

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÁO CÁO BÀI TẬP LỚN
MÔN: LẬP TRÌNH API
ĐỀ TÀI:
XÂY DỰNG WEBSITE QUẢN TRỊ DỮ LIỆU THUÊ NHÀ

Giảng viên hướng dẫn : ThS. PHAN THANH HY

Sinh viên thực hiện : HOÀNG GIA KIẾT
: NGUYỄN HƯƠNG DUYÊN

Lớp : CÔNG NGHỆ THÔNG TIN

Khoá : 62

TP. Hồ Chí Minh, tháng 4 năm 2024

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO BÀI TẬP LỚN
MÔN: LẬP TRÌNH API
ĐỀ TÀI:
XÂY DỰNG WEBSITE QUẢN TRỊ DỮ LIỆU THUÊ NHÀ**

Giảng viên hướng dẫn : ThS. PHAN THANH HY

Sinh viên thực hiện : HOÀNG GIA KIẾT
: NGUYỄN HƯƠNG DUYÊN

Lớp : CÔNG NGHỆ THÔNG TIN

Khoá : 62

TP. Hồ Chí Minh, tháng 4 năm 2024

LỜI CẢM ƠN

Lời đầu tiên nhóm chúng em xin gửi lời cảm ơn sâu sắc đến thầy Phan Thanh Hy đã hỗ trợ và giúp đỡ nhóm chúng em hoàn thành môn học Lập trình API.

Chúng em cũng xin gửi lời cảm ơn đến quý thầy cô Bộ môn Công nghệ thông tin Trường Đại học Giao thông Vận tải Phân hiệu tại TP. Hồ Chí Minh đã truyền đạt kiến thức và giúp đỡ nhóm chúng em trong quá trình học tập để chúng em có thể hoàn thành tốt bài tập lớn của mình.

Ngoài ra, chúng em xin gửi lời cảm ơn đến các anh, chị, bạn đã đồng hành và giúp đỡ chúng em về tài liệu.

Trong quá trình thực hiện đề tài, chúng em còn gặp nhiều khó khăn, thiếu kiến thức chuyên môn nên báo cáo còn nhiều thiếu sót. Mong thầy có thể xem xét, đánh giá cho bài tập lớn của nhóm chúng em được cập nhật và chỉnh sửa tốt nhất.

Chúng em xin chân thành cảm ơn thầy./.

TP. Hồ Chí Minh, ngày ... tháng ... năm 2024

Nhóm sinh viên thực hiện

This image shows a full page of a document template designed for writing. It features approximately 28 evenly spaced, thin grey horizontal lines across the entire width of the page. The background is white, and there are no margins, headers, or footers visible.

Giảng viên hướng dẫn

ii

MỤC LỤC

LỜI CẢM ƠN.....	i
NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN.....	ii
MỤC LỤC.....	iii
DANH MỤC CHỮ VIẾT TẮT	v
DANH MỤC BẢNG BIỂU	vi
DANH MỤC HÌNH ẢNH.....	viii
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT	1
1.1. Định nghĩa API.....	1
1.2. Yêu cầu khi thiết kế API	1
1.2.1. Rõ ràng và dễ hiểu	1
1.2.2. Tính nhất quán	1
1.2.3. Bảo mật	1
1.2.4. Hiệu suất	1
1.2.5. Khả năng mở rộng	1
1.2.6. Dễ sử dụng	1
1.3. HTTP và vai trò xây dựng API	1
CHƯƠNG 2: KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU	3
2.1. Khảo sát hiện trạng.....	3
2.1.1. Quy trình nghiệp vụ	3
2.1.2. Mô tả bài toán	3
2.2. Phân tích yêu cầu	3
2.2.1. Yêu cầu chức năng.....	3
2.2.2. Yêu cầu phi chức năng.....	4
CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	5
3.1. Các biểu đồ hệ thống	5
3.1.1. Biểu đồ thực thể.....	5
3.1.2. Biểu đồ lớp.....	5
3.1.3. Biểu đồ use case.....	6
3.1.4. Biểu đồ hoạt động	21

3.1.5. Biểu đồ tuần tự.....	26
3.2. Danh sách các API	28
3.2.1. Auth.....	28
3.2.2. User	29
3.2.3. Customer	29
3.2.4. Apartment	30
3.2.5. Contract.....	31
3.3. Đặc tả các bảng dữ liệu.....	31
3.3.1. Bảng User.....	31
3.3.2. Bảng Role.....	32
3.3.3. Bảng Customer	32
3.3.4. Bảng Apartment	33
3.3.5. Bảng Contract	33
CHƯƠNG 4: XÂY DỰNG CHƯƠNG TRÌNH	34
4.1. Tổng quan các công cụ, công nghệ hỗ trợ xây dựng hệ thống. 34	
4.1.1. Phát triển Backend	34
4.1.2. Phát triển Frontend.....	35
4.2. Xây dựng chương trình	35
4.2.1. Chức năng tiêu biểu chung phía Backend	35
4.2.2. Frontend	44
4.3. Giao diện hệ thống	47
KẾT LUẬN.....	52
1. Kết quả đạt được.....	52
2. Hạn chế	52
3. Hướng phát triển.....	52
PHÂN CÔNG CÔNG VIỆC	53
TÀI LIỆU THAM KHẢO	54

DANH MỤC CHỮ VIẾT TẮT

STT	VIẾT TẮT	Ý NGHĨA	DIỄN GIẢI
1	VSCode	Visual Studio Code	
2	API	Application Programming Interface	Giao diện lập trình ứng dụng
3	HTTP	HyperText Transfer Protocol	Giao thức truyền tải siêu văn bản
4	IDE	Integrated development environment	Môi trường phát triển tích hợp
5	XML	Extensible markup language	
6	JWT	JSON Web Token	
7	BA	Business Analyst	Phân tích nghiệp vụ
8	BE	Back End	
9	FE	Front End	

DANH MỤC BẢNG BIỂU

Bảng 3.1: Đặc tả use case thêm khách hàng	7
Bảng 3.2: Đặc tả use case sửa khách hàng	8
Bảng 3.3: Đặc tả use case import danh sách khách hàng	8
Bảng 3.4: Đặc tả use case export danh sách khách hàng.....	9
Bảng 3.5: Đặc tả use case xoá khách hàng	9
Bảng 3.6: Đặc tả use case thêm hợp đồng	11
Bảng 3.7: Đặc tả use case sửa hợp đồng.....	11
Bảng 3.8: Đặc tả use case import danh sách hợp đồng	12
Bảng 3.9: Đặc tả use case export danh sách hợp đồng	12
Bảng 3.10: Đặc tả use case xoá hợp đồng	12
Bảng 3.11: Đặc tả use case thêm căn hộ.....	14
Bảng 3.12: Đặc tả use case sửa căn hộ	14
Bảng 3.13: Đặc tả use case import danh sách căn hộ	15
Bảng 3.14: Đặc tả use case export danh sách căn hộ.....	15
Bảng 3.15: Đặc tả use case xoá căn hộ.....	15
Bảng 3.16: Đặc tả use case thêm người dùng.....	16
Bảng 3.17: Đặc tả use case sửa người dùng	17
Bảng 3.18: Đặc tả use case chặn người dùng	17
Bảng 3.19: Đặc tả use case thống kê	17
Bảng 3.20: Đặc tả use case đăng nhập.....	18
Bảng 3.21: Đặc tả use case đăng xuất.....	18
Bảng 3.22: Đặc tả use case đổi mật khẩu	19
Bảng 3.23: Đặc tả use case quên mật khẩu.....	19
Bảng 3.24: Đặc tả use case cập nhật thông tin	20
Bảng 3.25: Danh sách API Auth.....	29
Bảng 3.26: Danh sách API User	29
Bảng 3.27: Danh sách API Customer	30
Bảng 3.28: Danh sách API Apartment	30
Bảng 3.29: Danh sách API Contract.....	31

Bảng 3.30: Đặc tả bảng User	32
Bảng 3.31: Đặc tả bảng Role	32
Bảng 3.32: Đặc tả bảng Customer	32
Bảng 3.33: Đặc tả bảng Apartment.....	33
Bảng 3.34: Đặc tả bảng Contract.....	33
Bảng 4.1: Code xử lý đăng nhập.....	36
Bảng 4.2: Code xử lý lấy danh sách người dùng.....	37
Bảng 4.3: Code xử lý tạo mới người dùng	39
Bảng 4.4: Code xử lý Lấy thông tin người dùng theo token	40
Bảng 4.5: Code xử lý cập nhật thông tin người dùng.....	42
Bảng 4.6: Code xử lý xuất danh sách người dùng.....	43

DANH MỤC HÌNH ẢNH

Hình 3.1: Biểu đồ lớp quản trị dữ liệu thuê nhà	5
Hình 3.2: Biểu đồ ERD quản trị dữ liệu thuê nhà	5
Hình 3.3: Biểu đồ use case tổng quát	6
Hình 3.4: Biểu đồ use case quản lý khách hàng	7
Hình 3.5: Biểu đồ use case quản lý hợp đồng	10
Hình 3.6: Biểu đồ use case quản lý căn hộ	13
Hình 3.7: Biểu đồ use case quản lý người dùng	16
Hình 3.8: Biểu đồ hoạt động đăng nhập	21
Hình 3.9: Biểu đồ hoạt động đăng xuất	22
Hình 3.10: Biểu đồ hoạt động đổi mật khẩu	23
Hình 3.11: Biểu đồ hoạt động quên mật khẩu	24
Hình 3.12: Biểu đồ hoạt động cập nhật thông tin	25
Hình 3.19: Biểu đồ tuần tự đăng nhập	26
Hình 3.20: Biểu đồ tuần tự đăng xuất	26
Hình 3.21: Biểu đồ tuần tự đổi mật khẩu	27
Hình 3.22: Biểu đồ tuần tự quên mật khẩu	27
Hình 3.23: Biểu đồ tuần tự cập nhật thông tin	28
Hình 4.1: Kiểm tra API đăng nhập	36
Hình 4.2: Kiểm tra API lấy danh sách người dùng	38
Hình 4.3: Kiểm tra API tạo mới người dùng	39
Hình 4.4: Kiểm tra API Lấy thông tin người dùng theo token	40
Hình 4.5: Kiểm tra API cập nhật thông tin người dùng	43
Hình 4.6: Kiểm tra API xuất danh sách người dùng	44
Hình 4.7: Giao diện đăng nhập	47
Hình 4.8: Giao diện quên mật khẩu	48
Hình 4.9: Giao diện gửi mật khẩu mới qua email	48
Hình 4.10: Giao diện trang chủ	48
Hình 4.11: Giao diện quản lý khách hàng	49
Hình 4.12: Giao diện quản lý căn hộ	49

Hình 4.13: Giao diện quản lý hợp đồng.....	49
Hình 4.14: Giao diện quản lý người dùng	50
Hình 4.15: Giao diện tài khoản.....	50
Hình 4.16: Giao diện sau khi đổi mail mới.....	50
Hình 4.17: Giao diện đổi mật khẩu.....	51

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1. Định nghĩa API

API là lớp trung gian cho phép các ứng dụng khác nhau giao tiếp và trao đổi dữ liệu với nhau một cách hiệu quả. API cung cấp một tập hợp các quy tắc và định nghĩa rõ ràng về cách thức các ứng dụng có thể tương tác, bao gồm các yêu cầu, định dạng dữ liệu và các phương thức phản hồi.

1.2. Yêu cầu khi thiết kế API

1.2.1. Rõ ràng và dễ hiểu

Tài liệu API cần được viết rõ ràng, súc tích và dễ hiểu cho các nhà phát triển khác. Điều này bao gồm việc mô tả chi tiết các chức năng của API, định dạng dữ liệu, phương thức yêu cầu và phản hồi, cũng như các ví dụ sử dụng.

1.2.2. Tính nhất quán

API cần duy trì tính nhất quán trong cách đặt tên, định dạng dữ liệu và cấu trúc tổng thể. Việc tuân theo các quy ước chung sẽ giúp các nhà phát triển dễ dàng học hỏi và sử dụng API.

1.2.3. Bảo mật

API cần được thiết kế với các biện pháp bảo mật phù hợp để bảo vệ dữ liệu và ngăn chặn truy cập trái phép. Điều này bao gồm việc sử dụng xác thực, ủy quyền và mã hóa dữ liệu.

1.2.4. Hiệu suất

API cần được tối ưu hóa để đảm bảo hiệu suất cao và thời gian phản hồi nhanh chóng. Điều này đặc biệt quan trọng đối với các API được sử dụng cho các ứng dụng thời gian thực.

1.2.5. Khả năng mở rộng

API cần được thiết kế để có thể dễ dàng mở rộng và bổ sung các chức năng mới trong tương lai. Điều này giúp đảm bảo API có thể đáp ứng nhu cầu thay đổi của các nhà phát triển và ứng dụng.

1.2.6. Dễ sử dụng

API cần được thiết kế để dễ sử dụng cho các nhà phát triển, bao gồm việc cung cấp các thư viện, công cụ và các ví dụ mã để hỗ trợ việc tích hợp API.

1.3. HTTP và vai trò xây dựng API

HTTP là giao thức nền tảng được sử dụng để truyền tải thông tin giữa các máy tính trên mạng internet. **HTTP** đóng vai trò nền tảng trong việc xây dựng API, cung cấp giao thức tiêu chuẩn để trao đổi dữ liệu giữa các ứng dụng. [1]

HTTP quy định cách thức các ứng dụng gửi yêu cầu và nhận phản hồi. Các phương thức HTTP thường được sử dụng trong API bao gồm GET, POST, PUT và DELETE, mỗi phương thức đại diện cho một loại thao tác dữ liệu cụ thể:

- Phương thức GET: được sử dụng để lấy dữ liệu từ một tài nguyên được xác định bởi URL.

- Phương thức POST: được sử dụng để gửi dữ liệu đến server để tạo hoặc cập nhật một tài nguyên.

- Phương thức PUT: được sử dụng để cập nhật một tài nguyên hiện có.

- Phương thức DELETE: được sử dụng để xóa một tài nguyên.

Các phương thức kém phổ biến hơn bao gồm: OPTIONS, TRACE, PATCH, CONNECT được mô tả trong RFC9110.

CHƯƠNG 2: KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

2.1. Khảo sát hiện trạng

2.1.1. Quy trình nghiệp vụ

- Nhân viên kinh doanh trong phòng ban sẽ tìm khách hàng, căn hộ và hợp đồng cho công ty. Sau đó cập nhật thông tin vào file máy tính cá nhân

- Một nhân viên có thể tạo nhiều hợp đồng, mỗi hợp đồng thì sẽ có thông tin của khách hàng, căn hộ và nhân viên tạo ra hợp đồng đó.

- Cuối tuần sẽ gửi lại file cho người quản lý để cập nhật vào file của máy tính công ty. Có những trường hợp thông tin bị trùng thì rất tốn thời gian để xử lý.

- Khi có chỉnh sửa thì cập nhật thông tin với người quản lý.

- Hầu hết các nghiệp vụ quản lý đều được thực hiện theo cách thủ công, gây chậm trễ, phiền toái.

2.1.2. Mô tả bài toán

Một công ty cần chuyển đổi số vào việc quản lý dữ liệu khách hàng, căn hộ, hợp đồng cho phòng ban kinh doanh. Việc sử dụng hệ thống chung sẽ làm tăng khả năng quản lý, nhanh chóng và tránh trùng lặp dữ liệu. Nhân viên sẽ nhập thông tin vào hệ thống quản lý. Thông tin về khách hàng: Mã, Họ tên, Địa chỉ, Số CCCD, Ngày sinh, Số điện thoại. Thông tin về văn hộ: Mã, Địa chỉ, Giá tiền, Số phòng. Thông tin về hợp đồng: Mã, Ngày bắt đầu, Ngày kết thúc, Tổng tiền, Tiền thuê 1 tháng, Ngày tạo, Mã khách hàng, Mã căn hộ, Mã nhân viên tạo. Khi có yêu cầu tạo hợp đồng thì nhân viên sử dụng chương trình cần nhập thông tin khách hàng (nếu chưa có).

Ngoài ra chương trình cần có thêm các chức năng như:

- Import/Export danh sách thông tin của khách hàng, căn hộ, hợp đồng
- Tra cứu thông tin hợp đồng
- Thống kê

2.2. Phân tích yêu cầu

2.2.1. Yêu cầu chức năng

- Cho phép người dùng đăng nhập vào hệ thống
- Cho phép người dùng cập nhật thông tin cá nhân
- Cho phép người quản lý xem thống kê của hệ thống
- Phân quyền cho nhân viên và quản lý:

+ Quản lý, nhân viên: Thêm, xem và chỉnh sửa dữ liệu

- + Quản lý: Xoá dữ liệu

- Cho phép người quản lý, nhân viên nhập và xuất dữ liệu khách hàng, căn hộ, hợp đồng

2.2.2. Yêu cầu phi chức năng

Yêu cầu về chất lượng

- Tính cập nhật – phát triển:

- + Dễ dàng nâng cấp hoặc thêm các tính năng tiện ích

- + Bảo trì hệ thống

- Tính tiện dụng:

- + Có các đặc tả và hướng dẫn rõ ràng, dễ dàng thao tác

- + Hoạt động ổn định, đáng tin cậy

- Tính tương thích:

- + Hoạt động tốt trên tất cả các phương tiện phổ biến hiện nay

Yêu cầu về giao diện

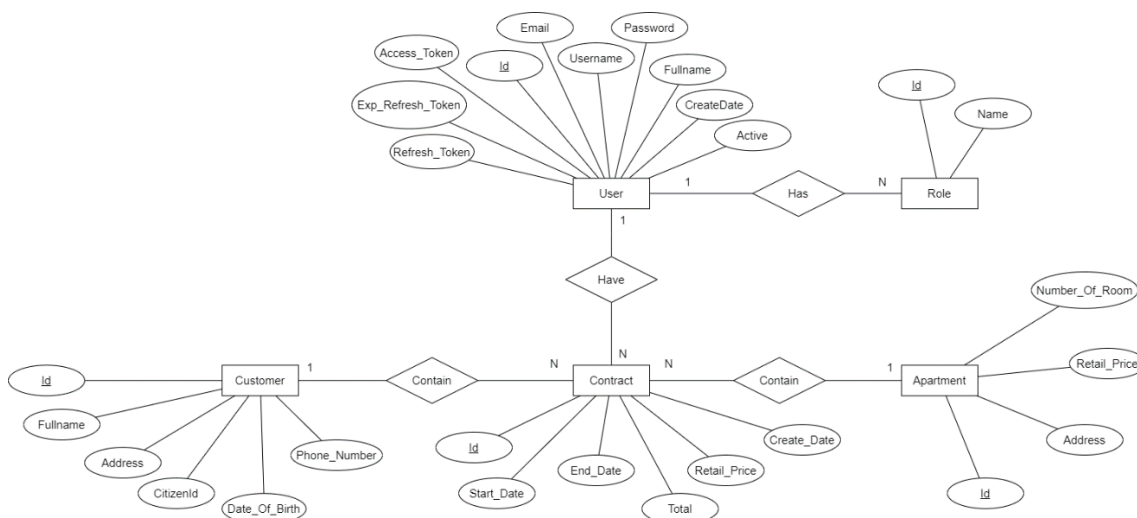
- + Thân thiện, dễ sử dụng

- + Hiện thị đầy đủ và chi tiết các thông tin, tính nhất quán

CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

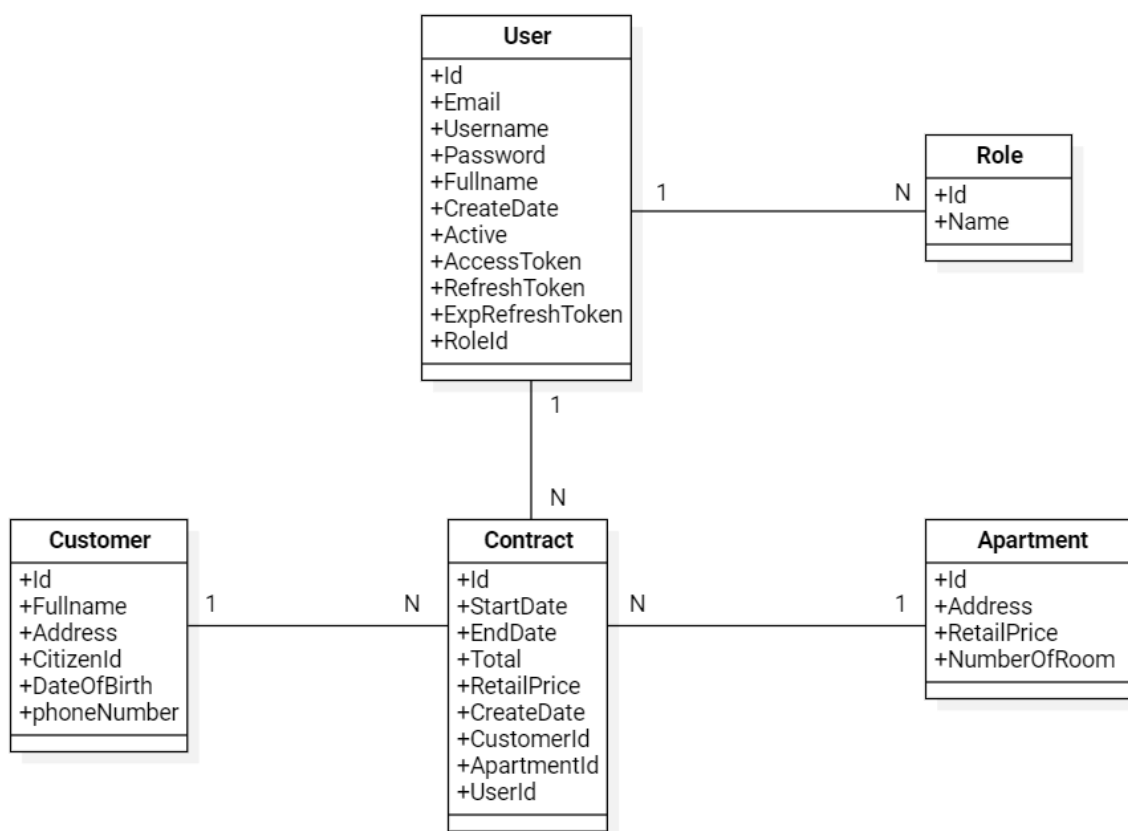
3.1. Các biểu đồ hệ thống

3.1.1. Biểu đồ thực thể



Hình 3.1: Biểu đồ lớp quản trị dữ liệu thuê nhà

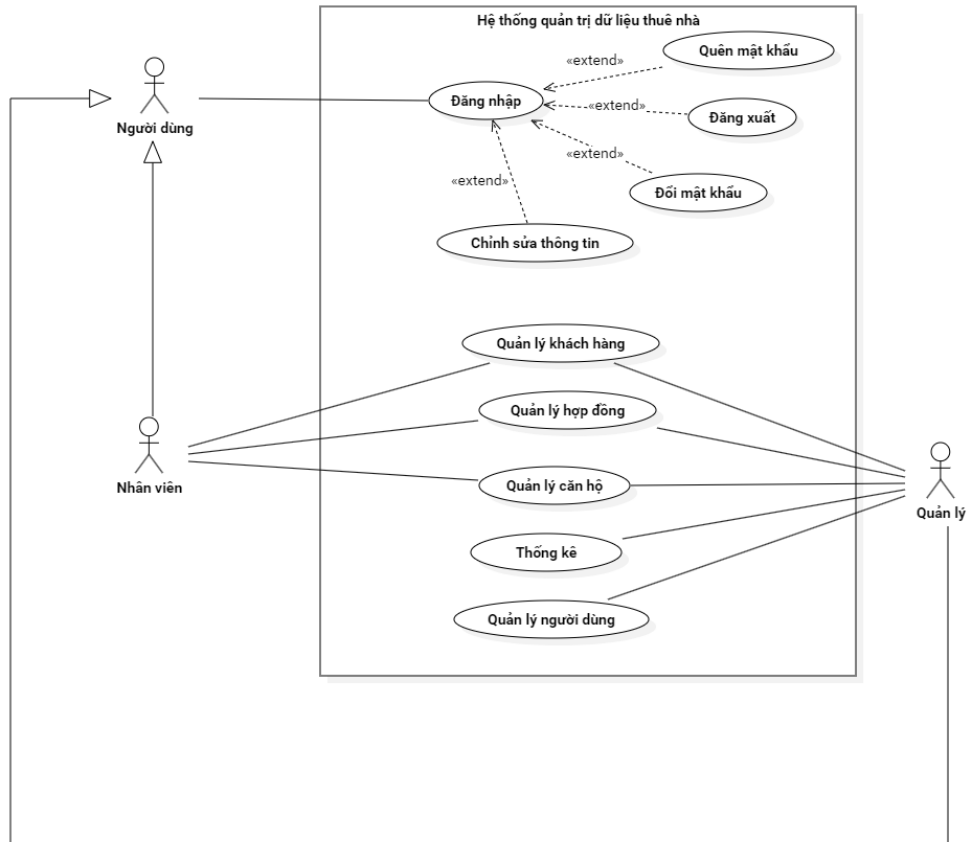
3.1.2. Biểu đồ lớp



Hình 3.2: Biểu đồ ERD quản trị dữ liệu thuê nhà

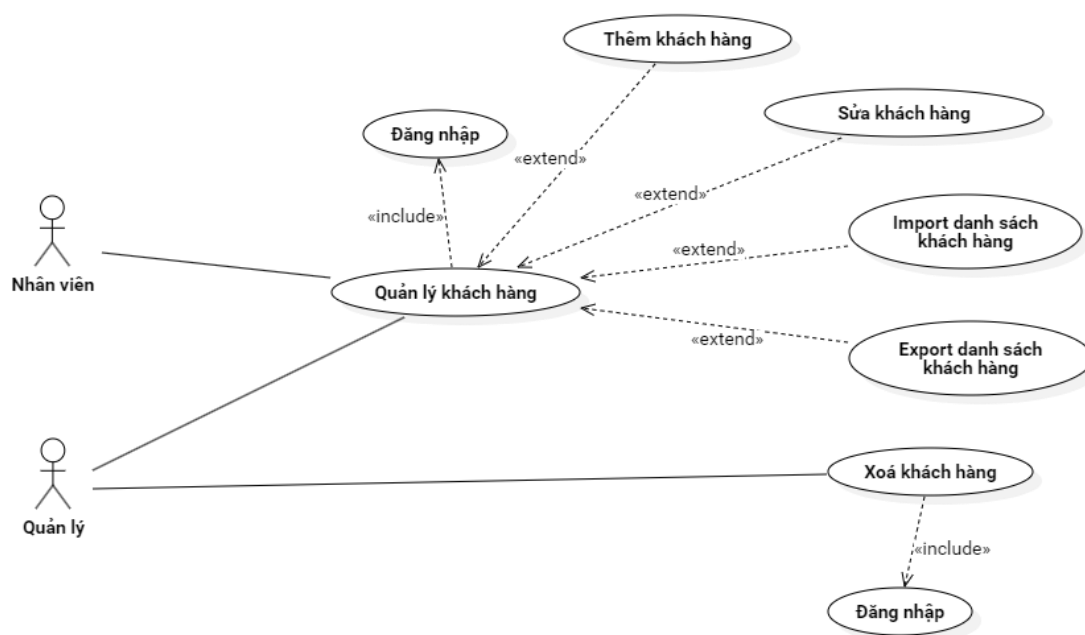
3.1.3. Biểu đồ use case

3.1.3.1. Use case tổng quát hệ thống



Hình 3.3: Biểu đồ use case tổng quát

3.1.3.2. Chức năng quản lý khách hàng



Hình 3.4: Biểu đồ use case quản lý khách hàng

- Đặc tả thêm khách hàng

Tên use case:	Thêm khách hàng
Tác nhân:	Quản lý, nhân viên
Điều kiện tiên quyết:	<ul style="list-style-type: none"> Người dùng phải đăng nhập Đang ở trang Quản lý khách hàng
Hậu điều kiện:	<ul style="list-style-type: none"> Thêm thành công khách hàng mới
Luồng sự kiện chính:	<ol style="list-style-type: none"> 1. Người dùng chọn nút “Thêm” 2. Hệ thống hiển thị form thêm khách hàng 3. Người dùng nhập thông tin bắt buộc 4. Người dùng nhấn nút “Lưu” 5. Hệ thống kiểm tra thông tin 6. Hệ thống cho phép thêm khách hàng 7. Hệ thống lưu khách hàng
Luồng sự kiện phụ:	<p>A. Thông tin không đúng yêu cầu</p> <ol style="list-style-type: none"> 1. Hệ thống báo lỗi 2. Người dùng nhập lại thông tin không đúng yêu cầu

Bảng 3.1: Đặc tả use case thêm khách hàng

- Đặc tả sửa khách hàng

Tên use case:	Sửa khách hàng
Tác nhân:	Quản lý, nhân viên

Điều kiện tiên quyết:	<ul style="list-style-type: none"> • Người dùng phải đăng nhập • Đang ở trang Quản lý khách hàng
Hậu điều kiện:	<ul style="list-style-type: none"> • Sửa thành công khách hàng
Luồng sự kiện chính:	<ol style="list-style-type: none"> 1. Người dùng chọn khách hàng mong muốn chỉnh sửa 2. Người dùng chọn nút “Sửa” 3. Hệ thống hiển thị form sửa khách hàng 4. Người dùng nhập thông tin chỉnh sửa 5. Người dùng nhấn nút “Lưu” 6. Hệ thống kiểm tra thông tin 7. Hệ thống cho phép cập nhật khách hàng 8. Hệ thống cập nhật khách hàng
Luồng sự kiện phụ:	<p>A. Thông tin không đúng yêu cầu</p> <ol style="list-style-type: none"> 1. Hệ thống báo lỗi 2. Người dùng nhập lại thông tin không đúng yêu cầu

Bảng 3.2: Đặc tả use case sửa khách hàng

- Đặc tả import danh sách khách hàng

Tên use case:	Import danh sách khách hàng
Tác nhân:	Quản lý, nhân viên
Điều kiện tiên quyết:	<ul style="list-style-type: none"> • Người dùng phải đăng nhập • Đang ở trang Quản lý khách hàng
Hậu điều kiện:	<ul style="list-style-type: none"> • Thêm thành công danh sách khách hàng
Luồng sự kiện chính:	<ol style="list-style-type: none"> 1. Người dùng chọn nút “Import” 2. Hệ thống hiển thị form import 3. Người dùng chọn file cần import 4. Người dùng nhấn nút “Gửi” 5. Hệ thống kiểm tra file được gửi lên 6. Hệ thống cho phép import 7. Hệ thống lưu danh sách khách hàng 8. Hệ thống thông báo kết quả import 9. Hệ thống import thành công
Luồng sự kiện phụ:	<p>A. File không đúng yêu cầu</p> <ol style="list-style-type: none"> 1. Hệ thống báo lỗi 2. Người dùng chọn file lại <p>B. Nội dung danh sách nhập không đúng yêu cầu</p> <ol style="list-style-type: none"> 1. Hệ thống báo lỗi 2. Người dùng chỉnh sửa lại nội dung

Bảng 3.3: Đặc tả use case import danh sách khách hàng

- Đặc tả export danh sách khách hàng

Tên use case:	Export danh sách khách hàng
Tác nhân:	Quản lý, nhân viên
Điều kiện tiên quyết:	<ul style="list-style-type: none"> • Người dùng phải đăng nhập • Đang ở trang Quản lý khách hàng
Hậu điều kiện:	<ul style="list-style-type: none"> • Tải thành công danh sách khách hàng
Luồng sự kiện chính:	<ol style="list-style-type: none"> 1. Người dùng chọn nút “Export” 2. Hệ thống xử lý file 3. Hệ thống gửi file về cho trình duyệt 4. Trình duyệt tự động tải file 5. Hệ thống export thành công
Luồng sự kiện phụ:	Không có

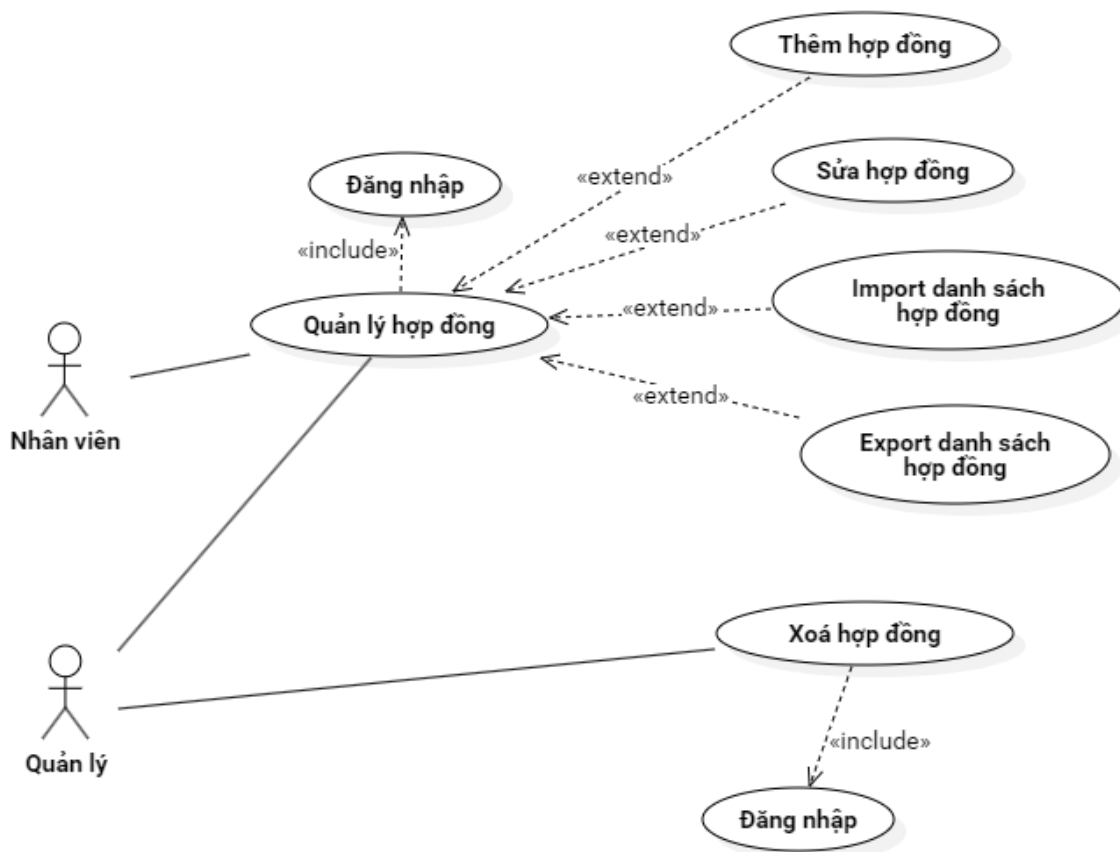
Bảng 3.4: Đặc tả use case export danh sách khách hàng

- Đặc tả xoá khách hàng

Tên use case:	Xoá khách hàng
Tác nhân:	Quản lý
Điều kiện tiên quyết:	<ul style="list-style-type: none"> • Người dùng phải đăng nhập • Đang ở trang Quản lý khách hàng
Hậu điều kiện:	<ul style="list-style-type: none"> • Xoá thành công khách hàng
Luồng sự kiện chính:	<ol style="list-style-type: none"> 1. Người dùng chọn khách hàng mong muốn xoá 2. Người dùng chọn nút “Xoá” 3. Hệ thống hỏi xác nhận xoá 4. Người dùng xác nhận xoá 5. Hệ thống xoá thành công khách hàng
Luồng sự kiện phụ:	<p>A. Người dùng xác nhận không xoá</p> <ol style="list-style-type: none"> 1. Hệ thống hiển thị trang Quản lý khách hàng

Bảng 3.5: Đặc tả use case xoá khách hàng

3.1.3.3. Chức năng quản lý hợp đồng



Hình 3.5: Biểu đồ use case quản lý hợp đồng

- Đặc tả thêm hợp đồng

Tên use case:	Thêm hợp đồng
Tác nhân:	Quản lý, nhân viên
Điều kiện tiên quyết:	<ul style="list-style-type: none"> • Người dùng phải đăng nhập • Đang ở trang Quản lý hợp đồng
Hậu điều kiện:	<ul style="list-style-type: none"> • Thêm thành công hợp đồng mới
Luồng sự kiện chính:	<ol style="list-style-type: none"> 1. Người dùng chọn nút “Thêm” 2. Hệ thống hiển thị form thêm hợp đồng 3. Người dùng nhập thông tin bắt buộc 4. Người dùng nhấn nút “Lưu” 5. Hệ thống kiểm tra thông tin 6. Hệ thống cho phép thêm hợp đồng 7. Hệ thống lưu hợp đồng
Luồng sự kiện phụ:	<p>A. Thông tin không đúng yêu cầu</p> <ol style="list-style-type: none"> 1. Hệ thống báo lỗi 2. Người dùng nhập lại thông tin không đúng yêu cầu

Bảng 3.6: Đặc tả use case thêm hợp đồng

- Đặc tả sửa hợp đồng

Tên use case:	Sửa hợp đồng
Tác nhân:	Quản lý, nhân viên
Điều kiện tiên quyết:	<ul style="list-style-type: none"> • Người dùng phải đăng nhập • Đang ở trang Quản lý hợp đồng
Hậu điều kiện:	<ul style="list-style-type: none"> • Thêm thành công hợp đồng
Luồng sự kiện chính:	<ol style="list-style-type: none"> 1. Người dùng chọn hợp đồng mong muốn chỉnh sửa 2. Người dùng chọn nút “Sửa” 3. Hệ thống hiển thị form sửa hợp đồng 4. Người dùng nhập thông tin chỉnh sửa 5. Người dùng nhấn nút “Lưu” 6. Hệ thống kiểm tra thông tin 7. Hệ thống cho phép cập nhật hợp đồng 8. Hệ thống cập nhật hợp đồng
Luồng sự kiện phụ:	<p>A. Thông tin không đúng yêu cầu</p> <ol style="list-style-type: none"> 1. Hệ thống báo lỗi 2. Người dùng nhập lại thông tin không đúng yêu cầu

Bảng 3.7: Đặc tả use case sửa hợp đồng

- Đặc tả import danh sách hợp đồng

Tên use case:	Import danh sách hợp đồng
Tác nhân:	Quản lý, nhân viên
Điều kiện tiên quyết:	<ul style="list-style-type: none"> • Người dùng phải đăng nhập • Đang ở trang Quản lý hợp đồng
Hậu điều kiện:	<ul style="list-style-type: none"> • Thêm thành công danh sách hợp đồng
Luồng sự kiện chính:	<ol style="list-style-type: none"> 1. Người dùng chọn nút “Import” 2. Hệ thống hiển thị form import 3. Người dùng chọn file cần import 4. Người dùng nhấn nút “Gửi” 5. Hệ thống kiểm tra file được gửi lên 6. Hệ thống cho phép import 7. Hệ thống lưu danh sách hợp đồng 8. Hệ thống thông báo kết quả import 9. Hệ thống import thành công

Luồng sự kiện phụ:	A. File không đúng yêu cầu 1. Hệ thống báo lỗi 2. Người dùng chọn file lại B. Nội dung danh sách nhập không đúng yêu cầu 1. Hệ thống báo lỗi 2. Người dùng chỉnh sửa lại nội dung
---------------------------	--

Bảng 3.8: Đặc tả use case import danh sách hợp đồng

- Đặc tả export danh sách hợp đồng

Tên use case:	Export danh sách hợp đồng
Tác nhân:	Quản lý, nhân viên
Điều kiện tiên quyết:	<ul style="list-style-type: none"> • Người dùng phải đăng nhập • Đang ở trang Quản lý hợp đồng
Hậu điều kiện:	<ul style="list-style-type: none"> • Tải thành công danh sách hợp đồng
Luồng sự kiện chính:	1. Người dùng chọn nút “Export” 2. Hệ thống xử lý file 3. Hệ thống gửi file về cho trình duyệt 4. Trình duyệt tự động tải file 5. Hệ thống export thành công
Luồng sự kiện phụ:	Không có

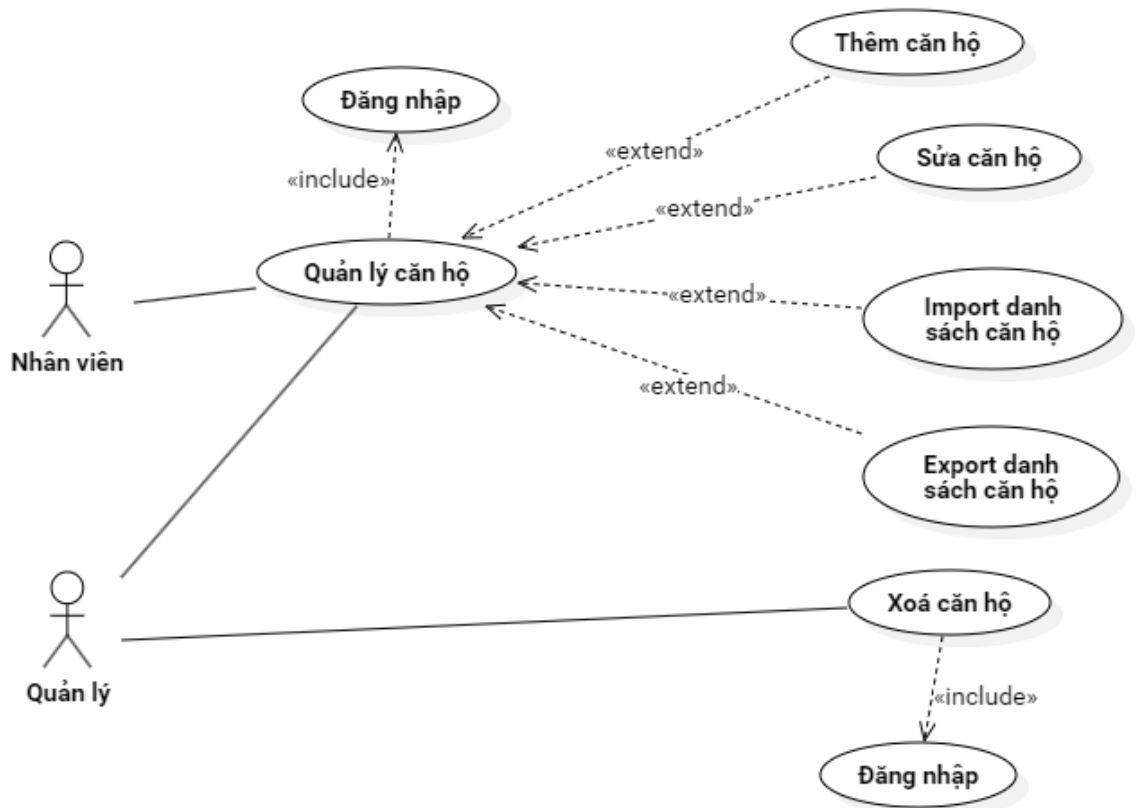
Bảng 3.9: Đặc tả use case export danh sách hợp đồng

- Đặc tả xoá hợp đồng

Tên use case:	Xoá hợp đồng
Tác nhân:	Quản lý
Điều kiện tiên quyết:	<ul style="list-style-type: none"> • Người dùng phải đăng nhập • Đang ở trang Quản lý hợp đồng
Hậu điều kiện:	<ul style="list-style-type: none"> • Xoá thành công hợp đồng
Luồng sự kiện chính:	1. Người dùng chọn hợp đồng mong muốn xoá 2. Người dùng chọn nút “Xoá” 3. Hệ thống hỏi xác nhận xoá 4. Người dùng xác nhận xoá 5. Hệ thống xoá thành công hợp đồng
Luồng sự kiện phụ:	A. Người dùng xác nhận không xoá 1. Hệ thống hiển thị trang Quản lý hợp đồng

Bảng 3.10: Đặc tả use case xoá hợp đồng

3.1.3.4. Chức năng quản lý căn hộ



Hình 3.6: Biểu đồ use case quản lý căn hộ

- Đặc tả thêm căn hộ

Tên use case:	Thêm căn hộ
Tác nhân:	Quản lý, nhân viên
Điều kiện tiên quyết:	<ul style="list-style-type: none"> Người dùng phải đăng nhập Đang ở trang Quản lý căn hộ
Hậu điều kiện:	<ul style="list-style-type: none"> Thêm thành công căn hộ mới
Luồng sự kiện chính:	<ol style="list-style-type: none"> 1. Người dùng chọn nút “Thêm” 2. Hệ thống hiển thị form thêm căn hộ 3. Người dùng nhập thông tin bắt buộc 4. Người dùng nhấn nút “Lưu” 5. Hệ thống kiểm tra thông tin 6. Hệ thống cho phép thêm căn hộ 7. Hệ thống lưu căn hộ

Luồng sự kiện phụ:	A. Thông tin không đúng yêu cầu 1. Hệ thống báo lỗi 2. Người dùng nhập lại thông tin không đúng yêu cầu
---------------------------	---

Bảng 3.11: Đặc tả use case thêm căn hộ

- Đặc tả sửa căn hộ

Tên use case:	Sửa căn hộ
Tác nhân:	Quản lý, nhân viên
Điều kiện tiên quyết:	<ul style="list-style-type: none"> • Người dùng phải đăng nhập • Đang ở trang Quản lý căn hộ
Hậu điều kiện:	<ul style="list-style-type: none"> • Thêm thành công căn hộ
Luồng sự kiện chính:	1. Người dùng chọn căn hộ mong muốn chỉnh sửa 2. Người dùng chọn nút “Sửa” 3. Hệ thống hiển thị form sửa căn hộ 4. Người dùng nhập thông tin chỉnh sửa 5. Người dùng nhấn nút “Lưu” 6. Hệ thống kiểm tra thông tin 7. Hệ thống cho phép cập nhật căn hộ 8. Hệ thống cập nhật căn hộ
Luồng sự kiện phụ:	A. Thông tin không đúng yêu cầu 1. Hệ thống báo lỗi 2. Người dùng nhập lại thông tin không đúng yêu cầu

Bảng 3.12: Đặc tả use case sửa căn hộ

- Đặc tả import danh sách căn hộ

Tên use case:	Import danh sách căn hộ
Tác nhân:	Quản lý, nhân viên
Điều kiện tiên quyết:	<ul style="list-style-type: none"> • Người dùng phải đăng nhập • Đang ở trang Quản lý căn hộ
Hậu điều kiện:	<ul style="list-style-type: none"> • Thêm thành công danh sách căn hộ
Luồng sự kiện chính:	1. Người dùng chọn nút “Import” 2. Hệ thống hiển thị form import 3. Người dùng chọn file cần import 4. Người dùng nhấn nút “Gửi” 5. Hệ thống kiểm tra file được gửi lên 6. Hệ thống cho phép import 7. Hệ thống lưu danh sách căn hộ 8. Hệ thống thông báo kết quả import 9. Hệ thống import thành công

Luồng sự kiện phụ:	A. File không đúng yêu cầu 1. Hệ thống báo lỗi 2. Người dùng chọn file lại B. Nội dung danh sách nhập không đúng yêu cầu 1. Hệ thống báo lỗi 2. Người dùng chỉnh sửa lại nội dung
---------------------------	--

Bảng 3.13: Đặc tả use case import danh sách căn hộ

- Đặc tả export danh sách căn hộ

Tên use case:	Export danh sách căn hộ
Tác nhân:	Quản lý, nhân viên
Điều kiện tiên quyết:	<ul style="list-style-type: none"> • Người dùng phải đăng nhập • Đang ở trang Quản lý căn hộ
Hậu điều kiện:	<ul style="list-style-type: none"> • Tải thành công danh sách căn hộ
Luồng sự kiện chính:	1. Người dùng chọn nút “Export” 2. Hệ thống xử lý file 3. Hệ thống gửi file về cho trình duyệt 4. Trình duyệt tự động tải file 5. Hệ thống export thành công
Luồng sự kiện phụ:	Không có

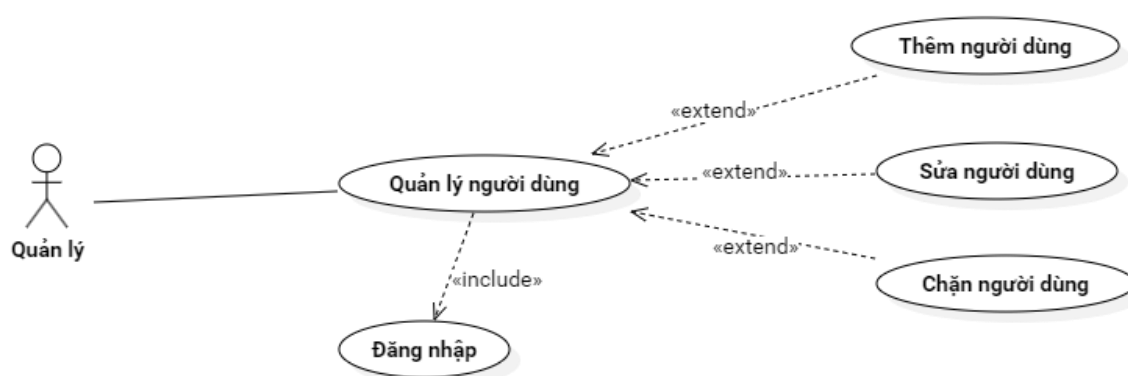
Bảng 3.14: Đặc tả use case export danh sách căn hộ

- Đặc tả xoá căn hộ

Tên use case:	Xoá căn hộ
Tác nhân:	Quản lý
Điều kiện tiên quyết:	<ul style="list-style-type: none"> • Người dùng phải đăng nhập • Đang ở trang Quản lý căn hộ
Hậu điều kiện:	<ul style="list-style-type: none"> • Xoá thành công căn hộ
Luồng sự kiện chính:	1. Người dùng chọn căn hộ mong muốn xoá 2. Người dùng chọn nút “Xoá” 3. Hệ thống hỏi xác nhận xoá 4. Người dùng xác nhận xoá 5. Hệ thống xoá thành công căn hộ
Luồng sự kiện phụ:	A. Người dùng xác nhận không xoá 1. Hệ thống hiển thị trang Quản lý căn hộ

Bảng 3.15: Đặc tả use case xoá căn hộ

3.1.3.5. Chức năng quản lý người dùng



Hình 3.7: Biểu đồ use case quản lý người dùng

- Đặc tả thêm căn hộ

Tên use case:	Thêm người dùng
Tác nhân:	Quản lý
Điều kiện tiên quyết:	<ul style="list-style-type: none"> Người dùng phải đăng nhập Đang ở trang Quản lý người dùng
Hậu điều kiện:	<ul style="list-style-type: none"> Thêm thành công người dùng mới
Luồng sự kiện chính:	<ol style="list-style-type: none"> Người dùng chọn nút “Thêm” Hệ thống hiển thị form thêm người dùng Người dùng nhập thông tin bắt buộc Người dùng nhấn nút “Lưu” Hệ thống kiểm tra thông tin Hệ thống cho phép thêm người dùng Hệ thống lưu người dùng
Luồng sự kiện phụ:	A. Thông tin không đúng yêu cầu <ol style="list-style-type: none"> Hệ thống báo lỗi Người dùng nhập lại thông tin không đúng yêu cầu

Bảng 3.16: Đặc tả use case thêm người dùng

- Đặc tả sửa người dùng

Tên use case:	Sửa người dùng
Tác nhân:	Quản lý
Điều kiện tiên quyết:	<ul style="list-style-type: none"> Người dùng phải đăng nhập Đang ở trang Quản lý người dùng
Hậu điều kiện:	<ul style="list-style-type: none"> Sửa thành công người dùng
Luồng sự kiện chính:	<ol style="list-style-type: none"> Người dùng chọn người dùng mong muốn chỉnh sửa Người dùng chọn nút “Sửa”

	3. Hệ thống hiển thị form sửa người dùng 4. Người dùng nhập thông tin chỉnh sửa 5. Người dùng nhấn nút “Lưu” 6. Hệ thống kiểm tra thông tin 7. Hệ thống cho phép cập nhật người dùng 8. Hệ thống cập nhật người dùng
Luồng sự kiện phụ:	A. Thông tin không đúng yêu cầu 1. Hệ thống báo lỗi 2. Người dùng nhập lại thông tin không đúng yêu cầu

Bảng 3.17: Đặc tả use case sửa người dùng

- Đặc tả chặn người dùng

Tên use case:	Chặn người dùng
Tác nhân:	Quản lý
Điều kiện tiên quyết:	<ul style="list-style-type: none"> • Người dùng phải đăng nhập • Đang ở trang Quản lý người dùng
Hậu điều kiện:	<ul style="list-style-type: none"> • Chặn thành công người dùng
Luồng sự kiện chính:	1. Người dùng chọn người dùng mong muốn chặn 2. Người dùng chọn nút “Chặn” 3. Hệ thống hỏi xác nhận chặn 4. Người dùng xác nhận chặn 5. Hệ thống chặn thành công người dùng
Luồng sự kiện phụ:	A. Người dùng xác nhận không chặn 1. Hệ thống hiển thị trang Quản lý người dùng

Bảng 3.18: Đặc tả use case chặn người dùng

3.1.3.6. Chức năng thống kê

- Đặc tả thống kê

Tên use case:	Thống kê
Tác nhân:	Quản lý
Điều kiện tiên quyết:	<ul style="list-style-type: none"> • Người dùng đã đăng nhập • Đang ở Trang chủ
Hậu điều kiện:	<ul style="list-style-type: none"> • Người dùng xem thống kê thành công
Luồng sự kiện chính:	1. Người dùng chọn loại thống kê muốn xem 2. Hệ thống hiển thị kết quả người dùng chọn
Luồng sự kiện phụ:	Không có

Bảng 3.19: Đặc tả use case thống kê

3.1.3.7. Chức năng đăng nhập

- Đặc tả đăng nhập

Tên use case:	Đăng nhập
Tác nhân:	Người dùng
Điều kiện tiên quyết:	<ul style="list-style-type: none">• Tài khoản người dùng đã được tạo sẵn• Đang ở trang Đăng nhập
Hậu điều kiện:	<ul style="list-style-type: none">• Người dùng đăng nhập ứng dụng thành công• Hệ thống ghi nhận hoạt động đăng nhập thành công
Luồng sự kiện chính:	<ol style="list-style-type: none">1. Người dùng nhập Email và Password2. Hệ thống kiểm tra thông tin3. Tài khoản đăng nhập đúng4. Hệ thống chuyển người dùng đến Trang chủ
Luồng sự kiện phụ:	A. Đăng nhập thất bại <ol style="list-style-type: none">1. Thông báo lỗi2. Về lại trang Đăng nhập

Bảng 3.20: Đặc tả use case đăng nhập

3.1.3.8. Chức năng đăng xuất

- Đặc tả đăng xuất

Tên use case:	Đăng xuất
Tác nhân:	Người dùng
Điều kiện tiên quyết:	<ul style="list-style-type: none">• Người dùng đã đăng nhập• Đang ở Trang chủ
Hậu điều kiện:	<ul style="list-style-type: none">• Người dùng đăng xuất ứng dụng thành công• Hệ thống ghi nhận hoạt động đăng xuất thành công
Luồng sự kiện chính:	<ol style="list-style-type: none">1. Người dùng chọn nút “Đăng xuất”2. Hệ thống hỏi xác nhận đăng xuất3. Hệ thống xoá phiên đăng nhập4. Hệ thống chuyển người dùng đến trang Đăng nhập
Luồng sự kiện phụ:	A. Người dùng huỷ đăng xuất <ol style="list-style-type: none">1. Hệ thống hiển thị Trang chủ

Bảng 3.21: Đặc tả use case đăng xuất

3.1.3.9. Chức năng đổi mật khẩu

- Đặc tả đổi mật khẩu

Tên use case:	Đổi mật khẩu
Tác nhân:	Người dùng

Điều kiện tiên quyết:	<ul style="list-style-type: none"> • Người dùng đã đăng nhập • Đang ở Trang chủ
Hậu điều kiện:	<ul style="list-style-type: none"> • Người dùng đổi mật khẩu thành công
Luồng sự kiện chính:	<ol style="list-style-type: none"> 1. Người dùng chọn vào Ảnh đại diện 2. Hệ thống hiển thị Trang thông tin 3. Người dùng chọn chức năng “Đổi mật khẩu” 4. Hệ thống hiển thị trang “Đổi mật khẩu” 5. Người dùng nhập thông tin đổi mật khẩu 6. Hệ thống kiểm tra thông tin 7. Hệ thống cho phép cập nhật mật khẩu mới 8. Hệ thống cập nhật thông tin 9. Hệ thống cập nhật mật khẩu mới
Luồng sự kiện phụ:	A. Thông tin không đúng yêu cầu <ol style="list-style-type: none"> 1. Hệ thống báo lỗi 2. Người dùng nhập lại thông tin không đúng yêu cầu

Bảng 3.22: Đặc tả use case đổi mật khẩu

3.1.3.10. Chức năng quên mật khẩu

- Đặc tả quên mật khẩu

Tên use case:	Quên mật khẩu
Tác nhân:	Người dùng
Điều kiện tiên quyết:	<ul style="list-style-type: none"> • Tài khoản người dùng được tạo sẵn • Người dùng không nhớ mật khẩu • Người dùng đang ở trang Đăng nhập
Hậu điều kiện:	<ul style="list-style-type: none"> • Người dùng đặt lại được mật khẩu mới • Đăng nhập thành công
Luồng sự kiện chính:	<ol style="list-style-type: none"> 1. Chọn Quên mật khẩu 2. Hệ thống hiển thị trang Quên mật khẩu 3. Nhập Email tài khoản quên 4. Hệ thống mật khẩu mới 5. Hệ thống hiển thị trang Đăng nhập
Luồng sự kiện phụ:	A. Email không hợp lệ <ol style="list-style-type: none"> 1. Hệ thống thông báo Email không hợp lệ

Bảng 3.23: Đặc tả use case quên mật khẩu

3.1.3.11. Chức năng cập nhật thông tin

- Đặc tả cập nhật thông tin

Tên use case:	Cập nhật thông tin
----------------------	--------------------

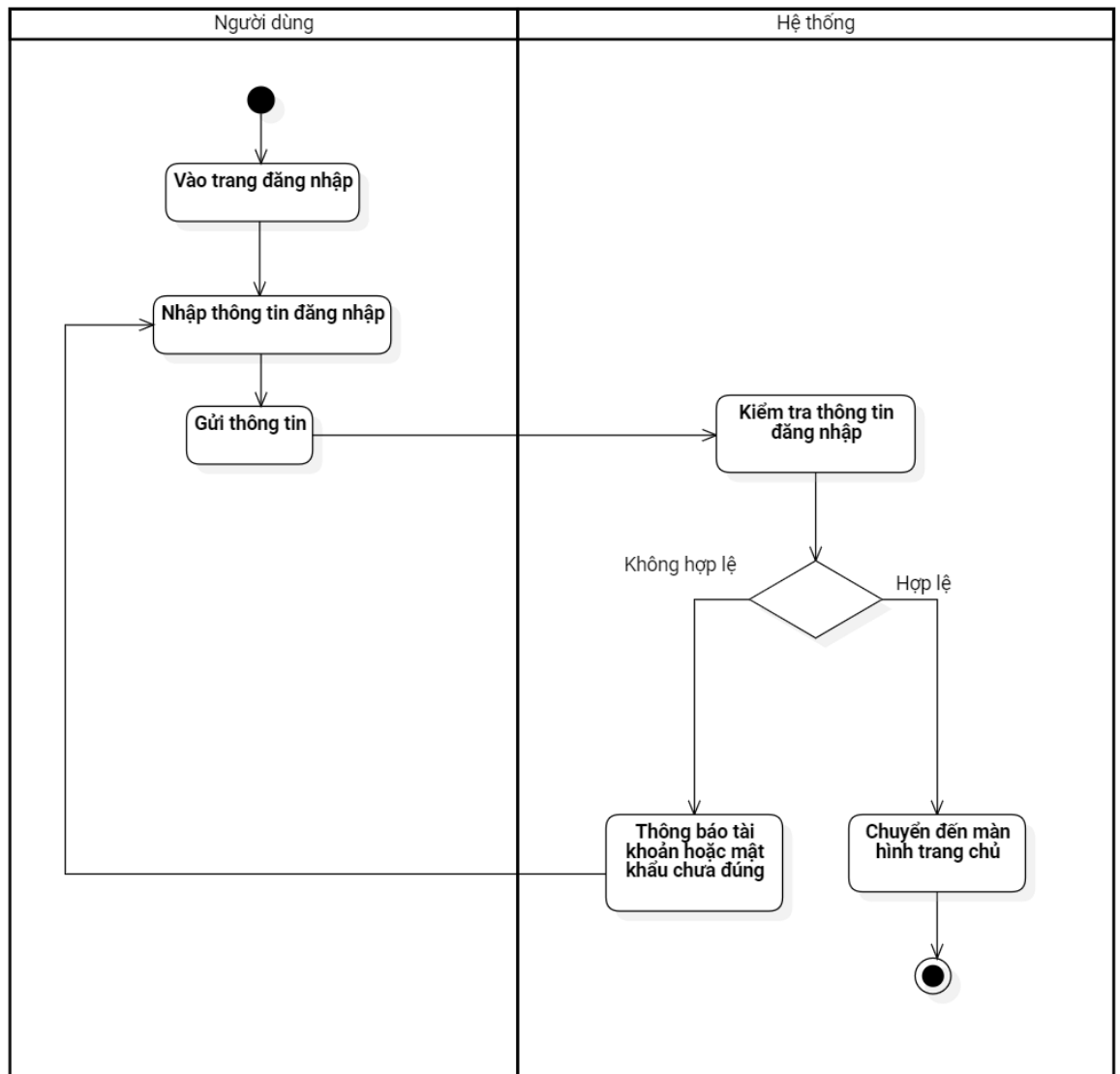
Tác nhân:	Người dùng
Điều kiện tiên quyết:	<ul style="list-style-type: none"> • Người dùng đã đăng nhập • Đang ở Trang chủ
Hậu điều kiện:	<ul style="list-style-type: none"> • Người dùng cập nhật thông tin thành công
Luồng sự kiện chính:	<ol style="list-style-type: none"> 1. Người dùng chọn vào Ảnh đại diện 2. Hệ thống hiển thị Trang thông tin 3. Người dùng nhập thông tin cần chỉnh sửa 4. Hệ thống kiểm tra thông tin 5. Hệ thống cho phép cập nhật thông tin 6. Hệ thống cập nhật thông tin
Luồng sự kiện phụ:	A. Thông tin không đúng yêu cầu <ol style="list-style-type: none"> 1. Hệ thống báo lỗi 2. Người dùng nhập lại thông tin không đúng yêu cầu

Bảng 3.24: Đặc tả use case cập nhật thông tin

3.1.4. Biểu đồ hoạt động

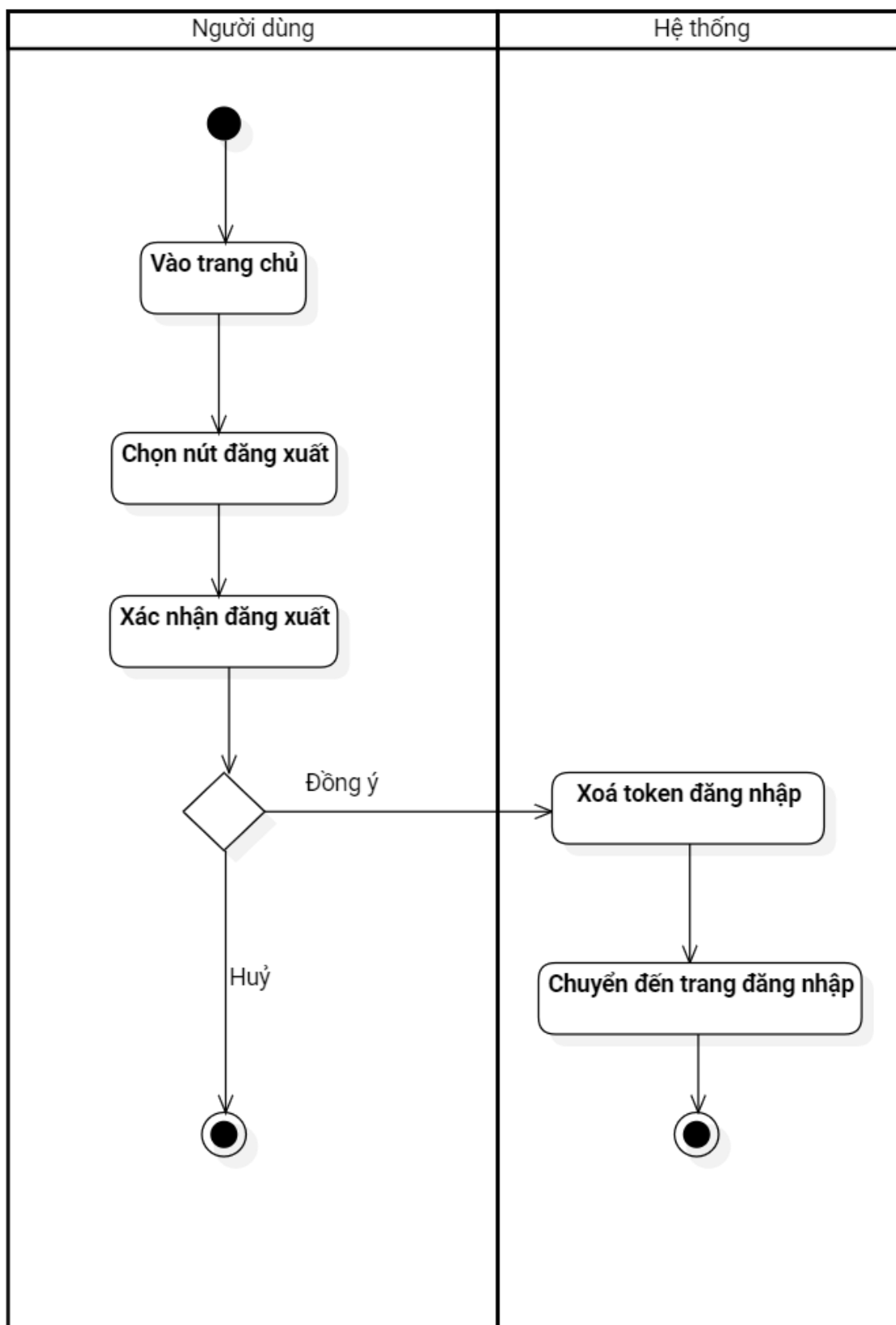
3.1.4.1. Chức năng người dùng chung

3.1.4.1.1. Biểu đồ hoạt động đăng nhập



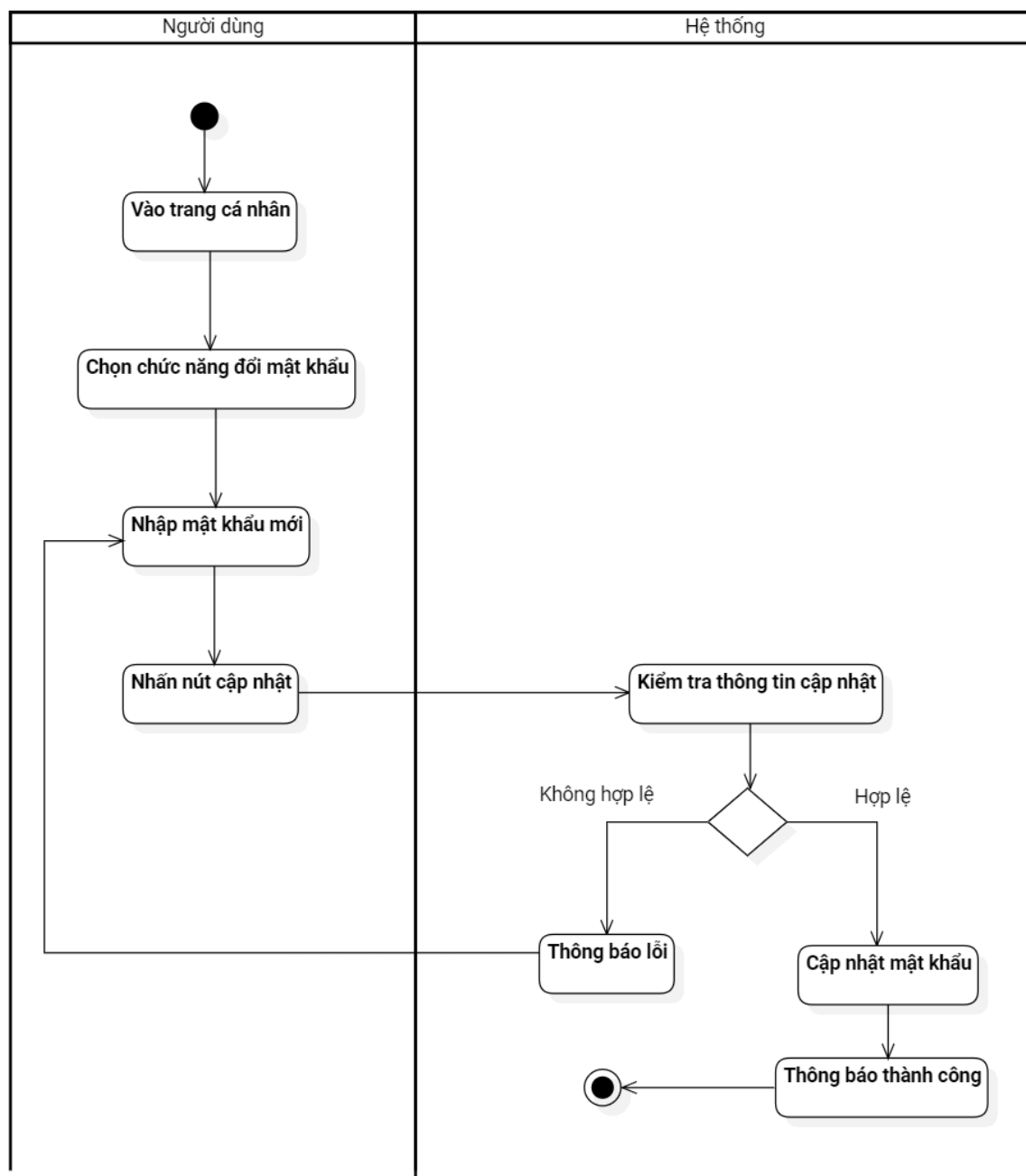
Hình 3.8: Biểu đồ hoạt động đăng nhập

3.1.4.1.2. Biểu đồ hoạt động đăng xuất



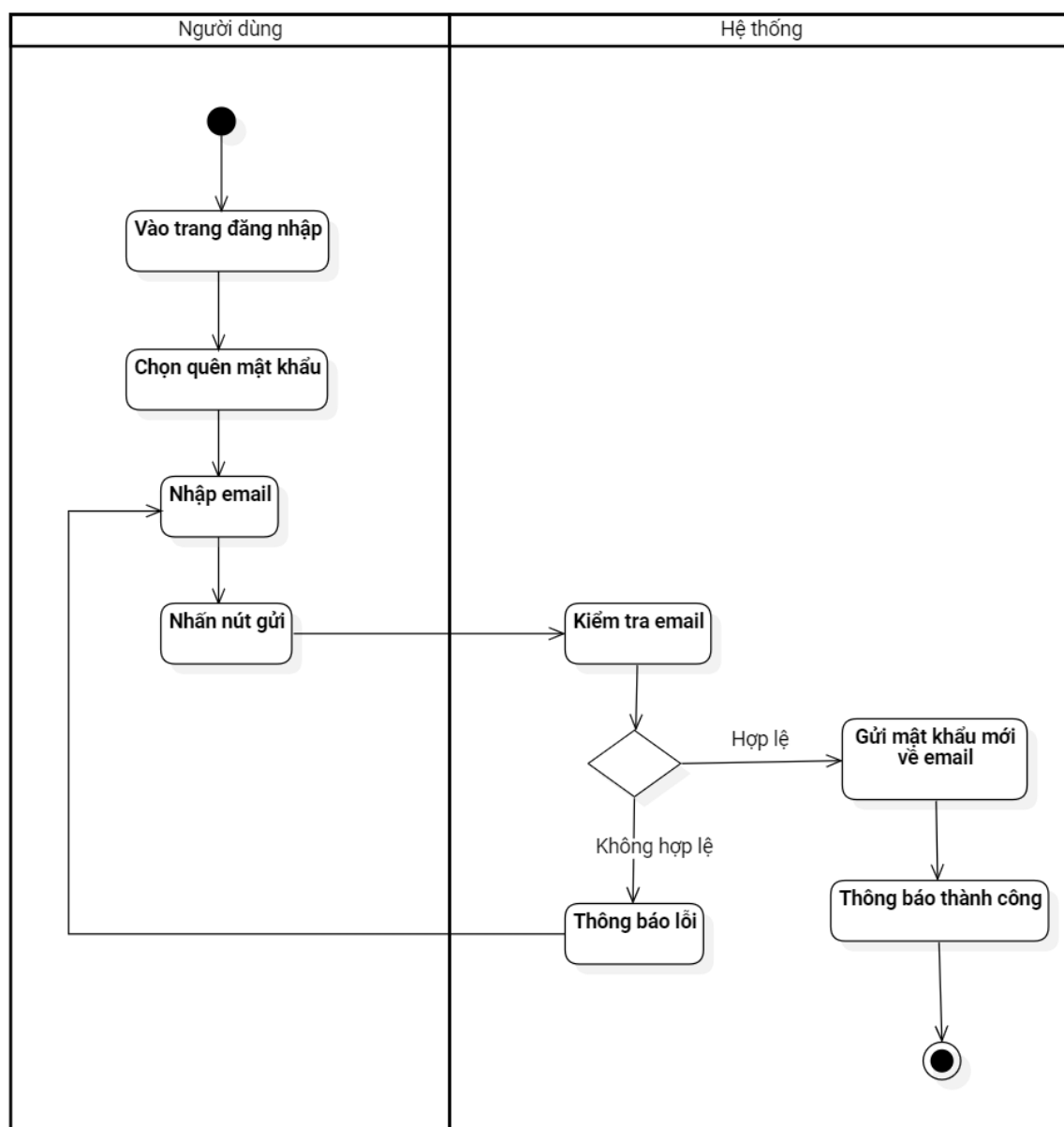
Hình 3.9: Biểu đồ hoạt động đăng xuất

3.1.4.1.3. Biểu đồ hoạt động đổi mật khẩu



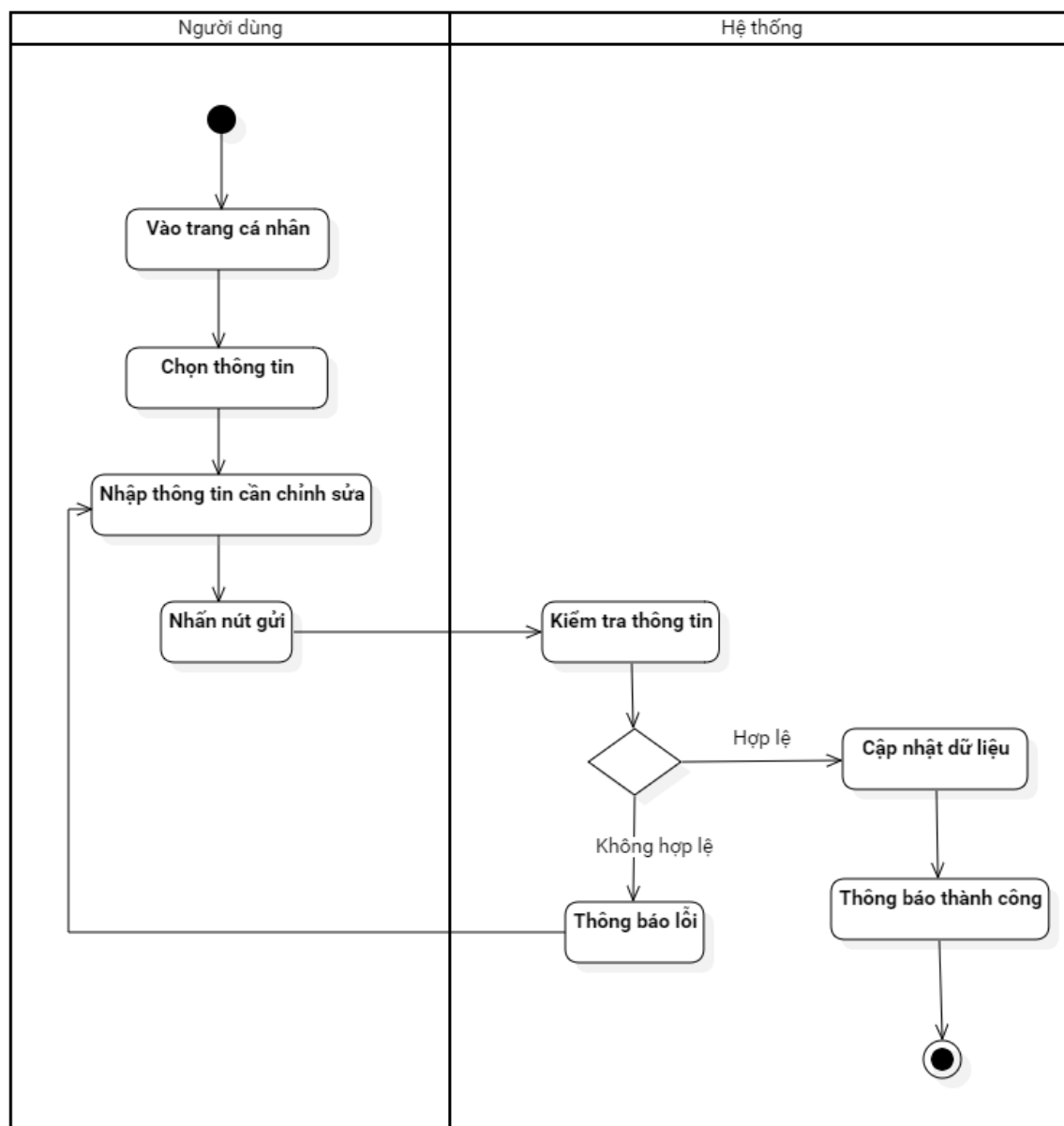
Hình 3.10: Biểu đồ hoạt động đổi mật khẩu

3.1.4.1.4. Biểu đồ hoạt động quên mật khẩu



Hình 3.11: Biểu đồ hoạt động quên mật khẩu

3.1.4.1.5. Biểu đồ hoạt động cập nhật thông tin

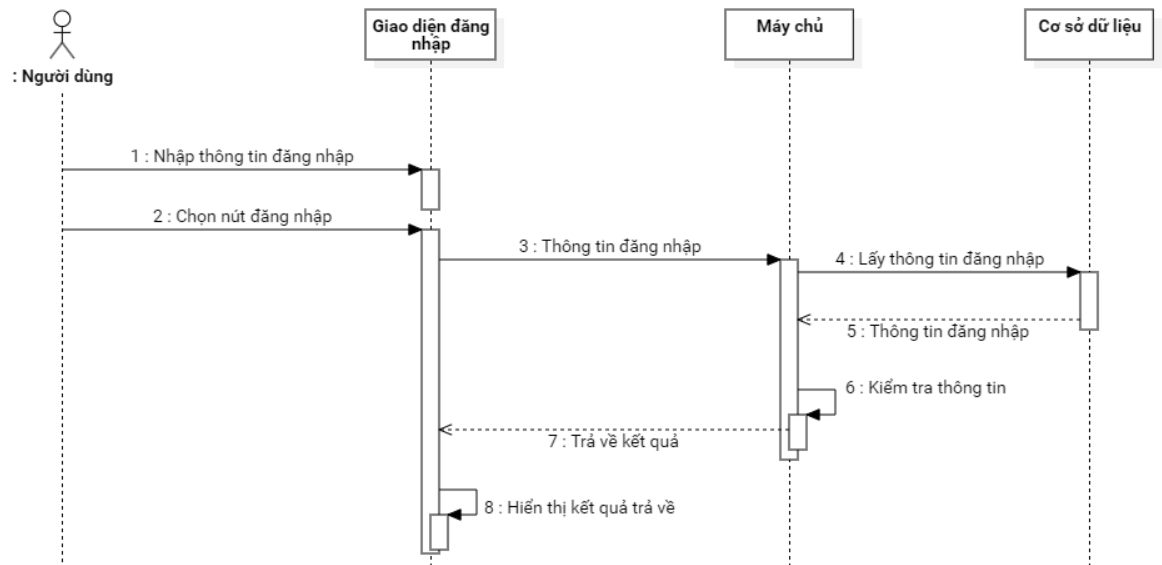


Hình 3.12: Biểu đồ hoạt động cập nhật thông tin

3.1.5. Biểu đồ tuần tự

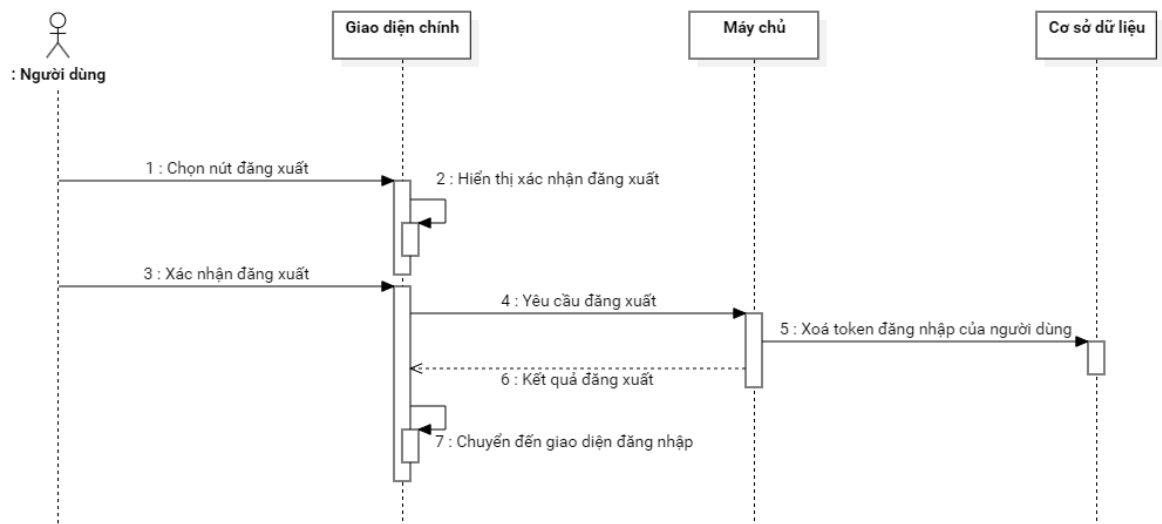
3.1.5.1. Chức năng người dùng chung

3.1.5.1.1. Biểu đồ tuần tự đăng nhập



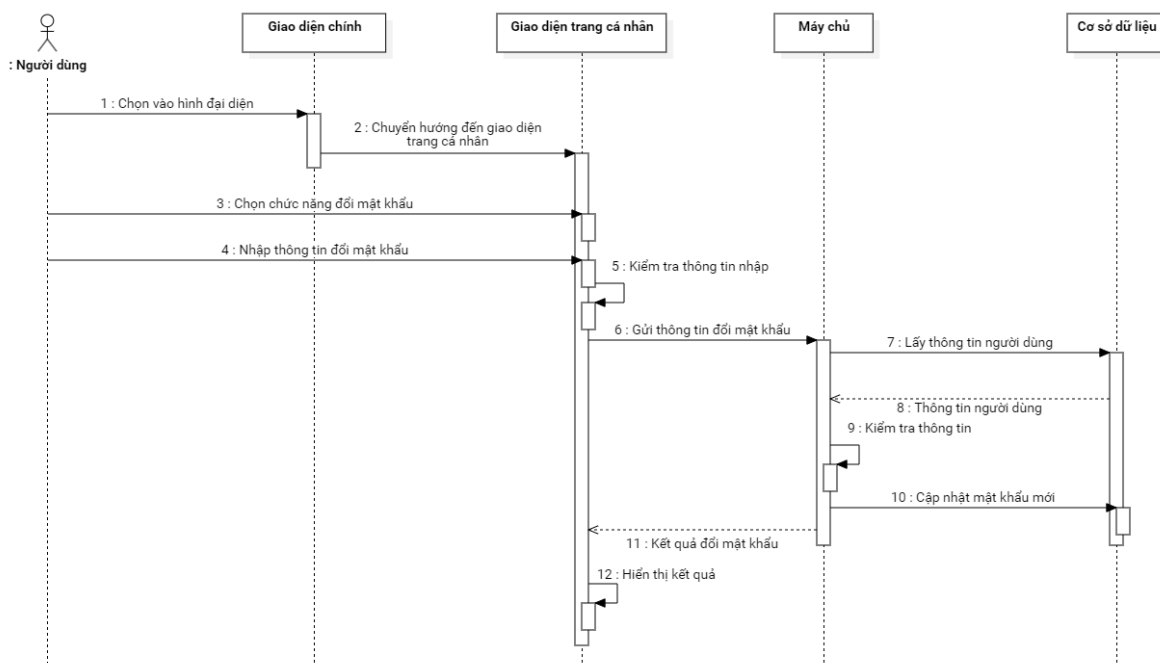
Hình 3.13: Biểu đồ tuần tự đăng nhập

3.1.5.1.2. Biểu đồ tuần tự đăng xuất



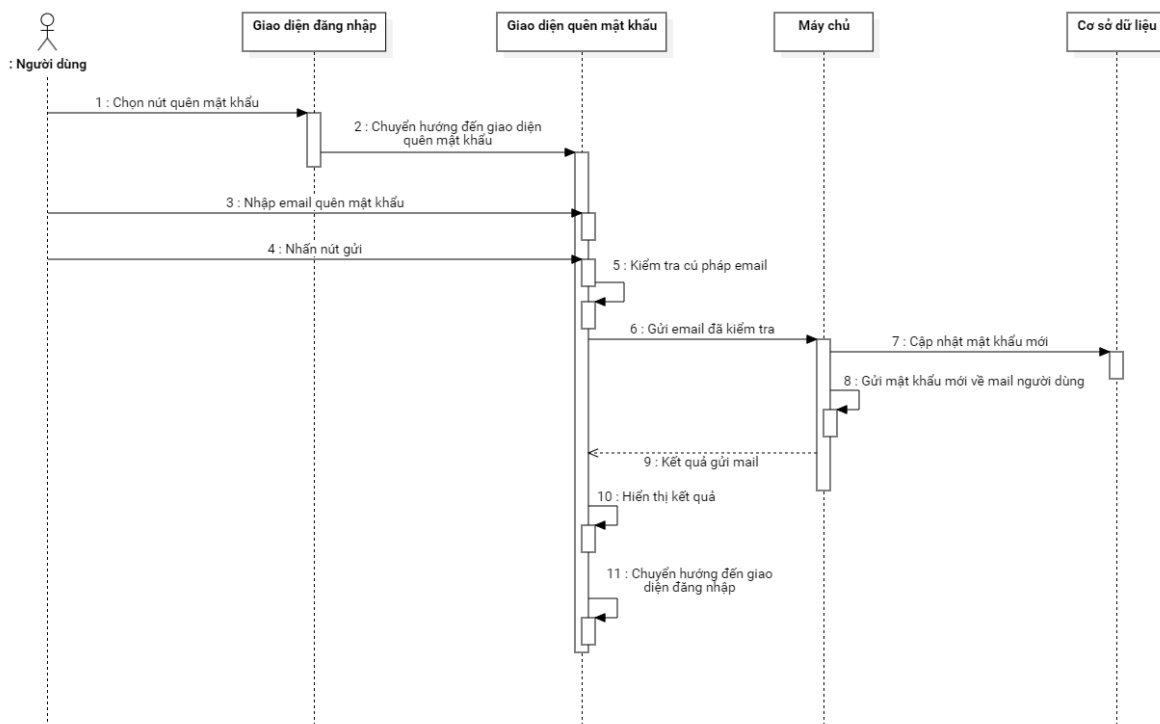
Hình 3.14: Biểu đồ tuần tự đăng xuất

3.1.5.1.3. Biểu đồ tuần tự đổi mật khẩu



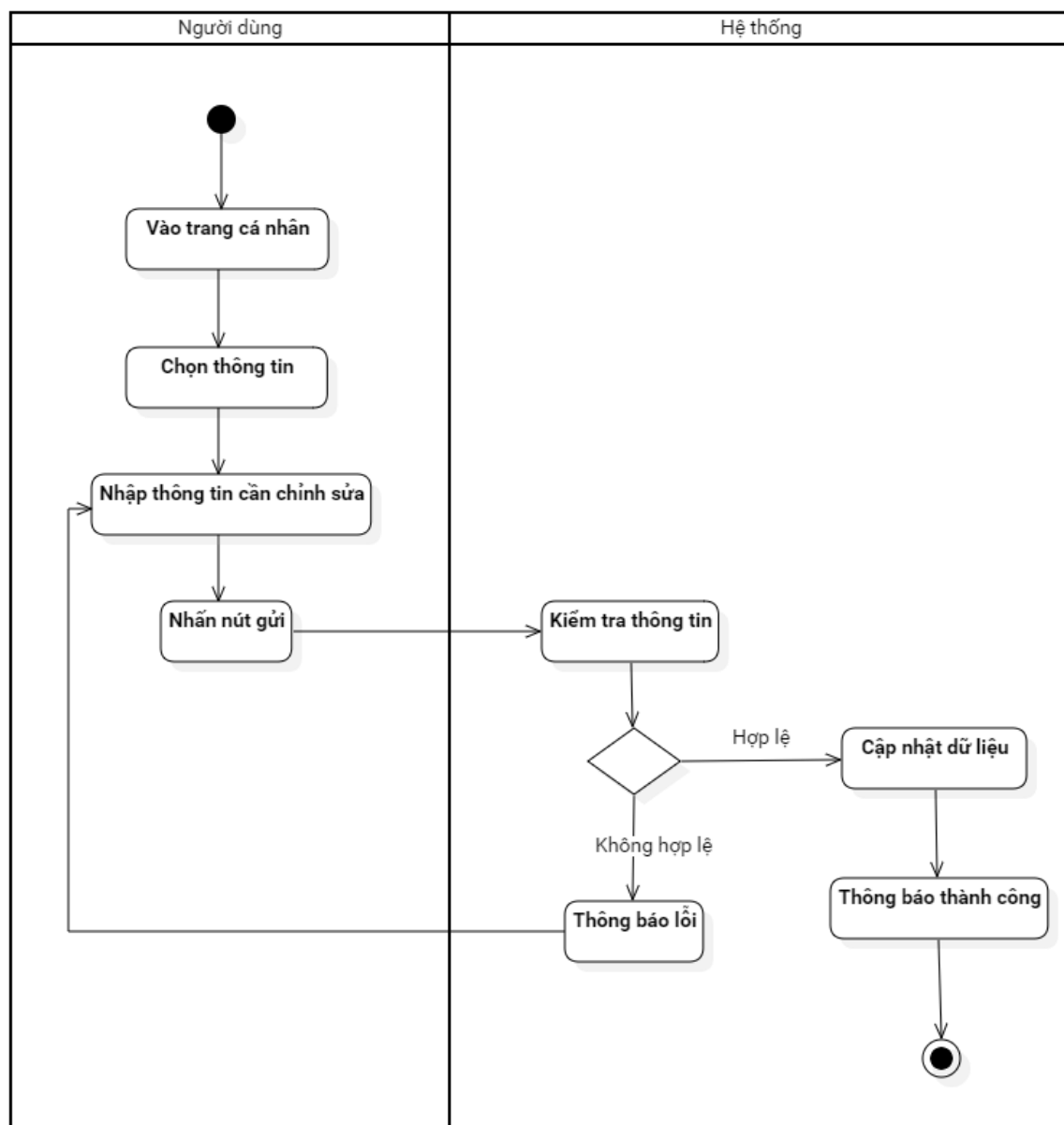
Hình 3.15: Biểu đồ tuần tự đổi mật khẩu

3.1.5.1.4. Biểu đồ tuần tự quên mật khẩu



Hình 3.16: Biểu đồ tuần tự quên mật khẩu

3.1.5.1.5. Biểu đồ tuần tự cập nhật thông tin



Hình 3.17: Biểu đồ tuần tự cập nhật thông tin

3.2. Danh sách các API

3.2.1. Auth

Phương thức	Đường dẫn	Ý nghĩa
POST	/auth/login	Đăng nhập
GET	/auth/logout	Đăng xuất

Phương thức	Đường dẫn	Ý nghĩa
POST	/auth/resetPassword	Đặt lại mật khẩu
POST	/auth/refreshToken	Làm mới token

Bảng 3.25: Danh sách API Auth

3.2.2. User

Phương thức	Đường dẫn	Diễn giải
GET	/users?page{int}&pageSize{int}&sortBy={string}	Danh sách người dùng
GET	/users/{id}	Lấy người dùng theo id
GET	/users/details	Lấy người dùng theo token
GET	/users/export	Xuất danh sách người dùng
GET	/users/statistic	Thống kê người dùng
POST	/users/update/{id}	Cập nhật người dùng theo id
POST	/users/changePassword/{id}	Đổi mật khẩu của người dùng theo id
POST	/users/block/{id}	Chặn người dùng theo id
POST	/users/unblock/{id}	Bỏ chặn người dùng theo id
POST	/users/add	Tạo mới người dùng

Bảng 3.26: Danh sách API User

3.2.3. Customer

Phương thức	Đường dẫn	Ý nghĩa
GET	/customers?page{int}&pageSize{int}&sortBy={string}	Lấy danh sách khách hàng
GET	/customers/{id}	Lấy khách hàng theo id

Phương thức	Đường dẫn	Ý nghĩa
GET	/customers/statistic	Thống kê khách hàng
GET	/customers/search	Tìm kiếm khách hàng
GET	/customers/export	Xuất danh sách khách hàng
POST	/customers/update/{id}	Cập nhật khách hàng theo id
POST	/customers/import	Nhập danh sách khách hàng
POST	/customers/add	Tạo mới khách hàng
DELETE	/customers/delete/{id}	Xoá khách hàng theo id

Bảng 3.27: Danh sách API Customer

3.2.4. Apartment

Phương thức	Đường dẫn	Ý nghĩa
GET	/apartments?page{int}&pageSize{int}&sortBy={string}	Lấy danh sách căn hộ
GET	/apartments/{id}	Lấy căn hộ theo id
GET	/apartments/statistic	Thống kê căn hộ
GET	/apartments/search	Tìm kiếm căn hộ
GET	/apartments/export	Xuất danh sách căn hộ
POST	/apartments/update/{id}	Cập nhật căn hộ theo id
POST	/apartments/import	Nhập danh sách căn hộ
POST	/apartments/add	Tạo mới căn hộ
DELETE	/apartments/delete/{id}	Xoá căn hộ theo id

Bảng 3.28: Danh sách API Apartment

3.2.5. Contract

Phương thức	Đường dẫn	Ý nghĩa
GET	/contracts?page{int}&pageSize{int}&sortBy={string}	Lấy danh sách hợp đồng
GET	/contracts/{id}	Lấy hợp đồng theo id
GET	/contracts/statistic	Thống kê hợp đồng
GET	/contracts/search	Tìm kiếm hợp đồng
GET	/contracts/export	Xuất danh sách hợp đồng
POST	/contracts/update/{id}	Cập nhật hợp đồng theo id
POST	/contracts/add	Tạo mới hợp đồng
DELETE	/contracts/delete/{id}	Xoá hợp đồng theo id

Bảng 3.29: Danh sách API Contract

3.3. Đặc tả các bảng dữ liệu

3.3.1. Bảng User

STT	Tên trường	Kiểu dữ liệu	Kích cỡ	Khoá	Diễn giải
1	Id	Varchar	255	Chính	Mã người dùng
2	Access_Token	Varchar	255		Mã truy cập
3	Active	Bit			Trạng thái tài khoản
4	Create_Date	Date			Ngày tạo tài khoản
5	Email	Varchar	255		Địa chỉ thư điện tử
6	Exp_Refresh_Token	Date Time			Thời gian hết hạn của mã làm mới
7	Full_Name	Varchar	255		Họ tên người dùng

STT	Tên trường	Kiểu dữ liệu	Kích cỡ	Khoá	Diễn giải
8	Password	Varchar	255		Mật khẩu người dùng
9	Refresh-Token	Varchar			Mã làm mới
10	Username	Varchar	255		Mã người dùng
11	Role_Id	Int		Ngoại	Mã vai trò

Bảng 3.30: Đặc tả bảng User

3.3.2. Bảng Role

STT	Tên trường	Kiểu dữ liệu	Kích cỡ	Khoá	Diễn giải
1	Id	Int		Chính	Mã vai trò
2	Name	Varchar	20		Tên vai trò

Bảng 3.31: Đặc tả bảng Role

3.3.3. Bảng Customer

STT	Tên trường	Kiểu dữ liệu	Kích cỡ	Khoá	Diễn giải
1	Id	Varchar	255	Chính	Mã khách hàng
2	Address	Varchar	255		Địa chỉ khách hàng
3	Citizen_Id	Varchar	255		Số căn cước công dân
4	Date_Of_Birth	Date			Ngày sinh khách hàng
5	Full_Name	Varchar	100		Họ tên khách hàng
6	Phone_Number	Varchar	20		Số điện thoại khách hàng

Bảng 3.32: Đặc tả bảng Customer

3.3.4. Bảng Apartment

STT	Tên trường	Kiểu dữ liệu	Kích cỡ	Khoá	Diễn giải
1	Id	Varchar	255	Chính	Mã căn hộ
2	Address	Varchar	255		Địa chỉ căn hộ
3	Number_Of_Room	Int			Số lượng phòng
4	Retail_Price	Double			Giá thuê

Bảng 3.33: Đặc tả bảng Apartment

3.3.5. Bảng Contract

STT	Tên trường	Kiểu dữ liệu	Kích cỡ	Khoá	Diễn giải
1	Id	Varchar	255	Chính	Mã hợp đồng
2	Create_Date	Date			Ngày tạo hợp đồng
3	Start_Date	Date			Ngày bắt đầu thuê
4	End_Date	Date			Ngày kết thúc thuê
5	Retail_Price	Double			Giá thuê
6	Total	Double			Tổng tiền thuê
7	Apartment_Id	Varchar	255	Ngoại	Mã căn hộ
8	Customer_Id	Varchar	255	Ngoại	Mã khách hàng
9	User_Id	Varchar	255	Ngoại	Mã người dùng

Bảng 3.34: Đặc tả bảng Contract

CHƯƠNG 4: XÂY DỰNG CHƯƠNG TRÌNH

4.1. Tổng quan các công cụ, công nghệ hỗ trợ xây dựng hệ thống

4.1.1. Phát triển Backend

4.1.1.1. IntelliJ IDEA

IntelliJ IDEA là một môi trường phát triển tích hợp (IDE) mạnh mẽ được thiết kế đặc biệt cho lập trình Java, và cũng có khả năng hỗ trợ một số ngôn ngữ khác như Node.js, Python hay nhiều ngôn ngữ khác. Phiên bản đầu tiên của IntelliJ IDEA được phát hành vào tháng 1/2001, từ đó đến nay nó đã trở thành một trong những IDE hàng đầu trong cộng đồng phát triển phần mềm.

Mặc dù ban đầu được biết đến chủ yếu là một IDE Java, IntelliJ IDEA đã mở rộng sự hỗ trợ của mình cho nhiều ngôn ngữ và framework khác nhau. Nó cung cấp một loạt các tính năng thông minh như trình soạn thảo mã thông minh, công cụ phân tích mã và bộ công cụ refactorings giúp tối ưu hóa quy trình phát triển phần mềm. [2]

4.1.1.2. MySQL

MySQL là hệ quản trị cơ sở dữ liệu mã nguồn mở phổ biến hàng đầu trên thế giới (gọi tắt là RDBMS) và đặc biệt MySQL được ưa chuộng trong quá trình xây dựng, phát triển ứng dụng. MySQL được đánh giá là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có khả năng thay đổi mô hình sử dụng phù hợp với điều kiện công việc. MySQL hoạt động trên nhiều hệ điều hành, cung cấp một hệ thống lớn các hàm tiện ích rất mạnh.

MySQL thích hợp với các ứng dụng có truy cập cơ sở dữ liệu trên Internet nhờ vào tốc độ cũng như tính bảo mật cao. MySQL có thể tải miễn phí từ trang chủ với nhiều phiên bản cho các hệ điều hành khác nhau như phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix,... [3]

4.1.1.3. Java Spring Boot

Spring Boot là một trong số các module của Spring framework chuyên cung cấp các tính năng RAD (Rapid Application Development) cho phép tạo ra và phát triển các ứng dụng độc lập dựa trên Spring một cách nhanh chóng.

Spring Boot ra đời với mục đích loại bỏ những cấu hình phức tạp của Spring, nó không yêu cầu cấu hình XML và nâng cao năng suất cho các nhà phát triển. Với sự góp mặt của Spring Boot, hệ sinh thái Spring đã trở nên mạnh mẽ, phổ biến và hiệu quả hơn bao giờ hết.

4.1.1.4. JSON Web Token (JWT)

JWT là một phương tiện đại diện cho các yêu cầu chuyển giao giữa hai bên Client – Server, các thông tin trong chuỗi JWT được định dạng bằng JSON. Trong

đó chuỗi Token phải có 3 phần là header, payload và signature được ngăn bằng dấu “.”.

4.1.2. Phát triển Frontend

4.1.2.1. Visual Studio Code

Visual Studio Code là trình soạn thảo, biên tập lập trình mã nguồn miễn phí được sử dụng trên 3 nền tảng đó là: Windows, macOS và Linux được xây dựng, phát triển bởi Microsoft. Visual Studio Code được các chuyên gia công nghệ thông tin đánh giá cao, nó là sự kết hợp hoàn hảo giữa IDE và Code Editor.

4.1.2.2. ReactJs

ReactJS được phát triển bởi Facebook và được giới thiệu lần đầu tiên vào năm 2011. Ban đầu, ReactJS được phát triển để xây dựng giao diện người dùng trên trang web Facebook, nhằm cải thiện tốc độ và hiệu suất của ứng dụng web.

Hiện nay, ReactJS đã trở thành một trong những thư viện phát triển web phổ biến nhất, được sử dụng rộng rãi bởi các công ty lớn và nhỏ trên toàn thế giới. Facebook cũng tiếp tục đầu tư phát triển và nâng cấp ReactJS để đáp ứng nhu cầu của cộng đồng phát triển.

4.2. Xây dựng chương trình

4.2.1. Chức năng tiêu biểu chung phía Backend

4.2.1.1. Đăng nhập

AuthServiceImpl.java

```
public JwtResponseDTO login(LoginDTO loginDTO) throws InvalidException {
    AuthValidator.validatorLoginDTO(loginDTO);

    User user = null;
    try {
        user = userService.getUserByUsernameOrEmail(loginDTO.getUsername());
    } catch (NotFoundException e) {
        throw new InvalidException(AuthMessage.CHECK_ACCOUNT);
    }

    if (!AuthUtils.isValidPassword(loginDTO.getPassword(), user.getPassword()))
        throw new InvalidException(AuthMessage.CHECK_ACCOUNT);

    if (!user.getActive())
```

```

        throw new InvalidException(AuthMessage.BLOCKED_USER);

    String refreshToken = tokenService.createRefreshToken(user);
    String accessToken =

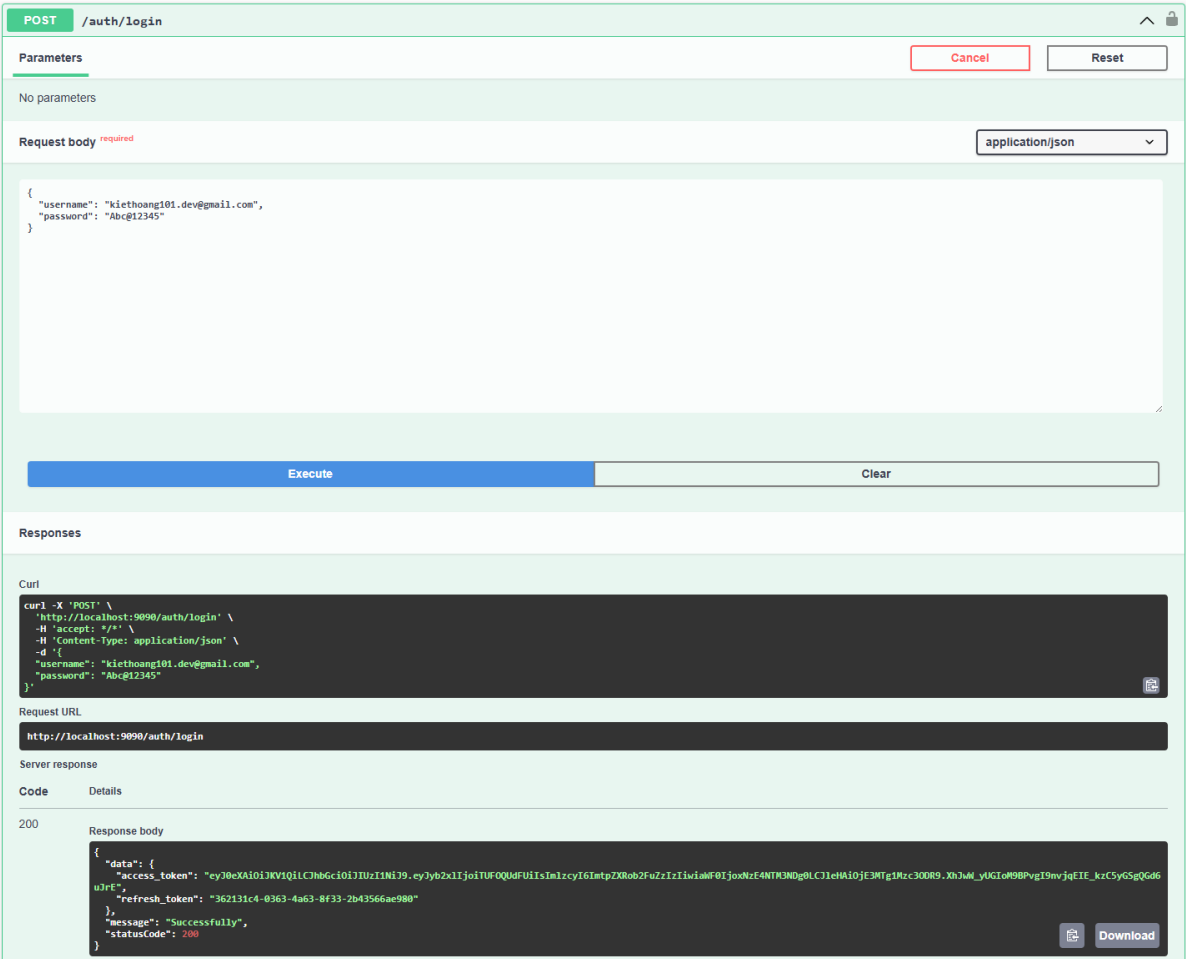
JwtUtil.createAccessToken(userService.loadUserByUsername(user.getUsername()));

    tokenService.updateAccessToken(user.getId(), accessToken);

    return JwtResponseDTO.builder()
        .accessToken(accessToken)
        .refreshToken(refreshToken)
        .build();
}

```

Bảng 4.1: Code xử lý đăng nhập



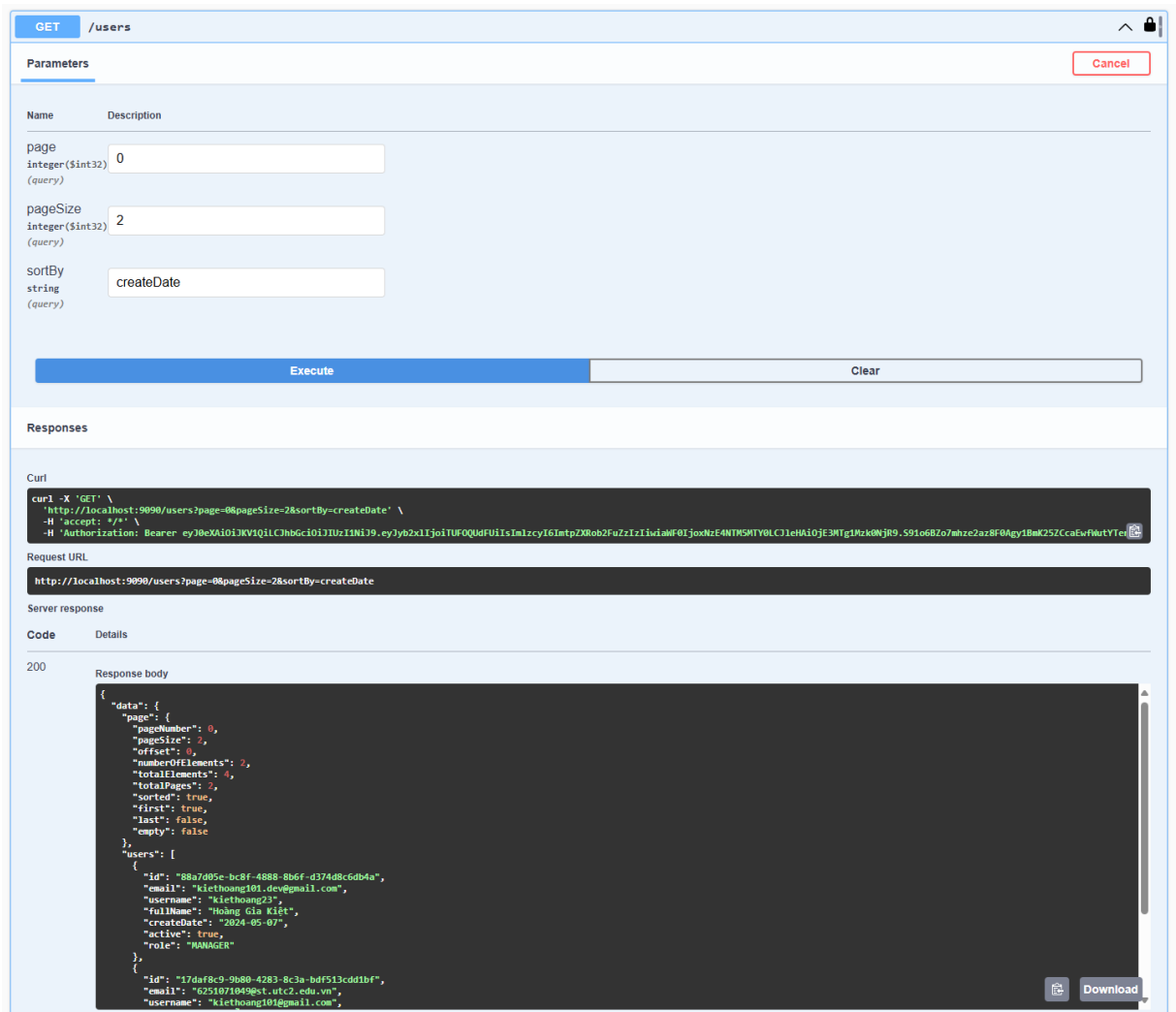
Hình 4.1: Kiểm tra API đăng nhập

4.2.1.2. Danh sách người dùng

UserServiceImpl.java

```
public Map<String, Object> getAll(Pageable pageable) throws NoContentException {  
    Map<String, Object> result = new HashMap<>();  
  
    Page<User> pageEntity =  
        userRepository.findAll(pageable);  
  
    Page<UserDto> pageDto =  
        pageEntity.map(EntityToDto::userToDto);  
  
    APIPageableDTO apiPageableDTO = new APIPageableDTO(pageDto);  
  
    result.put("page", apiPageableDTO);  
    result.put("users", pageDto.getContent());  
  
    return result;  
}
```

Bảng 4.2: Code xử lý lấy danh sách người dùng



Hình 4.2: Kiểm tra API lấy danh sách người dùng

4.2.1.3. Tạo mới người dùng

UserServiceImpl.java

```
public void createUser(UserDto userDto) {
    UserValidator.validatorUserDTO(userDto);

    checkDuplicated(userDto);

    // Default role
    Optional<Role> role =
        roleRepository.findRoleByName(ERole.STAFF);

    if (role.isEmpty()) throw new
        NotFoundException(AuthMessage.ROLE_NOT_FOUND);
}
```

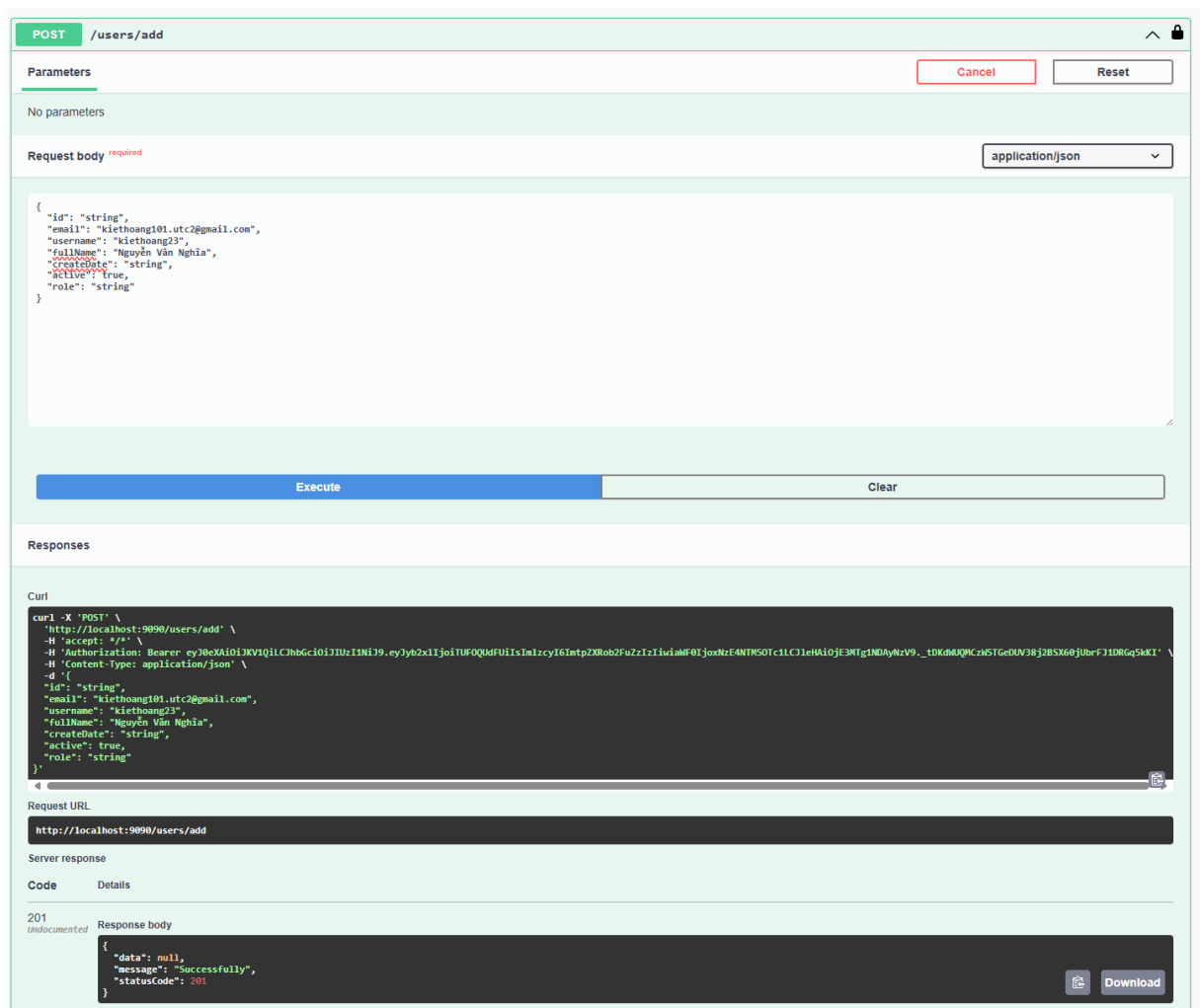
```

User user = User.builder()
    .email(userDto.getEmail())
    .username(userDto.getEmail())
    .password(AuthUtils.encodePassword(userDto.getPassword() == null ?
        DEFAULT_PASSWORD : userDto.getPassword()))
    .active(DEFAULT_STATUS)
    .fullName(userDto.getFullName())
    .role(role.get())
    .build();

userRepository.save(user);
}

```

Bảng 4.3: Code xử lý tạo mới người dùng



Hình 4.3: Kiểm tra API tạo mới người dùng

4.2.1.4. Lấy thông tin người dùng theo token

UserServiceImpl.java

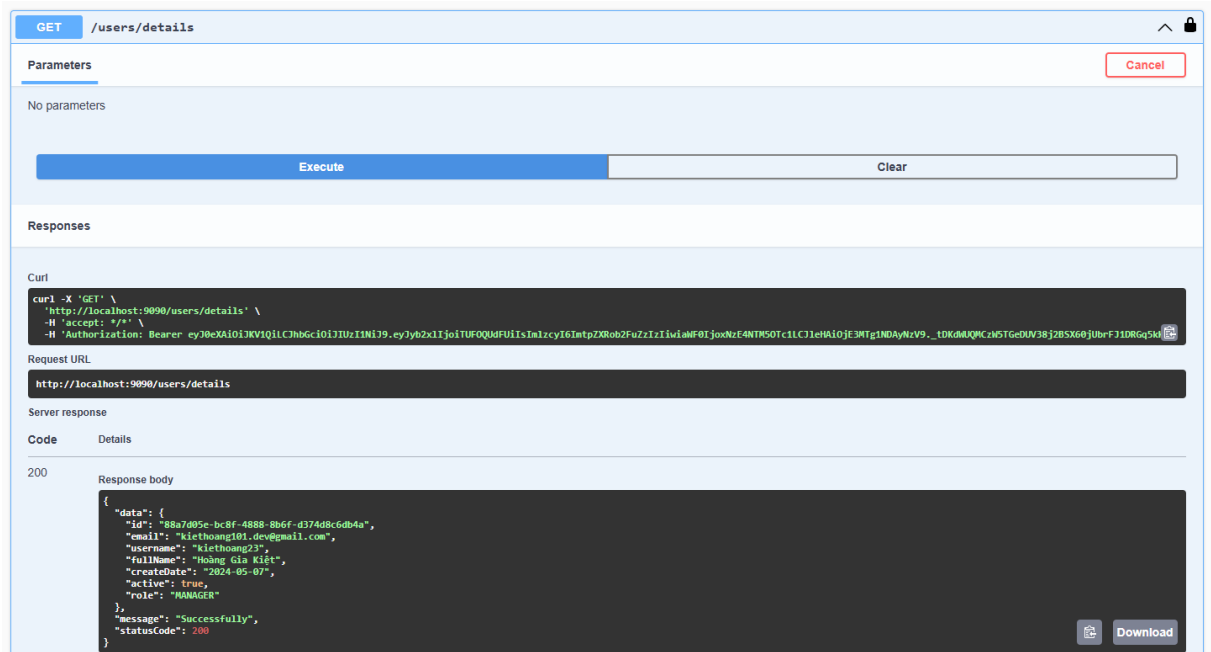
```
public UserDto getUserDetailsFromToken(String token) throws JwtException,
NotFoundException {
    if (JwtUtil.isAccessTokenExpired(token)) {
        throw new JwtException(AuthMessage.TOKEN_EXPIRED);
    }

    String username = JwtUtil.extractUsername(token);
    Optional<User> user = userRepository.findUserByUsername(username);

    if (user.isEmpty())
        throw new NotFoundException(UserMessage.NOT_FOUND);

    return EntityToDto.userToDto(user.get());
}
```

Bảng 4.4: Code xử lý Lấy thông tin người dùng theo token



Hình 4.4: Kiểm tra API Lấy thông tin người dùng theo token

4.2.1.5. Cập nhật thông tin người dùng

UserServiceImpl.java

```

public void updateUser(String id, UserUpdateDto userUpdateDto, String token) throws
JwtException, InvalidException, ForbiddenException {
    if (JwtUtil.isAccessTokenExpired(token)) {
        throw new JwtException(AuthMessage.TOKEN_EXPIRED);
    }

    // Kiểm tra có phải user đó đang cập nhật profile mình không
    String username = JwtUtil.extractUsername(token);
    User userFromDb = getUserById(id);

    if (JwtUtil.extractRole(token).equals(ERole.MANAGER.toString()) ||
username.equals(userFromDb.getUsername())) {
        UserValidator.validatorUserUpdateDTO(userUpdateDto);

        UserDto tempUser = EntityToDto.userToDto(userFromDb);

        // Update cho role staff
        if (userUpdateDto.getEmail() != null) {
            tempUser.setEmail(userUpdateDto.getEmail());
        }
        if (userUpdateDto.getFullName() != null) {
            tempUser.setFullName(userUpdateDto.getFullName());
        }

        // Update chỉ có manager
        if (JwtUtil.extractRole(token).equals(ERole.MANAGER.toString()) &&
tempUser.getRole().equals(ERole.STAFF.toString())) {
            if (userUpdateDto.getActive() != null) {
                tempUser.setActive(userUpdateDto.getActive());
            }
            if (userUpdateDto.getRole() != null) {
                tempUser.setRole(userUpdateDto.getRole());
            }
        }
    }
}

```

```

        checkDuplicated(tempUser);

        // Nếu email được cập nhật -> gửi mail
        if (!userFromDb.getEmail().equals(tempUser.getEmail())) {
            try {
                mailService.sendMailEmailChanged(userFromDb.getEmail(),
tempUser.getEmail());
            } catch (MessagingException e) {
                throw new RuntimeException(AuthMessage.ERROR_SEND_MAIL);
            }
        }

        Optional<Role> roleNew =
            roleRepository.findRoleByName(ERole.valueOf(tempUser.getRole()));

        if (roleNew.isEmpty()) throw new
NotFoundException(AuthMessage.ROLE_NOT_FOUND);

        userFromDb.setEmail(tempUser.getEmail());
        userFromDb.setFullName(tempUser.getFullName());
        userFromDb.setActive(tempUser.getActive());
        userFromDb.setRole(roleNew.get());

        userRepository.save(userFromDb);
    } else
        throw new ForbiddenException();
}

```

Bảng 4.5: Code xử lý cập nhật thông tin người dùng

POST

/users/update/{id}

Parameters

Cancel

Reset

Name	Description
id <small>* required</small>	
string (path)	88a7d05e-bc8f-4888-8b6f-d374d8c6db4a

Request body required

application/json

```

{
  "fullName": "Nguyễn Văn ABC"
}

```

Execute

Clear

Responses

Curl

```

curl -X 'POST' \
  'http://localhost:9090/users/update/88a7d05e-bc8f-4888-8b6f-d374d8c6db4a' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJ0eXAiOiJKV1QiOiJhbnN1bi19.eY3yb2x1Ijo1TUQOMFUiOiJmZy16ImtpZXR0b2FuZzIzIiwiaWF0IjoxNzE4NTQwMDU2Lj1eHA1OjE3MTg1NDQ3NTZ9.ZnFHChzMS1G10oo7Xn4X3OLsI3RAOk4ay8aTEPkThtc' \
  -H 'Content-type: application/json' \
  -d '{
    "fullName": "Nguyễn Văn ABC"
  }'

```

Request URL

http://localhost:9090/users/update/88a7d05e-bc8f-4888-8b6f-d374d8c6db4a

Server response

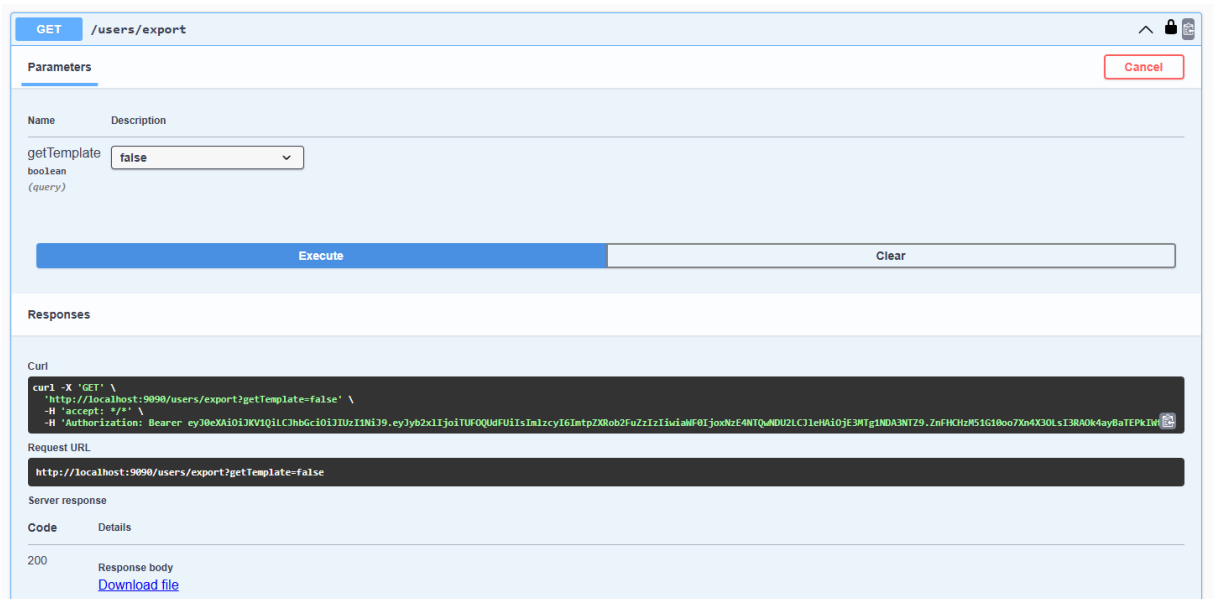
Code	Details
200	<div>Response body</div> <pre> { "data": null, "message": "Successfully", "statusCode": 200 } </pre> <div>Download</div>

Hình 4.5: Kiểm tra API cập nhật thông tin người dùng

4.2.1.6. Xuất danh sách người dùng

UserServiceImpl.java
<pre> public File exportCsv(boolean getTemplate) { try { List<UserDto> userDtos = new ArrayList<>(); userRepository.findAll().forEach(user userDtos.add(EntityToDto.userToDto(user)); return CsvHelper.exportUsers(userDtos, getTemplate); } catch (Exception e) { throw new RuntimeException(e.getMessage()); } } </pre>

Bảng 4.6: Code xử lý xuất danh sách người dùng



Hình 4.6: Kiểm tra API xuất danh sách người dùng

4.2.2. Frontend

4.2.2.1. Khởi tạo dự án ReactJs

index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width,initial-scale=1.0"/>
    <title>MiuKit</title>
  </head>
  <body>
    <div id="root" ></div>
    <script type="module" src="/src/main.tsx"></script>
  </body>
</html>
```

main.tsx

```
import React from "react";
import ReactDOM from "react-dom/client";
```

```
import './index.css';
import App from './App/App';

ReactDOM.createRoot(document.getElementById("root") as HTMLElement).render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

4.2.2.2. Cài đặt phương thức sử dụng Api từ phía server

api / index.ts

```
import axios from "axios";
const privateInstance = axios.create({
  baseURL: "http://localhost:9090",
  headers: {
    "Content-Type": "application/json",
  },
});

// -----

const FetchApi = async (...) => privateInstance.get(...);

const CreateApi = async (...) => privateInstance.post(...);

const UpdateApi = async (...) => privateInstance.post(...);

const DeleteApi = async (...) => privateInstance.delete(...);

const ImportApi = async (...) => privateInstance.post(...);

const ExportApi = async (...) => privateInstance.get(...);
```



```
const SearchApi = async () => privateInstance.get(...);
```

4.2.2.3. Cài đặt đường dẫn ứng dụng với yêu cầu đăng nhập

routes / index.ts

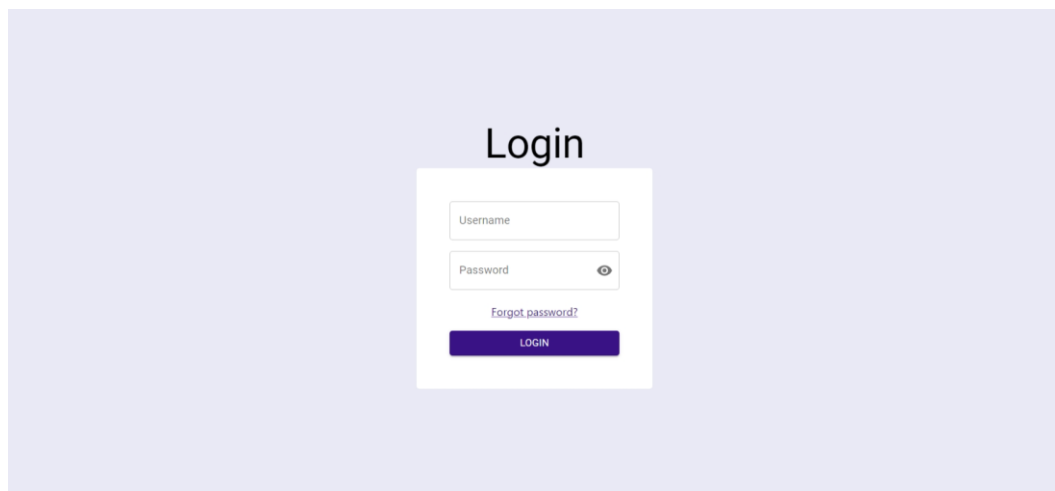
```
const requireSignInRoutes = [
  {
    path: "/",
    element: (<RequireSignIn />),
    children: [
      {
        path: "/",
        index: true,
        element: (<StatisticPage />),
      },
      {
        path: "/apartment",
        element: (<ApartmentIndex />),
      },
      {
        path: "/customers",
        element: (<CustomerIndex />),
      },
      {
        path: "/contracts",
        element: (<ContractIndex />),
      },
      {
        path: "/import",
        element: (<ImportPage />),
      },
    ],
  },
]
```

```
]

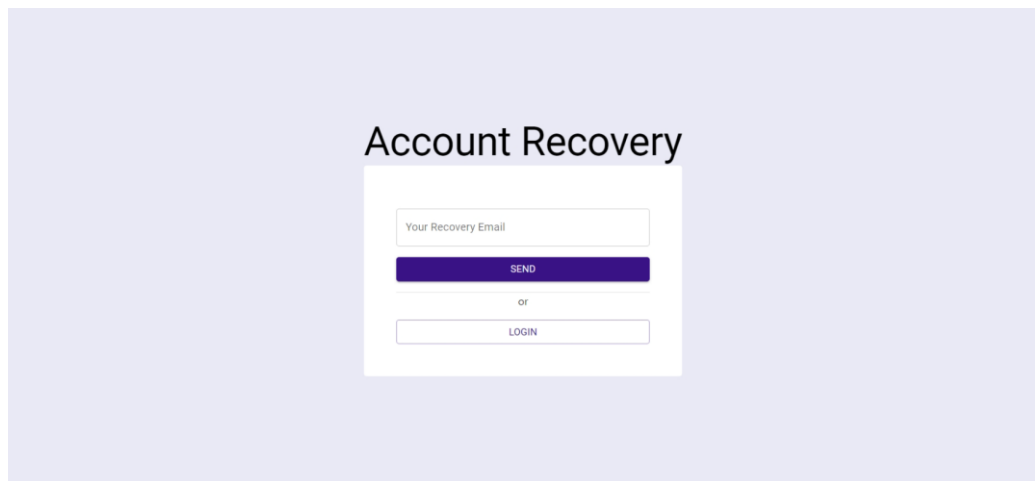
const router = createBrowserRouter(
  [
    ...requireSignInRoutes,
    {
      path: "/login",
      element: <LoginPage />,
    },
    {
      path: "/forgotPassword",
      element: (<ForgotPasswordPage />),
    },
  ],
);

export default router;
```

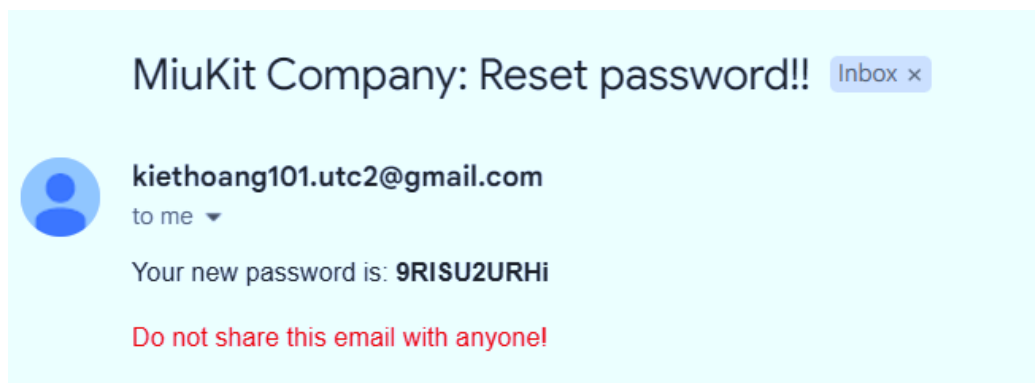
4.3. Giao diện hệ thống



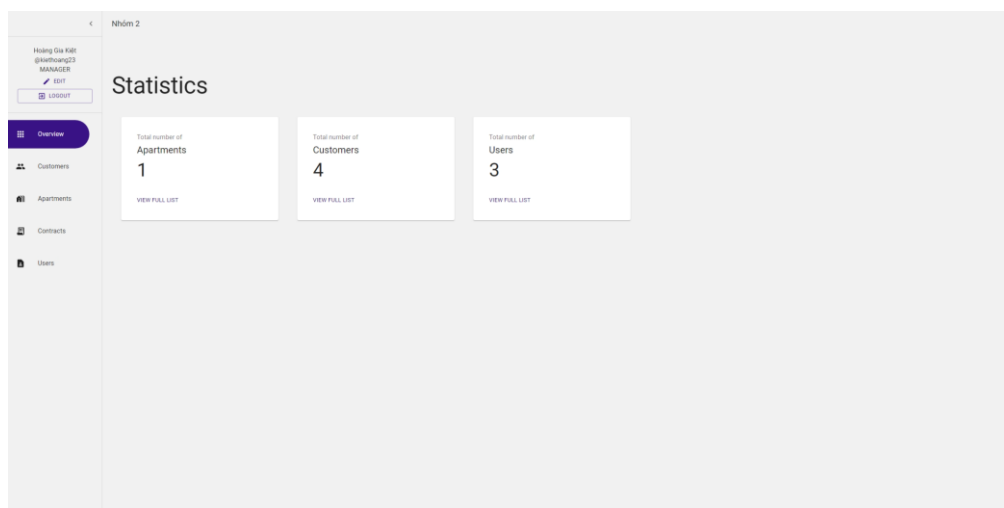
Hình 4.7: Giao diện đăng nhập



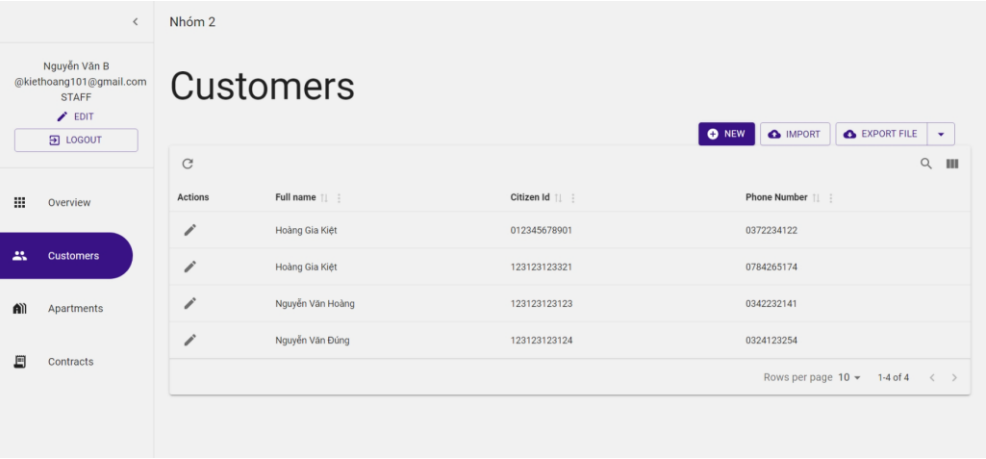
Hình 4.8: Giao diện quên mật khẩu



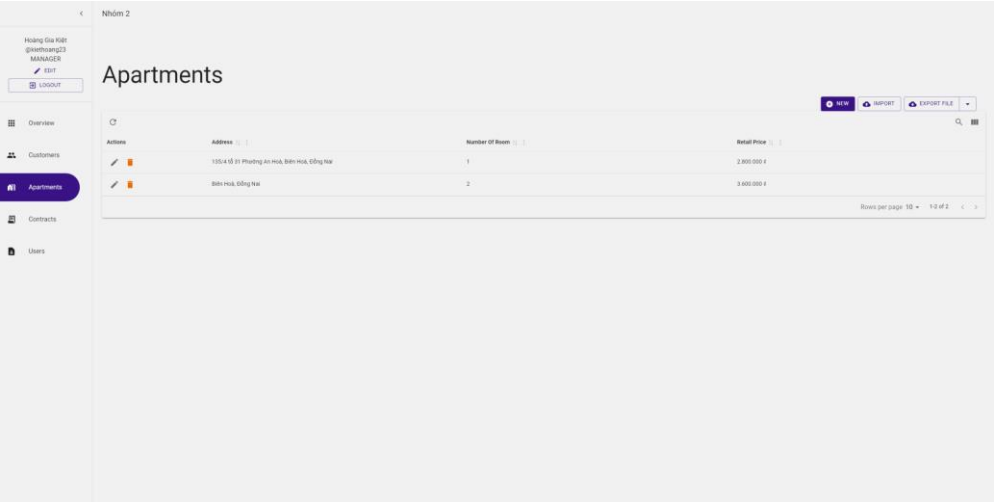
Hình 4.9: Giao diện gửi mật khẩu mới qua email



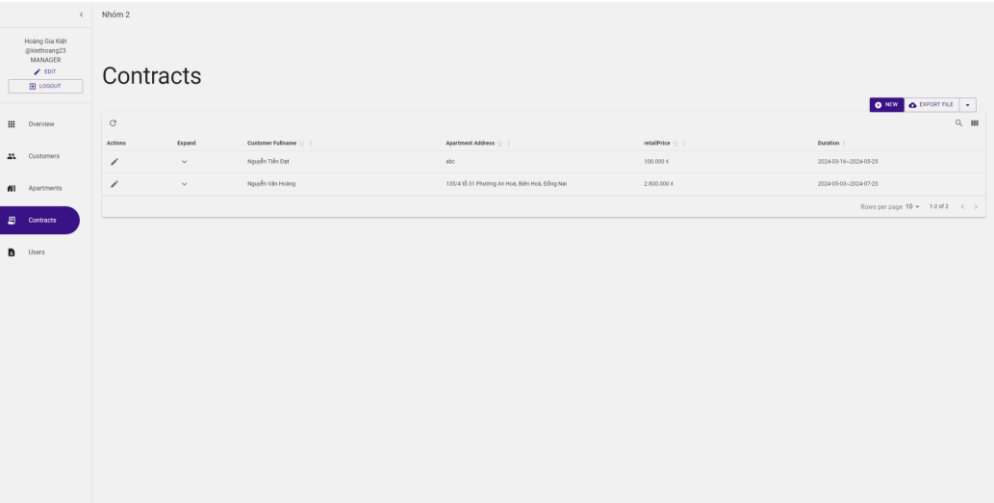
Hình 4.10: Giao diện trang chủ



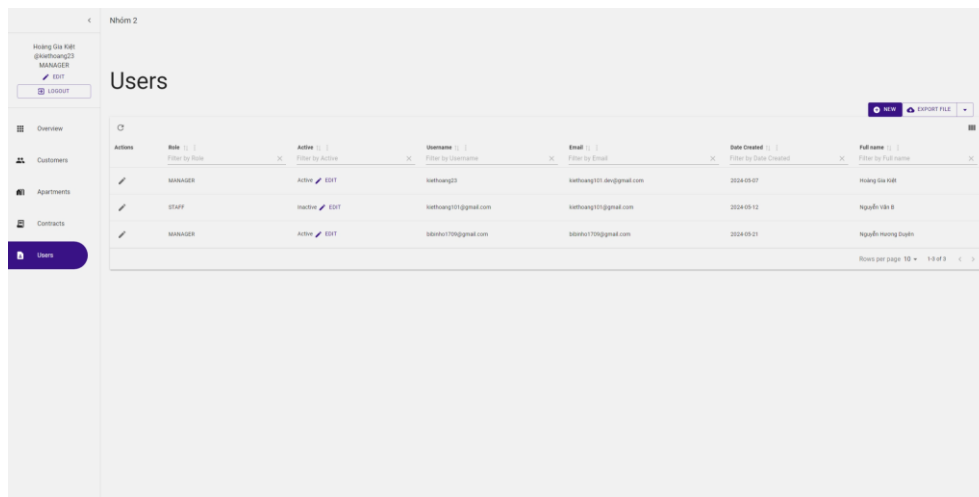
Hình 4.11: Giao diện quản lý khách hàng



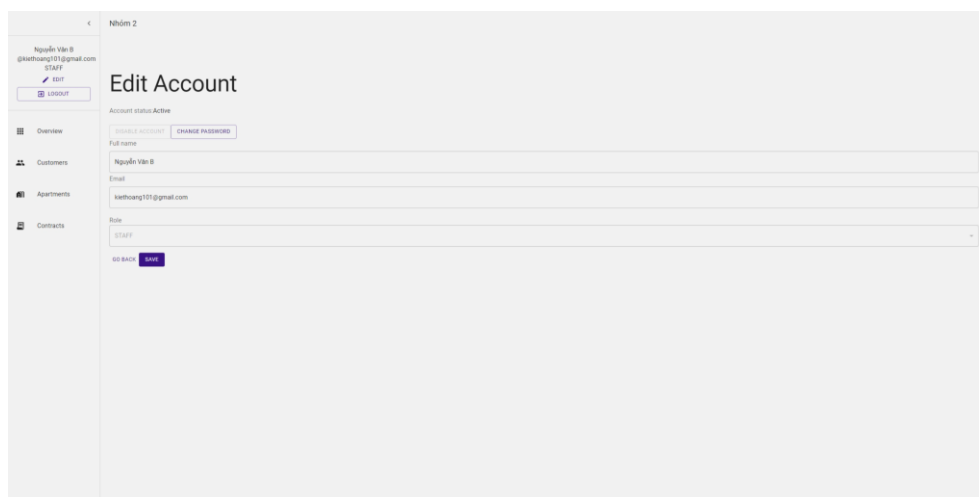
Hình 4.12: Giao diện quản lý căn hộ



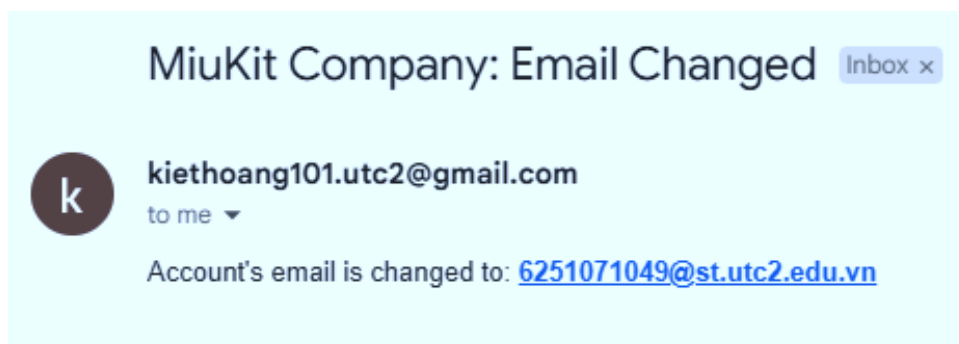
Hình 4.13: Giao diện quản lý hợp đồng



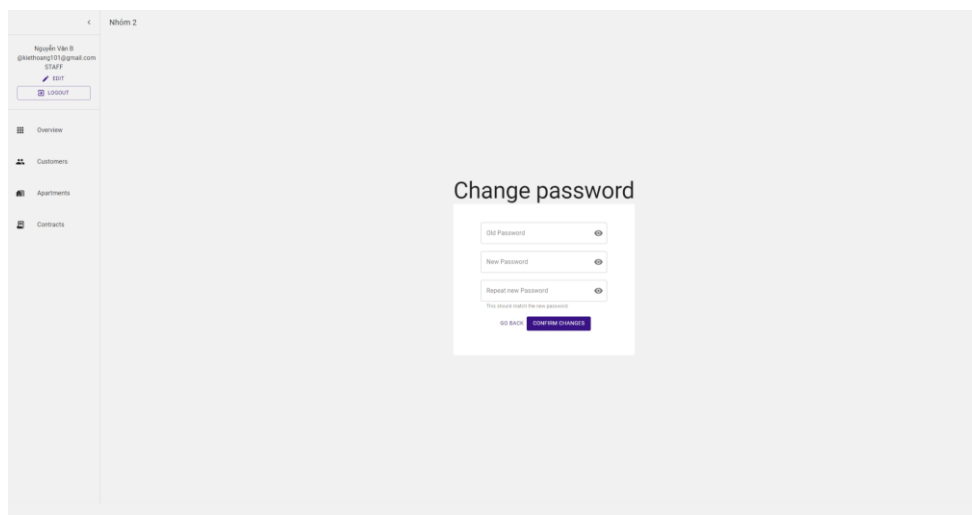
Hình 4.14: Giao diện quản lý người dùng



Hình 4.15: Giao diện tài khoản



Hình 4.16: Giao diện sau khi đổi mail mới



Hình 4.17: Giao diện đổi mật khẩu

KẾT LUẬN

1. Kết quả đạt được

- Quản lý khách hàng.
- Quản lý căn hộ.
- Quản lý hợp đồng.
- Quản lý người dùng.
- Thống kê
- Xác thực và phân quyền người dùng.
- Import/Export danh sách dữ liệu.

2. Hạn chế

- Giao diện còn chưa thân thiện với người dùng.
- Xử lý Backend chưa tối ưu.
- Chưa deploy được website.

3. Hướng phát triển

Trong thời gian tới, em sẽ cố gắng khắc phục những hạn chế trên và mở rộng thêm các tính năng mới của chương trình như:

- Thêm nhiều tính năng
- Tối ưu giao diện
- Tối ưu Backend
- Deploy websites

PHÂN CÔNG CÔNG VIỆC

STT	Họ và tên SV	Công việc
1	Hoàng Gia Kiệt	Phân tích yêu cầu (BA) Xây dựng API (BE) Cài đặt cơ sở dữ liệu
2	Nguyễn Hương Duyên	Phân tích yêu cầu (BA) Xây dựng ứng dụng người dùng (FE)

TÀI LIỆU THAM KHẢO

- [1] Fielding, R. T., Nottingham, M., & Reschke, J., "RFC," [Online]. Available: <https://www.rfc-editor.org/rfc/rfc9110>. [Accessed 14 05 2024].
- [2] N. Thuý, "FPTShop," 28 02 2024. [Online]. Available: <https://fptshop.com.vn/tin-tuc/danh-gia/phan-mem-intellij-idea-176513>. [Accessed 05 05 2024].
- [3] NhanHoa, "NhanHoa," 08 04 2021. [Online]. Available: <https://nhanhoa.com/tin-tuc/mysql-la-gi.html>. [Accessed 07 05 2024].