**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**



**BÀI TẬP LỚN**

**LẬP TRÌNH JAVA**

**Tên đề tài: QUẢN LÝ KHU CHUNG CƯ**

**ANLAND COMPLEX**

**Giáo viên hướng dẫn : Th.s Vũ Thị Dương**

**Lớp : ĐH – KTPM3 – K11**

**Nhóm : 17**

|  |  |  |
| --- | --- | --- |
| Mã sinh viên | Họ tên | Điểm |
| 1141360205 | Phạm Quốc Cường |  |
| 1141360215 | Phạm Văn Đại |  |
| 2017607983 | Hoàng Văn Độ |  |

(Tài liệu lưu hành nội bộ)

**Hà Nội 2019**

MỤC LỤC

[Chương 1. Phân tích thiết kế 3](#_Toc8250571)

[Quản lý khu chung cư Anland Complex 3](#_Toc8250572)

[1.1 Giới thiệu (Phạm Quốc Cường) 3](#_Toc8250573)

[1.2 Khảo sát hệ thống (Phạm Văn Đại) 3](#_Toc8250574)

[1.2.1 Khảo sát sơ bộ 3](#_Toc8250575)

[1.2.2 Đặc tả yêu cầu người dùng 5](#_Toc8250576)

[1.3 Phân tích hệ thống 7](#_Toc8250577)

[1.3.1 Mô hình hóa chức năng hệ thống (Phạm Văn Đại) 7](#_Toc8250578)

[1.3.2 Mô hình hóa dữ liệu của hệ thống (Hoàng Văn Độ) 17](#_Toc8250579)

[1.4 Thiết kế hệ thống 19](#_Toc8250580)

[1.4.1 Thiết kế giao diện (Phạm Quốc Cường) 19](#_Toc8250581)

[1.4.2 Thiết kế dữ liệu- Ánh xạ lớp sang bảng (Hoàng Văn Độ) 22](#_Toc8250582)

[Chương 2. CÀI ĐẶT <Quản lý chung cư > 26](#_Toc8250583)

[2.1 Giới thiệu công cụ. 26](#_Toc8250584)

[2.1.1 Giới thiệu về NeatBeans – Môi trường phát triển mã nguồn mở . 26](#_Toc8250585)

[2.1.2 NetBeans IDE phát hành : 26](#_Toc8250586)

[2.1.3 Giới thiệu Microsoft SQL Server : 27](#_Toc8250587)

[2.2 Thực hiện bài toán 31](#_Toc8250588)

[2.2.1 Phân công công việc 31](#_Toc8250589)

[2.2.2 Phạm Văn Đại – Thực hiện form Quản Lý với các nhóm chức năng Quản lý khu căn hộ, Quản lý căn hộ, Quản lý tài khoản. 31](#_Toc8250590)

[2.2.3 Hoàng Văn Độ - FORM đăng nhập hệ thống , FORM Quản lý TAB Thông tin mua bán, thông tin cư dân,Thống kê doanh thu . 52](#_Toc8250591)

[2.2.4 Phạm Quốc Cường – Thực hiện Form Nhân viên với các chức năng Xem thông tin mua bán, Quản lý cư dân, In hợp đồng 58](#_Toc8250592)

[2.3 Kết luận (Phạm Quốc Cường) 65](#_Toc8250593)

[2.3.1 Nội dung đã thực hiện 66](#_Toc8250594)

[2.3.2 Hướng phát triển. 66](#_Toc8250595)

# Phân tích thiết kế

# Quản lý khu chung cư Anland Complex

## Giới thiệu (Phạm Quốc Cường)

Chung cư Anland Complex được Tập đoàn Nam Cường khởi công và xây dựng, tọa lạc tại khu đô thị Dương Nội gần 200 ha gần tuyến đường Tố Hữu, quận Hà Đông, thành phố Hà Nội.

Công trình có quy mô gồm: 25 tầng nổi và 2 tầng hầm.

* Bãi đỗ xe: 2 tầng hầm.
* Khối đế: 2 tầng Thương mại dịch vụ.
* Căn hộ: Từ tầng 3 – 25 với 551 căn hộ.

Với số lượng căn hộ tương đối lớn như vậy, việc quản lý và vận hành các tòa nhà chung cư cần 1 đội ngũ nhân sự khá lớn. Việc ứng dụng công nghệ thông tin trong công tác quản lý là một trong những hiệu quả của tin học nhằm giảm nhẹ đến mức tối đa lao động thủ công, tiết kiệm thời gian tìm kiếm thông tin, làm hẹp không gian lưu trữ, hệ thống hóa và cụ thể hóa thông tin theo nhu cầu của người sử dụng, trợ giúp nhà quản lý trong việc ra quyết định.

Chính vì vậy nhóm em xin đề xuất “**Xây dựng phần mềm Quản lý khu chung cư Anland Complex**”.

## Khảo sát hệ thống (Phạm Văn Đại)

Khảo sát sơ bộ

* Mục tiêu :
* Phần mềm giúp nhân viên và người quản lý thao tác nghiệp vụ nhanh chóng và tự động.
* Dữ liệu về khu căn hộ, căn hộ và cư dân trong khu chung cư nhất quán.
* Thống kê doanh thu để biết được lợi nhuận của việc mua – bán căn hộ.
* Phương pháp :

|  |  |
| --- | --- |
| **Kế hoạch phỏng vấn** | |
| **Người được hỏi:** Trần Oanh | **Người phỏng vấn:** Phạm Văn Đại |
| **Địa chỉ:** Tầng 7, tòa nhà Nam Cường, đường Tố Hữu, Hà Đông, Hà Nội.  **SĐT**: 0123456789 | **Thời gian hẹn:** 20/03/2019  **Thời điểm bắt đầu:** 8h00  **Thời điểm kết thúc:** 8h30’ |
| **Đối tượng:**   * Đối tượng được hỏi là: Các thành phần thao tác với hệ thống. * Cần thu thập dữ liệu: Hoạt động hệ thống, yêu cầu chức năng, phi chức năng * Cần thỏa thuận về: thời gian, kinh phí và các yêu cầu khác | **Các yêu cầu đòi hỏi:**  Là người liên quan trực tiếp, có hiểu biết về hoạt động của hệ thống. |
| **Chương trình:**   * Giới thiệu: Bản thân, tên dự án, chủ đề sẽ hỏi. * Tổng quan về dự án: Tìm hiểu về phần mềm cần quản lý * Tổng quan về phỏng vấn: Phỏng vấn chi tiết về hệ thống. * Chủ đề sẽ đề cập: Xin phép được ghi âm. * Chủ đề 1: Hoạt động của hệ thống * Chủ đề 2: Hoạt động kinh doanh. * Tổng hợp các nội dung chính ý kiến của người được hỏi.   **Kết thúc:** Cảm ơn về cuộc phỏng vấn, bắt tay, chào tạm biệt. | **Ước lượng thời gian:**  1 phút  2 phút  1 phút  5 phút  10 phút  3 phút  2 phút |
|  | **Dự kiến tổng cộng:** 24 phút |

*Bảng 1.1 Kế hoạch phỏng vấn*

* Bảng kết quả phỏng vấn :

|  |  |  |
| --- | --- | --- |
| **STT** | **Câu hỏi phỏng vấn** | **Câu trả lời của khách hàng** |
| **1** | Ông/Bà mong muốn phần mềm có những chức năng gì ạ? | * Quản lý mua, bán căn hộ (3) * Quản lý thông tin cư dân (4) * Lập hợp đồng mua, bán nhà (5) * Thống kê doanh thu, cư dân (6) |
| **2** | Cư dân trong chung cư được chia thành nhóm hay loại hay không ạ ? | * Cư dân thì có bình dân và vip (2) |
| **3** | Căn hộ trong chung cư được chia thành nhóm hay không ạ ? | * Căn hộ có 2 loại chung cư và biệt thự (1) |
| **4** | Ông/Bà có những yêu cầu gì về dữ liệu ạ ? | * Dữ liệu phải đảm bảo an toàn, không bị đánh cắp (7) * Backup dữ liệu khi cần (10) |
| **5** | Ông/Bà có những yêu cầu gì về hệ thống ạ ? | * Chạy trên nhiều nền tảng: mobile, web, desktop... (9) * Chạy ổn định, không có lỗi (11) |
| **6** | Ông/Bà có những yêu cầu gì về giao diện ạ ? | * Hình ảnh đẹp màu sắc ưa nhìn (12) |
| **7** | Ông/Bà yêu cầu hoàn thành phần mềm trong bao lâu ? | * Thời gian hoàn thành kịp đưa vào sử dụng (8) |
| **8** | Ông/Bà có thể cụ thể thời gian phải hoàn thành được không ạ ? | * Khoảng 1 tháng, chậm tối đa 3 - 4 ngày.(8) |
| **9** | Ông/Bà sẽ dự định chi bao nhiêu cho phần mềm của mình ? | * Khoảng 100 triệu, có thể thưởng thêm nếu hoàn thành tốt.(-) |
| **10** | Ông/Bà còn những yêu cầu nào khác nữa không ạ ? | * Phần mềm độc quyền, mang dấu ấn tập đoàn.(-) * Bảo hành 10 năm, có hỏng hóc sửa chữa miễn phí.(13) |
| *Bảng 1.2 Kết quả phỏng vấn* | | |

### Đặc tả yêu cầu người dùng

* Phân tích yêu cầu người dùng:

|  |  |  |  |
| --- | --- | --- | --- |
| Nhóm yêu cầu | Yêu cầu | Khả năng thực hiện | Ghi chú |
| Căn hộ | (1) |  | Bỏ qua phần biệt thự |
| (3) |  |  |
| Cư dân | (2) |  | Cư dân có 1 loại duy nhất |
| (4) |  |  |
| Hợp đồng,thống kê | (5) |  |  |
| (6) |  |  |
| Dữ liệu | (7) |  | Dữ liệu vẫn có thể bị mất |
| (10) |  |  |
| Thời gian | (8) |  | Chậm tối đa 3 - 4 ngày |
| Hệ thống | (9) |  | Chỉ chạy trên nền desktop, hệ điều hành windows |
| (11) |  |  |
| (12) |  |  |
| Bảo trì | (13) |  |  |

*Bảng 1.3 Phân tích yêu cầu người dùng*

* Yêu cầu chức năng:

1. Nhóm quản lý căn hộ: ( Dữ liệu do ban quản lý hạ tầng cung cấp)

* Quản lý khu căn hộ, căn hộ.
* Xem thông tin căn hộ

1. Nhóm quản lý cư dân: ( Dữ liệu do khách hàng và cư dân cung cấp)

* Quản lý thông tin cư dân
* Xem thông tin cư dân

1. Nhóm quản lý mua - bán:

* Lập hợp đồng mua, bán
* Thống kê doanh thu
* Xem thông tin mua - bán căn hộ

1. Nhóm quản lý truy cập hệ thống:

* Đăng nhập
* Đăng xuất

1. Nhóm quản lý tài khoản:

* Thêm tài khoản
* Sửa tài khoản
* Xóa tài khoản
* Yêu cầu phi chức năng:

1. Yêu cầu về sản phẩm:

* Dữ liệu không bị hack hay mất mát
* Tốc độ xử lý, phản hồi nhanh
* Xử lý được lượng dữ liệu lớn
* Giao diện đẹp, tươi sáng
* Có thể thêm các chức năng cần thiết sau này
* Sửa chữa, bảo trì miễn phí
* Bảo hành 10 năm.

1. Yêu cầu về tổ chức/tiến trình phát triển:

* Ngôn ngữ lập trình: Java - Swing
* Hệ quản trị cơ sở dữ liệu: Microsoft SQL Server
* Chuẩn chất lượng phần mềm : ISO/IEC 9126
* Hệ điều hành: Windows

1. Yêu cầu từ bên ngoài:

* Thời gian: 1 tháng(chậm tối đa 3 – 4 ngày)
* Chi phí: 100.000.000đ
* Bản quyền: Độc quyền.
* Liên kết phần mềm khác: Không
* Mô tả hoạt động của hệ thống:

1. Giao diện đăng nhập( có 2 loại tài khoản: nhân viên và người quản lý): Nếu là nhân viên, phải đăng nhập bằng tài khoản được người quản lý cấp trước đó. Nếu là người quản lý, phải đăng nhập bằng tài khoản được cấp trước bởi nhà phát triển (có thể thay đổi sau này).
2. Giao diện dành cho nhân viên: Khi chọn đăng nhập dành cho nhân viên. Nhân viên có thể thực hiện các công việc sau:

* Nhập thông tin cư dân mua nhà.
* In hợp đồng mua nhà cho cư dân mới.
* Chỉnh sửa thông tin của cư dân.
* Xem danh sách thông tin mua – bán căn hộ.
* Xem thông tin căn hộ.

1. Giao diện dành cho người quản lý: Khi chọn đăng nhập dành cho quản lý. Người quản lý phải có username và password để đăng nhập vào hệ thống. Sau khi đăng nhập thành công, người quản lý có thể làm các công việc sau:

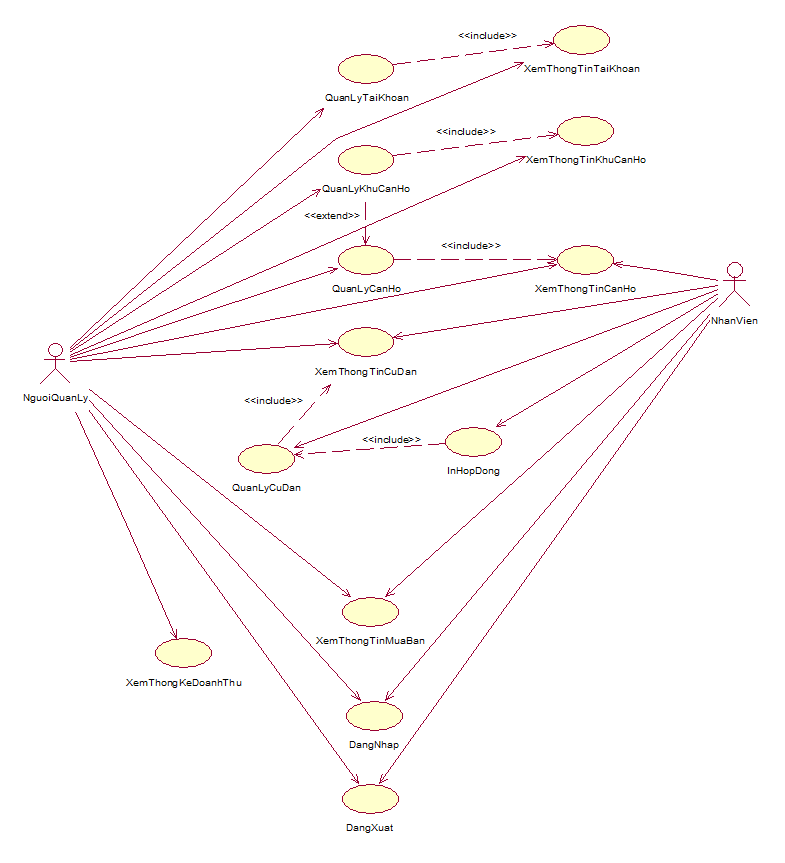
* Thêm khu căn hộ mới, sửa thông tin khu căn hộ. Xóa khu căn hộ khỏi hệ thống.
* Sửa thông tin căn hộ.
* Xem danh sách thông tin mua – bán căn hộ.
* Xem thống kê doanh thu.
* Xem thông tin cư dân
* Quản lý tài khoản

1. Hệ thống có nút đăng xuất ở bất kỳ giao diện nào. Khi chọn nút “Đăng Xuất” đóng các thao tác hiện tại mở lại giao diện đăng nhập.

## Phân tích hệ thống

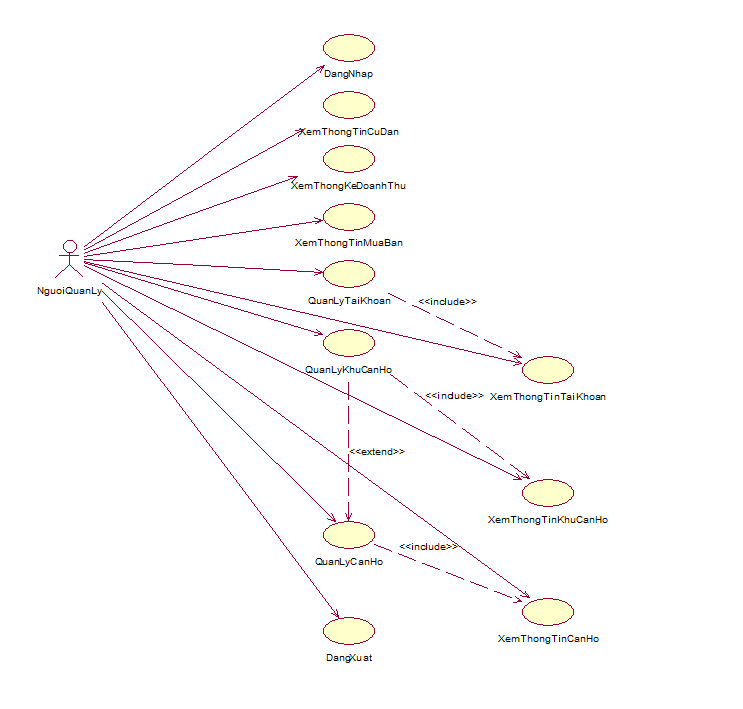
### Mô hình hóa chức năng hệ thống (Phạm Văn Đại)

* Biểu đồ use case của hệ thống :



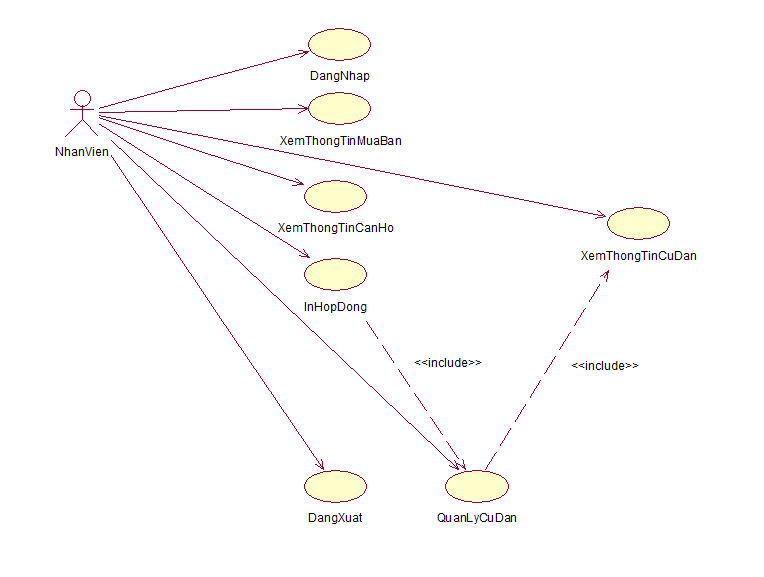
Hình 1.1 Biểu đồ use case của hệ thống

* Biểu đồ use case của actor người quản lý :



Hình 1.2 Biểu đồ use case của người quản lý

* Biểu đồ use case của actor nhân viên :



Hình 1.3 Biểu đồ use case của nhân viên

1. Mô tả use case: Đăng nhập

* **Mô tả vắn tắt:**

Use case này cho phép người dùng (người quản lý, nhân viên) đăng nhập hệ thống bằng tên đăng nhập và mật khẩu để truy nhập các chức năng của hệ thống theo vai trò.

* **Luồng sự kiện:**
* Luồng cơ bản:

1. Use case này bắt đầu khi người dùng kích vào nút “Đăng nhập” trên giao diện đăng nhập của hệ thống. Hệ thống kiểm tra tên và mật khẩu của người dùng trong bảng “TAIKHOAN”. Nếu là tài khoản dành cho nhân viên chuyển sang bước 2. Nếu là tài khoản của người quản lý chuyển sang bước 3.
2. Hệ thống sẽ hiển thị giao diện quản lý dành cho nhân viên.
3. Hệ thống sẽ hiển thị giao diện quản lý dành cho người quản lý.
4. Use case này kết thúc.

* Luồng rẽ nhánh:

1. Hủy bỏ: Tại Bước 2 trong luồng cơ bản, nếu người dùng kích vào nút “Hủy bỏ” thì use case kết thúc.
2. Sai tên hoặc mật khẩu: Tại bước 1 trong luồng cơ bản nếu người dùng nhập sai tên đăng nhập hay mật khẩu thì hệ thống sẽ thông báo lỗi. Người dùng có thể nhập lại và tiếp tục bước 2 hoặc chọn hủy bỏ để kết thúc use case.
3. Không thể kết được với cơ sở dữ liệu: Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

* **Các yêu cầu đặc biệt:**

Use case này được dùng để phân quyền của người dùng theo tên đăng nhập. Các quyền được kiểm soát theo vai trò người quản lý, nhân viên.

* **Tiền điều kiện:**

Người quản lý phải có tài khoản được nhà phát triển cung cấp trước đó .

Nhân viên phải có tài khoản do người quản lý cấp.

* **Hậu điều kiện:**

Người dùng đăng nhập được vào hệ thống.

1. Mô tả use case: Quản lý khu căn hộ

* **Mô tả vắn tắt:**

Use case này cho phép người quản lý có thể quản lý khu căn hộ

* **Luồng sự kiện:**
* Luồng cơ bản:

1. Use case này bắt đầu khi người quản lý bấm vào nút “Quản Lý Khu Căn Hộ”. Hệ thống sẽ lấy thông tin từ bảng “KHUCANHO” và hiển thị danh sách thông tin các khu căn hộ.
2. Thêm khu căn hộ: Người quản lý bấm vào nút “Thêm”. Hệ thống hiển thị một giao diện thêm khu căn hộ. Người quản lý nhập các thông tin cho khu căn hộ rồi bấm vào nút “Lưu”. Hệ thống sẽ thêm khu căn hộ mới vào bảng “KHUCANHO” và thêm mới các căn của khu vừa thêm vào bảng “CANHO” .
3. Sửa khu căn hộ: Người quản lý chọn khu căn hộ có thông tin sai, nhập lại thông tin cho đúng và nhấn nút “Sửa”. Hệ thống sẽ cập nhật lại thông tin về khu căn hộ rồi lưu lại vào CSDL.
4. Xóa khu căn hộ: Người quản lý chọn khu căn hộ cần xóa và bấm nút “Xóa”. Hệ thống sẽ xóa toàn bộ thông tin liên quan đến khu căn hộ đó. Bao gồm cả thông tin về các căn hộ trong khu đó.
5. Use case này kết thúc khi người quản lý bấm vào 1 mục quản lý khác trên giao diện quản lý.

* Luồng rẽ nhánh:

1. Tại bước 2 và 3 trong luồng cơ bản nếu người quản lý nhập thông tin cho khu căn hộ không hợp lệ. Hệ thống sẽ hiện thông báo lỗi. Người quản lý có thể nhập lại hoặc nhấn “Hủy” .
2. Không thể kết nối được với cơ sở dữ liệu: Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

* **Các yêu cầu đặc biệt:** Chỉ người quản lý mới sử dụng được use case này.
* **Tiền điều kiện:** Người quản lý phải đăng nhập vào hệ thống trước khi use case này bắt đầu.
* **Hậu điều kiện:** Không.

1. Mô tả use case: Quản lý căn hộ

* **Mô tả vắn tắt:** Use case này cho phép người quản lý có thể quản lý căn hộ
* **Luồng sự kiện:**
* Luồng cơ bản:

1. Use case này bắt đầu khi người quản lý bấm vào nút “Quản Lý Căn Hộ”. Hệ thống sẽ lấy thông tin từ bảng “CANHO” và hiển thị danh sách thông tin các căn hộ.
2. Thêm căn hộ: Hệ thống tự động thêm căn hộ mới khi có 1 khu mới được thêm.
3. Sửa căn hộ: Người quản lý chọn căn hộ có thông tin sai, nhập lại thông tin cho đúng và nhấn nút “Sửa”. Hệ thống sẽ cập nhật lại thông tin về căn hộ rồi lưu lại vào CSDL.
4. Xóa căn hộ: Hệ thống sẽ tự động xóa các căn hộ của 1 khu, khi khu đó bị xóa.
5. Use case này kết thúc khi người quản lý bấm vào 1 mục quản lý khác trên giao diện quản lý.

* Luồng rẽ nhánh:

1. Tại bước 3 trong luồng cơ bản nếu người quản lý nhập thông tin cho căn hộ không hợp lệ. Hệ thống sẽ hiện thông báo lỗi. Người quản lý có thể nhập lại hoặc nhấn “Hủy”.
2. Không thể kết nối được với cơ sở dữ liệu: Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

* **Các yêu cầu đặc biệt:** Chỉ người quản lý mới sử dụng được use case này.
* **Tiền điều kiện:** Người quản lý phải đăng nhập vào hệ thống trước khi use case này bắt đầu.
* **Hậu điều kiện:** Không.

1. Mô tả use case: Quản lý tài khoản

* **Mô tả vắn tắt:** Use case này cho phép người quản lý có thể quản lý tài khoản của mình và nhân viên.
* **Luồng sự kiện:**
* Luồng cơ bản:

1. Use case này bắt đầu khi người quản lý bấm vào nút “Quản Lý Tài Khoản”. Hệ thống sẽ lấy thông tin từ bảng “TAIKHOAN” và hiển thị danh sách thông tin của các tài khoản.
2. Thêm tài khoản( chỉ dành cho tài khoản của nhân viên): Người quản lý nhập các thông tin cho tài khoản mới rồi bấm vào nút “Thêm”. Hệ thống sẽ thêm một tài khoản mới vào CSDL.
3. Sửa tài khoản : Người quản lý chọn tài khoản của nhân viên hoặc của mình nhập lại thông tin cho đúng và nhấn nút “Sửa”. Hệ thống sẽ cập nhật lại thông tin về tài khoản rồi lưu lại vào CSDL.
4. Xóa tài khoản ( chỉ dành cho tài khoản của nhân viên): Người quản lý chọn tài khoản nhân viên cần xóa và bấm nút “Xóa”. Hệ thống sẽ xóa tài khoản đó khỏi cơ sở dữ liệu.
5. Use case này kết thúc khi người quản lý bấm vào 1 mục quản lý khác trên giao diện quản lý.

* Luồng rẽ nhánh:

1. Tại bước 2 và 3 trong luồng cơ bản nếu người quản lý nhập thông tin cho căn hộ không hợp lệ. Hệ thống sẽ hiện thông báo lỗi. Người quản lý có thể nhập lại hoặc nhấn “Hủy”.
2. Không thể kết nối được với cơ sở dữ liệu: Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

* **Các yêu cầu đặc biệt:** Chỉ người quản lý mới sử dụng được use case này.
* **Tiền điều kiện:** Người quản lý phải đăng nhập vào hệ thống trước khi use case này bắt đầu.
* **Hậu điều kiện:** Không.

1. Mô tả use case: Xem thống kê doanh thu

* **Mô tả vắn tắt:** Use case này cho phép người quản lý xem doanh thu của chung cư.
* **Luồng sự kiện:**
* Luồng cơ bản:

1. Use case này bắt đầu khi người quản lý bấm vào nút “Thống Kê Doanh Thu” trên giao diện quản lý. Hệ thống hiển thị giao diện thống kê.
2. Người dùng chọn các chỉ tiêu để thống kê rồi bấm nút “Thống Kê” để xem thống kê. Hệ thống hiển thị bảng thống kê theo các chỉ tiêu đã chọn.
3. Use case này kết thúc.

* Luồng rẽ nhánh:

1. Tại bước 2 trong luồng cơ bản, nếu người dùng bấm nút “Thoát”. Hệ thống đóng giao diện thống kê. Và chuyển đến giao diện quản lý. Use case này kết thúc.
2. Không thể kết nối được với cơ sở dữ liệu: Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

* **Các yêu cầu đặc biệt:** Chỉ người quản lý mới có thể sử dụng được use case này.
* **Tiền điều kiện:** Người quản lý phải đăng nhập vào hệ thống trước khi use case này bắt đầu.
* **Hậu điều kiện:** Người quản lý xem được doanh thu của chung cư.

1. Mô tả use case: Xem thông tin căn hộ

* **Mô tả vắn tắt:** Use case này cho phép người dùng xem thông tin căn hộ.
* **Luồng sự kiện:**
* Luồng cơ bản:

1. Use case này bắt đầu khi người dùng truy cập giao diện quản lý căn hộ . Hệ thống hiển thị thông tin của căn hộ lên 1 dữ liệu gồm các thông tin như tên căn hộ, số phòng, tên khu, diện tích,..
2. Use case này kết thúc.

* Luồng rẽ nhánh:

1. Không thể kết nối được với cơ sở dữ liệu: Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. Use case này kết thúc.

* **Các yêu cầu đặc biệt:** Không
* **Tiền điều kiện:** Phải đăng nhập vào hệ thống trước khi bắt đầu use case này.
* **Hậu điều kiện:** Hệ thống hiển thị thông tin căn hộ.

1. Mô tả use case: Xem thông tin mua bán

* **Mô tả vắn tắt:** Use case này cho phép người dùng xem thông tin mua bán nhà.
* **Luồng sự kiện:**
* Luồng cơ bản:

1. Use case này bắt đầu khi người dùng truy cập vào giao diện thông tin mua- bán nhà . Hệ thống hiển thị giao diện thông tin mua – bán nhà gồm ngày giờ mua, người mua, mua căn hộ và giá mua. Use case này kêt thúc.

* Luồng rẽ nhánh:

1. Không thể kết được với cơ sở dữ liệu: Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. Use case này kết thúc.

* **Các yêu cầu đặc biệt:** Không
* **Tiền điều kiện:** Phải đăng nhập vào hệ thống trước khi bắt đầu use case này.
* **Hậu điều kiện:** Hệ thống hiển thị giao diện mua bán nhà.

1. Mô tả use case: Xem thông tin cư dân

* **Mô tả vắn tắt:** Use case này cho phép người dùng xem thông tin cư dân( chủ hộ) trong chung cư.
* **Luồng sự kiện:**
* Luồng cơ bản:

1. Use case này bắt đầu khi người dùng truy cập vào giao diện quản lý cư dân. Hệ thống hiển thị giao diện gồm các thông tin liên quan đến cư dân. Use case này kết thúc.

* Luồng rẽ nhánh:

1. Không thể kết được với cơ sở dữ liệu: Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. Use case này kết thúc

* **Các yêu cầu đặc biệt:** Không
* **Tiền điều kiện:** Không
* **Hậu điều kiện:** Hệ thống hiển thị thông tin cư dân

1. Mô tả use case: Quản lý cư dân

* **Mô tả vắn tắt:** Use case này cho phép nhân viên có thể quản lý cư dân
* **Luồng sự kiện:**
* Luồng cơ bản:

1. Use case này bắt đầu khi nhân viên bấm vào nút “Quản Lý Cư Dân”. Hệ thống sẽ lấy thông tin từ bảng “CUDAN” và hiển thị danh sách thông tin các cư dân.
2. Thêm cư dân: Hệ thống sẽ thêm 1 cư dân mới vào CSDL khi có 1 hợp đồng mua nhà mới.
3. Sửa cư dân: Nhân viên chọn cư dân có thông tin sai, nhập lại thông tin cho đúng và nhấn nút “Sửa”. Hệ thống sẽ cập nhật lại thông tin về cư dân rồi lưu lại vào CSDL.
4. Use case này kết thúc khi người nhân viên bấm vào 1 mục quản lý khác trên giao diện quản lý.

* Luồng rẽ nhánh:

1. Tại bước 2 trong luồng cơ bản nếu cư dân đã có trong bảng “CUDAN” thì hệ thống bỏ qua bước 2.
2. Tại bước 3 trong luồng cơ bản nếu nhân viên nhập thông tin cho căn hộ không hợp lệ. Hệ thống sẽ hiện thông báo lỗi. Nhân viên có thể nhập lại hoặc nhấn “Hủy”.
3. Không thể kết được với cơ sở dữ liệu: Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

* **Các yêu cầu đặc biệt:** Chỉ nhân viên mới sử dụng được use case này.
* **Tiền điều kiện:** Nhân viên phải đăng nhập vào hệ thống trước khi use case này bắt đầu.
* **Hậu điều kiện:** Không.

1. Mô tả use case: In hợp đồng

* **Mô tả vắn tắt:** Use này cho cho phép nhân viên in hợp đồng cho khách hàng đến mua nhà.
* **Luồng sự kiện:**
* Luồng cơ bản:

1. Use case này bắt đầu khi nhân viên bấm vào nút “In Hợp Đồng”. Hệ thống hiển thị ra bản hợp đồng mua – bán nhà.
2. Hệ thống mở giao diện danh sách các căn hộ trống rồi chọn căn mà khách chọn.
3. Tích chọn khách hàng chưa là cư dân. Nhập các thông tin cần thiết rồi bấm vào nút “In”. Hệ thống in ra hợp đồng mua – bán nhà. Đồng thời, thêm mới 1 cư dân vào CSDL. Use case này kết thúc.
4. Tích chọn khách hàng là cư dân. Chọn mã cư dân , hoặc tên trong 1 list rồi bấm “In hợp đồng”. Use case này kết thúc.

* Luồng rẽ nhánh:

1. Tại bước 2 trong luồng cơ bản, nếu nhân viên nhập các thông tin của khách hàng không hợp lệ hệ thống sẽ hiển thị thông báo lỗi. Nhân viên có thể nhập lại thông tin hoặc chọn nút “Hủy” để hủy bỏ thao tác.
2. Không thể kết được với cơ sở dữ liệu: Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

* **Các yêu cầu đặc biệt:** Chỉ có nhân viên mới có thể sử dụng use case này
* **Tiền điều kiện:** Nhân viên phải đăng nhập trước khi bắt đầu use case.
* **Hậu điều kiện:** Hệ thống thêm mới 1 cư dân và in ra hợp đồng mua bán nhà cho cư dân đó.

1. Mô tả use case: Đăng xuất

* **Mô tả vắn tắt:** Use case này cho phép người dùng thoát khỏi hệ thống.
* **Luồng sự kiện:**
* Luồng cơ bản:

1. Use case này bắt đầu khi người dùng bấm vào nút “Đăng xuất” trên các giao diện quản lý của hệ thống. Hệ thống đóng các giao diện quản lý và mở ra giao diện đăng nhập. Use case này kết thúc.

* Luồng rẽ nhánh: Không
* **Các yêu cầu đặc biệt:** Không
* **Tiền điều kiện:** Không
* **Hậu điều kiện:** Người dùng thoát khỏi hệ thống quản lý.

### Mô hình hóa dữ liệu của hệ thống (Hoàng Văn Độ)

**Class Diagram là một trong những bản vẽ quan trọng nhất của thiết kế phần mềm, nó cho thấy cấu trúc và quan hệ giữa các thành phần tạo nên phần mềm. Trong quá trình xây dựng Class Diagram chúng ta sẽ phải quyết định rất nhiều yếu tố về thiết kế nên nó là bản vẽ khó xây dựng nhất. Bản vẽ này sẽ cho thấy cấu trúc tĩnh của phần mềm, tương tự như bản vẽ mặt bằng trong thiết kế của ngành xây dựng.**

1. **Cách xác định (lớp) đối tượng.**

Ý tưởng chung : Ánh xạ biểu đồ ca sử dụng sang biểu đồ lớp.

* Bước 1 : Xác định mục đích ca sử dụng
* Thường được thể hiện dưới dạng dữ liệu hoặc thông tin sẽ cung cấp cho các tác nhân.
* Ví dụ: người quản lý căn hộ có thể xem thông tin của căn hộ, chủ hộ….
* Bước 2: Dựa vào các mục đích để xác định đối tượng
* Để thực hiện được mục đích, phải có các thông tin (dữ liệu vào ra )
* Các dữ liệu này sẽ là các đối tượng , hoặc thông tin về đối tượng.
* Ví dụ: Để có thông tin về 1 căn hộ thì cần có : Tên căn hộ, diện tích, nội thất, giá thành, địa chỉ….
* Bước 3 : Xác định hàm thành phần
* Để thực hiên bước này ta trả lời cho các câu hỏi :

Ca sử dụng này cần làm gì với mỗi đối tượng ?

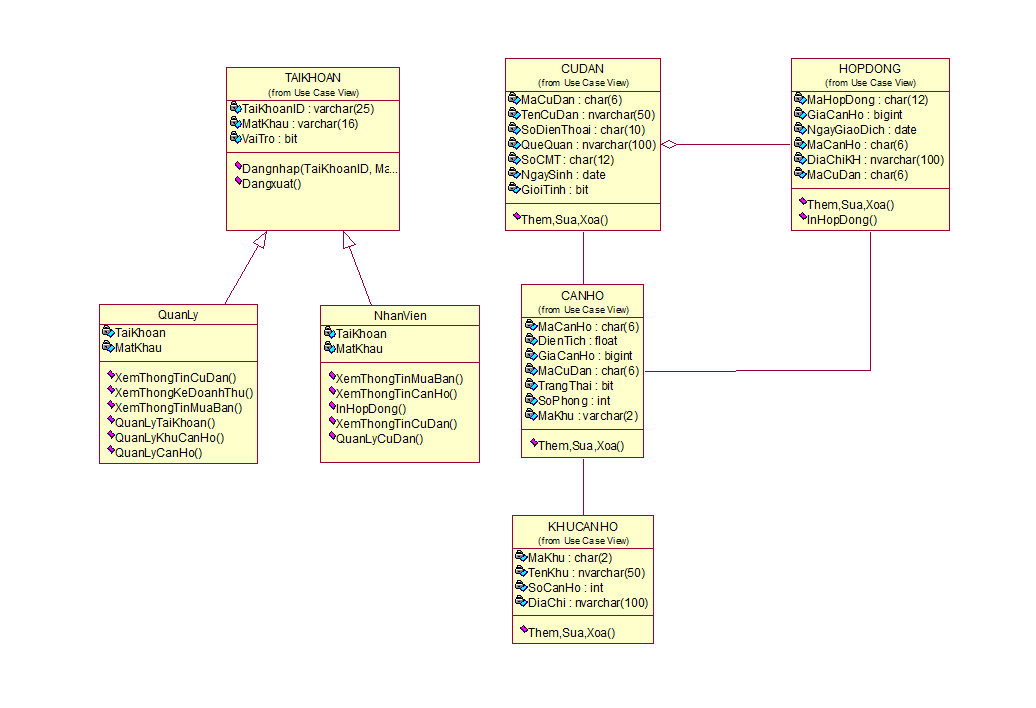
Ca sử dụng này cần biết gì về mỗi đối tượng ?

* Bước 4 : Xác định mối quan hệ .
* Với mỗi đối tượng: liên kết với đối tượng khác, phụ thuộc hay được sinh ra (quan hệ kết hợp).
* Với mỗi quan hệ : mỗi cá thể (thể hiện) thuộc lớp này có thể kết hợp được với bao nhiêu cá thể của lớp khác.

1. **Tìm các class dự kiến**

* Xem xét Use case Diagram của hệ thống :
* Use case Đăng nhập xác định thực thể thể là người dùng, được phân chia thành 2 loại là Người quản lý và Nhân viên xác định Class TaiKhoan
* Use case Quản lý khu căn hộ xác định Class Khu Căn Hộ
* Use case Quản lý căn hộ, thông tin căn hộ, thống kê kinh doanh Xác định Class Căn hộ.
* Use case Thông tin mua bán, in hợp đồng xác định class Hợp đồng.
* Use case Xem thông tin cư dân, quản lý cư dân xác định Class Cư dân .
* Xác định thuộc tính và quan hệ cho các lớp
* Class TaiKhoan bao gồm các thuộc tính : Tên tài khoản, Mật khẩu, Vai trò.
* Class Khu căn hộ: Mã khu căn hộ, Tên khu , Số tầng, Số căn/ tầng, Địa chỉ
* Class Căn hộ thì bao gồm : Mã căn hộ, Diện tích, Giá thành, Mã cư dân và Trạng Thái, Số Phòng.
* Class Hợp đồng : Mã hợp đồng, Giá căn hộ, Ngày giao dịch, Mã căn hộ, Mã Cư Dân, Địa chỉ Khách Hàng.
* Cư dân: Mã cư dân, Tên cư dân, Số điện thoại, Địa chỉ, số chứng minh thư, Số Điện Thoại, Ngày Sinh, Giới Tính, Quê Quán.
* Xác định phương thức của các lớp.
* Class Tài Khoản: có các phương thức đăng nhập đăng xuất, thêm sửa xóa.
* Class Khu căn hộ, căn hộ, cư dân, hợp đồng cũng có các phương thức Thêm, sửa, xóa.
* Phân tích mối quan hệ giữa các lớp.
* Khu căn hộ: Một khu căn hộ thì có nhiều căn hộ.
* Một căn hộ chỉ có duy nhất một bản hợp đồng.
* Cư dân : một cư dân có thể có nhiều bản hợp đồng mua-bán căn hộ.

1. **Sau khi tìm kiếm xác định phân tích các bước ở trên ta có biểu đồ lớp như sau :**

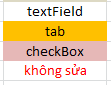
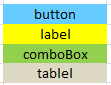


Hình 1.4 Biểu đồ lớp

## Thiết kế hệ thống

### Thiết kế giao diện (Phạm Quốc Cường)

1. Màn hình chính



* Chú thích:
* Màn hình đăng nhập:



Là quản lý

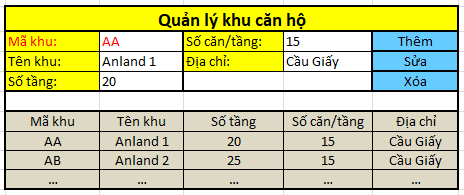
Là nhân viên

|  |  |
| --- | --- |
| Màn hình cho Người quản lý | Màn hình cho Nhân viên |
| C:\Users\quocc.QUOCCUONG\AppData\Local\Microsoft\Windows\INetCache\Content.Word\form-quan-ly.png  **(6)**  **(5)**  **(4)**  **(3)**  **(2)**  **(1)** | C:\Users\quocc.QUOCCUONG\AppData\Local\Microsoft\Windows\INetCache\Content.Word\form-nv.png  **(9)**  **(8)**  **(7)** |

1. Màn hình cho Người quản lý

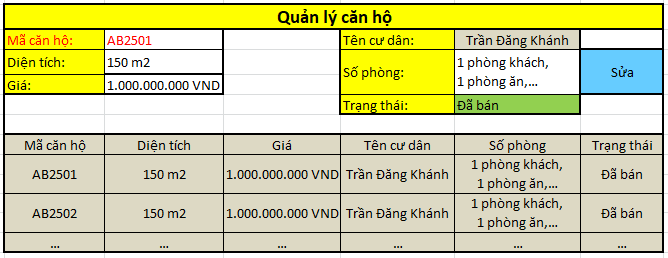
* Chức năng Quản lý khu căn hộ (1)

**(1)**



* Chức năng Quản lý căn hộ (2)

**(2)**



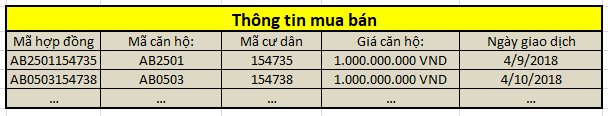
* Chức năng Quản lý tài khoản (3)

**(3)**



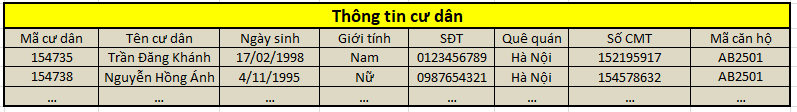
* Chức năng Xem thông tin mua bán (4)

**(4)**



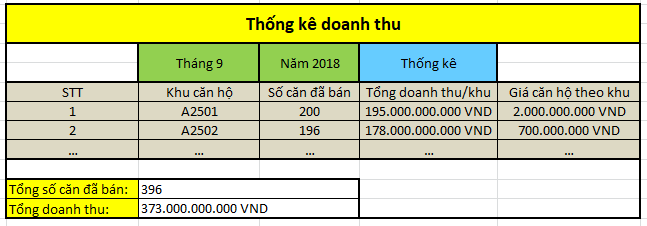
* Chức năng Xem thông tin cư dân (5)

**(5)**



* Chức năng Thống kê doanh thu (6)

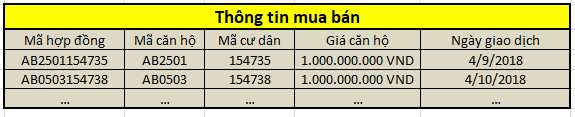
**(6)**



1. Màn hình cho Nhân viên

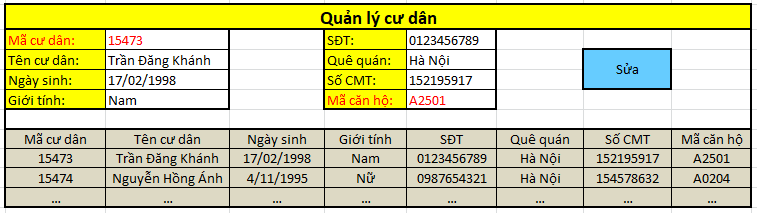
* Chức năng Xem thông tin mua bán (7)

**(7)**



* Chức năng Quản lý cư dân (8)

**(8)**



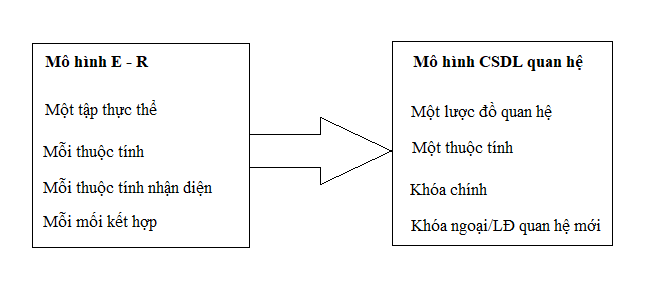
* Chức năng Xem thông tin căn hộ và In hợp đồng (9)

**(9)**



### Thiết kế dữ liệu- Ánh xạ lớp sang bảng (Hoàng Văn Độ)

#### Quy tắc chuyển đổi:



**Xử lý kiểu liên kết :**

Quy tắc 1 : với kiểu liên kết 1:1

Cách 1: Chuyển khóa chính của LĐQH này sang làm khóa ngoại của LĐQH hoặc ngược lại.

Cách 2: Nhập 2 kiểu thực thể và mối liên kết thành 1 LĐQH, chọn khóa chính cho phù hợp.

Quy tắc 2 : với kiểu liên kết 1:n

Chuyển khóa chính của LĐQH bên 1 (cha) sang làm khóa ngoại của LĐQH bên nhiều (con)

Quy tắc 3: Với kiểu liên kết n:n Chuyển mối liên kết thành một LĐQH có thuộc tính là thuộc tính của mối liên kết, thêm các thuộc tính khóa chính của các LĐQH có liên quan, khóa chính của LĐQH mới này là các thuộc tính mới thêm vào.

Quy tắc 4: Với thuộc tính đa trị Tách thuộc tính đa trị ra khỏi kiểu thực thể ban đầu, thêm một kiểu thực thể mới chứa thuộc tính đa trị này, xác định lại mối liên kết của kiểu thực thể này với kiểu thực thể ban đầu. Áp dụng các quy tắc từ 1 đến quy tắc 3

#### Ánh xạ lớp sang bảng

* **Từ Lớp TaiKhoan bao gồm lớp QuanLy và lớp NhanVien ta tạo Bảng TAIKHOAN.**

CREATE TABLE TAIKHOAN

(

TenTaiKhoan VARCHAR(25) CONSTRAINT pk\_TAIKHOAN PRIMARY KEY,

MatKhau VARCHAR(16) NOT NULL ,

VaiTro BIT NOT NULL

DEFAULT 0

)

* **Lớp CUDAN -> Bảng CUDAN**

CREATE TABLE CUDAN

(

MaCuDan CHAR(6) CONSTRAINT pk\_CUDAN PRIMARY KEY ,

TenCuDan NVARCHAR(50) NOT NULL ,

NgaySinh DATE NOT NULL

DEFAULT GETDATE() ,

GioiTinh BIT NOT NULL

DEFAULT 0 ,

SoDT CHAR(10) NOT NULL ,

SoCMT CHAR(12) NOT NULL ,

QueQuan NVARCHAR(100) NOT NULL

)

* **Lớp KHUCANHO -> Bảng KHUCANHO**

CREATE TABLE KHUCANHO

(

MaKhu CHAR(2) CONSTRAINT pk\_KHUCANHO PRIMARY KEY ,

TenKhu NVARCHAR(50) NOT NULL ,

SoTang INT NOT NULL

DEFAULT 1 ,

SoCanTT INT NOT NULL

DEFAULT 1 ,

DiaChi NVARCHAR(100) NOT NULL,

)

* **Lớp CANHO -> Bản CANHO**

CREATE TABLE CANHO

(

MaCanHo CHAR(6) CONSTRAINT pk\_CANHO PRIMARY KEY ,

DienTich FLOAT NOT NULL

DEFAULT 50 ,

Gia BIGINT NOT NULL ,

TrangThai BIT NOT NULL

DEFAULT 0 ,

SoPhong INT NOT NULL

DEFAULT 5 ,

MaCuDan CHAR(6)

CONSTRAINT fk1\_CANHO FOREIGN KEY REFERENCES dbo.CUDAN ( MaCuDan ) ,

MaKhu CHAR(2)

NOT NULL

DEFAULT 'AA'

CONSTRAINT fk2\_CANHO

FOREIGN KEY REFERENCES dbo.KHUCANHO ( MaKhu ) ON DELETE CASCADE

ON UPDATE CASCADE -- (AA -> ZZ)

)

* **Lớp HOPDONG -> Bảng HOPDONG**

CREATE TABLE HOPDONG

(

MaHopDong CHAR(12) NOT NULL

CONSTRAINT pk\_HOPDONG PRIMARY KEY , -- HD0000000001

NgayGiaoDich DATE NOT NULL

DEFAULT GETDATE() ,

DiaChiKH NVARCHAR(100) NOT NULL ,

MaCuDan CHAR(6)

NOT NULL

CONSTRAINT fk1\_HOPDONG FOREIGN KEY REFERENCES dbo.CUDAN ( MaCuDan ) ,

MaCanHo CHAR(6)

NOT NULL

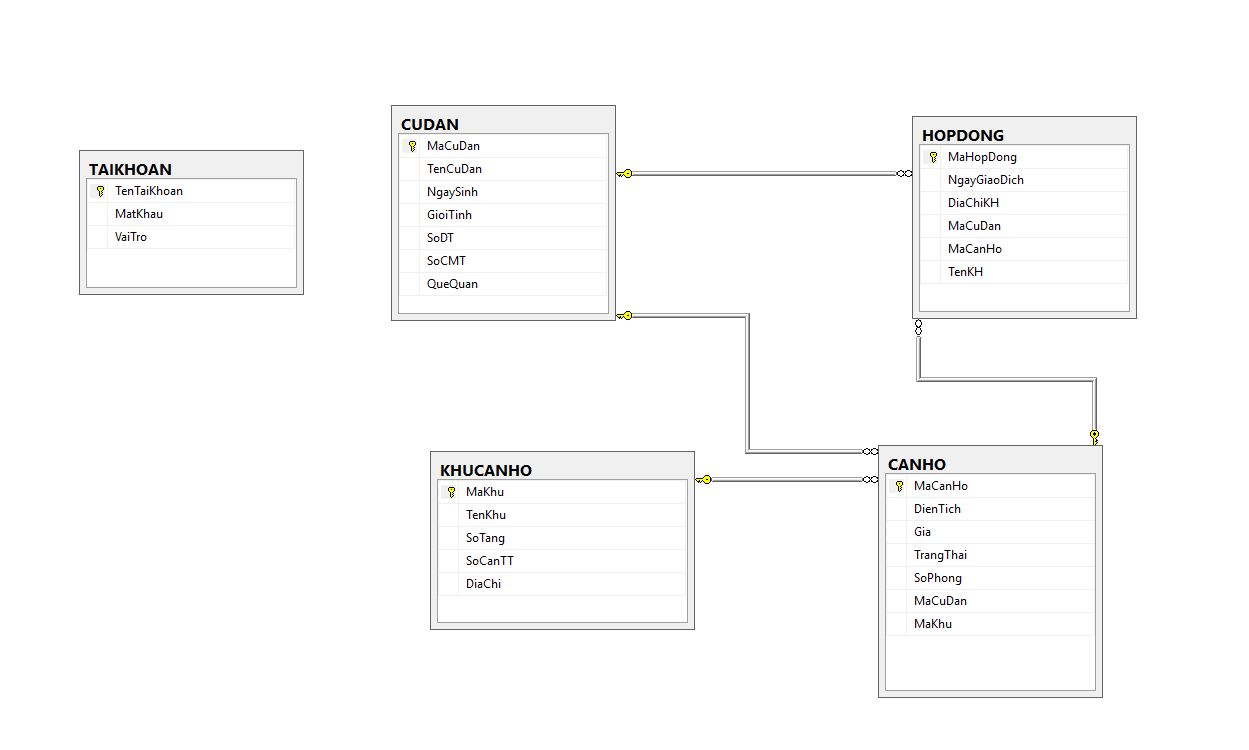
CONSTRAINT fk2\_HOPDONG

FOREIGN KEY REFERENCES dbo.CANHO ( MaCanHo ) ON DELETE CASCADE

ON UPDATE CASCADE,

)

#### Lược đồ quan hệ giữa các bảng



# CÀI ĐẶT <Quản lý chung cư >

## Giới thiệu công cụ.

### Giới thiệu về NeatBeans – Môi trường phát triển mã nguồn mở .

**NetBeans** là một [môi trường phát triển tích hợp](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) cho [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) . NetBeans cho phép các ứng dụng được phát triển từ một tập hợp các [thành phần phần mềm](https://en.wikipedia.org/wiki/Software_component)*mô-đun* được gọi là *mô-đun* .

 NetBeans chạy trên [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) , [macOS](https://en.wikipedia.org/wiki/MacOS) , [Linux](https://en.wikipedia.org/wiki/Linux) và [Solaris](https://en.wikipedia.org/wiki/Solaris_(operating_system)) . Ngoài việc phát triển Java, nó còn có các phần mở rộng cho các ngôn ngữ khác như [PHP](https://en.wikipedia.org/wiki/PHP) , [C](https://en.wikipedia.org/wiki/C_(programming_language)) , [C ++](https://en.wikipedia.org/wiki/C%2B%2B) , [HTML5](https://en.wikipedia.org/wiki/HTML5) , [[4]](https://en.wikipedia.org/wiki/NetBeans#cite_note-4) và [JavaScript](https://en.wikipedia.org/wiki/JavaScript) . Các ứng dụng dựa trên NetBeans, bao gồm NetBeans IDE, có thể được các nhà phát triển bên thứ ba mở rộng.

### NetBeans IDE phát hành :

NetBeans IDE 6.0 đã giới thiệu hỗ trợ phát triển các mô-đun IDE và các ứng dụng khách phong phú dựa trên nền tảng NetBeans, [trình](https://en.wikipedia.org/wiki/GUI) xây dựng [GUI](https://en.wikipedia.org/wiki/GUI) Java Swing (trước đây gọi là "Project Matisse"), hỗ trợ [CVS](https://en.wikipedia.org/wiki/Concurrent_Versions_System) cải tiến , hỗ trợ [WebLogic](https://en.wikipedia.org/wiki/WebLogic) 9 và [JBoss](https://en.wikipedia.org/wiki/JBoss) 4 và nhiều cải tiến trình soạn thảo.

NetBeans IDE 6.5, được phát hành vào tháng 11 năm 2008, đã mở rộng các tính năng [Java EE hiện](https://en.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition) có (bao gồm hỗ trợ tính bền vững của Java, EJB 3 và JAX-WS). Ngoài ra, Gói doanh nghiệp NetBeans hỗ trợ phát triển các ứng dụng doanh nghiệp Java EE 5, bao gồm [các](https://en.wikipedia.org/wiki/Service-oriented_architecture) công cụ thiết kế trực quan [SOA](https://en.wikipedia.org/wiki/Service-oriented_architecture) , công cụ lược đồ XML, phối hợp dịch vụ web (cho BPEL) và mô hình hóa [UML](https://en.wikipedia.org/wiki/Unified_Modeling_Language) .

NetBeans IDE 6.8 là IDE đầu tiên cung cấp hỗ trợ đầy đủ cho Java EE 6 và [GlassFish Enterprise Server v3](https://en.wikipedia.org/wiki/GlassFish) . Các nhà phát triển lưu trữ các dự án nguồn mở của họ trên [kenai.com](https://en.wikipedia.org/wiki/Project_Kenai)cũng được hưởng lợi từ việc tích hợp và điều hướng theo dõi vấn đề ngay lập tức trong IDE, hỗ trợ phát triển ứng dụng web với PHP 5.3 và khung Symfony và cải thiện hoàn thiện mã, bố cục, gợi ý và điều hướng trong các dự án JavaFX.

NetBeans IDE 6.9, được phát hành vào tháng 6 năm 2010, đã thêm hỗ trợ cho [OSGi](https://en.wikipedia.org/wiki/OSGi) , [Spring Framework](https://en.wikipedia.org/wiki/Spring_Framework) 3.0, tiêm phụ thuộc Java EE (JSR-299), [Zend Framework](https://en.wikipedia.org/wiki/Zend_Framework) cho [PHP](https://en.wikipedia.org/wiki/PHP) và điều hướng mã dễ dàng hơn (như chú thích "Bị ghi đè / triển khai"), định dạng , gợi ý và tái cấu trúc trên một số ngôn ngữ.

NetBeans IDE 7.0 được phát hành vào tháng 4 năm 2011. Vào ngày 1 tháng 8 năm 2011, Nhóm NetBeans đã phát hành NetBeans IDE 7.0.1, hỗ trợ đầy đủ cho việc phát hành chính thức nền tảng [Java SE 7](https://en.wikipedia.org/wiki/Java_version_history#Java_SE_7_(July_28,_2011)).

NetBeans IDE 7.3 được phát hành vào tháng 2 năm 2013 có thêm hỗ trợ cho các công nghệ HTML5 và web. [[13]](https://en.wikipedia.org/wiki/NetBeans#cite_note-13)

NetBeans IDE 7.4 đã được phát hành vào ngày 15 tháng 10 năm 2013.

NetBeans IDE 8.0 đã được phát hành vào ngày 18 tháng 3 năm 2014.

NetBeans IDE 8.1 đã được phát hành vào ngày 4 tháng 11 năm 2015.

NetBeans IDE 8.2 đã được phát hành vào ngày 3 tháng 10 năm 2016.

Netbeans 9.0, bổ sung hỗ trợ cho Java 9 và 10, được phát hành vào ngày 29 tháng 7 năm 2018, bởi dự án Vườn ươm Apache. [[10]](https://en.wikipedia.org/wiki/NetBeans#cite_note-incubator-10)[[11]](https://en.wikipedia.org/wiki/NetBeans#cite_note-roadmap-11)[[14]](https://en.wikipedia.org/wiki/NetBeans#cite_note-jax-incubate-14)[[15]](https://en.wikipedia.org/wiki/NetBeans#cite_note-15)

NetBeans 10.0 được phát hành vào ngày 27 tháng 12 năm 2018. Nó mang đến sự hỗ trợ cho Java 11 và cải thiện hỗ trợ cho PHP (7.0 .77.3).

NetBeans 11.0 được phát hành vào ngày 4 tháng 4 năm 2019.

Giới thiệu Microsoft SQL Server :

#### Lịch sử ra đời :

Phiên bản đầu tiên của Microsoft SQL Server ra đời đầu tiên vào năm 1989 cho các hệ điều hành chạy 16 bit với SQL Server phiên bản 1.0 và tiếp tục phát triển cho tới ngày nay.

SQL Server của Microsoft được thị trường chấp nhận rộng rãi kể từ version 6.5. Sau đó Microsoft đã cải tiến và hầu như viết lại một engine mới cho SQL Server 7.0. Cho nên có thể nói từ version 6.5 lên version 7.0 là một  bước nhảy vọt. Có một số đặc tính của SQL Server 7.0 không tương thích với version 6.5. Trong khi đó từ Version 7.0 lên version 8.0 (SQL Server 2000) thì những cải tiến chủ yếu là mở rộng các tính năng về web và làm cho SQL Server 2000 đáng tin cậy hơn.

Một điểm đặc biệt đáng lưu ý ở phiên bản 2000 là **Multiple-Instance**. Tức là bạn có thể cài đặt phiên bản 2000 chung với các phiên bản trước mà không cần phải gỡ chúng. Nghĩa là bạn có thể chạy song song version 6.5 hoặc 7.0 với phiên bản 2000 trên cùng một máy (điều này không thể xảy ra với các phiên bản trước đây). Khi đó phiên bản cũ trên máy bạn là **Default Instance** còn phiên bản 2000 mới vừa cài sẽ là **Named Instance**.

Từ tháng 10 năm 2016, các phiên bản sau được Microsoft hỗ trợ:

* SQL Server 2008 R2
* SQL Server 2012
* SQL Server 2014
* SQL Server 2016

Phiên bản hiện tại là Microsoft SQL Server 2016, xuất bản vào ngày 1/6/2016.

SQL Server 2016 chỉ hỗ trợ cho các bộ vi xử lý 64 bit.

#### Giới thiệu :

QL Server là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System (RDBMS) ) sử dụng câu lệnh SQL (**Transact-SQL)**để trao đổi dữ liệu giữa máy Client và máy cài SQL Server. Một RDBMS bao gồm databases, database engine và các ứng dụng dùng để quản lý dữ liệu và các bộ phận khác nhau trong RDBMS.

SQL Server được tối ưu để có thể chạy trên môi trường cơ sở dữ liệu rất lớn (Very Large Database Environment) lên đến Tera-Byte và có thể phục vụ cùng lúc cho hàng ngàn user. SQL Server có thể kết hợp “ăn ý” với các server khác như Microsoft Internet Information Server (IIS), E-Commerce Server, Proxy Server….

Một vài ấn bản SQL Server:

* **Enterprise** : chứa tất cả cá đặc điểm nổi bật của SQL Server, bao gồm nhân bộ máy cơ sở dữ liệu và các dịch vụ đi kèm cùng với các công cụ cho tạo và quản lý phân cụm SQL Server. Nó có thể quản lý các CSDL lớn tới 524 [petabytes](https://en.wikipedia.org/wiki/Petabyte) và đánh địa chỉ 12 [terabytes](https://en.wikipedia.org/wiki/Terabyte) bộ nhớ và hỗ trợ tới 640 bộ vi xử lý(các core của cpu)
* **Standard** : Rất thích hợp cho các công ty vừa và nhỏ vì giá thành rẻ hơn nhiều so với Enterprise Edition, nhưng lại bị giới hạn một số chức năng cao cấp (advanced features) khác, edition này có thể chạy tốt trên hệ thống lên đến 4 CPU và 2 GB RAM.
* **Developer** : Có đầy đủ các tính năng của Enterprise Edition nhưng được chế tạo đặc biệt như giới hạn số lượng người kết nối vào Server cùng một lúc…. Ðây là phiên bản sử dụng cho phát triển và kiểm tra ứng dụng. Phiên bản này phù hợp cho các cá nhân, tổ chức xây dựng và kiểm tra ứng dụng
* **Workgroup**: ấn bản SQL Server Workgroup bao gồm chức năng lõi cơ sở dữ liệu nhưng không có các dịch vụ đi kèm. Chú ý phiên bản này không còn tồn tại ở SQL Server 2012.
* **Express**: SQL Server Express dễ sử dụng và quản trị cơ sở dữ liệu đơn giản. Được tích hợp với Microsoft Visual Studio, nên dễ dàng để phát triển các ứng dụng dữ liệu, an toàn trong lưu trữ, và nhanh chóng triển khai. SQL Server Express là phiên bản miễn phí,  không giới hạn về số cơ sở dữ liệu hoặc người sử dụng, nhưng nó chỉ dùng cho 1 bộ vi xử lý với 1 GB bộ nhớ và 10 GB file cơ sở dữ liệu. SQL Server Express là lựa chọn tốt cho những người dùng chỉ cần một phiên bản SQL Server 2005 nhỏ gọn, dùng trên máy chủ có cấu hình thấp, những nhà phát triển ứng dụng không chuyên hay những người yêu thích xây dựng các ứng dụng nhỏ.

#### Cài đặt :

Cần có phiên bản **Developer Edition** và ít nhất là 64 MB RAM, 500 MB hard disk để có thể cài SQL Server. Bạn có thể cài trên Windows Server hay Windows 10, Windows 8, Windows 7, Windows XP …

Các bước cài đặt không quá khó khăn với các phiên bản SQL Server khác nhau, bạn có thể xem lại bài [Hướng dẫn cài đặt SQL Server 2008](http://timoday.edu.vn/huong-dan-cai-dat-va-su-dung-sql-server-2008-va-northwind-database/). Khi cài đặt bạn cần lưu ý các điểm sau:

Ở màn hình thứ hai bạn chọn **Install Database Server**. Sau khi install xong SQL Server bạn có thể install thêm Analysis Service nếu bạn thích.

Ở màn hình **Installation Definition**bạn chọn **Server and Client Tools**.

Sau đó bạn nên chọn kiểu **Custom** và **chọn tất cả** các bộ phận của SQL Server. Ngoài ra nên **chọn các giá trị mặc định**(default)

Ở màn hình **Authentication Mode** nhớ chọn **Mixed Mode**. Lưu ý vì SQL Server có thể dùng chung chế độ bảo mật (security) với hệ điều hành Windows và cũng có thể dùng chế độ bảo mật riêng của nó.  Trong Production Server người ta thường dùng Windows Authetication vì độ an toàn cao hơn và dễ dàng cho người quản lý mạng và cả cho người sử dụng. Nghĩa là một khi bạn được chấp nhận (authenticated) kết nối vào domain thì bạn có quyền truy cập dữ liệu (access data) trong SQL Server. Tuy nhiên ta nên chọn Mixed Mode để dễ dàng cho việc học tập.

Sau khi install bạn sẽ thấy một icon nằm ở góc phải bên dưới màn hình, đây chính là Service Manager. Bạn có thể Start, Stop các SQL Server services dễ dàng bằng cách double-click vào icon này.

#### Các thành phần cơ bản :

SQL Server được cấu tạo bởi nhiều thành phần như Database Engine, Reporting Services, Notification Services, Integration Services, Full Text Search Service…. Các thành phần này khi phối hợp với nhau tạo thành một giải pháp hoàn chỉnh giúp cho việc lưu trữ và phân tích dữ liệu một cách dễ dàng.

[](http://timoday.edu.vn/wp-content/uploads/2016/11/ThanhPhanSQLServer.gif)

*Các thành phần chính SQL Server*

**Database Engine**

Lõi của SQL Server:

Ðây là một engine có khả năng chứa data ở các quy mô khác nhau dưới dạng table và support tất cả các kiểu kết nối (data connection) thông dụng của Microsoft như

ActiveX Data Objects (ADO), OLE DB, and Open Database Connectivity (ODBC).

Ngoài ra nó còn có khả năng tự điều chỉnh (tune up) ví dụ như sử dụng thêm các tài nguyên (resource) của máy khi cần và trả lại tài nguyên cho hệ điều hành khi một user log off.

**Replication**

Cơ chế tạo bản sao (Replica):

Giả sử bạn có một database dùng để chứa dữ liệu được các ứng dụng thường xuyên cập nhật. Một ngày đẹp trời bạn muốn có một cái database giống y hệt như thế trên một server khác để chạy báo cáo (report database) (cách làm này thường dùng để tránh ảnh hưởng đến performance của server chính). Vấn đề là report server của bạn cũng cần phải được cập nhật thường xuyên để đảm bảo tính chính xác của các báo cáo. Bạn không thể dùng cơ chế back up and restore trong trường hợp này. Thế thì bạn phải làm sao? Lúc đó cơ chế replication của SQL Server sẽ được sử dụng để bảo đảm cho dữ liệu ở 2 database được đồng bộ (synchronized).

**Integration Services (DTS)**

Integration Services là một tập hợp các công cụ đồ họa và các đối tượng lập trình cho việc di chuyển, sao chép và chuyển đổi dữ liệu.

Nếu bạn làm việc trong một công ty lớn trong đó data được chứa trong nhiều nơi khác nhau và ở các dạng khác nhau cụ thể như chứa trong Oracle, DB2 (của IBM), SQL Server, Microsoft Access….Bạn chắc chắn sẽ có nhu cầu di chuyển data giữa các server này (migrate hay transfer) và không chỉ di chuyển bạn còn muốn định dạng (format) nó trước khi lưu vào database khác, khi đó bạn sẽ thấy DTS giúp bạn giải quyết công việc trên dễ dàng.

**Analysis Services**

Một dịch vụ phân tích dữ liệu rất hay của Microsoft

Dữ liệu (Data) chứa trong database sẽ chẳng có ý nghĩa gì nhiều nếu như bạn không thể lấy được những thông tin (Information) bổ ích từ đó. Do đó Microsoft cung cấp cho bạn một công cụ rất mạnh giúp cho việc phân tích dữ liệu trở nên dễ dàng và hiệu quả bằng cách dùng khái niệm hình khối nhiều chiều (multi-dimension cubes) và kỹ thuật “khai phá dữ liệu” (data mining).

**Notification Services**

Dịch vụ thông báo Notification Services là nền tảng cho sự phát triển và triển khai các ứng dụng tạo và gửi thông báo. Notification Services có thể gửi thông báo theo địch thời đến hàng ngàn người đăng ký sử dụng nhiều loại thiết bị khác nhau.

**Reporting  Services**

Reporting Services bao gồm các thành phần server và client cho việc tạo, quản lý và triển khai các báo cáo. Reporting Services cũng là nền tảng cho việc phát triển và xây dựng các ứng dụng báo cáo.

**Full Text Search Service**

Dịch vụ SQL Server Full Text Search là một dịch vụ đặc biệt cho đánh chỉ mục và truy vấn cho dữ liệu văn bản không cấu trúc được lưu trữ trong các CSDL SQL Server. Đánh chỉ mục với Full Text Search có thể dduwowcj tạo trên bất kỳ cột dựa trên dữ liệu văn bản. Nó sẽ rất hiệu quả cho việc tìm các sử dụng toán tử LIKE trong SQL với trường hợp tìm văn bản.

**Service Broker**

Được sử dụng bên trong mỗi Instance, là môi trường lập trình cho việc các ứng dụng nhảy qua các Instance. Service Broker giao tiếp qua giao thức TCP/IP và cho phép các component khác nhau có thể được đồng bộ cùng nhau theo hướng trao đổi các message. Service Broker chạy như một phần của bộ máy cơ sở dữ liệu, cung cấp một nền tảng truyền message tin cậy và theo hàng đợi cho các ứng dụng SQL Server.

## Thực hiện bài toán

### Phân công công việc

Nội dung trình bày ở đây : Nêu các chức năng sẽ cài đặt và nội dung phân công cho từng thành viên trong nhóm.

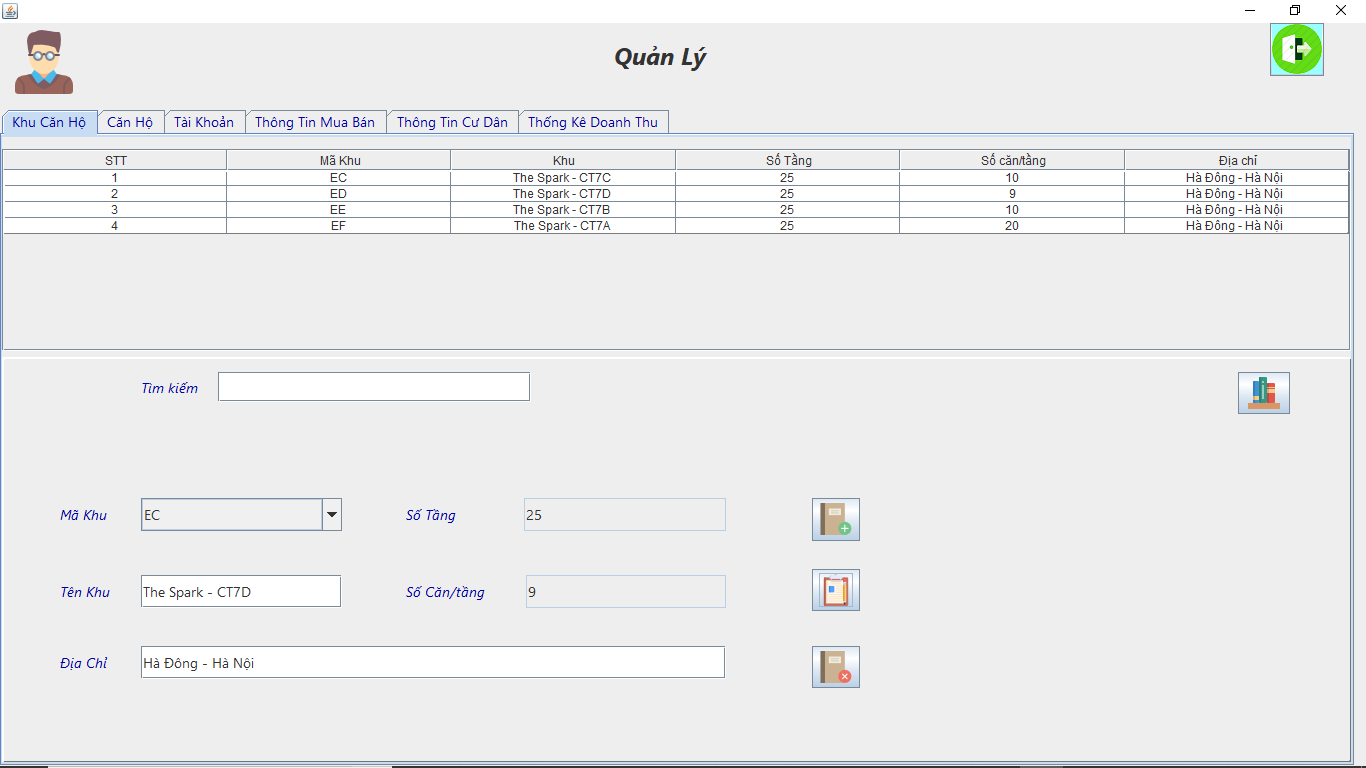
(Kiểu Normal - Font Times New Roman 13, justified)

|  |  |
| --- | --- |
| Tên sinh viên | Tên công việc |
|  |  |

### Phạm Văn Đại – Thực hiện form Quản Lý với các nhóm chức năng Quản lý khu căn hộ, Quản lý căn hộ, Quản lý tài khoản.

#### Giao diện và mô tả các đối tượng trên giao diện:

1. Giao diện nhóm chức năng quản lý khu căn hộ, căn hộ :



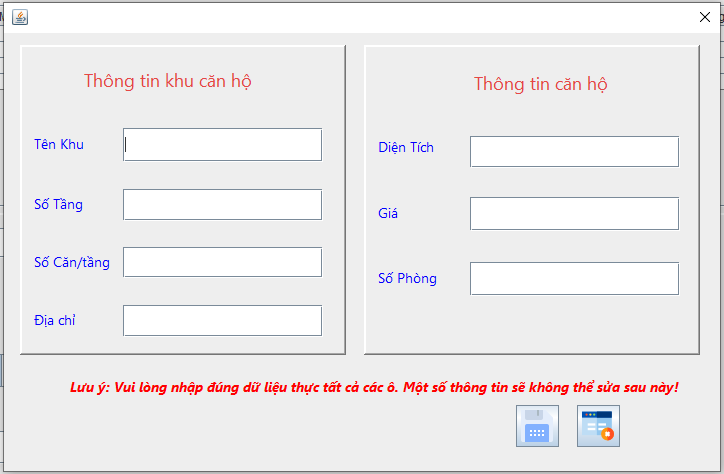
Hình 2.1 Giao diện nhóm chức năng quản lý khu căn hộ, căn hộ

* JButton  (Đăng xuất): Đăng xuất khỏi hệ thống.
* JTable : Hiển thị thông tin các khu căn hộ
* JButton  (Xem tất cả) : Xem tất cả thông tin của các khu căn hộ
* JTextField (Tìm kiếm) : Tìm kiếm thông tin khu căn hộ theo từ khóa mà người dùng nhập.
* JButton  (Thêm) : Thêm 1 khu căn hộ và các căn hộ mới trong khu.

Click mở giao diện thêm khu (*Hình 2.2*).

* JButton  (Sửa) : Sửa lại thông tin khu căn hộ sai sót.
* JButton  (Xóa) : Xóa khu căn hộ khi cần thiết. Click mở giao diện xác nhận (*Hình 2.3*).
* Jcombobox(Mã Khu), JTextField (Tên Khu, Địa Chỉ, Số Tầng, Số Căn/tầng) : Hiển thị thông tin của khu căn hộ người dùng chọn.

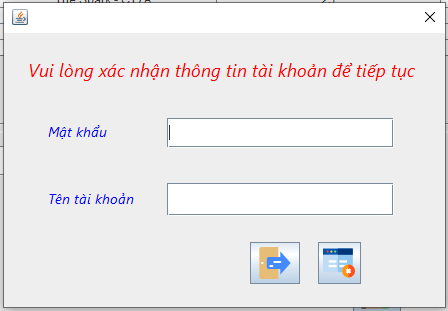
1. Giao diện chức năng thêm khu, căn hộ mới



Hình 2.2 Giao diện chức năng thêm khu, căn hộ mới

* JPanel (Thông tin khu căn hộ) : Nhập thông tin của khu căn hộ gồm các thông tin Tên Khu, Số Tầng, Số Căn/ tầng, Địa Chỉ.
* JPanel (Thông tin căn hộ) : Nhập thông tin chung cho các căn hộ trong khu.
* JButton  (Lưu) : Kiểm tra dữ liệu ở 2 panel trên. Nếu hợp lệ lưu vào csdl. Không hợp lệ hiện thông báo lỗi.
* JButton  (Hủy) : Hủy bỏ tất cả các thao tác đang thực hiện và đóng giao diện.

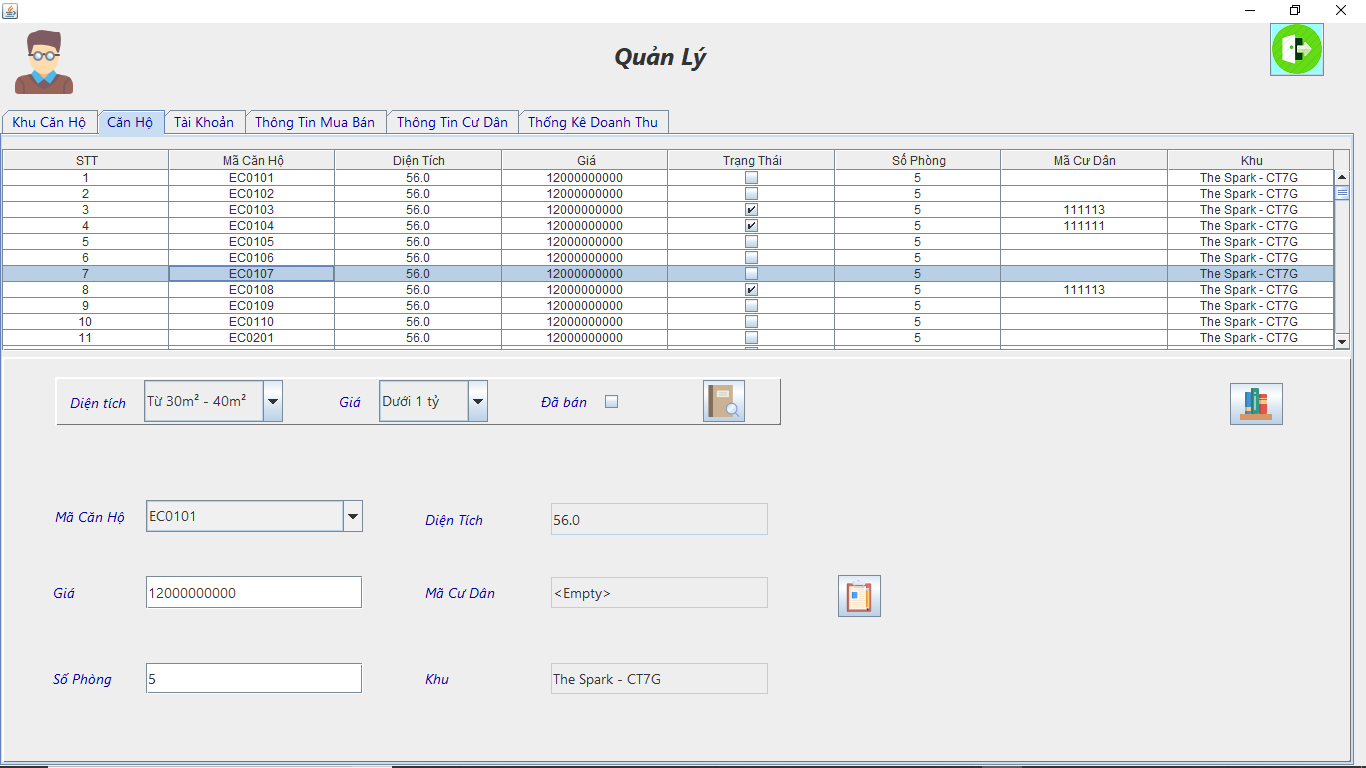
1. Giao diện xác nhận thông tin tài khoản :



Hình 2.3 Giao diện xác nhận thông tin tài khoản

* JTextField (Tên tài khoản, Mật khẩu) : Nhập thông tin tài khoản
* JButton  (Lưu) : Kiểm tra dữ liệu ở 2 textfield trên. Nếu đúng cho phép tiếp tục. Không đúng hiện thông báo lỗi.
* JButton  (Hủy) : Hủy bỏ tất cả các thao tác đang thực hiện và đóng giao diện.

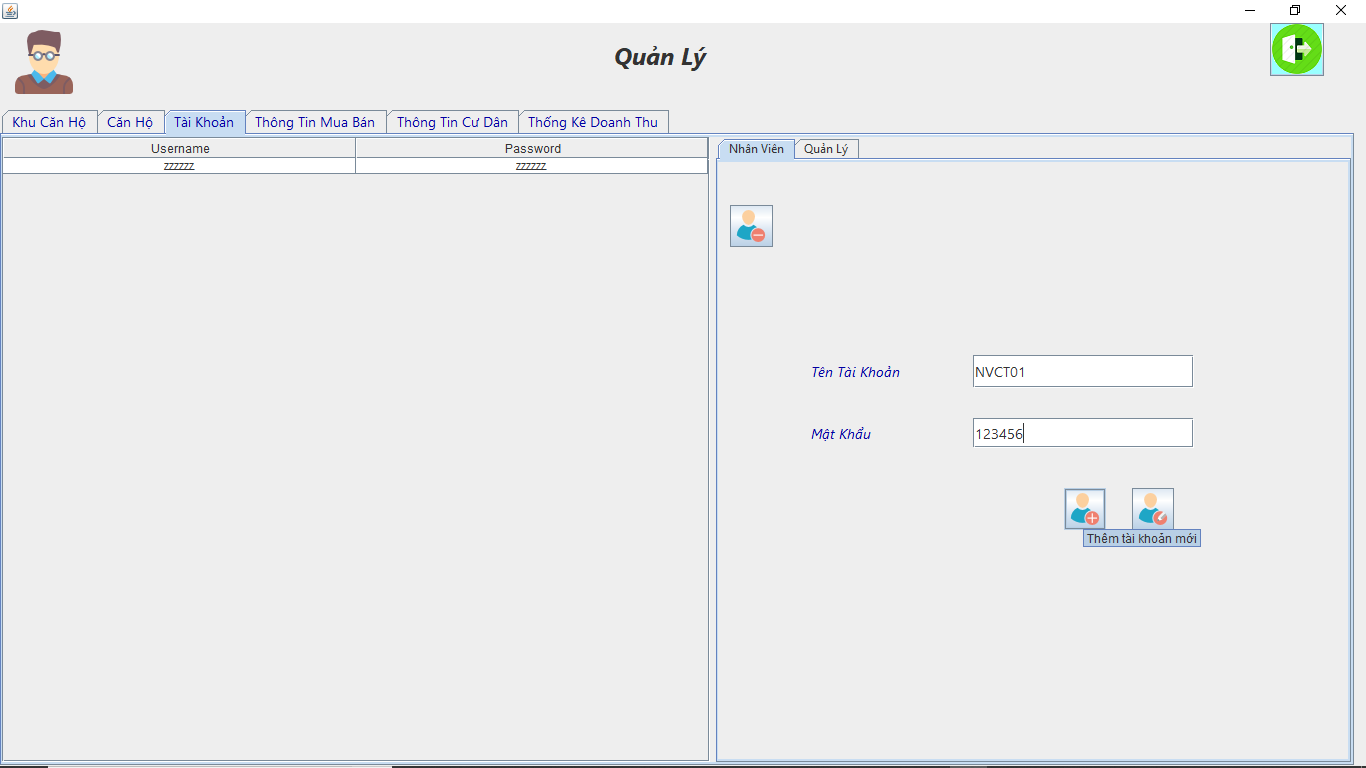
1. Giao diện chức năng sửa thông tin căn hộ(tab ‘Căn Hộ’) :



Hình 2.4 Giao diện chức năng sửa căn hộ

* JButton  (Xem tất cả) : Xem tất cả thông tin của các căn hộ
* Jcombobox (Diện tích, Giá), Checkbox (Đã bán) : Tiêu chí để người dùng lựa chọn khi tìm kiếm.
* JButton  (Tìm kiếm) : Tìm kiếm theo các tiêu chí người dùng muốn.
* Jcombobox (Mã Căn Hộ), JTextField (Diện Tích, Giá, Mã Cư Dân, Số Phòng, Khu) : Hiển thị thông tin người dùng chọn.
* JButton  (Sửa) : Sửa lại thông tin căn hộ sai sót.

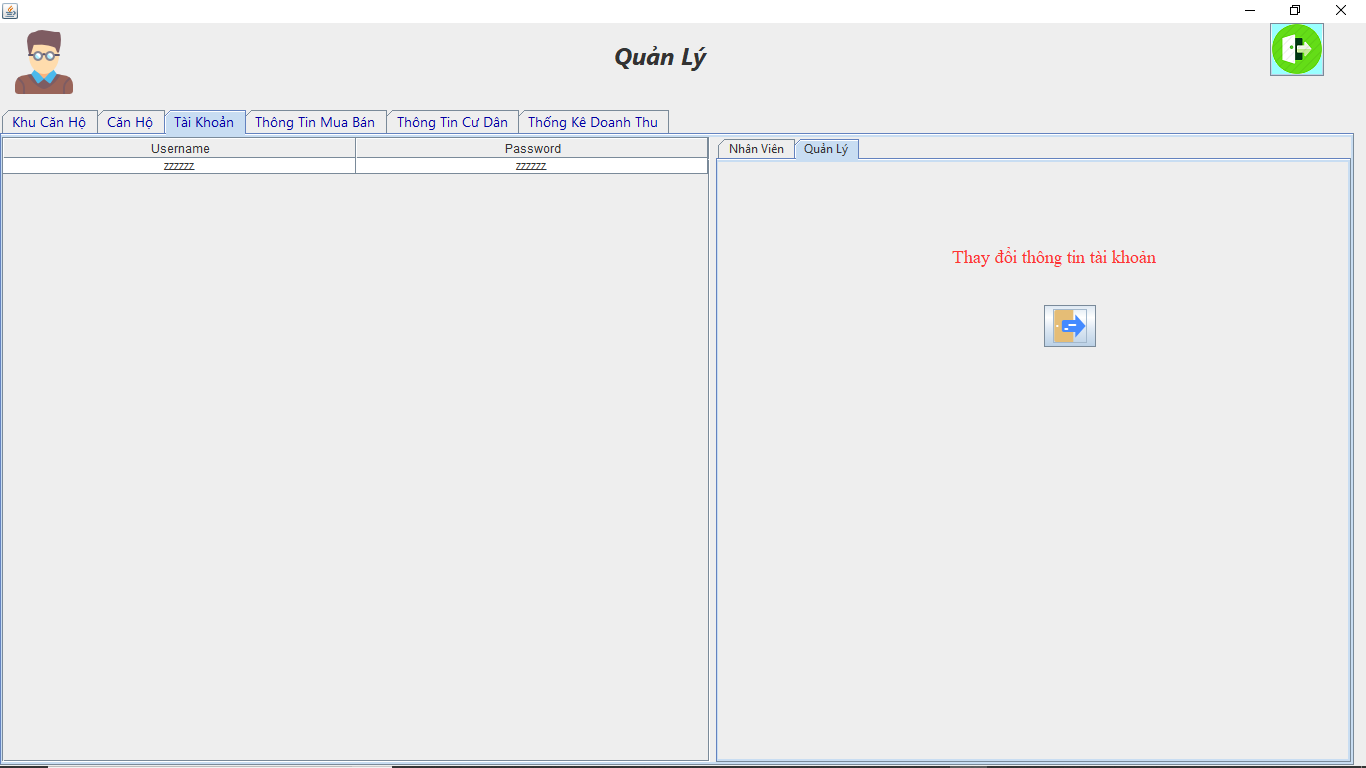
1. Giao diện nhóm chức năng quản lý tài khoản nhân viên(tab ‘Tài Khoản’, tab con ‘Nhân Viên’) :



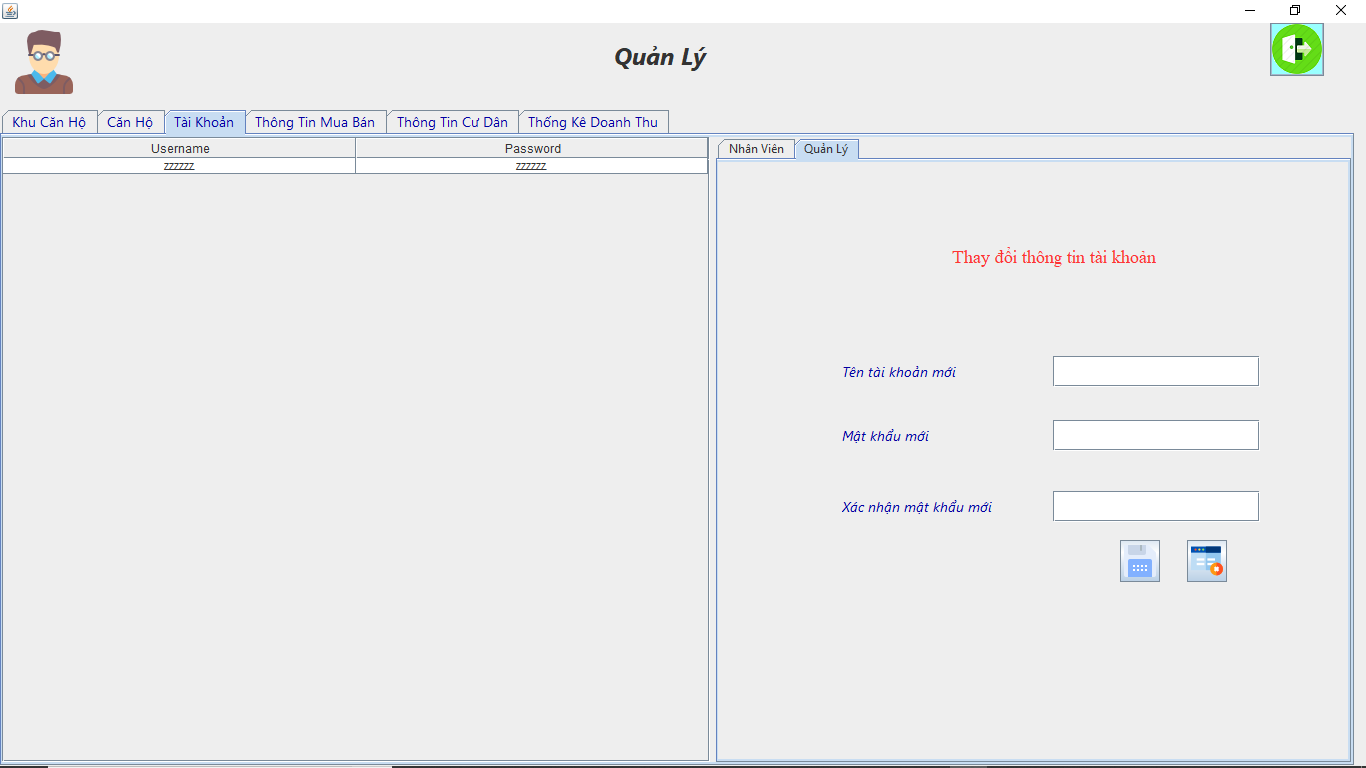
Hình 2.5 Giao diện nhóm chức năng quản lý tài khoản nhân viên

* JTable : Hiển thị thông tin tài khoản của nhân viên
* JButton  (Xóa) : Xóa tài khoản nhân viên đã chọn.
* JButton  (Thêm) : Thêm 1 tài khoản nhân viên mới.
* JButton  (Sửa) : Sửa mật khẩu tài khoản nhân viên đã chọn.
* JTextField (Tên Tài Khoản, Mật khẩu) : Nhập dữ liệu cho tài khoản.

1. Giao diện thay đổi thông tin tài khoản cho quản lý :



Hình 2.6 Giao diện thay đổi thông tin tài khoản quản lý

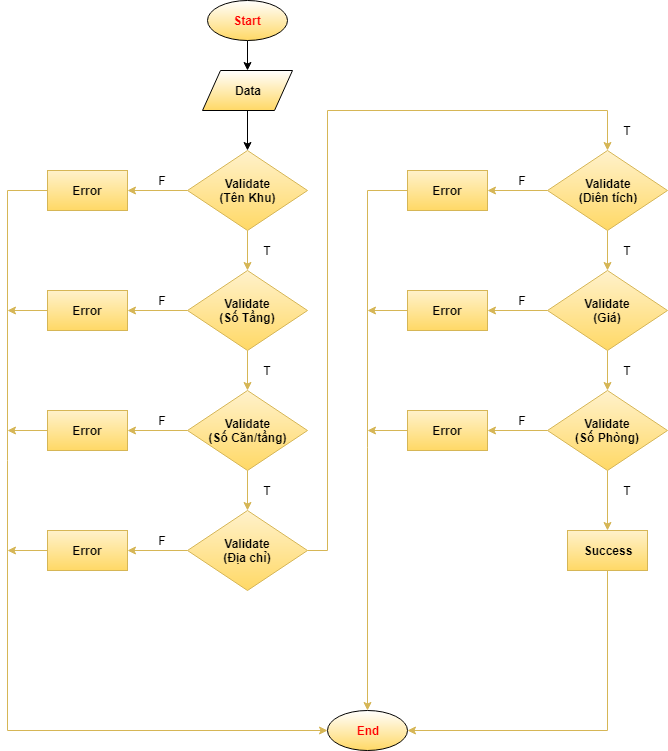


Hình 2.7 Giao diện lưu thông tin tài khoản mới.

* JButton  (Tiếp tục) : Mở giao diện xác nhận tài khoản mở giao diện xác nhận (*Hình 2.3*).
* JTextField (Tên tài khoản mới), JPaswordField (Mật khẩu mới, Xác nhận mật khẩu mới) : Nhập và xác nhận thông tin tài khoản mới.
* JButton  (Lưu) : Kiểm tra dữ liệu ở 3 ô text trên. Nếu đúng cho phép thay đổi thông tin tài khoản. Không đúng hiện thông báo lỗi.
* JButton  (Hủy) : Hủy bỏ tất cả các thao tác đang thực hiện.

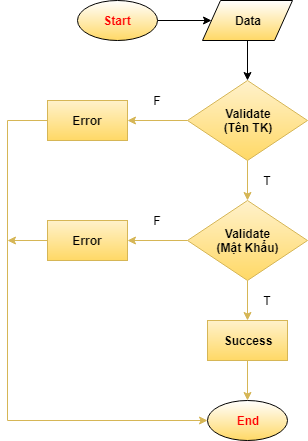
#### Kiểm tra tính hợp lệ của dữ liệu đầu vào (Algorithm Flowchart) :

1. Kiểm tra thông tin đầu vào khu căn hộ, căn hộ :



Hình 2.8 Lưu đồ thuật toán kiểm tra hợp lệ dữ liệu thông tin khu căn hộ, căn hộ

1. Kiểm tra thông tin tài khoản:

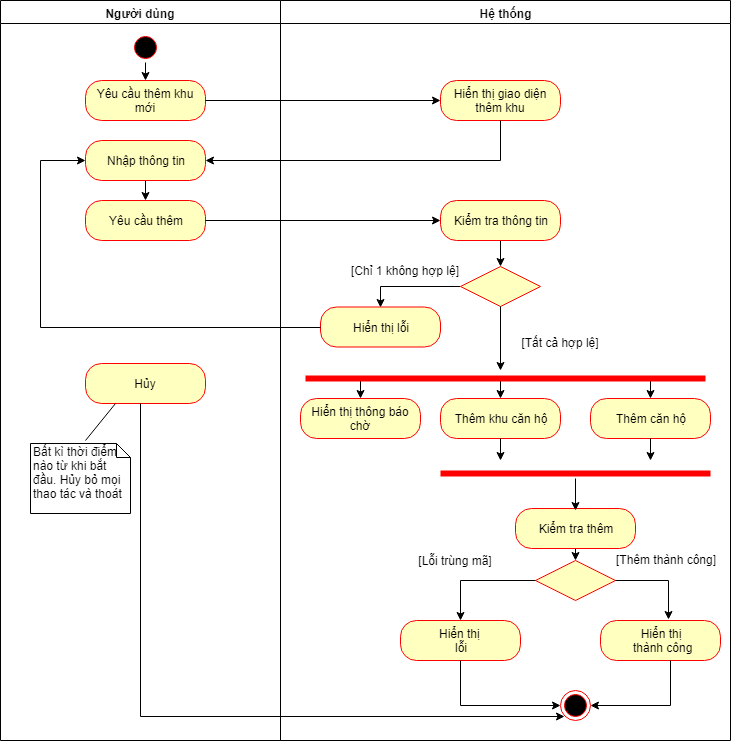


Hình 2.9 Lưu đồ thuật toán kiểm tra hợp lệ dữ liệu thông tin tài khoản

#### Xử lý thao tác người dùng(Activity Diagram) :

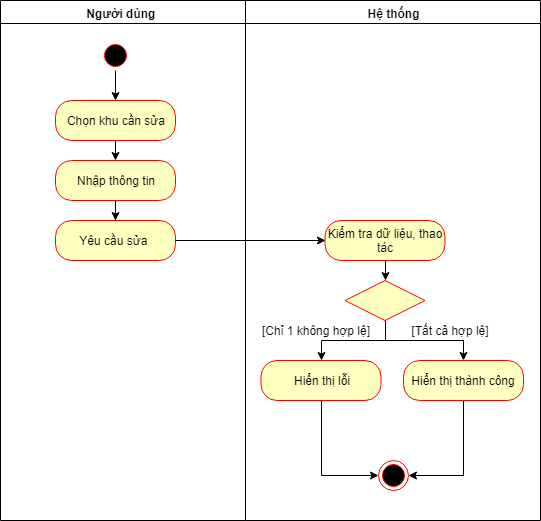
1. Tab ‘Khu Căn Hộ’ :

* Chức năng thêm khu căn hộ, căn hộ :



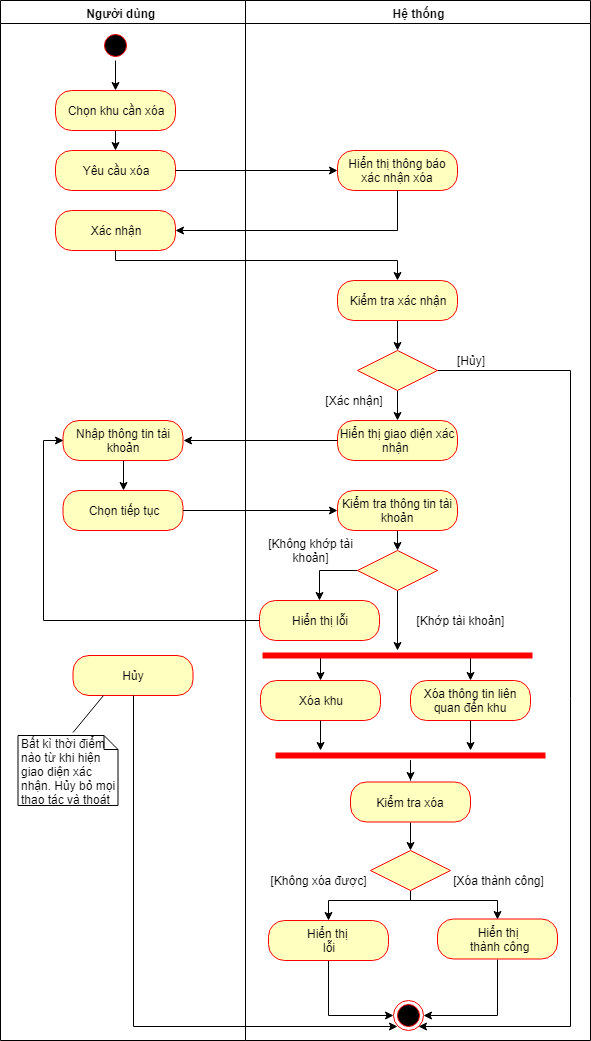
Hình 2.10 Biểu đồ hoạt động của chức năng thêm khu căn hộ, căn hộ

* Chức năng sửa thông tin khu :



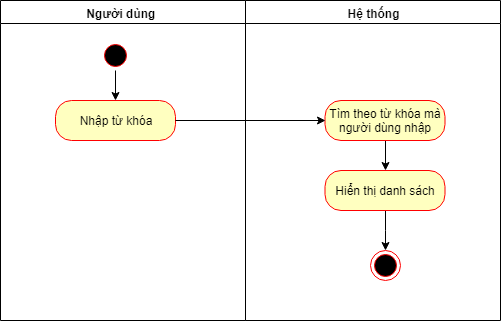
Hình 2.11 Biểu đồ hoạt động của chức năng sửa khu

* Chức năng xóa khu :



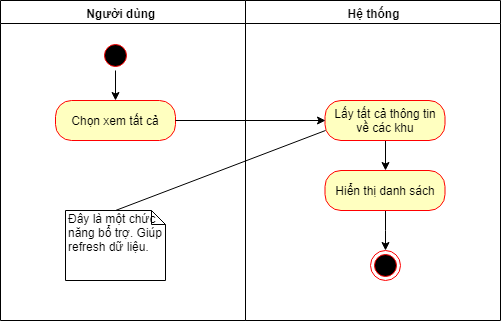
Hình 2.12 Biểu đồ hoạt động chức năng xóa khu

* Chức năng tìm kiếm khu :



Hình 2.13 Biểu đồ hoạt động của chức năng tìm kiếm khu căn hộ

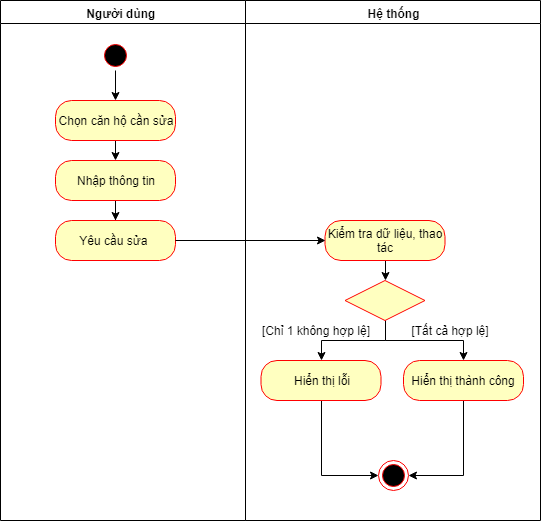
* Chức năng xem tất cả khu:



Hình 2.14 Biểu đồ hoạt động của chức năng xem tất cả khu

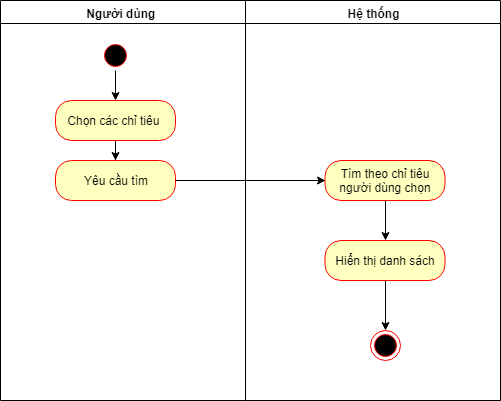
1. Tab ‘Căn Hộ’ :

* Chức năng sửa căn hộ :



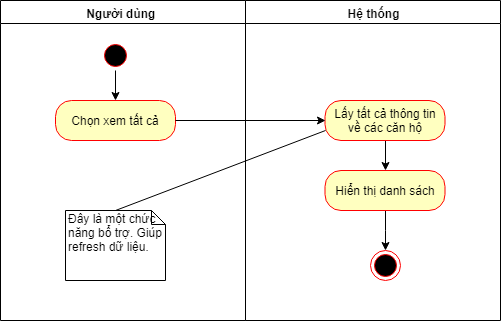
Hình 2.15 Biểu đồ hoạt động của chức năng sửa căn hộ

* Chức năng tìm kiếm căn hộ :



Hình 2.16 Biểu đồ hoạt động của chức năng tìm kiếm căn hộ

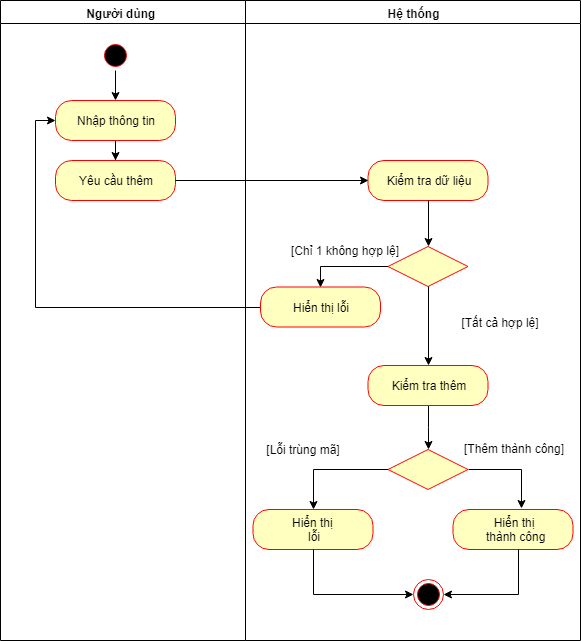
* Chức năng xem tất cả căn hộ:



Hình 2.17 Biểu đồ hoạt động của chức năng xem tất cả căn hộ

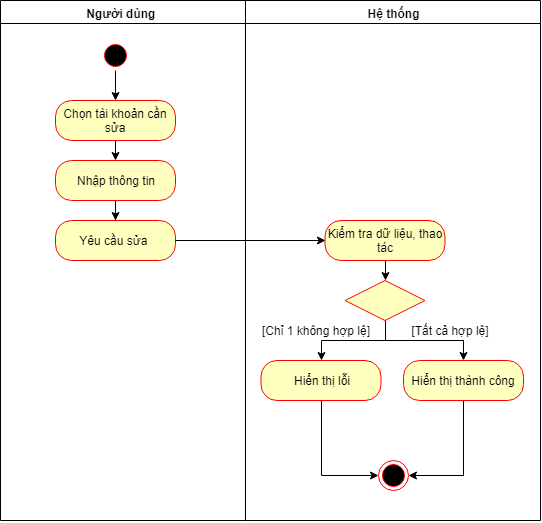
1. Tab ‘Tài Khoản’ :

* Chức năng thêm tài khoản nhân viên :



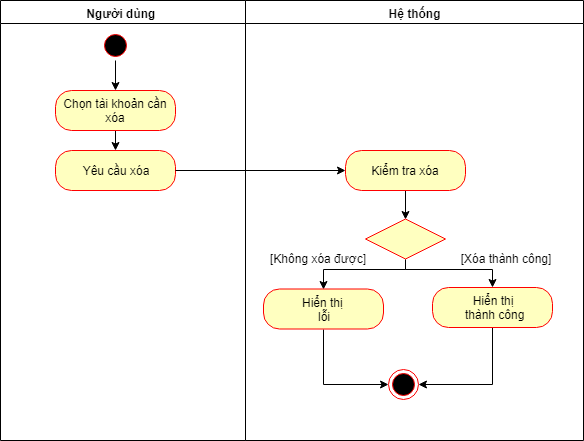
Hình 2.18 Biểu đồ hoạt động của chức năng thêm tài khoản nhân viên

* Chức năng sửa tài khoản nhân viên :



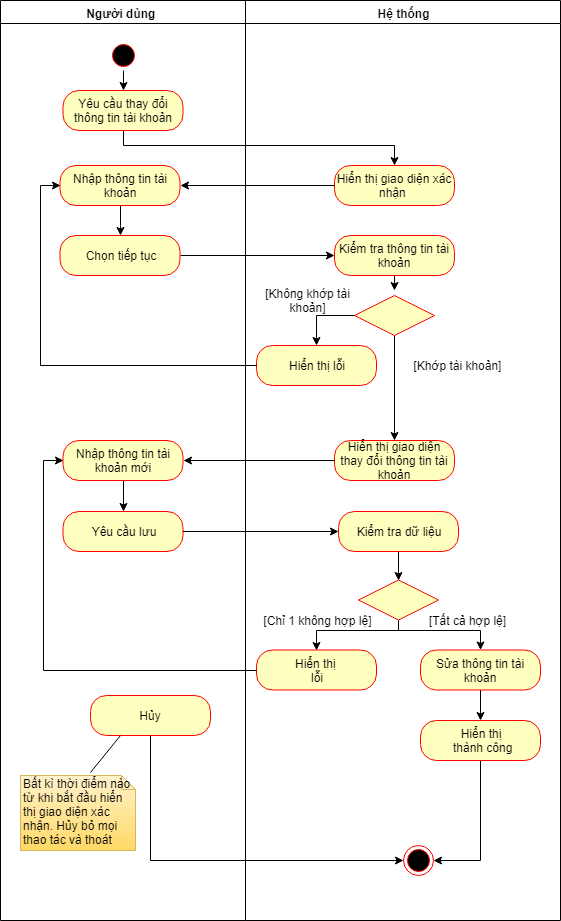
Hình 2.19 Biểu đồ hoạt động của chức năng sửa tài khoản nhân viên

* Chức năng xóa tài khoản nhân viên:



Hình 2.20 Biểu đồ hoạt động của chức năng xóa tài khoản nhân viên

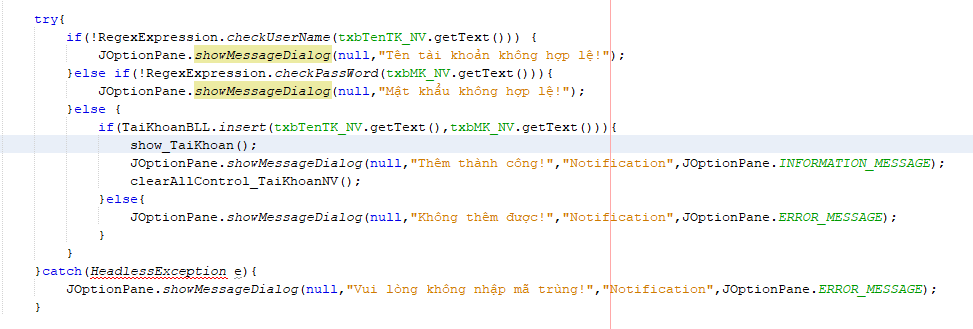
* Chức năng thay đổi thông tin tài khoản người quản lý :



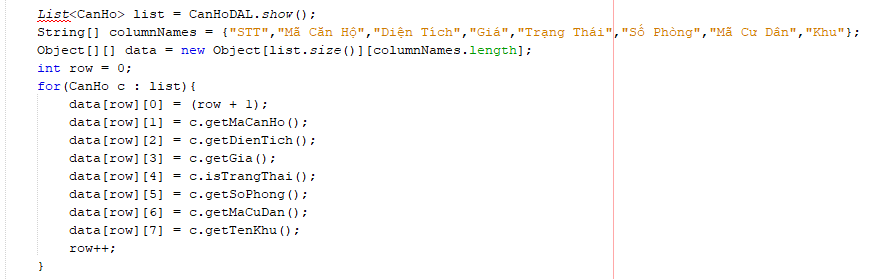
Hình 2.21 Biểu đồ hoạt động của chức năng thay đổi thông tin tài khoản quản lý

#### Sử dụng một số thư viện chuẩn :

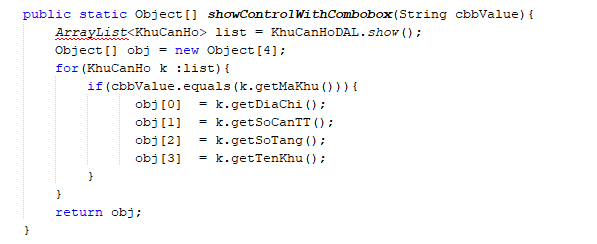
* ***java.awt*** :



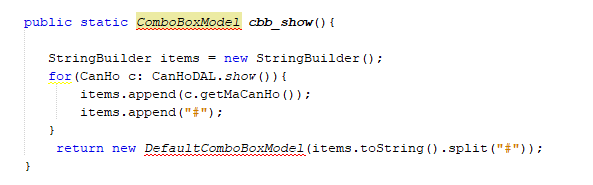
* *java.awt.HeadlessException : Package này hỗ trợ việc bắt ngoại lệ xảy ra do việc nhập dữ liệu không hợp lệ, cụ thể ở đây là trùng mã.*
* ***java.util*** :



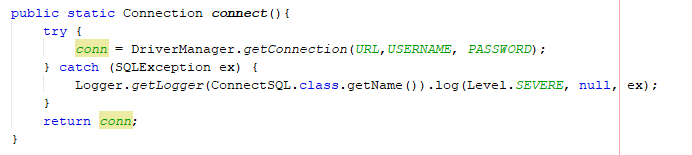
* *java.util.List : Package này hỗ trợ làm việc với List. Cụ thể ở đây là khai báo một đối tượng kiểu List <CanHo>*



* *java.util.ArrayList : Package này hỗ trợ làm với ArrayList. Cụ thể ở đây là khai báo một đối tượng kiểu ArrayList <KhuCanHo>*
* ***java.swing*** :



* *javax.swing.ComboBoxModel && javax.swing.DefaultComboBoxModel : 2 package này hỗ trợ làm việc với Combobox. Cụ thể ở đây là khai báo kiểu trả về cho hàm cbb\_show() và trả về giá trị hàm.*
* ***java.sql*** :



* *java.sql.Connection && java.sql.DriverManager && java.sql.SQLException : 3 package này hỗ trợ kết nối và làm việc với DataBase. Cụ thể đây là kết nối và bắt ngoại lệ khi kết nối.*

### Hoàng Văn Độ - FORM đăng nhập hệ thống , FORM Quản lý TAB Thông tin mua bán, thông tin cư dân,Thống kê doanh thu .

#### Đăn nhập hệ thống.

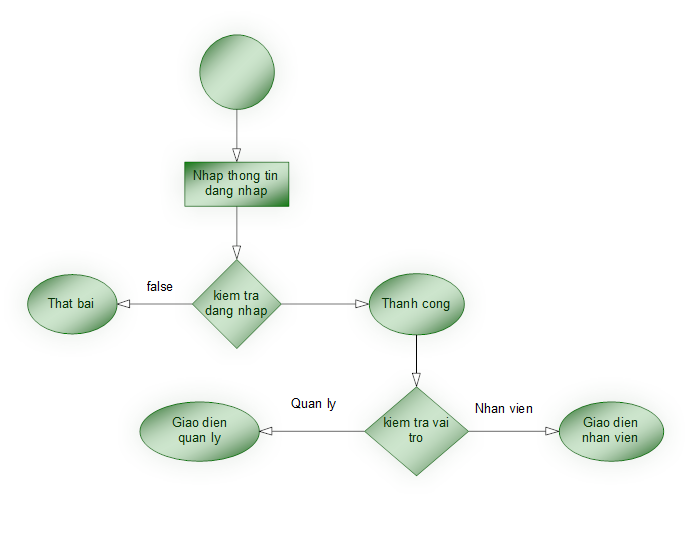
#### 

Hình 3.1 Form đăng nhập hệ thống

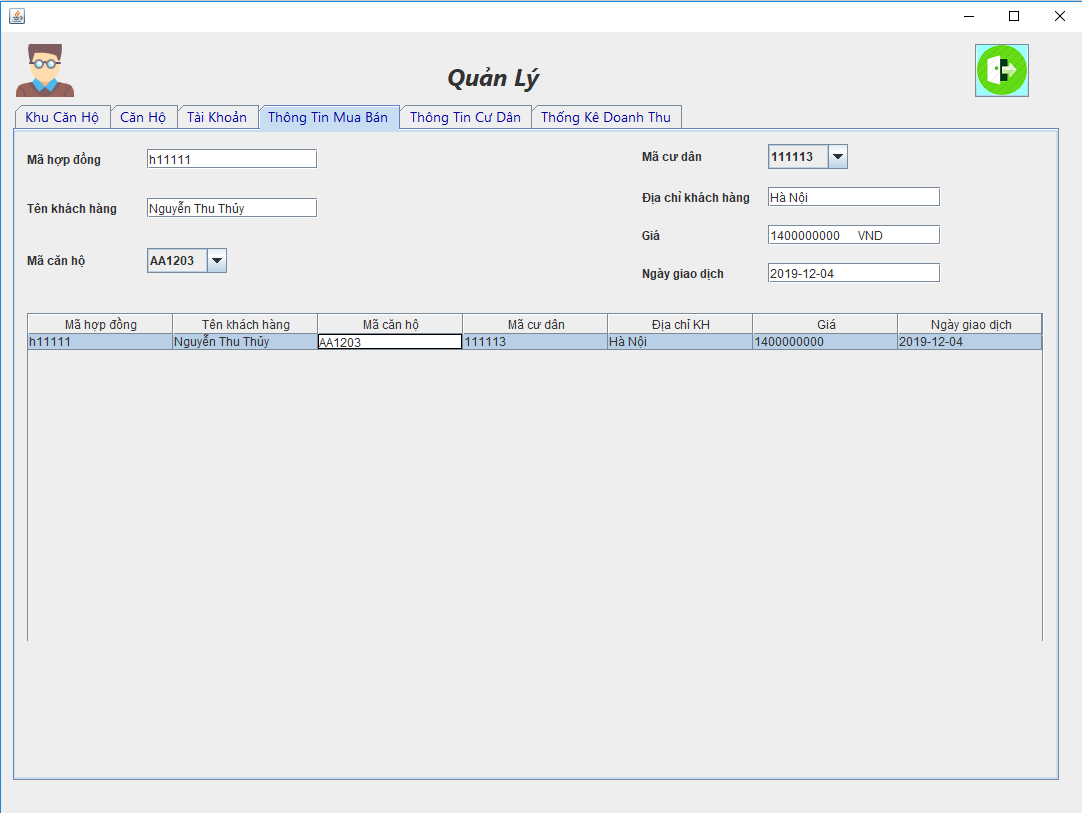
Mô tả : nội dung chính của form bao gồm 2 Textfield nhập thông tin đăng nhập

2 button Đăng nhập và exit .

Quy trình xử lý đăng nhập :



#### FORM quản lý Tab thông tin mua bán .



Hình 3.2 Tab Thông tin mua bán.

Mô tả :

Tab hiển thị thông tin mua bán .

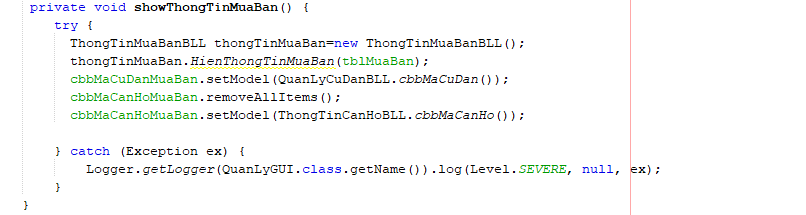
* + - * JTable – thông tin mua bán

Thông tin bao gồm :

* Mã hợp đồng
* Tên khách hàng
* Mã căn hộ
* Mã cư dân
* Địa chỉ khách hàng
* Giá
* Ngày giao dịch

Thông tin được hiển thị dưới dạng bảng .

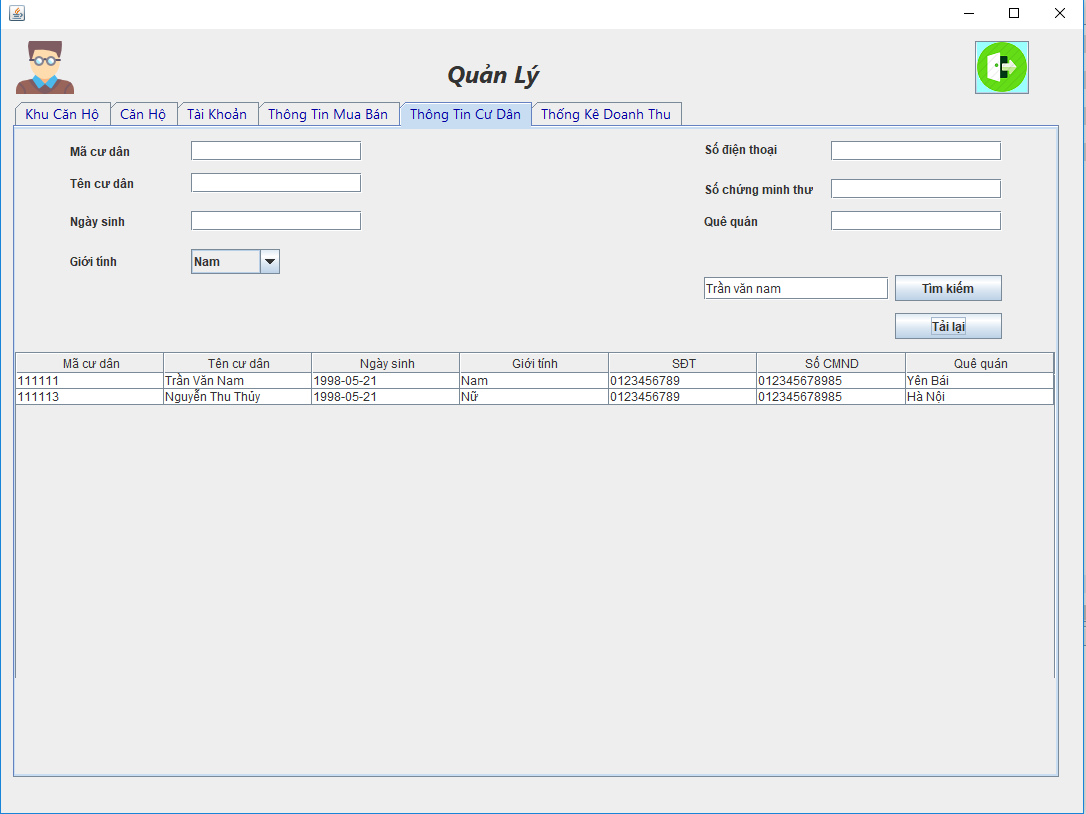
Thư viện đã sử dụng :



#### *Java Swing :*

* *JavaSwing.Jtable*
* *JavaSwing.Jcombobox*

#### FORM Quản lý TAB thông tin cư dân .



Hình 3.3 Tab Thông tin cư dân .

Mô tả chức năng :

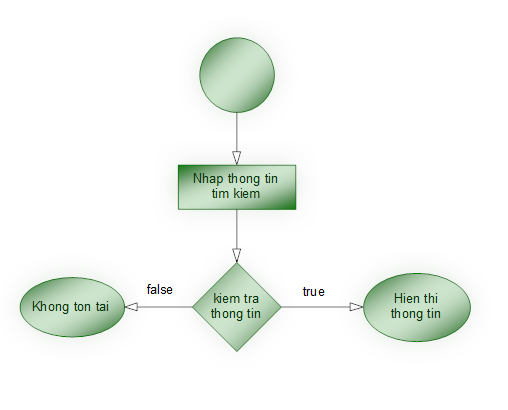
Tab hiển thị thông tin cư dân dưới dạng bảng dữ liệu

* + - * JTable – thông tin cư dân
      * JTextField tìm kiếm
      * JButton – tìm kiếm
      * JButton – tải lại thông tin

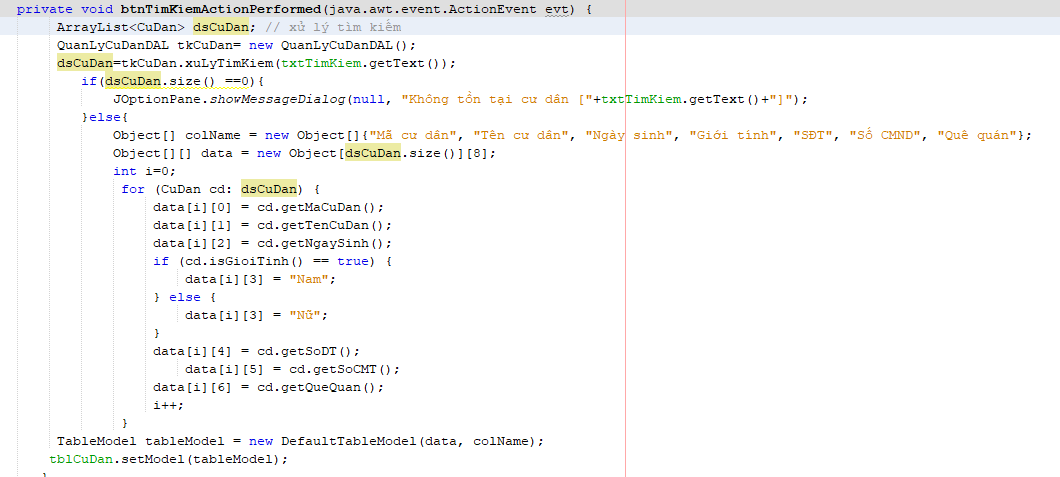
Có chức năng tìm kiếm thông tin

Thông tin bao gồm :

* Mã cư dân
* Tên cư dân
* Ngày sinh
* Giới tinh
* Số điện thoại
* Số chứng minh thư
* Quê quán

Trình tự xử lý chức năng tìm kiếm

Thư viện đã xử dụng :



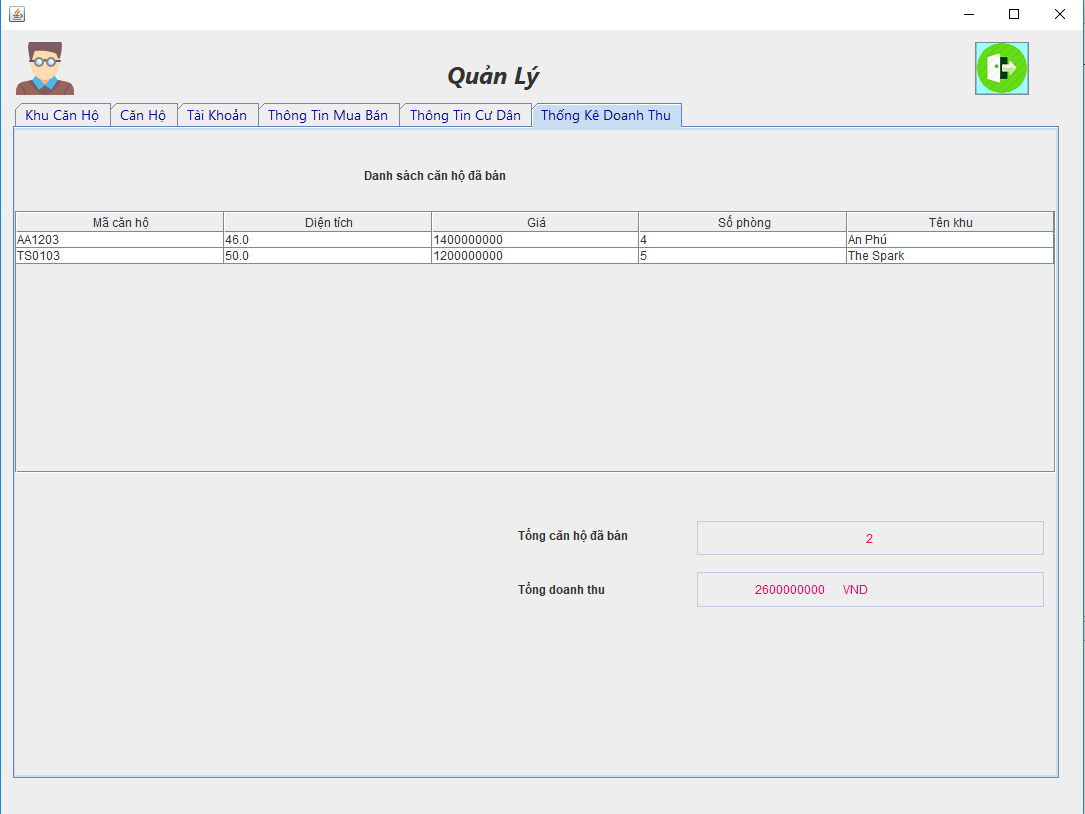
*Java.util*

* *Java.util.ArrayList*

*Java.swing*

* *Java.swing.TableModel*
* *Java.swing.JTable*

#### FORM Quản lý TAB Thống kê doanh thu



Hình 3.4 FORM quản lý tab thống kê doanh thu

Mô tả các đối tượng :

Hiển thị danh sách các căn hộ đã bán dưới dạng bảng

Bao gồm các thông tin :

* Mã căn hộ
* Diện tích
* Giá căn hộ
* Số phòng
* Tên khu

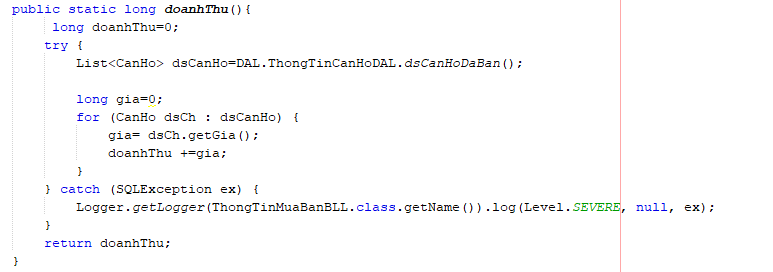
Đếm số căn hộ đã bán

Tổng doanh thu từ các căn hộ đã bán

Các thư viện đã được sử dụng :

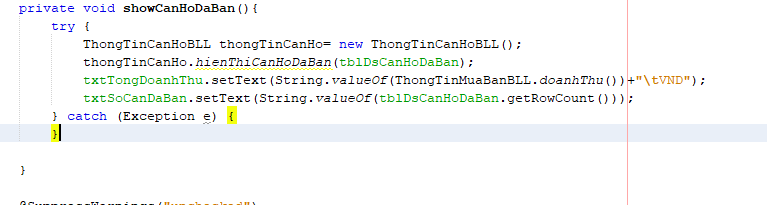
*Java.util*

* *Java.util.List*



*Java.swing*

* *Java.swing.JTable*



### Phạm Quốc Cường – Thực hiện Form Nhân viên với các chức năng Xem thông tin mua bán, Quản lý cư dân, In hợp đồng

#### Giao diện và mô tả các đối tượng trên giao diện

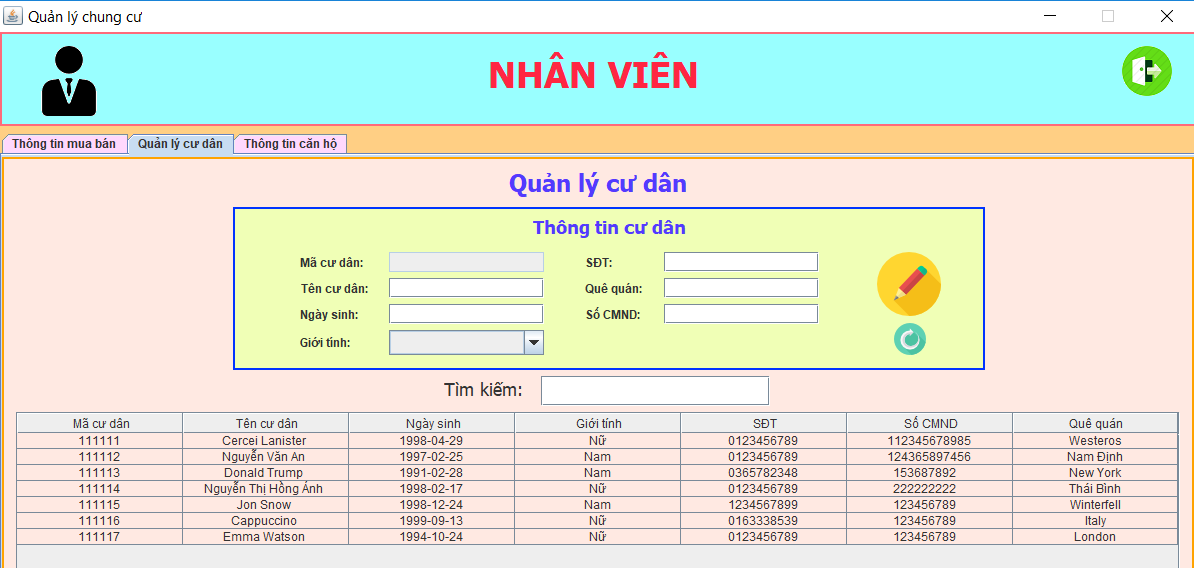
1. Giao diện nhóm chức năng Thông tin mua bán



*Hình 2. Giao diện nhóm chức năng Thông tin mua bán*

* Jbutton E:\Git\Quan-ly-khu-chung-cu\QuanLyChungCu\src\Images\logout.png (Đăng xuất) : Đăng xuất khỏi hệ thống
* JTextField Tìm kiếm : Nhập thông tin cần tìm kiếm.
* JButton E:\Git\Quan-ly-khu-chung-cu\QuanLyChungCu\src\Images\rotation.png (Tải lại) : Tải lại thông tin mua bán.
* JTable : Hiển thị thông tin hợp đồng mua bán.

1. Giao diện nhóm các chức năng Quản lý cư dân



* JButton  (Sửa) : Cho phép sửa thông tin cư dân
* JButton E:\Git\Quan-ly-khu-chung-cu\QuanLyChungCu\src\Images\rotation.png (Tải lại) : Tải lại thông tin cư dân
* JTextField Tìm kiếm : Nhập thông tin cần tìm kiếm
* JTable : Hiện thông tin cư dân

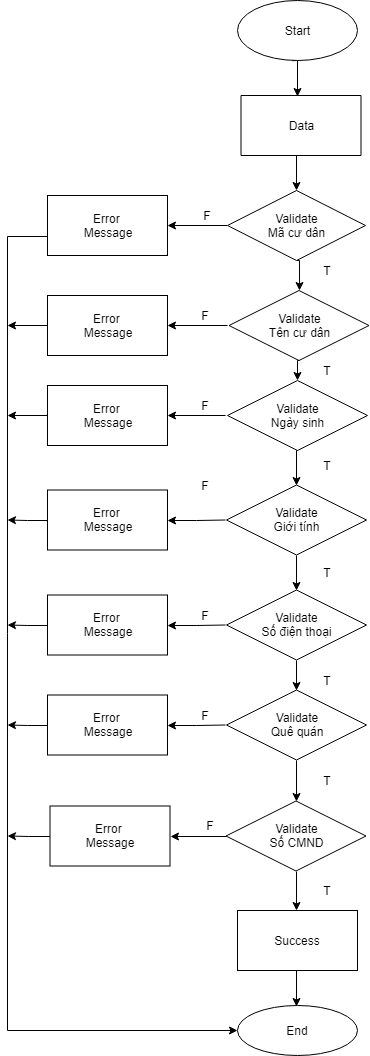
1. Giao diện nhóm chức năng Thông tin căn hộ



* JButton E:\Git\Quan-ly-khu-chung-cu\QuanLyChungCu\src\Images\find.png (Tìm) : Cho phép tìm thông tin căn hộ
* JButton E:\Git\Quan-ly-khu-chung-cu\QuanLyChungCu\src\Images\rotation.png (Tải lại) : Tải lại thông tin căn hộ
* JTable : Hiện thông tin căn hộ
* JCheckBox (Là cư dân) : Kiểm tra xem khách hàng đã từng thực hiện giao dịch nào hay chưa.
* JButton  (In hợp đồng) : Thực hiện lưu thông tin của khách hàng vào cơ sở dữ liệu, đồng thời in hợp đồng ra file Word.

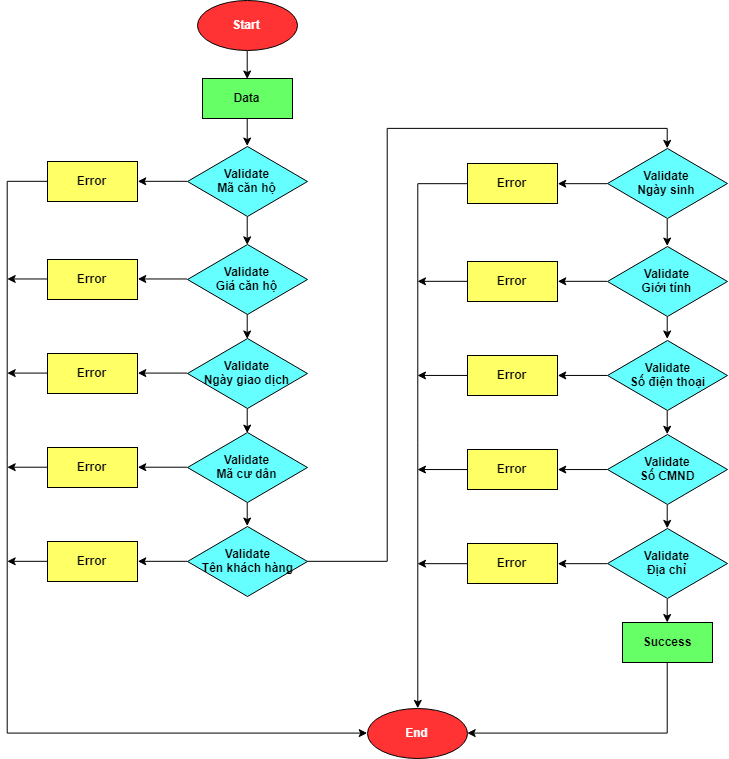
#### Kiểm tra tính hợp lệ của dữ liệu đầu vào

1. Kiểm tra thông tin đầu vào Quản lý cư dân



*Hình 2. Lưu đồ kiểm tra tính hợp lệ dữ liệu của Quản lý cư dân*

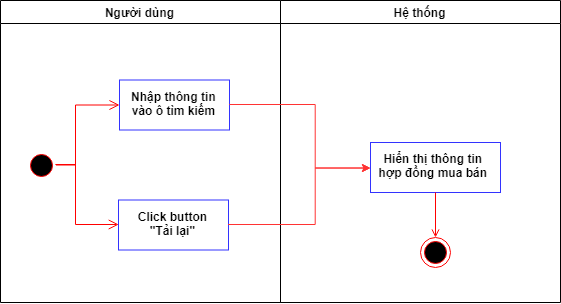
1. Kiểm tra thông tin đầu vào của Thông tin hợp đồng



*Hình 2. Lưu đồ kiểm tra thông tin đầu vào của Thông tin hợp đồng*

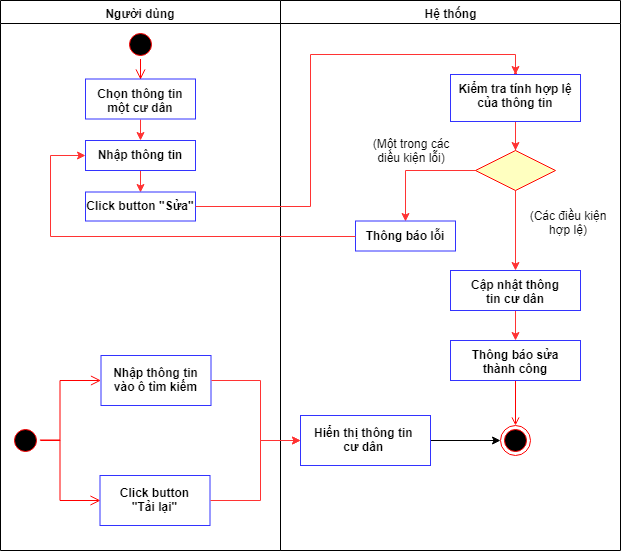
#### Xử lý thao tác người dùng

1. Tab Thông tin mua bán



*Hinh 2. Biểu đồ hoạt động Thông tin mua bán*

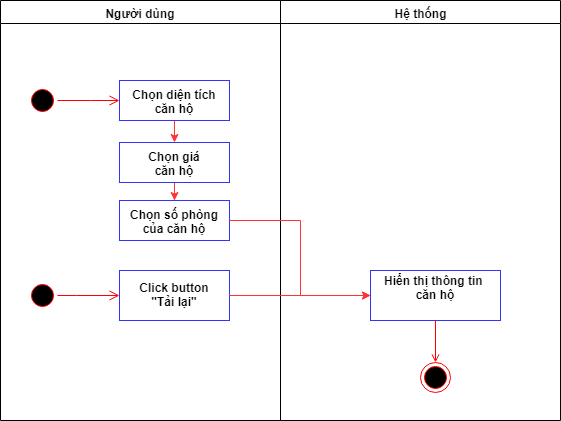
1. Tab Quản lý cư dân



*Hình 2. Biểu đồ hoạt động của Quản lý cư dân*

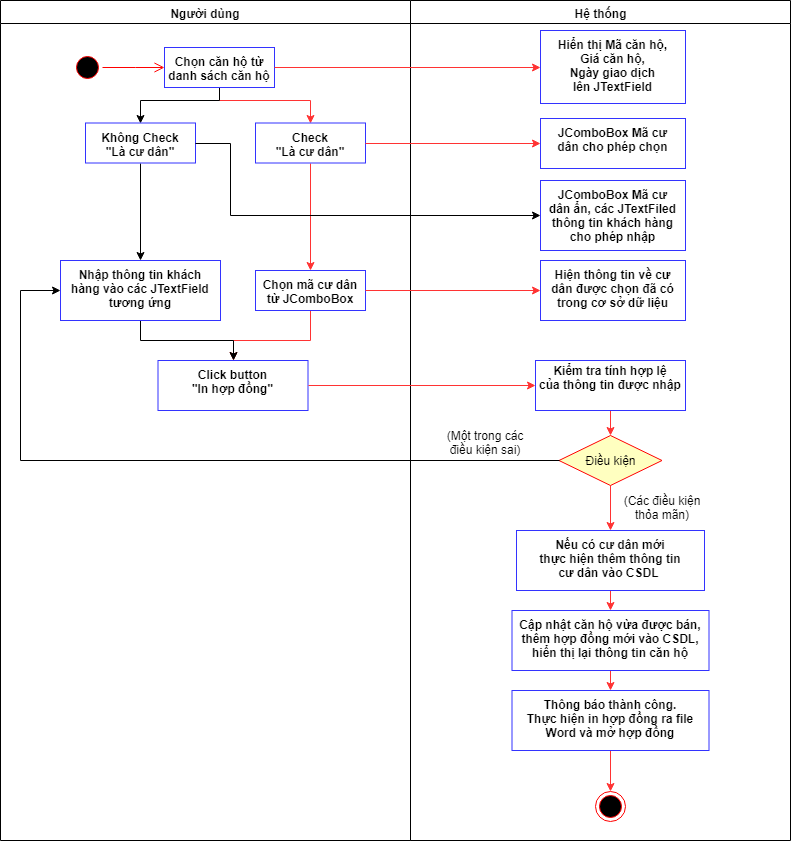
1. Tab Thông tin căn hộ

* Chức năng Tìm căn hộ và Tải lại thông tin căn hộ



*Hình 2. Biểu đồ hoạt động của Chức năng Tìm căn hộ và Tải lại thông tin căn hộ*

* Chức năng In hợp đồng



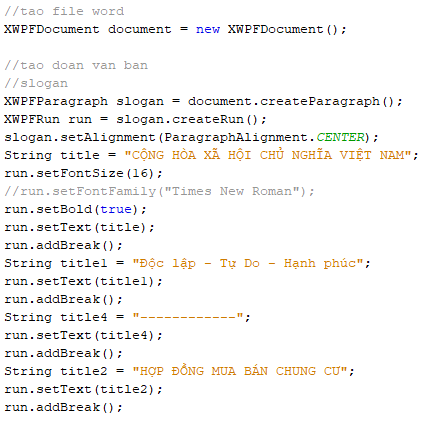
*Hình 2. Biểu đồ hoạt động của Chức năng In hợp đồng*

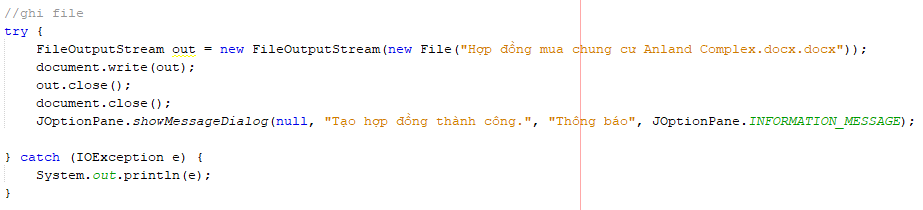
#### Sử dụng các thư viện chuẩn

1. Apache POI

* Apache POI là một API phổ biến nhất, cho phép các lập trình viên tạo, chỉnh sửa và hiển thị file MS Word sử dụng ngôn ngữ Java.
* Các Class chính của Apache POI Word:
* **XWPFDocument**:Được sử dụng để đọc ghi các file MS-Word với định dạng .docx (đọc, lưu file, thêm đoạn văn bản…)
* **XWPFParagraph**: Được dùng để tạo đoạn paragraph trong tài liệu word. (Tùy chỉnh căn lề, border)
* **XWPFRun**: Được sử dụng thể thêm text vào paragraph (Tùy chỉnh font, size…)
* **XWPFTable**: Được dùng để thêm mới bảng vào file (Thêm cột, thêm dòng, cài đặt chiều rộng cho cột)
* **XWPFWordExtractor**: Đây là class cơ bản được dùng để lấy text đơn giản từ tài liệu word (lấy tất cả text bên trong file word).
* Ví dụ ghi file Word:

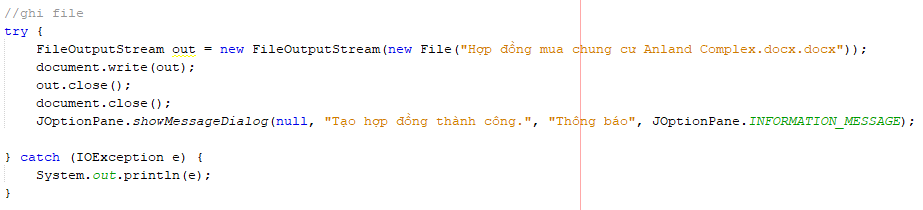






1. java.io

* java.io là tập hợp của các lớp và interface mà ta có thể sử dụng để thực hiện hầu như mọi thao tác vào - ra thông qua ứng dụng java.
* java.io.File đại diện cho một tập tin hoặc một thư mục trong hệ thống, nó đại diện cho một đường dẫn (pathname).
* java.io.FileOutputStream dùng để thêm nội dung File trong Java.



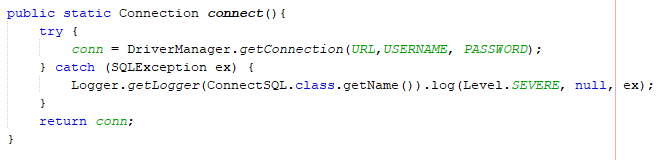
1. java.awt

* java.awt được viết hoàn toàn bằng Java cung cấp một tập hợp các lớp dùng để viết giao diện người dùng đồ họa, **AWT** là bộ framework cũ cho Java (Java 1.1).
* java.awt.Desktop: sử dụng để mở một file Word



1. java.sql

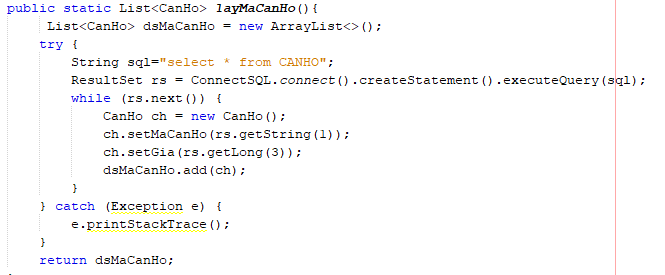
* java.sql dùng để tương tác với các loại cơ sở dữ liệu quan hệ
* java.sql.Connection & java.sql.DriverManager && java.sql.SQLException : 3 package này hỗ trợ kết nối và làm việc với DataBase. Cụ thể đây là kết nối và bắt ngoại lệ khi kết nối



1. java.util

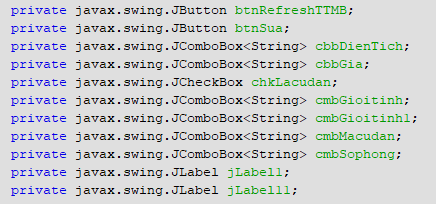
* java.util.List hỗ trợ làm việc với List

Khai báo đối tượng kiểu List<CanHo>



1. java.swing

* java.swing cung cấp các thành phần (Component) gọn nhẹ và độc lập nền tảng. Javax.swing. Package cung cấp các lớp cho Java Swing chảng hạn như JButton, JTextField, JTextArea, JRadioButton, JCheckbox,…



## Kết luận (Phạm Quốc Cường)

<đây là mục lớn cuối cùng, Nếu bài toán sinh viên làm có nhiều mục chi tiết hơn thì phần cuối cùng cũng để là kết luận

### Nội dung đã thực hiện

Sau một thời gian dài bắt tay vào nghiên cứu nhóm chúng em đã hoàn thành đề tài **“Quản lý khu chung cư Anland Complex”.** Qua đây bản thân em cũng như các thành viên trong nhóm đã học hỏi được nhiều kinh nghiệm.

* Làm việc nhóm một cách khoa học, hiệu quả.
* Biết được cách xây dựng mô hình cho một phần mềm.
* Tăng kỹ năng chuyên môn của bản thân.

Với những kiến thức thu nhận được nhóm chúng em đã hoàn thành bài tập lớn này. Tuy nhiên, phần mềm cũng còn nhiều hạn chế mà chúng em chưa nhận biết được. Chúng em rất mong có được những nhận xét, đánh giá từ phía cô giáo để có thể nhìn ra hạn chế của nhóm em trong việc xây dựng phần mềm.

### Hướng phát triển.

Trong quá trình hoàn thành bài tập lớn, các thành viên trong nhóm cũng đã nâng cao được phần nào các kỹ năng chuyên môn. Vì vậy, có thể nhóm phát triển sẽ không dừng lại ở mức độ là bài tập lớn mà có thể phát triển đi lên làm đồ án, hoặc có thể cao hơn nữa là quản lý những dự án trong thực tế.

Với hướng phát triển là xây dựng đồ án đi lên từ bài tập lớn này, điều này sẽ được thực hiện nếu các kiến thực đạt được của thành viên trong nhóm là tương đối tốt, đáp ứng được yêu cầu của Cô hay chính dự án.