# ĐẠI HỌC PHENIKAA

**TRƯỜNG CÔNG NGHỆ THÔNG TIN PHENIKAA**



BÁO CÁO ĐỒ ÁN CƠ SỞ

***TÊN ĐỀ TÀI: XÂY DỰNG WEBSITE QUẢN LÝ DỊCH VỤ***

***HOMESTAY***

Giảng viên hướng dẫn : ThS Nguyễn Quang Dũng,ThS Nguyễn Xuân Quế

Nhóm sinh viên thực hiện:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Nguyễn Mạnh Tiến | 22010151 | [22010151@st.phenikaa-uni.edu.vn](mailto:22010134@st.phenikaa-uni.edu.vn) |
| 2 | Lê Chí Hoàn | 22010063 | 22010063@st.phenikaa-uni.edu.vn |

*Khoá: K16 – 2022 – 2026*

*Lớp tín chỉ: Phân Tích và Thiết Kế Phần Mềm (N05)*

*Chương trình đào tạo: Công nghệ thông tin*

***Hà Nội, Tháng 10 năm 2025***

# LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành đến ThS Nguyễn Quang Dũng và ThS Nguyễn Xuân Quế giảng viên hướng dẫn, người đã tận tình giúp đỡ, định hướng và góp ý trong suốt quá trình thực hiện đồ án. Những kiến thức quý báu từ cô là nền tảng quan trọng giúp nhóm hoàn thiện sản phẩm một cách hiệu quả.

Chúng em cũng xin gửi lời cảm ơn đến Trường Công nghệ thông tin Phenikaa – Đại học Phenikaa đã tạo điều kiện và môi trường học tập tốt để chúng em có cơ hội tiếp cận thực tế và rèn luyện kỹ năng.

Cuối cùng, xin cảm ơn các thành viên trong nhóm đã nỗ lực làm việc tích cực, đoàn kết và phối hợp chặt chẽ để hoàn thành đồ án đúng tiến độ.

# LỜI CAM ĐOAN

Chúng em xin cam đoan:

* Báo cáo và sản phẩm đồ án là kết quả làm việc nghiêm túc của chính nhóm

chúng em.

* Mọi tài liệu tham khảo đều được trích dẫn rõ ràng.
* Nhóm không sao chép hay sử dụng trái phép bất kỳ sản phẩm nào từ các nguồn khác mà không được phép.

Nếu phát hiện có sự gian lận hay sao chép, chúng em xin hoàn toàn chịu trách nhiệm trước Hội đồng và nhà trường.

Hà Nội, ngày … tháng … năm

[1. Giới thiệu 4](#_Toc211974448)

[1.1. Đặt vấn đề 4](#_Toc211974449)

[1.2. Các giải pháp đã có 4](#_Toc211974450)

[1.3. Giải pháp đề xuất 5](#_Toc211974451)

[2. Thiết kế và triển khai 5](#_Toc211974454)

[2.1. Các yêu cầu chức năng 5](#_Toc211974455)

[2.1.1. Chức năng dành cho người dùng 5](#_Toc211974456)

[2.1.2. Chức năng dành cho quản trị viên (Admin) 6](#_Toc211974457)

[2.2. Các yêu cầu phi chức năng 7](#_Toc211974458)

[2.3. Công nghệ sử dụng 8](#_Toc211974459)

[2.3.1. Ngôn ngữ lập trình PHP 8](#_Toc211974461)

[2.3.2. Kỹ thuật AJAX 9](#_Toc211974469)

[2.3.3. Cơ sở dữ liệu MySQL 9](#_Toc211974476)

[2.4. Các ràng buộc khi triển khai 9](#_Toc211974489)

[2.4.1. Ràng buộc về kinh tế 10](#_Toc211974491)

[2.4.2. Ràng buộc về đạo đức 10](#_Toc211974496)

[3. Mô hình hệ thống / Thiết kế giải pháp 11](#_Toc211974503)

[3.1. Mô hình Use Case tổng quan 11](#_Toc211974504)

[3.2. Mô hình Use Case phân rã 11](#_Toc211974505)

[3.2.1. Mô hình Use Case người dùng 12](#_Toc211974506)

[3.2.2. Mô hình Use Case Admin 12](#_Toc211974507)

[3.3. Biểu đồ lớp (Class Diagram) 14](#_Toc211974508)

[3.4. Đặc tả Use Case 14](#_Toc211974509)

[3.4.1. Use Case: Đăng nhập / Đăng ký 14](#_Toc211974510)

[3.4.2. Use Case: Đặt phòng (Người dùng) 16](#_Toc211974516)

[3.4.3. Use Case: Quản lý đặt phòng (Admin) 17](#_Toc211974522)

[3.4.4. Use Case: Quản lý phòng (Admin) 18](#_Toc211974527)

[3.4.5. Use Case: Gửi tin nhắn (Người dùng) 19](#_Toc211974532)

[3.4.6. Use Case: Đánh giá phòng (Người dùng) 20](#_Toc211974537)

[3.4.7. Use Case: Quản lý người dùng (Admin) 20](#_Toc211974542)

[3.5. Biểu đồ tuần tự (Sequence Diagram) 21](#_Toc211974548)

[3.6. Biểu đồ trạng thái (State Diagram) 24](#_Toc211974549)

[3.7. Biểu đồ gói (Package Diagram) 26](#_Toc211974550)

[3.7.1 Sơ đồ database 28](#_Toc211974551)

[3.8. Áp dụng mẫu thiết kế – Singleton Pattern 28](#_Toc211974552)

[3.8.1. Giới thiệu 28](#_Toc211974553)

[3.8.2. Lý do lựa chọn Singleton Pattern 28](#_Toc211974554)

[3.8.3. Cấu trúc triển khai Singleton trong dự án 29](#_Toc211974555)

[3.9. Phác thảo giao diện 29](#_Toc211974556)

[TÀI LIỆU THAM KHẢO 31](#_Toc211974557)

# 

# PHÂN CÔNG NHIỆM VỤ ĐỒ ÁN

|  |  |
| --- | --- |
| **công việc/nhiệm**  **vụ** | **Mô tả tóm tắt công việc** |
| Công việc 1 | Thu thập dữ liệu, phân tích yêu cầu |
| Công việc 2 | Thiết kế hế thống: Use-case, CSDL, biểu đồ tuần tự… |
| Công việc 3 | Lập trình frontend |
| Công việc 4 | Lập trình backend |
| Công việc 5 | Viết báo cáo, chuẩn bị thuyết trình |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **T**  **T** | **MSV** | **Họ và**  **tên** | **Nhiệm vụ**  **phân công** | **Nội dung đã thực hiện** | **Đánh**  **giá** |
| 1 | 22010151 | Nguyễn Mạnh Tiến | Công việc 1,2,3,4,5 | Phân tích yêu cầu, lập trình  backend đặt phòng & người dùng, kiểm thử & báo cáo | 60% |
| 2 |  | Lê Chí Hoàn | Công việc 1,2,3,5 | Thiết kế giao diện frontend người dùng, lập trình phần quản lý phòng, thiết kế hệ  thống, viết báo cáo. | 40% |

# 1. Giới thiệu

## 1.1. Đặt vấn đề

Trên thế giới hiện nay, ngành Công nghệ Thông tin đang có những bước phát triển vô cùng mạnh mẽ, ứng dụng trong nhiều lĩnh vực đời sống. Thời kỳ công nghiệp đòi hỏi những thông tin nhanh chóng, chính xác. Máy tính điện tử trở nên gần gũi và phổ biến hơn đối với mọi người, việc đưa Công nghệ Thông tin vào công việc quản lý giúp giảm bớt sức lao động, tiết kiệm thời gian, nâng cao độ chính xác và hiệu quả hơn so với phương pháp quản lý thủ công trước đây.

Ngành CNTT ở Việt Nam cũng đang từng bước bứt phá, đạt được nhiều thành công trong việc áp dụng tin học vào phát triển kinh tế - xã hội, góp phần khẳng định vị thế của mình trên thị trường quốc tế.

Trong những năm gần đây, loại hình lưu trú ngắn hạn – đặc biệt là **Homestay** – đang phát triển mạnh mẽ. Nhu cầu khách hàng tăng cao kéo theo nhiều cá nhân, tổ chức đầu tư vào mô hình này nhờ các ưu điểm như tính tiện ích, giá cả hợp lý và trải nghiệm cá nhân hóa.

Tuy nhiên, việc quản lý hoạt động kinh doanh Homestay hiện vẫn còn nhiều bất cập. Phần lớn chủ cơ sở lưu trú vẫn sử dụng phương pháp thủ công (sổ sách, Excel hoặc điện thoại), gây khó khăn trong thống kê, kiểm soát đơn đặt phòng, và chăm sóc khách hàng.

**Do đó, việc xây dựng một hệ thống quản lý Homestay hiện đại, dễ sử dụng, hoạt động trên nền tảng web là cần thiết.** Ứng dụng không chỉ hỗ trợ quản lý phòng, khách hàng, đặt phòng, hóa đơn mà còn giúp **thống kê, phân tích hoạt động kinh doanh và nâng cao trải nghiệm quản lý cho chủ Homestay.**

## 1.2. Các giải pháp đã có

Hiện nay, một số giải pháp hỗ trợ quản lý Homestay phổ biến bao gồm:

* **Các nền tảng đặt phòng trực tuyến** (Booking.com, Airbnb, Agoda,...): có tích hợp công cụ quản lý chỗ ở, nhưng không hỗ trợ đầy đủ các chức năng quản lý nội bộ như thu chi, hóa đơn, nhân sự, thống kê.
* **Phần mềm quản lý khách sạn truyền thống (Hotel Management System - HMS)** như ezCloud, CiHMS,...: tuy nhiều chức năng nhưng cồng kềnh, phức tạp, chi phí cao, không phù hợp với mô hình Homestay nhỏ lẻ.
* **Quản lý bằng bảng tính Excel hoặc Google Sheet:** dễ triển khai và rẻ, nhưng thiếu tự động hóa, dễ sai sót và khó theo dõi tập trung.

## 1.3. Giải pháp đề xuất

Đề tài hướng tới việc xây dựng website quản lý Homestay đơn giản, dễ sử dụng, có thể triển khai trên nền tảng web, giải quyết những vấn đề trọng tâm trong hoạt động quản lý.

### a. Giao diện người dùng (Frontend)

* Đăng ký tài khoản: Người dùng tạo tài khoản bằng email, mật khẩu, thông tin cá nhân.
* Đăng nhập: Truy cập đầy đủ chức năng sau khi xác thực thông tin.
* Xem thông tin phòng: Hiển thị danh sách phòng, tiện nghi, hình ảnh, giá thuê, sức chứa.
* Tra cứu phòng trống: Lọc phòng theo ngày nhận – ngày trả.
* Đặt phòng trực tuyến: Tạo yêu cầu đặt phòng và chờ xác nhận.
* Xem dịch vụ: Cung cấp thông tin dịch vụ đi kèm (ăn uống, thuê xe, tour...).
* Gửi tin nhắn: Khách hàng có thể liên hệ nhanh với chủ Homestay.

### b. Giao diện Quản trị (Admin)

* Quản lý phòng: Thêm, sửa, xóa, cập nhật trạng thái (còn trống, đã đặt, bảo trì).
* Quản lý đặt phòng: Duyệt, từ chối, chỉnh sửa, hủy đơn đặt.
* Quản lý dịch vụ: Thêm/sửa dịch vụ kèm theo.
* Thống kê đặt phòng: Thống kê theo ngày, tháng, năm để theo dõi hiệu suất kinh doanh.
* Quản lý người dùng: Xem danh sách, phân quyền hoặc khóa tài khoản.
* Quản lý tiện ích và không gian: Cập nhật thông tin khu vực, tiện nghi (wifi, bếp, điều hòa,...).
* Quản lý đánh giá: Xem và phản hồi đánh giá khách hàng.
* Quản lý tin nhắn: Trả lời yêu cầu hỗ trợ, góp ý, khiếu nại từ người dùng.

# 2. Thiết kế và triển khai

## 2.1. Các yêu cầu chức năng

### 2.1.1. Chức năng dành cho người dùng

Hệ thống quản lý Homestay cho phép người dùng thực hiện các thao tác cơ bản sau:

* Đăng ký tài khoản:  
   Người dùng có thể tạo tài khoản mới bằng cách nhập thông tin cá nhân (email, mật khẩu, họ tên, số điện thoại).  
   Hệ thống kiểm tra dữ liệu hợp lệ và hiển thị thông báo khi đăng ký thành công hoặc thất bại.
* Đăng nhập / Đăng xuất:  
   Người dùng đăng nhập vào hệ thống bằng thông tin đã đăng ký.  
   Nếu thông tin sai, hệ thống hiển thị thông báo lỗi.  
   Khi đăng nhập thành công, người dùng có thể truy cập các chức năng chính và được phép đăng xuất an toàn khỏi hệ thống.
* Xem thông tin phòng:  
   Người dùng có thể xem danh sách các phòng Homestay đang hoạt động cùng thông tin chi tiết gồm: mô tả, hình ảnh, tiện nghi, giá thuê, sức chứa, tình trạng phòng.
* Tra cứu phòng trống:  
   Cung cấp công cụ lọc giúp người dùng tìm kiếm phòng theo ngày nhận – ngày trả.  
   Hệ thống chỉ hiển thị các phòng còn trống trong khoảng thời gian đã chọn.
* Đặt phòng trực tuyến:  
   Người dùng chọn phòng mong muốn và gửi yêu cầu đặt phòng.  
   Hệ thống ghi nhận thông tin, hiển thị trạng thái “Chờ xác nhận” và gửi thông báo đặt phòng thành công.
* Xem và sử dụng dịch vụ đi kèm:  
   Hiển thị thông tin chi tiết về các dịch vụ bổ sung như: cho thuê xe, giặt là, ăn uống, tour du lịch,...
* Gửi tin nhắn / liên hệ:  
   Người dùng có thể gửi yêu cầu, câu hỏi hoặc phản hồi cho chủ Homestay thông qua form liên hệ trên website.  
   Hệ thống lưu tin nhắn và hiển thị thông báo gửi thành công.
* Đánh giá phòng:  
   Sau khi hoàn tất lưu trú, người dùng có thể để lại đánh giá về chất lượng dịch vụ và chấm điểm sao cho phòng đã đặt.  
   Hệ thống ghi nhận và hiển thị đánh giá trên trang chi tiết phòng.

### 2.1.2. Chức năng dành cho quản trị viên (Admin)

Tài khoản quản trị viên có quyền truy cập và quản lý toàn bộ hệ thống với các chức năng chính sau:

* Quản lý phòng:  
   Thêm mới, chỉnh sửa hoặc xóa thông tin phòng (tên phòng, mô tả, hình ảnh, tiện nghi, giá thuê, trạng thái hoạt động).  
   Cập nhật trạng thái phòng theo thời gian thực (còn trống, đã đặt, đang bảo trì).
* Quản lý đơn đặt phòng:  
   Theo dõi danh sách đơn đặt phòng của người dùng.  
   Phê duyệt, từ chối hoặc hủy đơn đặt phòng khi cần.

Gửi thông báo cho người dùng về tình trạng xử lý đơn đặt.

* Quản lý dịch vụ đi kèm:  
   Thêm, sửa hoặc xóa các dịch vụ bổ sung (thuê xe, giặt là, ăn uống, tour du lịch,...).  
   Cập nhật giá và tình trạng hoạt động của từng dịch vụ.
* Thống kê đặt phòng và doanh thu:  
   Thống kê số lượng đơn đặt theo ngày, tháng, năm.  
   Tính toán tỷ lệ sử dụng phòng và doanh thu tổng hợp để hỗ trợ phân tích kinh doanh.
* Quản lý người dùng:  
   Hiển thị danh sách tài khoản khách hàng đã đăng ký.  
   Chỉnh sửa thông tin, khóa hoặc xóa tài khoản vi phạm.
* Quản lý không gian và tiện ích Homestay:  
   Cập nhật thông tin về các khu vực (phòng khách, bếp, sân vườn, khu vui chơi...) và tiện nghi (wifi, máy giặt, điều hòa, bếp, tủ lạnh,...).
* Quản lý đánh giá:  
   Xem và kiểm duyệt các đánh giá do người dùng gửi.  
   Có thể phản hồi, ẩn hoặc xóa những đánh giá không phù hợp.
* Quản lý tin nhắn / phản hồi từ người dùng:  
   Nhận và trả lời tin nhắn từ khách hàng gửi qua hệ thống.  
   Phân loại tin nhắn theo nhóm yêu cầu: hỗ trợ, góp ý, khiếu nại,...

## 2.2. Các yêu cầu phi chức năng

Ngoài các chức năng chính, hệ thống cần đáp ứng các yêu cầu phi chức năng sau để đảm bảo tính ổn định, bảo mật và khả năng mở rộng trong vận hành:

* Hiệu năng:

Hệ thống phải đảm bảo tốc độ phản hồi nhanh, thời gian tải trang ngắn, có thể xử lý đồng thời nhiều người dùng truy cập mà không gây gián đoạn dịch vụ.

* Bảo mật:  
   Dữ liệu người dùng (thông tin cá nhân, tài khoản, đơn đặt phòng) phải được lưu trữ an toàn.  
   Mật khẩu và thông tin nhạy cảm cần được mã hóa.  
   Cơ chế xác thực, phân quyền rõ ràng giữa quản trị viên và người dùng thông thường.
* Khả năng mở rộng:  
   Hệ thống được thiết kế theo hướng module hóa, cho phép dễ dàng bổ sung, chỉnh sửa hoặc tích hợp thêm các chức năng mới trong tương lai.
* Tính ổn định và độ tin cậy:  
   Hệ thống phải hoạt động ổn định, hạn chế tối đa lỗi phát sinh.  
   Khi xảy ra lỗi, cần có thông báo rõ ràng và cơ chế xử lý phù hợp.
* Khả năng bảo trì:  
   Mã nguồn được tổ chức khoa học, dễ đọc và dễ sửa đổi.  
   Cấu trúc thư mục rõ ràng giúp việc nâng cấp hoặc bảo trì hệ thống thuận tiện.
* Tương thích đa nền tảng:  
   Giao diện website phải hiển thị tốt và hoạt động ổn định trên nhiều thiết bị khác nhau: máy tính, máy tính bảng và điện thoại di động.
* Sao lưu và phục hồi dữ liệu:  
   Hệ thống cần có cơ chế sao lưu định kỳ để tránh mất mát dữ liệu.  
   Có khả năng phục hồi nhanh chóng trong trường hợp sự cố hoặc tấn công hệ thống.

## 2.3. Công nghệ sử dụng

### Trong dự án Hệ thống quản lý Homestay, nhóm lựa chọn các công nghệ phổ biến, dễ triển khai và phù hợp với yêu cầu của một ứng dụng web quản lý dữ liệu.

### 2.3.1. Ngôn ngữ lập trình PHP

### PHP (*PHP: Hypertext Preprocessor*) là ngôn ngữ lập trình mã nguồn mở chạy phía máy chủ (server-side), được sử dụng rộng rãi trong phát triển web. PHP cho phép kết hợp linh hoạt với HTML, CSS và JavaScript, giúp xây dựng các trang web động, thân thiện và dễ bảo trì.

### Ưu điểm khi lựa chọn PHP cho dự án:

### Miễn phí, phổ biến, dễ triển khai trên các máy chủ web thông dụng (Apache,Nginx).

### Hỗ trợ tốt việc thao tác với cơ sở dữ liệu MySQL.

### Cộng đồng lớn, tài liệu phong phú, dễ học và dễ mở rộng.

### Tích hợp tốt với các công nghệ web khác như AJAX, Bootstrap.

### PHP được dùng để xây dựng toàn bộ chức năng nghiệp vụ chính như: quản lý phòng, đặt phòng, quản lý dịch vụ, quản lý người dùng và hiển thị giao diện web.

### 2.3.2. Kỹ thuật AJAX

### AJAX (*Asynchronous JavaScript and XML*) là kỹ thuật cho phép trao đổi dữ liệu với máy chủ mà không cần tải lại toàn bộ trang web. Trong dự án, AJAX được sử dụng để thực hiện các thao tác như đặt phòng, tra cứu phòng trống, gửi tin nhắn, giúp trải nghiệm người dùng mượt mà hơn.

### Lợi ích khi sử dụng AJAX:

### Tăng tốc độ tải trang, chỉ cập nhật phần nội dung thay đổi.

### Tiết kiệm băng thông và giảm tải cho máy chủ.

### Cải thiện trải nghiệm người dùng với giao diện tương tác nhanh, liên tục.

### Dễ dàng tích hợp với PHP và MySQL trong mô hình ứng dụng web hiện đại.

### 2.3.3. Cơ sở dữ liệu MySQL

### MySQL là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở được lựa chọn cho hệ thống. Dữ liệu như thông tin người dùng, phòng, đặt phòng, đánh giá và dịch vụ được lưu trữ, truy xuất và xử lý thông qua MySQL.

### Ưu điểm của MySQL:

### Tốc độ xử lý nhanh, ổn định và độ tin cậy cao.

### Hỗ trợ tốt các truy vấn phức tạp và giao dịch (Transaction).

### Dễ tích hợp với PHP thông qua thư viện mysqli hoặc PDO.

### Dễ cài đặt và triển khai trên nhiều nền tảng khác nhau.

### Cấu trúc cơ sở dữ liệu của hệ thống được thiết kế theo mô hình quan hệ (Relational Database Schema) với các bảng chính:

### users – quản lý thông tin người dùng;

### rooms – lưu thông tin phòng;

### bookings – quản lý đơn đặt phòng;

### services – lưu danh sách dịch vụ;

### reviews – lưu trữ đánh giá từ khách hàng.

## 2.4. Các ràng buộc khi triển khai

### Khi triển khai hệ thống, nhóm cần xem xét một số ràng buộc đặc thù về kinh tế, kỹ thuật và đạo đức nhằm đảm bảo hệ thống có thể vận hành ổn định và bền vững.

### 2.4.1. Ràng buộc về kinh tế

### Hạn chế về ngân sách đầu tư: Do đây là dự án học tập, nguồn lực tài chính có giới hạn, nhóm ưu tiên sử dụng các công cụ mã nguồn mở miễn phí như: PHP, MySQL, Bootstrap, Visual Studio Code,... nhằm giảm thiểu chi phí.

### Hạn chế về hạ tầng triển khai: Hệ thống không sử dụng máy chủ vật lý riêng mà triển khai thử nghiệm trên các nền tảng lưu trữ đám mây miễn phí (Render, Heroku, GitHub Codespaces). Vì vậy, hiệu năng và dung lượng lưu trữ có giới hạn, chỉ phù hợp cho mục đích demo hoặc thử nghiệm.

### Chi phí bảo trì và nâng cấp: Việc bảo trì lâu dài sau khi kết thúc môn học phụ thuộc vào khả năng tự chỉnh sửa của sinh viên. Do đó, mã nguồn được viết đơn giản, dễ đọc và dễ bảo trì để người khác có thể kế thừa và phát triển tiếp.

### Không tích hợp cổng thanh toán thực tế: Hệ thống chỉ mô phỏng quá trình đặt phòng và thanh toán giả lập nhằm đảm bảo an toàn dữ liệu và tránh rủi ro tài chính trong môi trường thử nghiệm.

### 2.4.2. Ràng buộc về đạo đức

### Bảo vệ dữ liệu cá nhân: Thông tin người dùng (họ tên, email, số điện thoại) được bảo mật và chỉ thu thập ở mức tối thiểu cần thiết. Dữ liệu được mã hóa và chỉ người quản trị có quyền truy cập.

### Không phân biệt đối xử và đảm bảo công bằng: Giao diện và nội dung hệ thống không chứa yếu tố kỳ thị, phân biệt giới tính, vùng miền hoặc dân tộc. Mọi người dùng có quyền tiếp cận và sử dụng hệ thống như nhau.

### Không lan truyền thông tin sai lệch: Các thông tin phòng, dịch vụ, hình ảnh được yêu cầu chính xác, không vi phạm bản quyền hoặc gây hiểu nhầm cho khách hàng.

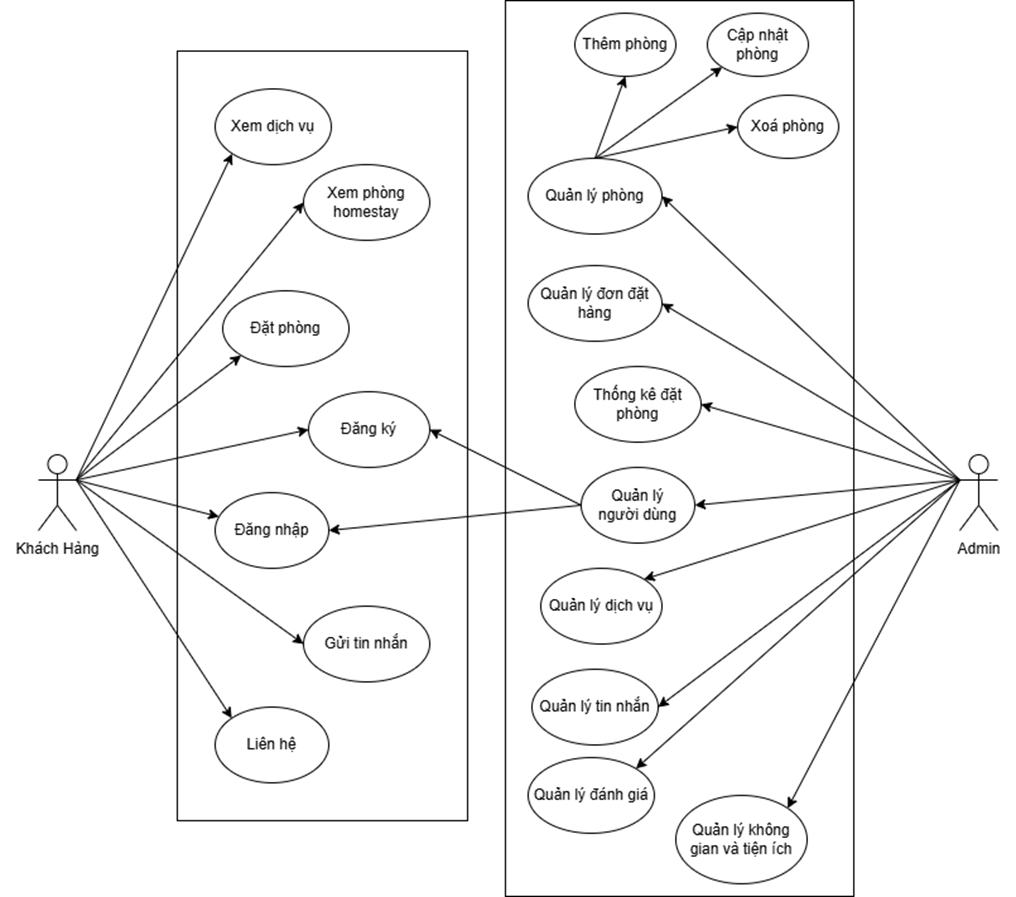
### Trách nhiệm với cộng đồng người dùng:

### Hệ thống có cơ chế kiểm soát nội dung do người dùng đăng tải (đánh giá, bình luận) và cho phép ẩn hoặc báo cáo các nội dung vi phạm tiêu chuẩn cộng đồng.

### Hướng đến mục tiêu giáo dục và minh bạch: Dự án được xây dựng phục vụ mục đích học tập, vì vậy toàn bộ thiết kế, mã nguồn và tài liệu sẽ được công bố minh bạch, có chú thích rõ ràng để giúp sinh viên khác có thể học tập và phát triển thêm.

# 3. Mô hình hệ thống / Thiết kế giải pháp

## 3.1. Mô hình Use Case tổng quan



*Hình 1. Biểu đồ Use Case tổng quát*

## 

## 

## 

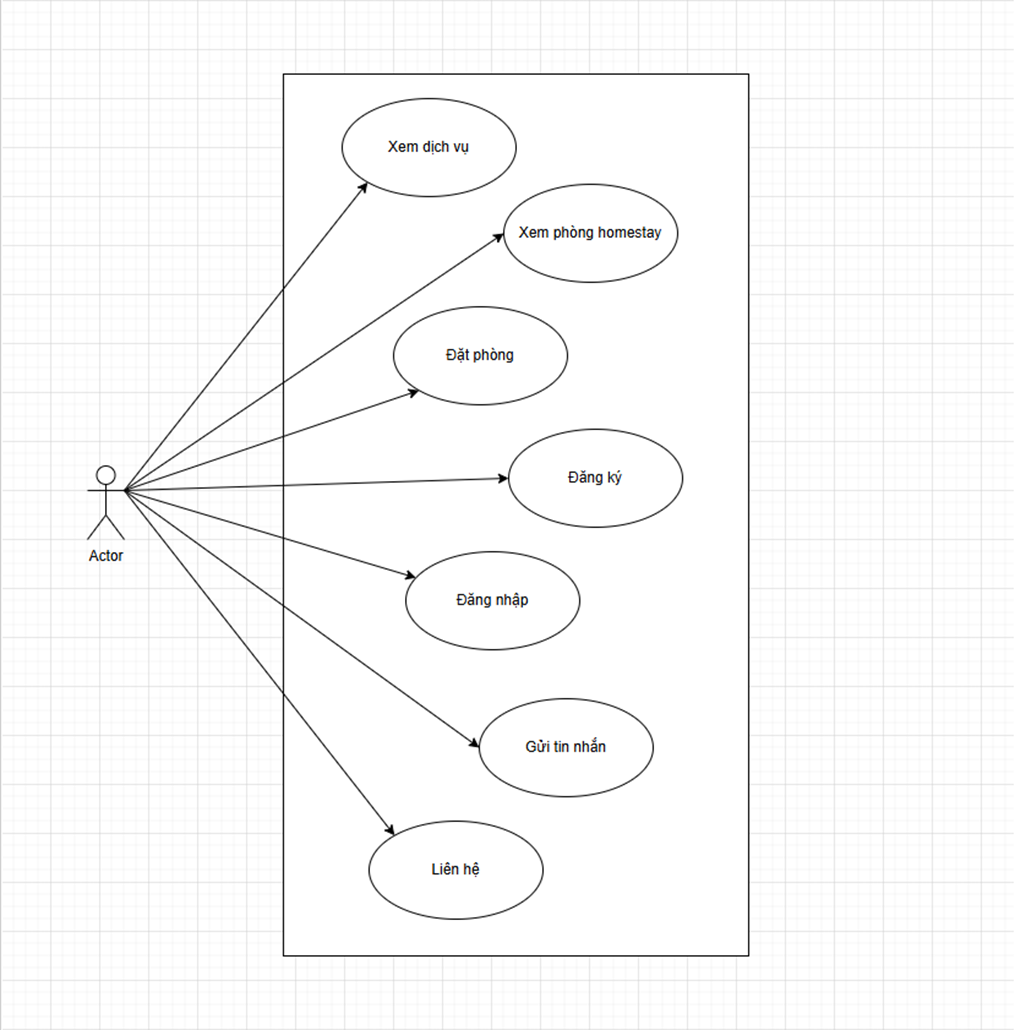
## 

## 

## 

## 3.2. Mô hình Use Case phân rã

### 3.2.1. Mô hình Use Case người dùng



*Hình 2. Biểu đồ Use Case người dùng*

### 

### 

### 

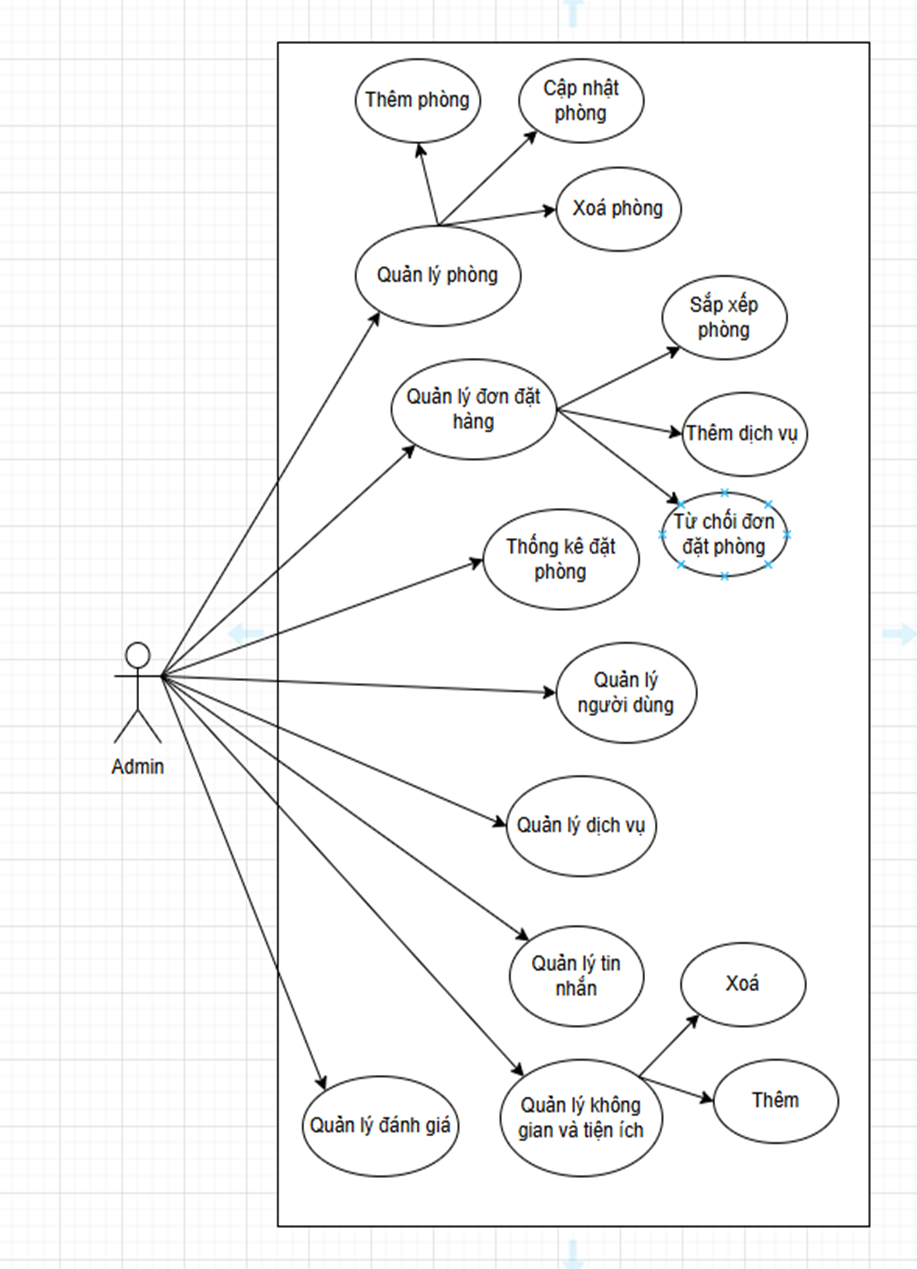
### 

### 

### 

### 

### 3.2.2. Mô hình Use Case Admin



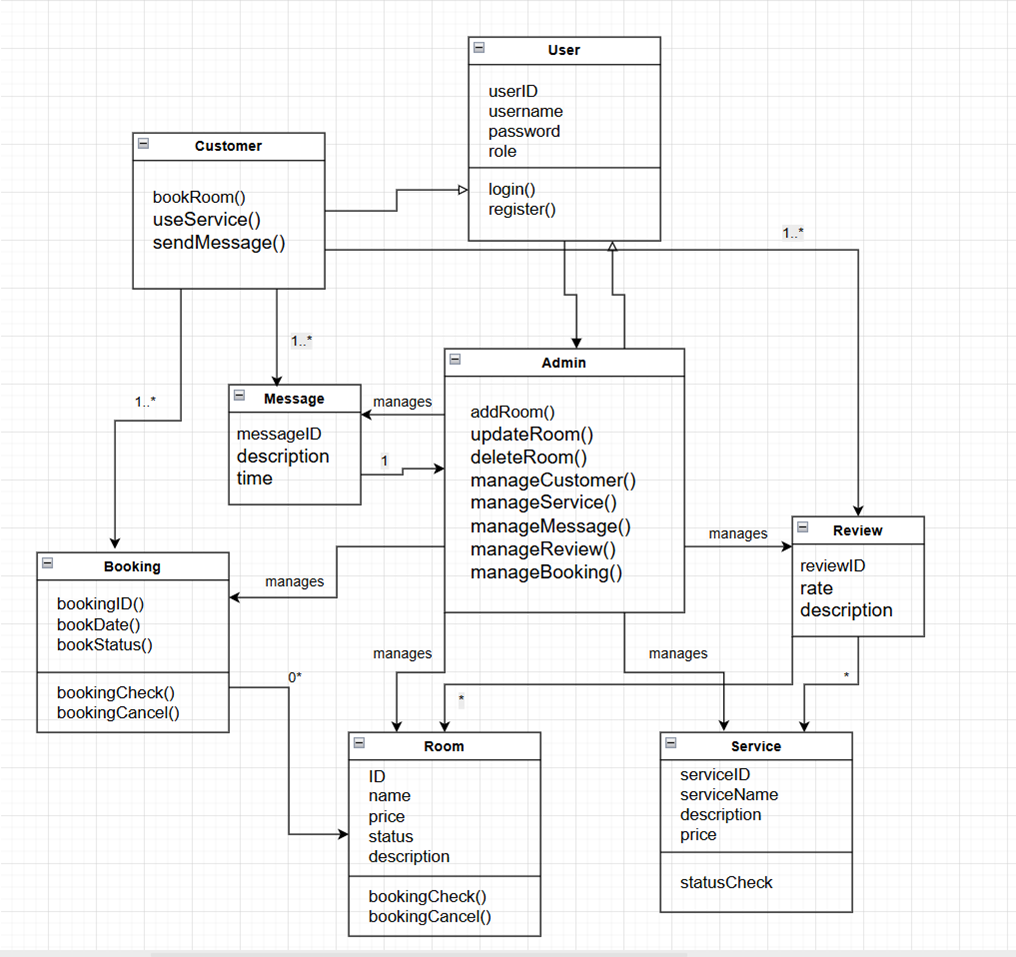
*Hình 3. Biểu đồ Use Case Admin*

## 

## 

## 

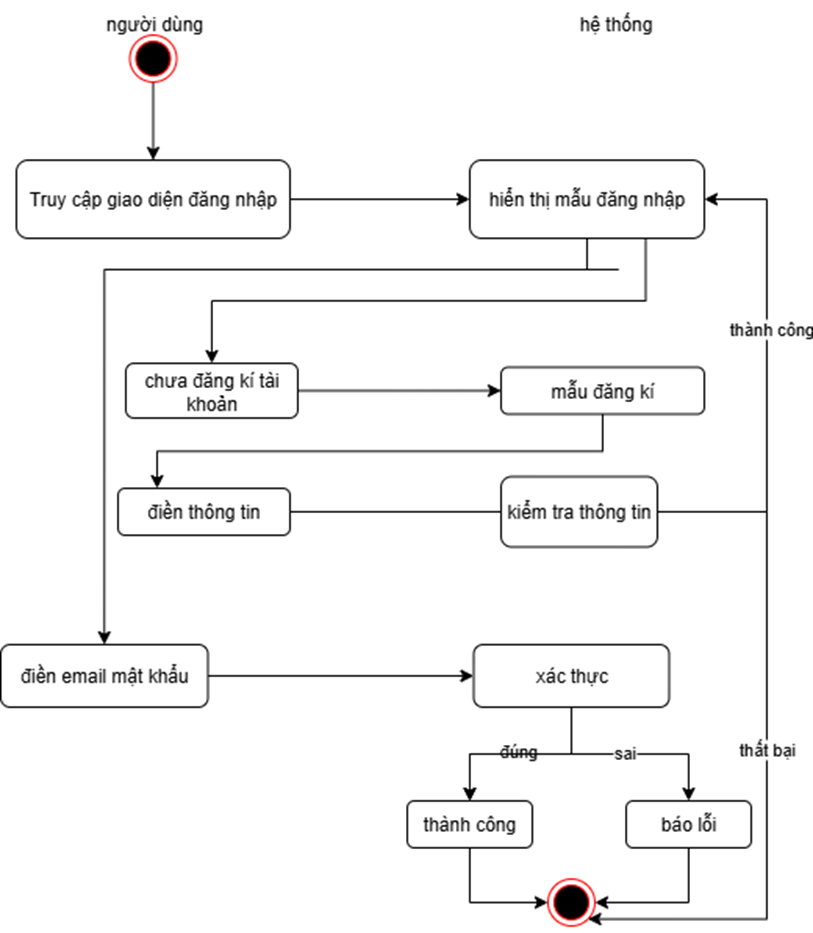
## 3.3. Biểu đồ lớp (Class Diagram)



*Hình 4. Biểu đồ thể hiện quan hệ giữa các lớp chính trong hệ thống*

## 3.4. Đặc tả Use Case

## 3.4.1. Use Case: Đăng nhập / Đăng ký



### Mô tả:

Người dùng hoặc quản trị viên thực hiện đăng nhập vào hệ thống để truy cập các chức năng phù hợp với quyền của mình.  
 Nếu chưa có tài khoản, người dùng có thể đăng ký tài khoản mới để sử dụng hệ thống.

### Tác nhân: Người dùng, Quản trị viên (Admin)

### Điều kiện kích hoạt:

Người dùng truy cập vào trang “Đăng nhập” hoặc “Đăng ký”.

### Tiền điều kiện:

Thiết bị được kết nối Internet.

### Hậu điều kiện:

Nếu thông tin hợp lệ: hệ thống chuyển đến trang chính tương ứng.  
Nếu sai thông tin: hiển thị thông báo lỗi.

**Các luồng sự kiện:**

|  |  |
| --- | --- |
| Bước | Mô tả |
| 1 | Người dùng mở ứng dụng và chọn đăng nhập |
| 2 | Hệ thống hiển thị giao diện đăng nhập |
| 3 | Người dùng nhập thông tin email, mật khẩu, ấn nút đăng nhập |
| 4 | Hệ thống kiểm tra thông tin và xác nhận thông tin |
| 5 | Nếu đúng thông tin chuyển sang giao diện của ứng dụng |
| 6 | Nếu sai hiển thị thông báo lỗi |

Bảng 1: Luồng sự kiện đăng nhập

Đăng ký

Mô tả: Người dùng nếu chưa tạo tài khoản trước đó sẽ thực hiện thao

tác đăng ký tài khoản mới.

Các tác nhân: Người dùng chưa đăng ký

Điều kiện kích hoạt sử dụng: Người dùng chưa có tài khoản Tiền điều kiện: Thiết bị được kết nối internet

Hậu điều kiện: Không có

## 3.4.2. Use Case: Đặt phòng (Người dùng)

### 

### Mô tả:

### Người dùng sau khi đăng nhập có thể tra cứu phòng trống và đặt phòng trực tuyến trong khoảng thời gian mong muốn.

### Tác nhân: Người dùng đã đăng nhập

### Tiền điều kiện:

Người dùng đã có tài khoản và đăng nhập thành công.

Phòng còn trống trong thời gian được chọn.

### Hậu điều kiện:

Hệ thống ghi nhận thông tin đặt phòng với trạng thái ban đầu là “Chờ xác nhận”.

**Các luồng sự kiện:**

|  |  |
| --- | --- |
| Bước | Mô tả |
| 1 | Người dùng tìm kiếm phòng trống |
| 2 | Hệ thống hiển thị danh sách phòng theo tiêu chí tìm kiếm |
| 3 | Người dùng chọn phòng, chọn ngày nhận và ngày trả |
| 4 | Người dùng nhấn “Đặt phòng” |
| 5 | Hệ thống lưu thông tin đặt phòng, chuyển trạng thái thành “Chờ xác nhận” |
| 6 | Hiển thị thông báo đặt phòng thành công |

## 3.4.3. Use Case: Quản lý đặt phòng (Admin)

### Mô tả:

Admin có thể xem, duyệt, hủy hoặc xác nhận các đơn đặt phòng của người dùng.

### Tác nhân: Admin

### Tiền điều kiện:

Admin đã đăng nhập hệ thống.

Có ít nhất một đơn đặt phòng ở trạng thái “Chờ xác nhận”.

### Hậu điều kiện:

Trạng thái đơn đặt được cập nhật (“Đã xác nhận” hoặc “Đã hủy”).

Người dùng nhận được thông báo thay đổi trạng thái.

**Các luồng sự kiện**

|  |  |
| --- | --- |
| Bước | Mô tả |
| 1 | Admin đăng nhập hệ thống |
| 2 | Truy cập trang “Quản lý đặt phòng” |
| 3 | Chọn đơn đặt cần xử lý |
| 4 | Chọn hành động: “Xác nhận” hoặc “Hủy” |
| 5 | Hệ thống cập nhật trạng thái đơn đặt |
| 6 | Gửi thông báo đến người dùng |

## 3.4.4. Use Case: Quản lý phòng (Admin)

### Mô tả:

Admin thực hiện các thao tác thêm mới, chỉnh sửa hoặc xóa thông tin phòng Homestay.

### Tác nhân: Admin

### Tiền điều kiện:

Admin đã đăng nhập hệ thống.

### Hậu điều kiện:

Dữ liệu phòng được cập nhật trong cơ sở dữ liệu.

**Các luồng sự kiện:**

|  |  |
| --- | --- |
| Bước | Mô tả |
| 1 | Admin chọn "Thêm phòng mới" hoặc chỉnh sửa/xóa phòng hiện có |
| 2 | Điền/hiệu chỉnh thông tin: tên phòng, giá, hình ảnh, mô tả… |
| 3 | Nhấn "Lưu" |
| 4 | Hệ thống cập nhật dữ liệu và hiển thị danh sách mới |

## 3.4.5. Use Case: Gửi tin nhắn (Người dùng)

### Mô tả:

Người dùng có thể gửi yêu cầu hỗ trợ hoặc câu hỏi đến chủ Homestay hoặc Admin thông qua form liên hệ.

### Tác nhân: Người dùng đã đăng nhập

### Tiền điều kiện:

Có kết nối Internet.

### Hậu điều kiện:

Tin nhắn được lưu vào hệ thống để quản trị viên xử lý.

**Các luồng sự kiện:**

|  |  |
| --- | --- |
| Bước | Mô tả |
| 1 | Người dùng truy cập trang “Liên hệ” |
| 2 | Nhập họ tên, email, tiêu đề, nội dung tin nhắn |
| 3 | Nhấn “Gửi” |
| 4 | Hệ thống lưu tin nhắn vào bảng user\_queries |
| 5 | Hiển thị thông báo “Gửi thành công” |

## 3.4.6. Use Case: Đánh giá phòng (Người dùng)

### Mô tả:

Sau khi hoàn tất lưu trú, người dùng có thể để lại đánh giá và chấm điểm cho phòng đã đặt.

### Tác nhân: Người dùng đã hoàn tất đặt phòng.

### Tiền điều kiện:

Đơn đặt phòng ở trạng thái “Hoàn tất”.  
Người dùng đã đăng nhập.

### Hậu điều kiện:

Hệ thống lưu đánh giá và hiển thị công khai trên trang chi tiết phòng.

**Các luồng sự kiện:**

|  |  |
| --- | --- |
| Bước | Mô tả |
| 1 | Người dùng truy cập lịch sử đặt phòng |
| 2 | Chọn phòng cần đánh giá |
| 3 | Nhập số sao và nội dung đánh giá |
| 4 | Nhấn “Gửi đánh giá” |
| 5 | Hệ thống lưu đánh giá, hiển thị trên chi tiết phòng |

## 3.4.7. Use Case: Quản lý người dùng (Admin)

### Mô tả:

**Admin theo dõi danh sách người dùng, có thể khóa hoặc mở lại tài khoản vi phạm.**

### Tác nhân: Admin

### Tiền điều kiện:

Admin đã đăng nhập.

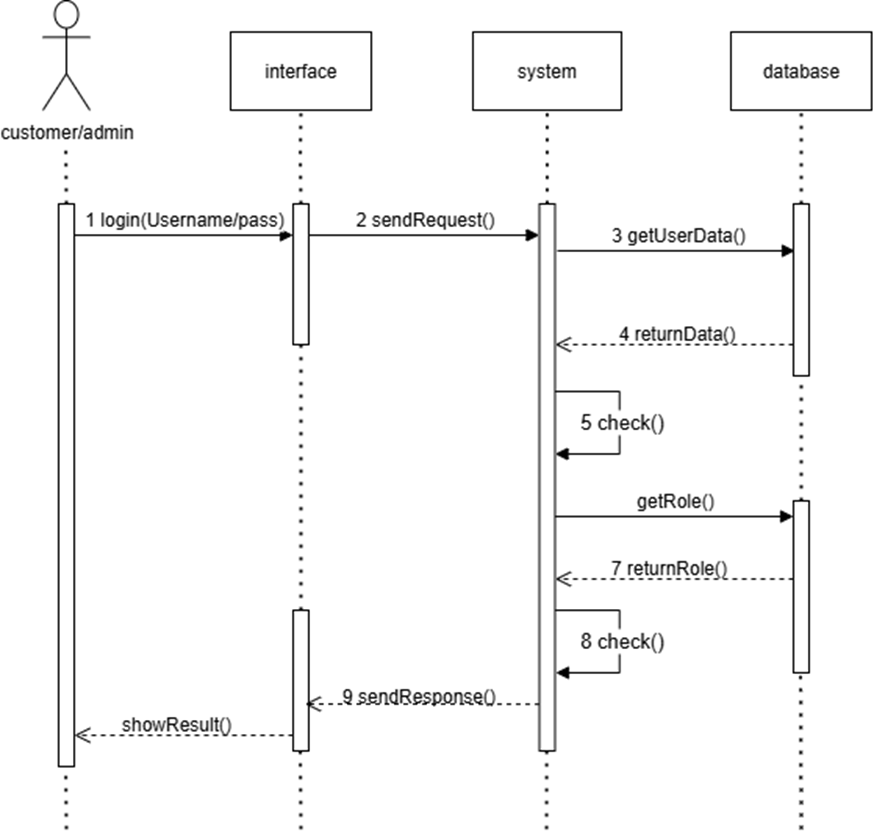
### Hậu điều kiện:

Trạng thái tài khoản người dùng được cập nhật trong cơ sở dữ liệu.

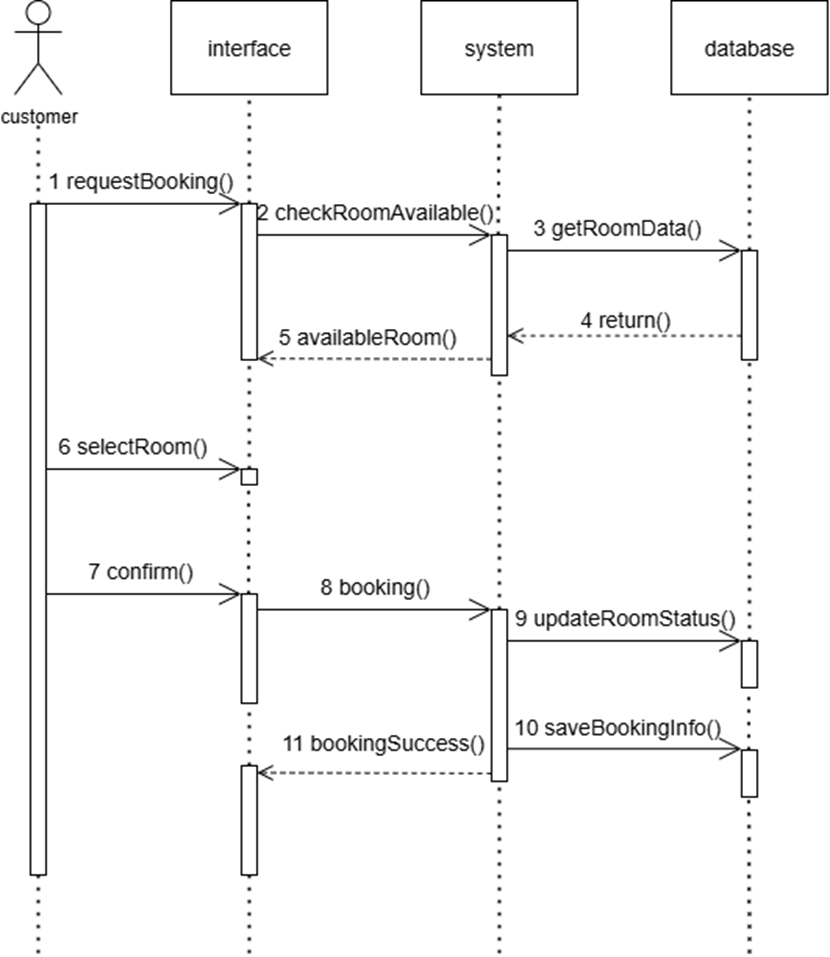
### Các luồn sự kiện:

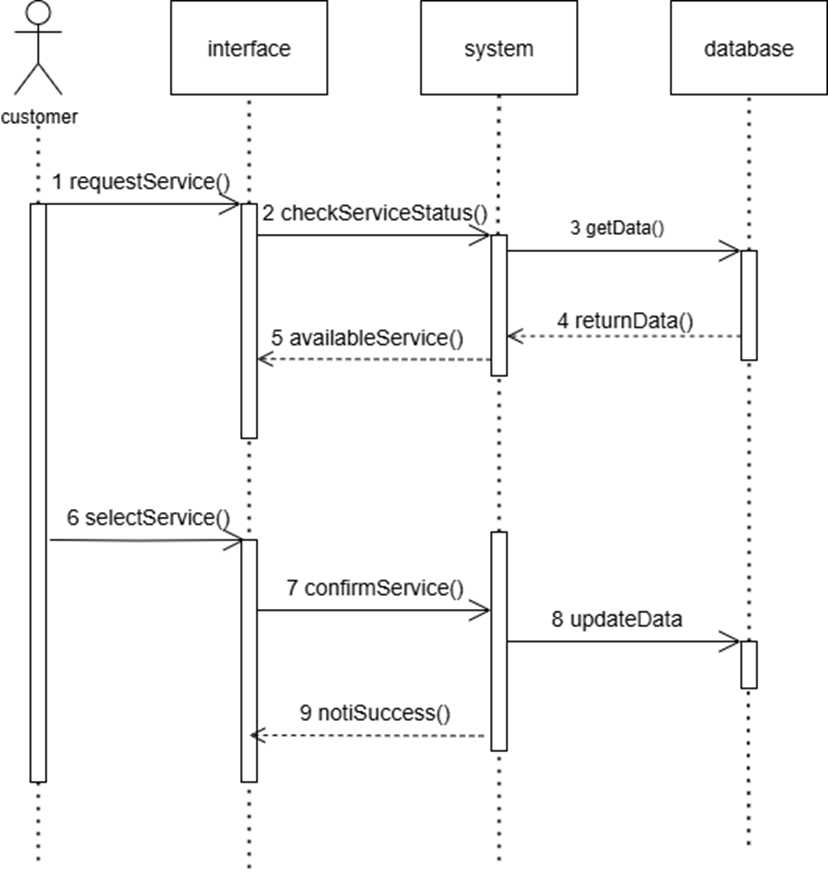
|  |  |
| --- | --- |
| Bước | Mô tả |
| 1 | Admin đăng nhập và vào phần quản lý người dùng |
| 2 | Tìm kiếm người dùng theo email hoặc tên |
| 3 | Chọn người dùng cần xử lý |
| 4 | Chọn hành động: Khóa tài khoản / Mở lại |
| 5 | Hệ thống cập nhật trạng thái trường status trong bảng user\_cred |

## 3.5. Biểu đồ tuần tự (Sequence Diagram)



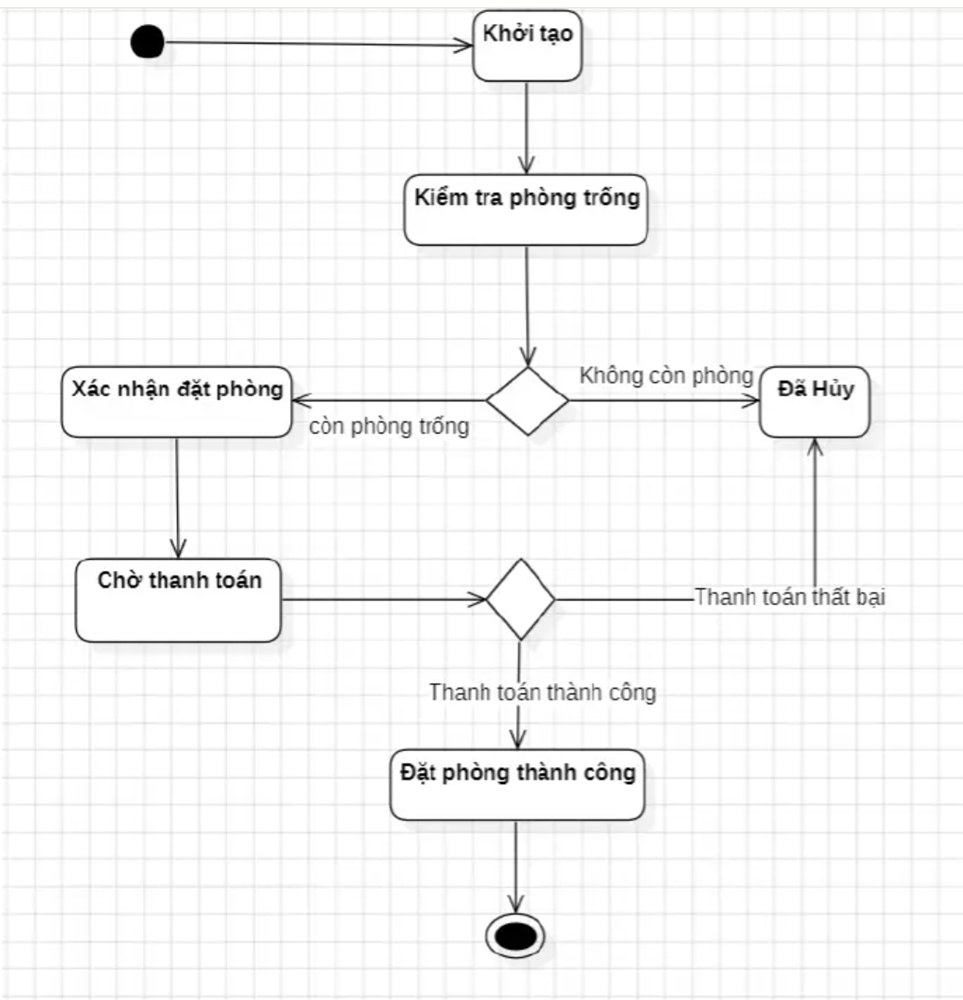
*Hình 5. Sơ đồ tuần tự đăng nhập*

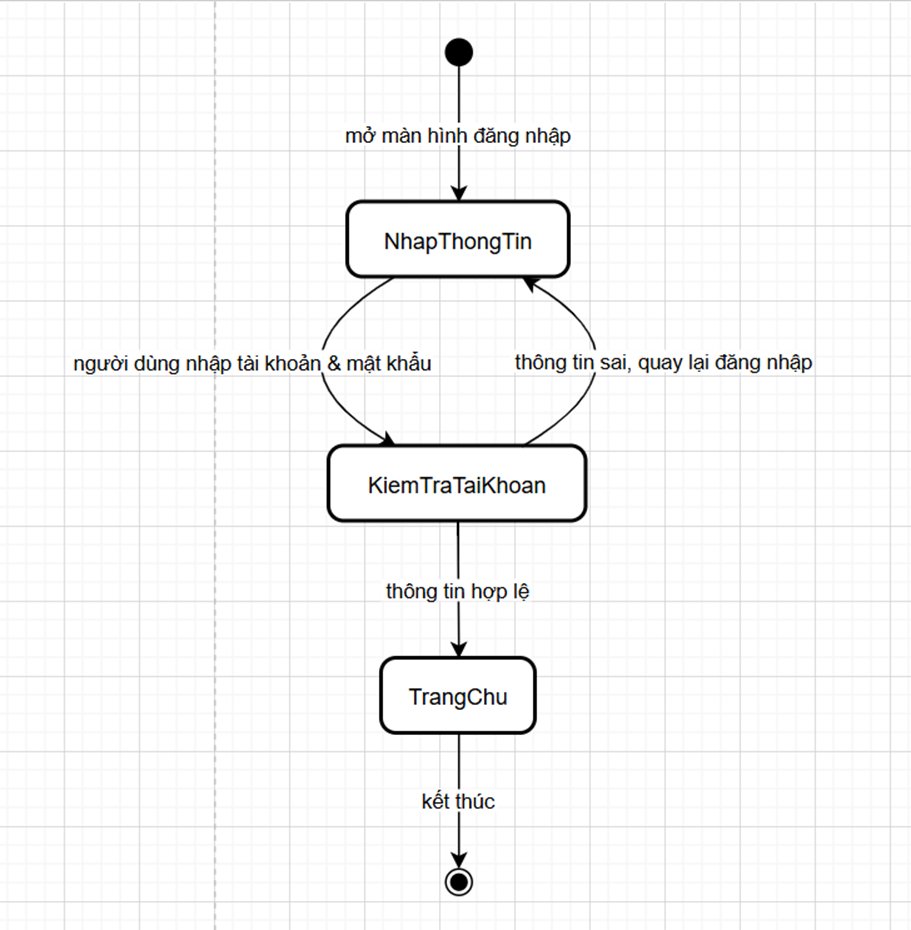
** *Hình 6. Sơ đồ tuần tự đặt phòng*

** *Hình 7. Sơ đồ tuần tự đặt dịch vụ*

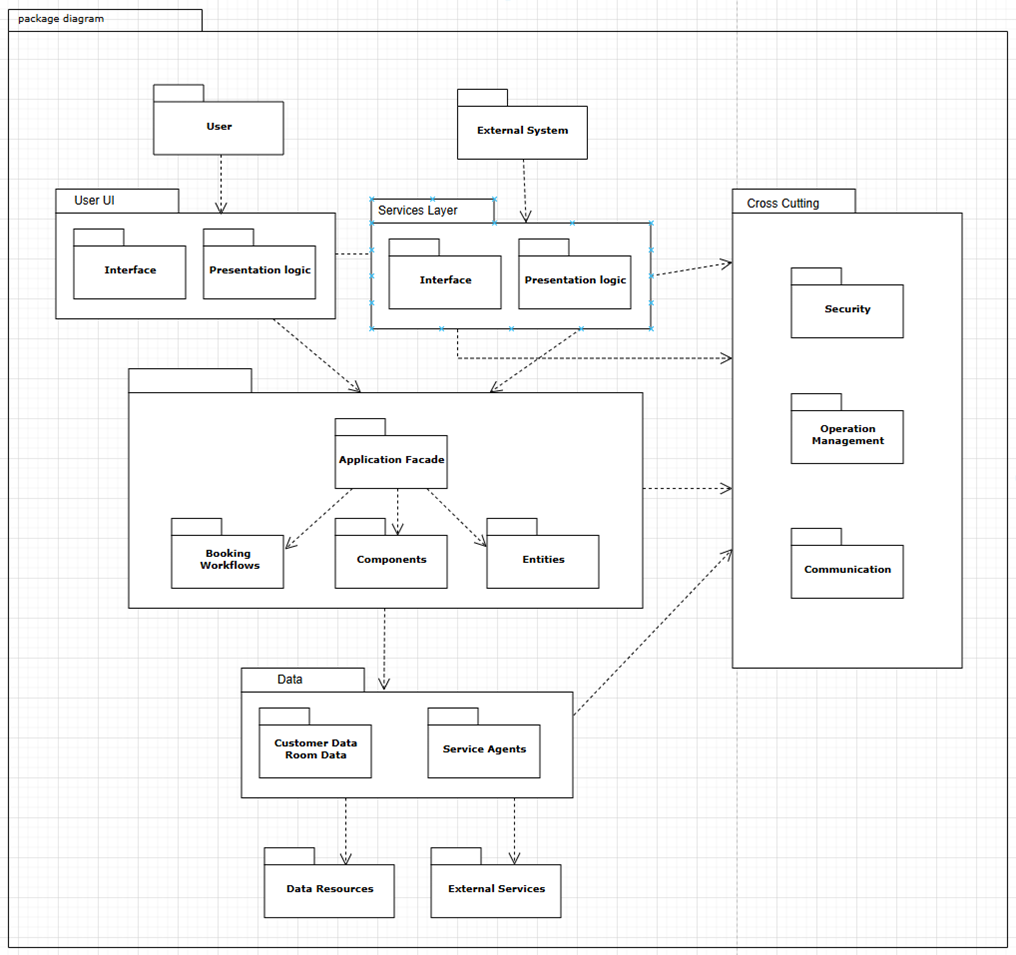
## 3.6. Biểu đồ trạng thái (State Diagram)

Mô tả trạng thái của các đối tượng chính:



* Đặt phòng: “Chờ xác nhận” → “Đã xác nhận” → “Hoàn tất” hoặc “Đã hủy”  
  
* Đăng nhập: “Chưa xác thực” → “Đã xác thực” → “Đăng xuất”

## 3.7. Biểu đồ gói (Package Diagram)



*Hình 8. Biểu đồ gói phân tầng kiến trúc hệ thống (UI – Service – Repository – Database)*

## 

## 

## 

## 

## 

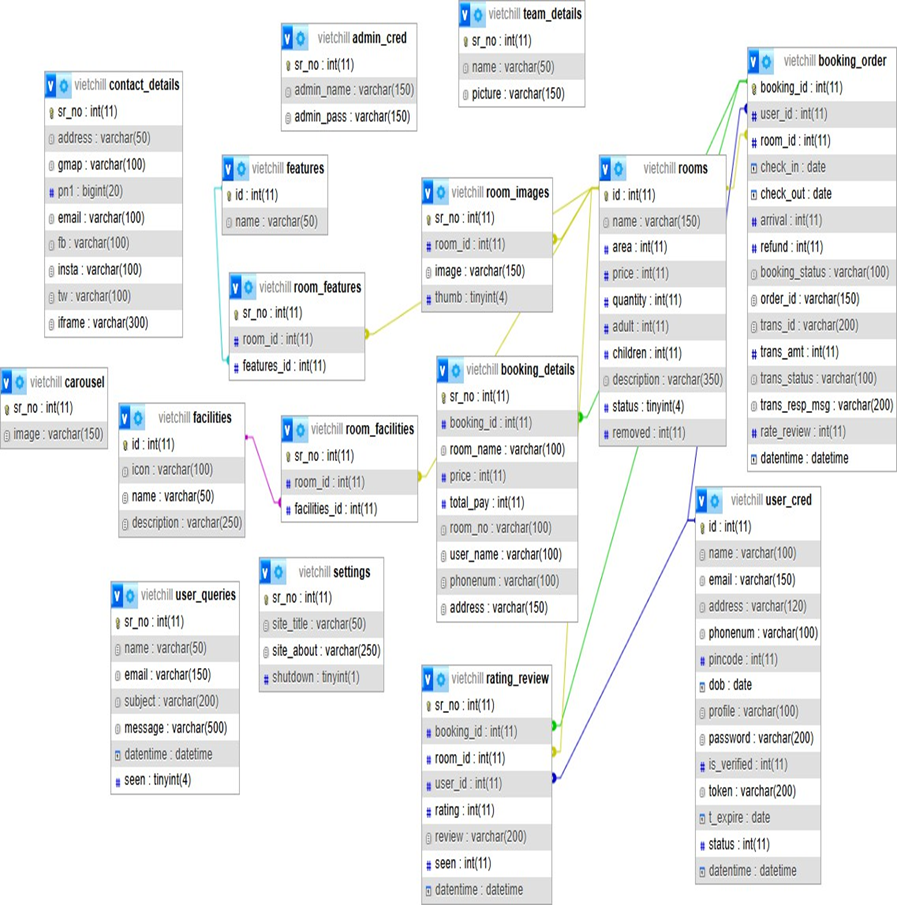
## 

## 

## 

## 

## 3.7.1 Sơ đồ database



*Hình 9. cấu trúc database*

## 3.8. Áp dụng mẫu thiết kế – Singleton Pattern

## 3.8.1. Giới thiệu

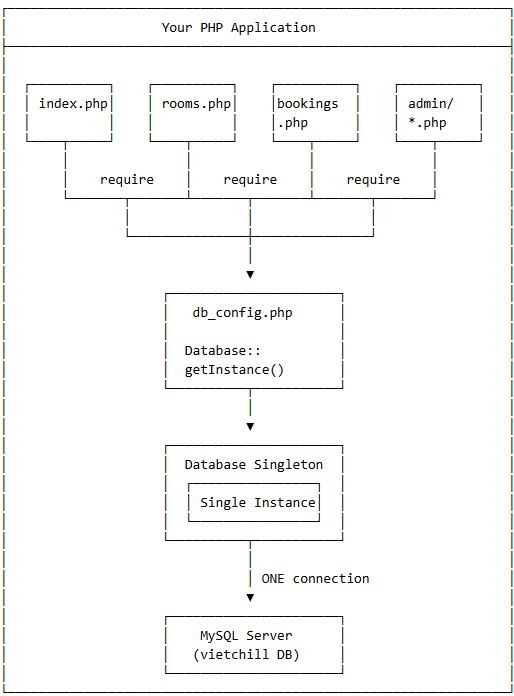
Trong quá trình xây dựng hệ thống quản lý Homestay, nhóm đã áp dụng mẫu thiết kế **Singleton (Singleton Pattern)** cho lớp quản lý kết nối cơ sở dữ liệu.  
Mục tiêu của mẫu thiết kế này là đảm bảo chỉ tồn tại duy nhất một thể hiện (instance) của lớp kết nối trong toàn bộ chương trình, giúp tối ưu hiệu năng và tránh xung đột khi nhiều phần của hệ thống truy cập cơ sở dữ liệu cùng lúc.

## 3.8.2. Lý do lựa chọn Singleton Pattern

Trong các ứng dụng web, mỗi khi một trang được tải, nếu hệ thống khởi tạo một kết nối mới đến cơ sở dữ liệu, điều này sẽ gây **tốn tài nguyên** và **làm giảm hiệu suất**.  
Việc sử dụng Singleton Pattern giúp:

Đảm bảo chỉ có một kết nối duy nhất đến cơ sở dữ liệu trong suốt vòng đời của ứng dụng.  
Tiết kiệm tài nguyên hệ thống (RAM, CPU) khi nhiều người dùng truy cập cùng lúc.  
Tăng độ ổn định và an toàn, vì tất cả các module đều dùng chung một điểm truy cập cơ sở dữ liệu.  
Dễ bảo trì: việc thay đổi thông tin kết nối (tên host, tài khoản, mật khẩu) chỉ cần thực hiện tại một nơi duy nhất.

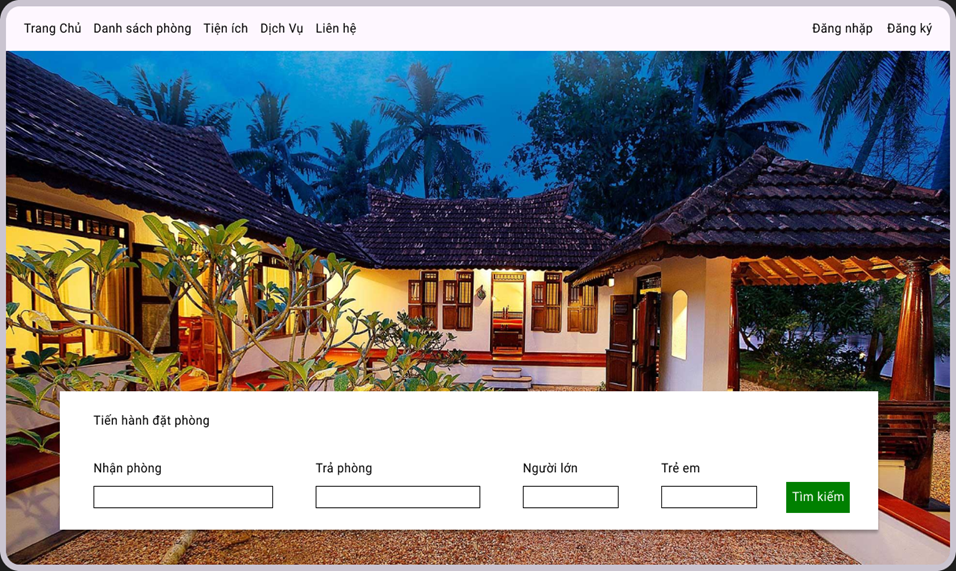
## 3.8.3. Cấu trúc triển khai Singleton trong dự án



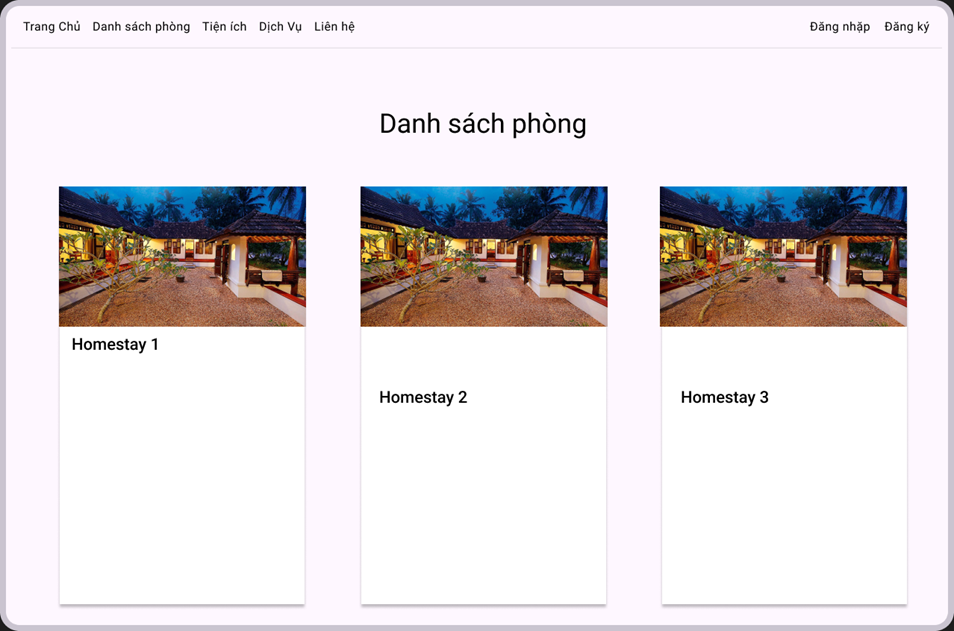
*Hình 10. cấu trúc singleton*

## 3.9. Phác thảo giao diện

Hiển thị các màn hình chính:

* Trang chủ  
  

*Hình 11. giao diện trang chủ dự kiến*

* Danh sách phòng  
  

*Hình 12. giao diện danh sách phòng sự kiến*

# TÀI LIỆU THAM KHẢO

[1]. PHP Official Documentation – <https://www.php.net/docs.php>

[2]. MySQL Documentation – <https://dev.mysql.com/doc/>

[3]. AJAX Tutorial – <https://www.w3schools.com/ajax/>

[4]. Bootstrap Documentation – <https://getbootstrap.com/docs/>

[5]. Mozilla Developer Network (MDN) – <https://developer.mozilla.org/>

[6]. Stack Overflow – <https://stackoverflow.com/>

[7]. GitHub – Nền tảng chia sẻ và lưu trữ mã nguồn – <https://github.com/>