

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC MỎ - ĐỊA CHẤT**



TRẦN VĂN TRÌNH

TRẦN VĂN TRÌNH

ĐỐ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ THÔNG TIN

TÊN ĐỀ TÀI:

XÂY DỰNG WEBSITE ĐẶT PHÒNG KHÁCH SẠN

KHOÁ K65

Hà Nội - Năm 2025

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC MỎ - ĐỊA CHẤT**

ĐỒ ÁN TỐT NGHIỆP

CHUYÊN NGÀNH: MẠNG MÁY TÍNH

GIÁO VIÊN HƯỚNG DẪN

SINH VIÊN THỰC HIỆN

Hà Nội – Năm 2025

LỜI CẢM ƠN

Trong suốt thời gian học tập, rèn luyện tại Trường Đại học Mỏ - Địa chất, em xin chân thành cảm ơn tới các thầy cô giáo trong nhà trường vì đã tạo điều kiện thuận lợi nhất cho em trong việc học tập và nghiên cứu.

Em cũng xin gửi lời cảm ơn chân thành tới tất cả các thầy cô giáo trong Khoa Công nghệ thông tin và các thầy cô bộ môn liên quan đã giảng dạy và truyền đạt lại những kiến thức bổ ích cho em trong suốt thời gian học tập tại trường, giúp em có thêm nhiều kiến thức để sẵn sàng có thể bước tiếp trong con đường sau này.

Đặc biệt em xin gửi lời cảm ơn chân thành tới cô Trần Thị Thu Thúy. Trong suốt thời gian làm đồ án tốt nghiệp vừa qua, cô đã dành nhiều thời gian để theo sát và hướng dẫn em hoàn thành đề tài.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế của một sinh viên, đồ án tốt nghiệp này của em không thể tránh được những thiếu sót. Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các thầy cô để em có điều kiện bổ sung, nâng cao ý thức của mình, phục vụ tốt hơn công việc thực tế sau này.

Em xin chân thành cảm ơn !

Hà Nội, ngày tháng 05 năm 2025

Sinh viên thực hiện

MỤC LỤC

| | |
|---|----|
| LỜI CẢM ƠN | 4 |
| MỤC LỤC | 5 |
| DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT | 9 |
| DANH MỤC HÌNH ẢNH | 10 |
| DANH MỤC BẢNG BIỂU | 13 |
| LỜI MỞ ĐẦU..... | 14 |
| Lý do chọn đề tài..... | 14 |
| Mục tiêu đề tài..... | 14 |
| Đối tượng người dùng và phạm vi nghiên cứu | 15 |
| Nội dung chính của đồ án | 15 |
| Kết quả đạt được của đề tài..... | 16 |
| CHƯƠNG 1. MÔ TẢ BÀI TOÁN | 18 |
| 1.1 Mục tiêu của bài toán | 18 |
| 1.2 Yêu cầu đặt ra của bài toán | 18 |
| 1.2.1 Về dữ liệu | 18 |
| 1.2.2 Về chức năng xử lý | 18 |
| 1.2.3 Về quy trình xử lý..... | 18 |
| 1.3 Mô tả các nguồn tài liệu, dữ liệu, mẫu biểu liên quan | 18 |
| 1.4 Phân tích hiện trạng và các vấn đề tồn tại..... | 19 |
| CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ..... | 20 |
| 2.1 Giới thiệu về HTML, CSS, JavaScript | 20 |
| 2.1.1 HTML (<i>Hypertext Markup Language</i>) | 20 |
| 2.1.2 CSS (<i>Cascading Style Sheets</i>)..... | 20 |
| 2.1.3 <i>JavaScript</i> | 21 |

| | |
|---|----|
| 2.2 Giới thiệu về Visual Studio..... | 21 |
| 2.3 Node.js | 22 |
| 2.4 Framework Express | 22 |
| 2.5 MongoDB..... | 23 |
| 2.6 Bootstrap | 24 |
| 2.7 Vue.js | 24 |
| 2.8 Tóm tắt chương | 25 |
| CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG..... | 30 |
| 3.1 Xác định Actor | 30 |
| 3.2 Sơ đồ Use case | 31 |
| 3.2.1 Use Case dành cho tổng quát | 31 |
| 3.2.2 Use Case dành cho administration..... | 32 |
| 3.2.3 Use Case dành cho customer..... | 32 |
| 3.2.4 Use Case dành cho nhân viên..... | 33 |
| 3.2.5 Use Case dành cho khách vãng lai..... | 33 |
| 3.3 Đặc tả Use case | 37 |
| 3.3.1 Use Case “Đăng nhập”..... | 37 |
| 3.3.2 Use Case “Đăng ký” | 38 |
| 3.3.3 Use Case “Cập nhật thông tin profile” | 39 |
| 3.3.4 Use Case “Thêm thông tin phòng” | 39 |
| 3.3.5 Use Case “Sửa thông tin phòng” | 40 |
| 3.3.6 Use Case “Xem thông tin phòng” | 42 |
| 3.3.7 Use Case “Cập nhật thông tin phòng” | 42 |
| 3.3.8 Use Case “Xem phòng đã đặt” | 43 |
| 3.3.9 Use Case “Xác nhận đặt phòng” | 44 |

| | |
|---|----|
| 3.3.10 Use Case “Đặt phòng” | 45 |
| 3.3.11 Use Case “Xem trang chủ” | 46 |
| 3.3.12 Use Case “Thống kê doanh thu”..... | 47 |
| 3.4 Sơ đồ hoạt động (Activity)..... | 48 |
| 3.4.1 Quản trị viên (Admin)..... | 48 |
| 3.4.2 Khách hàng (Customer)..... | 63 |
| 3.4.3 Khách vãng lai (Guest)..... | 72 |
| 3.4.4 Nhân viên(Staff) | 77 |
| 3.5 Sơ đồ tuần tự (Sequence)..... | 81 |
| 3.5.1 Quản trị viên (Admin)..... | 81 |
| 3.5.2 Khách hàng (Customer)..... | 85 |
| 3.5.3 Khách vãng lai (Guest) | 87 |
| 3.6 Biểu đồ Class | 88 |
| 3.7 Mô tả cơ sở dữ liệu | 89 |
| CHƯƠNG 4. KẾT QUẢ CÀI ĐẶT, THỬ NGHIỆM | 94 |
| 4.1 Yêu cầu cài đặt phần mềm | 94 |
| 4.2 Cấu trúc dự án | 94 |
| 4.2.1 Project Admin | 94 |
| 4.2.2 Project Enduser | 94 |
| 4.2.3 Project Backend..... | 95 |
| 4.3 Cài đặt | 95 |
| 4.3.1 Project Backend..... | 95 |
| 4.3.2 Project Admin và Enduser..... | 95 |
| 4.4 Thực nghiệm | 95 |
| 4.4.1 Đăng ký và đăng nhập | 95 |

| | |
|--|-----|
| 4.4.2 <i>Tìm kiếm và đặt phòng</i> | 97 |
| 4.4.3 <i>Quản lý phòng (Admin)</i> | 99 |
| 4.4.4 <i>Quản lý đơn đặt phòng (Admin)</i> | 101 |
| 4.4.5 <i>Cập nhật trạng thái đơn đặt phòng</i> | 102 |
| 4.5 Kết quả và đánh giá..... | 103 |
| 4.5.1 <i>Kết quả</i> | 103 |
| 4.5.2 <i>Đánh giá</i> | 103 |
| KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN..... | 104 |
| TÀI LIỆU THAM KHẢO | 105 |

DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

| STT | Tên viết tắt | Tiếng anh | Nghĩa tiếng việt |
|-----|--------------|-----------|------------------|
| 1 | CSDL | | Cơ sở dữ liệu |
| 2 | DB | Database | Cơ sở dữ liệu |

DANH MỤC HÌNH ẢNH

| | |
|--|----|
| Hình 2.1 Vòng đời của Vue | 27 |
| Hình 3.1 Use case tổng quát | 31 |
| Hình 3.2 Use Case dành cho administration..... | 32 |
| Hình 3.3 Use Case dành cho khách hàng | 32 |
| Hình 3.4 Use Case dành cho nhân viên | 33 |
| Hình 3.5 Use Case dành cho khách vãng lai | 33 |
| Hình 3.6 Sơ đồ activity đăng nhập | 48 |
| Hình 3.7 Sơ đồ activity thêm thành viên | 49 |
| Hình 3.8 Sơ đồ activity sửa thành viên..... | 50 |
| Hình 3.9 Sơ đồ activity khoá người dùng..... | 51 |
| Hình 3.10 Sơ đồ activity mở khoá người dùng..... | 52 |
| Hình 3.11 Sơ đồ activity xem phòng khách hàng đã đặt | 53 |
| Hình 3.12 Sơ đồ activity xem báo cáo doanh thu..... | 54 |
| Hình 3.13 Sơ đồ activity thêm loại phòng | 55 |
| Hình 3.14 Sơ đồ activity sửa loại phòng | 56 |
| Hình 3.15 Sơ đồ activity xoá loại phòng | 57 |
| Hình 3.16 Sơ đồ activity xoá bình luận | 58 |
| Hình 3.17 Sơ đồ activity thêm dịch vụ phòng | 59 |
| Hình 3.18 Sơ đồ activity sửa dịch vụ phòng..... | 60 |
| Hình 3.19 Sơ đồ activity xoá dịch vụ phòng | 61 |
| Hình 3.20 Sơ đồ activity đăng xuất | 62 |
| Hình 3.21 Sơ đồ activity xem thông tin phòng..... | 63 |
| Hình 3.22 Sơ đồ activity xem trang chủ | 64 |
| Hình 3.23 Sơ đồ activity tìm kiếm phòng..... | 65 |
| Hình 3.24 Sơ đồ activity lọc theo phòng điều kiện | 66 |
| Hình 3.25 Sơ đồ activity đăng nhập | 67 |
| Hình 3.26 Sơ đồ activity đặt phòng | 68 |
| Hình 3.27 Sơ đồ activity xem phòng đã đặt | 69 |
| Hình 3.28 Sơ đồ activity đánh giá phòng | 70 |
| Hình 3.29 Sơ đồ activity đăng xuất | 71 |

| | |
|--|-----|
| Hình 3.30 Sơ đồ activity đăng ký | 72 |
| Hình 3.31 Sơ đồ activity xem trang chủ | 73 |
| Hình 3.32 Sơ đồ activity xem thông tin phòng..... | 74 |
| Hình 3.33 Sơ đồ activity tìm kiếm phòng..... | 75 |
| Hình 3.34 Sơ đồ activity lọc phòng theo điều kiện | 76 |
| Hình 3.35 Sơ đồ activity đăng nhập | 77 |
| Hình 3.36 Sơ đồ activity xem phòng khách hàng đã đặt | 78 |
| Hình 3.37 Sơ đồ activity xem báo cáo..... | 79 |
| Hình 3.38 Sơ đồ activity đăng xuất | 80 |
| Hình 3.39 Sơ đồ Sequence đăng nhập | 81 |
| Hình 3.40 Sơ đồ Sequence khokhách hàng | 81 |
| Hình 3.41 Sơ đồ Sequence xem danh sách phòng đã được đặt | 82 |
| Hình 3.42 Sơ đồ Sequence xem báo cáo | 82 |
| Hình 3.43 Sơ đồ Sequence thêm nhân viên | 83 |
| Hình 3.44 Sơ đồ Sequence sửa nhân viên | 83 |
| Hình 3.45 Sơ đồ Sequence xóa nhân viên | 84 |
| Hình 3.46 Sơ đồ Sequence xem phòng..... | 85 |
| Hình 3.47 Sơ đồ Sequence xem trang chủ..... | 85 |
| Hình 3.48 Sơ đồ Sequence tìm kiếm phòng | 86 |
| Hình 3.49 Sơ đồ Sequence đặt phòng..... | 86 |
| Hình 3.50 Sơ đồ Sequence đánh giá phòng | 87 |
| Hình 3.51 Sơ đồ Sequence đăng ký khách hàng..... | 88 |
| Hình 4.1 Màn hình đăng ký | 96 |
| Hình 4.2 Màn hình đăng nhập | 96 |
| Hình 4.3 Màn hình danh sách tìm kiếm..... | 97 |
| Hình 4.4 Màn hình chi tiết phòng..... | 97 |
| Hình 4.5 Màn hình đặt phòng | 98 |
| Hình 4.6 Màn hình thêm phòng | 99 |
| Hình 4.7 Màn hình sửa phòng | 100 |
| Hình 4.8 Màn hình xóa phòng | 101 |
| Hình 4.9 Màn hình danh sách phòng chưa xác nhận | 101 |

Hình 4.10 Màn hình danh sách phòng đã xác nhận 102

DANH MỤC BẢNG BIỂU

| | |
|--|----|
| Bảng 3.1 Danh sách Actor | 30 |
| Bảng 3.3 Biểu đồ Class..... | 88 |
| Bảng 3.4 Bảng thông tin User..... | 89 |
| Bảng 3.5 Bảng thông tin chung | 90 |
| Bảng 3.6 Bảng thông tin dịch vụ chung | 91 |
| Bảng 3.7 Bảng thông tin loại phòng | 91 |
| Bảng 3.8 Bảng thông tin phòng đã book | 92 |
| Bảng 3.9 Bảng thông tin đánh giá phòng | 93 |
| Bảng 3.10 Bảng chức năng thanh toán | 93 |

LỜI MỞ ĐẦU

Lý do chọn đề tài

Đáp ứng nhu cầu đặt phòng nhanh chóng, tiện lợi cho khách hàng

- Ngành du lịch – khách sạn đang phát triển mạnh mẽ, đặc biệt sau đại dịch, nhu cầu đặt phòng trực tuyến ngày càng phổ biến.
- Hệ thống đặt phòng giúp khách hàng tiết kiệm thời gian, dễ dàng so sánh giá cả, tiện nghi và đặt phòng mọi lúc, mọi nơi.

Giúp khách sạn quản lý hiệu quả và mở rộng thị trường

- Nhiều khách sạn nhỏ, vừa vẫn quản lý phòng, khách hàng và đơn đặt phòng bằng giấy tờ hoặc Excel → dễ sai sót, tốn thời gian.
- Đề tài giúp số hóa và tự động hóa quy trình đặt phòng, nâng cao hiệu quả quản lý.

Phù hợp xu hướng số hóa và phát triển du lịch hiện nay

- Ứng dụng công nghệ vào quy trình vận hành khách sạn.
- Tăng khả năng tiếp cận khách hàng qua các nền tảng trực tuyến.

Khả năng ứng dụng thực tế và triển khai cao

- Hệ thống có thể áp dụng vào thực tế cho nhà nghỉ, khách sạn nhỏ hoặc phát triển thành sản phẩm startup.
- Có thể triển khai lên các nền tảng cloud như Heroku, Vercel, Firebase...

Mục tiêu đề tài

- Nghiên cứu về cách tổ chức và hoạt động của framework Vue.js và áp dụng giao diện ứng dụng.
- Nghiên cứu về dữ liệu và áp dụng xây dựng API cho ứng dụng.

- Xây dựng Website đáp ứng nhu cầu đặt phòng trực tuyến
- Xây dựng được hệ thống quản lý thông tin phòng, user, thông tin đặt phòng và doanh số theo từng tuần, tháng, năm.

Đối tượng người dùng và phạm vi nghiên cứu

Đối tượng người dùng

| ĐỐI TƯỢNG NGƯỜI SỬ DỤNG | |
|-------------------------|------------------------|
| STT | Tên đối tượng |
| 1 | Khách vãng lai (guest) |
| 2 | Khách hàng (customer) |
| 3 | Nhân viên (staff) |
| 4 | Quản trị viên (admin) |

Phạm vi nghiên cứu

- Nghiên cứu cách viết API trên Node.js
- Nghiên cứu về kiến trúc của Vue.js và cách xây dựng các Single Page Application
- Nghiên cứu cách kết nối dữ liệu gọi API từ phía Frontend để lấy dữ liệu trả về từ phía BackEnd
- Nghiên cứu hệ quản trị dữ liệu MongooseDB

Nội dung chính của đồ án

Nội dung chính của đồ án tốt nghiệp bao gồm:

- **Chương 1: Tổng quan** - Giới thiệu về mục tiêu, phạm vi và ý nghĩa của đề tài. Trình bày về các công nghệ và công cụ sẽ được sử dụng trong quá trình phát triển.
- **Chương 2: Cơ sở lý thuyết** - Tìm hiểu về các công nghệ nền tảng như Vue.js, Node.js và MongoDB. Cung cấp kiến thức cơ bản và nâng cao về các công nghệ này.
- **Chương 3: Phân tích thiết kế hệ thống** - Khảo sát và phân tích các yêu cầu của hệ thống. Thiết kế các thành phần chính của hệ thống, bao gồm các mô hình dữ liệu, giao diện người dùng và các quy trình nghiệp vụ.
- **Chương 4: Kết quả cài đặt, thử nghiệm** - Trình bày quá trình cài đặt và cấu hình hệ thống. Đánh giá kết quả thử nghiệm và hiệu quả của hệ thống sau khi triển khai.

Kết quả đạt được của đề tài

Xây dựng được hệ thống website hoàn chỉnh

- Giao diện thân thiện, dễ sử dụng, phù hợp với cả người dùng phổ thông và quản trị viên.
- Tương thích trên nhiều thiết bị như máy tính, máy tính bảng và điện thoại di động.

Cung cấp đầy đủ các chức năng cơ bản của một hệ thống đặt phòng trực tuyến

- Đăng ký, đăng nhập, quản lý tài khoản người dùng.
- Tìm kiếm khách sạn theo địa điểm, ngày đặt, loại phòng.
- Xem thông tin chi tiết khách sạn và các loại phòng.
- Đặt phòng, hủy phòng, kiểm tra lịch sử đặt phòng.
- Gửi và xem đánh giá của người dùng về khách sạn.

Chức năng quản trị hiệu quả

- Quản lý danh sách khách sạn, loại phòng, đơn đặt phòng và người dùng.
- Thống kê đơn đặt phòng theo thời gian, doanh thu, tỷ lệ lấp đầy phòng.
- Dễ dàng cập nhật thông tin khách sạn và phòng.

Ứng dụng các công nghệ web hiện đại

- Giao diện sử dụng HTML, CSS, JavaScript (React/Vue nếu có).
- Backend sử dụng Node.js, PHP hoặc Python, kết hợp với cơ sở dữ liệu MySQL, MongoDB hoặc PostgreSQL.
- Hệ thống được phân tích và thiết kế theo hướng mô-đun, dễ dàng mở rộng trong tương lai.

Nâng cao kỹ năng thực tế

- Sinh viên/nhóm thực hiện đề tài đã nâng cao kỹ năng lập trình, tư duy hệ thống, làm việc nhóm và quản lý dự án phần mềm.
- Có thêm kinh nghiệm về triển khai, bảo mật, kiểm thử và tối ưu hóa hiệu suất website.

CHƯƠNG 1. MÔ TẢ BÀI TOÁN

1.1 Mục tiêu của bài toán

Mục tiêu chính là xây dựng hệ thống quản lý đặt phòng khách sạn Golobe tại thành phố Đà Nẵng. Hệ thống này hỗ trợ trong công tác quản lý, thống kê, và tra cứu thông tin phòng khách sạn một cách hiệu quả, chính xác và nhanh chóng.

1.2 Yêu cầu đặt ra của bài toán

1.2.1 Về dữ liệu

- Dữ liệu cá nhân của khách hàng: Họ tên, số điện thoại, địa chỉ, địa chỉ email, mật khẩu,...
- Dữ liệu hệ thống: Dữ liệu phòng, loại phòng, dịch vụ, đơn đặt phòng, người dùng, đánh giá, thông tin thanh toán.

1.2.2 Về chức năng xử lý

- Cho **khách hàng**: Đăng ký/đăng nhập, tìm kiếm, đặt phòng, thanh toán, xem lịch sử, đánh giá. Quản lý nhân viên, quản lý khách hàng.
- Cho **admin/nhân viên**: Thêm/sửa/xóa phòng, xác nhận đơn, thống kê doanh thu, quản lý người dùng. Xuất báo cáo, thống kê.
- Cho **khách vãng lai**: Tìm kiếm phòng, xem thông tin phòng, đăng ký tài khoản.

1.2.3 Về quy trình xử lý

- Hệ thống nhận yêu cầu từ người dùng frontend (Vue.js / Nuxt.js), gửi API đến backend (Node.js + Express). Quản lý dữ liệu đầu vào (thông tin khách hàng).
- Backend xử lý logic và truy vấn MongoDB, sau đó trả về dữ liệu cho frontend hiển thị. Sao lưu, phục hồi dữ liệu khi cần thiết.

1.3 Mô tả các nguồn tài liệu, dữ liệu, mẫu biểu liên quan

- **Tài liệu tham khảo:**

- Tài liệu chính thống về Vue.js, Node.js, MongoDB, Bootstrap.
- Các tài liệu kỹ thuật về API RESTful, thiết kế web responsive.
- Một số nguồn online như: npmjs.com,
caodangquoctehanoi.edu.vn,...

- **Công cụ phát triển:**

- Visual Studio Code, GitLab, MongoDB Compass.
- Hệ điều hành: Windows 10/11 hoặc MacOS.

- **Mẫu biểu / Biểu đồ trong đồ án:**

- Use Case Diagram, Activity Diagram, Sequence Diagram.
- Biểu đồ Class, biểu đồ thông kê doanh thu.
- Màn hình đăng nhập, tìm kiếm, đặt phòng, quản lý phòng.

1.4 Phân tích hiện trạng và các vấn đề tồn tại

- **Hiện trạng:**

- Nhiều khách sạn nhỏ vẫn đang dùng phương pháp truyền thống như sổ sách hoặc Excel, gây tốn thời gian, dễ sai sót và khó thống kê.
- Nhu cầu đặt phòng online ngày càng cao nhưng chưa có giải pháp tối ưu, chi phí thấp cho các đơn vị nhỏ.

- **Vấn đề cần giải quyết:**

- Cần một hệ thống đặt phòng giao diện thân thiện, chi phí thấp, dễ triển khai.
- Cung cấp tính năng linh hoạt, hỗ trợ khách hàng đặt/hủy phòng dễ dàng, quản lý hiệu quả.
- Giải quyết bài toán thống kê, phân quyền, quản lý thông tin với khả năng mở rộng trong tương lai.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ

2.1 Giới thiệu về HTML, CSS, JavaScript

2.1.1 HTML (*Hypertext Markup Language*)

HTML hay còn gọi là “Ngôn ngữ đánh dấu siêu văn bản” . Nó là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web với các mẩu thông tin được trình bày trên World Wide Web. Cùng với CSS và JavaScript, HTML tạo ra bộ ba nền tảng kỹ thuật cho World Wide Web. HTML được định nghĩa như là một ứng dụng đơn giản của SGML và được sử dụng trong các tổ chức cần đến các yêu cầu xuất bản phức tạp. HTML đã trở thành một chuẩn Internet do tổ chức World Wide Web Consortium (W3C) duy trì.

HTML được sáng tạo bởi Tim Berners-Lee, nhà vật lý học của trung tâm nghiên cứu CERN ở Thụy Sĩ. Anh ta đã nghĩ ra được ý tưởng cho hệ thống hypertext trên nền Internet. Anh xuất bản phiên bản đầu tiên của HTML trong năm 1991 bao gồm 18 tag HTML. Từ đó, mỗi phiên bản mới của HTML đều có thêm tag mới và attributes mới. Nâng cấp mới nhất gần đây là vào năm 2014, khi ra mắt chuẩn HTML5.

2.1.2 CSS (*Cascading Style Sheets*)

CSS là chữ viết tắt cho “Cascading Style Sheets”, được sử dụng để mô tả giao diện và định dạng của một tài liệu viết bằng ngôn ngữ đánh dấu. Nó cung cấp một tính năng bổ sung cho HTML. Nó thường được sử dụng với HTML để thay đổi style của trang web và giao diện người dùng. Nó cũng có thể được sử dụng với bất kỳ loại tài liệu XML nào bao gồm cả XML đơn giản, SVG và XUL.

CSS được phát triển bởi W3C (World Wide Web Consortium) vào năm 1996, nhằm mục đích: Hạn chế tối thiểu việc làm rối mã HTML của trang Web bằng các thẻ quy định kiểu dáng khiến mã nguồn của trang Web được gọn gàng hơn, tách nội dung của trang Web và định dạng hiển thị, dễ dàng cho việc cập nhật nội dung; Tạo ra các kiểu dáng có thể áp dụng cho nhiều trang Web, giúp tránh phải lặp lại việc định dạng cho các trang Web giống nhau.

Giống như HTML, CSS không thực sự là một ngôn ngữ lập trình mà là một ngôn ngữ định kiểu – style. Điều này có nghĩa là nó cho phép bạn áp dụng kiểu có chọn lọc cho các phần tử trong tài liệu HTML.

2.1.3 JavaScript

JavaScript là một ngôn ngữ lập trình kịch bản đa nền tảng (Cross-platform), hướng đối tượng. Nó chủ yếu được sử dụng để tăng cường cung cấp trải nghiệm thân thiện hơn với người dùng website như là bao gồm các trang web cập nhật tự động, cải tiến giao diện người dùng như menu và hộp thoại, hình động, đồ họa 2D và 3D, bản đồ tương tác, trình phát video... JavaScript là một ngôn ngữ nhỏ và nhẹ, chạy trong môi trường máy chủ lưu trữ. Nó có thể được kết nối với các đối tượng của môi trường để cung cấp kiểm soát chương trình đối với chúng.

JavaScript được phát triển bởi Brendan Eich tại Hãng truyền thông Netscape với cái tên đầu tiên Mocha, rồi sau đó đổi tên thành LiveScript, và cuối cùng thành JavaScript. Giống Java, JavaScript có cú pháp tương tự C, nhưng nó gần với Self hơn Java. Hiện nay phiên bản mới nhất của JavaScript là ECMAScript 7 với nhiều cải tiến vượt bậc so với phiên bản tiền nhiệm.

JavaScript là ngôn ngữ kịch bản phía máy khách (client – side) duy nhất. Do đó, nó được sử dụng trên hầu hết các trang web mà bạn nhìn thấy trên internet. Tuy nhiên cú pháp của nó vô cùng lộn xộn và khó làm chủ, do đó Jquery – một thư viện của JavaScript ra đời nhằm giúp lập trình viên xây dựng những chức năng có sử dụng Javascript trở nên dễ dàng hơn như: đơn giản hóa việc duyệt tài liệu HTML, xử lý sự kiện, hoạt ảnh và tương tác Ajax để phát triển web nhanh chóng... jQuery là một bộ công cụ JavaScript gọn nhẹ nhưng đầy mạnh mẽ, được chứng minh qua hơn 90% các website trên thế giới sử dụng (trừ các website chạy bằng JavaScript Framework).

2.2 Giới thiệu về Visual Studio

Visual Studio cung cấp các mẫu dự án và công cụ để phát triển ứng dụng Web ASP.NET một cách dễ dàng và hiệu quả. Nó cung cấp các công cụ chỉnh sửa mã thông minh như IntelliSense, giúp bạn viết mã nhanh chóng và tránh các

lỗi cú pháp. Visual Studio hỗ trợ quản lý dự án, cung cấp các công cụ quản lý dự án mạnh mẽ cho phép lập trình viên quản lý mã nguồn, xây dựng và triển khai ứng dụng của họ một cách hiệu quả. Visual Studio tích hợp với các hệ thống quản lý mã nguồn như Git và Azure DevOps, giúp bạn quản lý mã nguồn và làm việc cộng tác với các thành viên trong nhóm phát triển.

2.3 Node.js

Node.js là một nền tảng (Platform) phát triển độc lập được xây dựng ở trên Javascript Runtime của Chrome mà chúng ta có thể xây dựng được các ứng dụng mạng một cách nhanh chóng và dễ dàng mở rộng.

Phần Core bên dưới của Node.js được viết hầu hết bằng C++ nên cho tốc độ xử lý và hiệu năng khá cao.

Node.js tạo ra được các ứng dụng có tốc độ xử lý nhanh, realtime thời gian thực.

Node.js áp dụng cho các sản phẩm có lượng truy cập lớn, cần mở rộng nhanh, cần đổi mới công nghệ, hoặc tạo ra các dự án Startup nhanh nhất có thể.

Node.js chạy đa nền tảng phía Server, sử dụng kiến trúc hướng sự kiện Event-driven, cơ chế non-blocking I/O làm cho nó nhẹ và hiệu quả..

2.4 Framework Express

Express là một trong những framework phổ biến dùng để xây dựng API và Website phổ biến nhất của NodeJS. Nó được sử dụng rộng rãi đến mức hầu như mọi dự án Web nào đều bắt đầu bằng việc tích hợp Express. Có rất nhiều lí do để chọn Express:

- Có nhiều tính năng hỗ trợ trong việc xây dựng web và API.
- Quản lý các route dễ dàng.
- Cung cấp một nền tảng phát triển cho API.
- Hỗ trợ nhiều thư viện và Plugin.
- Bảo mật và an toàn hơn so với việc code thuần.

2.5 MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, là CSDL thuộc NoSql và được hàng triệu người sử dụng

MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON thay vì dạng bảng như CSDL quan hệ nên truy vấn sẽ rất nhanh

Với CSDL quan hệ chúng ta có khái niệm bảng, các cơ sở dữ liệu quan hệ (như MySQL hay SQL Server...) sử dụng các bảng để lưu dữ liệu thì với MongoDB chúng ta sẽ dùng khái niệm là collection thay vì bảng

Các collection trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ không cần tuân theo một cấu trúc nhất định.

Ưu điểm của MongoDB

- Linh hoạt: Lưu dữ liệu dạng JSON, không cần schema cố định.
- Hiệu suất cao: Tốc độ truy vấn và ghi nhanh, phù hợp với dữ liệu lớn.
- Dễ mở rộng: Hỗ trợ sharding, phân tán dữ liệu tốt.
- Tích hợp tốt với Node.js: Dễ dàng xây dựng ứng dụng web hiện đại.
- Dễ triển khai: Miễn phí, cài đặt nhanh, có nhiều công cụ hỗ trợ.

Nhược điểm của MongoDB

- Không hỗ trợ JOIN phức tạp: Khó xử lý quan hệ nhiều bảng như SQL.
- Dữ liệu dư thừa: Do không có chuẩn hóa dữ liệu như RDBMS.
- Tiêu tốn bộ nhớ: Vì lưu trữ dữ liệu kèm key trong mỗi document.
- Thiếu tính toàn vẹn dữ liệu mạnh: Không hỗ trợ transaction phức tạp như SQL.

2.6 Bootstrap

Bootstrap là 1 framework HTML, CSS, và JavaScript cho phép người dùng dễ dàng thiết kế website theo 1 chuẩn nhất định, tạo các website thân thiện với các thiết bị cầm tay như mobile, ipad, tablet, ...

Bootstrap bao gồm những cái cơ bản có sẵn như: typography, forms, buttons, tables, navigation, modals, image carousels và nhiều thứ khác. Trong bootstrap có thêm nhiều Component, Javascript hỗ trợ cho việc thiết kế responsive của bạn dễ dàng, thuận tiện và nhanh chóng hơn.

Điểm thuận lợi khi sử dụng Bootstrap:

- Rất dễ để sử dụng: Nó đơn giản vì nó được base trên HTML, CSS và Javascript chỉ cần có kiến thức cơ bản về 3 cái đó là có thể sử dụng bootstrap tốt.
- Responsive: Bootstrap xây dựng sẵn responsive css trên các thiết bị Iphones, tablets, và desktops. Tính năng này khiến cho người dùng tiết kiệm được rất nhiều thời gian trong việc tạo ra một website thân thiện với các thiết bị điện tử, thiết bị cầm tay.
- Tương thích với trình duyệt: Nó tương thích với tất cả các trình duyệt (Chrome, Firefox, Internet Explorer, Safari, and Opera). Tuy nhiên, với IE browser, Bootstrap chỉ hỗ trợ từ IE9 trở lên. Điều này vô cùng dễ hiểu vì IE8 không support HTML5 và CSS3.

2.7 Vue.js

Vue.js là một **framework JavaScript** dùng để xây dựng **giao diện người dùng (UI)** và **ứng dụng web một trang (SPA)**.

Nhẹ, dễ học, dễ tích hợp.

Ưu điểm:

- Kích thước nhỏ: Vue chỉ nặng ~18KB, cài đặt nhanh, tối ưu cho SEO và trải nghiệm người dùng.

- DOM ảo hiệu suất cao: Tăng tốc độ cập nhật giao diện nhờ sử dụng Virtual DOM.
- Liên kết dữ liệu linh hoạt: Hệ thống phản ứng mạnh mẽ, hỗ trợ ràng buộc dữ liệu hai chiều dễ dàng.

Nhược điểm :

- Rào cản ngôn ngữ: Tài liệu và cộng đồng chủ yếu bằng tiếng Trung.
- Chưa phù hợp dự án lớn: Ít được hỗ trợ bởi các tập đoàn lớn, cộng đồng nhỏ hơn React/Angular.
- Quá linh hoạt: Dễ gây thiếu nhất quán trong nhóm phát triển.
- Tài nguyên hạn chế: Hệ sinh thái nhỏ hơn so với các framework phổ biến khác.

2.8 Tóm tắt chương

Vue.js (Front-end):

- Dễ học, dễ phát triển: Cấu trúc rõ ràng, cú pháp đơn giản, thân thiện với người mới.
- Component-based: Tái sử dụng mã nguồn hiệu quả, dễ bảo trì và mở rộng.
- Hiệu năng cao: Cập nhật giao diện nhanh chóng nhờ hệ thống reactivity mạnh mẽ.
- Tích hợp dễ dàng với HTML tĩnh hoặc project SPA.

Node.js + Express (Back-end):

- Hiệu năng cao: Node.js sử dụng kiến trúc bất đồng bộ (non-blocking I/O), giúp xử lý nhiều kết nối đồng thời – lý tưởng cho ứng dụng web thời gian thực.

- JavaScript toàn bộ stack: Dễ đồng bộ hóa dữ liệu giữa client và server vì cả hai đều dùng JavaScript.
- Express.js nhẹ và linh hoạt: Framework đơn giản nhưng mạnh mẽ để xây dựng RESTful API một cách nhanh chóng.

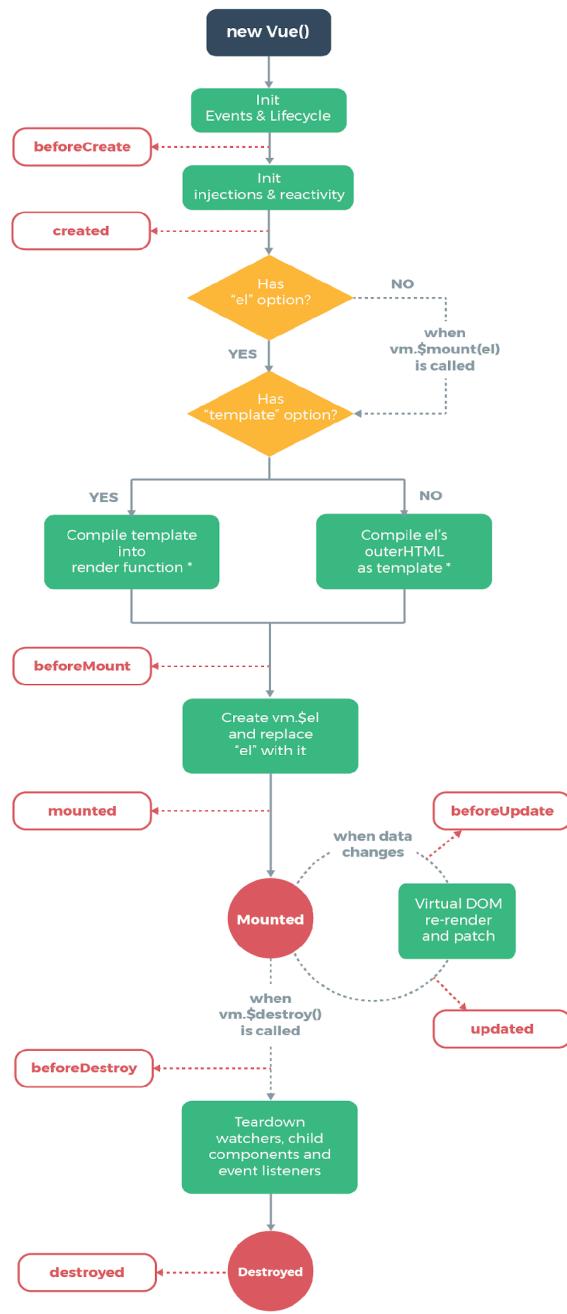
MongoDB (Cơ sở dữ liệu):

- NoSQL linh hoạt: Lưu trữ dữ liệu dạng JSON (document-based), rất phù hợp khi dữ liệu không có cấu trúc cố định.
- Tích hợp tốt với Node.js: Giao tiếp qua thư viện Mongoose hoặc native driver dễ dàng.
- Dễ mở rộng và triển khai: Phù hợp cho cả dự án nhỏ lẫn quy mô lớn.

Bootstrap (Giao diện UI):

- Nhanh chóng, tiện lợi: Thư viện CSS phổ biến, giúp tạo giao diện đẹp, responsive mà không tốn nhiều thời gian viết CSS thủ công.
- Tương thích tốt với Vue.js: Dễ tích hợp để nhanh chóng tạo bố cục, form, bảng, modal...

Vòng đời của Vue (Vue Instant)



* template compilation is performed ahead-of-time if using a build step, e.g. single-file components

Hình 2.1 Vòng đời của Vue

- Sơ đồ phía trên là vòng đời của một instance. Như chúng ta thấy, Lifecycle của một instance sẽ gồm:

- Khởi tạo một đối tượng Vue

- Gắn kết vào DOM
- Cập nhật DOM khi dữ liệu thay đổi
- Hủy instance

2.6.1 Một số tính năng nổi bật của Visual Studio

- Hỗ trợ phát triển ứng dụng trên nhiều ngôn ngữ lập trình như C#, C++, JavaScript, Python và nhiều ngôn ngữ khác.
- Với giao diện thân thiện và dễ sử dụng cho phép lập trình viên tập trung vào việc phát triển ứng dụng mà không mất nhiều thời gian để tìm hiểu công cụ.
- Nó cung cấp các công cụ hỗ trợ xây dựng và triển khai ứng dụng lên nhiều nền tảng khác nhau như Desktop, Web, di động và Cloud.
- Cùng với tích hợp các dịch vụ Cloud như Azure giúp người dùng triển khai và quản lý ứng dụng của mình trên Cloud một cách dễ dàng.
- Cho phép đặt các điểm ngắt để dừng mã tại những điểm cụ thể và kiểm tra trạng thái ứng dụng.
- Cho phép nhiều lập trình viên làm việc cùng nhau trên cùng một dự án trong thời gian thực.
- Cung cấp các số liệu về độ phức tạp, khả năng bảo trì và các yếu tố khác của mã nguồn.

2.6.2 Ưu và nhược điểm của Visual Studio

Ưu điểm:

- Giao diện trực quan, dễ sử dụng, giúp lập trình viên dễ dàng tiếp cận và làm việc hiệu quả.
- Hỗ trợ phát triển ứng dụng cho nhiều nền tảng như Windows, macOS, Linux, Android, IOS và Web.

- Hỗ trợ Git trực tiếp, giúp dễ dàng quản lý mã nguồn theo dõi các thay đổi và làm việc nhóm
- Tính năng Live Share cho phép nhiều lập trình viên cộng tác và làm việc cùng nhau trong thời gian thực
- Tích hợp sâu với Azure hỗ trợ triển khai và quản lý ứng dụng trên đám mây

Nhược điểm:

- Phiên bản Professional và Enterprise của Visual Studio có chi phí cao
- Tiêu tốn nhiều tài nguyên hệ thống, dẫn đến hiệu suất chậm trên các máy tính cấu hình thấp
- Quá trình cài đặt và cập nhật có thể mất nhiều thời gian do kích thước lớn của phần mềm và các gói đi kèm
- Sự đa dạng và phong phú của các tính năng có thể khiến người mới cảm thấy choáng ngợp và khó sử dụng hết các tính năng.

CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1 Xác định Actor

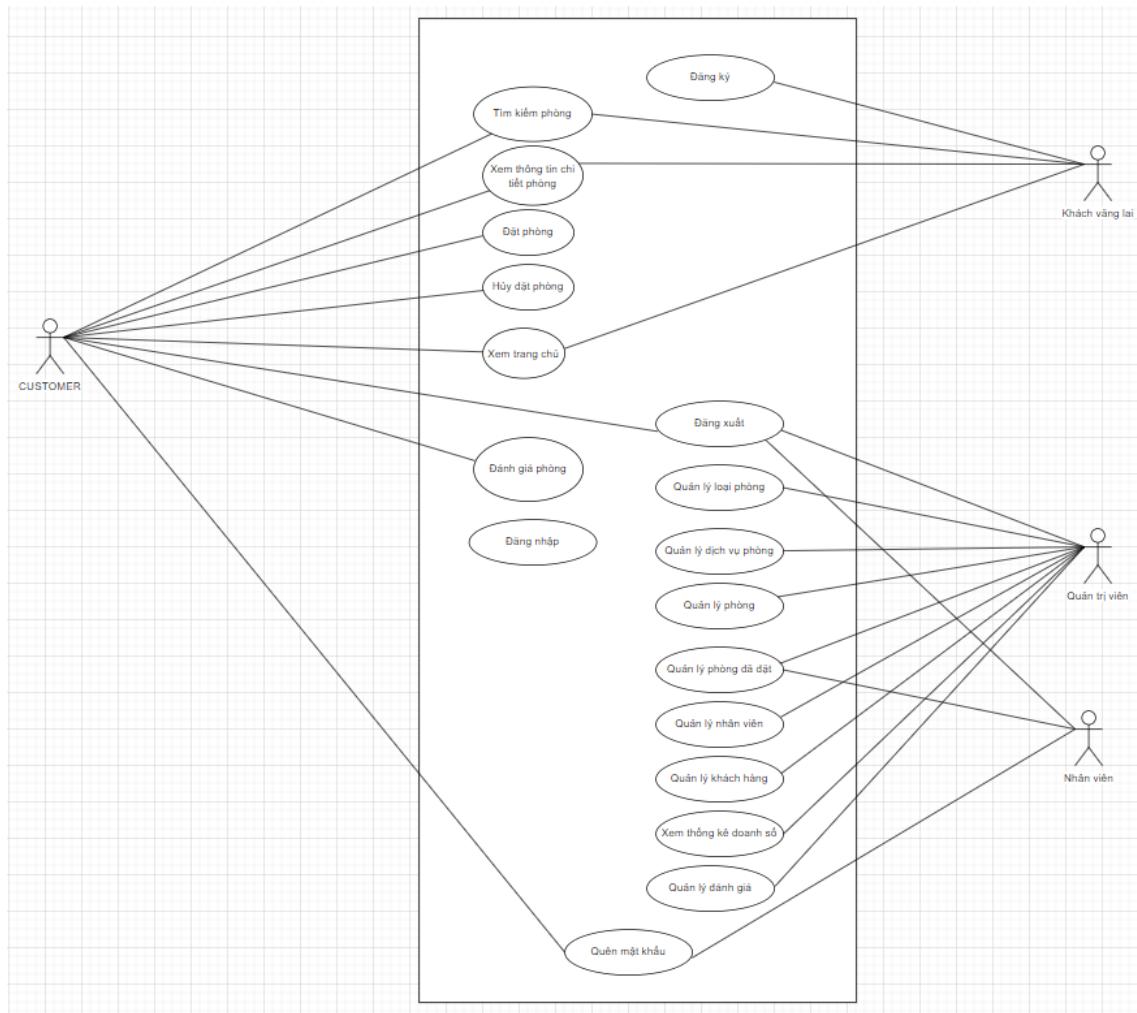
Xác định các Actor của hệ thống là bước quan trọng để nhận diện những tác nhân sẽ tương tác với hệ thống. Việc này không chỉ giúp định rõ vai trò của từng thành phần trong hệ thống mà còn tạo nền tảng cho việc phân tích các yêu cầu chức năng và phi chức năng. Trong hệ thống đặt phòng khách sạn, các Actor chính bao gồm:

| STT | Tác nhân | Ý nghĩa |
|-----|-----------------------|--|
| 1 | Quản trị viên (Admin) | Người quản trị hệ thống. |
| 2 | Khách hàng (Customer) | Khách hàng có nhu cầu đặt phòng truy cập vào hệ thống. |
| 3 | Khách vãng lai(Guest) | Khách chưa có tài khoản trên hệ thống |
| 4 | Nhân viên(Staff) | Nhân viên hệ thống phụ trách theo dõi thông tin phòng |

Bảng 3.1 Danh sách Actor

3.2 Sơ đồ Use case

3.2.1 Use Case dành cho tổng quát



Hình 3.1 Use case tóm quát

3.2.2 Use Case dành cho administration



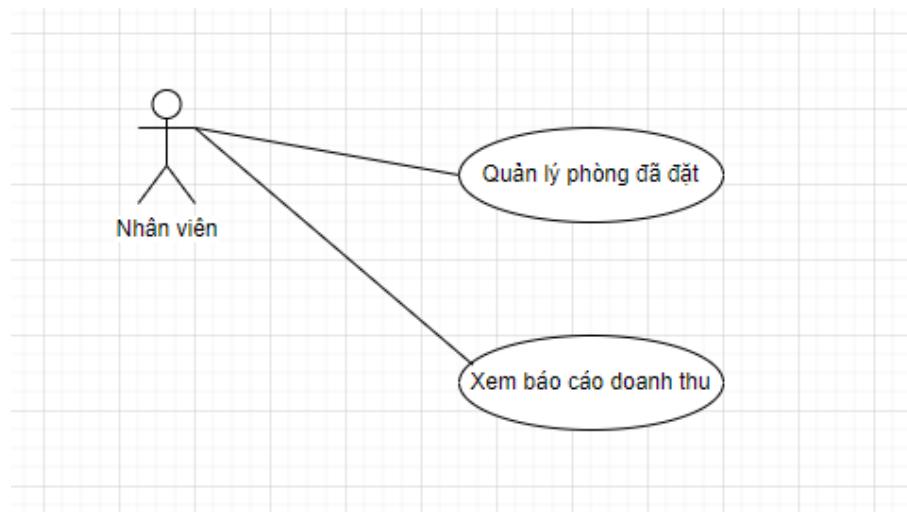
Hình 3.2 Use Case dành cho administration

3.2.3 Use Case dành cho customer



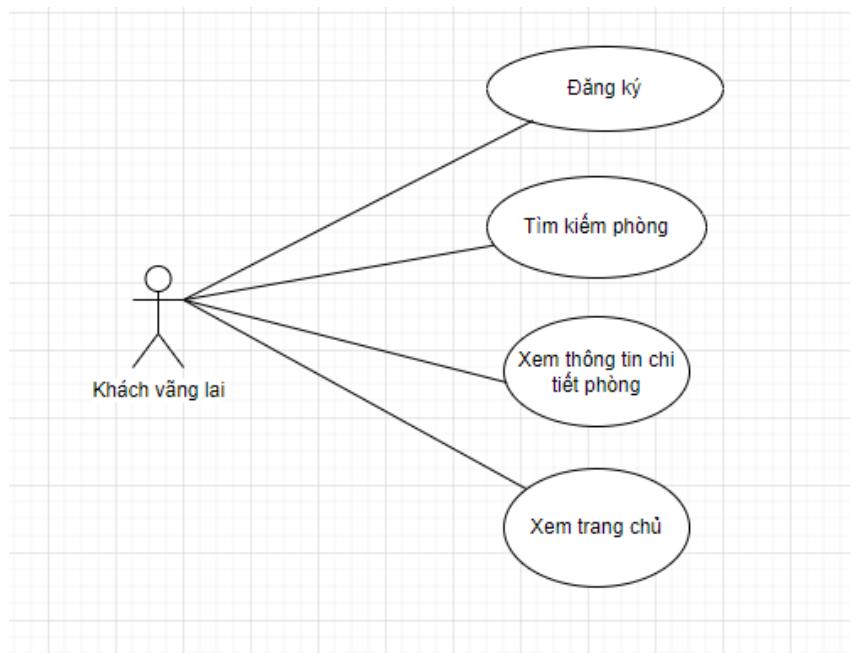
Hình 3.3 Use Case dành cho khách hàng

3.2.4 Use Case dành cho nhân viên



Hình 3.4 Use Case dành cho nhân viên

3.2.5 Use Case dành cho khách vãng lai



Hình 3.5 Use Case dành cho khách vãng lai

| STT | Tên Use Case | Ý nghĩa/Ghi chú | Admin | Customer | Staff | Guest |
|-----|--|--|-------|----------|-------|-------|
| 1 | Xem thông tin phòng | Cho phép người dùng, admin , nhân viên đều được phép xem thông tin phòng | X | X | X | X |
| 2 | Xem trang chủ | Cho phép người dùng và admin đều được phép xem giao diện của trang chủ | X | X | X | X |
| 3 | Tìm sản phẩm theo ngày nhận phòng và trả phòng | Người dùng có thể nhập ngày nhận và trả phòng để tìm kiếm phòng phù hợp | | X | | X |
| 4 | Tìm sản phẩm theo danh mục | Người dùng có thể tìm kiếm theo loại phòng, dịch vụ phòng | | X | | X |
| 5 | Đăng ký khách hàng | Người dùng đăng ký làm khách hàng để đặt phòng và nhận được các ưu đãi riêng của hệ thống | | | | X |
| 6 | Đăng nhập | Để người dùng, admin có thể đăng nhập vào trang quản trị riêng của mình (mỗi actor sẽ có 1 | X | X | X | |

| STT | Tên Use Case | Ý nghĩa/Ghi chú | Admin | Customer | Staff | Guest |
|-----|--------------------------|---|-------|----------|-------|-------|
| | | giao diện quản trị riêng và quyền hạn khác nhau) | | | | |
| 7 | Đánh giá phòng | Người dùng có thể đánh phòng (với điều kiện chỉ được đánh giá được các phòng họ đã đặt) | | X | | |
| 8 | Thêm thành viên | Người quản lý có thể thêm được thành viên cho hệ thống của mình | X | | | |
| 9 | Active/Logout thành viên | Người quản lý có thể xóa những thành viên thuộc hệ thống của mình | X | | | |
| 10 | Xem báo cáo | Người quản lý có thể xem báo cáo doanh thu theo khoảng thời gian nhất định | X | | | |
| 11 | Tạo phòng | admin có thể thêm phòng ở trang admin | X | | | |
| 12 | Sửa phòng | admin có thể sửa phòng ở trang admin | X | | | |
| 13 | Xóa phòng | admin có thể xoá phòng ở trang admin | X | | | |
| 14 | Xem phòng đã đặt | admin có thể xem phòng khách hàng đã đặt | X | | X | |

| STT | Tên Use Case | Ý nghĩa/Ghi chú | Admin | Customer | Staff | Guest |
|-----|----------------------------------|---|-------|----------|-------|-------|
| 15 | Xem phòng đã đặt theo trạng thái | admin có thể xem được các phòng đã đặt theo nhiều điều kiện | X | | X | |
| 16 | Xem đánh giá | admin có thể xem các đánh giá của khách hàng về các phòng hiện có trên hệ thống | X | | X | |
| 17 | Đặt phòng | Khách hàng có thể đặt phòng theo ý muốn | | X | | |
| 18 | Thanh toán | Khách hàng có thể thanh toán online bằng VNPay | | X | | |

Bảng 3.2 Sơ đồ Use case

3.3 Đặc tả Use case

3.3.1 Use Case “Đăng nhập”

Tóm tắt: Usecase bắt đầu khi có người dùng muốn đăng nhập vào hệ thống.

Dòng sự kiện chính:

- Hệ thống hiển thị trang đăng nhập.
- Người dùng nhập thông tin email và mật khẩu để đăng nhập vào hệ thống.
- Hệ thống xác thực thông tin đăng nhập.
- Trường hợp xác thực thành công, hệ thống cho phép người dùng thao tác dựa trên quyền của tài khoản đăng nhập. Ngược lại, từ chối đăng nhập.

Dòng sự kiện phụ:

Thiếu thông tin đăng nhập: Hệ thống kiểm tra các trường bắt buộc và yêu cầu người dùng nhập lại đầy đủ thông tin đăng nhập.

Thông tin đăng nhập không đúng:

- **Nhập sai email :** Hệ thống thông báo thông tin đăng nhập không đúng và yêu cầu người dùng nhập lại.
- **Nhập sai mật khẩu:** Hệ thống thông báo mật khẩu không đúng và yêu cầu người dùng nhập lại.
- **Nhập sai cả email và mật khẩu:** Hệ thống thông báo cả email và mật khẩu đều không đúng và yêu cầu người dùng nhập lại.

3. Đăng nhập thất bại:

Tiền điều kiện: Người dùng đã có tài khoản đăng nhập vào hệ thống

Hậu điều kiện: Hệ thống chuyển sang trang giao diện chính của chương trình

Chú thích: Use Case này áp dụng cho các đối tượng người dùng bao gồm: (**admin, customer, nhân viên**)

3.3.2 Use Case “Đăng ký”

Tóm tắt: Use Case bắt đầu khi có người dùng muốn đăng ký tài khoản.

Dòng sự kiện chính:

- Hệ thống hiển thị trang đăng ký.
- Người dùng nhập thông tin email và mật khẩu.
- Hệ thống xác thực thông tin đăng ký.
- Hệ thống lưu trữ thông tin của tài khoản đăng ký xuống CSDL.
- Hệ thống xuất kết quả lưu trữ, cập nhật ra màn hình.
- Trường hợp xác thực thành công, hệ thống thông báo đăng ký thành công. Ngược lại từ chối đăng ký.

Dòng sự kiện phụ:

- Nhập thiếu thông tin: Hệ thống kiểm tra các trường bắt buộc và yêu cầu người dùng nhập lại đầy đủ thông tin để có thể đăng ký.
- Tài khoản đăng ký bị trùng, hoặc không đúng: Hệ thống thông báo tài khoản đăng ký không đúng và yêu cầu người dùng nhập lại.
- Đăng ký thất bại: Hệ thống trả về lỗi đăng ký.

Tiền điều kiện: Người dùng đã có tài khoản mail.

Hậu điều kiện: Hệ thống chuyển sang trang giao diện đăng nhập.

Chú Thích: Use Case này áp dụng cho các đối tượng người dùng bao gồm: (**Khách vãng lai**)

3.3.3 Use Case “Cập nhật thông tin profile”

Tóm tắt: Use Case bắt đầu khi có người dùng muốn cập nhật thông tin profile.

Dòng sự kiện chính:

- Hệ thống hiển thị trang hiển thị thông tin profile của user.
- Người dùng cập nhật các thông tin của profile.
- Hệ thống xác thực thông tin đã cập nhật.
- Hệ thống cập nhật thông tin xuống CSDL.
- Hệ thống xuất kết quả lưu trữ, cập nhật ra màn hình.
- Trường hợp xác thực thành công, hệ thống thông báo cập nhật thông tin profile và đồng thời các thông tin của profile sẽ được cập nhật với thông tin người dùng đã cập nhật trước đó.

Dòng sự kiện phụ:

- Nhập thiếu thông tin: Hệ thống kiểm tra các trường bắt buộc và yêu cầu người dùng nhập lại đầy đủ thông tin để khi cập nhật.
- Cập nhật thất bại: Hệ thống trả về lỗi cập nhật.

Tiền điều kiện: Người dùng đã có tài khoản và đăng nhập vào website.

Hậu điều kiện: Hệ thống chuyển sang trang giao diện home của website.

3.3.4 Use Case “Thêm thông tin phòng”

Tóm tắt: Use Case bắt đầu khi người dùng chọn chức năng thêm phòng từ trang giao diện chính.

Dòng sự kiện chính:

- Hệ thống hiển thị giao diện thêm mới phòng.
- Người dùng nhập thông tin phòng mới

- Hệ thống kiểm tra và lưu trữ thông tin phòng mới vào CSDL.
- Hệ thống thông báo kết quả lưu trữ cho người dùng

Dòng sự kiện phụ:

- Người dùng nhập thiếu thông tin: Hệ thống kiểm tra các trường thông tin bắt buộc và yêu cầu người dùng nhập lại đầy đủ.
- Tạo mới phòng thất bại: Hệ thống trả về lỗi.

Tiền điều kiện: Người dùng đã đăng nhập thành công vào hệ thống với vai trò là admin.

Hậu điều kiện: Thông tin phòng mới được lưu trữ vào CSDL.

Chú thích: Các Use Case có quy trình tương tự : (Use Case thêm nhân viên, Use Case thêm loại phòng, Use Case thêm dịch vụ phòng)

3.3.5 Use Case “Sửa thông tin phòng”

Tóm tắt: Use Case bắt đầu khi người dùng chọn chức năng sửa thông tin phòng từ trang giao diện chính.

Dòng sự kiện chính:

- Hệ thống hiển thị danh sách các phòng hiện có.
- Người dùng chọn phòng cần sửa và hệ thống hiển thị giao diện sửa thông tin.
- Người dùng cập nhật thông tin phòng
- Hệ thống lưu trữ thông tin phòng đã cập nhật vào CSDL.
- Hệ thống thông báo kết quả cập nhật cho người dùng.

Dòng sự kiện phụ:

- Người dùng nhập thiếu thông tin: Hệ thống kiểm tra các trường thông tin bắt buộc và yêu cầu người dùng nhập lại đầy đủ.
- Cập nhật phòng thất bại: Hệ thống trả về lỗi.

Tiền điều kiện: Người dùng đã đăng nhập thành công vào hệ thống với vai trò là admin.

Hậu điều kiện: Thông tin phòng được cập nhật trong CSDL.

Chú thích: Các Use Case có quy trình tương tự:

- **Use Case sửa thông tin nhân viên,**
- **Use Case sửa thông tin loại phòng,**
- **Use Case sửa thông tin dịch vụ phòng**

Use Case Xóa thông tin phòng

Tóm tắt: Use Case bắt đầu khi người dùng chọn chức năng xóa phòng từ trang giao diện chính.

Dòng sự kiện chính:

- Hệ thống hiển thị danh sách các phòng hiện có.
- Người dùng chọn phòng cần xóa.
- Hệ thống yêu cầu xác nhận việc xóa phòng.
- Người dùng xác nhận xóa phòng.
- Hệ thống xóa thông tin phòng khỏi CSDL.
- Hệ thống thông báo kết quả xóa cho người dùng.

Dòng sự kiện phụ:

- Xóa phòng thất bại: Hệ thống trả về lỗi.

Tiền điều kiện: Người dùng đã đăng nhập thành công vào hệ thống với vai trò là admin.

Hậu điều kiện: Thông tin phòng được xóa khỏi CSDL.

Chú thích: Các Use Case có quy trình tương tự:

- **Use Case xóa thông tin nhân viên,**

- **Use Case xóa thông tin loại phòng,**
- **Use Case xóa thông tin dịch vụ phòng,**
- **Use Case xóa đánh giá**

3.3.6 Use Case “Xem thông tin phòng”

Tóm tắt: Use Case bắt đầu khi người dùng chọn chức năng xem thông tin phòng từ trang giao diện chính.

Dòng sự kiện chính:

- Hệ thống lấy dữ liệu danh sách các phòng từ CSDL và hiển thị ra màn hình.
- Người dùng chọn một phòng cụ thể để xem chi tiết.
- Hệ thống hiển thị thông tin chi tiết của phòng đã chọn.

Tiền điều kiện: Người dùng đã đăng nhập thành công vào hệ thống với vai trò là admin, nhân viên

Hậu điều kiện: Người dùng có thể xem chi tiết thông tin của phòng đã chọn.

Chú thích: Các Use Case có quy trình tương tự:

- **Use Case xem thông tin nhân viên,**
- **Use Case xem thông tin loại phòng,**
- **Use Case xem thông tin dịch vụ phòng,**
- **Use Case xem đánh giá**

3.3.7 Use Case “Cập nhật thông tin phòng”

Tóm tắt: Use Case bắt đầu khi người dùng chọn chức năng cập nhật thông tin phòng từ trang giao diện chính.

Dòng sự kiện chính:

- Hệ thống hiển thị danh sách các phòng hiện có.

- Người dùng chọn phòng cần cập nhật.
- Hệ thống hiển thị giao diện cập nhật thông tin phòng.
- Người dùng nhập các thông tin cập nhật.
- Hệ thống lưu trữ thông tin phòng đã cập nhật vào CSDL.
- Hệ thống thông báo kết quả cập nhật cho người dùng.

Dòng sự kiện phụ:

- Người dùng nhập thiếu thông tin: Hệ thống kiểm tra các trường thông tin bắt buộc và yêu cầu người dùng nhập lại đầy đủ.
- Hệ thống thông báo kết quả cập nhật cho người dùng.

Tiền điều kiện: Người dùng đã đăng nhập thành công vào hệ thống với vai trò là admin.

Hậu điều kiện: Thông tin phòng được cập nhật trong CSDL.

Chú thích: Các Use Case có quy trình tương tự:

- **Use Case cập nhật thông tin nhân viên,**
- **Use Case cập nhật thông tin loại phòng,**
- **Use Case cập nhật thông tin dịch vụ phòng**

3.3.8 Use Case “Xem phòng đã đặt”

Tóm tắt: Use Case bắt đầu khi người dùng chọn thực hiện quản lý đặt phòng từ trang giao diện quản lý của admin. Hệ thống sẽ xử lý yêu cầu của người dùng.

Dòng sự kiện chính:

- Hệ thống lấy dữ liệu danh sách các thông tin của các phòng đã đặt từ CSDL và hiển thị ra màn hình
- Người dùng chọn menu trạng thái phù hợp để xem những phòng đã được đặt

- Hệ thống hiển thị giao diện danh sách phòng đã đặt

Dòng sự kiện phụ:

- Lấy dữ liệu thất bại hệ thống trả về lỗi.

Tiền điều kiện: Người dùng đã đăng nhập thành công vào hệ thống với vai trò là admin, nhân viên

Hậu điều kiện: Hệ thống lưu trữ load danh sách phòng đã đặt theo trạng thái cụ thể mà người dùng muốn xem.

3.3.9 Use Case “Xác nhận đặt phòng”

Tóm tắt: Use Case bắt đầu khi người dùng muốn xác nhận việc đặt phòng từ trang giao diện chính.

Dòng sự kiện chính:

- Người dùng chọn chức năng "Xác nhận đặt phòng" từ trang giao diện chính.
- Hệ thống hiển thị danh sách các phòng đã đặt nhưng chưa được xác nhận.
- Người dùng chọn một phòng cần xác nhận từ danh sách.
- Hệ thống hiển thị chi tiết thông tin phòng đã đặt.
- Người dùng xác nhận việc đặt phòng.
- Hệ thống lưu trữ trạng thái xác nhận vào CSDL.
- Hệ thống thông báo kết quả xác nhận cho người dùng và cập nhật danh sách các phòng đã xác nhận.

Dòng sự kiện phụ:

- **Lấy dữ liệu thất bại:** Hệ thống trả về lỗi và yêu cầu người dùng thử lại.

- **Xác nhận thất bại:** Hệ thống trả về lỗi khi không thể lưu trữ trạng thái xác nhận và yêu cầu người dùng thử lại.

Tiền điều kiện: Người dùng đã truy cập vào trang web với vai trò admin hoặc nhân viên

Hậu điều kiện: Hệ thống hiển thị danh sách các phòng đã được xác nhận.

Chú thích: Các Use Case có quy trình tương tự:

- **Use Case hủy xác nhận đặt phòng,**
- **Use Case nhận thanh toán,**
- **Use Case hoàn thành phòng**

3.3.10 Use Case “Đặt phòng”

Tóm tắt: Use Case thực hiện người dùng tiến hành lựa chọn phòng muốn đặt.

Dòng sự kiện chính:

- Hệ thống lấy dữ liệu danh sách các phòng phù hợp với điều kiện tìm kiếm.
- Người dùng tiến hành xem phòng và tiến hành thanh toán online hoặc thanh toán tại khách sạn.
- Sau đó nhập thông tin cần thiết để tiến hành thanh toán.
- Hệ thống kiểm tra các thông tin cần thiết cập nhật lưu trữ thông tin đặt phòng của khách hàng xuống CSDL.
- Hệ thống xuất kết quả ra màn hình.

Dòng sự kiện phụ:

- Lấy dữ liệu thất bại hệ thống trả về lỗi.

- Người dùng nhập thiếu thông tin: Hệ thống kiểm tra các trường thông tin bắt buộc và yêu cầu người dùng nhập lại đầy đủ.
- Lưu dữ liệu thất bại: Hệ thống trả về lỗi
- Cập nhật dữ liệu thất bại: Hệ thống trả về lỗi

Tiền điều kiện: Người dùng đã đăng nhập thành công vào hệ thống với vai trò là khách hàng (customer).

Hậu điều kiện: Hệ thống lưu trữ và cập nhật thông tin xuống cơ sở dữ liệu.

3.3.11 Use Case “Xem trang chủ”

Tóm tắt

- Tác nhân: khách hàng, khách vãng lai
- Mô tả: cho phép khách hàng xem trang chủ website

Dòng sự kiện

- Dòng sự kiện chính
 - Khách hàng nhập tên miền
 - Lấy dữ liệu trang chủ
 - Truyền vào dữ liệu trang chủ cần lấy
 - Hệ thống trả về giao diện trang chủ

- Các dòng sự kiện khác

Không có

- Các yêu cầu đặc biệt

Không có

- Trạng thái hệ thống khi bắt đầu thực hiện Use Case

- Khách hàng không cần đăng nhập vào website

- Trạng thái hệ thống sau khi thực hiện Use Case
 - Thực hiện thành công khách hàng sẽ vào được trang chủ của website
- Điểm mở rộng
 - Không có

3.3.12 Use Case “Thông kê doanh thu”

Tóm tắt: Use Case thực hiện thống kê doanh thu của website với số lượng phòng được đặt và đã hoàn thành.

Dòng sự kiện chính:

- Hệ thống hiển thị trang lập dashboard của quản lý admin với biểu đồ doanh thu.
- Hệ thống tổng hợp dữ liệu, lập báo cáo thống kê theo yêu cầu của người dùng.

Dòng sự kiện phụ:

- Lấy dữ liệu thất bại hệ thống trả về lỗi.

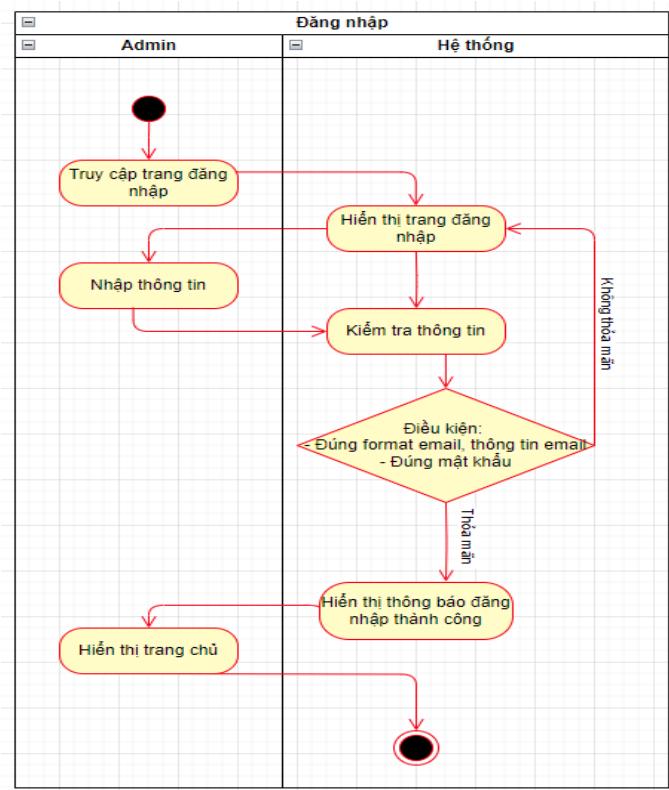
Tiền điều kiện: Người dùng đã đăng nhập thành công vào hệ thống với vai trò là admin, nhân viên

Hậu điều kiện: Hệ thống hiển thị biểu đồ cùng với tổng doanh thu của hệ thống.

3.4 Sơ đồ hoạt động (Activity)

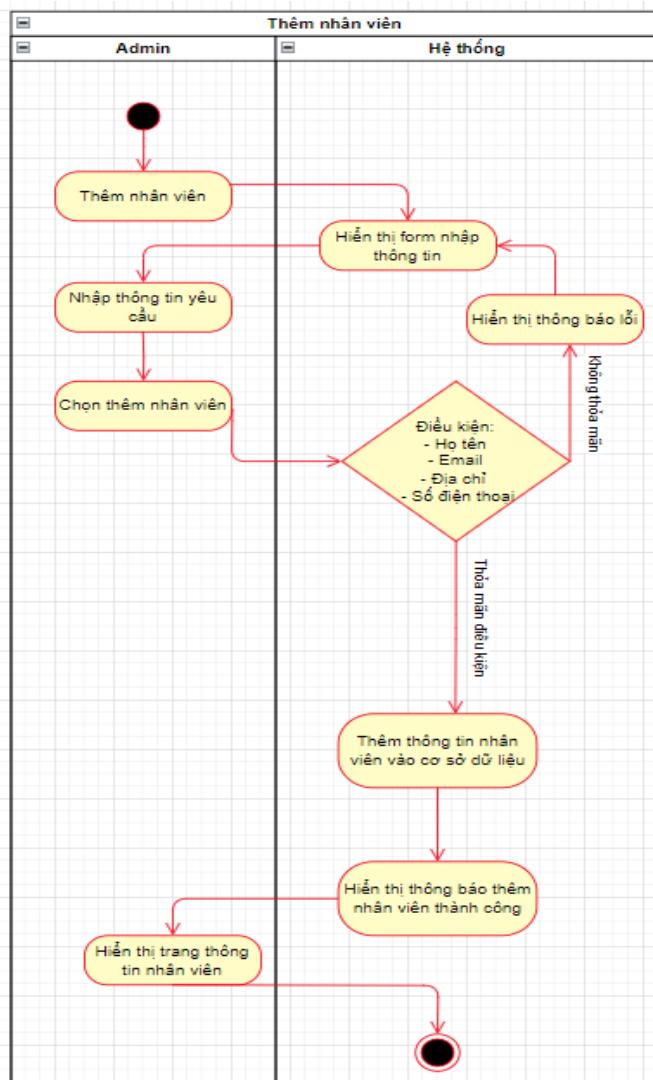
3.4.1 Quản trị viên (Admin)

- Sơ đồ activity đăng nhập



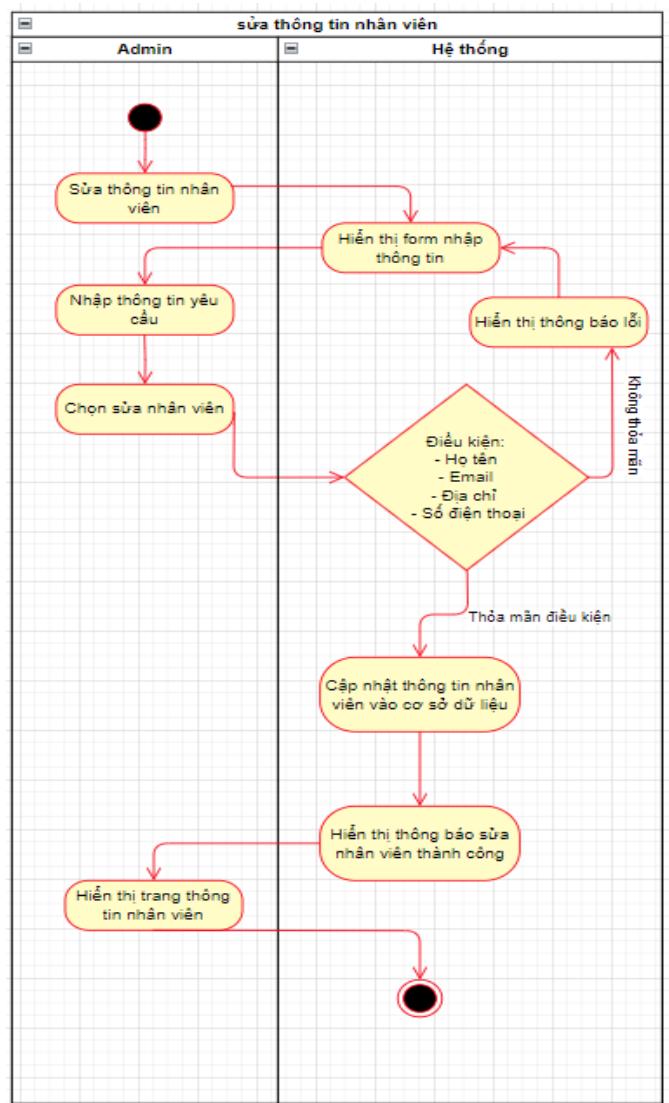
Hình 3.6 Sơ đồ activity đăng nhập

- Sơ đồ activity thêm thành viên



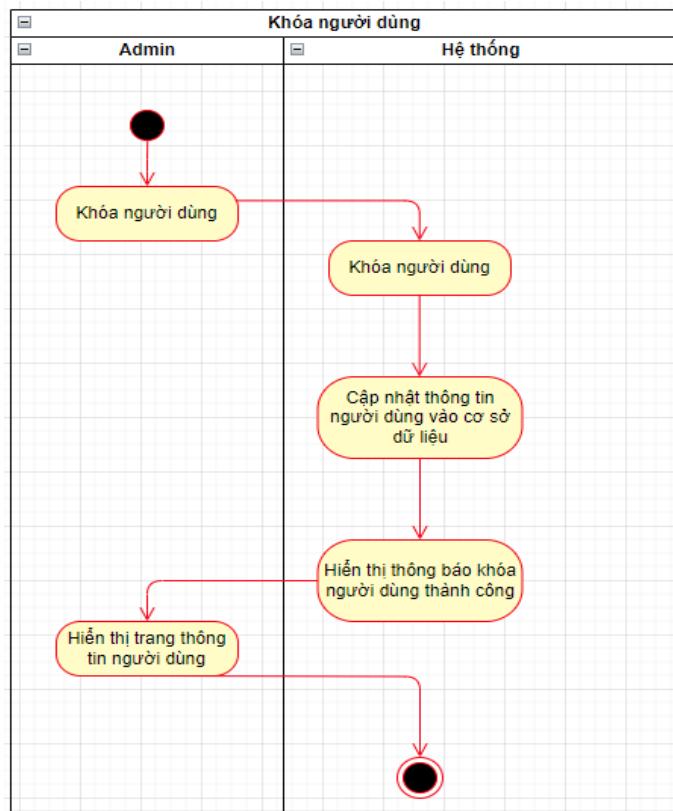
Hình 3.7 Sơ đồ activity thêm thành viên

- Sơ đồ activity sửa thông tin thành viên



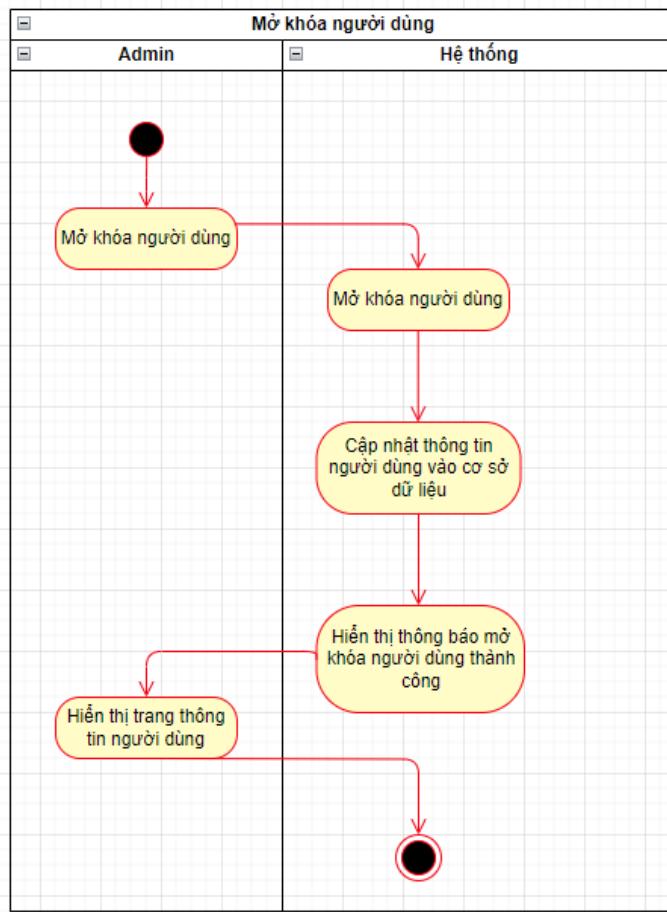
Hình 3.8 Sơ đồ activity sửa thành viên

- Sơ đồ activity khoá người dùng



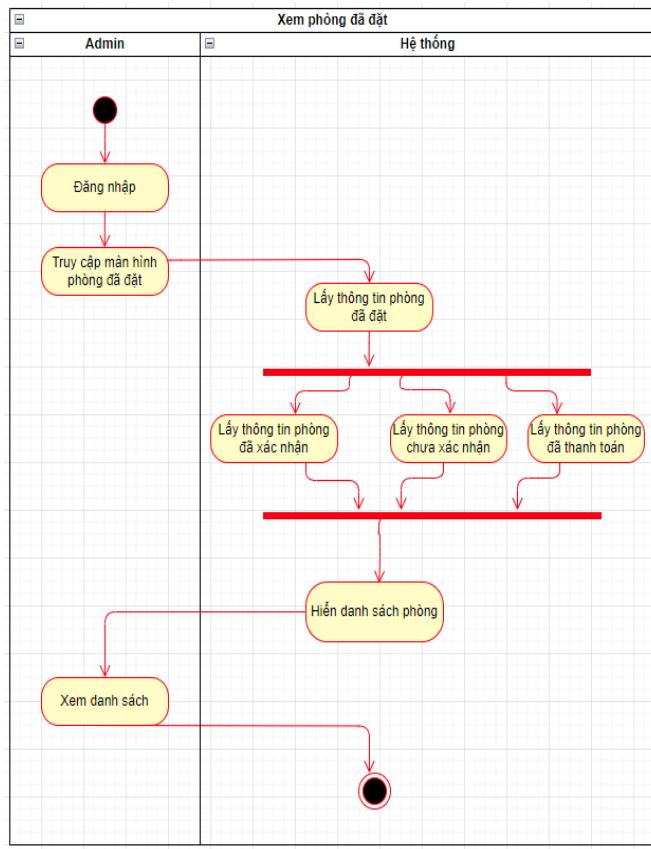
Hình 3.9 Sơ đồ activity khoá người dùng

- Sơ đồ activity mở khoá người dùng



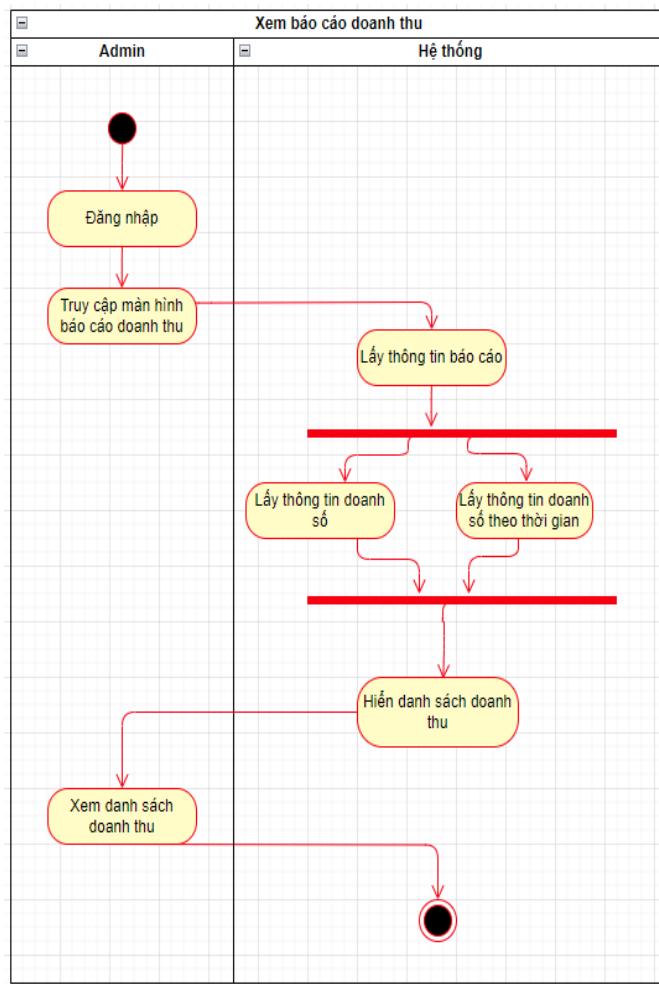
Hình 3.10 Sơ đồ activity mở khoá người dùng

- Sơ đồ activity xem phòng khách hàng đã đặt



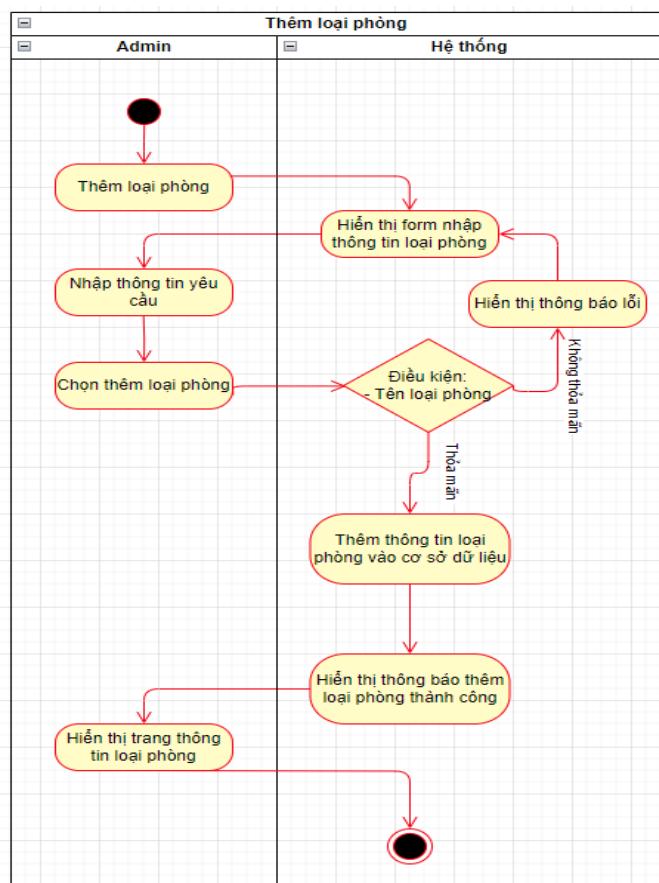
Hình 3.11 Sơ đồ activity xem phòng khách hàng đã đặt

- Sơ đồ activity xem báo cáo doanh thu



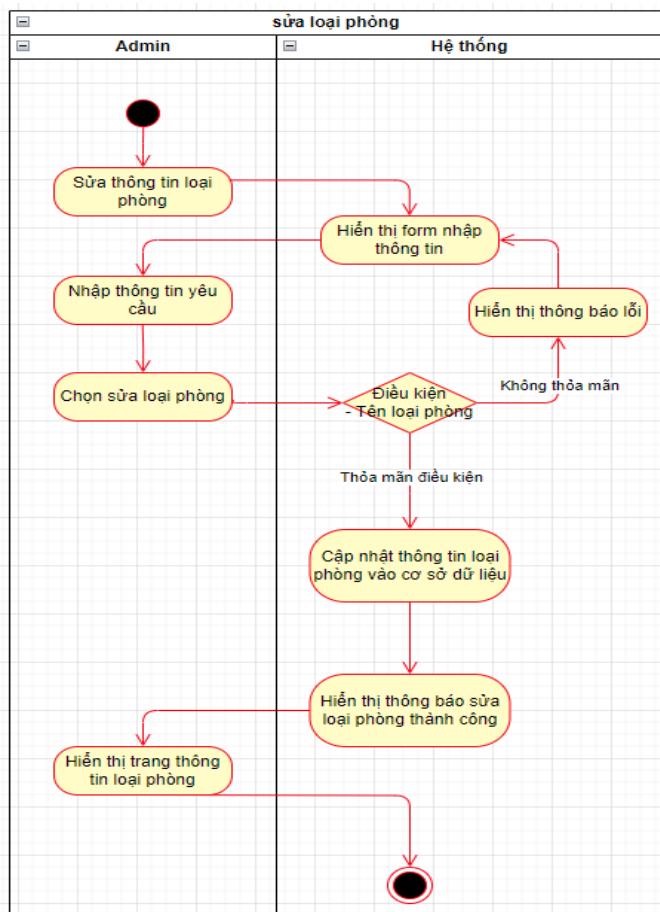
Hình 3.12 Sơ đồ activity xem báo cáo doanh thu

- Sơ đồ activity thêm loại phòng



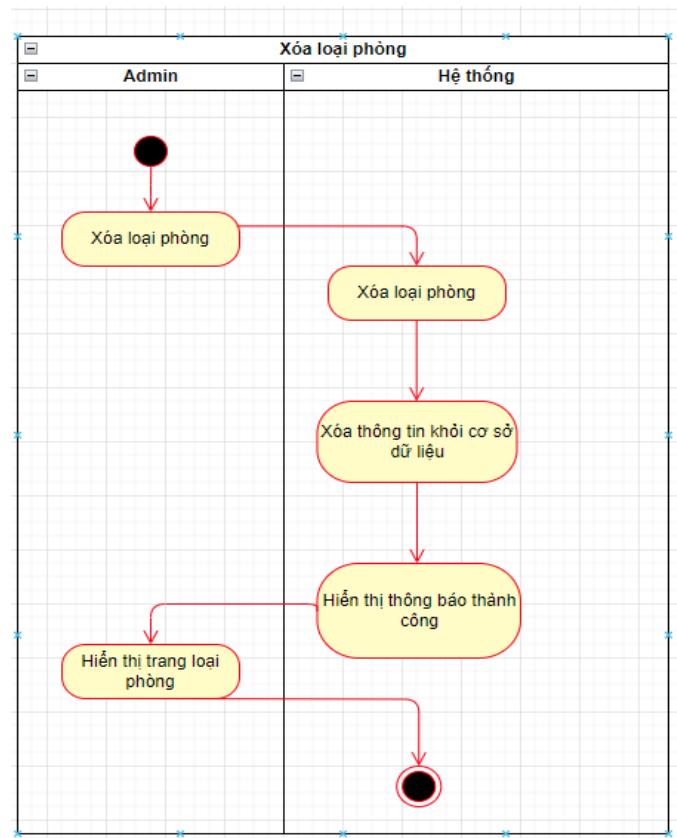
Hình 3.13 Sơ đồ activity thêm loại phòng

- Sơ đồ activity sửa loại phòng



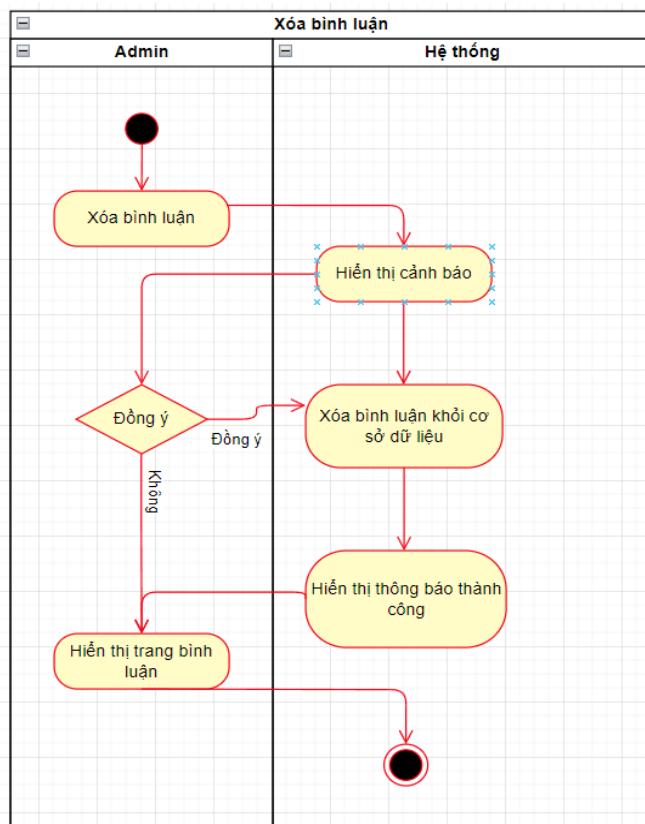
Hình 3.14 Sơ đồ activity sửa loại phòng

- Sơ đồ activity xoá loại phòng



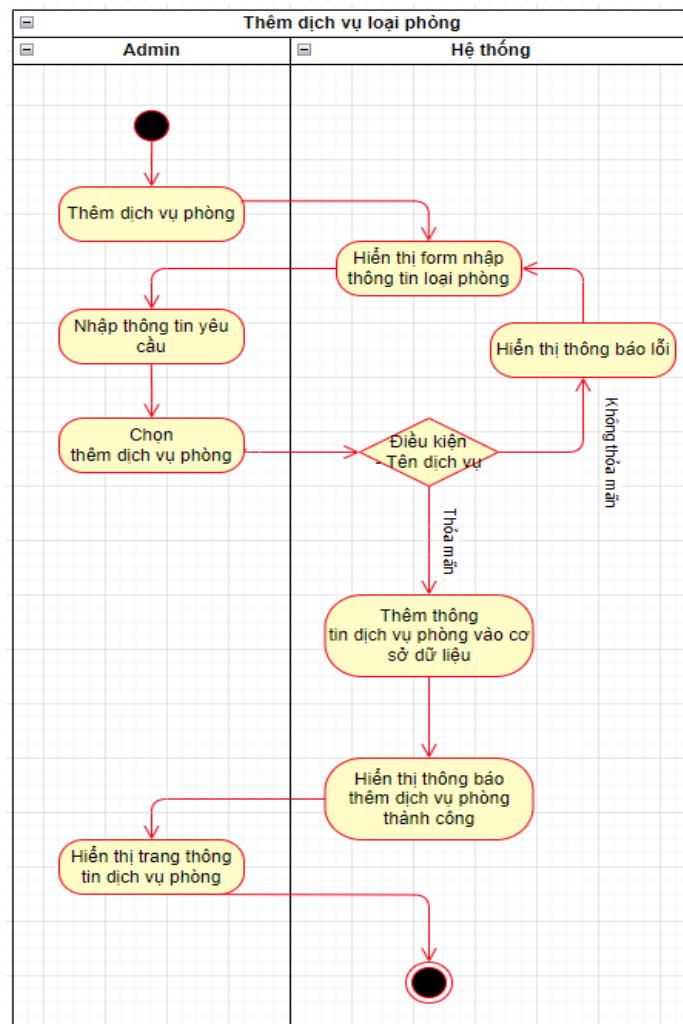
Hình 3.15 Sơ đồ activity xoá loại phòng

- Sơ đồ activity xoá bình luận



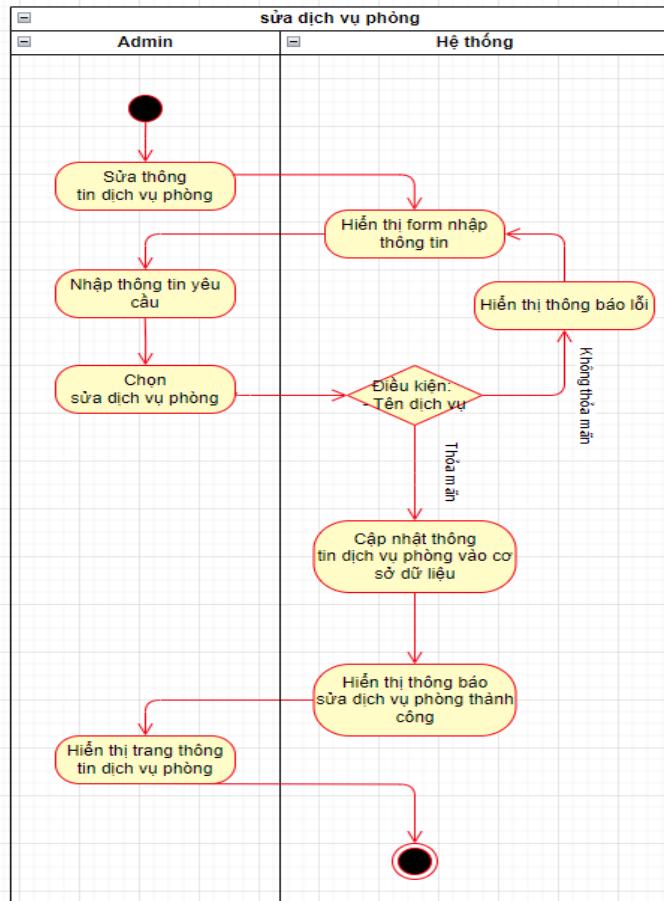
Hình 3.16 Sơ đồ activity xoá bình luận

- Sơ đồ activity thêm dịch vụ phòng



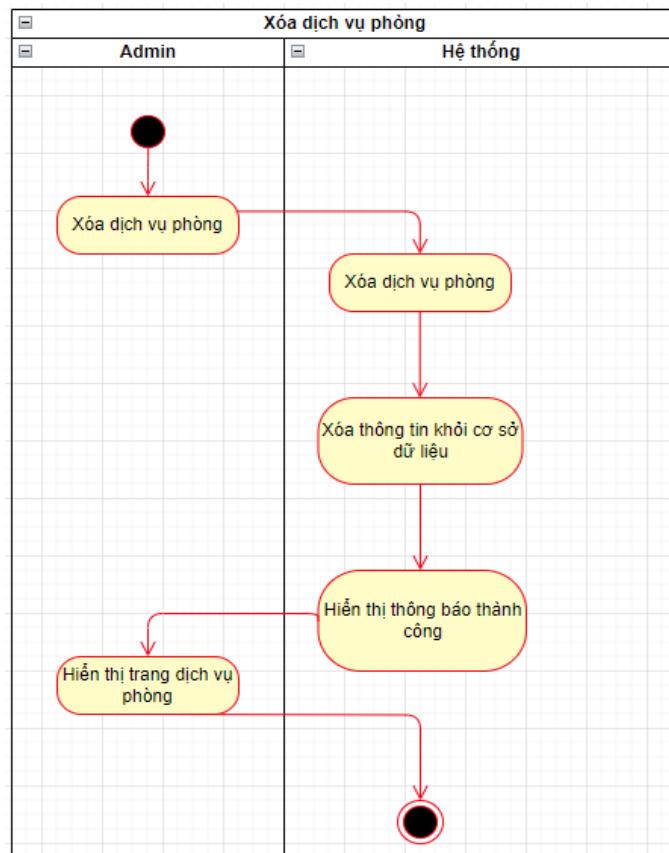
Hình 3.17 Sơ đồ activity thêm dịch vụ phòng

- Sơ đồ activity sửa dịch vụ phòng



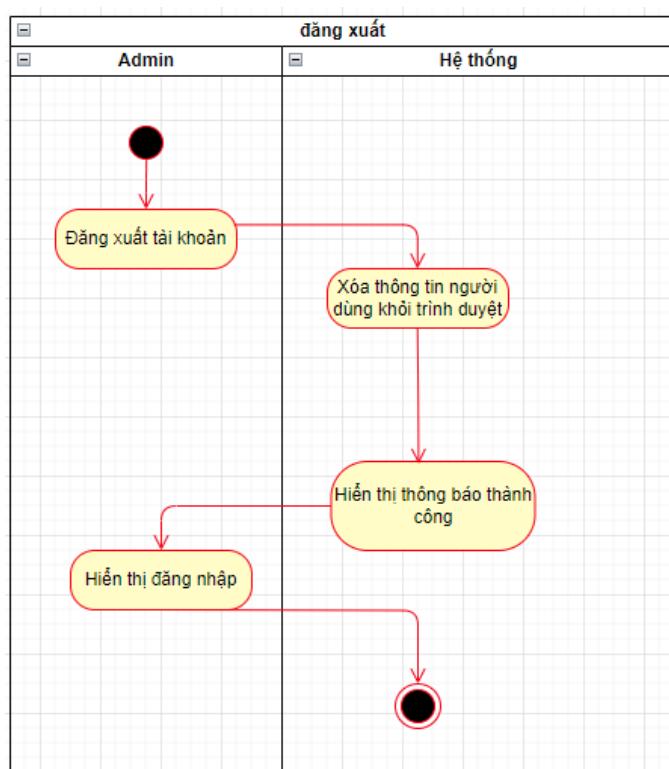
Hình 3.18 Sơ đồ activity sửa dịch vụ phòng

- Sơ đồ activity xoá dịch vụ phòng



Hình 3.19 Sơ đồ activity xoá dịch vụ phòng

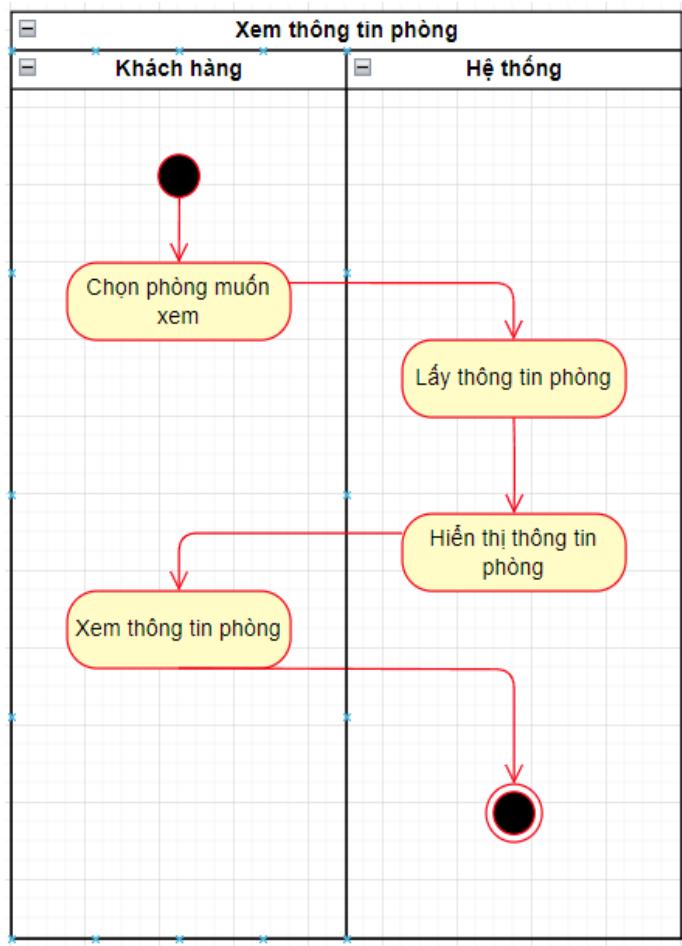
- Sơ đồ activity đăng xuất



Hình 3.20 Sơ đồ activity đăng xuất

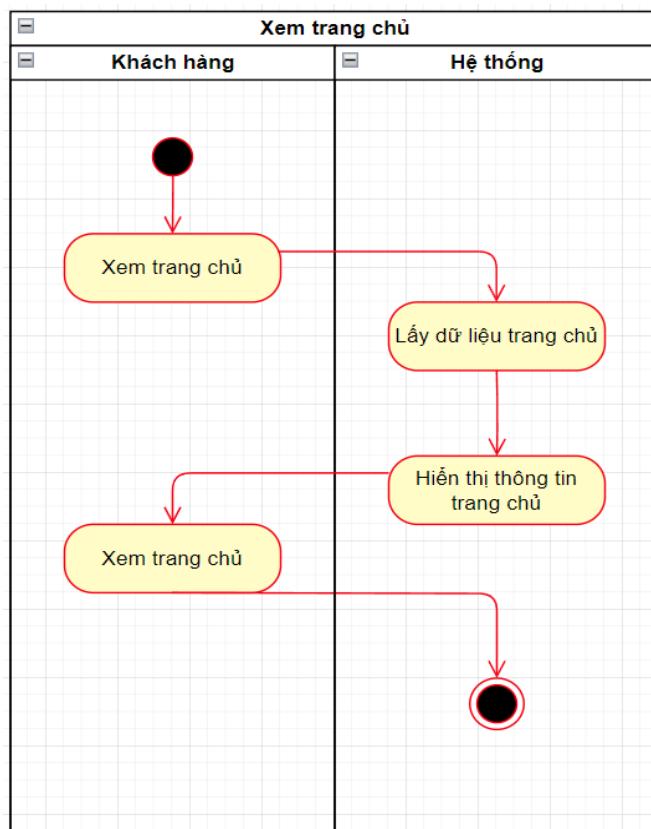
3.4.2 Khách hàng (Customer)

- Sơ đồ activity xem thông tin phòng



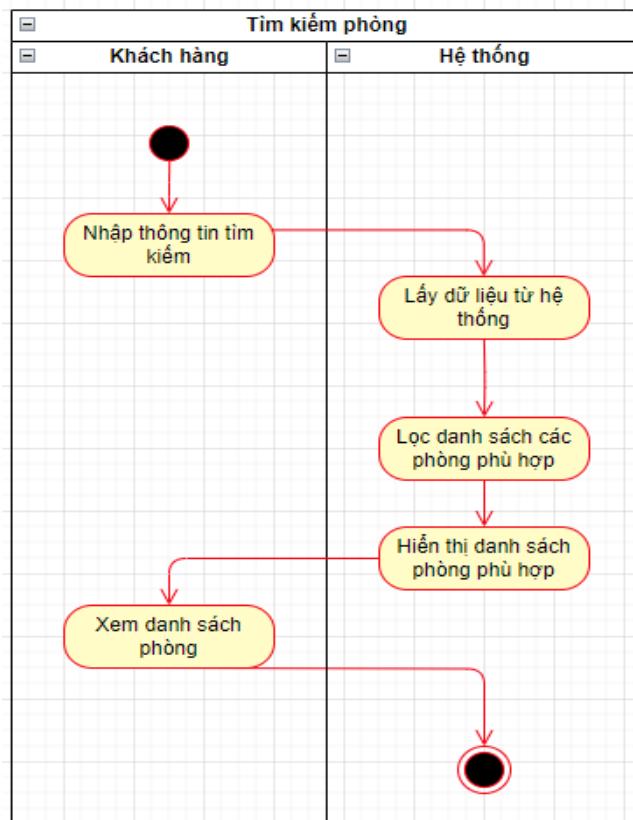
Hình 3.21 Sơ đồ activity xem thông tin phòng

- Sơ đồ activity xem trang chủ



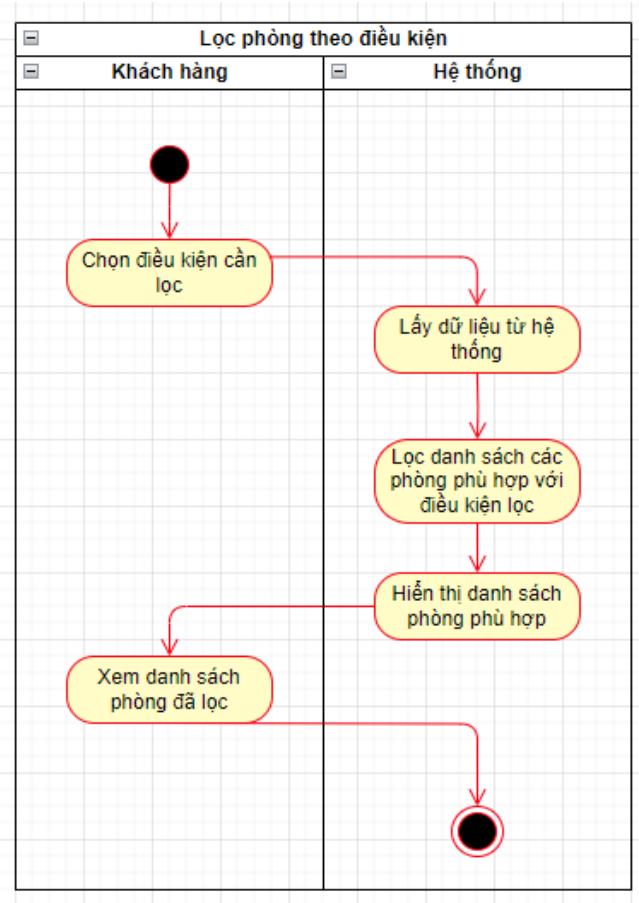
Hình 3.22 Sơ đồ activity xem trang chủ

- Sơ đồ activity tìm kiếm phòng



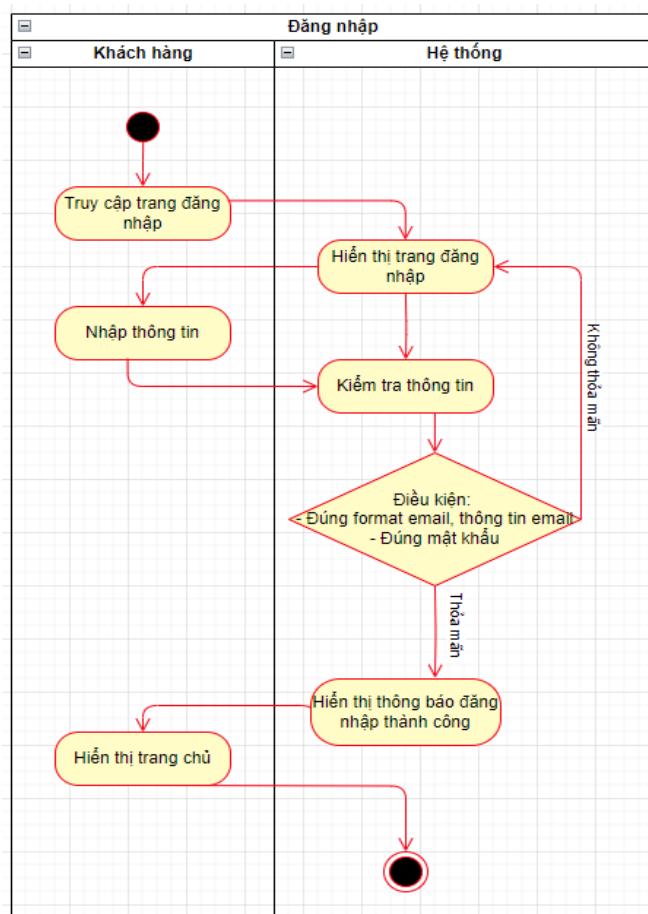
Hình 3.23 Sơ đồ activity tìm kiếm phòng

- Sơ đồ activity lọc phòng theo điều kiện



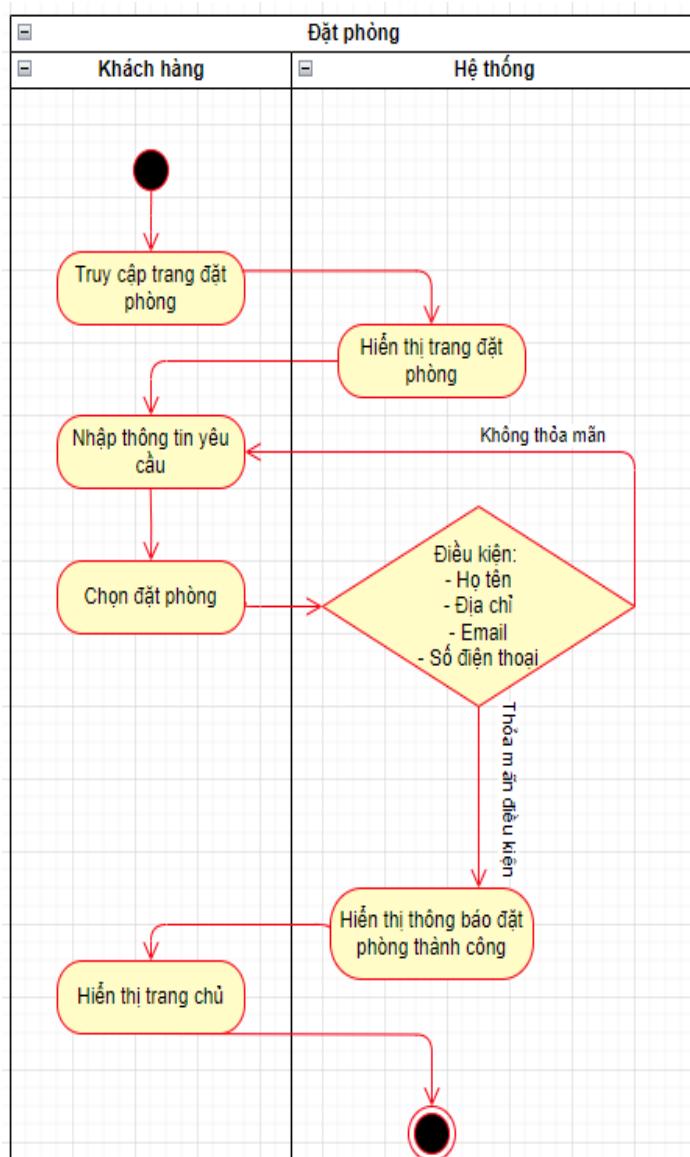
Hình 3.24 Sơ đồ activity lọc phòng điều kiện

- Sơ đồ activity đăng nhập



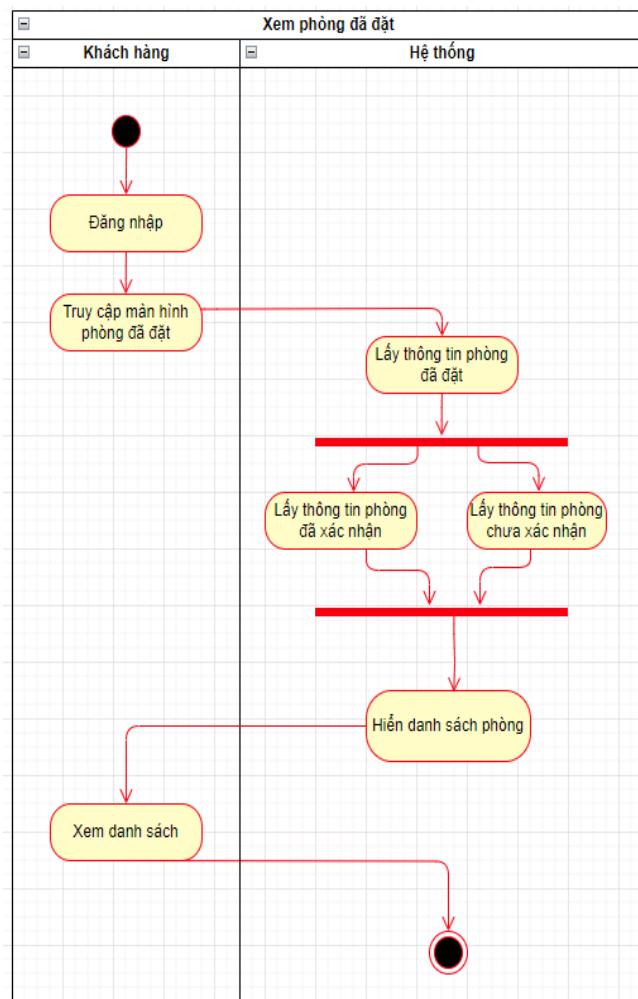
Hình 3.25 Sơ đồ activity đăng nhập

- Sơ đồ activity đặt phòng



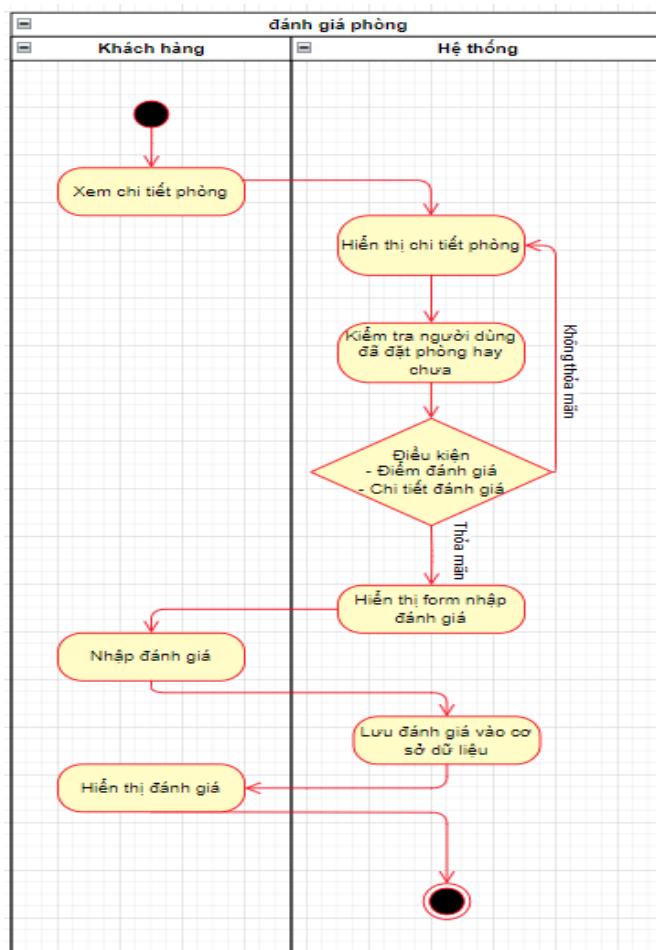
Hình 3.26 Sơ đồ activity đặt phòng

- Sơ đồ activity xem phòng đã đặt



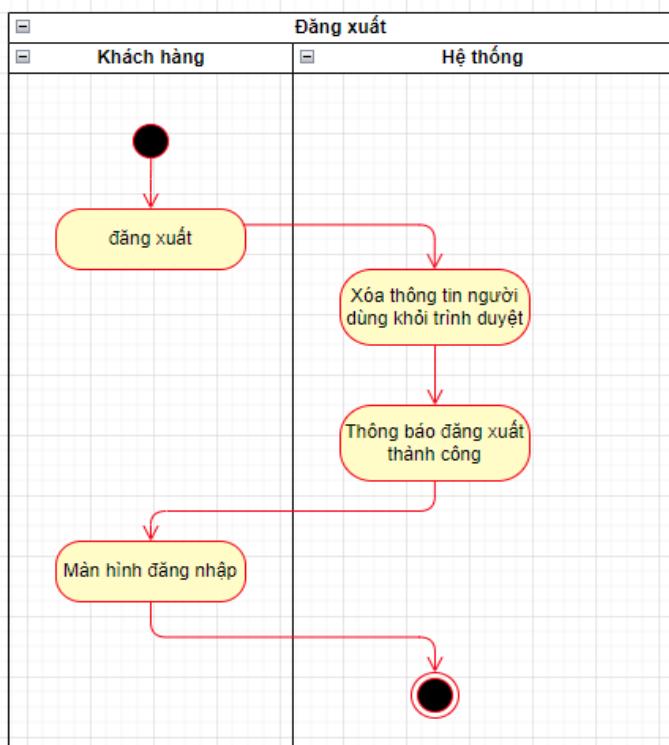
Hình 3.27 Sơ đồ activity xem phòng đã đặt

- Sơ đồ activity đánh giá phòng



Hình 3.28 Sơ đồ activity đánh giá phòng

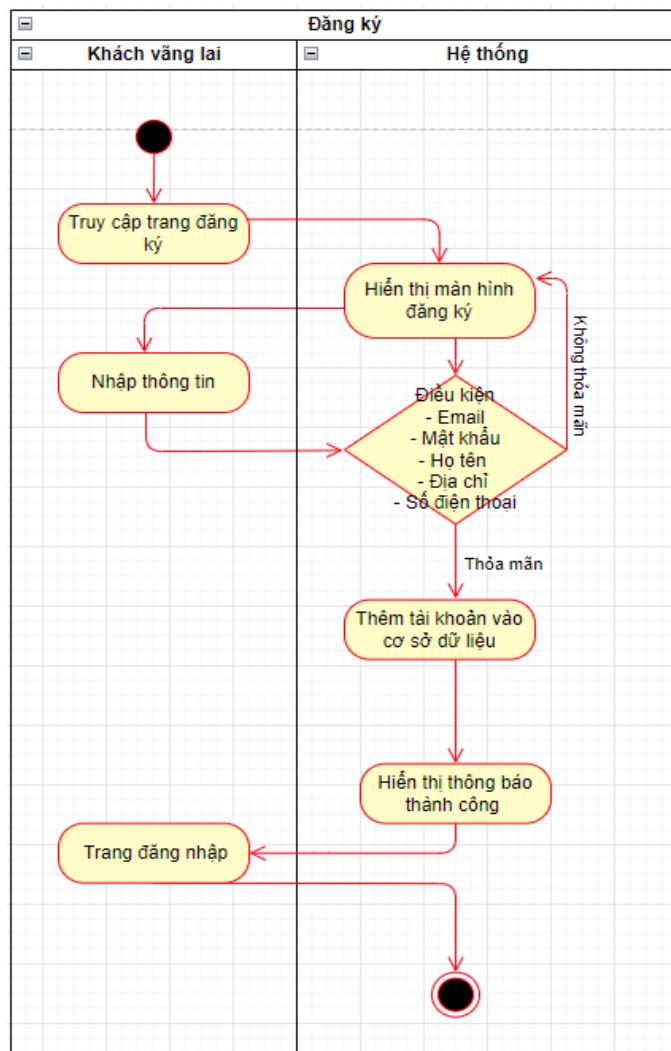
- Sơ đồ activity đăng xuất



Hình 3.29 Sơ đồ activity đăng xuất

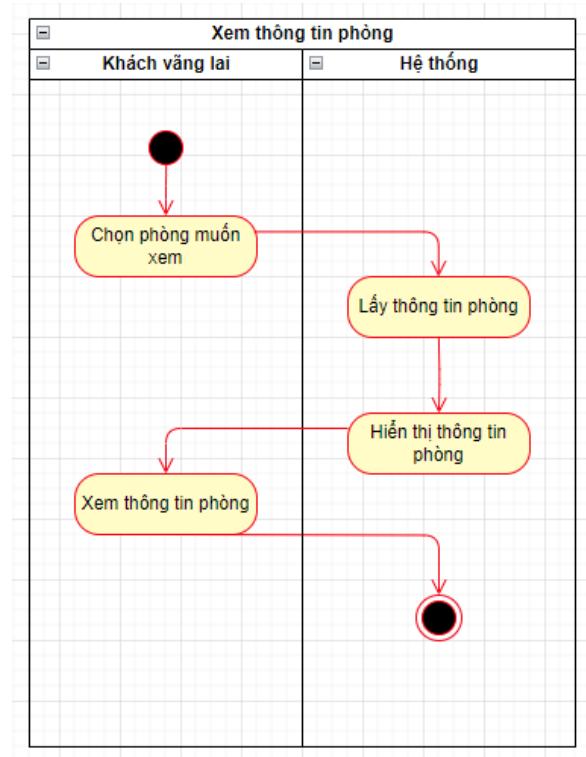
3.4.3 Khách vãng lai (Guest)

- Sơ đồ activity đăng ký



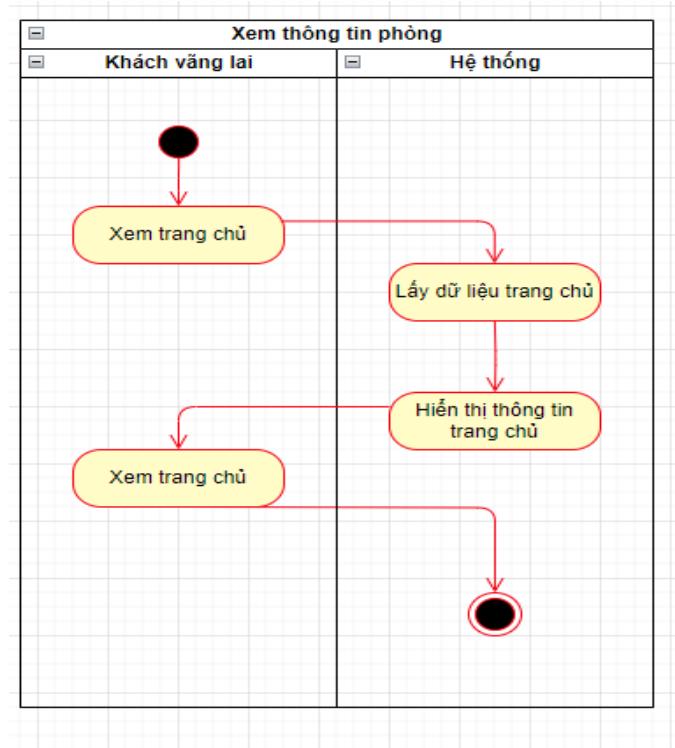
Hình 3.30 Sơ đồ activity đăng ký

- Sơ đồ activity xem trang chủ



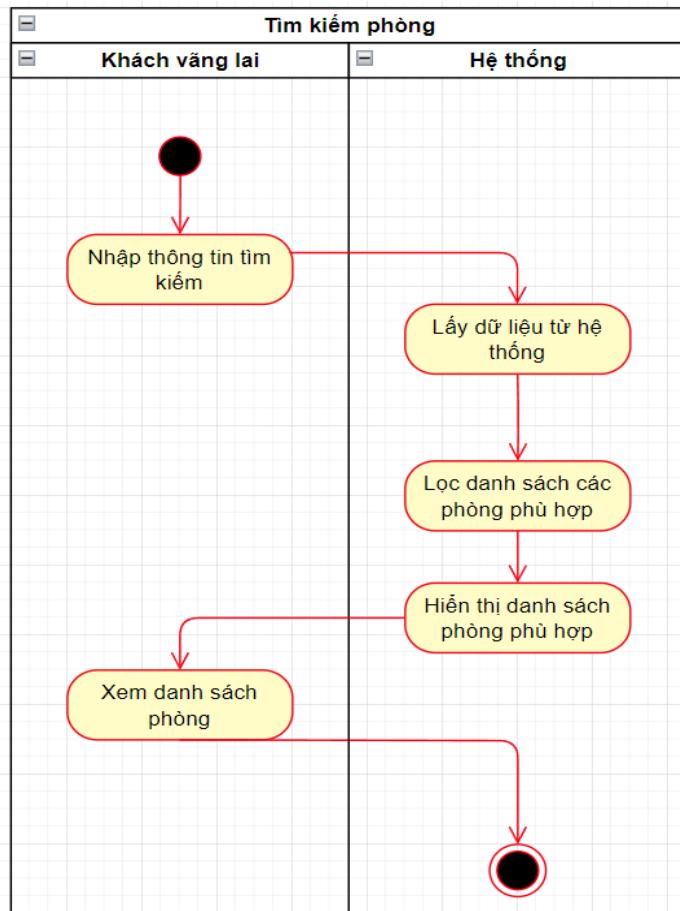
Hình 3.31 Sơ đồ activity xem trang chủ

- Sơ đồ activity xem thông tin phòng



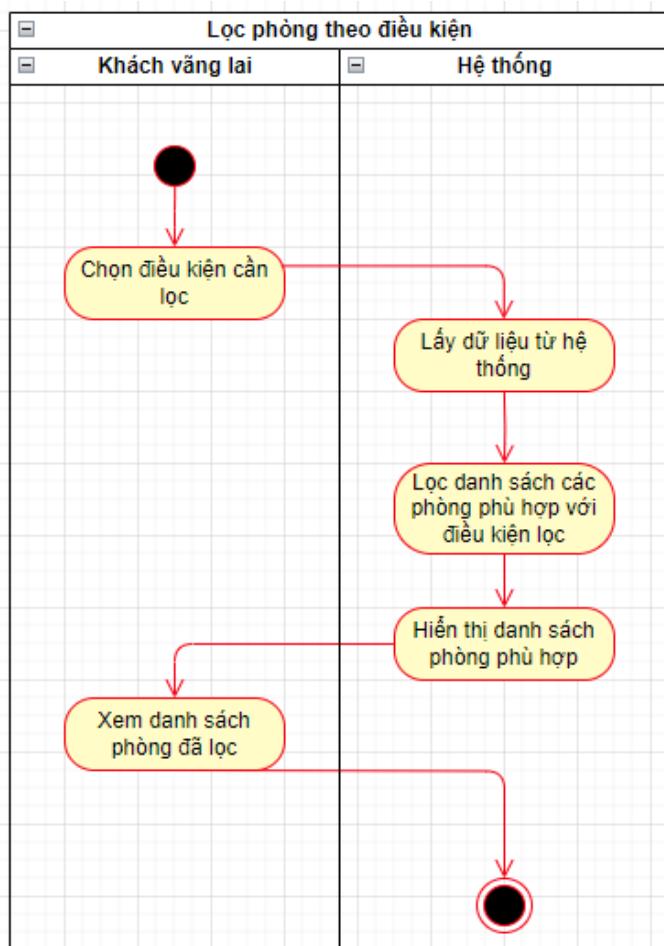
Hình 3.32 Sơ đồ activity xem thông tin phòng

- Sơ đồ activity tìm kiếm phòng



Hình 3.33 Sơ đồ activity tìm kiếm phòng

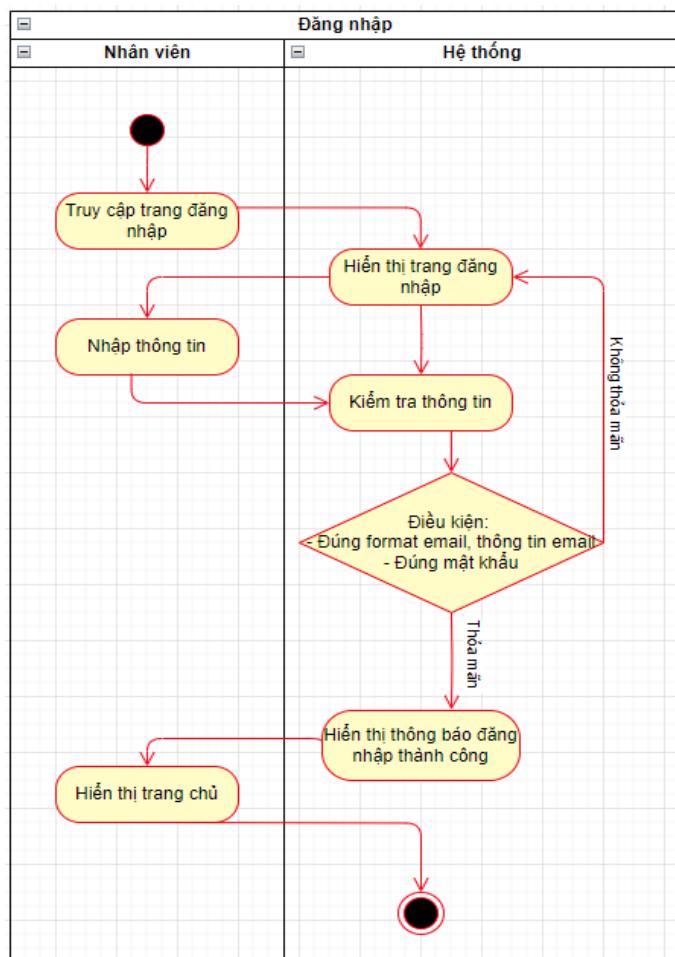
- Sơ đồ activity lọc phòng theo điều kiện



Hình 3.34 Sơ đồ activity lọc phòng theo điều kiện

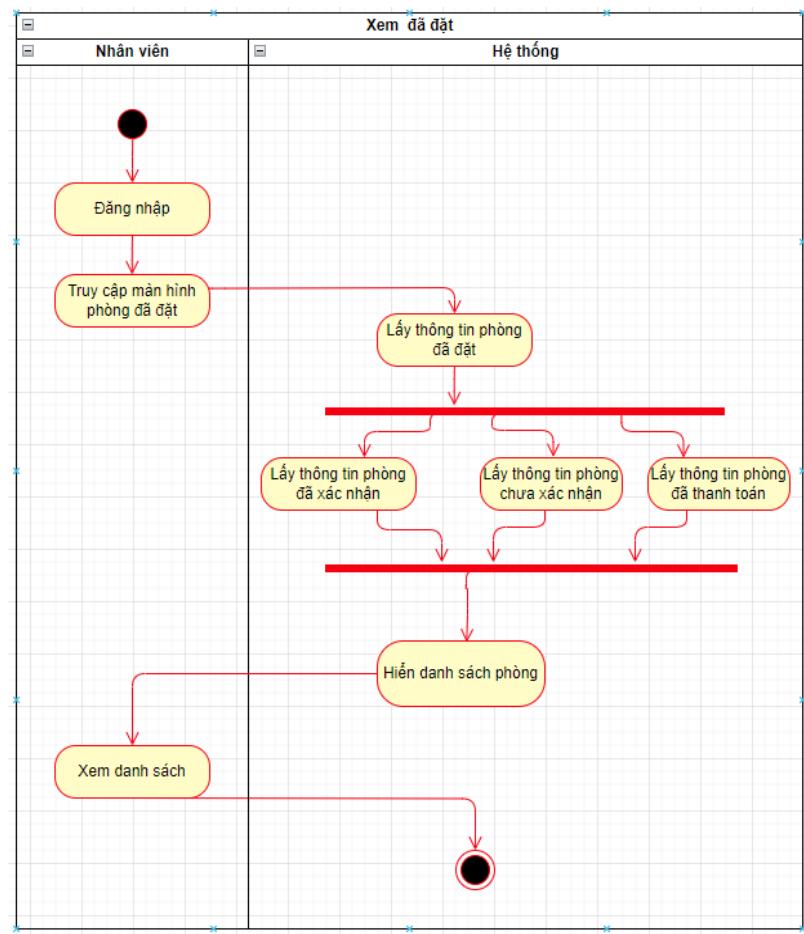
3.4.4 Nhân viên(Staff)

- Sơ đồ activity đăng nhập



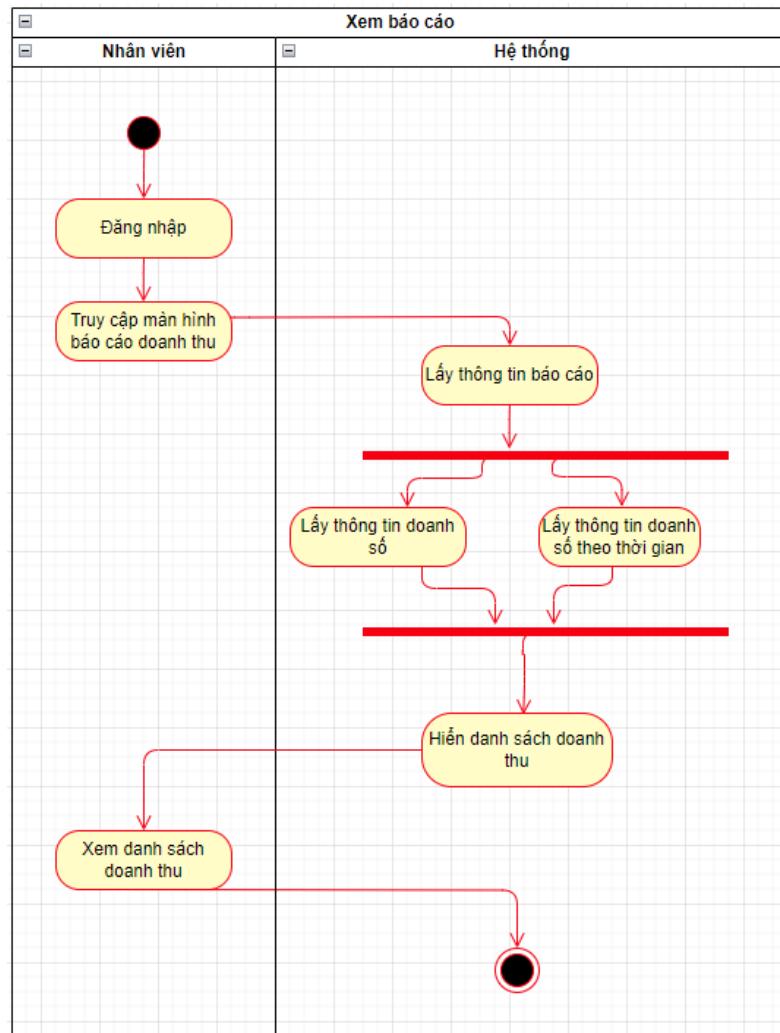
Hình 3.35 Sơ đồ activity đăng nhập

- Sơ đồ activity xem phòng khách hàng đã đặt



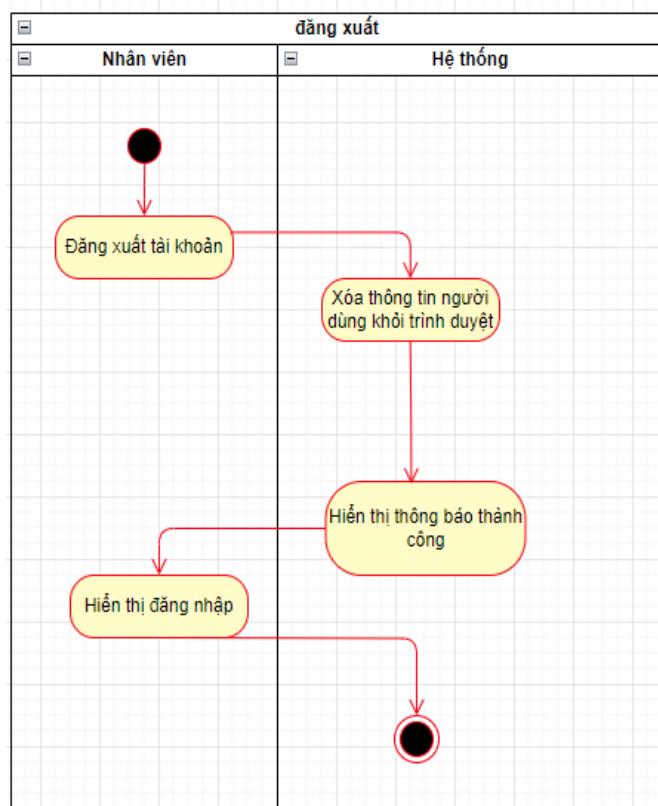
Hình 3.36 Sơ đồ activity xem phòng khách hàng đã đặt

- Sơ đồ activity xem báo cáo



Hình 3.37 Sơ đồ activity xem báo cáo

- Sơ đồ activity đăng xuất

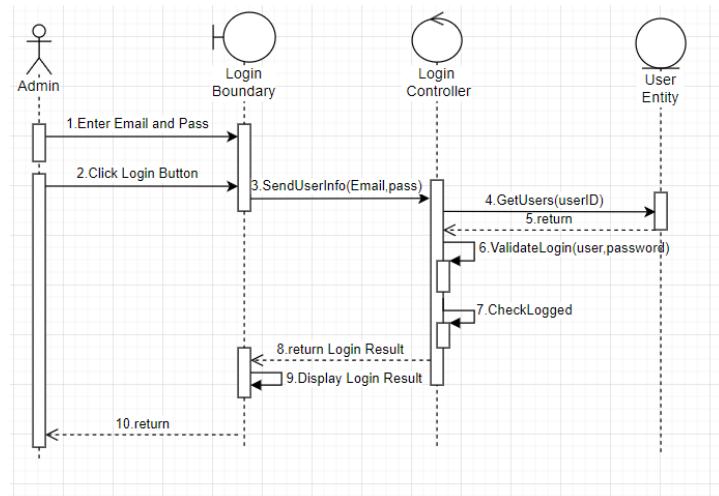


Hình 3.38 Sơ đồ activity đăng xuất

3.5 Sơ đồ tuần tự (Sequence)

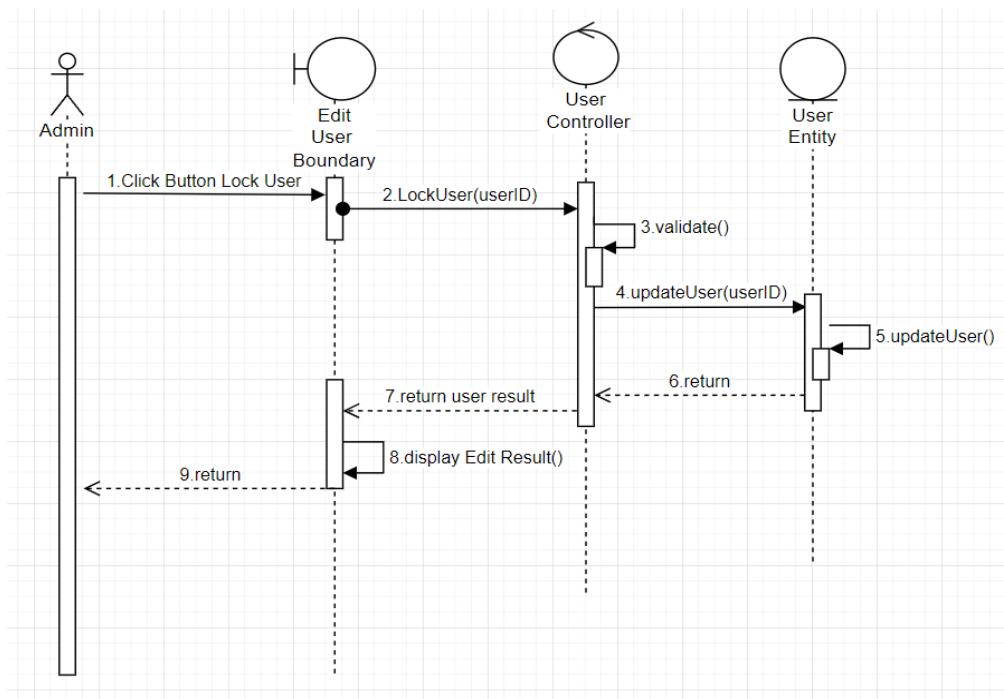
3.5.1 Quản trị viên (Admin)

- Sơ đồ Sequence đăng nhập



Hình 3.39 Sơ đồ Sequence đăng nhập

- Mô tả: cho phép admin đăng nhập hệ thống (tương tự với **nhân viên** và **khách hàng (customer)**)
- Sơ đồ Sequence khoá khách hàng

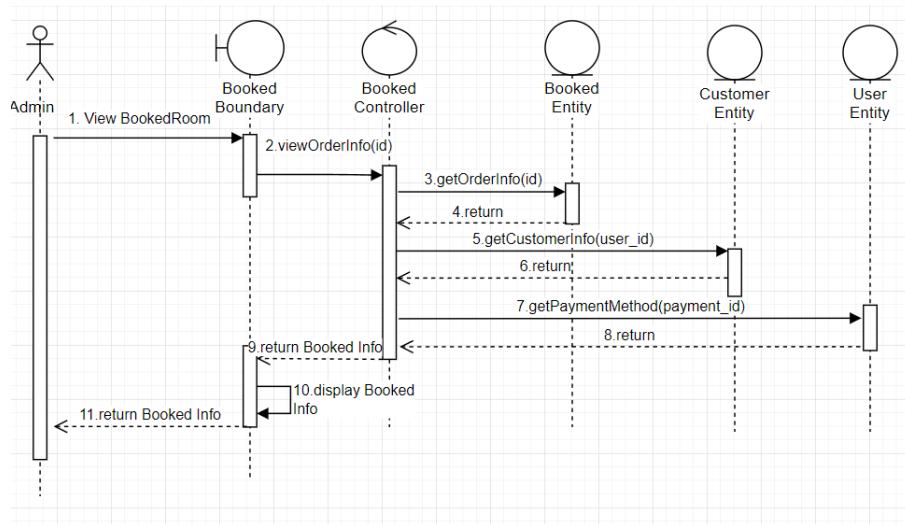


Hình 3.40 Sơ đồ Sequence khoá khách hàng

- Mô tả: cho phép **admin** khoá hoạt động khách hàng, Sequence có quy trình tương tự:

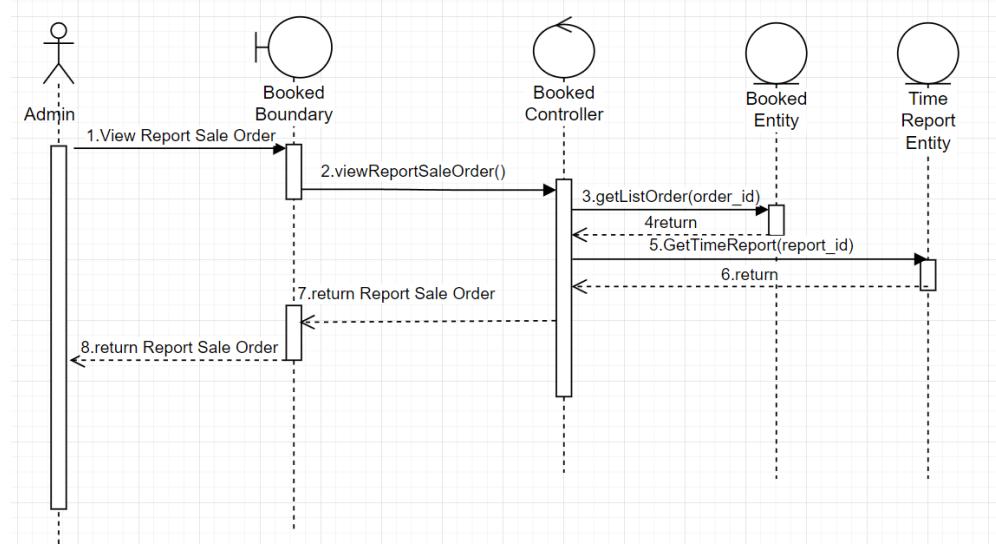
o Sequence Mở khóa khách hàng

- Sơ đồ Sequence xem danh sách phòng đã được đặt



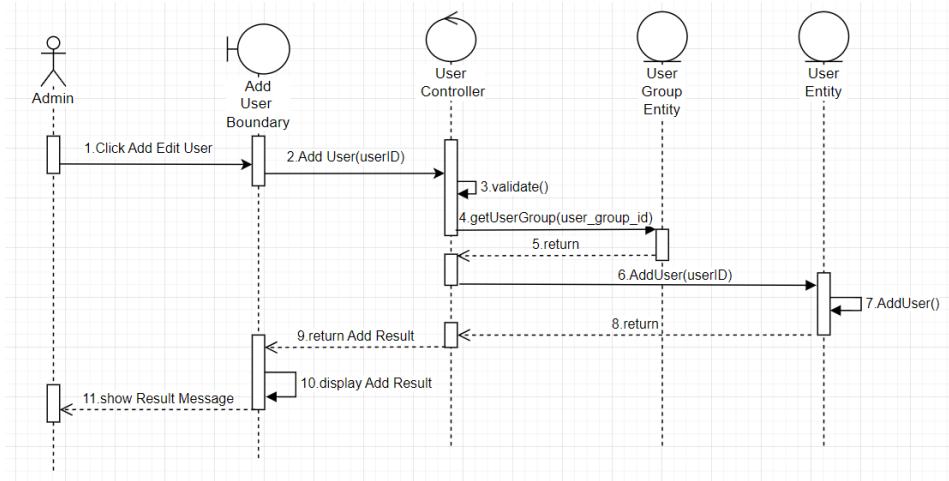
Hình 3.41 Sơ đồ Sequence xem danh sách phòng đã được đặt

- Mô tả: cho phép **admin** và **nhân viên** xem danh sách các phòng đã đặt trên hệ thống
- Sơ đồ Sequence xem báo cáo



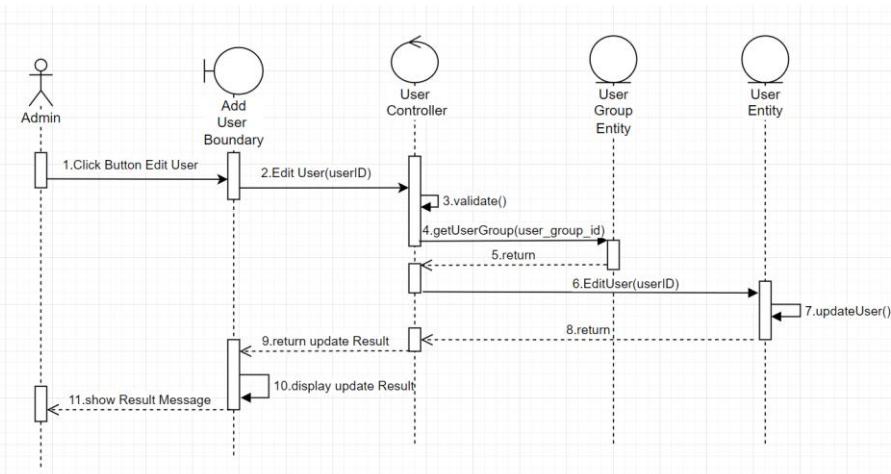
Hình 3.42 Sơ đồ Sequence xem báo cáo

- Mô tả: cho phép **admin** và **nhân viên** xem báo cáo thống kê
- Sơ đồ Sequence thêm nhân viên



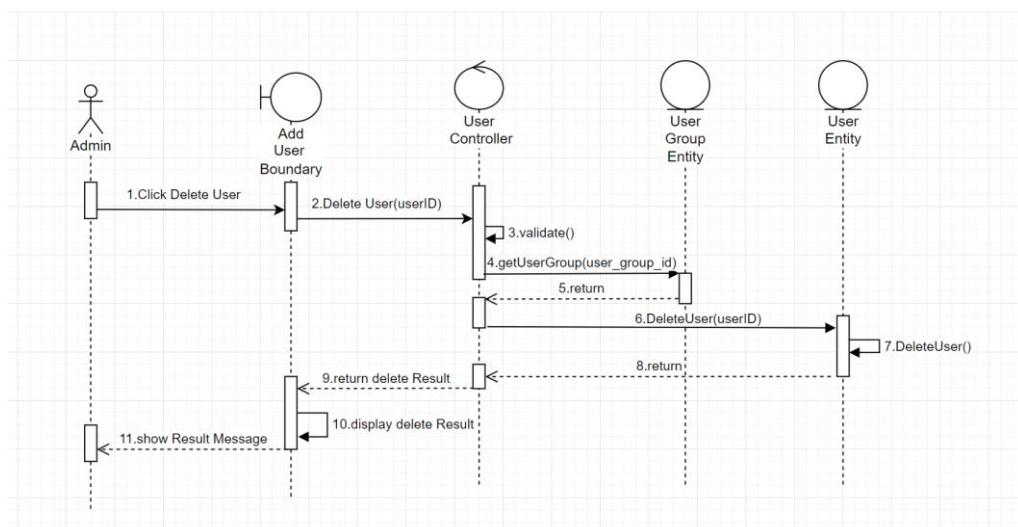
Hình 3.43 Sơ đồ Sequence thêm nhân viên

- Mô tả: cho phép **admin** thêm mới nhân viên, Sequence có quy trình tương tự:
 - **Sơ đồ Sequence thêm loại phòng ,**
 - **Sequence thêm dịch vụ phòng,**
 - **Sơ đồ Sequence thêm phòng**
- Sơ đồ Sequence sửa nhân viên



Hình 3.44 Sơ đồ Sequence sửa nhân viên

- Mô tả: cho phép **admin** thêm mới nhân viên, Sequence có quy trình tương tự:
 - **Sơ đồ Sequence sửa loại phòng ,**
 - **Sequence sửa dịch vụ phòng,**
 - **Sơ đồ Sequence sửa thông tin phòng**
- Sơ đồ Sequence xóa nhân viên

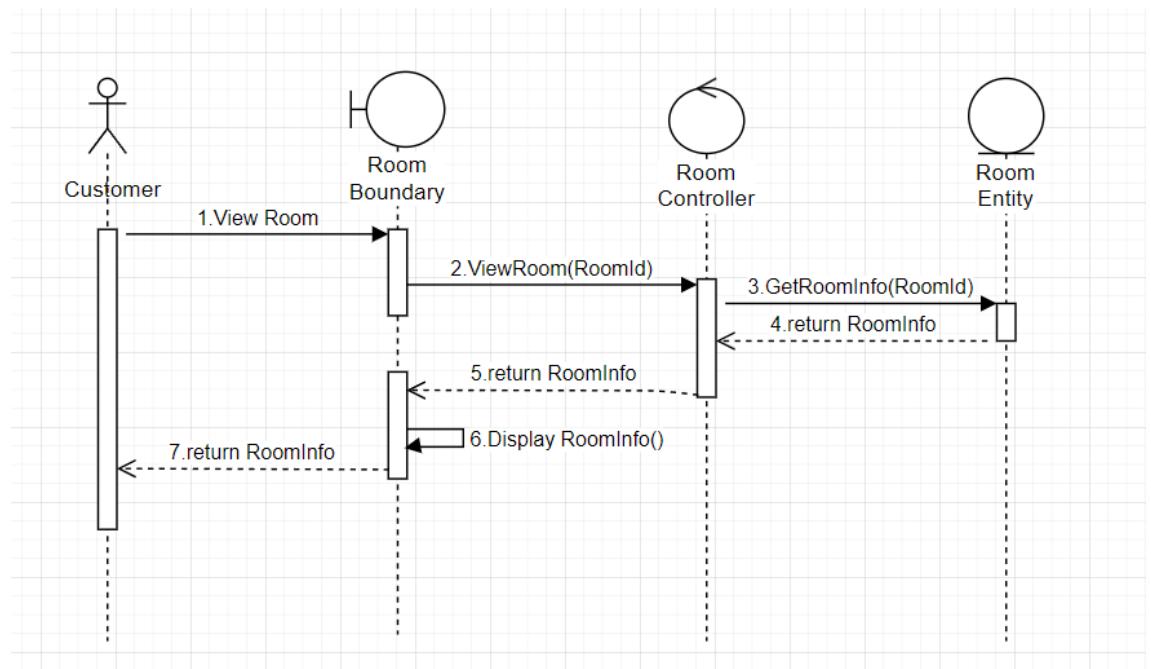


Hình 3.45 Sơ đồ Sequence xóa nhân viên

- Mô tả: cho phép **admin** thêm mới nhân viên, Sequence có quy trình tương tự:
 - **Sơ đồ Sequence xóa loại phòng**
 - **Sequence xóa dịch vụ phòng,**
 - **Sơ đồ Sequence xóa thông tin phòng**

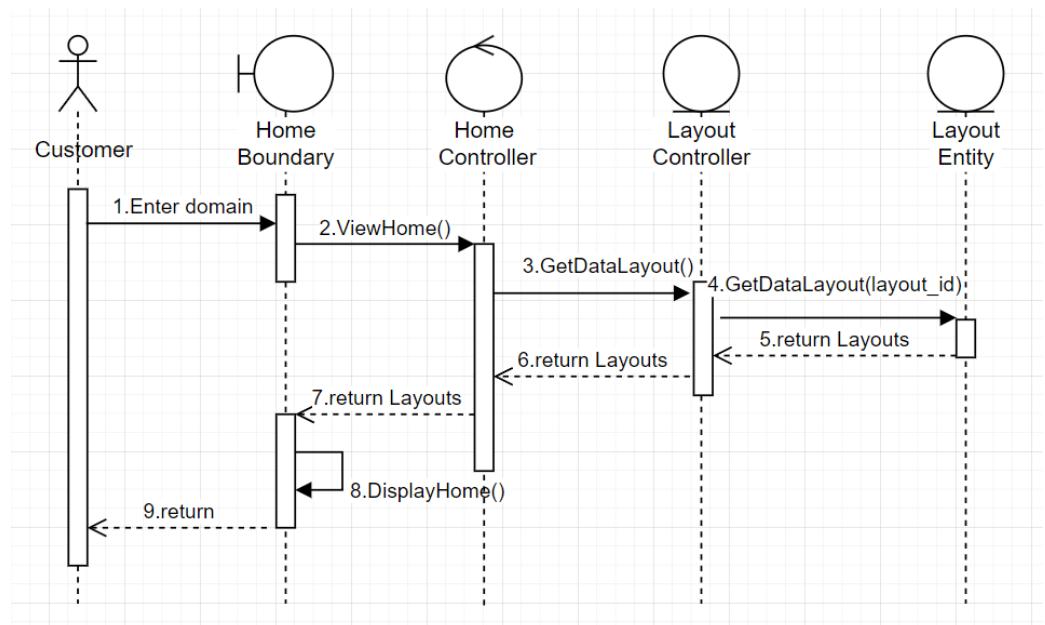
3.5.2 Khách hàng (Customer)

- Sơ đồ Sequence xem phòng



Hình 3.46 Sơ đồ Sequence xem phòng

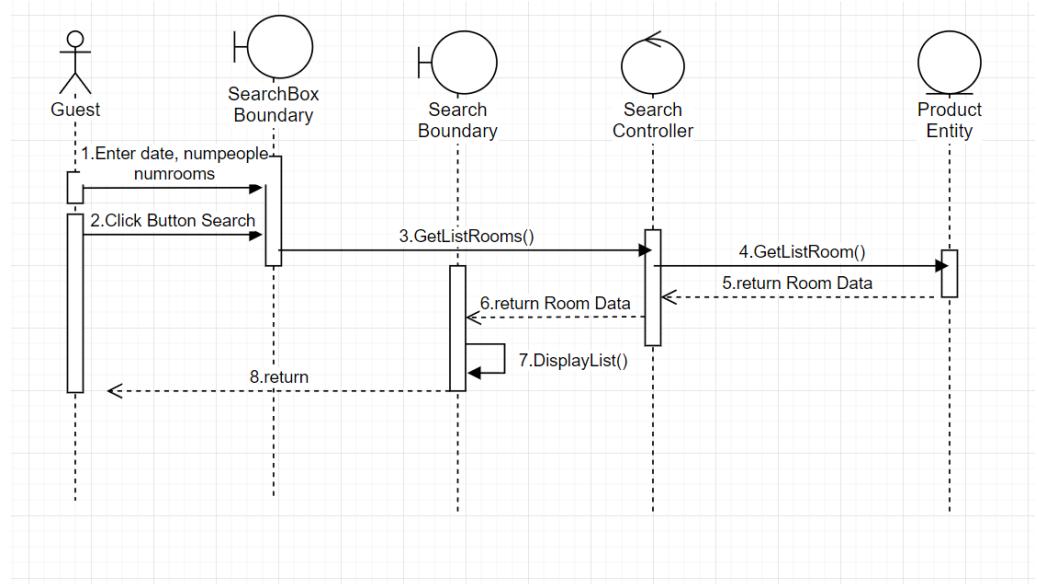
- Mô tả: cho phép user xem danh sách các tour đang hoạt động trên hệ thống, tương tự với đối tượng: **Khách vãng lai (Guest)**
- Sơ đồ Sequence xem trang chủ



Hình 3.47 Sơ đồ Sequence xem trang chủ

- Mô tả: cho phép **customer** và **khách vãng lai (Guest)** xem trang chủ hệ thống

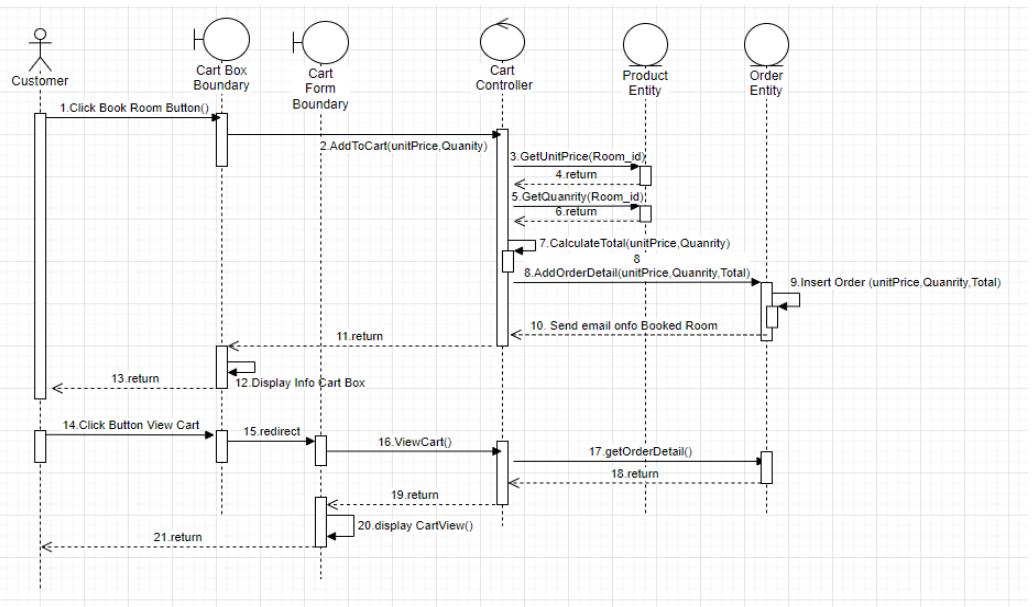
- Sơ đồ Sequence tìm kiếm phòng



Hình 3.48 Sơ đồ Sequence tìm kiếm phòng

- Mô tả: cho phép **customer** và **khách vãng lai (Guest)** tìm kiếm phòng

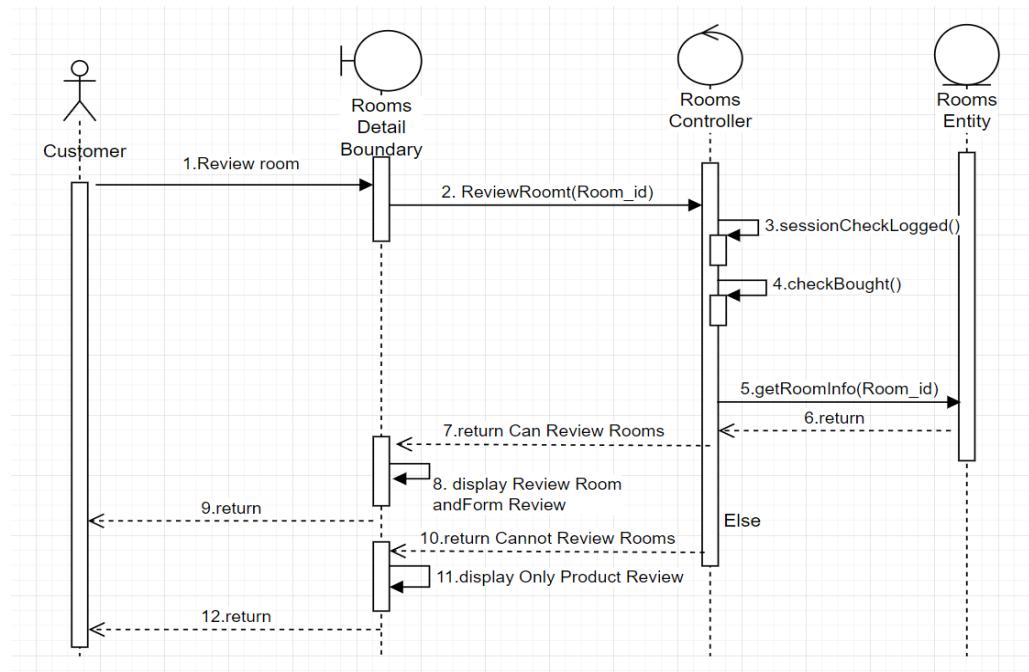
- Sơ đồ Sequence đặt phòng



Hình 3.49 Sơ đồ Sequence đặt phòng

- Mô tả: cho phép khách hàng đánh giá phòng

- Sơ đồ Sequence đánh giá phòng

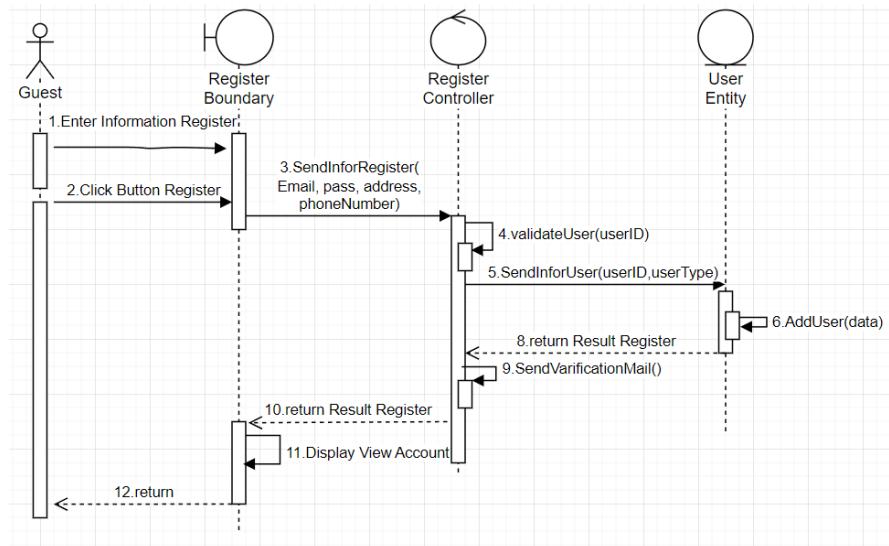


Hình 3.50 Sơ đồ Sequence đánh giá phòng

- Mô tả: cho phép **customer** (khách hàng) đánh giá phòng

3.5.3 Khách vãng lai (Guest)

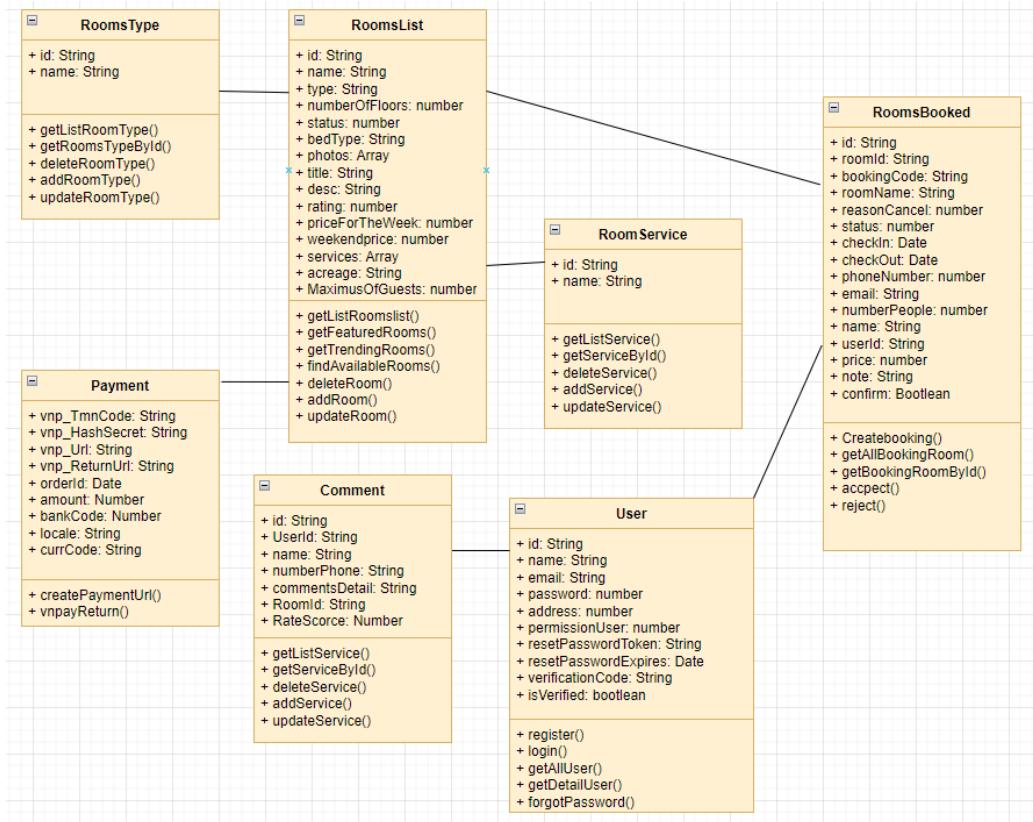
- Sơ đồ Sequence đăng ký khách hàng



Hình 3.51 Sơ đồ Sequence đăng ký khách hàng

- Mô tả: cho phép khách vãng lai (Guest) đăng ký người dùng trên hệ thống

3.6 Biểu đồ Class



Bảng 3.2 Biểu đồ Class

3.7 Mô tả cơ sở dữ liệu

- Bảng User (chứa thông tin người dùng)

| STT | TÊN | KIỂU DỮ LIỆU |
|-----|--------------------------|--------------|
| 1 | id | ObjectId |
| 2 | email | String |
| 3 | password | String |
| 4 | address | String |
| 5 | permissionUser | Number |
| 6 | resetPasswordToke n | String |
| 7 | resetPasswordExpi res | Date |
| 8 | verificationCode | String |
| 9 | isVerified | Boolean |

Bảng 3.3 Bảng thông tin User

- Bảng RoomsList (chứa thông tin phòng)

| STT | TÊN | KIỂU DỮ LIỆU |
|-----|-----------------|--------------|
| 1 | id | ObjectId |
| 2 | name | String |
| 3 | type | String |
| 4 | numberOfFloors | Number |
| 5 | status | Number |
| 6 | bedType | String |
| 7 | photos | Array |
| 8 | title | String |
| 9 | desc | String |
| 10 | priceForTheWeek | Number |
| 11 | weekendprice | Number |
| 12 | services | Array |
| 13 | acreage | String |
| 14 | MaximusOfGuests | Number |
| 15 | Comment | Array |
| 16 | Rating | Number |
| 17 | Payment-Status | String |

Bảng 3.4 Bảng thông tin chung

- Bảng RoomService (Chứa Thông tin dịch vụ phòng)

| STT | TÊN | KIỂU DỮ LIỆU |
|-----|------|--------------|
| 1 | id | ObjectId |
| 2 | name | String |

Bảng 3.5 Bảng thông tin dịch vụ chung

- Bảng RoomsType (Chứa thông tin loại phòng)

| STT | TÊN | KIỂU DỮ LIỆU |
|-----|------|--------------|
| 1 | id | ObjectId |
| 2 | name | String |

Bảng 3.6 Bảng thông tin loại phòng

- Bảng RoomsBooked (Chứa thông tin phòng đã book)

| STT | TÊN | KIỂU DỮ LIỆU |
|-----|--------------|--------------|
| 1 | id | ObjectId |
| 2 | roomId | String |
| 3 | bookingCode | String |
| 4 | roomName | String |
| 5 | reasonCancel | Number |
| 6 | status | Number |
| 7 | checkIn | Date |
| 8 | checkOut | Date |
| 9 | email | String |
| 10 | numberPeople | Number |
| 11 | name | String |
| 12 | userId | String |
| 13 | price | Number |
| 14 | note | String |
| 15 | confirm | Boolean |

Bảng 3.7 Bảng thông tin phòng đã book

- Bảng Comment (chứa thông tin đánh giá phòng)

| STT | TÊN | KIỂU DỮ LIỆU |
|-----|-------------|--------------|
| 1 | id | ObjectId |
| 2 | UserId | String |
| 3 | name | String |
| 4 | numberPhone | String |
| 5 | RoomId | String |
| 6 | RateScore | Number |

Bảng 3.8 Bảng thông tin đánh giá phòng

- Bảng Payment (Chứa chức năng thanh toán)

| STT | TÊN | KIỂU DỮ LIỆU |
|-----|----------------|--------------|
| 1 | vnp_TmnCode | String |
| 2 | vnp_HashSecret | String |
| 3 | vnp_Url | String |
| 4 | vnp_ReturnUrl | String |
| 5 | amount | Number |
| 6 | bankCode | Number |
| 7 | locale | String |
| 8 | currCode | String |

Bảng 3.9 Bảng chức năng thanh toán

CHƯƠNG 4. KẾT QUẢ CÀI ĐẶT, THỬ NGHIỆM

4.1 Yêu cầu cài đặt phần mềm

Để triển khai hệ thống website đặt phòng khách sạn cần tiến hành như sau

- Hệ điều hành: Window 10,11 hoặc MacOS
- Frontend: Nuxt.Js 3
- Backend: Node.Js
- Cơ sở dữ liệu: MongoDB
- Công cụ phát triển: Visual Studio Code, Git
- Quản lý dự án: GitLab

4.2 Cấu trúc dự án

Hệ thống được triển khai dưới dạng ba project chính: Project cho Admin, Project cho Enduser, và Project Backend.

4.2.1 Project Admin

- **Mục đích:** Quản lý thông tin khách sạn, phòng, giá cả, dịch vụ, và các đơn đặt phòng.
- **Chức năng:**
 - Đăng nhập/Đăng xuất
 - Quản lý phòng (thêm, sửa, xóa)
 - Quản lý đơn đặt phòng(xác nhận, ...)
 - Quản lý phản hồi từ khách hàng
 - Thông tin thống kê

4.2.2 Project Enduser

- **Mục đích:** Cho phép khách hàng và khách vãng lai tìm kiếm và đặt phòng.
- **Chức năng:**

- Đăng ký/Đăng nhập
- Tìm kiếm phòng
- Đặt phòng và thanh toán
- Quản lý thông tin cá nhân và các đơn đặt phòng
- Xem và để lại đánh giá

4.2.3 Project Backend

- **Mục đích:** Xử lý logic nghiệp vụ và tương tác với cơ sở dữ liệu.
- **Chức năng:**
 - Quản lý API cho Enduser và Admin
 - Xử lý các yêu cầu từ Frontend
 - Tương tác với MongoDB để lưu trữ và truy xuất dữ liệu

4.3 Cài đặt

4.3.1 Project Backend

1. “npm i” để cài đặt module
2. “npm start” để khởi động và connect DB

4.3.2 Project Admin và Enduser

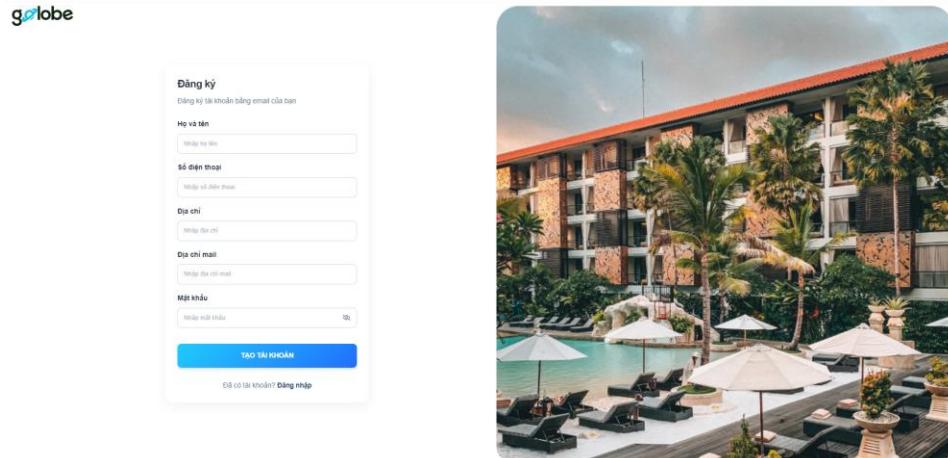
1. “npm i” để cài đặt module
2. “npm run dev” để khởi động giao diện

4.4 Thực nghiệm

4.4.1 Đăng ký và đăng nhập

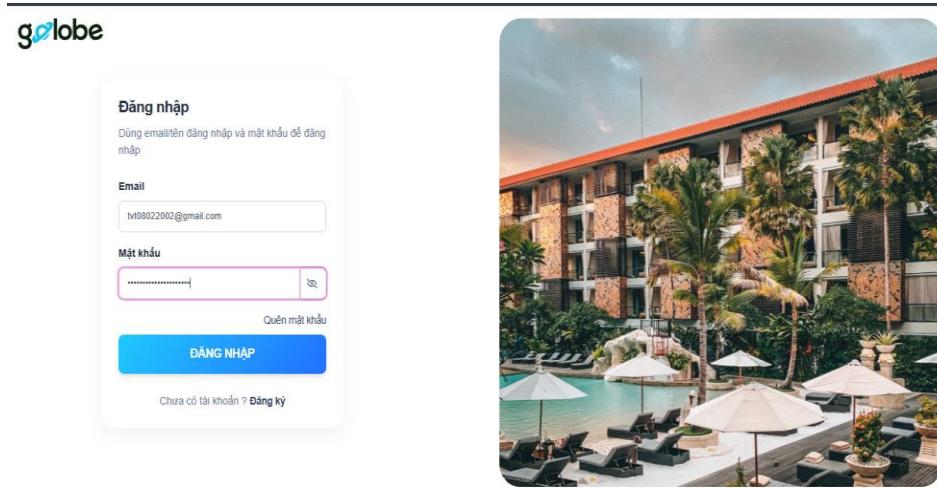
Đăng ký:

- Người dùng truy cập vào trang đăng ký và điền thông tin cá nhân (họ tên, email, mật khẩu, sđt, địa chỉ).
- Người dùng nhấn nút "Đăng ký" và hệ thống gửi yêu cầu POST tới API để lưu thông tin người dùng vào MongoDB
- Hệ thống trả về thông báo đăng ký thành công hoặc lỗi nếu có.



Hình 4.1 Màn hình đăng ký

Đăng nhập:



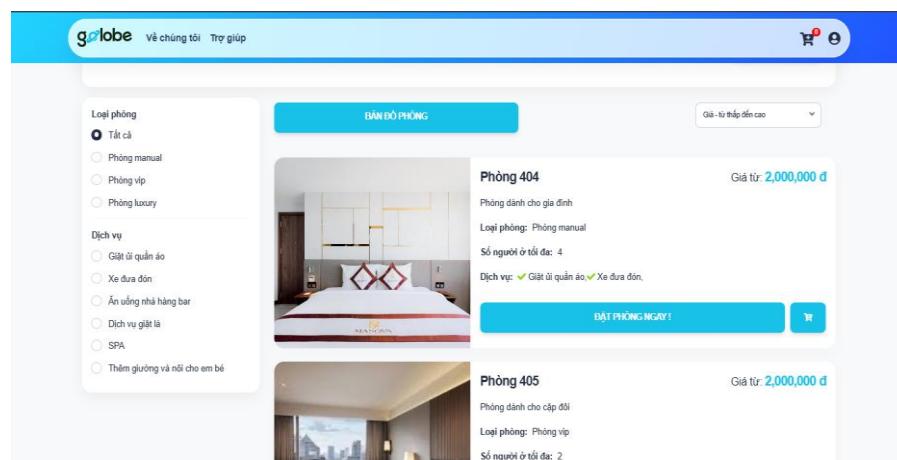
Hình 4.2 Màn hình đăng nhập

- Người dùng truy cập vào trang đăng nhập và nhập email cùng mật khẩu.
- Người dùng nhấn nút "Đăng nhập" và hệ thống gửi yêu cầu POST tới API để xác thực thông tin đăng nhập.
- Hệ thống trả về thông báo đăng nhập thành công hoặc lỗi nếu có, đồng thời lưu trữ token xác thực (nếu có) để quản lý phiên làm việc của người dùng.

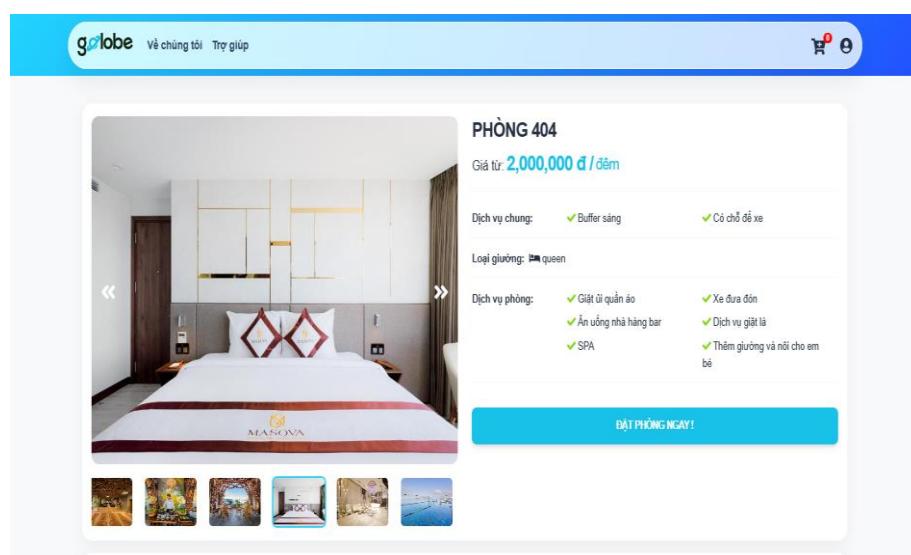
4.4.2 Tìm kiếm và đặt phòng

Tìm kiếm phòng:

- Người dùng truy cập trang tìm kiếm phòng và nhập các tiêu chí tìm kiếm (địa điểm, ngày nhận phòng, ngày trả phòng, số lượng khách).
- Hệ thống gửi yêu cầu GET tới API để lấy danh sách phòng phù hợp từ MongoDB.
- Hệ thống hiển thị danh sách phòng theo các tiêu chí tìm kiếm của người dùng.



Hình 4.3 Màn hình danh sách tìm kiếm



Hình 4.4 Màn hình chi tiết phòng

Đặt phòng:

- Người dùng chọn phòng từ danh sách và nhấn nút "Đặt phòng".
- Người dùng điền thông tin cá nhân và thông tin thanh toán.
- Người dùng nhấn nút "Xác nhận đặt phòng" và hệ thống gửi yêu cầu POST tới API để lưu thông tin đặt phòng vào MongoDB.
- Hệ thống trả về thông báo đặt phòng thành công hoặc lỗi nếu có.

The screenshot shows the GLOBE booking interface. At the top, there's a header with the GLOBE logo, search bar, and account information. Below the header, the main form is divided into several sections:

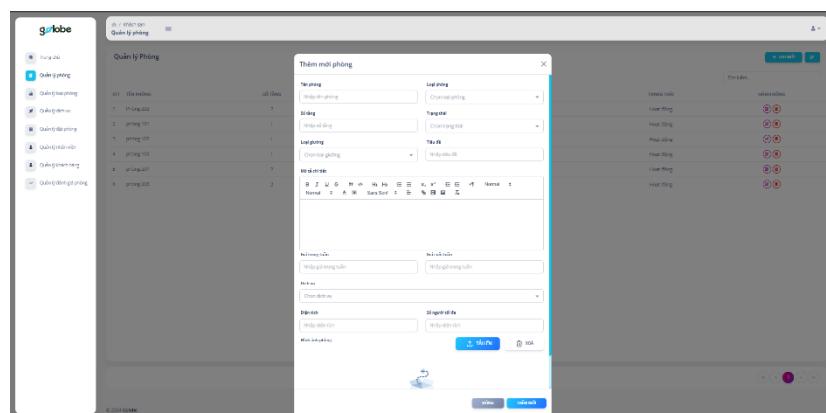
- Chi tiết phòng của bạn:** Shows check-in date (01/06/2025), check-out date (04/06/2025), and a note about a 3-day stay.
- Thông tin quý khách:** Fields for first name (Hà Văn Trình), last name (Trình), email (haovantrinh@gmail.com), phone number (0987654321), and address (Hà Nội).
- Yêu cầu đặc biệt:** A text area for special requests, with a note: "Vui lòng ghi yêu cầu đặc biệt của bạn tại đây (không bắt buộc)".
- Xác nhận đặt phòng:** A section for payment method selection:
 - Thương thức thanh toán:** Radio buttons for "Thanh toán online" (selected) and "Thanh toán tại khách sạn (dành cho khách hàng đã từng thanh toán online)".
 - Thanh toán với VNPAY giảm thêm 6%:** A note indicating a discount for VNPAY payment.
 - Công thanh toán VNPAY:** A placeholder for the payment gateway.
- ĐẶT PHÒNG NGAY:** A large blue button to confirm the booking.
- Đăng ký nhận tin:** A newsletter sign-up section with an input field for email and a "SUBMIT" button.

Hình 4.5 Màn hình đặt phòng

4.4.3 Quản lý phòng (Admin)

Thêm phòng:

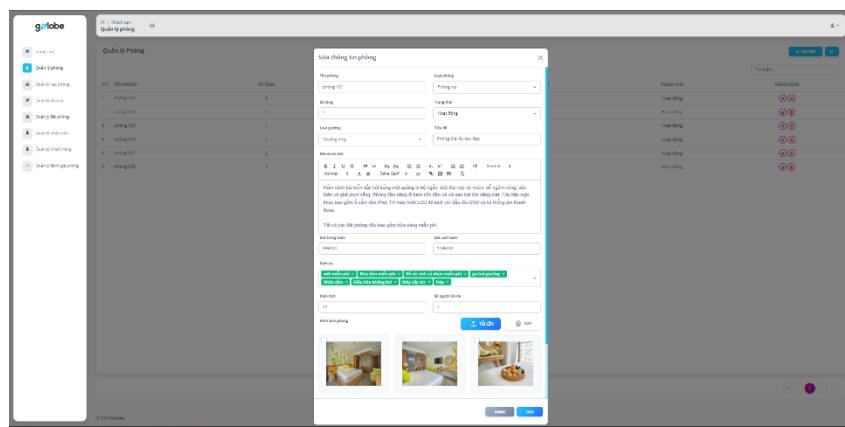
- Admin truy cập trang quản lý phòng và nhấn nút "Thêm phòng mới".
- Admin điền thông tin phòng (tên phòng, mô tả, giá cả, hình ảnh).
- Admin nhấn nút "Lưu" và hệ thống gửi yêu cầu POST tới API để lưu thông tin phòng mới vào MongoDB.
- Hệ thống trả về thông báo thêm phòng thành công hoặc lỗi nếu có.



Hình 4.6 Màn hình thêm phòng

Sửa phòng:

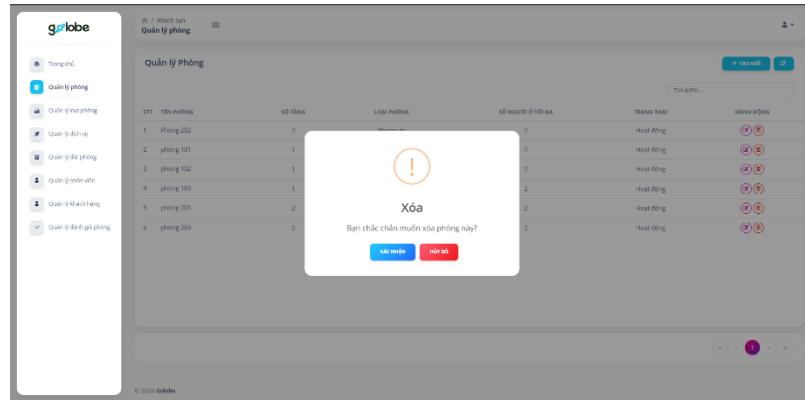
- Admin chọn phòng cần sửa từ danh sách và nhấn nút "Chỉnh sửa".
- Admin cập nhật thông tin phòng và nhấn nút "Lưu".
- Hệ thống gửi yêu cầu PUT tới API để cập nhật thông tin phòng trong MongoDB.
- Hệ thống trả về thông báo sửa phòng thành công hoặc lỗi nếu có



Hình 4.7 Màn hình sửa phòng

Xóa phòng:

- Admin chọn phòng cần xóa từ danh sách và nhấn nút "Xóa".
- Hệ thống gửi yêu cầu DELETE tới API để xóa phòng khỏi MongoDB.
- Hệ thống trả về thông báo xóa phòng thành công hoặc lỗi nếu có



Hình 4.8 Màn hình xóa phòng

4.4.4 Quản lý đơn đặt phòng (Admin)

Xem danh sách đơn đặt phòng:

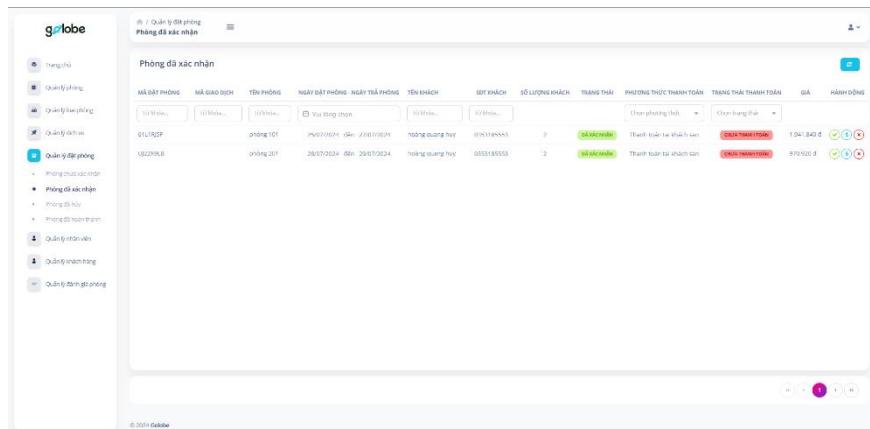
- Admin truy cập trang quản lý đơn đặt phòng để xem danh sách các đơn đặt phòng.
- Hệ thống gửi yêu cầu GET tới API để lấy danh sách đơn đặt phòng từ MongoDB.
- Hệ thống hiển thị danh sách đơn đặt phòng.

| Phòng chưa xác nhận | | | | | | | | | | |
|---------------------|-----------|-----------|----------------|----------------|-----------------|------------|----------------|--|--------------------------|---------------------|
| Số đặt phòng | Mã khách | Tên phòng | Ngày đặt phòng | Ngày trả phòng | Tên khách | Số khách | Số lượng khách | Trạng thái | Phòng thuê thành toán | Mức thuê phòng thuê |
| U221018 | phong 201 | phong 201 | 28/07/2024 | 29/07/2024 | hoàng dương may | 0932185552 | 2 | ĐÃ XÁC NHẬN | Thanh toán tại Khách sạn | 970.000 ₫ |
| F7670204 | 14934932 | phong 202 | 27/07/2024 | 28/07/2024 | hoàng quang may | 0932185558 | 2 | Chưa xác nhận | Thanh toán online | 1.177.474 ₫ |

Hình 4.9 Màn hình danh sách phòng chưa xác nhận

4.4.5 Cập nhật trạng thái đơn đặt phòng

- Admin chọn đơn đặt phòng cần cập nhật và nhấn nút "Xác nhận".
- Admin cập nhật trạng thái đơn đặt phòng (đã xác nhận, đã thanh toán, đã hủy).
- Hệ thống gửi yêu cầu PUT tới API để cập nhật thông tin đơn đặt phòng trong MongoDB.
- Hệ thống trả về thông báo cập nhật đơn đặt phòng thành công hoặc lỗi nếu có.



Hình 4.10 Màn hình danh sách phòng đã xác nhận

4.5 Kết quả và đánh giá

4.5.1 Kết quả

Hệ thống hoạt động ổn định: Các chức năng chính của hệ thống đã được triển khai và hoạt động mượt mà.

Hiệu suất tốt: Hệ thống có thể xử lý đồng thời nhiều yêu cầu mà không gặp vấn đề về hiệu năng.

Giao diện thân thiện: Người dùng có thể dễ dàng tương tác với hệ thống và thực hiện các thao tác cần thiết.

4.5.2 Đánh giá

Ưu điểm:

- Hệ thống có cấu trúc rõ ràng, dễ bảo trì và mở rộng.
- Sử dụng các công nghệ hiện đại và phổ biến như Nuxt.js, Node.js và MongoDB.
- Giao diện người dùng thân thiện và dễ sử dụng.

Nhược điểm:

- Một số tính năng phức tạp cần thời gian và nguồn lực

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Kết luận:

Qua việc thực hiện đề tài trên, em đã tạo ra một phần mềm đơn giản thân thiện với người sử dụng. Trong quá trình thực hiện đề tài em đã đạt được những mục tiêu sau:

- Nắm bắt được yêu cầu, phân tích, thiết kế.
- Phân quyền cho các đối tượng sử dụng.
- Sử dụng Framework NuxtJS, NodeJS.
- Thống kê.

Hạn chế:

- Chưa thử nghiệm trên Internet.
- Câu lệnh truy vấn CSDL còn chưa tối ưu.
- Chưa kiểm thử hết tất cả các trường hợp có thể xảy ra.

Hướng phát triển:

Với nỗ lực của bản thân, đồ án em đã cố gắng hoàn thành yêu cầu đề tài. Do thời gian và năng lực có hạn nên hệ thống của em mới chỉ đi sâu và chúc năng đặt phòng. Em muốn đề tài tốt nghiệp của em sẽ trở thành website đặt phòng online chuyên nghiệp. Cung cấp đầy đủ những dịch vụ hiện đang có trên thị trường với giá cả hợp lý, phải chăng. Đi kèm dịch vụ đặt phòng là những dịch vụ uy tín chất lượng để phục vụ đến du khách.

TÀI LIỆU THAM KHẢO

- [1] Thu.Mây, "Tổng quan ngành công nghệ thông tin Việt Nam," [Online]. Available: <https://caodangquoctehanoi.edu.vn/tong-quan-nganh-cong-nghe-thong-tin-viet-nam/>. [Accessed 11 10 2022].
- [2] "Thị trường Công nghệ thông tin - Viễn thông: Những xu hướng mới," 2021. [Online]. Available: <https://thoibaonganhang.vn/thi-truong-cong-nghe-thong-tin-vien-thong-nhung-xu-huong-moi-116867.html>. [Accessed 2022].
- [3] "Website là gì? Trang web là gì? Có những loại nào?". [Online]. Available: <https://carly.com.vn/blog/website-la-gi/> [Accessed 2019].
- [4] "NodeJS là gì? Những điều nên biết về NodeJS bạn cần biết năm 2022," 2021. [Online]. Available: <https://niithanoi.edu.vn/nodejs-la-gi-tong-hop-day-du-ve-nodejs-ban-can-biet.html>. [Accessed 2022].
- [5] "MongoDB là gì? Định nghĩa đầy đủ và chi tiết nhất về MongoDB,. [Online]. Available: <https://topdev.vn/blog/mongodb-la-gi/> [Accessed 2022].
- [6] "Vue.js là gì? Khái niệm cơ bản về Vue.js". [Online]. Available: <https://vi.vuejs.org/>. [Accessed 2022].
- [7] P. T. N. Mai, "Bootstrap là gì? Giới thiệu về Bootstrap," 2016. [Online]. Available: <https://viblo.asia/p/bai-1-bootstrap-la-gi-gioi-thieu-ve-bootstrap-DzVkpLbDknW>. [Accessed 2022].