = @tiennuoc,

Trangthai = 0,

Tiendichvu = @tiendv,

Tienphong = @tienphong,

Tongtien = @tiendien+ @tiennuoc+ @tiendv+ @tienphong

where Thoidiem = @thoidiem and Maphongtro = @mapt

end

### 2.6.11. Trigger xóa thông tin người thuê khi xóa hợp đồng

create trigger [dbo].[trig\_XoaToanBoNguoiThue] on [dbo].[HopDong]

after delete

as

begin

declare @manguoithue int, @maphongtro int

select @manguoithue = d.Manguoithue, @maphongtro = d.Maphongtro from deleted d

if(not exists (select 1 from HopDong where Maphongtro = @maphongtro))

begin

delete from ChiTietDichVu where Maphongtro = @maphongtro

delete from DienNuoc where Maphongtro = @maphongtro

end

end

# CHƯƠNG 3: THIẾT KẾ CÁC CHỨC NĂNG

## 3.1. Kết nối cơ sở dữ liệu

## Lớp DAL

public bool connectToDataBase(string nameDataBase, string username, string password)

{

err = "";

string connStr = $"Data Source=lenovo14\\SQLEXPRESS;Initial Catalog=QuanLyPhongTro;User ID={username};Password={password};Encrypt=False";

conn = new SqlConnection(connStr);

try

{

conn.Open();

conn.Close();

return true;

}

catch (Exception ex)

{

err = ex.Message;

return false;

}

}

## Lớp BAL

public static void ConnectDataBase(string dtb,string user,string pass)

{

nameDatabase = dtb;

username = user;

password = pass;

DataAccessLayer db = new DataAccessLayer(nameDatabase,username,password);

}

## Lớp Presentation

private void button\_dangNhap\_Click(object sender, EventArgs e)

{

string nameDataBase = comboBox\_Database.Text;

string userName = textbox\_username.Text;

string password = textbox\_password.Text;

if (string.IsNullOrWhiteSpace(nameDataBase))

{

MessageBox.Show("Vui lòng chọn database");

}

else if(BAL\_ConnectDatabase.SuccesConn(out string error) == false)

{

MessageBox.Show(error,"ERROR",MessageBoxButtons.OK,MessageBoxIcon.Error);

}

else

{

BAL\_ConnectDatabase.ConnectDataBase(nameDataBase,userName,password);

if (nameDataBase == "QuanLyPhongTro")

{

balAccount = new BAL\_Account();

var data = balAccount.LayThongTinTaiKhoang(userName).Tables[0].Rows[0];

string role\_of\_account = Convert.ToString(data["roles"]);

if (role\_of\_account == "user")

{

int maPhong = Convert.ToInt32(data["ma\_Phong"]);

LoadFormHomeUser(maPhong);

}

else if(role\_of\_account == "admin")

{

LoadFormHomeAdmin();

}

}

else

MessageBox.Show($"Kết nối thành công vào database {nameDataBase}");

}

}

## Các thủ tục về hoá đơn

### Tạo hoá đơn

create proc proc\_ShowHoaDon @PK\_ThoiDiem date, @PK\_MaPhong int

as

begin

select \* from HoaDon where Thoidiem = @PK\_ThoiDiem and Maphongtro = @PK\_MaPhong

end

Thực hiện trên C#

public DataSet ShowHoaDon(DateTime? thoidiem, int maPT)

{

return db.ExecuteQueryDataSet("proc\_ShowHoaDon", CommandType.StoredProcedure,

new SqlParameter("@PK\_ThoiDiem", thoidiem),

new SqlParameter("@PK\_MaPhong", maPT));

**}**

### Thêm hoá đơn

create proc proc\_ThemHoaDon @ThoiDiem date, @MaPhong int, @TienDien int, @TienNuoc int, @TrangThai bit, @TienDV int,@TienPhong int

as

begin

insert into HoaDon values(@ThoiDiem,@MaPhong,@TienDien,@TienNuoc,@TrangThai,@TienDV,@TienPhong,null)

end

Thực hiện trên C#

public bool ThemHoaDon(DateTime? thoidiem, int maPT, int tienDien, int tienNuoc, byte trangThai, int tienDV, int tienPhong, ref string err)

{

return db.MyExecuteNonQuery("proc\_ThemHoaDon", CommandType.StoredProcedure, ref err,

new SqlParameter("@ThoiDiem", thoidiem),

new SqlParameter("@MaPhong", maPT),

new SqlParameter("@Tiendien", tienDien),

new SqlParameter("@Tiennuoc", tienNuoc),

new SqlParameter("@Trangthai", trangThai),

new SqlParameter("@TienDV", tienDV),

new SqlParameter("@Tienphong", tienPhong));

}

### Xóa hoá đơn

create proc proc\_XoaHoaDon @PK\_ThoiDiem date, @PK\_MaPhong int

as

begin

delete from HoaDon where Thoidiem = @PK\_ThoiDiem and Maphongtro = @PK\_MaPhong

end

Thực hiện trên C#

public bool XoaHoaDon(DateTime? thoidiem, int maPT, ref string err)

{

return db.MyExecuteNonQuery("proc\_XoaHoaDon", CommandType.StoredProcedure, ref err,

new SqlParameter("@PK\_ThoiDiem", thoidiem),

new SqlParameter("@PK\_MaPhong", maPT));

}

### Cập nhật hoá đơn

create proc proc\_CapNhatHoaDon @ThoiDiem date, @MaPhong int, @TienDien int, @TienNuoc int, @TrangThai bit, @TienDV int,@TienPhong int

as

begin

update HoaDon

set Tiendien = @TienDien,

Tiennuoc = @TienNuoc,

Trangthai = @TrangThai,

Tiendichvu = @TienDV,

Tienphong = @TienPhong

where Thoidiem = @ThoiDiem and

Maphongtro = @MaPhong

end

Thực hiện trên C#

public bool CapNhatHoaDon(DateTime? thoidiem, int maPT, int tienDien, int tienNuoc, byte trangThai, int tienDV, int tienPhong, ref string err)

{

return db.MyExecuteNonQuery("proc\_CapNhatHoaDon", CommandType.StoredProcedure, ref err,

new SqlParameter("@ThoiDiem", thoidiem),

new SqlParameter("@MaPhong", maPT),

new SqlParameter("@Tiendien", tienDien),

new SqlParameter("@Tiennuoc", tienNuoc),

new SqlParameter("@Trangthai", trangThai),

new SqlParameter("@TienDV", tienDV),

new SqlParameter("@Tienphong", tienPhong));

}

## Các thủ tục về loại phòng trọ

### Show loại phòng trọ

create proc proc\_ShowLoaiPhong

@PK\_MaLoaiPhong int

as

begin

select \* from LoaiPhongTro where Maloaiphong = @PK\_MaLoaiPhong

end

**Thực hiện trên C#**

public DataSet ShowLoaiPhongTro(int maLoaiPhong)

{

return db.ExecuteQueryDataSet("proc\_ShowLoaiPhong", CommandType.StoredProcedure,

new SqlParameter("@PK\_MaLoaiPhong", maLoaiPhong));

}

### Thêm loại phòng trọ

create proc proc\_ThemLoaiPhong @MaLoaiPhong int, @DienTich int, @GiaTien int, @TienCoc int

as

begin

insert into LoaiPhongTro values(@MaLoaiPhong,@DienTich,@GiaTien,@TienCoc)

end

Thực hiện trên C#

public bool ThemLoaiPhongTro(int maLoaiPhong, int dienTich, int giaTien, int tienCoc, ref string err)

{

return db.MyExecuteNonQuery("proc\_ThemLoaiPhong", CommandType.StoredProcedure, ref err,

new SqlParameter("@MaLoaiPhong", maLoaiPhong),

new SqlParameter("@DienTich", dienTich),

new SqlParameter("@GiaTien", giaTien),

new SqlParameter("@TienCoc", tienCoc));

}

### Xóa loại phòng trọ

create proc proc\_XoaLoaiPhong @Maloaiphong int

as

begin

delete from LoaiPhongTro where Maloaiphong = @Maloaiphong

end

Thực hiện trên C#

public bool XoaLoaiPhongTro(int maLoaiPhong, ref string err)

{

return db.MyExecuteNonQuery("proc\_XoaLoaiPhong", CommandType.StoredProcedure, ref err,

new SqlParameter("@Maloaiphong", maLoaiPhong));

}

### Cập nhật loại phòng trọ

create proc proc\_CapNhatLoaiPhong @MaLoaiPhong int, @DienTich int, @GiaTien int, @TienCoc int

as

begin

update LoaiPhongTro

set Dientich = @DienTich,

Giatien = @GiaTien,

Tiencoc = @TienCoc

where Maloaiphong = @MaLoaiPhong

end

Thực hiện trên C#

public bool CapNhatLoaiPhongTro(int maLoaiPhong, int dienTich, int giaTien, int tienCoc, ref string err)

{

return db.MyExecuteNonQuery("proc\_CapNhatLoaiPhong", CommandType.StoredProcedure, ref err,

new SqlParameter("@MaLoaiPhong", maLoaiPhong),

new SqlParameter("@DienTich", dienTich),

new SqlParameter("@GiaTien", giaTien),

new SqlParameter("@TienCoc", tienCoc));

}

* + 1. Tìm kiếm theo mã loại phòng trọ

create proc [dbo].[proc\_ShowAllTheoMaLoaiPhong] @maLP int

as

begin

select \* from LoaiPhongTro where Maloaiphong = @maLP

end

GO

Thực hiện trên C#

public DataSet TimKiemTheoMaLP(int maLP)

{

return db.ExecuteQueryDataSet("proc\_ShowAllTheoMaLoaiPhong", CommandType.StoredProcedure,new SqlParameter("@maLP",maLP));

}

* + 1. Lấy mã loại phòng trọ lớn nhất

create proc [dbo].[proc\_LayMaxMaLoaiPhongTro]

as

begin

select dbo.func\_maxMaLoaiPhongTro() as Maloaiphongtro

end

GO

***Thực hiện trên C#***

public int layMaxMaloaiphong()

{

int maxMaloaiphong = -1;

DataSet maLoaiPhong = db.ExecuteQueryDataSet("proc\_LayMaxMaLoaiPhongTro", CommandType.StoredProcedure, null);

maxMaloaiphong = maLoaiPhong.Tables[0].Rows[0][0] is DBNull ? 0 : Convert.ToInt32(maLoaiPhong.Tables[0].Rows[0][0]);

return maxMaloaiphong;

}

## Các thủ tục về phòng trọ

### Show phòng trọ

create proc proc\_ShowPhongTro

@MaPhong Int

As

Begin

SELECT \* FROM PhongTro WHERE Maphongtro = @Maphong

End

**Thực hiện trên C#**

public DataSet ShowPhongTro(int maPhong)

{

return db.ExecuteQueryDataSet("proc\_ShowPhongTro",

CommandType.StoredProcedure,

new SqlParameter("@MaPhong", maPhong));

}

### Thêm phòng trọ

create proc proc\_ThemPhongTro

@MaPhong Int,

@MaLoaiPhong Int

AS

BEGIN

INSERT INTO PhongTro

VALUES (@MaPhong, @MaLoaiPhong,0)

END

Thực hiện trên C#

public bool ThemPhongTro(int maPhong, int maLoaiPhong, ref string err)

{

return db.MyExecuteNonQuery("proc\_ThemPhongTro",

CommandType.StoredProcedure,

ref err,

new SqlParameter("@MaPhong", maPhong),

new SqlParameter("@MaLoaiPhong", maLoaiPhong)

);

}

### Xóa phòng trọ

create proc proc\_XoaPhongTro

@MaPhong Int

As

Begin

Delete From PhongTro Where Maphongtro = @MaPhong

End

Thực hiện trên C#

public bool XoaPhongTro(int maPhong, ref string err)

{

return db.MyExecuteNonQuery("proc\_XoaPhongTro",

CommandType.StoredProcedure,

ref err, new SqlParameter("@MaPhong", maPhong));

}

### Cập nhật phòng trọ

create proc proc\_CapNhatPhongTro

@MaPhong Int,

@MaLoaiPhong Int,

@TrangThai BIT

As

Begin

UPDATE PhongTro

SET Maloaiphongtro = @MaLoaiphong, Trangthai = @TrangThai

WHERE Maphongtro = @MaPhong

END

Thực hiện trên C#

public bool CapNhatPhongTro(int maPhong, int maLoaiPhong, bool trangThai, ref string err)

{

return db.MyExecuteNonQuery("proc\_CapNhatPhongTro",

CommandType.StoredProcedure,

ref err, new SqlParameter("@MaPhong", maPhong),

new SqlParameter("@MaLoaiPhong", maLoaiPhong),

new SqlParameter("@TrangThai", trangThai));

}

* + 1. Tìm kiếm theo mã phòng trọ

create proc [dbo].[proc\_ShowAllTheoMaPhong] @maPhong int

as

begin

select \* from PhongTro where Maphongtro = @maPhong

end

GO

Thực hiện trên C#

public DataSet TimKiemTheoMaPhong(int maPhong)

{

return db.ExecuteQueryDataSet($"proc\_ShowAllTheoMaPhong", CommandType.StoredProcedure,new SqlParameter("@maPhong",maPhong));

}

* + 1. Lấy Maphongtro lớn nhất

create proc proc\_LayMaxMaPhong

as

begin

select dbo.func\_maxMaPhongTro() as Maphongtro

end

***Thực hiện trên C#***

public int layMaxMaphong()

{

int maxMaphong = -1;

DataSet maPhong = db.ExecuteQueryDataSet("proc\_LayMaxMaPhong", CommandType.StoredProcedure, null);

maxMaphong = maPhong.Tables[0].Rows[0][0] is DBNull ? 0 : Convert.ToInt32(maPhong.Tables[0].Rows[0][0]);

return maxMaphong;

}

## Các thủ tục về điện nước

### Show điện nước

Create Proc proc\_ShowDienNuoc

@Maphongtro INT,

@Thoidiem DATE = NULL

AS

BEGIN

-- nếu ThoiDiem null thì hiển thị tất cả các thông tin điện nước của phòng

IF @Thoidiem IS NULL

SELECT \* FROM DienNuoc WHERE Maphongtro = @Maphongtro Order by Thoidiem Desc

ELSE

SELECT \* FROM DienNuoc WHERE Maphongtro = @Maphongtro AND Thoidiem = @Thoidiem

END

**Thực hiện trên C#**

public DataSet ShowDienNuoc(int maPhongTro, DateTime? thoiDiem = null)

{

if (thoiDiem == null)

{

return db.ExecuteQueryDataSet("proc\_ShowDienNuoc", CommandType.StoredProcedure,

new SqlParameter("@Maphongtro", maPhongTro));

}

else

{

return db.ExecuteQueryDataSet("proc\_ShowDienNuoc", CommandType.StoredProcedure,

new SqlParameter("@Maphongtro", maPhongTro),

new SqlParameter("@Thoidiem", thoiDiem));

}

}

### Thêm điện nước

CREATE PROCEDURE proc\_ThemDienNuoc

@Maphongtro INT,

@Thoidiem DATE,

@Sodienhientai INT,

@Sonuochientai INT,

@Sodiencu INT,

@Sonuoccu INT

AS

BEGIN

INSERT INTO DienNuoc (Maphongtro, Thoidiem, Sodienhientai, Sonuochientai, Sodiencu, Sonuoccu)

VALUES (@Maphongtro, @Thoidiem, @Sodienhientai, @Sonuochientai, @Sodiencu, @Sonuoccu)

END

Thực hiện trên C#

public bool ThemDienNuoc(int maPhongTro, DateTime thoiDiem, int soDienHienTai, int soNuocHienTai,ref string err)

{

return db.MyExecuteNonQuery("proc\_ThemDienNuoc", CommandType.StoredProcedure, ref err,

new SqlParameter("@Maphongtro", maPhongTro),

new SqlParameter("@Thoidiem", thoiDiem),

new SqlParameter("@Sodienhientai", soDienHienTai),

new SqlParameter("@Sonuochientai", soNuocHienTai));

}

### Xóa điện nước

CREATE PROCEDURE proc\_XoaDienNuoc

@Maphongtro INT,

@Thoidiem DATE

AS

BEGIN

DELETE FROM DienNuoc WHERE Maphongtro = @Maphongtro AND Thoidiem = @Thoidiem

END

Thực hiện trên C#

public bool XoaDienNuoc(int maPhongTro, DateTime thoiDiem, ref string err)

{

return db.MyExecuteNonQuery("proc\_XoaDienNuoc", CommandType.StoredProcedure, ref err,

new SqlParameter("@Maphongtro", maPhongTro),

new SqlParameter("@Thoidiem", thoiDiem));

}

### Cập nhật điện nước

CREATE PROC proc\_CapNhatDienNuoc

@Maphongtro INT,

@Thoidiem DATE,

@Sodienhientai INT,

@Sonuochientai INT,

@Sodiencu INT,

@Sonuoccu INT

AS

BEGIN

UPDATE DienNuoc

SET Sodienhientai = @Sodienhientai,

Sonuochientai = @Sonuochientai,

Sodiencu = @Sodiencu,

Sonuoccu = @Sonuoccu

WHERE Maphongtro = @Maphongtro AND Thoidiem = @Thoidiem

END

Thực hiện trên C#

public bool CapNhatDienNuoc(int maPhongTro, DateTime thoiDiem, int soDienHienTai, int soNuocHienTai, int soDienCu, int soNuocCu, ref string err)

{

return db.MyExecuteNonQuery("proc\_CapNhatDienNuoc", CommandType.StoredProcedure, ref err,

new SqlParameter("@Maphongtro", maPhongTro),

new SqlParameter("@Thoidiem", thoiDiem),

new SqlParameter("@Sodienhientai", soDienHienTai),

new SqlParameter("@Sonuochientai", soNuocHienTai),

new SqlParameter("@Sodiencu", soDienCu),

new SqlParameter("@Sonuoccu", soNuocCu));

}

* + 1. Tìm kiếm điện nước theo mã phòng trọ

create proc [dbo].[proc\_ShowAllDienNuocTheoMaPhongTro] @maPT int

as

begin

select \* from DienNuoc where Maphongtro = @maPT

end

GO

Thực hiện trên C#

public DataSet TimKiemTheoMaPhong(int maPT)

{

return db.ExecuteQueryDataSet("proc\_ShowAllDienNuocTheoMaPhongTro",CommandType.StoredProcedure,new SqlParameter("@maPT",maPT));

}

* + 1. Lấy thời điểm tiếp theo

create proc [dbo].[proc\_LayThoiDiemTiepTheo] @maPhong int

as

begin

select dbo.func\_thoiDiemDienNuocTiepTheo(@maPhong) from DienNuoc where Maphongtro = @maPhong

end

***Thực hiện trên C#***

public DateTime LayThoiDiemTiepTheo(int maPhong)

{

DateTime time = DateTime.Now;

DataTable table\_time = db.ExecuteQueryDataSet("proc\_LayThoiDiemTiepTheo", CommandType.StoredProcedure,

new SqlParameter("Maphong", maPhong)).Tables[0];

time = Convert.ToDateTime(table\_time.Rows[0][0]);

return time;

}

## Các thủ tục về loại nội thất

### Show loại nội thất

create proc proc\_ShowLoaiNoiThat @PK\_MaLoaiNoiThat int

as

begin

select \* from LoaiNoiThat where Maloainoithat = @PK\_MaLoaiNoiThat;

end

**Thực hiện trên C#**

public DataSet ShowLoaiNoiThat(int maLoaiNoiThat)

{

return db.ExecuteQueryDataSet("proc\_ShowLoaiNoiThat", CommandType.StoredProcedure,

new SqlParameter("@PK\_MaLoaiNoiThat", maLoaiNoiThat));

}

### Thêm loại nội thất

create proc proc\_ThemLoaiNoiThat @MaLoaiNoiThat int, @TenLoaiNoiThat nvarchar(50)

as

begin

insert into LoaiNoiThat values (@MaLoaiNoiThat, @TenLoaiNoiThat);

end

Thực hiện trên C#

public bool ThemLoaiNoiThat(int maLoaiNoiThat, string tenLoaiNoiThat, ref string err)

{

return db.MyExecuteNonQuery("proc\_ThemLoaiNoiThat", CommandType.StoredProcedure, ref err,

new SqlParameter("@MaLoaiNoiThat", maLoaiNoiThat),

new SqlParameter("@TenLoaiNoiThat", tenLoaiNoiThat));

}

### Xóa loại nội thất

create proc proc\_XoaLoaiNoiThat @PK\_MaLoaiNoiThat int

as

begin

delete LoaiNoiThat where Maloainoithat = @PK\_MaLoaiNoiThat;

end

Thực hiện trên C#

public bool XoaLoaiNoiThat(int maLoaiNoiThat, ref string err)

{

return db.MyExecuteNonQuery("proc\_XoaLoaiNoiThat", CommandType.StoredProcedure, ref err,

new SqlParameter("@PK\_MaLoaiNoiThat", maLoaiNoiThat));

}

### Cập nhật loại nội thất

create proc proc\_CapNhatLoaiNoiThat @MaLoaiNoiThat int, @TenLoaiNoiThat nvarchar(50)

as

begin

update LoaiNoiThat

set Tenloainoithat = @TenLoaiNoiThat

where Maloainoithat = @MaLoaiNoiThat

end

Thực hiện trên C#

public bool CapNhatLoaiNoiThat(int maLoaiNoiThat, string tenLoaiNoiThat, ref string err)

{

return db.MyExecuteNonQuery("proc\_CapNhatLoaiNoiThat", CommandType.StoredProcedure, ref err,

new SqlParameter("@MaLoaiNoiThat", maLoaiNoiThat),

new SqlParameter("@TenLoaiNoiThat", tenLoaiNoiThat));

}

* + 1. Tìm kiếm theo mã loại nội thất

create proc [dbo].[proc\_ShowAllTheoMaLoaiNoiThat] @maLNT int

as

begin

select \* from LoaiNoiThat where Maloainoithat = @maLNT

end

GO

Thực hiện trên C#

public DataSet TimKiemTheoMaLNT(int maLoaiNoiThat)

{

return db.ExecuteQueryDataSet("proc\_ShowAllTheoMaLoaiNoiThat", CommandType.StoredProcedure,new SqlParameter("@maLNT",maLoaiNoiThat));

}

* + 1. Lấy Maloainoithat lớn nhất

create proc proc\_LayMaxMaLoaiNoiThat

as

begin

select dbo.func\_maxMaLoaiNoiThat() as Maloainoithat

end

**Thực hiện trên C#**

public DataSet LayMaLoaiNoiThat()

{

return db.ExecuteQueryDataSet("select Maloainoithat from LoaiNoiThat", CommandType.Text, null);

}

## Các hàm, thủ tục về nội thất

### Show nội thất

create proc proc\_ShowNoiThat @PK\_MaNoiThat int

as

begin

select \* from NoiThat where Manoithat = @PK\_MaNoiThat;

end

**Thực hiện trên C#**

public DataSet ShowNoiThat(int maNoiThat)

{

return db.ExecuteQueryDataSet("proc\_ShowNoiThat", CommandType.StoredProcedure,

new SqlParameter("@PK\_MaNoiThat", maNoiThat));

}

public DataSet ShowAllNoiThat()

{

return db.ExecuteQueryDataSet("select \* from NoiThat", CommandType.Text,

null);

}

### Thêm nội thất

create proc proc\_ThemNoiThat @MaNoiThat int, @TenNoiThat nvarchar(50), @MaPhongTro int, @MaLoaiNoiThat int, @GiaTien int, @NhaSanXuat nvarchar(50), @TrangThai bit

as

begin

insert into NoiThat values (@MaNoiThat, @TenNoiThat, @MaPhongTro, @MaLoaiNoiThat, @GiaTien, @NhaSanXuat, @TrangThai);

end

Thực hiện trên C#

public bool ThemNoiThat(int maNoiThat, string tenNoiThat, int maPhongTro, int maLoaiNoiThat, int giaTien, string nhaSanXuat, bool trangThai, ref string err)

{

return db.MyExecuteNonQuery("proc\_ThemNoiThat", CommandType.StoredProcedure, ref err,

new SqlParameter("@MaNoiThat", maNoiThat),

new SqlParameter("@TenNoiThat", tenNoiThat),

new SqlParameter("@MaPhongTro", maPhongTro),

new SqlParameter("@MaLoaiNoiThat", maLoaiNoiThat),

new SqlParameter("@GiaTien", giaTien),

new SqlParameter("@NhaSanXuat", nhaSanXuat),

new SqlParameter("@TrangThai", trangThai));

}

### Xóa nội thất

create proc proc\_XoaNoiThat @PK\_MaNoiThat int

as

begin

delete NoiThat where Manoithat = @PK\_MaNoiThat;

end

Thực hiện trên C#

public bool XoaNoiThat(int maNoiThat, ref string err)

{

return db.MyExecuteNonQuery("proc\_XoaNoiThat", CommandType.StoredProcedure, ref err,

new SqlParameter("@PK\_MaNoiThat", maNoiThat));

}

### Cập nhật nội thất

create proc proc\_CapNhatNoiThat @MaNoiThat int, @TenNoiThat nvarchar(50), @MaPhongTro int, @MaLoaiNoiThat int, @GiaTien int, @NhaSanXuat nvarchar(50), @TrangThai bit

as

begin

update NoiThat

set Tennoithat = @TenNoiThat, Maphongtro = @MaPhongTro, Maloainoithat = @MaLoaiNoiThat, Giatien = @GiaTien, Nhasanxuat = @NhaSanXuat, Trangthai = @TrangThai

where Manoithat = @MaNoiThat;

end

* + 1. Lấy Manoithat lớn nhất

create proc proc\_LayMaxMaNoiThat

as

begin

select dbo.func\_maxMaNoiThat() as Manoithat

end

Thực hiện trên C#

public int layMaxMaNoiThat()

{

int maxMaNoiThat = -1;

DataSet maPhong = db.ExecuteQueryDataSet("proc\_LayMaxMaNoiThat", CommandType.StoredProcedure, null);

maxMaNoiThat = Convert.ToInt32(maPhong.Tables[0].Rows[0][0]);

return maxMaNoiThat;

}

* + 1. Tìm kiếm theo mã nội thất

create proc [dbo].[proc\_ShowAllTheoMaNoiThat] @maNT int

as

begin

select \* from NoiThat where Manoithat = @maNT

end

GO

Thực hiện trên C#

public DataSet TimKiemTheoMaNT(int maNT)

{

return db.ExecuteQueryDataSet("proc\_ShowAllTheoMaNoiThat", CommandType.StoredProcedure,new SqlParameter("@maNT",maNT));

}

## Các hàm, thủ tục về dịch vụ

### Show dịch vụ

create proc proc\_ShowDichVu @PK\_MaDichVu int

as

begin

select \* from DichVu where Madichvu = @PK\_MaDichVu;

end

**Thực hiện trên C#**

public DataSet ShowDichVu(int maDV)

{

return db.ExecuteQueryDataSet("proc\_ShowDichVu", CommandType.StoredProcedure,

new SqlParameter("@PK\_MaDichVu", maDV));

}

### Thêm dịch vụ

create proc proc\_ThemDichVu @MaDichVu int, @TenDichVu nvarchar(50), @ChiPhi int

as

begin

insert into DichVu VALUES (@MaDichVu, @TenDichVu, @ChiPhi);

end

Thực hiện trên C#

public bool ThemDichVu(int maDV, string tenDV, int chiPhi, ref string err)

{

return db.MyExecuteNonQuery("proc\_ThemDichVu", CommandType.StoredProcedure, ref err,

new SqlParameter("@MaDichVu", maDV),

new SqlParameter("@TenDichVu", tenDV),

new SqlParameter("@ChiPhi", chiPhi));

}

### Xóa dịch vụ

create proc proc\_XoaDichVu @PK\_MaDichVu int

as

begin

delete DichVu where Madichvu = @PK\_MaDichVu;

end

Thực hiện trên C#

public bool XoaDichVu(int maDV, ref string err)

{

return db.MyExecuteNonQuery("proc\_XoaDichVu", CommandType.StoredProcedure, ref err,

new SqlParameter("@PK\_MaDichVu", maDV));

}

### Cập nhật dịch vụ

create proc proc\_CapNhatDichVu @MaDichVu int, @TenDichVu nvarchar(50), @ChiPhi int

as

begin

update DichVu

set Tendichvu = @TenDichVu, Chiphi = @ChiPhi

where Madichvu = @MaDichVu;

end

Thực hiện trên C#

public bool CapNhatDichVu(int maDV, string tenDV, int chiPhi, ref string err)

{

return db.MyExecuteNonQuery("proc\_CapNhatDichVu", CommandType.StoredProcedure, ref err,

new SqlParameter("@MaDichVu", maDV),

new SqlParameter("@TenDichVu", tenDV),

new SqlParameter("@ChiPhi", chiPhi));

}

* + 1. Tìm kiếm theo mã dịch vụ

create proc [dbo].[proc\_ShowAllDichVuTheoMaDichVu] @maDV int

as

begin

select \* from DichVu where Madichvu = @maDV

end

GO

Thực hiện trên C#

public DataSet TimKiemTheoMaDV(int maDV)

{

return db.ExecuteQueryDataSet("proc\_ShowAllDichVuTheoMaDichVu", CommandType.StoredProcedure, new SqlParameter("@maDV",maDV));

}

* + 1. Lấy Madichvu lớn nhất

create proc proc\_LayMaxMaDichVu

as

begin

select dbo.func\_maxMaDichVu() as Madichvu

end

**Thực hiện trên C#**

public int laymaxMaDichVu()

{

int maxMaDichVu = -1;

DataSet maDichVu = db.ExecuteQueryDataSet("proc\_LayMaxMaDichVu", CommandType.StoredProcedure, null);

maxMaDichVu = maDichVu.Tables[0].Rows[0][0] is DBNull ? 0 : Convert.ToInt32(maDichVu.Tables[0].Rows[0][0]);

return maxMaDichVu;

}

## Các thủ tục về người thuê trọ

### Show người thuê trọ

create proc proc\_ShowAllNguoiThueTro

as

begin

select \* from NguoiThueTro

end

**Thực hiện trên C#**

public DataSet ShowNguoiThueTro(int maNguoiThue)

{

return db.ExecuteQueryDataSet("proc\_ShowNguoiThueTro", CommandType.StoredProcedure,

new SqlParameter("@PK\_MaNguoiThue", maNguoiThue));

}

### Tìm kiếm theo mã người thuê trọ

create proc [dbo].[proc\_ShowAllTheoMaNguoiThueTro] @maNTT int

as

begin

select \* from NguoiThueTro where Manguoithue = @maNTT

end

GO

**Thực hiện trên C#**

public DataSet TimKiemTheoMaNTT(int maNTT)

{

return db.ExecuteQueryDataSet($"proc\_ShowAllTheoMaNguoiThueTro", CommandType.StoredProcedure, new SqlParameter("@maNTT", maNTT));

}

### Show chi tiết người thuê trọ theo mã

create proc proc\_ViewShowChiTietNguoiThueTroTheoMaPhongTro

@Manguoithue int

as

begin

select \* from View\_Ds\_ChiTietNguoiThueTro

where Manguoithue = @Manguoithue

end

### Thêm người thuê trọ

create proc proc\_ThemNguoiThueTro

@Manguoithue int,@Ngayvao date, @Ngaydi date, @Ho nvarchar(50), @Tenlot nvarchar(50), @Ten nvarchar(50), @SoCCCD char(12), @Quequan nvarchar(50), @Sodt char(10), @Job nvarchar(50)

as

begin

insert into NguoiThueTro

values(@Manguoithue,@Ngayvao,@Ngaydi,@Ho,@Tenlot,@Ten,@SoCCCD,@Quequan,@Sodt,@Job)

end

Thực hiện trên C#

public bool ThemNguoiThueTro(int maNguoiThue, DateTime? ngayVaoO, DateTime? ngayDonDi, string ho, string tenLot, string ten, string soCCCD, string queQuan, string soDT, string ngheNghiep, ref string err)

{

return db.MyExecuteNonQuery("proc\_ThemNguoiThueTro", CommandType.StoredProcedure, ref err,

new SqlParameter("@MaNguoiThue", maNguoiThue),

new SqlParameter("@NgayVao", ngayVaoO),

new SqlParameter("@Ngaydi", ngayDonDi),

new SqlParameter("@Ho", ho),

new SqlParameter("@TenLot", tenLot),

new SqlParameter("@Ten", ten),

new SqlParameter("@SoCCCD", soCCCD),

new SqlParameter("@QueQuan", queQuan),

new SqlParameter("@SoDT", soDT),

new SqlParameter("@job", ngheNghiep));

}

### Xoá người thuê trọ

create proc proc\_XoaNguoiThueTro

@Manguoithue int

as

begin

delete from NguoiThueTro

where Manguoithue = @Manguoithue

end

Thực hiện trên C#

public bool XoaNguoiThueTro(int maNguoiThue, ref string err)

{

return db.MyExecuteNonQuery("proc\_XoaNguoiThueTro", CommandType.StoredProcedure, ref err,

new SqlParameter("@Manguoithue", maNguoiThue));

}

### Cập nhật người thuê trọ

create proc proc\_CapNhatNguoiThueTro

@Manguoithue int,@Ngayvao date, @Ngaydi date, @Ho nvarchar(50), @Tenlot nvarchar(50), @Ten nvarchar(50), @SoCCCD char(12), @Quequan nvarchar(50), @Sodt char(10), @Job nvarchar(50)

as

begin

update NguoiThueTro

set

Ngayvaoo = @Ngayvao,

Ngaydondi = @Ngaydi,

Ho = @Ho,

Tenlot = @Tenlot,

Ten = @Ten,

SoCCCD = @SoCCCD,

Quequan = @Quequan,

Sodt = @Sodt,

Nghenghiep = @Job

where Manguoithue = @Manguoithue

end

Thực hiện trên C#

public bool CapNhatNguoiThueTro(int maNguoiThue, DateTime? ngayVaoO, DateTime? ngayDonDi, string ho, string tenLot, string ten, string soCCCD, string queQuan, string soDT, string ngheNghiep, ref string err)

{

return db.MyExecuteNonQuery("proc\_CapNhatNguoiThueTro", CommandType.StoredProcedure, ref err,

new SqlParameter("@MaNguoiThue", maNguoiThue),

new SqlParameter("@NgayVao", ngayVaoO),

new SqlParameter("@Ngaydi", ngayDonDi),

new SqlParameter("@Ho", ho),

new SqlParameter("@TenLot", tenLot),

new SqlParameter("@Ten", ten),

new SqlParameter("@SoCCCD", soCCCD),

new SqlParameter("@QueQuan", queQuan),

new SqlParameter("@SoDT", soDT),

new SqlParameter("@job", ngheNghiep));

}

* + 1. Lấy Manguoithue lớn nhất

create proc proc\_LayMaxMaNguoiThue

as

begin

select dbo.func\_maxMaNguoiThue() as Manguoithue

end

***Thực hiện trên C#***

public int layMaxMaNguoiThue()

{

int maxMaNguoiThue = -1;

DataSet maNguoiThue = db.ExecuteQueryDataSet("proc\_LayMaxMaNguoiThue", CommandType.StoredProcedure, null);

maxMaNguoiThue = Convert.ToInt32(maNguoiThue.Tables[0].Rows[0][0]);

return maxMaNguoiThue;

}

## Các thủ tục về hợp đồng

### Show hợp đồng

create proc proc\_ShowHopDong

@Manguoithue int, @Maphongtro int

as

begin

select \* from HopDong

where Manguoithue = @Manguoithue and Maphongtro = @Maphongtro

end

**Thực hiện trên C#**

public DataSet ShowHopDong(int nguoithue, int maPT)

{

return db.ExecuteQueryDataSet("proc\_ShowHopDong", CommandType.StoredProcedure,

new SqlParameter("@Manguoithue", nguoithue),

new SqlParameter("@Maphongtro", maPT));

}

### Thêm hợp đồng

create proc proc\_ThemHopDong

@Manguoithue int, @Maphongtro int, @Ngaytao date, @Ngaydondi date

as

begin

insert into HopDong

values(@Manguoithue,@Maphongtro,@Ngaytao,@Ngaydondi)

end

Thực hiện trên C#

public bool ThemHopDong(int nguoithue, int maPT, DateTime dangky,ref string err)

{

return db.MyExecuteNonQuery("proc\_ThemHopDong", CommandType.StoredProcedure, ref err,

new SqlParameter("@Manguoithue", nguoithue),

new SqlParameter("@Maphongtro", maPT),

new SqlParameter("@Ngaytao", dangky) );

}

### Xóa hợp đồng

create proc proc\_XoaHopDong

@Manguoithue int, @Maphongtro int

as

begin

delete from HopDong

where Manguoithue = @Manguoithue and Maphongtro = @Maphongtro

end

Thực hiện trên C#

public bool XoaHopDong(int nguoithue, int maPT, ref string err)

{

return db.MyExecuteNonQuery("proc\_XoaHopDong", CommandType.StoredProcedure, ref err,

new SqlParameter("@Manguoithue", nguoithue),

new SqlParameter("@Maphongtro", maPT));

}

### Cập nhật hợp đồng

create proc proc\_CapNhatHopDong

@Manguoithue int, @Maphongtro int, @Ngaytao date, @Ngaykt date

as

begin

update HopDong

set

Ngaytaohd = @Ngaytao,

Ngayketthuchopdong = @Ngaykt

where Manguoithue = @Manguoithue and Maphongtro = @Maphongtro

end

Thực hiện trên C#

public bool CapNhatHopDong(int nguoithue, int maPT, DateTime dangky, DateTime? ketthuc, ref string err)

{

SqlParameter ngaydondi;

if (ketthuc == null)

{

ngaydondi = new SqlParameter("@Ngaykt", DBNull.Value);

}

else

{

ngaydondi = new SqlParameter("@Ngaykt", ketthuc);

}

return db.MyExecuteNonQuery("proc\_CapNhatHopDong", CommandType.StoredProcedure, ref err,

new SqlParameter("@Manguoithue", nguoithue),

new SqlParameter("@Maphongtro", maPT),

new SqlParameter("@Ngaytao", dangky),

ngaydondi);

}

* + 1. Tìm kiếm hợp đồng theo mã người thuê trọ

create proc [dbo].[proc\_ShowAllHopDongTheoMaNguoiThue] @maNTT int

as

begin

select \* from HopDong where Manguoithue = @maNTT

end

GO

Thực hiện trên C#

public DataSet TimKiemTheoMaNTT(int maNTT)

{

return db.ExecuteQueryDataSet($"proc\_ShowAllHopDongTheoMaNguoiThue", CommandType.StoredProcedure,new SqlParameter("@maNTT",maNTT));

}

## Các hàm, thủ tục về chi tiết dịch vụ

### Show chi tiết dịch vụ

create proc proc\_ShowChiTietDichVu

@Maphongtro int, @Madichvu int, @Thoidiemdangky date

as

begin

select \* from ChiTietDichVu

where Maphongtro = @Maphongtro and Madichvu = @Madichvu and Thoidiemdangky = @Thoidiemdangky

end

**Thực hiện trên C#**

public DataSet ShowChiTietDichVu(int maPT, int maDV, DateTime? dangky)

{

return db.ExecuteQueryDataSet("proc\_ShowChiTietDichVu", CommandType.StoredProcedure,

new SqlParameter("@Maphongtro", maPT),

new SqlParameter("@Madichvu", maDV),

new SqlParameter("@Thoidiemdangky", dangky));

}

### Thêm chi tiết dịch vụ

create proc proc\_ThemChiTietDichVu

@Maphongtro int, @Madichvu int, @Thoidiemdangky date

as

begin

insert into ChiTietDichVu(Maphongtro,Madichvu,Thoidiemdangky)

values(@Maphongtro,@Madichvu,@Thoidiemdangky)

end

Thực hiện trên C#

public bool ThemCTDV(int maPT, int maDV, DateTime dangky,ref string err)

{

return db.MyExecuteNonQuery("proc\_ThemChiTietDichVu", CommandType.StoredProcedure, ref err,

new SqlParameter("@Maphongtro", maPT),

new SqlParameter("@Madichvu", maDV),

new SqlParameter("@Thoidiemdangky", dangky)

);

}

* + 1. Xoá chi tiết dịch vụ

create proc proc\_XoaChiTietDichVu

@Maphongtro int, @Madichvu int, @Thoidiemdangky date

as

begin

delete from ChiTietDichVu

where Maphongtro = @Maphongtro and Madichvu = @Madichvu and Thoidiemdangky = @Thoidiemdangky

end

Thực hiện trên C#

public bool XoaChiTietDichVu(int maPT, int maDV, DateTime? dangky, ref string err)

{

return db.MyExecuteNonQuery("proc\_XoaChiTietDichVu", CommandType.StoredProcedure, ref err,

new SqlParameter("@Maphongtro", maPT),

new SqlParameter("@Madichvu", maDV),

new SqlParameter("@Thoidiemdangky", dangky));

}

* + 1. Cập nhật chi tiết dịch vụ

create proc proc\_CapNhatChiTietDichVu

@Maphongtro int, @Madichvu int, @Thoidiemdangky date, @Thoidiemngungsudung date

as

begin

update ChiTietDichVu

set

Thoidiemngungsudung = @Thoidiemngungsudung

where Maphongtro = @Maphongtro and Madichvu = @Madichvu and Thoidiemdangky = @Thoidiemdangky

end

Thực hiện trên C#

public bool CapNhatChiTietDichVu(int maPT, int maDV, DateTime dangky, DateTime? ngungdung, ref string err)

{

SqlParameter nullPara;

if(ngungdung == null)

{

nullPara = new SqlParameter("@Thoidiemngungsudung", DBNull.Value);

}

else

{

nullPara = new SqlParameter("@Thoidiemngungsudung", ngungdung);

}

return db.MyExecuteNonQuery("proc\_CapNhatChiTietDichVu", CommandType.StoredProcedure, ref err,

new SqlParameter("@Maphongtro", maPT),

new SqlParameter("@Madichvu", maDV),

new SqlParameter("@Thoidiemdangky", dangky),

nullPara);

}

* + 1. Tìm kiếm chi tiết dịch vụ theo mã phòng trọ

create proc [dbo].[proc\_ShowAllHopDongTheoMaNguoiThue] @maNTT int

as

begin

select \* from HopDong where Manguoithue = @maNTT

end

GO

Thực hiện trên C#

public DataSet TimKiemTheoMaPT(int maPT)

{

return db.ExecuteQueryDataSet($"proc\_ShowAllChiTietDichVuTheoMaPhongTro", CommandType.StoredProcedure,new SqlParameter("@maPT",maPT));

}

* 1. Các hàm về phòng trọ
     1. Lấy mã phòng trọ lớn nhất đang có

create function func\_maxMaPhongTro()

returns int

as

begin

declare @max\_maphong int

set @max\_maphong = -1

select @max\_maphong = MAX(Maphongtro) from PhongTro

return @max\_maphong

end

Thực hiện trên C#

public int layMaxMaphong()

{

int maxMaphong = -1;

DataSet maPhong = db.ExecuteQueryDataSet("proc\_LayMaxMaPhong", CommandType.StoredProcedure, null);

maxMaphong = maPhong.Tables[0].Rows[0][0] is DBNull ? 0 : Convert.ToInt32(maPhong.Tables[0].Rows[0][0]);

return maxMaphong;

}

* 1. Các hàm về dịch vụ
     1. Lấy mã dịch vụ lớn nhất đang có

create function func\_maxMaDichVu()

returns int

as

begin

declare @MaxMaDichVu int

set @MaxMaDichVu = -1

select @MaxMaDichVu = max(Madichvu) from DichVu

return @MaxMaDichVu

end

Thực hiện trên C#

public int laymaxMaDichVu()

{

int maxMaDichVu = -1;

DataSet maDichVu = db.ExecuteQueryDataSet("proc\_LayMaxMaDichVu", CommandType.StoredProcedure, null);

maxMaDichVu = maDichVu.Tables[0].Rows[0][0] is DBNull ? 0 : Convert.ToInt32(maDichVu.Tables[0].Rows[0][0]);

return maxMaDichVu;

}

* 1. Các hàm về nội thất
     1. Lấy mã nội thất lớn nhất đang có

create function func\_maxMaNoiThat()

returns int

as

begin

declare @MaxMaNoiThat int

set @MaxMaNoiThat = -1

select @MaxMaNoiThat = max(Manoithat) from NoiThat

return @MaxMaNoiThat

end

* 1. Các hàm về loại nội thất
     1. Lấy mã loại nội thất lớn nhất đang có (Sử dụng trong proc)

create function func\_maxMaLoaiNoiThat()

returns int

begin

declare @MaxMaLoaiNoiThat int

set @MaxMaLoaiNoiThat = -1

select @MaxMaLoaiNoiThat = max(Maloainoithat) from LoaiNoiThat

return @MaxMaLoaiNoiThat

end

## Các hàm về chi tiết dịch vụ

* + 1. Kiểm tra dịch vụ phòng đang có hay không (Sử dụng trong trigger)

create function func\_kiemtradichvuphongdangco(@room\_id int, @dichvu\_id int)

returns int

as

begin

declare @trangthai date

select @Trangthai = Thoidiemngungsudung from ChiTietDichVu where Maphongtro = @room\_id and Madichvu = @dichvu\_id

if(@trangthai is null)

return 1

return 0

end

Thực hiện trên C#

* + 1. Tính tổng tiền dịch vụ của phòng tại một thời điểm nào đó

create or alter function func\_TinhTongTienDichVu(@thoiDiem date,@maPhong int)

returns int

as

begin

declare @tongTienDichVu int = 0

if(dbo.func\_KiemTraPhongTroCoDangDuocThue(@thoiDiem,@maPhong) = 1)

begin

select @tongTienDichVu = Sum(DichVu.Chiphi)

from ChiTietDichVu inner join DichVu on ChiTietDichVu.Madichvu = DichVu.Madichvu

where MONTH(Thoidiemdangky) <= MONTH(@thoiDiem) and YEAR(Thoidiemdangky) <= YEAR(@thoiDiem) and ChiTietDichVu.Maphongtro = @maPhong and Thoidiemngungsudung is null

end

return @tongTienDichVu

end

**Thực hiện trên C#**

public int layTongTienDichVu(DateTime thoiDiem, int maPhong)

{

DataTable result = db.ExecuteQueryDataSet("select dbo.func\_TinhTongTienDichVu(@thoiDiem,@maPhong)", CommandType.Text,

new SqlParameter("@thoiDiem", thoiDiem), new SqlParameter("@maPhong", maPhong)).Tables[0];

int tongTien = 0;

tongTien = Convert.ToInt32(result.Rows[0][0]);

return tongTien;

}

## Các hàm về loại phòng trọ

* + 1. Lấy mã loại phòng trọ lớn nhất đang có (Sử dụng trong proc)

create function func\_maxMaLoaiPhongTro()

returns int

begin

declare @MaxMaLoaiPhongTro int

set @MaxMaLoaiPhongTro = -1

select @MaxMaLoaiPhongTro = max(Maloaiphong) from LoaiPhongTro

return @MaxMaLoaiPhongTro

end

## Các hàm về người thuê trọ

* + 1. Lấy mã người thuê lớn nhất đang có (Sử dụng trong proc)

create function func\_maxMaNguoiThue()

returns int

begin

declare @MaxMaNguoiThue int

set @MaxMaNguoiThue = -1

select @MaxMaNguoiThue = max(Manguoithue) from NguoiThueTro

return @MaxMaNguoiThue

end

## Các hàm về hợp đồng

* + 1. Số người đang ở phòng hiện tại (Sử dụng trong trigger giới hạn người thuê)

create function func\_SoNguoiTrongPhongHienTai(@NoS int)

returns int

as

begin

return (select count(\*) from HopDong where HopDong.Maphongtro = @Nos)

end

* + 1. Kiểm tra có tồn tại hợp đồng của phòng nào đó (Sử dụng trong trigger cập nhật trạng thái phòng trọ)

create or alter function func\_KiemTraTonTaiHopDong(@Maphong int)

returns table

as

return (select HopDong.Manguoithue, HopDong.Maphongtro from NguoiThueTro inner join HopDong on NguoiThueTro.Manguoithue = HopDong.Manguoithue

where HopDong.Maphongtro = @Maphong and NguoiThueTro.Ngaydondi is null)

go

Thực hiện trên C#

* + 1. Ngày tạo hợp đồng nhỏ nhất của phòng đó (Sử dụng trong trigger)

create or alter function func\_MinNgayTaoHopDong(@roomId int)

returns table

as

return (select min(HopDong.Ngaytaohd) as MinNgayTaoHD

from NguoiThueTro inner join HopDong on NguoiThueTro.Manguoithue = HopDong.Manguoithue

where HopDong.Maphongtro = @roomId and NguoiThueTro.Ngaydondi is null)

go

* + 1. Kiểm tra ở một thời điểm phòng trọ có đang được thuê (Sử dụng trong trigger tính tổng tiền dịch vụ)

create or alter function func\_KiemTraPhongTroCoDangDuocThue(@thoiDiem date,@maPhong int)

returns bit

as

begin

declare @check bit = 0

if exists (select \*

from HopDong inner join PhongTro on HopDong.Maphongtro = PhongTro.Maphongtro

where @thoiDiem <= HopDong.Ngayketthuchopdong and @thoiDiem >= HopDong.Ngaytaohd and HopDong.Maphongtro = @maPhong)

begin

set @check = 1

end

return @check

end

## Các hàm về điện nước

* + 1. Hàm đếm số lượng phòng trong bảng điện nước (Sử dụng cho trigger cập nhật điện nước và trong các function khác)

create or alter function func\_DemSoLuongPhongTrongBangDienNuoc(@roomId int)

returns int

as

begin

declare @no\_room int = 0

select @no\_room = count(Maphongtro) from DienNuoc where Maphongtro = @roomId

return @no\_room

end

* + 1. Hàm lấy thời điểm mới nhất theo mã phòng của điện nước (Sử dụng cho procedure lấy thời điểm tiếp theo)

create function func\_thoiDiemDienNuocTiepTheo(@maPhong int)

returns date

begin

declare @thoiDiem date = null

select @thoiDiem = max(Thoidiem) from DienNuoc where Maphongtro = @maPhong

if (@thoiDiem is not null)

begin

set @thoiDiem = dateadd(mm,1, @thoiDiem)

end

else

begin

set @thoiDiem = getdate()

end

return @thoiDiem

end

* + 1. Lấy tiền điện

create or alter function func\_LayTienDienCuaPhong(@thoiDiem date,@maPhong int)

returns int

as

begin

declare @tienDien int = 0

if(dbo.func\_KiemTraPhongTroCoDangDuocThue(@thoiDiem,@maPhong) = 1)

begin

select @tienDien = Tiendien

from DienNuoc

where MONTH(Thoidiem) = MONTH(@thoiDiem) and YEAR(Thoidiem) = YEAR(@thoiDiem) and Maphongtro = @maPhong

end

return @tienDien

end

Thực hiện trên C#

public int LayTienDien(DateTime thoiDiem, int maPhong)

{

int tienDien = 0;

DataTable table\_result = db.ExecuteQueryDataSet("select dbo.func\_LayTienDienCuaPhong(@thoiDiem,@maPhong)", CommandType.Text,new SqlParameter("thoiDiem",thoiDiem),

new SqlParameter("Maphong", maPhong)).Tables[0];

tienDien = Convert.ToInt32(table\_result.Rows[0][0]);

return tienDien;

}

* + 1. Lấy tiền nước

create function [dbo].[func\_LayTienNuocCuaPhong](@thoiDiem date,@maPhong int)

returns int

as

begin

declare @tienNuoc int = 0

if(dbo.func\_KiemTraPhongTroCoDangDuocThue(@thoiDiem,@maPhong) = 1)

begin

select @tienNuoc = Tiennuoc

from DienNuoc

where MONTH(Thoidiem) = MONTH(@thoiDiem) and YEAR(Thoidiem) = YEAR(@thoiDiem) and Maphongtro = @maPhong

end

return @tienNuoc

Thực hiện trên C#

public int LayTienNuoc(DateTime thoiDiem, int maPhong)

{

int tienNuoc = 0;

DataTable table\_result = db.ExecuteQueryDataSet("select dbo.func\_LayTienNuocCuaPhong(@thoiDiem,@maPhong)", CommandType.Text, new SqlParameter("thoiDiem", thoiDiem),

new SqlParameter("Maphong", maPhong)).Tables[0];

tienNuoc = Convert.ToInt32(table\_result.Rows[0][0]);

return tienNuoc;

}

* + 1. Lấy số điện cũ (Sử dụng trong trigger cập nhật điện nước)

create or alter function func\_LaySoDienCu(@roomId int,@thoiDiem date)

returns int

as

begin

declare @Sodiencu int = 0

if(dbo.func\_DemSoLuongPhongTrongBangDienNuoc(@roomId) >= 2 )

begin

declare @thangtruoc date = DATEADD(mm,-1,@thoiDiem)

select @Sodiencu = Sodienhientai from DienNuoc where Maphongtro = @roomId and Thoidiem = @thangtruoc

end

return @Sodiencu

end

* + 1. Lấy số nước cũ (Sử dụng trong trigger cập nhật điện nước)

create or alter function func\_LaySoNuocCu(@roomId int,@thoiDiem date)

returns int

as

begin

declare @Sonuoccu int = 0

if(dbo.func\_DemSoLuongPhongTrongBangDienNuoc(@roomId) >= 2 )

begin

declare @thangtruoc date = DATEADD(mm,-1,@thoiDiem)

select @Sonuoccu = Sonuochientai from DienNuoc where Maphongtro = @roomId and Thoidiem = @thangtruoc

end

return @Sonuoccu

end

* 1. Thống kê chi phí phòng trọ theo tháng

create proc [dbo].[proc\_ThongKeChiPhi] @thoidiem date

as

begin

select sum(Tiendien) as TongTienDien,

sum(Tiennuoc) as TongTienNuoc,

sum(Tiendichvu) as TongTienDV,

sum(Tienphong) as TongTienPhong,

sum(Tongtien) as TongTien

from HoaDon where DATEDIFF(month,Thoidiem,@thoidiem) = -1

end

# CHƯƠNG 4. TẠO USER VÀ PHÂN QUYỀN

Ứng dụng có một tài khoảng admin duy nhất là ChuTro có role là sysadmin

**Tài khoảng chủ trọ:** ChuTro

**Password:** 123456

Những tài khoảng user là tài khoảng của mỗi phòng. Các tài khoảng này thuộc về role admin

+ Tạo tài khoản Admin

use master

go

create login ChuTro

with password = '123456',

DEFAULT\_DATABASE = QuanLyPhongTro,

check\_expiration=off,

check\_policy=off;

go

use QuanLyPhongTro

go

CREATE USER ChuTro FOR LOGIN ChuTro;

go

ALTER SERVER ROLE sysadmin ADD MEMBER ChuTro

+ Đối với Role NguoiThue (dành cho mỗi phòng):

Use QuanLyPhongTro

go

CREATE ROLE NguoiThue

go

--------Gán các quyền trên các bảng cho role NguoiThue---------

GRANT SELECT ON View\_Ds\_DichVuDangTrangBi TO NguoiThue

GRANT SELECT ON View\_Ds\_NguoiThueTro\_HienTai TO NguoiThue

GRANT SELECT ON View\_XemNoiThatDangDuocTrangBiChoTro TO NguoiThue

GRANT SELECT ON View\_ChiTietPhongTro TO NguoiThue

grant select on Account to NguoiThue

-- Gán quyền thực thi trên các procedure, function cho role Staff

grant execute on dbo.proc\_ShowAllDienNuocTheoMaPhongTro to NguoiThue

grant execute on dbo.proc\_ShowAllHoaDonTheoMaPhongTro to NguoiThue

grant execute on dbo.proc\_ShowDienNuoc to NguoiThue

grant execute on proc\_ShowHoaDon to NguoiThue

go

+ Đối với tài khoảng ChuTro ta sử dụng server role sysadmin.

**- Procedure tạo tài khoản:**

CREATE OR ALTER PROCEDURE proc\_ThemTaiKhoang

@username NVARCHAR(25),

@pass\_word VARCHAR(25),

@ma\_phong INT

AS

BEGIN

BEGIN TRAN

BEGIN TRY

-- Thêm tài khoản

INSERT INTO Account(username,pass\_word,ma\_phong, roles) VALUES (@username, @pass\_word,@ma\_phong,'user');

DECLARE @sqlString NVARCHAR(2000)

-- Tạo tài khoản login cho nhân viên, tên người dùng và mật khẩu là tài khoản được tạo trên bảng Account

SET @sqlString = 'CREATE LOGIN [' + @username + '] WITH PASSWORD=''' + @pass\_word + ''', DEFAULT\_DATABASE=[QuanLyPhongTro], CHECK\_EXPIRATION=OFF, CHECK\_POLICY=OFF'

EXEC (@sqlString)

PRINT(@sqlString);

-- Tạo tài khoản người dùng đối với nhân viên đó trên database (tên người dùng trùng với tên login)

SET @sqlString = 'CREATE USER [' + @username + '] FOR LOGIN [' + @username +']';

EXEC (@sqlString)

PRINT(@sqlString);

-- Thêm người dùng vào vai trò quyền tương ứng (Staff hoặc Manager(sysadmin))

SET @sqlString = 'ALTER ROLE NguoiThue ADD MEMBER [' + @username + ']';

EXEC (@sqlString)

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

DECLARE @err NVARCHAR(MAX)

SELECT @err = N'Lỗi ' + ERROR\_MESSAGE()

RAISERROR(@err, 16, 1)

END CATCH

END

**- Procedure cập nhật tài khoản:**

CREATE or ALTER PROC proc\_CapNhatTaiKhoang

(@username NVARCHAR(50), @password VARCHAR(25))

AS

BEGIN

UPDATE Account

SET pass\_word = @password

WHERE username = @username

DECLARE @sql VARCHAR(200) = 'ALTER LOGIN [' + @username + '] WITH PASSWORD=''' + @password + '''';

EXEC(@sql);

END

**-Procedure xóa tài khoảng:**

CREATE or ALTER PROCEDURE proc\_XoaTaiKhoangPhong

(@username nvarchar(25))

AS

BEGIN

-- Kết thúc phiên làm việc của user

DECLARE @name\_user varchar(15);

SELECT @name\_user=username FROM ACCOUNT WHERE username = @username

DECLARE @sql varchar(100)

DECLARE @SessionID INT;

SELECT @SessionID = session\_id

FROM sys.dm\_exec\_sessions

WHERE login\_name = @name\_user;

IF @SessionID IS NOT NULL

BEGIN

SET @sql = 'kill ' + Convert(NVARCHAR(20), @SessionID)

exec(@sql)

END

BEGIN TRANSACTION;

BEGIN TRY

DELETE FROM Account where username = @username

--

SET @sql = 'DROP USER ['+ @username + ']'

exec (@sql)

--

SET @sql = 'DROP LOGIN ['+ @username + ']'

exec (@sql)

--

COMMIT;

END TRY

BEGIN CATCH

DECLARE @err NVARCHAR(MAX)

SELECT @err = N'Lỗi ' + ERROR\_MESSAGE()

RAISERROR(@err, 16, 1)

ROLLBACK TRANSACTION;

THROW;

END CATCH

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 2022 (RTM) - 16.0.1000.6

- Phần mềm Winform được viết bằng Microsoft Visual Studio 2022 - trên nền tảng WinForm App (.NET Framework) C#.

- Phần mềm được xây dựng dựa trên kiến trúc 3 lớp.

***Giao diện và chức năng của chương trình:***

## 5.1. Giao điện đăng nhập

A person carrying a bag

Description automatically generated- Chọn database phù hợp cho hệ thống, nhập username và password cho tài khoản để đăng nhập. Tùy từng loại tài khoản (ứng với username, password đã nhập) mà sẽ có các quyền truy cập khác nhau

Code C#:

private void button\_dangNhap\_Click(object sender, EventArgs e)

{

string nameDataBase = comboBox\_Database.Text;

string userName = textbox\_username.Text;

string password = textbox\_password.Text;

if (string.IsNullOrWhiteSpace(nameDataBase))

{

MessageBox.Show("Vui lòng chọn database");

}

else

{

BAL\_ConnectDatabase.ConnectDataBase(nameDataBase, userName, password);

if (BAL\_ConnectDatabase.SuccesConn(out string error) == false)

MessageBox.Show(error, "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);

else

{

if (nameDataBase == "QuanLyPhongTro")

{

balAccount = new BAL\_Account();

var table\_account = balAccount.LayThongTinTaiKhoang(userName).Tables[0];

if(table\_account.Rows.Count > 0)

{

var data = table\_account.Rows[0];

string role\_of\_account = Convert.ToString(data["roles"]);

if (role\_of\_account == "user")

{

int maPhong = Convert.ToInt32(data["ma\_Phong"]);

LoadFormHomeUser(maPhong);

}

else if (role\_of\_account == "admin")

{

LoadFormHomeAdmin();

}

}

else

{

MessageBox.Show($"Không tồn tại tài khoảng {userName}");

}

}

else

MessageBox.Show($"Kết nối thành công vào database {nameDataBase}");

}

}

}

## 5.2. Giao điện của tài khoản Admin

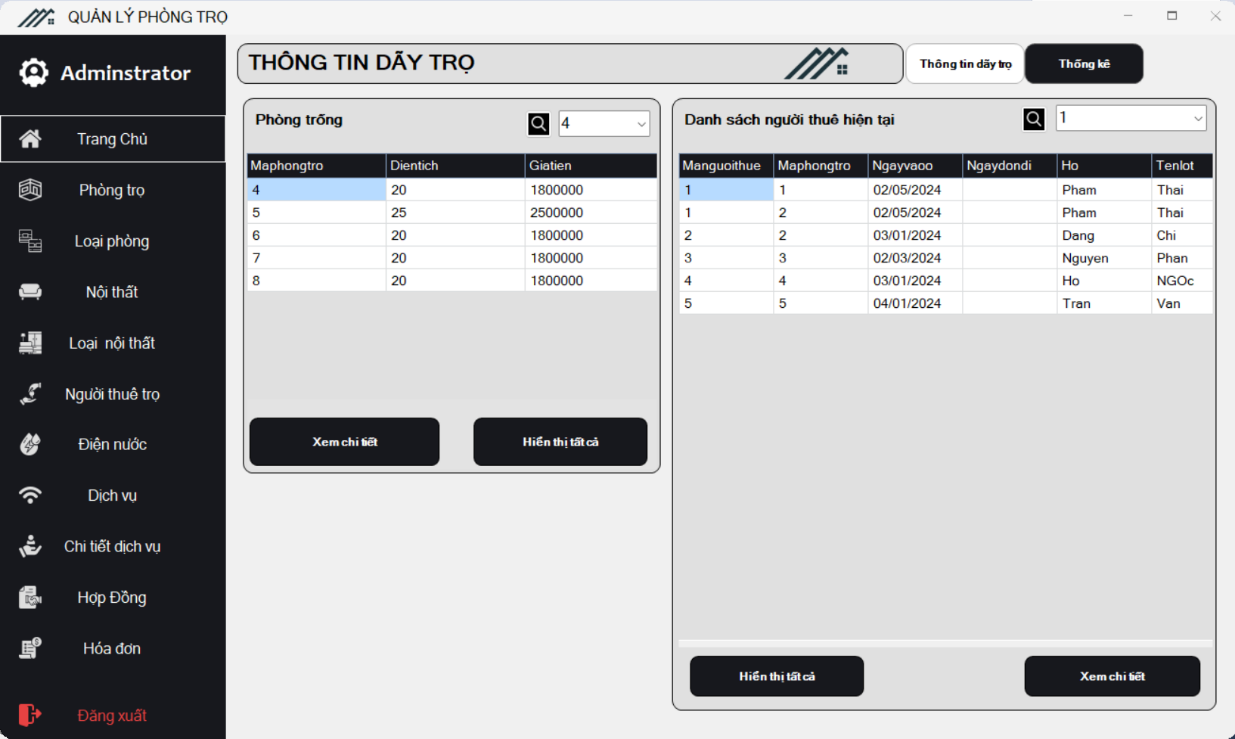
### 5.2.1. Giao diện trang chủ

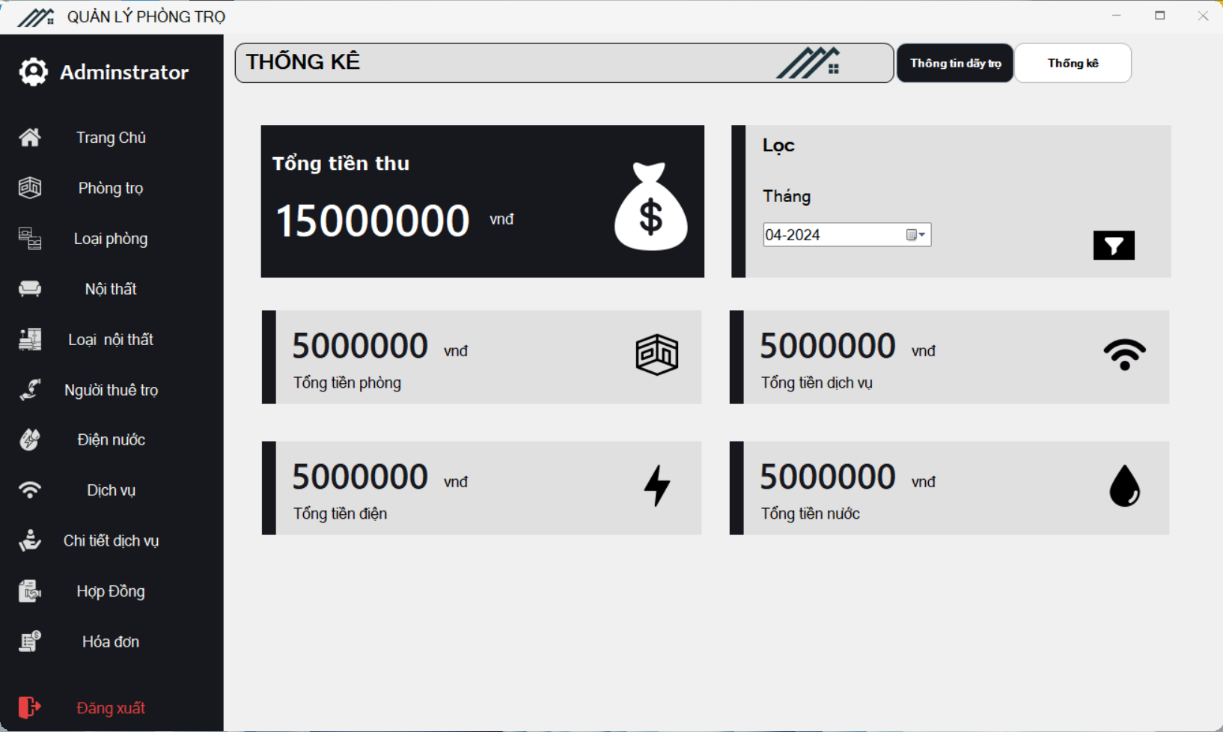
**Chức năng:**

- Hiển thị danh sách phòng đang trống và danh sách người thuê hiện tại

**Code SQL:**

- Thống kê các tổng doanh thu và khác khoản thu ( tổng tiền phòng, tổng tiền dịch vụ, tổng tiền điện, tổng tiền nước), lọc các khoản này theo tháng





### 5.2.2. Giao diện cài đặt hệ thống

**Chức năng:**

* Thêm, sửa, xóa các tài khoản để đăng nhập cho các phòng
* Tìm kiếm theo username

**A screenshot of a computer

Description automatically generated**

### 5.2.3. Giao diện quản lý phòng trọ

**Chức năng:**

* Thêm, sửa, xóa các thông tin các phòng
* Tìm kiểm theo phòng

**A screenshot of a computer

Description automatically generated**

### 5.2.4. Giao diện quản lý loại phòng trọ

**Chức năng:**

* Thêm, sửa, xóa các thông tin các loại phòng trọ
* Tìm kiểm theo mã loại phòng

**A screenshot of a computer

Description automatically generated**

### 5.2.5. Giao diện quản lý nội thất

**Chức năng:**

* Thêm, sửa, xóa các thông tin nội thất
* A screenshot of a computer

  Description automatically generatedTìm kiểm theo mã nội thất

### 5.2.6. Giao diện quản lý loại nội thất

**Chức năng:**

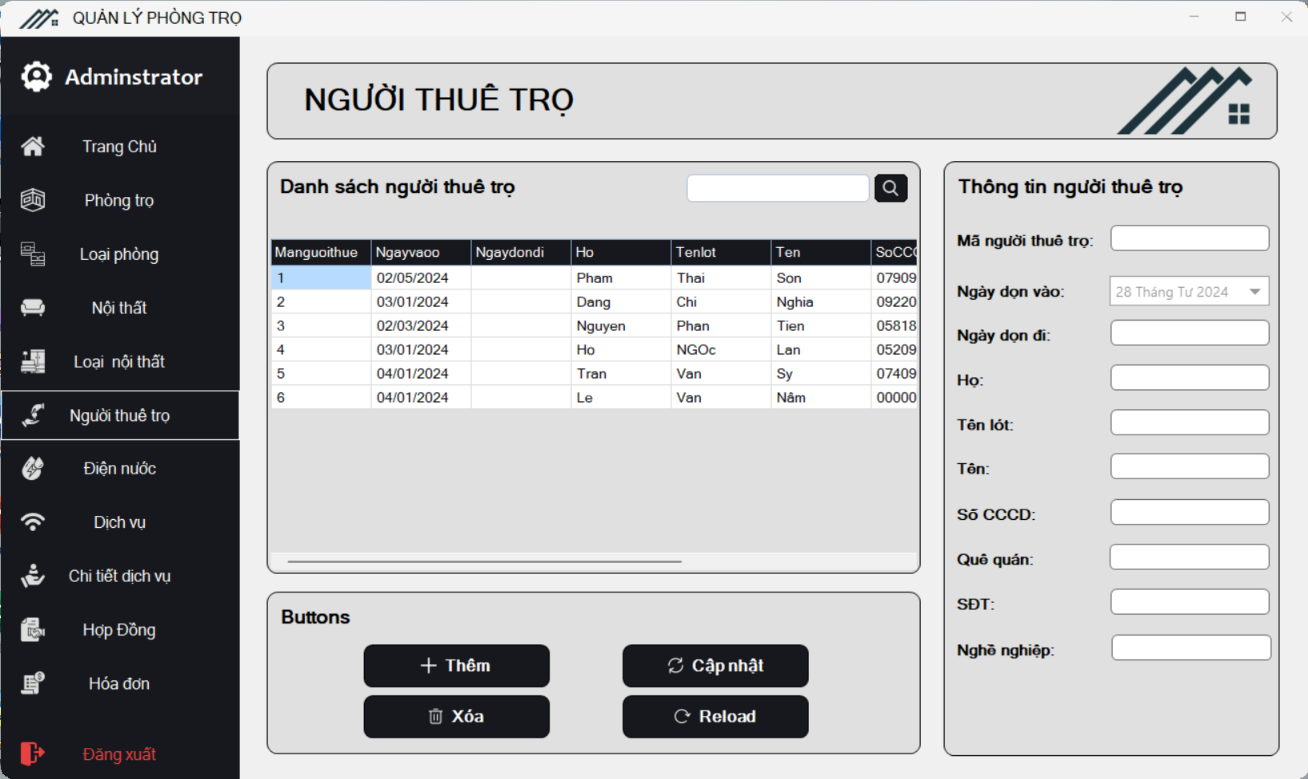
* Thêm, sửa, xóa các thông tin loại nội thất
* Tìm kiểm theo mã loại nội thất

**A screenshot of a computer

Description automatically generated**

### 5.2.7. Giao diện quản lý người thuê trọ

**Chức năng:**

* Thêm, sửa, xóa các thông tin người thuê trọ
* Tìm kiểm theo mã người thuê trọ

### 5.2.8. Giao diện quản lý điện nước

**Chức năng:**

* Thêm, sửa, xóa các thông tin điện nước
* Tìm kiểm theo mã điện nước

A screenshot of a computer

Description automatically generated

### 5.2.9. Giao diện quản lý dịch vụ

**Chức năng:**

* Thêm, sửa, xóa các thông tin dịch vụ
* Tìm kiểm theo mã dịch vụ

**A screenshot of a computer

Description automatically generated**

### 5.2.10. Giao diện quản lý chi tiết dịch vụ

**Chức năng:**

* Thêm, sửa, xóa các chi tiết dịch vụ
* Tìm kiểm theo mã chi tiết dịch vụ

**A screenshot of a computer

Description automatically generated**

### 5.2.11. Giao diện quản lý hợp đồng

**Chức năng:**

* Thêm, sửa, xóa các hợp đồng
* A screenshot of a computer

  Description automatically generatedTìm kiểm theo mã hợp đồng

### 5.2.12. Giao diện quản lý hóa đơn

**Chức năng:**

* Thêm, sửa, xóa các hóa đơn
* Tìm kiểm theo mã hóa đơn
* Xuất hóa đơn theo phòng

A screenshot of a computer

Description automatically generated