**TRƯỜNG ĐẠI HỌC PHƯƠNG ĐÔNG**

KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

NGÀNH CÔNG NGHỆ THÔNG TIN

🙞 **🕮** 🙜

****

**ĐỒ ÁN TỐT NGHIỆP**

***Đề tài:***

**XÂY DỰNG APP ĐẶT PHÒNG**

**VÀ QUẢN LÝ KHÁCH SẠN TRÊN ANDROID NATIVE**

**HN, 2025**



**TRƯỜNG ĐẠI HỌC PHƯƠNG ĐÔNG**

KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

NGÀNH CÔNG NGHỆ THÔNG TIN

🙞 **🕮** 🙜



**ĐỒ ÁN TỐT NGHIỆP**

***Đề tài:***

**XÂY DỰNG APP ĐẶT PHÒNG VÀ**

**QUẢN LÝ KHÁCH SẠN**

**TRÊN ANDROID NATIVE**

**Giáo viên hướng dẫn: Ths. Nguyễn Mười Phương**

**Sinh viên thực hiện: Trần Trung Dũng**

**Mã sinh viên: 521100139**

**HN, 2025**

**MỤC LỤC**

[LỜI NÓI ĐẦU 1](#_Toc197915375)

[CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG KHÁCH SẠN 2](#_Toc197915376)

[**1.1 Khái niệm chung về hệ thống khách sạn 2**](#_Toc197915377)

[**1.1.1 Các khái niệm về hệ thống quản lý khách sạn 2**](#_Toc197915378)

[**1.1.2 Các thành phần và mô thình của hệ thống quản lý khách sạn 3**](#_Toc197915379)

[**1.2 Đặc điểm của hệ thống quản lý khách sạn đơn lẻ 5**](#_Toc197915380)

[**1.3 Lợi ích của hệ thống quản lý khách sạn 7**](#_Toc197915381)

[**1.3.1 Lợi ích đối với khách sạn 7**](#_Toc197915382)

[**1.3.2 Lợi ích với khách hàng 10**](#_Toc197915383)

[**1.4 Thách thức và hạn chế của hệ thống quản lý khách sạn 13**](#_Toc197915384)

[**1.4.1 Hạn chế mang tính kỹ thuật 13**](#_Toc197915385)

[**1.4.2 Hạn chế mang tính vận hành và quản lý 14**](#_Toc197915386)

[Chương 2. LỰA CHỌN CÔNG NGHỆ 16](#_Toc197915387)

[**2.1 HTML 16**](#_Toc197915388)

[**2.1.1 Khái quát về HTML 16**](#_Toc197915389)

[**2.1.2 Ưu điểm và nhược điểm của HTML 16**](#_Toc197915390)

[**2.2 CSS 17**](#_Toc197915391)

[**2.2.1 Khái quát về CSS 17**](#_Toc197915392)

[**2.2.2 CSS hoạt động như thế nào? 17**](#_Toc197915393)

[**2.3 Javascript 18**](#_Toc197915394)

[**2.3.1 Khái quát về Javascript 18**](#_Toc197915395)

[**2.3.2 Ưu điểm, nhược điểm của Javascript 18**](#_Toc197915396)

[**2.4 View Engine với EJS 19**](#_Toc197915397)

[**2.4.1 Khái quát về EJS 19**](#_Toc197915398)

[**2.4.2 Ưu điểm, nhược điểm của EJS 19**](#_Toc197915399)

[**2.5 Framework Nodejs 20**](#_Toc197915400)

[**2.5.1 Khái quát về Nodejs và Express 20**](#_Toc197915401)

[**2.5.2 Vòng đời 22**](#_Toc197915402)

[**2.6 Firebase 22**](#_Toc197915403)

[**2.6.1 Khái quát về Firebase 22**](#_Toc197915404)

[**2.6.2 Các dịch vụ chính của Firebase 23**](#_Toc197915405)

[**2.6.3 Tích hợp Firebase với Node.js 24**](#_Toc197915406)

[**2.7 Android Native và Kotlin 25**](#_Toc197915407)

[**2.7.1. Android Native (Ứng dụng gốc Android) 25**](#_Toc197915408)

[**2.7.2. Kotlin 26**](#_Toc197915409)

[Chương 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 28](#_Toc197915410)

[**3.1 Khảo sát hiện trạng 28**](#_Toc197915411)

[**3.1.1 Mô tả đề tài 28**](#_Toc197915412)

[**3.1.2 Chức năng hệ thống 29**](#_Toc197915413)

[**3.2 Phân tích và thiết kế hệ thống 31**](#_Toc197915414)

[**3.2.1 Mô tả nghiệp vụ 31**](#_Toc197915415)

[**3.2.2 Đối với quản trị viên (QTV/Admin) 33**](#_Toc197915416)

[**3.3 Xây dựng biểu đồ usecase 34**](#_Toc197915417)

[**3.3.1 Biểu đồ Usecase tổng quát 34**](#_Toc197915418)

[**3.3.2 Use case đăng nhập Admin 35**](#_Toc197915419)

[**3.3.3 Use case quản lý phòng 36**](#_Toc197915420)

[**3.3.4 Use case quản lý loại phòng 40**](#_Toc197915421)

[**3.3.5 Use case quản lý tiện nghi 43**](#_Toc197915422)

[**3.3.6 Use case quản lý order 45**](#_Toc197915423)

[**3.3.7 Use case quản lý order items 48**](#_Toc197915424)

[**3.3.8 Usecase quản lý bookings 50**](#_Toc197915425)

[**3.3.9 Use case quản lý khách hàng 53**](#_Toc197915426)

[**3.3.10 Use case quản lý khướu nại 57**](#_Toc197915427)

[**3.3.11 Use case quản lý nhân viên 59**](#_Toc197915428)

[**3.3.12 Use case xem phòng 62**](#_Toc197915429)

[**3.3.13 Use case đăng nhập khách hàng 65**](#_Toc197915430)

[**3.3.14 Usecase đặt phòng 66**](#_Toc197915431)

[**3.3.15 Use case đặt đồ ăn/ dịch vụ 68**](#_Toc197915432)

[**3.3.16 Use case quản lý thông tin cá nhân 71**](#_Toc197915433)

[**3.3.17 Use case quản lý bookings 74**](#_Toc197915434)

[**3.3.18 Use case quản lý order 76**](#_Toc197915435)

[**3.4 Xây dựng biểu đồ lớp 79**](#_Toc197915436)

[**3.5 Biểu đồ hoạt động 80**](#_Toc197915437)

[**3.5.1 Biểu đồ hoạt động khách hàng 80**](#_Toc197915438)

[**3.5.2 Biểu đồ hoạt động quản trị viên 82**](#_Toc197915439)

[**3.6 Thiết kế 85**](#_Toc197915440)

[**3.6.1 Biểu đồ tuần tự 85**](#_Toc197915441)

[**3.6.2 Thiết kế cơ sở dữ liệu 87**](#_Toc197915442)

[Chương 4. DEMO CHƯƠNG TRÌNH 96](#_Toc197915443)

[**4.1 Giao diện khách hàng(+App/Web) 96**](#_Toc197915444)

[**4.1.1 Các thanh cấu tạo nên trang khách hàng 96**](#_Toc197915445)

[**4.1.2 Giao diện trang danh sách phòng 96**](#_Toc197915446)

[**4.1.3 Giao diện trang chi tiết phòng 97**](#_Toc197915447)

[**4.1.4 Giao diện trang đặt phòng 99**](#_Toc197915448)

[**4.1.5 Giao diện thông báo đặt phòng thành công(Hóa đơn) 101**](#_Toc197915449)

[**4.1.6 Giao diện trang đăng nhập/ đăng ký 102**](#_Toc197915450)

[**4.1.7 Giao diện trang my booking 102**](#_Toc197915451)

[**4.1.8 Giao diện menu order 103**](#_Toc197915452)

[**4.2 Giao diện trang quản trị(Web) 104**](#_Toc197915453)

[**4.2.1 Giao diện trang admin 104**](#_Toc197915454)

[**4.2.2 Giao diện trang quản lý booking 106**](#_Toc197915455)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 107](#_Toc197915456)

[**1. Kết luận 107**](#_Toc197915457)

[**2. Hướng phát triển 109**](#_Toc197915458)

[TÀI LIỆU THAM KHẢO 111](#_Toc197915459)

[LỜI CAM ĐOAN 112](#_Toc197915460)

# LỜI NÓI ĐẦU

Trong nền kinh tế hiện đại, sự phát triển mạnh mẽ của công nghệ thông tin đã mang lại những thay đổi căn bản trong phương thức quản lý và vận hành doanh nghiệp, đặc biệt trong ngành dịch vụ khách sạn và du lịch. Ngày nay, hầu hết các khách sạn trên thế giới, từ quy mô nhỏ đến các chuỗi khách sạn lớn, đều ứng dụng công nghệ thông tin vào hoạt động kinh doanh với các mức độ và hình thức khác nhau. Nhiều hệ thống quản lý khách sạn hiện đại đã ra đời nhằm tối ưu hóa quy trình vận hành, nâng cao chất lượng dịch vụ và gia tăng trải nghiệm khách hàng.

Việc ứng dụng công nghệ số vào quản lý khách sạn đang được các doanh nghiệp trong ngành lưu trú quan tâm đầu tư phát triển. Đến nay, có nhiều lý do thuyết phục để các khách sạn tích hợp hệ thống quản lý trực tuyến vào hoạt động kinh doanh nhằm tăng tính cạnh tranh, đáp ứng nhu cầu khách hàng và phát triển bền vững trong bối cảnh số hóa toàn cầu. Tại Việt Nam, ngành khách sạn đang dần chuyển đổi từ mô hình quản lý truyền thống sang mô hình quản lý số hóa, trong đó các hệ thống đặt phòng trực tuyến, quản lý khách hàng và dịch vụ phòng chiếm vị trí quan trọng và có tiềm năng phát triển rất lớn.

Việc xây dựng một website quản lý khách sạn toàn diện không chỉ giúp tiếp cận được số lượng lớn khách hàng mà còn tạo ra sự thuận tiện trong việc quản lý hoạt động nội bộ, từ đặt phòng, quản lý nhân sự đến cung cấp dịch vụ ăn uống và các tiện ích khác. Chính vì vậy, em đã lựa chọn đề tài "Xây dựng hệ thống quản lý khách sạn Lembo's Luxurious Hotel" làm đồ án của mình.

Đồ án gồm các chương:

Chương 1: Tổng quan về hệ thống khách sạn

Chương 2: Lựa chọn công nghệ

Chương 3: Phân tích thiết kế hệ thống

Chương 4: Demo chương trình

Em xin chân thành cảm ơn sự hướng dẫn, và giúp đỡ tận tình của Ths. Nguyễn Mười Phương, các Thầy Cô trong khoa Công nghệ thông tin Trường Đại học Phương Đông đã giúp đỡ em trong quá trình học tập cũng như trong quá trình làm đồ án.

CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG KHÁCH SẠN

* 1. Khái niệm chung về hệ thống khách sạn
     1. Các khái niệm về hệ thống quản lý khách sạn

Hệ thống quản lý khách sạn (Hotel Management System - HMS) được hiểu theo nhiều cách khác nhau tùy thuộc vào góc độ tiếp cận. Dưới đây là một số định nghĩa phổ biến:

Theo quan điểm công nghệ thông tin, hệ thống quản lý khách sạn là một phần mềm ứng dụng tích hợp các chức năng quản lý hoạt động kinh doanh của khách sạn từ đặt phòng, nhận phòng, quản lý phòng, dịch vụ ăn uống, dịch vụ bổ sung, cho đến thanh toán và quản lý khách hàng.

Theo quan điểm quản lý, hệ thống quản lý khách sạn là tập hợp các quy trình, công cụ và phương pháp được sử dụng để điều hành hiệu quả một cơ sở lưu trú, bao gồm việc quản lý nhân sự, tài chính, cơ sở vật chất, dịch vụ khách hàng và các hoạt động khác.

Theo quan điểm vận hành, hệ thống quản lý khách sạn là nền tảng trung tâm kết nối tất cả các bộ phận chức năng của khách sạn nhằm đảm bảo sự vận hành trơn tru, hiệu quả và mang lại trải nghiệm tốt nhất cho khách hàng.

Theo Hiệp hội Khách sạn Quốc tế (International Hotel Association - IHA), hệ thống quản lý khách sạn được định nghĩa là "hệ thống tổng thể bao gồm các phần mềm, phần cứng, quy trình và con người, được thiết kế để tối ưu hóa việc quản lý và vận hành tất cả các khía cạnh của một cơ sở lưu trú từ lễ tân, đặt phòng, quản lý tài sản, kế toán, nhân sự đến dịch vụ khách hàng."

Tổ chức Du lịch Thế giới (UNWTO) mô tả hệ thống quản lý khách sạn là "nền tảng kỹ thuật số tích hợp cho phép quản lý hiệu quả tất cả các khía cạnh của hoạt động khách sạn, từ đặt phòng và quản lý phòng, đến quản lý kho, tài chính, nhân sự và các dịch vụ khách hàng."

Theo nghĩa rộng, hệ thống quản lý khách sạn bao gồm toàn bộ các quy trình, công nghệ và con người tham gia vào việc điều hành, kiểm soát và phát triển dịch vụ lưu trú, từ khâu tiếp thị, bán hàng, đặt phòng, vận hành đến chăm sóc khách hàng và quản lý sau bán hàng.

Trong bối cảnh của Lembo's Luxurious Hotel, hệ thống quản lý khách sạn được xây dựng là một hệ thống toàn diện tích hợp các công nghệ hiện đại nhằm tối ưu hóa trải nghiệm của khách hàng và nâng cao hiệu quả vận hành, đồng thời mang lại lợi thế cạnh tranh trong thị trường khách sạn cao cấp.

* + 1. Các thành phần và mô thình của hệ thống quản lý khách sạn

Hệ thống quản lý khách sạn Lembo's Luxurious Hotel được xây dựng theo mô hình tích hợp toàn diện, bao gồm các thành phần chính sau:

**1. Hệ thống quản lý phòng (Room Management System - RMS):**

* Quản lý trạng thái phòng (trống, đã đặt, đang sử dụng, đang dọn dẹp, bảo trì)
* Quản lý loại phòng và giá phòng theo mùa, sự kiện
* Quản lý tiện nghi phòng
* Lập kế hoạch bảo trì và vệ sinh phòng
* Giám sát chất lượng phòng và tiêu chuẩn dịch vụ

**2. Hệ thống đặt phòng (Reservation System):**

* Quản lý đặt phòng trực tuyến và trực tiếp
* Tích hợp với các kênh đặt phòng (OTA) như Booking.com, Expedia
* Xác nhận đặt phòng tự động
* Quản lý hủy/thay đổi đặt phòng
* Lập kế hoạch công suất phòng

**3. Hệ thống lễ tân (Front Desk System):**

* Đăng ký nhận phòng (check-in) và trả phòng (check-out)
* Quản lý thông tin khách hàng
* Xử lý yêu cầu đặc biệt của khách
* Phân công phòng và quản lý chìa khóa/thẻ từ
* Xử lý các vấn đề phát sinh tại quầy lễ tân

**4. Hệ thống dịch vụ phòng và ẩm thực (Room Service & F&B System):**

* Quản lý đơn hàng dịch vụ phòng
* Quản lý nhà hàng và quầy bar
* Quản lý kho thực phẩm và đồ uống
* Lập kế hoạch thực đơn và chuẩn bị bữa ăn
* Quản lý tiệc và sự kiện

**5. Hệ thống thanh toán (Payment System):**

* Quản lý hóa đơn và chi phí
* Xử lý thanh toán bằng nhiều phương thức (tiền mặt, thẻ tín dụng, chuyển khoản)
* Tích hợp với hệ thống kế toán
* Quản lý thuế và báo cáo tài chính
* Xử lý hoàn tiền và điều chỉnh hóa đơn

**6. Hệ thống quản lý quan hệ khách hàng (CRM):**

* Quản lý hồ sơ khách hàng
* Theo dõi lịch sử lưu trú và chi tiêu
* Chương trình khách hàng thân thiết
* Quản lý phản hồi và đánh giá của khách
* Chiến dịch tiếp thị và chăm sóc khách hàng

**7. Hệ thống quản lý nhân sự (Human Resource Management System):**

* Quản lý thông tin nhân viên
* Lập lịch và phân công công việc
* Đào tạo và phát triển nhân viên
* Đánh giá hiệu suất công việc
* Quản lý lương và phúc lợi

**8. Hệ thống báo cáo và phân tích (Reporting & Analytics System):**

* Báo cáo doanh thu và lợi nhuận
* Phân tích xu hướng đặt phòng
* Báo cáo công suất sử dụng phòng
* Phân tích phản hồi của khách hàng
* Báo cáo hiệu suất nhân viên

**Mô hình tích hợp của Lembo's Luxurious Hotel:**

Hệ thống quản lý khách sạn Lembo's Luxurious Hotel được thiết kế theo mô hình tích hợp trung tâm dữ liệu (Centralized Data Integration Model), trong đó tất cả các hệ thống con đều kết nối và chia sẻ dữ liệu với nhau thông qua một cơ sở dữ liệu trung tâm. Mô hình này cho phép:

1. **Tích hợp liền mạch**: Thông tin từ tất cả các bộ phận được đồng bộ hóa thời gian thực, đảm bảo tính nhất quán của dữ liệu.
2. **Truy cập phân quyền**: Nhân viên ở các bộ phận khác nhau có thể truy cập vào các phần thông tin cần thiết cho công việc của họ dựa trên quyền được cấp.
3. **Quản lý tập trung**: Ban quản lý có thể giám sát tất cả các hoạt động và truy cập báo cáo tổng hợp từ tất cả các bộ phận.
4. **Đa nền tảng**: Hệ thống có thể được truy cập từ nhiều thiết bị khác nhau như máy tính để bàn, máy tính bảng, điện thoại thông minh.
5. **Mở rộng linh hoạt**: Có thể dễ dàng thêm các module hoặc tính năng mới khi nhu cầu phát triển.
   1. Đặc điểm của hệ thống quản lý khách sạn đơn lẻ

Hệ thống quản lý khách sạn Lembo's Luxurious Hotel mang những đặc điểm nổi bật của một hệ thống hiện đại:

- Tính cá nhân hóa cao

Hệ thống có khả năng nhận diện và ghi nhớ sở thích cá nhân của từng khách hàng, từ loại phòng ưa thích, dịch vụ thường sử dụng, đến các yêu cầu đặc biệt. Thông qua phân tích dữ liệu, hệ thống có thể đề xuất các dịch vụ phù hợp với từng khách hàng, tạo ra trải nghiệm riêng biệt cho mỗi người. Ví dụ, khi một khách hàng thường xuyên đặt dịch vụ spa đăng nhập vào hệ thống, họ sẽ nhận được các đề xuất về các liệu trình spa mới hoặc khung giờ phổ biến.

- Đáp ứng tức thời

Hệ thống được thiết kế để xử lý các yêu cầu của khách hàng trong thời gian thực, giảm thiểu thời gian chờ đợi và nâng cao trải nghiệm khách hàng. Các yêu cầu dịch vụ phòng, đặt bàn nhà hàng, yêu cầu dịch vụ đưa đón được xử lý ngay lập tức thông qua ứng dụng di động hoặc hệ thống trong phòng. Thời gian phản hồi trung bình cho các yêu cầu không vượt quá 5 phút, đảm bảo sự hài lòng của khách hàng.

- Giá cả linh hoạt

Hệ thống cho phép áp dụng chiến lược giá động (dynamic pricing) dựa trên nhiều yếu tố như mùa du lịch, tỷ lệ lấp đầy phòng, sự kiện đặc biệt và lịch sử của khách hàng. Khả năng này giúp tối ưu hóa doanh thu và cung cấp các ưu đãi cá nhân hóa cho khách hàng. Ví dụ, trong mùa thấp điểm, hệ thống có thể tự động giảm giá phòng hoặc đề xuất các gói khuyến mãi đặc biệt để kích thích đặt phòng.

- Đáp ứng mọi nơi, mọi lúc

Khách hàng có thể tương tác với hệ thống 24/7 thông qua nhiều kênh như website, ứng dụng di động, chatbot, hoặc qua điện thoại. Hệ thống hỗ trợ đa ngôn ngữ và có khả năng xử lý đặt phòng, yêu cầu dịch vụ, và thanh toán tại bất kỳ thời điểm nào, từ bất kỳ nơi nào trên thế giới. Điều này đặc biệt quan trọng đối với khách du lịch quốc tế ở các múi giờ khác nhau.

- Tích hợp trí tuệ nhân tạo và học máy

Hệ thống sử dụng các thuật toán AI và học máy để phân tích hành vi khách hàng, dự đoán nhu cầu, và tối ưu hóa hoạt động. Ví dụ, hệ thống có thể dự đoán công suất phòng trong các kỳ nghỉ lễ, đề xuất điều chỉnh giá phòng, hoặc gợi ý các dịch vụ bổ sung dựa trên hồ sơ khách hàng. AI cũng được sử dụng trong chatbot để trả lời các câu hỏi thường gặp của khách hàng một cách tức thời.

- Kết nối Internet vạn vật (IoT)

Các thiết bị thông minh trong phòng như điều hòa, đèn, TV, rèm cửa được kết nối với hệ thống trung tâm, cho phép khách hàng điều khiển thông qua ứng dụng di động hoặc bảng điều khiển trong phòng. Hệ thống IoT cũng giúp giám sát tiêu thụ năng lượng, phát hiện sự cố kỹ thuật sớm, và tối ưu hóa bảo trì. Ví dụ, khi khách hàng rời khỏi phòng, các thiết bị có thể tự động chuyển sang chế độ tiết kiệm năng lượng.

- Phân tích dữ liệu lớn

Hệ thống thu thập và phân tích khối lượng lớn dữ liệu về hành vi khách hàng, lịch sử đặt phòng, phản hồi, và xu hướng thị trường để đưa ra các quyết định kinh doanh thông minh. Các dashboard trực quan giúp ban quản lý nắm bắt tình hình hoạt động của khách sạn, xác định cơ hội cải tiến, và phát triển chiến lược kinh doanh hiệu quả.

- Tích hợp với các nền tảng xã hội và đánh giá

Hệ thống tích hợp với các nền tảng xã hội và website đánh giá như TripAdvisor, Booking.com, Google Reviews để theo dõi và phản hồi nhanh chóng các đánh giá của khách hàng. Điều này không chỉ giúp cải thiện hình ảnh thương hiệu mà còn cung cấp thông tin quý giá để nâng cao chất lượng dịch vụ.

- Bảo mật đa lớp

Hệ thống được bảo vệ bởi nhiều lớp bảo mật, bao gồm mã hóa dữ liệu, xác thực đa yếu tố, kiểm soát truy cập dựa trên vai trò, và giám sát liên tục để phát hiện các hoạt động đáng ngờ. Dữ liệu khách hàng nhạy cảm như thông tin thanh toán được bảo vệ theo các tiêu chuẩn bảo mật cao nhất như PCI DSS.

- Tối ưu hóa vận hành thông qua tự động hóa

Nhiều quy trình vận hành được tự động hóa như phân công nhân viên dọn phòng, cập nhật trạng thái phòng, gửi email xác nhận đặt phòng, nhắc nhở check-out, và tạo báo cáo. Điều này giúp giảm thiểu lỗi do con người, tăng hiệu quả vận hành, và cho phép nhân viên tập trung vào việc phục vụ khách hàng.

* 1. Lợi ích của hệ thống quản lý khách sạn
     1. Lợi ích đối với khách sạn

- Tối ưu hóa doanh thu và lợi nhuận

* Áp dụng chiến lược giá động dựa trên cầu thị trường, mùa vụ và các sự kiện
* Tăng doanh thu từ việc bán thêm dịch vụ (upselling) và bán chéo (cross-selling)
* Giảm chi phí vận hành thông qua tự động hóa quy trình
* Tối ưu hóa công suất phòng với hệ thống dự báo nhu cầu thông minh
* Giảm tỷ lệ hủy phòng và không đến (no-show) thông qua chính sách đặt cọc và nhắc nhở tự động

- Nâng cao hiệu quả vận hành

* Tự động hóa các quy trình thủ công, giảm khối lượng công việc hành chính
* Tối ưu hóa lịch trình nhân viên dựa trên dự báo công suất
* Quản lý kho hàng và vật tư hiệu quả, giảm lãng phí
* Giảm thời gian check-in/check-out với các quy trình kỹ thuật số
* Cải thiện truyền thông nội bộ giữa các bộ phận

- Cải thiện trải nghiệm khách hàng

* Cung cấp dịch vụ cá nhân hóa dựa trên lịch sử và sở thích của khách
* Đáp ứng yêu cầu khách hàng nhanh chóng và chính xác
* Giảm thời gian chờ đợi tại quầy lễ tân với check-in/check-out tự động
* Cung cấp nhiều kênh liên lạc và hỗ trợ
* Ghi nhớ và dự đoán nhu cầu của khách hàng thường xuyên

- Ra quyết định dựa trên dữ liệu

* Phân tích thời gian thực về hiệu suất kinh doanh
* Nhận biết xu hướng thị trường và điều chỉnh chiến lược
* Đánh giá hiệu quả của các chiến dịch tiếp thị
* Tối ưu hóa phân bổ nguồn lực dựa trên dữ liệu lịch sử
* Dự báo nhu cầu mùa vụ và lập kế hoạch nhân sự phù hợp

- Tăng cường bảo mật và tuân thủ quy định

* Bảo vệ dữ liệu khách hàng với các biện pháp bảo mật tiên tiến
* Tuân thủ các quy định về bảo vệ dữ liệu như GDPR
* Theo dõi và kiểm soát truy cập vào hệ thống
* Lưu trữ hồ sơ khách hàng an toàn và tuân thủ pháp luật
* Kiểm toán tự động để đảm bảo tuân thủ quy định tài chính

- Quản lý danh tiếng trực tuyến

* Theo dõi và phản hồi đánh giá của khách hàng trên các nền tảng
* Phân tích sentiments (cảm xúc) từ phản hồi của khách
* Cải thiện dịch vụ dựa trên phản hồi thực tế
* Xây dựng và duy trì hình ảnh thương hiệu tích cực
* Tạo ra người ủng hộ thương hiệu từ khách hàng hài lòng

- Tiết kiệm chi phí dài hạn

* Giảm chi phí nhân sự thông qua tự động hóa
* Tối ưu hóa tiêu thụ năng lượng với các hệ thống thông minh
* Giảm chi phí hành chính và giấy tờ
* Kéo dài tuổi thọ của trang thiết bị với bảo trì dự phòng
* Giảm chi phí đào tạo với hệ thống trực quan, dễ sử dụng

- Tạo lợi thế cạnh tranh

* Cung cấp dịch vụ hiện đại và tiện lợi hơn đối thủ
* Phản ứng nhanh với thay đổi thị trường và nhu cầu khách hàng
* Phát triển các gói dịch vụ độc đáo dựa trên dữ liệu khách hàng
* Duy trì mối quan hệ lâu dài với khách hàng trung thành
* Xây dựng danh tiếng cho sự đổi mới và chất lượng dịch vụ

- Mở rộng kênh phân phối

* Tích hợp với các OTA (Online Travel Agencies) và kênh phân phối khác
* Quản lý hiệu quả tính khả dụng phòng trên nhiều nền tảng
* Cập nhật giá và tình trạng phòng thời gian thực
* Phát triển và tối ưu hóa kênh đặt phòng trực tiếp
* Thiết lập quan hệ đối tác chiến lược với các công ty du lịch

- Phát triển bền vững

* Giảm sử dụng giấy với quy trình kỹ thuật số
* Tối ưu hóa tiêu thụ năng lượng và nước
* Giảm thiểu chất thải thông qua quản lý kho hiệu quả
* Theo dõi và cải thiện dấu chân carbon
* Hỗ trợ các sáng kiến du lịch bền vững
  + 1. Lợi ích với khách hàng

- Vượt giới hạn về không gian và thời gian: Thương mại điện tử cho phép khách hàng mua sắm mọi nơi, mọi lúc đối với các cửa hàng trên khắp thế giới

- Đặt phòng và thanh toán thuận tiện

* Đặt phòng trực tuyến 24/7 từ bất kỳ thiết bị nào
* Nhiều phương thức thanh toán an toàn và bảo mật
* Xác nhận đặt phòng tức thì và theo dõi trạng thái đặt phòng
* Khả năng thay đổi hoặc hủy đặt phòng trực tuyến
* Hệ thống giữ chỗ linh hoạt và minh bạch

- Check-in/check-out nhanh chóng

* Check-in trực tuyến trước khi đến
* Nhận phòng nhanh chóng với hệ thống kiosk tự phục vụ
* Check-out trên thiết bị di động không cần xếp hàng
* Nhận hóa đơn điện tử tức thì
* Khả năng gia hạn thời gian trả phòng thông qua ứng dụng

- Trải nghiệm cá nhân hóa

* Các dịch vụ được đề xuất dựa trên sở thích cá nhân
* Ghi nhớ yêu cầu đặc biệt (tầng cao, phòng không hút thuốc, gối đặc biệt)
* Chào đón cá nhân hóa với tên và ưu đãi đặc biệt
* Đề xuất các hoạt động và dịch vụ phù hợp với sở thích
* Tùy chỉnh trải nghiệm trong phòng (nhiệt độ, ánh sáng, giải trí)

- Tiếp cận thông tin đầy đủ

* Thông tin chi tiết về loại phòng, tiện nghi và dịch vụ
* Hình ảnh và video thực tế của phòng và khu vực công cộng
* Thông tin về các dịch vụ bổ sung và giá cả
* Thông tin điểm đến và các hoạt động du lịch
* Câu hỏi thường gặp và hỗ trợ trực tuyến

- Điều khiển phòng thông minh

* Điều khiển ánh sáng, nhiệt độ và rèm cửa thông qua ứng dụng
* Điều khiển hệ thống giải trí từ thiết bị cá nhân
* Gọi dịch vụ phòng thông qua ứng dụng hoặc thiết bị trong phòng
* Đặt lịch dịch vụ như dọn phòng, massage, đưa đón
* Nhận thông báo và tin nhắn từ khách sạn

- Tích lũy và sử dụng điểm thưởng

* Tham gia chương trình khách hàng thân thiết dễ dàng
* Theo dõi điểm thưởng và quyền lợi trực tuyến
* Sử dụng điểm thưởng cho các dịch vụ bổ sung hoặc giảm giá phòng
* Nhận ưu đãi đặc biệt dành riêng cho thành viên
* Thăng hạng dựa trên lịch sử lưu trú và chi tiêu

- Hỗ trợ đa ngôn ngữ

* Giao diện người dùng hỗ trợ nhiều ngôn ngữ
* Nhân viên chăm sóc khách hàng đa ngôn ngữ trực tuyến
* Dịch vụ phiên dịch theo yêu cầu
* Thông tin và hướng dẫn bằng ngôn ngữ bản địa
* Hỗ trợ thanh toán bằng nhiều loại tiền tệ

- Kết nối với cộng đồng du lịch

* Chia sẻ đánh giá và trải nghiệm với cộng đồng
* Tham gia diễn đàn hoặc nhóm khách hàng
* Nhận đề xuất từ khách hàng khác
* Tham gia các sự kiện xã hội tại khách sạn
* Kết nối với các doanh nghiệp địa phương

- Khả năng tiếp cận cho người khuyết tật

* Thông tin chi tiết về khả năng tiếp cận của phòng và cơ sở vật chất
* Đặt phòng và dịch vụ phù hợp với nhu cầu đặc biệt
* Hỗ trợ cho người khiếm thị hoặc khiếm thính
* Thông báo và hướng dẫn an toàn phù hợp
* Dịch vụ hỗ trợ cá nhân khi cần thiết

- Liền mạch với trải nghiệm du lịch tổng thể

* Tích hợp với các dịch vụ đặt vé máy bay, thuê xe
* Đề xuất và đặt tour du lịch địa phương
* Thông tin về giao thông công cộng và điểm tham quan
* Dịch vụ đưa đón sân bay và xe đưa đón
* Hỗ trợ trong trường hợp khẩn cấp hoặc thay đổi lịch trình
  1. Thách thức và hạn chế của hệ thống quản lý khách sạn
     1. Hạn chế mang tính kỹ thuật

**- Vấn đề bảo mật và an toàn dữ liệu** Trong bối cảnh các cuộc tấn công mạng ngày càng tinh vi, việc bảo vệ thông tin nhạy cảm của khách hàng như thông tin cá nhân, chi tiết thẻ tín dụng, và lịch trình du lịch là một thách thức lớn. Hệ thống quản lý khách sạn phải đối mặt với nguy cơ từ các cuộc tấn công ransomware, phishing, và đánh cắp dữ liệu. Mặc dù Lembo's Luxurious Hotel đã triển khai hệ thống bảo mật đa lớp, nguy cơ từ lỗ hổng zero-day hoặc lỗi của nhân viên vẫn luôn tồn tại.

**- Tính toàn vẹn dữ liệu** Với hàng nghìn giao dịch mỗi ngày, việc đảm bảo tính toàn vẹn và nhất quán của dữ liệu là một thách thức lớn. Sự cố mất kết nối, xung đột đồng bộ hóa, hoặc lỗi phần mềm có thể dẫn đến mất mát hoặc sai lệch dữ liệu quan trọng như đặt phòng, thanh toán, hoặc yêu cầu đặc biệt của khách. Điều này có thể gây ra những trải nghiệm tiêu cực cho khách hàng và ảnh hưởng đến uy tín của khách sạn.

**- Khả năng mở rộng hệ thống** Khi lượng khách tăng đột biến vào mùa cao điểm hoặc sự kiện lớn, hệ thống có thể gặp khó khăn trong việc xử lý khối lượng truy cập lớn, dẫn đến tình trạng trễ hoặc sập hệ thống. Việc nâng cấp và mở rộng hệ thống đòi hỏi đầu tư đáng kể về cơ sở hạ tầng CNTT, nhất là khi cần duy trì hoạt động liên tục của khách sạn. Lembo's Luxurious Hotel phải cân nhắc giữa chi phí đầu tư và hiệu suất hệ thống.

**- Tương thích với hệ thống bên thứ ba** Hệ thống quản lý khách sạn thường cần tích hợp với nhiều hệ thống bên thứ ba như các OTA (Booking.com, Expedia), cổng thanh toán, hệ thống POS nhà hàng, và hệ thống an ninh. Việc đảm bảo tính tương thích và đồng bộ hóa giữa các hệ thống này là một thách thức kỹ thuật lớn, đặc biệt khi các nhà cung cấp bên thứ ba cập nhật API hoặc thay đổi giao thức.

**- Độ tin cậy và thời gian hoạt động** Hệ thống quản lý khách sạn phải hoạt động 24/7/365 với độ tin cậy cao. Bất kỳ thời gian ngừng hoạt động nào cũng có thể gây ra tác động nghiêm trọng đến hoạt động của khách sạn và trải nghiệm của khách. Việc triển khai các giải pháp dự phòng, khôi phục sau thảm họa, và cân bằng tải là cần thiết nhưng làm tăng chi phí và độ phức tạp của hệ thống.

**- Phụ thuộc vào kết nối internet** Hệ thống dựa nhiều vào kết nối internet ổn định và tốc độ cao. Bất kỳ sự cố về kết nối như đứt cáp, tắc nghẽn mạng, hoặc tấn công DDoS đều có thể làm gián đoạn hoạt động của khách sạn. Mặc dù có các giải pháp dự phòng như kết nối dự phòng và chế độ ngoại tuyến, việc vận hành đầy đủ các tính năng vẫn phụ thuộc lớn vào chất lượng kết nối internet.

**- Khả năng thích ứng với công nghệ mới** Công nghệ phát triển nhanh chóng đòi hỏi hệ thống phải liên tục cập nhật để tích hợp các xu hướng mới như AI, blockchain, IoT, hoặc thực tế ảo tăng cường. Việc này đòi hỏi đầu tư liên tục vào nghiên cứu và phát triển, cũng như đào tạo nhân viên, trong khi vẫn đảm bảo tính ổn định và bảo mật của hệ thống hiện tại.

**- Lỗi và thời gian tải trang** Hiệu suất ứng dụng là yếu tố quan trọng đối với trải nghiệm người dùng. Các trang web hoặc ứng dụng tải chậm, bị lỗi, hoặc giao diện không thân thiện có thể làm khách hàng thất vọng và chuyển sang đối thủ cạnh tranh. Việc tối ưu hóa hiệu suất đòi hỏi kiểm thử liên tục và điều chỉnh hệ thống, đặc biệt là với sự đa dạng của thiết bị và trình duyệt mà khách hàng sử dụng.

* + 1. Hạn chế mang tính vận hành và quản lý

**- Chi phí triển khai và bảo trì cao** Việc triển khai một hệ thống quản lý khách sạn toàn diện như tại Lembo's Luxurious Hotel đòi hỏi đầu tư ban đầu lớn cho phần mềm, phần cứng, và cơ sở hạ tầng. Ngoài ra, chi phí bảo trì, cập nhật, và hỗ trợ kỹ thuật liên tục cũng đáng kể. Đối với các khách sạn vừa và nhỏ, chi phí này có thể là rào cản lớn cho việc áp dụng công nghệ tiên tiến.

**- Đào tạo nhân viên và thay đổi quy trình làm việc** Việc áp dụng hệ thống mới đòi hỏi đào tạo toàn diện cho nhân viên ở tất cả các cấp. Nhân viên có thể gặp khó khăn trong việc thích nghi với công nghệ mới, đặc biệt là những người đã quen với quy trình làm việc truyền thống. Sự kháng cự đối với thay đổi và thời gian học tập có thể ảnh hưởng đến hiệu quả vận hành trong giai đoạn chuyển đổi.

**- Phụ thuộc vào nhà cung cấp** Khi đã triển khai một hệ thống cụ thể, khách sạn trở nên phụ thuộc vào nhà cung cấp đó cho các cập nhật, bảo trì, và hỗ trợ kỹ thuật. Nếu nhà cung cấp ngừng hỗ trợ, thay đổi mô hình kinh doanh, hoặc tăng giá đáng kể, khách sạn có thể gặp khó khăn trong việc duy trì hoặc thay đổi hệ thống.

**- Quá tải thông tin** Hệ thống hiện đại có khả năng thu thập và phân tích lượng lớn dữ liệu, nhưng điều này cũng có thể dẫn đến tình trạng quá tải thông tin cho các nhà quản lý. Việc xác định những dữ liệu thực sự quan trọng và chuyển đổi chúng thành thông tin hữu ích cho việc ra quyết định là một thách thức lớn.

**- Cân bằng giữa tự động hóa và dịch vụ cá nhân** Trong khi tự động hóa có thể tăng hiệu quả và giảm chi phí, quá phụ thuộc vào nó có thể làm mất đi yếu tố con người trong dịch vụ khách sạn - một yếu tố quan trọng đối với trải nghiệm khách hàng, đặc biệt trong phân khúc khách sạn cao cấp. Tìm ra sự cân bằng phù hợp giữa tự động hóa và tương tác cá nhân là một thách thức liên tục.

**- Tuân thủ quy định về dữ liệu và quyền riêng tư** Các quy định như GDPR ở châu Âu hoặc CCPA ở California đặt ra các yêu cầu nghiêm ngặt về cách thu thập, lưu trữ, và xử lý dữ liệu khách hàng. Việc tuân thủ các quy định này, đặc biệt khi hoạt động quốc tế, đòi hỏi kiến thức chuyên môn và có thể làm tăng chi phí và độ phức tạp của hệ thống.

**- Quản lý rủi ro và kế hoạch dự phòng** Với sự phụ thuộc ngày càng tăng vào hệ thống kỹ thuật số, việc xây dựng và duy trì các kế hoạch dự phòng hiệu quả cho các tình huống như mất điện, sự cố hệ thống, hoặc tấn công mạng trở nên quan trọng hơn bao giờ hết. Các kế hoạch này cần được kiểm thử thường xuyên và cập nhật để đảm bảo tính hiệu quả.

**- Khả năng đáp ứng thị trường thay đổi nhanh chóng** Ngành khách sạn và du lịch có thể thay đổi nhanh chóng do các yếu tố như xu hướng du lịch, tình hình kinh tế, hoặc sự kiện toàn cầu (như đại dịch COVID-19). Hệ thống quản lý khách sạn cần đủ linh hoạt để thích ứng với những thay đổi này, nhưng việc xây dựng tính linh hoạt vào hệ thống phức tạp là một thách thức lớn.

**- Đo lường ROI và hiệu quả hệ thống** Việc đo lường chính xác lợi nhuận đầu tư (ROI) từ hệ thống quản lý khách sạn có thể khó khăn, đặc biệt là các lợi ích phi tài chính như cải thiện trải nghiệm khách hàng hoặc hiệu quả vận hành. Thiếu các chỉ số rõ ràng có thể làm khó khăn cho việc biện minh cho đầu tư liên tục vào công nghệ.

**- Giữ cân bằng giữa đổi mới và ổn định** Trong khi đổi mới công nghệ là cần thiết để duy trì tính cạnh tranh, việc liên tục triển khai các tính năng hoặc hệ thống mới có thể gây gián đoạn hoạt động và tạo ra sự không ổn định. Tìm ra sự cân bằng giữa áp dụng công nghệ mới và duy trì sự ổn định trong vận hành là một thách thức lớn cho các nhà quản lý khách sạn.

1. LỰA CHỌN CÔNG NGHỆ
   1. HTML
      1. Khái quát về HTML

* Hyper Text: Hay Siêu văn bản đơn giản có nghĩa là “Văn bản trong Văn bản”. Một văn bản có một liên kết bên trong nó, là một siêu văn bản. Bất cứ khi nào bạn nhấp vào một liên kết đưa bạn đến một trang web mới, bạn đã nhấp vào một siêu văn bản. Siêu văn bản là một cách để liên kết hai hoặc nhiều trang web (tài liệu HTML) với nhau.
* Markup Language: hay ngôn ngữ đánh dấu là ngôn ngữ máy tính được sử dụng để áp dụng các quy ước về bố cục và định dạng cho tài liệu văn bản. Ngôn ngữ đánh dấu làm cho văn bản tương tác và năng động hơn. Nó có thể biến văn bản thành hình ảnh, bảng biểu, liên kết, v.v.
* Trang web: Trang web là một tài liệu thường được viết bằng HTML và được dịch bởi trình duyệt web. Một trang web có thể được xác định bằng cách nhập một URL. Một trang Web có thể thuộc loại tĩnh hoặc động. Chỉ với sự trợ giúp của HTML, chúng ta có thể tạo các trang web tĩnh.

Do đó, HTML là ngôn ngữ đánh dấu được sử dụng để tạo các trang web hấp dẫn với sự trợ giúp của kiểu dáng và có định dạng đẹp trên trình duyệt web. Một tài liệu HTML được tạo thành từ nhiều thẻ HTML và mỗi thẻ HTML chứa nội dung khác nhau.

* + 1. Ưu điểm và nhược điểm của HTML
* Ưu điểm
* HTML được ra đời từ rất lâu, do đó HTML có nguồn tài nguyên khổng lồ, hỗ trợ một cộng đồng người dùng lớn. Bên cạnh đó, cộng đồng HTML ngày càng phát triển trên thế giới.
* Mã nguồn của HTML là mã nguồn mở, do đó người dùng có thể sử dụng hoàn toàn miễn phí.
* HTML được sử dụng và được sử dụng trên nhiều trình duyệt được nhiều người dùng ưa chuộng hiện nay như Internet Explorer, Chrome, FireFox, Cốc cốc,…
* Học và tìm hiểu HTML đơn giản nên người học dễ dàng nắm được kiến thức và vận dụng trong xây dựng trang web căn bạn.
* HTML được thực hiện dễ dàng bởi HTML được tích hợp nhiều ngôn ngữ khác nhau như PHP, Java, NodeJs, Ruby,…Điều này sẽ giúp tạo thành một website hoàn chỉnh với nhiều tính năng.
* Nhược điểm
* Nhược điểm lớn nhất của HTML đó chính là chỉ có thể web tĩnh, web tĩnh có thể hiểu là những trang web chỉ hiện thông tin mà không có sự tương tác cho người dùng. Do đó, khi xây dựng tính năng động hoặc xây dựng hệ thống website có sự tương tác với người dùng, lập trình viên cần phải dùng thêm JavaScript hoặc ngôn ngữ backend của bên thứ ba.
* HTML thường chỉ có thể thực thi những thứ logic và cấu trúc nhất định, HTML không có khả năng tạo sự khác biệt và mới mẻ.
* Một số trình duyệt vẫn còn chậm trong viết hỗ trợ các phiên bản mới của HTML, đặc biệt là HTML5.
  1. CSS
     1. Khái quát về CSS
* **CSS** là chữ viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để **tìm và định dạng** lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu ([HTML](https://topdev.vn/blog/html-la-gi/)). Nói ngắn gọn hơn là ngôn ngữ tạo phong cách cho trang web. Bạn có thể hiểu đơn giản rằng, nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng,…thì CSS sẽ giúp chúng ta có thể thêm style vào các phần tử HTML đó như đổi bố cục, màu sắc trang, đổi màu chữ, font chữ, thay đổi cấu trúc…
* CSS hoạt động dựa trên phương thức là nó sẽ tìm dựa vào các vùng chọn, vùng chọn có thể là tên một thẻ HTML (tag), lớp (class), ID hay nhiều kiểu khác. Sau đó, nó sẽ áp dụng các thuộc tính cần thay đổi lên vùng chọn đó.
  + 1. CSS hoạt động như thế nào?

Khi một trình duyệt web (browser) hiển thị một tài liệu, nó phải kết hợp nội dung cùng với kiểu trình bày thông tin của tài liệu đó.Và đây là những gì sẽ xảy ra:

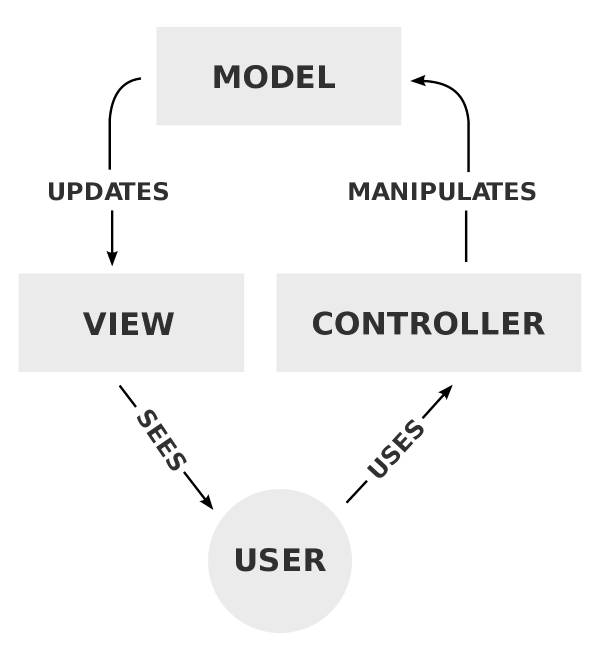
1. Trình duyệt sẽ tải HTML (ví dụ như: tải nó từ mạng)

2. Trình duyệt chuyển đổi HTML thành DOM (Mô hình đối tượng tài liệu). DOM đại diện cho tài liệu trong bộ nhớ của máy tính.

3. Sau đó, trình duyệt sẽ tìm nạp hầu hết các tài nguyên được liên kết với tài liệu HTML, chẳng hạn như hình ảnh và video được nhúng ... và CSS được liên kết.

4. Trình duyệt phân tích cú pháp CSS đã tìm nạp và sắp xếp các quy tắc khác nhau theo loại của chúng thành các "nhóm" khác nhau, ví dụ: phần tử, lớp, ID, v.v. Dựa trên các bộ chọn mà nó tìm thấy, trình duyệt sẽ tìm ra các quy tắc nên được áp dụng cho các nodes trong DOM

* 1. Javascript
     1. Khái quát về Javascript
* JavaScript là ngôn ngữ lập trình được nhà phát triển sử dụng để tạo trang web tương tác. Từ làm mới bảng tin trên trang mạng xã hội đến hiển thị hình ảnh động và bản đồ tương tác, các chức năng của JavaScript có thể cải thiện trải nghiệm người dùng của trang web. Là ngôn ngữ kịch bản phía máy khách, JavaScript là một trong những công nghệ cốt lõi của World Wide Web. Ví dụ: khi duyệt internet, bất cứ khi nào bạn thấy quảng cáo quay vòng dạng hình ảnh, menu thả xuống nhấp để hiển thị hoặc màu sắc phần tử thay đổi động trên trang web cũng chính là lúc bạn thấy các hiệu ứng của JavaScript.
  + 1. ***Ưu điểm, nhược điểm của Javascript***
* Ưu điểm
* Chương trình rất dễ học.
* Những lỗi Javascript rất dễ để phát hiện, từ đó giúp bạn sửa lỗi một cách nhanh chóng hơn.
* Những trình duyệt web có thể dịch thông qua HTML mà không cần sử dụng đến một compiler.
* JS có thể hoạt động ở trên nhiều nền tảng và các trình duyệt web khác nhau.
* Javascript còn có thể được gắn trên một số các element hoặc những events của các trang web.
* Nhược điểm
* Javascript dễ bị các hacker và scammer khai thác hơn.
* Javascript cũng không có khả năng đa luồng hoặc đa dạng xử lý.
* Có thể được dùng để thực thi những mã độc ở trên máy tính của người sử dụng.
* Những thiết bị khác nhau có thể sẽ thực hiện Javascript khác nhau, từ đó dẫn đến sự không đồng nhất.
  1. View Engine với EJS
     1. Khái quát về EJS
* EJS là từ viết tắt của Embedded JavaScript Templates (Các mẫu JavaScript nhúng). Đây là một công cụ tạo giao diện (view engine) phổ biến, cho phép nhúng mã JavaScript vào trong HTML để tạo ra các trang web động.
* Khi sử dụng EJS, các mẫu (templates) sẽ được xử lý trên máy chủ để từ đó sinh ra mã HTML thuần túy gửi đến client. Dựa vào cơ chế này, ứng dụng web của bạn có thể hiển thị nội dung động và tương tác với người dùng một cách linh hoạt.
* EJS hoạt động chặt chẽ với Node.js và đặc biệt là framework Express, giúp đơn giản hóa việc xây dựng giao diện người dùng trong các ứng dụng web hiện đại
  + 1. Ưu điểm, nhược điểm của EJS
* Ưu điểm
* **Tính đơn giản và thân thiện với người dùng**: EJS có cú pháp dễ học và dễ sử dụng, đặc biệt cho những người đã quen thuộc với HTML và JavaScript. Bạn chỉ cần thêm các thẻ đặc biệt <% %> để nhúng mã JavaScript vào HTML.
* **Cộng đồng hỗ trợ mạnh mẽ**: EJS được sử dụng rộng rãi trong cộng đồng Node.js, có nhiều tài liệu, ví dụ và diễn đàn hỗ trợ. Điều này giúp các nhà phát triển dễ dàng tìm kiếm giải pháp cho các vấn đề.
* **Hiệu suất tốt**: EJS cung cấp khả năng đệm (caching) cho các mẫu đã biên dịch, giúp ứng dụng chạy nhanh hơn và tiết kiệm tài nguyên máy chủ.
* **Tích hợp mượt mà với Express**: EJS được thiết kế để làm việc liền mạch với Express framework, giúp việc xây dựng ứng dụng web MVC (Model-View-Controller) trở nên đơn giản và nhất quán.
* **Hỗ trợ partial views và layouts**: EJS cho phép tạo các thành phần giao diện có thể tái sử dụng (partials) và bố cục chung (layouts), giúp cấu trúc code rõ ràng và dễ bảo trì
* Nhược điểm
* **Trộn lẫn logic và giao diện**: Mặc dù EJS cho phép nhúng JavaScript vào HTML, nhưng việc này có thể dẫn đến trộn lẫn logic xử lý và mã hiển thị, làm giảm tính bảo trì của ứng dụng nếu không được quản lý cẩn thận.
* **Thiếu các tính năng nâng cao**: So với một số view engine khác như Pug hoặc Handlebars, EJS có ít tính năng phức tạp hơn và không hỗ trợ sẵn một số khả năng như kế thừa mẫu (template inheritance).
* **Không tự động cập nhật giao diện**: EJS không có cơ chế binding hai chiều như một số framework front-end hiện đại, nghĩa là khi dữ liệu thay đổi, giao diện không tự động cập nhật mà cần tải lại trang.
* **Khả năng mở rộng hạn chế**: Việc tùy chỉnh và mở rộng EJS có thể khó khăn hơn so với một số view engine khác, đặc biệt khi cần các tính năng đặc biệt hoặc tùy chỉnh sâu.
  1. Framework Nodejs
     1. Khái quát về Nodejs và Express
* **Node.js** là một môi trường runtime JavaScript được xây dựng trên Chrome's V8 JavaScript engine, cho phép thực thi mã JavaScript ở phía máy chủ. Không phải là một ngôn ngữ lập trình mà là một platform, Node.js cho phép nhà phát triển xây dựng các ứng dụng mạng và máy chủ với hiệu suất cao bằng JavaScript.
* **Express.js** là một framework web tối giản, linh hoạt cho Node.js, cung cấp một bộ tính năng mạnh mẽ để phát triển các ứng dụng web và API. Express không áp đặt cấu trúc cứng nhắc lên ứng dụng, cho phép nhà phát triển tự do thiết kế kiến trúc phù hợp với yêu cầu của dự án.
* Các framework Node.js và thư viện đi kèm như Express cung cấp các tính năng được cấu hình sẵn, giúp nhà phát triển tích hợp vào ứng dụng của mình, từ đó giảm tải được số lượng code, giảm thời gian phát triển và tăng độ bảo mật. Hơn nữa, các ứng dụng được xây dựng trên các framework này thường yêu cầu ít bảo trì hơn so với các ứng dụng được xây dựng từ đầu.



Hình 2.1 Model-View-Controller

Node.js không áp đặt bất kỳ mô hình kiến trúc nào, nhưng Express và các framework khác cho phép dễ dàng triển khai mô hình MVC (Model-View-Controller). Đây là mô hình phân tách ứng dụng thành ba thành phần chính, giúp quản lý mã nguồn hiệu quả và dễ bảo trì:

* **Controller**: Đảm nhận vai trò nhận và xử lý các request từ người dùng, gọi đúng các phương thức xử lý, và trả về response phù hợp. Trong Node.js, controllers thường là các file JavaScript riêng biệt trong thư mục /controllers.
* **Model**: Chứa tất cả các logic nghiệp vụ, phương thức xử lý, truy xuất cơ sở dữ liệu (như MongoDB, MySQL, hoặc Firebase), và định nghĩa cấu trúc dữ liệu. Trong Node.js, models có thể được định nghĩa bằng các schema nếu sử dụng MongoDB với Mongoose, hoặc các class/object đơn giản.
* **View**: Đảm nhận việc hiển thị thông tin, tương tác với người dùng. Trong Node.js, views thường là các file template như EJS, Pug, Handlebars, hoặc thậm chí là JSON nếu xây dựng API.
* Luồng xử lý trong mô hình MVC của một ứng dụng Node.js:
* Request từ client được gửi đến server
* Router định tuyến request đến controller phù hợp
* Controller xử lý request, tương tác với model nếu cần
* Model thực hiện các thao tác với dữ liệu, giao tiếp với cơ sở dữ liệu
* Controller nhận dữ liệu từ model, gửi đến view
* View render dữ liệu và trả về HTML (hoặc JSON/XML trong trường hợp API)
* Response được gửi về cho client
* Bằng cách này, thông tin nội hàm được xử lý tách biệt với phần thông tin xuất hiện trong giao diện người dùng, đảm bảo các nguyên tắc nghề nghiệp của lập trình viên và tạo điều kiện cho việc phát triển, mở rộng, và bảo trì ứng dụng.
  + 1. Vòng đời
* MVC (Model-View-Controller) là mẫu kiến trúc phần mềm trên máy tính nhằm mục đích tạo lập giao diện cho người dùng. Theo đó, hệ thống MVC được chia thành ba phần có khả năng tương tác với nhau và tách biệt các nguyên tắc nghiệp vụ với giao diện người dùng.
* Ba thành phần ấy bao gồm:

1. Controller: Giữ nhiệm vụ nhận điều hướng các yêu cầu từ người dùng và gọi đúng những phương thức xử lý chúng.
2. Model: Là thành phần chứa tất cả các nghiệp vụ logic, phương thức xử lý, truy xuất database, đối tượng mô tả dữ liệu như các Class, hàm xử lý…
3. View: Đảm nhận việc hiển thị thông tin, tương tác với người dùng, nơi chứa tất cả các đối tượng GUI như textbox, images…

* Bằng cách này, thông tin nội hàm được xử lý tách biệt với phần thông tin xuất hiện trong giao diện người dùng. Bảo đảm các nguyên tắc nghề nghiệp của lập trình viên.

## 2.6 Firebase

### ***2.6.1 Khái quát về Firebase***

**Firebase** là một nền tảng phát triển ứng dụng di động và web do Google phát triển và cung cấp. Ban đầu là một startup độc lập được thành lập vào năm 2011, Firebase được Google mua lại vào năm 2014 và đã phát triển thành một bộ công cụ và dịch vụ toàn diện.

Firebase cung cấp các dịch vụ cloud-based, cho phép nhà phát triển tập trung vào việc xây dựng ứng dụng mà không cần quản lý cơ sở hạ tầng. Nó có thể được xem như một Backend-as-a-Service (BaaS), giảm bớt công việc phát triển backend cho các ứng dụng web và di động.

**Firebase Realtime Database** và **Firestore** là hai dịch vụ cơ sở dữ liệu NoSQL của Firebase, cung cấp khả năng lưu trữ và đồng bộ hóa dữ liệu theo thời gian thực giữa các client, cho phép xây dựng các ứng dụng có tính tương tác cao.

### ***2.6.2 Các dịch vụ chính của Firebase***

1. **Dịch vụ phát triển**:
   * **Firestore & Realtime Database**: Cơ sở dữ liệu NoSQL có khả năng đồng bộ hóa theo thời gian thực
   * **Authentication**: Hệ thống xác thực người dùng đa nền tảng, hỗ trợ nhiều phương pháp đăng nhập
   * **Storage**: Dịch vụ lưu trữ đám mây cho hình ảnh, âm thanh, video và các nội dung khác
   * **Hosting**: Dịch vụ hosting cho ứng dụng web tĩnh và động
   * **Cloud Functions**: Môi trường serverless cho phép thực thi mã phía server
   * **ML Kit**: Bộ công cụ machine learning cho ứng dụng di động
2. **Dịch vụ tăng trưởng**:
   * **Cloud Messaging**: Gửi thông báo đẩy miễn phí và đáng tin cậy
   * **Dynamic Links**: Liên kết thông minh điều hướng người dùng đến vị trí thích hợp
   * **Remote Config**: Thay đổi giao diện và hành vi của ứng dụng từ xa
3. **Dịch vụ phân tích**:
   * **Analytics**: Phân tích hành vi người dùng trên ứng dụng
   * **Crashlytics**: Báo cáo lỗi thời gian thực
   * **Performance Monitoring**: Theo dõi hiệu suất ứng dụng

### ***2.6.3 Tích hợp Firebase với Node.js***

Firebase Admin SDK cho Node.js cung cấp các API để tương tác với Firebase từ môi trường máy chủ. Quy trình tích hợp Firebase vào ứng dụng Node.js thường bao gồm:

1. **Cài đặt SDK**:

bash

npm install firebase-admin

1. **Khởi tạo ứng dụng Firebase**:

javascript

const admin = require('firebase-admin');

const serviceAccount = require('./path/to/serviceAccountKey.json');

admin.initializeApp({

credential: admin.credential.cert(serviceAccount),

databaseURL: "https://your-app.firebaseio.com"

});

1. **Sử dụng các dịch vụ Firebase**:

javascript

*// Firestore*

const db = admin.firestore();

const roomsRef = db.collection('rooms');

*// Authentication*

const auth = admin.auth();

*// Storage*

const storage = admin.storage();

Trong kiến trúc MVC của Node.js, Firebase thường được tích hợp vào phần Model, đóng vai trò là lớp trung gian giữa ứng dụng và cơ sở dữ liệu. Các model sẽ định nghĩa các hàm để tương tác với Firebase, thực hiện các thao tác CRUD (Create, Read, Update, Delete), trong khi controllers sẽ gọi các hàm này để xử lý các request từ client.

Việc sử dụng Firebase với Node.js trong mô hình MVC giúp đơn giản hóa việc phát triển backend, cung cấp khả năng mở rộng tự động, và tích hợp sẵn các tính năng quan trọng như xác thực và lưu trữ, cho phép nhà phát triển tập trung vào logic kinh doanh và trải nghiệm người dùng.

## **2.7 Android Native và Kotlin**

### ***2.7.1. Android Native (Ứng dụng gốc Android)***

* **Khái niệm:** Android Native dùng để chỉ việc phát triển ứng dụng di động dành riêng cho hệ điều hành Android. Các ứng dụng này được xây dựng bằng cách sử dụng các bộ công cụ phát triển phần mềm (SDK) và ngôn ngữ lập trình được Google hỗ trợ chính thức cho nền tảng Android.
* **Ngôn ngữ lập trình:** Các ngôn ngữ chính thức và phổ biến nhất để phát triển Android Native là Java và Kotlin.
* **Công cụ phát triển:** Công cụ chính được sử dụng là Android Studio, một môi trường phát triển tích hợp (IDE) cung cấp đầy đủ các công cụ để thiết kế, viết mã, biên dịch, kiểm thử và gỡ lỗi ứng dụng Android.
* **Ưu điểm:**
  + **Hiệu suất tối ưu:** Ứng dụng Native được biên dịch thành mã máy của thiết bị, tận dụng tối đa phần cứng và các tính năng của hệ điều hành, mang lại hiệu suất cao nhất.
  + **Trải nghiệm người dùng tốt nhất (UX):** Ứng dụng tuân thủ các nguyên tắc thiết kế và luồng hoạt động quen thuộc của Android, tạo cảm giác tự nhiên và mượt mà cho người dùng.
  + **Truy cập đầy đủ API và tính năng hệ thống:** Lập trình viên có thể truy cập trực tiếp vào tất cả các API hệ thống, cảm biến (GPS, camera, microphone,...) và tính năng mới nhất của Android mà không bị giới hạn.
  + **Độ tin cậy và ổn định cao:** Do được tối ưu hóa cho nền tảng, ứng dụng Native thường ít gặp lỗi và hoạt động ổn định hơn.
* **Nhược điểm:**
  + **Chỉ chạy trên Android:** Mã nguồn viết cho Android Native không thể sử dụng lại để chạy trên các nền tảng khác như iOS hay web.
  + **Thời gian và chi phí phát triển:** Nếu muốn có ứng dụng trên cả Android và iOS, bạn cần xây dựng hai ứng dụng riêng biệt, tốn nhiều thời gian và chi phí hơn.

### ***2.7.2. Kotlin***

* **Khái niệm:** Kotlin là một ngôn ngữ lập trình hiện đại, mã nguồn mở, kiểu tĩnh, được phát triển bởi công ty JetBrains (công ty tạo ra các IDE nổi tiếng như IntelliJ IDEA, Android Studio). Nó được thiết kế để chạy trên Máy ảo Java (JVM) và có khả năng tương tác hoàn toàn 100% với Java.
* **Lịch sử với Android:** Kotlin được JetBrains giới thiệu lần đầu vào năm 2011. Vào năm 2017, Google đã công bố Kotlin là ngôn ngữ lập trình chính thức và được ưu tiên hàng đầu cho phát triển ứng dụng Android. Hiện nay, Kotlin được hơn 60% các nhà phát triển Android chuyên nghiệp sử dụng.
* **Đặc điểm nổi bật:**
  + **Ngắn gọn và dễ đọc:** Cú pháp của Kotlin loại bỏ nhiều đoạn mã soạn sẵn (boilerplate code) thường thấy trong Java, giúp mã nguồn ngắn gọn, rõ ràng và dễ hiểu hơn.
  + **An toàn hơn (Null Safety):** Hệ thống kiểu của Kotlin được thiết kế để giảm thiểu lỗi NullPointerException (một lỗi rất phổ biến trong Java) bằng cách phân biệt rõ ràng giữa các kiểu có thể và không thể chứa giá trị null. Google cho biết các ứng dụng Android dùng Kotlin có tỷ lệ sự cố thấp hơn 20%.
  + **Tương thích hoàn toàn với Java:** Mã Kotlin có thể gọi mã Java và ngược lại một cách liền mạch trong cùng một dự án. Điều này cho phép các dự án Java hiện có có thể dần dần chuyển đổi sang Kotlin hoặc sử dụng cả hai ngôn ngữ song song.
  + **Hỗ trợ Coroutines:** Kotlin cung cấp Coroutines để xử lý các tác vụ bất đồng bộ (như gọi mạng, truy vấn cơ sở dữ liệu) một cách đơn giản và hiệu quả hơn so với các cơ chế truyền thống của Java.
  + **Hỗ trợ lập trình hàm:** Kotlin hỗ trợ các khái niệm lập trình hàm, giúp viết mã có cấu trúc tốt hơn và dễ kiểm thử hơn.
  + **Công cụ hỗ trợ tốt:** Kotlin được tích hợp sâu và hỗ trợ mạnh mẽ trong Android Studio và các IDE khác của JetBrains.
* **Ưu điểm trong phát triển Android:**
  + **Tăng năng suất:** Mã ngắn gọn hơn, an toàn hơn và các tính năng hiện đại giúp lập trình viên phát triển ứng dụng nhanh hơn và ít lỗi hơn.
  + **Dễ học (đối với người biết Java):** Cú pháp quen thuộc và khả năng tương tác tốt giúp các lập trình viên Java chuyển sang Kotlin dễ dàng.
  + **Là ngôn ngữ được Google ưu tiên:** Kotlin nhận được sự hỗ trợ hàng đầu từ Google cho các thư viện, công cụ và tài liệu mới nhất của Android, ví dụ như Jetpack Compose (bộ công cụ xây dựng giao diện người dùng hiện đại của Android được xây dựng hoàn toàn bằng Kotlin).
* **Nhược điểm (ít đáng kể):**
  + **Tốc độ biên dịch:** Đôi khi tốc độ biên dịch có thể chậm hơn một chút so với Java trong một số trường hợp, mặc dù đã được cải thiện nhiều.
  + **Số lượng lập trình viên kỳ cựu:** Mặc dù cộng đồng đang phát triển nhanh chóng, số lượng lập trình viên có nhiều năm kinh nghiệm sâu sắc với Kotlin có thể ít hơn so với Java (do Kotlin mới hơn).

1. PHÂN TÍCH THIẾT KẾ HỆ THỐNG
   1. Khảo sát hiện trạng
      1. Mô tả đề tài

Hệ thống quản lý và đặt phòng khách sạn trực tuyến là một giải pháp toàn diện cho ngành dịch vụ lưu trú, nơi thông tin về phòng ốc, dịch vụ và các chương trình ưu đãi được cập nhật liên tục để đáp ứng nhu cầu đa dạng của du khách.

Tính năng đặt phòng và thanh toán trực tuyến đóng vai trò thiết yếu, mang lại sự thuận tiện tối đa cho khách hàng và nâng cao đáng kể trải nghiệm của họ khi lựa chọn nơi ở. Đây là một trong những chức năng cốt lõi, giúp khách sạn tối ưu hóa quy trình và tăng cường khả năng cạnh tranh.

Việc quản lý hệ thống một cách hiệu quả là vô cùng quan trọng đối với hoạt động kinh doanh khách sạn trực tuyến. Người quản trị hệ thống được cung cấp đầy đủ quyền hạn để giám sát và điều chỉnh các khía cạnh then chốt, bao gồm:

* **Quản lý phòng:** Cập nhật thông tin chi tiết về các loại phòng, tình trạng phòng (trống, đã đặt, đang sửa chữa), giá cả, tiện nghi đi kèm.
* **Quản lý dịch vụ khách sạn:** Quản lý các dịch vụ cộng thêm như đặt đồ, v.v
* **Quản lý đặt phòng:** Theo dõi, xác nhận, sửa đổi hoặc hủy bỏ các yêu cầu đặt phòng từ khách hàng.
* **Quản lý hóa đơn và thanh toán:** Xử lý các giao dịch, xuất hóa đơn và theo dõi tình trạng thanh toán.
* **Quản lý khách hàng:** Lưu trữ thông tin khách hàng, lịch sử đặt phòng và các yêu cầu đặc biệt để cá nhân hóa dịch vụ.
* **Quản lý nội dung:** Đăng tải và cập nhật các bài viết giới thiệu về khách sạn, các chương trình khuyến mãi, thông tin du lịch địa phương.
* **Quản lý tài khoản người dùng:** Phân quyền và quản lý tài khoản cho nhân viên các bộ phận (lễ tân, buồng phòng, kinh doanh) và quản lý tài khoản của khách hàng thân thiết.
* **Quản lý hệ thống:** Cấu hình các cài đặt chung, sao lưu dữ liệu và đảm bảo an ninh hệ thống.

Song song với website, một **ứng dụng Android gốc** được phát triển nhằm mang lại trải nghiệm liền mạch và tiện lợi hơn nữa cho người dùng di động. Ứng dụng cho phép khách hàng dễ dàng tìm kiếm phòng, thực hiện đặt phòng nhanh chóng, quản lý các đặt chỗ hiện có, nhận thông báo về ưu đãi đặc biệt và tương tác trực tiếp với khách sạn.

Do hệ thống hướng tới phục vụ đa dạng đối tượng khách hàng, từ khách du lịch cá nhân đến khách công tác, giao diện của cả website và ứng dụng di động cần được thiết kế trực quan, hiện đại, chuyên nghiệp và dễ sử dụng. Các tính năng phải thân thiện, quy trình đặt phòng và thanh toán phải đơn giản và an toàn, đảm bảo mang lại sự hài lòng cao nhất cho khách hàng.

* + 1. Chức năng hệ thống

*3.1.2.1. Backend*

Quản trị hệ thống có quyền quản lý toàn bộ hoạt động của hệ thống quản lý khách sạn  
- **Chức năng bao gồm quản lý phòng**, quản lý loại phòng, quản lý tiện nghi phòng. Quản trị viên có thể thực hiện các chức năng như thêm, sửa, xóa, tìm kiếm, thay đổi trạng thái của phòng (sẵn sàng, đang sử dụng, đang dọn dẹp, bảo trì).  
**- Chức năng quản lý đặt phòng:** Chức năng bao gồm quản lý thông tin đặt phòng, lịch đặt phòng và check-in/check-out. Quản trị viên có thể thực hiện các chức năng như thêm đặt phòng mới, sửa thông tin đặt phòng, hủy đặt phòng, xác nhận đặt phòng, xử lý check-in và check-out.

- **Chức năng quản lý dịch vụ khách sạn** :Chức năng bao gồm quản lý các dịch vụ cung cấp cho khách hàng như dịch vụ phòng, nhà hàng, spa, đưa đón. Quản trị viên có thể thực hiện các chức năng như thêm, sửa, xóa, thay đổi trạng thái của các dịch vụ.  
- **Chức năng quản lý yêu cầu khách hàng**: Chức năng bao gồm quản lý thông tin liên hệ và yêu cầu từ khách hàng. Quản trị viên có thể thực hiện các chức năng xem, phản hồi, xử lý, cập nhật trạng thái và xóa các yêu cầu.

- **Chức năng quản lý khách hàng**: Cho phép tìm kiếm, xem thông tin, quản lý tài khoản khách hàng. Quản trị viên có thể thực hiện các chức năng như xem lịch sử đặt phòng, quản lý điểm thưởng, cập nhật thông tin cá nhân, quản lý cấp độ thành viên.

- **Chức năng quản lý nhân viên:** Chức năng bao gồm quản lý thông tin nhân viên, phân quyền, lịch làm việc. Quản trị viên có thể thực hiện các chức năng như thêm, sửa, xóa nhân viên, phân công công việc, theo dõi hiệu suất, quản lý quyền truy cập hệ thống.

- **Chức năng quản lý thanh toán**: Chức năng bao gồm quản lý các phương thức thanh toán và giao dịch trên hệ thống. Quản trị viên có thể thực hiện các chức năng như thêm, sửa, xóa, kích hoạt hoặc vô hiệu hóa các hình thức thanh toán, xem lịch sử giao dịch, xử lý hoàn tiền.

*3.1.2.2 Frontend*

Frontend của hệ thống quản lý khách sạn Lembo's Luxurious Hotel được xây dựng trên các tiêu chí:

* Giao diện sang trọng, chuyên nghiệp, các thao tác đơn giản và thân thiện với người dùng.
* Thông tin phòng và dịch vụ phong phú, đa dạng, luôn cập nhật những khuyến mãi và ưu đãi mới nhất.
* Có thông tin liên hệ và vị trí của khách sạn rõ ràng, dễ tiếp cận.
* Font chữ thanh lịch, hình ảnh chất lượng cao, màu sắc hài hòa phản ánh đẳng cấp của thương hiệu.
* Tính bảo mật dữ liệu của hệ thống phải cao, đặc biệt với thông tin cá nhân và thanh toán của khách hàng.
* Đảm bảo vận hành tốt khi có nhiều người dùng cùng tương tác tại cùng một thời điểm.
* Website tương thích với các trình duyệt phổ biến và thiết bị di động (responsive design).

Frontend của hệ thống quản lý khách sạn thực hiện các chức năng:

* Hiển thị thông tin chi tiết về các loại phòng, tiện nghi, dịch vụ kèm theo và giá cả.
* Hệ thống đặt phòng trực tuyến với giao diện trực quan, dễ sử dụng và bảo mật.
* Cho phép khách hàng kiểm tra tình trạng phòng trống theo ngày.
* Tìm kiếm phòng theo nhiều tiêu chí (loại phòng, giá, tiện nghi, ngày).
* Cho phép khách hàng đăng ký, đăng nhập và quản lý tài khoản cá nhân.
* Xem lịch sử đặt phòng và quản lý các đặt phòng hiện tại.
* Đặt dịch vụ phòng và các dịch vụ bổ sung trực tuyến.
* Hệ thống đánh giá và nhận xét sau khi sử dụng dịch vụ.
* Hiển thị thông tin về các điểm tham quan, ẩm thực và hoạt động giải trí gần khách sạn.
* Cho phép gửi yêu cầu đặc biệt và thông tin liên hệ trực tiếp đến bộ phận dịch vụ khách hàng.
* Hỗ trợ trực tuyến thông qua chat hoặc gọi điện thoại.
  1. Phân tích và thiết kế hệ thống
     1. Mô tả nghiệp vụ

*3.2.1.1. Đối với khách hàng*

Khách hàng có thể tìm kiếm hoặc click vào đường link của website. Website sẽ hiển thị trang chủ và các phân trang liên quan. Tương tự với app android cũng thế

Tại website và app khách hàng có thể:

* **Xem chi tiết phòng** bao gồm tên phòng, hình ảnh, đơn giá, tiện ích, mô tả chi tiết và các thông tin khác
* **Tìm kiếm phòng** theo nhiều tiêu chí khác nhau như tên phòng, khoảng giá, loại phòng, tiện nghi
* **Đặt phòng:** Khách hàng bấm vào phần đặt phòng và chọn loại phòng mong muốn
* **Quản lý đặt phòng:** Xem danh sách các loại phòng đã chọn, xem chi tiết giá tiền từng phòng và tổng giá tiền
* **Điều chỉnh đặt phòng:** Thực hiện thêm/bớt số lượng phòng, xóa phòng đã chọn

Khi khách hàng thực hiện đặt phòng, họ cần:

* **Đăng ký tài khoản** (nếu là khách hàng mới) với các thông tin như họ tên, tên đăng nhập, mật khẩu, số điện thoại, email
* Sau khi click vào nút đặt phòng, **hóa đơn** sẽ hiển thị với nội dung như: tên các phòng, đơn giá, số lượng, thành tiền và thuế
* Khách hàng có thể chọn **hình thức thanh toán** là thanh toán trực tiếp hoặc thanh toán online
* Khi chọn thanh toán online, khách hàng cần liên kết thẻ ngân hàng hoặc ví điện tử để thực hiện thanh toán (tính năng đang phát triển)

Đến ngày nhận phòng:

* Khách đến khách sạn và thực hiện **check-in** với lễ tân
* Quá trình có thể thực hiện thủ công hoặc qua quét mã QR (tính năng đang phát triển)

Các chức năng khác của tài khoản:

* **Quản lý tài khoản cá nhân:** Xem lịch sử đặt phòng, chi tiết hóa đơn (mã hóa đơn, tên phòng, thành tiền, thời gian check-in/check-out)
* **Đặt dịch vụ phòng:** Khi booking đang ở trạng thái "đã check-in", khách hàng có thể gọi các món ăn để mang trực tiếp lên phòng
* **Để lại đánh giá:** Khách hàng có thể để lại bình luận cho mỗi phòng đã sử dụng

**Trường hợp đặc biệt:**

* Khách hàng ở trạng thái "suspended" hoặc "banned" sẽ không thể truy cập các chức năng thông thường
* Những khách hàng này có lựa chọn liên hệ với admin qua số điện thoại hoặc hệ thống khiếu nại của khách sạn để được giải quyết
  + 1. Đối với quản trị viên (QTV/Admin)

Khi khởi tạo hệ thống:

* Khi bắt đầu khởi tạo phần mềm, thực hiện đăng ký 1 tài khoản và thủ công chỉnh tài khoản đó thành admin

Sau khi đăng nhập thành công, màn hình chính của trang quản trị gồm các chức năng:

* **Thống kê:** Cho phép theo dõi tổng các phòng đã đặt, tổng doanh thu theo tháng, tổng số khách hàng đã giao dịch
* **Quản lý phòng:** Xem và thao tác với các phòng trong khách sạn (thêm, sửa, xóa, cập nhật trạng thái)
* **Quản lý loại phòng:** Xem và thao tác các loại phòng. Đây là thành phần quan trọng nhất của hệ thống, xác định giá tiền của các phòng (tất cả các phòng thuộc cùng một loại sẽ có giá theo loại phòng)
* **Quản lý tiện nghi phòng:** Xem và cập nhật tiện nghi (thêm và xóa tiện nghi thực hiện tại quản lý phòng)
* **Quản lý khách hàng:** Xem thông tin khách hàng như liên hệ, số phòng đã đặt, ngày đăng ký và thông tin cá nhân. Có thể thay đổi mật khẩu cho khách hàng đã quên, giới hạn tài khoản khách hàng có hành vi không phù hợp
* **Quản lý bookings:** Xem và cập nhật các booking của khách hàng. Có thể thực hiện scan QR để check-in (đang phát triển) hoặc thực hiện thủ công
* **Quản lý menu đồ ăn:** Thêm, sửa, xóa đồ ăn của khách sạn, quản lý các đơn đặt hàng, phân công nhân viên giao đồ ăn
* **Quản lý nhân viên:** Thêm, sửa, xóa nhân viên và phân công vị trí làm việc tương ứng
  1. Xây dựng biểu đồ usecase

|  |  |  |
| --- | --- | --- |
| **STT** | **Tác nhân** | **Mô tả** |
| 1 | Admin | Admin là đối tượng chịu trách nhiệm quản trị hệ thống, giám sát việc sử dụng hế thống của các đối tượng khác, có quyền truy cập và sử dụng tất cả các chức năng trong hệ thống. |
| 2 | Người dùng/Khách hàng | Khách hàng đăng nhập vào hệ thống để xem thông tin phòng, đặt phòng, đồ ăn và đánh giá phòng. |

* + 1. Biểu đồ Usecase tổng quát

A diagram of a network

AI-generated content may be incorrect.

Hình 3.1 Biểu đồ Use case tổng quát

* + 1. Use case đăng nhập Admin

Ảnh có chứa văn bản, biểu đồ, vòng tròn, Phông chữ

Mô tả được tạo tự động

Hình 3.2 Biểu đồ Use case đăng nhâp

Tác nhân: Admin.

Mô tả: Use case cho admin đăng nhập vào hệ thống.

Điều kiện trước: admin chưa đăng nhập vào hệ thống.

Luồng sự kiện chính:

* Tìm kiếm website admin quản lý website.
* Giao diện đăng nhập hiển thị.
* Nhập tên đăng nhập (email), mật khẩu vào form đăng nhập.
* Hệ thống kiểm tra tài khoản admin và mật khẩu nhập của admin. Nếu nhập sai tài khoản admin hoặc mật khẩu thì sẽ yêu cầu nhập lại tài khoản mật khẩu. Nếu nhập đúng thì hệ thống sẽ chuyển tới trang quản trị.

Use case kết thúc.

* + 1. Use case quản lý phòng

A diagram of a diagram

AI-generated content may be incorrect.

Hình 3.3 Use case quản lý phòng

Tác nhân

Admin

Mô tả

Use case cho admin quản lý thông tin và trạng thái các phòng trong hệ thống khách sạn.

Điều kiện trước

* Admin đã đăng nhập vào hệ thống.
* Hệ thống đã được cấu hình với các loại phòng và tiện nghi cơ bản.

Luồng sự kiện chính

1. Admin truy cập vào mục "Quản lý phòng" từ menu chính của hệ thống.
2. Hệ thống hiển thị danh sách các phòng hiện có với thông tin cơ bản như mã phòng, tên phòng, loại phòng, giá, trạng thái hiện tại.
3. Admin có thể thực hiện các hành động sau:
   * Tìm kiếm phòng: Nhập từ khóa tìm kiếm (mã phòng, tên phòng) hoặc lọc theo loại phòng, trạng thái.
   * Xem chi tiết phòng: Chọn một phòng để xem thông tin chi tiết.
   * Thêm phòng mới: Chọn chức năng thêm phòng mới để nhập thông tin phòng.
   * Sửa thông tin phòng: Chọn phòng cần sửa và cập nhật thông tin.
   * Cập nhật trạng thái phòng: Thay đổi trạng thái phòng (sẵn sàng, đang sử dụng, đang dọn dẹp, bảo trì).
   * Xóa phòng: Chọn phòng cần xóa và xác nhận hành động.
4. Hệ thống cập nhật dữ liệu theo hành động của admin và hiển thị thông báo phù hợp (thành công hoặc lỗi).
5. Hệ thống cập nhật danh sách phòng sau mỗi thao tác thành công.

Luồng phụ

Luồng phụ: Thêm phòng mới

1. Admin chọn chức năng "Thêm phòng".
2. Hệ thống hiển thị form nhập thông tin phòng mới.
3. Admin nhập thông tin phòng bao gồm:
   * Mã phòng (nếu không tự động sinh)
   * Tên phòng
   * Chọn loại phòng từ danh sách có sẵn
   * Giá phòng (mặc định theo loại phòng)
   * Trạng thái phòng
   * Mô tả phòng
   * Chọn các tiện nghi phòng
   * Tải lên hình ảnh phòng
4. Admin nhấn nút "Lưu" để hoàn tất.
5. Hệ thống kiểm tra tính hợp lệ của thông tin:
   * Nếu thông tin hợp lệ: Lưu phòng mới vào cơ sở dữ liệu và hiển thị thông báo thành công.
   * Nếu thông tin không hợp lệ: Hiển thị thông báo lỗi và yêu cầu nhập lại.
6. Quay lại màn hình danh sách phòng với dữ liệu đã cập nhật.

Luồng phụ: Sửa phòng

1. Admin chọn phòng cần sửa từ danh sách và nhấn nút "Sửa".
2. Hệ thống hiển thị form với thông tin hiện tại của phòng.
3. Admin cập nhật thông tin phòng theo yêu cầu.
4. Admin nhấn nút "Cập nhật" để lưu thay đổi.
5. Hệ thống kiểm tra tính hợp lệ của thông tin:
   * Nếu thông tin hợp lệ: Lưu thay đổi vào cơ sở dữ liệu và hiển thị thông báo thành công.
   * Nếu thông tin không hợp lệ: Hiển thị thông báo lỗi và yêu cầu nhập lại.
6. Quay lại màn hình danh sách phòng với dữ liệu đã cập nhật.

Luồng phụ: Cập nhật trạng thái phòng

1. Admin chọn phòng cần cập nhật trạng thái.
2. Admin chọn trạng thái mới từ danh sách (sẵn sàng, đang sử dụng, đang dọn dẹp, bảo trì).
3. Hệ thống cập nhật trạng thái phòng và hiển thị thông báo thành công.
4. Danh sách phòng được cập nhật với trạng thái mới.

Luồng phụ: Xóa phòng

1. Admin chọn phòng cần xóa từ danh sách và nhấn nút "Xóa".
2. Hệ thống hiển thị hộp thoại xác nhận với thông tin cảnh báo.
3. Admin xác nhận hành động xóa:
   * Nếu xác nhận: Hệ thống kiểm tra xem phòng có đang được đặt hoặc có lịch sử đặt phòng không.
     + Nếu phòng không có ràng buộc: Xóa phòng khỏi hệ thống và hiển thị thông báo thành công.
     + Nếu phòng có ràng buộc: Hiển thị thông báo cảnh báo và cho phép admin chọn tiếp tục hoặc hủy.
   * Nếu hủy: Đóng hộp thoại và không thực hiện hành động xóa.
4. Danh sách phòng được cập nhật sau khi xóa thành công.

Luồng phụ: Tìm kiếm phòng

1. Admin nhập từ khóa tìm kiếm vào ô tìm kiếm và/hoặc chọn các bộ lọc (loại phòng, trạng thái).
2. Admin nhấn nút "Tìm kiếm" hoặc nhấn Enter.
3. Hệ thống tìm kiếm và hiển thị danh sách phòng phù hợp với điều kiện tìm kiếm.
4. Nếu không có kết quả, hiển thị thông báo "Không tìm thấy phòng phù hợp".

Điều kiện sau

* Thông tin phòng được cập nhật trong hệ thống.
* Admin có thể tiếp tục quản lý phòng hoặc chuyển sang chức năng khác của hệ thống.

Yêu cầu đặc biệt

* Quyền truy cập: Chỉ admin và nhân viên được cấp quyền mới có thể quản lý phòng.
* Giao diện: Hiển thị trực quan trạng thái phòng bằng màu sắc khác nhau.
* Hiệu suất: Tải danh sách phòng nhanh chóng, kể cả khi có nhiều phòng trong hệ thống.
* Bảo mật: Ghi lại nhật ký (log) cho tất cả các thao tác thêm/sửa/xóa phòng.

Use case liên quan

* Đăng nhập (include)
* Quản lý loại phòng
* Quản lý tiện nghi
* Quản lý đặt phòng
* Quản lý giá phòng theo mùa

Use case kết thúc

* + 1. Use case quản lý loại phòng

A diagram of a diagram

AI-generated content may be incorrect.

Hình 3.4 Use case quản lý loại phòng

Tác nhân

Admin

Mô tả

Use case cho admin quản lý các loại phòng trong hệ thống khách sạn.

Điều kiện trước

* Admin đã đăng nhập vào hệ thống.
* Hệ thống đã được thiết lập.

Luồng sự kiện chính

1. Admin truy cập vào mục "Quản lý loại phòng" từ menu chính.
2. Hệ thống hiển thị danh sách các loại phòng hiện có với thông tin cơ bản như mã loại phòng, tên loại phòng, mô tả, giá và các thông tin liên quan.
3. Admin có thể thực hiện các hành động sau:
   * Tìm kiếm loại phòng: Nhập từ khóa tìm kiếm để lọc danh sách loại phòng.
   * Thêm loại phòng mới: Chọn chức năng thêm loại phòng mới để nhập thông tin.
   * Sửa thông tin loại phòng: Chọn loại phòng cần sửa và cập nhật thông tin.
   * Xóa loại phòng: Chọn loại phòng cần xóa và xác nhận hành động.
4. Hệ thống cập nhật dữ liệu và hiển thị thông báo phù hợp.

Luồng phụ

Luồng phụ: Thêm loại phòng mới

1. Admin chọn chức năng "Thêm loại phòng".
2. Hệ thống hiển thị form nhập thông tin loại phòng mới.
3. Admin nhập thông tin bao gồm:
   * Tên loại phòng
   * Mô tả chi tiết
   * Giá cơ bản
   * Các thông tin bổ sung
4. Admin nhấn nút "Lưu" để hoàn tất.
5. Hệ thống kiểm tra tính hợp lệ của thông tin:
   * Nếu thông tin hợp lệ: Lưu loại phòng mới vào cơ sở dữ liệu và hiển thị thông báo thành công.
   * Nếu thông tin không hợp lệ: Hiển thị thông báo lỗi và yêu cầu nhập lại.

Luồng phụ: Sửa loại phòng

1. Admin chọn loại phòng cần sửa và nhấn nút "Sửa".
2. Hệ thống hiển thị form với thông tin hiện tại của loại phòng.
3. Admin cập nhật thông tin cần thiết.
4. Admin nhấn nút "Cập nhật" để lưu thay đổi.
5. Hệ thống kiểm tra tính hợp lệ của thông tin và cập nhật dữ liệu nếu hợp lệ.

Luồng phụ: Xóa loại phòng

1. Admin chọn loại phòng cần xóa và nhấn nút "Xóa".
2. Hệ thống hiển thị hộp thoại xác nhận.
3. Admin xác nhận hành động xóa:
   * Nếu xác nhận: Hệ thống kiểm tra xem loại phòng có đang được sử dụng không.
     + Nếu không có phòng nào thuộc loại này: Xóa loại phòng và hiển thị thông báo thành công.
     + Nếu có phòng đang sử dụng loại này: Hiển thị thông báo cảnh báo và không cho phép xóa.
   * Nếu hủy: Đóng hộp thoại và không thực hiện hành động xóa.

Điều kiện sau

* Thông tin loại phòng được cập nhật trong hệ thống.
* Danh sách loại phòng được hiển thị với thông tin mới nhất.

Yêu cầu đặc biệt

* Quyền truy cập: Chỉ admin mới có thể quản lý loại phòng.
* Tính toàn vẹn: Đảm bảo không xóa loại phòng đang được sử dụng.
* Hiệu suất: Cập nhật thông tin loại phòng phải được áp dụng cho tất cả các phòng thuộc loại đó.

Use case liên quan

* Đăng nhập (include)
* Quản lý phòng
* Quản lý giá phòng theo mùa
  + 1. Use case quản lý tiện nghi

A diagram of a diagram

AI-generated content may be incorrect.

Hình 3.5 Use case quản lý tiện nghi

Tác nhân: Admin

Mô tả: Use case cho admin quản lý các tiện nghi trong khách sạn.

Điều kiện trước

* Admin đã đăng nhập vào hệ thống.

Luồng sự kiện chính

1. Admin truy cập vào mục "Quản lý tiện nghi" từ menu chính.
2. Hệ thống hiển thị danh sách các tiện nghi hiện có với thông tin như tên tiện nghi, mô tả, biểu tượng.
3. Admin có thể thực hiện các hành động sau:
   * Tìm kiếm tiện nghi: Nhập từ khóa để lọc danh sách tiện nghi.
   * Thêm tiện nghi mới: Chọn chức năng thêm tiện nghi mới.
   * Sửa thông tin tiện nghi: Chọn tiện nghi cần sửa và cập nhật thông tin.
   * Xóa tiện nghi: Chọn tiện nghi cần xóa và xác nhận hành động.
4. Hệ thống cập nhật dữ liệu và hiển thị thông báo phù hợp.

Luồng phụ

Luồng phụ: Thêm tiện nghi mới

1. Admin chọn chức năng "Thêm tiện nghi".
2. Hệ thống hiển thị form nhập thông tin tiện nghi mới.
3. Admin nhập thông tin bao gồm:
   * Tên tiện nghi
   * Mô tả
   * Chọn biểu tượng (icon) hoặc tải lên hình ảnh minh họa
4. Admin nhấn nút "Lưu" để hoàn tất.
5. Hệ thống kiểm tra tính hợp lệ của thông tin và lưu tiện nghi mới vào cơ sở dữ liệu.

Luồng phụ: Sửa tiện nghi

1. Admin chọn tiện nghi cần sửa và nhấn nút "Sửa".
2. Hệ thống hiển thị form với thông tin hiện tại của tiện nghi.
3. Admin cập nhật thông tin cần thiết.
4. Admin nhấn nút "Cập nhật" để lưu thay đổi.
5. Hệ thống cập nhật dữ liệu và hiển thị thông báo thành công.

Luồng phụ: Xóa tiện nghi

1. Admin chọn tiện nghi cần xóa và nhấn nút "Xóa".
2. Hệ thống hiển thị hộp thoại xác nhận.
3. Admin xác nhận hành động xóa:
   * Nếu xác nhận: Hệ thống kiểm tra xem tiện nghi có đang được sử dụng bởi phòng nào không.
     + Nếu không có phòng nào sử dụng: Xóa tiện nghi và hiển thị thông báo thành công.
     + Nếu có phòng đang sử dụng: Hiển thị thông báo cảnh báo và không cho phép xóa.
   * Nếu hủy: Đóng hộp thoại và không thực hiện hành động xóa.

Điều kiện sau

* Thông tin tiện nghi được cập nhật trong hệ thống.
* Các phòng liên quan sẽ hiển thị tiện nghi mới hoặc đã cập nhật.

Yêu cầu đặc biệt

* Hệ thống phải hỗ trợ nhiều loại biểu tượng cho tiện nghi.
* Tiện nghi phải được hiển thị trực quan trên trang thông tin phòng.

Use case liên quan

* Đăng nhập (include)
* Quản lý phòng
* Xem chi tiết phòng (khách hàng)
  + 1. Use case quản lý order

A diagram of a diagram

AI-generated content may be incorrect.

Hình 3.6 Biểu đồ Use case quản lý order

Tác nhân

Admin

Mô tả

Use case cho admin quản lý các order dịch vụ phòng và thực phẩm.

Điều kiện trước

* Admin đã đăng nhập vào hệ thống.
* Có các order đã được tạo bởi khách hàng.

Luồng sự kiện chính

1. Admin truy cập vào mục "Quản lý order" từ menu chính.
2. Hệ thống hiển thị danh sách các order với thông tin như mã order, tên khách hàng, phòng, thời gian đặt, tổng tiền, trạng thái.
3. Admin có thể thực hiện các hành động sau:
   * Tìm kiếm order: Nhập từ khóa hoặc lọc theo trạng thái, ngày đặt.
   * Xem chi tiết order: Chọn order để xem thông tin chi tiết.
   * Sửa trạng thái order: Cập nhật trạng thái của order (đang xử lý, đang giao, đã hoàn thành).
   * Cài đặt người giao hàng: Phân công nhân viên phụ trách giao đơn hàng.
   * Xóa order: Hủy order và xác nhận hành động.
4. Hệ thống cập nhật dữ liệu và hiển thị thông báo phù hợp.

Luồng phụ

Luồng phụ: Xem chi tiết order

1. Admin chọn order cần xem chi tiết.
2. Hệ thống hiển thị thông tin chi tiết của order, bao gồm:
   * Thông tin khách hàng
   * Thông tin phòng
   * Danh sách các món ăn/dịch vụ đã đặt với số lượng và giá
   * Tổng tiền
   * Lịch sử cập nhật trạng thái
   * Ghi chú đặc biệt của khách hàng

Luồng phụ: Sửa trạng thái order

1. Admin chọn order cần cập nhật và nhấn nút "Cập nhật trạng thái".
2. Hệ thống hiển thị các lựa chọn trạng thái (đã nhận đơn, đang chuẩn bị, đang giao, đã hoàn thành, đã hủy).
3. Admin chọn trạng thái mới và xác nhận.
4. Hệ thống cập nhật trạng thái order và thông báo cho khách hàng (nếu cần).

Luồng phụ: Cài đặt người giao hàng

1. Admin chọn order và nhấn nút "Phân công giao hàng".
2. Hệ thống hiển thị danh sách nhân viên đang trực và có thể giao hàng.
3. Admin chọn nhân viên phù hợp.
4. Hệ thống cập nhật thông tin và gửi thông báo cho nhân viên được chọn.

Luồng phụ: Xóa order

1. Admin chọn order cần xóa và nhấn nút "Xóa".
2. Hệ thống hiển thị hộp thoại xác nhận.
3. Admin xác nhận hành động xóa:
   * Nếu xác nhận: Hệ thống xóa order và thông báo cho khách hàng.
   * Nếu hủy: Đóng hộp thoại và không thực hiện hành động xóa.

Điều kiện sau

* Thông tin order được cập nhật trong hệ thống.
* Khách hàng nhận được thông báo về các thay đổi trạng thái (nếu có).

Yêu cầu đặc biệt

* Thông báo theo thời gian thực khi có order mới hoặc cập nhật trạng thái.
* Khả năng in hóa đơn và phiếu giao hàng.

Use case liên quan

* Đăng nhập (include)
* Quản lý nhân viên
* Quản lý order items
* Đặt đồ ăn/dịch vụ (khách hàng)
  + 1. Use case quản lý order items

A diagram of a product

AI-generated content may be incorrect.

Hình 3.7 Biểu đồ Use case quản lý order items

Tác nhân

Admin

Mô tả

Use case cho admin quản lý các món ăn/đồ uống và dịch vụ trong menu của khách sạn.

Điều kiện trước

* Admin đã đăng nhập vào hệ thống.

Luồng sự kiện chính

1. Admin truy cập vào mục "Quản lý order items" từ menu chính.
2. Hệ thống hiển thị danh sách các món ăn/đồ uống và dịch vụ hiện có với thông tin như tên, mô tả, giá, danh mục, trạng thái.
3. Admin có thể thực hiện các hành động sau:
   * Tìm kiếm item: Nhập từ khóa tìm kiếm hoặc lọc theo danh mục, trạng thái.
   * Thêm item mới: Chọn chức năng thêm item mới.
   * Sửa thông tin item: Chọn item cần sửa và cập nhật thông tin.
   * Xóa item: Chọn item cần xóa và xác nhận hành động.
4. Hệ thống cập nhật dữ liệu và hiển thị thông báo phù hợp.

Luồng phụ

Luồng phụ: Thêm item mới

1. Admin chọn chức năng "Thêm item".
2. Hệ thống hiển thị form nhập thông tin item mới.
3. Admin nhập thông tin bao gồm:
   * Tên món ăn/dịch vụ
   * Mô tả chi tiết
   * Giá
   * Danh mục (món chính, món phụ, đồ uống, dịch vụ...)
   * Thông tin dinh dưỡng (calo, chất béo, protein...)
   * Thông tin dị ứng
   * Trạng thái sẵn có
   * Tải lên hình ảnh minh họa
4. Admin nhấn nút "Lưu" để hoàn tất.
5. Hệ thống kiểm tra tính hợp lệ của thông tin và lưu item mới vào cơ sở dữ liệu.

Luồng phụ: Sửa item

1. Admin chọn item cần sửa và nhấn nút "Sửa".
2. Hệ thống hiển thị form với thông tin hiện tại của item.
3. Admin cập nhật thông tin cần thiết.
4. Admin nhấn nút "Cập nhật" để lưu thay đổi.
5. Hệ thống cập nhật dữ liệu và hiển thị thông báo thành công.

Luồng phụ: Xóa item

1. Admin chọn item cần xóa và nhấn nút "Xóa".
2. Hệ thống hiển thị hộp thoại xác nhận.
3. Admin xác nhận hành động xóa:
   * Nếu xác nhận: Hệ thống kiểm tra xem item có nằm trong bất kỳ order hiện tại nào không.
     + Nếu không có order nào sử dụng: Xóa item và hiển thị thông báo thành công.
     + Nếu có order đang sử dụng: Hiển thị thông báo cảnh báo và đề xuất vô hiệu hóa item thay vì xóa.
   * Nếu hủy: Đóng hộp thoại và không thực hiện hành động xóa.

Điều kiện sau

* Thông tin order items được cập nhật trong hệ thống.
* Menu dịch vụ phòng được cập nhật với các thay đổi mới nhất.

Yêu cầu đặc biệt

* Hỗ trợ đánh dấu các món ăn đặc biệt (chay, thuần chay, không gluten, cay...).
* Khả năng quản lý các khuyến mãi và giá đặc biệt.
* Khả năng phân loại món ăn theo nhiều danh mục (ví dụ: bữa sáng, món chính, tráng miệng).

Use case liên quan

* Đăng nhập (include)
* Quản lý order
* Đặt đồ ăn/dịch vụ (khách hàng)
  + 1. Usecase quản lý bookings

A diagram of a company

AI-generated content may be incorrect.

Hình 3.8 Biểu đồ Use case quản lý booking

Tác nhân

Admin

Mô tả

Use case cho admin quản lý các đặt phòng trong hệ thống khách sạn.

Điều kiện trước

* Admin đã đăng nhập vào hệ thống.

Luồng sự kiện chính

1. Admin truy cập vào mục "Quản lý booking" từ menu chính.
2. Hệ thống hiển thị danh sách các booking với thông tin như mã booking, tên khách hàng, phòng, ngày check-in, ngày check-out, trạng thái booking, trạng thái thanh toán.
3. Admin có thể thực hiện các hành động sau:
   * Tìm kiếm bookings: Nhập từ khóa tìm kiếm hoặc lọc theo trạng thái, ngày.
   * Sửa trạng thái bookings: Cập nhật trạng thái của booking (đã đặt, đã check-in, đã check-out, đã hủy).
   * Sửa trạng thái giao dịch: Cập nhật trạng thái thanh toán (chưa thanh toán, đã thanh toán một phần, đã thanh toán đầy đủ).
   * Xóa bookings: Hủy booking và xác nhận hành động.
4. Hệ thống cập nhật dữ liệu và hiển thị thông báo phù hợp.

Luồng phụ

Luồng phụ: Tìm kiếm bookings

1. Admin nhập từ khóa tìm kiếm (mã booking, tên khách hàng, số phòng) hoặc chọn các bộ lọc (trạng thái booking, khoảng thời gian).
2. Admin nhấn nút "Tìm kiếm".
3. Hệ thống hiển thị danh sách các booking phù hợp với tiêu chí tìm kiếm.
4. Nếu không có kết quả, hiển thị thông báo "Không tìm thấy booking nào phù hợp".

Luồng phụ: Sửa trạng thái bookings

1. Admin chọn booking cần cập nhật và nhấn nút "Cập nhật trạng thái".
2. Hệ thống hiển thị các lựa chọn trạng thái (đã đặt, đã check-in, đã check-out, đã hủy).
3. Admin chọn trạng thái mới và xác nhận.
4. Hệ thống cập nhật trạng thái booking và thực hiện các hành động liên quan:
   * Nếu chuyển sang "đã check-in": Cập nhật trạng thái phòng thành "đang sử dụng".
   * Nếu chuyển sang "đã check-out": Cập nhật trạng thái phòng thành "cần dọn dẹp".
   * Nếu chuyển sang "đã hủy": Cập nhật trạng thái phòng thành "trống" nếu phòng đã được đặt trước.

Luồng phụ: Sửa trạng thái giao dịch

1. Admin chọn booking cần cập nhật và nhấn nút "Cập nhật thanh toán".
2. Hệ thống hiển thị thông tin thanh toán hiện tại và các lựa chọn cập nhật.
3. Admin cập nhật trạng thái thanh toán và nhập thông tin bổ sung nếu cần.
4. Hệ thống lưu thông tin cập nhật và gửi thông báo cho khách hàng nếu cần.

Luồng phụ: Xóa bookings

1. Admin chọn booking cần xóa và nhấn nút "Xóa".
2. Hệ thống hiển thị hộp thoại xác nhận với cảnh báo về các tác động có thể xảy ra.
3. Admin xác nhận hành động xóa:
   * Nếu xác nhận: Hệ thống xóa booking, cập nhật trạng thái phòng và gửi thông báo cho khách hàng.
   * Nếu hủy: Đóng hộp thoại và không thực hiện hành động xóa.

Điều kiện sau

* Thông tin booking được cập nhật trong hệ thống.
* Trạng thái phòng được cập nhật tương ứng.
* Khách hàng nhận được thông báo về các thay đổi nếu cần.

Yêu cầu đặc biệt

* Khả năng quản lý check-in/check-out hàng loạt cho các đoàn khách.
* Tích hợp với hệ thống quản lý phòng để cập nhật trạng thái phòng tự động.
* Hỗ trợ quét mã QR để xác nhận check-in/check-out nhanh chóng.

Use case liên quan

* Đăng nhập (include)
* Quản lý phòng
* Đặt phòng (khách hàng)
* Quản lý thống kê
  + 1. Use case quản lý khách hàng

A diagram of a user

AI-generated content may be incorrect.

Hình 3.9 Biểu đồ Use case quản lý khách hàng

Tác nhân

Admin

Mô tả

Use case cho admin quản lý tài khoản người dùng trong hệ thống khách sạn.

Điều kiện trước

* Admin đã đăng nhập vào hệ thống.

Luồng sự kiện chính

1. Admin truy cập vào mục "Quản lý user" từ menu chính.
2. Hệ thống hiển thị danh sách các tài khoản người dùng với thông tin như tên, email, số điện thoại, ngày đăng ký, vai trò, trạng thái.
3. Admin có thể thực hiện các hành động sau:
   * Tìm kiếm user: Nhập từ khóa tìm kiếm hoặc lọc theo trạng thái, vai trò.
   * Thay đổi mật khẩu user: Đặt lại mật khẩu cho người dùng khi cần thiết.
   * Xem thông tin user: Xem chi tiết thông tin và lịch sử hoạt động của người dùng.
   * Thay đổi quyền user: Nâng cấp hoặc hạ cấp quyền hạn của người dùng.
   * Quản lý giới hạn tài khoản user: Khóa, mở khóa hoặc giới hạn các tài khoản có hành vi vi phạm.
4. Hệ thống cập nhật dữ liệu và hiển thị thông báo phù hợp.

Luồng phụ

Luồng phụ: Tìm kiếm user

1. Admin nhập từ khóa tìm kiếm (tên, email, số điện thoại) hoặc chọn các bộ lọc (vai trò, trạng thái).
2. Admin nhấn nút "Tìm kiếm".
3. Hệ thống hiển thị danh sách các người dùng phù hợp với tiêu chí tìm kiếm.
4. Nếu không có kết quả, hiển thị thông báo "Không tìm thấy người dùng nào phù hợp".

Luồng phụ: Thay đổi mật khẩu user

1. Admin chọn tài khoản cần đặt lại mật khẩu và nhấn nút "Đặt lại mật khẩu".
2. Hệ thống hiển thị form với các lựa chọn:
   * Tạo mật khẩu ngẫu nhiên và gửi qua email.
   * Nhập mật khẩu mới trực tiếp.
3. Admin chọn phương thức và xác nhận.
4. Hệ thống cập nhật mật khẩu mới và thông báo cho người dùng qua email.

Luồng phụ: Xem thông tin user

1. Admin chọn tài khoản cần xem và nhấn nút "Xem chi tiết".
2. Hệ thống hiển thị thông tin chi tiết bao gồm:
   * Thông tin cá nhân
   * Lịch sử đặt phòng
   * Lịch sử đặt dịch vụ/đồ ăn
   * Lịch sử thanh toán
   * Các phản hồi và đánh giá
   * Hoạt động gần đây trên hệ thống

Luồng phụ: Thay đổi quyền user

1. Admin chọn tài khoản cần thay đổi quyền và nhấn nút "Thay đổi quyền".
2. Hệ thống hiển thị form với các lựa chọn quyền hạn (khách hàng, nhân viên, quản lý, admin).
3. Admin chọn quyền mới và cung cấp lý do thay đổi.
4. Hệ thống yêu cầu xác nhận và cập nhật quyền cho tài khoản.

Luồng phụ: Quản lý giới hạn tài khoản user

1. Admin chọn tài khoản cần giới hạn và nhấn nút "Quản lý giới hạn".
2. Hệ thống hiển thị form với các lựa chọn:
   * Tạm khóa tài khoản (với thời gian xác định)
   * Khóa vĩnh viễn
   * Giới hạn một số chức năng
   * Mở khóa tài khoản
3. Admin chọn hành động, nhập thời gian áp dụng (nếu cần) và lý do.
4. Hệ thống cập nhật trạng thái tài khoản và thông báo cho người dùng.

Điều kiện sau

* Thông tin và trạng thái người dùng được cập nhật trong hệ thống.
* Người dùng nhận được thông báo về các thay đổi đối với tài khoản của họ.

Yêu cầu đặc biệt

* Bảo mật cao cho thao tác thay đổi quyền và mật khẩu.
* Khả năng xuất báo cáo về hoạt động của người dùng.
* Lưu trữ lịch sử thay đổi trạng thái của tài khoản để kiểm tra sau này.

Use case liên quan

* Đăng nhập (include)
* Quản lý khướu nại
* Quản lý booking
  + 1. Use case quản lý khướu nại

A diagram of a diagram

AI-generated content may be incorrect.

Hình 3.10 Use case quản lý khướu nại

Tác nhân

Admin

Mô tả

Use case cho admin quản lý các khiếu nại và phản hồi từ khách hàng.

Điều kiện trước

* Admin đã đăng nhập vào hệ thống.

Luồng sự kiện chính

1. Admin truy cập vào mục "Quản lý khướu nại" từ menu chính.
2. Hệ thống hiển thị danh sách các khiếu nại với thông tin như mã khiếu nại, người gửi, ngày gửi, chủ đề, trạng thái xử lý.
3. Admin có thể thực hiện các hành động sau:
   * Tìm kiếm khướu nại: Nhập từ khóa tìm kiếm hoặc lọc theo trạng thái, khoảng thời gian.
   * Xem chi tiết khướu nại: Xem nội dung đầy đủ của khiếu nại và lịch sử trao đổi.
   * Thực hiện đồng ý/hủy bỏ với khướu nại: Xử lý khiếu nại của khách hàng và cập nhật trạng thái.
4. Hệ thống cập nhật dữ liệu và hiển thị thông báo phù hợp.

Luồng phụ

Luồng phụ: Tìm kiếm khướu nại

1. Admin nhập từ khóa tìm kiếm (mã khiếu nại, tên khách hàng) hoặc chọn các bộ lọc (trạng thái, khoảng thời gian).
2. Admin nhấn nút "Tìm kiếm".
3. Hệ thống hiển thị danh sách các khiếu nại phù hợp với tiêu chí tìm kiếm.
4. Nếu không có kết quả, hiển thị thông báo "Không tìm thấy khiếu nại nào phù hợp".

Luồng phụ: Xem chi tiết khướu nại

1. Admin chọn khiếu nại cần xem và nhấn nút "Xem chi tiết".
2. Hệ thống hiển thị thông tin chi tiết bao gồm:
   * Thông tin người gửi
   * Thông tin liên quan (booking, phòng, dịch vụ...)
   * Nội dung khiếu nại đầy đủ
   * Hình ảnh/tài liệu đính kèm (nếu có)
   * Lịch sử trao đổi và xử lý
   * Các khiếu nại liên quan từ cùng khách hàng (nếu có)
3. Admin có thể xem các thông tin liên quan như lịch sử đặt phòng, thông tin thanh toán để nắm bắt tổng thể vấn đề.

Luồng phụ: Thực hiện đồng ý/hủy bỏ với khướu nại

1. Admin chọn khiếu nại cần xử lý và nhấn nút "Xử lý khiếu nại".
2. Hệ thống hiển thị form xử lý với các lựa chọn:
   * Đồng ý với khiếu nại (chấp nhận)
   * Hủy bỏ khiếu nại (từ chối)
   * Yêu cầu thêm thông tin
3. Admin chọn hành động, nhập phản hồi chi tiết và các bước đã thực hiện để giải quyết.
4. Nếu đồng ý với khiếu nại, admin có thể chọn hình thức bồi thường:
   * Hoàn tiền (toàn bộ hoặc một phần)
   * Cung cấp dịch vụ miễn phí
   * Nâng cấp phòng/dịch vụ
   * Tặng điểm thưởng/voucher
5. Hệ thống cập nhật trạng thái khiếu nại và gửi thông báo cho khách hàng.
6. Khiếu nại được chuyển sang trạng thái "đã xử lý" hoặc "đang chờ phản hồi" tùy thuộc vào hành động.

Điều kiện sau

* Trạng thái khiếu nại được cập nhật trong hệ thống.
* Khách hàng nhận được thông báo về việc xử lý khiếu nại.
* Các hành động bồi thường/khắc phục được ghi nhận và thực hiện.

Yêu cầu đặc biệt

* Thời gian phản hồi ban đầu cho mỗi khiếu nại không quá 24 giờ.
* Các khiếu nại nghiêm trọng cần được ưu tiên và đánh dấu đặc biệt.
* Hệ thống phải theo dõi thời gian xử lý của từng khiếu nại để đảm bảo không bị bỏ sót.

Use case liên quan

* Đăng nhập (include)
* Quản lý user
* Khướu nại tài khoản (khách hàng)
  + 1. Use case quản lý nhân viên

A diagram of a company

AI-generated content may be incorrect.

Hình 3.11 Use case quản lý nhân viên

Tác nhân

Admin

Mô tả

Use case cho admin quản lý thông tin và phân công nhân viên trong khách sạn.

Điều kiện trước

* Admin đã đăng nhập vào hệ thống.

Luồng sự kiện chính

1. Admin truy cập vào mục "Quản lý nhân viên" từ menu chính.
2. Hệ thống hiển thị danh sách nhân viên với thông tin như mã nhân viên, tên, vị trí, phòng ban, trạng thái làm việc.
3. Admin có thể thực hiện các hành động sau:
   * Tìm kiếm nhân viên: Nhập từ khóa tìm kiếm hoặc lọc theo vị trí, phòng ban, trạng thái.
   * Thêm nhân viên mới: Nhập thông tin nhân viên mới vào hệ thống.
   * Sửa thông tin nhân viên: Cập nhật thông tin cá nhân và công việc của nhân viên.
   * Xóa nhân viên: Xóa nhân viên khỏi hệ thống hoặc đánh dấu là đã nghỉ việc.
4. Hệ thống cập nhật dữ liệu và hiển thị thông báo phù hợp.

Luồng phụ

Luồng phụ: Tìm kiếm nhân viên

1. Admin nhập từ khóa tìm kiếm (tên, mã nhân viên) hoặc chọn các bộ lọc (vị trí, phòng ban, trạng thái).
2. Admin nhấn nút "Tìm kiếm".
3. Hệ thống hiển thị danh sách nhân viên phù hợp với tiêu chí tìm kiếm.
4. Nếu không có kết quả, hiển thị thông báo "Không tìm thấy nhân viên nào phù hợp".

Luồng phụ: Thêm nhân viên mới

1. Admin chọn chức năng "Thêm nhân viên".
2. Hệ thống hiển thị form nhập thông tin nhân viên mới.
3. Admin nhập thông tin bao gồm:
   * Thông tin cá nhân (tên, ngày sinh, giới tính, CMND/CCCD, địa chỉ)
   * Thông tin liên hệ (email, số điện thoại)
   * Thông tin công việc (vị trí, phòng ban, ngày bắt đầu làm việc)
   * Thông tin tài khoản (tên đăng nhập, mật khẩu, quyền truy cập)
   * Tải lên ảnh đại diện (nếu có)
4. Admin nhấn nút "Lưu" để hoàn tất.
5. Hệ thống kiểm tra tính hợp lệ của thông tin và lưu nhân viên mới vào cơ sở dữ liệu.

Luồng phụ: Sửa nhân viên

1. Admin chọn nhân viên cần sửa và nhấn nút "Sửa".
2. Hệ thống hiển thị form với thông tin hiện tại của nhân viên.
3. Admin cập nhật thông tin cần thiết.
4. Admin nhấn nút "Cập nhật" để lưu thay đổi.
5. Hệ thống cập nhật dữ liệu và hiển thị thông báo thành công.

Luồng phụ: Xóa nhân viên

1. Admin chọn nhân viên cần xóa và nhấn nút "Xóa".
2. Hệ thống hiển thị hộp thoại xác nhận với các lựa chọn:
   * Xóa hoàn toàn (đối với nhân viên chưa có dữ liệu hoạt động)
   * Đánh dấu là đã nghỉ việc (giữ lại lịch sử hoạt động)
3. Admin chọn phương thức xóa và xác nhận.
4. Hệ thống cập nhật trạng thái nhân viên hoặc xóa khỏi hệ thống tùy theo lựa chọn.

Điều kiện sau

* Thông tin nhân viên được cập nhật trong hệ thống.
* Quyền truy cập của nhân viên được cập nhật tương ứng.

Yêu cầu đặc biệt

* Bảo mật thông tin cá nhân của nhân viên.
* Khả năng quản lý lịch làm việc và phân công công việc.
* Hỗ trợ đánh giá hiệu suất làm việc và theo dõi lịch sử công việc.

Use case liên quan

* Đăng nhập (include)
* Phân công công việc
* Quản lý lịch làm việc
  + 1. Use case xem phòng

A diagram of a diagram

AI-generated content may be incorrect.

Hình 3.13 Use case xem phòng

Tác nhân

Khách hàng

Mô tả

Use case cho khách hàng xem thông tin chi tiết về phòng tại khách sạn.

Điều kiện trước

* Khách hàng đã truy cập vào website của khách sạn.

Luồng sự kiện chính

1. Khách hàng truy cập vào mục "Phòng" hoặc chọn xem phòng từ trang chủ.
2. Hệ thống hiển thị danh sách các loại phòng với thông tin cơ bản và hình ảnh thu nhỏ.
3. Khách hàng chọn một phòng cụ thể để xem chi tiết.
4. Hệ thống hiển thị thông tin chi tiết về phòng, bao gồm:
   * Tên và loại phòng
   * Hình ảnh phòng (nhiều góc nhìn)
   * Mô tả chi tiết
   * Giá phòng (có thể thay đổi theo mùa/sự kiện)
   * Tiện nghi phòng
   * Kích thước phòng và sức chứa
   * Đánh giá và bình luận từ khách hàng trước
5. Khách hàng có thể thực hiện các hành động sau:
   * Đặt phòng: Tiến hành đặt phòng này ngay.
   * Để lại bình luận: Viết đánh giá hoặc bình luận về phòng (nếu đã từng ở).
   * Chọn phòng khác cùng loại: Xem các phòng tương tự trong cùng hạng mục.

Luồng phụ

Luồng phụ: Để lại bình luận

1. Khách hàng chọn chức năng "Để lại bình luận".
2. Hệ thống kiểm tra xem khách hàng đã đăng nhập chưa:
   * Nếu chưa, yêu cầu đăng nhập/đăng ký trước.
   * Nếu đã đăng nhập, hệ thống kiểm tra lịch sử đặt phòng để xác nhận khách đã từng sử dụng phòng này.
3. Hệ thống hiển thị form bình luận với các trường:
   * Đánh giá (1-5 sao)
   * Tiêu đề bình luận
   * Nội dung chi tiết
4. Khách hàng nhập thông tin và gửi bình luận.
5. Hệ thống lưu bình luận và hiển thị thông báo thành công.

Luồng phụ: Đặt phòng

1. Khách hàng chọn chức năng "Đặt phòng".
2. Hệ thống chuyển hướng đến chức năng đặt phòng với thông tin phòng đã được lựa chọn sẵn.

Luồng phụ: Chọn phòng khác cùng loại

1. Khách hàng chọn chức năng "Xem phòng tương tự".
2. Hệ thống hiển thị danh sách các phòng cùng loại hoặc có đặc điểm tương tự.
3. Khách hàng có thể chọn phòng khác để xem chi tiết.

Điều kiện sau

* Khách hàng có đầy đủ thông tin về phòng để quyết định có đặt phòng hay không.
* Hệ thống ghi nhận các tương tác của khách hàng để cá nhân hóa đề xuất trong tương lai.

Yêu cầu đặc biệt

* Hình ảnh phòng phải chất lượng cao và đa dạng góc nhìn.
* Thông tin phòng phải được cập nhật thường xuyên và chính xác.
* Hiển thị tính khả dụng của phòng theo thời gian thực.
* Tối ưu hóa tốc độ tải trang và hình ảnh.

Use case liên quan

* Đặt phòng
* Tìm kiếm phòng
* Quản lý bookings (khách hàng)
  + 1. Use case đăng nhập khách hàng

Ảnh có chứa văn bản, biểu đồ, vòng tròn, Phông chữ

Mô tả được tạo tự động

Hình 3.15 Use case đăng nhập khách hàng

Tác nhân: khách hàng.

Mô tả: cho phép khách hàng đăng nhập vào hệ thống.

Luồng sự kiện chính:

1. Khách hàng chọn chức năng đăng nhập.
2. Hiển thị Form đăng nhập và Nhập đủ thông tin.
3. Hệ thống kiểm tra thông tin.
4. Nếu đăng nhập thành công thì hiển thị thông tin, nếu sau thì chuyển sang luồng nhánh A1.
5. Kết thúc.

Luồng sự kiện rẽ nhánh:

* Nhánh A1: Khách hàng đăng nhập không thành công.

1. Hệ thống thông báo sai thông tin đăng nhập.
2. Chọn đăng kí hay nhập lại, nếu chọn đăng ký thì chuyển sang form đăng ký tài khoản, nếu khách hàng chọn nhập lại thì chuyển về bước 2 luồng sự kiện chính.
   * 1. Usecase đặt phòng

A diagram of a company

AI-generated content may be incorrect.

Hình 3.17 Use case đặt phòng

Tác nhân

Khách hàng

Mô tả

Use case cho khách hàng đặt phòng trong hệ thống khách sạn.

Điều kiện trước

* Khách hàng đã truy cập vào website của khách sạn.
* Có phòng còn trống trong thời gian khách hàng muốn đặt.

Luồng sự kiện chính

1. Khách hàng chọn mục "Đặt phòng" hoặc nhấn "Đặt ngay" từ trang xem chi tiết phòng.
2. Hệ thống yêu cầu khách hàng đăng nhập (nếu chưa đăng nhập).
3. Khách hàng thực hiện các bước đặt phòng bao gồm:
   * Chọn loại phòng cần đặt: Lựa chọn từ danh sách loại phòng có sẵn.
   * Chọn ngày đặt: Xác định ngày check-in và check-out.
   * Chọn giờ đặt: Xác định giờ check-in dự kiến.
   * Chọn loại booking muốn đặt: Lựa chọn loại đặt phòng (có thể hủy/không thể hủy, trả trước/trả sau).
4. Hệ thống kiểm tra tính khả dụng của phòng và hiển thị các lựa chọn phù hợp.
5. Khách hàng xác nhận thông tin đặt phòng và tiến hành thanh toán (nếu cần).
6. Hệ thống xác nhận đặt phòng thành công và gửi thông tin xác nhận đến khách hàng.

Luồng phụ

Luồng phụ: Chọn loại phòng cần đặt

1. Hệ thống hiển thị danh sách các loại phòng có sẵn với thông tin cơ bản (hình ảnh, tên, mô tả ngắn, giá).
2. Khách hàng xem và chọn loại phòng phù hợp.
3. Hệ thống hiển thị thông tin chi tiết hơn về loại phòng đã chọn, bao gồm tiện nghi và chính sách.

Luồng phụ: Chọn ngày đặt

1. Hệ thống hiển thị lịch với các ngày có phòng trống được đánh dấu.
2. Khách hàng chọn ngày check-in và check-out.
3. Hệ thống kiểm tra và xác nhận tính khả dụng của phòng trong khoảng thời gian đã chọn.

Luồng phụ: Chọn giờ đặt

1. Hệ thống hiển thị các khung giờ check-in có sẵn.
2. Khách hàng chọn giờ check-in dự kiến.
3. Hệ thống ghi nhận thông tin và cập nhật kế hoạch đón tiếp.

Luồng phụ: Chọn loại booking muốn đặt

1. Hệ thống hiển thị các loại đặt phòng:
   * Đặt trước (có thể hủy với hoàn tiền đầy đủ trước ngày X)
   * Đặt không hoàn tiền (giá rẻ hơn nhưng không thể hủy)
   * Đặt linh hoạt (có thể thay đổi ngày với phí phụ thu)
2. Khách hàng chọn loại đặt phòng phù hợp với nhu cầu.
3. Hệ thống hiển thị các điều khoản và điều kiện tương ứng.

Điều kiện sau

* Thông tin đặt phòng được lưu vào hệ thống.
* Phòng được đánh dấu là đã đặt trong khoảng thời gian tương ứng.
* Khách hàng nhận được email/SMS xác nhận đặt phòng với mã đặt phòng.

Yêu cầu đặc biệt

* Hệ thống cập nhật tính khả dụng của phòng theo thời gian thực.
* Chính sách hủy và thay đổi đặt phòng phải được hiển thị rõ ràng.
* Thông tin thanh toán phải được bảo mật theo tiêu chuẩn ngành.

Use case liên quan

* Đăng nhập (include)
* Xem phòng
* Thanh toán
* Quản lý bookings (khách hàng)
  + 1. Use case đặt đồ ăn/ dịch vụ

A diagram of a diagram

AI-generated content may be incorrect.

Hình 3.18 Use case đặt đồ ăn/ dịch vụ

Tác nhân

Khách hàng

Mô tả

Use case cho khách hàng đặt đồ ăn/đồ uống và các dịch vụ bổ sung trong thời gian lưu trú tại khách sạn.

Điều kiện trước

* Khách hàng đã đăng nhập vào tài khoản.
* Khách hàng đang có booking đã check-in (đang lưu trú tại khách sạn).

Luồng sự kiện chính

1. Khách hàng truy cập vào mục "Dịch vụ phòng" hoặc "Đặt đồ ăn/dịch vụ" từ trang cá nhân hoặc ứng dụng di động.
2. Hệ thống yêu cầu khách hàng xác nhận booking đang sử dụng.
3. Khách hàng thực hiện các bước đặt hàng:
   * Chọn booking đã check-in: Xác nhận phòng đang sử dụng.
   * Chọn đồ ăn/dịch vụ: Lựa chọn từ menu hiện có.
   * Chọn phương thức thanh toán: Trả trước hoặc tính vào hóa đơn phòng.
   * Chọn phương thức vận chuyển: Giao tận phòng hoặc tự đến nhận.
4. Hệ thống kiểm tra và xác nhận đơn hàng.
5. Khách hàng xác nhận đặt hàng.
6. Hệ thống ghi nhận đơn hàng và chuyển thông tin đến bộ phận liên quan.

Luồng phụ

Luồng phụ: Chọn booking đã check-in

1. Hệ thống hiển thị danh sách các booking hiện tại đã check-in của khách hàng.
2. Khách hàng chọn booking (phòng) cần đặt dịch vụ.
3. Hệ thống ghi nhận thông tin và hiển thị các dịch vụ có sẵn cho loại phòng này.

Luồng phụ: Chọn đồ ăn/dịch vụ

1. Hệ thống hiển thị menu với các danh mục:
   * Đồ ăn (bữa sáng, bữa trưa, bữa tối, đồ ăn nhẹ)
   * Đồ uống (nóng, lạnh, có cồn, không cồn)
   * Dịch vụ bổ sung (spa, giặt ủi, đưa đón)
2. Khách hàng duyệt qua menu và chọn các món cần đặt, điều chỉnh số lượng.
3. Hệ thống cập nhật giỏ hàng và tính tổng tiền tạm thời.
4. Khách hàng có thể thêm ghi chú đặc biệt cho từng món (ví dụ: không cay, không hành...).

Luồng phụ: Chọn phương thức thanh toán

1. Hệ thống hiển thị các phương thức thanh toán có sẵn:
   * Thanh toán online ngay
   * Tính vào hóa đơn phòng
   * Thanh toán khi nhận hàng
2. Khách hàng chọn phương thức thanh toán phù hợp.
3. Nếu chọn thanh toán online, hệ thống chuyển đến cổng thanh toán an toàn.

Luồng phụ: Chọn phương thức vận chuyển

1. Hệ thống hiển thị các lựa chọn vận chuyển:
   * Giao tận phòng (với thời gian dự kiến)
   * Tự đến nhận tại nhà hàng/quầy dịch vụ
2. Khách hàng chọn phương thức vận chuyển mong muốn.
3. Nếu chọn giao tận phòng, khách hàng có thể xác nhận phòng và thời gian giao.

Điều kiện sau

* Đơn hàng được ghi nhận trong hệ thống.
* Bộ phận liên quan (nhà bếp, dịch vụ) nhận được thông báo về đơn hàng mới.
* Khách hàng nhận được xác nhận đặt hàng với mã đơn và thời gian dự kiến.

Yêu cầu đặc biệt

* Menu phải cập nhật theo thời gian thực, hiển thị các món không còn sẵn có.
* Thời gian giao hàng phải chính xác và có thông báo nếu có sự chậm trễ.
* Khả năng đáp ứng các yêu cầu đặc biệt về ăn uống (dị ứng, chế độ ăn kiêng).

Use case liên quan

* Đăng nhập (include)
* Quản lý bookings (khách hàng)
* Quản lý orders (khách hàng)
* Thanh toán
  + 1. Use case quản lý thông tin cá nhân

A diagram with text and words

AI-generated content may be incorrect.

Hình 3.19 Use case quản lý thông tin cá nhân

Tác nhân

Khách hàng

Mô tả

Use case cho khách hàng xem và cập nhật thông tin cá nhân, quản lý các bookings và orders đã đặt.

Điều kiện trước

* Khách hàng đã đăng nhập vào tài khoản của mình.

Luồng sự kiện chính

1. Khách hàng truy cập vào mục "Tài khoản" hoặc "Thông tin cá nhân" từ menu chính.
2. Hệ thống hiển thị trang quản lý thông tin cá nhân với các phần:
   * Thông tin cơ bản (tên, email, số điện thoại...)
   * Lịch sử đặt phòng
   * Lịch sử đặt đồ ăn/dịch vụ
   * Điểm thưởng và ưu đãi
3. Khách hàng có thể thực hiện các hành động sau:
   * Thay đổi thông tin: Cập nhật thông tin cá nhân.
   * Quản lý booking: Xem và quản lý các đặt phòng.
   * Quản lý đồ ăn/dịch vụ đã đặt: Xem lịch sử các đơn hàng dịch vụ.
4. Hệ thống lưu trữ và cập nhật thông tin theo yêu cầu của khách hàng.

Luồng phụ

Luồng phụ: Thay đổi thông tin

1. Khách hàng chọn chức năng "Thay đổi thông tin cá nhân".
2. Hệ thống hiển thị form với thông tin hiện tại của khách hàng.
3. Khách hàng cập nhật các thông tin cần thiết:
   * Thông tin cá nhân (tên, ngày sinh, giới tính)
   * Thông tin liên hệ (địa chỉ, số điện thoại)
   * Thông tin bổ sung (sở thích, yêu cầu đặc biệt)
4. Khách hàng nhấn nút "Lưu thay đổi".
5. Hệ thống xác thực thông tin:
   * Nếu hợp lệ: Cập nhật thông tin và hiển thị thông báo thành công.
   * Nếu không hợp lệ: Hiển thị thông báo lỗi và yêu cầu sửa đổi.

Luồng phụ: Quản lý booking

1. Khách hàng chọn chức năng "Quản lý booking".
2. Hệ thống chuyển hướng đến trang quản lý bookings, hiển thị danh sách các đặt phòng hiện tại và lịch sử.
3. Khách hàng có thể xem chi tiết, hủy, hoặc thay đổi đặt phòng (nếu chính sách cho phép).

Luồng phụ: Quản lý đồ ăn/dịch vụ đã đặt

1. Khách hàng chọn chức năng "Quản lý đồ ăn/dịch vụ đã đặt".
2. Hệ thống hiển thị lịch sử các đơn hàng dịch vụ, bao gồm:
   * Đơn hàng đang xử lý
   * Đơn hàng đã hoàn thành
   * Đơn hàng đã hủy
3. Khách hàng có thể xem chi tiết đơn hàng, theo dõi trạng thái, hoặc hủy đơn nếu còn trong thời gian cho phép.

Điều kiện sau

* Thông tin cá nhân được cập nhật trong hệ thống.
* Khách hàng có thể xem và quản lý các dịch vụ đã đặt.

Yêu cầu đặc biệt

* Bảo mật thông tin cá nhân theo quy định GDPR hoặc các luật bảo vệ dữ liệu liên quan.
* Khả năng xuất dữ liệu cá nhân theo yêu cầu.
* Lưu trữ lịch sử thay đổi thông tin để phục vụ giải quyết tranh chấp.

Use case liên quan

* Đăng nhập (include)
* Quản lý bookings (khách hàng)
* Quản lý orders (khách hàng)
* Đổi mật khẩu
  + 1. Use case quản lý bookings

A diagram of a company

AI-generated content may be incorrect.

Hình 3.20 Use case quản lý bookings

Tác nhân

Khách hàng

Mô tả

Use case cho khách hàng xem, theo dõi và quản lý các đặt phòng của mình.

Điều kiện trước

* Khách hàng đã đăng nhập vào tài khoản.
* Khách hàng đã thực hiện ít nhất một lần đặt phòng.

Luồng sự kiện chính

1. Khách hàng truy cập vào mục "Đặt phòng của tôi" hoặc "Quản lý bookings" từ trang cá nhân.
2. Hệ thống hiển thị danh sách các đặt phòng với thông tin cơ bản (mã đặt phòng, ngày, loại phòng, trạng thái).
3. Khách hàng có thể thực hiện các hành động sau:
   * Xem chi tiết booking: Xem thông tin chi tiết về đặt phòng.
   * Hủy booking: Hủy đặt phòng (nếu chính sách cho phép).
   * Tìm kiếm: Tìm kiếm đặt phòng theo các tiêu chí khác nhau.
4. Hệ thống cập nhật thông tin đặt phòng theo yêu cầu của khách hàng.

Luồng phụ

Luồng phụ: Xem chi tiết booking

1. Khách hàng chọn một đặt phòng cụ thể từ danh sách và nhấn "Xem chi tiết".
2. Hệ thống hiển thị thông tin chi tiết về đặt phòng, bao gồm:
   * Thông tin phòng (loại, số phòng nếu đã được phân bổ)
   * Thời gian check-in và check-out
   * Thanh toán (tổng tiền, đã thanh toán, còn lại)
   * Trạng thái đặt phòng hiện tại
   * Các dịch vụ bổ sung đã đặt
   * Chính sách hủy và thay đổi
3. Khách hàng có thể tải xuống xác nhận đặt phòng hoặc hóa đơn (nếu có).

Luồng phụ: Hủy booking

1. Khách hàng chọn đặt phòng cần hủy và nhấn nút "Hủy booking".
2. Hệ thống kiểm tra chính sách hủy và hiển thị:
   * Phí hủy phòng (nếu có)
   * Tiền hoàn trả (nếu có)
   * Thời gian xử lý hoàn tiền
3. Khách hàng xác nhận hủy đặt phòng.
4. Hệ thống cập nhật trạng thái đặt phòng thành "Đã hủy" và xử lý hoàn tiền nếu áp dụng.
5. Hệ thống gửi email xác nhận hủy đặt phòng cho khách hàng.

Luồng phụ: Tìm kiếm

1. Khách hàng nhập thông tin tìm kiếm (mã đặt phòng, khoảng thời gian, trạng thái).
2. Hệ thống hiển thị các đặt phòng phù hợp với tiêu chí tìm kiếm.
3. Khách hàng có thể sắp xếp kết quả theo ngày, trạng thái hoặc giá trị.

Điều kiện sau

* Thông tin đặt phòng được cập nhật trong hệ thống.
* Khách hàng nhận được thông báo về các thay đổi đối với đặt phòng.

Yêu cầu đặc biệt

* Hiển thị rõ ràng các chính sách hủy và thay đổi đặt phòng.
* Thông báo kịp thời qua email/SMS khi có thay đổi trạng thái đặt phòng.
* Khả năng xuất thông tin đặt phòng dưới dạng PDF để in.

Use case liên quan

* Đăng nhập (include)
* Đặt phòng
* Quản lý thông tin cá nhân
* Thanh toán
  + 1. Use case quản lý order

A diagram of a diagram

AI-generated content may be incorrect.

Hình 3.21 Use case quản lý orders

Tác nhân

Khách hàng

Mô tả

Use case cho khách hàng xem, theo dõi và quản lý các đơn đặt đồ ăn/dịch vụ của mình.

Điều kiện trước

* Khách hàng đã đăng nhập vào tài khoản.
* Khách hàng đã thực hiện ít nhất một lần đặt đồ ăn/dịch vụ.

Luồng sự kiện chính

1. Khách hàng truy cập vào mục "Đơn hàng của tôi" hoặc "Quản lý orders" từ trang cá nhân.
2. Hệ thống hiển thị danh sách các đơn hàng với thông tin cơ bản (mã đơn, ngày đặt, tổng tiền, trạng thái).
3. Khách hàng có thể thực hiện các hành động sau:
   * Xem chi tiết order: Xem thông tin chi tiết về đơn hàng.
   * Hủy order: Hủy đơn hàng (nếu chưa xử lý).
   * Tìm kiếm order: Tìm kiếm đơn hàng theo các tiêu chí khác nhau.
   * Sửa order: Sửa đổi đơn hàng (nếu chưa xử lý).
4. Hệ thống cập nhật thông tin đơn hàng theo yêu cầu của khách hàng.

Luồng phụ

Luồng phụ: Xem chi tiết order

1. Khách hàng chọn một đơn hàng cụ thể từ danh sách và nhấn "Xem chi tiết".
2. Hệ thống hiển thị thông tin chi tiết về đơn hàng, bao gồm:
   * Danh sách các món đã đặt với số lượng và giá
   * Thời gian đặt và thời gian giao dự kiến
   * Phương thức thanh toán và trạng thái thanh toán
   * Trạng thái đơn hàng hiện tại
   * Ghi chú đặc biệt (nếu có)
3. Khách hàng có thể theo dõi trạng thái đơn hàng theo thời gian thực.

Luồng phụ: Hủy order

1. Khách hàng chọn đơn hàng cần hủy và nhấn nút "Hủy order".
2. Hệ thống kiểm tra trạng thái đơn hàng:
   * Nếu đơn hàng chưa xử lý: Cho phép hủy
   * Nếu đơn hàng đang xử lý hoặc đã giao: Thông báo không thể hủy
3. Khách hàng xác nhận hủy đơn hàng.
4. Hệ thống cập nhật trạng thái đơn hàng thành "Đã hủy" và xử lý hoàn tiền nếu đã thanh toán.
5. Hệ thống gửi thông báo xác nhận hủy đơn hàng.

Luồng phụ: Tìm kiếm order

1. Khách hàng nhập thông tin tìm kiếm (mã đơn, khoảng thời gian, trạng thái).
2. Hệ thống hiển thị các đơn hàng phù hợp với tiêu chí tìm kiếm.
3. Khách hàng có thể sắp xếp kết quả theo ngày, trạng thái hoặc giá trị.

Luồng phụ: Sửa order

1. Khách hàng chọn đơn hàng cần sửa và nhấn nút "Sửa order".
2. Hệ thống kiểm tra trạng thái đơn hàng:
   * Nếu đơn hàng chưa xử lý: Cho phép sửa
   * Nếu đơn hàng đang xử lý hoặc đã giao: Thông báo không thể sửa
3. Hệ thống hiển thị form sửa đơn hàng với thông tin hiện tại.
4. Khách hàng thực hiện các thay đổi (thêm/bớt món, điều chỉnh số lượng, thay đổi ghi chú).
5. Hệ thống cập nhật tổng tiền dựa trên các thay đổi.
6. Khách hàng xác nhận thay đổi.
7. Hệ thống cập nhật đơn hàng và gửi thông báo xác nhận.

Điều kiện sau

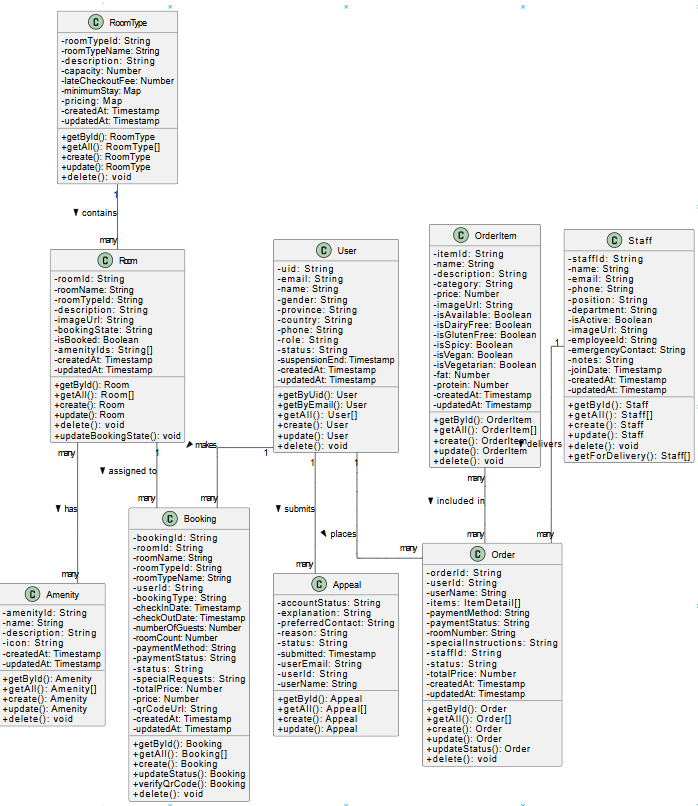
* Thông tin đơn hàng được cập nhật trong hệ thống.
* Khách hàng nhận được thông báo về các thay đổi đối với đơn hàng.

Yêu cầu đặc biệt

* Cập nhật trạng thái đơn hàng theo thời gian thực.
* Thông báo kịp thời khi đơn hàng sắp được giao.
* Khả năng đánh giá món ăn/dịch vụ sau khi hoàn thành.

Use case liên quan

* Đăng nhập (include)
* Đặt đồ ăn/dịch vụ
* Quản lý thông tin cá nhân
* Thanh toán
  1. Xây dựng biểu đồ lớp



Hình 3.23 Biểu đồ lớp

* 1. Biểu đồ hoạt động
     1. Biểu đồ hoạt động khách hàng

*3.5.1.1 Chức năng đăng nhập*

A picture containing text, diagram, screenshot, plan

Description automatically generated

Hình 3.24 Biểu đồ hoạt động chức năng đăng nhập

*3.5.1.2 Chức năng đăng ký*

A diagram of a work flow

Description automatically generated

Hình 3.25 Biểu đồ hoạt động chức năng đăng ký

* + - 1. *Chức năng đặt phòng*

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 3.26 Biểu đồ hoạt động chức năng đặt phòng

*3.5.1.4 Chức năng đặt đồ ăn/ dịch vụ*

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 3.27 Biểu đồ hoạt động chức năng đặt đồ ăn/ dịch vụ

* + 1. Biểu đồ hoạt động quản trị viên

3.5.2.1Chức năng quản lý phòng

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 3.28 Biểu đồ hoạt động chức năng quản lý phòng

3.5.2.2 Chức năng quản lý booking

A diagram of a process

AI-generated content may be incorrect.

Hình 3.29 Biểu đồ hoạt động chức năng quản lý bookings

*3.5.2.3 Chức năng quản lý order*

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 3.30 Biểu đồ hoạt động chức năng quản lý order

*3.5.2.6 Chức năng quản lý user*

*A diagram of a flowchart

AI-generated content may be incorrect.*

Hình 3.31 Biểu đồ hoạt động chức năng quản lý user

*3.5.2.7 Chức năng quản lý khướu nại(appeals)*

*A diagram of a work flow

AI-generated content may be incorrect.*

Hình 3.32 Biểu đồ hoạt động chức năng quản lý khướu nại

* 1. Thiết kế
     1. Biểu đồ tuần tự

3.6.1.1 Chức năng đăng ký

A picture containing text, diagram, line, parallel

Description automatically generated

Hình 3.33: Biểu đồ tuần tự đăng ký

3.6.1.2 Chức năng đăng nhập

A diagram of a diagram

AI-generated content may be incorrect.

Hình 3.34: Biểu đồ tuần tự đăng nhập

3.6.1.3 Chức năng đặt phòng

A diagram of a diagram

AI-generated content may be incorrect.

Hình 3.35: Biểu đồ tuần tự đặt phòng

3.6.1.4 Chức năng đặt đồ ăn/ dịch vụ

A diagram of a diagram

AI-generated content may be incorrect.

Hình 3.36: Biểu đồ tuần tự đặt đồ ăn

*3.6.1.5 Chức năng quản lý booking(admin)*

A diagram of a diagram

AI-generated content may be incorrect.

Hình 3.37: Biểu đồ tuần tự quản lý booking(admin)

* + 1. Thiết kế cơ sở dữ liệu

*-* Các collection:

* Amenities
* Appeals
* bookingRoom
* bookings
* comments
* orderItems
* orders
* roomTypes
* rooms
* staff
* users

- Cụ thể các collection:

**Collection: amenities (Tiện nghi)**

Lưu trữ thông tin về các tiện nghi của khách sạn hoặc phòng.

|  |  |
| --- | --- |
| **Tên trường** | **Giải thích** |
| amenitysId | ID định danh duy nhất cho tiện nghi. |
| createdAt | Thời gian tạo bản ghi tiện nghi. |
| description | Mô tả chi tiết về tiện nghi. |
| icon | Đường dẫn hoặc mã icon đại diện cho tiện nghi. |
| name | Tên của tiện nghi (ví dụ: "Wi-Fi miễn phí", "Bể bơi"). |
| updatedAt | Thời gian cập nhật bản ghi tiện nghi lần cuối. |

**Collection: appeals (Khiếu nại/Yêu cầu hỗ trợ)**

Lưu trữ thông tin về các khiếu nại hoặc yêu cầu hỗ trợ từ người dùng.

|  |  |
| --- | --- |
| **Tên trường** | **Giải thích** |
| accountsStatus | Trạng thái tài khoản liên quan đến khiếu nại (nếu có). |
| explanation | Giải trình hoặc mô tả chi tiết về vấn đề/khiếu nại. |
| preferredContact | Phương thức liên hệ ưu tiên của người dùng (ví dụ: email, điện thoại). |
| reason | Lý do chính của khiếu nại/yêu cầu. |
| status | Trạng thái xử lý của khiếu nại (ví dụ: "Mới", "Đang xử lý", "Đã giải quyết"). |
| submitted | Thời gian gửi khiếu nại/yêu cầu. |
| userEmail | Email của người dùng gửi khiếu nại. |
| userId | ID định danh của người dùng gửi khiếu nại. |
| username | Tên người dùng (username) của người gửi. |

**Collection: bookingRooms (Phòng trong Đặt phòng)**

Lưu trữ chi tiết về từng phòng cụ thể trong một đơn đặt phòng.

|  |  |
| --- | --- |
| **Tên trường** | **Giải thích** |
| bookingId | ID của đơn đặt phòng liên quan. |
| createdAt | Thời gian tạo bản ghi phòng trong đặt phòng. |
| price | Giá của phòng này trong đơn đặt phòng cụ thể. |
| roomId | ID của phòng được đặt. |
| roomName | Tên của phòng được đặt. |
| roomTypeId | ID của loại phòng. |
| roomTypeName | Tên của loại phòng. |

**Collection: bookings (Đặt phòng)**

Lưu trữ thông tin về các đơn đặt phòng của khách hàng.

|  |  |
| --- | --- |
| **Tên trường** | **Giải thích** |
| bookingId | ID định danh duy nhất cho đơn đặt phòng. |
| bookingType | Loại đặt phòng (ví dụ: "Online", "Tại quầy", "Qua điện thoại"). |
| checkInDate | Ngày nhận phòng dự kiến. |
| checkOutDate | Ngày trả phòng dự kiến. |
| createdAt | Thời gian tạo đơn đặt phòng. |
| numberOfGuests | Số lượng khách trong đơn đặt phòng. |
| paymentMethod | Phương thức thanh toán (ví dụ: "Thẻ tín dụng", "Tiền mặt", "Chuyển khoản"). |
| paymentStatus | Trạng thái thanh toán (ví dụ: "Chưa thanh toán", "Đã thanh toán", "Hoàn tiền"). |
| qrCodeUrl | URL của mã QR cho việc check-in nhanh hoặc xác nhận đặt phòng. |
| roomCount | Số lượng phòng được đặt trong đơn này. |
| specialRequests | Các yêu cầu đặc biệt từ khách hàng (ví dụ: phòng không hút thuốc, giường phụ). |
| status | Trạng thái của đơn đặt phòng (ví dụ: "Đã xác nhận", "Đã hủy", "Đã check-in"). |
| totalPrice | Tổng giá trị của đơn đặt phòng. |
| updatedAt | Thời gian cập nhật đơn đặt phòng lần cuối. |
| userId | ID của người dùng thực hiện đặt phòng. |
|  |  |

**Collection: comments (Bình luận/Đánh giá)**

Lưu trữ các bình luận hoặc đánh giá của khách hàng về phòng hoặc dịch vụ.

|  |  |
| --- | --- |
| **Tên trường** | **Giải thích** |
| commentId | ID định danh duy nhất cho bình luận. |
| createdAt | Thời gian tạo bình luận. |
| rating | Điểm đánh giá (ví dụ: từ 1 đến 5 sao). |
| roomId | ID của phòng được bình luận (nếu có). |
| text | Nội dung chi tiết của bình luận. |
| updatedAt | Thời gian cập nhật bình luận lần cuối. |
| userId | ID của người dùng viết bình luận. |

**Collection: orderItems (Mặt hàng trong Đơn hàng Dịch vụ)**

Lưu trữ thông tin chi tiết về các mặt hàng dịch vụ (ví dụ: đồ ăn, thức uống tại nhà hàng).

|  |  |
| --- | --- |
| **Tên trường** | **Giải thích** |
| allergens | Thông tin về các chất gây dị ứng có trong mặt hàng. |
| calories | Lượng calo của mặt hàng. |
| carbs | Lượng carbohydrate của mặt hàng (tính bằng gram). |
| category | Danh mục của mặt hàng (ví dụ: "Món chính", "Tráng miệng", "Đồ uống"). |
| createdAt | Thời gian tạo bản ghi mặt hàng. |
| description | Mô tả chi tiết về mặt hàng. |
| fat | Lượng chất béo của mặt hàng (tính bằng gram). |
| imageUrl | URL hình ảnh của mặt hàng. |
| isAvailable | Trạng thái còn hàng/sẵn có của mặt hàng (true/false). |
| isDairyFree | Mặt hàng có không chứa sữa không (true/false). |
| isGlutenFree | Mặt hàng có không chứa gluten không (true/false). |
| isSpicy | Mặt hàng có vị cay không (true/false). |
| isVegan | Mặt hàng có phải là đồ thuần chay không (true/false). |
| isVegetarian | Mặt hàng có phải là đồ chay không (true/false). |
| itemId | ID định danh duy nhất cho mặt hàng. |
| name | Tên của mặt hàng. |
| price | Giá của mặt hàng. |
| protein | Lượng protein của mặt hàng (tính bằng gram). |
| updatedAt | Thời gian cập nhật bản ghi mặt hàng lần cuối. |

**Collection: orders (Đơn hàng Dịch vụ)**

Lưu trữ thông tin về các đơn hàng dịch vụ của khách (ví dụ: đặt đồ ăn tại phòng, dịch vụ spa).

|  |  |
| --- | --- |
| **Tên trường** | **Giải thích** |
| createdAt | Thời gian tạo đơn hàng. |
| deliveryAddress | Địa chỉ giao hàng (nếu có, ví dụ: số phòng). |
| items | Danh sách các mặt hàng trong đơn hàng (mảng các đối tượng). |
| items.itemId | ID của mặt hàng trong danh sách. |
| items.name | Tên của mặt hàng trong danh sách. |
| items.price | Giá của một đơn vị mặt hàng trong danh sách. |
| items.quantity | Số lượng của mặt hàng trong danh sách. |
| items.total | Tổng giá của mặt hàng này (price \* quantity). |
| orderId | ID định danh duy nhất cho đơn hàng. |
| paymentMethod | Phương thức thanh toán cho đơn hàng. |
| paymentStatus | Trạng thái thanh toán của đơn hàng. |
| roomNumber | Số phòng của khách đặt hàng (nếu áp dụng). |
| specialInstructions | Hướng dẫn đặc biệt cho đơn hàng. |
| staffId | ID của nhân viên xử lý đơn hàng (nếu có). |
| status | Trạng thái của đơn hàng (ví dụ: "Đang chờ xử lý", "Đang giao", "Hoàn thành"). |
| totalPrice | Tổng giá trị của đơn hàng. |
| updatedAt | Thời gian cập nhật đơn hàng lần cuối. |
| userId | ID của người dùng đặt hàng. |
| userName | Tên của người dùng đặt hàng. |

**Collection: roomTypes (Loại phòng)**

Lưu trữ thông tin về các loại phòng khác nhau trong khách sạn.

|  |  |
| --- | --- |
| **Tên trường** | **Giải thích** |
| capacity | Sức chứa tối đa của loại phòng (số người). |
| createdAt | Thời gian tạo bản ghi loại phòng. |
| description | Mô tả chi tiết về loại phòng. |
| lateCheckoutFee | Phí trả phòng trễ (nếu có). |
| minimumStay | Số đêm ở tối thiểu yêu cầu (nếu có). |
| pricing | Đối tượng chứa thông tin giá cho các hình thức thuê khác nhau. |
| pricing.daily | Giá thuê theo ngày. |
| pricing.monthly | Giá thuê theo tháng. |
| pricing.weekly | Giá thuê theo tuần. |
| pricing.dayUse | Giá thuê sử dụng trong ngày (không qua đêm). |
| pricing.hourly | Đối tượng chứa thông tin giá thuê theo giờ. |
| pricing.hourly.basePrice | Giá cơ bản cho một khoảng thời gian thuê theo giờ nhất định. |
| pricing.hourly.additionalHourPrice | Giá cho mỗi giờ thuê thêm sau khoảng thời gian cơ bản. |
| pricing.overnight | Giá thuê qua đêm. |
| roomTypeId | ID định danh duy nhất cho loại phòng. |
| roomTypeName | Tên của loại phòng (ví dụ: "Standard", "Deluxe", "Suite"). |
| updatedAt | Thời gian cập nhật bản ghi loại phòng lần cuối. |

**Collection: rooms (Phòng)**

Lưu trữ thông tin chi tiết về từng phòng cụ thể trong khách sạn.

|  |  |
| --- | --- |
| **Tên trường** | **Giải thích** |
| amenityIds | Mảng các ID tiện nghi có trong phòng này. |
| bookingState | Trạng thái đặt của phòng (ví dụ: "Available", "Booked", "Maintenance"). |
| createdAt | Thời gian tạo bản ghi phòng. |
| description | Mô tả chi tiết về phòng. |
| imageUrl | URL hình ảnh của phòng. |
| isBooked | Trạng thái phòng đã được đặt hay chưa (true/false) - có thể dư thừa nếu có bookingState. |
| roomId | ID định danh duy nhất cho phòng. |
| roomName | Tên hoặc số phòng (ví dụ: "Phòng 101", "Phòng Hướng Biển A"). |
| roomTypeId | ID của loại phòng mà phòng này thuộc về. |
| updatedAt | Thời gian cập nhật bản ghi phòng lần cuối. |

**Collection: staff (Nhân viên)**

Lưu trữ thông tin về các nhân viên của khách sạn.

|  |  |
| --- | --- |
| **Tên trường** | **Giải thích** |
| createdAt | Thời gian tạo hồ sơ nhân viên. |
| department | Bộ phận làm việc của nhân viên (ví dụ: "Lễ tân", "Buồng phòng", "Kinh doanh"). |
| email | Địa chỉ email của nhân viên. |
| emergencyContact | Thông tin liên hệ khẩn cấp của nhân viên. |
| employeeId | Mã số nhân viên. |
| imageUrl | URL hình ảnh đại diện của nhân viên. |
| isActive | Trạng thái hoạt động của tài khoản nhân viên (true/false). |
| joinDate | Ngày nhân viên bắt đầu làm việc. |
| name | Họ và tên của nhân viên. |
| notes | Ghi chú thêm về nhân viên. |
| phone | Số điện thoại của nhân viên. |
| position | Chức vụ của nhân viên (ví dụ: "Quản lý", "Nhân viên lễ tân"). |
| staffId | ID định danh duy nhất cho nhân viên. |
| updatedAt | Thời gian cập nhật hồ sơ nhân viên lần cuối. |

**Collection: users (Người dùng)**

Lưu trữ thông tin về tài khoản của khách hàng và có thể cả quản trị viên (phân biệt bằng role).

|  |  |
| --- | --- |
| **Tên trường** | **Giải thích** |
| country | Quốc gia của người dùng. |
| createdAt | Thời gian tạo tài khoản người dùng. |
| email | Địa chỉ email của người dùng (thường dùng để đăng nhập). |
| gender | Giới tính của người dùng. |
| name | Họ và tên của người dùng. |
| phone | Số điện thoại của người dùng. |
| province | Tỉnh/Thành phố của người dùng. |
| role | Vai trò của người dùng trong hệ thống (ví dụ: "customer", "admin", "staff"). |
| status | Trạng thái tài khoản (ví dụ: "active", "inactive", "suspended"). |
| suspensionEnd | Thời gian kết thúc đình chỉ tài khoản (nếu status là "suspended"). |
| uid | ID định danh duy nhất cho người dùng (thường là từ hệ thống xác thực). |
| updatedAt | Thời gian cập nhật thông tin người dùng lần cuối. |

1. DEMO CHƯƠNG TRÌNH
   1. Giao diện khách hàng(+App/Web)
      1. Các thanh cấu tạo nên trang khách hàng

A screenshot of a phone

AI-generated content may be incorrect.

Hình 4.1 Giao diện nav\_header\_main

* + 1. Giao diện trang danh sách phòng

A screenshot of a cell phone

AI-generated content may be incorrect.

Hình 4.2: Giao diện trang danh sách phòng

A screenshot of a phone

AI-generated content may be incorrect.

Hình 4.3: Giao diện danh sách phòng - Web

* + 1. Giao diện trang chi tiết phòng

A screenshot of a phone

AI-generated content may be incorrect.

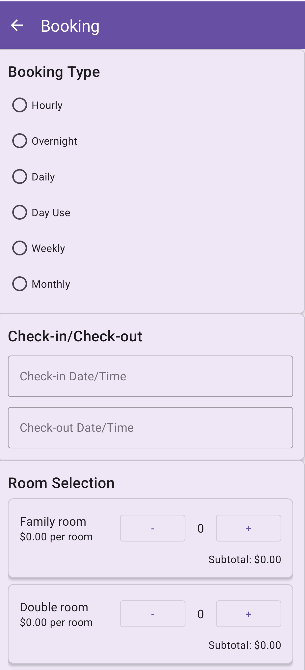
Hình 4.4: Giao diện trang chi tiết phòng

A screenshot of a web page

AI-generated content may be incorrect.

Hình 4.5: Giao diện chi tiết phòng – Web

* + 1. Giao diện trang đặt phòng



Hình 4.6 Giao diện trang đặt phòng – điện thoại 1

A screenshot of a phone

AI-generated content may be incorrect.

Hình 4.7 Giao diện trang đặt phòng – điện thoại 2

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.8 Giao diện trang đặt phòng(Web)

* + 1. Giao diện thông báo đặt phòng thành công(Hóa đơn)

A screenshot of a phone

AI-generated content may be incorrect.

Hình 4.9 Giao diện thông báo thành công

A screenshot of a hotel

AI-generated content may be incorrect.

Hình 4.10 Giao diện hóa đơn (Web)

* + 1. Giao diện trang đăng nhập/ đăng ký

A screenshot of a login screen

AI-generated content may be incorrect.A screenshot of a login form

AI-generated content may be incorrect.A screenshot of a login form

AI-generated content may be incorrect.

Hình 4.11 Giao diện đăng nhập/ đăng ký/ quên mật khẩu

* + 1. Giao diện trang my booking

Screens screenshot of a phone

AI-generated content may be incorrect.

Hình 4.12: Giao diện trang my booking

*A screenshot of a computer

AI-generated content may be incorrect.*

Hình 4.13: Giao diện trang my booking- Web

* + 1. Giao diện menu order

A screenshot of a phone

AI-generated content may be incorrect.A white background with pink and black lines

AI-generated content may be incorrect.

Hình 4.14: Giao diện dining/ cart – điện thoại

**A screenshot of a menu

AI-generated content may be incorrect.**

Hình 4.15: Giao diện trang dining -web

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.16: Giao diện trang cart- web

* 1. Giao diện trang quản trị(Web)
     1. Giao diện trang admin

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.17: Giao diện trang admin 1

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.18: Giao diện trang admin

* + 1. Giao diện trang quản lý booking

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.19: Giao diện trang danh sách booking

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.20: Giao diện trang cập nhật booking 1

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4.21: Giao diện trang cập nhật booking 2

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## **1. Kết luận**

Website Hệ thống quản lý khách sạn Lembo's Luxurious Hotel đã được phát triển thành công với mục tiêu số hóa và tối ưu hóa các hoạt động quản lý khách sạn, cung cấp nền tảng toàn diện cho cả khách hàng và nhân viên quản lý. Dự án này đã được xây dựng dựa trên nền tảng Node.js, Express, và Firebase, tạo ra một hệ thống linh hoạt và mạnh mẽ phục vụ nhu cầu kinh doanh hiện đại của khách sạn.

**Những tính năng đã đạt được:**

* **Hệ thống quản trị toàn diện**: Trang quản trị dành cho admin với khả năng quản lý phòng, đặt phòng, loại phòng, tiện nghi, nhân viên, người dùng và các thao tác khác.
* **Quản lý phòng và loại phòng**: Hệ thống cho phép tạo, cập nhật, xóa thông tin phòng và loại phòng với đầy đủ chi tiết như giá, tiện nghi, mô tả và hình ảnh.
* **Đặt phòng trực tuyến**: Khách hàng có thể dễ dàng xem và đặt phòng trực tuyến với đầy đủ tính năng như chọn ngày, số lượng khách, và yêu cầu đặc biệt.
* **Quản lý dịch vụ phòng**: Hệ thống đặt đồ ăn và dịch vụ phòng tích hợp, cho phép khách đặt món từ menu và theo dõi đơn hàng.
* **Hệ thống xác thực người dùng**: Đăng ký, đăng nhập, và quản lý tài khoản người dùng với phân quyền rõ ràng giữa khách hàng và quản trị viên.
* **Quản lý nhân viên**: Quản lý thông tin và phân công nhiệm vụ cho nhân viên khách sạn.
* **Quản lý đánh giá và bình luận**: Khách hàng có thể để lại đánh giá và nhận xét sau khi sử dụng dịch vụ.
* **Hệ thống QR code cho check-in/check-out**: Tích hợp mã QR để đơn giản hóa quy trình check-in và check-out cho khách hàng.

**Khó khăn gặp phải trong quá trình phát triển:**

* **Tích hợp với Firebase**: Việc sử dụng cả Firebase Client SDK và Admin SDK đồng thời tạo ra nhiều thách thức về cấu hình và quản lý kết nối. Đặc biệt, việc xử lý các timestamp và references giữa các collection trong Firestore đòi hỏi cách tiếp cận cẩn thận. Nhiều lần tôi gặp lỗi với timestamps khi chuyển đổi giữa dạng lưu trữ của Firestore và JavaScript Date objects.
* **Xử lý đồng bộ/bất đồng bộ**: Quản lý các hoạt động bất đồng bộ, đặc biệt trong các tác vụ phức tạp như đặt phòng và xử lý đơn đặt dịch vụ, đòi hỏi phải xử lý nhiều promises và async/await cùng lúc. Điều này trở nên cực kỳ khó khăn khi cần đảm bảo dữ liệu nhất quán giữa nhiều collections như rooms, bookings, users.
* **Quản lý session và authentication**: Tôi gặp nhiều vấn đề khi triển khai hệ thống xác thực kết hợp Firebase Authentication với session phía máy chủ, đặc biệt khi xử lý các trường hợp như quên mật khẩu, tài khoản bị khóa, hay phiên đăng nhập hết hạn. Việc đồng bộ trạng thái giữa Firebase Auth và session là một thách thức lớn.
* **Xử lý và lưu trữ file**: Tôi phải xây dựng hệ thống tải lên và quản lý hình ảnh từ đầu thay vì sử dụng Firebase Storage, điều này đòi hỏi nhiều công sức để đảm bảo tính bảo mật và hiệu quả. Việc tạo các thư mục con cho từng loại upload và xử lý các tên file trùng lặp khá phức tạp.
* **Thiết kế cơ sở dữ liệu NoSQL**: Việc thiết kế cấu trúc dữ liệu trên Firestore (NoSQL) khác hoàn toàn so với kinh nghiệm trước đây của tôi với SQL. Tôi gặp nhiều thách thức trong việc thiết kế các mối quan hệ giữa entities như room-roomType-amenities hay order-orderItems.
* **Xử lý tình huống người dùng bị hạn chế**: Việc triển khai hệ thống xử lý tài khoản bị đình chỉ hoặc cấm mà vẫn cho phép người dùng gửi kháng nghị đòi hỏi logic phức tạp, đặc biệt là trong việc giới hạn quyền truy cập nhưng vẫn cho phép một số tính năng hoạt động.
* **Hiệu suất và tối ưu hóa truy vấn**: Firestore có giới hạn về loại truy vấn và không hỗ trợ nhiều phép join phức tạp, nên tôi phải nghĩ ra các giải pháp thay thế như lưu cache và xử lý trong bộ nhớ cho một số tính năng như lọc phòng hoặc hiển thị đánh giá.

2. Hướng phát triển

Cải tiến ngắn hạn:

* Hoàn thiện giao diện người dùng: Cải thiện trải nghiệm người dùng với thiết kế responsive hơn và tối ưu cho các thiết bị di động. Cụ thể, tôi muốn thêm animations và transitions để tạo cảm giác cao cấp và sang trọng phù hợp với thương hiệu khách sạn.
* Tối ưu hóa quy trình đặt phòng: Đơn giản hóa quy trình đặt phòng từ 5 bước hiện tại xuống còn 3 bước, giảm tỷ lệ bỏ giỏ hàng. Thêm tính năng giữ phòng tạm thời trong 15 phút để khách hàng hoàn tất thanh toán.
* Hoàn thiện hệ thống QR code: Triển khai đầy đủ hệ thống check-in/check-out bằng QR code, bao gồm cả ứng dụng dành cho nhân viên lễ tân để quét mã và xác nhận khách.
* Tích hợp thanh toán trực tuyến: Hiện tại hệ thống mới chỉ hỗ trợ thanh toán tại khách sạn, tôi muốn tích hợp các cổng thanh toán như Stripe, PayPal và các phương thức thanh toán địa phương.

Phát triển dài hạn:

* Hệ thống quản lý nhiều cơ sở: Mở rộng hệ thống để quản lý nhiều cơ sở khách sạn với tính năng phân quyền và báo cáo tập trung. Đây là yếu tố quan trọng vì công ty dự định mở thêm 3 cơ sở trong 2 năm tới.
* Hệ thống phân tích và báo cáo nâng cao: Xây dựng dashboard với biểu đồ và số liệu trực quan về tỷ lệ lấp đầy, doanh thu, đánh giá khách hàng, và các KPI quan trọng khác để hỗ trợ ra quyết định kinh doanh.
* Tích hợp AI và học máy: Phát triển thuật toán dynamic pricing tự động điều chỉnh giá phòng dựa trên mùa du lịch, sự kiện trong thành phố, tỷ lệ lấp đầy và hành vi đặt phòng trước đây.
* Ứng dụng di động: Phát triển ứng dụng mobile native cho iOS và Android với tính năng đặt phòng, check-in/out không tiếp xúc, truy cập phòng bằng điện thoại, và đặt dịch vụ.

Tính năng kinh doanh mới:

* Gói combo và khuyến mãi: Phát triển hệ thống tạo và quản lý các gói combo (phòng + dịch vụ ăn uống + spa), chương trình khuyến mãi theo mùa và chương trình giảm giá cho khách hàng thành viên.
* Hệ thống thành viên và tích điểm: Xây dựng chương trình khách hàng thân thiết với nhiều cấp độ thành viên (Silver, Gold, Platinum), tích điểm và đổi ưu đãi.
* Trang blog và nội dung marketing: Tích hợp trang blog để chia sẻ thông tin về điểm đến, sự kiện địa phương, và mẹo du lịch để cải thiện SEO và tăng lưu lượng truy cập tự nhiên.
* Hệ thống tổ chức sự kiện: Phát triển module cho phép đặt và quản lý không gian hội nghị, tiệc cưới và sự kiện đặc biệt với tính năng tùy chỉnh layout, menu và dịch vụ đi kèm.
* Tích hợp dịch vụ taxi và tour du lịch: Mở rộng sang lĩnh vực dịch vụ lữ hành với tính năng đặt đưa đón sân bay, tour tham quan thành phố, và các hoạt động địa phương.

Lý do cho những phát triển này:

* Đáp ứng kỳ vọng của khách hàng hiện đại: Khách du lịch ngày nay, đặc biệt là thế hệ Millennials và Gen Z, mong đợi trải nghiệm số hóa hoàn chỉnh từ đặt phòng đến check-out.
* Vượt qua áp lực cạnh tranh: Các đối thủ lớn trong ngành như Marriott và Hilton đang đầu tư mạnh vào công nghệ, buộc chúng tôi phải nâng cấp để không bị tụt hậu.
* Tối ưu hóa vận hành và giảm chi phí: Tự động hóa nhiều quy trình sẽ giúp giảm 30% chi phí nhân sự và 25% chi phí hành chính, theo phân tích chi phí-lợi ích của chúng tôi.
* Ứng phó với thách thức hậu Covid-19: Ngành khách sạn đang phải thích nghi với những thay đổi dài hạn trong hành vi khách hàng, đặc biệt là nhu cầu về công nghệ không tiếp xúc và linh hoạt trong đặt/hủy phòng.
* Hỗ trợ kế hoạch mở rộng kinh doanh: Công ty có kế hoạch mở rộng từ 1 lên 4 cơ sở trong vòng 5 năm tới, và cần một hệ thống có khả năng mở rộng để hỗ trợ sự phát triển này.

TÀI LIỆU THAM KHẢO

[1] Thạc Bình Cường, Phân tích và thiết kế hệ thống thông tin, nhà xuất bản Khoa học và Kỹ thuật, 2002.

[2] Giáo trình môn lập trình hướng đối tượng - Lê Thị Mỹ Hạnh

[3] <https://vi.wikipedia.org/wiki/HTML>

[4] <https://vi.wikipedia.org/wiki/Laravel>

[5] <https://vi.wikipedia.org/wiki/Php>

[6] https://en.wikipedia.org/wiki/MySQL

[7]https://vi.wikipedia.org/wiki/CSS

# LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi. Những tài liệu trong đồ án tốt nghiệp là hoàn toàn trung thực. Các kết quả nghiên cứu do chính tôi thực hiện dưới sự chỉ đạo của giáo viên hướng dẫn.