

TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO CUỐI KỲ

ĐỀ TÀI: HỆ THỐNG QUẢN LÝ KHÁCH SẠN (WEB APP)

Môn học: Lập trình cơ sở dữ liệu

GVHD: ThS. Phạm Chí Công

LỚP: DH22IT01

SINH VIÊN THỰC HIỆN:

Nguyễn Thành Nam - 2251050045

Nguyễn Phong Phú - 2251050057

Thành phố Hồ Chí Minh, 2025

MỤC LỤC

Chương 1. GIỚI THIỆU ĐỀ TÀI	2
1. Giới thiệu	2
2. Lý do chọn đề tài	2
Chương 2. CƠ SỞ LÝ THUYẾT	3
1. Cơ sở hạ tầng và nền tảng triển khai	3
2. Công nghệ phát triển giao diện người dùng (Frontend)	3
3. Công nghệ phát triển logic xử lý phía máy chủ (Backend)	4
4. Hệ quản trị cơ sở dữ liệu	4
5. Dịch vụ bên ngoài tích hợp	5
Chương 3. THIẾT KẾ HỆ THỐNG	6
1. Kiến trúc hệ thống	6
2. Lược đồ usecase	6
3. Đặc tả usecase	7
4. Sơ đồ lớp	11
5. Sơ đồ hoạt động	14
6. Sơ đồ tuần tự	17
7. Lược đồ cơ sở dữ liệu quan hệ	20
Chương 4. HỆ THỐNG QUẢN LÝ KHÁCH SẠN	22
1. Mô tả chức năng của đề tài	22
2. Mô tả sản phẩm	22
Chương 5. KẾT LUẬN	26
1. Kết quả đạt được	26
2. Hạn chế của đề tài	27
3. Phát triển đề tài trong tương lai	28
TÀI LIỆU THAM KHẢO	28
Tài liệu giấy	28
Tài liệu trực tuyến	29
BẢNG PHÂN CÔNG CÔNG VIỆC	29

Chương 1. GIỚI THIỆU ĐỀ TÀI

1. Giới thiệu

Ngành du lịch Việt Nam đang phát triển mạnh mẽ, đặc biệt với sự gia tăng lượng khách quốc tế (hơn 3 triệu lượt trong 2 tháng đầu năm 2024, tăng 68,7% so với cùng kỳ năm 2023). Trong bối cảnh đó, lĩnh vực khách sạn đóng vai trò quan trọng, đòi hỏi các giải pháp quản lý hiện đại để đáp ứng nhu cầu ngày càng cao của khách hàng. Đề tài "Web Quản lý Khách sạn" tập trung xây dựng một ứng dụng web nhằm tự động hóa và tối ưu hóa các quy trình như đặt phòng, thanh toán, quản lý thông tin khách hàng và báo cáo thống kê, mang lại hiệu quả kinh doanh và trải nghiệm tốt hơn cho khách hàng.

2. Lý do chọn đề tài

Ngành du lịch Việt Nam đang trên đà phát triển mạnh mẽ, dẫn đến nhu cầu bức thiết về một hệ thống quản lý khách sạn hiện đại và hiệu quả. Với sự gia tăng đáng kể về lượng khách quốc tế, các khách sạn cần một giải pháp toàn diện để tối ưu hóa quy trình vận hành, từ quản lý đặt phòng, thanh toán, thông tin khách hàng đến phân tích dữ liệu. Một hệ thống mạnh mẽ sẽ không chỉ giảm thiểu sai sót, nâng cao trải nghiệm khách hàng mà còn giúp doanh nghiệp duy trì lợi thế cạnh tranh trong bối cảnh thị trường sôi động.

Đề tài "Web Quản lý Khách sạn" được lựa chọn không chỉ vì tính cấp thiết về mặt thực tiễn mà còn bởi tiềm năng ứng dụng các công nghệ tiên tiến. Dự án này tận dụng sức mạnh của một kiến trúc Monorepo hiện đại, kết hợp Backend mạnh mẽ với Django và Django REST Framework sử dụng PostgreSQL làm cơ sở dữ liệu, cùng với Frontend linh hoạt được xây dựng bằng React và Material-UI. Các công nghệ này cho phép xây dựng một hệ thống với đầy đủ tính năng như quản lý phòng, đặt phòng thông minh, tích hợp thanh toán VNPay, phân tích doanh thu, và các tác vụ tự động hóa thông qua [Cron-job.org](#).

Ngoài ra, việc thực hiện đề tài này mang lại cơ hội quý báu để phát triển và củng cố kỹ năng chuyên môn trong lập trình Full-stack, thiết kế kiến trúc hệ thống, quản lý cơ sở dữ liệu và triển khai ứng dụng trên các nền tảng đám mây như Render.com. Hệ thống được thiết kế với khả năng mở rộng và thích ứng cao, đảm bảo giá trị sử dụng lâu dài và linh hoạt trước những biến động của thị trường khách sạn trong tương lai.

Chương 2. CƠ SỞ LÝ THUYẾT

1. Cơ sở hạ tầng và nền tảng triển khai

Toàn bộ ứng dụng được triển khai trên một nền tảng đám mây thông nhất, giúp đơn giản hóa việc quản lý và vận hành.

Render: Nền tảng cloud hosting này đóng vai trò là môi trường triển khai (deploy) cho toàn bộ ứng dụng, bao gồm cả Frontend và Backend. Render cung cấp một giải pháp quản lý máy chủ hoàn chỉnh, tự động xử lý các tác vụ như chạy ứng dụng, thiết lập môi trường và cung cấp chứng chỉ HTTPS, cho phép tập trung vào phát triển tính năng thay vì cấu hình hạ tầng phức tạp.

2. Công nghệ phát triển giao diện người dùng (Frontend)

Giao diện người dùng được xây dựng để mang lại trải nghiệm tương tác tốt nhất, sử dụng các công nghệ hiện đại và hiệu quả.

ReactJS: Là một thư viện JavaScript phổ biến, được sử dụng để xây dựng giao diện người dùng theo mô hình **component**. Phương pháp này giúp chia nhỏ giao diện thành các thành phần độc lập, dễ quản lý và tái sử dụng, tăng tính tổ chức cho toàn bộ dự án.

Material-UI (MUI): Bộ thư viện giao diện này được xây dựng trên ReactJS, cung cấp một hệ thống các component theo chuẩn **Material Design**. Việc sử

dụng MUI giúp đảm bảo giao diện có tính thẩm mỹ, hiện đại và nhất quán, đồng thời đẩy nhanh tốc độ thiết kế.

Vite: Công cụ build và dev server này được áp dụng để tăng tốc độ phát triển. Với khả năng hot-reload cực nhanh, Vite giúp giảm đáng kể thời gian chờ đợi, nâng cao năng suất làm việc.

3. Công nghệ phát triển logic xử lý phía máy chủ (Backend)

Phần backend chịu trách nhiệm xử lý các nghiệp vụ cốt lõi, quản lý dữ liệu và giao tiếp với frontend.

Django: Là một framework web mạnh mẽ của Python, Django được lựa chọn để xây dựng backend nhờ các tính năng tích hợp sẵn như ORM (Object-Relational Mapping), hệ thống bảo mật và cơ chế quản trị.

Django REST Framework (DRF): Đây là một thư viện mở rộng cho Django, chuyên dùng để xây dựng các **API RESTful**. DRF hỗ trợ việc **serialize** dữ liệu từ cơ sở dữ liệu sang định dạng JSON và ngược lại, giúp việc giao tiếp giữa frontend và backend trở nên hiệu quả.

JWT (JSON Web Token): Để đảm bảo an toàn cho các giao dịch và tài nguyên, một cơ chế xác thực và phân quyền người dùng thông qua **token** bảo mật đã được triển khai, sử dụng chuẩn JWT.

4. Hệ quản trị cơ sở dữ liệu

PostgreSQL: Hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở này được sử dụng để lưu trữ và quản lý toàn bộ dữ liệu của ứng dụng. Điều đặc biệt là cơ sở dữ liệu này được triển khai trực tiếp trên nền tảng **Render.com**, tạo nên một hệ sinh thái đồng bộ, giúp việc kết nối và quản lý dữ liệu trở nên đơn giản và hiệu quả hơn.

5. Dịch vụ bên ngoài tích hợp

Để mở rộng chức năng và cải thiện trải nghiệm người dùng, ứng dụng đã tích hợp một số dịch vụ của bên thứ ba.

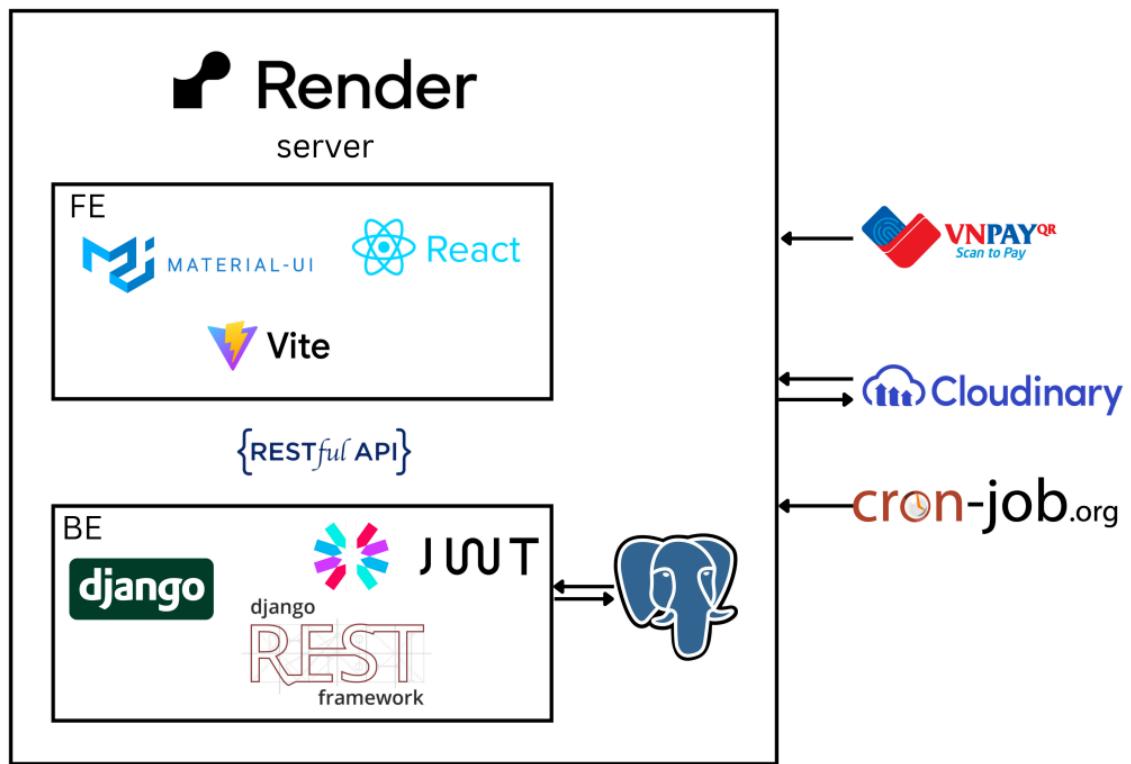
VNPay: Đây là cổng thanh toán trực tuyến, được tích hợp để xử lý các giao dịch thanh toán điện tử thông qua mã QR hoặc các phương thức khác, mang lại sự tiện lợi và an toàn cho người dùng.

Cloudinary: Dịch vụ lưu trữ và quản lý hình ảnh/video trên cloud này được sử dụng để xử lý các tài nguyên đa phương tiện. Cloudinary hỗ trợ việc upload, tối ưu hóa và phân phối nội dung, giúp giảm tải cho máy chủ và tăng tốc độ tải trang.

cron-job.org: Để tự động hóa các tác vụ định kỳ, dịch vụ này được sử dụng để lập lịch gọi các API của hệ thống. Điều này giúp thực hiện các công việc lặp lại như gửi thông báo hoặc đồng bộ dữ liệu mà không cần can thiệp thủ công.

Chương 3. THIẾT KẾ HỆ THỐNG

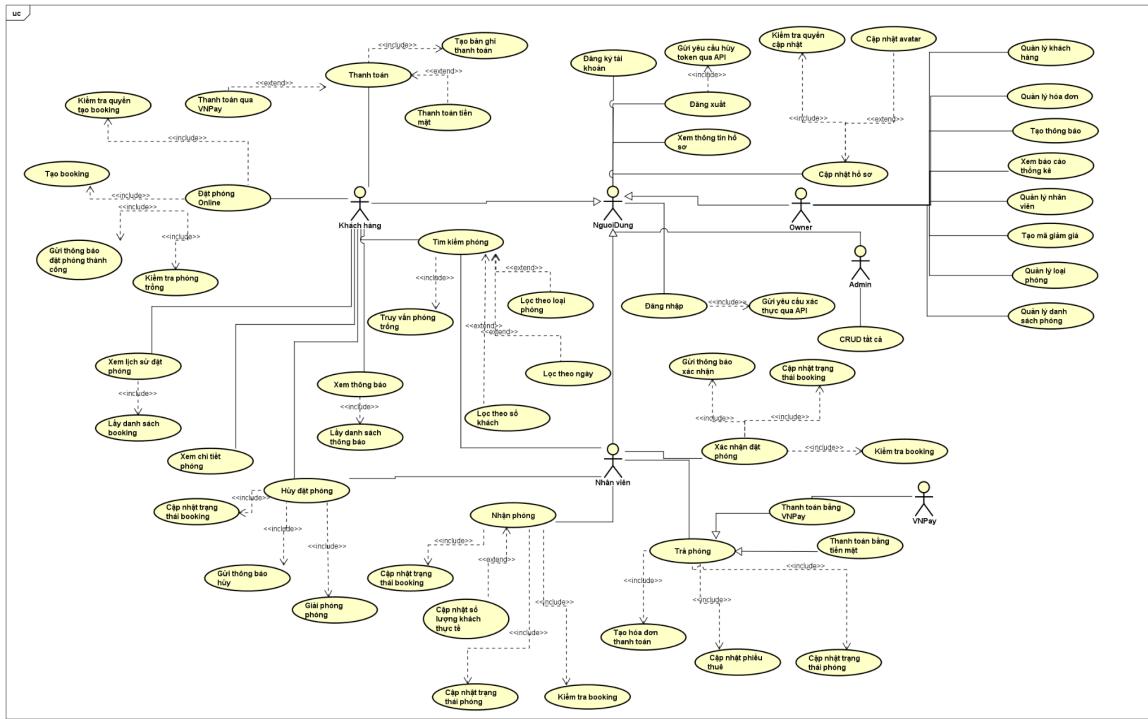
1. Kiến trúc hệ thống



Bảng 3.1: Kiến trúc hệ thống

2. Lược đồ usecase

Trình bày lược đồ tổng quát use case



Hình 3.2: Use case toàn hệ thống

3. Đặc tả usecase

Đặc tả use case “Đặt phòng”

Tên trường	Ý nghĩa
Use case Id	A001
Tên use case	Đặt phòng
Mô tả văn tắt	Khách hàng hoặc nhân viên tạo booking bằng cách chọn phòng, nhập thông tin (ngày nhận/trả phòng, số khách, yêu cầu đặc biệt). Hệ thống phân bổ khách thông minh và lưu booking.
Actor chính	Khách hàng, Nhân viên
Actor phụ	Không có.
Tiền điều kiện	- Khách hàng đã đăng nhập hoặc nhân viên có quyền tạo booking. - Phòng trống đã được xác định.
Hậu điều kiện	- Booking được tạo với trạng thái "pending". - Giá booking được tính và lưu.
Luồng hoạt động	B1: Khách hàng hoặc nhân viên truy cập mục đặt phòng. B2: Người dùng chọn phòng, nhập ngày nhận/trả phòng, số khách, yêu cầu đặc biệt, và mã giảm giá (tùy chọn). B3: Người dùng nhấn nút "Đặt phòng" để tiếp tục. B4: Hệ thống kiểm tra phòng trống và kiểm tra thời gian. B5: Hệ thống phân bổ khách vào phòng bằng và tính giá tạm tính.

	B6: Hệ thống tạo Booking với trạng thái "pending" và hiển thị thông tin booking.
Luồng thay thế	B5.1: Nếu số khách vượt sức chứa tối đa, hệ thống đề xuất chọn thêm phòng hoặc giảm số khách.
Luồng ngoại lệ	Nếu phòng đã được đặt bởi booking khác, hệ thống thông báo lỗi. Nếu số khách vượt sức chứa tổng, hệ thống thông báo lỗi và yêu cầu điều chỉnh.

Đặc tả use case “Nhận phòng (Check-in)”

Tên trường	Ý nghĩa
Use case Id	A002
Tên use case	Nhận phòng
Mô tả vắn tắt	Nhân viên hoặc khách hàng thực hiện check-in cho booking đã xác nhận, nhập số khách thực tế, tạo RoomRental, và cập nhật trạng thái phòng thành "occupied".
Actor chính	Nhân viên, Khách hàng.
Actor phụ	Không có.
Tiền điều kiện	- Booking ở trạng thái "confirmed". - Ngày hiện tại \geq ngày check-in đã đặt.
Hậu điều kiện	- RoomRental được tạo và liên kết với các phòng. - Trạng thái booking cập nhật thành "checked_in". - Trạng thái phòng được cập nhật thành "occupied". - Giá thực tế được tính dựa trên số khách thực tế (bao gồm phụ thu nếu có).
Luồng hoạt động	B1: Nhân viên truy cập hệ thống và tra cứu booking bằng ID hoặc thông tin khách hàng. B2: Hệ thống kiểm tra trạng thái booking là "confirmed" và hiển thị thông tin booking. B3: Nhân viên nhập số khách thực tế. B4: Hệ thống kiểm tra sức chứa phòng với số khách thực tế B5: Hệ thống kiểm tra trạng thái tất cả phòng trong booking phải là "booked" hoặc "available". B6: Hệ thống tính giá thực tế dựa trên số khách thực tế (áp dụng phụ thu nếu vượt sức chứa cơ bản). B8: Hệ thống tạo RoomRental, cập nhật trạng thái booking thành "checked_in", cập nhật trạng thái phòng thành "occupied", và hiển thị thông tin check-in.
Luồng thay thế	B2.1: Nếu booking không ở trạng thái "confirmed", hệ thống thông báo cần xác nhận trước. B5.1: Nếu số khách thực tế vượt 150% sức chứa, hệ thống thông báo lỗi với chi tiết: "Không thể check-in! Số khách thực tế (X) vượt quá giới hạn 150% sức chứa phòng (tối đa: Y khách)".

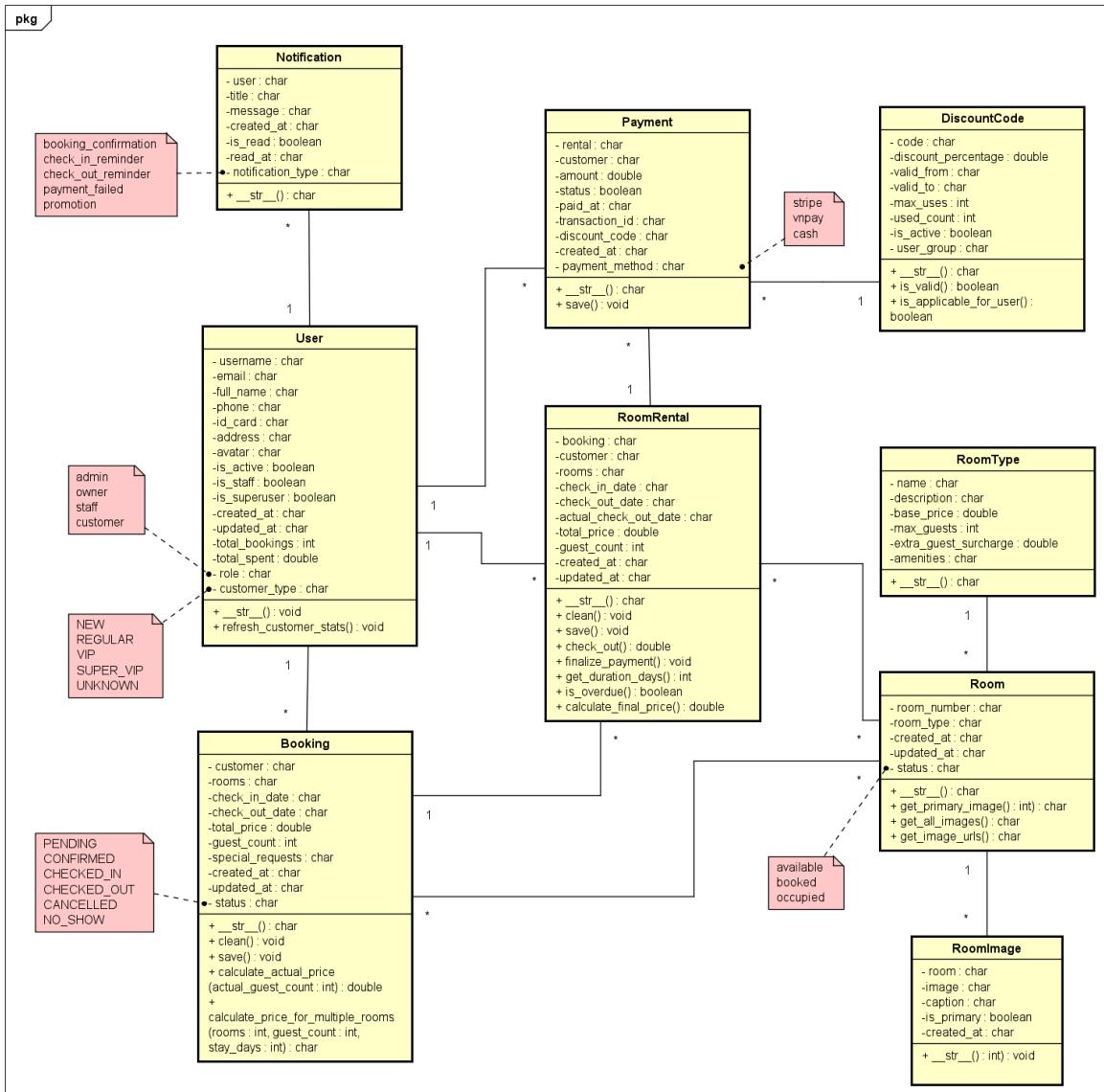
	B6.1: Nếu phòng không ở trạng thái "booked"/"available", hệ thống thông báo "Phòng {room_number} không khả dụng (trạng thái: {status})".
Luồng ngoại lệ	E1: Nếu booking không tồn tại, hệ thống thông báo "Booking not found". E2: Nếu chưa đến ngày check-in, hệ thống thông báo "Chưa đến ngày check-in. Ngày check-in: {date}, Ngày hiện tại: {current_date}". E3: Nếu số khách thực tế không hợp lệ (≤ 0 hoặc không phải số), hệ thống thông báo "Số khách thực tế không hợp lệ".

Đặc tả các use case “Trả phòng (Checkout)”

Tên trường	Ý nghĩa
Use case Id	A003
Tên usecase	Trả phòng
Mô tả văn tắt	Nhân viên thực hiện check-out cho khách hàng đã check-in, tính toán chi phí cuối cùng có thể áp dụng mã giảm giá, và xử lý thanh toán qua phương thức được chọn. Hệ thống cập nhật trạng thái phòng và hoàn tất quá trình lưu trú.
Actor chính	Nhân viên
Actor phụ	Hệ thống thanh toán VNPay,...
Tiền điều kiện	- Booking tồn tại và ở trạng thái "checked_in" - Khách hàng đã thực hiện check-in thành công - RoomRental tồn tại cho booking
Hậu điều kiện	- Hóa đơn thanh toán được tạo - Trạng thái booking được cập nhật thành "checked_out" - Thời gian trả phòng thực tế được ghi nhận - Trạng thái phòng được cập nhật thành "available" - Thông báo hoàn tất được gửi cho khách hàng
Luồng hoạt động	B1: Nhân viên chọn booking cần check-out từ danh sách B2: Hệ thống kiểm tra booking đã check-in và hiển thị thông tin chi tiết B3: Hệ thống hiển thị thông tin tính phí bao gồm thời gian lưu trú và số khách thực tế B4: Nhân viên có thể chọn mã giảm giá khả dụng (nếu có) B5: Hệ thống tính toán chi phí cuối cùng sau khi áp dụng giảm giá

	<p>B6: Nhân viên chọn phương thức thanh toán (tiền mặt hoặc VNPay)</p> <p>B7: Hệ thống xử lý thanh toán theo phương thức đã chọn:</p> <p>B7.1 - Thanh toán tiền mặt: Hệ thống ghi nhận thanh toán hoàn tất ngay lập tức Cập nhật trạng thái booking và phòng Hiển thị thông báo check-out thành công</p> <p>B7.2 - Thanh toán VNPay: Hệ thống tạo liên kết thanh toán VNPay Chuyển hướng khách hàng đến cổng thanh toán Chờ xác nhận kết quả từ VNPay</p> <p>B8: Hệ thống gửi thông báo hoàn tất cho khách hàng</p>
Luồng thay thế	<p>B2.1: Nếu booking chưa check-in, hệ thống thông báo "Booking chưa check-in hoặc đã check-out"</p> <p>B2.2: Nếu đã check-out trước đó, hệ thống thông báo "Đã check-out trước đó"</p> <p>B4.1: Nếu mã giảm giá không hợp lệ, hệ thống thông báo "Mã giảm giá không áp dụng được"</p> <p>B7.2.1: Nếu tạo liên kết VNPay thất bại, hệ thống thông báo lỗi và cho phép chọn phương thức khác</p> <p>Xử lý VNPay:</p> <p>C1: VNPay gửi kết quả thanh toán về hệ thống</p> <p>C2: Nếu thanh toán thành công: Hệ thống hoàn tất quá trình check-out Cập nhật trạng thái booking và phòng Gửi thông báo thành công cho khách hàng</p> <p>C3: Nếu thanh toán thất bại: Hệ thống giữ nguyên trạng thái booking</p> <p>C4: Hệ thống chuyển hướng về trang quản lý với thông báo kết quả</p>
Luồng ngoại lệ	<p>E1: Nếu booking không tồn tại, hệ thống thông báo "Không tìm thấy booking"</p> <p>E2: Nếu xảy ra lỗi hệ thống, hệ thống khôi phục trạng thái ban đầu và thông báo lỗi</p> <p>E3: Nếu VNPay không phản hồi, hệ thống ghi nhận lỗi và chuyển hướng về trang chính</p>

4. Sơ đồ lớp



Bảng 3.3: Sơ đồ lớp

Phân tích, giải thích các mối quan hệ thiết lập

1. Mối quan hệ Association giữa User và Booking (một - nhiều):

Một người dùng (User) với vai trò "customer" có thể thực hiện nhiều đặt phòng (Booking), nhưng mỗi đặt phòng chỉ thuộc về một khách hàng. Mỗi quan hệ này được thiết lập thông qua khóa ngoại.

2. Mối quan hệ Association giữa Booking và Room (nhiều - nhiều):

Một đặt phòng (Booking) có thể bao gồm nhiều phòng (Room), và một phòng có thể xuất hiện trong nhiều đặt phòng khác nhau vào các thời điểm khác nhau. Mỗi quan hệ này được thiết lập thông qua trường ManyToManyField trong model Booking.

3. Mối quan hệ Association giữa Room và RoomType (nhiều - một):

Nhiều phòng (Room) có thể thuộc về một loại phòng (RoomType), nhưng mỗi phòng chỉ thuộc về một loại phòng cụ thể. Mỗi quan hệ này được thiết lập thông qua ForeignKey từ Room đến RoomType.

4. Mối quan hệ Association giữa Room và RoomImage (một - nhiều):

Một phòng (Room) có thể có nhiều ảnh (RoomImage), nhưng mỗi ảnh chỉ thuộc về một phòng cụ thể. Mỗi quan hệ này được thiết lập thông qua ForeignKey từ RoomImage đến Room.

5. Mối quan hệ Association giữa Booking và RoomRental (một - nhiều):

Một đặt phòng (Booking) có thể dẫn đến nhiều phiếu thuê phòng (RoomRental) trong các trường hợp đặc biệt, nhưng mỗi phiếu thuê phòng thường chỉ liên kết với một đặt phòng. Mỗi quan hệ này được thiết lập thông qua ForeignKey từ RoomRental đến Booking.

6. Mối quan hệ Association giữa RoomRental và Room (nhiều - nhiều):

Một phiếu thuê phòng (RoomRental) có thể bao gồm nhiều phòng (Room), và một phòng có thể xuất hiện trong nhiều phiếu thuê phòng khác nhau. Mỗi quan hệ này được thiết lập thông qua ManyToManyField trong RoomRental.

7. Mối quan hệ Association giữa RoomRental và Payment (một - nhiều):

Một phiếu thuê phòng (RoomRental) có thể có nhiều thanh toán (Payment), nhưng mỗi thanh toán chỉ thuộc về một phiếu thuê phòng. Mỗi quan hệ này được thiết lập thông qua ForeignKey từ Payment đến RoomRental.

8. Mối quan hệ Association giữa User và Payment (một - nhiều):

Một người dùng (User) với vai trò "customer" có thể có nhiều thanh toán (Payment), nhưng mỗi thanh toán chỉ thuộc về một khách hàng. Mỗi quan hệ này được thiết lập thông qua khóa ngoại từ Payment đến User.

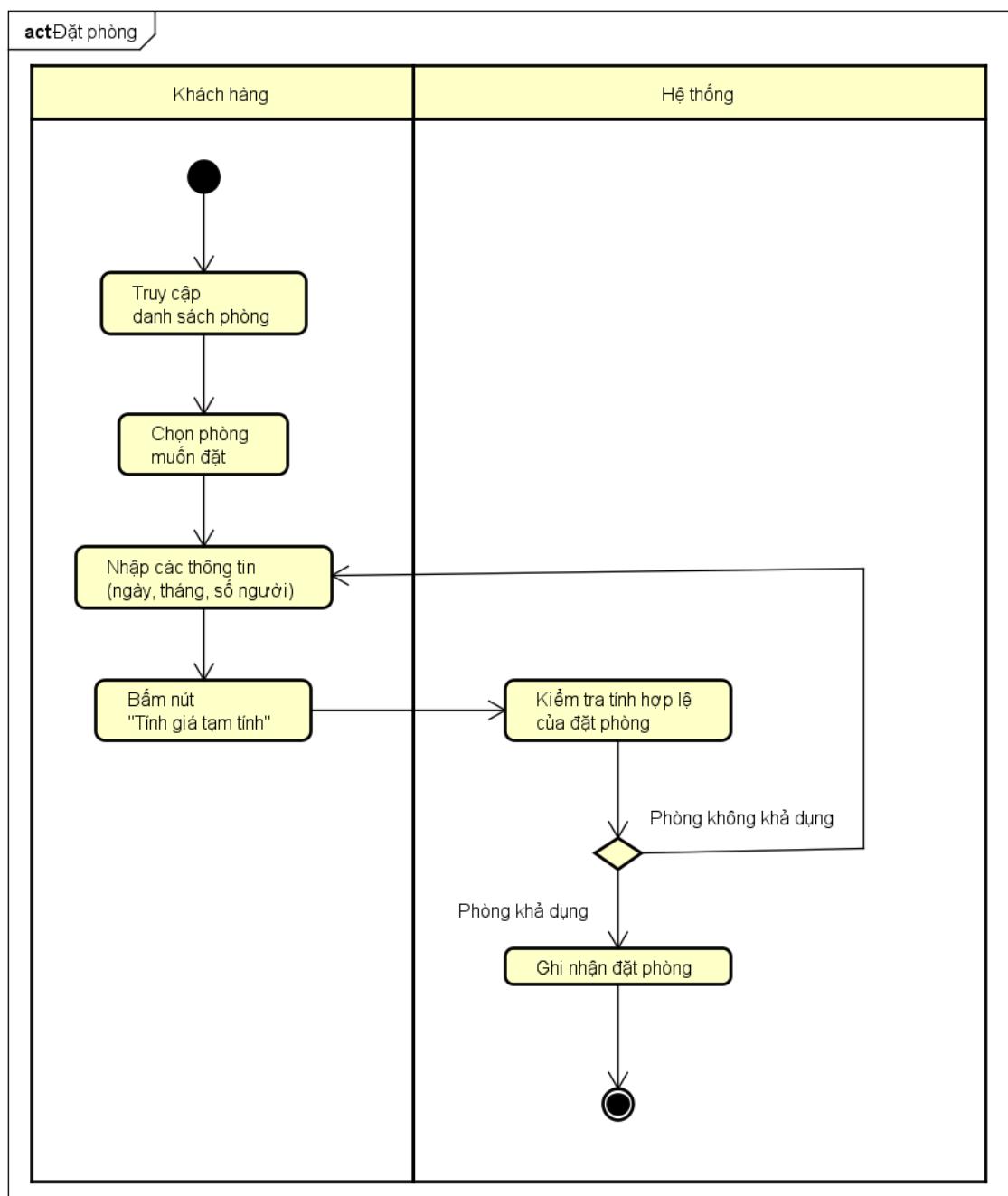
9. Mối quan hệ Association giữa Payment và DiscountCode (nhiều - một):

Nhiều thanh toán (Payment) có thể sử dụng một mã giảm giá (DiscountCode), nhưng mỗi thanh toán chỉ sử dụng một mã giảm giá (nếu có). Mỗi quan hệ này được thiết lập thông qua khóa ngoại từ Payment đến DiscountCode.

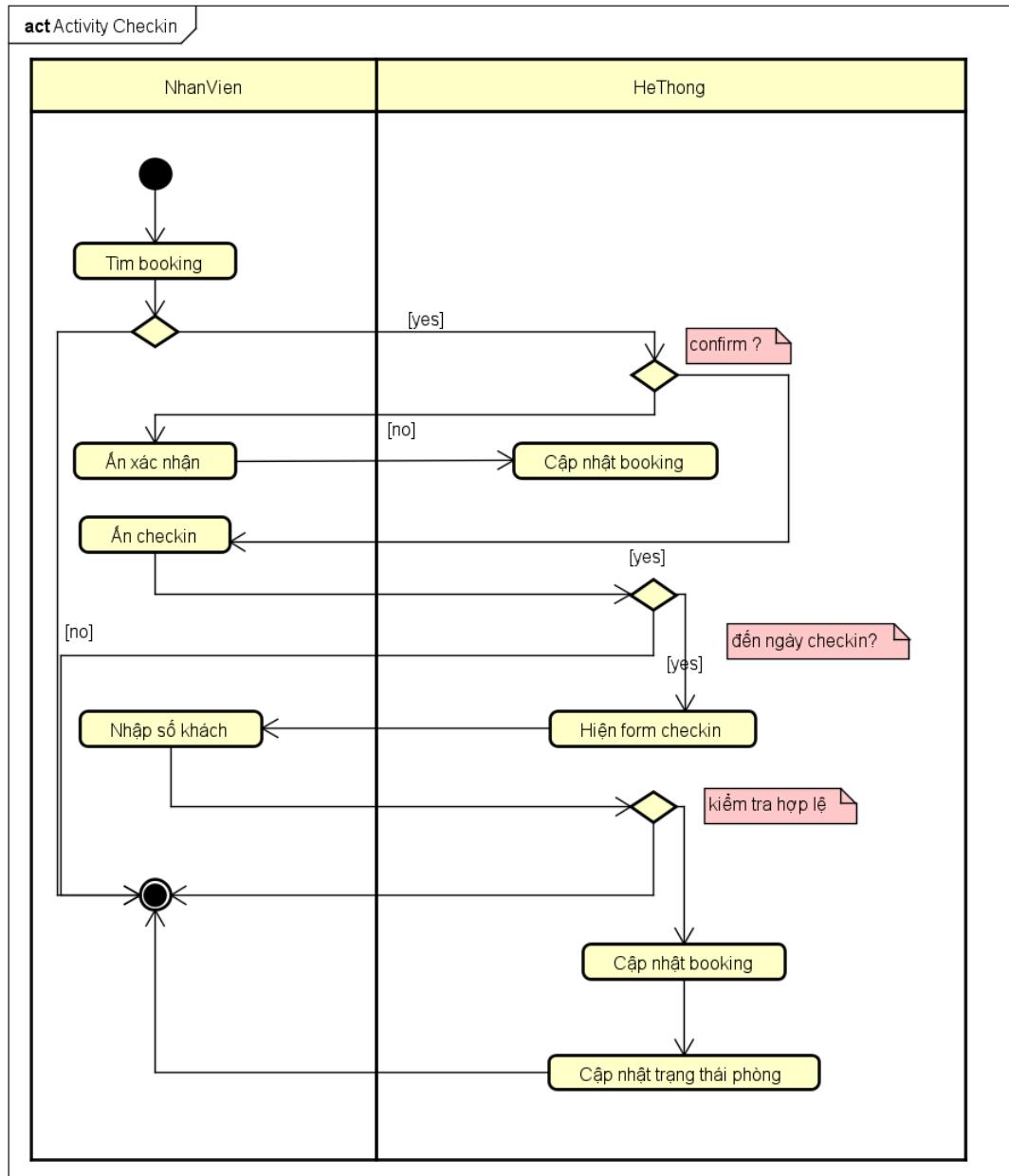
10. Mối quan hệ Association giữa User và Notification (một - nhiều):

Một người dùng (User) có thể nhận nhiều thông báo (Notification), nhưng mỗi thông báo chỉ được gửi đến một người dùng cụ thể. Mỗi quan hệ này được thiết lập thông qua ForeignKey từ Notification đến User.

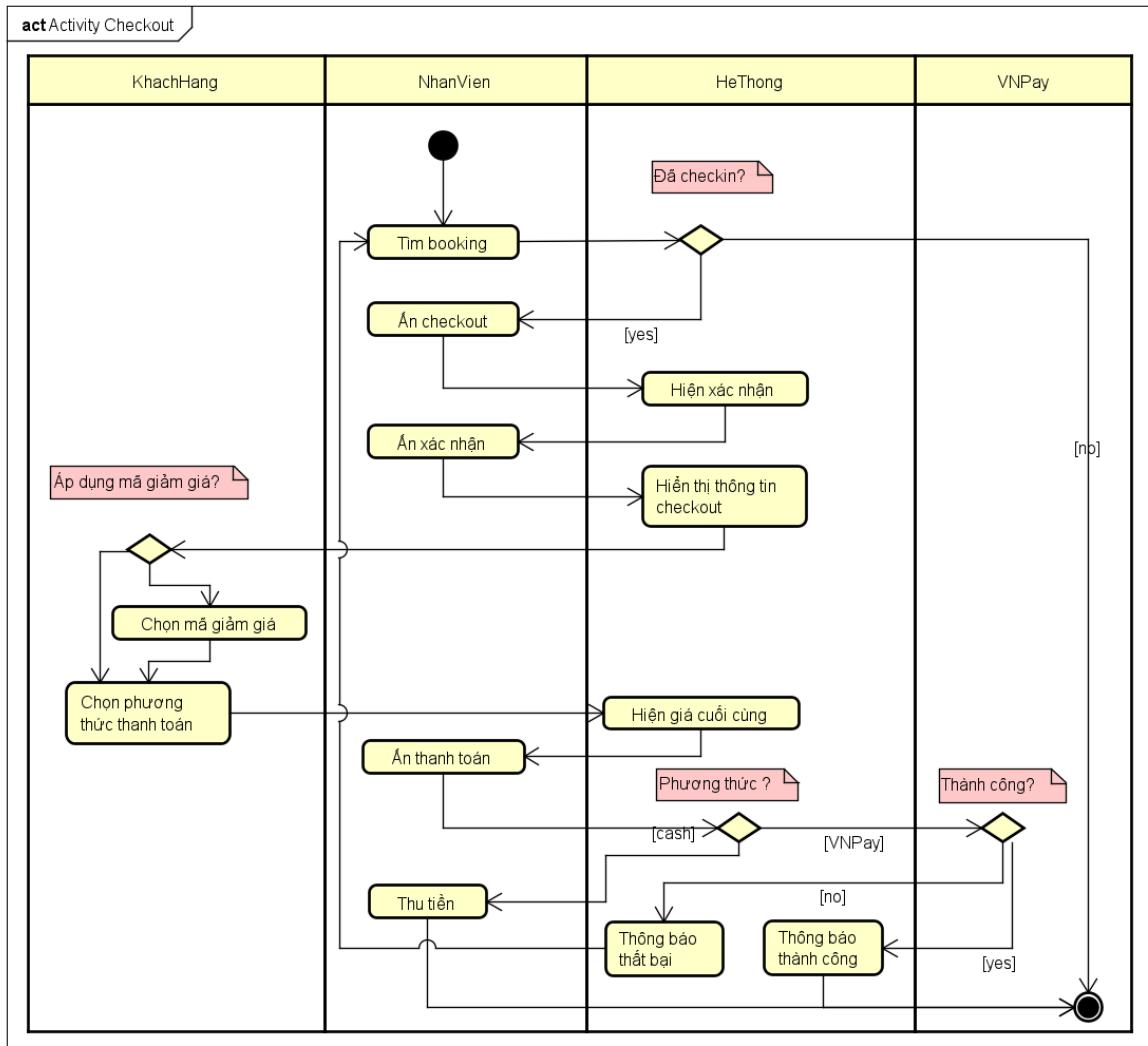
5. Sơ đồ hoạt động



Hình 3.4: Activity Diagram “Đặt phòng”

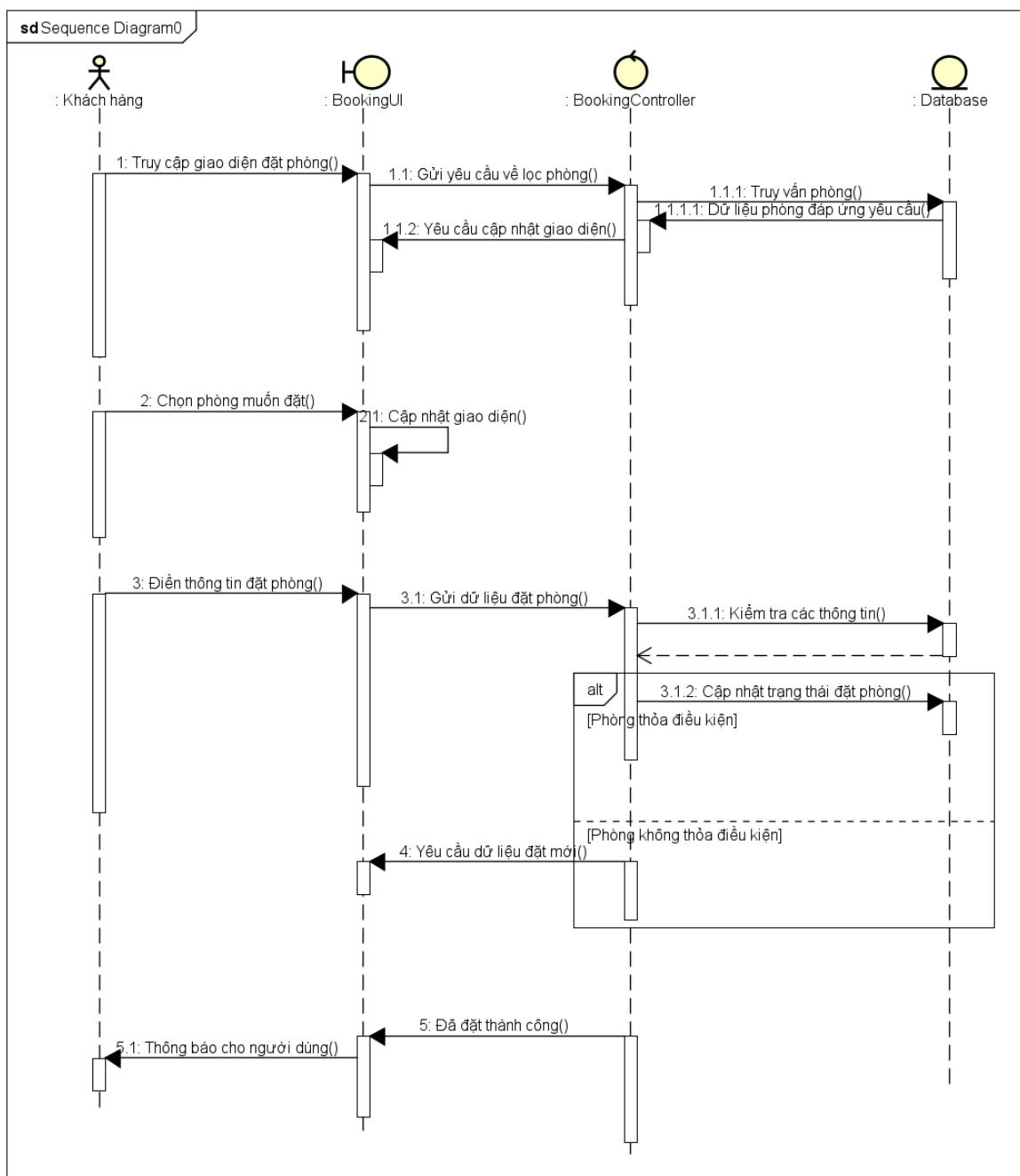


Hình 3.5: Activity Diagram “Nhận phòng”

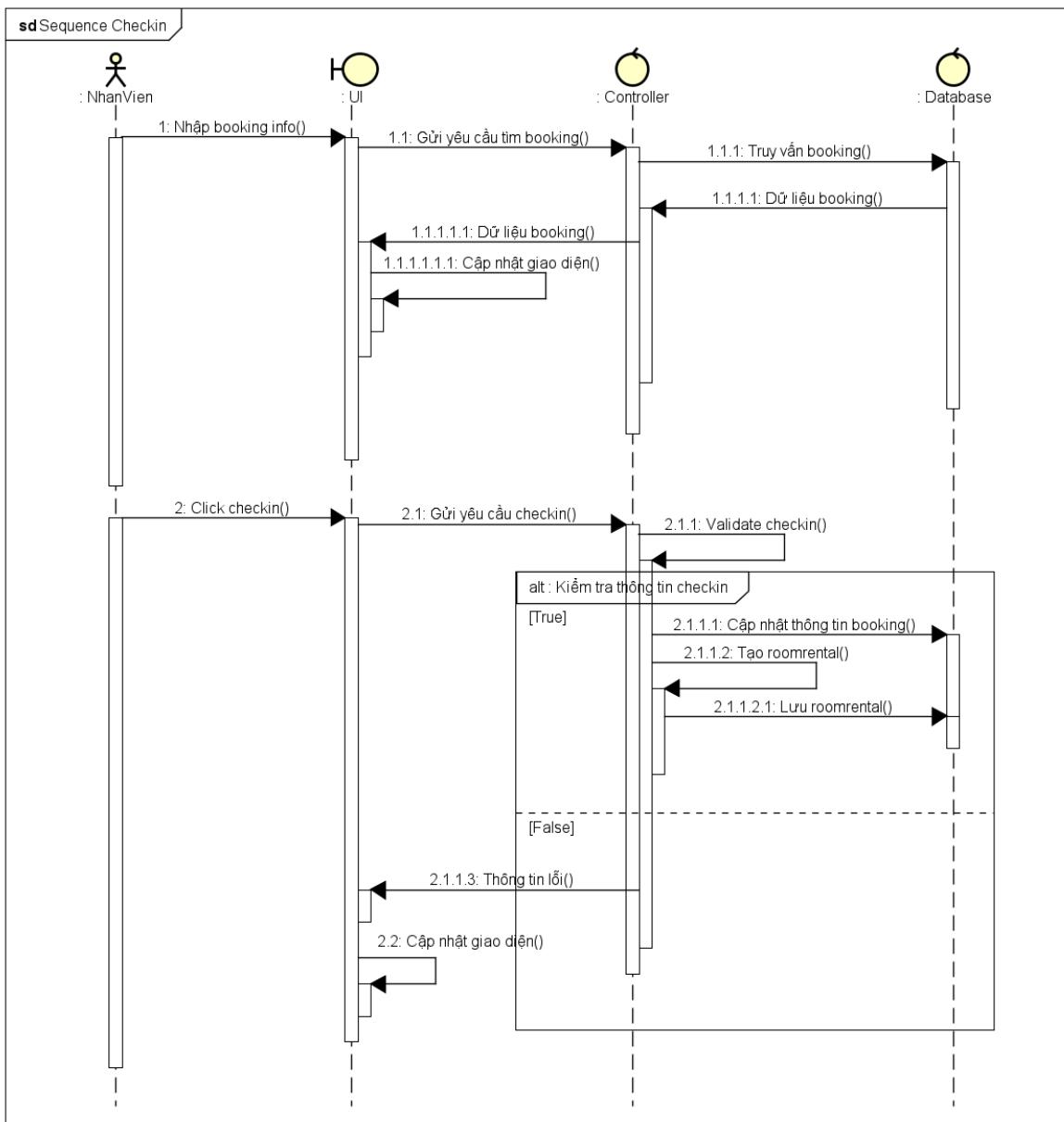


Hình 3.6: Activity Diagram “Trả phòng”

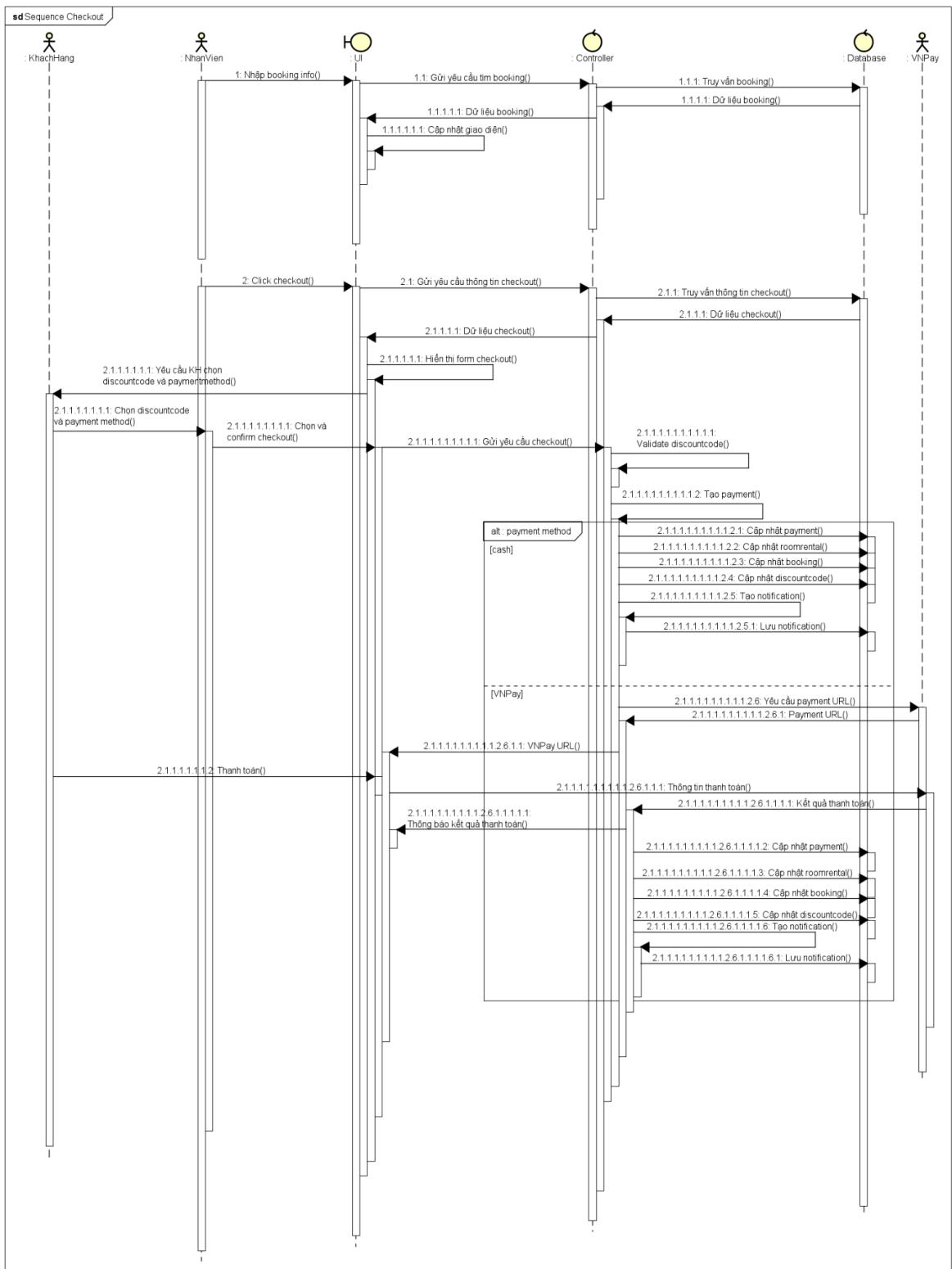
6. Sơ đồ tuần tự



Hình 3.7: Sequence Diagram “Đặt phòng”

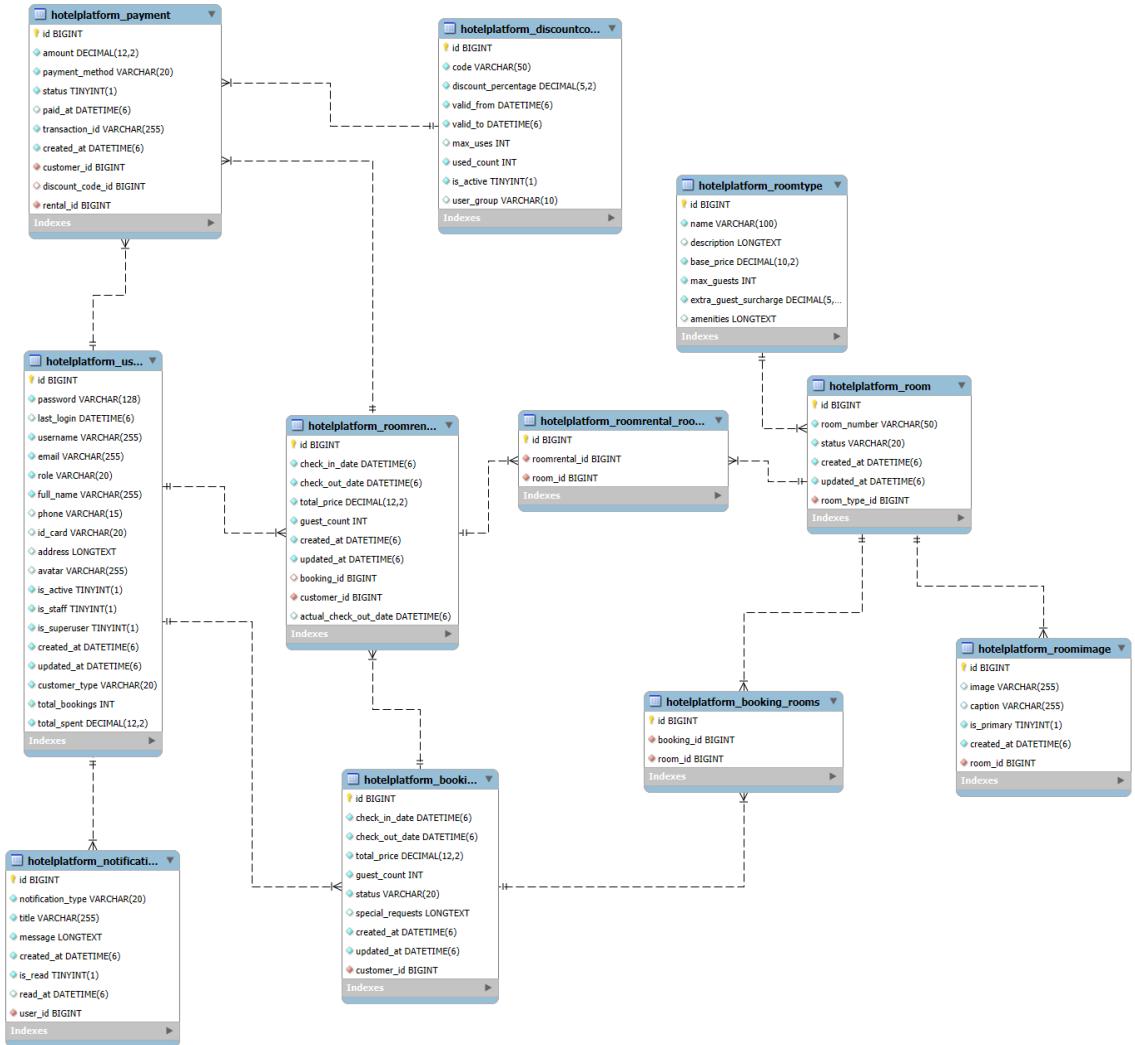


Hình 3.8: Sequence Diagram “Nhận phòng”



Hình 3.9: Sequence Diagram “Trả phòng”

7. Lược đồ cơ sở dữ liệu quan hệ



Bảng 3.10: Cơ sở dữ liệu quan hệ

User([id](#), [username](#), [email](#), [role](#), [full_name](#), [phone](#), [id_card](#), [address](#), [avatar](#), [is_active](#), [is_staff](#), [is_superuser](#), [created_at](#), [updated_at](#), [customer_type](#), [total_bookings](#), [total_spent](#), [password](#), [last_login](#))

RoomType([id](#) , [name](#), [description](#), [base_price](#), [max_guests](#), [extra_guest_surcharge](#), [amenities](#))

Room([id](#), [room_number](#), #[room_type_id](#), [status](#), [created_at](#), [updated_at](#))

RoomImage(id , #room_id , image, caption, is_primary, created_at)

Booking(id , #customer_id , check_in_date, check_out_date, total_price, guest_count, status, special_requests, created_at, updated_at, #rooms)

booking_rooms(id , #booking_id , #room_id)

RoomRental(id , #customer_id, #rooms, check_in_date, check_out_date, actual_check_in_date, actual_check_out_date, total_price, guest_count, status, special_requests, created_at, updated_at)

roomrental_rooms(id , #roomrental_id, #room_id)

Payment(id , #rental_id, #customer_id), amount, payment_method, status, paid_at, transaction_id, #discount_code_id, created_at)

DiscountCode(id , code, discount_percentage, valid_from, valid_to, max_uses, used_count, is_active, user_group)

Notification(id , #user_id, notification_type, title, message, created_at, is_read, read_at)

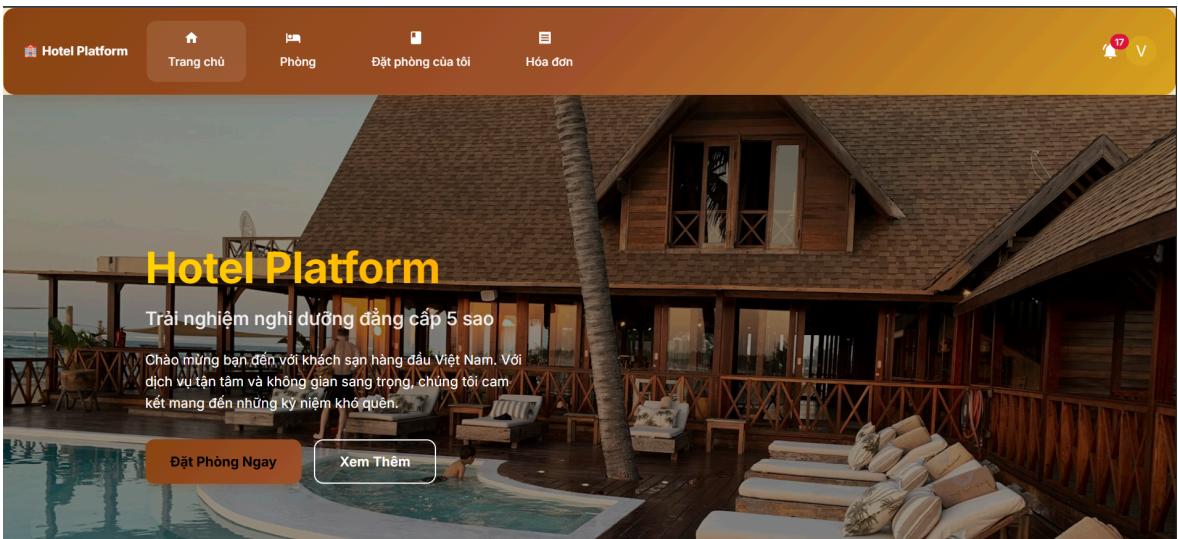
Chương 4. HỆ THỐNG QUẢN LÝ KHÁCH SẠN

1. Mô tả chức năng của đề tài

Hệ thống Quản lý Khách sạn là ứng dụng web được xây dựng trên nền tảng Django REST Framework nhằm tự động hóa và tối ưu hóa các quy trình vận hành khách sạn, phục vụ cả khách hàng, nhân viên và quản trị viên. Với khách hàng, hệ thống cung cấp chức năng tìm kiếm phòng theo loại, giá; đặt phòng kèm thông tin chi tiết; thanh toán trực tuyến qua VNPay hoặc tiền mặt; quản lý hồ sơ cá nhân và nhận thông báo về tình trạng đặt phòng, thanh toán. Đối với nhân viên và quản lý, hệ thống cho phép quản lý phòng và loại phòng, xử lý các đơn đặt với nhiều trạng thái, thực hiện check-in/check-out, tính toán chi phí, quản lý hóa đơn thanh toán, và quản lý khách hàng. Chức năng thống kê và báo cáo hỗ trợ phân tích doanh thu, lượng khách để đưa ra quyết định kinh doanh. Ngoài ra, hệ thống tích hợp xác thực bảo mật bằng JWT, phân quyền chi tiết theo vai trò, tự động hóa các tác vụ định kỳ như cập nhật trạng thái phòng và gửi thông báo, quản lý dữ liệu ảnh qua Cloudinary, đồng thời cung cấp API RESTful hỗ trợ tìm kiếm, phân trang và tích hợp với hệ thống bên ngoài. Những chức năng này giúp nâng cao hiệu quả vận hành, giảm sai sót thủ công và mang lại trải nghiệm dịch vụ tốt hơn cho người dùng.

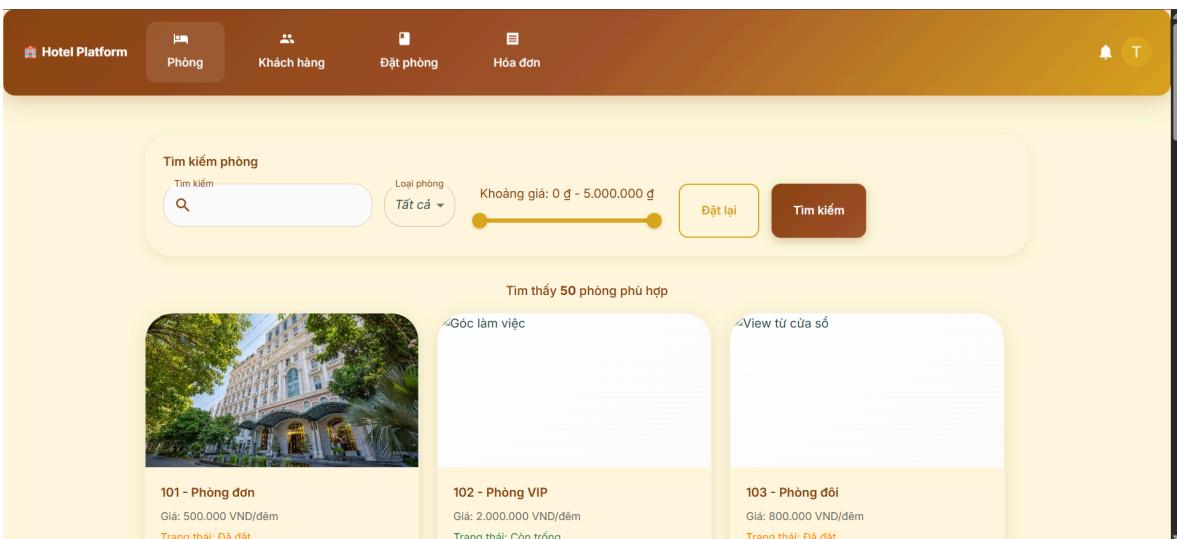
2. Mô tả sản phẩm

Hệ thống Quản lý Khách sạn là một ứng dụng web hiện đại, tự động hóa quy trình đặt phòng, thanh toán và quản lý khách sạn, mang lại trải nghiệm tiện lợi và hiệu quả vận hành tối ưu. Sản phẩm giúp giảm sai sót và hỗ trợ quyết định kinh doanh qua các công cụ phân tích mạnh mẽ.



Hình 4.1: Giao diện trang chủ.

Giao diện nhân viên cung cấp trang chủ trực quan để tìm kiếm phòng theo ngày, giá cả và loại phòng đồng thời có thể đặt phòng và quản lý hóa đơn.



Hình 4.2: Giao diện của Nhân viên

Quản lý đặt phòng										
ID	Khách hàng	Số điện thoại	Phòng	Check-in	Check-out	Tổng tiền	Số khách	Trạng thái	Hành động	
61	Vũ Lê	+84-79-039880	103	07:00 12/8/25	07:00 15/8/25	2.400.000 ₫	1	Chờ xác nhận		
60	Vũ Lê	+84-79-039880	202	07:00 10/8/25	07:00 12/8/25	1.000.000 ₫	1	Đã check-out		
59	Vũ Lê	+84-79-039880	201	07:00 12/8/25	07:00 15/8/25	10.500.000 ₫	5	Chờ xác nhận		
58	Bà Linh Lê	01 0676 9709	407	10:08 10/8/25	10:08 12/8/25	7.000.000 ₫	3	Đã check-in		
57	Bà Hà Hoàng	+84-93-229 1...	206	10:08 10/8/25	10:08 13/8/25	3.600.000 ₫	3	Đã check-in		
56	Bà Phương Vũ	(04) 1427 7533	108	10:08 10/8/25	10:08 11/8/25	500.000 ₫	1	Đã check-in		
55	Bà Hà Hoàng	+84-93-229 1...	101	10:08 16/8/25	10:08 19/8/25	1.500.000 ₫	1	Chờ xác nhận		

Hình 4.3: Giao diện quản lý đặt phòng của nhân viên

Khách hàng có thể đặt phòng, nhập thông tin như ngày nhận/trả, số khách, yêu cầu đặc biệt và áp dụng mã giảm giá.

Đặt Phòng

Thông Tin Phòng
 Phòng đã chọn: 106 - Phòng đơn (500.000 ₫/đêm)

Thông Tin Đặt Phòng

Ngày nhận phòng *

Ngày trả phòng *

Số khách *

Tối đa 1 khách

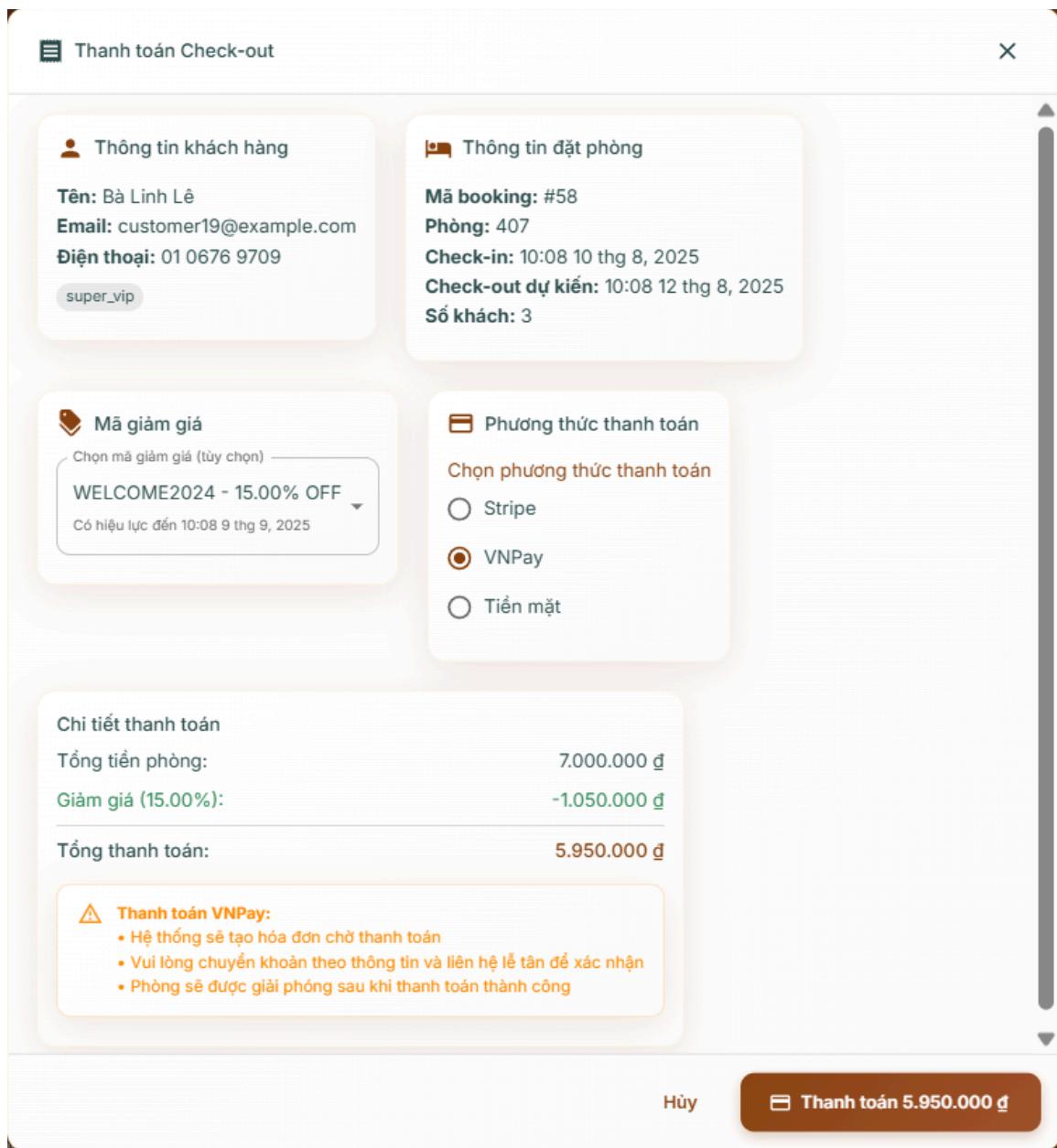
Yêu cầu đặc biệt

Tính Giá Tạm Tính

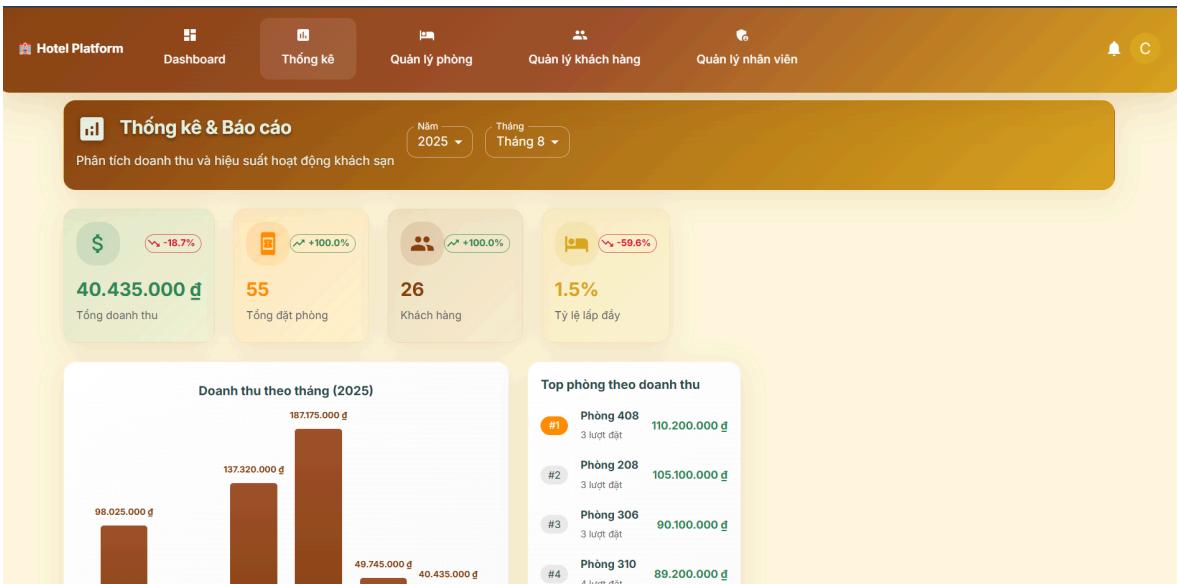
Giá Tạm Tính
 Số đêm: 3
Tổng giá: 1.500.000 ₫

Hình 4.3: Giao diện đặt phòng

Thanh toán được thực hiện an toàn qua cổng VNPay.



Hình 4.4:Giao diện Thanh toán



Hình 4.5: Giao diện thống kê báo cáo

Nhân viên sử dụng bảng điều khiển để quản lý phòng, check-in/check-out, thông tin khách hàng và theo dõi hóa đơn, đồng thời tạo báo cáo doanh thu chi tiết. Hệ thống tích hợp cron-job.org để tự động hóa tác vụ như gửi thông báo và đồng bộ dữ liệu. Được xây dựng trên ReactJS, Material-UI cho giao diện, Django và PostgreSQL cho backend, cùng các dịch vụ như VNPay và Cloudinary, sản phẩm triển khai trên Render.com, đảm bảo tính ổn định và khả năng mở rộng để đáp ứng nhu cầu khách sạn hiện đại.

Chương 5. KẾT LUẬN

1. Kết quả đạt được

Đề tài "**Web Quản lý Khách sạn**" đã thành công xây dựng một hệ thống ứng dụng web hoàn chỉnh, tự động hóa và tối ưu hóa hiệu quả các nghiệp vụ chính. Các chức năng cốt lõi như **quản lý phòng**, **đặt phòng**, **nhận/trả phòng** được triển khai linh hoạt, tích hợp cả **cổng thanh toán VNPay** an toàn. Hệ thống còn bao gồm quản lý thông tin khách hàng, nhân viên và tính năng **báo cáo thống kê doanh thu** chính xác, cùng với tự động hóa tác vụ định kỳ qua cron-job.org.

Về mặt kỹ thuật, ứng dụng được phát triển theo mô hình **Full-stack** với **Django/Django REST Framework** ở backend và **ReactJS/Material-UI** ở frontend, sử dụng cơ sở dữ liệu **PostgreSQL**. Hệ thống đã được triển khai ổn định trên nền tảng đám mây **Render.com** và tích hợp dịch vụ **Cloudinary** để tối ưu hóa lưu trữ đa phương tiện. Giá trị thực tiễn của dự án là rõ ràng: rút ngắn thời gian thao tác, giảm thiểu sai sót, nâng cao trải nghiệm khách hàng và cung cấp dữ liệu chính xác hỗ trợ quyết định kinh doanh. Quá trình này đã giúp củng cố toàn diện năng lực lập trình Full-stack, thiết kế kiến trúc, quản lý cơ sở dữ liệu và triển khai cloud, đồng thời rèn luyện kỹ năng làm việc nhóm và giải quyết vấn đề, tạo tiền đề vững chắc cho khả năng mở rộng trong tương lai.

2. Hạn chế của đề tài

Chưa hỗ trợ đăng nhập bằng tài khoản bên thứ ba:

Hiện tại, khách hàng phải tạo tài khoản riêng trên hệ thống để đăng nhập. Hệ thống chưa cho phép đăng nhập nhanh bằng Google, Facebook hay các ứng dụng phổ biến khác. Điều này có thể làm khách hàng thấy bất tiện, vì nhiều người đã quen đăng nhập chỉ bằng một cú nhấp chuột.

Bảo mật người dùng chưa thực sự tối ưu:

Mặc dù hệ thống dùng JWT để bảo vệ API, nhưng chưa thấy đề cập đến các tính năng bảo mật nâng cao như xác thực hai yếu tố (2FA) – ví dụ, gửi mã OTP qua điện thoại hoặc email. Nếu mật khẩu của người dùng yếu hoặc bị lộ, tài khoản có thể dễ bị xâm nhập hơn.

Chưa hỗ trợ nhiều ngôn ngữ:

Điều này có thể gây khó khăn cho các khách sạn phục vụ khách quốc tế, vì khách hàng hoặc nhân viên có thể gặp rắc rối nếu không hiểu ngôn ngữ chính của hệ thống.

Chưa tích hợp thanh toán đa dạng:

Hệ thống có xử lý thanh toán, nhưng chưa có hỗ trợ nhiều phương thức thanh toán phổ biến như ví điện tử (Momo, ZaloPay), thẻ quốc tế (Visa, MasterCard), hay tiền mặt tại quầy. Với khách hàng ở Việt Nam, việc thiếu các phương thức thanh toán quen thuộc có thể gây bất tiện.

Chưa có hỗ trợ khách hàng 24/7:

Chưa có kênh hỗ trợ người dùng (như chat trực tuyến, hotline, hoặc email hỗ trợ). Nếu khách sạn gặp lỗi kỹ thuật (ví dụ: không đặt phòng được hoặc báo cáo sai), họ có thể phải tự xử lý hoặc chờ lâu để được hỗ trợ.

3. Phát triển để tài trong tương lai

- Bổ sung đăng nhập bằng tài khoản bên thứ ba như Google, Facebook hoặc các nền tảng phổ biến khác, giúp người dùng truy cập nhanh chóng và thuận tiện hơn.
- Tăng cường bảo mật tài khoản người dùng bằng việc tích hợp các tính năng như xác thực hai yếu tố (2FA) hoặc mã OTP, đảm bảo an toàn hơn cho dữ liệu cá nhân.
- Hỗ trợ đa ngôn ngữ để đáp ứng nhu cầu của khách hàng và nhân viên từ nhiều quốc gia khác nhau, nâng cao trải nghiệm sử dụng.
- Mở rộng phương thức thanh toán, bổ sung các hình thức phổ biến tại Việt Nam và quốc tế như ví điện tử, thẻ ngân hàng nội địa, thẻ quốc tế hoặc thanh toán tại quầy.
- Xây dựng kênh hỗ trợ khách hàng 24/7 thông qua chat trực tuyến, hotline hoặc email hỗ trợ, đảm bảo người dùng luôn nhận được sự trợ giúp kịp thời.

TÀI LIỆU THAM KHẢO

Tài liệu giấy

- [1] Dương Hữu Thành, *Phát triển ứng dụng Django rest framework và react*. TP. Hồ Chí Minh: Nhà Xuất Bản Thông Tin Và Truyền Thông, 2024.

[2] Dương Hữu Thành, *Công nghệ phần mềm*. TP. Hồ Chí Minh: Nhà Xuất Bản Thông Tin Và Truyền Thông, 2023.

Tài liệu trực tuyến

[1] Cloudinary, "Managing Media Files in Django (Images, Videos, and Audio)", Cloudinary Blog, [Online]. Available:

<https://cloudinary.com/blog/managing-media-files-in-django>. [Accessed: Aug. 13, 2025].

[2] VNPAY, "Kết nối Cổng thanh toán VNPAY", VNPAY Developer Documentation, [Online]. Available:
<https://sandbox.vnpayment.vn/apis/docs/thanh-toan-pay/pay.html>. [Accessed: Aug. 13, 2025].

BẢNG PHÂN CÔNG CÔNG VIỆC

Họ và tên	Công việc	Mức độ hoàn thành
Nguyễn Phong Phú	Cơ sở dữ liệu; Xem phòng; Xem chi tiết phòng; Đặt phòng; Xem hóa đơn; Đăng ký tài khoản; Nhận phòng; Trả phòng; Báo cáo	100%
Nguyễn Thanh Nam	Cơ sở dữ liệu; QL nhân viên; QL phòng và loại phòng; QL khách hàng; Thống kê và báo cáo; Đăng nhập; Đăng ký; Hồ sơ cá nhân; Quản lý đặt phòng; Nhận phòng; Trả phòng; Thanh toán VNPay; Thông báo; Deploy; Báo cáo	100%