

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: AN TOÀN THÔNG TIN

ĐỀ TÀI: XÂY DỰNG WEBSITE QUẢN LÝ KHÁCH SẠN

Người hướng dẫn: TS. NGUYỄN VĂN HIỆU

Sinh viên thực hiện: LÊ ĐỨC TUẤN

Số thẻ sinh viên: 102200239

Lớp: 20TCLC_DT5

Đà Nẵng, 6/2024

TÓM TẮT

Tên đề tài: XÂY DỰNG WEBSITE QUẢN LÝ KHÁCH SẠN

Sinh viên thực hiện: LÊ ĐỨC TUẤN

Số thẻ SV: 102200239

Lớp: 20TCLC_DT5

Xây dựng website quản lý khách sạn là một đề tài mang đến khả năng giúp các khách sạn quản lý hoạt động kinh doanh một cách hiệu quả và tiện lợi. Đề tài website quản lý khách sạn sẽ giúp người dùng dễ dàng quản lý thông tin đặt phòng, theo dõi tình trạng phòng, quản lý khách hàng, và cung cấp các thông tin cần thiết một cách nhanh chóng và chính xác.

Đề tài áp dụng các thuật toán quản lý dữ liệu, kỹ thuật bảo mật thông tin và các giao thức truyền thông để đảm bảo việc quản lý diễn ra mượt mà và an toàn. Đề xuất kỹ thuật được áp dụng cho dự án bao gồm:

- Ngôn ngữ lập trình: Java, Java Script
- Không gian/Công cụ phát triển: Visual Studio Code, IntelliJ, Post Man, My SQL WorkBench
- Framework/Thư viện: Java Spring Boot, ReactJs

Với mục tiêu đặt ra là thiết kế và triển khai được một website quản lý khách sạn cho phép người dùng có thể xem thông tin và đặt phòng khách sạn online ,người quản lý khách sạn có thể quản lý thông tin khách sạn một cách nhanh chóng và an toàn, đề tài tập trung giải quyết bài toán về quản lý dữ liệu, tối ưu hóa hiệu suất và đảm bảo tính bảo mật của hệ thống để có thể đáp ứng nhu cầu của người dùng trong môi trường thực tế.

ĐỒ ÁN TỐT NGHIỆP

ĐẠI HỌC ĐÀ NẴNG

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

TRƯỜNG ĐẠI HỌC BÁCH KHOA

Độc lập - Tự do - Hạnh phúc

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Lê Đức Tuấn

Số thẻ sinh viên: 102200239

Lớp: 20TCLC-DT5

Khoa: Công nghệ thông tin

Ngành: An toàn thông tin

1. Tên đề tài đồ án: Xây dựng Website quản lý khách sạn

2. Đề tài thuộc diện: ☐ Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện

3. Các số liệu và dữ liệu ban đầu: Không

4. Nội dung các phần thuyết minh và tính toán:

- **Mở đầu:** Giới thiệu tổng quan vấn đề, mục đích, phạm vi của đề tài, hướng tiếp cận và bố cục của đồ án.
- **Chương 1:** Cơ sở lý thuyết: Trình bày các lý thuyết, công cụ, công nghệ đã sử dụng.
- **Chương 2:** Phân tích và thiết kế hệ thống: Trình bày thiết kế hệ thống, hướng tiếp cận vấn đề.
- **Chương 3:** Triển khai và đánh giá: Trình bày cách cài đặt, vận hành. Trình bày những kết quả và đánh giá thu được
- **Kết luận:** Trình bày các kết quả đạt được, chỉ ra những hạn chế còn tồn tại và đề xuất hướng phát triển.

5. Các bản vẽ, đồ thị (ghi rõ các loại và kích thước bản vẽ): Không

6. Họ tên người hướng dẫn: TS. Nguyễn Văn Hiệu

7. Ngày giao nhiệm vụ đồ án:/...../2024

8. Ngày hoàn thành đồ án:/...../2024

Đà Nẵng, ngày tháng năm 201

Trưởng Bộ môn

Người hướng dẫn

LỜI NÓI ĐẦU

Trước tiên, em xin phép gửi lời cảm ơn tới thầy Nguyễn Văn Hiệu đã tận tình hỗ trợ, theo sát tiến độ của em trong quá trình hoàn thành dự án này.

Trong thời đại hiện đại, sự chuyển đổi toàn diện từ thế giới offline sang môi trường trực tuyến đã đặt ra những thách thức mới, nhưng cũng mở ra những cơ hội lớn trong nhiều lĩnh vực, đặc biệt là trong lĩnh vực quản lý và dịch vụ khách sạn. Đối mặt với sự phát triển không ngừng của công nghệ thông tin, việc tích hợp các giải pháp số vào quản lý khách sạn không chỉ trở nên cần thiết mà còn là chìa khóa mở ra những tiện ích và lợi ích vô song. Trong bối cảnh mà việc quản lý hiệu quả và nâng cao trải nghiệm khách hàng ngày càng trở thành ưu tiên hàng đầu, một hệ thống quản lý khách sạn trực tuyến không chỉ mang lại tầm quan trọng trong việc tăng cường hiệu quả hoạt động mà còn giúp cải thiện sự hài lòng của khách hàng.

Vậy nên, em quyết định xây dựng website quản lý khách sạn tên là Suny Hotel.

Suny Hotel cho phép các khách sạn quản lý toàn diện thông tin đặt phòng, theo dõi tình trạng phòng, quản lý khách hàng và cung cấp thông tin cần thiết một cách nhanh chóng và chính xác. Website này không chỉ là một công cụ quản lý hiệu quả mà còn là một đối tác đáng tin cậy trong việc nâng cao chất lượng dịch vụ và trải nghiệm của khách hàng. Ngoài ra, hệ thống còn có các chức năng như thống kê lịch sử đặt phòng, quản lý thông tin khách hàng, thiết lập và điều chỉnh giá phòng, cùng nhiều tính năng khác sẽ được em trình bày chi tiết hơn trong báo cáo này.

Suny Hotel được xây dựng trên nền tảng web, cho phép người dùng có thể linh động trong việc sử dụng. Với giao diện thân thiện và tính năng phong phú, Suny Hotel hứa hẹn mang lại sự tiện ích và hiệu quả cho mọi khách sạn và người dùng.

Tổng kết lại, Suny Hotel không chỉ giúp các khách sạn tối ưu hóa quy trình quản lý mà còn nâng cao chất lượng dịch vụ và sự hài lòng của khách hàng, góp phần vào sự phát triển bền vững của ngành dịch vụ khách sạn.

CAM ĐOAN

Tôi xin cam đoan:

1. Báo cáo đồ án tốt nghiệp Tên đề tài: Xây dựng Website quản lí khách sạn là do chính cá nhân tôi chính thực hiện dưới sự hướng dẫn trực tiếp của giảng viên TS. Nguyễn Văn Hiệu.

2. Tôi đã tự đọc nghiên cứu, dịch tài liệu và tổng hợp các kiến thức đã làm nên báo cáo này và đảm bảo không sao chép ở bất cứ đâu.

3. Những lý thuyết trong báo cáo đều được sử dụng tài liệu như tôi đã tham khảo ở phần tài liệu tham khảo đã có trong báo cáo.

Nếu có vi phạm, tôi xin chịu hoàn toàn trách nhiệm.

Sinh viên thực hiện

Lê Đức Tuấn

Mục lục	
NHẬN XÉT CỦA NGƯỜI HƯỚNG DẪN	ii
NHẬN XÉT CỦA NGƯỜI PHẢN BIỆN	iii
LỜI NÓI ĐẦU.....	vi
MỞ ĐẦU.....	1
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT.....	3
1.1 Công nghệ sử dụng	3
1.1.1. Xây dựng Website với ReactJs và Java SpringBoot	3
1.1.2. Xây dựng FE bằng React JS	4
1.1.3. Xây dựng BE bằng Java SpringBoot.....	6
1.1.4. Giao tiếp giữa front-end và back-end – Restful API.....	8
1.1.5. Xây dựng cơ sở dữ liệu với My Sql WordBench	10
1.1.6. Authenticate – Json Web Token.....	11
1.2 Mô hình triển khai.....	12
1.2.1. Mô hình Client-Server	12
1.2.2. Mô hình ba lớp	13
1.3 Tổng kết	15
CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....	16
2.1 Phân tích nghiệp vụ hệ thống.....	16
2.1.1. Các tác nhân và nghiệp vụ.....	16
2.1.2. Sơ đồ phân rã chức năng	17
2.2 Thiết kế hệ thống	18
2.2.1. Sơ đồ use case tổng quát của hệ thống	18
2.2.2. Sơ đồ hoạt động.....	22
2.2.2. Sơ đồ tuần tự	27
2.2.3. Cơ sở dữ liệu của hệ thống	33
2.3 Tổng kết	38
CHƯƠNG 3: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ	39
3.1 Phát triển ứng dụng	39
3.2 Kết quả triển khai.....	39
3.2.1 Giao diện bên phía người dùng	39
3.2.2 Giao diện bên phía người quản lí	48
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	54
TÀI LIỆU THAM KHẢO	56

DANH SÁCH CÁC BẢNG

<i>Bảng 2.1 Bảng mô tả tác nhân và nghiệp vụ</i>	16
<i>Bảng 2.2 Bảng đặc tả Use Case chức năng quản lý người dùng</i>	19
<i>Bảng 2.3 Bảng đặc tả Use Case chức năng quản lý phòng</i>	20
<i>Bảng 2.4 Bảng đặc tả Use Case chức năng quản lý đặt phòng</i>	21
<i>Bảng 2.5 Bảng account</i>	34
<i>Bảng 2.6 Bảng booking</i>	35
<i>Bảng 2.7 Bảng review</i>	36
<i>Bảng 2.8 Bảng booked_room</i>	36
<i>Bảng 2.9 Bảng room</i>	37
<i>Bảng 2.10 Bảng images</i>	38
<i>Bảng 2.11 Bảng facilities</i>	38

DANH SÁCH CÁC HÌNH VẼ

<i>Hình 1.1 SpringBoot kết hợp React Js</i>	3
<i>Hình 1.2 Giới thiệu Reactjs</i>	4
<i>Hình 1.3 Giới thiệu RestFul API</i>	9
<i>Hình 1.4 Giới thiệu Json Web Token</i>	11
<i>Hình 1.5 Mô hình client-server</i>	13
<i>Hình 1.6 Mô hình 3 layer</i>	14
<i>Hình 2.1 Sơ đồ phân rã chức năng</i>	17
<i>Hình 2.2 Sơ đồ Use Case tổng quát</i>	18
<i>Hình 2.3 Sơ đồ Use Case chức năng quản lý người dùng</i>	19
<i>Hình 2.4 Sơ đồ Use Case chức năng quản lý thông tin phòng khách sạn</i>	20
<i>Hình 2.5 Sơ đồ Use Case chức năng quản lý đặt phòng khách sạn</i>	21
<i>Hình 2.6 Sơ đồ hoạt động chức năng đăng nhập</i>	22
<i>Hình 2.7 Sơ đồ hoạt động chức năng đặt phòng của người dùng</i>	23
<i>Hình 2.8 Sơ đồ hoạt động chức năng quản lý người dùng</i>	24
<i>Hình 2.9 Sơ đồ hoạt động chức năng quản lý phòng</i>	25
<i>Hình 2.10 Sơ đồ hoạt động chức năng quản lý đặt phòng</i>	26
<i>Hình 2.11 Sơ đồ tuần tự chức năng đăng nhập</i>	27
<i>Hình 2.12 Sơ đồ tuần tự chức năng đặt phòng</i>	28
<i>Hình 2.13 Sơ đồ tuần tự chức năng tạo phòng</i>	29
<i>Hình 2.14 Sơ đồ tuần tự chức năng cập nhật phòng</i>	30
<i>Hình 2.15 Sơ đồ tuần tự chức năng xóa phòng</i>	31
<i>Hình 2.16 Sơ đồ tuần tự chức năng quản lý đặt phòng</i>	32
<i>Hình 2.17 Cơ sở dữ liệu</i>	33
<i>Hình 3.1 Giao diện đăng nhập</i>	34
<i>Hình 3.2 Giao diện đăng kí</i>	35
<i>Hình 3.3 Giao diện trang chủ của khách sạn</i>	41
<i>Hình 3.4 Giao diện trang xem danh sách phòng khách sạn</i>	42

<i>Hình 3.5 Giao diện trang thông tin phòng khách sạn và đặt phòng</i>	<i>43</i>
<i>Hình 3.6 Menu của người dùng để xem thông tin cá nhân</i>	<i>44</i>
<i>Hình 3.7 Dialog cập nhật thông tin cá nhân</i>	<i>45</i>
<i>Hình 3.8 Giao diện trang xem danh sách đặt phòng.....</i>	<i>46</i>
<i>Hình 3.9 Giao diện trang đánh giá phòng khách sạn</i>	<i>47</i>
<i>Hình 3.10 Giao diện trang quản lý phòng khách sạn</i>	<i>48</i>
<i>Hình 3.11 Giao diện tạo mới một phòng khách sạn.....</i>	<i>49</i>
<i>Hình 3.12 Giao diện chỉnh sửa phòng khách sạn</i>	<i>50</i>
<i>Hình 3.13 Giao diện quản lý người dùng.....</i>	<i>51</i>
<i>Hình 3.14 Giao diện chỉnh sửa thông tin người dùng.....</i>	<i>51</i>
<i>Hình 3.15 Giao diện nhận phòng trả phòng</i>	<i>52</i>
<i>Hình 3.16 Giao diện quản lý trạng thái doanh thu của khách sạn.....</i>	<i>53</i>

DANH SÁCH CÁC KÝ HIỆU, VIẾT TẮT

Chữ viết tắt	Diễn giải
SQL	Structured Query Language
API	Application Programming Interface
AOP	Aspect-Oriented Programming
MVC	Model – View – Controller
FE, BE	Front-end, Back-end
JWT	Json Web Token
DOM	Document Object Model
UI	User Interface
REST	Representational State Transfer

MỞ ĐẦU

1. Tổng quan đề tài

Trong thời đại kỹ thuật số hiện nay, sự chuyển đổi toàn diện từ thế giới offline sang môi trường trực tuyến đã đặt ra những thách thức mới, nhưng cũng mở ra nhiều cơ hội lớn trong nhiều lĩnh vực, đặc biệt là trong việc quản lý thông tin và dịch vụ khách hàng. Đối mặt với sự phát triển không ngừng của công nghệ thông tin, việc tích hợp các giải pháp số vào các hoạt động quản lý không chỉ trở nên cần thiết mà còn là chìa khóa mở ra những tiện ích và lợi ích vượt trội. Trong bối cảnh đó, việc quản lý khách sạn trực tuyến không chỉ mang lại sự tiện lợi và hiệu quả cho cả khách hàng và nhân viên mà còn nâng cao chất lượng dịch vụ và trải nghiệm khách hàng.

Suny Hotel là một hệ thống quản lý khách sạn trực tuyến, cho phép khách hàng đặt phòng, hủy đặt phòng, và xem thông tin phòng một cách dễ dàng và nhanh chóng. Đối với nhân viên, hệ thống cung cấp các chức năng quản lý đặt phòng, quản lý khách hàng, cập nhật tình trạng phòng. Ứng dụng này không chỉ là một công cụ thuận tiện mà còn là một đối tác đáng tin cậy trong việc duy trì và nâng cao chất lượng dịch vụ khách sạn. Với giao diện thân thiện và tính năng phong phú, Suny Hotel hứa hẹn mang lại sự tiện ích và hiệu quả cho cả khách hàng và nhân viên khách sạn.

2. Mục tiêu

- **Về lý thuyết**
 - + Tìm hiểu về các thư viện, framework và công cụ hữu ích khi phát triển ứng dụng web, chẳng hạn như MySQL Workbench, Hibernate, JWT, v.v.
 - + Sử dụng thành thạo công cụ quản lý mã nguồn Git.
- **Về ứng dụng**
 - + Từ các kiến thức về mặt lý thuyết áp dụng vào xây dựng ứng dụng quản lý khách sạn trực tuyến với giao diện hấp dẫn, hệ thống quản lý cơ sở dữ liệu hiệu quả với MySQL, tích hợp các tính năng quản lý đặt phòng, quản lý khách hàng và thông tin phòng, đảm bảo quyền riêng tư và an toàn dữ liệu cho người dùng.

3. Đối tượng nghiên cứu

- Nhu cầu quản lý đặt phòng và khách hàng của khách sạn.
- Xây dựng và phát triển ứng dụng web quản lý khách sạn.
- Các phương pháp định vị và quản lý dữ liệu trên ứng dụng web.

4. Phương pháp thực hiện

- Với những mục tiêu nêu trên, để có được kết quả tốt nhất, em tiến hành phát triển hệ thống quản lý khách sạn trực tuyến theo các bước:
- Thực hiện nghiên cứu, tìm hiểu các yêu cầu mà hệ thống cần phải đáp ứng để đưa ra các chức năng của ứng dụng.
- Tìm hiểu, xây dựng hệ thống cơ sở dữ liệu với MySQL.
- Áp dụng các thư viện và framework để xây dựng giao diện và các chức năng cho hệ thống.

5. Bố cục báo cáo

Báo cáo bao gồm các nội dung sau:

Mở đầu

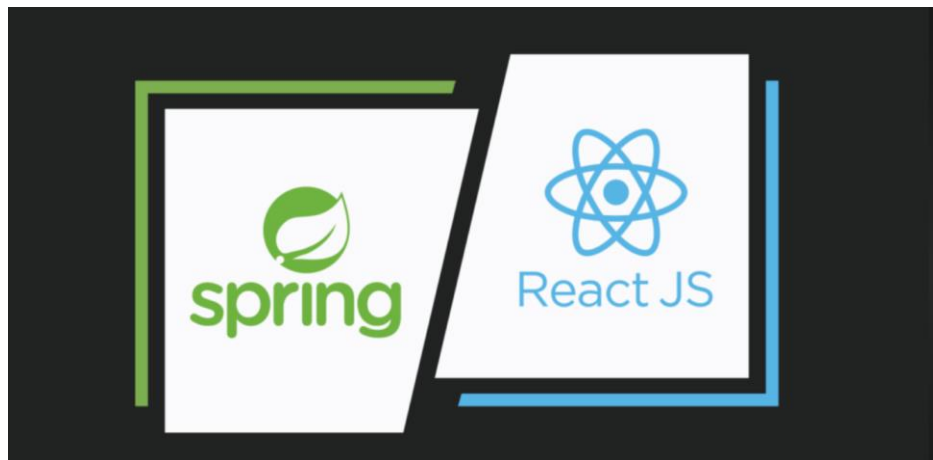
- Chương 1: Cơ sở lý thuyết
- Chương 2: Phân tích thiết kế hệ thống
- Chương 3: Triển khai và đánh giá kết quả
- Kết luận và hướng phát triển

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1 Công nghệ sử dụng

1.1.1. Xây dựng Website với ReactJs và Java SpringBoot

ReactJS là một thư viện JavaScript mạnh mẽ do Facebook phát triển, cho phép tạo ra các ứng dụng web nhanh chóng và hiệu quả với giao diện người dùng phong phú và hiệu suất cao. Khi kết hợp với Spring Boot, một framework phát triển backend dựa trên Java, các nhà phát triển có thể xây dựng các ứng dụng web toàn diện với khả năng tùy biến linh hoạt. Sử dụng ReactJS cho frontend và Spring Boot cho backend giúp tối ưu hóa quy trình phát triển, nâng cao hiệu quả và chất lượng sản phẩm



Hình 1.1 SpringBoot kết hợp React Js

Tại sao lại chọn ReactJS kết hợp với Spring Boot?

Tiết kiệm thời gian và chi phí: Việc sử dụng ReactJS cho frontend và Spring Boot cho backend giúp giảm thiểu công sức và thời gian phát triển nhờ khả năng tái sử dụng mã nguồn và phân chia công việc hiệu quả giữa frontend và backend.

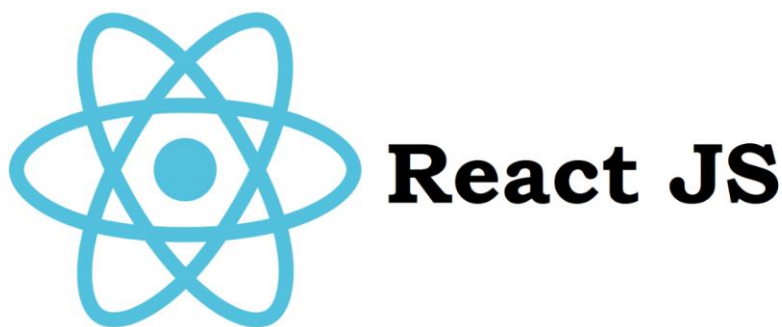
Hiệu suất cao: ReactJS cung cấp khả năng cập nhật và render giao diện người dùng một cách nhanh chóng và hiệu quả, trong khi Spring Boot cung cấp một nền tảng mạnh mẽ cho việc phát triển các ứng dụng backend với hiệu suất cao.

Giao diện phong phú: ReactJS cung cấp nhiều thư viện và component tùy biến, giúp tạo ra giao diện người dùng đẹp mắt và nhất quán trên các thiết bị khác nhau.

Hỗ trợ mạnh mẽ: Cả ReactJS và Spring Boot đều có cộng đồng lớn và tài liệu phong phú, giúp dễ dàng tìm kiếm hỗ trợ và giải quyết vấn đề trong quá trình phát triển.

1.1.2. Xây dựng FE bằng React JS

React.js là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook, được sử dụng rộng rãi để xây dựng giao diện người dùng (UI) cho các ứng dụng web và di động. React tập trung vào việc xây dựng các component, giúp cho việc tái sử dụng và quản lý mã nguồn trở nên dễ dàng hơn.



Hình 1.2 Giới thiệu Reactjs

Ưu điểm:

- Component-Based Architecture (Kiến trúc dựa trên Component):
 - + Tái sử dụng: Các component có thể tái sử dụng giúp giảm thiểu mã nguồn lặp lại và làm cho việc phát triển trở nên dễ dàng hơn.
 - + Dễ bảo trì: Mỗi component có thể được quản lý và cập nhật một cách độc lập, giúp dễ dàng bảo trì và mở rộng ứng dụng.
- Virtual DOM (DOM Ảo):
 - + Hiệu suất cao: Virtual DOM cải thiện hiệu suất render bằng cách chỉ cập nhật các phần của DOM thực sự cần thay đổi. Điều này làm giảm thiểu số lượng thao tác với DOM, giúp ứng dụng chạy nhanh và mượt mà hơn.
- JSX:
 - + Dễ đọc và viết: JSX cho phép viết mã HTML trong JavaScript, làm cho mã nguồn trở nên trực quan và dễ hiểu hơn.

- + Động: Kết hợp JavaScript với HTML, giúp tạo ra các UI phức tạp dễ dàng hơn.
- React Hooks:
 - + Quản lý state dễ dàng: Hooks cung cấp các API đơn giản để quản lý state và các side effect trong component function.
 - + Giảm sự phức tạp: Hooks loại bỏ nhu cầu sử dụng class-based components, giúp mã nguồn ngắn gọn và dễ quản lý hơn.
- Large Ecosystem (Hệ sinh thái lớn):
 - + Cộng đồng mạnh mẽ: React có một cộng đồng lớn và sôi động, cung cấp nhiều tài liệu, hướng dẫn và hỗ trợ.
 - + Thư viện phong phú: Nhiều thư viện và công cụ hỗ trợ như Redux, React Router, và Material-UI giúp mở rộng và tăng cường chức năng của React.

Nhược điểm:

- JSX không quen thuộc: JSX là một cú pháp mở rộng của JavaScript, cho phép bạn viết HTML trực tiếp trong JavaScript. Mặc dù mạnh mẽ, nhưng cú pháp này có thể gây khó khăn cho những người không quen với nó hoặc những người đến từ các ngôn ngữ lập trình khác.
- SEO: Mặc dù React có thể được cấu hình để hỗ trợ SEO tốt hơn thông qua các kỹ thuật như server-side rendering (SSR), việc thiết lập và duy trì SSR có thể phức tạp và đòi hỏi thêm tài nguyên.
- Không phải là giải pháp toàn diện: React chỉ là một thư viện cho việc xây dựng giao diện người dùng. Để xây dựng một ứng dụng hoàn chỉnh, bạn thường cần phải kết hợp với các thư viện khác cho các chức năng như quản lý trạng thái (Redux, MobX), routing (React Router), và các công cụ khác, làm tăng sự phức tạp.
- Performance issues: Mặc dù React được thiết kế để tối ưu hóa hiệu suất, nhưng việc sử dụng không đúng cách hoặc không tối ưu có thể dẫn đến các vấn đề về hiệu suất, chẳng hạn như rendering lại không cần thiết.

1.1.3. Xây dựng BE bằng Java SpringBoot

Spring là một framework mã nguồn mở mạnh mẽ cho việc phát triển các ứng dụng Java doanh nghiệp. Được phát triển bởi Rod Johnson và lần đầu ra mắt vào năm 2002, Spring cung cấp một bộ công cụ toàn diện giúp quản lý các phụ thuộc, xử lý dữ liệu, và xây dựng các dịch vụ web một cách dễ dàng và hiệu quả. Với một cộng đồng phát triển lớn và một hệ sinh thái phong phú, Spring đã trở thành một lựa chọn phổ biến cho việc xây dựng backend của nhiều ứng dụng quy mô lớn.

Ưu điểm:

- **Dependency Injection:**
 - + **Linh hoạt:** Giúp quản lý các phụ thuộc giữa các đối tượng dễ dàng, làm cho mã nguồn linh hoạt và dễ bảo trì.
 - + **Kiểm tra dễ dàng:** DI hỗ trợ việc kiểm thử đơn vị bằng cách cho phép thay thế các phụ thuộc bằng các đối tượng giả (mock objects).
- **Spring Boot:**
 - + **Khởi động nhanh:** Giúp đơn giản hóa việc cấu hình và khởi động ứng dụng với các cấu hình mặc định và công cụ tiện lợi.
 - + **Microservices:** Hỗ trợ mạnh mẽ cho việc xây dựng các dịch vụ vi mô, với các tính năng như cấu hình tự động, quản lý phụ thuộc, và tích hợp với các công cụ DevOps.
- **Spring MVC:**
 - + **Kiến trúc MVC:** Tách biệt rõ ràng giữa logic nghiệp vụ, giao diện người dùng và điều khiển, giúp mã nguồn dễ quản lý và mở rộng.
- **Spring Data:**
 - + **Quản lý dữ liệu dễ dàng:** Cung cấp các giải pháp mạnh mẽ và dễ sử dụng cho việc làm việc với cơ sở dữ liệu, hỗ trợ nhiều loại cơ sở dữ liệu như SQL, NoSQL và cơ sở dữ liệu phân tán.
- **Spring Security:**
 - + **Bảo mật toàn diện:** Cung cấp các cơ chế bảo mật mạnh mẽ và linh hoạt, giúp bảo vệ ứng dụng khỏi các mối đe dọa bảo mật thông thường.
- **Spring Cloud:**
 - + **Hệ thống phân tán:** Cung cấp các công cụ và dịch vụ để xây dựng các hệ thống phân tán và dịch vụ vi mô, hỗ trợ các tính năng như cân bằng tải, quản lý cấu hình và khám phá dịch vụ.
- **Cộng đồng lớn và tài liệu phong phú:**
 - **Hỗ trợ tốt:** Với một cộng đồng người dùng rộng lớn và nhiều tài liệu, giúp giải quyết các vấn đề nhanh chóng và hiệu quả.

Nhược điểm:

- Độ phức tạp:
 - + Khó học: Số lượng lớn các thành phần và tính năng của Spring có thể làm cho việc học tập trở nên phức tạp và khó khăn đối với người mới bắt đầu.
 - + Cấu hình phức tạp: Dù Spring Boot đã giảm bớt phần nào sự phức tạp, việc cấu hình các ứng dụng lớn và phức tạp vẫn tốn nhiều thời gian và công sức.
- Hiệu suất:
 - + Chi phí tài nguyên: Nếu không được tối ưu hóa đúng cách, các ứng dụng Spring có thể đòi hỏi nhiều tài nguyên hệ thống hơn.
 - + Thời gian khởi động: Các ứng dụng Spring, đặc biệt khi sử dụng nhiều thư viện, có thể mất thời gian khởi động lâu hơn.
- Quản lý phụ thuộc:
 - + Xung đột phụ thuộc: Khi sử dụng nhiều thư viện và module, có thể xảy ra xung đột phụ thuộc, đòi hỏi phải quản lý và giải quyết cẩn thận.
- Kích thước ứng dụng:
 - + Ứng dụng lớn: Các ứng dụng Spring thường có kích thước lớn do việc tích hợp nhiều thành phần và thư viện, làm tăng kích thước của file JAR/WAR.

Kết luận:

Spring là một framework mạnh mẽ và linh hoạt cho việc xây dựng backend của các ứng dụng Java. Với các ưu điểm như tiêm phụ thuộc, hỗ trợ microservices, và bảo mật toàn diện, Spring giúp phát triển các ứng dụng từ nhỏ đến lớn một cách hiệu quả. Tuy nhiên, nhược điểm về độ phức tạp và quản lý phụ thuộc cần được xem xét kỹ lưỡng. Việc nắm vững Spring có thể mở ra nhiều cơ hội trong lĩnh vực phát triển phần mềm, đặc biệt trong các ứng dụng doanh nghiệp và dịch vụ vi mô.

1.1.4. Giao tiếp giữa front-end và back-end – Restful API

Web API là phương thức dùng để cho phép các ứng dụng khác nhau có thể giao tiếp, trao đổi dữ liệu qua lại. Dữ liệu được web api trả về thường có dạng JSON hoặc XML thông qua giao thức HTTP hoặc HTTPS.

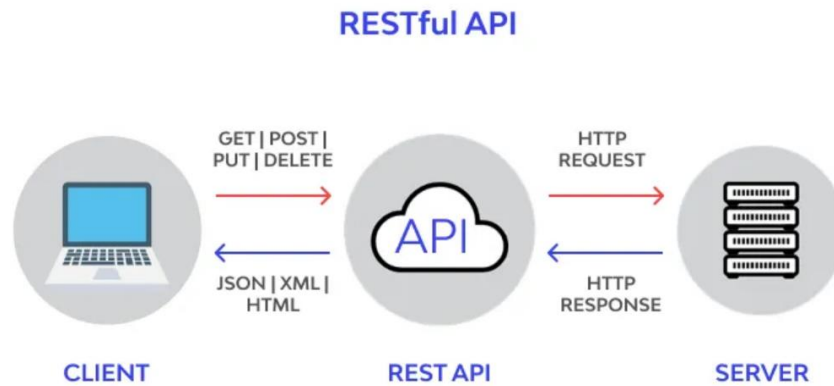
Một số điểm nổi bật:

- Web API hỗ trợ restful đầy đủ các phương thức: GET, POST, PUT, DELETE dữ liệu, giúp xây dựng các HTTP service một cách rất đơn giản và nhanh chóng. Web API cũng có khả năng hỗ trợ đầy đủ các thành phần HTTP: URI, request/response, header, caching, versioning, content format.
- Tự động hóa sản phẩm: với web api chúng ta sẽ tự động hóa quản lý công việc, cập nhật luồng công việc, giúp tăng năng suất và hiệu quả công việc hơn.
- Khả năng tích hợp linh động: cho phép lấy nội dung từ một website một cách dễ dàng nếu được cho phép, tăng trải nghiệm người dùng, API hoạt động như một chiếc công, cho phép các công ty chia sẻ thông tin được chọn nhưng vẫn tránh được những yêu cầu không mong muốn.

RESTful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động...), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.

Chức năng quan trọng nhất của REST là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE...) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một RESTful API.

Cách thức hoạt động:



Hình 1.3 Giới thiệu RestFul API

- REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu dưới sẽ sử dụng những phương thức HTTP riêng.
- GET: Trả về một Resource hoặc một danh sách Resource.
- POST: Tạo mới một Resource.
- PUT: Cập nhật thông tin cho Resource.
- PATCH: Cập nhật thông tin cho một phần Resource.
- DELETE: Xoá một Resource.

Một số status code khi ta request một API:

- 200 OK – Trả về thành công cho những phương thức GET, PUT, PATCH hoặc DELETE.
- 304 Not Modified – Client có thể sử dụng dữ liệu cache.
- 400 Bad Request – Request không hợp lệ
- 401 Unauthorized – Request cần có auth.
- 403 Forbidden – bị từ chối không cho phép.
- 404 Not Found – Không tìm thấy resource từ URI
- 405 Method Not Allowed – Phương thức không cho phép với user hiện tại.
- 410 Gone – Resource không còn tồn tại, Version cũ đã không còn hỗ trợ.
- 429 Too Many Requests – Request bị từ chối do bị giới hạn.

Một số quy ước khi triển khai API:

- Nên sử dụng đúng status code: nếu API bị lỗi, hãy trả về đúng status code, tránh trường hợp luôn trả về status 2xx.
- Không nên dùng underscore (_), nên dùng hyphen (-)
- Nên viết thường trong URL
- Không nên sử dụng đuôi file extension trong URI (VD:.html,.xml,.json,...)

1.1.5. Xây dựng cơ sở dữ liệu với My Sql WordBench

MySQL Workbench là một công cụ mạnh mẽ được phát triển và duy trì bởi Oracle, hỗ trợ các nhà phát triển trong việc thiết kế, phát triển, và quản lý cơ sở dữ liệu MySQL.

MySQL Workbench cung cấp một loạt các công cụ toàn diện để giúp việc quản lý cơ sở dữ liệu trở nên dễ dàng và hiệu quả hơn. Nó giúp giảm bớt gánh nặng của việc quản lý cơ sở dữ liệu và cho phép các nhà phát triển tập trung vào việc tạo ra các tính năng và trải nghiệm người dùng tốt hơn.

Các thành phần chính của MySQL Workbench:

- SQL Development: MySQL Workbench cung cấp một giao diện đồ họa mạnh mẽ để viết và thực thi các câu lệnh SQL, quản lý các phiên làm việc, và duyệt dữ liệu một cách trực quan. Nó giúp các nhà phát triển làm việc với cơ sở dữ liệu một cách hiệu quả hơn.
- Data Modeling: Công cụ này cho phép thiết kế, tạo và quản lý các mô hình dữ liệu phức tạp. Người dùng có thể tạo các mô hình EER (Entity-Relationship), thực hiện reverse engineering từ cơ sở dữ liệu hiện có và tạo các mô hình logic và vật lý.
- Database Design & Maintenance: MySQL Workbench cung cấp các công cụ để thiết kế và bảo trì cơ sở dữ liệu, bao gồm các tính năng như forward engineering, reverse engineering, và khả năng tạo các sơ đồ ERD (Entity Relationship Diagram) từ các cơ sở dữ liệu hiện có.
- Visual Database Design: Người dùng có thể trực quan hóa cấu trúc cơ sở dữ liệu, tạo các bảng, khóa, và quan hệ một cách trực quan. Điều này giúp đơn giản hóa quá trình thiết kế cơ sở dữ liệu và đảm bảo tính nhất quán của cấu trúc dữ liệu.

- **Server Administration:** MySQL Workbench cung cấp các công cụ quản lý máy chủ mạnh mẽ, bao gồm quản lý người dùng và quyền hạn, sao lưu và phục hồi, và cấu hình máy chủ. Nó giúp quản trị viên cơ sở dữ liệu dễ dàng quản lý và bảo trì hệ thống.
- **Data Migration:** Công cụ này hỗ trợ di chuyển dữ liệu từ các hệ quản trị cơ sở dữ liệu khác sang MySQL, bao gồm các hệ quản trị cơ sở dữ liệu như Oracle, Microsoft SQL Server, và PostgreSQL. Nó cung cấp một quy trình di chuyển dữ liệu mượt mà và hiệu quả.

Lợi ích khi sử dụng MySQL Workbench:

- **Dễ dàng tích hợp:** MySQL Workbench cung cấp các công cụ và giao diện trực quan, dễ dàng tích hợp vào các quy trình phát triển và quản lý cơ sở dữ liệu.
- **Thời gian triển khai nhanh chóng:** Các công cụ của MySQL Workbench được thiết kế để giảm thiểu thời gian thiết lập và triển khai, giúp các nhà phát triển tập trung vào việc phát triển tính năng.
- **Bảo mật và mở rộng:** MySQL Workbench sử dụng cơ sở hạ tầng mạnh mẽ của MySQL, đảm bảo bảo mật và khả năng mở rộng cho các cơ sở dữ liệu từ nhỏ đến lớn.
- **Tích hợp toàn diện:** MySQL Workbench cung cấp một bộ công cụ toàn diện từ thiết kế, phát triển, đến quản lý và bảo trì cơ sở dữ liệu, giúp tối ưu hóa quy trình làm việc và nâng cao hiệu quả phát triển.

Sử dụng MySQL Workbench không chỉ giúp tối ưu hóa quy trình quản lý cơ sở dữ liệu mà còn nâng cao hiệu quả và chất lượng của sản phẩm cuối cùng.

1.1.6. Authenticate – Json Web Token



Hình 1.4 Giới thiệu Json Web Token

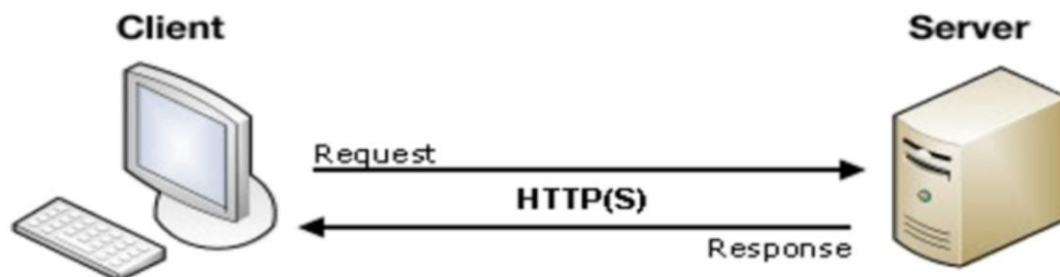
JSON Web Mã (JWT) [7] là một chuẩn mở (RFC 7519) định nghĩa một cách nhỏ gọn và khép kín để truyền một cách an toàn thông tin giữa các bên dưới dạng đối tượng JSON. Thông tin này có thể được xác minh và đáng tin cậy vì có chứa chữ ký số. JWTs có thể được ký bằng một thuật toán bí mật (với thuật toán HMAC) hoặc một public / private key sử dụng mã hoá RSA. Thành phần chính:

- Header: Bao gồm hai phần chính: loại token (mặc định là JWT - Thông tin này cho biết đây là một Token JWT) và thuật toán đã dùng để mã hóa (HMAC SHA256 - HS256 hoặc RSA);
- Payload: Chứa các claims. Claims là một các biểu thức về một thực thể (chẳng hạn user) và một số metadata phụ trợ. Có 3 loại claims thường gặp trong Payload: reserved, public và private claims.
 - + Reserved claims: Reserved claims: Đây là một số metadata được định nghĩa trước, trong đó một số metadata là bắt buộc, số còn lại nên tuân theo để JWT hợp lệ và đầy đủ thông tin: iss (issuer), iat (issued-at time) exp (expiration time), sub (subject), aud (audience), jti (Unique Identifier cho JWT, Có thể được sử dụng để ngăn JWT bị thay thế);
 - + Public Claims - Claims được cộng đồng công nhận và sử dụng rộng rãi;
 - + Private Claims: Claims tự định nghĩa (không được trùng với Reserved Claims và Public Claims), được tạo ra để chia sẻ thông tin giữa 2 parties đã thỏa thuận và thống nhất trước đó.
- Signature: Chữ ký Signature trong JWT là một chuỗi được mã hóa bởi header, payload cùng với một chuỗi.

1.2 Mô hình triển khai

1.2.1. Mô hình Client-Server

Mô hình Client-Server (khách hàng - máy chủ) [9] là một kiến trúc phân tán phổ biến trong lĩnh vực phát triển phần mềm. Trong mô hình này, ứng dụng được chia thành hai thành phần chính: client (khách hàng) và server (máy chủ), mỗi thành phần có trách nhiệm riêng.



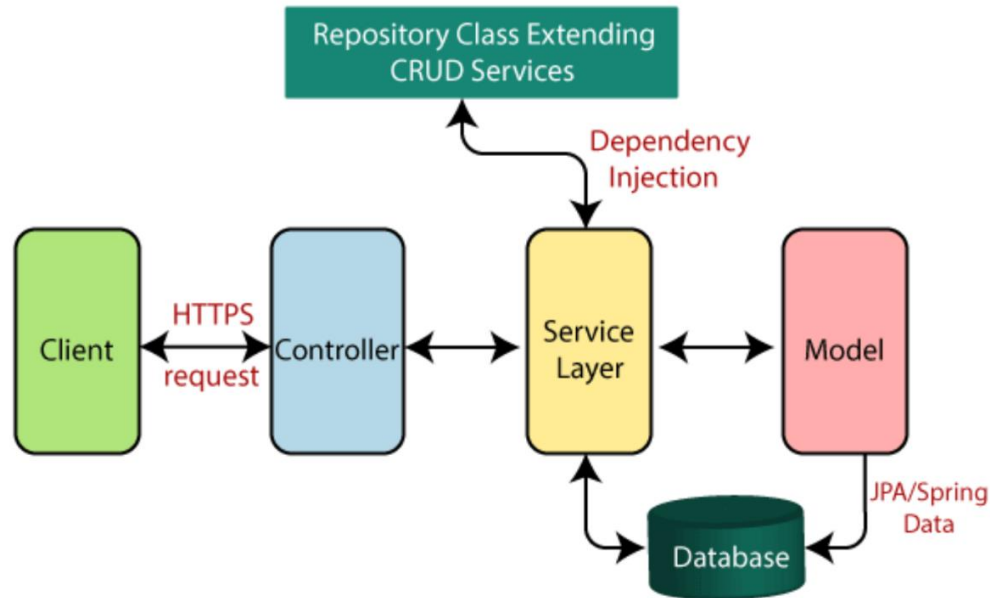
Hình 1.5 Mô hình client-server

- Client (Khách hàng):
 - + Đây là phần của ứng dụng chạy trên máy tính hoặc thiết bị của người dùng cuối (người dùng);
 - + Chức năng của client là gửi yêu cầu đến server và nhận phản hồi từ server;
 - + Client thường có giao diện người dùng (UI) để tương tác với người dùng, hiển thị dữ liệu và thực hiện các thao tác người dùng;
 - + Client gửi yêu cầu (request) đến server và chờ nhận phản hồi (response).
- Server (Máy chủ):
 - + Đây là phần của ứng dụng chạy trên máy chủ hoặc nền tảng phía máy chủ;
 - + Chức năng của server là xử lý yêu cầu từ client và gửi lại kết quả (response) cho client;
 - + Server thường chứa logic kinh doanh, xử lý dữ liệu và tương tác với cơ sở dữ liệu;
 - + Server có thể cung cấp dịch vụ cho nhiều client khác nhau và xử lý các yêu cầu từ chúng;
 - + Server có khả năng xử lý đồng thời nhiều yêu cầu từ client khác nhau.

Mô hình Client-Server cho phép tách biệt giữa phần giao diện người dùng và phần xử lý logic kinh doanh và cơ sở dữ liệu. Giúp tăng tính mô đun hóa, khả năng mở rộng và quản lý dễ dàng trong quá trình phát triển ứng dụng. Ngoài ra, mô hình này cũng cho phép phân chia trách nhiệm và phân phối công việc giữa client và server để đạt hiệu suất và hiệu quả tốt hơn trong việc xử lý yêu cầu và truy cập dữ liệu.

1.2.2. Mô hình ba lớp

Mô hình 3 lớp [10] trong Spring Boot là một kiến trúc phần mềm phổ biến được sử dụng để phân chia ứng dụng thành các lớp khác nhau để tăng tính bảo mật, quản lý dễ dàng và tái sử dụng mã.



Hình 1.6 Mô hình 3 layer

Dưới đây là mô tả của từng lớp trong mô hình 3 lớp của Spring Boot:

- Lớp Presentation Layer (Lớp Giao diện): Là lớp đầu tiên mà người dùng tương tác khi sử dụng ứng dụng. Nhiệm vụ là nhận và xử lý các yêu cầu từ người dùng và hiển thị dữ liệu cho họ. Trong Spring Boot, lớp Presentation Layer thường được triển khai bằng các API REST hoặc các Controller. Tương tác với lớp Business Layer để truy xuất dữ liệu và xử lý yêu cầu từ người dùng;
- Lớp Business Layer (Lớp Logic kinh doanh): Chứa logic kinh doanh của ứng dụng. Xử lý các yêu cầu từ lớp Presentation Layer, thực hiện các quy tắc kinh doanh, gọi các phương thức từ lớp Data Access Layer để truy xuất và cập nhật dữ liệu. Lớp Business Layer giúp tách biệt logic kinh doanh khỏi giao diện và cung cấp sự linh hoạt và khả năng tái sử dụng mã;
- Lớp Data Access Layer (Lớp Truy xuất dữ liệu): là nơi chúng ta thao tác với cơ sở dữ liệu. Cung cấp các phương thức để truy xuất và cập nhật dữ liệu từ cơ sở dữ liệu. Trong Spring Boot, lớp Data Access Layer thường được triển khai bằng các Repository hoặc DAO (Data

Access Object). Tương tác với cơ sở dữ liệu và truyền dữ liệu giữa lớp Business Layer và cơ sở dữ liệu.

Trong mô hình 3 lớp, DTO (Data Transfer Object) được sử dụng để chuyển dữ liệu giữa các lớp trong hệ thống. DTO là một lớp đơn giản, chỉ chứa các thuộc tính và các phương thức getter/setter để truy cập dữ liệu.

Mô hình 3 lớp trong Spring Boot giúp phân chia rõ ràng và tách biệt các lớp chức năng, tăng tính module hóa và dễ dàng quản lý và bảo trì ứng dụng. Tạo điều kiện cho việc phát triển song song của các lớp và khả năng kiểm thử đơn vị dễ dàng.

1.3 Tổng kết

Chương này trình bày về các công nghệ, nền tảng đã sử dụng trong quá trình triển khai đề tài. Ngoài ra còn mô tả chi tiết các mô hình và kỹ thuật đã áp dụng để xây dựng hệ thống.

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

2.1 Phân tích nghiệp vụ hệ thống

2.1.1. Các tác nhân và nghiệp vụ

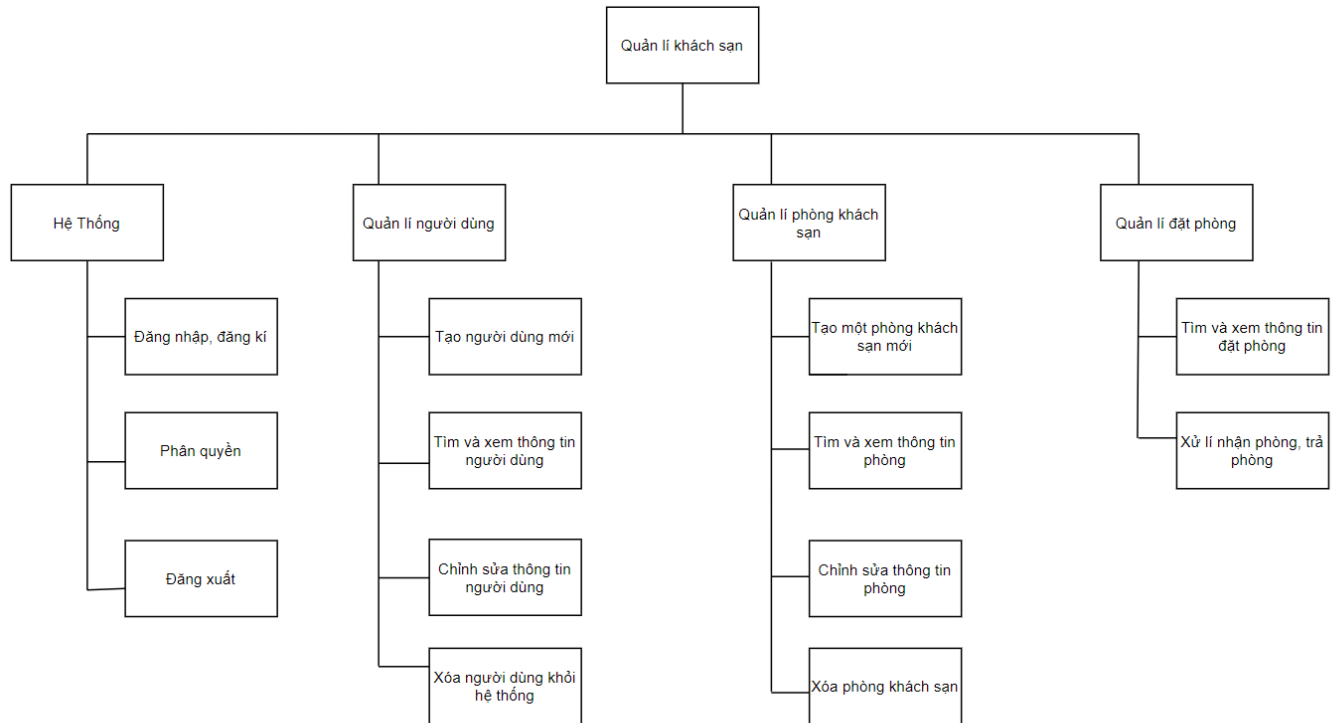
Bảng phân mô tả tác nhân chính và nghiệp vụ của các tác nhân trong hệ thống Website quản lí khách sạn

Tác nhân	Nghiệp vụ
Admin	Có quyền quản lý tài khoản người dùng Có quyền quản lý Phòng khách sạn Có quyền quản lý lịch đặt, trả phòng Có quyền quản lý doanh thu Có quyền đăng nhập, đăng xuất
User	Có quyền xem phòng khách sạn Có quyền quản lý thông tin cá nhân Có quyền xem các thông tin lịch đặt cá nhân Có quyền đặt phòng khách sạn Có quyền đăng ký, đăng nhập, đăng xuất

Bảng 2.1 Bảng mô tả tác nhân và nghiệp vụ

2.1.2. Sơ đồ phân rã chức năng

Sơ đồ phân rã chức năng dưới đây giúp chia nhỏ bài toán phức tạp, thiết kế và xây dựng phần mềm một cách có hệ thống, quản lý dự án tốt hơn, tái sử dụng các chức năng, và đảm bảo chất lượng

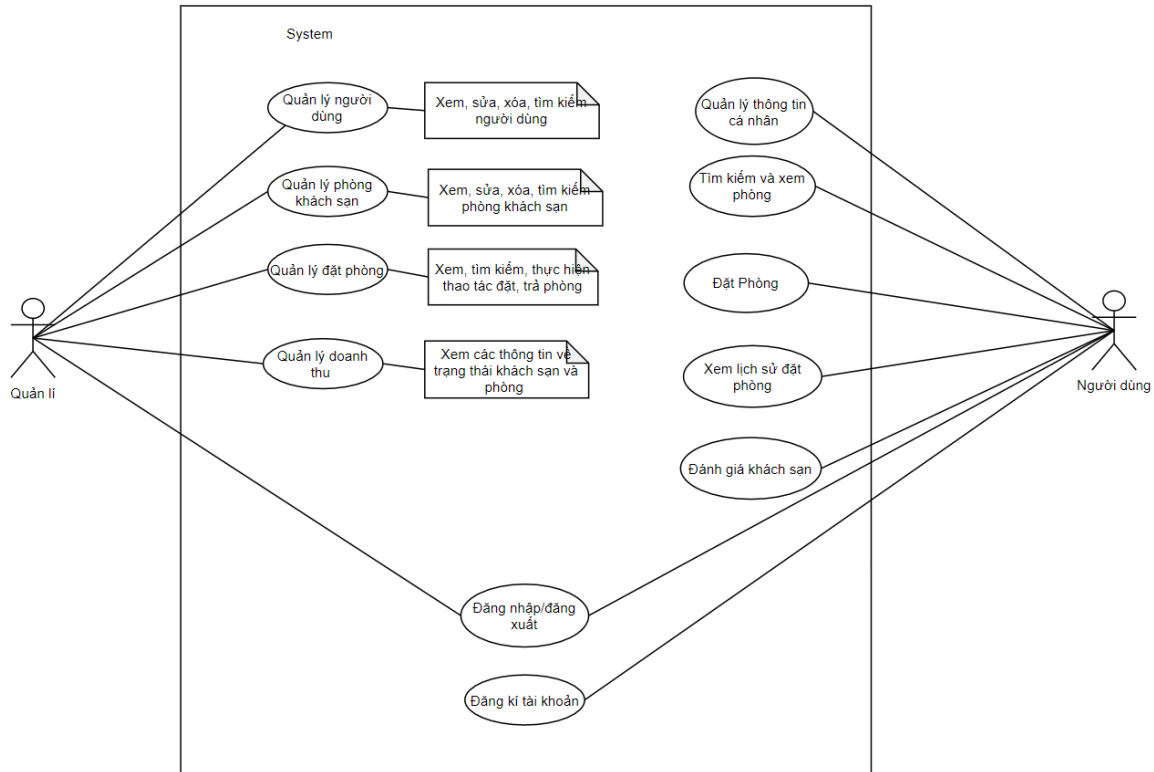


Hình 2.1 Sơ đồ phân rã chức năng

2.2 Thiết kế hệ thống

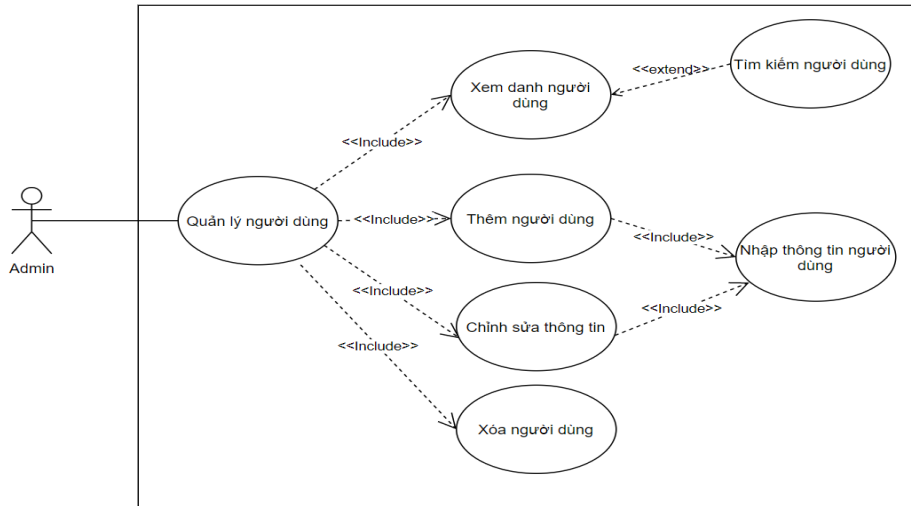
2.2.1. Sơ đồ use case tổng quát của hệ thống

Dưới đây là sơ đồ use case tổng quát mô tả các chức năng chính mà các tác nhân có thể sử dụng trong hệ thống



Hình 2.2 Sơ đồ Use Case tổng quát

Đây là sơ đồ use case cho chức năng quản lí người dùng



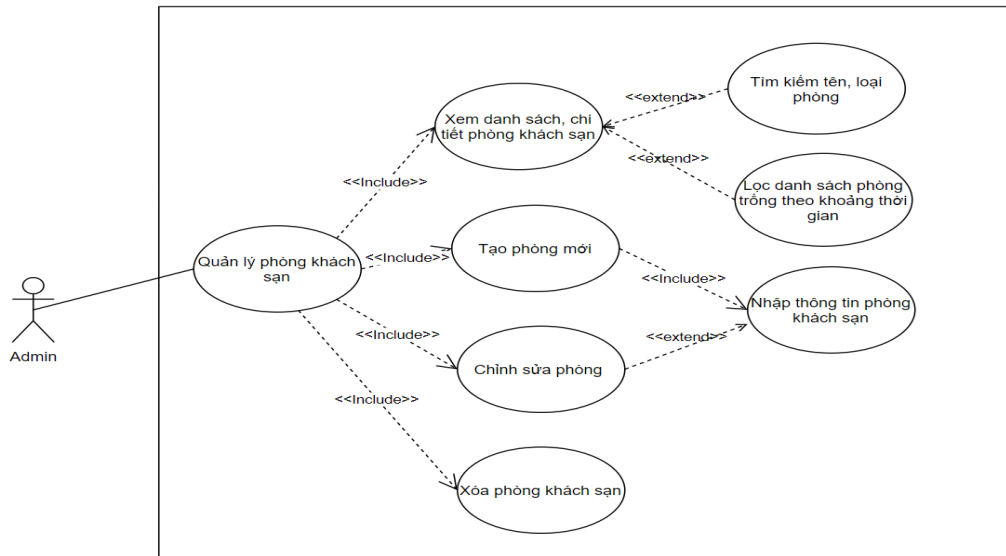
Hình 2.3 Sơ đồ Use Case chức năng quản lí người dùng

Bảng dưới đây đặc tả ca sử dụng chức năng quản lí người dùng

Tên ca sử dụng	Quản lý người dùng	
Tác nhân	ADMIN	
Mô tả	Cho phép ADMIN quản lý người dùng	
Điều kiện kích hoạt	ADMIN phải đăng nhập thành công	
Kịch bản	Hành động đầu vào	Hệ thống phản hồi
	Người dùng nhấn vào nút “User” bên thanh menu trái	Hệ thống hiển thị màn hình quản lý người dùng
	Người dùng nhấn vào nút “Add User”	Hệ thống hiển thị dialog tạo tài khoản người dùng
	Người dùng nhấn vào nút ở thanh action của cuối thông tin người dùng, bấm vào “Update”	Hệ thống hiển thị form chỉnh sửa thông tin người dùng, nhấn “Update ” để cập nhật thông tin
	Người dùng nhấn vào nút ở thanh action của cuối thông tin người dùng, bấm vào “Delete”	Hệ thống hiển thị dialog xác nhận xóa tài khoản người dùng, nhấn “OK ” để xác nhận xóa

Bảng 2.2 Bảng đặc tả Use Case chức năng quản lí người dùng

Sơ đồ usecase quản lí thông tin phòng khách sạn



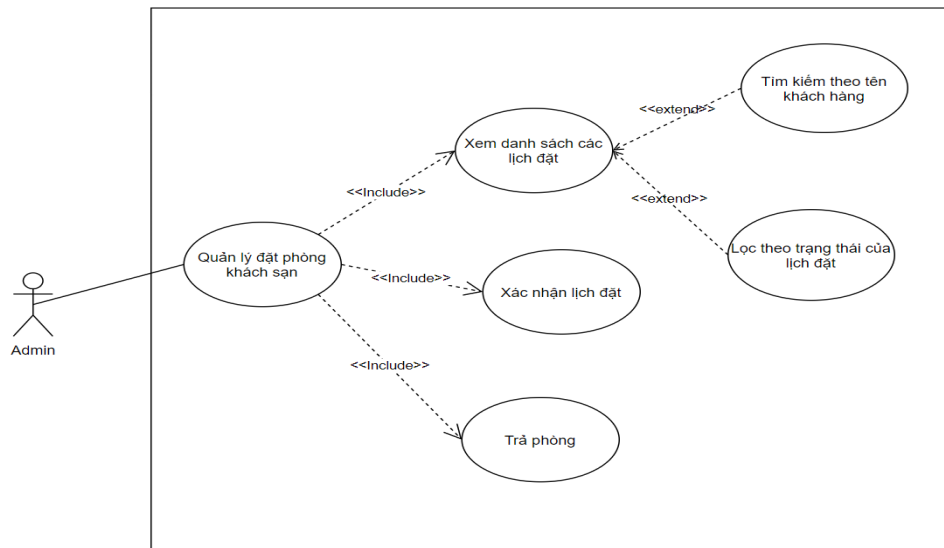
Hình 2.4 Sơ đồ Use Case chức năng quản lí thông tin phòng khách sạn

Bảng dưới đây đặc tả ca sử dụng chức năng quản lí phòng khách sạn của ADMIN

Tên ca sử dụng	Quản lý phòng	
Tác nhân	ADMIN	
Mô tả	Cho phép ADMIN quản lý phòng	
Điều kiện kích hoạt	ADMIN phải đăng nhập thành công	
Kịch bản	Hành động đầu vào	Hệ thống phản hồi
	Người dùng nhấn vào nút “Room” bên thanh menu trái	Hệ thống hiển thị màn hình quản lý phòng khách sạn
	Người sử dụng các thanh filter để lọc phòng theo các điều kiện	Hệ thống hiển thị danh sách các phòng theo điều kiện
	Người dùng nhấn vào nút “Add Room”	Hệ thống hiển thị trang tạo phòng khách sạn
	Người dùng nhấn vào phòng bất kì	Hệ thống hiển thị trang chỉnh sửa thông tin phòng, nhấn Update để cập nhật

Bảng 2.3 Bảng đặc tả Use Case chức năng quản lí phòng

Sơ đồ use case quản lí đặt phòng khách sạn



Hình 2.5 Sơ đồ Use Case chức năng quản lí đặt phòng khách sạn

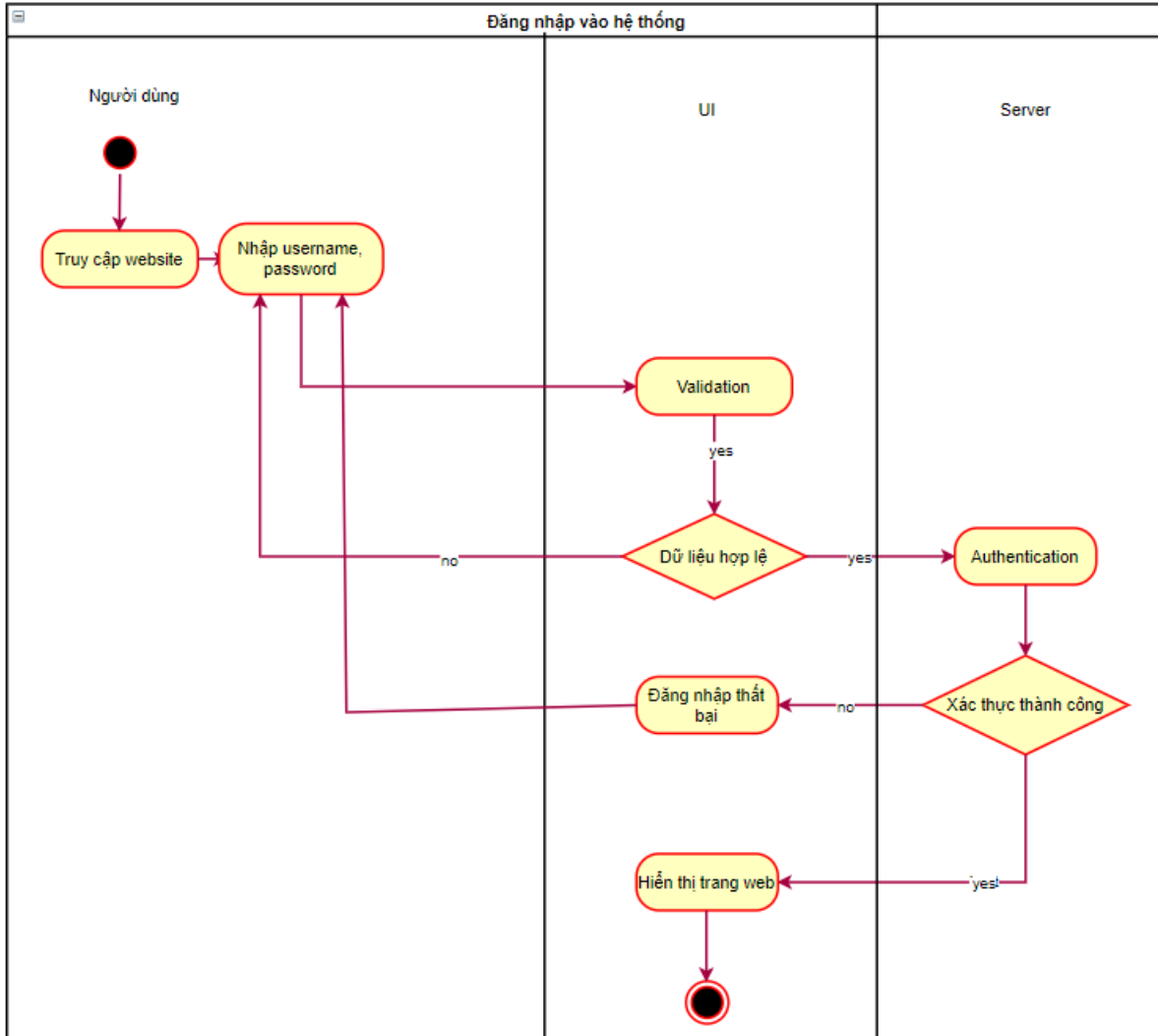
Bảng dưới đây đặc tả ca sử dụng chức năng quản lí đặt phòng khách sạn của ADMIN

Tên ca sử dụng	Quản lý đặt phòng	
Tác nhân	ADMIN	
Mô tả	Cho phép ADMIN quản lý phòng	
Điều kiện kích hoạt	ADMIN phải đăng nhập thành công	
Kịch bản	Hành động đầu vào	Hệ thống phản hồi
	Người dùng nhấn vào nút “Booking” bên thanh menu trái	Hệ thống hiển thị màn hình quản lí lịch đặt phòng khách sạn
	Người sử dụng thanh tìm kiếm để tìm các lịch đặt	Hệ thống hiển thị danh sách các lịch đặt theo điều kiện
	Người dùng nhấn vào nút “Confirm”	Hệ thống cập nhật trạng thái cho lịch đặt đó
	Người dùng nhấn vào nút “Return”	Hệ thống cập nhật trạng thái cho lịch đặt đó

Bảng 2.4 Bảng đặc tả Use Case chức năng quản lí đặt phòng

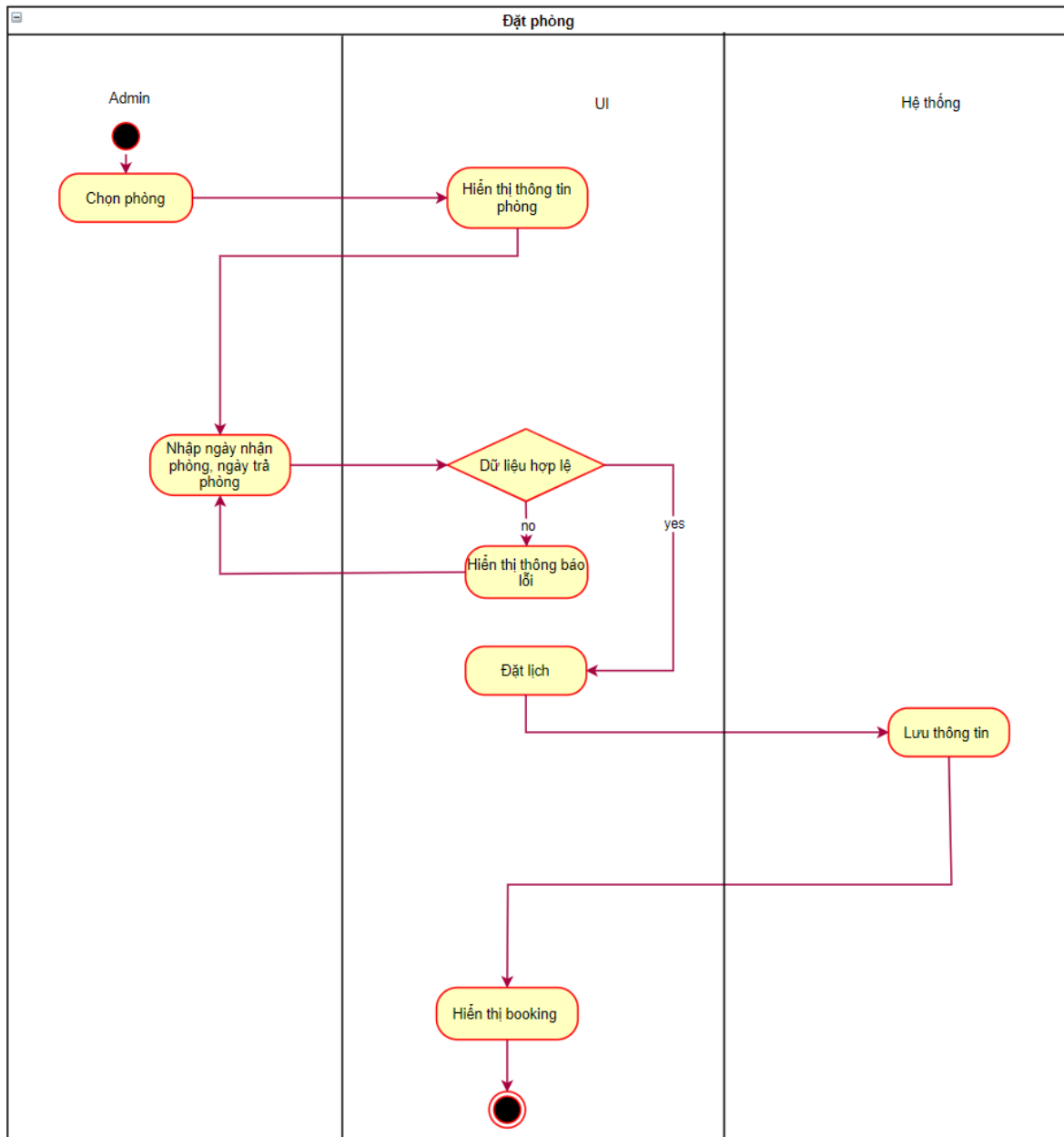
2.2.2. Sơ đồ hoạt động

Dưới đây là sơ đồ hoạt động của chức năng đăng nhập của người dùng



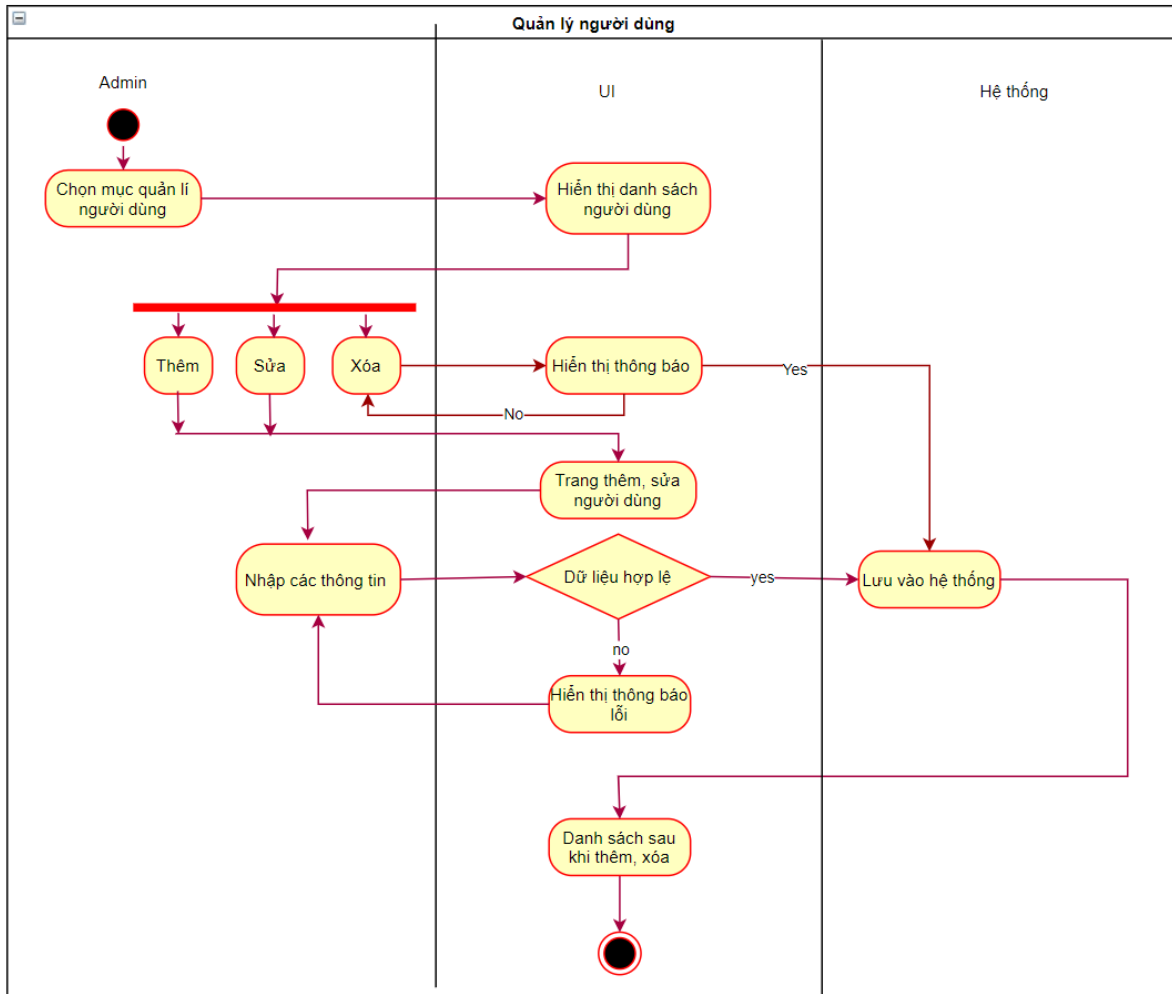
Hình 2.6 Sơ đồ hoạt động chức năng đăng nhập

Dưới đây là sơ đồ hoạt động của chức năng đặt phòng của người dùng



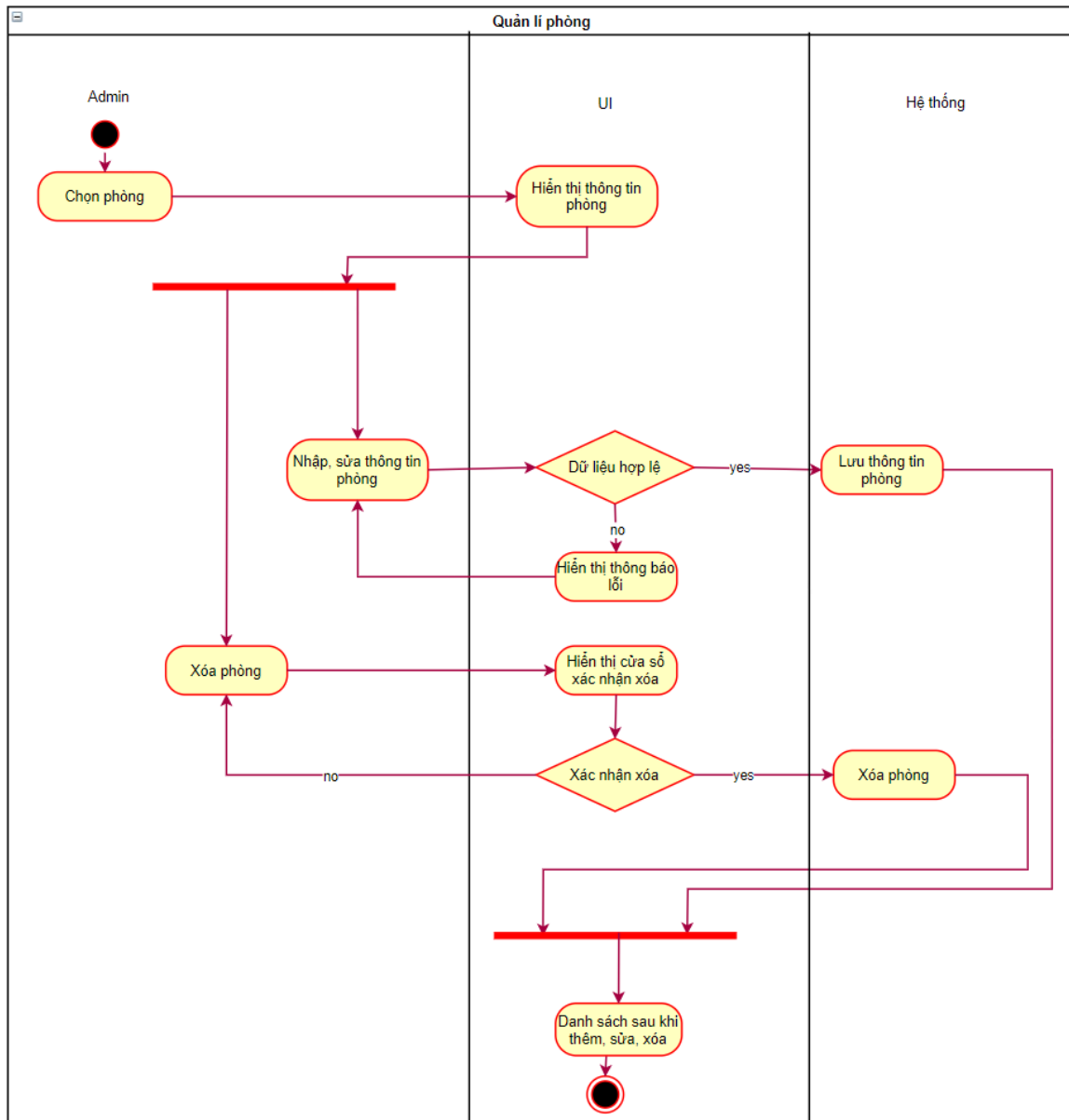
Hình 2.7 Sơ đồ hoạt động của chức năng đặt phòng của người dùng

Dưới đây là sơ đồ hoạt động của chức năng quản lí người dùng



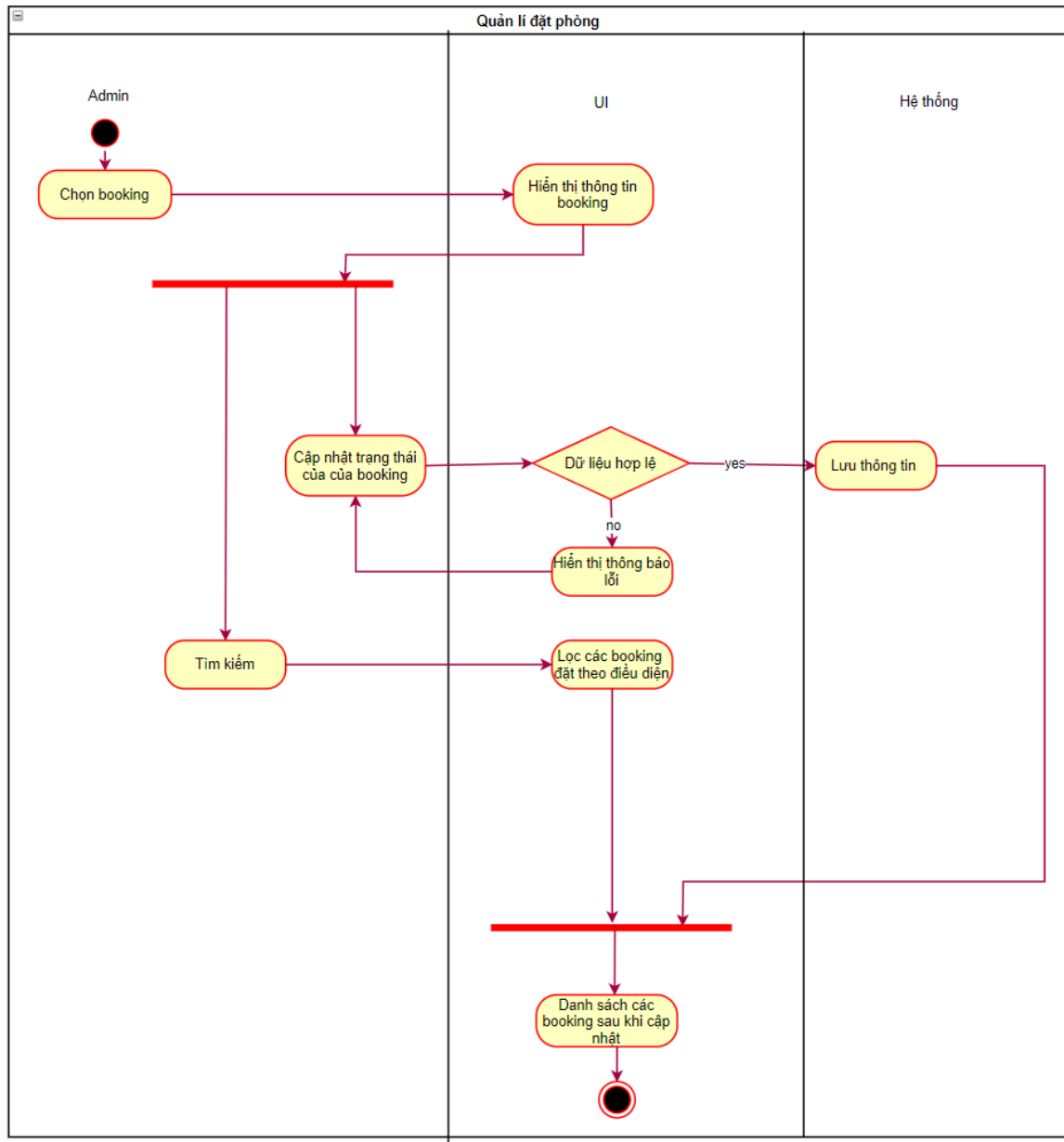
Hình 2.8 Sơ đồ hoạt động chức năng quản lí người dùng

Dưới đây là sơ đồ hoạt động của chức năng quản lí phòng khách sạn



Hình 2.9 Sơ đồ hoạt động chức năng quản lí phòng

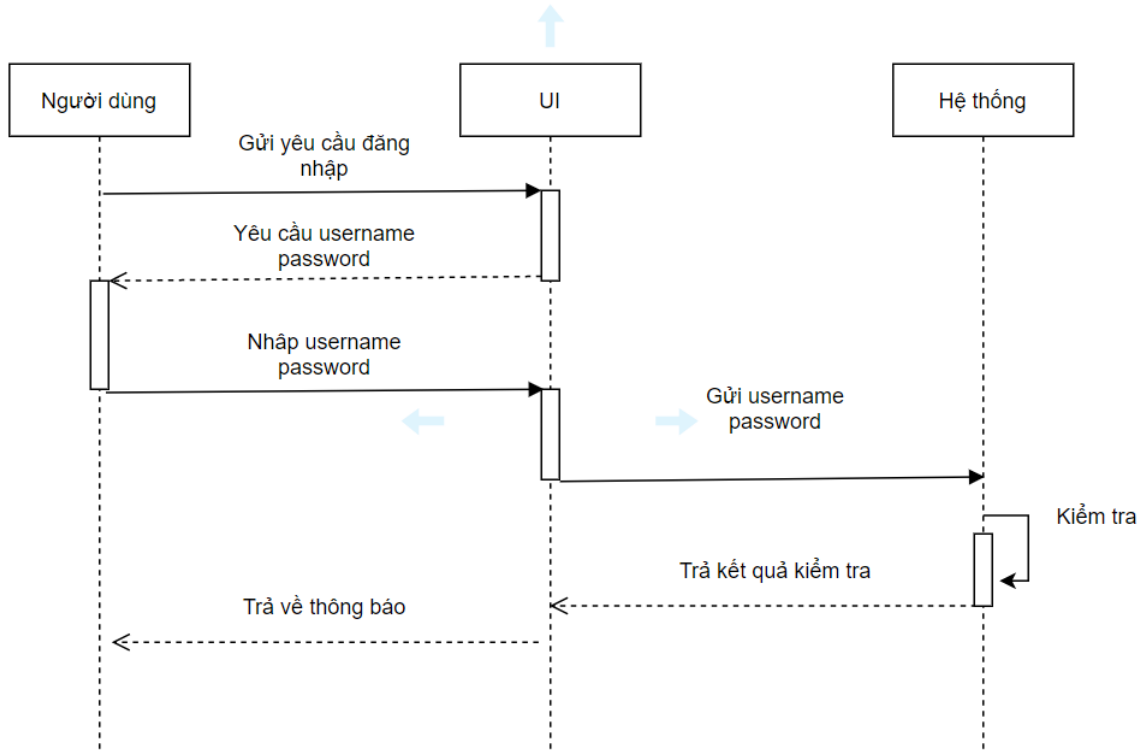
Dưới đây là sơ đồ hoạt động của chức năng đặt phòng của người quản lí



Hình 2.10 Sơ đồ hoạt động chức năng quản lí đặt phòng

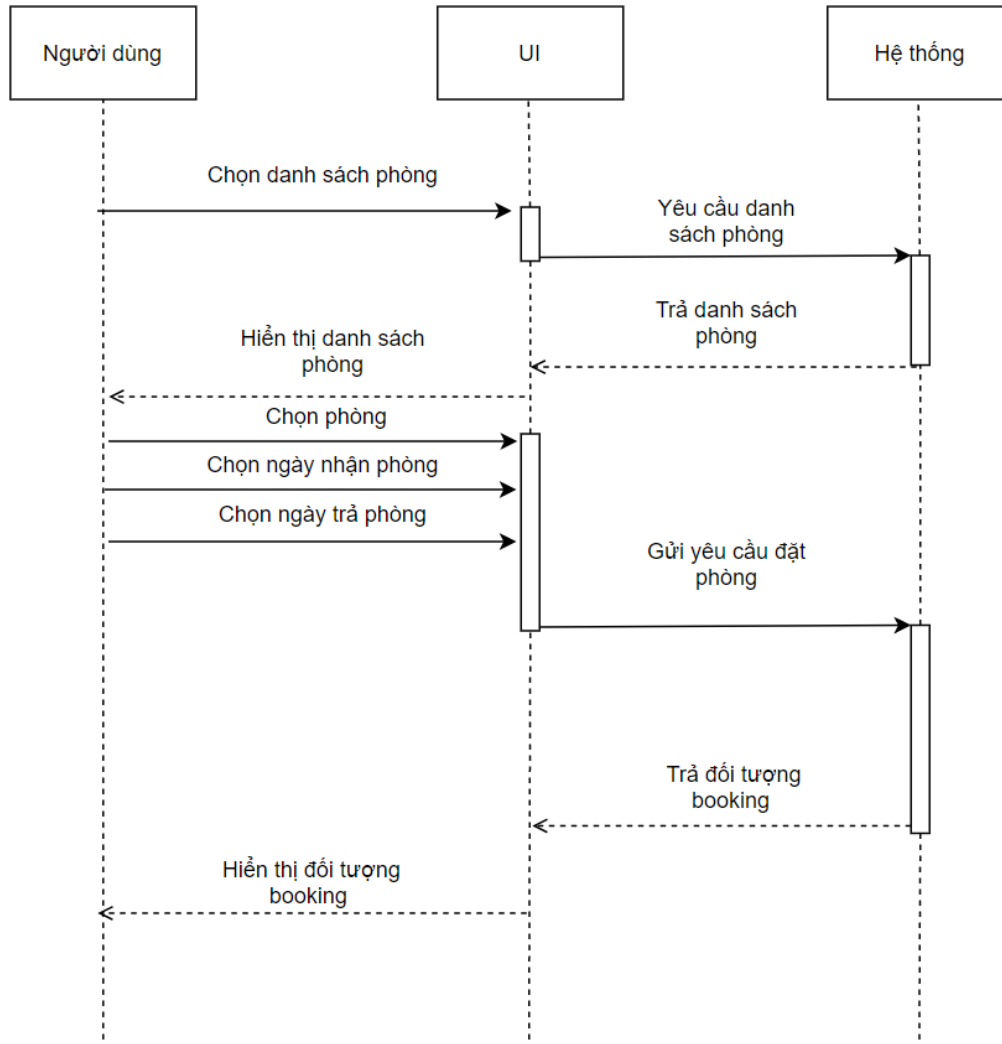
2.2.2. Sơ đồ tuần tự

Dưới đây là sơ đồ tuần tự đăng nhập mô tả các bước từ khi người dùng nhập thông tin đăng nhập cho đến khi hệ thống xác thực thông tin đăng nhập



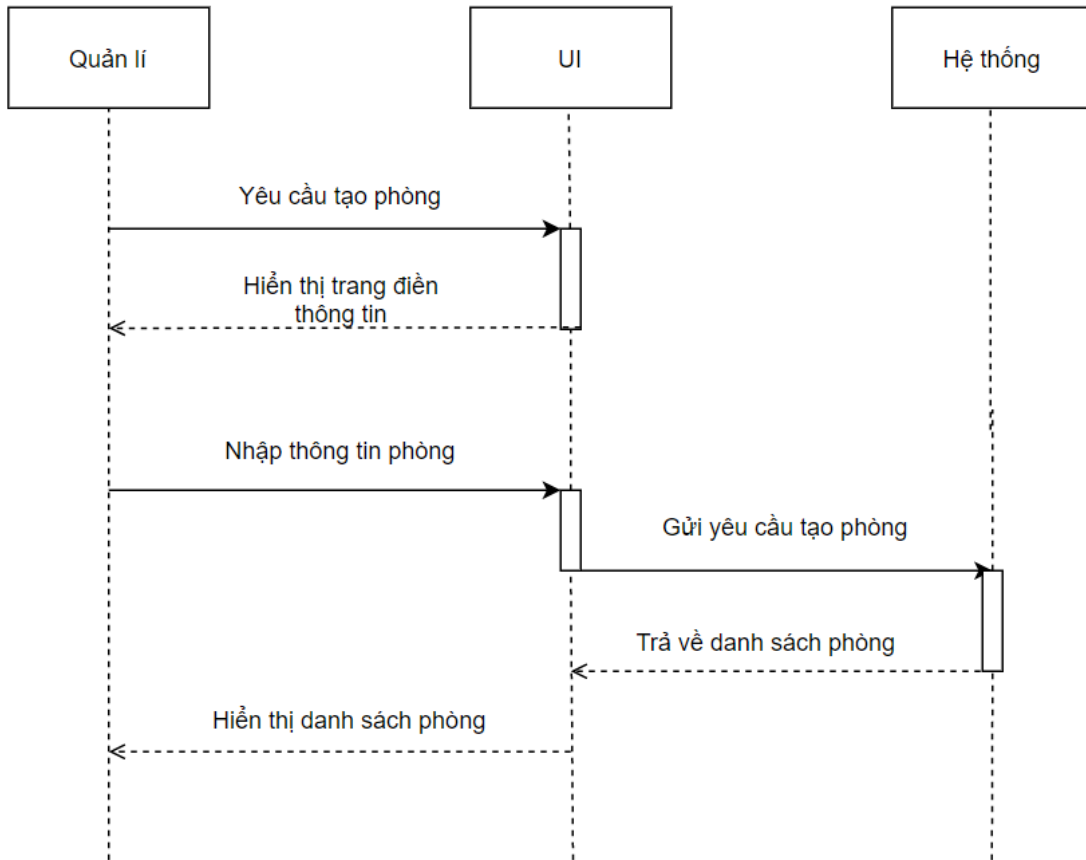
Hình 2.11 Sơ đồ tuần tự chức năng đăng nhập

Dưới đây là sơ đồ tuần tự chức năng đặt phòng, nó mô tả các bước thực hiện của khách hàng khi thực hiện đặt phòng khách sạn



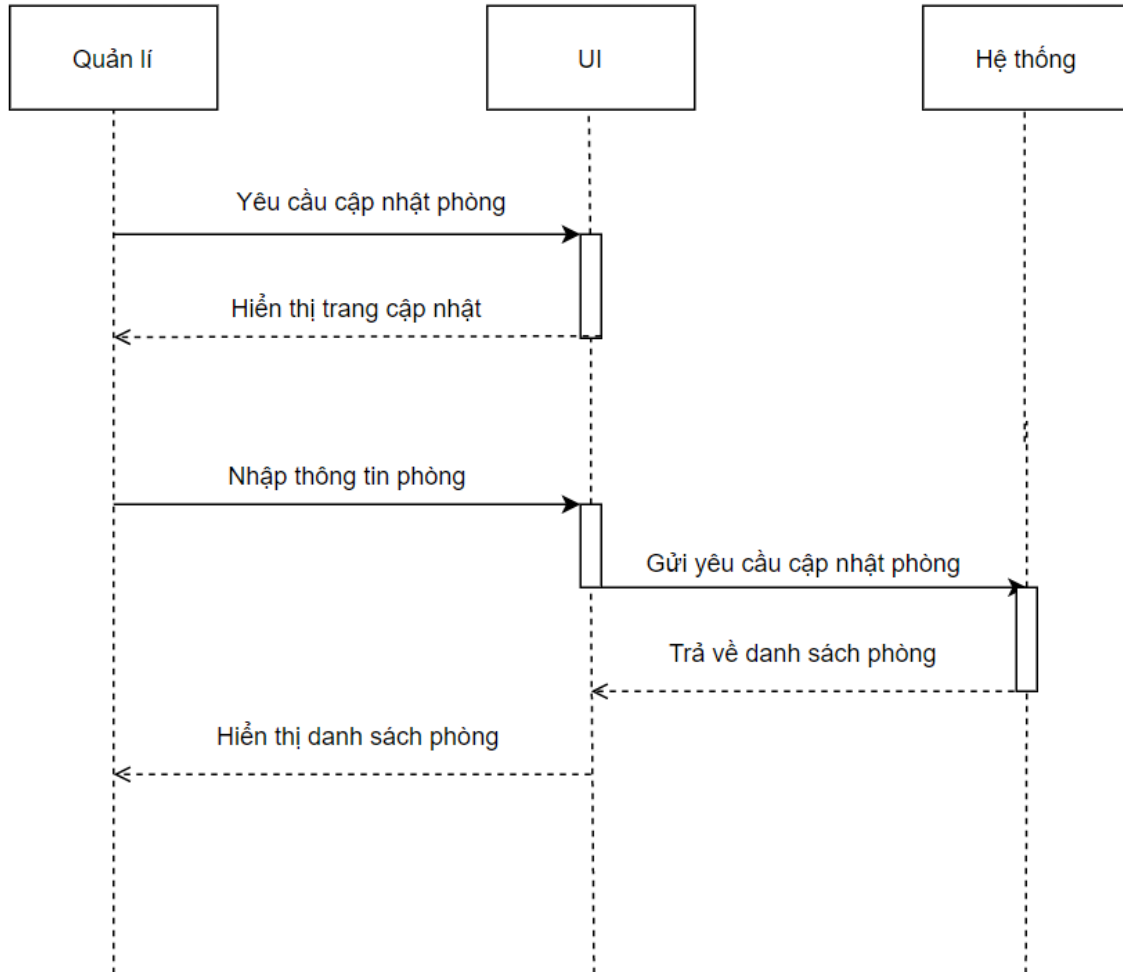
Hình 2.12 Sơ đồ tuần tự chức năng đặt phòng

Dưới đây là sơ đồ tuần tự chức năng tạo phòng mô tả tuần tự các bước người quản lý tạo phòng khách sạn



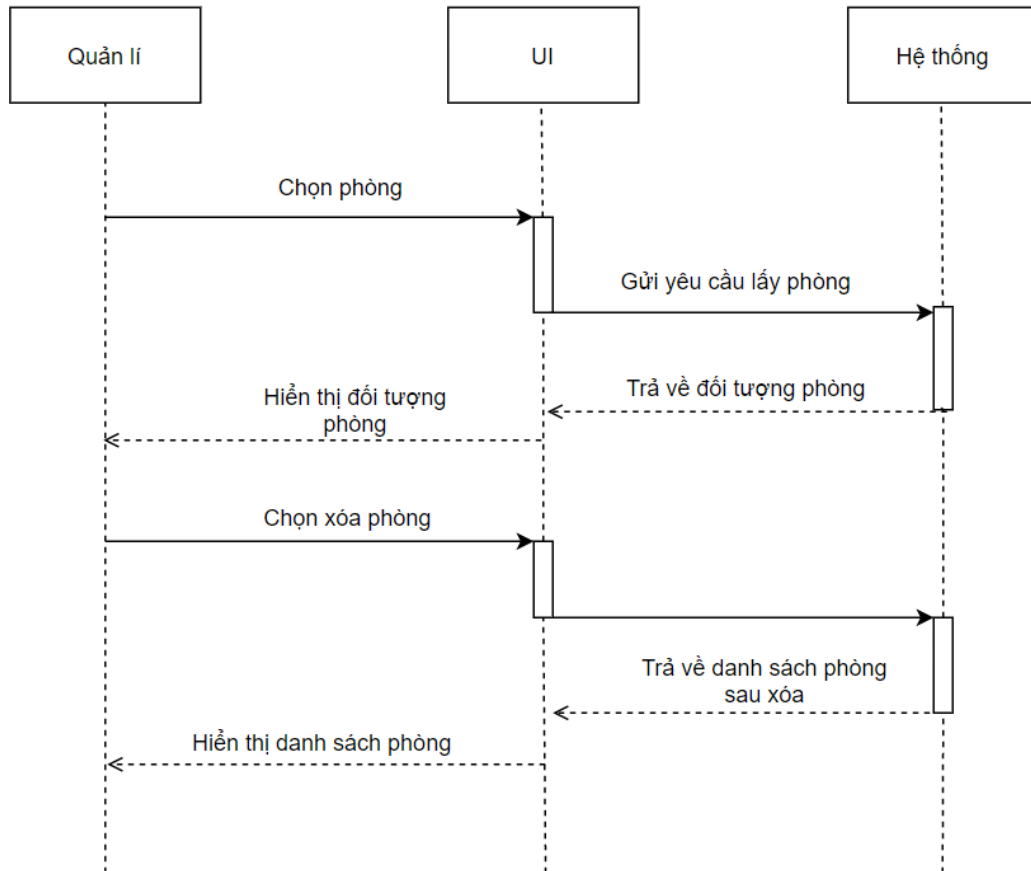
Hình 2.13 Sơ đồ tuần tự chức năng tạo phòng

Dưới đây là sơ đồ tuần tự chức năng cập nhật phòng mô tả tuần tự các bước người quản lí sửa thông tin của phòng khách sạn



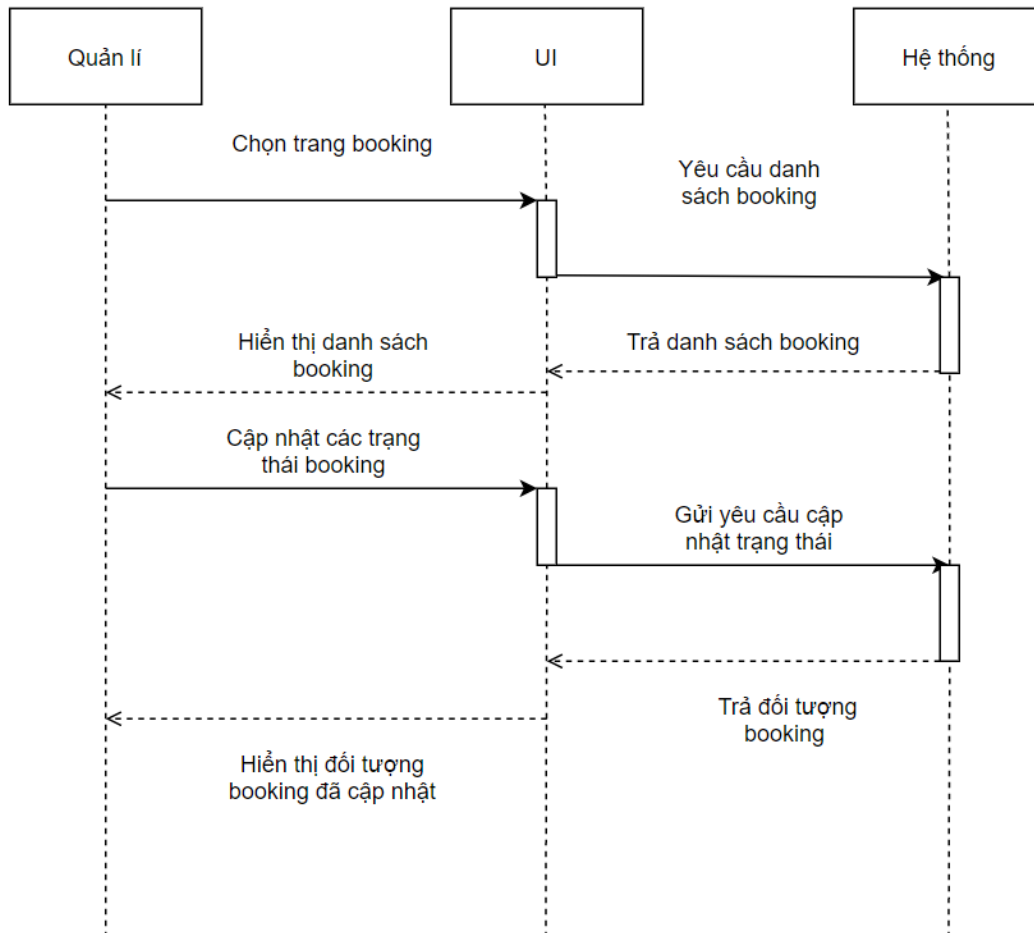
Hình 2.14 Sơ đồ tuần tự chức năng cập nhật phòng

Dưới đây là sơ đồ tuần tự chức năng tạo phòng mô tả tuần tự các bước người quản lý thực hiện xóa phòng khách sạn



Hình 2.15 Sơ đồ tuần tự chức năng xóa phòng

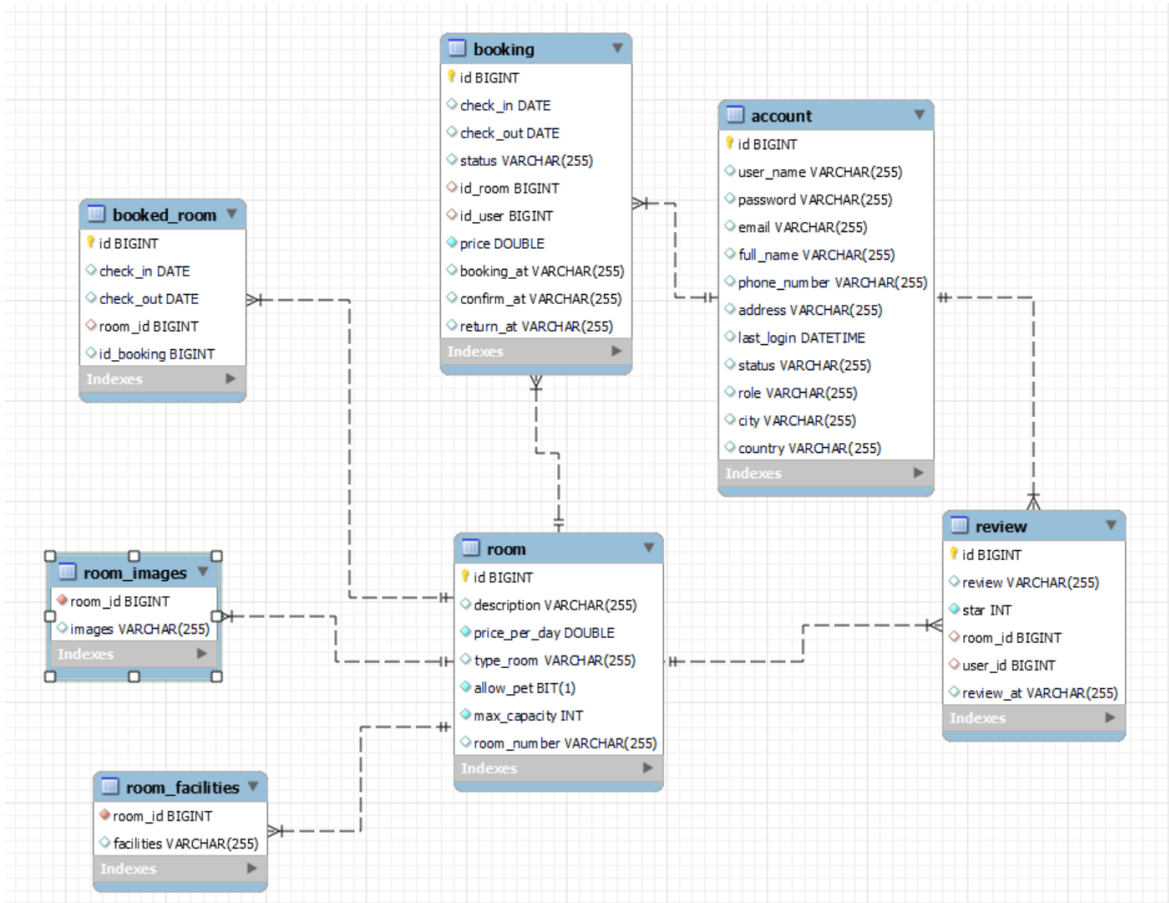
Dưới đây là sơ đồ tuần tự chức năng tạo phòng mô tả tuần tự các bước người quản lý thực hiện xác nhận trạng thái của các lịch đặt phòng (booking)



Hình 2.16 Sơ đồ tuần tự chức năng quản lý đặt phòng

2.2.3. Cơ sở dữ liệu của hệ thống

Dưới đây là phần sơ đồ cơ sở dữ liệu của hệ thống quản lý khách sạn



Hình 2.17 Cơ sở dữ liệu

Dưới đây là bảng mô tả bảng “account” trong Database

Tên trường	Kiểu dữ liệu	Ràng buộc	Diễn giải
id	Long	Khóa chính	Khóa chính
email	String	Unique	Email người dùng
fullName	String		Tên đầy đủ của người dùng
userName	String		Tài khoản đăng nhập của người dùng
password	String		Mật khẩu tài khoản người dùng
phoneNumber	String	Unique	Số điện thoại người dùng
role	String		Quyền người dùng
address	String		Địa chỉ người dùng người dùng
lastLogin	Date		Thời gian người dùng sử dụng cuối cùng

Bảng 2.5 Bảng account

Dưới đây là bảng mô tả bảng “booking” trong Database

Tên trường	Kiểu dữ liệu	Ràng buộc	Diễn giải
id	Long	Khóa chính	Khóa chính
idRoom	Long	Khóa ngoại	Id phòng được tham chiếu
idUser	Long	Khóa ngoại	Id người dùng được tham chiếu
checkIn	Date		Ngày nhận phòng
checkOut	Date		Ngày trả phòng
price	double		Giá phòng
status	String		Trạng thái booking
bookingAt	String		Thời gian đặt phòng
confirmAt	String		Thời gian được xác nhận
returnAt	String		Thời gian trả phòng

Bảng 2.6 Bảng booking

Dưới đây là bảng mô tả bảng “review” trong Database

Tên trường	Kiểu dữ liệu	Ràng buộc	Diễn giải
id	Long	Khóa chính	Khóa chính
idRoom	Long	Khóa ngoại	Id phòng được tham chiếu
idUser	Long	Khóa ngoại	Id người dùng được tham chiếu
star	int		Số sao được đánh giá
reviewAt	String		Thời gian đặt đánh giá

Bảng 2.7 Bảng review

Dưới đây là bảng mô tả bảng “bookedRoom” trong Database

Tên trường	Kiểu dữ liệu	Ràng buộc	Diễn giải
id	Long	Khóa chính	Khóa chính
idRoom	Long	Khóa ngoại	Id phòng được tham chiếu
idBooking	Long	Khóa ngoại	Id lịch đặt được tham chiếu
checkIn	String		Ngày nhận phòng
checkOut	String		Ngày trả phòng

Bảng 2.8 Bảng booked_room

Dưới đây là bảng mô tả bảng “room” trong Database

Tên trường	Kiểu dữ liệu	Ràng buộc	Diễn giải
id	Long	Khóa chính	Khóa chính
idImage	Long	Khóa ngoại	Id ảnh được tham chiếu
idFacilities	Long	Khóa ngoại	Id tiện ích được tham chiếu
typeRoom	String		Loại phòng
roomNumber	String		Số phòng
description	String		Mô tả chi tiết
maxCapacity	int		Sức chứa tối đa
allowPet	boolean		Cho phép thú cưng
pricePerDay	double		Giá trên 1 ngày

Bảng 2.9 Bảng room

Dưới đây là bảng mô tả bảng “images” trong Database

Tên trường	Kiểu dữ liệu	Ràng buộc	Diễn giải
id	Long	Khóa chính	Khóa chính
images	String		Tên ảnh được lưu trong máy

Bảng 2.10 Bảng images

Dưới đây là bảng mô tả bảng “facilities” trong Database

Tên trường	Kiểu dữ liệu	Ràng buộc	Diễn giải
id	Long	Khóa chính	Khóa chính
facilities	String		Các tiện ích của phòng khách sạn

Bảng 2.11 Bảng facilities

2.3 Tổng kết

Chương này trình bày về các tác nhân tham gia vào hệ thống, biểu đồ sơ đồ ca sử dụng trực quan hóa các chức năng của các tác nhân, sơ đồ hoạt động mô tả luồng đi các chức năng chính của hệ thống, cho ta cái nhìn tổng quát về cả hệ thống

CHƯƠNG 3: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

3.1 Phát triển ứng dụng

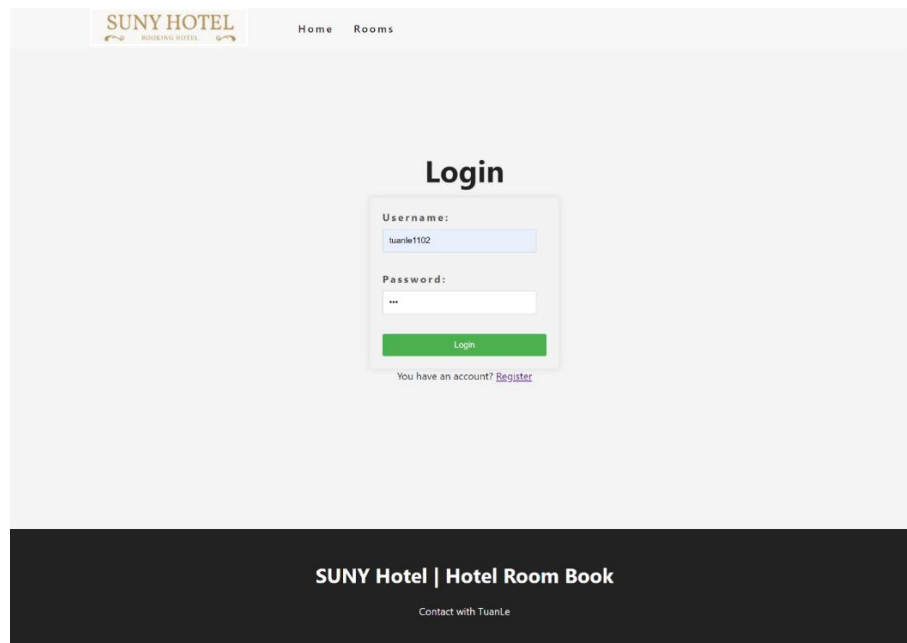
- Môi trường: Hệ điều hành Windows 11, cơ sở dữ liệu MySQL;
- Công cụ phát triển: Visual Studio Code, Postman, InterliJ;
- Cài đặt và cấu hình:
 - + Back-end: Java Spring Boot , MySQL,.
 - + Front-end: React Js,..
- Kiểm thử và sửa lỗi: Tạo và thực hiện các bài kiểm tra để đảm bảo tính đúng đắn của các chức năng, sửa lỗi và điều chỉnh nếu cần thiết.

3.2 Kết quả triển khai

Dưới đây là phần kết quả của quá trình triển khai website quản lý khách sạn của hai quyền là người dùng và quản lý

3.2.1 Giao diện bên phía người dùng

Màn hình trang đăng nhập vào hệ thống, người dùng nhập đúng user name và password để được tiếp tục truy cập vào hệ thống



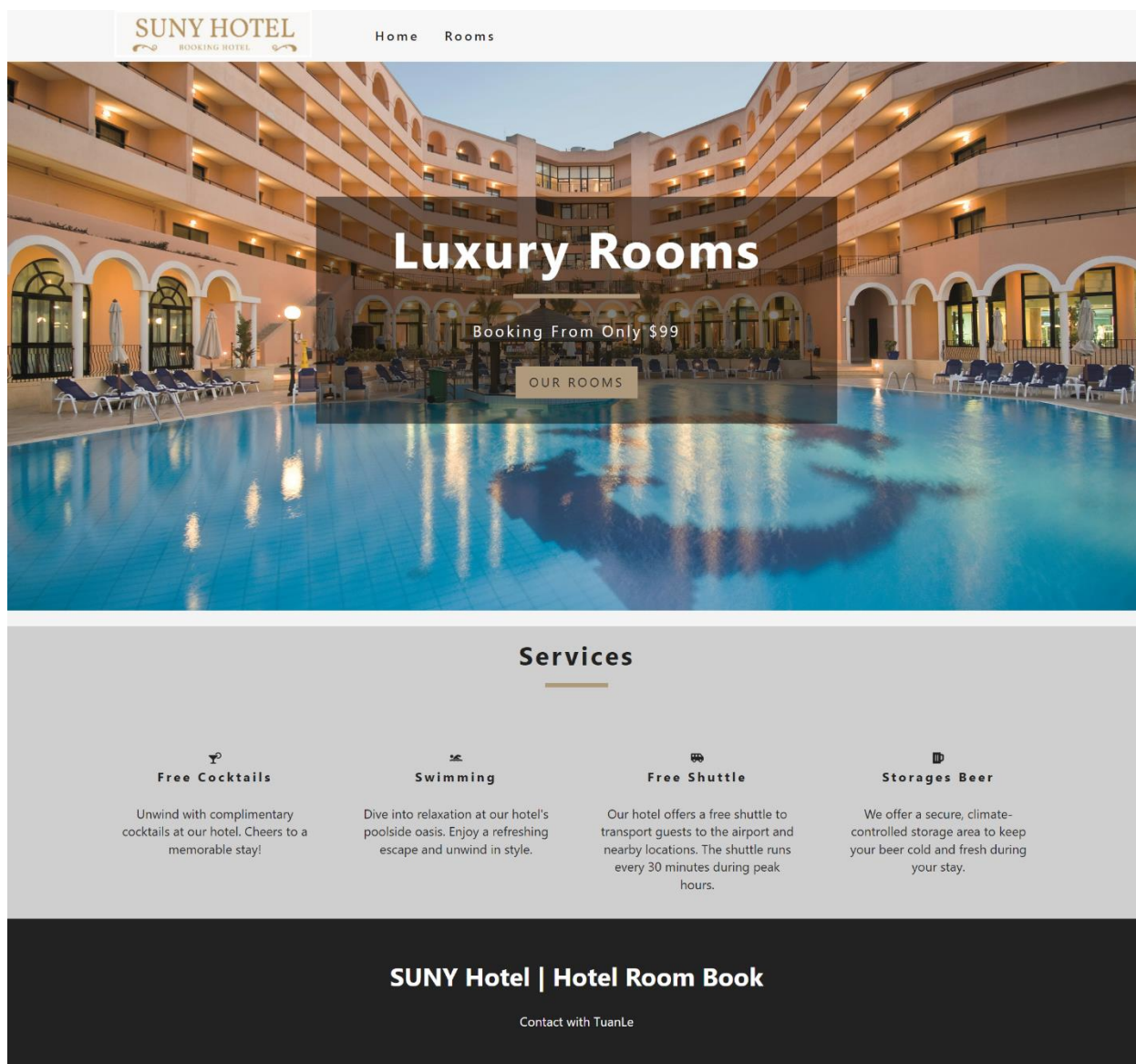
Hình 3.1 Giao diện đăng nhập

Nếu chưa có tài khoản người dùng có thể đăng kí một tài khoản mới để sử dụng

The image shows a web interface for 'SUNY HOTEL'. At the top, there is a header with the hotel's logo and two navigation links: 'Home' and 'Rooms'. Below the header, the main section is titled 'Register'. It contains a registration form with the following fields: 'Username' (with the value 'thanhphu123'), 'Password' (masked with dots), 'Confirm Password' (masked with dots), and 'Email' (with the value 'thanhphu123@gmail.com'). A green button labeled 'Đăng ký' is positioned below the email field. The bottom of the page features a dark footer with the text 'SUNY Hotel | Hotel Room Book' and a smaller line 'Contact with TuanLe'.

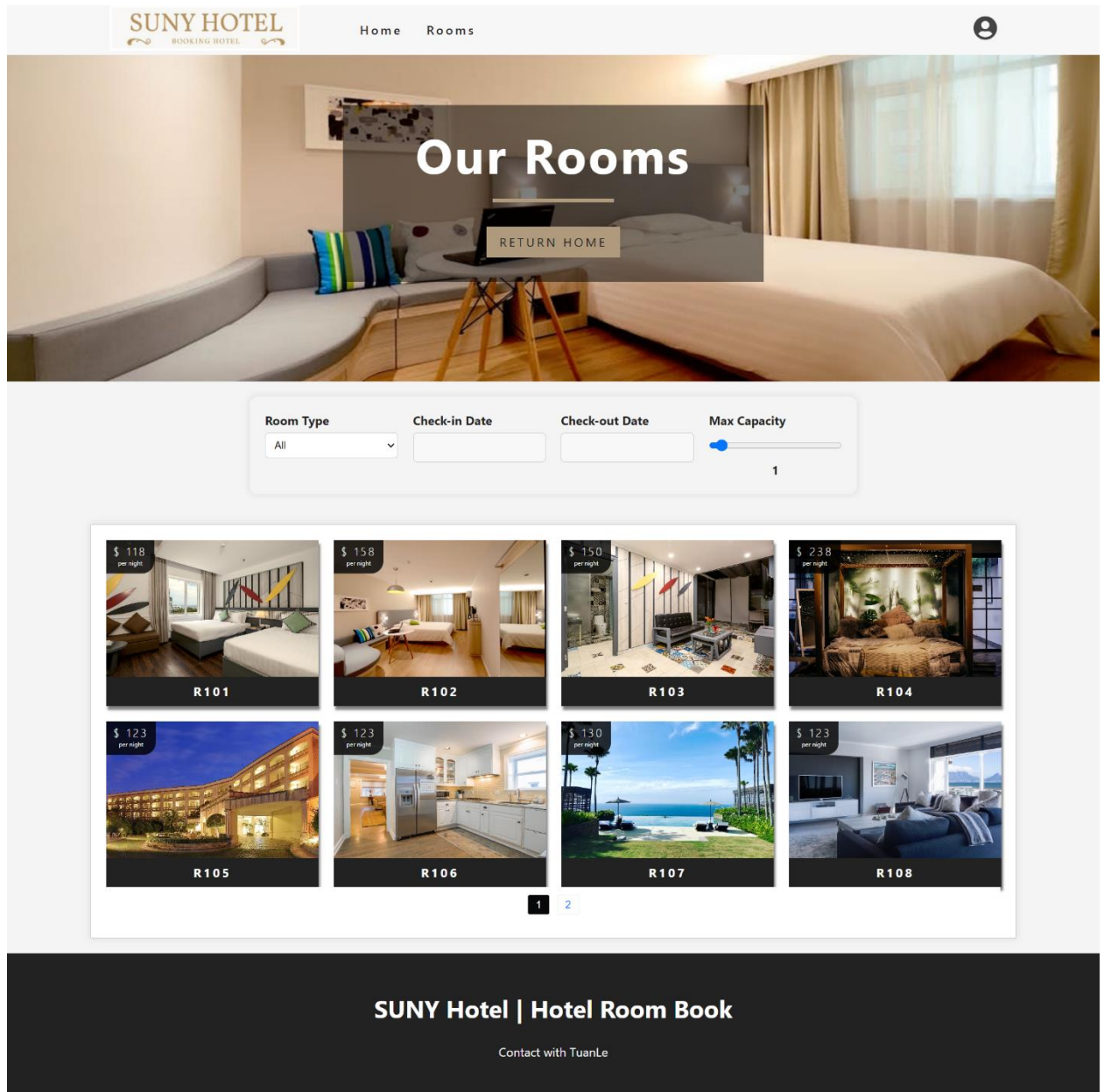
Hình 3.2 Giao diện đăng kí

Đây là giao diện trang chủ của khách sạn, có thể click vào “OurRoom” để xem các phòng của khách sạn này



Hình 3.3 Giao diện trang chủ của khách sạn

Giao diện khi nhấn vào để xem danh sách phòng, ở đây có thể lựa chọn được các phòng phù hợp với nhu cầu dựa theo các tiêu chí như loại phòng, số lượng tối đa cho phép, có thể lọc phòng còn trống trong khoảng thời gian ta muốn đặt



Hình 3.4 Giao diện trang xem danh sách phòng khách sạn

Khi nhấn vào xem phòng, ta có thể đặt phòng đó bằng cách chọn ngày nhận phòng, trả phòng, nếu phòng còn trống trong khoảng thời gian đó phòng sẽ được thành công. Dưới đây là giao diện trang xem thông tin phòng và đặt phòng

SUNY HOTEL
BOOKING HOTEL

Home Rooms



Single Room

[BACK TO ROOMS](#)



Single Room

The room features a plush king-size bed draped in crisp, high-quality linens, ensuring a restful night's sleep lulled by the gentle sound of the waves.

Information room: R101

Max Capacity: 2 people
Allow Pets: Yes
Facilities:

- wifi
- parking
- pool
- spa

Price: \$118/day

Reviews:

Ok!!!
★★★★★
P Phạm Huỳnh Văn Phát
13:09:37, 13/06/2024

Good!!
★★★★★
N Nguyễn Thành Phú
02:50:32, 14/06/2024

Okkkk

Check-in Date
2024-06-27

Check-out Date
2024-06-29

Total Price: \$236

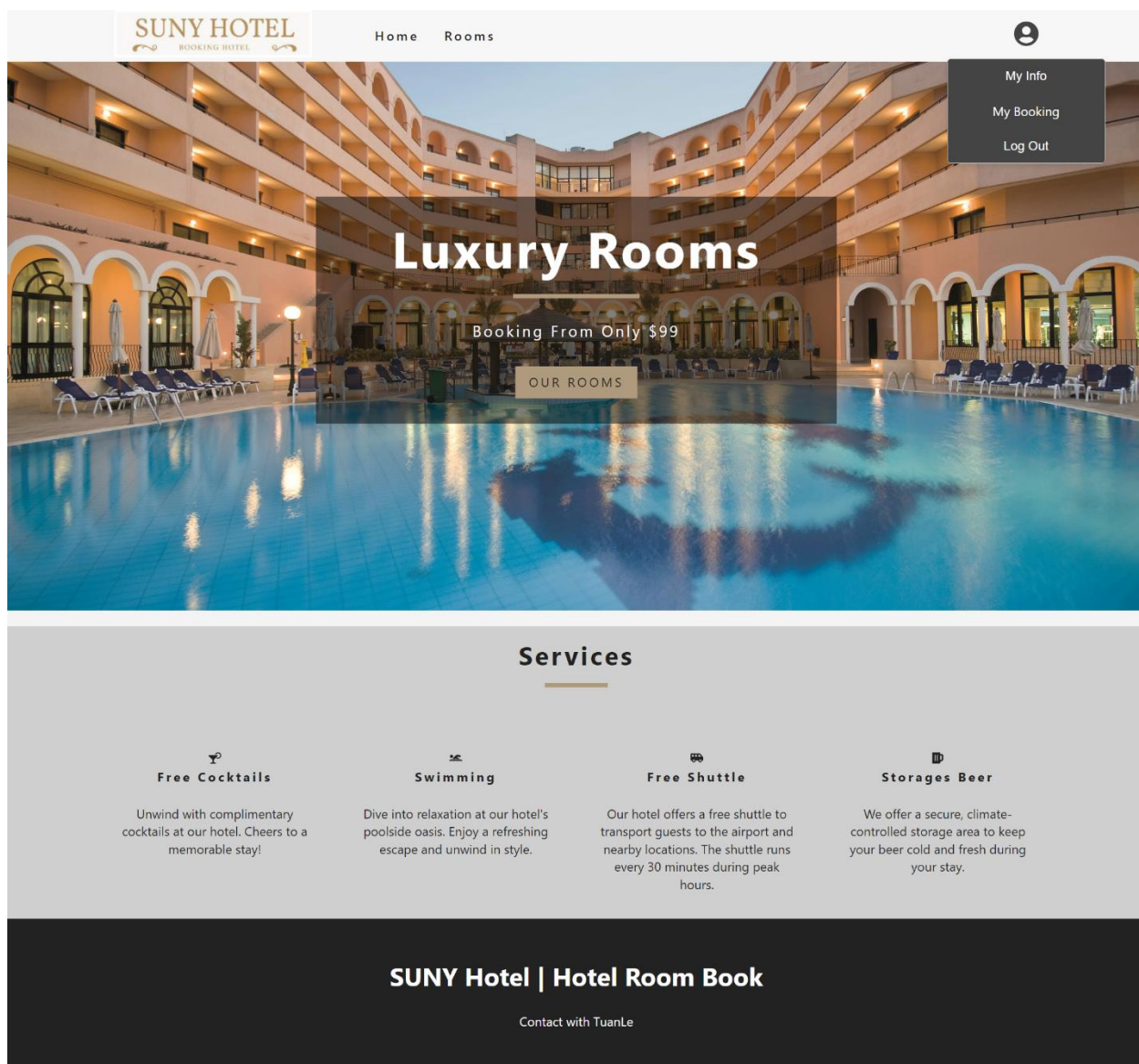
[Book Now](#)

SUNY Hotel | Hotel Room Book

Contact with TuanLe

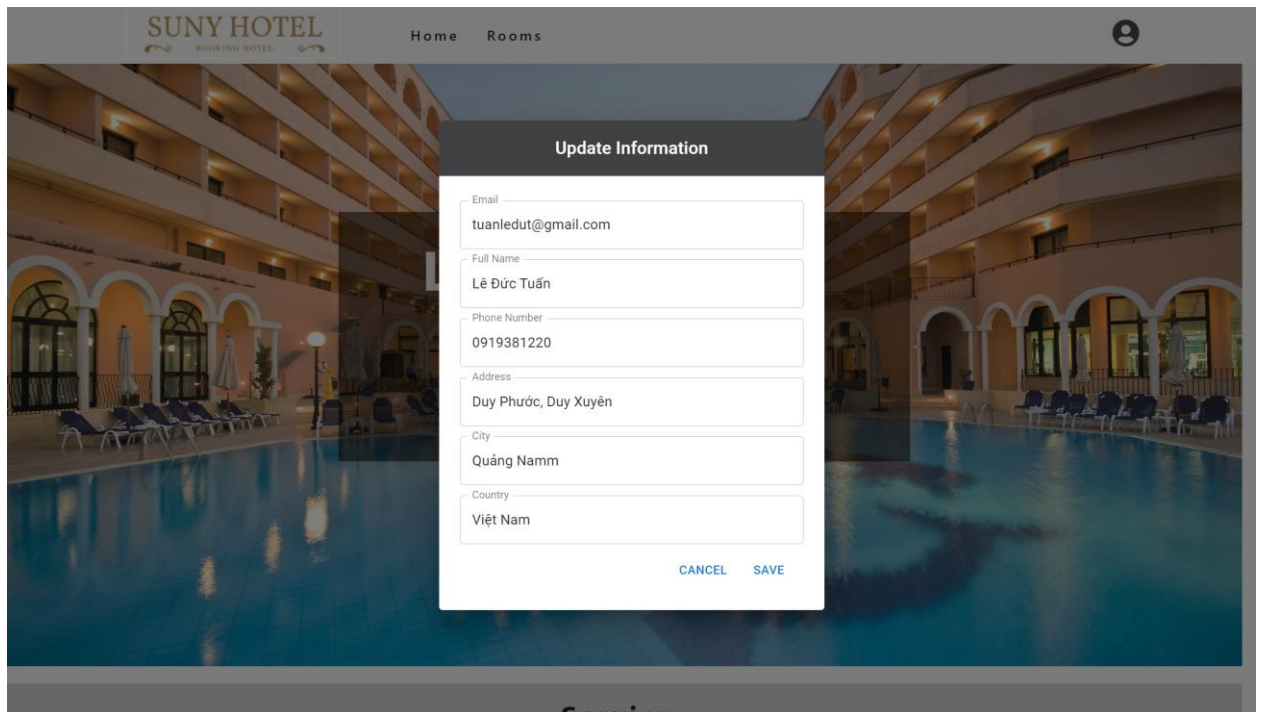
Hình 3.5 Giao diện trang thông tin phòng khách sạn và đặt phòng

Đây là phần menu của người dùng, ở đây có thể chọn để em thông tin cá nhân cũng như các lịch đặt phòng của cá nhân



Hình 3.6 Menu của người dùng để xem thông tin cá nhân

Dưới đây là phần dialog dùng để cập nhật thông tin của cá nhân, nhấn vào “My Info” trên menu người dùng để cập nhật thông tin người dùng



The screenshot displays the SUNY HOTEL website interface. At the top, the header includes the hotel's name, navigation links for 'Home' and 'Rooms', and a user profile icon. The main background image shows a large, modern hotel lobby with a swimming pool. Overlaid on this is a 'Update Information' dialog box. This dialog box contains several input fields: 'Email' (filled with 'tuanledut@gmail.com'), 'Full Name' (filled with 'Lê Đức Tuấn'), 'Phone Number' (filled with '0919381220'), 'Address' (filled with 'Duy Phước, Duy Xuyên'), 'City' (filled with 'Quảng Namm'), and 'Country' (filled with 'Việt Nam'). At the bottom right of the dialog box are two buttons: 'CANCEL' and 'SAVE'.

Hình 3.7 Dialog cập nhật thông tin cá nhân

Dưới đây là giao diện xem thông tin đặt phòng khi người dùng nhấn vào “My Booking” trên menu người dùng, ở đây chúng ta sẽ xem được thông tin lịch đặt của bảng thân cũng như trạng thái lịch đặt đó như thế nào, ở đây chúng ta có thể nhấn vào action “review” để thực hiện đánh giá phòng khách sạn



[Home](#) [Rooms](#)



My Booking

Search by room number

ID Booking	ID Room	Type Room	Room Number	Check-in Date	Check-out Date	Total Price	Booking At	Status	Action
57	8	Double Room	R108	2024-06-13	2024-06-16	369.0	01:39:51, 12/06/2024	Returned at 02:52:37, 13/06/2024	:
58	1	Single Room	R101	2024-06-13	2024-06-15	236.0	02:51:39, 13/06/2024	Returned at 02:52:38, 13/06/2024	 Review
71	2	Double Room	R102	2024-06-20	2024-06-21	158.0	00:00:40, 20/06/2024	Waiting confirm..	:

SUNY Hotel | Hotel Room Book

Contact with TuanLe

Hình 3.8 Giao diện trang xem danh sách đặt phòng

3.2.2 Giao diện bên phía người quản lý

Dưới đây là giao diện trang quản lý phòng khách sạn, ta có thể tìm kiếm hoặc lọc cách phòng theo các điều kiện như, loại phòng, tên phòng, hoặc những phòng còn trong trong khoảng thời gian nào đó

ADMIN

- Dashboard
- User
- Rooms
- Booking
- Logout

Room Manager

Check-in:

Check-out:

Room Type:

Search Room Number:

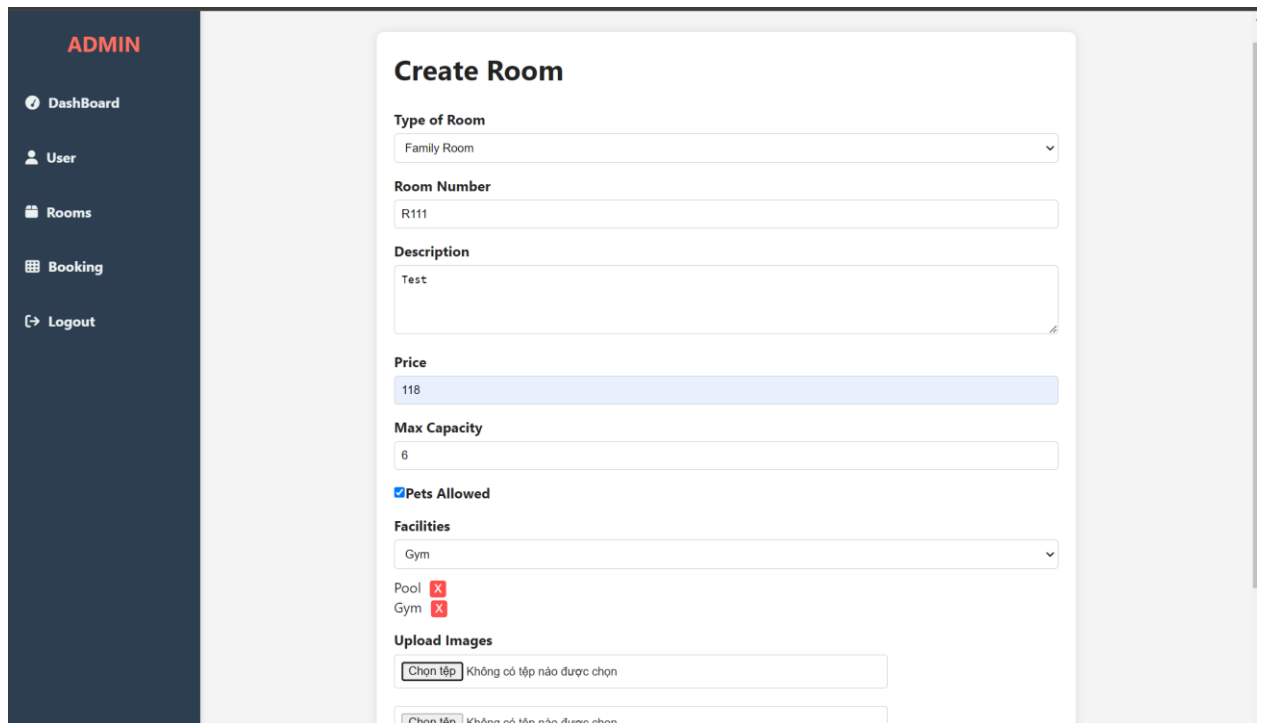
You can check which rooms are available during this period

R101	R102	R103	R104
R105	R106	R107	R108
R109	R110	R110	R111

Add Room

Hình 3.10 Giao diện trang quản lý phòng khách sạn

Ta có thể tạo một phòng mới bằng cách nhập đầy đủ các thông tin tiện ích của phòng, thêm ảnh mô tả cho phòng rồi nhấn “Create Room”



The screenshot shows an admin dashboard with a sidebar on the left and a main content area. The sidebar is dark blue with the word 'ADMIN' in red at the top. It contains a 'DashBoard' link with a checkmark icon, and links for 'User', 'Rooms', 'Booking', and 'Logout' with corresponding icons. The main content area is white and features a 'Create Room' form. The form has several sections: 'Type of Room' with a dropdown menu showing 'Family Room'; 'Room Number' with a text input field containing 'R111'; 'Description' with a text area containing 'Test'; 'Price' with a blue slider bar set to '118'; 'Max Capacity' with a text input field containing '6'; 'Pets Allowed' with a checked checkbox; 'Facilities' with a dropdown menu showing 'Gym'; and 'Upload Images' with two file upload buttons, each labeled 'Chọn tệp' and 'Không có tệp nào được chọn'. There are also small red 'X' icons next to 'Pool' and 'Gym' labels.

Hình 3.11 Giao diện tạo mới một phòng khách sạn

Chúng ta có thể chỉnh sửa thông tin của một phòng bằng cách nhấn vào phòng đó, trang update room sẽ mở ra và chỉ cần sửa đổi thông tin rồi nhấn “Update”

ADMIN

- Dashboard
- User
- Rooms
- Booking
- Logout

Description

The room features a plush king-size bed draped in crisp, high-quality linens, ensuring a restful night's sleep lulled by the gentle sound of the waves.

Price

118

Max Capacity

2

☒ **Pets Allowed**

Facilities

Select facility

Wifi ☒
Parking ☒
Pool ☒
Spa ☒

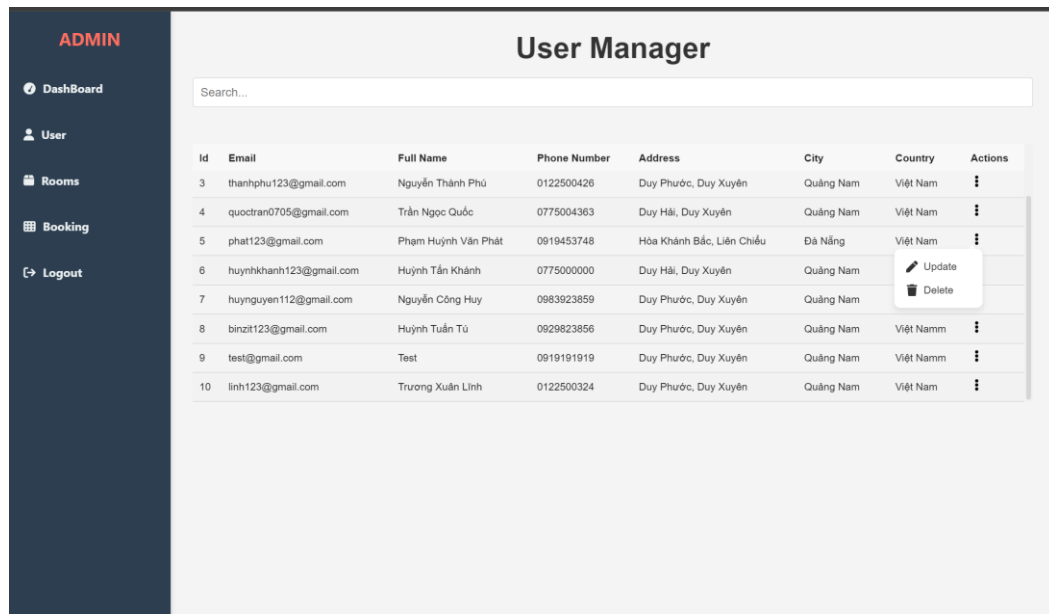
Upload Images

161853625.jpg 168166131.jpg 168211730.jpg

Update Room

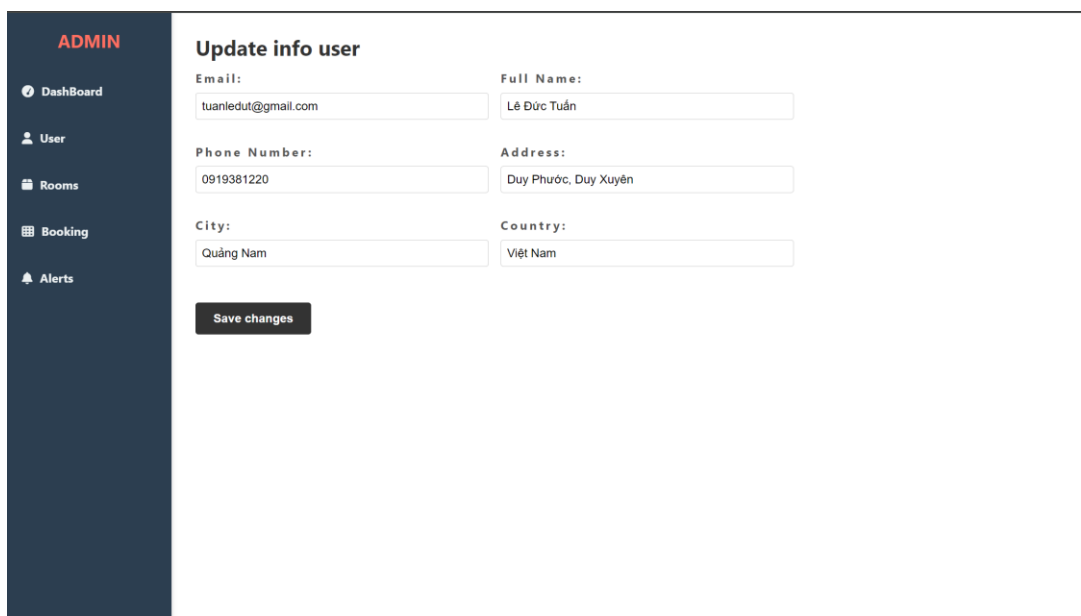
Hình 3.12 Giao diện chỉnh sửa phòng khách sạn

Đây là giao diện quản lý người dùng, ở đây chúng ta có thể tìm kiếm, chỉnh sửa thông tin và xóa người dùng khỏi hệ thống



Hình 3.13 Giao diện quản lý người dùng

Đây là trang chỉnh sửa để chỉnh sửa thông tin người dùng, nhập thông tin mới và nhấn “Save change”



Hình 3.14 Giao diện chỉnh sửa thông tin người dùng

Đây là trang quản lí đặt phòng trả phòng, ở đây sẽ hiển thị tất cả những lịch sử booking của người dùng, ở đây chúng ta có thể thao tác xác nhận đặt phòng hoặc trả phòng

ADMIN

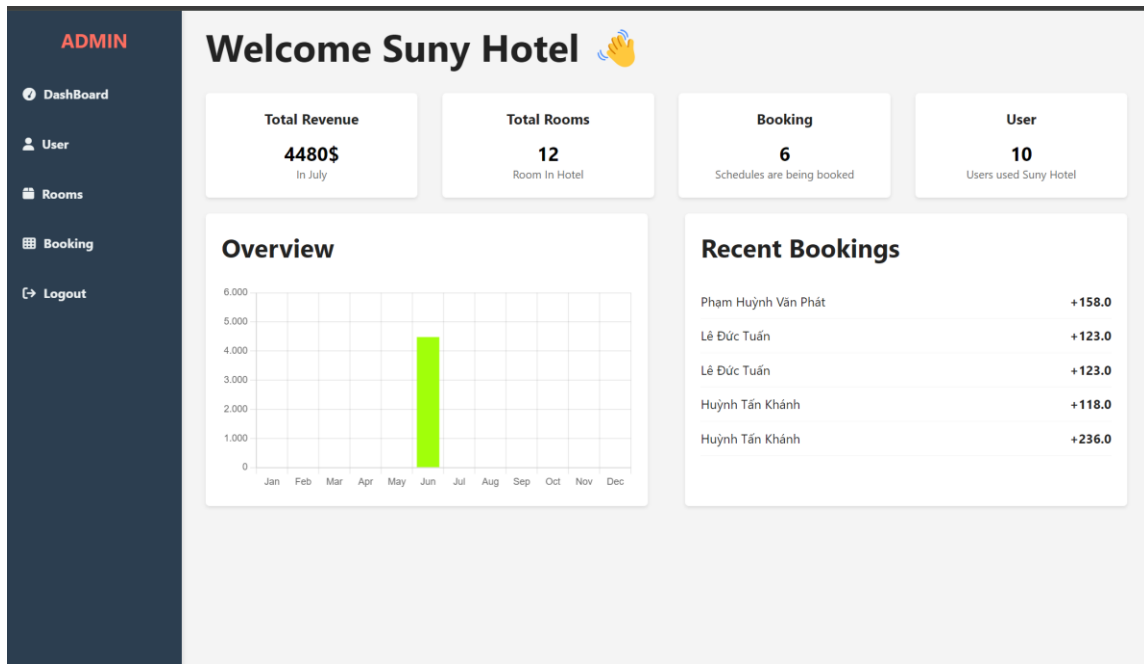
- DashBoard
- User
- Rooms
- Booking
- Alerts

Booking Manager

ID Booking	ID Room	Type Room	Full Name	Check-in Date	Check-out Date	Total Price	Booking At	Status	Confirm At	Action
69	6	Single Room	Lê Đức Tuấn	2024-06-18	2024-06-19	123.0	04:16:51, 17/06/2024	Waiting confirm..		<button>Confirm</button>
68	1	Single Room	Huỳnh Tấn Khánh	2024-06-18	2024-06-19	118.0	13:22:14, 16/06/2024	Confirmed	13:22:45, 16/06/2024	<button>Return</button>
67	1	Single Room	Huỳnh Tấn Khánh	2024-06-24	2024-06-26	236.0	13:13:41, 16/06/2024	Confirmed	13:14:17, 16/06/2024	<button>Return</button>
66	1	Single Room	Huỳnh Tấn Khánh	2024-06-27	2024-06-29	236.0	12:26:27, 16/06/2024	Confirmed	12:26:54, 16/06/2024	<button>Return</button>
65	1	Single Room	Huỳnh Tấn Khánh	2024-06-16	2024-06-17	118.0	12:22:38, 16/06/2024	Returned	12:23:16, 16/06/2024	Return at: 13:29:37, 16/06/2024
63	1	Single Room	Huỳnh Tấn Khánh	2024-06-23	2024-06-25	236.0	12:12:27, 16/06/2024	Returned	12:12:57, 16/06/2024	Return at: 12:30:48, 16/06/2024

Hình 3.15 Giao diện nhận phòng trả phòng

Đây là trang dùng để quản lí các thông tin, trạng thái tổng quan và doanh thu của khách sạn, chúng ta có thể theo dõi được doanh thu của khách sạn để có kế hoạch cho phù hợp



Hình 3.16 Giao diện quản lí trạng thái doanh thu của khách sạn

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận

Trong quá trình nghiên cứu lý thuyết và triển khai website công nghệ, đồ án đã đạt được những kết quả đáng kể.

Về mặt lý thuyết, đồ án đã thành công trong việc nghiên cứu và xây dựng mô hình lý thuyết cho website quản lý khách sạn. Các khía cạnh về quản lý dữ liệu, quy trình nghiệp vụ, và bảo mật thông tin đã được xem xét kỹ lưỡng. Đồ án đã phân tích và thiết kế cơ sở dữ liệu, xây dựng cấu trúc hệ thống và lập trình các chức năng cần thiết cho việc quản lý khách sạn.

Về mặt thực tiễn ứng dụng, đồ án đã triển khai và xây dựng thành công một website quản lý khách sạn. Website này không chỉ đáp ứng nhu cầu quản lý nội bộ của khách sạn mà còn cung cấp các tiện ích cho khách hàng trong việc đặt phòng trực tuyến. Website đã được thử nghiệm và chứng minh có khả năng quản lý thông tin khách hàng, đặt phòng, thanh toán, và các dịch vụ khác một cách hiệu quả. Điều này giúp nâng cao trải nghiệm của khách hàng và tối ưu hóa quy trình hoạt động của khách sạn.

2. Hướng phát triển

Mặc dù đồ án đã đạt được nhiều thành tựu, nhưng vẫn còn một số hạn chế cần được khắc phục để hoàn thiện hệ thống.

Về mặt độ ổn định và giao diện người dùng, hiện tại website vẫn còn một vài vấn đề về độ ổn định và giao diện chưa thực sự bắt mắt. Giao diện cần được thiết kế lại để thân thiện hơn với người dùng, tăng tính thẩm mỹ và dễ dàng thao tác.

Về các chức năng bổ sung, đồ án vẫn chưa tích hợp chức năng thanh toán online, điều này làm giảm tính tiện lợi cho khách hàng khi đặt phòng. Một số hướng phát triển cụ thể bao gồm:

Cải thiện giao diện: Đầu tư vào việc thiết kế giao diện người dùng sao cho trực quan, đẹp mắt và dễ sử dụng. Điều này có thể bao gồm việc sử dụng màu sắc hài hòa, bố cục hợp lý và các biểu tượng dễ nhận diện.

Thêm chức năng thanh toán online: Tích hợp các cổng thanh toán trực tuyến để khách hàng có thể thanh toán một cách nhanh chóng và an toàn. Điều này sẽ giúp tăng tính tiện lợi và sự hài lòng của khách hàng.

Chức năng chat trực tuyến: Xây dựng chức năng chat để khách hàng có thể liên hệ trực tiếp với khách sạn khi cần thiết. Chức năng này giúp giải đáp thắc mắc và hỗ trợ khách hàng một cách nhanh chóng và hiệu quả.

Hỗ trợ đa ngôn ngữ: Phát triển tính năng đa ngôn ngữ để phục vụ khách hàng quốc tế. Điều này bao gồm việc dịch toàn bộ nội dung website sang các ngôn ngữ phổ biến như tiếng Anh, tiếng Pháp, tiếng Trung, v.v.

Tăng cường bảo mật: Nâng cao các biện pháp bảo mật để bảo vệ thông tin khách hàng và dữ liệu của khách sạn khỏi các nguy cơ tấn công mạng.

Với những cải tiến và bổ sung này, đồ án sẽ không chỉ hoàn thiện hơn về mặt chức năng mà còn nâng cao trải nghiệm người dùng, giúp website trở thành một công cụ hữu ích và đáng tin cậy trong quản lý khách sạn.

TÀI LIỆU THAM KHẢO

- [1] Json Web Token, Link: <https://jwt.io/introduction>
- [2] Mô hình Client – Server, Link: [Client-Server Overview - Learn web development | MDN \(mozilla.org\)](#)
- [3] Mô hình 3 lớp, Link: <https://www.ibm.com/topics/three-tier-architecture>
- [4] <https://thaunguyen.com/learn-react/>
- [5] <https://codegym.vn/blog/reactjs-tim-hieu-ve-component-api/>
- [6] https://www.tutorialspoint.com/spring_boot/index.htm
- [7] <https://www.javatpoint.com/spring-boot-tutorial>