

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN
ĐỀ TÀI:
XÂY DỰNG WEBSITE QUẢN LÝ CHUNG CƯ
MINI

Người hướng dẫn: **TS. TRƯƠNG NGỌC CHÂU**

Sinh viên thực hiện: **KIỀU PHƯỚC VINH**

Mã số sinh viên: **2151220204**

Lớp: **21CT4**

Đà Nẵng, 12/2025

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN
ĐỀ TÀI:
XÂY DỰNG WEBSITE QUẢN LÝ CHUNG CƯ
MINI

Người hướng dẫn: **TS. TRƯƠNG NGỌC CHÂU**

Sinh viên thực hiện: **KIỀU PHƯỚC VINH**

Mã số sinh viên: **2151220204**

Lớp: **21CT4**

Đà Nẵng, 12/2025

TÓM TẮT

Tên đề tài: Xây dựng website quản lý chung cư mini.

Sinh viên thực hiện: Kiều Phước Vinh

Mã số SV: 2151220204

Lớp: 21CT4

Trong bối cảnh chuyển đổi số phát triển mạnh mẽ, việc quản lý chung cư mini bằng phương pháp truyền thống như ghi chép, thu tiền trực tiếp hay thông báo thủ công đã bộc lộ nhiều hạn chế như khó theo dõi, dễ sai sót và thiếu tính minh bạch. Điều này đặt ra nhu cầu xây dựng một hệ thống quản lý trực tuyến hiện đại, giúp tối ưu hóa quy trình vận hành và nâng cao trải nghiệm cho cả chủ chung cư lẫn người thuê.

Đề tài “Xây dựng website quản lý chung cư mini sử dụng ReactJS và NodeJS” được thực hiện nhằm tạo ra một nền tảng quản lý tập trung, hỗ trợ các chức năng như quản lý căn hộ, hợp đồng, theo dõi thanh toán, cập nhật thông báo và xử lý phản hồi. Việc ứng dụng ReactJS cho giao diện và NodeJS cho backend giúp hệ thống có tốc độ xử lý nhanh, giao diện thân thiện, khả năng bảo mật tốt và dễ mở rộng, qua đó đáp ứng nhu cầu quản lý hiệu quả trong mô hình chung cư mini hiện nay.

LỜI NÓI ĐẦU

Để hoàn thành được Đồ án tốt nghiệp này, em xin chân thành cảm ơn quý thầy cô trong Khoa Công Nghệ Thông Tin – Trường Đại Học Kiến Trúc Đà Nẵng đã tận tình giảng dạy và truyền đạt kiến thức cho em trong suốt thời gian học tập tại trường. Đặc biệt, em xin bày tỏ lòng biết ơn sâu sắc đến Thầy TS. Trương Ngọc Châu, người đã tận tâm hướng dẫn, chỉ bảo và giúp đỡ em trong suốt quá trình thực hiện đồ án tốt nghiệp. Sự hỗ trợ quý báu của thầy không chỉ giúp em hoàn thành tốt đồ án mà còn mang đến cho em nhiều kiến thức, kinh nghiệm giá trị để tự tin hơn trên con đường nghề nghiệp sau này.

Trong quá trình làm đồ án, do kiến thức và kinh nghiệm thực tiễn còn hạn chế nên chắc chắn không tránh khỏi những thiếu sót. Em rất mong nhận được những ý kiến đóng góp của quý thầy cô để em có thể hoàn thiện bản thân hơn nữa.

Em xin trân trọng cảm ơn !

LỜI CAM ĐOAN

Em xin cam đoan rằng toàn bộ kết quả trong đồ án tốt nghiệp là do em tự thực hiện dưới sự hướng dẫn của TS. Trương Ngọc Châu, không sao chép từ bất kỳ nguồn nào mà không trích dẫn. Các nội dung được trình bày trong đồ án hoặc là do em trực tiếp nghiên cứu và xây dựng, hoặc được tổng hợp từ những tài liệu tham khảo có nguồn gốc rõ ràng và được trích dẫn đầy đủ, hợp pháp. Em xin hoàn toàn chịu trách nhiệm và chấp nhận mọi hình thức kỷ luật theo quy định đối với lời cam đoan này.

Đà Nẵng, ngày 7 tháng 12 năm 2025

Sinh viên thực hiện:

Kiều Phước Vinh

MỤC LỤC

LỜI NÓI ĐẦU	ii
LỜI CAM ĐOAN	iii
DANH SÁCH BẢNG.....	viii
DANH SÁCH CÁC KÍ HIỆU CHỮ VIẾT TẮT	ix
MỞ ĐẦU	1
1. Mục đích thực hiện đề tài.....	1
2. Mục tiêu đề tài	1
3. Phạm vi đối tượng nghiên cứu	1
4. Phương pháp nghiên cứu	2
5. Cấu trúc của đồ án tốt nghiệp.....	2
CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI XÂY DỰNG WEBSITE QUẢN LÝ	
CHUNG CƯ MINI	3
1.1. Lý do chọn đề tài	3
1.2. Lịch sử nghiên cứu	3
1.3. Mục tiêu nghiên cứu	4
1.4. Đối tượng phạm vi nghiên cứu.....	5
1.4.1. Đối tượng nghiên cứu.....	5
1.4.2. Phạm vi nghiên cứu.....	6
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	8
2.1. Tổng quan về phát triển website hiện đại.....	8
2.2. Giới thiệu về ngôn ngữ JavaScript	8
2.3. Thư viện ReactJS trong phát triển giao diện người dùng.....	9
2.3.1. Giới thiệu về ReactJS.....	9
2.3.2. Virtual DOM trong ReactJS.....	9
2.3.3. Giới thiệu về ReactJS.....	9
2.3.4 Ứng dụng ReactJS trong hệ thống chung cư mini.....	9
2.4. Nền tảng Node.js và ExpressJS trong Backend.....	10
2.4.1 Tổng quan về Node.js.....	10
2.4.2. ExpressJS Framework.....	10
2.4.3. Mô hình RESTful API	10
2.4.4. Ứng dụng Node.js và ExpressJS trong hệ thống chung cư mini	11
2.5. Cơ sở dữ liệu MongoDB	11
2.5.1. Tổng quan về MongoDB	11

2.5.2. Cấu trúc dữ liệu trong MongoDB.....	11
2.5.3. Ưu điểm của MongoDB	11
2.5.4. Ứng dụng MongoDB trong hệ thống chung cư mini	12
2.6. Kiến trúc tổng thể hệ thống website quản lý chung cư mini	12
2.7. Công nghệ xác thực và bảo mật hệ thống.....	12
2.8. Yêu cầu kỹ thuật.....	13
2.9. Responsive Design và hỗ trợ đa thiết bị.....	13
CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....	14
3.1. Xác định yêu cầu xây dựng website quản lý chung cư mini	14
3.1.1. Yêu cầu chức năng theo từng tác nhân	14
3.1.2. Yêu cầu phi chức năng.....	15
3.1.3. Yêu cầu hệ thống	16
3.2. Phân tích theo yêu cầu chức năng và tác nhân.....	16
3.2.1 Chức năng chi tiết theo bảng.....	16
3.2.2. Yêu cầu phi chức năng.....	18
3.3. Mô hình hóa yêu cầu	18
CHƯƠNG 4: CÀI ĐẶT VÀ PHÁT TRIỂN WEBSITE	48
4.1. Môi trường phát triển và cài đặt.....	48
4.1.1. Phần cứng và phần mềm.....	48
4.1.2. Các thư viện và công nghệ chính.....	48
4.1.2. Cài đặt môi trường và phát triển	49
4.2 Cấu trúc thư mục code	49
4.3. Kết quả đạt được	51
4.4. Kiểm thử hệ thống.....	57
4.4.1. Kế hoạch kiểm thử.....	57
4.4.2. Kết quả kiểm thử	57
TÀI LIỆU THAM KHẢO	61

DANH SÁCH HÌNH ẢNH

Hình 3.1 Mô hình Use case tổng quát.	18
Hình 3.2 Mô hình Use case admin.	21
Hình 3.3 Mô hình Use case chủ Người thuê.....	23
Hình 3.4 Mô hình Use case đăng nhập.....	25
Hình 3.5 Mô hình Use case quản lý căn hộ.....	25
Hình 3.6 Mô hình Use case chức năng quản lý hợp đồng.....	27
Hình 3.7 Mô hình Use case chức năng quản lý hóa đơn.....	28
Hình 3.8 Mô hình Use case chức năng xử lý yêu cầu.....	29
Hình 3.9 Mô hình Use case chức năng báo cáo thống kê.	30
Hình 3.10 Mô hình Use case chức năng quản lý thông báo.	31
Hình 3.11 Mô hình Use case chức năng gửi yêu cầu dịch vụ.....	32
Hình 3.12 Mô hình Use case chức năng thanh toán.....	33
Hình 3.13 Sơ đồ class diagram.	34
Hình 3.14 Mô hình quan hệ.	35
Hình 3.15 Sơ đồ chức năng đăng nhập.....	42
Hình 3.16 Sơ đồ trình tự chức năng quản lý căn hộ.....	42
Hình 3.17 Sơ đồ trình tự chức năng ký hợp đồng.....	43
Hình 3.18 Sơ đồ trình tự chức năng báo cáo và thống kê.	43
Hình 3.19 Sơ đồ trình tự chức năng thanh toán.....	44
Hình 3.20 Sơ đồ hoạt động chức năng báo cáo và thống kê.	45
Hình 3.21 Sơ đồ hoạt động chức năng quản lý căn hộ.....	45
Hình 3.22 Sơ đồ hoạt động chức năng thanh toán.	46
Hình 3.23 Sơ đồ hoạt động chức năng xử lý yêu cầu dịch vụ.....	46
Hình 3.24 Sơ đồ hoạt động chức năng đăng nhập.	47
Hình 4.1 Giao diện trang home.....	51
Hình 4.2 Giao diện xem chi tiết căn hộ.....	52
Hình 4.3 Màn hình đăng nhập.....	52
Hình 4.4 Giao diện trang admin (chủ chung cư).....	53
Hình 4.5 Giao diện chức năng quản lý căn hộ.....	53
Hình 4.6 Giao diện chức năng quản lý hợp đồng.	54
Hình 4.7 Giao diện chức năng quản lý hóa đơn.	54
Hình 4.8 Giao diện chức năng Quản lý yêu cầu dịch vụ.....	55
Hình 4.9 Giao diện chức năng xem chi tiết hóa đơn.....	55
Hình 4.10 Giao diện chức năng thanh toán chuyển khoản.....	56

Hình 4.11 Giao diện chức năng xem chi tiết hóa đơn.....	56
---	----

DANH SÁCH BẢNG

Bảng 3.1 Mô tả chức năng	16
Bảng 3.2 Mô tả Use Case tổng quát.	19
Bảng 3.3 Mô tả Use Case Chủ chung cư	21
Bảng 3.4 Mô tả Use Case Người thuê	24
Bảng 3.5 Mô tả Use Case chức năng đăng nhập.....	25
Bảng 3.6 Mô tả Use Case chức năng Quản lý căn hộ.....	26
Bảng 3.7 Mô tả Use Case chức năng quản lý hợp đồng.....	27
Bảng 3.8 Mô tả Use Case chức năng quản lý hóa đơn.....	28
Bảng 3.9 Mô tả Use Case chức năng xử lý yêu cầu.	29
Bảng 3.10 Mô tả Use Case chức năng báo cáo thống kê.	30
Bảng 3.11 Mô tả Use Case chức năng quản lý thông báo.....	31
Bảng 3.12 Mô tả Use Case chức năng gửi yêu cầu dịch vụ.....	32
Bảng 3.13 Mô tả Use Case chức năng đăng nhập.....	33
Bảng 3.14 Users	35
Bảng 3.15 Apartments.	36
Bảng 3.16 Contracts.	37
Bảng 3.17 Bill	37
Bảng 3.18 ServiceRequest.	39
Bảng 3.19 Occupants.....	40
Bảng 4.1 Test case chức năng đăng nhập	57

DANH SÁCH CÁC KÍ HIỆU CHỮ VIẾT TẮT

Kí Hiệu	Ý nghĩa
API	Application Programming Interface – Giao diện lập trình ứng dụng
UI	User Interface – Giao diện người dùng
UX	User Experience – Trải nghiệm người dùng
DB	Database – Cơ sở dữ liệu
NoSQL	Not Only SQL – Cơ sở dữ liệu phi quan hệ (MongoDB)
HTTP	HyperText Transfer Protocol – Giao thức truyền tải siêu văn bản
HTTPS	HyperText Transfer Protocol Secure – Giao thức truyền tải siêu văn bản bảo mật
JWT	JSON Web Token – Chuẩn giao thức xác thực dạng token
REST	Representational State Transfer – Kiến trúc API nhẹ dùng cho NodeJS
FE	Frontend – Giao diện người dùng (ReactJS)
BE	Backend – Xử lý phía máy chủ (NodeJS/ExpressJS)
ID	Identification – Mã định danh
UID	User ID – Mã định danh người dùng
AID	Apartment ID – Mã định danh căn hộ
CID	Contract ID – Mã định danh hợp đồng
IID	Invoice ID – Mã định danh hóa đơn
RID	Request ID – Mã định danh yêu cầu bảo trì

MỞ ĐẦU

1. Mục đích thực hiện đề tài

Trong bối cảnh đô thị hóa nhanh chóng và số lượng chung cư mini tại Việt Nam ngày càng tăng, nhu cầu quản lý hiệu quả các hoạt động trong tòa nhà như quản lý căn hộ, người thuê, hợp đồng thuê, hóa đơn điện nước, phiếu yêu cầu (ticket), thanh toán và thông báo ngày càng trở nên quan trọng. Tuy nhiên, phần lớn các chủ chung cư mini vẫn quản lý bằng Excel, Zalo hoặc ghi chép thủ công dẫn đến nhiều hạn chế như sai sót dữ liệu, khó theo dõi lịch sử, thiếu minh bạch và tốn thời gian.

Việc xây dựng một “Hệ thống quản lý chung cư mini” là cần thiết nhằm hỗ trợ chủ tòa nhà và người thuê tương tác với nhau một cách nhanh chóng, thuận tiện và minh bạch hơn. Thông qua nền tảng web, các bên có thể quản lý thông tin mọi lúc, mọi nơi; chủ động xem hóa đơn, xem hợp đồng, gửi yêu cầu sửa chữa; đồng thời ban quản lý có thể tạo hóa đơn, quản lý căn hộ và theo dõi tình trạng cư dân dễ dàng.

Đề tài “Xây dựng Website quản lý chung cư mini” không chỉ mang ý nghĩa học thuật giúp sinh viên ứng dụng kiến thức về phân tích – thiết kế hệ thống và lập trình web hiện đại, mà còn mang giá trị thực tiễn cao. Hệ thống giúp số hóa quy trình quản lý, giảm thất thoát, nâng cao minh bạch và cải thiện trải nghiệm của cư dân. Sản phẩm hoàn thiện được kỳ vọng trở thành một giải pháp hữu ích, có thể triển khai thực tế cho các chung cư mini quy mô nhỏ đến trung bình.

2. Mục tiêu đề tài

Mục tiêu của đề tài là xây dựng một Website hỗ trợ quản lý chung cư mini giúp ban quản lý theo dõi căn hộ, người thuê, hợp đồng và hóa đơn một cách thuận tiện, đồng thời hỗ trợ cư dân xem thông tin và gửi yêu cầu nhanh chóng. Hệ thống hướng tới giao diện hiện đại, dễ dùng, hoạt động tốt trên cả máy tính và điện thoại. Về chức năng, website tập trung vào quản lý căn hộ, quản lý người thuê, tạo và theo dõi hóa đơn, xử lý yêu cầu sửa chữa, thông báo và thống kê. Về công nghệ, đề tài áp dụng bộ MERN Stack (ReactJS, NodeJS, ExpressJS, MongoDB) kết hợp xác thực bảo mật JWT nhằm đảm bảo hiệu năng, an toàn dữ liệu và khả năng mở rộng trong tương lai.

3. Phạm vi đối tượng nghiên cứu

a. Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài gồm các công nghệ phát triển web hiện đại như ReactJS dùng để xây dựng giao diện người dùng, NodeJS và ExpressJS xử lý

ng nghiệp vụ phía máy chủ, cùng MongoDB để lưu trữ dữ liệu. Ngoài ra, đề tài còn nghiên cứu các nghiệp vụ thực tế trong quản lý chung cư mini như quản lý căn hộ, người thuê, hợp đồng thuê nhà, hóa đơn điện nước và xử lý yêu cầu hỗ trợ. Người sử dụng chính của hệ thống gồm ban quản lý chung cư và cư dân sinh sống trong tòa nhà

b. Phạm vi nghiên cứu

Phạm vi nghiên cứu của đề tài tập trung vào việc xây dựng Website quản lý chung cư mini với các chức năng chính như quản lý căn hộ, quản lý người thuê, quản lý hợp đồng, tạo và theo dõi hóa đơn, xử lý yêu cầu sửa chữa và gửi thông báo. Hệ thống hoạt động trên nền tảng web, hỗ trợ truy cập bằng trình duyệt trên máy tính và thiết bị di động.

4. Phương pháp nghiên cứu

Đề tài được thực hiện dựa trên hai nhóm phương pháp chính. Thứ nhất là phương pháp nghiên cứu lý thuyết, bao gồm tìm hiểu tài liệu về ReactJS, NodeJS, ExpressJS, MongoDB và các nguyên tắc phân tích – thiết kế hệ thống thông tin. Thứ hai là phương pháp thực nghiệm, trong đó tiến hành khảo sát yêu cầu thực tế, thiết kế cơ sở dữ liệu, xây dựng giao diện, lập trình chức năng và kiểm thử hệ thống. Việc kết hợp hai phương pháp giúp đảm bảo sản phẩm vừa phù hợp thực tiễn vừa đạt yêu cầu kỹ thuật.

5. Cấu trúc của đồ án tốt nghiệp

Đồ án được trình bày theo cấu trúc gồm phần mở đầu, phần kết luận và 4 chương nội dung chính:

Chương 1: Tổng quan về đề tài xây dựng quản lý chung cư mini

Chương 2: Cơ sở lý thuyết

Chương 3: Phân tích và thiết kế hệ thống

Chương 4: Cài đặt và phát triển WEBSITE

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI XÂY DỰNG WEBSITE QUẢN LÝ CHUNG CƯ MINI

1.1. Lý do chọn đề tài

Trong thời đại công nghệ số phát triển mạnh mẽ, việc ứng dụng các hệ thống quản lý trực tuyến đã trở thành xu hướng tất yếu trong nhiều lĩnh vực của đời sống. Đặc biệt, tại các đô thị lớn, mô hình chung cư mini ngày càng phát triển mạnh nhờ đáp ứng nhu cầu nhà ở với chi phí phù hợp cho sinh viên, người lao động và các hộ gia đình nhỏ. Tuy nhiên, dù mô hình này phát triển nhanh chóng nhưng nhiều chủ chung cư vẫn đang quản lý bằng phương pháp thủ công như ghi chép sổ sách, sử dụng bảng tính rời hoặc trao đổi qua tin nhắn, dẫn đến nhiều bất cập như thất lạc thông tin, khó theo dõi doanh thu, chậm trễ trong việc cập nhật hóa đơn điện – nước, và gặp khó khăn khi xử lý yêu cầu từ người thuê.

Bên cạnh đó, người thuê phòng cũng gặp không ít trở ngại khi muốn kiểm tra thông tin phòng trống, giá thuê, hợp đồng, hoặc các thông báo quan trọng từ chủ chung cư. Việc thiếu một hệ thống quản lý tập trung khiến quy trình vận hành trở nên kém hiệu quả, dễ xảy ra nhầm lẫn và làm giảm trải nghiệm của cư dân.

Từ thực tế này, tôi nhận thấy nhu cầu xây dựng một hệ thống quản lý chung cư mini hiện đại, trực tuyến và dễ sử dụng là vô cùng cần thiết. Vì vậy, tôi lựa chọn thực hiện đề tài “Xây dựng website quản lý chung cư mini sử dụng ReactJS, NodeJS và MongoDB” với mục tiêu mang lại giải pháp quản lý khoa học, hiệu quả và minh bạch hơn. ReactJS được sử dụng để xây dựng giao diện thân thiện, mượt mà; NodeJS đảm nhiệm phần xử lý backend và các nghiệp vụ quản lý; MongoDB hỗ trợ lưu trữ dữ liệu linh hoạt, phù hợp với các mô hình dữ liệu thay đổi theo thời gian như cư dân, hợp đồng, hóa đơn và lịch sử thanh toán.

Đề tài không chỉ có giá trị về mặt học thuật, giúp tôi củng cố và vận dụng các kiến thức đã học về lập trình web, kiến trúc hệ thống và cơ sở dữ liệu, mà còn mang ý nghĩa thực tiễn cao. Hệ thống sau khi hoàn thiện có thể trở thành một mô hình mẫu để các chủ chung cư mini hoặc các cơ sở cho thuê phòng áp dụng, góp phần nâng cao hiệu quả quản lý và cải thiện trải nghiệm của người thuê theo hướng hiện đại hóa, chuyên nghiệp hóa.

1.2. Lịch sử nghiên cứu

Việc ứng dụng công nghệ thông tin vào lĩnh vực quản lý nhà ở, căn hộ và chung cư đã xuất hiện từ rất sớm, chủ yếu dưới dạng các phần mềm quản lý nội bộ

trên máy tính. Trong giai đoạn đầu, các hệ thống này thường là ứng dụng desktop với các tính năng cơ bản như lưu trữ thông tin cư dân, ghi chú thu tiền phòng hoặc theo dõi tình trạng phòng. Tuy nhiên, các phần mềm truyền thống còn nhiều hạn chế: khó mở rộng, thiếu khả năng truy cập từ xa, giao diện phức tạp và không hỗ trợ tương tác trực tuyến giữa chủ trọ và người thuê.

Từ những năm 2010 trở lại đây, cùng với sự phát triển mạnh mẽ của Internet và xu hướng chuyển đổi số, nhu cầu quản lý nhà trọ và chung cư nhỏ qua nền tảng web ngày càng tăng. Nhiều chủ nhà, chủ trọ bắt đầu sử dụng các hệ thống quản lý trên nền tảng web hoặc các ứng dụng di động để theo dõi hợp đồng, hóa đơn điện nước, phản hồi cư dân và tình trạng phòng. Tuy nhiên, phần lớn các nền tảng phổ biến hiện nay lại thiên về mô hình quản lý bất động sản quy mô lớn, có chi phí cao, hoặc có quá nhiều tính năng không phù hợp với mô hình chung cư mini—nơi chỉ cần hệ thống đơn giản, dễ dùng, tiết kiệm chi phí và phù hợp với nhu cầu quản lý thực tế.

Về mặt công nghệ, kiến trúc tách rời frontend – backend ngày càng trở thành xu hướng trong các hệ thống web hiện đại. ReactJS – xuất hiện từ năm 2013 – nhanh chóng được ưa chuộng nhờ khả năng xây dựng giao diện linh hoạt, tối ưu hiệu năng và dễ mở rộng. Trong khi đó, các công nghệ backend như NodeJS, ExpressJS hoặc Spring Boot giúp xây dựng API một cách hiệu quả, bảo mật và dễ dàng kết nối với cơ sở dữ liệu. Các hệ quản lý cơ sở dữ liệu như MongoDB hoặc MySQL cũng được sử dụng rộng rãi để lưu trữ thông tin thuê phòng, hóa đơn, hợp đồng và phản hồi cư dân.

Trong bối cảnh đó, việc xây dựng một website quản lý chung cư mini không chỉ đáp ứng xu hướng hiện đại hóa quy trình quản lý mà còn giải quyết thực tiễn nhu cầu của các chủ trọ: quản lý phòng, hóa đơn, hợp đồng, và tương tác với người thuê một cách nhanh chóng, tập trung và thuận tiện. Đề tài này kế thừa những ưu điểm của công nghệ web hiện đại nhằm mang đến một hệ thống phù hợp, dễ sử dụng và có khả năng mở rộng trong tương lai.

1.3. Mục tiêu nghiên cứu

Mục tiêu chính của đề tài là xây dựng một website quản lý chung cư mini sử dụng ReactJS, NodeJS và MongoDB nhằm hỗ trợ chủ chung cư và người thuê trong việc quản lý thông tin căn hộ, hợp đồng thuê, hóa đơn và các hoạt động liên quan. Hệ thống hướng đến việc số hóa quy trình quản lý truyền thống, giúp giảm thiểu thao tác thủ công, tiết kiệm thời gian, hạn chế sai sót và nâng cao hiệu quả quản lý. Đồng thời, website mang lại cho người thuê một nền tảng trực tuyến rõ ràng, minh bạch, thuận tiện trong việc theo dõi các thông tin về hợp đồng, chi phí và các thông báo từ chủ chung cư.

Để hiện thực hóa mục tiêu chung, đề tài tập trung vào việc xây dựng giao diện người dùng trực quan với ReactJS, hỗ trợ người dùng truy cập dễ dàng trên nhiều thiết bị. Trang chủ sẽ hiển thị danh sách các căn hộ của chung cư, bao gồm hình ảnh, diện tích, giá thuê và tình trạng còn trống nhằm giúp người dùng tham khảo trước khi quyết định thuê. Bên cạnh đó, website cung cấp chức năng ký hợp đồng trực tuyến, cho phép người thuê điền form thông tin và nhận hợp đồng dưới dạng PDF sau khi hoàn tất. Hệ thống tự động tạo tài khoản đăng nhập cho người thuê sau khi hợp đồng được ký, giúp quá trình tiếp nhận và quản lý thông tin trở nên mạch lạc và tiện lợi hơn.

Ngoài ra, đề tài đặt mục tiêu xây dựng chức năng quản lý hóa đơn điện, nước và tiền thuê hằng tháng. Chủ chung cư có thể tạo, cập nhật hoặc xóa hóa đơn, trong khi người thuê có thể xem chi tiết hóa đơn, lịch sử thanh toán và các thông báo liên quan. Bên cạnh đó, hệ thống còn hỗ trợ chức năng gửi phản hồi để người thuê trao đổi với chủ chung cư về các vấn đề phát sinh như sửa chữa, thắc mắc hoặc yêu cầu hỗ trợ.

Về mặt kỹ thuật, NodeJS và ExpressJS được sử dụng để xây dựng backend xử lý các nghiệp vụ hệ thống, cung cấp API kết nối với frontend và đảm bảo dữ liệu được xử lý nhanh chóng, ổn định. ReactJS được sử dụng để phát triển giao diện người dùng với khả năng tương tác mượt mà, tối ưu hiệu suất và nâng cao trải nghiệm người dùng. MongoDB được lựa chọn làm cơ sở dữ liệu vì tính linh hoạt, dễ mở rộng và phù hợp với dữ liệu phi quan hệ trong các ứng dụng quản lý như căn hộ, hợp đồng, người thuê và hóa đơn.

Nhìn chung, mục tiêu của đề tài là xây dựng một hệ thống quản lý chung cư mini hoàn chỉnh, trực quan và hiệu quả, đáp ứng đúng nhu cầu thực tế, dễ triển khai và dễ mở rộng trong tương lai.

1.4. Đối tượng phạm vi nghiên cứu

1.4.1. Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài tập trung vào các công nghệ và giải pháp hiện đại phục vụ cho việc xây dựng hệ thống quản lý chung cư mini, nhằm đáp ứng nhu cầu số hóa các nghiệp vụ quản lý và vận hành nhà ở quy mô nhỏ một cách hiệu quả và tối ưu. Các nội dung kỹ thuật chính bao gồm việc ứng dụng ReactJS để phát triển giao diện người dùng thân thiện, trực quan, dễ thao tác, giúp nâng cao trải nghiệm sử dụng trên nhiều loại thiết bị khác nhau; sử dụng NodeJS và ExpressJS để xây dựng hệ thống xử lý nghiệp vụ phía máy chủ, đảm bảo tốc độ phản hồi nhanh, kiến trúc API rõ ràng, linh hoạt trong xử lý logic và kết nối dữ liệu; cùng với MongoDB nhằm lưu trữ dữ liệu theo mô hình NoSQL, hỗ trợ lưu trữ linh hoạt, dễ thay đổi cấu trúc và

đặc biệt phù hợp khi cần mở rộng dữ liệu trong tương lai. Bên cạnh đó, nghiên cứu còn hướng đến việc áp dụng các mô hình quản lý hiệu quả trong công tác vận hành chung cư mini, bao gồm: quản lý và cập nhật trạng thái phòng theo từng thời điểm; theo dõi xuyên suốt quá trình tạo mới, gia hạn hoặc chấm dứt hợp đồng thuê; quản lý hóa đơn điện nước với khả năng tra cứu lịch sử rõ ràng; và xử lý phản ánh từ cư dân để đảm bảo thông tin được tiếp nhận nhanh, phản hồi minh bạch và hạn chế sai sót. Đối tượng sử dụng hệ thống được xác định là chủ các chung cư mini và cư dân đang sinh sống tại chung cư, trong đó chủ chung cư và người quản lý đóng vai trò quản trị vận hành, cập nhật dữ liệu và giám sát hệ thống, còn cư dân là nhóm trực tiếp khai thác thông tin và gửi các yêu cầu hỗ trợ khi cần. Hệ thống hướng đến việc hỗ trợ quản lý vận hành minh bạch, giảm tải công việc thủ công, hạn chế nhập liệu giấy tờ, tiết kiệm thời gian quản lý, đồng thời giúp cư dân dễ dàng theo dõi thông tin, gửi yêu cầu hỗ trợ và thanh toán dịch vụ một cách nhanh chóng, chủ động hơn, qua đó nâng cao hiệu suất vận hành chung cư, tăng tính hài lòng của người thuê và tạo ra môi trường quản lý hiện đại, có tính ứng dụng thực tiễn cao. Ngoài ra, đề tài còn xem xét các giải pháp để đảm bảo tính ổn định, an toàn dữ liệu và thuận lợi cho việc phát triển mở rộng sau này, góp phần hoàn thiện mô hình quản lý chung cư mini theo hướng đơn giản nhưng hiệu quả và bền vững.

1.4.2. Phạm vi nghiên cứu

Phạm vi nội dung của đề tài tập trung vào quá trình phân tích, thiết kế và xây dựng hệ thống website quản lý chung cư mini nhằm số hóa hầu hết các nghiệp vụ quản lý nội bộ và tương tác với cư dân. Hệ thống được triển khai với các nhóm chức năng chính, bao gồm: quản lý thông tin phòng (số phòng, trạng thái thuê, giá thuê, tiện ích đi kèm); quản lý hợp đồng thuê (tạo mới, chỉnh sửa, gia hạn, chấm dứt, lưu trữ lịch sử hợp đồng); quản lý danh sách cư dân và khách thuê (thông tin cá nhân, tình trạng lưu trú, phân quyền truy cập); ghi nhận và quản lý chỉ số điện, nước định kỳ; tự động tạo và theo dõi hóa đơn hàng tháng; tra cứu lịch sử thanh toán và trạng thái hóa đơn; quản lý các yêu cầu sửa chữa và bảo trì (tiếp nhận, phân loại, xử lý, phản hồi kết quả); cũng như hiển thị các thông báo quan trọng từ ban quản lý đến cư dân một cách nhanh chóng, trực quan.

Bên cạnh các chức năng nghiệp vụ cốt lõi, đề tài còn nghiên cứu chuyên sâu vào việc tối ưu trải nghiệm người dùng (UX/UI) để đảm bảo giao diện thân thiện, dễ thao tác trên cả thiết bị máy tính và di động. Đồng thời, hệ thống chú trọng đến tính bảo mật và toàn vẹn dữ liệu, bao gồm các cơ chế như: mã hóa và bảo vệ thông tin nhạy cảm, xác thực đăng nhập an toàn, phân quyền rõ ràng giữa các tác nhân (chủ chung cư, admin, cư dân), ngăn chặn truy cập trái phép, backup dữ liệu và đảm bảo

hệ thống vận hành ổn định. Ngoài ra, hệ thống cũng tuân thủ các nguyên tắc bảo vệ dữ liệu cá nhân và xây dựng quy trình xử lý thông tin minh bạch, an toàn, phù hợp cho mô hình quản lý dân cư.

Phạm vi không gian nghiên cứu được giới hạn trong mô hình quản lý của một chung cư mini cụ thể tại thành phố Đà Nẵng, với mục đích sử dụng làm môi trường thử nghiệm, thẩm định tính khả thi và hiệu quả của hệ thống. Đây là mô hình thực tế nhằm giúp kết quả nghiên cứu có tính ứng dụng cao và phản ánh chính xác các nhu cầu phát sinh trong quá trình quản lý, vận hành chung cư mini hiện nay. Mặc dù phạm vi triển khai trước mắt chỉ áp dụng cho một tòa nhà, hệ thống được thiết kế theo kiến trúc hướng mở và có khả năng mở rộng cao (scalable), cho phép dễ dàng tích hợp thêm tòa nhà mới, mở rộng số lượng phòng, tăng quy mô cư dân, hoặc triển khai trên nhiều chung cư mini khác trong tương lai mà không cần thay đổi nhiều về cấu trúc lõi của hệ thống. Nhờ đó, sản phẩm không chỉ mang ý nghĩa là một mô hình demo mà còn là nền tảng cho giải pháp quản lý chung cư mini đa điểm, sẵn sàng thích nghi khi nhu cầu sử dụng gia tăng.

Ngoài ra, đề tài cũng xem xét khả năng phát triển hệ thống theo hướng chuẩn hóa và tái sử dụng quy trình, nhằm giúp các chủ chung cư mini khác có thể nhanh chóng áp dụng vào mô hình quản lý của họ với thời gian triển khai ngắn, chi phí tối ưu và ít rủi ro về tái cấu trúc công nghệ. Điều này góp phần khẳng định tính bền vững của hệ thống, không chỉ phục vụ nghiên cứu mà còn hướng đến giá trị ứng dụng lâu dài trong thực tế.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Tổng quan về phát triển website hiện đại

Trong những năm gần đây, cùng với sự phát triển mạnh mẽ của Internet và quá trình chuyển đổi số, website không chỉ dừng lại ở vai trò cung cấp thông tin mà còn trở thành nền tảng vận hành chính cho nhiều tổ chức, doanh nghiệp và mô hình dịch vụ. Website hiện đại ngày nay được xây dựng theo hướng tương tác cao, xử lý dữ liệu thời gian thực và có khả năng mở rộng linh hoạt.

Mô hình phát triển website hiện đại thường được xây dựng theo kiến trúc tách biệt giữa frontend và backend. Frontend đảm nhiệm phần giao diện người dùng và trải nghiệm tương tác, trong khi backend chịu trách nhiệm xử lý nghiệp vụ, bảo mật và quản lý dữ liệu. Kiến trúc này giúp hệ thống dễ bảo trì, dễ nâng cấp và phù hợp với các hệ thống quản lý như website quản lý chung cư mini.

2.2. Giới thiệu về ngôn ngữ JavaScript

JavaScript (JS) là ngôn ngữ lập trình phổ biến giúp xây dựng các website tương tác, sinh động và linh hoạt. Đây là ngôn ngữ kịch bản (scripting language), dựa trên các đối tượng có sẵn hoặc tự định nghĩa, cho phép các lập trình viên xử lý sự kiện, thay đổi nội dung trang web, hoặc tương tác trực tiếp với người dùng.

JS có thể chạy trên tất cả các trình duyệt phổ biến như Chrome, Firefox, Edge, và được hỗ trợ đầy đủ trên các thiết bị di động, giúp phát triển ứng dụng đa nền tảng. Ngôn ngữ JavaScript được phát triển bởi Netscape và hiện nay được sử dụng trên hơn 92% các website. JS có thể nhúng trực tiếp vào HTML hoặc lưu trong file riêng để sử dụng nhiều lần, giúp tối ưu khả năng bảo trì và phát triển dự án.

JavaScript có nhiều ưu điểm nổi bật. Trước hết, nó dễ học và sử dụng nhờ cú pháp đơn giản, gần gũi với tiếng Anh, giúp người mới tiếp cận nhanh chóng. Bên cạnh đó, JavaScript có khả năng chạy trên nhiều trình duyệt và thiết bị khác nhau, bao gồm cả máy tính và thiết bị di động, đảm bảo tính tương thích cao. Ngoài ra, JavaScript còn hỗ trợ tương tác thời gian thực, có thể xử lý các sự kiện như click, hover và cập nhật nội dung trên trang mà không cần tải lại, mang đến trải nghiệm người dùng mượt mà và linh hoạt.

Trong dự án quản lý chung cư, việc sử dụng JavaScript mang lại nhiều lợi ích thiết thực. Nó giúp tương tác trực tiếp với người dùng bằng cách hiển thị thông báo khi thêm hóa đơn, đăng ký hợp đồng hoặc gửi nhắc nhở thanh toán. Đồng thời, JavaScript cho phép cập nhật dữ liệu thời gian thực, ví dụ như thông tin phòng trống,

hợp đồng hay đơn hàng, mà không cần tải lại trang, giúp trải nghiệm người dùng mượt mà hơn. Bên cạnh đó, JavaScript dễ dàng tích hợp với HTML, CSS và ReactJS, tạo ra giao diện trực quan, dễ sử dụng cho cả chủ chung cư lẫn khách thuê.

2.3. Thư viện ReactJS trong phát triển giao diện người dùng

2.3.1. Giới thiệu về ReactJS

ReactJS là thư viện JavaScript mã nguồn mở, được phát triển bởi Facebook, dùng để xây dựng giao diện người dùng (UI) theo kiến trúc component-based (dựa trên các thành phần). ReactJS giúp các ứng dụng web trở nên tương tác, dễ bảo trì và hiệu suất cao.

2.3.2. Virtual DOM trong ReactJS

Virtual DOM là một bản sao ảo của DOM thực, được ReactJS sử dụng để theo dõi và quản lý các thay đổi trong giao diện người dùng. Khi dữ liệu trong ứng dụng thay đổi, ReactJS sẽ cập nhật vào Virtual DOM trước, sau đó tiến hành so sánh (diffing) với DOM trước đó để xác định chính xác những phần tử nào thực sự cần thay đổi. Nhờ cơ chế này, ReactJS chỉ cập nhật những thành phần bị thay đổi thay vì render lại toàn bộ giao diện như cách truyền thống. Điều này giúp giảm đáng kể số lần thao tác trực tiếp với DOM thật, từ đó tối ưu hiệu suất xử lý, giảm tải cho trình duyệt và nâng cao tốc độ phản hồi của website, đặc biệt hiệu quả đối với các hệ thống có nhiều dữ liệu động như website quản lý chung cư mini.

2.3.3. Giới thiệu về ReactJS

Trong ReactJS, dữ liệu được quản lý chủ yếu thông qua hai khái niệm quan trọng là state và props. State là dữ liệu nội bộ của mỗi component, dùng để lưu trữ trạng thái và được cập nhật trong quá trình người dùng tương tác với hệ thống, ví dụ như trạng thái đăng nhập, danh sách căn hộ, hóa đơn hoặc yêu cầu dịch vụ. Props là các tham số được truyền từ component cha xuống component con để chia sẻ dữ liệu giữa các thành phần giao diện, giúp đảm bảo tính nhất quán trong quá trình hiển thị. Bên cạnh cơ chế state và props cơ bản, ReactJS còn cho phép tích hợp các thư viện quản lý trạng thái như Redux hoặc Context API nhằm quản lý dữ liệu ở quy mô lớn, đặc biệt hữu ích đối với các hệ thống có nhiều chức năng và nhiều người dùng như website quản lý chung cư mini. Việc quản lý trạng thái tập trung giúp dữ liệu được đồng bộ trên toàn hệ thống, hạn chế lỗi phát sinh, dễ kiểm soát luồng dữ liệu và nâng cao hiệu quả bảo trì cũng như mở rộng ứng dụng trong tương lai.

2.3.4 Ứng dụng ReactJS trong hệ thống chung cư mini

ReactJS mang lại nhiều ưu điểm quan trọng cho dự án quản lý chung cư. Trước hết, nhờ khả năng tái sử dụng thành phần (Reusable Components), các phần như danh sách căn hộ, hóa đơn hay hợp đồng có thể dùng nhiều lần, tiết kiệm thời gian phát

triển. ReactJS còn có hiệu suất cao nhờ sử dụng Virtual DOM, chỉ cập nhật những phần thay đổi, giúp trải nghiệm người dùng mượt mà. Cơ chế định hướng dữ liệu một chiều (Unidirectional Data Flow) giúp quản lý dữ liệu dễ kiểm soát và giảm rủi ro lỗi. Ngoài ra, cấu trúc component rõ ràng giúp mã nguồn dễ bảo trì và thuận tiện cho việc mở rộng dự án. Trong thực tế, ReactJS có thể được ứng dụng để hiển thị danh sách phòng trống theo tầng hoặc loại căn hộ, tạo form nhập liệu hợp đồng, thêm/sửa/xóa thông tin khách thuê, cũng như cập nhật trạng thái thanh toán của khách theo thời gian thực.

2.4. Nền tảng Node.js và ExpressJS trong Backend

2.4.1 Tổng quan về Node.js

Node.js là một môi trường cho phép chạy JavaScript ở phía máy chủ (server), được xây dựng dựa trên V8 Engine của Google Chrome. Khác với mô hình xử lý đồng bộ truyền thống, Node.js hoạt động theo cơ chế bất đồng bộ (asynchronous) và non-blocking I/O, cho phép xử lý nhiều yêu cầu cùng lúc mà không làm nghẽn hệ thống. Điều này giúp Node.js có hiệu suất cao, đặc biệt phù hợp với các hệ thống web có số lượng người dùng truy cập lớn và yêu cầu xử lý dữ liệu liên tục. Trong hệ thống quản lý chung cư mini, Node.js đóng vai trò là nền tảng chính để xây dựng backend, đảm bảo hệ thống hoạt động ổn định, nhanh chóng và hiệu quả.

2.4.2. ExpressJS Framework

ExpressJS là một framework phổ biến được xây dựng trên nền tảng Node.js, giúp đơn giản hóa quá trình xây dựng máy chủ web và phát triển các ứng dụng backend. ExpressJS cung cấp nhiều công cụ mạnh mẽ như định tuyến (routing), xử lý request – response, middleware, xử lý lỗi và tích hợp các thư viện bên ngoài. Nhờ ExpressJS, việc xây dựng các chức năng backend trở nên nhanh chóng, rõ ràng và dễ quản lý hơn so với việc lập trình thuần Node.js. Trong dự án hệ thống quản lý chung cư mini, ExpressJS được sử dụng để xây dựng toàn bộ hệ thống API, phục vụ cho việc giao tiếp giữa frontend và backend.

2.4.3. Mô hình RESTful API

RESTful API là mô hình giao tiếp giữa client và server thông qua các phương thức HTTP như GET, POST, PUT và DELETE. Mỗi API đại diện cho một tài nguyên cụ thể và cho phép frontend thực hiện các thao tác thêm, sửa, xóa, truy vấn dữ liệu một cách linh hoạt. Việc sử dụng RESTful API giúp hệ thống hoạt động độc lập giữa frontend và backend, dễ dàng mở rộng, tích hợp với các ứng dụng khác trong tương lai. Trong hệ thống quản lý chung cư mini, toàn bộ chức năng như quản lý căn hộ, hợp đồng, hóa đơn, người dùng và dịch vụ đều được xây dựng thông qua mô hình RESTful API.

2.4.4. Ứng dụng Node.js và ExpressJS trong hệ thống chung cư mini

Trong hệ thống website quản lý chung cư mini, Node.js và ExpressJS được sử dụng để xây dựng toàn bộ tầng xử lý nghiệp vụ của hệ thống. Backend đảm nhiệm các chức năng quan trọng như xác thực và phân quyền người dùng bằng JWT, xử lý tạo – sửa – hủy hợp đồng thuê, quản lý hóa đơn dịch vụ, ghi nhận lịch sử thanh toán, tiếp nhận và xử lý yêu cầu dịch vụ từ người thuê. Ngoài ra, hệ thống backend còn kết nối với cơ sở dữ liệu MongoDB thông qua thư viện Mongoose để quản lý dữ liệu một cách chặt chẽ và an toàn. Backend của hệ thống được triển khai hoạt động trên cổng 5000, đảm bảo khả năng tiếp nhận và xử lý yêu cầu từ frontend một cách ổn định và liên tục.

2.5. Cơ sở dữ liệu MongoDB

2.5.1. Tổng quan về MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL phổ biến hiện nay, được thiết kế để lưu trữ và xử lý dữ liệu theo dạng document với cấu trúc BSON (Binary JSON). Khác với các hệ quản trị cơ sở dữ liệu quan hệ truyền thống như MySQL hay SQL Server, MongoDB không yêu cầu một cấu trúc bảng cố định, cho phép dữ liệu được lưu trữ linh hoạt theo từng đối tượng cụ thể. Nhờ đặc điểm này, MongoDB rất phù hợp với các hệ thống có yêu cầu thay đổi dữ liệu thường xuyên trong quá trình phát triển. Bên cạnh đó, MongoDB còn có hiệu suất truy vấn cao, khả năng mở rộng tốt và hỗ trợ làm việc với dữ liệu lớn, đáp ứng tốt nhu cầu của các hệ thống web hiện đại như website quản lý chung cư mini.

2.5.2. Cấu trúc dữ liệu trong MongoDB

Trong MongoDB, dữ liệu được tổ chức theo mô hình collection và document. Collection tương đương với bảng trong cơ sở dữ liệu quan hệ, còn document tương đương với một bản ghi (record). Mỗi document được lưu trữ dưới dạng cặp key–value giống với cấu trúc JSON, giúp việc đọc, ghi và xử lý dữ liệu trở nên trực quan và dễ hiểu. Các document trong cùng một collection không bắt buộc phải có cấu trúc hoàn toàn giống nhau, điều này giúp hệ thống dễ dàng mở rộng và thay đổi dữ liệu khi cần thiết. Cấu trúc linh hoạt của MongoDB giúp lập trình viên dễ dàng bổ sung thêm các thuộc tính mới mà không ảnh hưởng đến các dữ liệu đã có trước đó.

2.5.3. Ưu điểm của MongoDB

MongoDB sở hữu nhiều ưu điểm nổi bật như tốc độ truy vấn nhanh, khả năng xử lý dữ liệu lớn ổn định và dễ dàng mở rộng theo chiều ngang. Do không bị ràng buộc chặt chẽ về cấu trúc dữ liệu, MongoDB cho phép thay đổi, bổ sung hoặc chỉnh sửa dữ liệu một cách linh hoạt trong quá trình phát triển hệ thống. Bên cạnh đó, MongoDB còn hỗ trợ tốt cho các ứng dụng thời gian thực, khả năng sao lưu và phục

hồi dữ liệu hiệu quả, đảm bảo an toàn cho hệ thống. Chính những ưu điểm này đã giúp MongoDB trở thành lựa chọn phù hợp cho các hệ thống quản lý hiện đại, trong đó có hệ thống quản lý chung cư mini.

2.5.4. Ứng dụng MongoDB trong hệ thống chung cư mini

Trong hệ thống quản lý chung cư mini, MongoDB được sử dụng để lưu trữ toàn bộ dữ liệu quan trọng của hệ thống. Cụ thể, MongoDB lưu trữ thông tin người dùng, thông tin căn hộ, hợp đồng thuê, hóa đơn dịch vụ, lịch sử thanh toán, các yêu cầu dịch vụ của người thuê và hệ thống thông báo. Việc sử dụng MongoDB giúp hệ thống dễ dàng quản lý, truy vấn và cập nhật dữ liệu một cách nhanh chóng và chính xác. Đồng thời, cơ sở dữ liệu MongoDB còn đảm bảo khả năng mở rộng trong tương lai khi số lượng người dùng và dữ liệu của hệ thống ngày càng tăng.

2.6. Kiến trúc tổng thể hệ thống website quản lý chung cư mini

Hệ thống website quản lý chung cư mini được xây dựng theo mô hình kiến trúc 3 tầng gồm: tầng giao diện (Frontend), tầng xử lý nghiệp vụ (Backend) và tầng dữ liệu (Database). Trong đó, tầng giao diện sử dụng ReactJS để xây dựng các thành phần hiển thị và tương tác trực tiếp với người dùng như quản lý căn hộ, hợp đồng, hóa đơn, dịch vụ và tài khoản. Tầng xử lý nghiệp vụ sử dụng NodeJS kết hợp với ExpressJS để xây dựng hệ thống máy chủ, xử lý các yêu cầu từ frontend, kiểm soát luồng dữ liệu và thực hiện các chức năng nghiệp vụ của hệ thống. Tầng dữ liệu sử dụng MongoDB làm nơi lưu trữ toàn bộ thông tin của hệ thống như người dùng, căn hộ, hợp đồng, hóa đơn và lịch sử thanh toán. Mô hình kiến trúc 3 tầng giúp hệ thống hoạt động ổn định, dễ bảo trì, dễ nâng cấp và có khả năng mở rộng tốt khi số lượng người dùng và dữ liệu ngày càng tăng trong tương lai.

2.7. Công nghệ xác thực và bảo mật hệ thống

Bảo mật là một trong những yếu tố quan trọng trong hệ thống website quản lý chung cư mini, do hệ thống xử lý nhiều dữ liệu nhạy cảm liên quan đến người dùng, hợp đồng và tài chính. Hệ thống sử dụng công nghệ JWT (JSON Web Token) để thực hiện xác thực và phân quyền người dùng. Khi người dùng đăng nhập thành công, hệ thống sẽ tạo ra một token và gửi về phía client, token này sẽ được sử dụng để xác thực trong các lần truy cập tiếp theo vào hệ thống. Mật khẩu của người dùng được mã hóa trước khi lưu vào cơ sở dữ liệu nhằm đảm bảo an toàn thông tin. Ngoài ra, hệ thống còn phân quyền rõ ràng giữa các vai trò như Admin và Người thuê, giúp kiểm soát chặt chẽ quyền truy cập và đảm bảo dữ liệu chỉ được sử dụng đúng mục đích.

2.8. Yêu cầu kỹ thuật

Giao diện người dùng của website quản lý chung cư mini được thiết kế theo phong cách hiện đại, hướng đến trải nghiệm thân thiện, dễ sử dụng và trực quan cho người dùng. Hệ thống sử dụng CSS3 kết hợp với Flexbox, Grid Layout và CSS Variables để xây dựng bố cục linh hoạt, dễ quản lý và tối ưu hiển thị trên nhiều kích thước màn hình khác nhau. Ngoài ra, giao diện còn được áp dụng các hiệu ứng hiện đại như Glassmorphism và Gradient Background nhằm tăng tính thẩm mỹ và tạo cảm giác chuyên nghiệp cho website. Việc thiết kế giao diện hợp lý giúp người dùng dễ dàng thao tác, giảm thời gian làm quen và nâng cao hiệu quả sử dụng hệ thống.

2.9. Responsive Design và hỗ trợ đa thiết bị

Hệ thống website quản lý chung cư mini được xây dựng theo hướng Responsive Design nhằm đảm bảo hiển thị tốt trên nhiều loại thiết bị khác nhau như máy tính để bàn, máy tính bảng và điện thoại di động. Giao diện được tự động điều chỉnh kích thước, bố cục và các thành phần hiển thị sao cho phù hợp với từng loại màn hình. Nhờ đó, người dùng có thể dễ dàng truy cập và sử dụng hệ thống ở mọi lúc, mọi nơi mà không gặp khó khăn về giao diện. Việc hỗ trợ đa thiết bị không chỉ giúp nâng cao trải nghiệm người dùng mà còn tăng tính linh hoạt và hiệu quả trong quá trình quản lý và vận hành chung cư mini.

CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

3.1. Xác định yêu cầu xây dựng website quản lý chung cư mini

Trong đề tài “**Xây dựng website quản lý chung cư mini**”, việc xác định các yêu cầu và phân loại chức năng theo từng nhóm người dùng là bước quan trọng nhằm đảm bảo hệ thống đáp ứng đúng nhu cầu thực tế. Website được xây dựng với mục tiêu hỗ trợ chủ chung cư trong việc quản lý phòng, người thuê, hợp đồng và các hóa đơn hằng tháng, đồng thời mang đến cho người thuê một nền tảng thuận tiện để theo dõi thông tin và tương tác với ban quản lý.

Hệ thống tập trung vào 2 đối tượng sử dụng chính:

1. Chủ chung cư (Admin)

Đây là người trực tiếp quản lý toàn bộ hoạt động của khu chung cư mini. Admin có quyền truy cập toàn bộ hệ thống để thực hiện các chức năng như: quản lý phòng, quản lý người thuê, tạo và theo dõi hợp đồng thuê, nhập chỉ số điện – nước, xuất hóa đơn, kiểm tra tình trạng thanh toán, đăng thông báo và xử lý các yêu cầu phản ánh từ người thuê.

2. Người thuê

Đây là những cư dân đang sinh sống trong chung cư và có nhu cầu theo dõi thông tin liên quan đến căn phòng của mình. Người thuê có thể đăng nhập vào hệ thống để xem thông tin cá nhân, hợp đồng thuê, hóa đơn điện – nước hằng tháng, lịch sử thanh toán, nhận thông báo từ chủ nhà và gửi yêu cầu hỗ trợ khi cần.

Việc xác định rõ hai đối tượng người dùng giúp đảm bảo tính logic trong thiết kế hệ thống, đồng thời hỗ trợ phân chia chức năng hợp lý, dễ sử dụng và có khả năng mở rộng trong tương lai.

3.1.1. Yêu cầu chức năng theo từng tác nhân

Người thuê: Người thuê là đối tượng sử dụng hệ thống để theo dõi thông tin cá nhân và các dịch vụ liên quan đến phòng đang thuê. Các chức năng của người thuê bao gồm:

- Xem thông tin phòng đang thuê: bao gồm diện tích, giá thuê, trạng thái hợp đồng và các thông tin liên quan.
- Xem hợp đồng thuê: hiển thị các điều khoản hợp đồng, thời hạn thuê, ngày bắt đầu – ngày kết thúc.
- Xem hóa đơn điện, nước, internet và phí dịch vụ hằng tháng: người thuê có thể xem chi tiết từng khoản phí, chỉ số điện – nước và tổng số tiền cần thanh toán.

- Theo dõi lịch sử thanh toán: xem các hóa đơn đã trả, thời gian thanh toán và trạng thái xử lý.
- Nhận thông báo từ chủ chung cư: ví dụ thông báo bảo trì, tăng giá dịch vụ, nhắc đóng tiền,...
- Gửi yêu cầu – phản ánh: người thuê có thể gửi yêu cầu sửa chữa, đề nghị hỗ trợ hoặc phản ánh lỗi hệ thống đến chủ chung cư.

Admin (chủ chung cư): người quản lý toàn bộ hệ thống, chịu trách nhiệm vận hành, giám sát và xử lý tất cả các hoạt động liên quan đến căn hộ, người thuê, hợp đồng, hóa đơn và các dịch vụ. Các chức năng admin:

- Quản lý thông tin căn hộ: Thêm mới, cập nhật, xóa và theo dõi trạng thái các căn hộ (còn trống, đang cho thuê, đang bảo trì), bao gồm diện tích, giá thuê, hình ảnh và tiện ích.
- Quản lý hợp đồng thuê: theo dõi thời hạn thuê, ngày bắt đầu – ngày kết thúc, và tình trạng hợp đồng. Theo dõi lịch sử thanh toán: xem các hóa đơn đã trả, thời gian thanh toán và trạng thái xử lý.
- Quản lý hóa đơn: Tạo hóa đơn điện, nước, internet và phí dịch vụ hằng tháng cho từng phòng; cập nhật trạng thái thanh toán; xuất hóa đơn khi cần.
- Quản lý người thuê và người ở: Theo dõi thông tin người thuê chính và danh sách người ở trong từng căn hộ.
- Xử lý yêu cầu – phản ánh từ người thuê: Tiếp nhận, phân công xử lý, cập nhật trạng thái và phản hồi các yêu cầu sửa chữa, bảo trì, khiếu nại.

3.1.2. Yêu cầu phi chức năng

Hiệu suất:

Hệ thống cần đảm bảo tốc độ phản hồi nhanh, đặc biệt trong các thao tác quan trọng như tải danh sách phòng, hiển thị hóa đơn hoặc xử lý yêu cầu từ người thuê. Dữ liệu điện – nước và hóa đơn phải được truy xuất mượt mà ngay cả khi có nhiều người truy cập cùng lúc.

Bảo mật:

Thông tin cá nhân của người thuê, hợp đồng, chỉ số điện – nước và dữ liệu thanh toán phải được bảo mật chặt chẽ. Các tài khoản đăng nhập cần được mã hóa mật khẩu, phân quyền rõ ràng nhằm tránh truy cập trái phép và đảm bảo an toàn dữ liệu.

Khả năng mở rộng:

Hệ thống phải dễ dàng tích hợp thêm các chức năng trong tương lai như thanh toán trực tuyến, quản lý camera an ninh, quản lý bãi giữ xe hoặc phát triển ứng dụng di động dành cho người thuê và chủ chung cư.

Tương thích thiết bị:

Giao diện website phải thân thiện, hiển thị tốt trên nhiều thiết bị như máy tính, máy tính bảng và điện thoại di động. Hệ thống cần đảm bảo trải nghiệm người dùng nhất quán dù sử dụng trên bất kỳ kích thước màn hình nào.

3.1.3. Yêu cầu hệ thống

Hệ thống quản lý chung cư mini được phát triển với kiến trúc hiện đại gồm 3 thành phần chính. Ở tầng giao diện, ứng dụng web sử dụng ReactJS để xây dựng giao diện người dùng thân thiện, hỗ trợ đa thiết bị với hai vai trò chính là quản trị viên và cư dân. Người dùng có thể dễ dàng quản lý thông tin cá nhân, xem hợp đồng, thanh toán hóa đơn, quản lý người ở và gửi yêu cầu bảo trì thông qua giao diện trực quan.

Tầng xử lý nghiệp vụ được xây dựng bằng Node.js và ExpressJS, cung cấp các API RESTful đảm bảo bảo mật thông qua cơ chế xác thực JWT. Hệ thống xử lý đa dạng nghiệp vụ từ quản lý người dùng, hợp đồng, thanh toán đến xử lý yêu cầu dịch vụ và gửi thông báo. Dữ liệu được lưu trữ trên nền tảng MongoDB với các collection được thiết kế tối ưu cho việc quản lý thông tin cư dân, hợp đồng, hóa đơn và yêu cầu dịch vụ.

Về triển khai, hệ thống được thiết kế linh hoạt với khả năng mở rộng cao, có thể triển khai trên các nền tảng đám mây như Vercel hoặc máy chủ riêng. Tính năng bảo mật được chú trọng với việc mã hóa dữ liệu nhạy cảm, phân quyền chi tiết và kiểm soát truy cập chặt chẽ. Hệ thống cũng hỗ trợ tích hợp với các dịch vụ thanh toán trực tuyến và gửi email thông báo tự động, đem lại trải nghiệm toàn diện cho cả người quản lý và cư dân.

3.2. Phân tích theo yêu cầu chức năng và tác nhân

3.2.1 Chức năng chi tiết theo bảng

Bảng 3.1 Mô tả chức năng

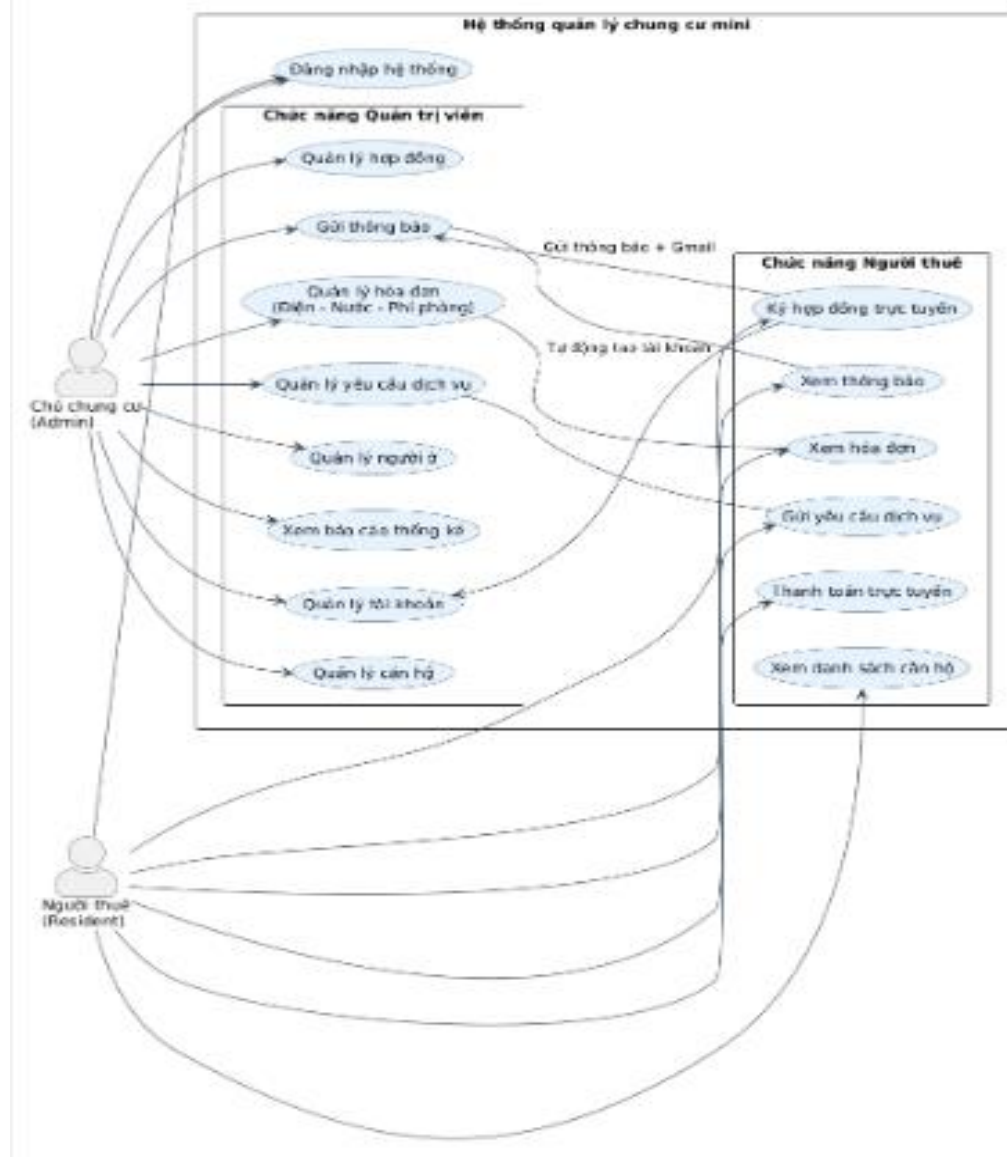
STT	Tác nhân	Chức năng	Mô tả
1	Admin (Chủ chung cư)	Quản lý tài khoản	Xem, xóa tài khoản người dùng và cấp quyền truy cập
2	Admin (Chủ chung cư)	Quản lý căn hộ	Thêm mới, cập nhật thông tin, xóa căn hộ trong tòa nhà
3	Admin (Chủ chung cư)	Quản lý hợp đồng	Xem , tải file pdf và xóa hợp đồng

STT	Tác nhân	Chức năng	Mô tả
4	Admin (Chủ chung cư)	Quản lý hóa đơn	Tạo hóa đơn, xác nhận thanh toán, in hóa đơn và báo cáo thu chi
5	Admin (Chủ chung cư)	Quản lý dịch vụ	Tiếp nhận, xử lý và phản hồi các yêu cầu bảo trì từ cư dân
6	Admin (Chủ chung cư)	Báo cáo thống kê	Xem báo cáo doanh thu, tỷ lệ lấp đầy và các thống kê khác
7	Người thuê	Thông tin cá nhân	Xem và cập nhật thông tin cá nhân, đổi mật khẩu
8	Người thuê	Hợp đồng	Xem thông tin hợp đồng, tải hợp đồng dạng PDF
9	Người thuê	Hóa đơn	Xem lịch sử hóa đơn, thanh toán trực tuyến, tải hóa đơn
10	Người thuê	Quản lý người ở	Thêm, xem, cập nhật thông tin người cùng ở
11	Người thuê	Yêu cầu dịch vụ	Tạo mới, theo dõi trạng thái yêu cầu bảo trì, sửa chữa
12	Người thuê	Thông báo	Nhận và xem các thông báo từ ban quản lý
13	Hệ thống	Gửi thông báo	Tự động gửi thông báo qua email/ứng dụng về hóa đơn, bảo trì

3.2.2. Yêu cầu phi chức năng

Hệ thống được thiết kế đảm bảo độ tin cậy cao với cơ chế xác thực JWT và phân quyền chặt chẽ cho từng vai trò người dùng. Giao diện đáp ứng linh hoạt trên mọi thiết bị từ desktop đến mobile, tối ưu hóa trải nghiệm người dùng. Về bảo mật, hệ thống áp dụng mã hóa dữ liệu nhạy cảm, kiểm tra đầu vào và bảo vệ chống tấn công phổ biến. Kiến trúc module hóa giúp dễ dàng mở rộng thêm tính năng trong tương lai như thanh toán trực tuyến, đặt lịch bảo trì. Hệ thống được tối ưu về hiệu năng với thời gian phản hồi nhanh, hỗ trợ đa người dùng đồng thời và dễ dàng bảo trì nhờ cấu trúc code rõ ràng, tài liệu đầy đủ.

3.3. Mô hình hóa yêu cầu

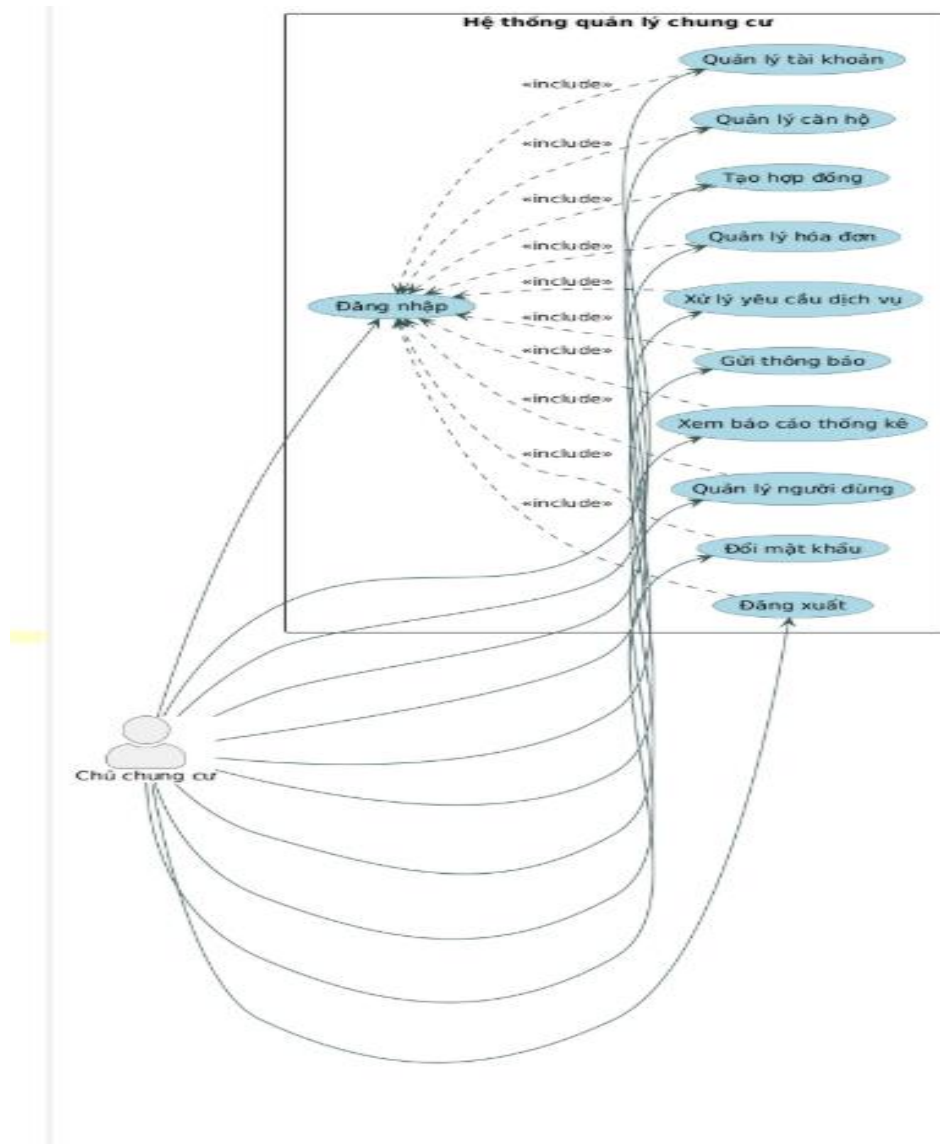


Hình 3.1 Mô hình Use case tổng quát.

Bảng 3.2 Mô tả Use Case tổng quát.

STT	Tác nhân	Use Case chính	Use Case phụ (nếu có)	Mô tả chức năng
1	Chủ chung cư (Admin)	Đăng nhập	—	Admin đăng nhập để truy cập hệ thống và các chức năng quản lý.
2	Người thuê	Xem danh sách căn hộ	—	Người thuê xem thông tin căn hộ: diện tích, giá, tiện nghi, tình trạng.
3	Người thuê	Ký hợp đồng trực tuyến	—	Người thuê ký hợp đồng thuê phòng trên web. Sau khi ký, hệ thống gửi tài khoản + mật khẩu về Gmail.
4	Người thuê	Đăng nhập hệ thống	—	Người thuê đăng nhập để xem hóa đơn, thông báo, yêu cầu dịch vụ.
5	Chủ chung cư (Admin)	Quản lý tài khoản người thuê	—	Xem và xóa tài khoản
6	Chủ chung cư (Admin)	Quản lý căn hộ	—	Quản lý thông tin căn hộ, thêm sửa xóa và trạng thái (trống/đã thuê).
7	Chủ chung cư (Admin)	Quản lý hóa đơn (Điện – Nước – Phí phòng)	—	Tạo và gửi hóa đơn cho người thuê hằng tháng và xác nhận thanh toán.
8	Chủ chung cư (Admin)	Quản lý hợp đồng	—	Quản lý hợp đồng: tạo, cập nhật, chấm dứt, lưu trữ.

9	Người thuê	Xem hóa đơn	—	Người thuê xem hóa đơn điện, nước, phí phòng.
10	Chủ chung cư (Admin)	Quản lý yêu cầu dịch vụ	—	Xử lý các yêu cầu hỗ trợ, sửa chữa của người thuê.
11	Người thuê	Gửi yêu cầu dịch vụ	—	Gửi yêu cầu sửa chữa, hỗ trợ.
12	Chủ chung cư (Admin)	Gửi thông báo	—	Gửi thông báo đến tất cả cư dân hoặc từng phòng.
13	Người thuê	Xem thông báo	—	Người thuê xem thông báo mà Admin gửi.
14	Chủ chung cư (Admin)	Quản lý người ở	—	Quản lý thông tin người ở trong từng căn hộ.
15	Người thuê	Thanh toán trực tuyến	—	Thanh toán hóa đơn qua ngân hàng hoặc ví điện tử.
16	Chủ chung cư (Admin)	Xem báo cáo thống kê	—	Xem tổng hợp số liệu: doanh thu, số phòng, số hợp đồng.

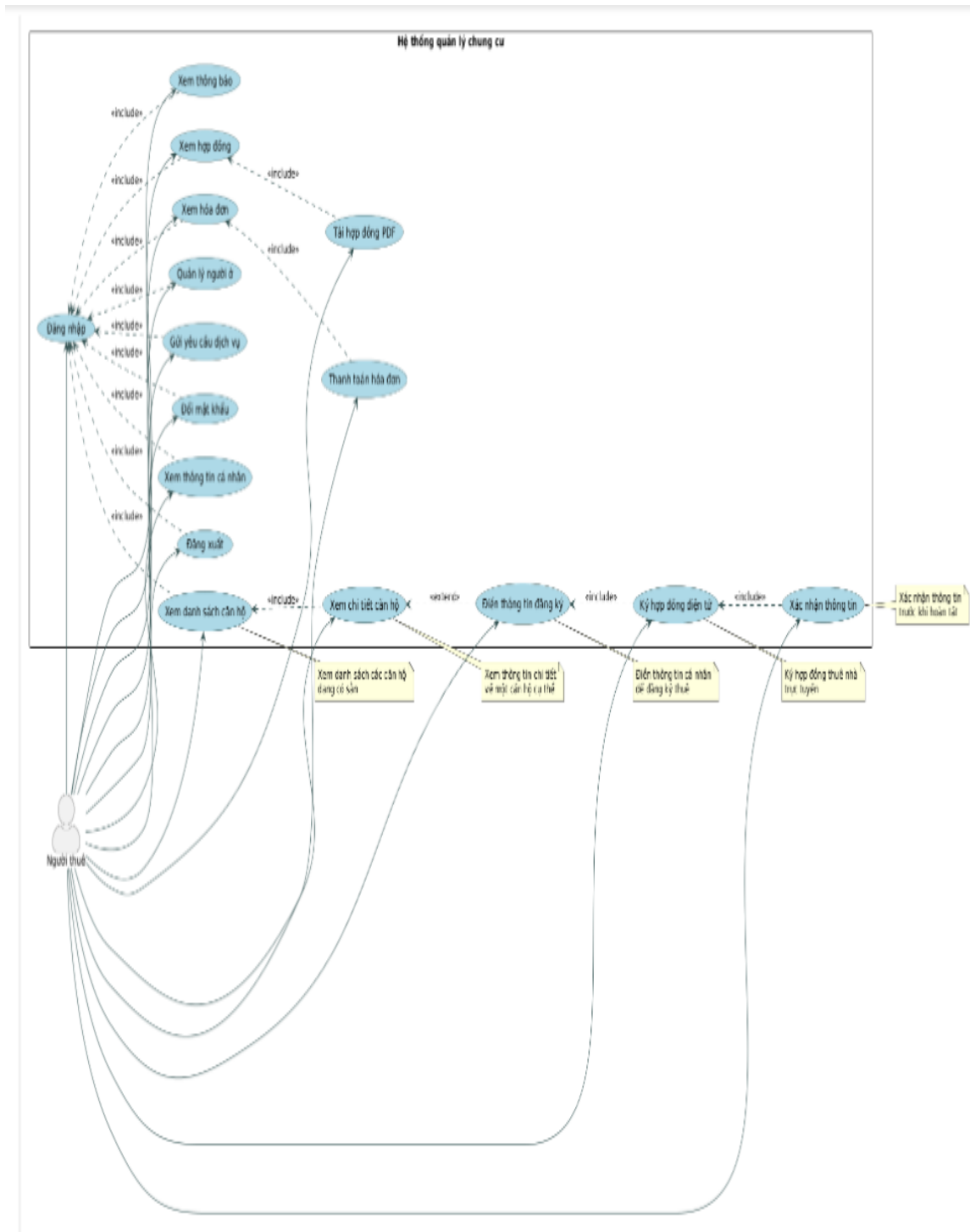


Hình 3.2 Mô hình Use case admin.

Bảng 3.3 Mô tả Use Case Chủ chung cư

STT	Use Case chính	Use Case phụ (Include)	Mô tả chức năng
1	Đăng nhập	Kiểm tra tài khoản	Cho phép Chủ chung cư truy cập hệ thống
2	Quản lý tài khoản	Kiểm tra email trùng	Quản lý tài khoản tenant trong hệ thống
3	Quản lý căn hộ	Thêm căn hộ	Quản lý danh sách phòng/căn hộ

		Sửa thông tin Cập nhật trạng thái (trống/đã thuê)	
4	Xem hợp đồng	Tải PDF	Xem chi tiết và tải hợp đồng PDF
5	Quản lý hóa đơn	Tính tiền điện Tính tiền nước Cập nhật tiền dịch vụ Gửi hóa đơn	Tạo và gửi hóa đơn hằng tháng
6	Xử lý yêu cầu dịch vụ	Xem yêu cầu Phản hồi yêu cầu Cập nhật trạng thái yêu cầu	Xử lý yêu cầu sửa chữa, khiếu nại
7	Gửi thông báo	Chọn đối tượng nhận Gửi thông báo qua hệ thống	Gửi tin nhắn hoặc thông báo chung
8	Quản lý người dùng		Quản lý tài khoản admin/người thuê
9	Quản lý người ở		Xem thông tin từng người ở tại căn hộ
10	Xem báo cáo thống kê	Lấy dữ liệu doanh thu	Xem báo cáo tổng quan hệ thống

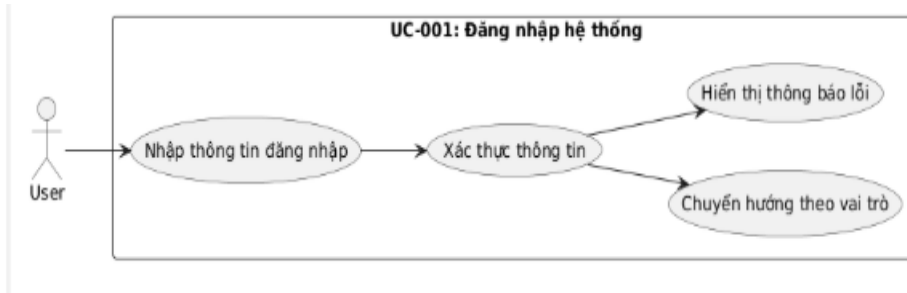


Hình 3.3 Mô hình Use case chủ Người thuê.

Bảng 3.4 Mô tả Use Case Người thuê

STT	Tên Use Case chính	Use Case phụ (Include/Extend)	Mô tả
1	Xem căn hộ & ký hợp đồng	Xem danh sách căn hộ Xem chi tiết căn hộ Gửi yêu cầu ký hợp đồng	Người thuê chọn căn → yêu cầu ký hợp đồng
2	Nhận tài khoản Gmail sau khi ký hợp đồng	Gửi Gmail tự động (NodeJS)	Sau khi ký hợp đồng, hệ thống gửi tài khoản
3	Đăng nhập	Kiểm tra thông tin	Người đăng nhập trang cá nhân
4	Xem hợp đồng	Tải hợp đồng PDF	Xem hoặc tải bản hợp đồng đã ký
5	Xem hóa đơn	Lọc hóa đơn theo tháng Hiển thị chi tiết	Người thuê xem hóa đơn được chủ chung cư gửi
6	Thanh toán hóa đơn	Chọn phương thức Xác nhận thanh toán	Thanh toán chuyển khoản
7	Gửi yêu cầu hỗ trợ	Chọn loại yêu cầu Gửi ảnh/sự cố	Báo hỏng, yêu cầu sửa chữa
8	Xem thông báo	Lọc thông báo Đánh dấu đã đọc	Nhận thông báo từ admin

- Sơ đồ use case chức năng đăng nhập:

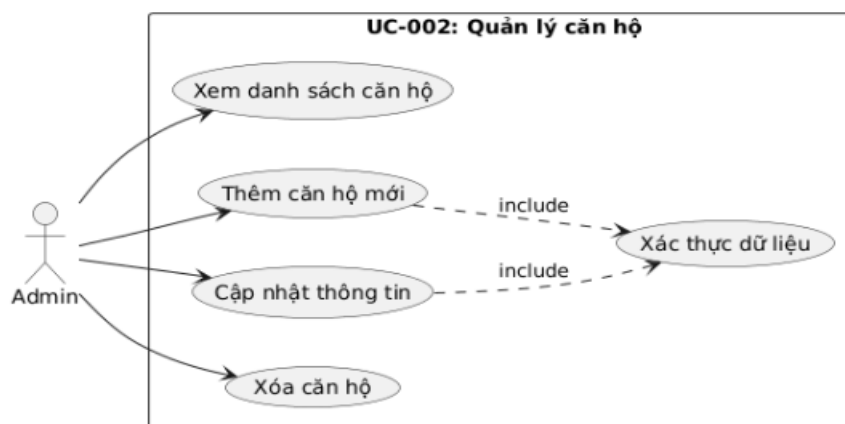


Hình 3.4 Mô hình Use case đăng nhập.

Bảng 3.5 Mô tả Use Case chức năng đăng nhập

Tên Use Case	Đăng nhập hệ thống
Tác nhân	User (Người thuê / Admin)
Mục tiêu	Cho phép người dùng đăng nhập vào hệ thống theo đúng vai trò.
Điều kiện trước	Người dùng đã có tài khoản hợp lệ.
Điều kiện sau	- Đăng nhập thành công → chuyển hướng theo vai trò. - Thất bại → hiển thị thông báo lỗi.
Luồng chính	1. User truy cập trang đăng nhập. 2. Nhập tên đăng nhập và mật khẩu. 3. Hệ thống kiểm tra thông tin. 4. Xác thực thành công → chuyển vào hệ thống.
Luồng ngoại lệ	1: Email không tồn tại → Thông báo lỗi. 2: Sai mật khẩu → Thông báo lỗi.

- Sơ đồ use case chức năng quản lý căn hộ:

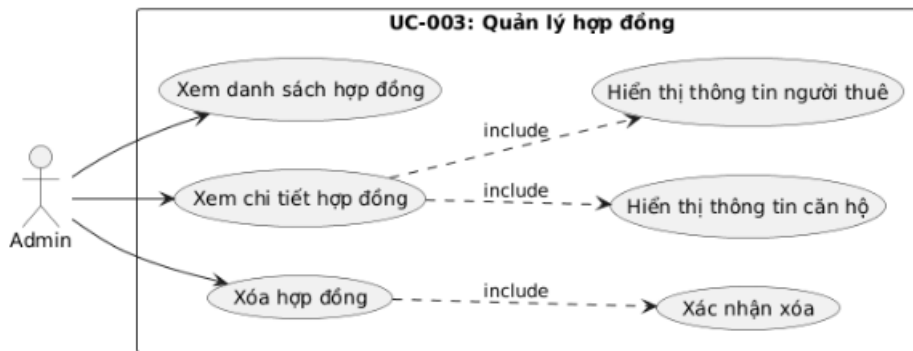


Hình 3.5 Mô hình Use case quản lý căn hộ.

Bảng 3.6 Mô tả Use Case chức năng Quản lý căn hộ.

Tên Use Case	Quản lý căn hộ
Tác nhân	Admin
Mục tiêu	Admin có thể xem, thêm, cập nhật và xóa thông tin căn hộ trong hệ thống.
Điều kiện trước	Admin đã đăng nhập hệ thống.
Điều kiện sau	- Thông tin căn hộ được lưu/ cập nhật/ xóa thành công. - Danh sách căn hộ hiển thị chính xác.
Luồng chính	1. Xem danh sách căn hộ - Admin truy cập danh sách căn hộ. 2. Thêm căn hộ mới - Admin nhập thông tin căn hộ. - Hệ thống <i>include</i> bước “Xác thực dữ liệu”. - Lưu căn hộ mới vào DB. 3. Cập nhật thông tin - Admin chỉnh sửa thông tin căn hộ. - Hệ thống <i>include</i> “Xác thực dữ liệu”. - Lưu cập nhật vào DB. 4. Xóa căn hộ - Admin chọn căn hộ cần xóa. - Hệ thống xác nhận và xóa.
Luồng ngoại lệ	1: Thiếu thông tin khi thêm/sửa → Báo lỗi. 2: Dữ liệu không hợp lệ → Không lưu vào DB. 3: Xóa căn hộ đang có hợp đồng → Từ chối xóa.

- Sơ đồ use case chức năng quản lý hợp đồng:

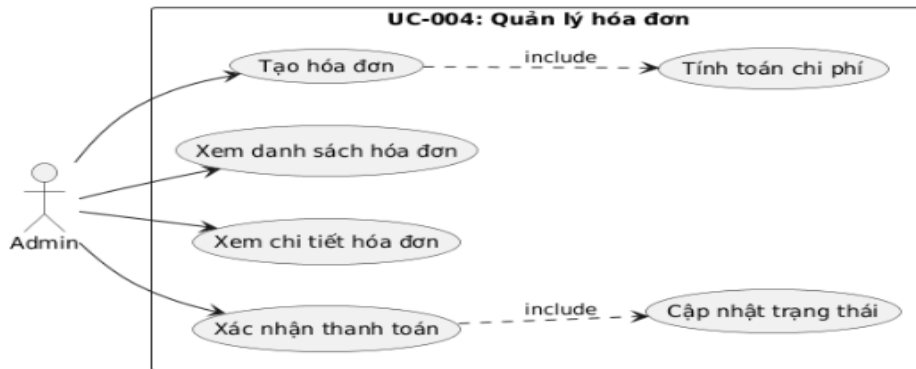


Hình 3.6 Mô hình Use case chức năng quản lý hợp đồng.

Bảng 3.7 Mô tả Use Case chức năng quản lý hợp đồng.

Tên Use Case	Quản lý hợp đồng
Tác nhân	Admin
Mục tiêu	Cho phép Admin xem danh sách hợp đồng, xem chi tiết hợp đồng và xóa hợp đồng.
Điều kiện trước	Admin đã đăng nhập hệ thống.
Điều kiện sau	Thông tin hợp đồng được hiển thị hoặc hợp đồng được xóa thành công.
Luồng chính	- Xem danh sách hợp đồng - Admin truy cập danh sách hợp đồng. - Xem chi tiết hợp đồng - Admin chọn hợp đồng cần xem. - Hệ thống hiển thị thông tin hợp đồng. <i>include</i>: Hiển thị thông tin người thuê. <i>include</i>: Hiển thị thông tin căn hộ. - Xóa hợp đồng - Admin chọn hợp đồng cần xóa. <i>include</i>: Hệ thống yêu cầu xác nhận xóa. - Hợp đồng được xóa khỏi DB.
Luồng ngoại lệ	- Hợp đồng không tồn tại → Thông báo lỗi. - Không thể xóa vì đang còn hiệu lực hoặc liên quan dữ liệu khác → Từ chối xóa

- Sơ đồ use case chức năng quản lý hóa đơn:



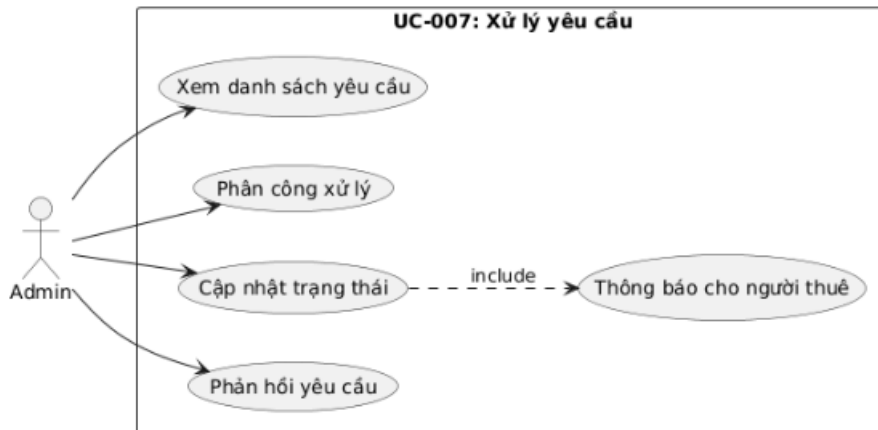
Hình 3.7 Mô hình Use case chức năng quản lý hóa đơn.

Bảng 3.8 Mô tả Use Case chức năng quản lý hóa đơn.

Tên Use Case	Quản lý hóa đơn
Tác nhân	Admin
Mục tiêu	Cho phép Admin tạo hóa đơn, xem danh sách – chi tiết hóa đơn và xác nhận thanh toán.
Điều kiện trước	Admin đã đăng nhập hệ thống.
Điều kiện sau	Hóa đơn được tạo, hiển thị đúng hoặc cập nhật trạng thái thanh toán thành công.
Luồng chính	- Tạo hóa đơn - Admin chọn “Tạo hóa đơn”. - <i>include</i>: Hệ thống tự động tính toán chi phí (điện, nước, dịch vụ). - Hệ thống lưu hóa đơn vào DB. - Xem danh sách hóa đơn - Admin truy cập mục “Danh sách hóa đơn”. - Hệ thống hiển thị toàn bộ hóa đơn. - Xem chi tiết hóa đơn - Admin chọn hóa đơn cần xem. - Hệ thống hiển thị đầy đủ các khoản phí và trạng thái thanh toán. - Xác nhận thanh toán

	- Admin chọn hóa đơn đã được chuyển khoản. - <i>include</i>: Hệ thống cập nhật trạng thái thanh toán.
Luồng ngoại lệ	1: Hóa đơn không tồn tại → Thông báo lỗi. 2: Không thể xác nhận thanh toán khi trạng thái không hợp lệ.

- Sơ đồ use case chức năng xử lý yêu cầu:



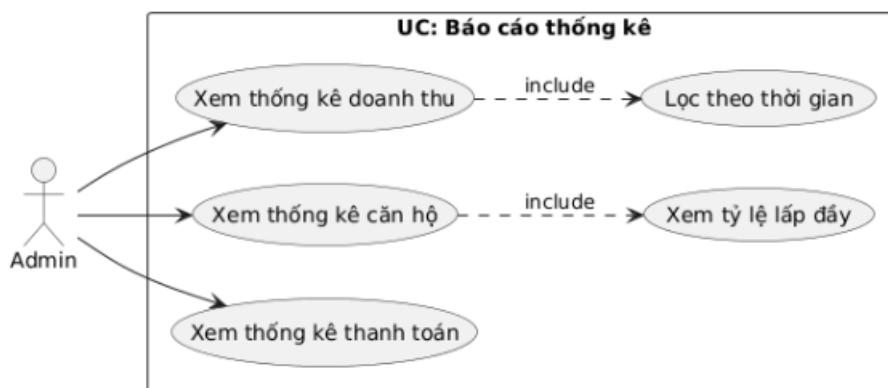
Hình 3.8 Mô hình Use case chức năng xử lý yêu cầu.

Bảng 3.9 Mô tả Use Case chức năng xử lý yêu cầu.

Tên Use Case	Xử lý yêu cầu
Tác nhân	Admin
Mục tiêu	Cho phép Admin xem, phân công, cập nhật trạng thái và phản hồi yêu cầu dịch vụ của người thuê.
Điều kiện trước	Admin đã đăng nhập hệ thống.
Điều kiện sau	Yêu cầu được cập nhật, phản hồi hoặc thông báo đến người thuê thành công.
Luồng chính	1. Xem danh sách yêu cầu - Admin mở mục “Yêu cầu dịch vụ”. - Hệ thống hiển thị danh sách tất cả yêu cầu. 2. Phân công xử lý - Admin chọn yêu cầu và phân công cho người phụ trách (nếu có). 3. Cập nhật trạng thái yêu cầu

	- Admin chuyển trạng thái: Chờ xử lý → Đang xử lý → Hoàn thành. - <i>include</i>: Hệ thống gửi thông báo cho người thuê. 4. Phản hồi yêu cầu - Admin gửi phản hồi trực tiếp cho người thuê qua hệ thống.
Luồng ngoại lệ	1: Yêu cầu không tồn tại → Thông báo lỗi. 2: Không thể cập nhật trạng thái → Dữ liệu không hợp lệ. 3: Lỗi khi gửi thông báo → Hệ thống ghi log và yêu cầu thử lại.

- Sơ đồ use case chức năng báo cáo thống kê:



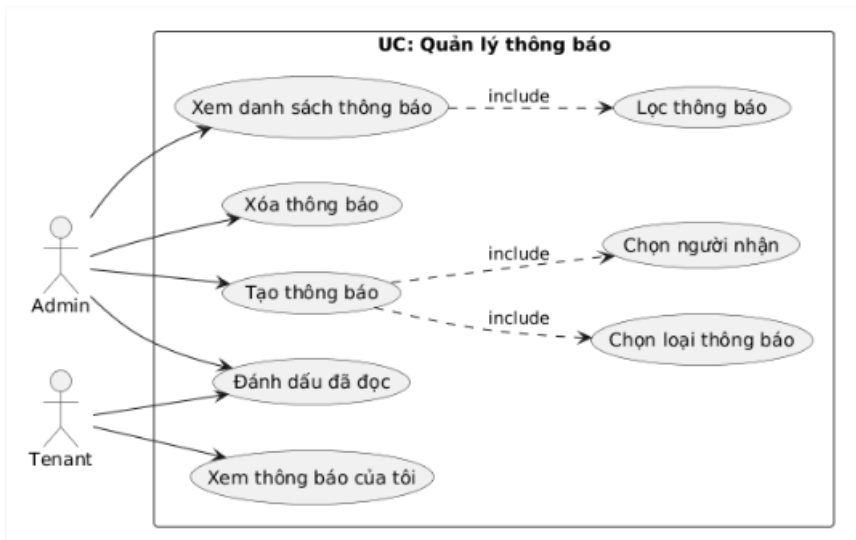
Hình 3.9 Mô hình Use case chức năng báo cáo thống kê.

Bảng 3.10 Mô tả Use Case chức năng báo cáo thống kê.

Tên Use Case	Báo cáo thống kê
Tác nhân	Admin
Mục tiêu	Cho phép Admin xem thống kê doanh thu, thống kê căn hộ và thống kê thanh toán.
Điều kiện trước	Admin đã đăng nhập hệ thống.
Điều kiện sau	Báo cáo được hiển thị chính xác theo lựa chọn của Admin.
Luồng chính	1. Xem thống kê doanh thu - Admin chọn mục doanh thu. - <i>include</i>: Lọc theo thời gian. 2. Xem thống kê căn hộ - Admin chọn mục thống kê căn hộ. - <i>include</i>: Xem tỷ lệ lấp đầy.

	3. Xem tổng kê thanh toán - Admin chọn mục tổng kê thanh toán. - Hệ thống hiển thị các hóa đơn đã/ chưa thanh toán.
Luồng ngoại lệ	Không có dữ liệu theo thời gian lọc → Thông báo “Không có dữ liệu”.

- Sơ đồ use case chức năng quản lý thông báo:



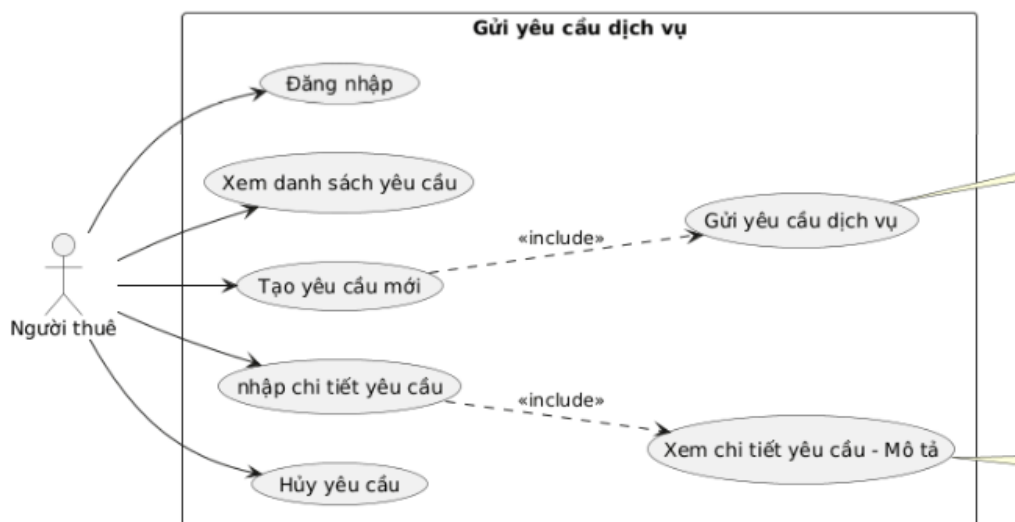
Hình 3.10 Mô hình Use case chức năng quản lý thông báo.

Bảng 3.11 Mô tả Use Case chức năng quản lý thông báo.

Tên Use Case	Quản lý thông báo
Tác nhân	Admin, Tenant (Người thuê)
Mục tiêu	Cho phép Admin tạo, xem, lọc, xóa thông báo. Tenant có thể xem và đánh dấu thông báo đã đọc.
Điều kiện trước	Người dùng đã đăng nhập hệ thống.
Điều kiện sau	Thông báo được tạo, hiển thị, cập nhật hoặc xóa thành công.
Luồng chính	1. Xem danh sách thông báo (Admin/Tenant) - Người dùng mở danh sách thông báo. - <i>include</i>: Lọc thông báo theo thời gian hoặc loại. 2. Tạo thông báo (Admin) - Admin nhập nội dung thông báo. - <i>include</i>: Chọn người nhận.

	- <i>include</i>: Chọn loại thông báo. - Hệ thống lưu thông báo vào DB. 3. Xóa thông báo (Admin) - Admin chọn thông báo và thực hiện xóa. 4. Xem thông báo của tôi (Tenant) - Tenant xem danh sách thông báo dành riêng cho mình. 5. Đánh dấu đã đọc (Tenant) - Tenant chọn thông báo → đánh dấu “Đã đọc”.
Luồng ngoại lệ	1: Không có thông báo phù hợp với bộ lọc → Thông báo “Không có dữ liệu”. 2: Thông báo không tồn tại → Lỗi và quay lại danh sách. 3: Người thuê không có quyền xóa thông báo.

- Sơ đồ use case chức năng gửi yêu cầu dịch vụ:



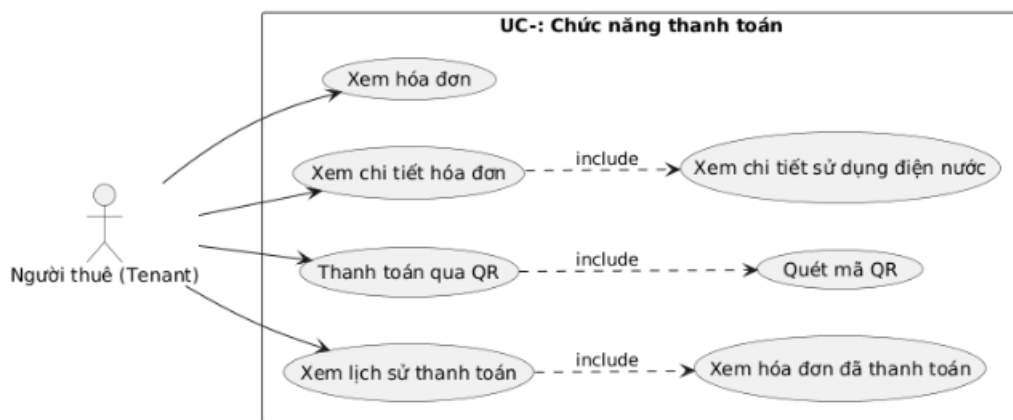
Hình 3.11 Mô hình Use case chức năng gửi yêu cầu dịch vụ.

Bảng 3.12 Mô tả Use Case chức năng gửi yêu cầu dịch vụ.

Tên Use Case	Gửi yêu cầu dịch vụ
Tác nhân	Tenant (Người thuê)
Mục tiêu	Cho phép người thuê gửi yêu cầu sửa chữa – bảo trì đến Admin một cách đầy đủ và chính xác.
Điều kiện trước	Người thuê đã đăng nhập hệ thống.

Điều kiện sau	Yêu cầu được gửi và lưu thành công, chờ Admin xử lý.
Luồng chính	1. Chọn loại yêu cầu - Tenant chọn loại dịch vụ: điện, nước, khóa, hư hỏng,... 2. Mô tả chi tiết - Tenant nhập mô tả sự cố, mức độ ảnh hưởng. 3. Đính kèm hình ảnh - Tenant tải lên hình ảnh minh họa (nếu có). 4. Gửi yêu cầu - Tenant nhấn “Gửi yêu cầu”. - <i>include</i> : Hệ thống xác nhận gửi và lưu yêu cầu vào DB.
Luồng ngoại lệ	1: Người thuê không nhập mô tả → Thông báo yêu cầu bắt buộc. 2: Upload hình ảnh lỗi → Hiển thị lỗi và yêu cầu thử lại. 3: Lỗi kết nối khi gửi yêu cầu → Thông báo “Gửi thất bại, vui lòng thử lại”.

- Sơ đồ use case chức năng thanh toán:



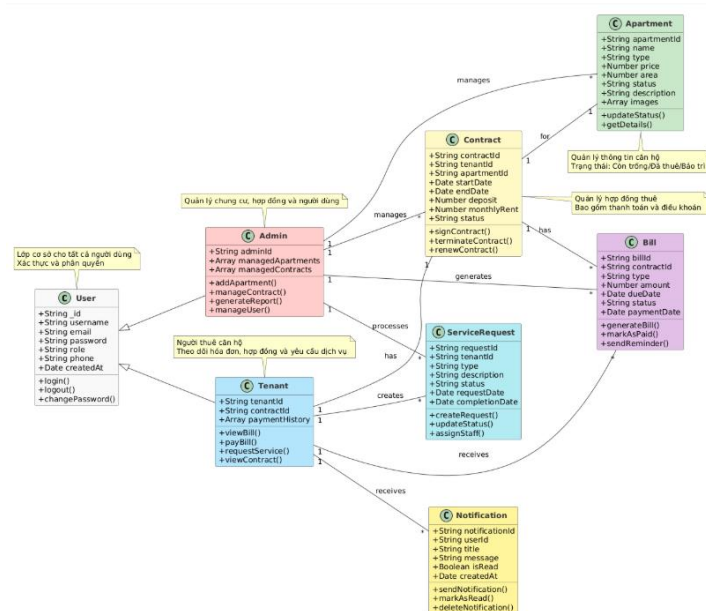
Hình 3.12 Mô hình Use case chức năng thanh toán.

Bảng 3.13 Mô tả Use Case chức năng đăng nhập.

Tên Use Case	Chức năng thanh toán
Tác nhân	Người thuê (Tenant)
Mục tiêu	Cho phép người thuê xem hóa đơn, kiểm tra chi tiết, thanh toán qua QR và xem lịch sử thanh toán.

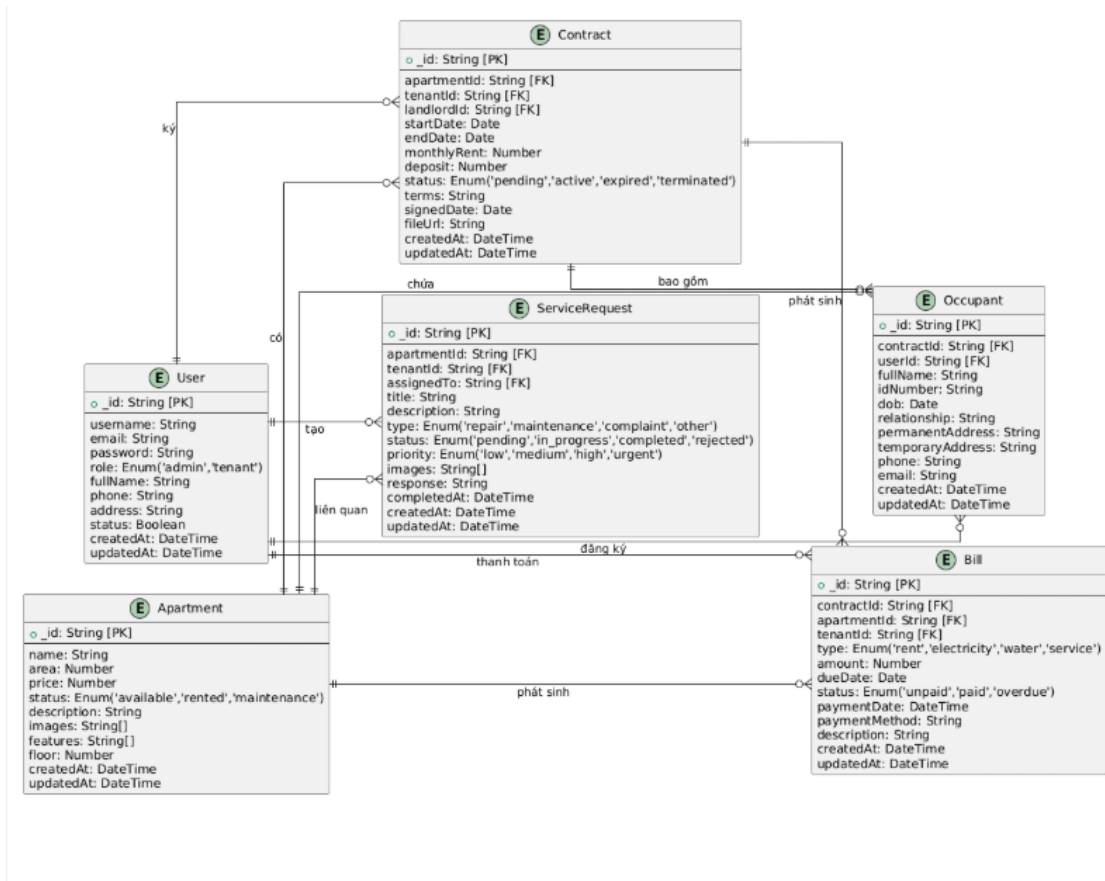
Điều kiện trước	Tenant đã đăng nhập hệ thống.
Điều kiện sau	Hệ thống ghi nhận việc xem hóa đơn, thanh toán hoặc xem lịch sử.
Luồng chính	- Xem hóa đơn - Tenant truy cập danh sách hóa đơn của mình. - Xem chi tiết hóa đơn - Tenant xem chi tiết số tiền, phí điện – nước, dịch vụ. - <i>include</i>: Hệ thống hiển thị chi tiết sử dụng điện nước. - Thanh toán qua QR - Tenant chọn “Thanh toán”. - <i>include</i>: Hệ thống cung cấp mã QR để quét. - Xem lịch sử thanh toán - Tenant xem danh sách các hóa đơn đã thanh toán. - <i>include</i>: Hiển thị hóa đơn đã thanh toán.
Luồng ngoại lệ	Không có dữ liệu theo thời gian lọc → Thông báo “Không có dữ liệu”.

- Sơ đồ class diagram:



Hình 3.13 Sơ đồ class diagram.

- Sơ đồ mô hình quan hệ:



Hình 3.14 Mô hình quan hệ.

Bảng 3.14 Users

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
_id	ObjectId	PRIMARY KEY	ID tự động tạo
username	String	UNIQUE, NOT NULL	Tên đăng nhập
password	String	NOT NULL	Mật khẩu (đã mã hóa)
role	String	ENUM: ['admin', 'tenant'], DEFAULT: 'tenant'	Vai trò người dùng
fullname	String	NOT NULL	Họ và tên đầy đủ
Phone	String	NOT NULL	Số điện thoại
email	String	NOT NULL	Địa chỉ email

idCard	String	NOT NULL	Số CMND/CCCD
occupants	Number	DEFAULT: 1	Số người ở
contractId	ObjectId	FOREIGN KEY (Contracts._id)	Hợp đồng thuê nhà
createAt	Date	AUTO	Ngày tạo
updateAt	Date	AUTO	Ngày cập nhật

Bảng 3.15 Apartments.

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
_id	ObjectId	PRIMARY KEY	ID tự động tạo
name	String	NOT NULL	Tên căn hộ (VD: A101)
floor	Number	NOT NULL	Tầng
price	Number	NOT NULL	Giá thuê (VNĐ)
area	Number	NOT NULL	Diện tích (VD: "30m ² ")
rooms	String	NOT NULL	Thông tin phòng (VD: "1 phòng ngủ, 1 phòng khách")
notes	String	NOT NULL	Ghi chú thêm
imgeUrls	[String]	NOT NULL	Danh sách URL hình ảnh
status	String	ENUM: ['available', 'rented'], DEFAULT: 'available'	Trạng thái căn hộ

Bảng 3.16 Contracts.

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
_id	ObjectId	PRIMARY KEY	ID tự động tạo
tenant	ObjectId	FOREIGN KEY (Users._id), NOT NULL	ID người thuê
tenantInfo	Object	NOT NULL	Thông tin người thuê (lưu trữ tĩnh)
apartment	ObjectId	FOREIGN KEY (Apartments._id), NOT NULL	ID căn hộ
apartmentInfo	Object	NOT NULL	Thông tin căn hộ (lưu trữ tĩnh)
rentAmount	Number	NOT NULL	Giá thuê (VNĐ)
contractDate	Date	DEFAULT: Date.now	Ngày lập hợp đồng
status	String	ENUM: ['active', 'expired', 'terminated'], DEFAULT: 'active'	Trạng thái hợp đồng
signedAt	Date	DEFAULT: Date.now	Ngày ký hợp đồng
createAt	Date	AUTO	Ngày tạo
updateAt	Date	AUTO	Ngày cập nhật

Bảng 3.17 Bill

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
_id	ObjectId	PRIMARY KEY	ID tự động tạo
apartment	ObjectId	FOREIGN KEY (Apartments._id), NOT NULL	ID căn hộ
tenant	ObjectId	FOREIGN KEY (Users._id), NOT NULL	ID người thuê
month	Number	MIN: 1, MAX: 12, NOT NULL	Tháng tính phí
year	Number	NOT NULL	Năm tính phí
electricityUsage	Number	DEFAULT: 0	Số điện tiêu thụ (kWh)
electricityPrice	Number	DEFAULT: 3500	Đơn giá điện (VNĐ/kWh)
electricityTotal	Number	DEFAULT: 0	Thành tiền điện
waterUsage	Number	DEFAULT: 0	Số nước tiêu thụ (m ³)
waterPrice	Number	DEFAULT: 15000	Đơn giá nước (VNĐ/m ³)
waterTotal	Number	DEFAULT: 0	Thành tiền nước
updateAt	Date	AUTO	Ngày cập nhật
rentAmount	Number	DEFAULT: 0	Tiền thuê nhà
otherFees	Number	DEFAULT: 0	Phí khác

otherFeesNote	String		Ghi chú phí khác
totalAmount	Number	DEFAULT: 0	Tổng tiền phải thanh toán
status	String	ENUM: ['unpaid', 'paid', 'overdue'], DEFAULT: 'unpaid'	Trạng thái thanh toán
paidDate	Date		Ngày thanh toán
note	String		Ghi chú
createdBy	ObjectId	FOREIGN KEY (Users._id), NOT NULL	ID người tạo hóa đơn
createAt	Date	AUTO	Ngày tạo
updateAt	Date	AUTO	Ngày cập nhật

Bảng 3.18 ServiceRequest.

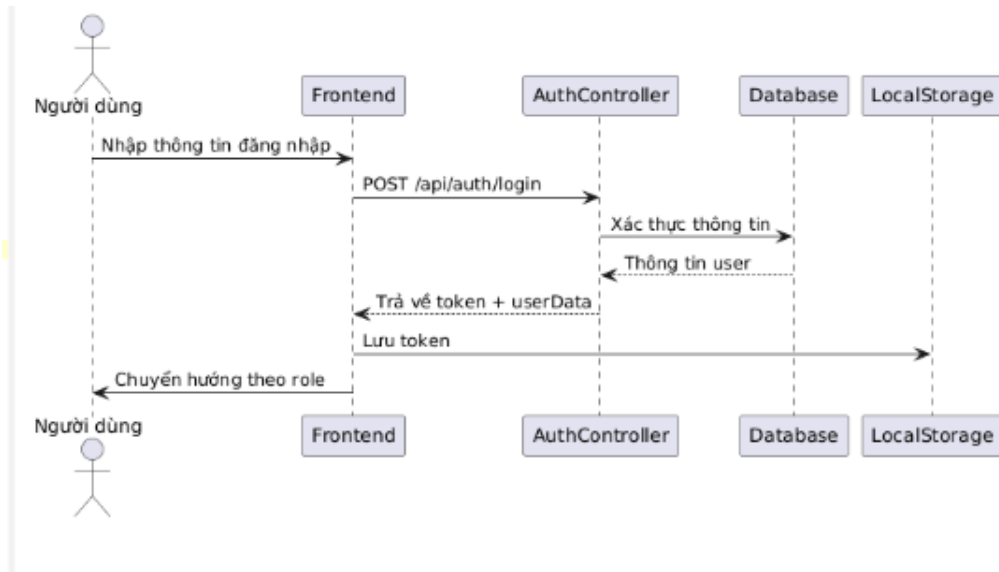
Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
_id	ObjectId	PRIMARY KEY	ID tự động tạo
tenant	ObjectId	FOREIGN KEY (Users._id), NOT NULL	ID người gửi yêu cầu
type	String	ENUM: ['electricity', 'water', 'maintenance', 'cleaning', 'security', 'other', 'lease_cancellation'], NOT NULL	Loại yêu cầu
priority	String	ENUM: ['low', 'medium', 'high', 'urgent'], DEFAULT: 'medium'	Mức độ ưu tiên

title	String	MAX: 200, NOT NULL	Tiêu đề yêu cầu
description	String	MAX: 1000, NOT NULL	Mô tả chi tiết
apartment	ObjectId	FOREIGN KEY (Apartments._id), NOT NULL	ID căn hộ liên quan
status	String	ENUM: ['pending', 'in_progress', 'completed', 'cancelled'], DEFAULT: 'pending'	Trạng thái yêu cầu
occupants	Number	DEFAULT: 1	Số người ở
adminResponse	String	MAX: 1000	Phản hồi từ admin
respondedAt	Date		Thời điểm phản hồi
completedAt	Date		Thời điểm hoàn thành
images	[String]		Danh sách tên file hình ảnh đính kèm
internalNotes	String	MAX: 500	Ghi chú nội bộ (chỉ admin thấy)
estimatedCompletion	Date		Thời gian dự kiến hoàn thành
createAt	Date	AUTO	Ngày tạo
updateAt	Date	AUTO	Ngày cập nhật

Bảng 3.19 Occupants.

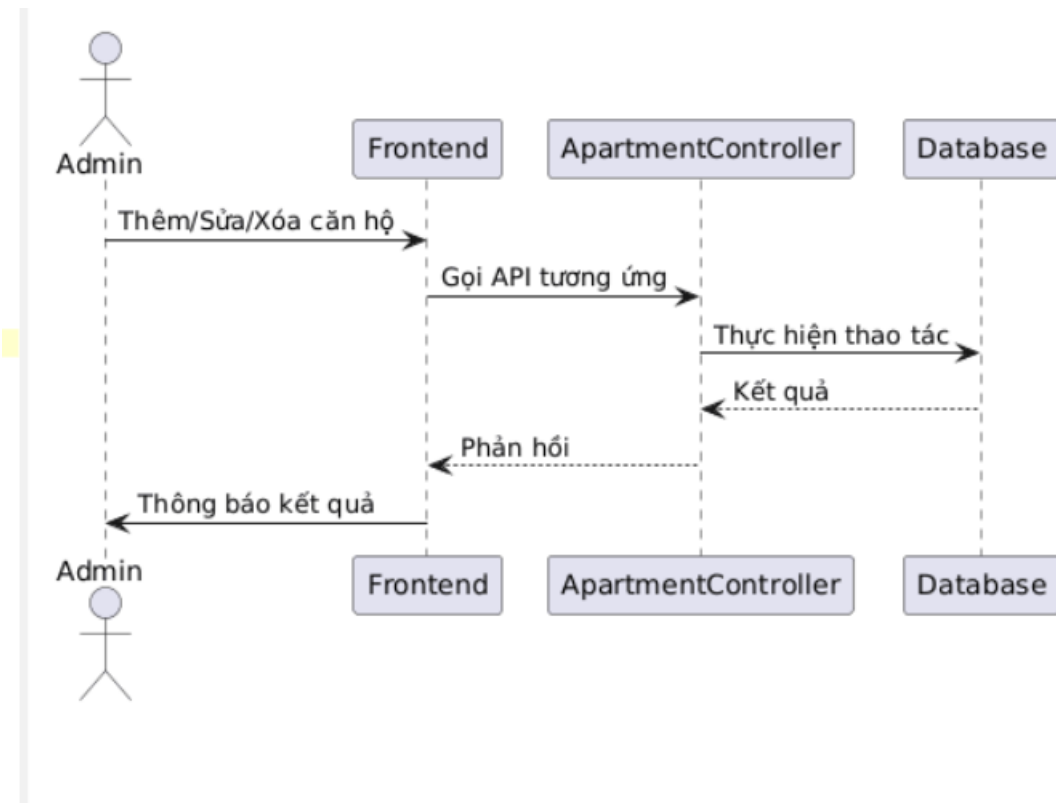
Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
_id	ObjectId	PRIMARY KEY	ID tự động tạo
name	String	NOT NULL	Họ và tên
idCard	String	NOT NULL	Số CMND/CCCD
phone	String		Số điện thoại
relationship	String		Mối quan hệ với chủ hợp đồng
contract	ObjectId	FOREIGN KEY (Contracts._id), NOT NULL	ID hợp đồng liên quan
email	String	NOT NULL	Địa chỉ email
isActive	Boolean	DEFAULT: true	Trạng thái hoạt động
notes	String		Ghi chú thêm
createAt	Date	AUTO	Ngày tạo
updateAt	Date	AUTO	Ngày cập nhật

- Sơ đồ trình tự đăng nhập:



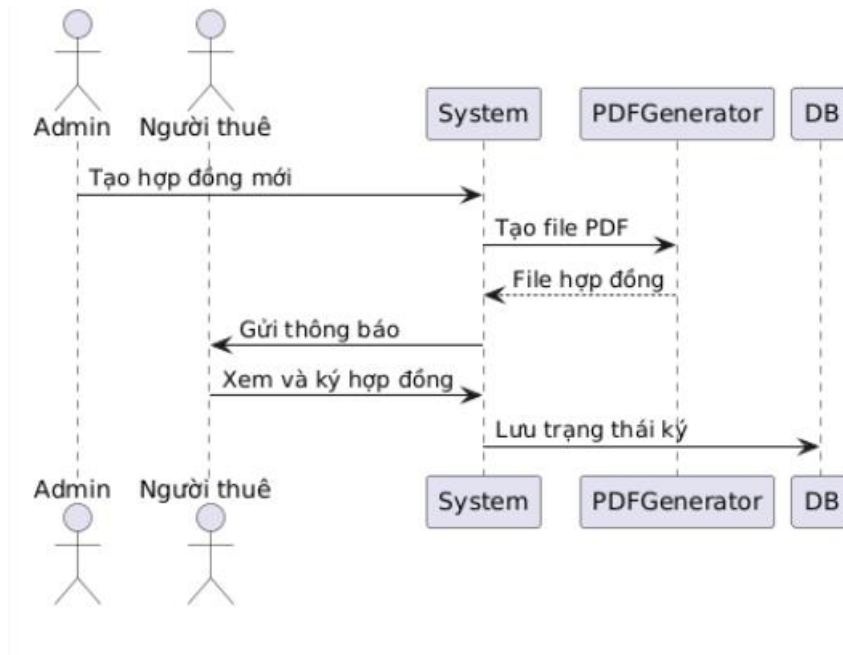
Hình 3.15 Sơ đồ chức năng đăng nhập.

- Sơ đồ trình tự admin quản lý căn hộ:



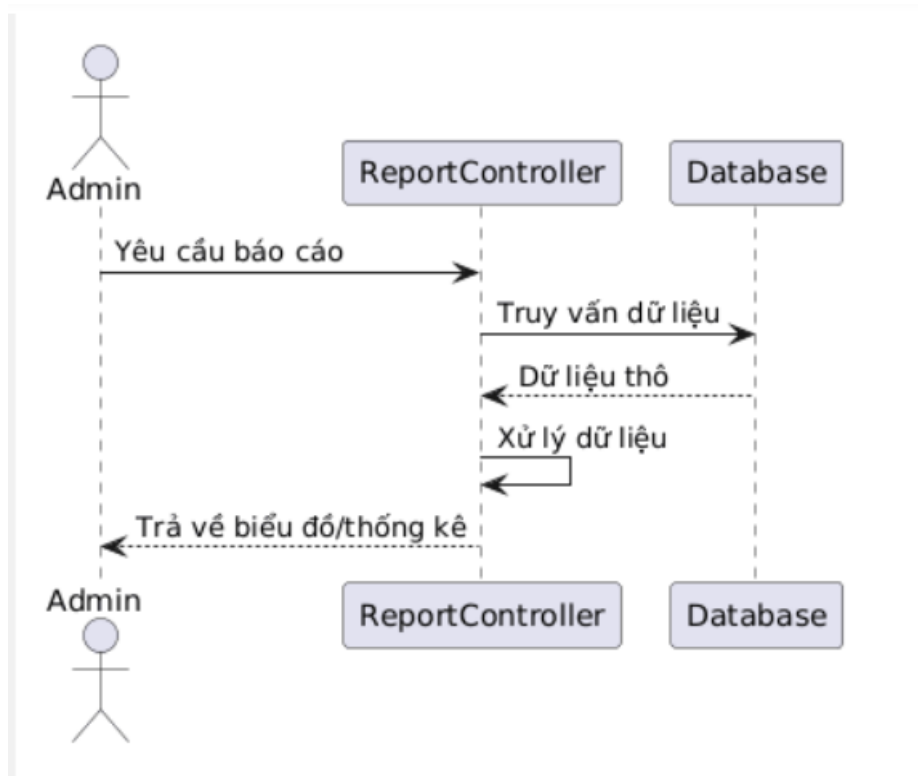
Hình 3.16 Sơ đồ trình tự chức năng quản lý căn hộ.

- Sơ đồ trình chức năng ký hợp đồng :



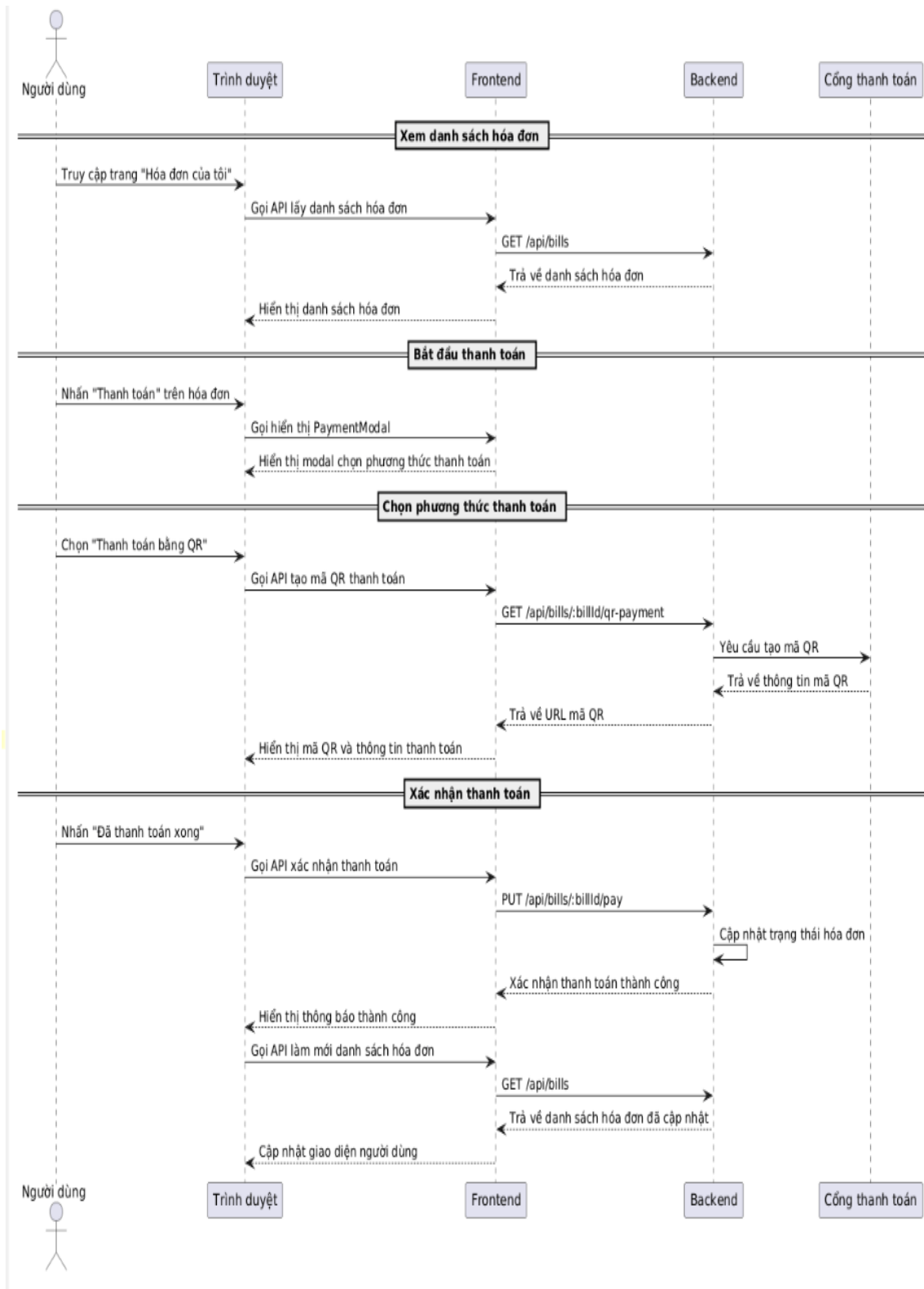
Hình 3.17 Sơ đồ trình tự chức năng ký hợp đồng.

- Sơ đồ trình tự báo cáo và thống kê:



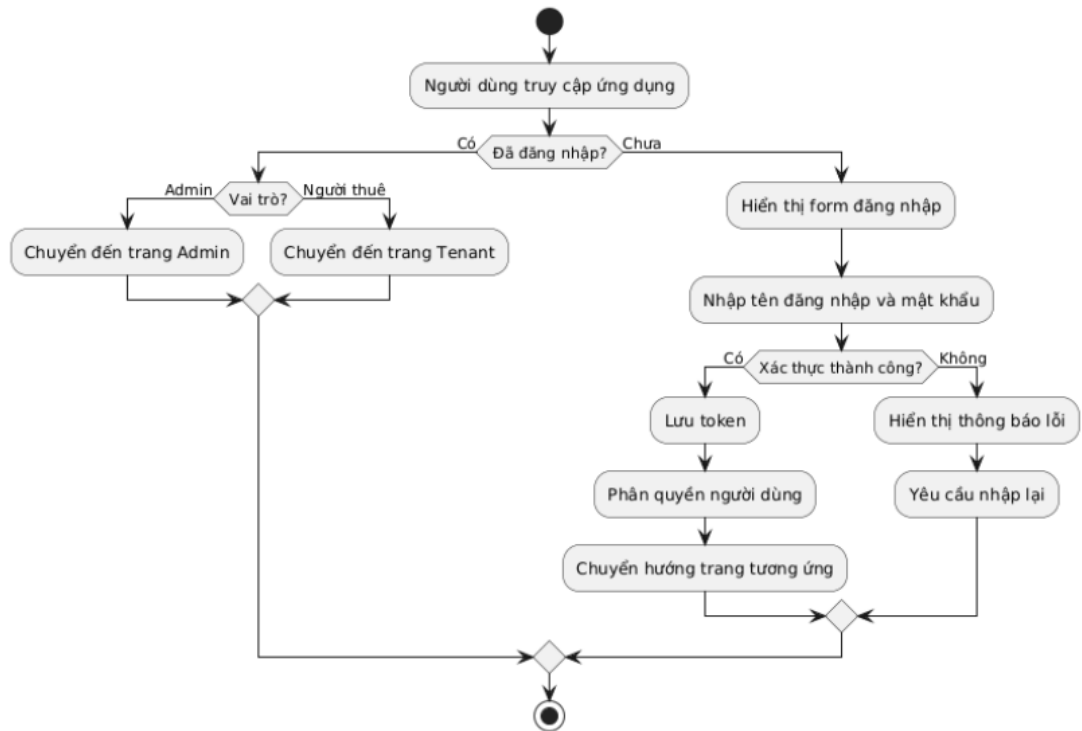
Hình 3.18 Sơ đồ trình tự chức năng báo cáo và thống kê.

- Sơ đồ trình tự chức năng thanh toán:



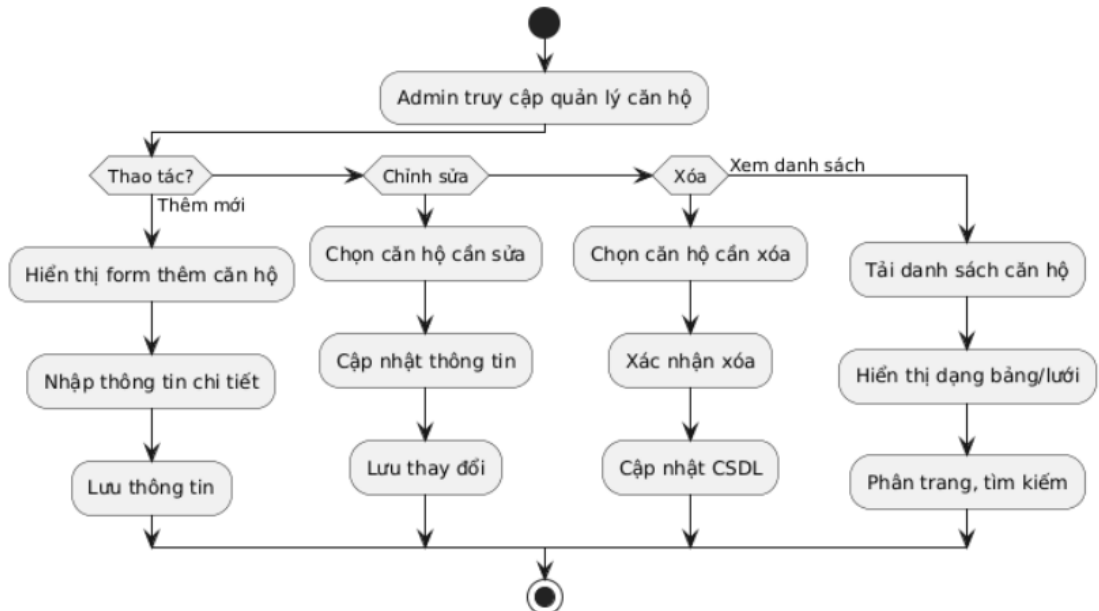
Hình 3.19 Sơ đồ trình tự chức năng thanh toán.

- Sơ đồ hoạt động chức năng đăng nhập:



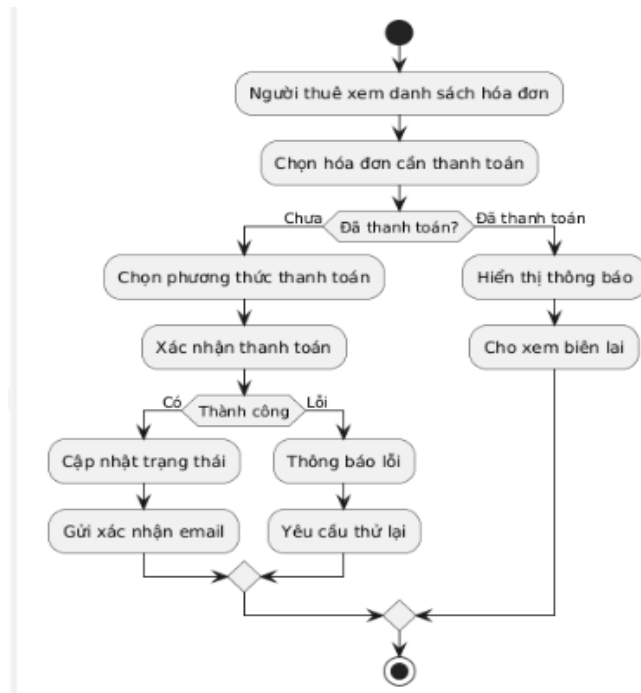
Hình 3.20 Sơ đồ hoạt động chức năng báo cáo và thống kê.

- Sơ đồ hoạt động chức năng quản lý căn hộ:



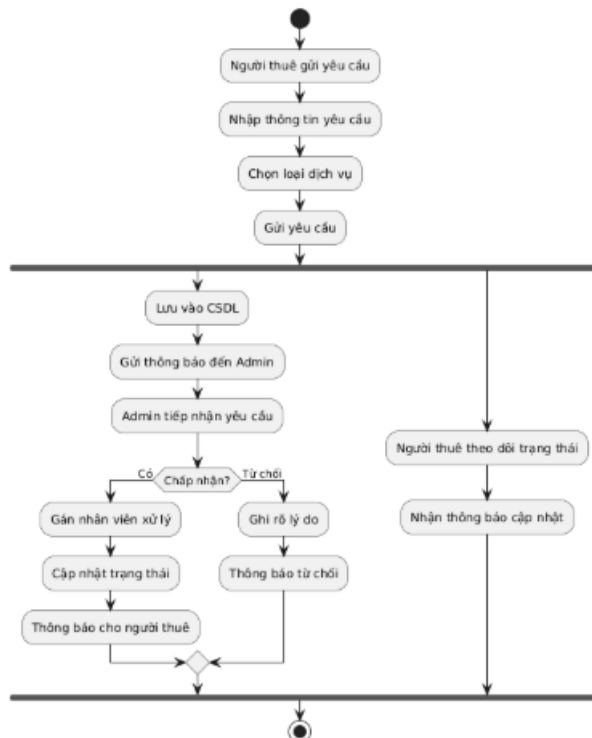
Hình 3.21 Sơ đồ hoạt động chức năng quản lý căn hộ.

- Sơ đồ hoạt động chức năng thanh toán:



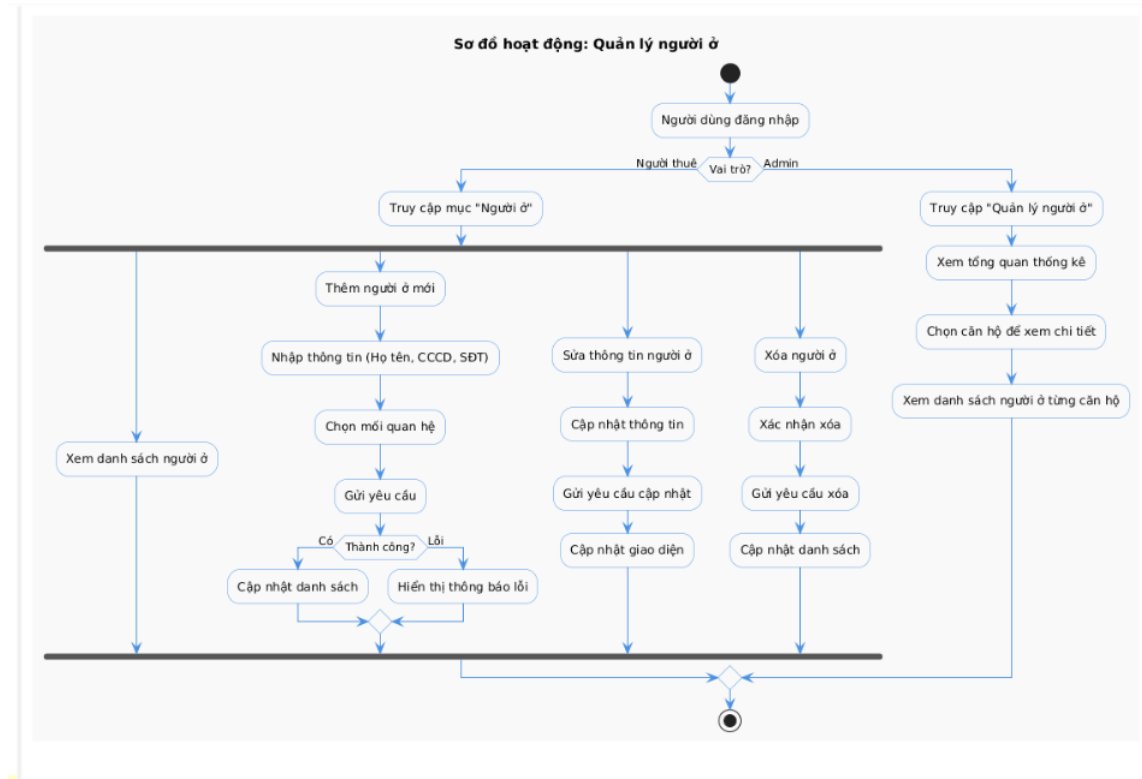
Hình 3.22 Sơ đồ hoạt động chức năng thanh toán.

- Sơ đồ hoạt động chức năng xử lý yêu cầu dịch vụ:



Hình 3.23 Sơ đồ hoạt động chức năng xử lý yêu cầu dịch vụ.

- Sơ đồ hoạt động chức năng đăng nhập:



Hình 3.24 Sơ đồ hoạt động chức năng đăng nhập.

CHƯƠNG 4: CÀI ĐẶT VÀ PHÁT TRIỂN WEBSITE

4.1. Môi trường phát triển và cài đặt

4.1.1. Phần cứng và phần mềm

Hệ thống được phát triển và triển khai trên các hệ điều hành phổ biến hiện nay như Windows 10, Windows 11, macOS hoặc Linux. Trình soạn thảo mã nguồn được sử dụng trong quá trình phát triển là Visual Studio Code do có giao diện thân thiện, hỗ trợ nhiều tiện ích mở rộng cho cả frontend và backend. Môi trường chạy của hệ thống sử dụng Node.js phiên bản từ 14.x trở lên kết hợp với trình quản lý gói npm hoặc yarn để cài đặt và quản lý các thư viện. Cơ sở dữ liệu được sử dụng là MongoDB phiên bản từ 4.4 trở lên, đáp ứng tốt nhu cầu lưu trữ dữ liệu của hệ thống. Công cụ Postman hoặc Insomnia được sử dụng để kiểm thử các API trong quá trình phát triển. Ngoài ra, hệ thống được kiểm tra và vận hành trên các trình duyệt web phổ biến như Google Chrome, Mozilla Firefox và Microsoft Edge phiên bản mới nhất nhằm đảm bảo khả năng tương thích.

4.1.2. Các thư viện và công nghệ chính

Ở phía frontend, hệ thống sử dụng React làm thư viện lõi để xây dựng giao diện người dùng theo hướng hiện đại và linh hoạt. React Router DOM được sử dụng để quản lý điều hướng giữa các trang trong mô hình ứng dụng SPA. Axios được sử dụng để thực hiện các yêu cầu HTTP từ frontend đến backend. React Context API có vai trò quản lý trạng thái toàn cục trong toàn bộ ứng dụng. Bên cạnh đó, hệ thống còn sử dụng các thư viện hỗ trợ giao diện và trải nghiệm người dùng như React Icons để hiển thị biểu tượng, React DatePicker để xử lý chọn ngày tháng, React Toastify để hiển thị thông báo và React Slick để xây dựng các hiệu ứng trình chiếu hình ảnh.

Ở phía backend, hệ thống sử dụng Express.js làm framework chính để xây dựng API trên nền tảng Node.js. Mongoose được sử dụng như một thư viện trung gian giúp kết nối và tương tác với cơ sở dữ liệu MongoDB. JSON Web Token được sử dụng để thực hiện xác thực người dùng, trong khi Bcryptjs được sử dụng để mã hóa mật khẩu nhằm đảm bảo an toàn thông tin. Multer là middleware hỗ trợ xử lý upload file. CORS được sử dụng để cho phép chia sẻ tài nguyên giữa các domain khác nhau. Dotenv giúp quản lý các biến môi trường trong quá trình phát triển. Express Validator được sử dụng để kiểm tra và xác thực dữ liệu đầu vào. Nodemon được sử dụng trong môi trường phát triển để tự động khởi động lại server khi có sự thay đổi trong mã nguồn.

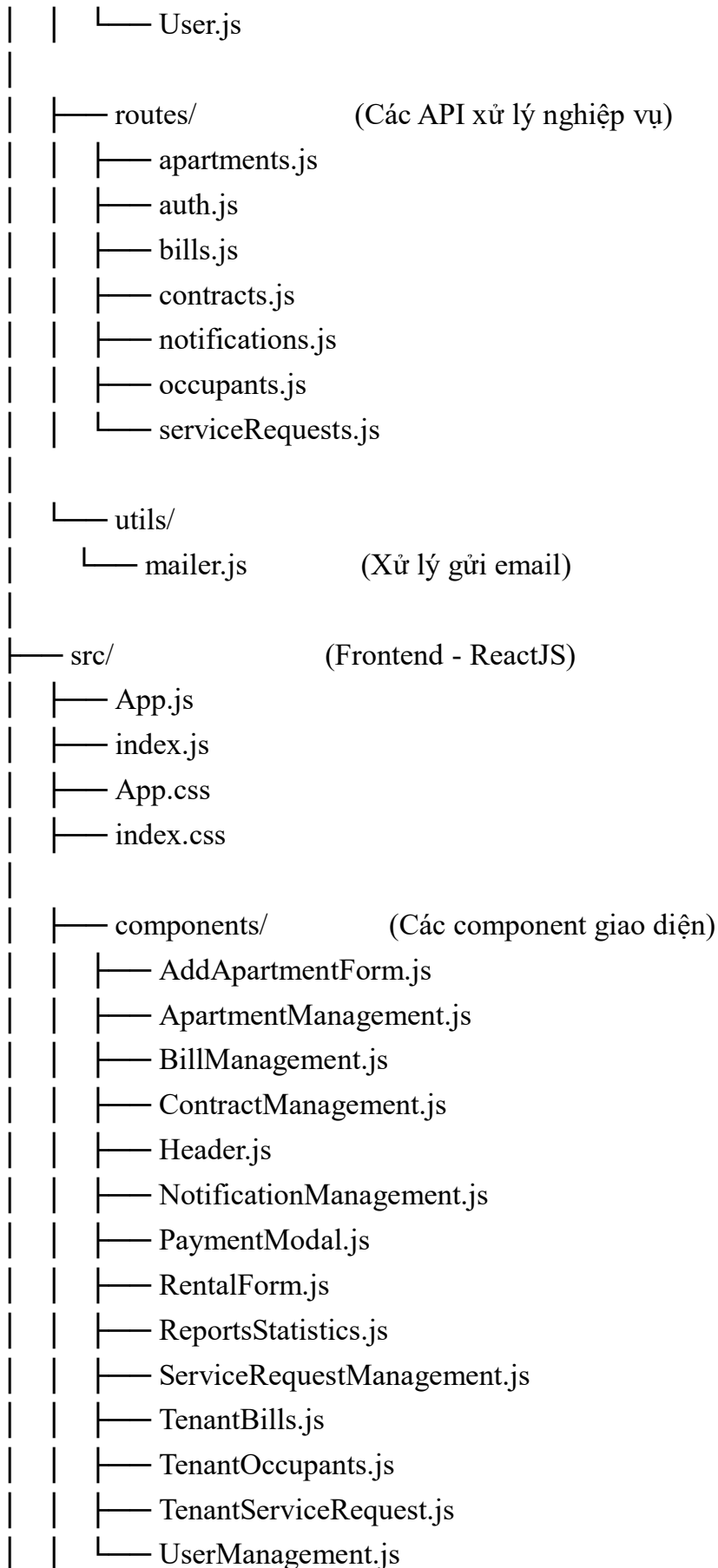
4.1.2. Cài đặt môi trường và phát triển

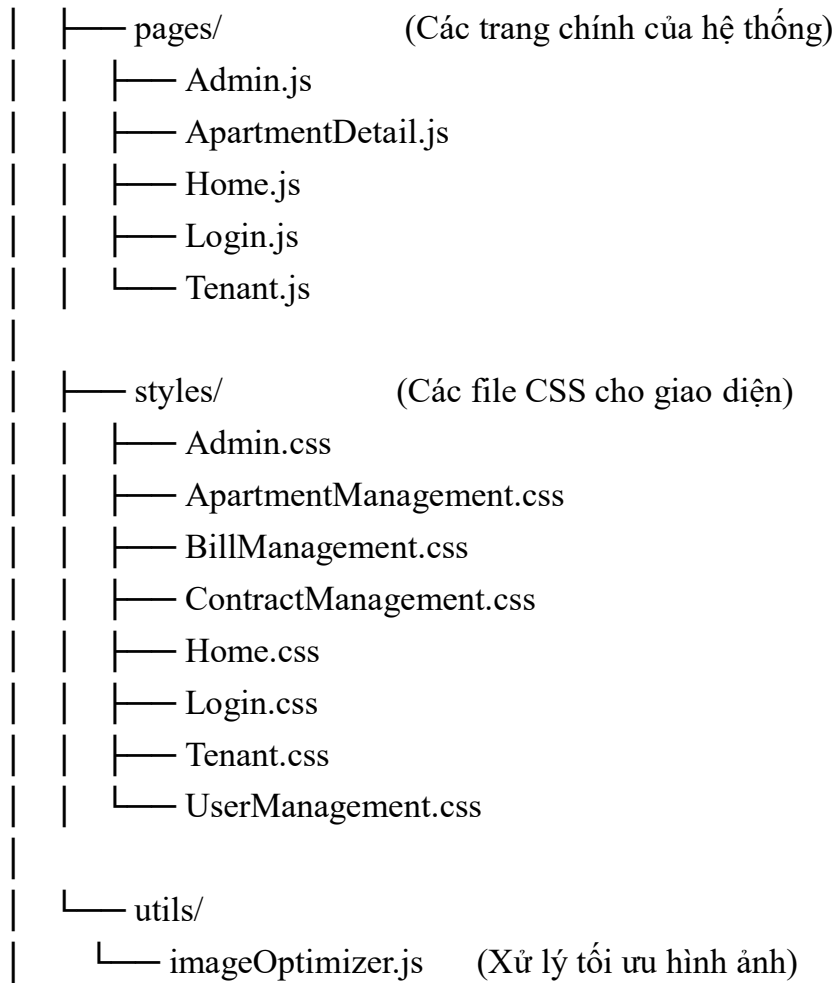
Quá trình cài đặt môi trường phát triển được thực hiện bằng cách cài đặt Node.js phiên bản LTS để đảm bảo tính ổn định cho hệ thống. Tiếp theo, MongoDB Community Server được cài đặt làm cơ sở dữ liệu chính và MongoDB Compass được sử dụng để hỗ trợ quản lý dữ liệu trực quan. Dự án được tải về từ kho mã nguồn và tiến hành cài đặt các thư viện cho cả backend và frontend bằng npm. Hệ thống sau đó được cấu hình thông qua file .env với các thông tin cần thiết như cổng server, chuỗi kết nối MongoDB và khóa bảo mật JWT. Cuối cùng, ứng dụng được khởi động và truy cập thông qua địa chỉ localhost cho cả giao diện người dùng và API.

4.2 Cấu trúc thư mục code

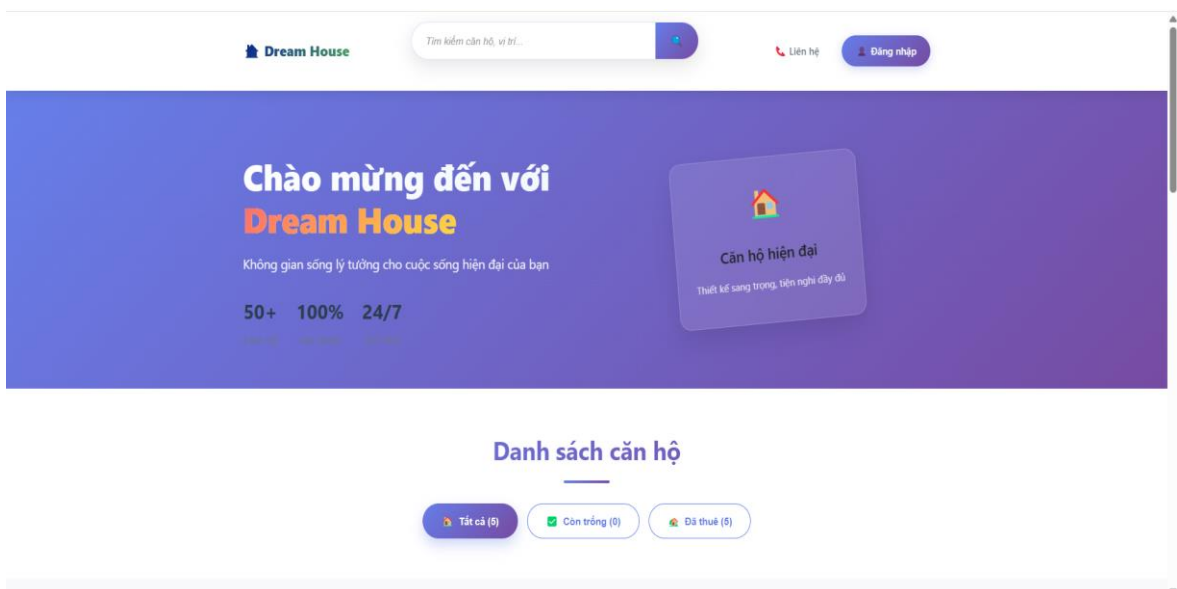
chungcumini/

```
|— .gitignore
|— package.json
|— package-lock.json
|— public/
|   |— index.html
|   |— favicon.ico
|   |— manifest.json
|   |— robots.txt
|
|— server/                                (Backend - Node.js & Express)
|   |— .env                              (Biến môi trường)
|   |— server.js                         (File khởi động server)
|   |— seed.js                          (Dữ liệu mẫu)
|   |— package.json
|
|   |— middleware/
|       |— auth.js                      (Xác thực người dùng)
|
|   |— models/                          (Các mô hình dữ liệu MongoDB)
|       |— Apartment.js
|       |— Bill.js
|       |— Contract.js
|       |— Notification.js
|       |— Occupant.js
|       |— ServiceRequest.js
```

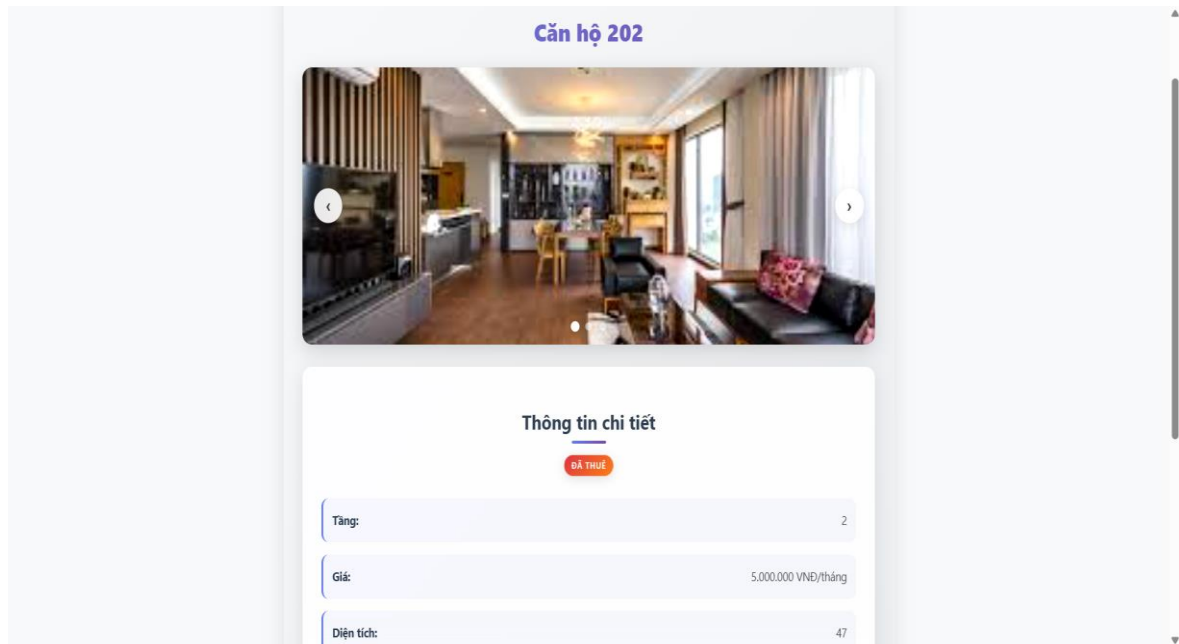




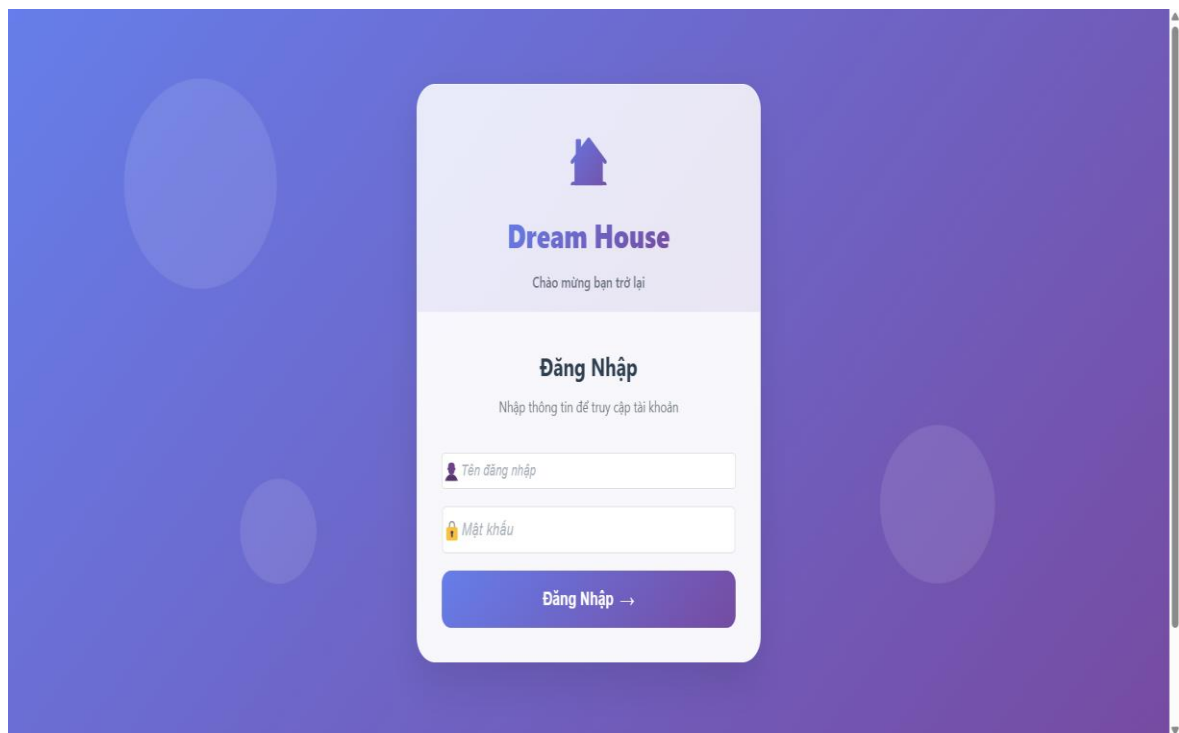
4.3. Kết quả đạt được



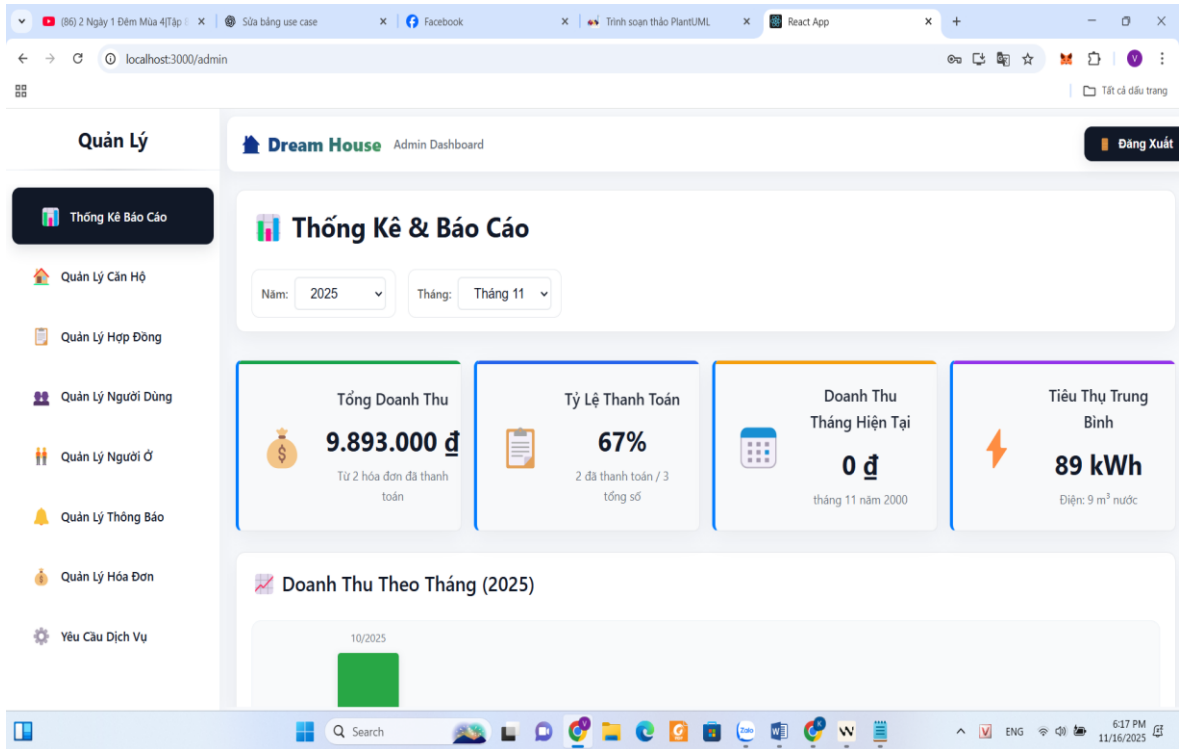
Hình 4.1 Giao diện trang home.



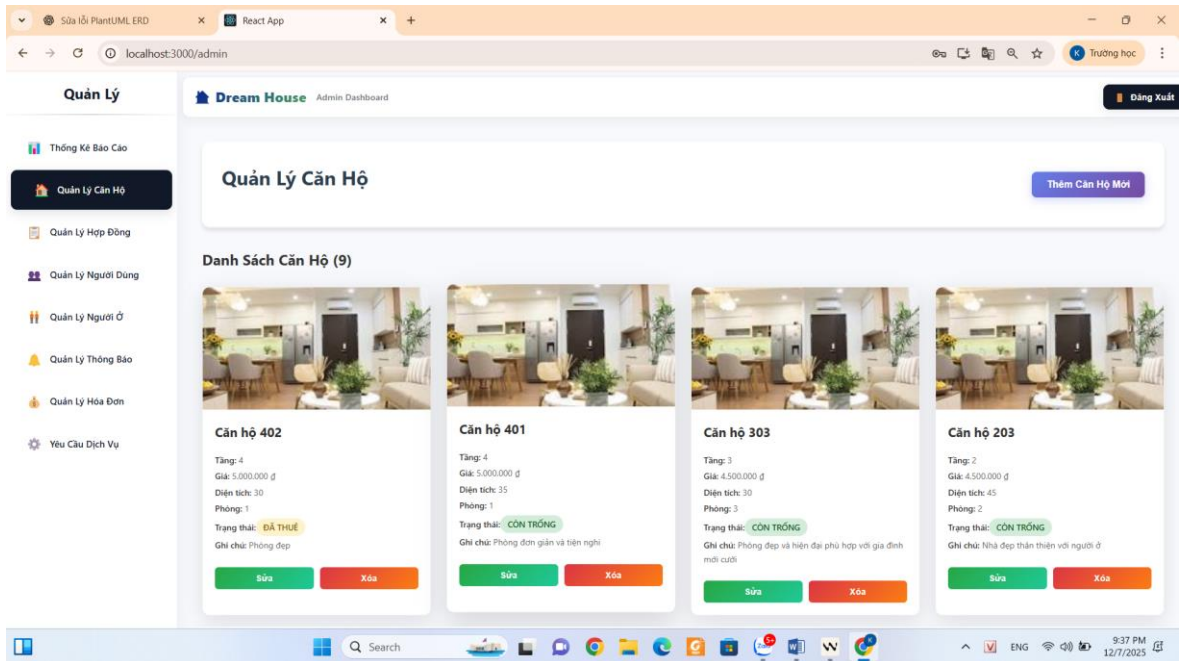
Hình 4.2 Giao diện xem chi tiết căn hộ.



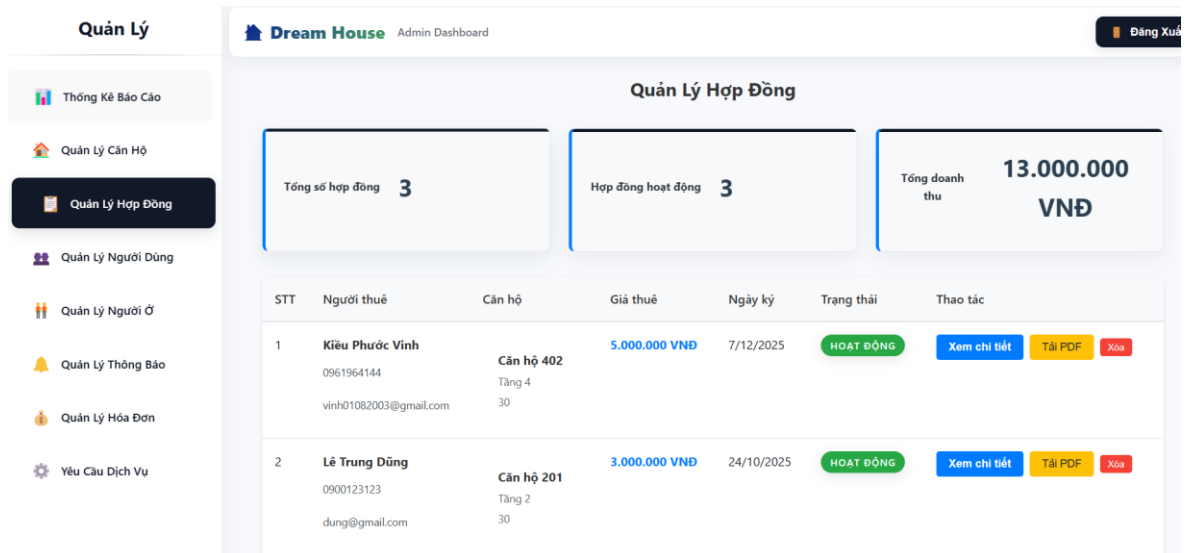
Hình 4.3 Màn hình đăng nhập.



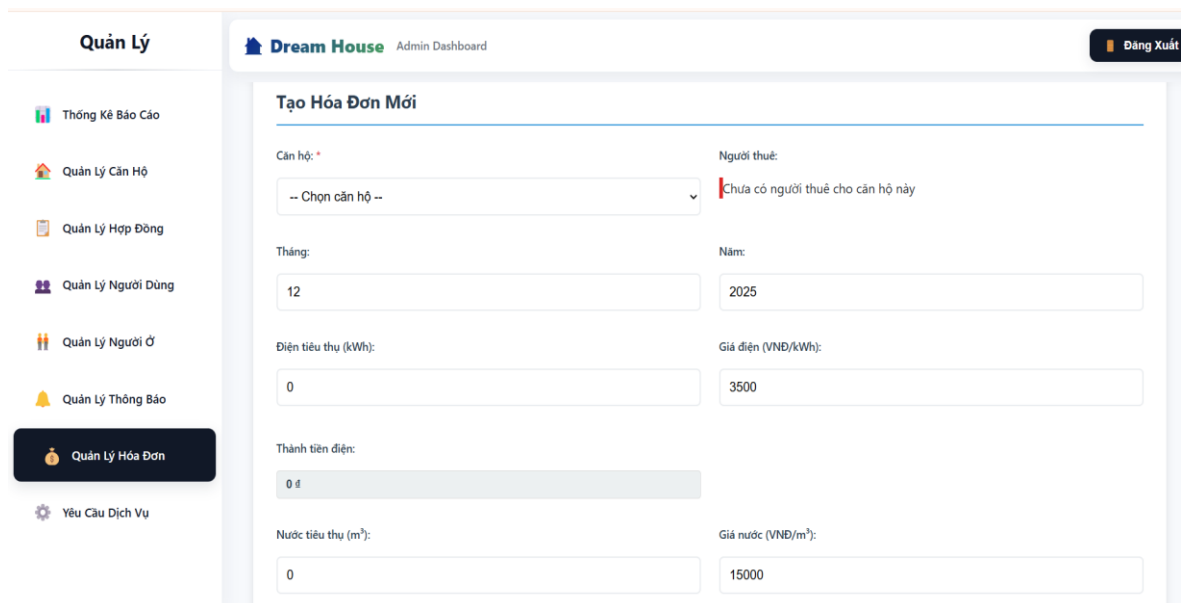
Hình 4.4 Giao diện trang admin (chủ chung cư).



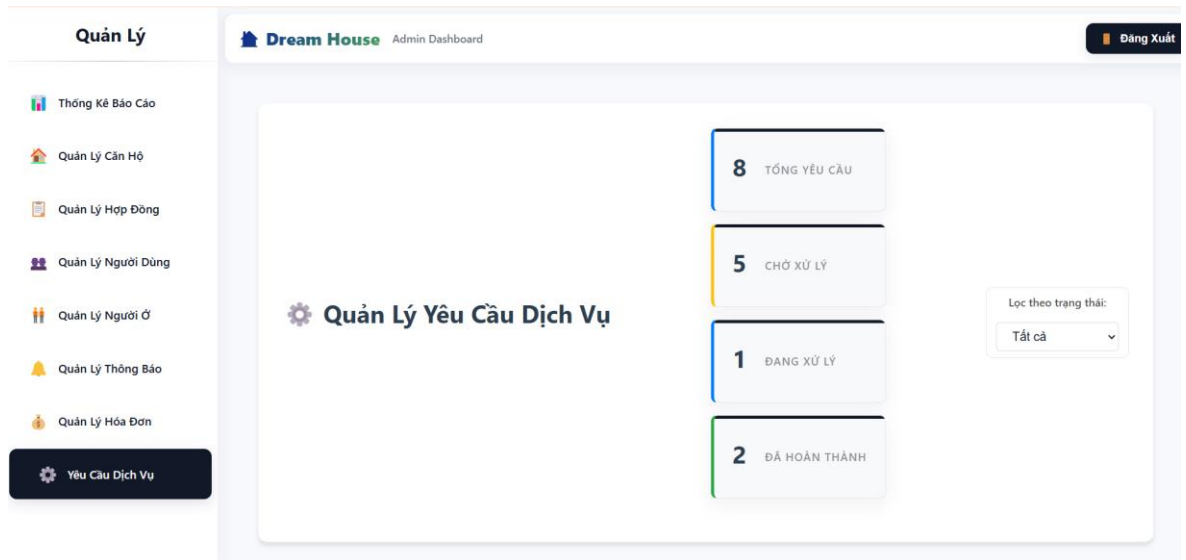
Hình 4.5 Giao diện chức năng quản lý căn hộ.



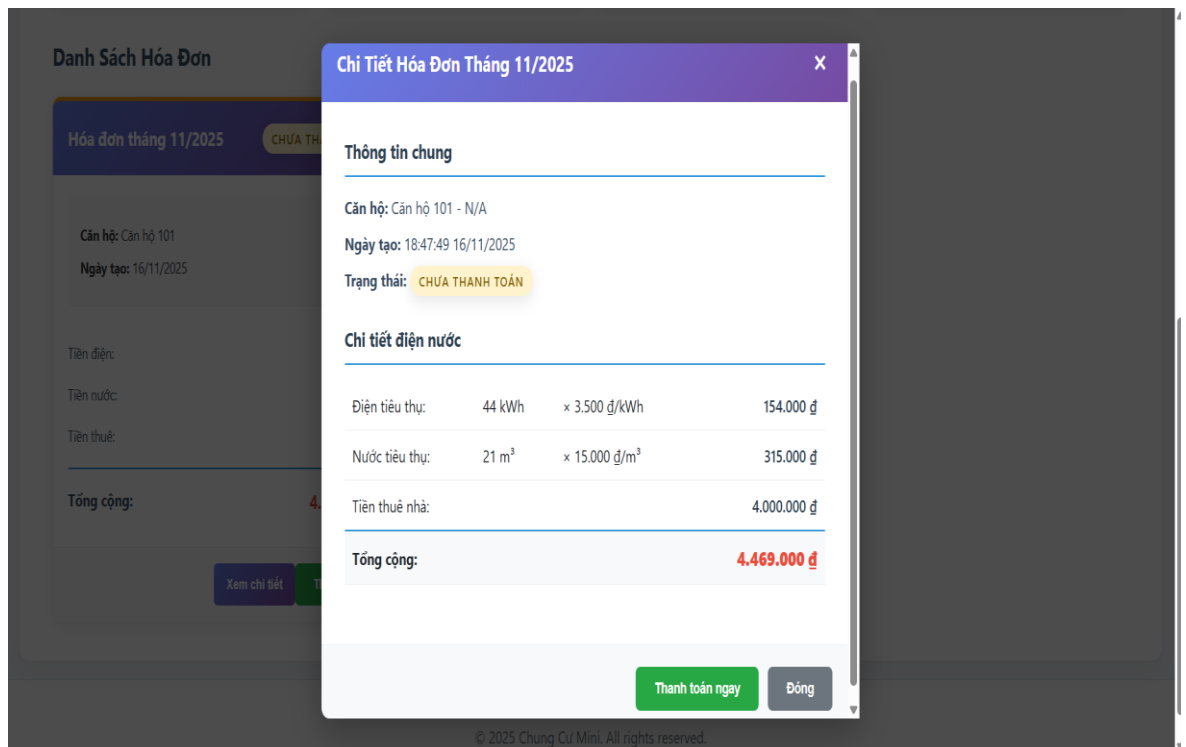
Hình 4.6 Giao diện chức năng quản lý hợp đồng.



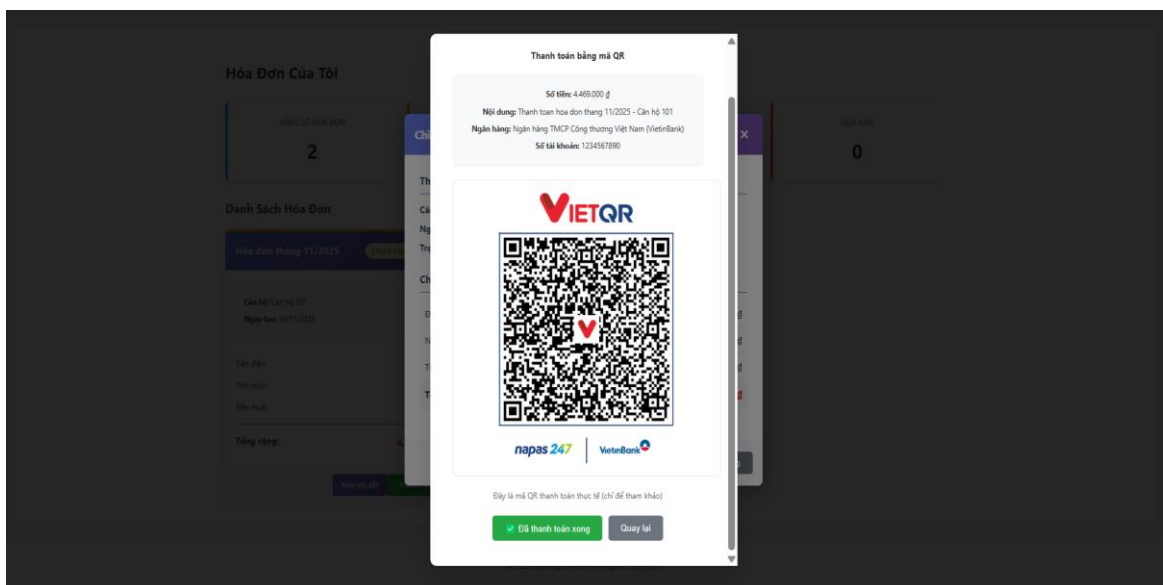
Hình 4.7 Giao diện chức năng quản lý hóa đơn.



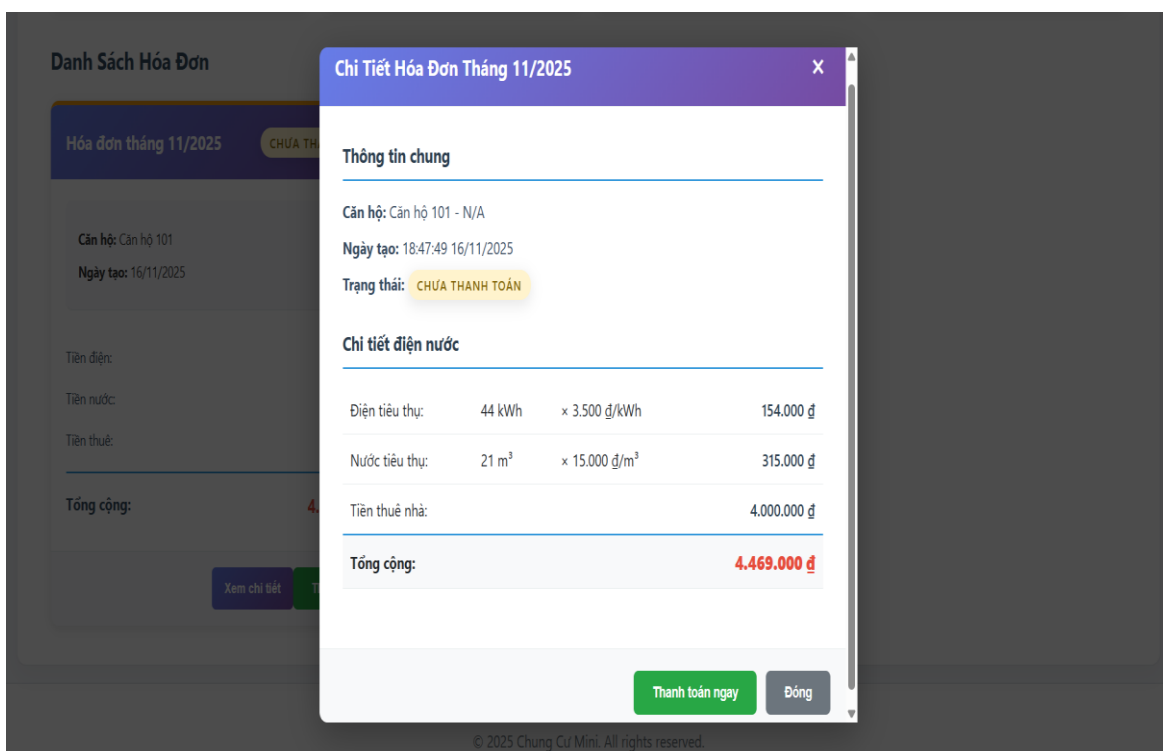
Hình 4.8 Giao diện chức năng Quản lý yêu cầu dịch vụ.



Hình 4.9 Giao diện chức năng xem chi tiết hóa đơn.



Hình 4.10 Giao diện chức năng thanh toán chuyển khoản.



Hình 4.11 Giao diện chức năng xem chi tiết hóa đơn.

4.4. Kiểm thử hệ thống

4.4.1. Kế hoạch kiểm thử

Mục tiêu kiểm thử:

- Đảm bảo tất cả các chức năng hoạt động đúng theo yêu cầu
- Phát hiện và sửa lỗi trước khi triển khai
- Kiểm tra tính tương thích trên nhiều trình duyệt
- Đánh giá hiệu năng và trải nghiệm người dùng

Phương pháp kiểm thử:

- Unit Testing: Kiểm thử từng function/component độc lập
- Integration Testing: Kiểm thử tương tác giữa Frontend - Backend - Database
- System Testing: Kiểm thử toàn bộ luồng nghiệp vụ End-to-End
- User Acceptance Testing (UAT): Kiểm thử với người dùng thực tế

Công cụ kiểm thử:

- Manual Testing (kiểm thử thủ công)
- Chrome DevTools (kiểm tra lỗi, network, performance)
- Postman (kiểm thử API)
- MongoDB Compass (kiểm tra dữ liệu)

4.4.2. Kết quả kiểm thử

Bảng 4.1 Test case chức năng đăng nhập

ID	Tên Test Case	Mô tả	Bước thực hiện	Kết quả mong đợi	Kết quả thực tế	Trạng thái
TC 01	Đăng nhập thành công	Kiểm tra đăng nhập với tài khoản hợp lệ	1. Truy cập trang đăng nhập 2. Nhập username đúng 3. Nhập password đúng 4. Nhấn nút Đăng nhập	Đăng nhập thành công, chuyển vào trang chính	Đúng như mong đợi	Pass

TC 02	Sai mật khẩu	Kiểm tra đăng nhập khi nhập sai mật khẩu	1. Truy cập trang đăng nhập 2. Nhập username đúng 3. Nhập password sai 4. Nhấn Đăng nhập	Hiện thị thông báo sai mật khẩu	Đúng như mong đợi	Pass
TC 03	Tài khoản không tồn tại	Kiểm tra đăng nhập với tài khoản không tồn tại	1. Truy cập trang đăng nhập 2. Nhập username không tồn tại 3. Nhấn Đăng nhập	Hiện thị thông báo tài khoản không tồn tại	Đúng như mong đợi	Pass
TC 04	Bỏ trống dữ liệu	Kiểm tra khi không nhập thông tin đăng nhập	1. Truy cập trang đăng nhập 2. Không nhập username và password 3. Nhấn Đăng nhập	Hiện thị thông báo yêu cầu nhập dữ liệu	Đúng như mong đợi	Pass
TC 05	Sai định dạng username	Kiểm tra username có ký tự không hợp lệ	1. Truy cập trang đăng nhập 2. Nhập username có ký tự đặc biệt 3. Nhấn Đăng nhập	Hiện thị thông báo dữ liệu không hợp lệ	Đúng như mong đợi	Pass

KẾT LUẬN

Trên đây là toàn bộ nội dung báo cáo đồ án tốt nghiệp. Trong quá trình nghiên cứu và xây dựng hệ thống, em đã hoàn thành đề tài “Xây dựng website quản lý chung cư mini” theo đúng yêu cầu đề ra. Hệ thống đã ứng dụng thành công các công nghệ web hiện đại như ReactJS ở phía giao diện người dùng và Node.js ở phía máy chủ, kết hợp MongoDB để quản lý dữ liệu, giúp website có khả năng phản hồi nhanh, giao diện trực quan, dễ sử dụng và tương thích với nhiều thiết bị.

Cụ thể, hệ thống cho phép chủ chung cư quản lý căn hộ, cư dân và các hóa đơn tiện ích, đồng thời cư dân có thể xem thông tin căn hộ, tra cứu hóa đơn, gửi yêu cầu bảo trì hoặc liên hệ quản lý. Admin có thể quản lý toàn bộ người dùng, căn hộ, hóa đơn và đảm bảo hệ thống vận hành ổn định.

Tuy nhiên, bên cạnh những kết quả đạt được, hệ thống vẫn còn một số hạn chế như chưa tích hợp thanh toán trực tuyến, chưa có tính năng thông báo tự động qua email/SMS, và chưa tối ưu được trải nghiệm giao diện trên mobile. Trong tương lai, hệ thống có thể được nâng cấp với các tính năng như tích hợp cổng thanh toán điện tử, xây dựng ứng dụng mobile, nâng cao bảo mật và tự động hóa thông báo cho cư dân và chủ chung cư.

I KẾT QUẢ ĐẠT ĐƯỢC

Hệ thống quản lý chung cư mini được xây dựng thành công, vận hành mượt mà và đáp ứng đầy đủ các yêu cầu đề ra. Về mặt công nghệ, đồ án đã áp dụng hiệu quả ReactJS cho giao diện người dùng, Node.js/Express cho phần backend và MongoDB để quản lý dữ liệu, giúp hệ thống phản hồi nhanh, ổn định và dễ mở rộng. Người dùng, bao gồm cư dân và chủ chung cư, có thể thực hiện các thao tác tra cứu thông tin căn hộ, xem và thanh toán hóa đơn, gửi yêu cầu bảo trì hoặc phản hồi trực tiếp tới ban quản lý. Admin hệ thống có thể quản lý toàn bộ thông tin cư dân, căn hộ, hóa đơn và giám sát hoạt động của hệ thống. Qua quá trình thực hiện, đồ án giúp người thực hiện nắm vững quy trình phát triển ứng dụng web theo mô hình frontend – backend tách biệt, xây dựng API RESTful, thiết kế cơ sở dữ liệu linh hoạt và áp dụng các kiến thức lập trình web vào thực tiễn.

II HẠN CHẾ

Mặc dù đạt được nhiều kết quả tích cực, hệ thống vẫn còn một số hạn chế cần khắc phục. Hiện tại, hiệu suất của hệ thống có thể chưa tối ưu khi xử lý đồng thời

nhiều người dùng, dẫn đến độ trễ trong các thao tác tra cứu hóa đơn hoặc gửi yêu cầu bảo trì. Về bảo mật, hệ thống cần bổ sung các lớp xác thực nâng cao, cơ chế mã hóa dữ liệu nhạy cảm và biện pháp chống tấn công như XSS hay CSRF để đảm bảo an toàn thông tin. Ngoài ra, các tính năng mở rộng như thông báo tự động qua email/SMS, tích hợp thanh toán trực tuyến và gợi ý lịch bảo trì thông minh vẫn chưa được triển khai, làm hạn chế trải nghiệm và tiện ích cho cư dân cũng như chủ chung cư.

III HƯỚNG PHÁT TRIỂN

Để hoàn thiện và nâng cao chất lượng hệ thống trong tương lai, có thể triển khai một số hướng phát triển như sau tối ưu hiệu suất bằng cách nghiên cứu và cải thiện. Trong tương lai, hệ thống có thể được nâng cấp để tăng cường hiệu suất, bảo mật và mở rộng tính năng. Cụ thể, việc tối ưu tốc độ phản hồi API, cải thiện truy vấn cơ sở dữ liệu và áp dụng cơ chế cache sẽ giúp hệ thống vận hành ổn định khi có nhiều người dùng truy cập đồng thời. Về bảo mật, có thể triển khai xác thực hai lớp (2FA), mã hóa dữ liệu nhạy cảm và kiểm soát truy cập theo vai trò người dùng (RBAC) để bảo vệ thông tin. Về mặt tính năng, hệ thống có thể tích hợp thanh toán trực tuyến cho các loại hóa đơn, thiết lập thông báo tự động qua email hoặc SMS, phát triển ứng dụng mobile cho Android và iOS, cũng như áp dụng các thuật toán thông minh để gợi ý lịch bảo trì, nhắc nhở thanh toán hoặc đề xuất dịch vụ phù hợp cho cư dân, từ đó nâng cao trải nghiệm người dùng và giá trị quản lý chung cư.

TÀI LIỆU THAM KHẢO

1. "ExpressJS" [Online]. Available: <https://expressjs.com/>.
2. "MongoDB" [Online]. Available: <https://www.mongodb.com>.
3. "NodeJS" [Online]. Available: <https://nodejs.org/en/docs/>.
4. Tài liệu JavaScript: <https://javascript.info/37>
5. Tài liệu React: <https://reactjs.org/docs/getting-started.html>
6. StackOverFlow: <https://stackoverflow.com/>
7. Youtube: (1906) Hỏi Dân IT - YouTub

