

**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 ỨNG DỤNG ĐĂNG BÀI, TÌM KIẾM  
PHÒNG TRỌ SỬ DỤNG JAVA, EXPRESSJS VÀ  
MONGODB**

Người hướng dẫn: **TS. LÊ VĂN MINH**  
Sinh viên thực hiện: **NGUYỄN VĂN DUY NHẤT**  
Mã số sinh viên: 2151220205  
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 ỨNG DỤNG ĐĂNG BÀI, TÌM KIẾM  
PHÒNG TRỢ SỬ DỤNG JAVA, EXPRESSJS VÀ  
MONGODB**

Người hướng dẫn: **TS. LÊ VĂN MINH**  
Sinh viên thực hiện: **NGUYỄN VĂN DUY NHẤT**  
Mã số sinh viên: 215122025  
Lớp: **21CT4**

**Đà Nẵng, 12/2025**





## TÓM TẮT

Tên đề tài: Xây dựng ứng dụng đăng bài ,tìm kiếm phòng trọ sử dụng Java ,ExpressJS và MongoDB

Sinh viên thực hiện: Nguyễn Văn Duy Nhất

Mã số SV:2151220205 Lớp: 21CT4

Đề tài tập trung nghiên cứu và ứng dụng các công nghệ hiện đại gồm Java làm backend, ExpressJS làm backend, cùng cơ sở dữ liệu MongoDB để xây dựng một nền tảng trực tuyến hỗ trợ sinh viên và chủ trọ đăng tin, tìm kiếm và quản lý thông tin phòng trọ một cách nhanh chóng, chính xác và tiện lợi. Hệ thống được tối ưu hóa giao diện người dùng (UX/UI) responsive trên cả máy tính và điện thoại, đảm bảo hiệu suất cao, khả năng mở rộng tốt, bảo mật thông tin cá nhân, đồng thời dễ dàng tích hợp thêm các tính năng trong tương lai như chat realtime, xác minh danh tính chủ trọ, hiển thị vị trí phòng trọ trên bản đồ Google Maps, hệ thống đánh giá – nhận xét và thanh toán tiền cọc trực tuyến. Nghiên cứu cũng đề xuất các giải pháp triển khai, bảo trì và mở rộng hệ thống một cách bền vững trong dài hạn.

Ứng dụng đăng tin và tìm kiếm phòng trọ sẽ là giải pháp hiệu quả kết nối trực tiếp giữa sinh viên đang tìm chỗ ở và các chủ trọ, giúp tiết kiệm thời gian, tăng tính minh bạch của thông tin, giảm thiểu rủi ro lừa đảo, từ đó mang lại trải nghiệm tiện lợi và đáng tin cậy cho cộng đồng sinh viên.

## 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, đặc biệt là các thầy cô giáo đã tận tình truyền đạt kiến thức trong những năm học vừa qua. Với vốn kiến thức được tiếp thu trong quá trình học không chỉ là nền tảng cho quá trình làm đồ án tốt nghiệp mà còn là hành trang quý báu để em bước vào đời một cách vững chắc và tự tin.

Em cũng xin gửi lời cảm ơn chân thành và sự tri ân sâu sắc của thầy Lê Văn Minh trong suốt thời gian vừa qua đã nhiệt tình chỉ dạy, giúp đỡ em để có thể hoàn thành tốt đồ án tốt nghiệp.

Trong quá trình làm bài đồ án tốt nghiệp, do trình độ lý luận cũng như kinh nghiệm thực tiễn còn hạn chế nên không thể tránh khỏi những thiếu sót, em rất mong nhận được ý kiến đóng góp của quý báu của các thầy cô để em có thể học hỏi thêm được nhiều kinh nghiệm, kiến thức.

***Em xin trân trọng cảm ơn!***

## **CAM ĐOAN**

Em xin cam đoan kết quả đạt được trong đồ án tốt nghiệp là sản phẩm của riêng cá nhân dưới sự hướng dẫn của TS.Lê Văn Minh , không sao chép lại của người khác.

Trong toàn bộ nội dung của đồ án, những điều được trình bày hoặc là của bản thân hoặc là được tổng hợp từ nhiều nguồn tài liệu. Tất cả các tài liệu tham khảo đều có xuất xứ rõ ràng và được trích dẫn hợp pháp. Em xin hoàn toàn chịu trách nhiệm và chịu mọi hình thức kỷ luật theo quy định cho lời cam đoan của mình.

Sinh viên thực hiện

Nguyễn Văn Duy Nhất

# MỤC LỤC

|                                                        |           |
|--------------------------------------------------------|-----------|
| <b>TÓM TẮT .....</b>                                   | <b>5</b>  |
| <b>LỜI NÓI ĐẦU .....</b>                               | <b>i</b>  |
| <b>CAM ĐOAN .....</b>                                  | <b>ii</b> |
| <b>DANH SÁCH BẢNG .....</b>                            | <b>v</b>  |
| <b>DANH SÁCH HÌNH ẢNH .....</b>                        | <b>vi</b> |
| <b>MỞ ĐẦU .....</b>                                    | <b>1</b>  |
| <b>1. Lý do chọn đề tài .....</b>                      | <b>1</b>  |
| <b>2. Lịch sử nghiên cứu .....</b>                     | <b>1</b>  |
| <b>3. Mục tiêu đề tài .....</b>                        | <b>2</b>  |
| <b>4. Đối tượng , phạm vi nghiên cứu .....</b>         | <b>3</b>  |
| <b>Chương 1: CƠ SỞ LÝ THUYẾT .....</b>                 | <b>5</b>  |
| <b>1.1.Giới thiệu tổng quan về ngôn ngữ Java .....</b> | <b>5</b>  |
| <b>1.2.Giới thiệu về NodeJs .....</b>                  | <b>7</b>  |
| <b>1.3.Giới thiệu về ExpressJs .....</b>               | <b>9</b>  |
| <b>1.4.Giới thiệu về MongoDB .....</b>                 | <b>10</b> |
| <b>Chương 2: PHÂN TÍCH, THIẾT KẾ HỆ THỐNG .....</b>    | <b>13</b> |
| <b>2.1.Xác định yêu cầu .....</b>                      | <b>13</b> |
| <b>2.2.Phân tích yêu cầu .....</b>                     | <b>13</b> |
| <b>2.2.1.Chức năng đăng kí .....</b>                   | <b>13</b> |
| <b>2.2.2.Chức năng đăng nhập .....</b>                 | <b>14</b> |
| <b>2.2.3.Chức năng tìm kiếm .....</b>                  | <b>14</b> |
| <b>2.2.4.Chức năng đăng bài .....</b>                  | <b>14</b> |
| <b>2.2.5.Chức năng quản lý đặt phòng .....</b>         | <b>15</b> |
| <b>2.2.6.Chức năng thêm sửa xóa bài đăng .....</b>     | <b>15</b> |
| <b>2.2.7.Chức năng đặt phòng .....</b>                 | <b>15</b> |

|                                                        |           |
|--------------------------------------------------------|-----------|
| 2.2.8. Quản lý tài khoản người dùng .....              | 15        |
| 2.3. Mô hình hóa yêu cầu.....                          | 16        |
| 2.3.1. Chức năng đăng nhập.....                        | 16        |
| 2.3.2. Chức năng đăng kí.....                          | 18        |
| 2.3.3. Chức năng tìm kiếm .....                        | 20        |
| 2.3.4. Chức năng đăng bài.....                         | 21        |
| 2.3.5. Chức năng quản lý đặt phòng .....               | 24        |
| 2.3.6. Chức năng quản lý bài đăng .....                | 27        |
| 2.3.7. Chức năng xem thống kê .....                    | 30        |
| 2.3.8. Chức năng quản lý tài khoản .....               | 32        |
| 2.4. Thiết kế cơ sở dữ liệu.....                       | 35        |
| 2.4.1. Mô hình thực thể liên kết.....                  | 35        |
| 2.4.2. Mô hình quan hệ.....                            | 36        |
| 2.5. Danh mục các bảng.....                            | 36        |
| 2.6. Thiết kế giao diện.....                           | 42        |
| 2.6.1. Giao diện đăng ký.....                          | 42        |
| 2.6.2. Giao diện người thuê .....                      | 43        |
| 2.6.3. Giao diện người cho thuê .....                  | 46        |
| 2.6.4. Giao diện Admin.....                            | 51        |
| <b>Chương 3: CÀI ĐẶT VÀ PHÁT TRIỂN ỨNG DỤNG .....</b>  | <b>54</b> |
| 3.1. Cấu trúc thư mục code .....                       | 54        |
| 3.1.1. Cấu trúc thư mục code Back End .....            | 54        |
| 3.1.2. Cấu trúc thư mục code FrontEnd .....            | 55        |
| 3.2. Chức năng cần phát triển sắp tới .....            | 56        |
| 3.2.1. Chức năng đặt lịch khi thanh toán tiền cọc..... | 56        |
| 3.2.2. Chức năng đánh giá bài đăng trọ.....            | 57        |
| <b>KẾT LUẬN .....</b>                                  | <b>60</b> |
| <b>TÀI LIỆU THAM KHẢO.....</b>                         | <b>62</b> |

## DANH SÁCH BẢNG

|                                                       |    |
|-------------------------------------------------------|----|
| Bảng 2.1.Mô tả Usecase đăng nhập.....                 | 17 |
| Bảng 2.2 Dòng chính Use Case đăng nhập .....          | 17 |
| Bảng 2.3 Mô tả Use Case đăng kí .....                 | 18 |
| Bảng 2.4 Dòng chính Use Case đăng nhập .....          | 18 |
| Bảng 2.5 Mô tả Use Case tìm kiếm.....                 | 20 |
| Bảng 2.6 Dòng chính Use Case tìm kiếm.....            | 20 |
| Bảng 2.7 Mô tả Use Case đăng bài.....                 | 22 |
| Bảng 2.8 Dòng chính Use Case đăng bài .....           | 22 |
| Bảng 2.9 Mô tả Use Case quản lý đặt phòng .....       | 24 |
| Bảng 2.10 Dòng chính Use Case quản lý đặt phòng ..... | 26 |
| Bảng 2.11 Mô tả Use Case quản lý bài đăng.....        | 27 |
| Bảng 2.12 Dòng chính Use Case quản lý bài đăng .....  | 28 |
| Bảng 2.13 Mô tả Use Case xem thống kê .....           | 30 |
| Bảng 2.14 Dòng chính Use Case xem thống kê .....      | 31 |
| Bảng 2.15 Mô tả Use Case quản lý tài khoản .....      | 32 |
| Bảng 2.17 Mô tả các bảng cơ sở dữ liệu .....          | 36 |
| Bảng 2.18 Mô tả bảng User.....                        | 37 |
| Bảng 2.19 Mô tả bảng Userrole.....                    | 38 |
| Bảng 2.20 Mô tả bảng Room.....                        | 38 |
| Bảng 2.21 Mô tả bảng Booking.....                     | 38 |
| Bảng 2.22 Mô tả bảng Payment .....                    | 39 |
| Bảng 2.23 Mô tả bảng Invoice .....                    | 40 |
| Bảng 2.25 Mô tả bảng Notification .....               | 40 |

## DANH SÁCH HÌNH ẢNH

|                                                            |    |
|------------------------------------------------------------|----|
| Hình 2.1 Biểu đồ Usecase tổng quát .....                   | 16 |
| Hình 2.2. Biểu đồ Usecase đăng nhập .....                  | 16 |
| Hình 2.3 Biểu đồ tuần tự đăng nhập .....                   | 17 |
| Hình 2.4. Biểu đồ Usecase đăng kí .....                    | 18 |
| Hình 2.5 Biểu đồ tuần tự đăng kí .....                     | 19 |
| Hình 2.6 Biểu đồ Use Case tìm kiếm phòng trọ .....         | 20 |
| Hình 2.7 Biểu đồ tuần tự tìm kiếm phòng trọ .....          | 21 |
| Hình 2.8 Biểu đồ Use Case đăng bài .....                   | 21 |
| Hình 2.9 Biểu đồ tuần tự đăng bài .....                    | 23 |
| Hình 2.10 Biểu đồ Use Case quản lý đặt phòng .....         | 24 |
| Hình 2.11 Biểu đồ tuần tự quản lý đặt phòng .....          | 26 |
| Hình 2.12 Biểu đồ Use Case quản lý bài đăng .....          | 27 |
| Hình 2.13 Biểu đồ tuần tự quản lý bài đăng .....           | 29 |
| Hình 2.14 Biểu đồ Use Case xem thống kê .....              | 30 |
| Hình 2.15 Biểu đồ tuần tự quản lý đặt phòng .....          | 31 |
| Hình 2.16 Biểu đồ Use Case quản lý tài khoản .....         | 32 |
| Hình 2.17 Biểu đồ tuần tự quản lý tài khoản .....          | 34 |
| Hình 2.18 Sơ đồ thực thể liên kết .....                    | 35 |
| Hình 2.19 Mô hình quan hệ .....                            | 36 |
| Hình 2.20 Màn hình Login .....                             | 42 |
| Hình 2.21 Màn hình người thuê .....                        | 43 |
| Hình 2.22 Giao diện danh sách đặt phòng .....              | 44 |
| Hình 2.23 Giao diện tìm kiếm .....                         | 45 |
| Hình 2.24 Giao diện người cho thuê .....                   | 46 |
| Hình 2.25 Giao diện đăng bài .....                         | 47 |
| Hình 2.26 Giao diện quản lý bài đăng .....                 | 48 |
| Hình 2.27 Giao diện quản lý các yêu cầu .....              | 49 |
| Hình 2.28 Giao diện quản lý các thanh toán .....           | 50 |
| Hình 2.29 Màn hình Admin .....                             | 51 |
| Hình 2.30 Giao diện quản lý người dùng .....               | 52 |
| Hình 2.30 Giao diện thống kê thanh toán của hệ thống ..... | 53 |

## MỞ ĐẦU

### 1. Lý do chọn đề tài

Trong bối cảnh đô thị hóa nhanh chóng và nhu cầu nhà ở giá rẻ ngày càng tăng cao tại Việt Nam, việc tìm kiếm và cho thuê phòng trọ đang trở thành một vấn đề thực tiễn cấp thiết đối với hàng triệu người, đặc biệt là sinh viên, người lao động trẻ và các gia đình thu nhập thấp. Tuy nhiên, các phương thức truyền thống (nhờ môi giới, dán tờ rơi) và một số nền tảng hiện có vẫn tồn tại nhiều hạn chế lớn: thông tin không được kiểm chứng, hình ảnh không trung thực, giá bị đội lên, dễ xảy ra lừa đảo đặt cọc, tính năng tìm kiếm còn sơ sài và trải nghiệm trên thiết bị di động chưa thực sự tối ưu.

Với sự phát triển mạnh mẽ của công nghệ thông tin và sự phổ biến của Internet, việc xây dựng một nền tảng trực tuyến chuyên biệt, minh bạch và tiện lợi cho việc đăng tin và tìm kiếm phòng trọ là hoàn toàn khả thi và cần thiết. Đây chính là cơ sở thực tiễn để tác giả lựa chọn đề tài nghiên cứu.

Về mặt công nghệ, Java và ExpressJS hiện là hai trong số những công nghệ backend phổ biến và mạnh mẽ nhất. Java nổi bật với độ ổn định cao, hệ sinh thái phong phú và khả năng xử lý logic nghiệp vụ phức tạp; ExpressJS lại mang đến sự linh hoạt, tốc độ xử lý vượt trội và khả năng xây dựng RESTful API nhanh chóng. Khi kết hợp với cơ sở dữ liệu NoSQL MongoDB – vốn rất phù hợp để lưu trữ dữ liệu phi cấu trúc như hình ảnh, tiện ích, vị trí địa lý của phòng trọ – sẽ tạo nên một hệ thống hiện đại, dễ mở rộng và hiệu năng cao.

Xuất phát từ nhu cầu thực tế của xã hội và mong muốn áp dụng kiến thức đã học vào giải quyết một bài toán có ý nghĩa thực tiễn, tác giả quyết định thực hiện đề tài: **“Xây dựng ứng dụng đăng bài và tìm kiếm phòng trọ sử dụng Java, ExpressJS và MongoDB”**.

### 2. Lịch sử nghiên cứu

Trong nước:

- Hiện nay tại Việt Nam đã xuất hiện một số website và ứng dụng hỗ trợ tìm kiếm phòng trọ như: Batdongsan.com.vn, Phongtro123.com, Nhà trọ 24h, Mogi.vn, Sinhvienphongtro.com, các nhóm Facebook (“Phòng trọ sinh viên Hà Nội/TP.HCM”, “Cho thuê trọ giá rẻ”...), ứng dụng Homedy, 123nhatro...

- Tuy nhiên, phần lớn các nền tảng này tồn tại những hạn chế:
- + Thông tin chưa được kiểm chứng kỹ, nhiều tin ảo, hình ảnh cũ hoặc không đúng thực tế.
- + Tính năng tìm kiếm còn đơn giản (chủ yếu theo quận/huyện, giá), thiếu bộ lọc nâng cao như khoảng cách đến trường học/công ty, tiện ích (máy lạnh, giờ giấc tự do, gần chợ...), hoặc tìm kiếm theo bản đồ.
- + Không có hoặc rất ít công cụ hỗ trợ quản lý tin đăng, lưu phòng yêu thích, so sánh phòng, chat trực tiếp với chủ trọ.
- + Giao diện trên điện thoại chưa thực sự thân thiện, tốc độ tải chậm khi có nhiều hình ảnh.

#### **Nước ngoài:**

Các nền tảng quốc tế nổi bật như **Airbnb**, **Booking.com**, **Zillow**, **99acres** (Ấn Độ), **Nestoria**, **Rightmove** (Anh)... tập trung chủ yếu vào cho thuê ngắn hạn (homestay, căn hộ du lịch) hoặc bất động sản cao cấp.

Một số ứng dụng chuyên về phòng trọ sinh viên như **Uniplaces**, **Student.com**, **Amber Student**, **HousingAnywhere** có giao diện hiện đại, bộ lọc chi tiết, hỗ trợ chat và đặt cọc online, nhưng:

- Được thiết kế chủ yếu cho thị trường châu Âu/Mỹ/Úc, không phù hợp với đặc thù phòng trọ giá rẻ tại Việt Nam (giá thấp, tiện ích cơ bản, hợp đồng miệng, không cần giấy tờ phức tạp).
- Phí dịch vụ cao, yêu cầu xác thực danh tính nghiêm ngặt, không hỗ trợ tiếng Việt đầy đủ.

### **3. Mục tiêu đề tài**

Đề tài “Xây dựng ứng dụng tìm kiếm và đăng bài cho thuê phòng trọ sử dụng Android Java, Node.js, Express.js và MongoDB” hướng đến phát triển một ứng dụng di động tiện lợi, phục vụ chủ yếu cho sinh viên, với giao diện thân thiện, dễ sử dụng và tích hợp các tính năng thông minh. Ứng dụng được xây dựng bằng Android Java cho giao diện di động, kết hợp với backend sử dụng Node.js, Express.js và cơ sở dữ liệu MongoDB, đảm bảo hiệu quả, linh hoạt và khả năng mở rộng. Mục tiêu cụ thể bao gồm: xây dựng các chức năng cơ bản như tìm kiếm phòng trọ theo giá cả, vị trí, diện tích, tiện ích hỗ trợ chủ nhà đăng bài với mô tả chi tiết, hình ảnh, thông tin liên hệ và quản lý bài đăng (tạo, chỉnh sửa, xóa); sử dụng thuật toán đơn giản; cung cấp công cụ lọc và tìm kiếm nâng cao theo tiêu chí tùy chỉnh; xây dựng backend mạnh mẽ để quản lý dữ liệu nhanh chóng và linh hoạt. Ứng dụng sử dụng tương thích với nhiều thiết bị Android. Đề tài mang lại giải pháp thực tiễn, giúp sinh viên dễ dàng tìm phòng trọ phù hợp, hỗ trợ

chủ nhà tiếp cận người thuê hiệu quả, đồng thời là tài liệu tham khảo cho việc ứng dụng công nghệ di động và backend hiện đại trong lĩnh vực bất động sản.

#### **4. Đối tượng , phạm vi nghiên cứu**

**Đối tượng nghiên cứu** của đề tài “Xây dựng ứng dụng tìm kiếm và đăng bài cho thuê phòng trọ” bao gồm hai nhóm người dùng chính, được xác định dựa trên các chức năng của ứng dụng nhằm đáp ứng nhu cầu thực tế trong việc tìm kiếm và quản lý phòng trọ tại thị trường Việt Nam, đặc biệt tại các thành phố lớn như Đà Nẵng, TP.HCM và Hà Nội. Nhóm đối tượng đầu tiên là **người thuê phòng trọ**, bao gồm sinh viên, người lao động, hoặc bất kỳ cá nhân nào cần tìm kiếm phòng trọ hoặc nhà ở giá rẻ, phù hợp với ngân sách và nhu cầu sinh hoạt. Nhóm này thường gặp khó khăn do thông tin phòng trọ phân tán trên nhiều nền tảng, thiếu minh bạch, hoặc không có công cụ tìm kiếm hiệu quả để so sánh các lựa chọn. Họ cần một ứng dụng di động với giao diện thân thiện, hỗ trợ tìm kiếm nhanh chóng thông qua từ khóa hoặc bộ lọc nâng cao như thành phố, quận/huyện, mức giá, diện tích, và tiện ích (wifi, điều hòa, nhà vệ sinh riêng, v.v.). Người thuê mong muốn xem chi tiết phòng trọ (mô tả, hình ảnh, giá cả), lưu danh sách phòng yêu thích, gửi đặt phòng, quản lý các khoản thanh toán, và tra cứu lịch sử hợp đồng. Nhóm đối tượng thứ hai là **chủ nhà trọ**, bao gồm các cá nhân hoặc tổ chức sở hữu và quản lý phòng trọ, căn hộ cho thuê, chẳng hạn như chủ nhà trọ tư nhân, quản lý ký túc xá, hoặc các đơn vị kinh doanh bất động sản quy mô nhỏ. Họ cần một nền tảng để quảng bá phòng trọ, tiếp cận đúng đối tượng khách hàng và quản lý giao dịch hiệu quả. Các chức năng chính bao gồm đăng bài giới thiệu phòng trọ với thông tin chi tiết (mô tả, hình ảnh, giá, tiện ích), quản lý bài đăng (thêm, xóa, sửa, tìm kiếm), và chờ admin duyệt bài để đảm bảo chất lượng. Chủ trọ nhận thông báo trong ứng dụng khi có đặt phòng từ khách hàng, quản lý hoặc hủy hợp đồng, và theo dõi các khoản thanh toán từ người thuê thông qua tính năng quản lý chi tiêu. Nhóm này tập trung vào việc tối ưu hóa quy trình quản lý, đảm bảo thông tin minh bạch và giao dịch nhanh chóng, từ đó tăng khả năng cho thuê và giảm thời gian trống phòng.

Cả hai nhóm đối tượng đều đóng vai trò quan trọng trong việc định hình các yêu cầu thiết kế và phát triển ứng dụng, từ giao diện người dùng trên Android đến hệ thống backend, nhằm mang lại trải nghiệm hiệu quả và thân thiện.

**Phạm vi nghiên cứu** tập trung vào việc phát triển một ứng dụng di động trên nền tảng Android sử dụng ngôn ngữ lập trình Java, kết hợp với hệ thống backend được xây dựng bằng Node.js, Express.js và cơ sở dữ liệu MongoDB. Ứng dụng được thiết kế để đáp ứng nhu cầu tìm kiếm và quản lý phòng trọ tại các thành phố lớn ở Việt Nam, với giao diện tiếng/ Các chức năng chính của ứng dụng gồm: quản lý hồ sơ người dùng

(người thuê đăng ký, đăng nhập, cập nhật thông tin, khôi phục mật khẩu; chủ trọ quản lý bài đăng và xác thực thông tin); tìm kiếm phòng trọ bằng từ khóa hoặc bộ lọc nâng cao (theo thành phố, quận/huyện, giá, diện tích, tiện ích). Ứng dụng tập trung vào thị trường Việt Nam, phục vụ nhu cầu thuê và cho thuê phòng trọ của sinh viên, người lao động và chủ trọ quy mô nhỏ, không bao gồm các đối tượng như môi giới bất động sản chuyên nghiệp hoặc tổ chức quản lý bất động sản lớn. Các ràng buộc bao gồm: chỉ phát triển cho Android, sử dụng tiếng Việt, không tích hợp dịch vụ bên ngoài, và tập trung vào trải nghiệm người dùng và hiệu suất hệ thống. Phạm vi này đảm bảo tính khả thi trong thời gian thực hiện đồ án, đồng thời mang lại giá trị thực tiễn cao trong bối cảnh đô thị hóa và nhu cầu nhà ở ngày càng tăng tại Việt Nam.

## Chương 1: CƠ SỞ LÝ THUYẾT

### 1.1. Giới thiệu tổng quan về ngôn ngữ Java

Java chính là một ngôn ngữ lập trình hướng đối tượng mạnh mẽ, được sử dụng rộng rãi trong phát triển phần mềm, đặc biệt là các ứng dụng có quy mô lớn. Đây là một ngôn ngữ độc lập nền tảng, hoạt động dựa trên mô hình “Write Once, Run Anywhere” (viết một lần, chạy mọi nơi) thông qua Java Virtual Machine (JVM). Chính vì tính ổn định, bảo mật cao và khả năng ứng dụng rộng rãi mà Java ngày càng được tin dùng trong nhiều lĩnh vực, từ web, mobile đến hệ thống doanh nghiệp.

Việc sử dụng ngôn ngữ Java có thể áp dụng cho nhiều nền tảng khác nhau, từ Windows, macOS, Linux cho tới các hệ thống nhúng. Java cũng là ngôn ngữ chính thức cho việc phát triển ứng dụng Android. Bởi thế mà việc sử dụng ngày càng đa dạng, có thể đáp ứng tốt cho nhiều nhu cầu và những đòi hỏi khác nhau của người dùng lẫn doanh nghiệp.

Nhìn chung, ngôn ngữ lập trình Java được đánh giá dễ học đối với người mới, đồng thời cực kỳ mạnh mẽ khi ứng dụng trong các dự án lớn. Java được phát triển bởi Sun Microsystems (nay thuộc Oracle) và vẫn đang là một trong những ngôn ngữ phổ biến nhất thế giới. Việc sử dụng Java có thể được nhúng trong các ứng dụng web (Java EE, Spring), ứng dụng desktop (JavaFX), hoặc hệ thống backend mạnh mẽ. Chính khả năng hoạt động ổn định, an toàn và linh hoạt giúp ngôn ngữ này được ứng dụng rộng rãi trong lĩnh vực công nghệ thông tin.

Những ưu điểm làm nên chất lượng của ngôn ngữ Java:

- **Là ngôn ngữ lập trình hướng đối tượng mạnh mẽ:** Java hỗ trợ đầy đủ các tính chất OOP như kế thừa, đa hình, trừu tượng... giúp tổ chức mã nguồn khoa học, dễ bảo trì và mở rộng.
- **Độc lập nền tảng (Platform-Independent):** Java được biên dịch sang bytecode và chạy trên JVM, cho phép phần mềm hoạt động trên mọi hệ điều hành mà không cần viết lại.
- **Hiệu suất ổn định và đáng tin cậy:** JVM tối ưu quá trình thực thi nên Java có tốc độ xử lý tốt, phù hợp cho các hệ thống lớn cần độ ổn định cao như ngân hàng, doanh nghiệp,...

- **Bảo mật vượt trội:** Java cung cấp nhiều cơ chế bảo mật tích hợp, chẳng hạn như Security Manager và kiểm soát truy cập, giúp xây dựng các ứng dụng an toàn.
- **Cộng đồng lớn và tài liệu phong phú:** Đây cũng là ưu điểm nổi bật bởi Java có cộng đồng toàn cầu mạnh mẽ, thư viện nhiều, framework đa dạng hỗ trợ lập trình viên triển khai dễ dàng hơn.
- **Khả năng mở rộng cao:** Java hoạt động tốt trong các hệ thống lớn với lưu lượng cao, dễ dàng mở rộng khi nhu cầu tăng trưởng.

#### **Ví dụ:**

// Hiển thị thông báo chào mừng người dùng

```
public class Main {  
    public static void greetUser(String name) {  
        System.out.println("Xin chào, " + name + "! Chào mừng bạn đến với Java.");  
    }  
  
    public static void main(String[] args) {  
        // Gọi hàm  
        greetUser("Tiến");  
    }  
}
```

#### **Lợi ích của Java:**

- **Ứng dụng rộng rãi trong doanh nghiệp:** Java được sử dụng để xây dựng hệ thống quản lý, phần mềm ngân hàng, ứng dụng thương mại điện tử,...
- **Khả năng đa nền tảng tuyệt vời:** Java chạy được trên mọi loại hệ điều hành thông qua JVM mà không cần chỉnh sửa mã.
- **Sử dụng cho ứng dụng Android:** Toàn bộ hệ sinh thái Android ban đầu được xây dựng dựa trên Java.
- **Hỗ trợ hệ thống thời gian thực:** Java có thể xây dựng các ứng dụng yêu cầu xử lý liên tục như server game, hệ thống theo dõi, phần mềm giao dịch,...
- **Hệ thống thư viện và framework mạnh mẽ:** Spring, Hibernate, Struts, JavaFX,... giúp phát triển ứng dụng nhanh và hiệu quả.

- Dễ tích hợp với các công nghệ khác: Java dễ dàng làm việc với cơ sở dữ liệu, web service, cloud và nhiều kiến trúc hiện đại.

## 1.2. Giới thiệu về NodeJs

Node.js là một nền tảng chạy JavaScript phía server, giúp các nhà phát triển xây dựng các ứng dụng mạng nhanh chóng, hiệu quả và có khả năng mở rộng cao. Node.js được phát triển dựa trên V8 JavaScript Engine của Google, trình thông dịch JavaScript được sử dụng trong trình duyệt Chrome. Điểm đặc trưng của Node.js là kiến trúc hướng sự kiện (event-driven) và mô hình không đồng bộ (asynchronous), giúp xử lý các yêu cầu từ client nhanh chóng và hiệu quả.

JavaScript phía server: Node.js cho phép các nhà phát triển sử dụng JavaScript ở phía server. Điều này mang lại sự thống nhất về ngôn ngữ lập trình giữa client và server, giúp giảm bớt sự phức tạp và cải thiện năng suất phát triển.

- Không đồng bộ và hướng sự kiện: Node.js sử dụng mô hình non-blocking I/O (nhập/xuất không chặn) và xử lý theo hướng sự kiện, giúp hệ thống có thể xử lý nhiều yêu cầu cùng lúc mà không bị chặn bởi các yêu cầu trước đó. Điều này giúp tối ưu hiệu suất, đặc biệt trong các ứng dụng có nhiều người dùng hoặc yêu cầu thời gian phản hồi nhanh.
- Cross-platform: Node.js hoạt động trên nhiều hệ điều hành như Windows, macOS, Linux và thậm chí trên các thiết bị nhúng như Raspberry Pi, giúp nó trở thành một nền tảng linh hoạt cho các loại dự án khác nhau.
- Xử lý đơn luồng với Event Loop: Mặc dù Node.js chỉ sử dụng một luồng để xử lý các yêu cầu, nó sử dụng Event Loop để quản lý nhiều kết nối cùng một lúc mà không cần nhiều luồng. Điều này giúp Node.js có thể xử lý nhiều yêu cầu đồng thời mà không gặp các vấn đề thường gặp với các hệ thống đa luồng như cạnh tranh tài nguyên (resource contention).
- Hỗ trợ WebSockets: Node.js hỗ trợ các kết nối WebSocket, giúp dễ dàng xây dựng các ứng dụng thời gian thực (real-time) như ứng dụng chat, các trò chơi nhiều người chơi, hoặc hệ thống theo dõi sự kiện.

### Ví dụ:

```
// Import module HTTP có sẵn trong Node.js
const http = require('http');
// Tạo một server
const server = http.createServer((req, res) => {
    res.statusCode = 200; // Trả về mã trạng thái 200 thành công
});
```

```
res.setHeader('Content-Type', 'text/plain'); // Đặt kiểu dữ liệu trả về
res.end('Hello, World!'); // Trả về nội dung
});
// Khởi động server tại cổng 3000
server.listen(3000, () => {
    console.log('Server chạy tại
http://localhost:3000/');
});
```

### **Ứng dụng của Node.Js:**

- API Backend: Node.js rất phổ biến để xây dựng các API RESTful với các thư viện như Express.js.
- Ứng dụng real-time: Node.js lý tưởng cho các ứng dụng thời gian thực như chat hoặc hệ thống thông báo.
- Xử lý file: Node.js có thể xử lý file (đọc, ghi, xóa) trên server một cách dễ dàng.
- Microservices: Node.js rất phù hợp cho các kiến trúc microservices nhờ khả năng nhẹ và hiệu suất cao.
- Ứng dụng IoT: Các thiết bị IoT thường sử dụng Node.js để xử lý dữ liệu cảm biến trong thời gian thực.

### **Các thư viện phổ biến của Node.Js:**

- Express.js: Framework nhẹ, linh hoạt để xây dựng web server và API.
- Socket.io: Hỗ trợ giao tiếp real-time giữa client và server.
- Mongoose: Thư viện để làm việc với MongoDB dễ dàng hơn.
- JWT (jsonwebtoken): Dùng để xác thực người dùng qua token.
- Lodash: Cung cấp các hàm tiện ích để xử lý dữ liệu.

### **Lợi ích của Node.Js:**

- Đơn giản hóa ngăn xếp công nghệ: Cả frontend và backend đều dùng JavaScript, giúp dễ dàng quản lý và đồng bộ mã nguồn.
- Hiệu suất cao: Xử lý hàng ngàn yêu cầu đồng thời mà không bị nghẽn.
- Cộng đồng lớn: Node.js có một cộng đồng nhà phát triển sôi động, luôn cập nhật công cụ và giải pháp mới.
- Tối ưu chi phí: Hiệu suất cao của Node.js giúp giảm tài nguyên phần cứng cần thiết, từ đó tiết kiệm chi phí.
- Khi nào sử dụng Node.Js:
- Khi cần xử lý real-time (chat, game, thông báo).
- Khi cần xây dựng API backend nhanh chóng.

- Khi làm việc với các ứng dụng nhiều kết nối như IoT. Khi cần hiệu suất cao với tài nguyên tối thiểu.

### **1.3. Giới thiệu về ExpressJs**

Express.js là một framework web nhẹ và linh hoạt, thường được biết đến trong phát triển ứng dụng server. Khi được áp dụng trong môi trường Java, nó có thể được tích hợp thông qua các mô hình bridge, API gateway hoặc các dịch vụ trung gian, giúp xây dựng các ứng dụng web và API một cách nhanh chóng và hiệu quả. Express.js đơn giản hóa quá trình phát triển backend, cung cấp bộ công cụ mạnh mẽ giúp xử lý yêu cầu HTTP, định tuyến, quản lý session và nhiều chức năng server khác.

#### **Đặc điểm nổi bật của Express.js :**

- Tích hợp với hệ thống Java: Express.js có thể được sử dụng như một service độc lập hoặc kết nối với các ứng dụng Java thông qua REST API, microservices hoặc API Gateway. Điều này giúp các dự án Java mở rộng khả năng xử lý, giao tiếp dịch vụ và tối ưu hóa tốc độ phản hồi từ tầng backend.
- Tối ưu hóa hiệu suất: Nhờ cấu trúc nhẹ và hướng xử lý tin gọn, Express.js phù hợp khi dùng làm lớp API trung gian cho các ứng dụng Java lớn. Điều này giúp giảm tải cho server Java chính và tăng hiệu suất tổng thể của hệ thống.
- Hệ thống middleware mạnh mẽ: Các middleware trong Express.js có thể được dùng để xử lý xác thực, bảo mật, lọc request, log request/response hoặc các tác vụ xử lý dữ liệu trước khi gửi đến dịch vụ Java phía sau.
- Hỗ trợ tốt REST API: Express.js là lựa chọn lý tưởng để xây dựng các endpoint giao tiếp với Java backend. Việc tách biệt lớp API giúp ứng dụng Java dễ mở rộng, dễ bảo trì và xử lý tốt hơn các luồng dữ liệu phức tạp.
- Hệ sinh thái phong phú: Express.js có nhiều module hỗ trợ mà bạn có thể tích hợp để bổ sung chức năng cho hệ thống Java như bảo mật, lưu session, phân tích dữ liệu, API throttling...
- Hỗ trợ nhiều engine hiển thị (view engines): Khi hoạt động như một lớp giao diện hoặc render HTML trước khi gửi đến ứng dụng Java, Express.js hỗ trợ nhiều engine template như Pug, EJS, Handlebars giúp tạo giao diện động.
- Cộng đồng lớn: Express.js có cộng đồng đông đảo, tài liệu dễ tìm, ví dụ rõ ràng – giúp việc tích hợp với Java trở nên dễ dàng, đặc biệt trong các hệ thống microservices.

#### **Ví dụ :**

```
const express = require('express');  
const app = express();  
// Route cơ bản để giao tiếp với dịch vụ Java phía sau
```

```
app.get('/java-service', (req, res) => {  
  // Gửi yêu cầu đến API Java hoặc xử lý dữ liệu trung gian  
  res.send('Express.js đang giao tiếp với dịch vụ Java!');  
});  
// Khởi chạy server tại cổng 3000  
app.listen(3000, () => {  
  console.log('API Gateway Express.js chạy tại http://localhost:3000');  
});
```

Ví dụ trên mô tả Express.js hoạt động như lớp API trung gian để ứng dụng Java có thể giao tiếp với frontend hoặc với các dịch vụ khác.

#### **Lợi ích khi sử dụng Express.js :**

- Dễ học, dễ tích hợp: Khi kết hợp với Java, Express.js hoạt động như một lớp API hoặc service nhẹ, cấu hình đơn giản và dễ mở rộng.
- Xử lý yêu cầu nhanh: Nhờ khả năng xử lý tốt luồng request, Express.js giúp giảm tải cho server Java chính trong hệ thống lớn.
- Tùy chỉnh linh hoạt: Có thể thêm middleware, module tùy chỉnh để phù hợp với kiến trúc Java doanh nghiệp.
- Phát triển nhanh: Express.js rút ngắn thời gian xây dựng lớp API hoặc dịch vụ hỗ trợ cho Java backend.
- Hỗ trợ mạnh cho REST API: Tạo và quản lý API dễ dàng, tối ưu cho các hệ thống Java microservices hoặc API Gateway.

#### **Khi nào nên sử dụng Express.js :**

- Khi bạn muốn xây dựng REST API nhanh chóng mà vẫn giữ backend chính bằng Java.
- Khi cần lớp trung gian (API Gateway, BFF – Backend for Frontend) để giảm tải cho Java server.
- Khi xây dựng hệ thống microservices trong đó Java và Express.js phối hợp xử lý.

### **1.4. Giới thiệu về MongoDB**

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL mã nguồn mở phổ biến, được phát triển bởi MongoDB Inc. Khác với các hệ quản trị cơ sở dữ liệu quan hệ truyền thống, MongoDB lưu trữ dữ liệu dưới dạng tài liệu (document) linh hoạt theo cấu trúc BSON (Binary JSON). Điều này cho phép MongoDB xử lý dữ liệu phi cấu trúc hoặc bán cấu trúc một cách hiệu quả, đặc biệt phù hợp cho các ứng dụng web hiện đại và hệ thống lớn.

### **Đặc điểm của MongoDB:**

- **Kiến trúc NoSQL dạng document:**

MongoDB không sử dụng bảng và hàng như cơ sở dữ liệu quan hệ. Thay vào đó, dữ liệu được lưu trữ trong các collection (tương tự bảng) và mỗi record là một document dạng JSON/BSON.

- **Hiệu suất cao và khả năng mở rộng:**

MongoDB được tối ưu cho tốc độ xử lý với khả năng đọc/ghi nhanh. Hỗ trợ mở rộng theo chiều ngang (sharding) giúp phân chia dữ liệu trên nhiều máy chủ, phù hợp với ứng dụng có lưu lượng lớn và yêu cầu phân tán dữ liệu.

- **Tính bảo mật cao:**

MongoDB cung cấp các cơ chế kiểm soát truy cập, xác thực người dùng, mã hóa dữ liệu ở mức ổ đĩa (Encryption at Rest) và mã hóa trong truyền tải (TLS/SSL). Các tính năng này giúp bảo vệ dữ liệu khỏi truy cập trái phép.

- **Tính toàn vẹn và độ tin cậy của dữ liệu:**

MongoDB hỗ trợ ACID trong phạm vi document, đảm bảo tính toàn vẹn dữ liệu trong các giao dịch đơn hoặc nhiều document (phiên bản 4.0+). Điều này rất quan trọng cho các ứng dụng yêu cầu độ tin cậy cao.

- **Tương thích với nhiều ngôn ngữ lập trình:**

MongoDB hỗ trợ tích hợp với nhiều ngôn ngữ như Java, Python, Node.js, C#, Go... giúp dễ dàng triển khai trong các ứng dụng web, mobile hoặc microservices.

### **Ví dụ:**

```
db.HocSinh.insertOne({
  Ten: "Nguyen Van A",
  Lop: "10A1",
  Diem: 8.5
});
```

### **Lợi ích của MongoDB:**

- **Hiệu suất cao:**

MongoDB tối ưu hóa tốc độ đọc ghi, phù hợp cho các hệ thống yêu cầu xử lý dữ liệu nhanh theo thời gian thực.

- **Mã nguồn mở và miễn phí:**

MongoDB Community Edition là phiên bản miễn phí, mạnh mẽ, phù hợp cho cả cá nhân lẫn doanh nghiệp.

- **Tương thích tốt:**

Hỗ trợ nhiều ngôn ngữ lập trình phổ biến như Java, Python, Node.js, C#, đồng thời hoạt động tốt trên nhiều hệ điều hành.

- **Hỗ trợ giao dịch (Transactions):**

MongoDB hỗ trợ giao dịch ACID giúp đảm bảo tính toàn vẹn của dữ liệu khi thực hiện nhiều thay đổi liên quan.

- **Bảo mật cao:**

Cung cấp nhiều cơ chế bảo mật như phân quyền chi tiết, mã hóa dữ liệu, xác thực người dùng, IP Whitelist.

- **Dễ học và sử dụng:**

MongoDB sử dụng cú pháp document JSON quen thuộc, trực quan, dễ tiếp cận cho cả người mới lẫn chuyên gia.

- **Hỗ trợ dữ liệu lớn:**

Với khả năng sharding mạnh mẽ, MongoDB có thể xử lý lượng dữ liệu khổng lồ, phù hợp cho các hệ thống lớn với lượng truy cập cao.

### **Ứng dụng thực tế của MongoDB:**

- Lưu trữ dữ liệu người dùng cho mạng xã hội.
- Xây dựng hệ thống quản lý sản phẩm cho website thương mại điện tử.
  - Lưu trữ dữ liệu thời gian thực trong các ứng dụng IoT.
  - Xây dựng hệ thống blog, tin tức, CMS hiện đại.
  - Phân tích và lưu trữ dữ liệu lớn trong các hệ thống Big Data.

### **Một số công cụ hỗ trợ MongoDB:**

- **MongoDB Compass:** Công cụ giao diện đồ họa giúp trực quan hóa và quản lý dữ liệu.
- **MongoDB Atlas:** Dịch vụ đám mây của MongoDB cung cấp triển khai, giám sát và bảo mật tự động.
- **Robo 3T / Studio 3T:** Công cụ quản lý MongoDB mạnh mẽ cho lập trình viên.
- **DBeaver:** Công cụ mã nguồn mở hỗ trợ nhiều hệ quản trị cơ sở dữ liệu, bao gồm MongoDB.

## **Chương 2: PHÂN TÍCH, THIẾT KẾ HỆ THỐNG**

### **2.1. Xác định yêu cầu**

Tiến hành nghiên cứu các quy trình và luồng hoạt động của ứng dụng, đảm bảo hiểu rõ sâu sắc về cơ cấu và cách thức hoạt động của nó.

Nghiên cứu các nhu cầu thực tế của người dùng về một ứng dụng hỗ trợ đăng bài, tìm kiếm trọ,.. từ đó phân tích chức năng cần có của ứng dụng

Xây dựng các chức năng từ những yêu cầu của người dùng:

- **Đăng bài cho thuê trọ :** Đóng vai trò là phương tiện giúp chủ trọ hoặc người cho thuê dễ dàng cung cấp thông tin về phòng trọ đang cần cho thuê. Chức năng này cho phép chủ trọ mô tả đầy đủ về phòng như diện tích, mức giá, địa chỉ, tiện nghi, các điều kiện sinh hoạt, cũng như tải lên hình ảnh hoặc video giúp người thuê đánh giá trực quan về chất lượng phòng. Đồng thời, hệ thống hỗ trợ chủ trọ quản lý bài đăng, chỉnh sửa hoặc cập nhật lại thông tin khi cần thiết. Việc kiểm duyệt nội dung cũng là yếu tố quan trọng nhằm đảm bảo thông tin đăng tải luôn chính xác, tránh bài đăng ảo hoặc thông tin không rõ ràng.
- **Tìm kiếm phòng trọ :** Giúp người dùng dễ dàng tìm được phòng phù hợp với nhu cầu cá nhân. Người dùng có thể tìm kiếm theo từ khóa hoặc sử dụng các bộ lọc như mức giá, diện tích, vị trí, loại phòng, tiện nghi hay tình trạng phòng. Hệ thống cũng cần hiển thị đầy đủ thông tin phòng trọ kèm hình ảnh, mô tả chi tiết và vị trí bản đồ để người dùng xem xét nhanh chóng và đưa ra lựa chọn chính xác. Khả năng sắp xếp kết quả theo giá tăng dần, mới đăng nhất hoặc vị trí gần nhất giúp tối ưu trải nghiệm tìm kiếm của người thuê.
- **Đặt cọc phòng :** Được thiết kế nhằm hỗ trợ người thuê giữ phòng một cách an toàn và giảm rủi ro bị mất phòng. Sau khi người thuê đã xem phòng hoặc xác nhận quyết định thuê, họ có thể tiến hành thanh toán tiền cọc trực tuyến thông qua VNPAY. Hệ thống trạng thái phòng đã được giữ chỗ.

### **2.2. Phân tích yêu cầu**

#### **2.2.1. Chức năng đăng kí**

##### **a. Người thuê**

- Khi truy cập vào hệ thống, người thuê có thể thực hiện chức năng đăng ký tài khoản bằng cách nhập các thông tin cá nhân như họ tên, số điện thoại, email, mật khẩu và

lựa chọn loại tài khoản. Sau khi đăng ký thành công, người thuê sẽ sở hữu một tài khoản riêng để sử dụng các chức năng của hệ thống.

#### ***b. Người cho thuê***

- Người cho thuê có thể đăng ký tài khoản bằng cách cung cấp các thông tin như họ tên, số điện thoại, email, mật khẩu và lựa chọn loại tài khoản. Sau khi hoàn tất đăng ký, hệ thống sẽ tạo giao diện quản lý dành riêng cho chủ trọ.

### ***2.2.2. Chức năng đăng nhập***

#### ***a. Người thuê***

- Sau khi đăng nhập thành công, người thuê sẽ có thể xem các thống kê, tìm kiếm phòng trọ, thanh toán tiền cọc và lịch sử thanh toán, quản lý yêu cầu các phòng đã đặt. Người thuê có thể cập nhật lại thông tin cá nhân bất cứ khi nào cần thiết.

#### ***b. Người cho thuê***

Khi người cho thuê đăng nhập vào ứng dụng, hệ thống sẽ hiển thị giao diện quản lý dành riêng cho chủ trọ, bao gồm các thống kê về tổng phòng đã đăng, số phòng đã được thuê, doanh thu và số đặt phòng phòng cần xác nhận. Sau đăng nhập, chủ trọ có thể thực hiện các chức năng như đăng bài mới, chỉnh sửa thông tin phòng trọ, theo dõi các khoản tiền cọc đã được thanh toán. Chủ trọ cũng có thể cập nhật hồ sơ cá nhân.

#### ***c. Admin***

- Khi Admin đăng nhập, hệ thống sẽ hiển thị các chức năng quản trị bao gồm quản lý người thuê, người cho thuê và các bài đăng trên hệ thống. Admin có quyền thêm, sửa, xóa, khóa tài khoản hoặc cập nhật thông tin của người thuê và người cho thuê. Bên cạnh đó, Admin còn có thể xóa bài đăng, xem thống kê về tổng số người dùng, tổng doanh thu.

### ***2.2.3. Chức năng tìm kiếm***

Chức năng tìm kiếm được xây dựng dành riêng cho người thuê trọ nhằm giúp họ dễ dàng tìm được phòng phù hợp với nhu cầu cá nhân. Khi đăng nhập vào hệ thống, người thuê có thể sử dụng thanh tìm kiếm để nhập các từ khóa liên quan đến loại phòng mình đang cần. Chỉ cần nhập từ khóa liên quan tới tên đường, thành phố, giá trị, khu vực sẽ xuất hiện cái loại phòng liên quan.

### ***2.2.4. Chức năng đăng bài***

- Chức năng đăng bài được thiết kế dành cho người cho thuê trọ nhằm giúp họ dễ dàng giới thiệu phòng trọ của mình đến người thuê một cách rõ ràng và đầy đủ nhất. Sau khi đăng nhập vào hệ thống, người cho thuê có thể truy cập vào giao diện phần thêm phòng và bắt đầu tạo bài viết mới. Hệ thống cho phép họ nhập các thông tin quan

trọng như tiêu đề bài đăng, địa chỉ phòng trọ, mức giá cho thuê, diện tích, tiện nghi, điều kiện sinh hoạt, quy định phòng và mô tả chi tiết về không gian. Người cho thuê cũng có thể tải lên nhiều hình ảnh nhằm giúp người thuê có cái nhìn trực quan và chính xác hơn về phòng trọ. Ngoài ra, hệ thống hỗ trợ chỉnh sửa, cập nhật hoặc xóa bài đăng khi có sự thay đổi về giá cả, tình trạng phòng hoặc thông tin liên quan. Chức năng đăng bài giúp người cho thuê trọ dễ dàng tiếp cận khách hàng tiềm năng, đồng thời tạo sự minh bạch và chuyên nghiệp trong việc chia sẻ thông tin phòng trọ.

#### **2.2.5. Chức năng quản lý đặt phòng**

##### **a. Người thuê**

- Người thuê có thể xem lại các yêu cầu đã gửi, kiểm tra trạng thái xử lý như “đang chờ duyệt”, “đã duyệt” hoặc “bị từ chối”, từ đó dễ dàng theo dõi tiến trình yêu cầu của mình.

##### **b. Người cho thuê**

- Người cho thuê có thể xem chi tiết thông tin yêu cầu của người thuê, đánh giá mức độ phù hợp và thực hiện các hành động như chấp nhận, từ chối. Các

#### **2.2.6. Chức năng thêm sửa xóa bài đăng**

##### **a. Người cho thuê**

- Sau khi đăng nhập, người cho thuê có thể tạo bài đăng mới bằng cách nhập các thông tin như tiêu đề, giá thuê, diện tích, địa chỉ, tiện nghi và hình ảnh của phòng trọ. Bên cạnh đó, họ cũng có thể chỉnh sửa bài đăng khi có sự thay đổi về giá cả, tình trạng phòng hoặc các thông tin liên quan, giúp đảm bảo dữ liệu luôn được cập nhật đầy đủ và chính xác. Trong trường hợp phòng trọ không còn cho thuê nữa, người cho thuê có thể xóa bài đăng để tránh gây nhầm lẫn cho người tìm trọ.

##### **b. Admin**

- Hệ thống cung cấp quyền quản trị cho phép Admin kiểm duyệt các bài đăng trước khi hiển thị công khai, đảm bảo nội dung tuân thủ quy định và không vi phạm chính sách. Admin cũng có thể gỡ bỏ những bài đăng sai phạm, chứa thông tin không chính xác

#### **2.2.7. Chức năng đặt phòng**

- Sau khi tìm kiếm bài đăng về phòng phù hợp người dùng có thể đặt phòng, hệ thống sẽ cho người dùng biết giá cả thuê, người dùng có thể chọn thuê trong bao nhiêu tháng và gửi đặt phòng. Phòng khi được gửi yêu cầu sẽ vào trạng thái chờ xác nhận, khi chủ trọ xác nhận thì người dùng có thể thanh toán tiền cọc phòng cho chủ phòng thông qua VNPay và sẽ được thông kê trong giao diện người dùng

#### **2.2.8. Quản lý tài khoản người dùng**

- Chức năng quản lý tài khoản của Admin cho phép Admin có thể xem danh sách người dùng, thực hiện các thao tác thêm, sửa và xóa tài khoản trong hệ thống. Với quyền quản trị cao nhất, Admin có thể tạo mới tài khoản cho người thuê hoặc người cho thuê khi cần thiết, đồng thời chỉnh sửa thông tin tài khoản nếu có sai sót hoặc khi người dùng yêu cầu hỗ trợ.

### 2.3. Mô hình hóa yêu cầu

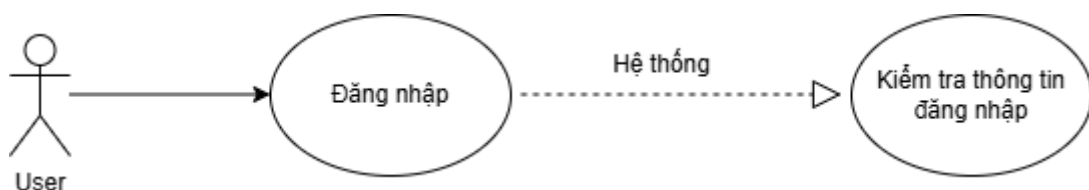
- Usecase tổng quát



Hình 2.1 Biểu đồ Usecase tổng quát

#### 2.3.1. Chức năng đăng nhập

- Biểu đồ Usecase đăng nhập



Hình 2.2. Biểu đồ Usecase đăng nhập

- Mô tả Usecase đăng nhập

Bảng 2.1.Mô tả Usecase đăng nhập

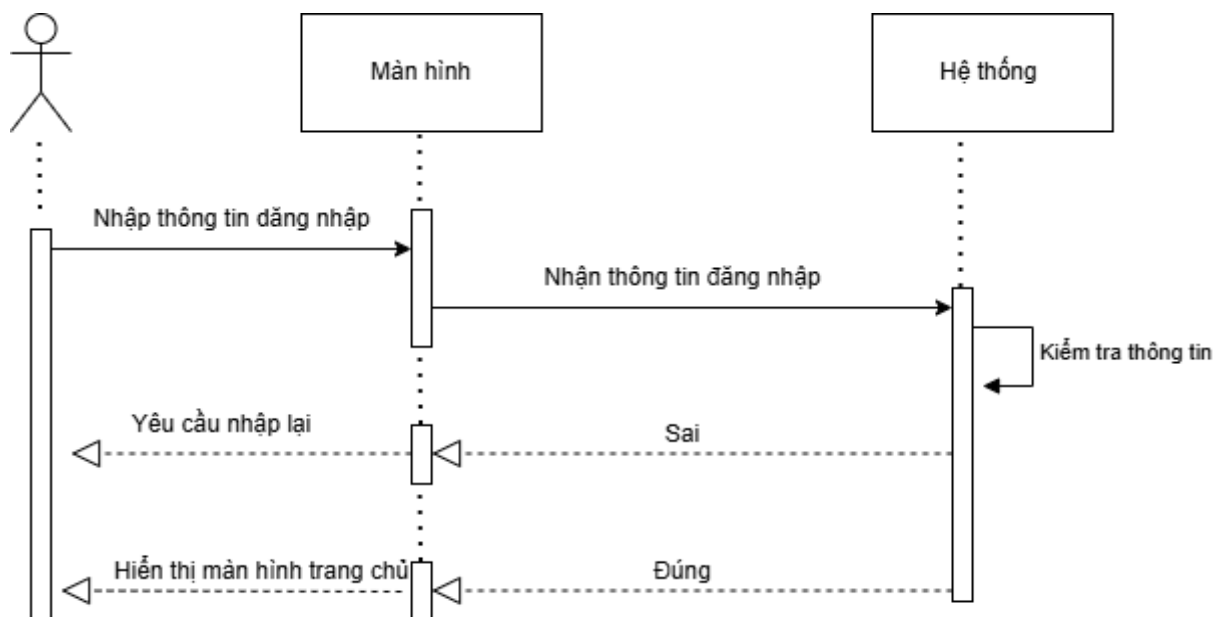
|                   |                                                          |
|-------------------|----------------------------------------------------------|
| Tên Usecase       | Đăng nhập                                                |
| Tác nhân          | Admin ,người thuê và người cho thuê                      |
| Điều kiện đầu vào | Admin ,người thuê và người cho thuê chưa đăng nhập       |
| Kết quả đầu ra    | Admin ,người thuê và người cho thuê đăng nhập thành công |
| Mô tả             | Use case cho phép Admin, nha sĩ và người dùng đăng nhập  |

- Dòng chính Usecase đăng nhập

Bảng 2.2 Dòng chính Use Case đăng nhập

| STT | Thực hiện | Hành động                                                 |
|-----|-----------|-----------------------------------------------------------|
| 1   | Tác nhân  | Admin ,người thuê và người cho thuê truy cập vào ứng dụng |
| 2   | Hệ thống  | Kiểm tra email và mật khẩu có đúng không                  |
| 3   | Tác nhân  | Nhập email, mật khẩu                                      |

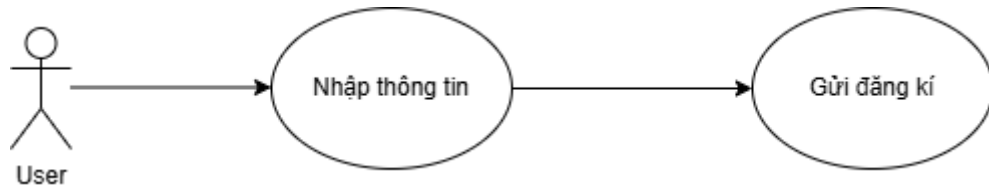
- Biểu đồ tuần tự đăng nhập



Hình 2.3 Biểu đồ tuần tự đăng nhập

### 2.3.2. Chức năng đăng kí

- Biểu đồ Usecase đăng kí



Hình 2.4. Biểu đồ Usecase đăng kí

- Mô tả UseCase đăng kí

Bảng 2.3 Mô tả Use Case đăng kí

|                   |                                                                                                                                                                                                                            |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tên Usecase       | Đăng kí                                                                                                                                                                                                                    |
| Tác nhân          | Người thuê và người cho thuê                                                                                                                                                                                               |
| Điều kiện đầu vào | Người thuê và người cho thuê chưa có tài khoản trong hệ thống                                                                                                                                                              |
| Kết quả đầu ra    | Người thuê và người cho thuê đăng ký tài khoản thành công hoặc thất bại với thông báo lỗi rõ ràng                                                                                                                          |
| Mô tả             | Use case cho phép Người thuê và Người cho thuê tạo tài khoản mới bằng cách nhập email và mật khẩu, gửi thông tin đăng ký lên hệ thống và nhận phản hồi xác nhận đăng ký thành công hoặc thông báo lỗi nếu có vấn đề xảy ra |

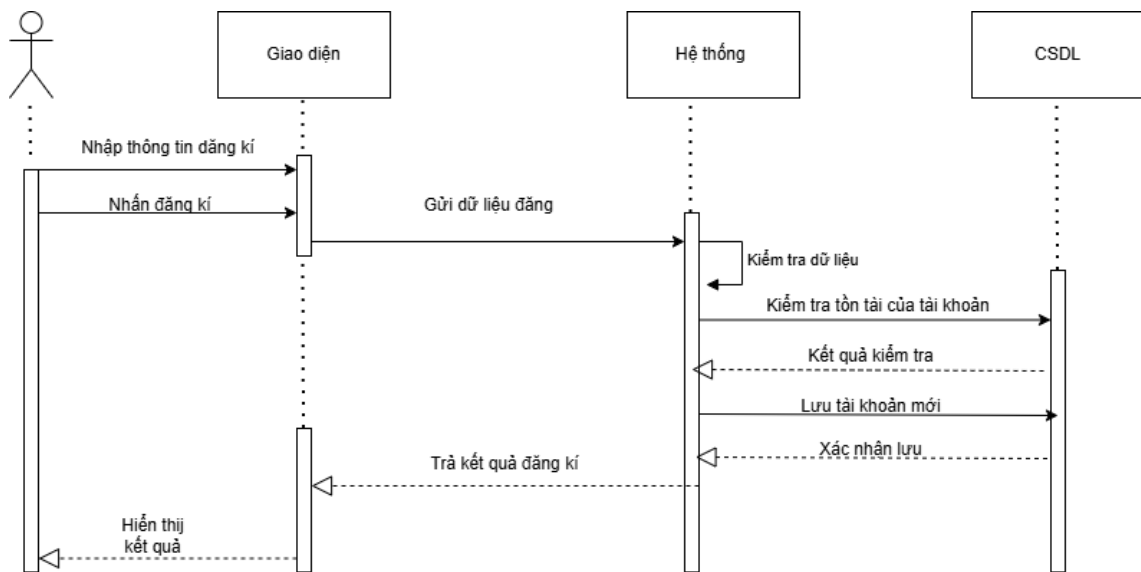
- Dòng chính Usecase đăng kí

Bảng 2.4 Dòng chính Use Case đăng nhập

| STT | Thực hiện | Hành động                                                                |
|-----|-----------|--------------------------------------------------------------------------|
| 1   | Tác nhân  | Người thuê hoặc người cho thuê truy cập vào trang đăng nhập của ứng dụng |

|   |          |                                                                         |
|---|----------|-------------------------------------------------------------------------|
| 2 | Tác nhân | Nhập email và mật khẩu<br>chọn loại tài khoản để<br>tạo tài khoản mới   |
| 3 | Hệ thống | Kiểm tra tính hợp lệ của<br>email và mật khẩu                           |
| 4 | Hệ thống | Kiểm tra xem email đã<br>được đăng ký chưa                              |
| 5 | Hệ thống | Lưu thông tin tài khoản<br>mới nếu hợp lệ và chưa<br>tồn tại            |
| 6 | Hệ thống | Gửi thông báo đăng ký<br>thành công hoặc lỗi (nếu<br>có) tới người dùng |

- Biểu đồ tuần tự đăng kí



Hình 2.5 Biểu đồ tuần tự đăng kí

### 2.3.3. Chức năng tìm kiếm

- Biểu đồ Usecase tìm kiếm



Hình 2.6 Biểu đồ Use Case tìm kiếm phòng trọ

- Mô tả Usecase tìm kiếm

Bảng 2.5 Mô tả Use Case tìm kiếm

| Tên Usecase       | Tìm kiếm phòng trọ                                                                                                                                                                                         |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tác nhân          | Người thuê trọ                                                                                                                                                                                             |
| Điều kiện đầu vào | Người thuê đã đăng nhập vào hệ thống                                                                                                                                                                       |
| Kết quả đầu ra    | Hiển thị danh sách phòng trọ phù hợp với từ khóa và bộ lọc tìm kiếm                                                                                                                                        |
| Mô tả             | Use case cho phép người thuê trọ nhập từ khóa liên quan tới nhu cầu để tìm kiếm phòng trọ phù hợp. Hệ thống xử lý yêu cầu và trả về kết quả tìm kiếm giúp người thuê dễ dàng lựa chọn phòng trọ mong muốn. |

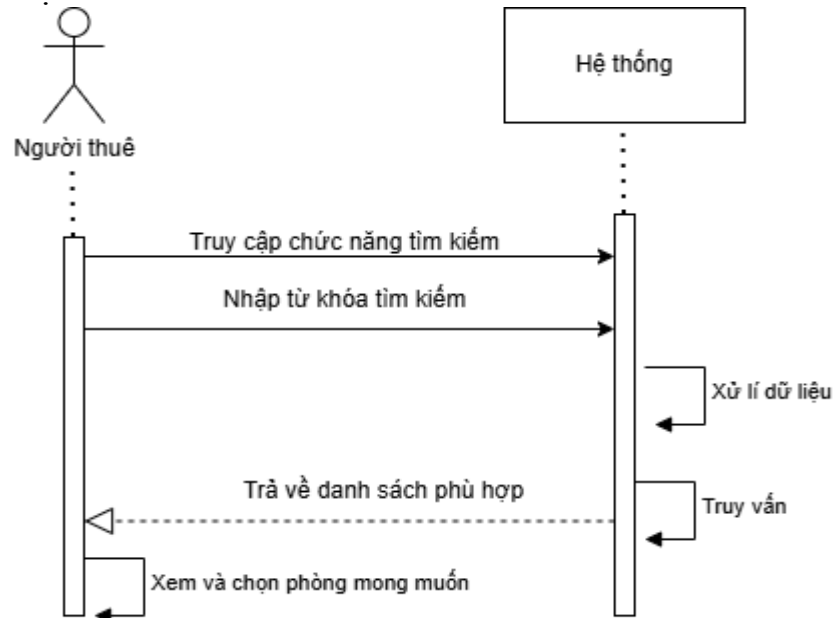
- Dòng chính Use Case tìm kiếm

Bảng 2.6 Dòng chính Use Case tìm kiếm

| STT | Thực hiện  | Hành động                                                  |
|-----|------------|------------------------------------------------------------|
| 1   | Người thuê | Truy cập vào chức năng tìm kiếm trên màn hình hệ thống     |
| 2   | Người thuê | Nhập từ khóa phù hợp với nhu cầu phòng cần thuê            |
| 3   | Hệ thống   | Nhận yêu cầu tìm kiếm và xử lý dữ liệu phù hợp             |
| 4   | Hệ thống   | Truy vấn các bài đăng trọ theo từ khóa và bộ lọc được chọn |

|   |            |                                                      |
|---|------------|------------------------------------------------------|
| 5 | Hệ thống   | Trả về danh sách phòng trọ phù hợp với yêu cầu       |
| 6 | Người thuê | Xem kết quả tìm kiếm và lựa chọn phòng trọ mong muốn |

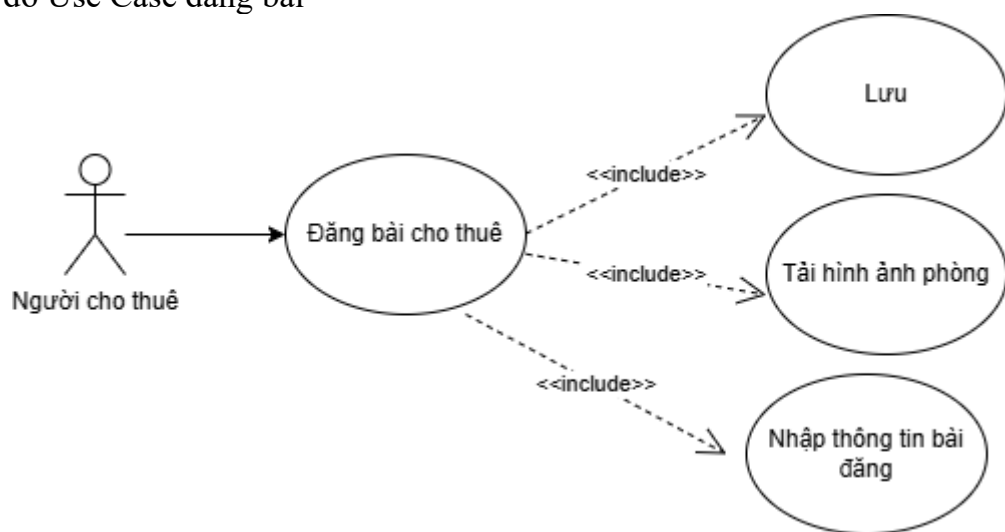
- Biểu đồ tuần tự tìm kiếm



Hình 2.7 Biểu đồ tuần tự tìm kiếm phòng trọ

#### 2.3.4. Chức năng đăng bài

- Biểu đồ Use Case đăng bài



Hình 2.8 Biểu đồ Use Case đăng bài

- Mô tả Use Case đăng bài

Bảng 2.7 Mô tả Use Case đăng bài

|                   |                                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tên Usecase       | Đăng bài                                                                                                                                                                                                                                                                                                                                                                         |
| Tác nhân          | Người cho thuê                                                                                                                                                                                                                                                                                                                                                                   |
| Điều kiện đầu vào | Người cho thuê đã đăng nhập vào hệ thống                                                                                                                                                                                                                                                                                                                                         |
| Kết quả đầu ra    | Bài đăng phòng trọ được tạo mới và hiển thị trên hệ thống                                                                                                                                                                                                                                                                                                                        |
| Mô tả             | Use Case cho phép người cho thuê tạo bài đăng mới về phòng trọ, bao gồm các thông tin như tiêu đề, mô tả, hình ảnh, giá cả, diện tích khu vực. Sau khi nhập đầy đủ thông tin, người cho thuê gửi bài đăng lên hệ thống, hệ thống lưu và hiển thị bài đăng để người thuê có thể tìm kiếm hoặc xem. Nếu có quản lý bởi admin, bài đăng sẽ được duyệt trước khi hiển thị công khai. |

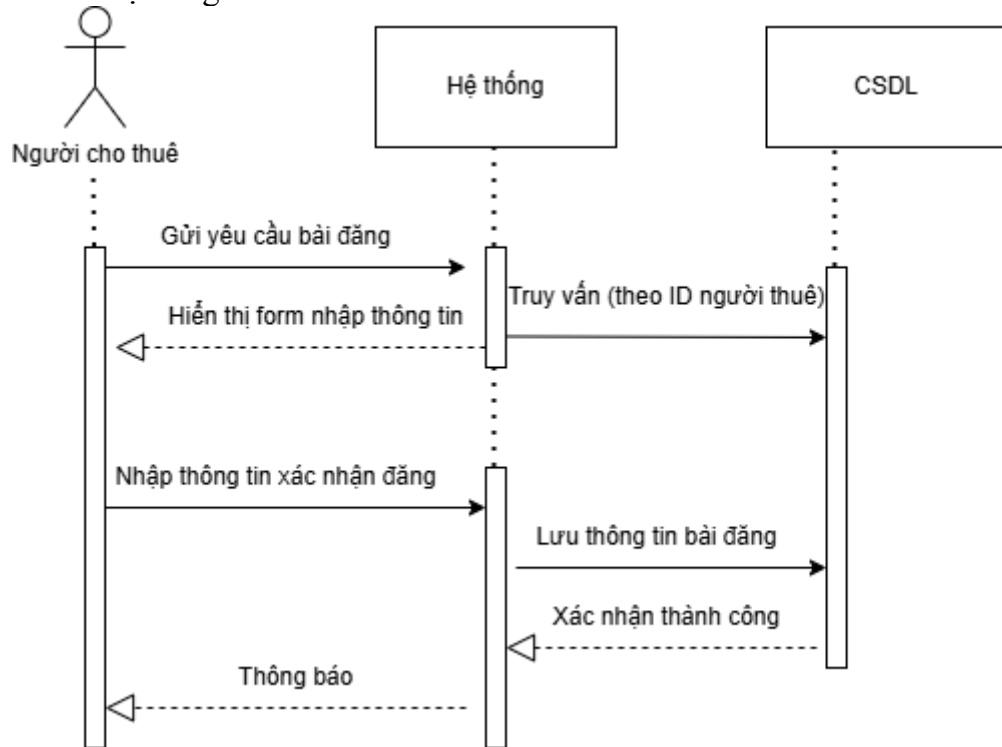
- Dòng chính Use Case đăng bài

Bảng 2.8 Dòng chính Use Case đăng bài

| STT | Thực hiện      | Hành động                                                                                           |
|-----|----------------|-----------------------------------------------------------------------------------------------------|
| 1   | Người cho thuê | Truy cập vào chức năng “Thêm phòng” trên màn hình hệ thống                                          |
| 2   | Người cho thuê | Nhập đầy đủ thông tin về phòng trọ: tiêu đề, mô tả, giá cả, diện tích, khu vực, tiện nghi, hình ảnh |
| 3   | Người cho thuê | Xem lại thông tin bài đăng và xác nhận gửi bài                                                      |
| 4   | Hệ thống       | Nhận thông tin bài đăng và lưu vào hệ thống                                                         |
| 5   | Hệ thống       | Hiển thị bài đăng công khai                                                                         |

|   |                |                                                                    |
|---|----------------|--------------------------------------------------------------------|
| 6 | Người cho thuê | Nhận thông báo xác nhận đăng bài thành công và trạng thái bài đăng |
|---|----------------|--------------------------------------------------------------------|

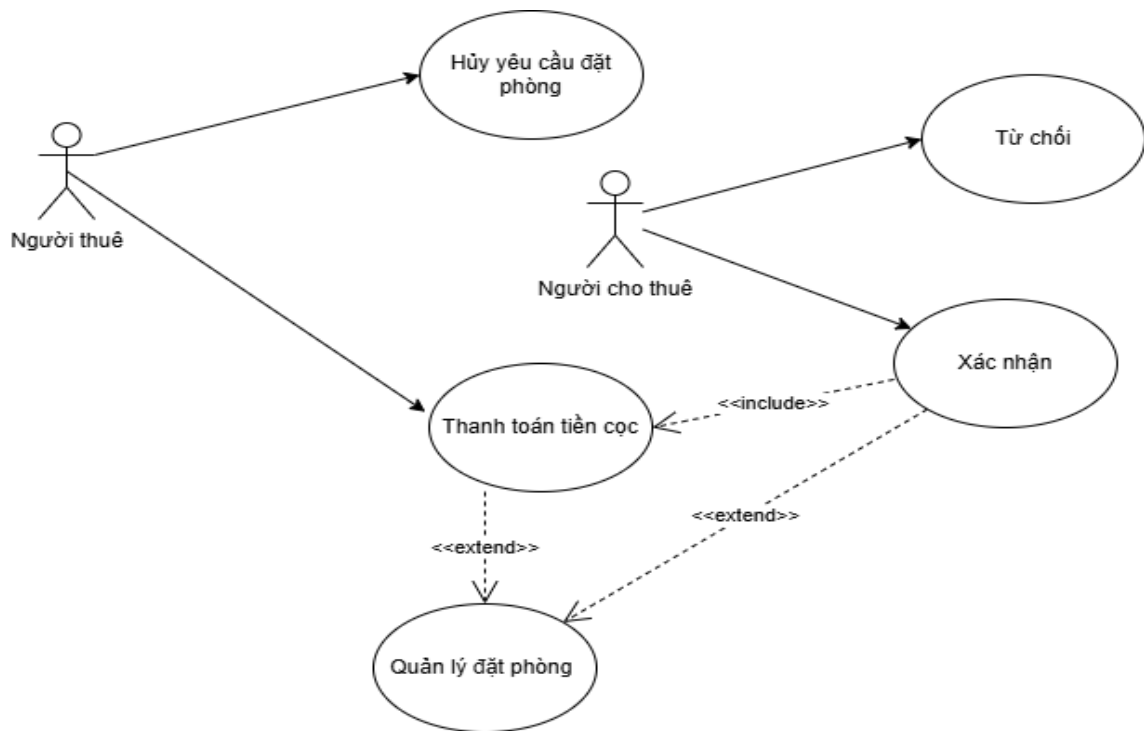
- Biểu đồ tuần tự đăng bài



Hình 2.9 Biểu đồ tuần tự đăng bài

### 2.3.5. Chức năng quản lý đặt phòng

- Biểu đồ Use Case quản lý đặt phòng



Hình 2.10 Biểu đồ Use Case quản lý đặt phòng

- Mô tả Use Case quản lý đặt phòng

Bảng 2.9 Mô tả Use Case quản lý đặt phòng

|                   |                                                                                                                                                                                                                                                                                                             |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tên Usecase       | Quản lý đặt phòng                                                                                                                                                                                                                                                                                           |
| Tác nhân          | Người cho thuê và Người thuê                                                                                                                                                                                                                                                                                |
| Điều kiện đầu vào | Người thuê hoặc người cho thuê đã đăng nhập vào hệ thống                                                                                                                                                                                                                                                    |
| Kết quả đầu ra    | - Yêu cầu đặt phòng được cập nhật trạng thái: <ul style="list-style-type: none"> <li>• <b>Đã xác nhận</b> (do Người cho thuê).</li> <li>• <b>Đã thanh toán tiền cọc</b> (do Người thuê, sau khi được xác nhận).</li> <li>• <b>Bị từ chối</b> (do Người thuê hoặc bị từ chối bởi Người cho thuê).</li> </ul> |
| Mô tả             | <ul style="list-style-type: none"> <li>• Use Case này cho phép cả Người thuê và Người cho thuê tham gia vào quá trình quản lý yêu cầu đặt phòng. Người thuê có thể gửi yêu cầu đặt phòng và chủ động hủy yêu cầu nếu</li> </ul>                                                                             |

|  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <p>không còn nhu cầu hoặc khi yêu cầu chưa được xử lý. Khi Người cho thuê xác nhận yêu cầu, hệ thống sẽ mở quyền cho Người thuê thực hiện bước tiếp theo là thanh toán tiền cọc.</p> <ul style="list-style-type: none"><li>• Ở phía Người cho thuê, họ có thể xem danh sách các yêu cầu đặt phòng liên quan đến phòng trọ của mình và đưa ra quyết định xác nhận hoặc từ chối. Việc xác nhận sẽ kích hoạt giai đoạn thanh toán của Người thuê, trong khi việc từ chối sẽ kết thúc quy trình đặt phòng.</li><li>• Trong suốt quá trình, hệ thống tự động cập nhật trạng thái của yêu cầu theo từng thao tác và hiển thị thông tin kịp thời cho cả hai bên, đảm bảo sự minh bạch và chính xác trong quản lý đặt phòng.</li></ul> |
|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

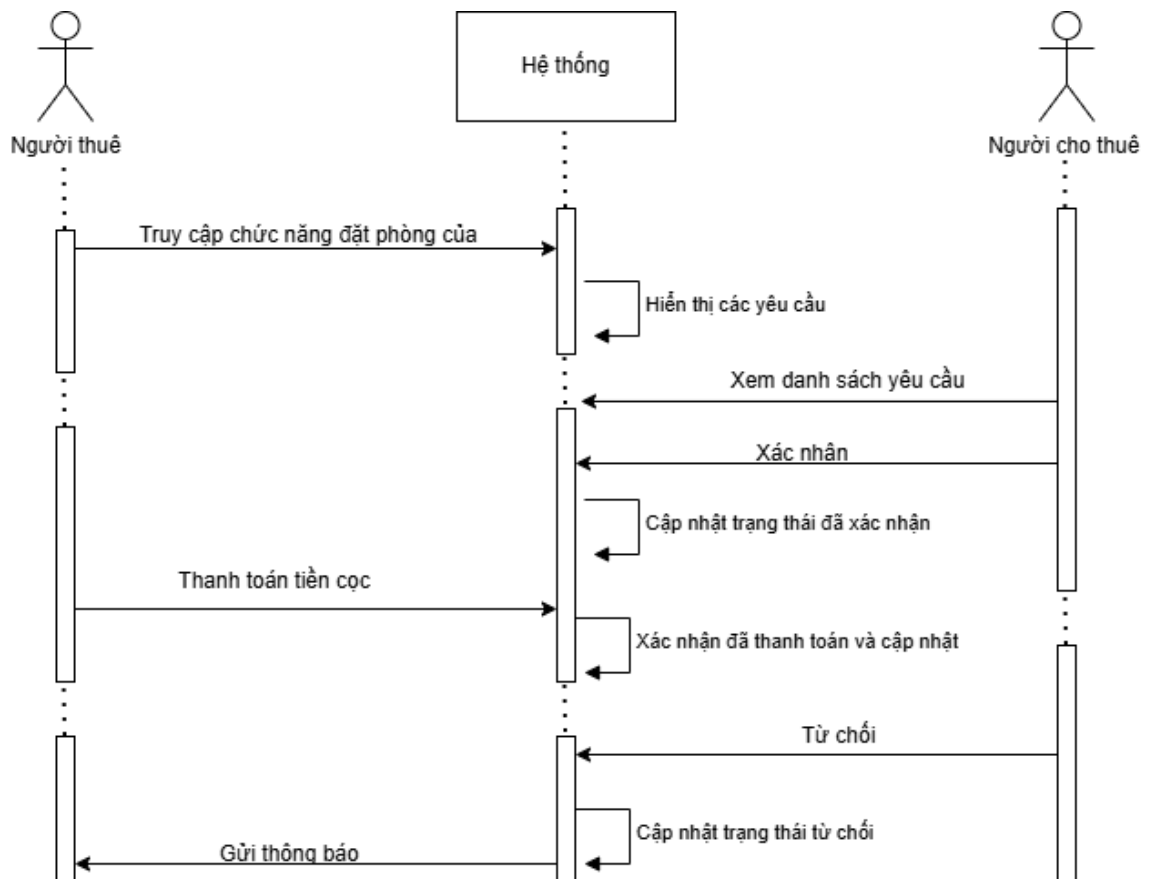
- Dòng chính Use Case quản lý đặt phòng

| STT | Thực hiện      | Hành động                                                               |
|-----|----------------|-------------------------------------------------------------------------|
| 1   | Người thuê     | Truy cập chức năng đặt phòng của tôi.                                   |
| 2   | Hệ thống       | Tiếp nhận yêu cầu và ghi nhận trạng thái “Chờ xác nhận”.                |
| 3   | Người cho thuê | Xem danh sách các yêu cầu đặt phòng.                                    |
| 4   | Người cho thuê | Chọn một yêu cầu và <b>xác nhận</b> hoặc <b>từ chối</b> .               |
| 5   | Hệ thống       | Cập nhật trạng thái yêu cầu tương ứng: “Đã xác nhận” hoặc “Bị từ chối”. |
| 6   | Người thuê     | Nếu yêu cầu đã được xác nhận, người thuê                                |

|   |          |                                                                                             |
|---|----------|---------------------------------------------------------------------------------------------|
|   |          | <b>có thể thực hiện thanh toán tiền cọc.</b>                                                |
| 7 | Hệ thống | Xác nhận thanh toán thành công và chuyển trạng thái yêu cầu thành “Đã thanh toán tiền cọc”. |

Bảng 2.10 Dòng chính Use Case quản lý đặt phòng

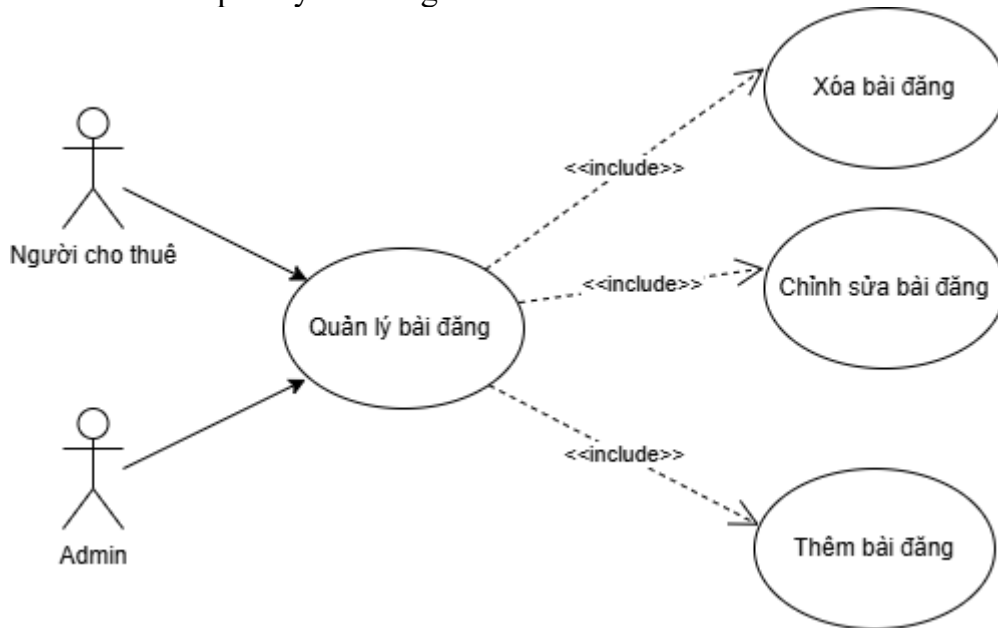
- Biểu đồ tuần tự quản lý đặt phòng



Hình 2.11 Biểu đồ tuần tự quản lý đặt phòng

### 2.3.6. Chức năng quản lý bài đăng

- Biểu đồ Use Case quản lý bài đăng



Hình 2.12 Biểu đồ Use Case quản lý bài đăng

- Mô tả Use Case quản lý bài đăng

Bảng 2.11 Mô tả Use Case quản lý bài đăng

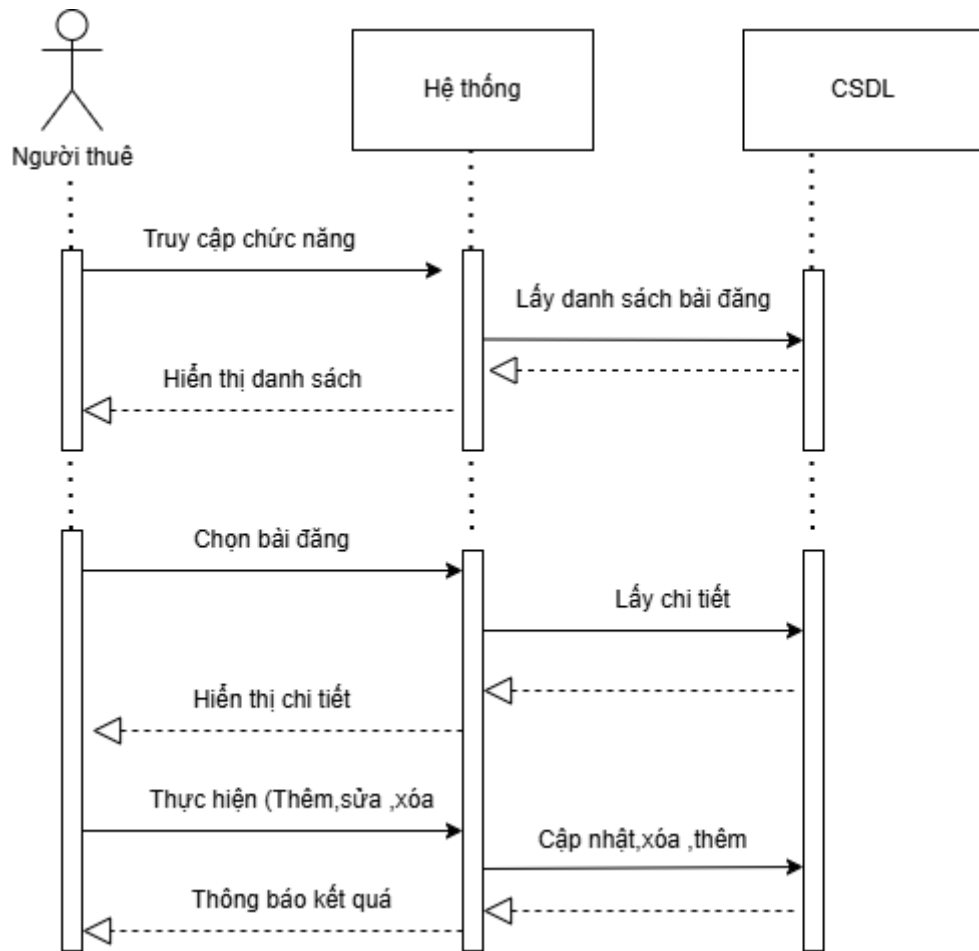
| Tên Usecase       | Quản lý bài đăng                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tác nhân          | Người cho thuê và Admin                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Điều kiện đầu vào | Tác nhân đã đăng nhập vào hệ thống và có quyền quản lý bài đăng                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Kết quả đầu ra    | Bài đăng được thêm ,xem, chỉnh sửa hoặc xóa tùy theo thao tác của tác nhân                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Mô tả             | Use Case cho phép người cho thuê và quản trị viên thực hiện các chức năng quản lý bài đăng phòng trọ đã được đăng lên hệ thống. Người cho thuê có thể xem danh sách các bài đăng của mình, chỉnh sửa nội dung (tiêu đề, mô tả, hình ảnh, giá cả, diện tích, địa chỉ...) hoặc xóa bài đăng không còn phù hợp. Quản trị viên cũng có thể xem, chỉnh sửa hoặc xóa bài đăng để đảm bảo chất lượng thông tin và tuân thủ quy định hệ thống. Mọi thay đổi được lưu và cập nhật để người thuê có thể xem thông tin chính xác và mới nhất. |

• Dòng chính Use Case quản lý bài đăng

| STT | Thực hiện              | Hành động                                                                                                                                               |
|-----|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1   | Người cho thuê /Admin  | Truy cập vào chức năng <b>“Quản lý bài đăng”</b> trên hệ thống                                                                                          |
| 2   | Hệ thống               | Hiển thị danh sách các bài đăng mà người dùng có quyền quản lý.                                                                                         |
| 3   | Người cho thuê / Admin | Chọn một bài đăng để thực hiện thao tác: <b>Thêm, Sửa, hoặc Xóa</b>                                                                                     |
| 4   | Hệ thống               | Hiển thị thông tin chi tiết bài đăng đã chọn                                                                                                            |
| 5   | Người cho thuê / Admin | Thực hiện một trong các hành động:<br>• <b>Sửa</b> :cập nhật thông tin bài đăng<br>• <b>Xóa</b> : xác nhận xóa bài đăng<br>• <b>Thêm</b> :Thêm bài đăng |
| 6   | Hệ thống               | Nếu người dùng chọn <b>Sửa</b> , hệ thống kiểm tra thông tin hợp lệ và cập nhật vào cơ sở dữ liệu                                                       |
| 7   | Hệ thống               | Nếu người dùng chọn <b>Xóa</b> , hệ thống ghi nhận thao tác và xóa bài đăng khỏi danh sách                                                              |
| 8   | Hệ thống               | Gửi thông báo cho người dùng về kết quả thao tác (sửa thành công, xóa thành công...)                                                                    |
| 9   | Người cho thuê / Admin | Nhận thông báo và xem lại trạng thái bài đăng sau khi thao tác                                                                                          |

Bảng 2.12 Dòng chính Use Case quản lý bài đăng

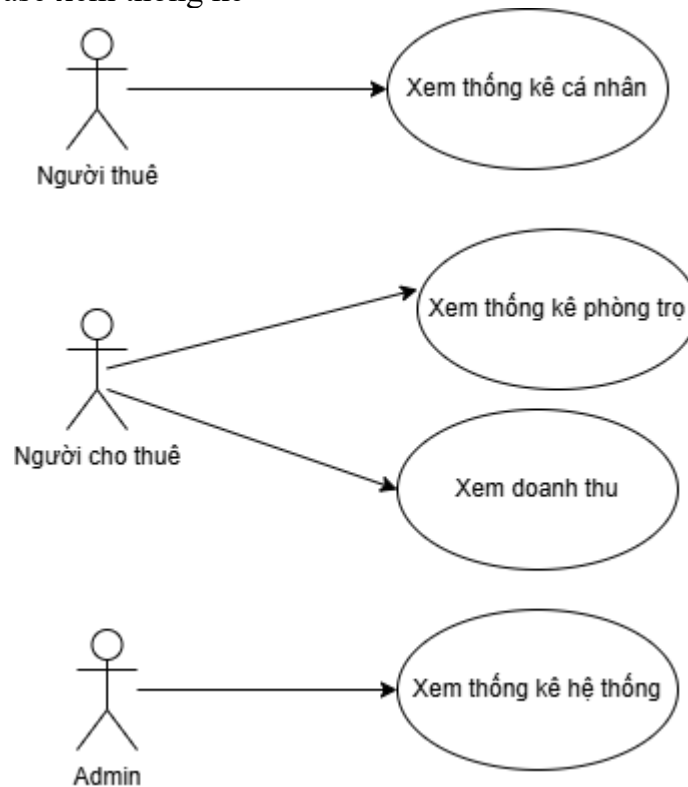
- Biểu đồ tuần tự quản lý bài đăng



Hình 2.13 Biểu đồ tuần tự quản lý bài đăng

### 2.3.7. Chức năng xem thống kê

- Biểu đồ Use Case xem thống kê



Hình 2.14 Biểu đồ Use Case xem thống kê

- Mô tả Use Case xem thống kê

Bảng 2.13 Mô tả Use Case xem thống kê

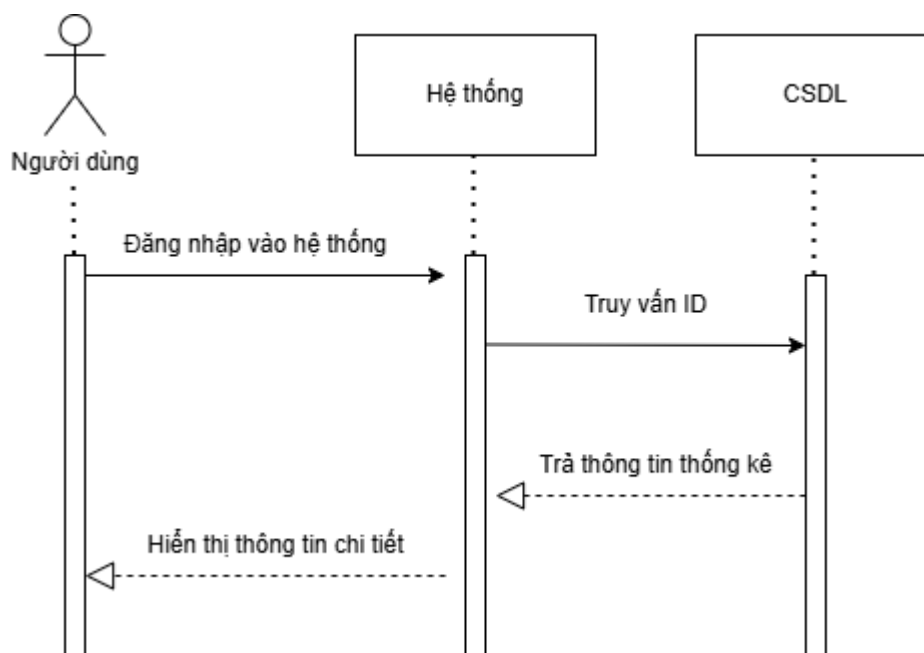
| Tên Usecase       | Xem thống kê                                                                                                                                            |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tác nhân          | Người cho thuê ,người thuê ,admin                                                                                                                       |
| Điều kiện đầu vào | Người thuê ,người cho thuê ,admin đã truy cập hệ thống và đã có các thao tác đặt phòng thanh toán ,đăng bài                                             |
| Kết quả đầu ra    | Thông tin chi tiết của thống kê được hiển thị đầy đủ và chính xác để người thuê ,người cho thuê , admin xem.                                            |
| Mô tả             | Use Case cho phép người thuê ,người cho thuê ,admin xem đầy đủ thông tin thống kê của từng tác nhân ,qua đó có thể nắm tình hình các hoạt động của mình |

- Dòng chính Use Case thống kê

| STT | Thực hiện                        | Hành động                                                                    |
|-----|----------------------------------|------------------------------------------------------------------------------|
| 1   | Người thuê ,người cho thuê,admin | Truy cập vào hệ thống.                                                       |
| 2   | Hệ thống                         | Hiển thị bảng thống kê tùy từng tác nhân mà sẽ hiện những thống kê khác nhau |
| 3   | Người thuê ,người cho thuê,admin | Xem các thống kê                                                             |

Bảng 2.14 Dòng chính Use Case xem thống kê

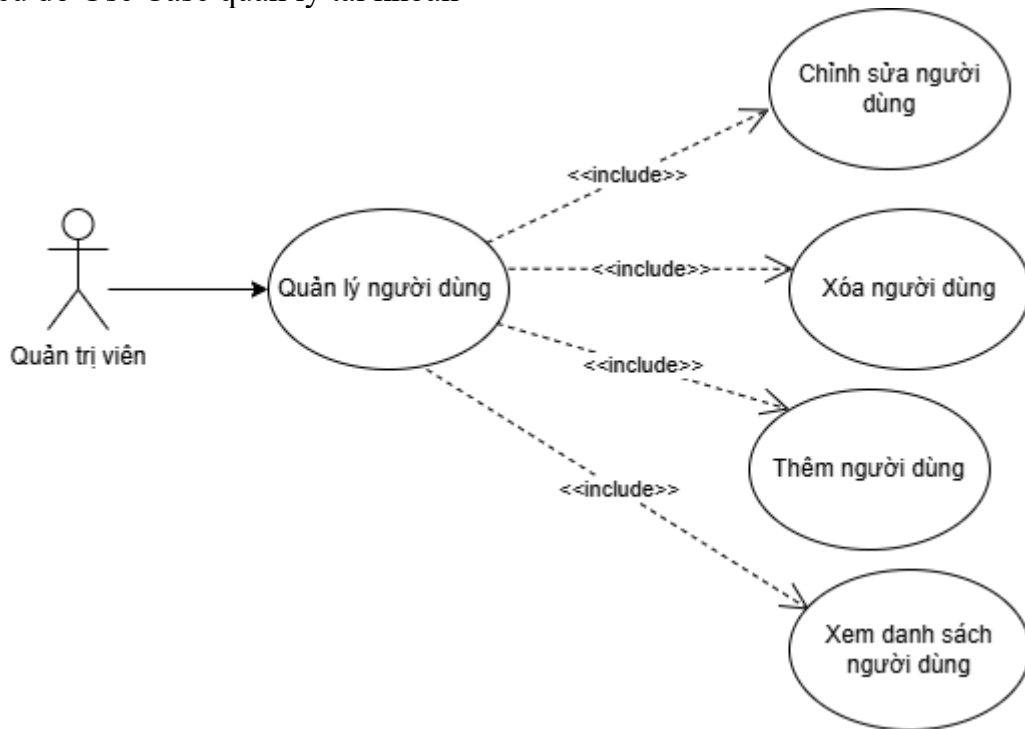
- Biểu đồ tuần tự quản lý bài đăng



Hình 2.15 Biểu đồ tuần tự quản lý đặt phòng

### 2.3.8. Chức năng quản lý tài khoản

- Biểu đồ Use Case quản lý tài khoản



Hình 2.16 Biểu đồ Use Case quản lý tài khoản

- Mô tả Use Case quản lý tài khoản

Bảng 2.15 Mô tả Use Case quản lý tài khoản

| Tên Usecase       | Quản lý tài khoản                                                                                                                                                                                                           |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tác nhân          | Admin                                                                                                                                                                                                                       |
| Điều kiện đầu vào | <ul style="list-style-type: none"> <li>• Quản trị viên đã đăng nhập hệ thống.</li> <li>• Có quyền quản trị người dùng.</li> </ul>                                                                                           |
| Kết quả đầu ra    | Người dùng được thêm, sửa, xóa hoặc danh sách người dùng được hiển thị chính xác tùy theo thao tác của quản trị viên.                                                                                                       |
| Mô tả             | Use Case “Quản lý người dùng” cho phép quản trị viên thực hiện đầy đủ các chức năng cần thiết để kiểm soát và điều chỉnh thông tin tài khoản của người dùng trên hệ thống. Use Case này bao gồm các chức năng con như: Thêm |

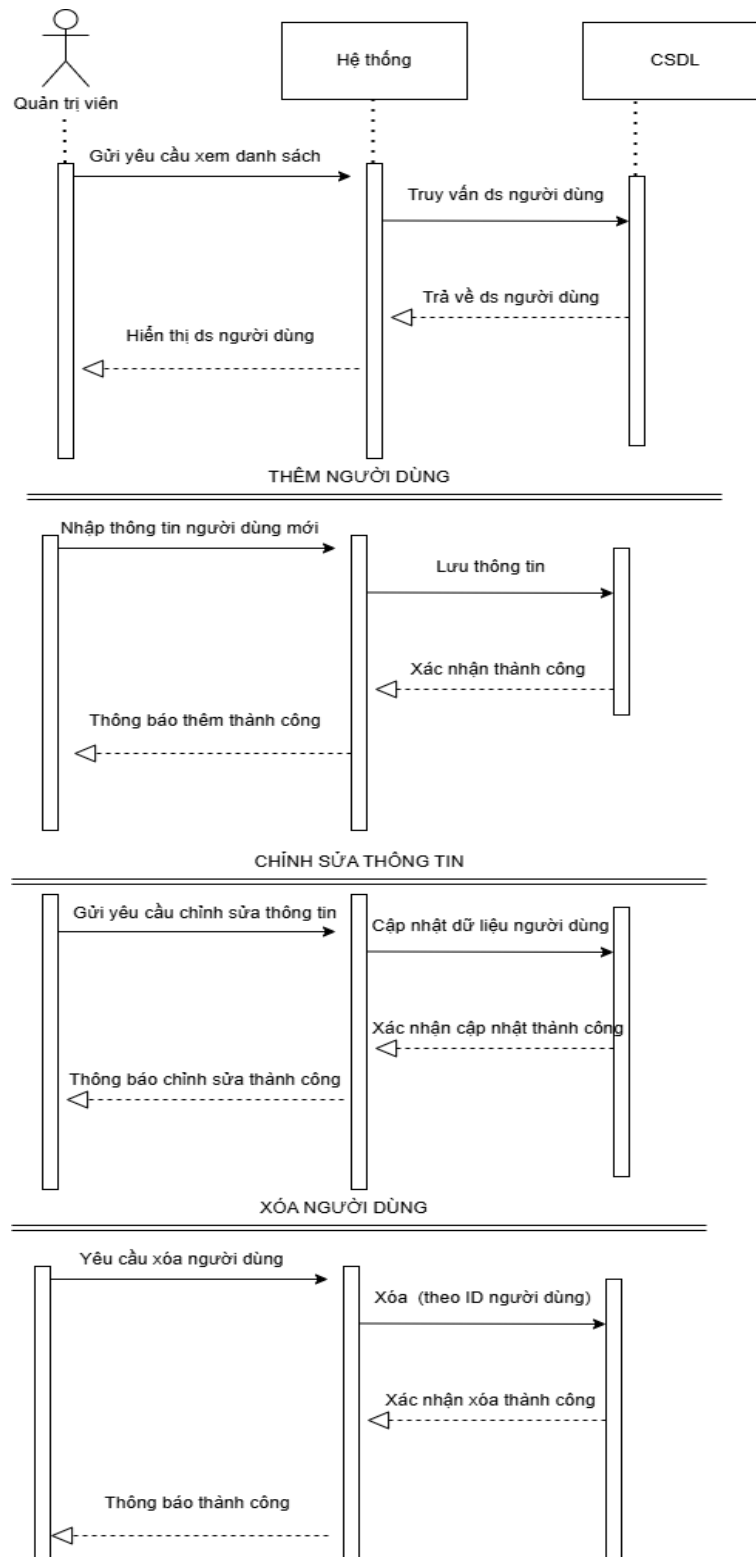
|  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|--|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | người dùng, Chinh sửa người dùng, Xóa người dùng và Xem danh sách người dùng. Thông qua các chức năng này, quản trị viên có thể tạo mới tài khoản cho người dùng, cập nhật các thông tin quan trọng như tên, email hoặc quyền hạn, xóa tài khoản, đồng thời xem danh sách toàn bộ người dùng để phục vụ công tác quản lý. Hệ thống đảm bảo mọi thao tác và thay đổi của quản trị viên đều được lưu trữ và cập nhật chính xác trong cơ sở dữ liệu, giúp quá trình quản lý người dùng diễn ra hiệu quả, minh bạch và an toàn. |
|--|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- Dòng chính Use Case quản lý tài khoản

| STT | Thực hiện | Hành động                                                                                               |
|-----|-----------|---------------------------------------------------------------------------------------------------------|
| 1   | Admin     | Đăng nhập vào hệ thống.                                                                                 |
| 2   | Hệ thống  | Xác thực quyền quản trị và cho phép truy cập chức năng quản lý người dùng.                              |
| 3   | Admin     | Chọn chức năng “Quản lý tài khoản”.                                                                     |
| 4   | Hệ thống  | Hiển thị các tùy chọn: Thêm người dùng, Chinh sửa người dùng, Xóa người dùng, Xem danh sách người dùng. |
| 5   | Admin     | Chọn một chức năng cần thực hiện.                                                                       |
| 6   | Hệ thống  | Tiếp nhận yêu cầu và xử lý theo chức năng được chọn.                                                    |
| 7   | Hệ thống  | Cập nhật thông tin vào cơ sở dữ liệu và hiển thị kết quả cho quản trị viên.                             |

Bảng 2.16 Dòng chính Use Case quản lý tài khoản

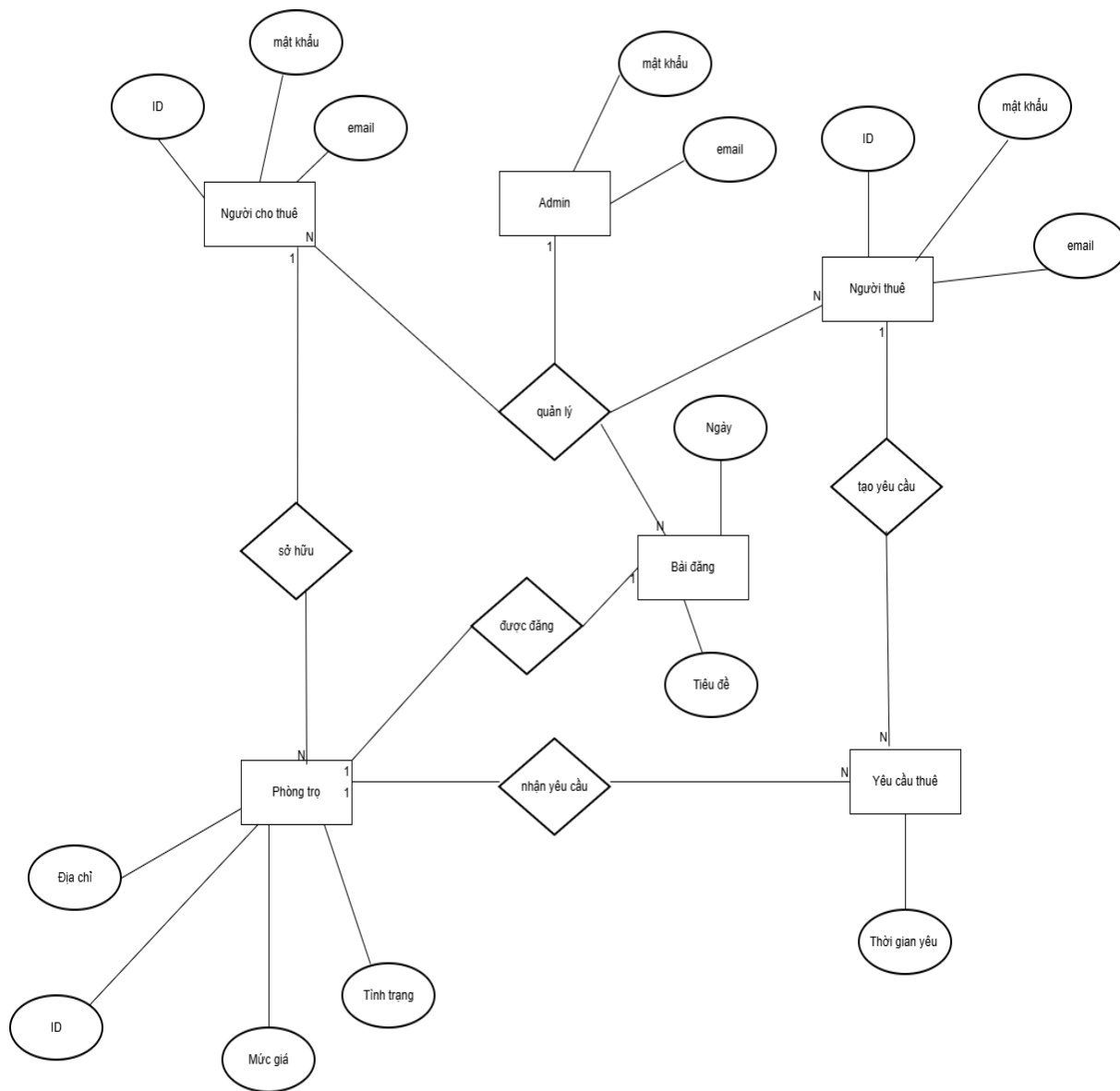
- Biểu đồ tuần tự quản lý tài khoản



Hình 2.17 Biểu đồ tuần tự quản lý tài khoản

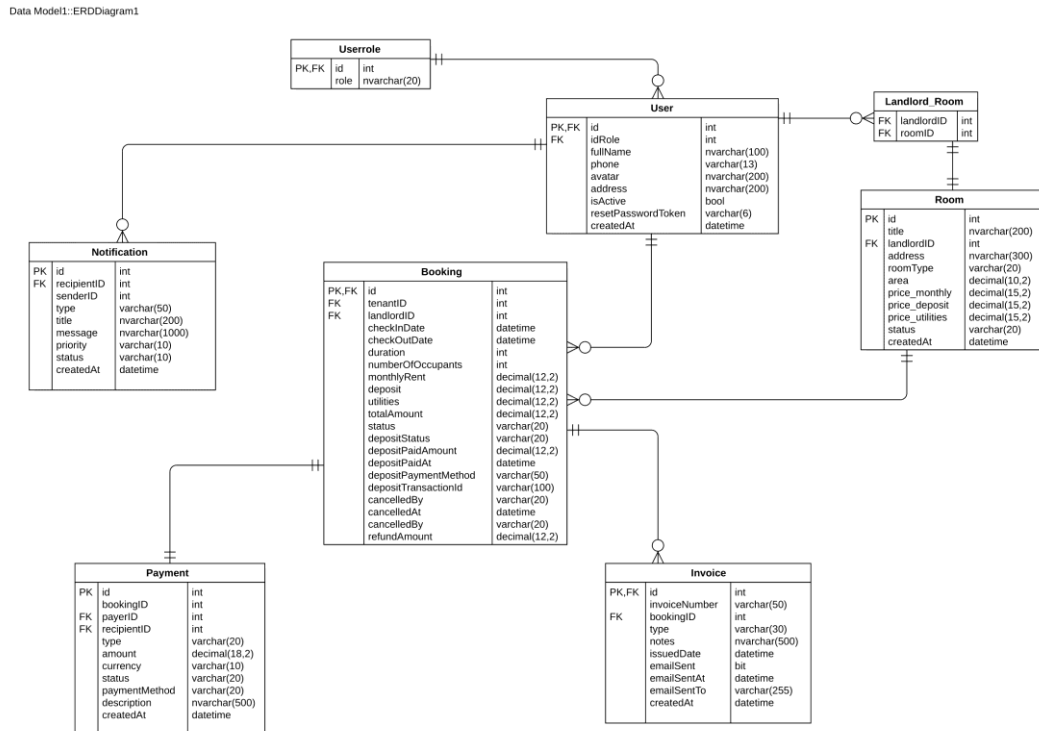
## 2.4. Thiết kế cơ sở dữ liệu

### 2.4.1. Mô hình thực thể liên kết



Hình 2.18 Sơ đồ thực thể liên kết

## 2.4.2. Mô hình quan hệ



Hình 2.19 Mô hình quan hệ

## 2.5. Danh mục các bảng

Bảng 2.17 Mô tả các bảng cơ sở dữ liệu

| STT | Tên bảng      | Mô tả                                                                                              |
|-----|---------------|----------------------------------------------------------------------------------------------------|
| 1   | Userrole      | Lưu trữ thông tin vai trò người dùng (admin, landlord, tenant...).                                 |
| 2   | User          | Lưu trữ thông tin người dùng: họ tên, số điện thoại, email, địa chỉ, avatar, trạng thái kích hoạt. |
| 3   | Landlord_Room | Liên kết giữa chủ phòng trọ và phòng trọ mà họ sở hữu.                                             |
| 4   | Room          | Lưu trữ thông tin phòng trọ: tiêu đề, địa chỉ, loại phòng, diện tích, giá                          |

|   |              |                                                                                                                             |
|---|--------------|-----------------------------------------------------------------------------------------------------------------------------|
|   |              | thuê, giá cọc, giá tiện ích, trạng thái.                                                                                    |
| 5 | Booking      | Lưu trữ thông tin đặt phòng: người thuê, chủ phòng, thời gian thuê, chi tiết chi phí, trạng thái đặt cọc, hủy, hoàn tiền... |
| 6 | Payment      | Lưu trữ thông tin thanh toán: số tiền, loại thanh toán, người gửi – người nhận, mã giao dịch, trạng thái.                   |
| 7 | Invoice      | Lưu trữ hóa đơn thanh toán: mã hóa đơn, ngày xuất, nội dung, người nhận email, trạng thái gửi email.                        |
| 8 | Notification | Lưu trữ thông báo gửi giữa các người dùng: tiêu đề, nội dung, mức độ ưu tiên, trạng thái.                                   |
| 9 | Review       | Lưu trữ đánh giá phòng: điểm đánh giá, loại đánh giá, bình luận, thời gian tạo.                                             |

## Bảng User

Bảng 2.18 Mô tả bảng User

| Tên trường      | Kiểu dữ liệu  | Null  | Key         | Ghi chú              |
|-----------------|---------------|-------|-------------|----------------------|
| <b>Id</b>       | int           | Không | Primary Key | Mã người dùng        |
| <b>idRole</b>   | int           | Có    | Foreign Key | Quyền người dùng     |
| <b>fullName</b> | nvarchar(100) | Có    |             | Họ tên               |
| <b>Phone</b>    | varchar(13)   | Có    |             | Số điện thoại        |
| <b>Avatar</b>   | nvarchar(200) | Có    |             | Ảnh đại diện         |
| <b>Address</b>  | nvarchar(200) | Có    |             | Địa chỉ              |
| <b>isActive</b> | boolean       | Có    |             | Trạng thái kích hoạt |

|                           |            |    |  |                   |
|---------------------------|------------|----|--|-------------------|
| <b>resetPasswordToken</b> | varchar(6) | Có |  | Mã reset password |
| <b>createdAt</b>          | datetime   | Có |  | Ngày tạo          |

### Bảng Userrole

Bảng 2.19 Mô tả bảng Userrole

| Tên trường    | Kiểu dữ liệu | Null  | Key         | Ghi chú          |
|---------------|--------------|-------|-------------|------------------|
| <b>Id</b>     | Int          | Không | Primary Key | Mã người dùng    |
| <b>idRole</b> | Int          | Có    | Foreign Key | Quyền người dùng |

### Bảng Room

Bảng 2.20 Mô tả bảng Room

| Tên trường             | Kiểu dữ liệu  | Null  | Key         | Ghi chú             |
|------------------------|---------------|-------|-------------|---------------------|
| <b>Id</b>              | int           | Không | Primary Key | Mã phòng            |
| <b>landlordID</b>      | int           | Có    | Foreign Key | Chủ phòng           |
| <b>Title</b>           | nvarchar(200) | Có    |             | Tiêu đề phòng       |
| <b>Address</b>         | nvarchar(300) | Có    |             | Địa chỉ             |
| <b>roomType</b>        | varchar(20)   | Có    |             | Loại phòng          |
| <b>Area</b>            | decimal(10,2) | Có    |             | Diện tích           |
| <b>price_monthly</b>   | decimal(15,2) | Có    |             | Giá thuê theo tháng |
| <b>price_deposit</b>   | decimal(15,2) | Có    |             | Tiền cọc            |
| <b>price_utilities</b> | decimal(15,2) | Có    |             | Giá dịch vụ         |
| <b>Status</b>          | varchar(20)   | Có    |             | Trạng thái          |
| <b>createdAt</b>       | datetime      | Có    |             | Ngày tạo            |

### Bảng Booking

Bảng 2.21 Mô tả bảng Booking

| Tên trường      | Kiểu dữ liệu | Null  | Key         | Ghi chú      |
|-----------------|--------------|-------|-------------|--------------|
| <b>Id</b>       | int          | Không | Primary Key | Mã đặt phòng |
| <b>tenantID</b> | int          | Có    | Foreign Key | Người thuê   |

|                             |               |    |             |                            |
|-----------------------------|---------------|----|-------------|----------------------------|
| <b>landlordID</b>           | int           | Có | Foreign Key | Chủ nhà                    |
| <b>roomID</b>               | int           | Có | Foreign Key | Phòng                      |
| <b>checkinDate</b>          | datetime      | Có |             | Ngày bắt đầu               |
| <b>checkoutDate</b>         | datetime      | Có |             | Ngày kết thúc              |
| <b>duration</b>             | int           | Có |             | Số tháng/tuần thuê         |
| <b>numberOfOccupants</b>    | int           | Có |             | Số người                   |
| <b>monthlyRent</b>          | decimal(12,2) | Có |             | Tiền thuê                  |
| <b>deposit</b>              | decimal(12,2) | Có |             | Đặt cọc                    |
| <b>utilities</b>            | decimal(12,2) | Có |             | Dịch vụ                    |
| <b>totalAmount</b>          | decimal(12,2) | Có |             | Tổng tiền                  |
| <b>status</b>               | varchar(20)   | Có |             | Trạng thái                 |
| <b>depositStatus</b>        | varchar(20)   | Có |             | TT đặt cọc                 |
| <b>depositPaidAmount</b>    | decimal(12,2) | Có |             | Số tiền đã cọc             |
| <b>depositPaidAt</b>        | datetime      | Có |             | Ngày thanh toán cọc        |
| <b>depositPaymentMethod</b> | varchar(50)   | Có |             | Phương thức thanh toán cọc |
| <b>depositTransactionId</b> | varchar(100)  | Có |             | Mã giao dịch               |
| <b>cancelledBy</b>          | varchar(20)   | Có |             | Người hủy                  |
| <b>cancelledAt</b>          | datetime      | Có |             | Thời gian hủy              |
| <b>refundAmount</b>         | decimal(12,2) | Có |             | Số tiền hoàn cọc           |

## Bảng Payment

Bảng 2.22 Mô tả bảng Payment

| Tên trường | Kiểu dữ liệu | Null  | Key         | Ghi chú       |
|------------|--------------|-------|-------------|---------------|
| <b>Id</b>  | int          | Không | Primary Key | Mã thanh toán |

|                      |               |    |             |                  |
|----------------------|---------------|----|-------------|------------------|
| <b>bookingID</b>     | int           | Có | Foreign Key | Mã đặt phòng     |
| <b>payerID</b>       | int           | Có | Foreign Key | Người thanh toán |
| <b>recipientID</b>   | int           | Có | Foreign Key | Người nhận       |
| <b>Type</b>          | varchar(20)   | Có |             | Loại thanh toán  |
| <b>Amount</b>        | decimal(18,2) | Có |             | Số tiền          |
| <b>Currency</b>      | varchar(10)   | Có |             | Đơn vị tiền      |
| <b>Status</b>        | varchar(20)   | Có |             | Trạng thái       |
| <b>paymentMethod</b> | varchar(20)   | Có |             | Phương thức      |
| <b>description</b>   | nvarchar(500) | Có |             | Mô tả            |
| <b>createdAt</b>     | datetime      | Có |             | Ngày tạo         |

## Bảng Invoice

Bảng 2.23 Mô tả bảng Invoice

| Tên trường           | Kiểu dữ liệu  | Null  | Key         | Ghi chú          |
|----------------------|---------------|-------|-------------|------------------|
| <b>Id</b>            | int           | Không | Primary Key | Mã hóa đơn       |
| <b>invoiceNumber</b> | varchar(50)   | Có    |             | Số hóa đơn       |
| <b>bookingID</b>     | int           | Có    | Foreign Key | Đặt phòng        |
| <b>Type</b>          | varchar(30)   | Có    |             | Loại hóa đơn     |
| <b>Notes</b>         | nvarchar(500) | Có    |             | Ghi chú          |
| <b>issuedDate</b>    | datetime      | Có    |             | Ngày xuất        |
| <b>emailSent</b>     | boolean       | Có    |             | Đã gửi email?    |
| <b>emailSentAt</b>   | datetime      | Có    |             | Thời gian gửi    |
| <b>emailSentTo</b>   | varchar(255)  | Có    |             | Email người nhận |
| <b>createdAt</b>     | datetime      | Có    |             | Ngày tạo         |

## Bảng Notification

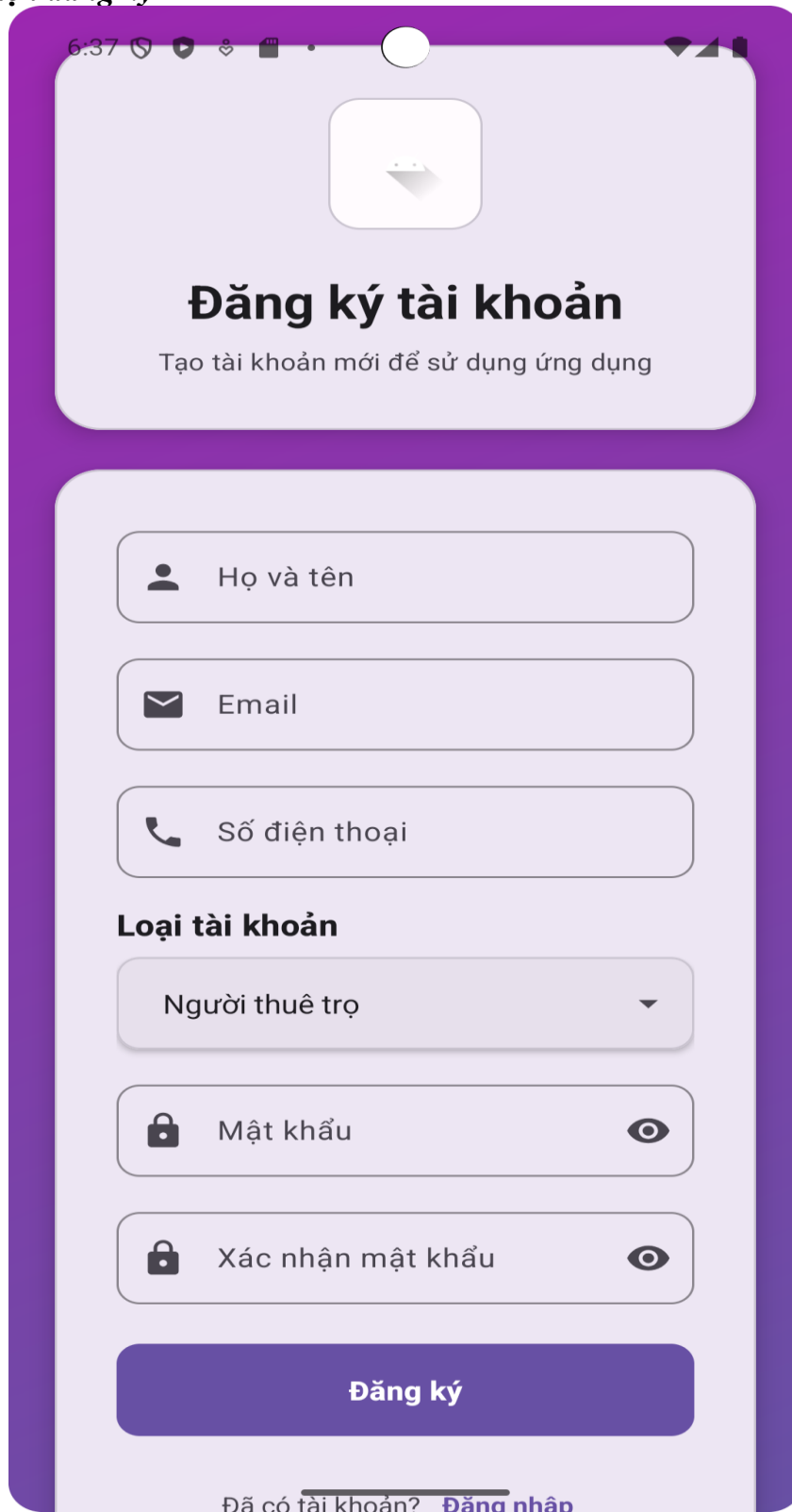
Bảng 2.25 Mô tả bảng Notification

| Tên trường         | Kiểu dữ liệu | Null  | Key         | Ghi chú        |
|--------------------|--------------|-------|-------------|----------------|
| <b>Id</b>          | int          | Không | Primary Key | Mã thông báo   |
| <b>recipientID</b> | int          | Có    | Foreign Key | Người nhận     |
| <b>senderID</b>    | int          | Có    | Foreign Key | Người gửi      |
| <b>type</b>        | varchar(50)  | Có    |             | Loại thông báo |

|                  |                |    |  |            |
|------------------|----------------|----|--|------------|
| <b>title</b>     | nvarchar(200)  | Có |  | Tiêu đề    |
| <b>message</b>   | nvarchar(1000) | Có |  | Nội dung   |
| <b>priority</b>  | varchar(10)    | Có |  | Ưu tiên    |
| <b>status</b>    | varchar(10)    | Có |  | Trạng thái |
| <b>createdAt</b> | datetime       | Có |  | Ngày tạo   |

## 2.6. Thiết kế giao diện

### 2.6.1. Giao diện đăng ký

A mobile application registration screen with a purple header and footer. The main content area is white with rounded corners. At the top, there's a status bar showing the time 6:37 and various icons. Below the status bar is a white square icon with a stylized house. The title "Đăng ký tài khoản" is in bold black text, followed by the subtitle "Tạo tài khoản mới để sử dụng ứng dụng". There are four input fields: "Họ và tên" (with a person icon), "Email" (with an envelope icon), "Số điện thoại" (with a phone icon), and "Loại tài khoản" (a dropdown menu with "Người thuê trọ" selected). Below these are two password fields: "Mật khẩu" and "Xác nhận mật khẩu", both with lock icons and toggle eyes. A large purple button labeled "Đăng ký" is at the bottom. At the very bottom, there's a link "Đã có tài khoản? Đăng nhập".

6:37

**Đăng ký tài khoản**

Tạo tài khoản mới để sử dụng ứng dụng

Họ và tên

Email

Số điện thoại

**Loại tài khoản**

Người thuê trọ

Mật khẩu

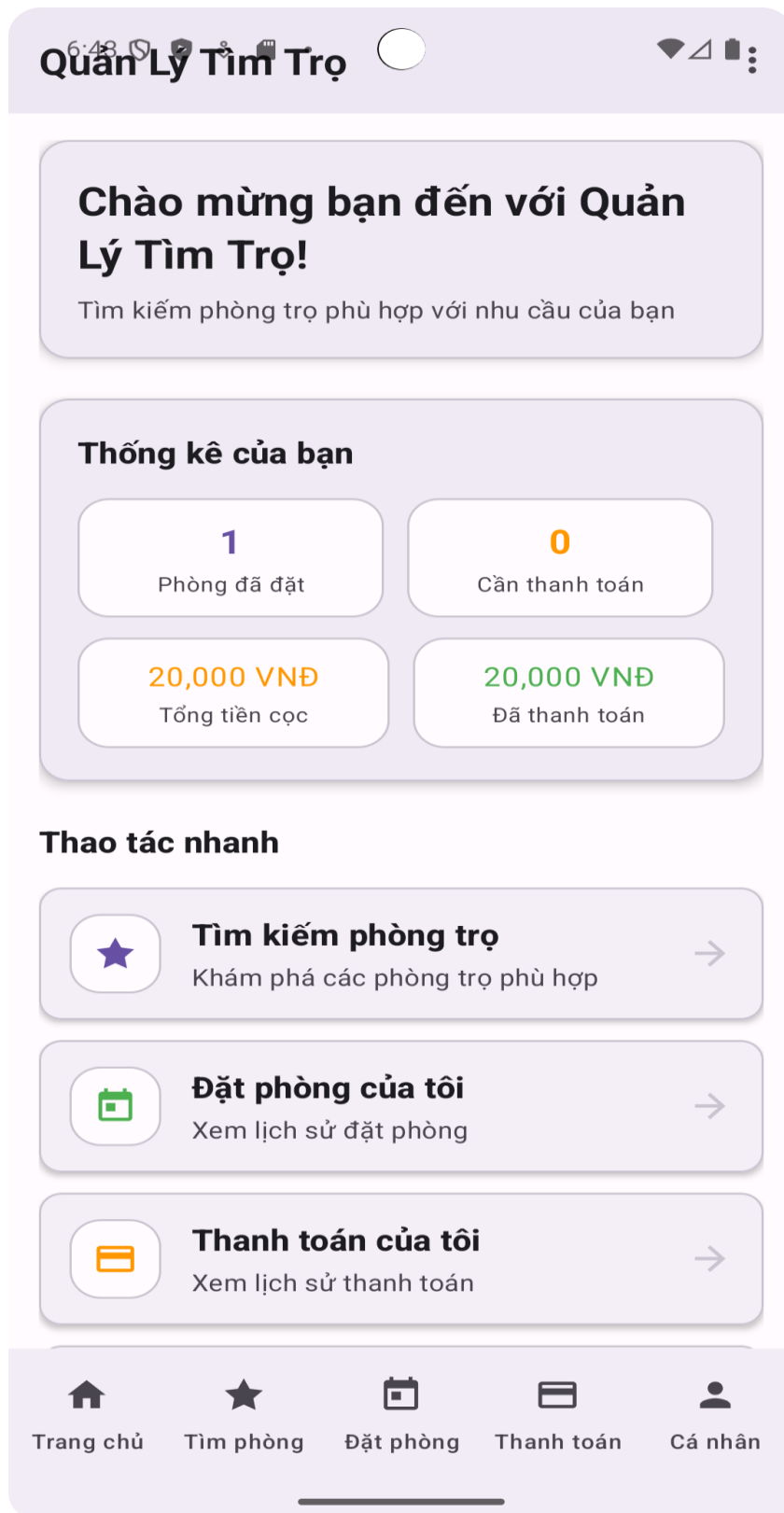
Xác nhận mật khẩu

**Đăng ký**

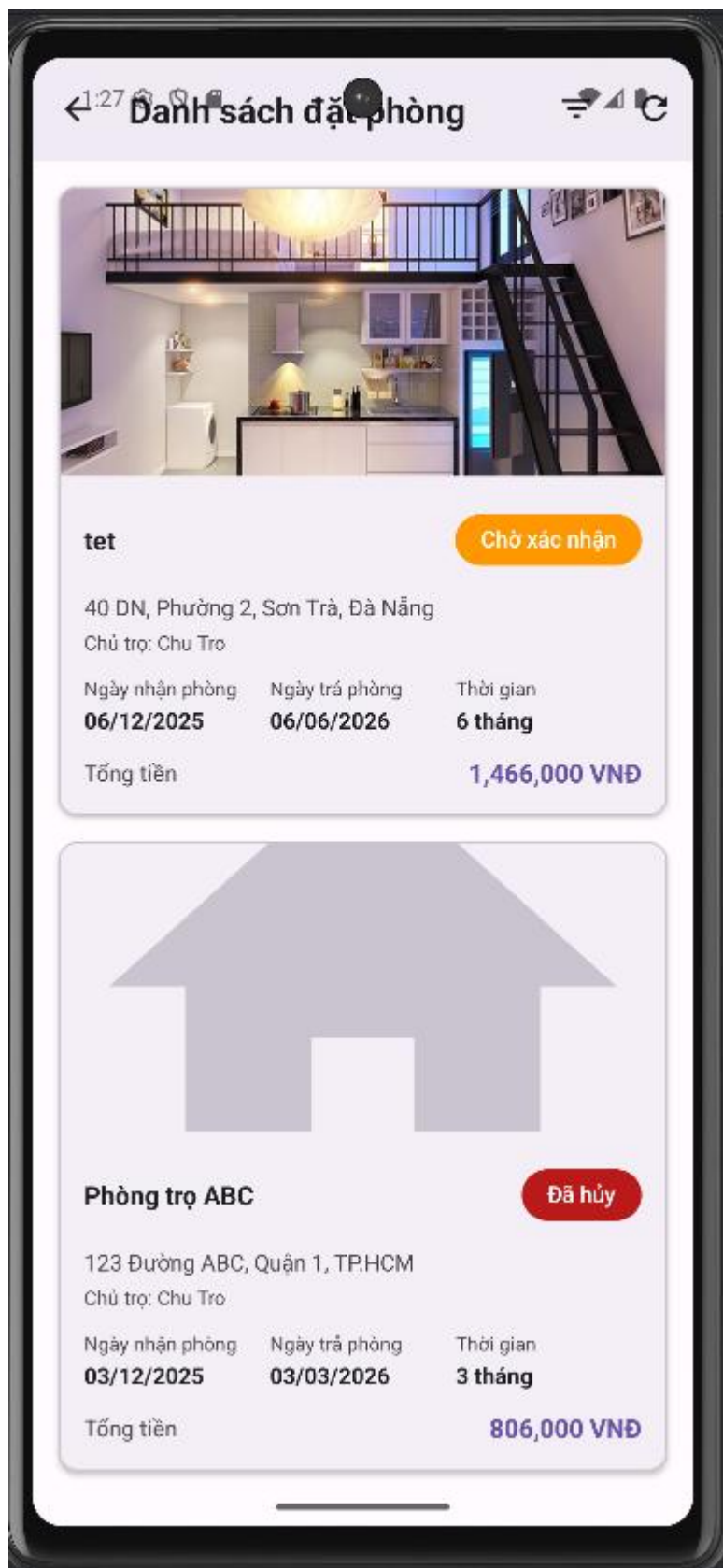
Đã có tài khoản? [Đăng nhập](#)

Hình 2.20 Màn hình Login

### 2.6.2. Giao diện người thuê



Hình 2.21 Màn hình người thuê

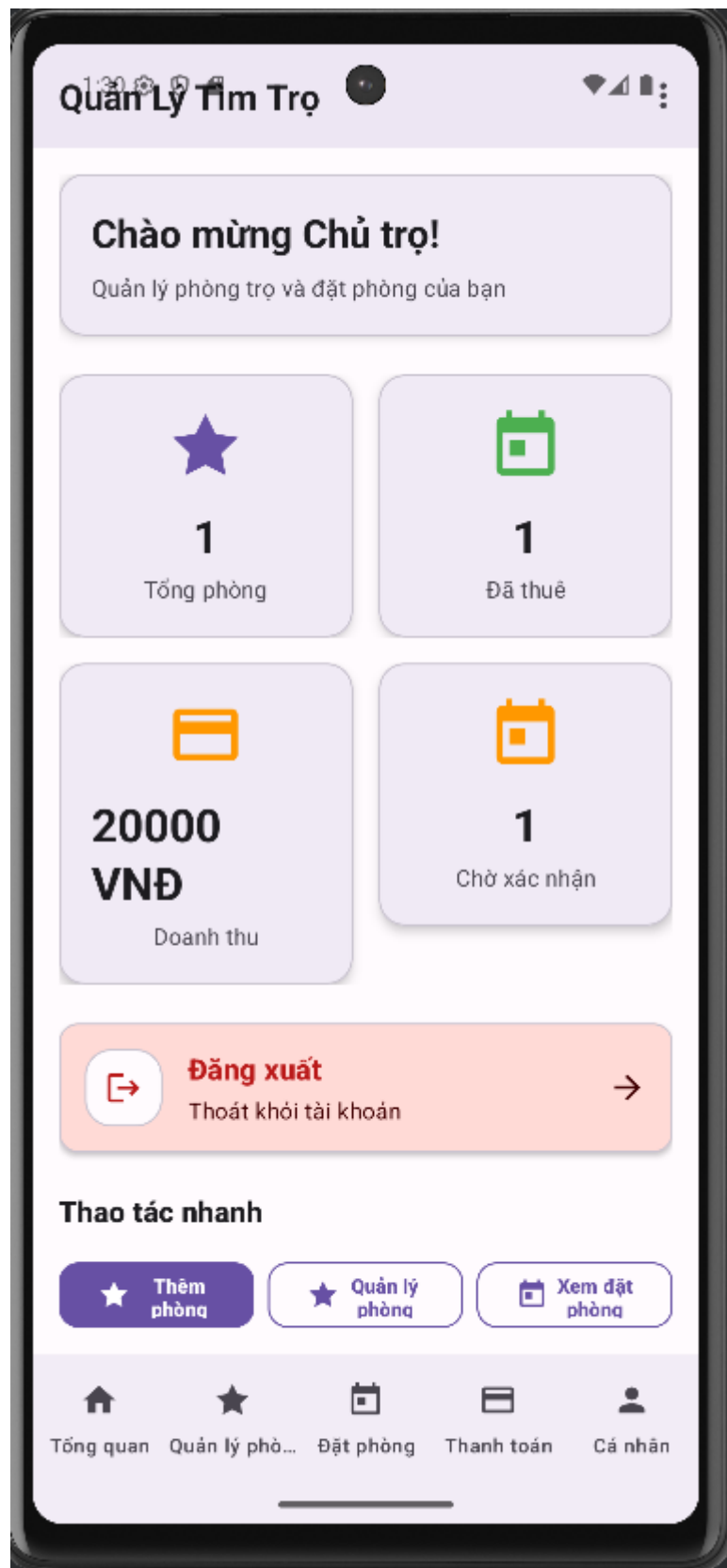


Hình 2.22 Giao diện danh sách đặt phòng



Hình 2.23 Giao diện tìm kiếm

### 2.6.3. Giao diện người cho thuê



Hình 2.24 Giao diện người cho thuê

← 1:32 Thêm phòng trọ

**Thông tin cơ bản**

Tiêu đề phòng trọ \*

Mô tả chi tiết

Loại phòng ▼

Diện tích (m²)

**Địa chỉ**

Thành phố \* ▼

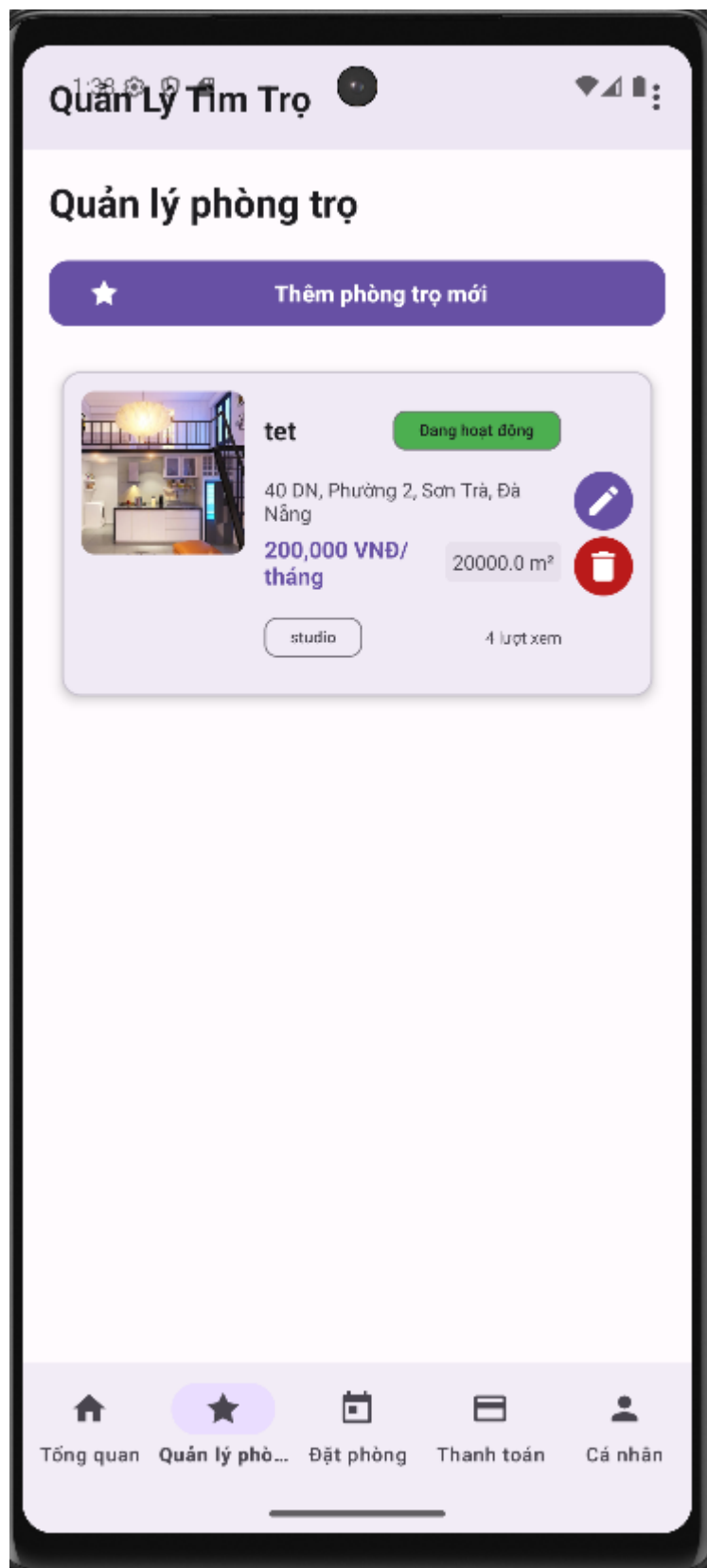
Quận/Huyện ▼

Phường/Xã ▼

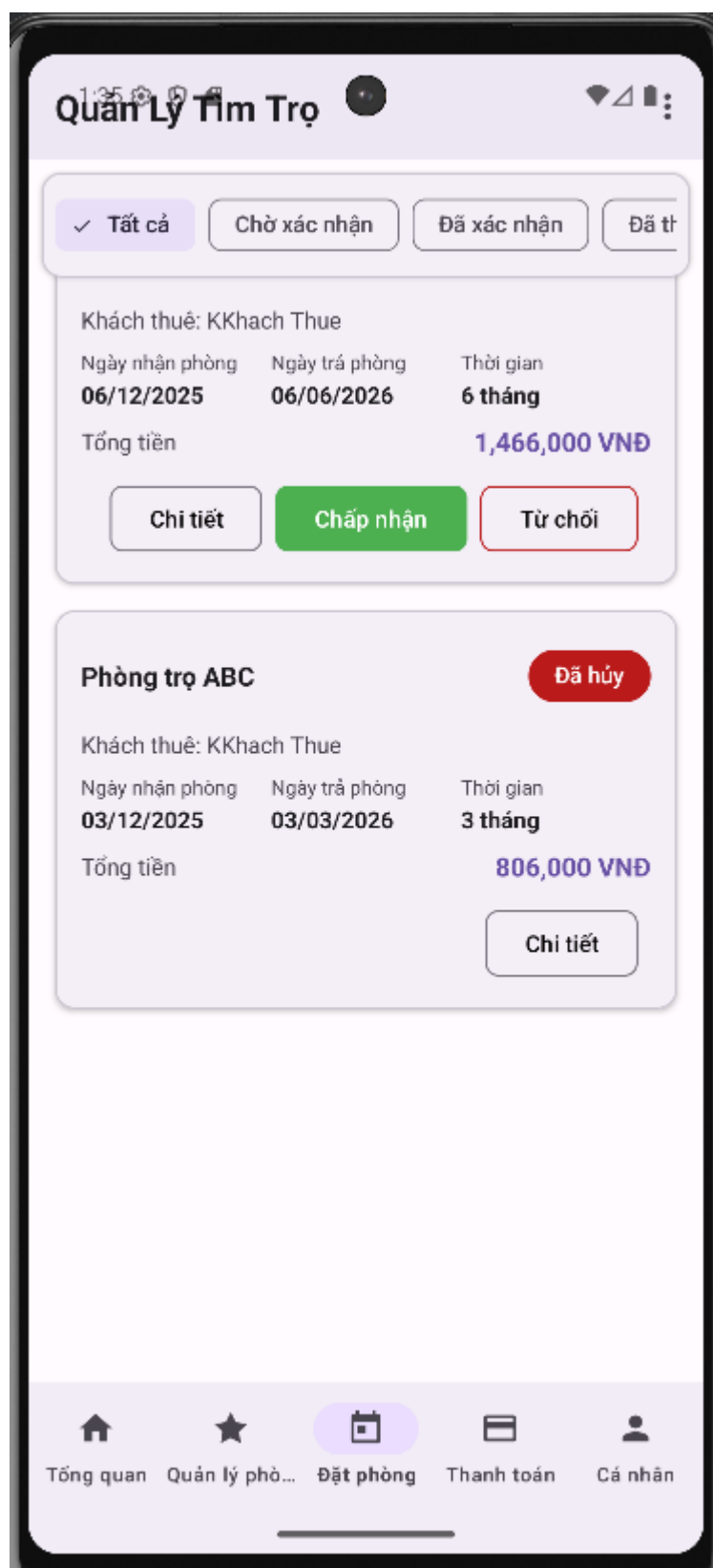
Số nhà, tên đường

**Giá cả**

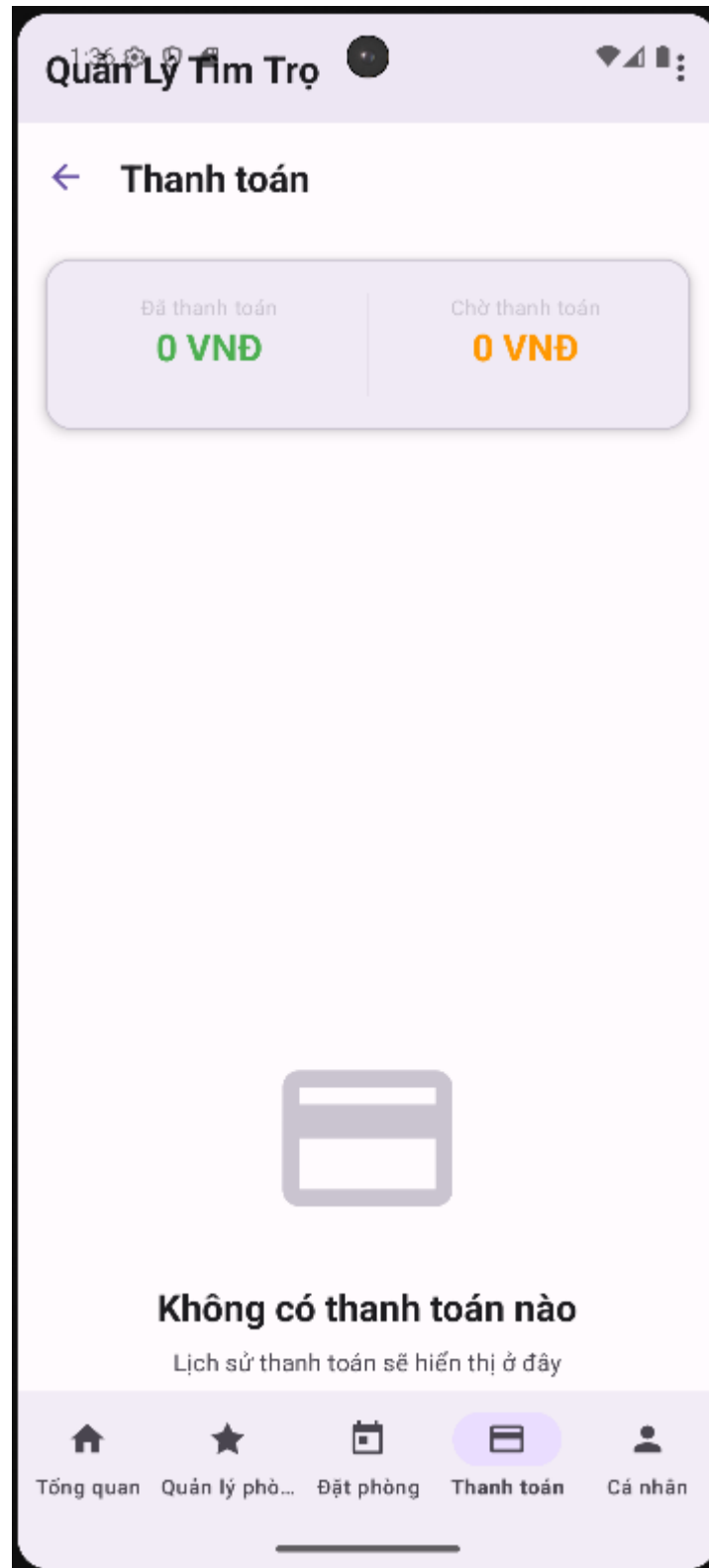
Hình 2.25 Giao diện đăng bài



Hình 2.26 Giao diện quản lý bài đăng

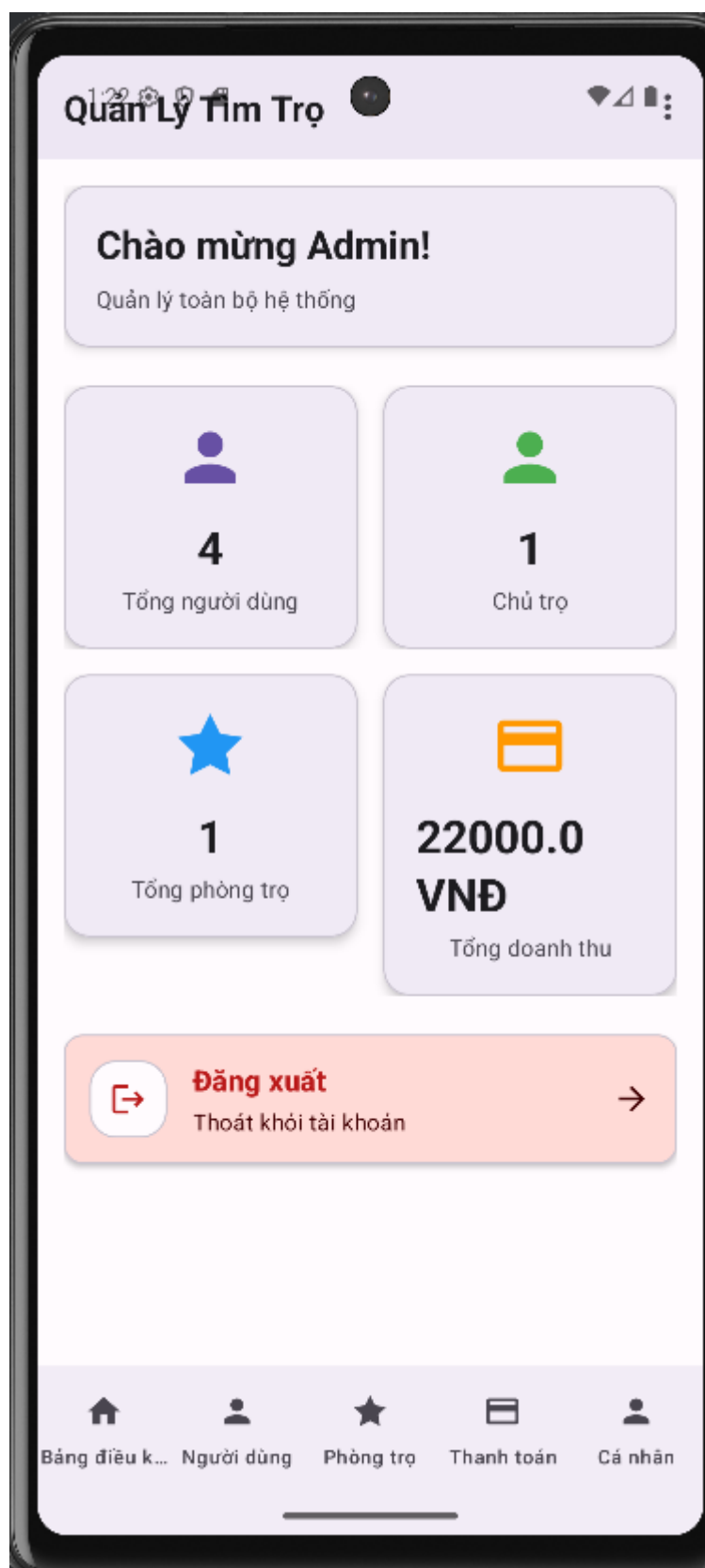


Hình 2.27 Giao diện quản lý các yêu cầu

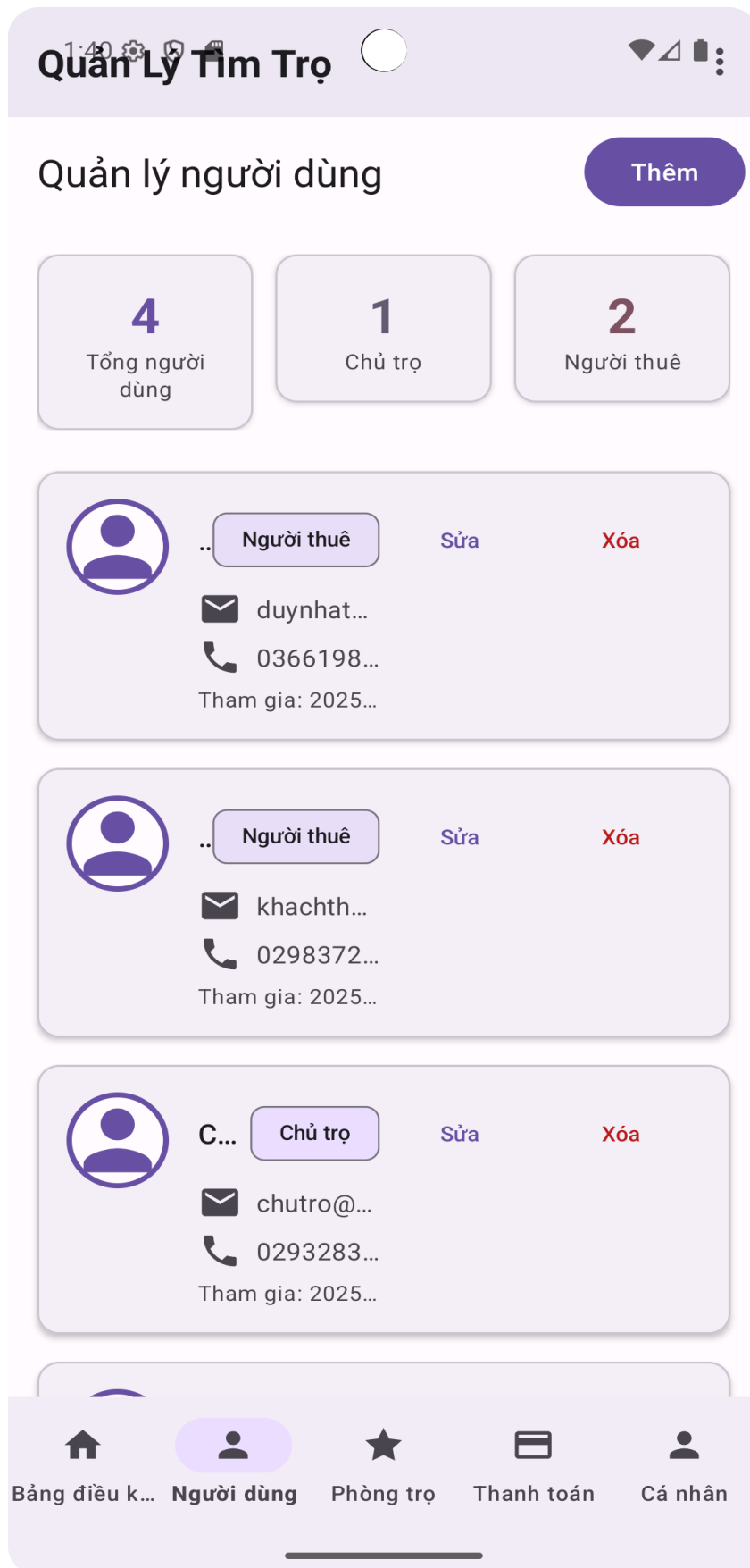


Hình 2.28 Giao diện quản lý các thanh toán

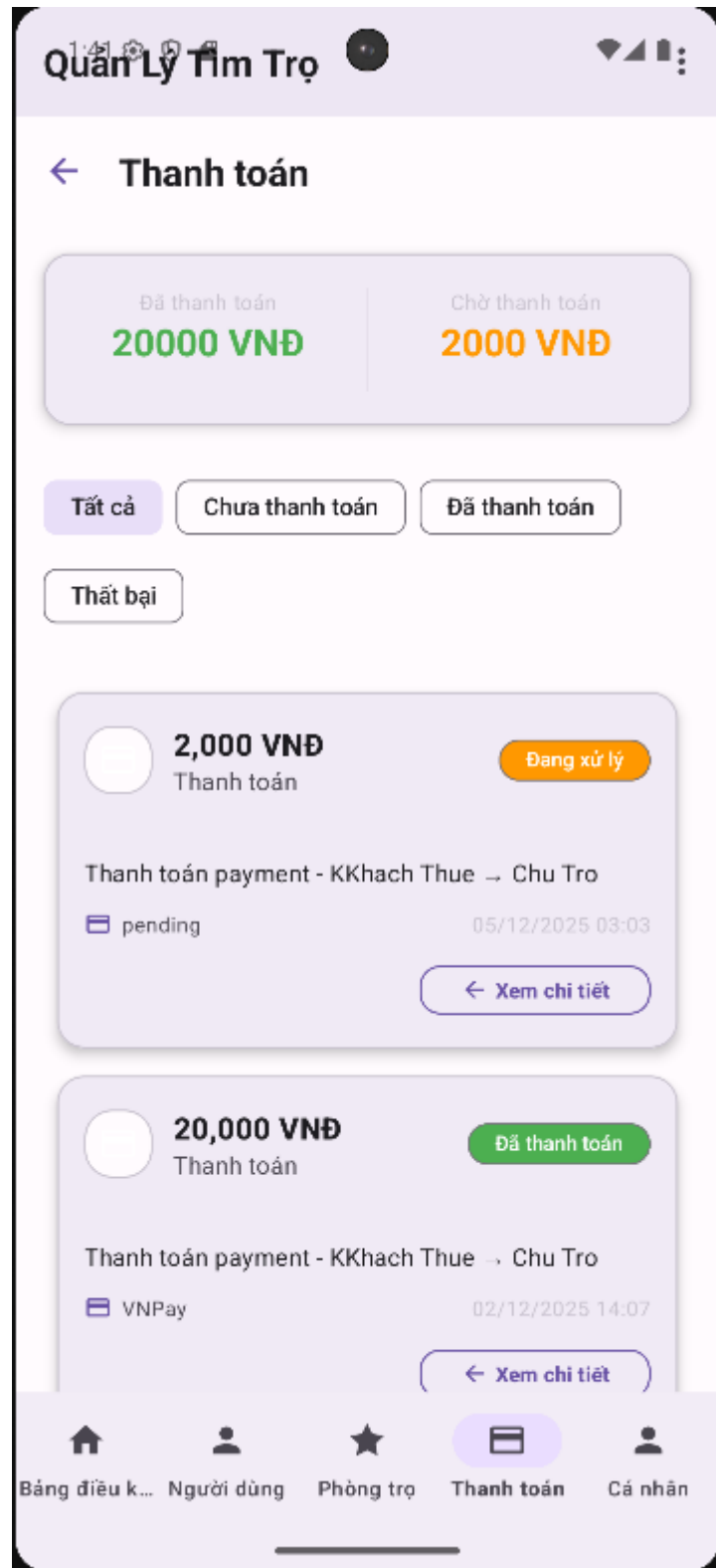
#### 2.6.4. Giao diện Admin



Hình 2.29 Màn hình Admin



Hình 2.30 Giao diện quản lý người dùng



Hình 2.30 Giao diện thống kê thanh toán của hệ thống

## Chương 3: CÀI ĐẶT VÀ PHÁT TRIỂN ỨNG DỤNG

### 3.1. Cấu trúc thư mục code

#### 3.1.1. Cấu trúc thư mục code Back End

|                       |                                     |
|-----------------------|-------------------------------------|
| backend/              | # Mã nguồn chính                    |
| ├── .env              | # Biến môi trường (bảo mật)         |
| ├── .gitignore        | # Cấu hình Git bỏ qua file          |
| ├── config.env        | # Tập cấu hình môi trường           |
| ├── package.json      | # Thông tin dự án & dependency      |
| ├── package-lock.json | # Lock file cho npm dependency      |
| ├── server.js         | # Điểm khởi tạo chính của server    |
|                       |                                     |
| ├── middleware/       | # Các middleware xử lý request      |
| └── auth.js           | # Xác thực & phân quyền người dùng  |
| └── validation.js     | # Kiểm tra & xác thực dữ liệu input |
|                       |                                     |
| ├── models/           | # Định nghĩa schema & ORM model     |
| └── Booking.js        | # Model đặt phòng                   |
| └── Notification.js   | # Model thông báo                   |
| └── Payment.js        | # Model thanh toán                  |
| └── Review.js         | # Model đánh giá                    |
| └── Room.js           | # Model phòng                       |
| └── User.js           | # Model người dùng                  |
|                       |                                     |
| ├── routes/           | # Định nghĩa các endpoint API       |
| └── auth.js           | # API xác thực & đăng ký            |
| └── bookings.js       | # API quản lý đặt phòng             |
| └── notifications.js  | # API quản lý thông báo             |
| └── payments.js       | # API quản lý thanh toán            |
| └── rooms.js          | # API quản lý phòng                 |
| └── statistics.js     | # API thống kê                      |

```
|   └─ users.js                # API quản lý người dùng
|
|   └─ scripts/                # Script tiện ích
|       └─ createAdmin.js      # Script tạo tài khoản admin
|       └─ ensureUploadDirs.js # Script tạo thư mục upload
|
|   └─ uploads/                # Thư mục lưu file tải lên
|       └─ rooms/              # Hình ảnh phòng
|       └─ temp/               # File tạm thời
|       └─ users/              # Avatar & hình ảnh người dùng
|
|   └─ utils/                  # Hàm tiện ích & service
|       └─ emailService.js     # Service gửi email
|       └─ vnpayService.js     # Service thanh toán VNPay
|
|   └─ node_modules/           # Các package dependencies (tự động)
```

### **3.1.2. Cấu trúc thư mục code FrontEnd**

```
quanlyphongtro/
|
|   └─ appquanlytimtro/        # Android App (Gradle Project)
|       └─ build.gradle.kts    # Build configuration cho toàn project
|       └─ settings.gradle.kts # Settings gradle
|       └─ gradle.properties   # Gradle properties
|       └─ gradlew              # Gradle wrapper script (Linux/Mac)
|       └─ gradlew.bat          # Gradle wrapper script (Windows)
|       └─ local.properties    # Local build properties
|       └─ output.txt          # Build output log
|       └─ tree.txt             # Project structure file
|
|   └─ gradle/                 # Gradle wrapper directory
|       └─ wrapper/
|           └─ gradle-wrapper.properties # Gradle version configuration
|
|   └─
```

```
|   └── app/                                # Main Android application module
|       ├── build.gradle.kts                # App-specific build configuration
|       ├── proguard-rules.pro              # ProGuard rules for code obfuscation
|       └── build/                           # Build output directory
|           ├── generated/                   # Auto-generated files
|               ├── ap_generated_sources/    # Annotation processor sources
|               ├── res/                     # Generated resources
|               └── source/                   # Generated source files
|           └── intermediates/                # Intermediate build artifacts
|               ├── aar_metadata_check/
|               ├── annotation_processor_list/
|               ├── apk/                     # APK files
|               ├── assets/                  # App assets
|               ├── dex/                     # DEX files (compiled bytecode)
|               ├── javac/                   # Compiled Java files
|               ├── lint_vital_*/            # Lint check outputs
|               └── ... (other build intermediates)
|   └──
```

### **3.2.Chức năng cần phát triển sắp tới**

#### **3.2.1.Chức năng đặt lịch khi thanh toán tiền cọc**

- **Vấn đề:**

Trong phiên bản hiện tại, hệ thống chỉ ghi nhận giao dịch thanh toán tiền cọc của người thuê trọ, chưa có chức năng đặt lịch hẹn để nhận phòng hoặc hoàn tất thủ tục ký hợp đồng. Điều này gây bất tiện cho cả chủ trọ lẫn người thuê, vì hai bên phải liên hệ thủ công để thống nhất thời gian. Ngoài ra, không có công cụ kiểm tra trùng lịch hoặc thông báo nhắc lịch, dễ dẫn đến bỏ sót hoặc trùng hẹn.

- **Cách giải quyết:**

Để khắc phục, hệ thống sẽ được phát triển thêm khả năng tự động mở chức năng đặt lịch sau khi thanh toán thành công, cho phép người thuê chọn ngày giờ phù hợp. Lịch hẹn sẽ được ghi vào hệ thống, gửi thông báo cho chủ trọ và hỗ trợ xác nhận hoặc đề xuất lại thời gian. Hệ thống cũng cần bổ sung tính năng nhắc lịch, kiểm tra trùng lịch và cho phép chỉnh sửa hoặc hủy lịch khi cần thiết.

- **Modul code (dự kiến phát triển):**

1. Module hiển thị giao diện đặt lịch sau thanh toán

- Tự động hiển thị popup/ màn hình chọn ngày – giờ sau khi thanh toán thành công.
- Tiền xử lý dữ liệu: mã người thuê, mã phòng, mã giao dịch.

2. Module xử lý đặt lịch (Booking Service)

- API tạo lịch hẹn:
  - `/api/booking/create`
- Kiểm tra trùng giờ giữa các lịch đã có.
- Ghi lịch vào database, trạng thái “Chờ xác nhận”.

3. Module đồng bộ với chủ trọ

- Gửi thông báo đến chủ trọ (notification/email).
- API xác nhận hoặc từ chối lịch:
  - `/api/booking/approve`
  - `/api/booking/reject`

4. Module nhắc lịch

- Cron job hoặc scheduler kiểm tra các lịch sắp tới.
- Gửi thông báo nhắc người thuê và chủ trọ trước thời điểm hẹn.

5. Module quản lý lịch hẹn

- Người thuê: xem lịch, chỉnh sửa, hủy lịch.
- Chủ trọ: xem lịch theo ngày/tuần/tháng, lọc theo phòng.

### **3.2.2. Chức năng đánh giá bài đăng trọ**

- **Vấn đề:**

Hiện tại hệ thống chỉ hiển thị thông tin bài đăng trọ nhưng chưa hỗ trợ người thuê để lại đánh giá hoặc nhận xét về phòng trọ sau khi đã thuê. Điều này khiến người dùng mới không có nguồn tham khảo đáng tin cậy, đồng thời chủ trọ cũng không nhận được phản hồi để cải thiện chất lượng dịch vụ. Việc thiếu đánh giá cũng làm giảm mức độ minh bạch và độ tin cậy của nền tảng.

- **Cách giải quyết:**

Hệ thống cần bổ sung chức năng **đánh giá và nhận xét bài đăng trọ**, cho phép người thuê sau khi sử dụng dịch vụ để lại điểm đánh giá (rating) và bình luận. Chức năng này cần đảm bảo:

- Chỉ người đã từng thuê phòng mới được quyền đánh giá (tránh spam).
- Cho phép đánh giá bằng số sao (1–5 sao) và nhập bình luận chi tiết.
- Tổng hợp điểm trung bình và hiển thị dưới bài đăng.
- Chủ trọ có thể phản hồi bình luận để tăng tính tương tác.
- Hệ thống kiểm duyệt bình luận để loại bỏ các nội dung vi phạm.

Chức năng này sẽ giúp tăng độ minh bạch, hỗ trợ người thuê đưa ra lựa chọn chính xác, đồng thời nâng cao uy tín của nền tảng.

- **Modul code (dự kiến phát triển):**

### 1. Module giao diện đánh giá

- Form nhập đánh giá (rating + bình luận).
- Nút gửi đánh giá chỉ xuất hiện khi người dùng đã hoàn tất thuê phòng.
- Hiển thị danh sách đánh giá theo thời gian.

### 2. Module xử lý đánh giá (Rating Service)

- API tạo đánh giá:
  - POST /api/rating/create
  - Kiểm tra quyền đánh giá (chỉ người từng thuê).
- API cập nhật đánh giá:
  - PUT /api/rating/update/{id}
- API xóa đánh giá:
  - DELETE /api/rating/delete/{id}

### 3. Module tính điểm trung bình

- Tự động tính điểm trung bình dựa trên tất cả rating.
- Cập nhật điểm trung bình vào bảng bài đăng.
- Hiển thị điểm tổng hợp trên giao diện bài đăng.

#### **4. Module phản hồi của chủ trọ**

- API chủ trọ phản hồi đánh giá:
  - POST /api/rating/reply/{ratingId}
- Hiển thị phản hồi bên dưới bình luận người dùng.

#### **5. Module kiểm duyệt**

- Lọc từ ngữ không phù hợp.
- Cho phép admin ẩn hoặc xóa đánh giá vi phạm.

#### **6. Module hiển thị danh sách đánh giá**

- Lọc đánh giá theo:
  - Mới nhất
  - Nhiều sao nhất
  - Ít sao nhất
- Phân trang danh sách đánh giá.

## KẾT LUẬN

Trên đây là toàn bộ báo cáo đồ án tốt nghiệp. Như vậy, sau một thời gian tìm hiểu, nghiên cứu thì em đã hoàn thành đồ án tốt nghiệp “Xây dựng ứng dụng đăng bài, tìm kiếm tro sử dụng Java ,ExpressJS và MongoDB”.

### I. KẾT QUẢ ĐẠT ĐƯỢC

- **Về công nghệ**
  - Hiểu rõ hơn và sử dụng thành thạo Java, ExpressJS và MongoDB trong quá trình xây dựng ứng dụng.
  - Nắm được cách thiết kế API, quản lý dữ liệu NoSQL và tổ chức kiến trúc backend.
  - Áp dụng hiệu quả các kiến thức về lập trình ứng dụng vào việc xây dựng đồ án.
- **Về chương trình**
  - Hoàn thiện các chức năng chính theo đúng mục tiêu ban đầu của đồ án.
  - Giao diện đơn giản, dễ sử dụng, hỗ trợ tốt cho người dùng trong việc đăng bài và tìm kiếm tro.
  - Đáp ứng được nhu cầu cơ bản của một ứng dụng tìm kiếm và quản lý phòng trọ trực tuyến.

### II. HẠN CHẾ

Mặc dù đã đạt được những kết quả tích cực, em nhận thức rõ rằng đồ án còn tồn tại một số hạn chế cần được cải thiện trong tương lai. Dưới đây là một số điểm hạn chế mà em đã nhận diện:

- **Hiệu suất:** Có thể cần tối ưu hóa hiệu suất của hệ thống để đảm bảo ứng dụng chạy mượt mà, đặc biệt là khi có số lượng người dùng lớn.
- **Mở rộng tính năng:** Cần xem xét và phát triển thêm các tính năng mới để làm cho ứng dụng trở nên đa dạng hơn và đáp ứng được nhiều nhu cầu sử dụng hơn.

### III. HƯỚNG PHÁT TRIỂN

Dựa vào những hạn chế đã đề cập, dưới đây là những hướng phát triển mà tôi đề xuất để nâng cao chất lượng và khả năng sử dụng của đồ án:

- **Tối ưu hóa hiệu suất:** Nghiên cứu và áp dụng các kỹ thuật tối ưu hóa mã nguồn và cơ sở dữ liệu để cải thiện hiệu suất hệ thống.

- Mở rộng tính năng: Nghiên cứu và thêm các tính năng mới như đặt lịch hẹn, đánh giá.
- Hỗ trợ đa nền tảng: Phát triển thành ứng dụng để có thể chạy trên nhiều nền tảng, mở rộng đối tượng người dùng.

## TÀI LIỆU THAM KHẢO

- [1] <https://fr.scribd.com/document/556890138/Xay-D%E1%BB%B1ng-Website-Tim-Ki%E1%BA%BFm-Phong-Tr%E1%BB%8D>
- [2]<https://github.com/0xl4p/Giao-Trinh-PTIT/blob/main/Gi%C3%A1o%20tr%C3%ACnh%20ph%C3%A2n%20t%C3%ADch%20v%C3%A0%20thi%E1%BA%BFt%20k%E1%BA%BF%20h%E1%BB%87%20t%E1%BB%91ng%20th%C3%B4ng%20tin%20-%20202013.pdf>
- [3] <https://github.com/lehuygiang28/vnpay>
- [4] <https://www.youtube.com/watch?v=2wkOnBj8EpI>
- [5] <https://github.com/thanqphan/mojoin>
- [6] <https://github.com/idiotman-2212/web-dang-tin-cho-thue-tro>
- [7]<https://www.studocu.vn/vn/document/truong-dai-hoc-thu-dau-mot/cong-nghe-thong-tin/xay-dung-ung-dung-di-dong-quan-ly-nha-tro/42923844>
- [8] <https://www.youtube.com/watch?v=SccSCuHhOw0>
- [9] <https://www.youtube.com/watch?v=VH1SMW8Y1nU>
- [10] chat gpt
- [11] claude
- [12] grok
- [13] ohana
- [14] <https://www.youtube.com/watch?v=lagbuQzGk6U>
- [15] <https://www.youtube.com/watch?v=DvatT4VDOps>
- [16] <https://batdongsan.com.vn/>
- [17]<https://www.slideshare.net/slideshow/giaotrinhphantichthietkehethong-120411053120phpapp01/38065556>
- [18] <https://codedoan.com/san-pham/do-an-tot-nghiep-ung-dung-ho-tro-tim-kiem-nha-tro/>
- [19] <https://www.youtube.com/watch?v=vhFK6hXV-Qo>
- [20] <https://swagger.io/>
- [21] <https://developer.android.com/develop?hl=vi>

[22] <https://github.com/iamashish1/findRoomies>

[23] <https://sourcecodec.net/downloads/source-code-do-an-website-tim-phong-tro-bang-spring-boot-angular/>

[24] <https://www.youtube.com/watch?v=3I3psqOSGkc>

[25] [https://tromoi.com/?utm\\_source=googleads&utm\\_medium=cpc&utm\\_campaign=timtro-trangchu&utm\\_term=timtro-trangchu&gad\\_source=1&gad\\_campaignid=22631404299&gbraid=0AAAAA9\\_V0ySDWFDbS\\_a4x\\_xCtrb4p8Ql-&gclid=Cj0KCCQiAosrJBhD0ARIsAHebCNo2mhmR3FAfWOI4NqqxIQI6YljkZlRSqnDUGotMUVUjoslYAu8fyS0aAsPREALw\\_wcB](https://tromoi.com/?utm_source=googleads&utm_medium=cpc&utm_campaign=timtro-trangchu&utm_term=timtro-trangchu&gad_source=1&gad_campaignid=22631404299&gbraid=0AAAAA9_V0ySDWFDbS_a4x_xCtrb4p8Ql-&gclid=Cj0KCCQiAosrJBhD0ARIsAHebCNo2mhmR3FAfWOI4NqqxIQI6YljkZlRSqnDUGotMUVUjoslYAu8fyS0aAsPREALw_wcB)

