A picture containing font, logo, graphics, brand

Description automatically generated

**FPT POLYTECHNIC**

**BÁO CÁO GIAI ĐOẠN I**

***DỰ ÁN 1: QUẢN LÝ PHÒNG TRỌ***

**NGÀNH CÔNG NGHỆ THÔNG TIN (UDPM – JAVA)**

A house with trees and a sign

Description automatically generated with low confidence

**GIẢNG VIÊN HƯỚNG DẪN: NGUYỄN LÝ MINH THY**

**LỚP: SD18202 - Mã môn: PRO1041**

**NHÓM 5:** Vũ Thị Thu Phương (PS26491) – Trưởng nhóm

Chu Minh Thành (PS26487) – Thành viên

Ngô Khắc Sáng (PS26482) – Thành viên

Cao Hoàng Duy (PS26495) – Thành viên

Nguyễn Trường Sơn (PS26480) – Thành viên

TP Hồ Chí Minh, 2023

Mục lục

[**1.** **Giới thiệu dự án:** 3](#_Toc142491149)

[1.1. Hiện trạng thực tế: 3](#_Toc142491150)

[1.2. Lý do chọn đề tài: 3](#_Toc142491151)

[1.3. Lập kế hoạch dự án: 4](#_Toc142491152)

[**2.** **Phân tích yêu cầu khách hàng:** 5](#_Toc142491153)

[2.1. Sơ đồ use case: 5](#_Toc142491154)

[2.2. Đặc tả yêu cầu hệ thống (SRS): 6](#_Toc142491155)

[2.2.1. Chủ trọ: 6](#_Toc142491156)

[2.2.2. Người thuê/ Người dùng: 8](#_Toc142491157)

[2.2.3. Đăng nhập / Đăng xuất: 9](#_Toc142491158)

[2.2.4. Đổi mật khẩu, quên mật khẩu: 9](#_Toc142491159)

[2.2.5. Yêu cầu về bảo mật: 9](#_Toc142491160)

[2.3. Các sơ đồ và yêu cầu hệ thống: 10](#_Toc142491161)

[2.3.1. Sơ đồ phân rã chức năng phần mềm: 10](#_Toc142491162)

[2.3.2. Sơ đồ triển khai: 11](#_Toc142491163)

[2.3.3. Sơ đồ DFD: 11](#_Toc142491164)

[2.3.4. Sơ đồ hoạt động: 12](#_Toc142491165)

[2.3.5. Sơ đồ lớp 19](#_Toc142491166)

[2.3.6. Yêu cầu hệ thống: 19](#_Toc142491167)

[**3.** **Thiết kế ứng dụng:** 19](#_Toc142491168)

[3.1. Mô hình công nghệ ứng dụng: 19](#_Toc142491169)

[3.2. Thực thể: 20](#_Toc142491170)

[3.2.1. Sơ đồ quan hệ thực thể ERD: 20](#_Toc142491171)

[3.2.2. Sơ đồ quan hệ thực thể EER: 20](#_Toc142491172)

[3.2.3. Sơ đồ Database: 21](#_Toc142491173)

[3.2.4. Chi tiết thực thể: 21](#_Toc142491174)

[3.3. Cơ sở dữ liệu: 24](#_Toc142491175)

[3.3.1 Tạo cơ sở dữ liệu 24](#_Toc142491176)

[3.3.2. Tìm dữ liệu cho cơ sở dữ liệu 28](#_Toc142491177)

[3.4. Thiết kế giao diện 44](#_Toc142491178)

[3.4.1. Logo 44](#_Toc142491179)

[3.4.2. Giao diện login 45](#_Toc142491180)

[3.4.3. Giao diện Chủ Trọ 49](#_Toc142491181)

[3.4.4. Giao diện Người Dùng – Người Thuê 54](#_Toc142491182)

[**4.** **Thực hiện dự án** 58](#_Toc142491183)

[4.1. Tạo giao diện với Swing: 58](#_Toc142491184)

[4.1.1. Logo. 58](#_Toc142491185)

[4.1.2. Giao diện login. 59](#_Toc142491186)

[4.1.3. Giao diện chủ trọ. 61](#_Toc142491187)

[4.1.4. Giao diện người dùng. 69](#_Toc142491188)

[5.1. Tạo CSDL với SQL Server: 73](#_Toc142491189)

[5.1.1. Sơ đồ quan hệ: 73](#_Toc142491190)

[5.1.2. Chi tiết các bảng: 73](#_Toc142491191)

[5.1.3. Thủ tục lưu: 76](#_Toc142491192)

[5.2. Lập trình CSDL: 76](#_Toc142491193)

[5.2.1. JDBC: 77](#_Toc142491194)

[5.2.2. Entity class và DAO class: 77](#_Toc142491195)

[5.3. Thư viện tiện ích: 125](#_Toc142491196)

[5.4. Lập trình nghiệp vụ: 125](#_Toc142491197)

[**6.** **Kiểm thử phần mềm và sửa lỗi:** 130](#_Toc142491198)

[**7.** **Đóng gói và triển khai:** 141](#_Toc142491199)

[7.1. Sản phẩm phần mềm: 141](#_Toc142491200)

[7.2. Hướng dẫn cài đặt: 141](#_Toc142491201)

[**8.** **Kết luận:** 145](#_Toc142491202)

[8.1. Thuận lợi: 145](#_Toc142491203)

[8.2. Khó khăn: 145](#_Toc142491204)

# **Giới thiệu dự án:**

## Hiện trạng thực tế:

Hiện trạng quản lý phòng trọ bằng công nghệ thông tin đã đem lại nhiều lợi ích và tiện ích cho việc quản lý và vận hành phòng trọ. Dưới đây là tóm tắt về hiện trạng này:

* Hệ thống quản lý thông tin khách hàng: Công nghệ thông tin giúp tổ chức thông tin khách hàng, bao gồm thông tin cá nhân, hợp đồng thuê, lịch thanh toán và các yêu cầu đặc biệt. Các công cụ phần mềm quản lý thông tin chính xác và tiện lợi, giúp quản lý dễ dàng tìm kiếm và cập nhật thông tin khách hàng một cách nhanh chóng.
* Hệ thống đặt phòng trực tuyến: Khách hàng có thể đặt phòng và thực hiện thanh toán trực tuyến thông qua các ứng dụng di động hoặc trang web. Điều này tiết kiệm thời gian, giúp khách hàng có thể tìm hiểu thông tin về phòng trọ, giá cả và tiện nghi trước khi đặt phòng.
* Hệ thống thanh toán điện tử: Quản lý phòng trọ có thể sử dụng các hình thức thanh toán điện tử như ví điện tử, thẻ tín dụng hoặc chuyển khoản trực tuyến để thu tiền phòng. Điều này giúp giảm thiểu rủi ro mất mát tiền mặt và giúp quản lý dễ dàng theo dõi các giao dịch.
* Hệ thống giám sát và bảo mật: Công nghệ thông tin cung cấp các giải pháp giám sát và bảo mật như hệ thống camera an ninh, điều khiển truy cập thông qua mã pin hoặc thẻ từ. Điều này giúp giảm thiểu các vụ vi phạm, bảo vệ an ninh cho cư dân và tài sản trong phòng trọ.
* Hệ thống quản lý dịch vụ: Công nghệ thông tin cho phép quản lý phòng trọ quản lý dịch vụ như sửa chữa và bảo trì phòng, vệ sinh, điện, nước và internet. Hệ thống có thể tự động cập nhật và thông báo cho quản lý về các yêu cầu bảo trì, đảm bảo dịch vụ được cung cấp đúng thời điểm và đúng chất lượng.

Tóm lại, quản lý phòng trọ bằng công nghệ thông tin đã giúp tiết kiệm thời gian, nâng cao hiệu quả quản lý và cung cấp dịch vụ tốt hơn cho khách hàng. Các hệ thống này đang ngày càng phổ biến trong ngành bất động sản và có tiềm năng phát triển trong tương lai.

## Lý do chọn đề tài:

Thông qua khảo sát thực tế, nhìn chung nhà trọ hoạt động khá tốt. Nhưng bên cạnh đó vẫn còn nhiều công đoạn chiếm nhiều thời gian do thao tác thủ công. Nguy cơ sai sót khi nhập sổ sách, khi thống kê báo cáo, và việc lưu trữ thông tin chủ yếu trên giấy tờ vì thế gây khó khăn trong việc bảo quản, lưu trữ. Nhóm em chọn đề tài để công nghệ hóa các bước lưu trữ thông tin để thuận tiện hơn trong việc lưu trữ, tra cứu và chỉnh sửa.

## Lập kế hoạch dự án:

Giai đoạn 1: Phân tích dự án

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mô tả** | **Thực hiện** | **Bắt đầu** | **Kết thúc** | **Trạng thái** |
| Chọn đề tài | Cả nhóm | 27/06/2023 | 27/06/2023 | OFF |
| Phân tích đề tài | Cả nhóm | 27/06/2023 | 27/06/2023 | OFF |
| Vẽ sơ đồ ERD | Sáng | 30/06/2023 | 03/07/2023 | OFF |
| Vẽ sơ đồ phân rã chức năng phần mềm | Phương | 30/06/2023 | 03/07/2023 | OFF |
| Vẽ sơ đồ Use Case | Thành | 30/06/2023 | 03/07/2023 | OFF |
| Vẽ sơ đồ DFD | Sơn | 30/06/2023 | 03/07/2023 | OFF |
| Vẽ sơ đồ hoạt động | Duy | 01/07/2023 | 03/07/2023 | OFF |
| Vẽ sơ đồ lớp | Sơn | 01/07/2023 | 03/07/2023 | OFF |
| Vẽ sơ đồ kiến trúc | Sáng | 01/07/2023 | 03/07/2023 | OFF |
| Vẽ sơ đồ mô hình công nghệ | Duy | 01/07/2023 | 03/07/2023 | OFF |
| **Tạo CSDL** | Thành | 01/07/2023 | 03/07/2023 | OFF |
| **Tìm dữ liệu cho CSDL** | Thành | 01/07/2023 | 03/07/2023 | OFF |
| Thiết kế Logo | Phương | 01/07/2023 | 03/07/2023 | OFF |
| Thiết kế giao diện | Phương | 01/07/2023 | 03/07/2023 | OFF |

Giai đoạn 2: Thực hiện dự án

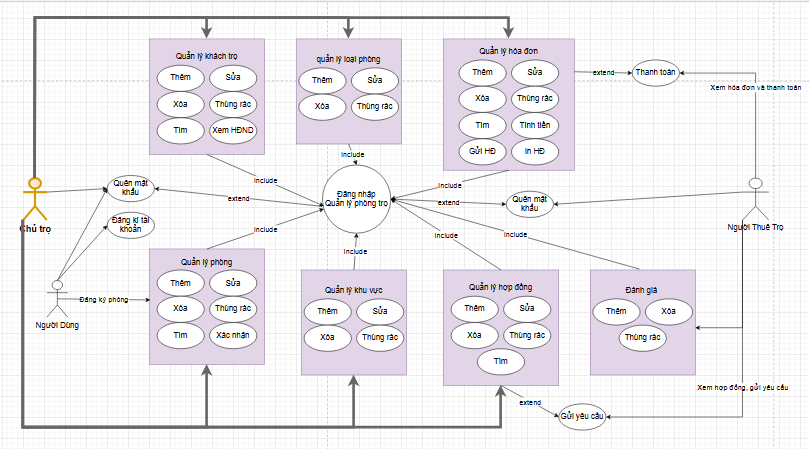
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mô tả** | **Thực hiện** | **Bắt đầu** | **Kết thúc** | **Trạng thái** |
| Code login | Thành | 04/07/2023 | 14/07/2023 | OFF |
| Trang chủ chung panel | Phương | 04/07/2023 | 10/07/2023 | OFF |
| Menu của chủ trọ | Phương | 04/07/2023 | 10/07/2023 | OFF |
| Menu của người dùng | Sơn | 04/07/2023 | 10/07/2023 | OFF |
| Code quản lý người dùng | Sáng | 04/07/2023 | 22/07/2023 | OFF |
| Code quản lý hoạt động người dùng | Sáng | 04/07/2023 | 22/07/2023 | OFF |
| Code quản lý khu vực và loại phòng | Sơn | 04/07/2023 | 22/07/2023 | OFF |
| Code quản lý phòng trọ | Sơn | 04/07/2023 | 22/07/2023 | OFF |
| Code quản lý hợp đồng | Thành | 04/07/2023 | 14/07/2023 | OFF |
| Code quản lý hóa đơn | Duy | 04/07/2023 | 25/07/2023 | OFF |
| Code in hóa đơn | Phương | 04/07/2023 | 26/07/2023 | OFF |
| Code in hợp đồng | Phương | 04/07/2023 | 26/07/2023 | OFF |
| Code gửi email hóa đơn | Thành | 04/07/2023 | 29/07/2023 | OFF |
| Code thông tin cá nhân và lịch sử hoạt động | Sáng | 17/07/2023 | 03/08/2023 | OFF |
| Code trang chủ người dùng | Phương | 17/07/2023 | 03/08/2023 | OFF |
| Code đánh giá | Phương | 17/07/2023 | 03/08/2023 | OFF |
| Code hợp đồng | Thành | 17/07/2023 | 29/07/2023 | OFF |
| Code hóa đơn | Duy | 17/07/2023 | 25/07/2023 | OFF |

Giai đoạn 2: Báo cáo và đóng gói

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mô tả** | **Thực hiện** | **Bắt đầu** | **Kết thúc** | **Trạng thái** |
| Viết test case | Cả nhóm | 30/07/2023 | 07/08/2023 | OFF |
| Viết báo cáo | Cả nhóm | 01/08/2023 | 09/08/2023 | OFF |
| Đóng gói sản phẩm | Phương | 10/08/2023 | 10/08/2023 | OFF |

# **Phân tích yêu cầu khách hàng:**

## Sơ đồ use case:



## Đặc tả yêu cầu hệ thống (SRS):

### Chủ trọ:

* Mô tả chức năng: Chức năng quản lý phòng trọ được sử dụng để quản lí phòng trọ hoặc dãy các phòng trọ, khu vực trọ.
* Yêu cầu này bao gồm: Quản lý người dùng , quản lí hoạt động người dùng, quản lí phòng trọ , khu vực và loại phòng, quản lí hợp đồng, quản lí hóa đơn, quản lí thông tin cá nhân, quản lí lịch sử hoạt động.

#### Quản lý người dùng:

* Chức năng thêm: Thêm vào danh sách thông tin của người dùng mới với các thuộc tính: họ tên, số điện thoại, email, địa chỉ.
* Chức năng cập nhật : Cập nhật lại các thông tin của người dùng.
* Chức năng xóa: Xóa thông tin người dùng khỏi danh sách quản lý.
* Chức năng tìm kiếm: Tìm kiếm thông tin người dùng theo họ tên, số điện
* Dữ liệu liên quan: Thông tin chung cư gồm: Mã người dùng, họ tên, số điện thoại, email, địa chỉ.
* Đối tượng sử dụng: Chủ nhà trọ

#### Quản lý hoạt động người dùng:

* Mô tả chức năng: Chức năng quản lý hoạt động người dùng thuộc chủ nhà trọ, bảng danh sách hoạt động người dùng
* Dữ liệu liên quan: mã người dùng, thời gian đăng nhập, thời gian đăng xuất, nội dung.
* Đối tượng sử dụng: Chủ nhà trọ

#### Quản lý phòng trọ:

* Mô tả chức năng: Chủ nhà trọ có quyền tìm kiếm phòng trọ, xem thông tin phòng trọ và có thể cập nhật trạng thái phòng trọ các chức năng thêm, sửa, xóa, thùng rác, tìm kiếm.
* Chức năng thêm: Thêm vào danh sách thông tin của phòng trọ mới với các thuộc tính: Mã loại phòng ,mã người dùng, mã khu vực, tên phòng, diện tích, giá phòng, địa chỉ, mô tả, tình trạng.
* Chức năng cập nhật: Cập nhật lại các thông tin của phòng trọ
* Chức năng xóa: Xóa các thông tin của phòng trọ đó
* Chức năng tìm kiếm: Tìm kiếm thông tin phòng trọ theo mã phòng trọ
* Chức năng thùng rác: Khi nhấn xóa sẽ chuyển dữ liệu vào thùng rác có thể khôi phục dữ liệu hoặc xóa vĩnh viễn.
* Dữ liệu liên quan: Mã loại phòng ,mã người dùng, mã khu vực, tên phòng, diện tích, giá phòng, địa chỉ, mô tả, tình trạng.
* Đối tượng sử dụng: Chủ nhà trọ

#### Quản lý khu vực và loại phòng:

* Mô tả chức năng: Chủ nhà trọ sẽ kiểm soát thông tin củahợp đồng
* Chức năng thêm: Thêm người thuê trọ mới vào danh sách người thuê trọ với các thuộc tính: tên khu vực, tầng, dãy,tên loại phòng.
* Chức năng cập nhật: Cập nhật lại các thông tin của người thuê trọ
* Chức năng xóa: Xóa thông tin người thuê trọ ra khỏi danh sách người thuê trọ
* Chức năng thùng rác: Khi nhấn xóa sẽ chuyển dữ liệu vào thùng rác có thể khôi phục dữ liệu hoặc xóa vĩnh viễn.
* Dữ liệu liên quan: tên khu vực, tầng, dãy,tên loại phòng.
* Đối tượng sử dụng: Chủ nhà trọ

#### Quản lý hợp đồng:

* Mô tả chức năng: Chủ nhà trọ sẽ có quyền thay đổi thông tin hợp đồng.
* Chức năng thêm: Thêm người thuê trọ mới vào danh sách người thuê trọ với các thuộc tính: mã phòng trọ, mã người dùng, ngày bd thuê, ngày kt thuê, đặt cọc, giá phòng, giá điện giá nước, giá internet, giá rác, tình trạng, ngày tạo.
* Chức năng cập nhật: Cập nhật lại các thông tin của người thuê trọ
* Chức năng xóa: Xóa thông tin người thuê trọ ra khỏi danh sách người thuê trọ
* Chức năng thùng rác: Khi nhấn xóa sẽ chuyển dữ liệu vào thùng rác có thể khôi phục dữ liệu hoặc xóa vĩnh viễn.
* Dữ liệu liên quan: Dữ liệu liên quan: mã phòng trọ, mã người dùng, ngày bd thuê, ngày kt thuê, đặt cọc, giá phòng, giá điện giá nước, giá internet, giá rác, tình trạng, ngày tạo.
* Đối tượng sử dụng: Chủ nhà trọ

#### Quản lý hóa đơn:

* Mô tả chức năng: chức năng giúp chủ nhà trọ tạo hóa đơn mới, thêm hóa đơn vào danh sách hóa đơn, in hóa đơn và gửi hóa đơn đến người thuê trọ.
* Chức năng thêm hóa đơn: Thêm hóa đơn mới vào danh sách hóa đơn với các thuộc tính: mã hợp đồng, ngày tạo, ngày bd, ngày kt, số điện cũ, số điện mới, số nước cũ, số nước mới, tiền internet, tiền rác, khấu trừ, nợ, tổng cộng , tình trạng.
* Chức năng cập nhật: Cập nhật thông tin hóa đơn
* Chức năng xóa: Xóa hóa đơn khỏi danh sách hóa đơn
* Chức năng thùng rác: Khi nhấn xóa sẽ chuyển dữ liệu vào thùng rác có thể khôi phục dữ liệu hoặc xóa vĩnh viễn.
* Chức năng tìm kiếm: Tìm kiếm hóa đơn theo mã hóa đơn
* Chức năng in hóa đơn: In hóa đơn
* Chức năng gửi hóa đơn: Gửi hóa đơn đến người thuê trọ đăng ký hợp đồng
* Dữ liệu liên quan: mã hợp đồng, ngày tạo, ngày bd, ngày kt, số điện cũ, số điện mới, số nước cũ, số nước mới, tiền internet, tiền rác, khấu trừ, nợ, tổng cộng , tình trạng.
* Đối tượng sử dụng: Chủ nhà trọ

#### Thông tin cá nhân:

* Mô tả chức năng: chức năng này quản lí thông tin của 1 cá nhân với vai trò phân quyền.
* Chức năng cập nhật thông tin: Cập nhật trạng thái thông tin cá nhân : mã người dùng, họ tên, email, số điện thoại, địa chỉ, mật khẩu, vai trò.
* Dữ liệu liên quan: mã người dùng, họ tên, email, số điện thoại, địa chỉ, mật khẩu, vai trò.
* Đối tượng sử dụng: Chủ nhà trọ

#### Lịch sử hoạt động:

* Mô tả chức năng: chức năng này quản lí lịch sử đã đăng nhập của người dùng.
* Dữ liệu liên quan: mã session, mã người dùng, tg đăng nhập, tg đăng xuất, nội dung
* Đối tượng sử dụng: Chủ nhà trọ

### Người thuê/ Người dùng:

#### Đánh giá:

* Mô tả chức năng: Người dùng đăng nhập vào hệ thống để đánh giá phòng trọ.
* Chức năng gửi: Chức năng sẽ được gửi cho chủ nhà trọ với các thông tin: mã người dùng, mã phòng trọ, đánh giá, nội dung
* Chức năng xóa: Xóa hóa đơn đánh giá
* Chức năng thùng rác: Khi nhấn xóa sẽ chuyển dữ liệu vào thùng rác có thể khôi phục dữ liệu hoặc xóa vĩnh viễn.
* Dữ liệu liên quan: mã người dùng, mã phòng trọ, đánh giá, nội dung
* Đối tượng sử dụng: người thuê trọ

#### Hợp đồng:

* Mô tả chức năng: người thuê trọ sẽ gửi yêu cầu hợp đồng cho chủ nhà trọ
* chức năng gửi yêu cầu: gửi yêu cầu theo tình trạng của hợp đồng:
* còn hạn, hết hạn, yêu cầu hủy hợp đồng.
* Dữ liệu liên quan: mã phòng trọ, mã người dùng, ngày bd thuê, ngày kt thuê, đặt cọc, giá phòng, giá điện giá nước, giá internet, giá rác, tình trạng, ngày tạo.
* Đối tượng sử dụng: người thuê trọ

#### Hóa đơn:

* Mô tả chức năng: người thuê trọ có thể thanh toán hóa đơn cho chủ nhà trọ.
* Chức năng thanh toán hóa đơn: người thuê trọ có thể thực hiện thanh toán hóa đơn với các thông tin sau: mã hợp đồng, ngày bắt đầu, ngày kết thúc, số điện cũ, số điện mới, số nước cũ, số nước mới, tiền internet, tiền rác, khấu trừ, nợ , tổng cộng , tình trạng.
* Dữ liệu liên quan: mã hợp đồng, ngày bắt đầu, ngày kết thúc, số điện cũ, số điện mới, số nước cũ, số nước mới, tiền internet, tiền rác, khấu trừ, nợ , tổng cộng , tình trạng.
* Đối tượng sử dụng: người thuê trọ

### Đăng nhập / Đăng xuất:

* Mô tả chức năng: chức năng này quản lý tài khoản và phân quyền tài khoản bao gồm: tài khoản chủ quản lý và nhiều tài khoản người thuê trọ. Đăng xuất có chức năng thoát ứng dụng.
* Dữ liệu liên quan: tên tài khoản, mật khẩu
* Đối tượng sử dụng: Quản lý, người thuê trọ

### Đổi mật khẩu, quên mật khẩu:

* Mô tả chức năng: Chủ nhà người thuê trọ có quyền đổi mật khẩu và quên mật khẩu chỉ cần nhập email, mã xác nhận ngẫu nhiên sẽ được gửi về email, nhập mã xác nhận và đổi mật khẩu mới.
* Dữ liệu liên quan: đổi mật khẩu, mã xác nhận , email.
* Đối tượng sử dụng: Chủ nhà trọ, người thuê trọ

### Yêu cầu về bảo mật:

* Chủ nhà trọ và người dùng phải đăng nhập mới sử dụng được phần mềm
* Chủ nhà trọ được phép thực hiện tất cả các chức năng
* Hoạt động người dùng không được phép xóa dữ liệu và cũng không được xem thông tin về doanh thu

## Các sơ đồ và yêu cầu hệ thống:

### Sơ đồ phân rã chức năng phần mềm:

A diagram of a company

Description automatically generated

### Sơ đồ triển khai:

A diagram of a computer network

Description automatically generated

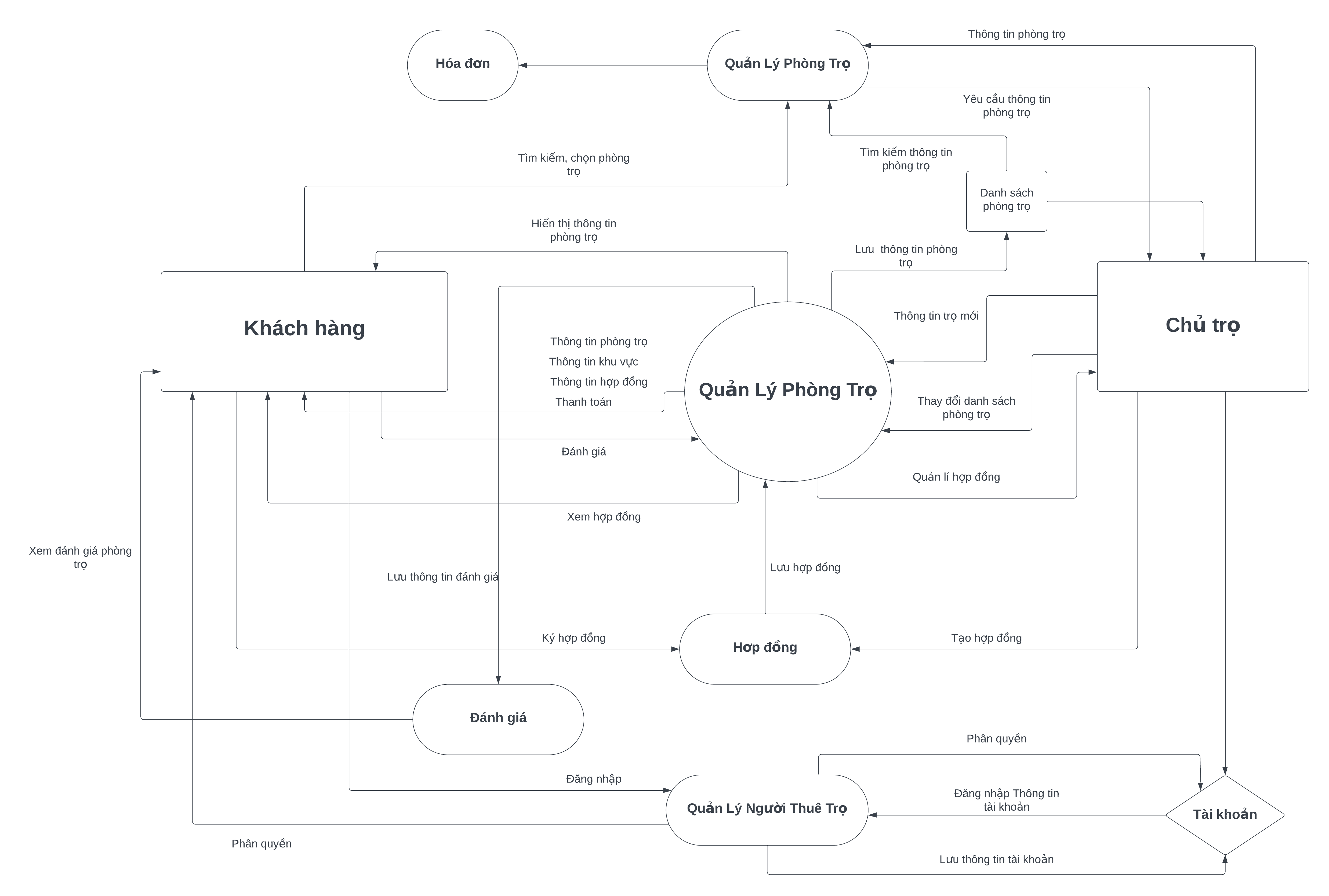
### Sơ đồ DFD:

* DFD level 0:

A diagram of a circle with arrows

Description automatically generated

* DFD level 1:



### Sơ đồ hoạt động:

* Chủ trọ:

A diagram of a flowchart

Description automatically generatedA diagram of a company

Description automatically generatedA diagram of a flowchart

Description automatically generatedA diagram of a company

Description automatically generatedA diagram of a diagram

Description automatically generated

A diagram of a diagram

Description automatically generated

* Người dùng/ Người thuê:

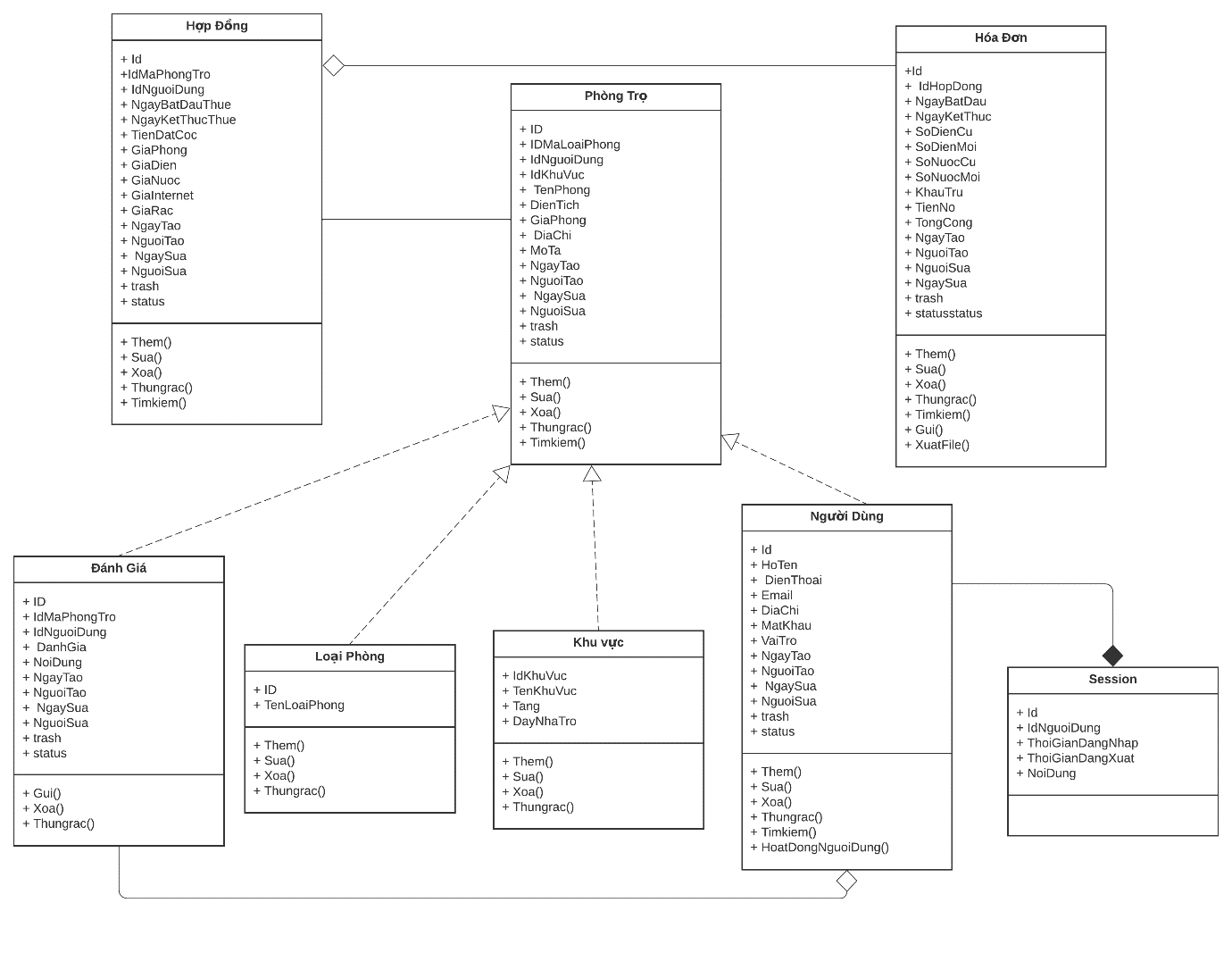
A diagram of a work flow

Description automatically generatedA diagram of a flowchart

Description automatically generatedA diagram of a flowchart

Description automatically generated

### Sơ đồ lớp



### Yêu cầu hệ thống:

* SQL Server
* JDK 19

# **Thiết kế ứng dụng:**

## Mô hình công nghệ ứng dụng:

A screenshot of a computer

Description automatically generated

## Thực thể:

### Sơ đồ quan hệ thực thể ERD:

A diagram of a diagram

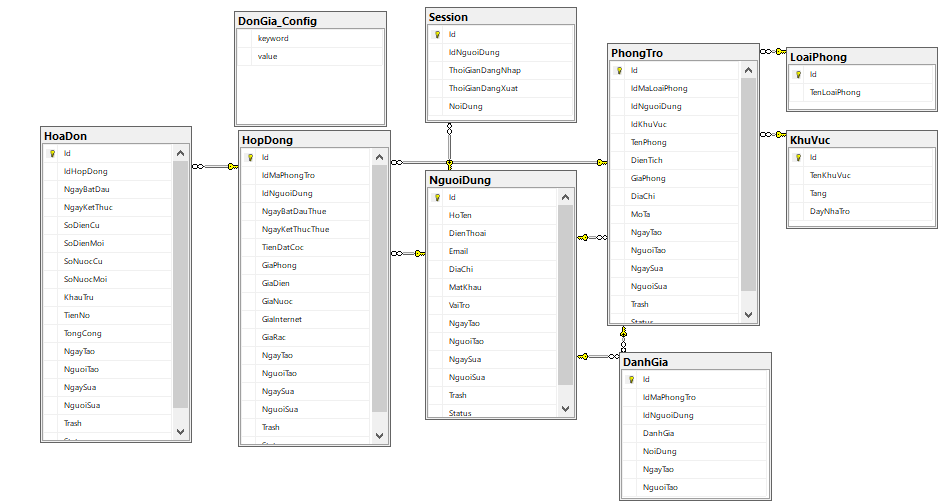
Description automatically generated

### Sơ đồ quan hệ thực thể EER:

A screenshot of a computer screen

Description automatically generated

### Sơ đồ Database:



### Chi tiết thực thể:

#### Khu vực:

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu | Ràng buộc | Ghi chú |
| Id | int | Identity(1,1), PK | Mã khu vực |
| TenKhuVuc | Nvarchar(255) | Not null | Tên khu vực |
| Tang | int |  | Tầng |
| DayNhaTro | int |  | Dãy nhà trọ |

#### Loại phòng:

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu | Ràng buộc | Ghi chú |
| Id | int | Identity(1,1), PK | Mã loại phòng |
| TenLoaiPhong | Nvarchar(255) | Not null | Tên loại phòng |

#### Phòng trọ:

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu | Ràng buộc | Ghi chú |
| Id | bigint | Identity(1,1), PK | Mã phòng trọ |
| IdMaLoaiPhong | int | FK | Mã loại phòng |
| IdNguoiDung | int | FK | Mã người dùng |
| TenPhong | Nvarchar(100) | Not null | Tên phòng |
| DienTich | Float |  | Diện tích |
| GiaPhong | Float |  | Giá phòng |
| DiaChi | Nvarchar(100) | Not null | Địa chỉ |
| MoTa | Nvarchar(100) | Not null | Mô tả |
| NgayTao | datetime |  | Ngày tạo |
| NguoiTao | Nvarchar(100) |  | Người tạo |
| NgaySua | datetime |  | Ngày sửa |
| NguoiSua | Nvarchar(100) |  | Người sửa |
| Trash | int | Default 1 | Thùng rác  0 là tạm xóa  1 là không xóa |
| Status | int |  | Trạng thái:  0 là chưa thuê  1 là đã thuê  2 là đang chờ duyệt |

#### Người dùng:

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu | Ràng buộc | Ghi chú |
| Id | bigint | Identity(1,1), PK | Mã người dùng |
| HoTen | Nvarchar(100) | Not null | Họ tên |
| DienThoai | Varchar(10) | Not null, unique | Số điện thoại |
| Email | Nvarchar(100) | Not null, unique | Email |
| MatKhau | Nvarchar(100) | Not null | Mật khẩu |
| DiaChi | Nvarchar(100) | Not null | Địa chỉ |
| VaiTro | Nvarchar(100) | Not null | Vai trò |
| NgayTao | datetime |  | Ngày tạo |
| NguoiTao | Nvarchar(100) |  | Người tạo |
| NgaySua | datetime |  | Ngày sửa |
| NguoiSua | Nvarchar(100) |  | Người sửa |
| Trash | int | Default 1 | Thùng rác:  0 là tạm xóa  1 là không xóa |
| Status | int |  | Trạng thái:  0 là người thuê trọ (Tenants)  1 là người dùng (User)  2 là chủ trọ (Admin) |

#### Session:

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu | Ràng buộc | Ghi chú |
| Id | int | Identity(1,1), PK | Mã session |
| IdNguoiDung | Bigint | Not null | Mã người dùng |
| ThoiGianDangNhap | datetime |  | Thời gian đăng nhập |
| ThoiGianDangXuat | datetime |  | Thời gian đăng xuất |
| NoiDung | Nvarchar(100) |  | Nội dung |

#### Đánh giá:

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu | Ràng buộc | Ghi chú |
| Id | bigint | Identity(1,1), PK | Mã đánh giá |
| IdMaPhongTro | bigint | FK | Mã phòng trọ |
| IdNguoiDung | bigint | FK | Mã người dùng |
| DanhGia | int | Not null | Đánh giá |
| NoiDung | Nvarchar(100) | Not null | Nội dung |
| NgayTao | datetime |  | Ngày tạo |
| NguoiTao | Nvarchar(100) |  | Người tạo |
| NgaySua | datetime |  | Ngày sửa |
| NguoiSua | Nvarchar(100) |  | Người sửa |
| Trash | int | Default 1 | Thùng rác:  0 là tạm xóa  1 là không xóa |
| Status | int |  | Trạng thái |

#### Hợp đồng:

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu | Ràng buộc | Ghi chú |
| Id | bigint | Identity(1,1), PK | Mã hợp đồng |
| IdMaPhongTro | bigint | FK | Mã phòng trọ |
| IdNguoiDung | bigint | FK | Mã người dùng |
| NgayBatDauThue | Datetime |  | Ngày BĐ thuê |
| NgayKetThucThue | Datetime |  | Ngày KT thuê |
| TienDatCoc | Float |  | Tiền đặt cọc |
| GiaPhong | Float |  | Giá phòng |
| GiaDien | Float |  | Giá điện |
| GiaNuoc | Float |  | Giá nước |
| GiaInternet | Float |  | Giá internet |
| GiaRac | Float |  | Giá rác |
| NgayTao | datetime |  | Ngày tạo |
| NguoiTao | Nvarchar(100) |  | Người tạo |
| NgaySua | datetime |  | Ngày sửa |
| NguoiSua | Nvarchar(100) |  | Người sửa |
| Trash | int | Default 1 | Thùng rác:  0 là tạm xóa  1 là không xóa |
| Status | int |  | Trạng thái:  0 là hết hạn  1 là đang sử dụng  2 là bị hủy |

#### Hóa đơn:

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu | Ràng buộc | Ghi chú |
| Id | bigint | Identity(1,1), PK | Mã hóa đơn |
| IdHopDong | bigint | Not null | Mã hợp đồng |
| NgayBatDau | Datetime |  | Ngày bắt đầu |
| NgayKetThuc | Datetime |  | Ngày kết thúc |
| SoDienCu | Float |  | Số điện cũ |
| SoDienMoi | Float |  | Số điện mới |
| SoNuocCu | Float |  | Số nước cũ |
| SoNuocMoi | Float |  | Số nước mới |
| KhauTru | Float |  | Khấu trừ |
| No | Float |  | Nợ |
| TongCong | Float |  | Tổng cộng |
| NgayTao | datetime |  | Ngày tạo |
| NguoiTao | Nvarchar(100) |  | Người tạo |
| NgaySua | datetime |  | Ngày sửa |
| NguoiSua | Nvarchar(100) |  | Người sửa |
| Trash | int | Default 1 | Thùng rác:  0 là tạm xóa  1 là không xóa |
| Status | int |  | Trạng thái:  0 là chưa thanh toán  1 là đã thanh toán |

#### DonGia\_Config:

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu | Ràng buộc | Ghi chú |
| Keyword | Nvarchar(255) | Not null | Tên của đơn giá |
| value | float | Not null | Giá tiền |

## Cơ sở dữ liệu:

### 3.3.1 Tạo cơ sở dữ liệu



A white background with black text

Description automatically generated

A close up of text

Description automatically generated

A screenshot of a computer code

Description automatically generated

A white screen with blue text

Description automatically generated

A computer screen shot of a code

Description automatically generated

A screenshot of a computer code

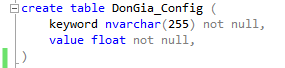
Description automatically generated

A screenshot of a computer code

Description automatically generated

A screen shot of a computer code

Description automatically generated



### 3.3.2. Tìm dữ liệu cho cơ sở dữ liệu

#### 3.3.2.1. Insert đánh giá:

SET IDENTITY\_INSERT [dbo].[DanhGia] ON

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (1, 4, 2, 4, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Nguyễn Văn A')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (2, 5, 3, 5, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Phạm Thị B')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (3, 6, 4, 3, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Nguyễn Thị C')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (4, 9, 5, 2, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Lê Văn D')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (5, 10, 6, 1, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Vũ Thị E')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (6, 12, 7, 4, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Hoàng Văn F')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (7, 15, 8, 5, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Đỗ Thị G')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (8, 17, 9, 4, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Trần Văn H')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (9, 18, 10, 3, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Lý Thị I')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (10, 20, 11, 2, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Nguyễn Văn K')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (11, 21, 12, 1, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Phạm Thị L')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (12, 23, 13, 1, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Nguyễn Thị M')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (13, 24, 14, 2, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Lê Văn N')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (14, 25, 15, 3, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Vũ Thị O')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (15, 27, 16, 4, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Hoàng Văn P')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (16, 28, 17, 5, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Đỗ Thị Q')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (17, 30, 18, 1, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Trần Văn R')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (18, 31, 19, 2, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Lý Thị S')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (19, 33, 20, 3, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Nguyễn Văn T')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (20, 35, 2, 4, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Phạm Thị U')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (21, 36, 2, 5, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Nguyễn Văn A')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (22, 38, 2, 5, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Nguyễn Văn A')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (23, 39, 2, 4, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Nguyễn Văn A')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (24, 41, 2, 3, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Nguyễn Văn A')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (25, 42, 2, 1, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Nguyễn Văn A')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (26, 44, 2, 1, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Nguyễn Văn A')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (27, 45, 2, 4, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Nguyễn Văn A')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (28, 46, 2, 5, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Nguyễn Văn A')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (29, 48, 2, 3, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Nguyễn Văn A')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (30, 49, 2, 2, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Nguyễn Văn A')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (31, 50, 2, 1, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Nguyễn Văn A')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (32, 51, 2, 1, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Nguyễn Văn A')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (33, 52, 2, 4, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Nguyễn Văn A')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (34, 54, 2, 2, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Nguyễn Văn A')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (35, 55, 2, 3, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Nguyễn Văn A')

INSERT [dbo].[DanhGia] ([Id], [IdMaPhongTro], [IdNguoiDung], [DanhGia], [NoiDung], [NgayTao], [NguoiTao]) VALUES (36, 56, 2, 5, N'Nhà thoáng mát', CAST(N'2023-07-31T14:33:16.017' AS DateTime), N'Nguyễn Văn A')

SET IDENTITY\_INSERT [dbo].[DanhGia] OFF

#### 3.3.2.2. Insert đơn giá:

INSERT [dbo].[DonGia\_Config] ([keyword], [value]) VALUES (N'Giá điện', 3000)

INSERT [dbo].[DonGia\_Config] ([keyword], [value]) VALUES (N'Giá nước', 17000)

INSERT [dbo].[DonGia\_Config] ([keyword], [value]) VALUES (N'Giá internet', 150000)

INSERT [dbo].[DonGia\_Config] ([keyword], [value]) VALUES (N'Giá rác', 30000)

#### 3.3.2.3. Insert hóa đơn:

SET IDENTITY\_INSERT [dbo].[HoaDon] ON

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (1, 1, CAST(N'2023-05-01T00:00:00.000' AS DateTime), CAST(N'2023-05-31T00:00:00.000' AS DateTime), 230, 250, 22, 28, 0, 0, 2332000, CAST(N'2023-06-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T14:39:33.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (2, 2, CAST(N'2023-05-01T00:00:00.000' AS DateTime), CAST(N'2023-05-31T00:00:00.000' AS DateTime), 250, 280, 27, 35, 0, 0, 1896000, CAST(N'2023-06-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T14:39:33.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (3, 3, CAST(N'2023-05-01T00:00:00.000' AS DateTime), CAST(N'2023-05-31T00:00:00.000' AS DateTime), 42, 100, 24, 38, 0, 0, 4502000, CAST(N'2023-06-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T14:39:33.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (4, 4, CAST(N'2023-05-01T00:00:00.000' AS DateTime), CAST(N'2023-05-31T00:00:00.000' AS DateTime), 42, 79, 24, 35, 0, 0, 3008000, CAST(N'2023-06-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T14:39:33.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (5, 5, CAST(N'2023-05-01T00:00:00.000' AS DateTime), CAST(N'2023-05-31T00:00:00.000' AS DateTime), 42, 79, 24, 35, 0, 0, 2468000, CAST(N'2023-06-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T14:39:33.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (6, 6, CAST(N'2023-05-01T00:00:00.000' AS DateTime), CAST(N'2023-05-31T00:00:00.000' AS DateTime), 42, 100, 24, 40, 0, 0, 3686000, CAST(N'2023-06-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T14:39:33.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (7, 7, CAST(N'2023-05-01T00:00:00.000' AS DateTime), CAST(N'2023-05-31T00:00:00.000' AS DateTime), 42, 62, 24, 29, 0, 0, 2315000, CAST(N'2023-06-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T14:39:33.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (8, 8, CAST(N'2023-05-01T00:00:00.000' AS DateTime), CAST(N'2023-05-31T00:00:00.000' AS DateTime), 42, 100, 24, 33, 0, 0, 3567000, CAST(N'2023-06-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T14:39:33.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (9, 9, CAST(N'2023-05-01T00:00:00.000' AS DateTime), CAST(N'2023-05-31T00:00:00.000' AS DateTime), 42, 100, 24, 33, 0, 0, 3567000, CAST(N'2023-06-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T14:39:33.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (10, 10, CAST(N'2023-05-01T00:00:00.000' AS DateTime), CAST(N'2023-05-31T00:00:00.000' AS DateTime), 42, 120, 24, 38, 0, 0, 4567000, CAST(N'2023-06-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T14:39:33.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (11, 11, CAST(N'2023-05-01T00:00:00.000' AS DateTime), CAST(N'2023-05-31T00:00:00.000' AS DateTime), 42, 90, 24, 33, 0, 0, 3537000, CAST(N'2023-06-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T14:39:33.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (12, 12, CAST(N'2023-05-01T00:00:00.000' AS DateTime), CAST(N'2023-05-31T00:00:00.000' AS DateTime), 42, 50, 24, 27, 0, 0, 1745000, CAST(N'2023-06-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T14:39:33.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (13, 13, CAST(N'2023-05-01T00:00:00.000' AS DateTime), CAST(N'2023-05-31T00:00:00.000' AS DateTime), 42, 88, 24, 30, 0, 0, 3480000, CAST(N'2023-06-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T14:39:33.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (14, 14, CAST(N'2023-05-01T00:00:00.000' AS DateTime), CAST(N'2023-05-31T00:00:00.000' AS DateTime), 42, 110, 24, 33, 0, 0, 3597000, CAST(N'2023-06-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T14:39:33.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (15, 15, CAST(N'2023-05-01T00:00:00.000' AS DateTime), CAST(N'2023-05-31T00:00:00.000' AS DateTime), 42, 59, 24, 28, 0, 0, 2289000, CAST(N'2023-06-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T14:39:33.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (16, 1, CAST(N'2023-06-01T00:00:00.000' AS DateTime), CAST(N'2023-06-30T00:00:00.000' AS DateTime), 250, 330, 28, 40, 0, 0, 2614000, CAST(N'2023-07-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T14:52:57.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (17, 2, CAST(N'2023-06-01T00:00:00.000' AS DateTime), CAST(N'2023-06-30T00:00:00.000' AS DateTime), 280, 340, 35, 40, 0, 0, 1935000, CAST(N'2023-07-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T14:52:57.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (18, 3, CAST(N'2023-06-01T00:00:00.000' AS DateTime), CAST(N'2023-06-30T00:00:00.000' AS DateTime), 100, 300, 38, 48, 0, 0, 4860000, CAST(N'2023-07-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T14:52:57.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (19, 4, CAST(N'2023-06-01T00:00:00.000' AS DateTime), CAST(N'2023-06-30T00:00:00.000' AS DateTime), 79, 150, 35, 39, 0, 0, 2991000, CAST(N'2023-07-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T14:52:57.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (20, 5, CAST(N'2023-06-01T00:00:00.000' AS DateTime), CAST(N'2023-06-30T00:00:00.000' AS DateTime), 79, 148, 35, 40, 0, 0, 2462000, CAST(N'2023-07-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T14:52:57.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (21, 6, CAST(N'2023-06-01T00:00:00.000' AS DateTime), CAST(N'2023-06-30T00:00:00.000' AS DateTime), 100, 170, 40, 51, 0, 0, 3637000, CAST(N'2023-07-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T14:52:57.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (22, 7, CAST(N'2023-06-01T00:00:00.000' AS DateTime), CAST(N'2023-06-30T00:00:00.000' AS DateTime), 62, 130, 29, 37, 0, 0, 2510000, CAST(N'2023-07-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T15:34:21.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (23, 8, CAST(N'2023-06-01T00:00:00.000' AS DateTime), CAST(N'2023-06-30T00:00:00.000' AS DateTime), 100, 176, 33, 44, 0, 0, 3655000, CAST(N'2023-07-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T15:34:21.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (24, 9, CAST(N'2023-06-01T00:00:00.000' AS DateTime), CAST(N'2023-06-30T00:00:00.000' AS DateTime), 100, 166, 33, 41, 0, 0, 3574000, CAST(N'2023-07-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T15:34:21.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (25, 10, CAST(N'2023-06-01T00:00:00.000' AS DateTime), CAST(N'2023-06-30T00:00:00.000' AS DateTime), 120, 190, 38, 47, 0, 0, 4458000, CAST(N'2023-07-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T15:34:21.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (26, 11, CAST(N'2023-06-01T00:00:00.000' AS DateTime), CAST(N'2023-06-30T00:00:00.000' AS DateTime), 90, 170, 33, 38, 0, 0, 3565000, CAST(N'2023-07-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T15:34:21.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (27, 12, CAST(N'2023-06-01T00:00:00.000' AS DateTime), CAST(N'2023-06-30T00:00:00.000' AS DateTime), 50, 130, 27, 31, 0, 0, 1978000, CAST(N'2023-07-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T15:34:21.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (28, 13, CAST(N'2023-06-01T00:00:00.000' AS DateTime), CAST(N'2023-06-30T00:00:00.000' AS DateTime), 88, 154, 30, 36, 0, 0, 3540000, CAST(N'2023-07-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T15:34:21.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (29, 14, CAST(N'2023-06-01T00:00:00.000' AS DateTime), CAST(N'2023-06-30T00:00:00.000' AS DateTime), 110, 200, 33, 41, 0, 0, 3646000, CAST(N'2023-07-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T15:34:21.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (30, 15, CAST(N'2023-06-01T00:00:00.000' AS DateTime), CAST(N'2023-06-30T00:00:00.000' AS DateTime), 59, 124, 28, 35, 0, 0, 2484000, CAST(N'2023-07-01T00:00:00.000' AS DateTime), N'Phương Vũ', CAST(N'2023-07-31T15:34:21.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (31, 1, CAST(N'2023-07-01T00:00:00.000' AS DateTime), CAST(N'2023-07-31T00:00:00.000' AS DateTime), 330, 452, 40, 47, 0, 0, 2655000, CAST(N'2023-08-01T00:00:00.000' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (32, 2, CAST(N'2023-07-01T00:00:00.000' AS DateTime), CAST(N'2023-07-31T00:00:00.000' AS DateTime), 340, 407, 40, 45, 0, 0, 1956000, CAST(N'2023-08-01T00:00:00.000' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (33, 3, CAST(N'2023-07-01T00:00:00.000' AS DateTime), CAST(N'2023-07-31T00:00:00.000' AS DateTime), 300, 474, 48, 50, 0, 0, 4646000, CAST(N'2023-08-01T00:00:00.000' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (34, 4, CAST(N'2023-07-01T00:00:00.000' AS DateTime), CAST(N'2023-07-31T00:00:00.000' AS DateTime), 150, 230, 39, 48, 0, 0, 3103000, CAST(N'2023-08-01T00:00:00.000' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (35, 5, CAST(N'2023-07-01T00:00:00.000' AS DateTime), CAST(N'2023-07-31T00:00:00.000' AS DateTime), 148, 207, 40, 49, 0, 0, 2500000, CAST(N'2023-08-01T00:00:00.000' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (36, 6, CAST(N'2023-07-01T00:00:00.000' AS DateTime), CAST(N'2023-07-31T00:00:00.000' AS DateTime), 170, 284, 51, 63, 0, 0, 3786000, CAST(N'2023-08-01T00:00:00.000' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (37, 7, CAST(N'2023-07-01T00:00:00.000' AS DateTime), CAST(N'2023-07-31T00:00:00.000' AS DateTime), 130, 246, 37, 46, 0, 0, 2671000, CAST(N'2023-08-01T00:00:00.000' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (38, 8, CAST(N'2023-07-01T00:00:00.000' AS DateTime), CAST(N'2023-07-31T00:00:00.000' AS DateTime), 176, 263, 44, 54, 0, 0, 3671000, CAST(N'2023-08-01T00:00:00.000' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (39, 9, CAST(N'2023-07-01T00:00:00.000' AS DateTime), CAST(N'2023-07-31T00:00:00.000' AS DateTime), 166, 270, 41, 51, 0, 0, 3722000, CAST(N'2023-08-01T00:00:00.000' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (40, 10, CAST(N'2023-07-01T00:00:00.000' AS DateTime), CAST(N'2023-07-31T00:00:00.000' AS DateTime), 190, 294, 47, 57, 0, 0, 4577000, CAST(N'2023-08-01T00:00:00.000' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (41, 11, CAST(N'2023-07-01T00:00:00.000' AS DateTime), CAST(N'2023-07-31T00:00:00.000' AS DateTime), 170, 278, 38, 46, 0, 0, 3700000, CAST(N'2023-08-01T00:00:00.000' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (42, 12, CAST(N'2023-07-01T00:00:00.000' AS DateTime), CAST(N'2023-07-31T00:00:00.000' AS DateTime), 130, 180, 31, 37, 0, 0, 1922000, CAST(N'2023-08-01T00:00:00.000' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (43, 13, CAST(N'2023-07-01T00:00:00.000' AS DateTime), CAST(N'2023-07-31T00:00:00.000' AS DateTime), 154, 243, 36, 45, 0, 0, 3660000, CAST(N'2023-08-01T00:00:00.000' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (44, 14, CAST(N'2023-07-01T00:00:00.000' AS DateTime), CAST(N'2023-07-31T00:00:00.000' AS DateTime), 200, 281, 41, 52, 0, 0, 3670000, CAST(N'2023-08-01T00:00:00.000' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (45, 15, CAST(N'2023-07-01T00:00:00.000' AS DateTime), CAST(N'2023-07-31T00:00:00.000' AS DateTime), 124, 241, 35, 41, 0, 0, 2623000, CAST(N'2023-08-01T00:00:00.000' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[HoaDon] ([Id], [IdHopDong], [NgayBatDau], [NgayKetThuc], [SoDienCu], [SoDienMoi], [SoNuocCu], [SoNuocMoi], [KhauTru], [TienNo], [TongCong], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (46, 17, CAST(N'2023-08-01T00:00:00.000' AS DateTime), CAST(N'2023-08-31T00:00:00.000' AS DateTime), 0, 50, 0, 10, 0, 0, 2000000, CAST(N'2023-07-01T00:00:00.000' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

SET IDENTITY\_INSERT [dbo].[HoaDon] OFF

#### 3.3.2.4. Insert hợp đồng:

SET IDENTITY\_INSERT [dbo].[HopDong] ON

INSERT [dbo].[HopDong] ([Id], [IdMaPhongTro], [IdNguoiDung], [NgayBatDauThue], [NgayKetThucThue], [TienDatCoc], [GiaPhong], [GiaDien], [GiaNuoc], [GiaInternet], [GiaRac], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (1, 4, 6, CAST(N'2022-01-01T00:00:00.000' AS DateTime), CAST(N'2024-01-01T00:00:00.000' AS DateTime), 1000000, 2000000, 3000, 17000, 150000, 20000, CAST(N'2023-07-30T14:45:40.000' AS DateTime), N'Phương Vũ', CAST(N'2022-01-01T00:00:00.000' AS DateTime), N'thanh', 1, 1)

INSERT [dbo].[HopDong] ([Id], [IdMaPhongTro], [IdNguoiDung], [NgayBatDauThue], [NgayKetThucThue], [TienDatCoc], [GiaPhong], [GiaDien], [GiaNuoc], [GiaInternet], [GiaRac], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (2, 8, 7, CAST(N'2023-01-01T00:00:00.000' AS DateTime), CAST(N'2024-07-01T00:00:00.000' AS DateTime), 700000, 1500000, 3000, 17000, 150000, 20000, CAST(N'2023-07-30T14:48:08.000' AS DateTime), N'Phương Vũ', CAST(N'2023-01-01T00:00:00.000' AS DateTime), N'thanh', 1, 1)

INSERT [dbo].[HopDong] ([Id], [IdMaPhongTro], [IdNguoiDung], [NgayBatDauThue], [NgayKetThucThue], [TienDatCoc], [GiaPhong], [GiaDien], [GiaNuoc], [GiaInternet], [GiaRac], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (3, 12, 8, CAST(N'2023-05-01T00:00:00.000' AS DateTime), CAST(N'2025-05-01T00:00:00.000' AS DateTime), 1900000, 3800000, 3000, 17000, 250000, 40000, CAST(N'2023-07-30T14:50:31.000' AS DateTime), N'Phương Vũ', CAST(N'2023-05-01T00:00:00.000' AS DateTime), N'thanh', 1, 1)

INSERT [dbo].[HopDong] ([Id], [IdMaPhongTro], [IdNguoiDung], [NgayBatDauThue], [NgayKetThucThue], [TienDatCoc], [GiaPhong], [GiaDien], [GiaNuoc], [GiaInternet], [GiaRac], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (4, 17, 9, CAST(N'2023-01-01T00:00:00.000' AS DateTime), CAST(N'2024-01-01T00:00:00.000' AS DateTime), 1500000, 2500000, 3000, 17000, 180000, 30000, CAST(N'2023-07-30T15:00:51.000' AS DateTime), N'Phương Vũ', CAST(N'2023-01-01T00:00:00.000' AS DateTime), N'thanh', 1, 1)

INSERT [dbo].[HopDong] ([Id], [IdMaPhongTro], [IdNguoiDung], [NgayBatDauThue], [NgayKetThucThue], [TienDatCoc], [GiaPhong], [GiaDien], [GiaNuoc], [GiaInternet], [GiaRac], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (5, 2, 10, CAST(N'2023-01-01T00:00:00.000' AS DateTime), CAST(N'2024-01-01T00:00:00.000' AS DateTime), 1000000, 2000000, 3000, 17000, 150000, 20000, CAST(N'2023-07-30T15:03:26.000' AS DateTime), N'Phương Vũ', CAST(N'2023-01-01T00:00:00.000' AS DateTime), N'thanh', 1, 1)

INSERT [dbo].[HopDong] ([Id], [IdMaPhongTro], [IdNguoiDung], [NgayBatDauThue], [NgayKetThucThue], [TienDatCoc], [GiaPhong], [GiaDien], [GiaNuoc], [GiaInternet], [GiaRac], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (6, 27, 11, CAST(N'2023-01-01T00:00:00.000' AS DateTime), CAST(N'2024-01-01T00:00:00.000' AS DateTime), 1500000, 3000000, 3000, 17000, 200000, 40000, CAST(N'2023-07-30T15:05:54.000' AS DateTime), N'Phương Vũ', CAST(N'2023-01-01T00:00:00.000' AS DateTime), N'thanh', 1, 1)

INSERT [dbo].[HopDong] ([Id], [IdMaPhongTro], [IdNguoiDung], [NgayBatDauThue], [NgayKetThucThue], [TienDatCoc], [GiaPhong], [GiaDien], [GiaNuoc], [GiaInternet], [GiaRac], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (7, 22, 12, CAST(N'2023-01-01T00:00:00.000' AS DateTime), CAST(N'2024-01-01T00:00:00.000' AS DateTime), 1000000, 2000000, 3000, 17000, 150000, 20000, CAST(N'2023-07-31T14:15:41.000' AS DateTime), N'Phương Vũ', CAST(N'2023-01-01T00:00:00.000' AS DateTime), N'thanh', 1, 1)

INSERT [dbo].[HopDong] ([Id], [IdMaPhongTro], [IdNguoiDung], [NgayBatDauThue], [NgayKetThucThue], [TienDatCoc], [GiaPhong], [GiaDien], [GiaNuoc], [GiaInternet], [GiaRac], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (8, 30, 13, CAST(N'2023-01-01T00:00:00.000' AS DateTime), CAST(N'2024-01-01T00:00:00.000' AS DateTime), 1500000, 3000000, 3000, 17000, 200000, 40000, CAST(N'2023-07-31T14:18:25.000' AS DateTime), N'Phương Vũ', CAST(N'2023-01-01T00:00:00.000' AS DateTime), N'thanh', 1, 1)

INSERT [dbo].[HopDong] ([Id], [IdMaPhongTro], [IdNguoiDung], [NgayBatDauThue], [NgayKetThucThue], [TienDatCoc], [GiaPhong], [GiaDien], [GiaNuoc], [GiaInternet], [GiaRac], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (9, 37, 14, CAST(N'2023-01-01T00:00:00.000' AS DateTime), CAST(N'2024-01-01T00:00:00.000' AS DateTime), 1500000, 3000000, 3000, 17000, 200000, 40000, CAST(N'2023-07-31T14:21:11.000' AS DateTime), N'Phương Vũ', CAST(N'2023-01-01T00:00:00.000' AS DateTime), N'thanh', 1, 1)

INSERT [dbo].[HopDong] ([Id], [IdMaPhongTro], [IdNguoiDung], [NgayBatDauThue], [NgayKetThucThue], [TienDatCoc], [GiaPhong], [GiaDien], [GiaNuoc], [GiaInternet], [GiaRac], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (10, 43, 15, CAST(N'2023-01-01T00:00:00.000' AS DateTime), CAST(N'2024-01-01T00:00:00.000' AS DateTime), 1900000, 3800000, 3000, 17000, 250000, 45000, CAST(N'2023-07-31T14:21:11.000' AS DateTime), N'Phương Vũ', CAST(N'2023-01-01T00:00:00.000' AS DateTime), N'thanh', 1, 1)

INSERT [dbo].[HopDong] ([Id], [IdMaPhongTro], [IdNguoiDung], [NgayBatDauThue], [NgayKetThucThue], [TienDatCoc], [GiaPhong], [GiaDien], [GiaNuoc], [GiaInternet], [GiaRac], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (11, 48, 16, CAST(N'2023-01-01T00:00:00.000' AS DateTime), CAST(N'2024-01-01T00:00:00.000' AS DateTime), 1500000, 3000000, 3000, 17000, 200000, 40000, CAST(N'2023-07-31T14:21:11.000' AS DateTime), N'Phương Vũ', CAST(N'2023-01-01T00:00:00.000' AS DateTime), N'thanh', 1, 1)

INSERT [dbo].[HopDong] ([Id], [IdMaPhongTro], [IdNguoiDung], [NgayBatDauThue], [NgayKetThucThue], [TienDatCoc], [GiaPhong], [GiaDien], [GiaNuoc], [GiaInternet], [GiaRac], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (12, 10, 17, CAST(N'2023-01-01T00:00:00.000' AS DateTime), CAST(N'2024-01-01T00:00:00.000' AS DateTime), 1000000, 1500000, 3000, 17000, 150000, 20000, CAST(N'2023-07-31T14:21:11.000' AS DateTime), N'Phương Vũ', CAST(N'2023-01-01T00:00:00.000' AS DateTime), N'Phương Vũ', 1, 1)

INSERT [dbo].[HopDong] ([Id], [IdMaPhongTro], [IdNguoiDung], [NgayBatDauThue], [NgayKetThucThue], [TienDatCoc], [GiaPhong], [GiaDien], [GiaNuoc], [GiaInternet], [GiaRac], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (13, 53, 18, CAST(N'2023-01-01T00:00:00.000' AS DateTime), CAST(N'2024-01-01T00:00:00.000' AS DateTime), 1500000, 3000000, 3000, 17000, 200000, 40000, CAST(N'2023-07-31T14:27:10.000' AS DateTime), N'Phương Vũ', CAST(N'2023-01-01T00:00:00.000' AS DateTime), N'thanh', 1, 1)

INSERT [dbo].[HopDong] ([Id], [IdMaPhongTro], [IdNguoiDung], [NgayBatDauThue], [NgayKetThucThue], [TienDatCoc], [GiaPhong], [GiaDien], [GiaNuoc], [GiaInternet], [GiaRac], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (14, 56, 19, CAST(N'2023-01-01T00:00:00.000' AS DateTime), CAST(N'2024-01-01T00:00:00.000' AS DateTime), 1500000, 3000000, 3000, 17000, 200000, 40000, CAST(N'2023-07-31T14:27:10.000' AS DateTime), N'Phương Vũ', CAST(N'2023-01-01T00:00:00.000' AS DateTime), N'thanh', 1, 1)

INSERT [dbo].[HopDong] ([Id], [IdMaPhongTro], [IdNguoiDung], [NgayBatDauThue], [NgayKetThucThue], [TienDatCoc], [GiaPhong], [GiaDien], [GiaNuoc], [GiaInternet], [GiaRac], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (15, 6, 20, CAST(N'2023-01-01T00:00:00.000' AS DateTime), CAST(N'2024-01-01T00:00:00.000' AS DateTime), 1000000, 2000000, 3000, 17000, 150000, 20000, CAST(N'2023-07-31T14:27:10.000' AS DateTime), N'Phương Vũ', CAST(N'2023-01-01T00:00:00.000' AS DateTime), N'thanh', 1, 1)

INSERT [dbo].[HopDong] ([Id], [IdMaPhongTro], [IdNguoiDung], [NgayBatDauThue], [NgayKetThucThue], [TienDatCoc], [GiaPhong], [GiaDien], [GiaNuoc], [GiaInternet], [GiaRac], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (16, 5, 21, CAST(N'2023-08-01T00:00:00.000' AS DateTime), CAST(N'2023-12-31T00:00:00.000' AS DateTime), 2000000, 2000000, 3000, 17000, 150000, 30000, CAST(N'2023-08-03T13:09:59.000' AS DateTime), N'Phương Vũ', CAST(N'2023-08-01T00:00:00.000' AS DateTime), N'thanh', 1, 1)

INSERT [dbo].[HopDong] ([Id], [IdMaPhongTro], [IdNguoiDung], [NgayBatDauThue], [NgayKetThucThue], [TienDatCoc], [GiaPhong], [GiaDien], [GiaNuoc], [GiaInternet], [GiaRac], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (17, 9, 22, CAST(N'2023-08-01T00:00:00.000' AS DateTime), CAST(N'2023-12-31T00:00:00.000' AS DateTime), 1500000, 1500000, 3000, 17000, 150000, 30000, CAST(N'2023-08-03T14:08:53.000' AS DateTime), N'Phương Vũ', CAST(N'2023-08-01T00:00:00.000' AS DateTime), N'thanh', 1, 1)

SET IDENTITY\_INSERT [dbo].[HopDong] OFF

#### 3.3.2.5. Insert khu vực:

SET IDENTITY\_INSERT [dbo].[KhuVuc] ON

INSERT [dbo].[KhuVuc] ([Id], [TenKhuVuc], [Tang], [DayNhaTro]) VALUES (1, N'Khu A', 1, 2)

INSERT [dbo].[KhuVuc] ([Id], [TenKhuVuc], [Tang], [DayNhaTro]) VALUES (2, N'Khu B', 2, 1)

INSERT [dbo].[KhuVuc] ([Id], [TenKhuVuc], [Tang], [DayNhaTro]) VALUES (3, N'Khu C', 3, 2)

INSERT [dbo].[KhuVuc] ([Id], [TenKhuVuc], [Tang], [DayNhaTro]) VALUES (4, N'Khu D', 1, 3)

INSERT [dbo].[KhuVuc] ([Id], [TenKhuVuc], [Tang], [DayNhaTro]) VALUES (5, N'Khu E', 2, 3)

SET IDENTITY\_INSERT [dbo].[KhuVuc] OFF

#### 3.3.2.6. Insert loại phòng:

SET IDENTITY\_INSERT [dbo].[LoaiPhong] ON

INSERT [dbo].[LoaiPhong] ([Id], [TenLoaiPhong]) VALUES (1, N'Phòng đơn')

INSERT [dbo].[LoaiPhong] ([Id], [TenLoaiPhong]) VALUES (2, N'Phòng đôi')

INSERT [dbo].[LoaiPhong] ([Id], [TenLoaiPhong]) VALUES (3, N'Phòng gia đình')

INSERT [dbo].[LoaiPhong] ([Id], [TenLoaiPhong]) VALUES (4, N'Phòng studio')

INSERT [dbo].[LoaiPhong] ([Id], [TenLoaiPhong]) VALUES (5, N'Phòng suite')

SET IDENTITY\_INSERT [dbo].[LoaiPhong] OFF

#### 3.3.2.7. Insert người dùng:

SET IDENTITY\_INSERT [dbo].[NguoiDung] ON

INSERT [dbo].[NguoiDung] ([Id], [HoTen], [DienThoai], [Email], [DiaChi], [MatKhau], [VaiTro], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (1, N'Phương Vũ', N'0386139125', N'phuongvu.xu@gmail.com', N'Quận 12', N'$2a$10$TUrGXJPs/RhQXqCrZC/zruo5OM96OpNot.3fvpidnRf8qAsYCA26y', N'Admin', CAST(N'2023-07-29T22:59:28.807' AS DateTime), N'Phương Vũ', NULL, NULL, 1, NULL)

INSERT [dbo].[NguoiDung] ([Id], [HoTen], [DienThoai], [Email], [DiaChi], [MatKhau], [VaiTro], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (2, N'Khắc Sáng', N'0963700685', N'sangnkps26482@fpt.edu.vn', N'Bình Dương', N'$2a$10$K6jofPHKfIFuwpNMmsfUeeT/FUlL0n06aoqUlbJ0qXTcT1o76YC9C', N'Admin', CAST(N'2023-07-30T10:34:51.980' AS DateTime), N'Khắc Sáng', NULL, NULL, 1, NULL)

INSERT [dbo].[NguoiDung] ([Id], [HoTen], [DienThoai], [Email], [DiaChi], [MatKhau], [VaiTro], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (3, N'Hoàng Duy', N'0933277546', N'duychps26595@fpt.edu.vn', N'Biên Hòa', N'$2a$10$sAON4D.oYYil0sNKnqUlKOTCsUK/Y2rplEXylM2AjFieWNKNgAGVu', N'Admin', CAST(N'2023-07-30T10:36:19.597' AS DateTime), N'Hoàng Duy', NULL, NULL, 1, NULL)

INSERT [dbo].[NguoiDung] ([Id], [HoTen], [DienThoai], [Email], [DiaChi], [MatKhau], [VaiTro], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (4, N'Trường Sơn', N'0364490645', N'sonntps26480@fpt.edu.vn', N'Quận 12', N'$2a$10$p7u//x5j/j6fyZIg4JKm0.xsue.b84gdJ2Eog3ZUjD3XIWo8Eitvm', N'Admin', CAST(N'2023-07-30T10:37:02.443' AS DateTime), N'Trường Sơn', NULL, NULL, 1, NULL)

INSERT [dbo].[NguoiDung] ([Id], [HoTen], [DienThoai], [Email], [DiaChi], [MatKhau], [VaiTro], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (5, N'Minh Thành', N'0965843004', N'jaycedtp@gmail.com', N'Bình Dương', N'$2a$10$D1qu5Apck1GkQYZNE1sxXefVSZEKUpigtJnGH8fqU7Ylh1yMolFpq', N'Admin', CAST(N'2023-07-30T10:41:58.187' AS DateTime), N'Minh Thành', NULL, NULL, 1, NULL)

INSERT [dbo].[NguoiDung] ([Id], [HoTen], [DienThoai], [Email], [DiaChi], [MatKhau], [VaiTro], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (6, N'Nguyễn Văn An', N'0123456789', N'annguyen@gmail.com', N'An Giang', N'$2a$10$1L4T0iuz7pMne36gN411degtlwyiW87aYh9TepdWcvaVWwBaasPjm', N'Tenants', CAST(N'2023-07-30T14:04:41.880' AS DateTime), N'Phương Vũ', CAST(N'2023-07-30T14:47:11.917' AS DateTime), N'Phương Vũ', 1, NULL)

INSERT [dbo].[NguoiDung] ([Id], [HoTen], [DienThoai], [Email], [DiaChi], [MatKhau], [VaiTro], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (7, N'Trần Thị Bình', N'0987654322', N'binhtran@gmail.com', N'Hà Nội', N'$2a$10$.Im5Y73EzIRG0v/vCyDRR.B047nb/fzAvZeg5xIDwkBrBofGM6jV2', N'Tenants', CAST(N'2023-07-30T14:08:09.507' AS DateTime), N'Phương Vũ', CAST(N'2023-07-30T14:47:31.493' AS DateTime), N'Phương Vũ', 1, NULL)

INSERT [dbo].[NguoiDung] ([Id], [HoTen], [DienThoai], [Email], [DiaChi], [MatKhau], [VaiTro], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (8, N'Lê Văn Cường', N'0369852147', N'cuongle@gmail.com', N'Đà Nẵng', N'$2a$10$ek4E72TVKzlfVaIKfDCPZOOdBD3W.0.v3rnrc.Hsw8D8pY/oV97UK', N'Tenants', CAST(N'2023-07-30T14:09:28.790' AS DateTime), N'Phương Vũ', CAST(N'2023-07-30T14:49:58.620' AS DateTime), N'Phương Vũ', 1, NULL)

INSERT [dbo].[NguoiDung] ([Id], [HoTen], [DienThoai], [Email], [DiaChi], [MatKhau], [VaiTro], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (9, N'Phạm Thị Diễm', N'0912345678', N'diempham@gmail.com', N'Hội An', N'$2a$10$/3Fgn/SDkNJyk/FsrXp2Qe36HvtOcQY6ZQmF4Aw8hoopsDDDuN75K', N'Tenants', CAST(N'2023-07-30T14:10:29.407' AS DateTime), N'Phương Vũ', CAST(N'2023-07-30T15:00:19.620' AS DateTime), N'Phương Vũ', 1, NULL)

INSERT [dbo].[NguoiDung] ([Id], [HoTen], [DienThoai], [Email], [DiaChi], [MatKhau], [VaiTro], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (10, N'Hoàng Văn Dũng', N'0987123456', N'dunghoang@gmail.com', N'Bắc Ninh', N'$2a$10$P4jP.6vThRc57CD/lUaD0eWZ1waql7kK4DGWtT/O.f8PxTnSlR8Hq', N'Tenants', CAST(N'2023-07-30T14:13:32.017' AS DateTime), N'Phương Vũ', CAST(N'2023-07-30T15:02:51.350' AS DateTime), N'Phương Vũ', 1, NULL)

INSERT [dbo].[NguoiDung] ([Id], [HoTen], [DienThoai], [Email], [DiaChi], [MatKhau], [VaiTro], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (11, N'Huỳnh Thái Hà', N'0909090901', N'hahuynh@gmail.com', N'Thành phố Hồ Chí Minh', N'$2a$10$D.wKbUhnWrINUV19ta44WuiCl2PKSILU2j7.BI1OtvAcN/Uz1J86.', N'Tenants', CAST(N'2023-07-30T14:14:11.730' AS DateTime), N'Phương Vũ', CAST(N'2023-07-30T15:05:05.407' AS DateTime), N'Phương Vũ', 1, NULL)

INSERT [dbo].[NguoiDung] ([Id], [HoTen], [DienThoai], [Email], [DiaChi], [MatKhau], [VaiTro], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (12, N'Phan Văn Giang', N'0987654321', N'giangphan@gmail.com', N'Ninh Bình', N'$2a$10$Aqo6nADo9Lc99wZaO0hTseihHLutFJ4TWtc5LQk/.naUMrJUa3deq', N'Tenants', CAST(N'2023-07-30T14:14:55.897' AS DateTime), N'Phương Vũ', CAST(N'2023-07-30T15:16:10.790' AS DateTime), N'Phương Vũ', 1, NULL)

INSERT [dbo].[NguoiDung] ([Id], [HoTen], [DienThoai], [Email], [DiaChi], [MatKhau], [VaiTro], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (13, N'Vũ Thu Hương', N'0369852148', N'huongvu@gmai.com', N'Tây Ninh', N'$2a$10$pD9vNJ6oae/3aBtwAaZIsOjvoMJNV/WP6kcGIdST6XIjYIHpDbQXe', N'Tenants', CAST(N'2023-07-30T14:15:59.063' AS DateTime), N'Phương Vũ', CAST(N'2023-07-30T15:16:17.680' AS DateTime), N'Phương Vũ', 1, NULL)

INSERT [dbo].[NguoiDung] ([Id], [HoTen], [DienThoai], [Email], [DiaChi], [MatKhau], [VaiTro], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (14, N'Đỗ Văn Khánh', N'0912345671', N'khanhdo@gmail.com', N'Đồng Nai', N'$2a$10$W8U1/ZvsVOPYN9PC9JSNweetnJb2Mvi/al486LuPgEDEBvba.qtRW', N'Tenants', CAST(N'2023-07-30T14:16:58.053' AS DateTime), N'Phương Vũ', CAST(N'2023-07-30T15:16:27.403' AS DateTime), N'Phương Vũ', 1, NULL)

INSERT [dbo].[NguoiDung] ([Id], [HoTen], [DienThoai], [Email], [DiaChi], [MatKhau], [VaiTro], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (15, N'Ngô Văn Minh', N'0987123457', N'minhngo@gmail.com', N'Thái Bình', N'$2a$10$utpIpzFuh.nHPcBLfGsgUuRKE0qjO9NEh1sZjHpjhkfwq1YKNAwP2', N'Tenants', CAST(N'2023-07-30T14:17:30.313' AS DateTime), N'Phương Vũ', CAST(N'2023-07-30T15:16:34.620' AS DateTime), N'Phương Vũ', 1, NULL)

INSERT [dbo].[NguoiDung] ([Id], [HoTen], [DienThoai], [Email], [DiaChi], [MatKhau], [VaiTro], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (16, N'Dương Thanh Ngọc', N'0909090902', N'ngocduong@gmail.com', N'Thành phố Hồ Chí Minh', N'$2a$10$L5a3FcFU5e8ylJnHWvs6Vez0IDBRxRXd/7zuBi5ygySCFeN4KpKrm', N'Tenants', CAST(N'2023-07-30T14:18:57.083' AS DateTime), N'Phương Vũ', CAST(N'2023-07-30T15:16:41.583' AS DateTime), N'Phương Vũ', 1, NULL)

INSERT [dbo].[NguoiDung] ([Id], [HoTen], [DienThoai], [Email], [DiaChi], [MatKhau], [VaiTro], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (17, N'Lý Uyên Thanh', N'0369852149', N'thanhly@gmail.com', N'Vũng Tàu', N'$2a$10$4/p1KcKuY2Xa1bI/9XRDRe4Dp875PCfA46ZC/8Jy3J2XxhayqJmvG', N'Tenants', CAST(N'2023-07-30T14:19:49.743' AS DateTime), N'Phương Vũ', CAST(N'2023-07-30T15:19:42.427' AS DateTime), N'Phương Vũ', 1, NULL)

INSERT [dbo].[NguoiDung] ([Id], [HoTen], [DienThoai], [Email], [DiaChi], [MatKhau], [VaiTro], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (18, N'Phạm Phương Yến', N'0912345674', N'yenpham@gmail.com', N'Bảo Lộc', N'$2a$10$Jxp94xSkPFFwhaMP.nD81uGfysARW58Hlr./MZFMBxsCyHjeCPkym', N'Tenants', CAST(N'2023-07-30T14:20:27.130' AS DateTime), N'Phương Vũ', CAST(N'2023-07-30T15:19:48.110' AS DateTime), N'Phương Vũ', 1, NULL)

INSERT [dbo].[NguoiDung] ([Id], [HoTen], [DienThoai], [Email], [DiaChi], [MatKhau], [VaiTro], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (19, N'Đặng Văn Đan', N'0987123458', N'dandang@gmail.com', N'Hà Nội', N'$2a$10$Onunv5nU.0IYXsYs6muFauJIQA/NseHuE0P3zL7./l978Azi7SI.y', N'Tenants', CAST(N'2023-07-30T14:21:23.710' AS DateTime), N'Phương Vũ', CAST(N'2023-07-30T15:19:54.197' AS DateTime), N'Phương Vũ', 1, NULL)

INSERT [dbo].[NguoiDung] ([Id], [HoTen], [DienThoai], [Email], [DiaChi], [MatKhau], [VaiTro], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (20, N'Đỗ Thị Xuyến', N'0909090903', N'xuyendo@gmail.com', N'Đà Nẵng', N'$2a$10$4gRF0ntCInHupGSpkTv/5OzKKs7U.K.7Lv7R6WmXSKEFOF3BlUd5K', N'Tenants', CAST(N'2023-07-30T14:22:24.987' AS DateTime), N'Phương Vũ', CAST(N'2023-07-30T15:20:04.893' AS DateTime), N'Phương Vũ', 1, NULL)

INSERT [dbo].[NguoiDung] ([Id], [HoTen], [DienThoai], [Email], [DiaChi], [MatKhau], [VaiTro], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (21, N'Huỳnh Văn Ái', N'0987654323', N'aihuynh@gmail.com', N'Hà Nội', N'$2a$10$kwBcmWPLeb3kz.GS4ejwaukJ7C.fRbPULIwFmg6J/nLUgSJbTllei', N'Tenants', CAST(N'2023-07-30T14:22:57.940' AS DateTime), N'Phương Vũ', NULL, NULL, 1, NULL)

INSERT [dbo].[NguoiDung] ([Id], [HoTen], [DienThoai], [Email], [DiaChi], [MatKhau], [VaiTro], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (22, N'Phan Tú Tâm', N'0369852141', N'tamphan@gmail.com', N'Kiên Giang', N'$2a$10$vaCpK9xtTumMUzkFO/MtgeV//MuZJZxA4QUtOJcvGUVWwS69cpe7O', N'Tenants', CAST(N'2023-07-30T14:23:49.010' AS DateTime), N'Phương Vũ', NULL, NULL, 1, NULL)

INSERT [dbo].[NguoiDung] ([Id], [HoTen], [DienThoai], [Email], [DiaChi], [MatKhau], [VaiTro], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (23, N'Trần Thị Tuyết', N'0912345677', N'tuyettran@gmail.com', N'Long An', N'$2a$10$Xt1bYA5bjHKjaSay6QoVu.IDRYkbhLdSVv9.m/PBrbz38UHcM4VW6', N'User', CAST(N'2023-07-30T14:24:32.713' AS DateTime), N'Phương Vũ', NULL, NULL, 1, NULL)

INSERT [dbo].[NguoiDung] ([Id], [HoTen], [DienThoai], [Email], [DiaChi], [MatKhau], [VaiTro], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (24, N'Lương Văn Ôn', N'0987123459', N'onluong@gmail.com', N'Hà Nội', N'$2a$10$al8Cnsy7ENuA6sES9Ckd2OuvmmnfimjbFmQES5G9gDWXXsiUDSReG', N'User', CAST(N'2023-07-30T14:25:20.137' AS DateTime), N'Phương Vũ', NULL, NULL, 1, NULL)

INSERT [dbo].[NguoiDung] ([Id], [HoTen], [DienThoai], [Email], [DiaChi], [MatKhau], [VaiTro], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (25, N'Vũ Văn Uy', N'0909090909', N'uyvu@gmail.com', N'Thành phố Hồ Chí Minh', N'$2a$10$e0ecVlxe6FEZsw3ZX7UGOublKnl2TLkeO46VP1ad9pBH/qvrpbPr.', N'User', CAST(N'2023-07-30T14:26:00.420' AS DateTime), N'Phương Vũ', NULL, NULL, 1, NULL)

SET IDENTITY\_INSERT [dbo].[NguoiDung] OFF

#### 3.3.2.8. Insert phòng trọ:

SET IDENTITY\_INSERT [dbo].[PhongTro] ON

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (2, 1, 10, 1, N'Phòng đơn 1', 20, 2000000, N'Bình Thạnh, HCM', N'Phòng gác lửng, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 1)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (3, 1, NULL, 1, N'Phòng đơn 2', 20, 2000000, N'Bình Thạnh, HCM', N'Phòng gác lửng, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (4, 1, 6, 1, N'Phòng đơn 3', 20, 2000000, N'Bình Thạnh, HCM', N'Phòng gác lửng, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 1)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (5, 1, 21, 1, N'Phòng đơn 4', 20, 2000000, N'Bình Thạnh, HCM', N'Phòng gác lửng, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 1)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (6, 1, 20, 1, N'Phòng đơn 5', 20, 2000000, N'Bình Thạnh, HCM', N'Phòng gác lửng, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 1)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (7, 1, 23, 1, N'Phòng đơn 6', 20, 2000000, N'Bình Thạnh, HCM', N'Phòng gác lửng, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (8, 1, 7, 1, N'Phòng nhỏ 1', 15, 1500000, N'Bình Thạnh, HCM', N'Phòng không có gác, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 1)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (9, 1, 22, 1, N'Phòng nhỏ 2', 15, 1500000, N'Bình Thạnh, HCM', N'Phòng không có gác, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 1)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (10, 1, 17, 1, N'Phòng nhỏ 3', 15, 1500000, N'Bình Thạnh, HCM', N'Phòng không có gác, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 1)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (11, 1, NULL, 1, N'Phòng nhỏ 4', 15, 1500000, N'Bình Thạnh, HCM', N'Phòng không có gác, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (12, 3, 8, 2, N'Phòng gia đình 1', 40, 3800000, N'Bình Thạnh, HCM', N'Phòng có 1 phòng lớn, 2 phòng nhỏ, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 1)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (13, 3, NULL, 2, N'Phòng gia đình 2', 40, 3800000, N'Bình Thạnh, HCM', N'Phòng có 1 phòng lớn, 2 phòng nhỏ, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (14, 3, NULL, 2, N'Phòng gia đình 3', 40, 3800000, N'Bình Thạnh, HCM', N'Phòng có 1 phòng lớn, 2 phòng nhỏ, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (15, 3, NULL, 2, N'Phòng gia đình 4', 40, 3800000, N'Bình Thạnh, HCM', N'Phòng có 1 phòng lớn, 2 phòng nhỏ, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (16, 3, NULL, 2, N'Phòng gia đình 5', 40, 3800000, N'Bình Thạnh, HCM', N'Phòng có 1 phòng lớn, 2 phòng nhỏ, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (17, 4, 9, 3, N'Phòng studio 1', 25, 2500000, N'Bình Thạnh, HCM', N'Phòng không có vách ngăn, nhà vệ sinh riêng, có bếp, có ban công', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 1)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (18, 4, NULL, 3, N'Phòng studio 2', 25, 2500000, N'Bình Thạnh, HCM', N'Phòng không có vách ngăn, nhà vệ sinh riêng, có bếp, có ban công', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (19, 4, NULL, 3, N'Phòng studio 3', 25, 2500000, N'Bình Thạnh, HCM', N'Phòng không có vách ngăn, nhà vệ sinh riêng, có bếp, có ban công', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (20, 4, NULL, 3, N'Phòng studio 4', 25, 2500000, N'Bình Thạnh, HCM', N'Phòng không có vách ngăn, nhà vệ sinh riêng, có bếp, có ban công', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (21, 4, NULL, 3, N'Phòng studio 5', 25, 2500000, N'Bình Thạnh, HCM', N'Phòng không có vách ngăn, nhà vệ sinh riêng, có bếp, có ban công', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (22, 4, 12, 3, N'Phòng đơn 7', 20, 2000000, N'Bình Thạnh, HCM', N'Phòng gác lửng, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 1)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (23, 4, NULL, 3, N'Phòng đơn 8', 20, 2000000, N'Bình Thạnh, HCM', N'Phòng gác lửng, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (24, 4, NULL, 3, N'Phòng đơn 9', 20, 2000000, N'Bình Thạnh, HCM', N'Phòng gác lửng, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (25, 4, NULL, 3, N'Phòng đơn 10', 20, 2000000, N'Bình Thạnh, HCM', N'Phòng gác lửng, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (26, 4, NULL, 3, N'Phòng đơn 11', 20, 2000000, N'Bình Thạnh, HCM', N'Phòng gác lửng, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (27, 2, 11, 4, N'Phòng đôi 1', 35, 3000000, N'Bình Thạnh, HCM', N'Phòng có 2 phòng nhỏ, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 1)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (28, 2, NULL, 4, N'Phòng đôi 2', 35, 3000000, N'Bình Thạnh, HCM', N'Phòng có 2 phòng nhỏ, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (29, 2, NULL, 4, N'Phòng đôi 3', 35, 3000000, N'Bình Thạnh, HCM', N'Phòng có 2 phòng nhỏ, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (30, 2, 13, 4, N'Phòng đôi 4', 35, 3000000, N'Bình Thạnh, HCM', N'Phòng có 2 phòng nhỏ, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 1)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (31, 2, NULL, 4, N'Phòng đôi 5', 35, 3000000, N'Bình Thạnh, HCM', N'Phòng có 2 phòng nhỏ, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (32, 2, NULL, 4, N'Phòng đôi 6', 35, 3000000, N'Bình Thạnh, HCM', N'Phòng có 2 phòng nhỏ, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (33, 2, NULL, 4, N'Phòng đôi 7', 35, 3000000, N'Bình Thạnh, HCM', N'Phòng có 2 phòng nhỏ, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (34, 2, NULL, 4, N'Phòng đôi 8', 35, 3000000, N'Bình Thạnh, HCM', N'Phòng có 2 phòng nhỏ, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (35, 2, NULL, 4, N'Phòng đôi 9', 35, 3000000, N'Bình Thạnh, HCM', N'Phòng có 2 phòng nhỏ, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (36, 2, NULL, 4, N'Phòng đôi 10', 35, 3000000, N'Bình Thạnh, HCM', N'Phòng có 2 phòng nhỏ, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (37, 5, 14, 4, N'Phòng suite 1', 35, 3000000, N'Bình Thạnh, HCM', N'Phòng có 2 phòng, ban công, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 1)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (38, 5, NULL, 4, N'Phòng suite 2', 35, 3000000, N'Bình Thạnh, HCM', N'Phòng có 2 phòng, ban công, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (39, 5, NULL, 4, N'Phòng suite 3', 35, 3000000, N'Bình Thạnh, HCM', N'Phòng có 2 phòng, ban công, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (40, 5, NULL, 4, N'Phòng suite 4', 35, 3000000, N'Bình Thạnh, HCM', N'Phòng có 2 phòng, ban công, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (41, 5, NULL, 4, N'Phòng suite 5', 35, 3000000, N'Bình Thạnh, HCM', N'Phòng có 2 phòng, ban công, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (42, 5, NULL, 4, N'Phòng suite 5', 35, 3000000, N'Bình Thạnh, HCM', N'Phòng có 2 phòng, ban công, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (43, 3, 15, 5, N'Phòng gia đình 6', 40, 3800000, N'Bình Thạnh, HCM', N'Phòng có 1 phòng lớn, 2 phòng nhỏ, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 1)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (44, 3, NULL, 5, N'Phòng gia đình 7', 40, 3800000, N'Bình Thạnh, HCM', N'Phòng có 1 phòng lớn, 2 phòng nhỏ, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (45, 3, NULL, 5, N'Phòng gia đình 8', 40, 3800000, N'Bình Thạnh, HCM', N'Phòng có 1 phòng lớn, 2 phòng nhỏ, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (46, 3, NULL, 5, N'Phòng gia đình 9', 40, 3800000, N'Bình Thạnh, HCM', N'Phòng có 1 phòng lớn, 2 phòng nhỏ, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (47, 3, NULL, 5, N'Phòng gia đình 10', 40, 3800000, N'Bình Thạnh, HCM', N'Phòng có 1 phòng lớn, 2 phòng nhỏ, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (48, 3, 16, 5, N'Phòng suite 6', 35, 3000000, N'Bình Thạnh, HCM', N'Phòng có 2 phòng, ban công, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 1)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (49, 3, NULL, 5, N'Phòng suite 7', 35, 3000000, N'Bình Thạnh, HCM', N'Phòng có 2 phòng, ban công, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (50, 3, NULL, 5, N'Phòng suite 8', 35, 3000000, N'Bình Thạnh, HCM', N'Phòng có 2 phòng, ban công, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (51, 3, NULL, 5, N'Phòng suite 9', 35, 3000000, N'Bình Thạnh, HCM', N'Phòng có 2 phòng, ban công, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (52, 3, NULL, 5, N'Phòng suite 10', 35, 3000000, N'Bình Thạnh, HCM', N'Phòng có 2 phòng, ban công, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (53, 2, 18, 5, N'Phòng đôi 11', 35, 3000000, N'Bình Thạnh, HCM', N'Phòng có 2 phòng nhỏ, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 1)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (54, 2, NULL, 5, N'Phòng đôi 12', 35, 3000000, N'Bình Thạnh, HCM', N'Phòng có 2 phòng nhỏ, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (55, 2, NULL, 5, N'Phòng đôi 13', 35, 3000000, N'Bình Thạnh, HCM', N'Phòng có 2 phòng nhỏ, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (56, 2, 19, 5, N'Phòng đôi 14', 35, 3000000, N'Bình Thạnh, HCM', N'Phòng có 2 phòng nhỏ, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 1)

INSERT [dbo].[PhongTro] ([Id], [IdMaLoaiPhong], [IdNguoiDung], [IdKhuVuc], [TenPhong], [DienTich], [GiaPhong], [DiaChi], [MoTa], [NgayTao], [NguoiTao], [NgaySua], [NguoiSua], [Trash], [Status]) VALUES (57, 2, NULL, 5, N'Phòng đôi 15', 35, 3000000, N'Bình Thạnh, HCM', N'Phòng có 2 phòng nhỏ, nhà vệ sinh riêng, có bếp', CAST(N'2023-07-29T23:17:20.710' AS DateTime), N'Phương Vũ', NULL, NULL, 1, 0)

SET IDENTITY\_INSERT [dbo].[PhongTro] OFF

## Thiết kế giao diện

### Logo

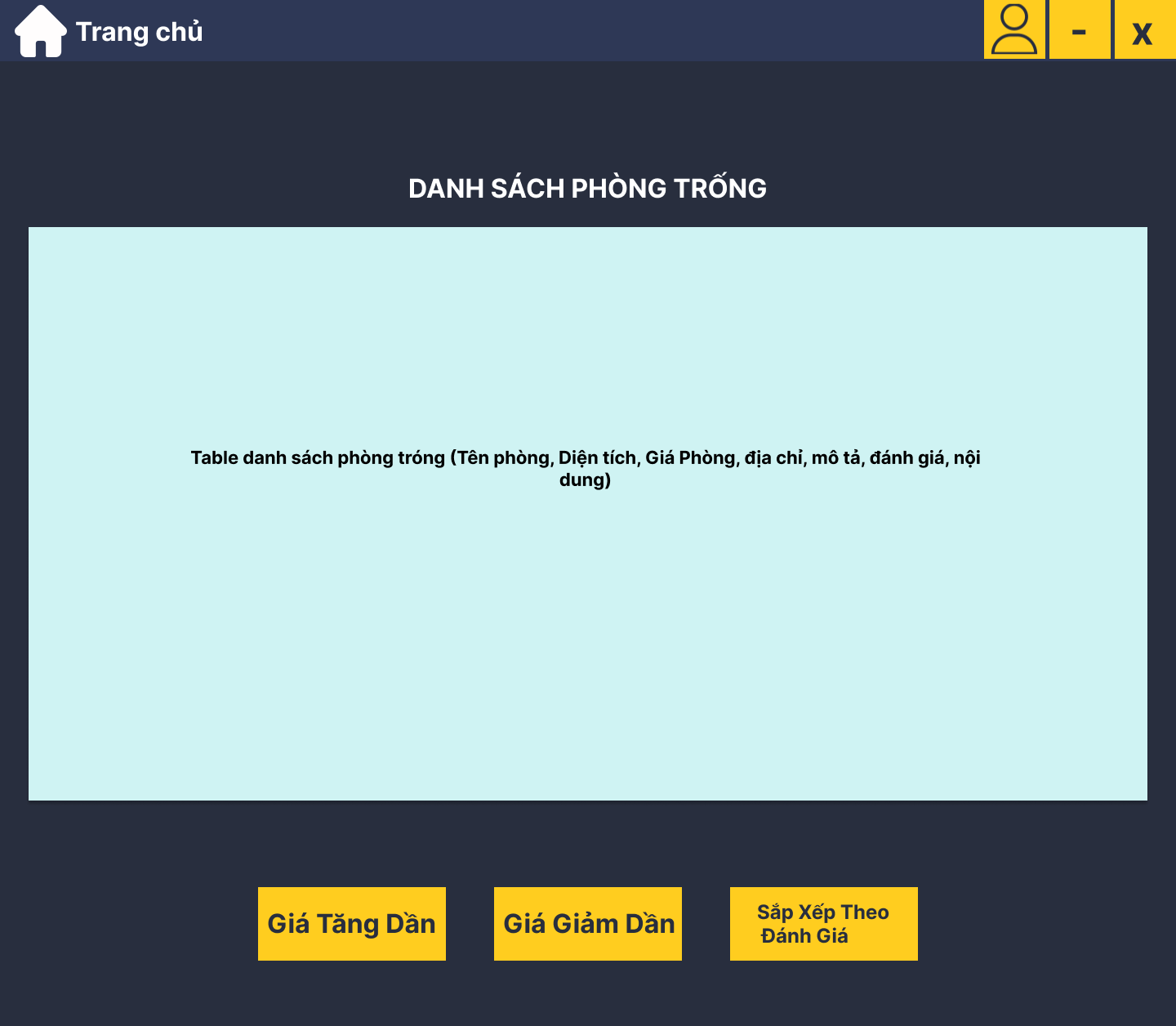
A house with trees and a sign

Description automatically generated

### Giao diện login

A screen shot of a house

Description automatically generated



A screenshot of a computer

Description automatically generated

A screenshot of a login screen

Description automatically generated

A screenshot of a computer

Description automatically generated



A screenshot of a login form

Description automatically generated

### Giao diện Chủ Trọ

A screenshot of a computer

Description automatically generated

A screenshot of a game

Description automatically generated



A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

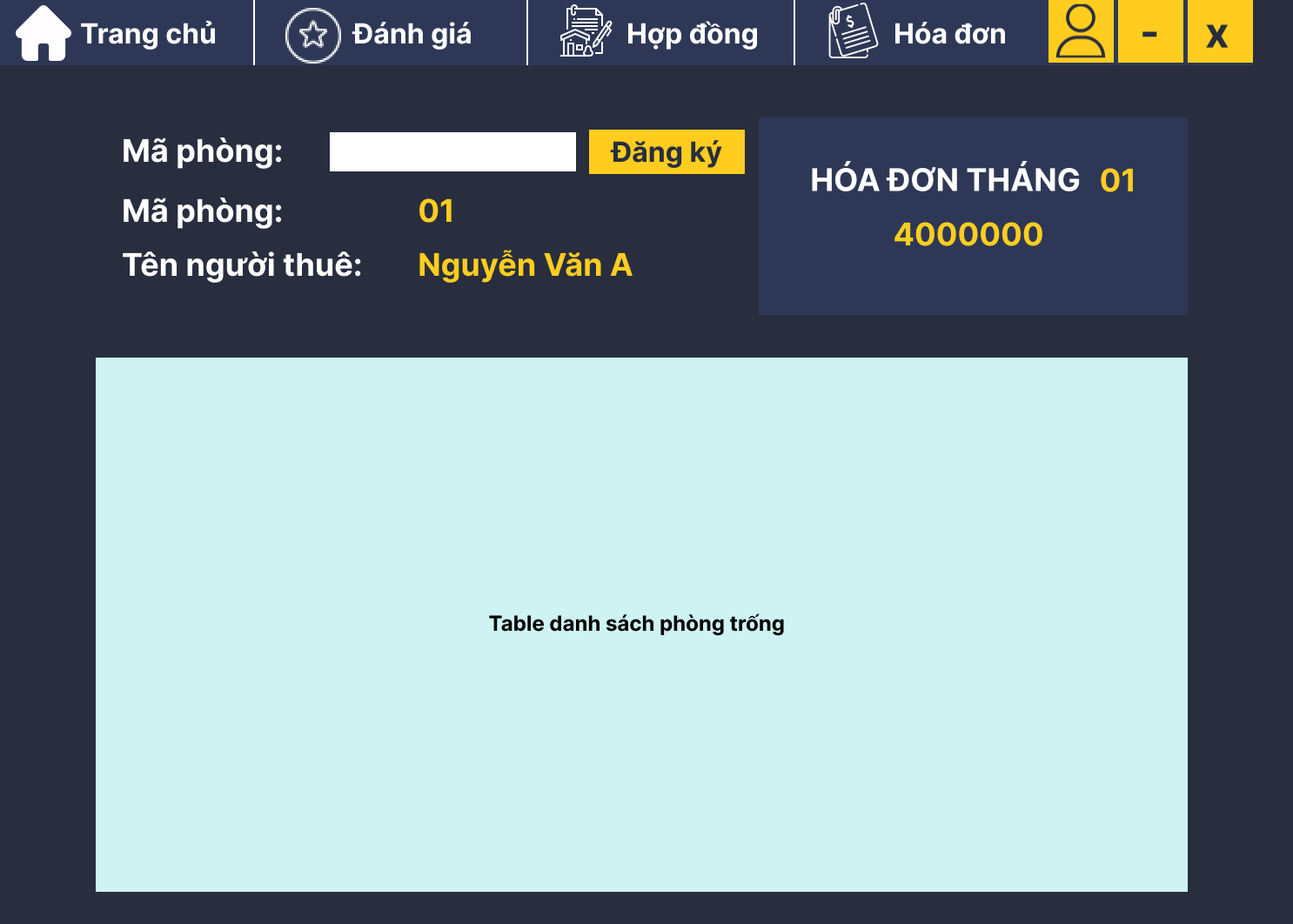
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

### Giao diện Người Dùng – Người Thuê



A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

# **Thực hiện dự án**

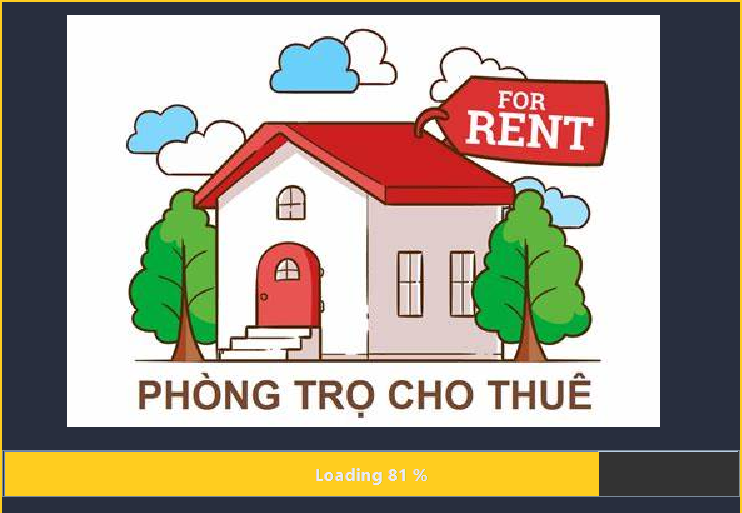
## Tạo giao diện với Swing:

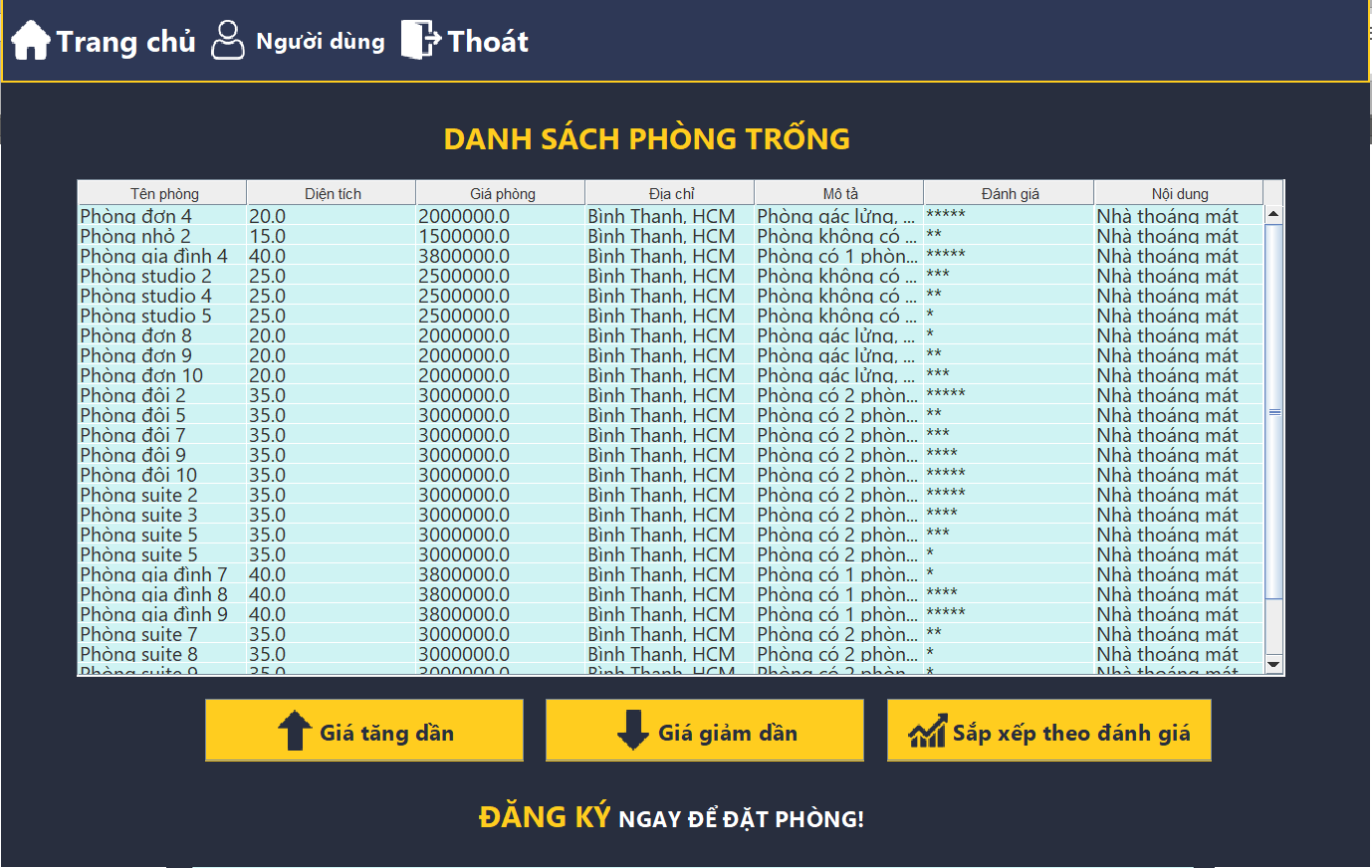
### Logo.

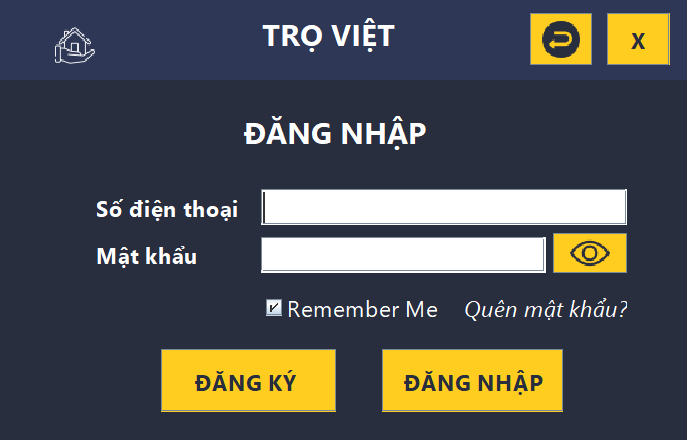
A house with trees and a sign

Description automatically generated

### Giao diện login.

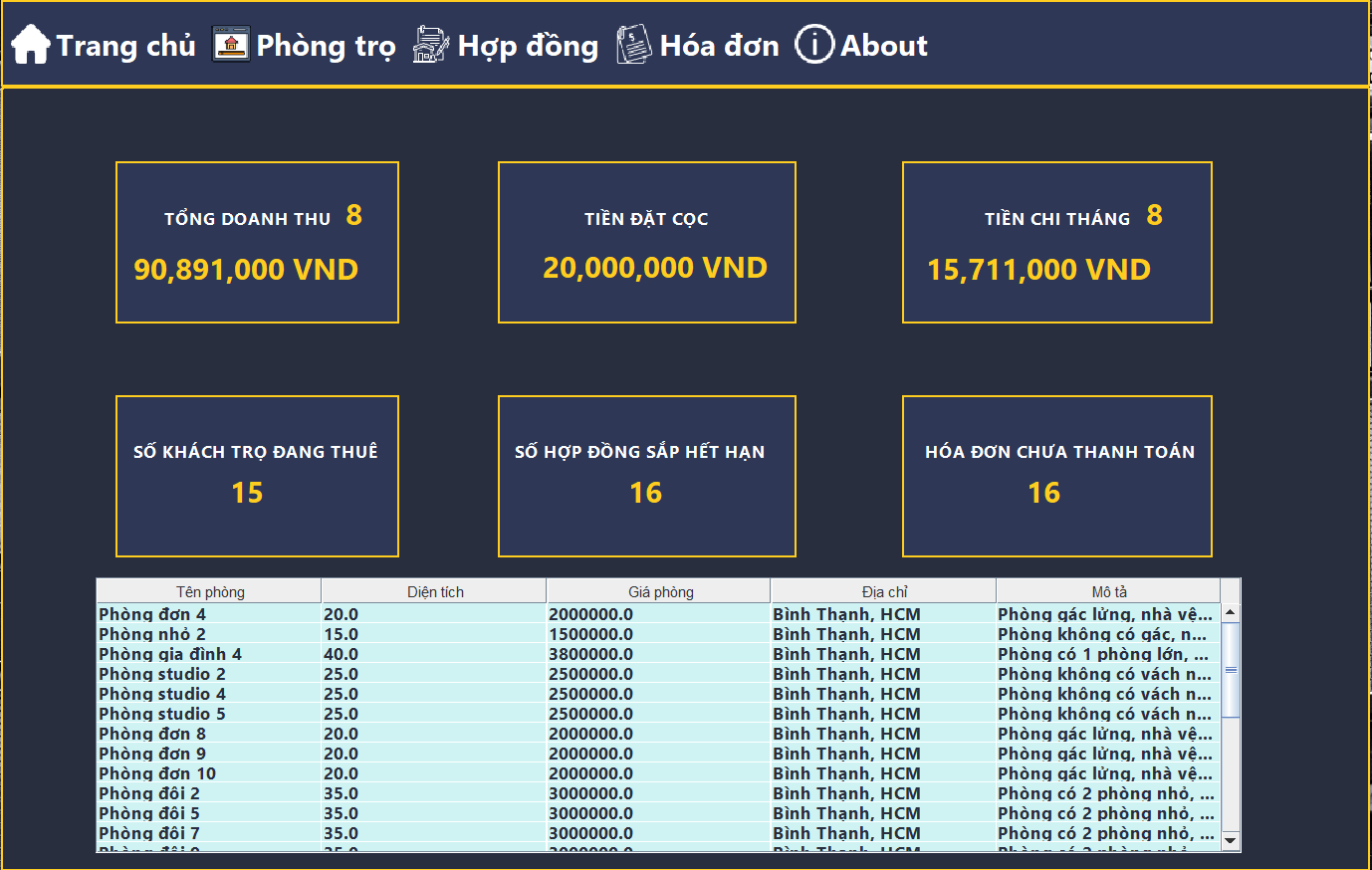


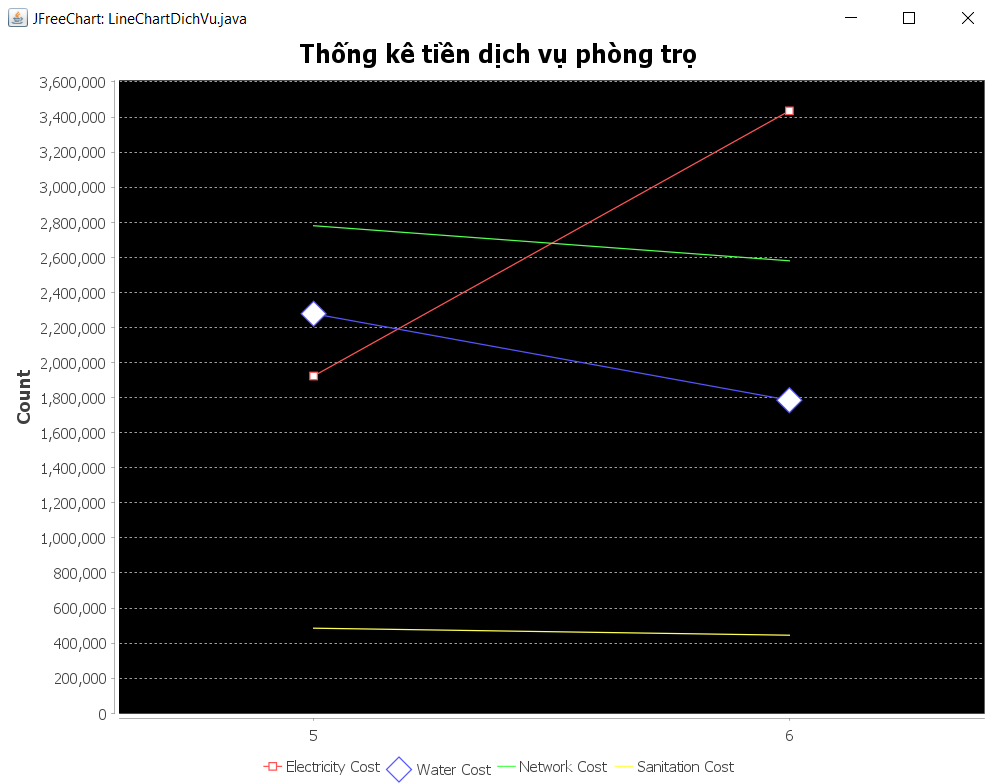






### Giao diện chủ trọ.

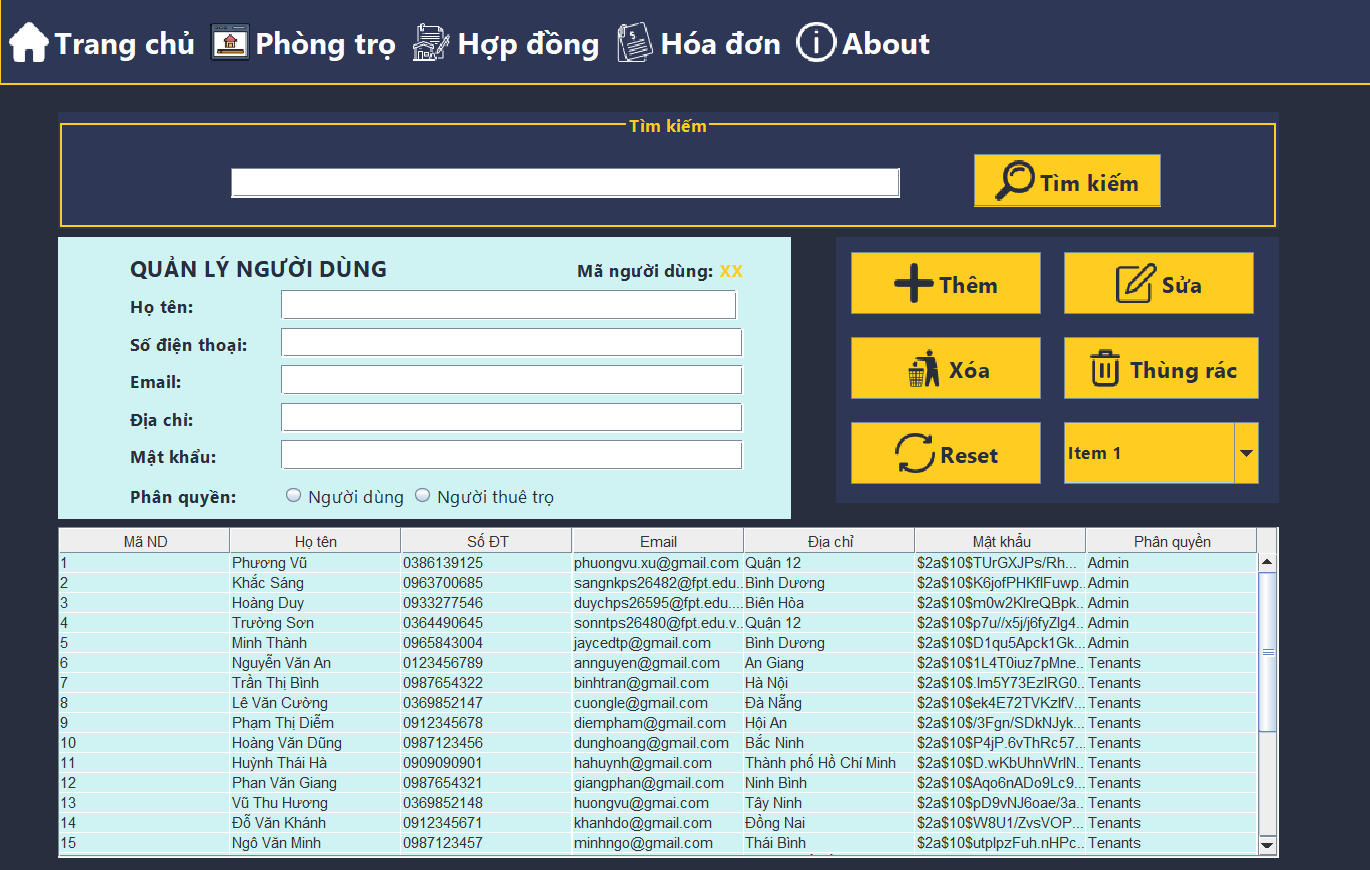


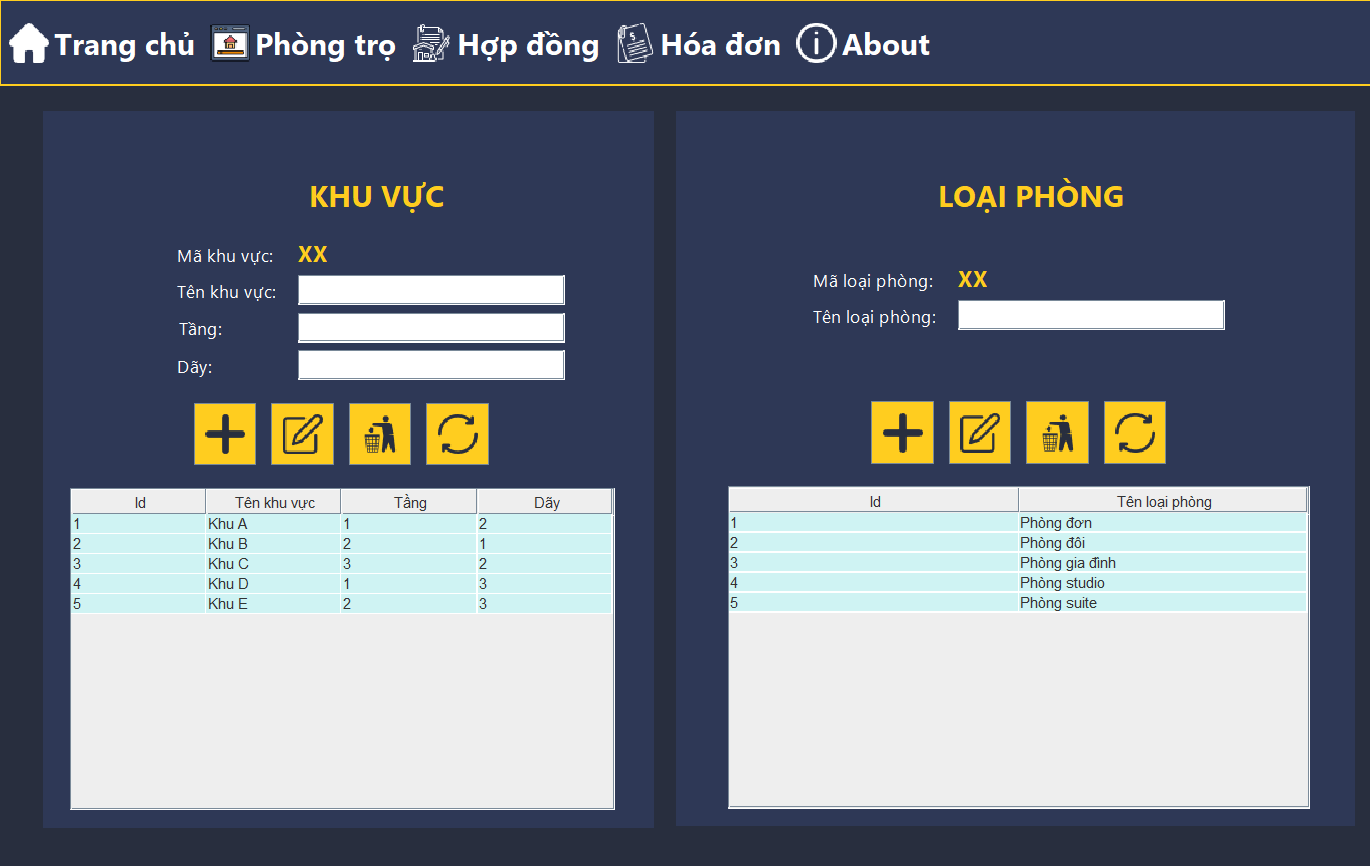


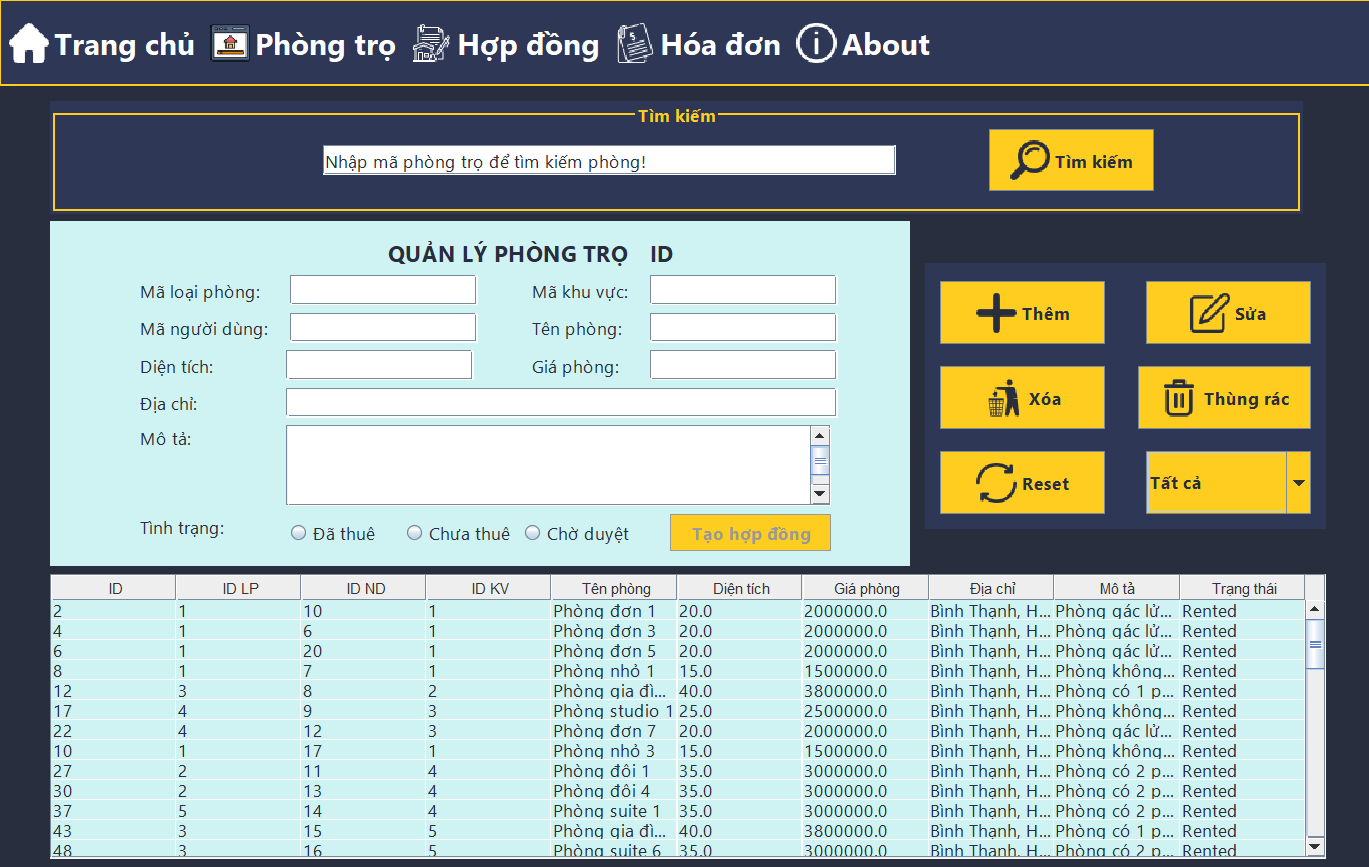


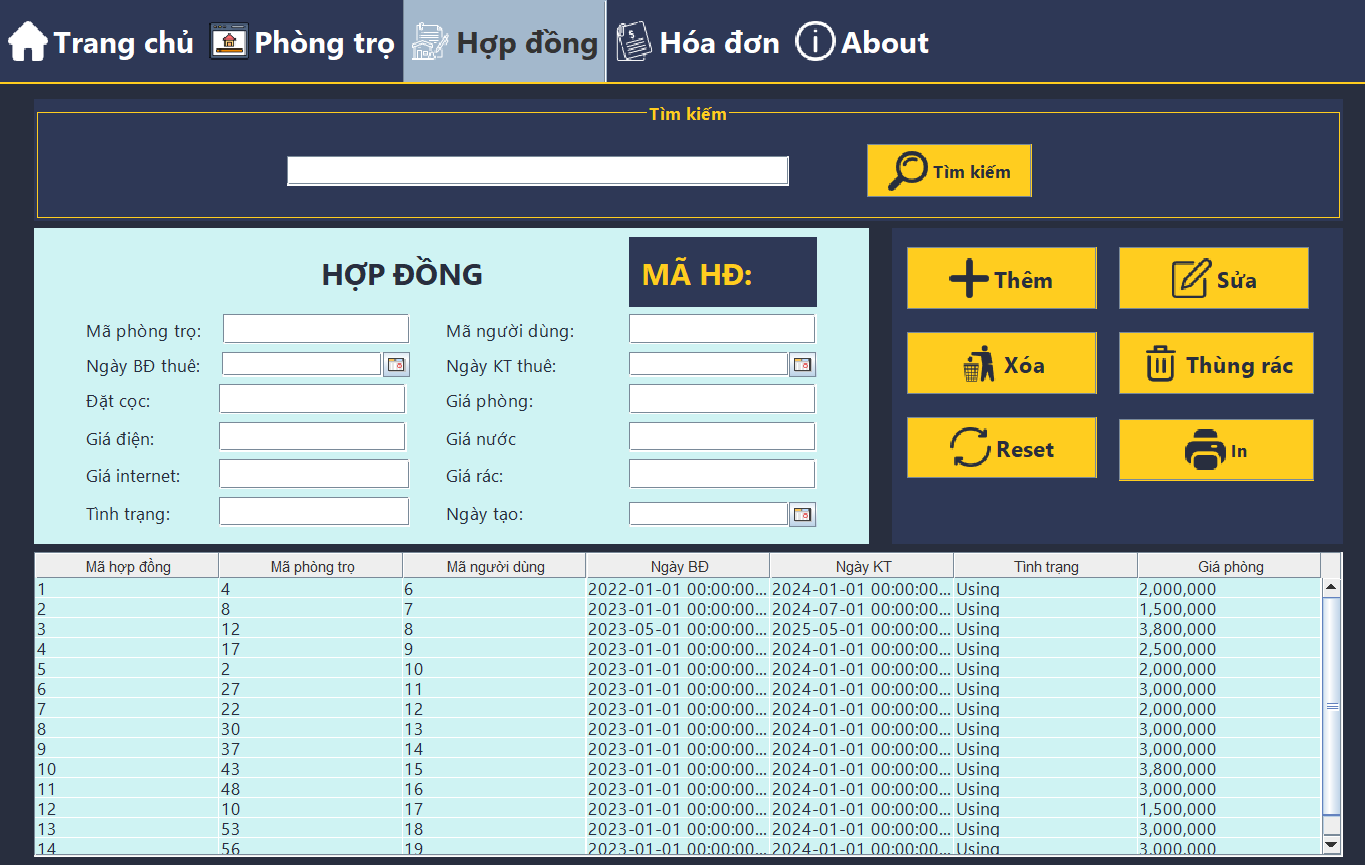


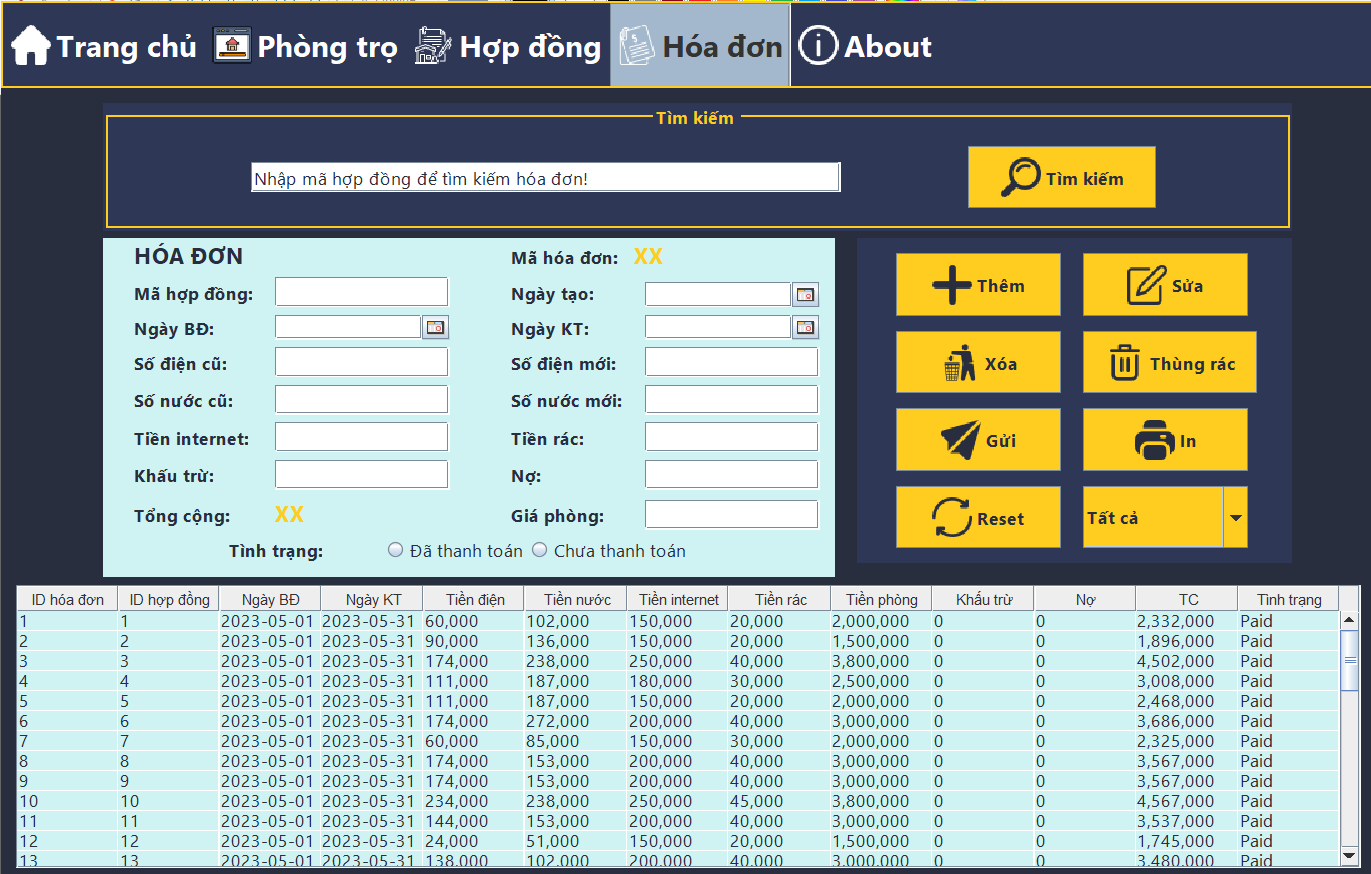




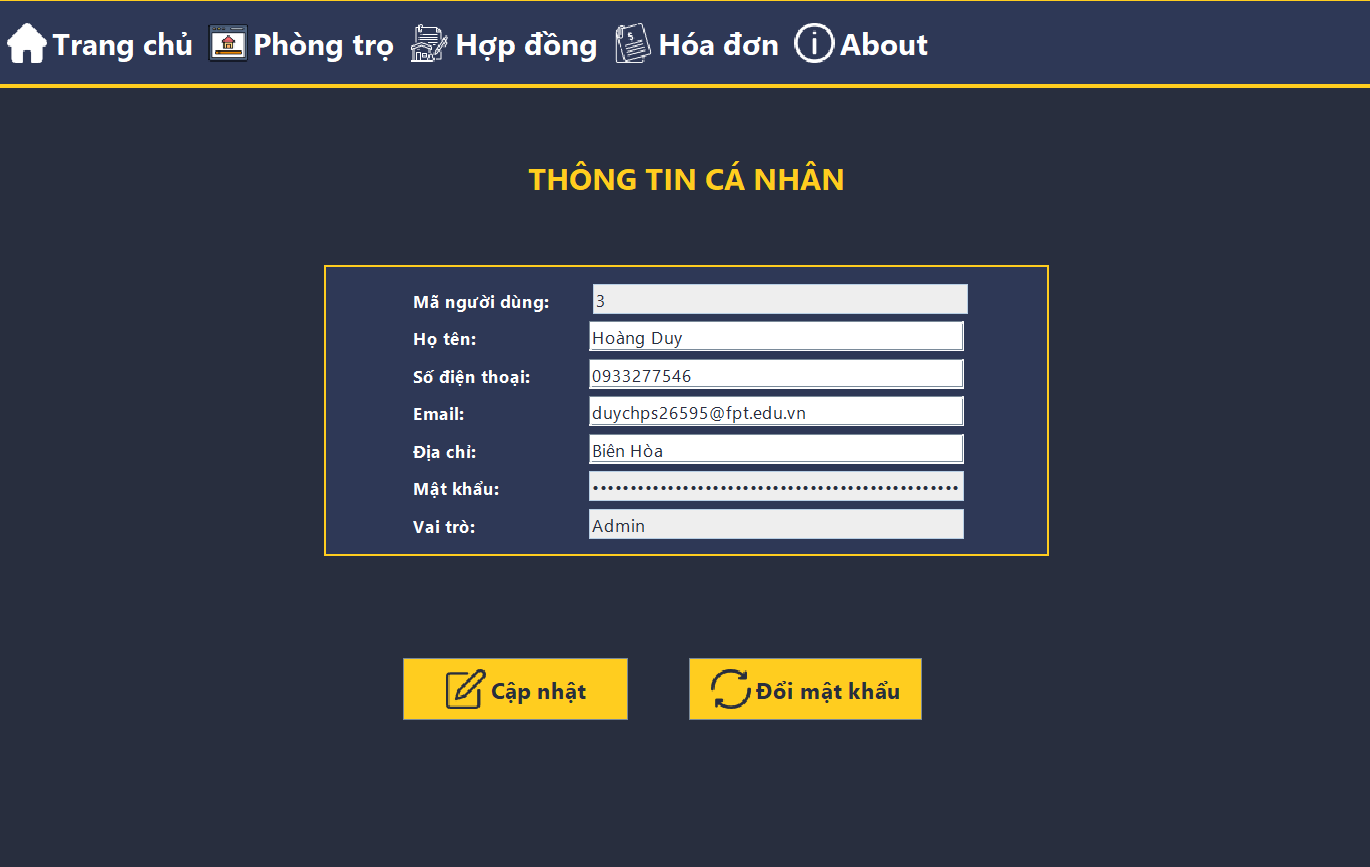










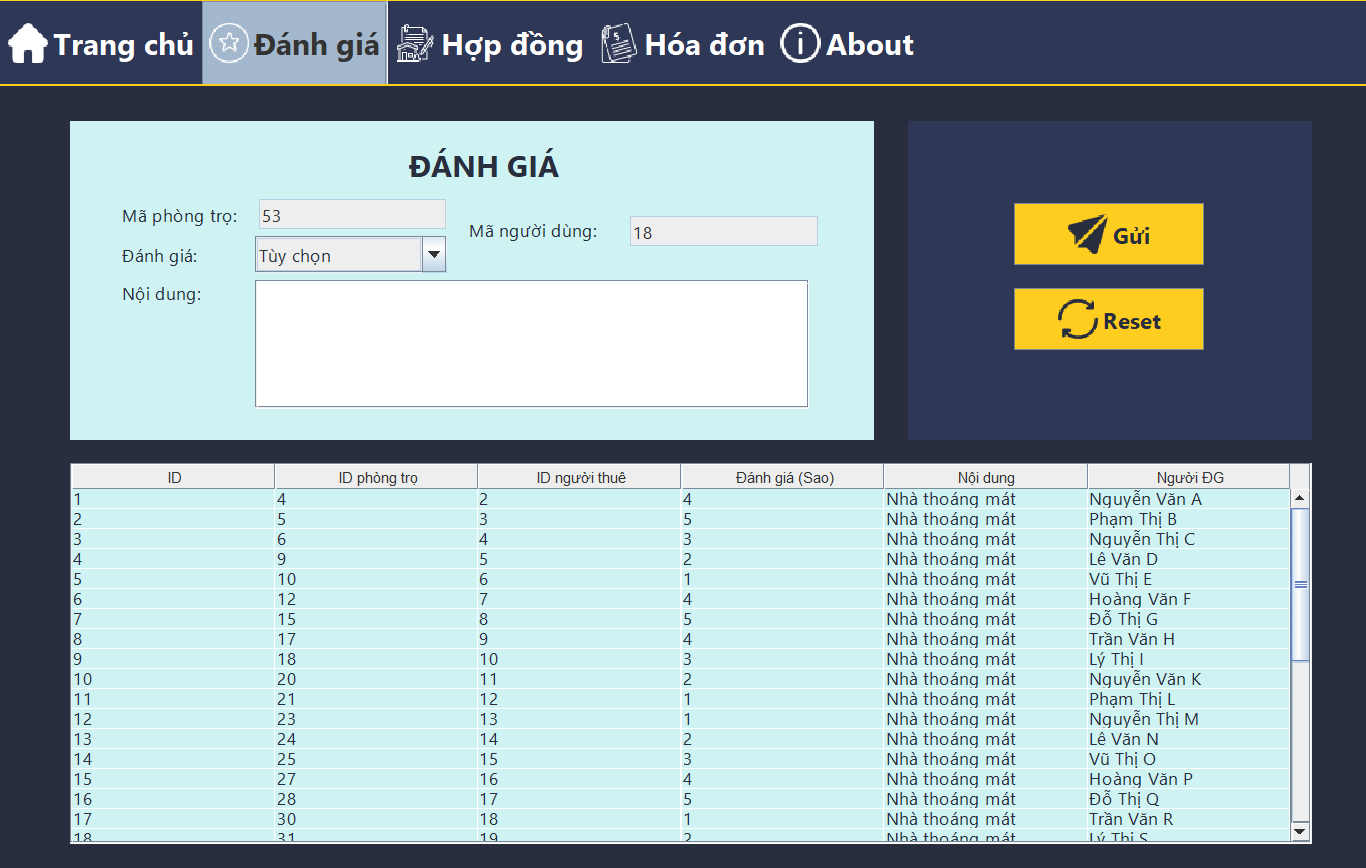




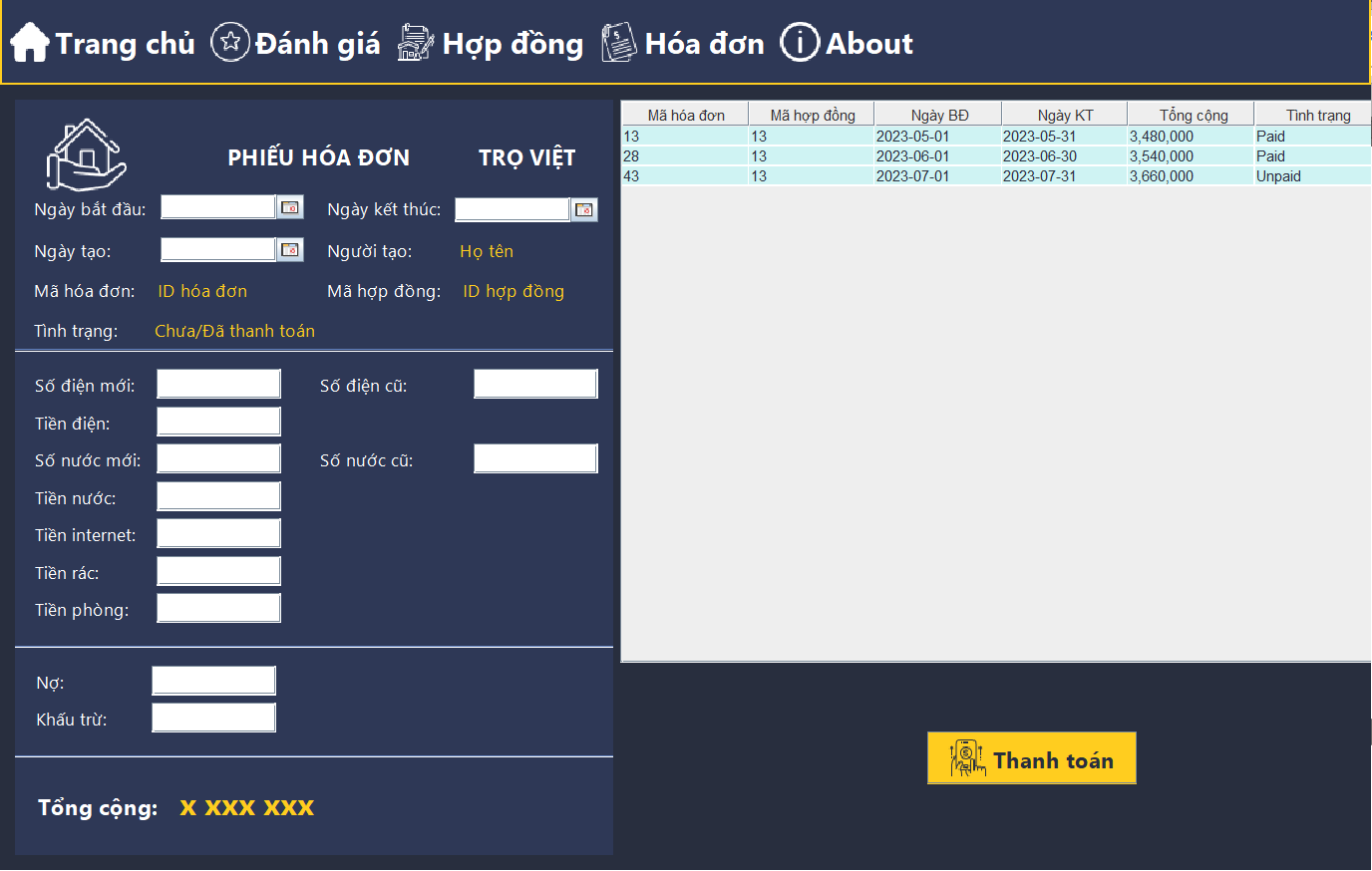


### 4.1.4. Giao diện người dùng.





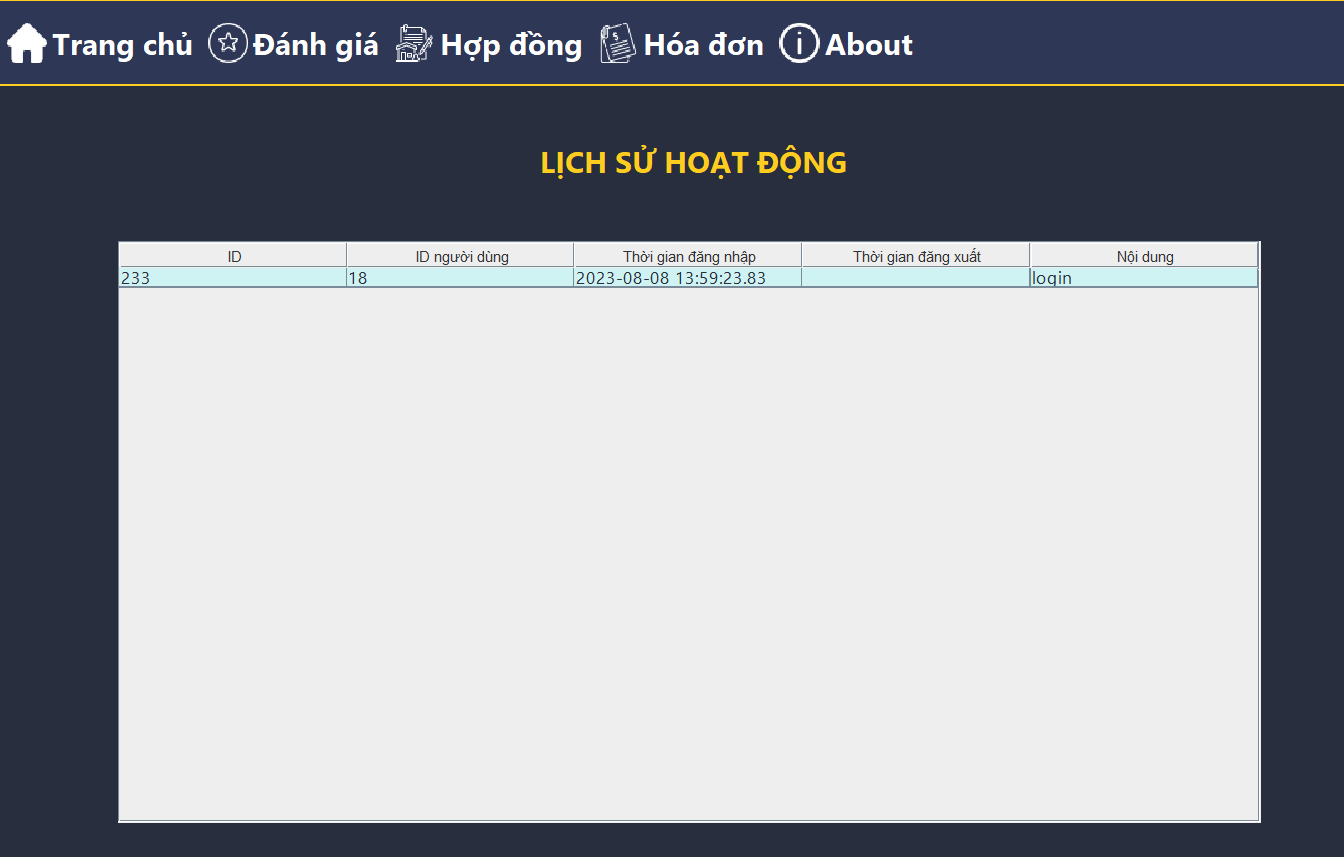






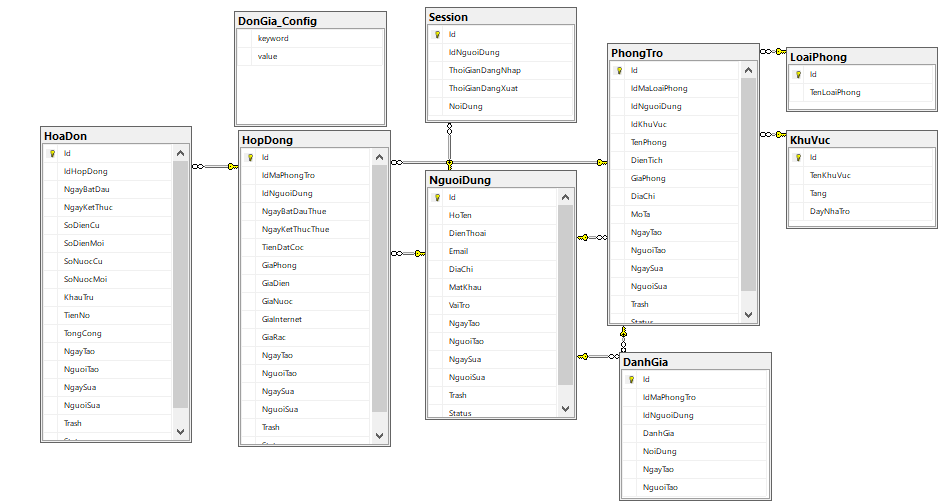




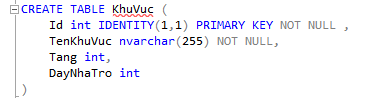


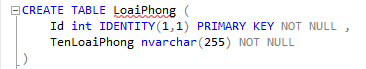
## Tạo CSDL với SQL Server:

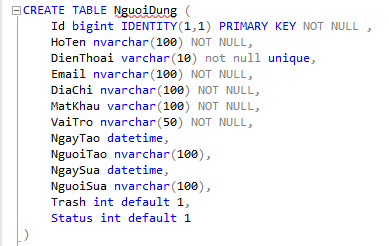
### Sơ đồ quan hệ:

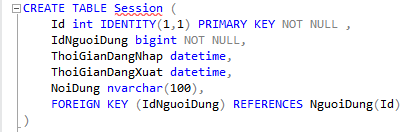


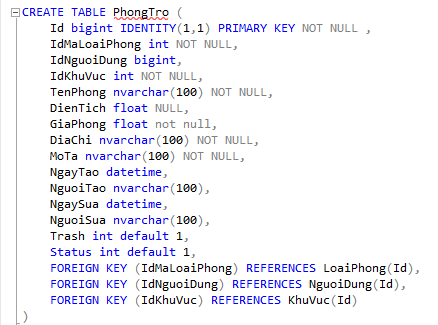
### Chi tiết các bảng:





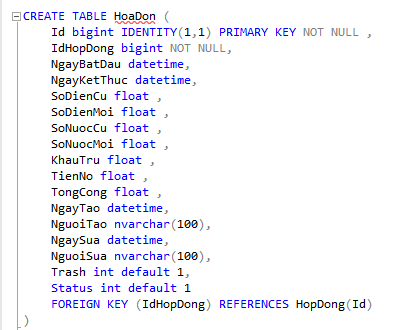




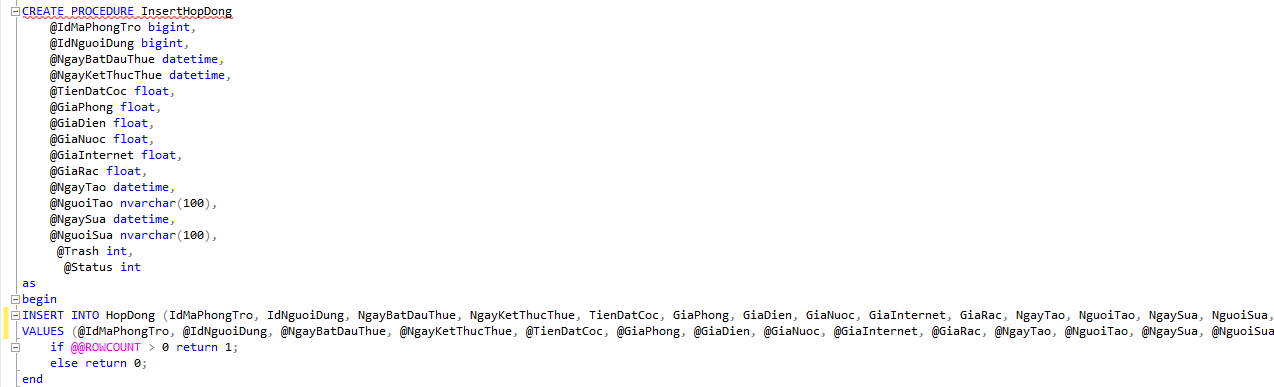






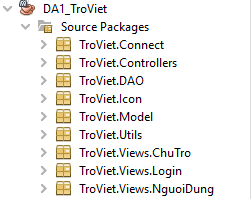


### Thủ tục lưu:



## Lập trình CSDL:

### Mô hình tổ chức lập trình CSDL:



### JDBC:



### Entity class và DAO class:

#### Entity khu vực và KhuVucDAO:

##### Entity khu vực:

package TroViet.Model;

public class KhuVuc {

private int id;

private String tenKhuVuc;

private int tang, day;

public KhuVuc() {

}

public KhuVuc(int id, String tenKhuVuc, int tang, int day) {

this.id = id;

this.tenKhuVuc = tenKhuVuc;

this.tang = tang;

this.day = day;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getTenKhuVuc() {

return tenKhuVuc;

}

public void setTenKhuVuc(String tenKhuVuc) {

this.tenKhuVuc = tenKhuVuc;

}

public int getTang() {

return tang;

}

public void setTang(int tang) {

this.tang = tang;

}

public int getDay() {

return day;

}

public void setDay(int day) {

this.day = day;

}

}

##### KhuVucDAO:

package TroViet.DAO;

import TroViet.Model.KhuVuc;

import TroViet.Connect.SQL\_Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

public class KhuVucDAO extends SQL\_Connection{

public ArrayList<KhuVuc> getALL() throws SQLException {

ArrayList<KhuVuc> listKhuVuc = new ArrayList<>();

String query = "select \* from KhuVuc;";

PreparedStatement ps = con.prepareStatement(query);

ResultSet rs = ps.executeQuery();

while (rs.next()) {

KhuVuc kv = new KhuVuc();

kv.setId(rs.getInt("Id"));

kv.setTenKhuVuc(rs.getString("TenKhuVuc"));

kv.setTang(rs.getInt("Tang"));

kv.setDay(rs.getInt("DayNhaTro"));

listKhuVuc.add(kv);

}

return listKhuVuc;

}

public KhuVuc get(int id) throws SQLException {

Statement st = con.createStatement();

String query = "select \* from KhuVuc where Id = " + id;

ResultSet rs = st.executeQuery(query);

if (rs.next()) {

KhuVuc kv = new KhuVuc();

kv.setId(rs.getInt("Id"));

kv.setTenKhuVuc(rs.getString("TenKhuVuc"));

kv.setTang(rs.getInt("Tang"));

kv.setDay(rs.getInt("DayNhaTro"));

return kv;

}

return null;

}

public void saveKV(KhuVuc kv) throws SQLException {

String query = "insert into KhuVuc values (?,?,?);";

PreparedStatement ps = con.prepareStatement(query);

// ps.setInt(1, kv.getId());

ps.setString(1, kv.getTenKhuVuc());

ps.setInt(2, kv.getTang());

ps.setInt(3, kv.getDay());

int result = ps.executeUpdate();

}

public String existsKV(String tenKhuVuc) throws SQLException {

String query = "select \* from KhuVuc where TenKhuVuc = ?;";

PreparedStatement ps = con.prepareStatement(query);

ps.setString(1, tenKhuVuc);

ResultSet rs = ps.executeQuery();

if (rs.next()) {

String yes = rs.getString("TenKhuVuc");

return yes;

}

return null;

}

public int updateKVByID(KhuVuc kv) throws SQLException {

try {

String query ="update KhuVuc set TenKhuVuc=?, Tang=?, DayNhaTro=? where Id=?;";

PreparedStatement ps = con.prepareStatement(query);

ps.setString(1, kv.getTenKhuVuc());

ps.setInt(2, kv.getTang());

ps.setInt(3, kv.getDay());

ps.setInt(4, kv.getId());

if (ps.executeUpdate() > 0) {

return 1;

}

} catch (Exception e) {

System.out.println("Success");

e.printStackTrace();

}

if (kv == null) {

throw new SQLException("Khuvuc rỗng");

}

return -1;

}

public void movedToBin(int idKV) throws SQLException {

String query = "UPDATE KhuVuc SET trash = ? WHERE Id = ?";

PreparedStatement stmt = con.prepareStatement(query);

stmt.setInt(1, 0);

stmt.setLong(2, idKV);

stmt.executeUpdate();

}

public ArrayList<KhuVuc> getAllFromBin() throws SQLException {

ArrayList<KhuVuc> listKV = new ArrayList<>();

Statement statement = con.createStatement();

String query = "SELECT \* FROM Khuvuc where Trash = 0 ";

ResultSet rs = statement.executeQuery(query);

while (rs.next()) {

KhuVuc kv = new KhuVuc();

kv.setId(rs.getInt("Id"));

kv.setTenKhuVuc(rs.getString("TenKhuVuc"));

kv.setTang(rs.getInt("Tang"));

kv.setDay(rs.getInt("DayNhaTro"));

listKV.add(kv);

}

return listKV;

}

public void restoredFromBin(int idKV) throws SQLException {

String query = "UPDATE KhuVuc SET trash = ? WHERE Id = ?";

PreparedStatement stmt = con.prepareStatement(query);

stmt.setInt(1, 1);

stmt.setLong(2, idKV);

stmt.executeUpdate();

}

public void deleteKVById(int idKV) throws SQLException {

PreparedStatement stmt = con.prepareStatement("DELETE FROM KhuVuc WHERE Id = ?");

stmt.setLong(1, idKV);

stmt.executeUpdate();

}

}

#### Entity loại phòng và LoaiPhongDAO:

##### Entity loại phòng:

package TroViet.Model;

public class LoaiPhong {

private int id;

private String tenLoaiPhong;

public LoaiPhong() {

}

public LoaiPhong(int id, String tenLoaiPhong) {

this.id = id;

this.tenLoaiPhong = tenLoaiPhong;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getTenLoaiPhong() {

return tenLoaiPhong;

}

public void setTenLoaiPhong(String tenLoaiPhong) {

this.tenLoaiPhong = tenLoaiPhong;

}

}

##### LoaiPhongDAO:

package TroViet.DAO;

import TroViet.Model.LoaiPhong;

import TroViet.Connect.SQL\_Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

public class LoaiPhongDAO extends SQL\_Connection{

public ArrayList<LoaiPhong> getAll() throws SQLException {

ArrayList<LoaiPhong> listLoaiPhong = new ArrayList<>();

String query = "select \* from LoaiPhong;";

PreparedStatement ps = con.prepareStatement(query);

ResultSet rs = ps.executeQuery();

while(rs.next()) {

LoaiPhong lp = new LoaiPhong();

lp.setId(rs.getInt("Id"));

lp.setTenLoaiPhong(rs.getString("TenLoaiPhong"));

listLoaiPhong.add(lp);

}

return listLoaiPhong;

}

public LoaiPhong get(int id) throws SQLException {

Statement st = con.createStatement();

String query = "select \* from LoaiPhong where Id = " + id;

ResultSet rs = st.executeQuery(query);

if (rs.next()) {

LoaiPhong lp = new LoaiPhong();

lp.setId(rs.getInt("Id"));

lp.setTenLoaiPhong(rs.getString("TenLoaiPhong"));

return lp;

}

return null;

}

public void saveLP(LoaiPhong lp) throws SQLException {

String query = "insert into LoaiPhong values (?);";

PreparedStatement ps = con.prepareStatement(query);

// ps.setInt(1, lp.getId());

ps.setString(1, lp.getTenLoaiPhong());

int result = ps.executeUpdate();

}

public String existsLP(String tenLoaiPhong) throws SQLException {

String query = "select \* from LoaiPhong where TenLoaiPhong = ?;";

PreparedStatement ps = con.prepareStatement(query);

ps.setString(1, tenLoaiPhong);

ResultSet rs = ps.executeQuery();

if (rs.next()) {

String yes = rs.getString("TenLoaiPhong");

return yes;

}

return null;

}

public int updateLPByID(LoaiPhong lp) throws SQLException {

try {

String query ="update LoaiPhong set TenLoaiPhong=? where Id=?;";

PreparedStatement ps = con.prepareStatement(query);

ps.setString(1, lp.getTenLoaiPhong());

ps.setInt(2, lp.getId());

if (ps.executeUpdate() > 0) {

return 1;

}

} catch (Exception e) {

System.out.println("Success");

e.printStackTrace();

}

if (lp == null) {

throw new SQLException("Khuvuc rỗng");

}

return -1;

}

public void deleteLPById(int idLP) throws SQLException {

PreparedStatement stmt = con.prepareStatement("DELETE FROM loaiPhong WHERE Id = ?");

stmt.setLong(1, idLP);

stmt.executeUpdate();

}

}

#### Entity phòng trọ, PhongTroDAO và PhongTroEnum:

##### Entity phòng trọ:

package TroViet.Model;

import TroViet.DAO.PhongTroEnum;

import java.security.Timestamp;

import java.sql.ResultSet;

import java.sql.SQLException;

public class PhongTro {

protected long id;

protected String tenphong,diachi,mota;

protected double dientich,giaphong;

protected Timestamp ngaytao,ngaysua;

protected String nguoitao,nguoisua;

protected int trash;

private PhongTroEnum status;

private int idMaloaiphong;

private long idNguoidung;

private int idKhuvuc;

public PhongTro() {

}

public long getId() {

return id;

}

public void setId(long id) {

this.id = id;

}

public String getTenphong() {

return tenphong;

}

public void setTenphong(String tenphong) {

this.tenphong = tenphong;

}

public String getDiachi() {

return diachi;

}

public void setDiachi(String diachi) {

this.diachi = diachi;

}

public String getMota() {

return mota;

}

public void setMota(String mota) {

this.mota = mota;

}

public double getDientich() {

return dientich;

}

public void setDientich(double dientich) {

this.dientich = dientich;

}

public double getGiaphong() {

return giaphong;

}

public void setGiaphong(double giaphong) {

this.giaphong = giaphong;

}

public Timestamp getNgaytao() {

return ngaytao;

}

public void setNgaytao(Timestamp ngaytao) {

this.ngaytao = ngaytao;

}

public Timestamp getNgaysua() {

return ngaysua;

}

public void setNgaysua(Timestamp ngaysua) {

this.ngaysua = ngaysua;

}

public String getNguoitao() {

return nguoitao;

}

public void setNguoitao(String nguoitao) {

this.nguoitao = nguoitao;

}

public String getNguoisua() {

return nguoisua;

}

public void setNguoisua(String nguoisua) {

this.nguoisua = nguoisua;

}

public int getTrash() {

return trash;

}

public void setTrash(int trash) {

this.trash = trash;

}

public PhongTroEnum getStatus() {

return status;

}

public void setStatus(PhongTroEnum status) {

this.status = status;

}

public int getIdMaloaiphong() {

return idMaloaiphong;

}

public void setIdMaloaiphong(int idMaloaiphong) {

this.idMaloaiphong = idMaloaiphong;

}

public long getIdNguoidung() {

return idNguoidung;

}

public void setIdNguoidung(long idNguoidung) {

this.idNguoidung = idNguoidung;

}

public int getIdKhuvuc() {

return idKhuvuc;

}

public void setIdKhuvuc(int idKhuvuc) {

this.idKhuvuc = idKhuvuc;

}

public static PhongTro getFromResultSet(ResultSet rs) throws SQLException{

PhongTro s = new PhongTro();

s.setId(rs.getLong("Id"));

s.setIdMaloaiphong(rs.getInt("IdMaLoaiPhong"));

s.setIdNguoidung(rs.getLong("IdNguoiDung"));

s.setIdKhuvuc(rs.getInt("IdKhuVuc"));

s.setTenphong(rs.getString("TenPhong"));

s.setDientich(rs.getDouble("DienTich"));

s.setGiaphong(rs.getDouble("GiaPhong"));

s.setDiachi(rs.getString("DiaChi"));

s.setMota(rs.getString("MoTa"));

s.setNguoitao(rs.getString("NguoiTao"));

s.setNguoisua(rs.getString("NguoiSua"));

s.setTrash(rs.getInt("Trash"));

s.setStatus(PhongTroEnum.getById(rs.getString("Status")));

return s;

}

@Override

public String toString() {

return tenphong;

}

}

##### PhongTroDAO:

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

package TroViet.DAO;

import TroViet.Connect.SQL\_Connection;

import TroViet.Model.DanhGia;

import TroViet.Model.PhongTro;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.Vector;

import java.util.logging.Level;

import java.util.logging.Logger;

/\*\*

\*

\* @author WINDOWS

\*/

public class PhongTroDAO extends SQL\_Connection {

public ArrayList<PhongTro> loadPhongTroData(String selectedStatus) throws SQLException {

ArrayList<PhongTro> phongtro = new ArrayList<>();

Statement statement = con.createStatement();

String query;

if (selectedStatus == PhongTroEnum.Not\_Yet.getName()) {

query = "SELECT \* FROM PhongTro WHERE Status = " + PhongTroEnum.Not\_Yet.getId();

} else if(selectedStatus == PhongTroEnum.Rented.getName()) {

query = "SELECT \* FROM PhongTro WHERE Status = " + PhongTroEnum.Rented.getId();

} else if(selectedStatus == PhongTroEnum.Waiting.getName()) {

query = "SELECT \* FROM PhongTro WHERE Status = " + PhongTroEnum.Waiting.getId();

}else {

query = "SELECT \* FROM PhongTro WHERE trash != 0";

}

ResultSet rs = statement.executeQuery(query);

while (rs.next()) {

PhongTro pt = PhongTro.getFromResultSet(rs);

phongtro.add(pt);

}

return phongtro;

}

public ArrayList<PhongTro> troInTrangChu() {

ArrayList<PhongTro> ptro = new ArrayList<>();

try {

String query = "select TenPhong, DienTich, GiaPhong, DiaChi, MoTa from PhongTro where PhongTro.idNguoiDung is null;";

PreparedStatement ps = con.prepareStatement(query);

ResultSet rs = ps.executeQuery();

while (rs.next()) {

PhongTro tro = new PhongTro();

tro.setTenphong(rs.getString("TenPhong"));

tro.setDientich(rs.getDouble("DienTich"));

tro.setGiaphong(rs.getDouble("GiaPhong"));

tro.setDiachi(rs.getString("DiaChi"));

tro.setMota(rs.getString("MoTa"));

ptro.add(tro);

}

} catch (SQLException ex) {

ex.printStackTrace();

}

return ptro;

}

public ArrayList<PhongTro> getAll() throws SQLException {

ArrayList<PhongTro> phongtro = new ArrayList<>();

Statement statement = con.createStatement();

String query = "SELECT \* FROM PhongTro where trash !=0 order by Status desc";

ResultSet rs = statement.executeQuery(query);

while (rs.next()) {

PhongTro pt = PhongTro.getFromResultSet(rs);

phongtro.add(pt);

}

return phongtro;

}

public PhongTro get(Long id) throws SQLException {

Statement statement = con.createStatement();

String query = "SELECT \* FROM PhongTro WHERE PhongTro.id = " + id;

ResultSet rs = statement.executeQuery(query);

if (rs.next()) {

PhongTro pt = PhongTro.getFromResultSet(rs);

return pt;

}

return null;

}

public void save(PhongTro t) throws SQLException {

if (t == null) {

throw new SQLException("Phòng trọ trống");

}

String query = "INSERT INTO PhongTro (idMaLoaiPhong, idNguoiDung, idKhuvuc, TenPhong,DienTich, GiaPhong, DiaChi, MoTa, Status)"

+ " VALUES (?, ?, ?, ?, ?,?,?,?,?)";

PreparedStatement stmt = con.prepareStatement(query);

stmt.setInt(1, t.getIdMaloaiphong());

stmt.setLong(2, t.getIdNguoidung());

stmt.setInt(3, t.getIdKhuvuc());

stmt.setString(4, t.getTenphong());

stmt.setDouble(5, t.getDientich());

stmt.setDouble(6, t.getGiaphong());

stmt.setString(7, t.getDiachi());

stmt.setString(8, t.getMota());

stmt.setString(9, t.getStatus().getId());

int row = stmt.executeUpdate();

}

public boolean dangKyPhong(long idNguoiDung, long idPhongTro) throws SQLException {

String query = "update PhongTro set Status = 2, IdNguoiDung = ? where Id = ?;";

PreparedStatement ps = con.prepareStatement(query);

ps.setLong(1, idNguoiDung);

ps.setLong(2, idPhongTro);

int rowsAffected = ps.executeUpdate();

return rowsAffected > 0; // Trả về true nếu cập nhật thành công, ngược lại trả về false

}

public void update(PhongTro t) throws SQLException {

if (t == null) {

throw new SQLException("Phòng Trọ rỗng");

}

String query = "UPDATE PhongTro SET IdMaLoaiPhong = ?, IdNguoiDung = ?, IdKhuVuc = ?, TenPhong = ?, DienTich = ?, GiaPhong = ?, DiaChi = ?, MoTa = ?, Status = ? WHERE Id = ?";

PreparedStatement stmt = con.prepareStatement(query);

stmt.setInt(1, t.getIdMaloaiphong());

stmt.setLong(2, t.getIdNguoidung());

stmt.setInt(3, t.getIdKhuvuc());

stmt.setString(4, t.getTenphong());

stmt.setDouble(5, t.getDientich());

stmt.setDouble(6, t.getGiaphong());

stmt.setString(7, t.getDiachi());

stmt.setString(8, t.getMota());

stmt.setString(9, t.getStatus().getId());

stmt.setLong(10, t.getId());

System.out.println("DEBUG - SQL Query: " + stmt.toString()); // In câu truy vấn SQL

int rowsAffected = stmt.executeUpdate();

System.out.println("DEBUG - Rows Affected: " + rowsAffected); // In số dòng bị ảnh hưởng

if (rowsAffected > 0) {

System.out.println("Dữ liệu đã được cập nhật thành công.");

} else {

System.out.println("Không tìm thấy dữ liệu phù hợp để cập nhật.");

}

}

public int updateStatus(Long id) {

try {

String query = "update PhongTro set Status=1 where Id=?;";

PreparedStatement ps = con.prepareStatement(query);

ps.setLong(1, id);

if (ps.executeUpdate() > 0) {

return 1;

}

} catch (Exception e) {

e.printStackTrace();

}

return -1;

}

public void deleteFromRecycle(Long id) throws SQLException {

if (id == null) {

throw new SQLException("Phòng Trọ rỗng");

}

String query = "UPDATE PhongTro SET trash = ? WHERE PhongTro.id = ?";

PreparedStatement stmt = con.prepareStatement(query);

stmt.setInt(1, 0);

stmt.setLong(2, id);

stmt.executeUpdate();

}

public ArrayList<PhongTro> searchByKey(String keyword) throws SQLException {

ArrayList<PhongTro> pt = new ArrayList<>();

Statement statement = con.createStatement();

String query = "SELECT \* FROM PhongTro WHERE idMaLoaiPhong LIKE '%" + keyword + "%';";

ResultSet rs = statement.executeQuery(query);

while (rs.next()) {

PhongTro phongtro = PhongTro.getFromResultSet(rs);

pt.add(phongtro);

}

return pt;

}

public ArrayList<PhongTro> getAllFromRecycle() throws SQLException {

ArrayList<PhongTro> phongtro = new ArrayList<>();

Statement statement = con.createStatement();

String query = "SELECT \* FROM PhongTro where trash = 0";

ResultSet rs = statement.executeQuery(query);

while (rs.next()) {

PhongTro pt = PhongTro.getFromResultSet(rs);

phongtro.add(pt);

}

return phongtro;

}

public void deleteById(Long id) throws SQLException {

PreparedStatement stmt = con.prepareStatement("DELETE FROM PhongTro WHERE PhongTro.id = ?");

stmt.setLong(1, id);

stmt.executeUpdate();

}

public void RestoreFromRecycle(Long id) throws SQLException {

if (id == null) {

throw new SQLException("phòng trọ rỗng");

}

String query = "UPDATE PhongTro SET trash = ? WHERE PhongTro.id = ?";

PreparedStatement stmt = con.prepareStatement(query);

stmt.setInt(1, 1);

stmt.setLong(2, id);

stmt.executeUpdate();

}

}

##### PhongTroEnum:

package TroViet.DAO;

public enum PhongTroEnum {

Not\_Yet("0", "Còn trống"),

Rented("1", "Đã thuê"),

Waiting ("2", "Chờ duyệt");

private String id, name;

PhongTroEnum(String id, String name) {

this.id = id;

this.name = name;

}

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public static PhongTroEnum getById(String id) {

for (PhongTroEnum e : values()) {

if (e.id.equals(id)) {

return e;

}

}

return Not\_Yet;

}

public static PhongTroEnum getByName(String name) {

for (PhongTroEnum e : values()) {

if (e.name.equals(name)) {

return e;

}

}

return Not\_Yet;

}

}

#### Entity user, UserDAO và UserRole:

##### Entity user:

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

package TroViet.Model;

import TroViet.DAO.UserRole;

public class UserEntity {

private Long id;

private String hoTen;

private String email;

private String DienThoai;

private String diaChi;

private String password;

protected UserRole role;

private int trash;

private UserRole status;

private String ngayTao;

private String nguoiTao;

private String ngaySua;

private String nguoiSua;

public boolean checkPassword(String password) {

return this.password.equals(password);

}

public void setId(Long id) {

this.id = id;

}

public void setHoTen(String hoTen) {

this.hoTen = hoTen;

}

public void setEmail(String email) {

this.email = email;

}

public void setDienThoai(String DienThoai) {

this.DienThoai = DienThoai;

}

public void setDiaChi(String diaChi) {

this.diaChi = diaChi;

}

public void setPassword(String password) {

this.password = password;

}

public void setRole(UserRole role) {

this.role = role;

}

public void setTrash(int trash) {

this.trash = trash;

}

public void setStatus(UserRole status) {

this.status = status;

}

public void setNgayTao(String ngayTao) {

this.ngayTao = ngayTao;

}

public void setNguoiTao(String nguoiTao) {

this.nguoiTao = nguoiTao;

}

public void setNgaySua(String ngaySua) {

this.ngaySua = ngaySua;

}

public void setNguoiSua(String nguoiSua) {

this.nguoiSua = nguoiSua;

}

public Long getId() {

return id;

}

public String getHoTen() {

return hoTen;

}

public String getEmail() {

return email;

}

public String getDienThoai() {

return DienThoai;

}

public String getDiaChi() {

return diaChi;

}

public String getPassword() {

return password;

}

public UserRole getRole() {

return role;

}

public int getTrash() {

return trash;

}

public UserRole getStatus() {

return status;

}

public String getNgayTao() {

return ngayTao;

}

public String getNguoiTao() {

return nguoiTao;

}

public String getNgaySua() {

return ngaySua;

}

public String getNguoiSua() {

return nguoiSua;

}

}

##### UserDAO:

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

package TroViet.DAO;

import TroViet.Connect.SQL\_Connection;

import java.sql.\*;

import java.time.LocalDateTime;

import javax.swing.JOptionPane;

import TroViet.Model.UserEntity;

import org.mindrot.jbcrypt.BCrypt;

import TroViet.Utils.SessionManager;

import java.util.ArrayList;

/\*\*

\*

\* @author WINDOWS

\*/

public class UserDAO extends SQL\_Connection {

// String url = "jdbc:sqlserver://localhost:1433;databaseName=TroViet";

// String username = "sa";

// String password = "123";

//

// private Connection connection;

//

// public UserDAO() {

// try {

// connection = DriverManager.getConnection(url, username, password);

// } catch (SQLException e) {

// e.printStackTrace();

// }

// }

public void saveAccountToDatabase(String hoTen, String dienThoai, String email, String diaChi, String matKhau, String vaiTro) {

// String user = SessionManager.getSession().getUser().getHoTen();

try {

LocalDateTime currentTime = LocalDateTime.now();

String sql = "INSERT INTO NguoiDung (HoTen, DienThoai, Email, DiaChi, MatKhau, VaiTro, NgayTao, NguoiTao) "

+ "VALUES (?, ?, ?, ?, ?, ?, ?, ?)";

PreparedStatement statement = con.prepareStatement(sql);

statement.setString(1, hoTen);

statement.setString(2, dienThoai);

statement.setString(3, email);

statement.setString(4, diaChi);

statement.setString(5, matKhau);

statement.setString(6, vaiTro);

statement.setObject(7, currentTime);

statement.setString(8, SessionManager.getSession().getUser().getHoTen()); // Thay thế "your\_nguoitao\_value" bằng giá trị tương ứng

// Thay thế "your\_nguoisua\_value" bằng giá trị tương ứng

statement.executeUpdate();

System.out.println("success");

} catch (SQLException ex) {

System.out.println("falll");

ex.printStackTrace();

}

}

public void saveNewPassword(String email, String newPassword) {

String hashedPassword = BCrypt.hashpw(newPassword, BCrypt.gensalt());

try {

String query = "UPDATE NguoiDung SET MatKhau = ? WHERE Email = ?";

PreparedStatement statement = con.prepareStatement(query);

statement.setString(1, hashedPassword);

statement.setString(2, email);

statement.executeUpdate();

} catch (SQLException e) {

e.printStackTrace();

}

}

public UserEntity getUserEntity(String username) {

UserEntity ad = new UserEntity();

try {

String query = "Select \* from NguoiDung where HoTen = ?";

PreparedStatement ps = con.prepareStatement(query);

ps.setString(1, username);

ResultSet rs = ps.executeQuery();

while (rs.next()) {

ad.setHoTen(rs.getNString(2));

ad.setPassword(rs.getNString(6));

return ad;

}

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

public static UserEntity getFromResultSet(ResultSet rs) throws SQLException {

UserEntity e = new UserEntity();

e.setId(rs.getLong("Id"));

e.setHoTen(rs.getString("HoTen"));

e.setDienThoai(rs.getString("DienThoai"));

e.setEmail(rs.getString("Email"));

e.setDiaChi(rs.getString("DiaChi"));

e.setPassword(rs.getString("MatKhau"));

e.setRole(UserRole.getById(rs.getString("VaiTro")));

return e;

}

public UserEntity findByUsername(String name) throws SQLException {

PreparedStatement statement = null;

ResultSet rs = null;

UserEntity nguoiDung = null;

try {

String query = "SELECT \* FROM NguoiDung WHERE NguoiDung.HoTen = ?";

statement = con.prepareStatement(query);

statement.setString(1, name);

rs = statement.executeQuery();

if (rs.next()) {

nguoiDung = getFromResultSet(rs);

}

} finally {

}

return nguoiDung;

}

public UserEntity findByPhone(String phone) throws SQLException {

PreparedStatement statement = null;

ResultSet rs = null;

UserEntity nguoiDung = null;

try {

String query = "SELECT \* FROM NguoiDung WHERE NguoiDung.DienThoai = ?";

statement = con.prepareStatement(query);

statement.setString(1, phone);

rs = statement.executeQuery();

if (rs.next()) {

nguoiDung = getFromResultSet(rs);

}

} finally {

}

return nguoiDung;

}

public boolean checkLoginUserEntity(String username, String password) throws SQLException {

UserEntity ad = getUserEntity(username);

if (ad != null) {

if (ad.getPassword().equals(password) && ad.getRole().getName().equals("Chủ trọ")) {

// QLCHMainFrame mainViewUserEntity = new QLCHMainFrame();

// mainViewUserEntity.setVisible(true);

JOptionPane.showMessageDialog(null, "Đăng nhập với vai trò Ban Quản Lý thành công!");

return true;

}

}

JOptionPane.showMessageDialog(null, "Đăng nhập thất bại!");

return false;

}

public UserEntity get(Long id) throws SQLException {

Statement statement = con.createStatement();

String query = "SELECT \* FROM NguoiDung WHERE NguoiDung.id = " + id;

ResultSet rs = statement.executeQuery(query);

if (rs.next()) {

UserEntity NguoiDung = getFromResultSet(rs);

return NguoiDung;

}

return null;

}

public UserEntity get(int id) throws SQLException {

Statement statement = con.createStatement();

String query = "SELECT \* FROM NguoiDung WHERE NguoiDung.id = " + id;

ResultSet rs = statement.executeQuery(query);

if (rs.next()) {

UserEntity NguoiDung = getFromResultSet(rs);

return NguoiDung;

}

return null;

}

public boolean isEmailExists(String email) throws SQLException {

PreparedStatement statement = null;

ResultSet rs = null;

UserEntity nguoiDung = null;

String query = "SELECT COUNT(\*) FROM NguoiDung WHERE Email = ?";

statement = con.prepareStatement(query);

statement.setString(1, email);

rs = statement.executeQuery();

if (rs.next()) {

int count = rs.getInt(1);

return count > 0;

}

return false;

}

public boolean isPhoneExists(String phone) throws SQLException {

String query = "SELECT COUNT(\*) FROM NguoiDung WHERE DienThoai = ?";

PreparedStatement ps = con.prepareStatement(query);

ps.setString(1, phone);

ResultSet rs = ps.executeQuery();

if(rs.next()) {

int count = rs.getInt(1);

return count>0;

}

return false;

}

public int updateUser(UserEntity user) throws SQLException {

try {

String query = "UPDATE NguoiDung SET HoTen=?, DiaChi=?, Email=?, DienThoai=? WHERE Id = ?";

PreparedStatement ps = con.prepareStatement(query);

ps.setString(1, user.getHoTen());

ps.setString(2, user.getDiaChi());

ps.setString(3, user.getEmail());

ps.setString(4, user.getDienThoai());

ps.setLong(5, user.getId());

if (ps.executeUpdate() > 0) {

return 1;

}

} catch (Exception e) {

e.printStackTrace();

}

if (user == null) {

throw new SQLException("User rỗng");

}

return -1;

}

public int updateRole(Long id) {

try {

String query = "update NguoiDung set VaiTro='Tenants' where Id=?;";

PreparedStatement ps = con.prepareStatement(query);

ps.setLong(1, id);

if (ps.executeUpdate() > 0) {

return 1;

}

} catch (Exception e) {

e.printStackTrace();

}

return -1;

}

public String getPasswordByID(long id) throws SQLException {

String query = "select MatKhau from NguoiDung where ID = " + id;

PreparedStatement ps = con.prepareStatement(query);

ResultSet rs = ps.executeQuery();

if(rs.next()) {

return rs.getString(1);

}

return null;

}

public boolean updatePassword(Long id, String newPassword) throws SQLException {

// Cập nhật mật khẩu mới vào CSDL

String updateSql = "UPDATE NguoiDung SET MatKhau = ? WHERE Id = ?";

PreparedStatement updateStatement = con.prepareStatement(updateSql);

updateStatement.setString(1, newPassword);

updateStatement.setLong(2, id);

int rowsAffected = updateStatement.executeUpdate();

return rowsAffected > 0; // Trả về true nếu cập nhật thành công, ngược lại trả về false

}

public ArrayList<UserEntity> getAll() throws SQLException {

ArrayList<UserEntity> orders = new ArrayList<>();

Statement statement = con.createStatement();

String query = "SELECT \* FROM NguoiDung where trash !=0 ";

ResultSet rs = statement.executeQuery(query);

while (rs.next()) {

UserEntity order = getFromResultSet(rs);

orders.add(order);

}

return orders;

}

public ArrayList<UserEntity> searchByKey(String word) throws SQLException {

ArrayList<UserEntity> orders = new ArrayList<>();

Statement statement = con.createStatement();

String query = "SELECT \* FROM NguoiDung WHERE Id = '" + word + "'OR DienThoai = '" + word + "' AND status = 1 AND trash != 0";

ResultSet rs = statement.executeQuery(query);

while (rs.next()) {

UserEntity order = getFromResultSet(rs);

orders.add(order);

}

return orders;

}

public int SuaND(UserEntity t)throws SQLDataException, SQLException{

try {

LocalDateTime currentTime = LocalDateTime.now();

if (t == null) {

throw new SQLException("NguoiDungEntity rỗng");

}

String query = "UPDATE NguoiDung SET HoTen = ?, DienThoai= ?, Email= ?, DiaChi= ?, VaiTro=?, NgaySua=?, NguoiSua=? WHERE Id = ?;";

PreparedStatement stmt = con.prepareStatement(query);

stmt.setString(1, t.getHoTen());

stmt.setString(2, t.getDienThoai());

stmt.setString(3, t.getEmail());

stmt.setString(4, t.getDiaChi());

stmt.setString(5, t.getRole().toString());

stmt.setLong(8, t.getId());

stmt.setObject(6, currentTime);

stmt.setString(7, SessionManager.getSession().getUser().getHoTen());

if (stmt.executeUpdate() > 0) {

System.out.println("Success");

return 1;

}

} catch (Exception ex) {

ex.printStackTrace();

}

if (t == null) {

throw new SQLException("NguoiDung rỗng");

}

return -1;

}

public void deleteFromRecycle(Long id) throws SQLException {

if (id == null) {

throw new SQLException("Order rỗng");

}

String query = "UPDATE NguoiDung SET trash = ? WHERE Id = ?";

PreparedStatement stmt = con.prepareStatement(query);

stmt.setInt(1, 0);

stmt.setLong(2, id);

stmt.executeUpdate();

}

public ArrayList<UserEntity> getAllFromRecycle() throws SQLException {

ArrayList<UserEntity> orders = new ArrayList<>();

Statement statement = con.createStatement();

String query = "SELECT \* FROM NguoiDung where trash = 0 ";

ResultSet rs = statement.executeQuery(query);

while (rs.next()) {

UserEntity order = getFromResultSet(rs);

orders.add(order);

}

return orders;

}

public void RestoreFromRecycle(Long id) throws SQLException {

if (id == null) {

throw new SQLException("NguoiDung rỗng");

}

String query = "UPDATE NguoiDung SET trash = ? , Status = 1 WHERE Id = ?";

PreparedStatement stmt = con.prepareStatement(query);

stmt.setInt(1, 1);

stmt.setLong(2, id);

stmt.executeUpdate();

}

public void deleteById(Long id) throws SQLException {

PreparedStatement stmt = con.prepareStatement("DELETE FROM NguoiDung WHERE Id = ?");

stmt.setLong(1, id);

stmt.executeUpdate();

}

}

##### UserRole:

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

package TroViet.DAO;

/\*\*

\*

\* @author ADMIN

\*/

public enum UserRole {

Inactive("Inactive", "inactive", 3),

Admin("Admin", "admin", 2),

User("User", "user", 1),

Tenants("Tenants", "tenants", 0);

private String id, name;

private int priority;

UserRole(String id, String name, int priority) {

this.id = id;

this.name = name;

this.priority = priority;

}

public static UserRole getById(String id) {

for (UserRole e : values()) {

if (e.id.equals(id)) {

return e;

}

}

return User;

}

public static UserRole getByName(String name) {

for (UserRole e : values()) {

if (e.name.equals(name)) {

return e;

}

}

return User;

}

public String getId() {

return id;

}

public String getName() {

return name;

}

public int compare(UserRole other) {

return priority - other.priority;

}

}

#### Entity hợp đồng, HopDongDAO và HopDongEnum:

##### Entity hợp đồng:

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

package TroViet.Model;

import TroViet.DAO.HopDongEnum;

/\*\*

\*

\* @author ADMIN

\*/

public class HopDongEntity {

private long id;

private long idMaPhongTro;

private long idNguoiDung;

private String ngayBatDauThue;

private String ngayKetThucThue;

private float tienDatCoc;

private float giaPhong;

private float giaDien;

private float giaNuoc;

private float giaInternet;

private float giaRac;

private int trash;

private HopDongEnum status;

private String ngayTao;

private String nguoiTao;

private String ngaySua;

private String nguoiSua;

public HopDongEnum getStatus() {

return status;

}

public void setStatus(HopDongEnum status) {

this.status = status;

}

public String getNgayTao() {

return ngayTao;

}

public String getNguoiTao() {

return nguoiTao;

}

public String getNgaySua() {

return ngaySua;

}

public String getNguoiSua() {

return nguoiSua;

}

public void setNgayTao(String ngayTao) {

this.ngayTao = ngayTao;

}

public void setNguoiTao(String nguoiTao) {

this.nguoiTao = nguoiTao;

}

public void setNgaySua(String ngaySua) {

this.ngaySua = ngaySua;

}

public void setNguoiSua(String nguoiSua) {

this.nguoiSua = nguoiSua;

}

public long getId() {

return id;

}

public long getIdMaPhongTro() {

return idMaPhongTro;

}

public long getIdNguoiDung() {

return idNguoiDung;

}

public String getNgayBatDauThue() {

return ngayBatDauThue;

}

public String getNgayKetThucThue() {

return ngayKetThucThue;

}

public float getTienDatCoc() {

return tienDatCoc;

}

public float getGiaPhong() {

return giaPhong;

}

public float getGiaDien() {

return giaDien;

}

public float getGiaNuoc() {

return giaNuoc;

}

public float getGiaInternet() {

return giaInternet;

}

public float getGiaRac() {

return giaRac;

}

public int getTrash() {

return trash;

}

public void setId(long id) {

this.id = id;

}

public void setIdMaPhongTro(long idMaPhongTro) {

this.idMaPhongTro = idMaPhongTro;

}

public void setIdNguoiDung(long idNguoiDung) {

this.idNguoiDung = idNguoiDung;

}

public void setNgayBatDauThue(String ngayBatDauThue) {

this.ngayBatDauThue = ngayBatDauThue;

}

public void setNgayKetThucThue(String ngayKetThucThue) {

this.ngayKetThucThue = ngayKetThucThue;

}

public void setTienDatCoc(float tienDatCoc) {

this.tienDatCoc = tienDatCoc;

}

public void setGiaPhong(float giaPhong) {

this.giaPhong = giaPhong;

}

public void setGiaDien(float giaDien) {

this.giaDien = giaDien;

}

public void setGiaNuoc(float giaNuoc) {

this.giaNuoc = giaNuoc;

}

public void setGiaInternet(float giaInternet) {

this.giaInternet = giaInternet;

}

public void setGiaRac(float giaRac) {

this.giaRac = giaRac;

}

public void setTrash(int trash) {

this.trash = trash;

}

}

##### HopDongDAO:

package TroViet.DAO;

import TroViet.Connect.SQL\_Connection;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import javax.swing.JOptionPane;

import TroViet.Model.HopDongEntity;

public class HopDongDAO extends SQL\_Connection {

// String url = "jdbc:sqlserver://localhost:1433;databaseName=TroViet";

// String username = "sa";

// String password = "123";

//

// private Connection connection;

//

// public HopDongDAO() {

// try {

// connection = DriverManager.getConnection(url, username, password);

// } catch (SQLException e) {

// e.printStackTrace();

// }

// }

public void save(HopDongEntity t) throws SQLException {

if (t == null) {

throw new SQLException("HopDong rỗng");

}

String query = "{CALL InsertHopDong(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)}";

PreparedStatement stmt = con.prepareStatement(query);

stmt.setLong(1, t.getIdMaPhongTro());

stmt.setLong(2, t.getIdNguoiDung());

stmt.setString(3, t.getNgayBatDauThue());

stmt.setString(4, t.getNgayKetThucThue());

stmt.setFloat(5, t.getTienDatCoc());

stmt.setFloat(6, t.getGiaPhong());

stmt.setFloat(7, t.getGiaDien());

stmt.setFloat(8, t.getGiaNuoc());

stmt.setFloat(9, t.getGiaInternet());

stmt.setFloat(10, t.getGiaRac());

stmt.setString(11, t.getNgayTao());

stmt.setString(12, t.getNguoiTao());

stmt.setString(13, t.getNgayBatDauThue()); // Set null for NgaySua

stmt.setString(14, "thanh"); // Set null for NguoiSua

stmt.setInt(15, t.getTrash());

stmt.setString(16, t.getStatus().getId());

// stmt.setLong(1, 1);

// stmt.setLong(2, 1);

// stmt.setString(3, "2023-07-15");

// stmt.setString(4, "2023-07-15");

// stmt.setFloat(5, 1);

// stmt.setFloat(6, 1);

// stmt.setFloat(7, 1);

// stmt.setFloat(8, 1);

// stmt.setFloat(9, 1);

// stmt.setFloat(10, 1);

// stmt.setString(11, "2023-07-15");

// stmt.setString(12, "thanh123");

// stmt.setString(13, "2023-07-15"); // Set null for NgaySua

// stmt.setString(14, "thanh"); // Set null for NguoiSua

// stmt.setInt(15, 1);

// stmt.setString(16, t.getStatus().getId());

int row = stmt.executeUpdate();

}

public int updateOrderById(HopDongEntity t) throws SQLException {

try {

if (t == null) {

throw new SQLException("HopDongEntity rỗng");

}

// String query = "{CALL UpdateHopDong(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)}";

String query = "UPDATE HopDong SET IdMaPhongTro = ?, IdNguoiDung = ?,NgayBatDauThue = ?, NgayKetThucThue = ?, TienDatCoc = ?, GiaPhong = ?, "

+ "GiaDien = ?, GiaNuoc = ?, GiaInternet = ?, GiaRac = ?, NgaySua = ?, NguoiSua = ?, Trash = ?, Status = ? "

+ "WHERE Id = ?";

PreparedStatement stmt = con.prepareStatement(query);

stmt.setLong(1, t.getIdMaPhongTro());

stmt.setLong(2, t.getIdNguoiDung());

stmt.setString(3, t.getNgayBatDauThue());

stmt.setString(4, t.getNgayKetThucThue());

stmt.setFloat(5, t.getTienDatCoc());

stmt.setFloat(6, t.getGiaPhong());

stmt.setFloat(7, t.getGiaDien());

stmt.setFloat(8, t.getGiaNuoc());

stmt.setFloat(9, t.getGiaInternet());

stmt.setFloat(10, t.getGiaRac());

stmt.setString(11, t.getNgayBatDauThue()); // Set null for NgaySua

stmt.setString(12, t.getNguoiSua()); // Set null for NguoiSua

stmt.setInt(13, t.getTrash());

stmt.setString(14, t.getStatus().getId());

stmt.setLong(15, t.getId());

if (stmt.executeUpdate() > 0) {

System.out.println("Success");

return 1;

}

} catch (Exception ex) {

ex.printStackTrace();

}

if (t == null) {

throw new SQLException("HopDong rỗng");

}

return -1;

}

public void deleteFromRecycle(Long id) throws SQLException {

if (id == null) {

throw new SQLException("Order rỗng");

}

String query = "UPDATE HopDong SET trash = ? and status = 0 WHERE Id = ?";

PreparedStatement stmt = con.prepareStatement(query);

stmt.setInt(1, 0);

stmt.setLong(2, id);

stmt.executeUpdate();

}

public static HopDongEntity getFromResultSet(ResultSet rs) throws SQLException {

HopDongEntity o = new HopDongEntity();

o.setId(rs.getLong("Id"));

o.setIdMaPhongTro(rs.getLong("IdMaPhongTro"));

o.setIdNguoiDung(rs.getLong("IdNguoiDung"));

o.setNgayBatDauThue(rs.getString("NgayBatDauThue"));

o.setNgayKetThucThue(rs.getString("NgayKetThucThue"));

o.setTienDatCoc(rs.getFloat("TienDatCoc"));

o.setGiaPhong(rs.getFloat("GiaPhong"));

o.setGiaDien(rs.getFloat("GiaDien"));

o.setGiaNuoc(rs.getFloat("GiaNuoc"));

o.setGiaInternet(rs.getFloat("GiaInternet"));

o.setGiaRac(rs.getFloat("GiaRac"));

o.setNgayTao(rs.getString("NgayTao"));

o.setNguoiTao(rs.getString("NguoiTao"));

o.setNgaySua(rs.getString("NgaySua"));

o.setNguoiSua(rs.getString("NguoiSua"));

o.setTrash(rs.getInt("Trash"));

o.setStatus(HopDongEnum.getById(rs.getString("status")));

return o;

}

public HopDongEntity get(Long id) throws SQLException {

Statement statement = con.createStatement();

String query = "SELECT \* FROM HopDong WHERE Id = " + id;

ResultSet rs = statement.executeQuery(query);

if (rs.next()) {

HopDongEntity hopDong = getFromResultSet(rs);

// order.setMonth(monthDao.get(order.getMonthId()));

// order.setService(serviceDao.get(order.getServiceId()));

// order.setApartment(apartmentDao.get(order.getApartmentId()));

return hopDong;

}

return null;

}

public boolean isHopDongExists(Long id) throws SQLException {

Statement statement = con.createStatement();

String query = "SELECT COUNT(\*) FROM HopDong WHERE Id = " + id;

ResultSet rs = statement.executeQuery(query);

if (rs.next()) {

int count = rs.getInt(1);

return count > 0;

}

return false;

}

public boolean isIdMaPhongTroExists(Long id) throws SQLException {

Statement statement = con.createStatement();

String query = "SELECT IdMaPhongTro FROM HopDong WHERE Trash = 1 and Status = 1 and IdMaPhongTro = " + id;

ResultSet rs = statement.executeQuery(query);

if (rs.next()) {

int count = rs.getInt(1);

return count > 0;

}

return false;

}

public boolean isIdNguoiDungTroExists(Long id) throws SQLException {

Statement statement = con.createStatement();

String query = "SELECT IdNguoiDung FROM HopDong WHERE Trash = 1 and Status = 1 and IdNguoiDung = " + id;

ResultSet rs = statement.executeQuery(query);

if (rs.next()) {

int count = rs.getInt(1);

return count > 0;

}

return false;

}

public ArrayList<HopDongEntity> getAll() throws SQLException {

ArrayList<HopDongEntity> orders = new ArrayList<>();

Statement statement = con.createStatement();

String query = "SELECT \* FROM HopDong where trash !=0 and status = 1 ";

ResultSet rs = statement.executeQuery(query);

while (rs.next()) {

HopDongEntity order = getFromResultSet(rs);

orders.add(order);

}

return orders;

}

public ArrayList<HopDongEntity> getAllFromRecycle() throws SQLException {

ArrayList<HopDongEntity> orders = new ArrayList<>();

Statement statement = con.createStatement();

String query = "SELECT \* FROM HopDong where trash = 0 ";

ResultSet rs = statement.executeQuery(query);

while (rs.next()) {

HopDongEntity order = getFromResultSet(rs);

orders.add(order);

}

return orders;

}

public void ngayKetThucThue() throws SQLException {

Statement statement = con.createStatement();

String query = "UPDATE HopDong SET Status = 1 WHERE GETDATE() >= NgayKetThucThue";

int rowsUpdated = statement.executeUpdate(query);

statement.close();

}

public ArrayList<HopDongEntity> searchByKey(String word) throws SQLException {

ArrayList<HopDongEntity> orders = new ArrayList<>();

Statement statement = con.createStatement();

String query = "SELECT \* FROM HopDong WHERE Id = '" + word + "' AND status = 1 AND trash != 0";

ResultSet rs = statement.executeQuery(query);

while (rs.next()) {

HopDongEntity order = getFromResultSet(rs);

orders.add(order);

}

return orders;

}

public void deleteById(Long id) throws SQLException {

PreparedStatement stmt = con.prepareStatement("DELETE FROM HopDong WHERE Id = ?");

stmt.setLong(1, id);

stmt.executeUpdate();

}

public void RestoreFromRecycle(Long id) throws SQLException {

if (id == null) {

throw new SQLException("HopDong rỗng");

}

String query = "UPDATE HopDong SET trash = ? , Status = 1 WHERE Id = ?";

PreparedStatement stmt = con.prepareStatement(query);

stmt.setInt(1, 1);

stmt.setLong(2, id);

stmt.executeUpdate();

}

public ArrayList<HopDongEntity> getHopDongByEmailUser(String email) throws SQLException {

ArrayList<HopDongEntity> hopDong = new ArrayList<>();

// String query = "SELECT TOP 1 \* " +

// "FROM HopDong " +

// "JOIN NguoiDung ON HopDong.IdNguoiDung = NguoiDung.Id " +

// "WHERE NguoiDung.Email = ? AND HopDong.Trash = 1 " +

// "ORDER BY HopDong.NgayTao DESC;";

String query = "SELECT TOP 1 \* FROM HopDong WHERE HopDong.IdMaPhongTro IN (SELECT PhongTro.Id FROM PhongTro join NguoiDung on PhongTro.IdNguoiDung = NguoiDung.Id WHERE NguoiDung.Email = ?) AND HopDong.Trash = 1 AND HopDong.Status = 1 ORDER BY HopDong.NgayTao DESC";

try (PreparedStatement statement = con.prepareStatement(query)) {

statement.setString(1, email);

ResultSet rs = statement.executeQuery();

if (rs.next()) {

HopDongEntity hopDongs = getFromResultSet(rs);

hopDong.add(hopDongs);

}

}

return hopDong;

}

}

##### HopDongEnum:

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

package TroViet.DAO;

/\*\*

\*

\* @author ADMIN

\*/

public enum HopDongEnum {

Time\_expired("0", "Hết hạn"),

Using("1", "Đang sử dụng"),

CANCEL("2", "Bị hủy");

private String id, name;

HopDongEnum(String id, String name) {

this.id = id;

this.name = name;

}

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public static HopDongEnum getById(String id) {

for (HopDongEnum e : values()) {

if (e.id.equals(id)) {

return e;

}

}

return Time\_expired;

}

public static HopDongEnum getByName(String name) {

for (HopDongEnum e : values()) {

if (e.name.equals(name)) {

return e;

}

}

return Time\_expired;

}

}

#### Entity hóa đơn, HoaDonDAO và HoaDonEnum:

##### Entity hóa đơn:

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

package TroViet.Model;

import TroViet.DAO.HoaDonEnum;

import java.util.Date;

/\*\*

\*

\* @author WINDOWS

\*/

public class HoaDon {

long id, idHopDong;

String ngayBD, ngayKT;

float soDienCu, soDienMoi, soNuocCu, soNuocMoi, khauTru, no, tongCong;

String ngayTao;

String nguoiTao;

String ngaySua;

String nguoiSua;

int trash;

HoaDonEnum status;

public HoaDon() {

}

public HoaDon(long id, long idHopDong, String ngayBD, String ngayKT, float soDienCu, float soDienMoi, float soNuocCu, float soNuocMoi, float khauTru, float no, float tongCong, String ngayTao, String nguoiTao, String ngaySua, String nguoiSua, int trash, HoaDonEnum status) {

this.id = id;

this.idHopDong = idHopDong;

this.ngayBD = ngayBD;

this.ngayKT = ngayKT;

this.soDienCu = soDienCu;

this.soDienMoi = soDienMoi;

this.soNuocCu = soNuocCu;

this.soNuocMoi = soNuocMoi;

this.khauTru = khauTru;

this.no = no;

this.tongCong = tongCong;

this.ngayTao = ngayTao;

this.nguoiTao = nguoiTao;

this.ngaySua = ngaySua;

this.nguoiSua = nguoiSua;

this.trash = trash;

this.status = status;

}

public long getId() {

return id;

}

public void setId(long id) {

this.id = id;

}

public long getIdHopDong() {

return idHopDong;

}

public void setIdHopDong(long idHopDong) {

this.idHopDong = idHopDong;

}

public String getNgayBD() {

return ngayBD;

}

public void setNgayBD(String ngayBD) {

this.ngayBD = ngayBD;

}

public String getNgayKT() {

return ngayKT;

}

public void setNgayKT(String ngayKT) {

this.ngayKT = ngayKT;

}

public float getSoDienCu() {

return soDienCu;

}

public void setSoDienCu(float soDienCu) {

this.soDienCu = soDienCu;

}

public float getSoDienMoi() {

return soDienMoi;

}

public void setSoDienMoi(float soDienMoi) {

this.soDienMoi = soDienMoi;

}

public float getSoNuocCu() {

return soNuocCu;

}

public void setSoNuocCu(float soNuocCu) {

this.soNuocCu = soNuocCu;

}

public float getSoNuocMoi() {

return soNuocMoi;

}

public void setSoNuocMoi(float soNuocMoi) {

this.soNuocMoi = soNuocMoi;

}

public float getKhauTru() {

return khauTru;

}

public void setKhauTru(float khauTru) {

this.khauTru = khauTru;

}

public float getNo() {

return no;

}

public void setNo(float no) {

this.no = no;

}

public float getTongCong() {

return tongCong;

}

public void setTongCong(float tongCong) {

this.tongCong = tongCong;

}

public String getNgayTao() {

return ngayTao;

}

public void setNgayTao(String ngayTao) {

this.ngayTao = ngayTao;

}

public String getNguoiTao() {

return nguoiTao;

}

public void setNguoiTao(String nguoiTao) {

this.nguoiTao = nguoiTao;

}

public String getNgaySua() {

return ngaySua;

}

public void setNgaySua(String ngaySua) {

this.ngaySua = ngaySua;

}

public String getNguoiSua() {

return nguoiSua;

}

public void setNguoiSua(String nguoiSua) {

this.nguoiSua = nguoiSua;

}

public int getTrash() {

return trash;

}

public void setTrash(int trash) {

this.trash = trash;

}

public HoaDonEnum getStatus() {

return status;

}

public void setStatus(HoaDonEnum status) {

this.status = status;

}

}

##### HoaDonDAO:

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

package TroViet.DAO;

import TroViet.Connect.SQL\_Connection;

import TroViet.Model.HoaDon;

import java.beans.Statement;

import java.sql.Date;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

/\*\*

\*

\* @author WINDOWS

\*/

public class HoaDonDAO extends SQL\_Connection{

public static HoaDon getFromResultSet(ResultSet rs) throws SQLException {

HoaDon hdon = new HoaDon();

hdon.setId(rs.getLong("Id"));

hdon.setIdHopDong(rs.getLong("IdHopDong"));

hdon.setNgayBD(rs.getString("NgayBatDau"));

hdon.setNgayKT(rs.getString("NgayKetThuc"));

hdon.setSoDienCu(rs.getFloat("SoDienCu"));

hdon.setSoDienMoi(rs.getFloat("SoDienMoi"));

hdon.setSoNuocCu(rs.getFloat("SoNuocCu"));

hdon.setSoNuocMoi(rs.getFloat("SoNuocMoi"));

hdon.setKhauTru(rs.getFloat("KhauTru"));

hdon.setNo(rs.getFloat("No"));

hdon.setTongCong(rs.getFloat("TongCong"));

hdon.setNgayTao(rs.getString("NgayTao"));

hdon.setNguoiTao(rs.getString("NguoiTao"));

hdon.setNgaySua(rs.getString("NgaySua"));

hdon.setNguoiSua(rs.getString("NguoiSua"));

hdon.setTrash(rs.getInt("Trash"));

hdon.setStatus(HoaDonEnum.getById((rs.getString("Status"))));

return hdon;

}

public ArrayList<HoaDon> searchByID(long idHopDong) throws SQLException {

ArrayList<HoaDon> listHDon = new ArrayList<>();

String query = "select \* from HoaDon where IdHopDong = ? and trash!=0;";

PreparedStatement ps = con.prepareStatement(query);

ps.setLong(1, idHopDong);

ResultSet rs = ps.executeQuery();

while (rs.next()) {

HoaDon hdon = getFromResultSet(rs);

listHDon.add(hdon);

}

return listHDon;

}

public void add(HoaDon hdon) throws SQLException{

String query = "insert into HoaDon (IdHopDong, NgayBatDau, NgayKetThuc, SoDienCu, SoDienMoi, SoNuocCu, SoNuocMoi, KhauTru, TienNo, TongCong, NgayTao, NguoiTao, Status)\n"

+ "values (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?);";

PreparedStatement ps = con.prepareStatement(query);

ps.setLong(1, hdon.getIdHopDong());

ps.setString(2, hdon.getNgayBD());

ps.setString(3, hdon.getNgayKT());

ps.setFloat(4, hdon.getSoDienCu());

ps.setFloat(5, hdon.getSoDienMoi());

ps.setFloat(6, hdon.getSoNuocCu());

ps.setFloat(7, hdon.getSoNuocMoi());

ps.setFloat(8, hdon.getKhauTru());

ps.setFloat(9, hdon.getNo());

ps.setFloat(10, hdon.getTongCong());

ps.setString(11, hdon.getNgayTao());

ps.setString(12, hdon.getNguoiTao());

ps.setString(13, hdon.getStatus().getId());

int row = ps.executeUpdate();

}

public int updateByID(HoaDon hdon) {

try {

String query = "update HoaDon \n"

+ "set IdHopDong=?, NgayBatDau=?, NgayKetThuc=?, SoDienCu=?, SoDienMoi=?, SoNuocCu=?, SoNuocMoi=?, KhauTru=?, TienNo=?, TongCong=?, NgaySua=?, NguoiSua=?, Trash=?, Status=?\n"

+ "where Id=?;";

PreparedStatement ps = con.prepareStatement(query);

ps.setLong(1, hdon.getIdHopDong());

ps.setString(2, hdon.getNgayBD());

ps.setString(3, hdon.getNgayKT());

ps.setFloat(4, hdon.getSoDienCu());

ps.setFloat(5, hdon.getSoDienMoi());

ps.setFloat(6, hdon.getSoNuocCu());

ps.setFloat(7, hdon.getSoNuocMoi());

ps.setFloat(8, hdon.getKhauTru());

ps.setFloat(9, hdon.getNo());

ps.setFloat(10, hdon.getTongCong());

ps.setString(11, hdon.getNgaySua());

ps.setString(12, hdon.getNguoiSua());

ps.setInt(13, hdon.getTrash());

ps.setString(14, hdon.getStatus().getId());

ps.setLong(15, hdon.getId());

if (ps.executeUpdate()>0) {

System.out.println("Success");

return 1;

}

} catch (Exception e) {

e.printStackTrace();

}

return -1;

}

public void deleteFromRecycle(Long id) throws SQLException {

if (id == null) {

throw new SQLException("ID trống");

}

String query = "UPDATE HoaDon SET Trash = ? WHERE Id = ?";

PreparedStatement stmt = con.prepareStatement(query);

stmt.setInt(1, 0);

stmt.setLong(2, id);

stmt.executeUpdate();

}

public void RestoreFromRecycle(Long id) throws SQLException {

if (id == null) {

throw new SQLException("ID rỗng");

}

String query = "UPDATE HoaDon SET trash = ? WHERE Id = ?";

PreparedStatement stmt = con.prepareStatement(query);

stmt.setInt(1, 1);

stmt.setLong(2, id);

stmt.executeUpdate();

}

public void deleteById(Long id) throws SQLException {

PreparedStatement stmt = con.prepareStatement("DELETE FROM HoaDon WHERE Id = ?");

stmt.setLong(1, id);

stmt.executeUpdate();

}

}

##### HoaDonEnum:

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Enum.java to edit this template

\*/

package TroViet.DAO;

/\*\*

\*

\* @author WINDOWS

\*/

public enum HoaDonEnum {

Paid("1", "Đã thanh toán"),

Unpaid("0", "Chưa thanh toán");

private String id, name;

HoaDonEnum(String id, String name) {

this.id = id;

this.name = name;

}

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public static HoaDonEnum getById(String id) {

for (HoaDonEnum e : values()) {

if (e.id.equals(id)) {

return e;

}

}

return Unpaid;

}

public static HoaDonEnum getByName(String name) {

for (HoaDonEnum e : values()) {

if (e.name.equals(name)) {

return e;

}

}

return Unpaid;

}

}

#### Entity đánh giá và DanhGiaDAO:

##### Entity đánh giá:

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

package TroViet.Model;

/\*\*

\*

\* @author WINDOWS

\*/

public class DanhGia {

private Long id, idPhongTro, idNguoiDung;

private int DanhGia;

private String noiDung;

private String ngayTao;

private String nguoiTao;

private String ngaySua;

private String nguoiSua;

private int trash;

private int status;

public DanhGia() {

}

public DanhGia(Long id, Long idPhongTro, Long idNguoiDung, int DanhGia, String noiDung, String ngayTao, String nguoiTao, String ngaySua, String nguoiSua, int trash, int status) {

this.id = id;

this.idPhongTro = idPhongTro;

this.idNguoiDung = idNguoiDung;

this.DanhGia = DanhGia;

this.noiDung = noiDung;

this.ngayTao = ngayTao;

this.nguoiTao = nguoiTao;

this.ngaySua = ngaySua;

this.nguoiSua = nguoiSua;

this.trash = trash;

this.status = status;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public Long getIdPhongTro() {

return idPhongTro;

}

public void setIdPhongTro(Long idPhongTro) {

this.idPhongTro = idPhongTro;

}

public Long getIdNguoiDung() {

return idNguoiDung;

}

public void setIdNguoiDung(Long idNguoiDung) {

this.idNguoiDung = idNguoiDung;

}

public int getDanhGia() {

return DanhGia;

}

public void setDanhGia(int DanhGia) {

this.DanhGia = DanhGia;

}

public String getNoiDung() {

return noiDung;

}

public void setNoiDung(String noiDung) {

this.noiDung = noiDung;

}

public String getNgayTao() {

return ngayTao;

}

public void setNgayTao(String ngayTao) {

this.ngayTao = ngayTao;

}

public String getNguoiTao() {

return nguoiTao;

}

public void setNguoiTao(String nguoiTao) {

this.nguoiTao = nguoiTao;

}

public String getNgaySua() {

return ngaySua;

}

public void setNgaySua(String ngaySua) {

this.ngaySua = ngaySua;

}

public String getNguoiSua() {

return nguoiSua;

}

public void setNguoiSua(String nguoiSua) {

this.nguoiSua = nguoiSua;

}

public int getTrash() {

return trash;

}

public void setTrash(int trash) {

this.trash = trash;

}

public int getStatus() {

return status;

}

public void setStatus(int status) {

this.status = status;

}

}

##### DanhGiaDAO:

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

package TroViet.DAO;

import TroViet.Connect.SQL\_Connection;

import TroViet.Model.DanhGia;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

/\*\*

\*

\* @author WINDOWS

\*/

public class DanhGiaDAO extends SQL\_Connection{

public ArrayList<DanhGia> getAll() throws SQLException {

ArrayList<DanhGia> listDG = new ArrayList<>();

String query = "select \* from DanhGia";

PreparedStatement ps = con.prepareStatement(query);

ResultSet rs = ps.executeQuery();

while (rs.next()) {

DanhGia dg = new DanhGia();

dg.setId(rs.getLong(1));

dg.setIdPhongTro(rs.getLong(2));

dg.setIdNguoiDung(rs.getLong(3));

dg.setDanhGia(rs.getInt(4));

dg.setNoiDung(rs.getString(5));

dg.setNguoiTao(rs.getString(7));

listDG.add(dg);

}

return listDG;

}

public void add(DanhGia dgia) throws SQLException {

String query = "insert into DanhGia (IdMaPhongTro, IdNguoiDung, DanhGia, NoiDung, NgayTao, NguoiTao) values (?,?,?,?,?,?);";

PreparedStatement ps = con.prepareStatement(query);

ps.setLong(1,dgia.getIdPhongTro());

ps.setLong(2, dgia.getIdNguoiDung());

ps.setInt(3, dgia.getDanhGia());

ps.setString(4, dgia.getNoiDung());

ps.setString(5, dgia.getNgayTao());

ps.setString(6, dgia.getNguoiTao());

int row = ps.executeUpdate();

}

public static DanhGia getFromResultSet(ResultSet rs) throws SQLException {

DanhGia dg = new DanhGia();

dg.setId(rs.getLong("Id"));

dg.setIdPhongTro(rs.getLong("IdMaPhongTro"));

dg.setIdNguoiDung(rs.getLong("IdNguoiDung"));

dg.setDanhGia(rs.getInt("DanhGia"));

dg.setNoiDung(rs.getString("NoiDung"));

dg.setNgayTao(rs.getString("NgayTao"));

dg.setNguoiTao(rs.getString("NguoiTao"));

return dg;

}

public DanhGia get(Long id) throws SQLException {

Statement statement = con.createStatement();

String query = "SELECT \* FROM DanhGia WHERE Id = " + id;

ResultSet rs = statement.executeQuery(query);

if (rs.next()) {

DanhGia danhGia = getFromResultSet(rs);

return danhGia;

}

return null;

}

}

#### Entity session, SessionDAO và SessionManager:

##### Entity session:

package TroViet.Model;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Timestamp;

public class SessionEntity {

private Long idUser;

private int id;

private Timestamp startTime, endTime;

private UserEntity user;

private String message;

public SessionEntity() {

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public Timestamp getStartTime() {

return startTime;

}

public Timestamp getEndTime() {

return endTime;

}

public UserEntity getUser() {

return user;

}

public void setStartTime(Timestamp startTime) {

this.startTime = startTime;

}

public void setEndTime(Timestamp endTime) {

this.endTime = endTime;

}

public void setUser(UserEntity user) {

this.user = user;

if (user != null) {

this.idUser = user.getId();

}

}

public Long getIdUser() {

return idUser;

}

public void setIdUser(Long idUser) {

this.idUser = idUser;

}

public String getMessage() {

return message;

}

public void setMessage(String message) {

this.message = message;

}

public static SessionEntity getFromResultSet(ResultSet rs) throws SQLException {

SessionEntity s = new SessionEntity();

s.setId(rs.getInt("id"));

s.setIdUser(rs.getLong("idUser"));

s.setMessage(rs.getNString("message"));

s.setStartTime(rs.getTimestamp("startTime"));

s.setEndTime(rs.getTimestamp("endTime"));

return s;

}

@Override

public String toString() {

return "Session{" + "id=" + id + ", idUser=" + idUser + ", startTime=" + startTime + ", endTime=" + endTime + ", message=" + message + '}';

}

}

##### SessionDAO:

package TroViet.DAO;

import TroViet.Connect.SQL\_Connection;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.sql.Timestamp;

import java.util.ArrayList;

import TroViet.Model.SessionEntity;

public class SessionDAO extends SQL\_Connection{

UserDAO userDao = new UserDAO();

// String url = "jdbc:sqlserver://localhost:1433;databaseName=TroViet";

// String username = "sa";

// String password = "123";

//

// private Connection connection;

//

// public SessionDAO() {

// try {

// connection = DriverManager.getConnection(url, username, password);

// } catch (SQLException e) {

// e.printStackTrace();

// }

// }

public static SessionEntity session;

public static SessionEntity getSession() {

return session;

}

public static SessionEntity getFromResultSet(ResultSet rs) throws SQLException {

SessionEntity s = new SessionEntity();

s.setId(rs.getInt("id"));

s.setIdUser(rs.getLong("IdNguoiDung"));

s.setMessage(rs.getNString("NoiDung"));

s.setStartTime(rs.getTimestamp("ThoiGianDangNhap"));

s.setEndTime(rs.getTimestamp("ThoiGianDangXuat"));

return s;

}

public ArrayList<SessionEntity> getAll() throws SQLException {

ArrayList<SessionEntity> sessions = new ArrayList<>();

Statement statement = con.createStatement();

String query = "SELECT \* FROM session ORDER BY session.ThoiGianDangNhap DESC";

ResultSet rs = statement.executeQuery(query);

while (rs.next()) {

SessionEntity session = getFromResultSet(rs);

session.setUser(userDao.get(session.getIdUser()));

sessions.add(session);

}

return sessions;

}

public SessionEntity get(Long id) throws SQLException {

Statement statement = con.createStatement();

String query = "SELECT \* FROM session WHERE id = " + id;

ResultSet rs = statement.executeQuery(query);

if (rs.next()) {

SessionEntity session = getFromResultSet(rs);

session.setUser(userDao.get(session.getIdUser()));

return session;

}

return null;

}

public ArrayList<SessionEntity> getSessionUser(Long idUser) throws SQLException {

ArrayList<SessionEntity> listSession = new ArrayList<>();

String query = "SELECT \* FROM session WHERE IdNguoiDung = ? ORDER BY session.ThoiGianDangNhap DESC";

PreparedStatement ps = con.prepareStatement(query);

ps.setLong(1, idUser);

ResultSet rs = ps.executeQuery();

while (rs.next()) {

SessionEntity session = getFromResultSet(rs);

session.setUser(userDao.get(session.getIdUser()));

listSession.add(session);

}

return listSession;

}

public ArrayList<SessionEntity> getSession(Long id) throws SQLException {

ArrayList<SessionEntity> sessions = new ArrayList<>();

Statement statement = con.createStatement();

String query = "SELECT \* FROM session WHERE IdNguoiDung = " + id + " ORDER BY session.ThoiGianDangNhap DESC";

ResultSet rs = statement.executeQuery(query);

while (rs.next()) {

SessionEntity session = getFromResultSet(rs);

session.setUser(userDao.get(session.getIdUser()));

sessions.add(session);

}

return sessions;

}

public void save(SessionEntity t) throws SQLException {

if (t == null) {

throw new SQLException("Shipment rỗng");

}

String query = "INSERT INTO session (IdNguoiDung, ThoiGianDangNhap, ThoiGianDangXuat , NoiDung) VALUES (?, ?, ?, ?)";

PreparedStatement stmt = con.prepareStatement(query);

stmt.setLong(1, t.getIdUser());

stmt.setTimestamp(2, t.getStartTime());

stmt.setTimestamp(3, t.getEndTime());

stmt.setNString(4, t.getMessage());

int row = stmt.executeUpdate();

}

public void update(SessionEntity t) throws SQLException {

if (t == null) {

throw new SQLException("Entity rỗng");

}

String query = "UPDATE session SET ThoiGianDangNhap = ?, ThoiGianDangXuat = ?, NoiDung = ? WHERE session.id = ?";

PreparedStatement stmt = con.prepareStatement(query);

stmt.setTimestamp(1, t.getStartTime());

stmt.setTimestamp(2, t.getEndTime());

stmt.setNString(3, t.getMessage());

stmt.setInt(4, t.getId());

int row = stmt.executeUpdate();

}

public void delete(SessionEntity t) throws SQLException {

throw new UnsupportedOperationException("Not supported yet.");

}

public void deleteById(Long id) throws SQLException {

throw new UnsupportedOperationException("Not supported yet.");

}

public SessionEntity getLast(Long IdNguoiDung) throws SQLException {

Statement statement = con.createStatement();

String query = "SELECT top 1\* FROM session WHERE IdNguoiDung = " + IdNguoiDung

+ " ORDER BY id DESC ";

ResultSet rs = statement.executeQuery(query);

if (rs.next()) {

SessionEntity session = getFromResultSet(rs);

session.setUser(userDao.get(session.getIdUser()));

return session;

}

return null;

}

public ArrayList<SessionEntity> getAll(Timestamp start, Timestamp end) throws SQLException {

ArrayList<SessionEntity> sessions = new ArrayList<>();

String query = "SELECT \* FROM session WHERE NoiDung = ? AND DATE(ThoiGianDangNhap) >= DATE(?) AND DATE(ThoiGianDangNhap) <= DATE(?) ORDER BY session.ThoiGianDangNhap DESC";

PreparedStatement statement = con.prepareStatement(query);

statement.setNString(1, "logout");

statement.setTimestamp(2, start);

statement.setTimestamp(3, end);

ResultSet rs = statement.executeQuery();

while (rs.next()) {

SessionEntity session = getFromResultSet(rs);

session.setUser(userDao.get(session.getIdUser()));

sessions.add(session);

}

return sessions;

}

}

##### SessionManager:

package TroViet.Utils;

import TroViet.DAO.SessionDAO;

import java.sql.SQLException;

import java.sql.Timestamp;

import TroViet.Model.SessionEntity;

import TroViet.Model.UserEntity;

import javax.swing.JOptionPane;

public class SessionManager {

public static SessionEntity session;

static SessionDAO sessionDao = new SessionDAO();

public SessionManager() {

}

public static SessionEntity getSession() {

return session;

}

public static void setSession(SessionEntity session) {

SessionManager.session = session;

}

public static void create(UserEntity e) throws SQLException {

if (e == null) {

throw new SQLException("Nguoi dung khong hop le!");

}

SessionEntity ss = new SessionEntity();

ss.setUser(e);

ss.setMessage("login");

ss.setStartTime(new Timestamp(System.currentTimeMillis()));

sessionDao.save(ss);

SessionEntity sss = sessionDao.getLast(e.getId());

setSession(sss);

}

public static void update() throws SQLException {

if (session == null) {

throw new SQLException("Ban chua dang nhap!");

}

session.setMessage("logout");

session.setEndTime(new Timestamp(System.currentTimeMillis()));

sessionDao.update(session);

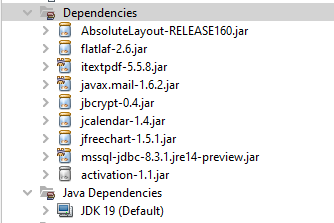
setSession(null);

}

}

## Thư viện tiện ích:

Thư viện tiện ích sử dụng trong Swing



## Lập trình nghiệp vụ:

### TrangChuChung:

|  |  |  |
| --- | --- | --- |
| **Thứ tự** | **Phương thức** | **Mô tả** |
| **1.** | ManHinhChao chao = new ManHinhChao(this, true);    chao.setVisible(true); | Khi khởi động và vào trang chủ chung hiển thị cửa sổ loading bằng Màn hình chào. |
| **2.** | setLocationRelativeTo(null); | Hiển thị màn hình ở giữa |
| **3.** | initTable(); | Đặt table Phòng trống = default sau đó hiển thị màn hình với bảng danh sách phòng trống theo đúng tiêu đề đã lưu với String[] columns. |
| **4.** | fillToPhongTrong(); | danh sách phòng trống |

### TrangChuPanel:

| **Thứ tự** | **Phương thức** | **Mô tả** |
| --- | --- | --- |
| **1.** | initTable(); | Hiển thị các form tbl: tổng doanh thu,tiền đặt cọc,tiền chi,tiền đặt cọc,số khách trọ đang thuê,số hợp đồng sắp hết hạn,hóa đơn chưa thanh toán. Đặt table Trang chủ = default sau đó hiển thị màn hình với bảng danh sách trang chủ theo đúng tiêu đề đã lưu với String[] columns. |
| **2.** | fillTableTrangChu(); | Lấy và hiển thị dữ liệu từ database lên bảng Trang Chủ |

### NguoiDungPanel:

|  |  |  |
| --- | --- | --- |
| **Thứ tự** | **Phương thức** | **Mô tả** |
| **1.** | initComponents(); |  |
| **2.** | initTable(); | Hiển thị màn hình table người dùng với các tiêu đề đã lưu với String[] columns. |
| **3.** | fillToTableUser(); | Lấy dữ liệu và hiển thị lên table người dùng đúng dữ liệu với tiêu đề. |

### HoatDongNguoiDungPanel:

|  |  |  |
| --- | --- | --- |
| **Thứ tự** | **Phương thức** | **Mô tả** |
| **1.** | initComponents(); |  |
| **2.** | initTable(); | Hiển thị cửa sổ table hoạt động người dùng với các tiêu đề đã lưu với String[] columns. |
| **3.** | fillTableSession(); | Lấy dữ liệu từ database và hiển thị lên bảng Hoạt động người dùng đúng dữ liệu của tiêu đề. |

### KhuVucLoaiPhongPanel:

|  |  |  |
| --- | --- | --- |
| **Thứ tự** | **Phương thức** | **Mô tả** |
| 1. | initKhuVuc(); | Dùng để show tiêu đề của bảng khu vực |
| 2. | initLoaiPhong(); | Dùng để show tiêu đề của bảng loại phòng |
| 3. | fillToKhuVuc(); | Lấy dữ liệu từ database và hiển thị lên bảng Khu Vực đúng dữ liệu của tiêu đề. |
| 4. | fillToLoaiPhong(); | Lấy dữ liệu từ database và hiển thị lên bảng Loại Phòng đúng dữ liệu tiêu đề |
| 5. | getIDKV(); | Dùng để lấy ID hiển thị chi tiết khu vực cần xem |
| 6. | getIDLP(); | Dùng để lấy ID hiển thị chi tiết loại phòng cần xem |
| 7. | tblKhuVucMouseClicked | Sự kiện nhấp vào hiển thị chi tiết Khu Vực |
| 8. | tblLoaiPhongMouseClicked | Sự kiện nhấp vào hiển thị chi tiết Loại Phòng |

### PhongTroPanel:

|  |  |  |
| --- | --- | --- |
| **Thứ tự** | **Phương thức** | **Mô tả** |
| 1. | initTable(); | Dùng để show tiêu đề của bảng hợp đồng |
| 2. | LoadDataTable(); | Lấy dữ liệu từ database và hiển thị lên bảng phòng trọ  đúng dữ liệu của tiêu đề. |
| 3. | getSelectedIds(); | Dùng để lấy ID hiển thị chi tiết phòng trọ admin cần xem |
| 4. | tblPhongTro | Sự kiện nhấp vào hiển thị chi tiết phòng trọ |

### HopDongPanel:

|  |  |  |
| --- | --- | --- |
| **Thứ tự** | **Phương thức** | **Mô tả** |
| 1. | initTable(); | Dùng để show tiêu đề của bảng hợp đồng |
| 2. | fillToTableOrder(); | Lấy dữ liệu từ database và hiển thị lên bảng hợp đồng  đúng dữ liệu của tiêu đề. |
| 3. | getSelectedID(); | Dùng để lấy ID hiển thị chi tiết hợp đồng admin cần xem |
| 4. | tblHopDongMouseClicked | Sự kiện nhấp vào hiển thị chi tiết Hợp Đồng |

### HoaDonPanel:

|  |  |  |
| --- | --- | --- |
| **Thứ tự** | **Phương thức** | **Mô tả** |
| 1. | initTable(); | Dùng để show tiêu đề của bảng hóa đơn |
| 2. | fillHoaDon(); | Lấy dữ liệu từ database và hiển thị lên bảng hóa đơn  đúng dữ liệu của tiêu đề. |
| 3. | getSelectedID(); | Dùng để lấy ID hiển thị chi tiết hóa đơn admin cần xem |
| 4. | tblHoaDonMouseClicked | Sự kiện nhấp vào hiển thị chi tiết hóa đơn |

### ThongTinCaNhanPanel:

|  |  |  |
| --- | --- | --- |
| **Thứ tự** | **Phương thức** | **Mô tả** |
| 1. | initComponents(); |  |
| 2. | fillToFormSession(); | Lấy dữ liệu từ database từ bảng User Hiển thị các thông tin của người dùng lên form |
| 3. | getFrom(); | Lấy các thực thể người dùng |
| 4. | btnCapNhatActionPerformed(); | Sự kiện button cập nhật thông tin cá nhân |
| 5. | btnDoiMKActionPerformed(); | Sự kiện button chuyển đổi mà hình tới form đổi mật khẩu tài khoản người dùng |

### LichSuHoatDongPanel:

|  |  |  |
| --- | --- | --- |
| **Thứ tự** | **Phương thức** | **Mô tả** |
| 1. | initComponents(); |  |
| 2. | initTable(); | Set table, đặt tên tên tiêu đề cho các cột trong bảng đúng với String[] columns |
| 3. | fillTableSession(); | Lấy dữ liệu từ database từ bảng session cho biết thời gian đăng nhập và đăng xuất của người dùng |

### TrangChuNDPanel:

|  |  |  |
| --- | --- | --- |
| **Thứ tự** | **Phương thức** | **Mô tả** |
| 1. | initComponents(); |  |
| 2. | setLocationRelativeTo(null); | Hiển thị màn hình ở giữa |
| 3. | ChuyenManHinhNguoiDung control = new ChuyenManHinhNguoiDung(jpnView); | Màn hình được hiển thị  khi đăng nhập vai trò user thành công |
| 4. | List<DanhMuc> listItem = new ArrayList<>(); | Click chọn các tab trên menu người dùng gồm Trang chủ, đánh giá, hợp đồng, hóa đơn, thông tin cá nhân, lịch sử hoạt động. |
| 5. | btnThoatActionPerformed(); | Sự kiện button Thoát khi click chọn “Thoát” => đóng màn hình |
| 6. | btnDangXuatActionPerformed() | Sự kiện button Đăng xuất khi click chọn “Đăng xuất”  => Chuyển lại trang chủ chung |

### DanhGiaPanel:

|  |  |  |
| --- | --- | --- |
| **Thứ tự** | **Phương thức** | **Mô tả** |
| 1. | initComponents(); |  |
| 2. | initTable(); | Set table, đặt tên tên tiêu đề cho các cột trong bảng đúng với String[] columns |
| 3. | fillToDanhGia(); | Lấy dữ liệu từ database từ bảng DanhGia hiển thị lên bảng các dữ liệu đánh giá phòng trọ |
| 4. | now(); | Câu lệnh truy vấn khi select mã người dùng và mã phòng trọ từ bảng hợp đồng |
| 5. | getSelectedID(); | Chọn hàng khi click trên bảng DanhGia |
| 6. | btnSuaActionPerformed(); | Sự kiện button Sửa khi click chọn “Sửa” => thông tin được cập nhật |
| 7. | btnResetActionPerformed(); | Dữ liệu trong form trống |
| 8. | tblDanhGiaMouseClicked(); | Sự kiện table cho phép click chọn hàng trên bảng |

### HopDongNDPanel:

|  |  |  |
| --- | --- | --- |
| **Thứ tự** | **Phương thức** | **Mô tả** |
| 1 | fillToTableOrder(); | Lấy dữ liệu từ database và hiển thị lên bảng Hợp Đồng người dùng đúng dữ liệu của tiêu đề. |

### HoaDonNDPanel:

|  |  |  |
| --- | --- | --- |
| **Thứ tự** | **Phương thức** | **Mô tả** |
| 1 | initTable(); | Dùng để show tiêu đề của bảng |
| 2 | fillHoadon(); | Lấy dữ liệu từ database và hiển thị lên bảng Hóa Đơn người dùng đúng dữ liệu của tiêu đề. |
| 3 | btnThanhToanActionPerformed | Bấm vào để hiển thị tài khoản ngân hàng của chủ trọ để thanh toán |
| 4 | getSelectedID() | Dùng để lấy ID hiển thị chi tiết hóa đơn người dùng cần xem |
| 5 | tblHoaDonMouseClicked | Sự kiện nhấp vào hiển thị chi tiết hóa đơn người dùng |

### ThongTinCaNhanNDPanel:

|  |  |  |
| --- | --- | --- |
| **Thứ tự** | **Phương thức** | **Mô tả** |
| 1 | fillToFormSession() | Lấy dữ liệu từ database và hiển thị lên bảng Thông tin cá nhân người dùng đúng dữ liệu của tiêu đề. |
| 2 | getFrom() | Dùng để lấy thông tin trên form table |
| 3 | btnCapNhatActionPerformed() | Sự kiện để  cập nhập thông tin người dùng |
|  | btnDoiMKActionPerformed | Sự kiện mở cửa sổ Panel đổi mật khẩu |

### LichSuHDNDPanel:

|  |  |  |
| --- | --- | --- |
| **Thứ tự** | **Phương thức** | **Mô tả** |
| 1 | initTable(); | Dùng để show tiêu đề của bảng |
| 2 | fillTableSession(); | Lấy dữ liệu từ database và hiển thị lên bảng Lịch sử người dùng đúng dữ liệu của tiêu đề. |

# **Kiểm thử phần mềm và sửa lỗi:**





























# **Đóng gói và triển khai:**

## Sản phẩm phần mềm:

|  |  |  |
| --- | --- | --- |
| **STT** | **Thành phần** | **Mô tả** |
| 1 | TroVietSetUp.exe | File cài đặt |
| 2 | DuAn1\_TroViet.sql | Cơ sở dữ liệu |
| 3. | HuongDanCaiDat.docx | Hướng dẫn cài đặt file |
| 4. | HuongDanSuDung.docx | Hướng dẫn sử dụng file |

## Hướng dẫn cài đặt:

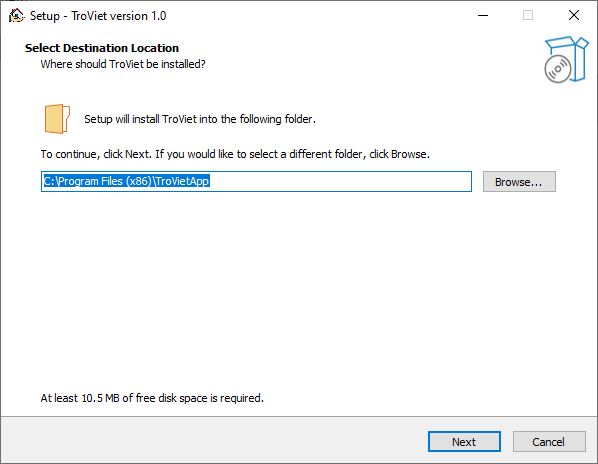
* Bước 1: Tải SQL Server, tạo tài khoản đăng nhập có username là “sa” và password là “123”, sau đó tiến hành chạy file “DuAn1\_TroViet.sql” để tạo database và dữ liệu



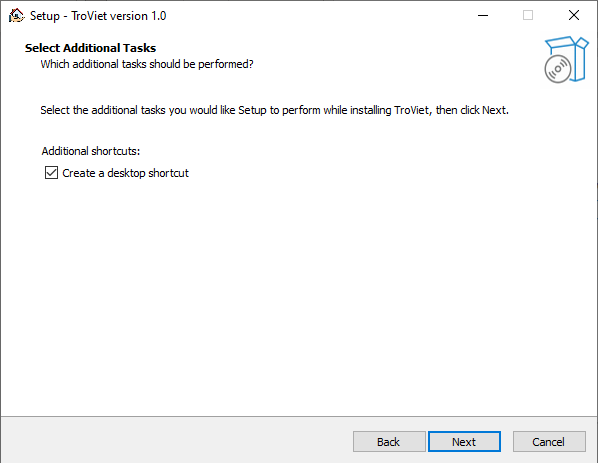
* Bước 2: Sau khi tạo cơ sở dữ liệu thành công, tiến hành chạy file “TroVietSetUp.exe” để cài đặt ứng dụng



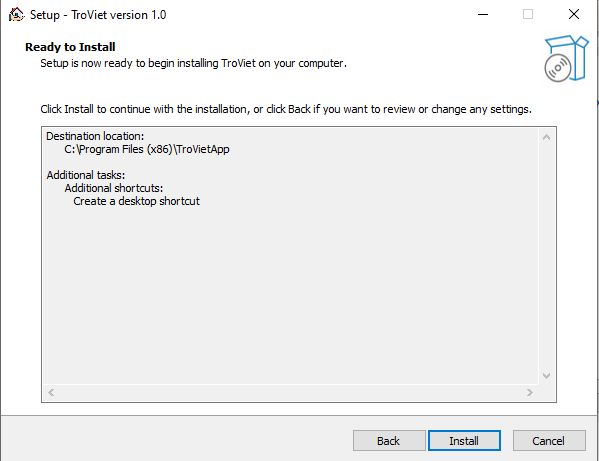
* Bước 3: Lựa chọn vị trí cài đặt cho ứng dụng



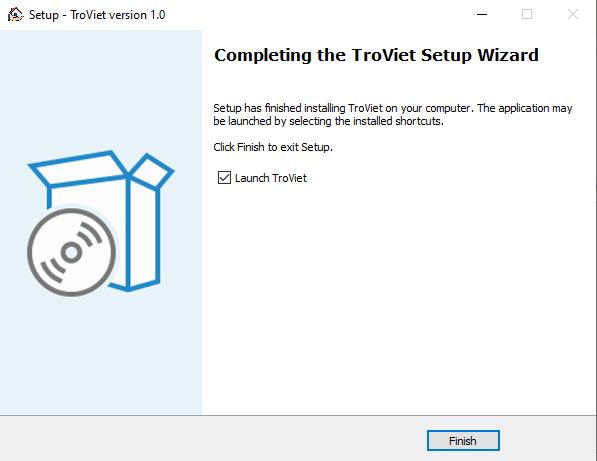
* Bước 4: Lựa chọn tạo shortcut trên desktop



* Bước 5: Chọn Install



* Bước 6: Finish và ngoài desktop sẽ hiển thị ứng dụng





# **Kết luận:**

## Thuận lợi:

* Thực hiện dự án với sự hướng dẫn của thầy Thy và các thành viên nên phát triển dự án khá thuận lợi.
* Các thành viên trao đổi code với nhau

## Khó khăn:

* Vẫn còn hiện tượng trễ deadline, không chịu tìm hiểu code, thống nhất ý tưởng giữa các thành viên.
* Trễ họp, vắng họp không lý do.
* Chưa tích cực tương tác trong group trao đổi khi cần.

HẾT!