

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM



ĐỒ ÁN TỐT NGHIỆP

TÊN HỌC PHẦN: ĐỒ ÁN TỐT NGHIỆP

**ĐỀ TÀI: XÂY DỰNG HỆ THỐNG QUẢN LÝ ĐẶT PHÒNG VÀ
QUẢN LÝ KHÁCH HÀNG CHO KHÁCH SẠN TRÊN NỀN TẢNG
ASP.NET Core WEB API.**

Giáo viên hướng dẫn: TS. LƯƠNG CAO ĐÔNG

Sinh viên thực hiện:

Mã sinh viên	Họ và tên	Lớp
1451020083	NGUYỄN TRỌNG HIẾU	CNTT 14-02

Hà Nội, tháng 4 năm 2024

BỘ GIÁO DỤC ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM



NGUYỄN TRỌNG HIẾU

XÂY DỰNG HỆ THỐNG QUẢN LÝ ĐẶT PHÒNG VÀ
QUẢN LÝ KHÁCH HÀNG CHO KHÁCH SẠN TRÊN
NỀN TẢNG ASP.NET Core WEB API

CHUYÊN NGÀNH : CÔNG NGHỆ THÔNG TIN

MÃ SỐ : 74.80.201

NGƯỜI HƯỚNG DẪN: TS. LƯƠNG CAO ĐÔNG

HÀ NỘI - 2024

LỜI CAM ĐOAN

Trước hết, em xin bày tỏ lòng biết ơn sâu sắc tới thầy **TS. LƯƠNG CAO ĐÔNG**, người đã dành nhiều thời gian và công sức để hướng dẫn, chỉ bảo và truyền đạt kiến thức quý báu cho em trong suốt quá trình thực hiện đề án này. Sự tận tâm, nhiệt tình và những góp ý, định hướng của thầy đã giúp em hoàn thiện bài đề án này một cách tốt nhất.

Em cũng xin chân thành cảm ơn Ban Giám hiệu, Phòng Đào tạo, cùng toàn thể các thầy cô giáo trong khoa Công nghệ thông tin - trường Đại học Đại Nam đã tạo điều kiện thuận lợi và cung cấp những kiến thức, kỹ năng cần thiết trong suốt quá trình học tập và nghiên cứu tại trường.

Em xin gửi lời cảm ơn chân thành tới các anh chị, bạn bè, đồng nghiệp đã luôn hỗ trợ, chia sẻ tài liệu, kinh nghiệm và động viên tôi trong quá trình nghiên cứu và hoàn thành đề án này.

Cuối cùng, em xin bày tỏ lòng biết ơn sâu sắc tới gia đình, đặc biệt là bố mẹ, đã luôn là nguồn động viên, khích lệ và tạo điều kiện tốt nhất cho tôi trong suốt thời gian học tập và nghiên cứu.

Hà Nội, tháng 05 năm 2024

Sinh viên thực hiện

Nguyễn Trọng Hiếu

LỜI CẢM ƠN

Lời đầu tiên em xin được bày tỏ lòng biết ơn sâu sắc đối với sự hỗ trợ tận tâm của Thầy trong suốt học phần xây dựng đồ án tốt nghiệp. Dưới sự dẫn dắt và hướng dẫn của Thầy, em đã không chỉ học được kiến thức chuyên sâu về lập trình web mà còn nhận thức được sức mạnh của công nghệ trong cuộc sống hàng ngày của chúng ta.

Thầy không chỉ là một người thầy, mà còn là nguồn động viên không ngừng cho em. Những hướng dẫn và lời khuyên của Thầy không chỉ là việc truyền đạt thông tin mà còn là hành trình khám phá với những ý tưởng sáng tạo và ứng dụng thực tế. Sự nhiệt huyết và lòng đam mê của Thầy đã là nguồn động viên lớn giúp em vượt qua các thách thức trong quá trình phát triển dự án này.

Em cũng muốn bày tỏ lòng biết ơn đặc biệt về sự kiên nhẫn của Thầy. Những lúc em gặp khó khăn và không hiểu rõ, Thầy luôn sẵn lòng giải đáp mọi thắc mắc của tôi một cách tỉ mỉ và thấu đáo. Điều này không chỉ giúp em hiểu sâu hơn về chủ đề mà còn giúp em phát triển kỹ năng tự học và tư duy logic.

Những lời nhận xét của thầy rất chi tiết và tỉ mỉ, em cảm thấy thật may mắn vì được một người dày dặn kinh nghiệm và trải nghiệm như thầy dẫn dắt em trong quá trình xây dựng và phát triển đề tài này. Em chắc chắn sẽ áp dụng những gì đã học được từ Thầy vào học tập và cuộc sống để không ngừng phát triển bản thân và sẽ sớm trở thành một chuyên gia lập trình web giỏi.

Một lần nữa, em xin thay mặt toàn thể những cá nhân sinh viên nói riêng và những tập thể được thầy dẫn dắt nói chung, chân thành cảm ơn Thầy vì tất cả những gì Thầy đã dành cho chúng em. Sự hiểu biết và lòng trân trọng của chúng em dành cho Thầy không thể nào diễn tả hết được. Mong rằng, trong tương lai, em có thể tiếp tục nhận được sự hướng dẫn và ủng hộ từ Thầy để có thể phát triển hơn nữa trong lĩnh vực này.

Em xin chân thành cảm ơn Thầy!

DANH MỤC VIẾT TẮT

API	Appication Program Interface
SWOT	Strengths – Weakness – Opportunities – Threats
MSSQL	Microsoft SQL Server
WCF	
MVC	Model – View - Controller
IIS	Internet Information Services
DB	Database
IDE	Integrated Development Environment
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
UC	Usecase
ID	Identity
CN	Chức năng

MỤC LỤC

DANH MỤC VIẾT TẮT	3
DANH MỤC HÌNH ẢNH	8
DANH MỤC BẢNG	9
DANH MỤC SƠ ĐỒ	10
CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI VÀ CÔNG NGHỆ.....	11
1.1. Tổng quan đề tài.....	11
1.1.2. Mô tả đề tài	11
1.2. Tổng quan công nghệ.....	13
1.2.1. API là gì?	13
1.2.2. Giới thiệu về ASP.NET Core.	16
1.2.3. ASP.NET Core Web API là gì?.....	16
1.3. Công cụ hỗ trợ.....	16
1.3.1. Microsoft SQL Server.	17
1.3.2. Visual Studio.	18
1.3.3. Visual Studio Code.....	18
1.3.4. Phân biệt Visual Studio và Visual Studio Code.....	18
CHƯƠNG II: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	20
2.1. Phát biểu bài toán.	20
2.1.1. Định nghĩa dự án.....	20
2.1.2. Mô tả vấn đề.....	20
2.1.3. Mô tả giải pháp.	20
2.2. Thiết kế chương trình.....	22
2.2.1. Tổng quan hệ thống.	22
2.2.2. Mô tả ngữ cảnh hệ thống.....	22
2.2.3. Các chức năng cơ bản của hệ thống.	23
2.2.4. Sự khác biệt so với các hệ thống hiện có.....	24

2.3. Công nghệ ràng buộc.....	25
2.3.1. Kỹ thuật phát triển hệ thống.	25
2.3.2. Môi trường phát triển.....	25
2.3.3. Các ràng buộc khác.....	26
2.4. Phân tích SWOT (Strengths – Weakness – Opportunities – Threats).....	26
2.5. Đặc tả yêu cầu hệ thống.	26
2.5.1. Yêu cầu chức năng.	26
2.5.2. Yêu cầu phi chức năng.....	27
2.5.3. Hạn chế và giả định.....	28
2.5.4. Tiêu chí chấp nhận.....	30
2.6. Thiết kế usecase hệ thống.....	32
2.6.1. Biểu đồ usecase tổng quát hệ thống.	32
2.6.2. Biểu đồ usecase cho từng chức năng.	36
2.6.3. Bảng chức năng và độ ưu tiên.....	54
2.6.4. Thiết kế sơ đồ thực thể liên kết.	58
2.7. Thiết kế database.	59
2.7.1. Lựa chọn cơ sở dữ liệu.....	59
2.7.2. Lược đồ cơ sở dữ liệu.	59
2.7.3. Định nghĩa bảng (table).	61
2.8. Thiết kế giao diện.....	68
2.8.1. Giao diện đăng nhập.	68
2.8.2. Giao diện đăng ký.....	69
2.8.3. Giao diện trang chủ user.....	69
2.8.4. Giao diện danh sách phòng user.....	70
2.8.5. Giao diện đặt phòng user.....	70
2.8.6. Giao diện thanh toán user.....	71

2.8.7. Giao diện quản lý tài khoản user.	71
2.8.8. Giao diện trang chủ admin.	72
2.8.9. Giao diện quản lý phòng admin.....	72
2.8.10. Giao diện quản lý user cho admin.	73
2.8.11. Giao diện quản lý Blog cho admin.....	73
CHƯƠNG III: CÀI ĐẶT HỆ THỐNG VÀ KIỂM THỬ	74
3.1. Các Api chức năng của hệ thống.	74
3.2. Giao diện hệ thống.....	78
3.2.1. Giao diện đăng nhập.	78
3.2.2. Giao diện đăng ký.....	79
3.2.3. Giao diện trang chủ user.....	79
3.2.4. Giao diện danh sách phòng user.....	80
3.2.5. Giao diện đặt phòng user.....	80
3.2.6. Giao diện thanh toán user.....	81
3.2.7. Giao diện quản lý tài khoản user.	82
3.2.8. Giao diện trang chủ admin.	82
3.2.9. Giao diện quản lý phòng admin.....	83
3.2.10. Giao diện quản lý user cho admin.	83
3.2.11. Giao diện quản lý Blog cho admin.....	84
3.3. Kiểm thử hệ thống.	84
3.3.1. Kiểm thử chức năng tìm kiếm phòng.	84
3.3.2. Kiểm thử chức năng đặt phòng trực tuyến.	85
3.3.3. Kiểm thử chức năng thanh toán online.	87
3.3.4. Kiểm thử chức năng quản lý phòng.	90
3.3.5. Kiểm thử chức năng quản lý doanh thu.....	92
3.3.6. Kiểm thử chức năng quản lý hoá đơn.	94

KẾT LUẬN	95
Tổng kết:.....	95
Hướng phát triển:	95
TÀI LIỆU THAM KHẢO.....	96

DANH MỤC HÌNH ẢNH

Hình 2. 1: Mô hình MVC trong dự án.....	21
Hình 2. 2: Layout giao diện đăng nhập	68
Hình 2. 3: Layout giao diện đăng ký.	69
Hình 2. 4: Layout giao diện trang chủ user.	69
Hình 2. 5: Layout giao diện danh sách phòng user	70
Hình 2. 6: Layout giao diện đặt phòng user.	70
Hình 2. 7: Layout giao diện thanh toán user.....	71
Hình 2. 8: Layout giao diện quản lý tài khoản user.....	71
Hình 2. 9: Layout giao diện trang chủ admin.....	72
Hình 2. 10: Layout giao diện quản lý phòng admin.	72
Hình 2. 11: Layout giao diện quản lý user cho admin.....	73
Hình 2. 12: Layout giao diện quản lý bài viết cho admin.	73
Hình 3. 1: Giao diện Swagger API definition 1.0 cho user.	74
Hình 3. 2: Các api dùng để xác thực người dùng và thay đổi mật khẩu	74
Hình 3. 3: Các api dùng để người dùng xem và tìm kiếm blog	74
Hình 3. 4: Các api thanh toán bằng Momo và VNPAY	75
Hình 3. 5: API dùng cho người dùng quản lý thông tin phòng.	75
Hình 3. 6: API quản lý thông tin cá nhân và thông tin thanh toán cho người dùng.	75
Hình 3. 7: API quản lý thông tin hoá đơn cho người dùng.	76
Hình 3. 8: Giao diện Swagger API definition 2.0 cho admin.....	76
Hình 3. 9: Các Api quản lý xác thực và quản lý tài khoản cho admin	76
Hình 3. 10: Api quản lý phòng dành cho admin.....	77
Hình 3. 11: Các Api quản lý giảm giá và quản lý dịch vụ phòng cho admin.....	77
Hình 3. 12: Api phân quyền cho admin.....	77
Hình 3. 13: Api thống kê doanh thu cho admin.....	78
Hình 3. 14: Giao diện đăng nhập bằng email và mật khẩu.	78
Hình 3. 15: Giao diện đăng ký bằng tên – email – số điện thoại - và mật khẩu.	79
Hình 3. 16: Giao diện trang chủ user sau khi đăng nhập thành công.	79
Hình 3. 17: Giao diện danh sách phòng cho user sau khi lựa chọn tìm kiếm.	80
Hình 3. 18: Giao diện đặt phòng cho user sau khi lựa chọn booking phòng.	80

Hình 3. 19: Giao diện thanh toán Momo cho user khi chọn phương thức thanh toán Momo. [5]	81
Hình 3. 20: Giao diện thanh toán bằng VN Pay cho user khi chọn phương thức thanh toán VnPay. [5]	81
Hình 3. 21: Giao diện quản lý tài khoản cho user khi muốn xem lại thông tin của mình.	82
Hình 3. 22: Giao diện trang chủ admin sau khi đăng nhập bằng tài khoản Admin.....	82
Hình 3. 23: Giao diện quản lý phòng admin khi muốn xem danh sách và thông tin phòng.	83
Hình 3. 24: Giao diện quản lý user cho admin khi muốn xem danh sách người dùng.	83
Hình 3. 25: Giao diện quản lý blog cho admin.	84

DANH MỤC BẢNG

Bảng 2. 1: So sánh sự khác biệt với các hệ thống hiện có	24
Bảng 2. 2: Phân tích SWOT	26
Bảng 2. 3: Mô tả các tác nhân hệ thống.	33
Bảng 2. 4: Danh sách các usecase trong hệ thống.....	34
Bảng 2. 5: Đặc tả các thành phần trong usecase	35
Bảng 2. 6: Đặc tả usecase quản lý tài khoản.	37
Bảng 2. 7: Đặc tả usecase quản lý người dùng.....	39
Bảng 2. 8: Đặc tả usecase quản lý phòng.	42
Bảng 2. 9: Đặc tả usecase quản lý hoá đơn.	45
Bảng 2. 10: Đặc tả usecase quản lý phản hồi – liên hệ.	47
Bảng 2. 11: Đặc tả usecase thống kê.	49
Bảng 2. 12: Đặc tả usecase quản lý blog.....	52
Bảng 2. 13: Bảng chi tiết chức năng và độ ưu tiên.....	54
Bảng 2. 14: Table ServiceRoom.	61
Bảng 2. 15: Table ServiceAttach	61
Bảng 2. 16: Table ServiceAttachDetail	61
Bảng 2. 17: Table RoomStar.....	62
Bảng 2. 18: Table Room	62
Bảng 2. 19: Table OderService.....	62
Bảng 2. 20: Table Reservation.....	63
Bảng 2. 21: Table ReservationChat	63

Bảng 2. 22: Table Discount	63
Bảng 2. 23: Table DiscountReservationDetail	64
Bảng 2. 24: Table ApsNetUser	64
Bảng 2. 25: Table DiscountType	65
Bảng 2. 26: Table ReservationPayment.....	65
Bảng 2. 27: Table Blog	65
Bảng 2. 28: Table BlogTypeDetail	65
Bảng 2. 29: Table BlogType	66
Bảng 2. 30: Table InvoiceReservation.....	66
Bảng 2. 31: Table DiscountServiceDetail.....	66
Bảng 2. 32: Table DiscountRoomDetail.....	66
Bảng 2. 33: Table RoomType	66
Bảng 2. 34: Table Contact	67

Bảng 3. 1: Bảng mô tả kiểm thử tìm kiếm phòng	84
Bảng 3. 2: Bảng mô tả kiểm thử đặt phòng online.....	85
Bảng 3. 3: Bảng mô tả kiểm thử thanh toán online.	87
Bảng 3. 4: Bảng mô tả kiểm thử quản lý phòng.....	90
Bảng 3. 5: Bảng mô tả kiểm thử quản lý doanh thu.	92
Bảng 3. 6: Bảng mô tả kiểm thử quản lý hoá đơn.....	94

DANH MỤC SƠ ĐỒ

Sơ đồ 2. 1: Sơ đồ tổng quan hệ thống.	22
Sơ đồ 2. 2: Usecase tổng quát hệ thống.....	33
Sơ đồ 2. 3: Usecase Quản lý tài khoản.....	37
Sơ đồ 2. 4: Usecase quản lý người dùng	39
Sơ đồ 2. 5: Usecase quản lý phòng.....	42
Sơ đồ 2. 6: Usecase quản lý hoá đơn.....	45
Sơ đồ 2. 7: Usecase quản lý phản hồi – liên hệ	47
Sơ đồ 2. 8: Usecase thống kê.....	49
Sơ đồ 2. 9: Usecase quản lý blog.....	52
Sơ đồ 2. 10: Sơ đồ thực thể liên kết.	59

CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI VÀ CÔNG NGHỆ

1.1. Tổng quan đề tài.

Việc xây dựng một trang web quản lý đặt phòng và quản lý khách hàng cho khách sạn không chỉ đơn thuần là việc tạo ra một giao diện trực quan mà còn là cách để tôn vinh và kể câu chuyện của một không gian ẩm thực đặc biệt. Trang web không chỉ đơn giản là một công cụ truyền thông, mà là bức tranh tường trình bày về kiến trúc của người sáng tạo.

Khách sạn không chỉ là nơi cung cấp chỗ nghỉ ngơi mà còn là điểm đến tinh tế, nơi gặp gỡ và trải nghiệm. Website quản lý đặt phòng và quản lý khách hàng cho khách sạn không chỉ là nền tảng truyền thông, mà còn là một cửa sổ mở ra thế giới của kiến trúc sư, view đa dạng và dịch vụ chuyên nghiệp.

Từ việc thiết kế giao diện hấp dẫn, tới việc thể hiện thông tin chi tiết về danh sách các loại phòng đa dạng, không gian trang trí tinh tế và chất lượng dịch vụ, việc xây dựng một trang web quản lý đặt phòng và quản lý khách hàng cho khách sạn đòi hỏi sự tinh tế và sáng tạo. Mỗi chi tiết trên trang web không chỉ là một mẫu thông tin mà còn là một câu chuyện, một trải nghiệm tuyệt vời mà khách hàng sẽ không quên.

1.1.2. Mô tả đề tài

Mục tiêu đề tài:

Đề tài xây dựng quản lý đặt phòng và quản lý khách hàng cho khách sạn không chỉ đề cập đến việc tạo ra một giao diện trực quan và thu hút, mà còn góp phần vào việc tạo nên trải nghiệm đầy đủ cho người dùng. Mục tiêu của dự án này là xây dựng một nền tảng trực tuyến đồng nhất, mở cửa sổ tới không gian đặc biệt của khách sạn, thể hiện sự chuyên nghiệp và tận tâm trong dịch vụ.

Qua việc tập trung vào việc thiết kế giao diện sáng tạo, dễ sử dụng và phản ánh đúng bản sắc của khách sạn, đề tài này đặt ra mục tiêu tạo ra một trang web tương tác, thân thiện với người dùng, giúp họ dễ dàng tìm hiểu thông tin về lựa chọn phòng khách sạn đa dạng, không gian ẩm cúng và những dịch vụ chất lượng.

Mục tiêu cũng tập trung vào việc nâng cao trải nghiệm người dùng, đưa ra thông tin chi tiết và hấp dẫn về món ăn, đồ uống, không gian trang trí và các sự kiện đặc biệt tại khách sạn. Bằng cách thể hiện chân thực, dễ dàng tiếp cận thông tin, và tạo cảm nhận tốt,

mục tiêu của dự án là tạo ra một trang web không chỉ là điểm đến ảo mà còn là điểm đến trải nghiệm, thu hút và tạo sự gắn kết với khách hàng.

Tận dụng công nghệ để tạo ra một giao diện thú vị, chất lượng cao và chuyên nghiệp, đề tài này nhằm mục tiêu nâng cao trải nghiệm của khách hàng, tạo ấn tượng và tạo ra môi trường tương tác tốt nhất giữa khách sạn và khách hàng. [1]

Các tính năng chủ yếu bao gồm:

- Đăng nhập / Đăng ký / phân quyền.
- Tìm kiếm phòng.
- Đặt phòng.
- Quản lý phòng.
- Quản lý khách hàng.
- Quản lý bài viết.
- Quản lý phòng, thông tin phòng.
- Thanh toán.
- Thống kê doanh thu.
- Hiện thị giao diện khách hàng.

Công nghệ và ngôn ngữ lập trình:

- Angular: Được sử dụng để xây dựng giao diện cho chương trình.
- ASP.NET Core Web API: Để xây dựng và quản lý backend, xử lý logic thêm, sửa, xóa, tìm kiếm và giao tiếp với cơ sở dữ liệu.
- MSSQL Database: Lưu trữ dữ liệu phòng.
- Firebase: Lưu ảnh của người dùng và ảnh phòng.
- API Momo / VN Pay for Developer: Dùng để tạo môi trường thanh toán cho khách hàng.
- Mailtrap: Sử dụng để gửi email xác thực tài khoản cho khách hàng.

Phạm vi dự án:

Dự án sẽ tập trung vào việc phát triển các chức năng cơ bản cần thiết cho người dùng, bao gồm quản lý phòng, tìm kiếm, đặt phòng, thanh toán, quản lý dành cho admin. Giao diện người dùng sẽ được thiết kế sao cho dễ sử dụng và hấp dẫn.

Kiến trúc hệ thống:

Hệ thống sẽ được xây dựng theo kiến trúc MVC (Model – View – Controller), với frontend (View) sử dụng Angular, backend (Model – Controller) sử dụng ASP.NET Core Web API và cơ sở dữ liệu SQL. Các phần này sẽ tương tác với nhau để cung cấp trải nghiệm của người dùng.

1.2. Tổng quan công nghệ.

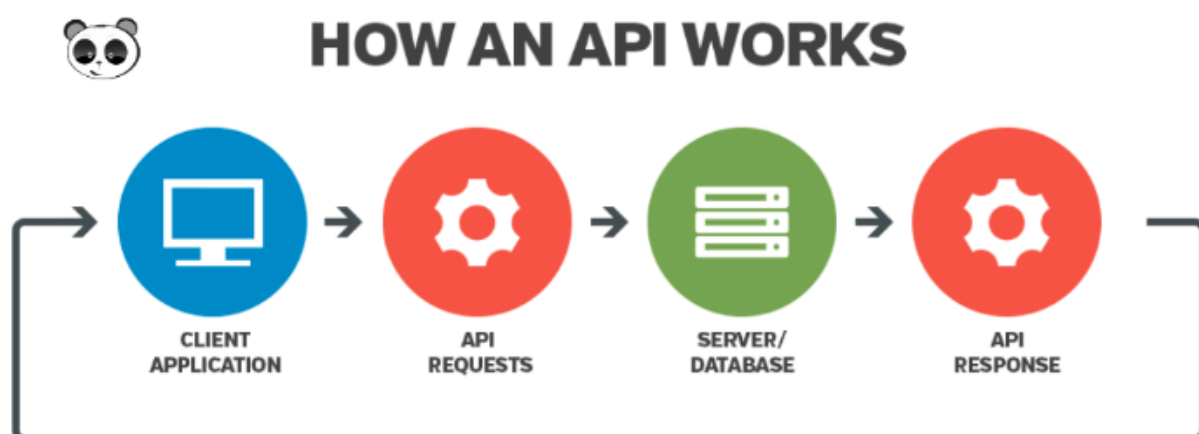
1.2.1. API là gì?

API được viết tắt bởi cụm từ tiếng Anh Application Programming Interface, tức giao diện lập trình ứng dụng. Đây là phương tiện cho hai hoặc nhiều ứng dụng trao đổi, tương tác với nhau, tạo ra tương tác giữa người dùng với ứng dụng hiệu quả và tiện lợi hơn.

Với API, các lập trình viên có thể tiếp cận, truy xuất dữ liệu từ máy chủ thể hiện chúng trên ứng dụng phần mềm hoặc website của mình một cách dễ dàng hơn.

Có thể hình dung như thế này: Giả sử bạn đang viết một phần mềm cho Windows 10 và bạn muốn phần mềm của mình có thể dễ dàng tương tác được với các tính năng hay dịch vụ của Windows như thay đổi hình nền, điều khiển con lăn, các thao tác liên quan tới giao diện người dùng hay thậm chí tính năng/ứng dụng cao cấp hơn. Tất cả điều đó đều có thể thực hiện được qua Windows API miễn phí dành cho Windows.

Tính tới nay, API đã phát triển với nhiều loại ứng dụng và phần mềm khác nhau. Thế hệ mới nhất của web/ app API có thể ứng dụng được ở mọi hệ thống từ cơ sở dữ liệu, hệ điều hành, hệ thống nền web, thư viện hay thậm chí là phần cứng máy tính.



Hình 1. 1: Cách hoạt động của API

Tên của một số phương thức API chính là các từ mô tả hành động của API. Một số hành động có thể xảy ra như:

- GET
- POST
- PUT
- DELETE

Ưu điểm của API

- API được tích hợp trên hầu hết các ứng dụng của desktop, mobile và các ứng dụng trên website;
- Có sự linh hoạt trong việc định dạng dữ liệu khác nhau khi được trả về client;
- API là một công cụ có mã nguồn mở, việc này giúp bạn có thể sử dụng API mọi lúc bất kể đâu với điều kiện là phải có kết nối với internet. Giúp hỗ trợ chức năng RESTful;
- Hỗ trợ giao tiếp hai chiều và được xác nhận trong những giao dịch khác nhau. Điều này giúp đảm bảo được độ tin cậy cao;
- Về cấu hình của API khi so sánh với cấu hình của WCF thì vô cùng thân thiện và đơn giản mang lại các trải nghiệm tuyệt vời cho người dùng;
- API hỗ trợ đầy đủ các thành phần MVC từ Model Binder, Action cho đến Unit Test, ...
- API giúp người dùng nhanh chóng xây dựng HTTP service điều này giúp công việc lập trình đơn giản hơn.

Nhược điểm của API

- API chỉ hỗ trợ mặc định POST, GET chứ chưa hoàn toàn như là RESTful service thông thường;
- Bạn cần có một kiến thức chuyên sâu thì việc sử dụng API mới thật sự hiệu quả;
- Để vận hành và phát triển API tốn rất nhiều thời gian cũng như chi phí;
- Hệ thống bảo mật sẽ bị tấn công nếu chủ sở hữu chưa giới hạn kỹ các điều kiện trên hệ thống.

Xu hướng phát triển của API

API hiện có 2 chính sách bảo vệ cơ bản gồm:

- Các công ty bảo vệ API của chính mình
- Các công ty chuyên cung cấp API miễn phí

Với loại thứ 1: ở đây các công ty thường sẽ chủ trương bảo vệ API và thu lợi từ các nhà phát triển phần mềm thứ 3 đăng ký, xin phép họ. Họ thường là những công ty sản xuất thiết bị, game, công nghệ, ví dụ như Sony cùng hệ thống playstation.

Với loại thứ 2: họ là các công ty chuyên cung cấp các phòng API miễn phí. Người dùng có thể thoải mái sử dụng API này để viết lên phần mềm bên thứ 3. Tuy nhiên, người dùng vẫn cần mua thêm phần mềm để sử dụng. Đây cũng chính là nguồn lợi của các nhà cung cấp API miễn phí, đồng thời vừa có thể đem tới hiệu quả lan tỏa và marketing mạnh mẽ hơn.

Điển hình cho loại thứ 2 đó là hệ sinh thái của Microsoft, Google hay Apple. Họ hầu như đều cung cấp các API miễn phí, đổi lại là các lập trình viên sẽ viết phần mềm hệ điều hành, người dùng phải mua của Window, Google để có thể sử dụng phần mềm đó. Đặc biệt trong đó, API của Google chính là ứng dụng trong hệ sinh thái giúp họ ngày càng phổ biến và dễ dàng tiếp cận mọi người khắp thế giới hơn.

Hiện nay, xu hướng người dùng chủ yếu thiên về loại thứ 2. Điều này khiến cho các API miễn phí ngày càng trở nên phổ biến hơn, vận hành và có mặt ở mọi nơi trên internet và cuộc sống.

Mức độ phổ biến của công cụ này ngày càng tăng khi chúng được nâng cao về các tiêu chuẩn về sự thân thiện, đơn giản, dễ sử dụng. Không những thế, tiêu chuẩn về bảo mật cũng được nâng cấp, giúp giảm thiểu các rủi ro hoặc khai thác các lỗ hổng bảo mật tốt hơn. Mọi hoạt động của API đều được giám sát nên hiệu suất của chúng cũng được cải thiện tốt hơn.

Với sự phát triển của các thiết bị di động và ứng dụng đi kèm, API giờ đây cũng được nâng cấp nhằm thích nghi tốt hơn với các thiết bị di động. Thực tế cho thấy, hầu hết các ứng dụng nền web và ứng dụng di động tận dụng nhiều API hơn, tạo động lực tăng trưởng API vượt bậc.

Trang Programmableweb cũng từng đánh giá, một trong những kho API miễn phí lớn nhất thế giới hiện nay có tới hơn 21.000 API được chia thành 450 mục khác nhau, bao phủ toàn bộ các lĩnh vực trong đời sống con người. Có thể thấy, API trở thành công cụ quan

trọng giúp cho hỏa động vận hành hệ thống tốt hơn, trở thành chìa khóa thông minh, đồng bộ và tự động hóa đặc biệt của con người.

1.2.2. Giới thiệu về ASP.NET Core.

ASP.NET Core được thiết kế nhằm đáp ứng các yêu cầu:

- Phát triển và hoạt động đa nền tảng
- Có kiến trúc dựa trên các module
- Phát triển hoàn toàn ở dạng mã nguồn mở
- Phù hợp với xu hướng hiện đại của ứng dụng web.

Để đạt được các yêu cầu trên, Microsoft xây dựng một platform mới đảm bảo nhẹ – nhanh – đa nền tảng. Platform mới này được đặt tên là .NET Core. Hiện nay .NET Core hoạt động được trên Windows, macOS và Linux.

.NET Core chứa rất nhiều API giống như của .NET Framework. Điều này giúp lập trình viên dễ dàng chuyển đổi sang .NET Core. Nếu nhìn từ phía các API thì có thể hình dung .NET Core tương tự như một bộ phận (độc lập) tách ra từ .NET Framework.

Ở những phiên bản đầu tiên .NET Core chỉ cho phép phát triển ứng dụng console đa nền tảng. ASP.NET Core là một tầng bổ sung xây dựng bên trên ứng dụng console để chuyển đổi nó thành một ứng dụng web.

ASP.NET Core có thể dễ dàng triển khai trên các hệ điều hành khác nhau, triển khai như một web server độc lập, hoặc kết hợp cùng các chương trình web server khác (IIS, Apache, Nginx). [8]

1.2.3. ASP.NET Core Web API là gì?

ASP.NET Core Web API là một khuôn khổ để xây dựng các dịch vụ HTTP có thể được truy cập từ bất kỳ ứng dụng khách nào bao gồm cả trình duyệt và thiết bị di động. Nó là một nền tảng lý tưởng để xây dựng các ứng dụng RESTful trên .NET Framework.

Các hướng dẫn về API Web này sẽ giúp người dùng tìm hiểu các kiến thức cơ bản về ASP.NET Core Web API bắt đầu từ cấp độ cơ bản đến nâng cao. Các hướng dẫn được chia thành các phần, trong đó mỗi phần chứa một số chủ đề liên quan được đóng gói với các giải thích dễ hiểu, các ví dụ thực tế, các mẹo hữu ích và phần ghi chú thông tin. [2]

1.3. Công cụ hỗ trợ.

1.3.1. Microsoft SQL Server.

SQL Server chính là một hệ quản trị dữ liệu quan hệ sử dụng câu lệnh SQL để trao đổi dữ liệu giữa máy cài SQL Server và máy Client. Một Relational Database Management System – RDBMS gồm có: databases, database engine và các chương trình ứng dụng dùng để quản lý các bộ phận trong RDBMS và những dữ liệu khác.

Các thành phần cơ bản trong SQL Server

Các thành phần cơ bản trong SQL Server gồm có: Reporting Services, Database Engine, Integration Services, Notification Services, Full Text Search Service, ... Tất cả kết hợp với nhau tạo thành một giải pháp hoàn chỉnh giúp cho việc phân tích và lưu trữ dữ liệu trở nên dễ dàng hơn.

- Database Engine: Đây là một engine có khả năng chứa dữ liệu ở các quy mô dưới dạng support và table. Ngoài ra, nó còn có khả năng tự điều chỉnh ví dụ: trả lại tài nguyên cho hệ điều hành khi một user log off và sử dụng thêm các tài nguyên của máy khi cần.
- Integration Services: là tập hợp các đối tượng lập trình và các công cụ đồ họa cho việc sao chép, di chuyển và chuyển đổi dữ liệu. Khi bạn làm việc trong một công ty lớn thì dữ liệu được lưu trữ ở nhiều nơi khác nhau như được chứa trong: Oracle, SQL Server, DB2, Microsoft Access, ... và bạn chắc chắn sẽ có nhu cầu di chuyển dữ liệu giữa các server này. Ngoài ra, bạn còn muốn định dạng dữ liệu trước khi lưu vào database. Chắc chắn Integration Services sẽ giúp bạn giải quyết được công việc này dễ dàng.
- Analysis Services: Đây là một dịch vụ phân tích dữ liệu rất hay của Microsoft. Dữ liệu khi được lưu trữ vào trong database mà bạn không thể lấy được những thông tin bổ ích thì coi như không có ý nghĩa gì. Chính vì thế, công cụ này ra đời giúp bạn trong việc phân tích dữ liệu một cách hiệu quả và dễ dàng bằng cách dùng kỹ thuật khai thác dữ liệu – datamining và khái niệm hình khối nhiều chiều – multi dimension cubes.
- Notification Services: Dịch vụ thông báo này là nền tảng cho sự phát triển và triển khai các ứng dụng soạn và gửi thông báo. Ngoài ra, dịch vụ này còn có chức năng gửi thông báo theo lịch thời đến hàng ngàn người đăng ký sử dụng trên nhiều loại thiết bị khác nhau.

- Reporting Services: là một công cụ tạo, quản lý và triển khai báo cáo bao gồm: server và client. Ngoài ra, nó còn là nền tảng cho việc phát triển và xây dựng các ứng dụng báo cáo.
- Full Text Search Service: là một thành phần đặc biệt trong việc truy vấn và đánh chỉ mục dữ liệu văn bản không cấu trúc được lưu trữ trong các cơ sở dữ liệu SQL Server.
- Service Broker: là một môi trường lập trình cho việc tạo ra các ứng dụng trong việc nhảy qua các Instance.

1.3.2. Visual Studio.

Visual studio là một trong những công cụ **hỗ trợ lập trình website** rất nổi tiếng nhất hiện nay của Microsoft và chưa có một phần mềm nào có thể thay thế được nó. Visual Studio được viết bằng 2 ngôn ngữ đó chính là C# và VB+. Đây là 2 ngôn ngữ lập trình giúp người dùng có thể lập trình được hệ thống một cách dễ dàng và nhanh chóng nhất thông qua Visual Studio.

Visual Studio là một phần mềm lập trình hệ thống được sản xuất trực tiếp từ Microsoft. Từ khi ra đời đến nay, Visual Studio đã có rất nhiều các phiên bản sử dụng khác nhau. Điều đó, giúp cho người dùng có thể lựa chọn được phiên bản tương thích với dòng máy của mình cũng như cấu hình sử dụng phù hợp nhất.

Bên cạnh đó, Visual Studio còn cho phép người dùng có thể tự chọn lựa giao diện chính cho máy của mình tùy thuộc vào nhu cầu sử dụng.

1.3.3. Visual Studio Code.

Visual Studio Code là trình soạn thảo, biên tập lập trình mã nguồn miễn phí được sử dụng trên 3 nền tảng đó là: Windows, macOS và Linux được xây dựng, phát triển bởi Microsoft. Visual Studio Code được các chuyên gia công nghệ thông tin đánh giá cao, nó là sự kết hợp hoàn hảo giữa IDE và CODE Editor.

Visual Studio Code còn có nhiệm vụ hỗ trợ các nền tảng như: JavaScript, TypeScript và Node.js. Bạn có thể hiểu cụ thể công việc của nó là mang đến một hệ sinh thái mới vô cùng phong phú cho các ngôn ngữ lập trình.

1.3.4. *Phân biệt Visual Studio và Visual Studio Code.*

Nhiều người đang nhầm lẫn giữa Visual Studio Code và Visual Code bởi chúng có tên gọi khá tương đồng với nhau. Thực chất, hai phiên bản này hoàn toàn khác nhau. Cụ thể:

- Visual Studio là một môi trường phát triển “đầy đủ tính năng” và “tiện lợi”.
- Visual Studio là một trình soạn thảo văn bản trên phần mềm.
- Visual Studio Code là trình chỉnh sửa văn bản đa nền tảng bao gồm Linux, macOS, Windows. Và nó có thể được mở rộng với các plugin theo nhu cầu của bạn.
- Visual Studio chỉ có thể chạy 1 trong hai nền tảng Windows hoặc Mac. Đây là hai hệ điều hành khác nhau.
- Tốc độ xử lý của Visual Studio Code được đánh giá là nhanh hơn so với Visual Studio.
- Dung lượng của Visual Studio Code nhỏ hơn Visual Studio.

CHƯƠNG II: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1. Phát biểu bài toán.

2.1.1. Định nghĩa dự án.

Đây là website quản lý cho khách sạn và đặt phòng trực tuyến cho khách hàng, với mục đích giúp cho khách hàng dễ dàng đặt được phòng từ xa mà không cần phải di chuyển đến tận nơi, giúp cho khách hàng tiết kiệm được thời gian và tạo sự thuận lợi khi đặt phòng khách sạn. Bên cạnh đó, website này còn giúp cho quản lý và nhân viên của khách sạn dễ dàng quản lý và kiểm soát được thông tin đặt phòng của khách hàng, thống kê được các doanh số của khách sạn.

2.1.2. Mô tả vấn đề.

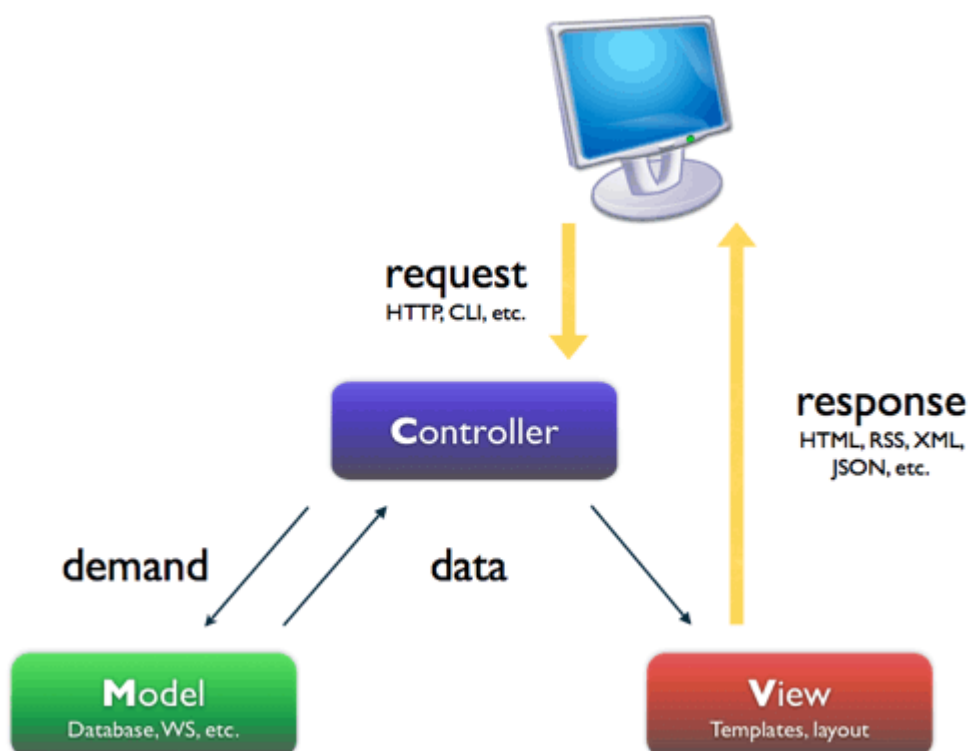
Hiện nay, hầu hết khách hàng đều muốn việc đặt phòng đều có thể được thực hiện bằng hình thức online và nhiều khách sạn khác cũng đã sẵn sàng đáp ứng được các nhu cầu này. Để bắt kịp được xu hướng hiện đại hoá này thì khách sạn của chúng ta nên sử dụng 1 trang web có thể phục vụ được việc đặt phòng trực tuyến cho khách hàng, và 1 hệ thống quản lý khách sạn là giải pháp đang được rất nhiều nhà quản lý khách sạn tin tưởng sử dụng, để người quản lý có thể quản lý được dữ liệu khách hàng, tình trạng hoạt động của khách sạn, ...

2.1.3. Mô tả giải pháp.

Dựa vào thực tiễn mà bài toán đặt ra, em đã quyết định xây dựng và sử dụng website quản lý và đặt phòng khách sạn với một số tính năng đặc biệt cho người dùng:

- Khách hàng có thể dễ dàng tìm kiếm và đặt phòng ở vị trí, khung cảnh của phòng theo nhu cầu và sở thích.
- Khách hàng có thể lựa chọn xem và đặt trước được các dịch vụ của khách sạn thông qua việc gọi điện.
- Khách hàng có thể giao dịch trực tuyến một cách thuận tiện và nhanh chóng.
- Quản lý của khách sạn có thể quản lý hệ thống dữ liệu bao gồm việc cập nhật và quản lý nhân viên, khách hàng, doanh thu một cách chính xác và nhanh chóng.
- Quản lý được tình hình hoạt động của khách sạn
- Công nghệ thực hiện: C#, ASP.Net, API with ASP.NET, Bootstrap, Angular.

Thiết kế kiến trúc ứng dụng (mô hình MVC)



Hình 2. 1: Mô hình MVC trong dự án.

View:

Chức năng và Công Nghệ: Cung cấp giao diện người dùng cho người dùng cuối để tương tác với ứng dụng. Sử dụng HTML, CSS, và JavaScript cho Web Browser, và có thể sử dụng các ngôn ngữ lập trình như Swift (cho iOS) hoặc Kotlin/Java (cho Android) cho ứng dụng di động.

Resquest:

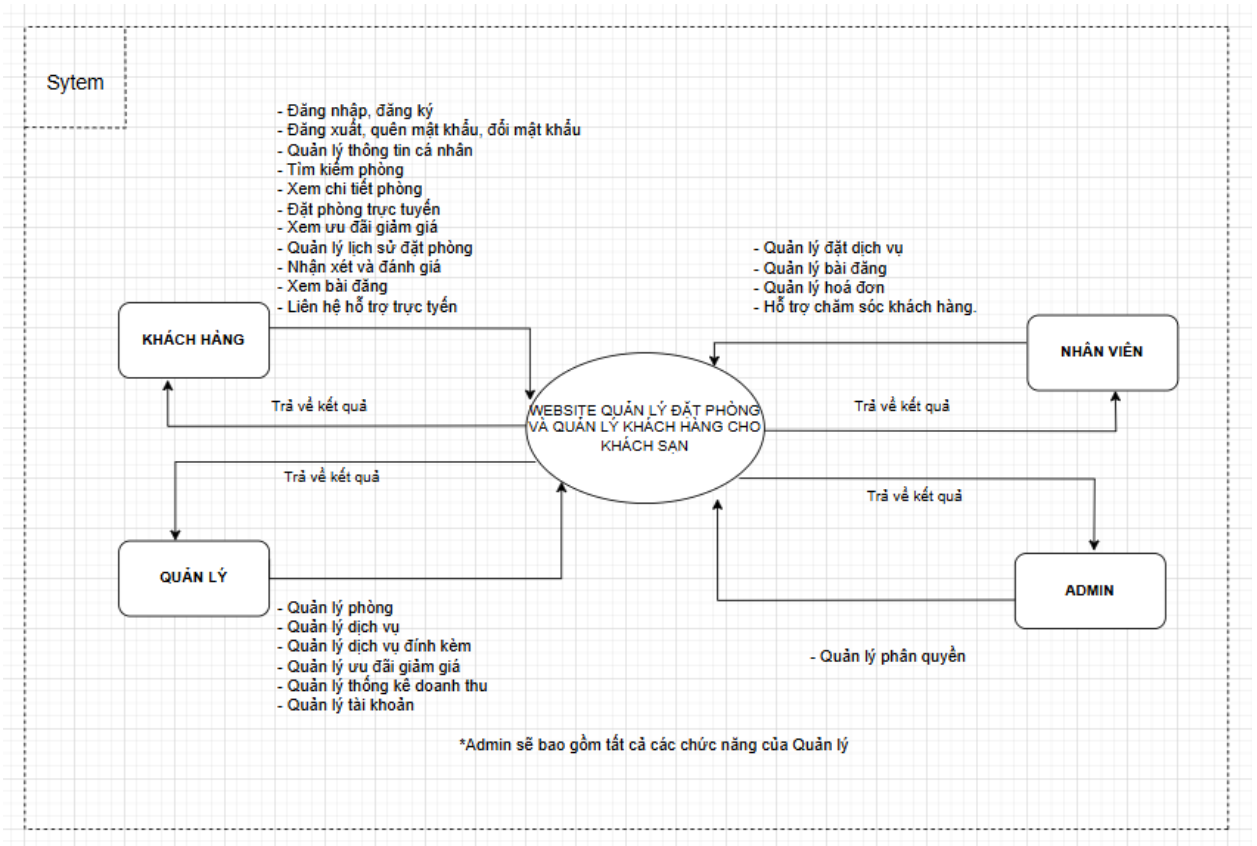
Chức năng và Công Nghệ: Đóng vai trò là cầu nối giữa giao diện người dùng và phần backend. Xử lý yêu cầu từ client và trả về dữ liệu. Sử dụng giao thức HTTP/HTTPS, thường là RESTful API, để truyền thông tin giữa client và server.

Backend:

Chức năng Công Nghệ: Xử lý logic kinh doanh, xử lý dữ liệu, và tương tác với cơ sở dữ liệu. Quản lý tất cả các yêu cầu từ client thông qua Rest API. Sử dụng các ngôn ngữ lập trình như Node.js, Python (Flask/Django), Java (Spring), hoặc Ruby (Ruby on Rails). Các framework này thường hỗ trợ xử lý HTTP requests và responses.

2.2. Thiết kế chương trình.

2.2.1. Tổng quan hệ thống.



Sơ đồ 2. 1: Sơ đồ tổng quan hệ thống.

2.2.2. Mô tả ngữ cảnh hệ thống.

Người dùng có trách nhiệm:

- Đối với Khách hàng: Có thể thực hiện đăng ký tài khoản và đăng nhập vào website khách sạn để tìm kiếm phòng, xem chi tiết và thực hiện đặt phòng trực tuyến, huỷ đặt phòng trực tuyến, quản lý lịch sử đặt phòng, nhận xét, đánh giá, khách hàng còn có thể xem các ưu đãi và xem các bài đăng về khách sạn (có thể bình luận và chia sẻ về bài đăng), ngoài ra còn có thể liên hệ hỗ trợ trực tuyến từ nhân viên của khách sạn thông qua Chat-online.
- Đối với Nhân viên: Có thể quản lý thông tin khách hàng, đặt dịch vụ, đặt phòng cho khách hàng, quản lý phòng đã đặt, quản lý dịch vụ đã đặt. Gọi xác nhận lại phòng đặt khi khách hàng đặt phòng online thành công, thực hiện các thủ tục check-in, check-out cho khách hàng. Ngoài ra, nhân viên còn Xem các hoá đơn đã thanh toán,

quản lý hoá đơn, đăng bài viết giới thiệu và hỗ trợ chăm sóc trực tuyến cho khách hàng, ...

- Đối với Quản lý: Có thể quản lý phòng, quản lý dịch vụ, tài khoản nhân viên và khách hàng, thông tin nhân viên, khách hàng, phiếu ưu đãi giảm giá và những nhận xét đánh giá từ khách hàng.
- Đối với Admin: Có quyền thực hiện được tất cả các chức của vai trò quản lý và có thể quản lý thêm được: phân quyền tài khoản, thống kê báo cáo doanh thu, thống kê phòng và dịch vụ được đặt.

2.2.3. Các chức năng cơ bản của hệ thống.

- **Chức năng chung:**

- Đăng ký, đăng nhập, đổi mật khẩu, quên mật khẩu, đăng xuất.
- Quản lý thông tin cá nhân.
- Tìm kiếm, xem chi tiết.
- Xem bài đăng.

- **Khách hàng:**

- Tìm kiếm phòng.
- Xem chi tiết phòng.
- Đặt phòng trực tuyến.
- Thanh toán trực tuyến.
- Quản lý lịch sử đặt phòng.
- Xem ưu đãi giảm giá.
- Nhận xét và đánh giá.
- Xem bài đăng (Bình luận và chia sẻ).
- Liên hệ hỗ trợ trực tuyến.

- **Nhân viên**

- Đặt dịch vụ trực tiếp.
- Xem chi tiết dịch vụ và phòng.
- Đặt phòng trực tiếp.
- Quản lý đặt dịch vụ.
- Quản lý đặt phòng.
- Quản lý hoá đơn.

- Quản lý bài đăng.
- Hỗ trợ chăm sóc khách hàng.
- **Quản lý**
 - Quản lý phòng.
 - Quản lý dịch vụ.
 - Quản lý dịch vụ đính kèm.
 - Quản lý ưu đãi giảm giá.
 - Quản lý nhận xét và đánh giá.
 - Quản lý thống kê doanh thu.
 - Quản lý tài khoản.
- **Admin**
 - Quản lý nhân viên.
 - Quản lý phân quyền.

2.2.4. Sự khác biệt so với các hệ thống hiện có.

Bảng 2. 1: So sánh sự khác biệt với các hệ thống hiện có

Lĩnh vực so sánh	Hệ thống hiện có	Hệ thống được em (tôi) xây dựng
Công nghệ	<ul style="list-style-type: none"> • Thường sử dụng các công nghệ cũ để dễ dàng dùng chung cho hàng loạt doanh nghiệp mà ko có tính tùy biến theo từng doanh nghiệp. • Các chức năng bị hạn chế và chưa có sự đa dạng cũng như chiều sâu trong hệ thống. 	<ul style="list-style-type: none"> • Sử dụng các công nghệ mới và có thể tùy biến theo nhu cầu của doanh nghiệp. • Đáp ứng đầy đủ các nhu cầu mà doanh nghiệp, khách sạn có thể đưa ra. • Dễ dàng triển khai cập nhật khi có yêu cầu thay đổi về công nghệ. • Tích hợp những công nghệ được phổ biến nhất hiện nay như thanh toán bằng Momo hoặc VNPay.

Bảo mật	<ul style="list-style-type: none"> Hệ thống thường dùng chung những công nghệ bảo mật nên khi một doanh nghiệp bị tấn công vào hệ thống thì các doanh nghiệp khác cũng sẽ có khả năng bị ảnh hưởng theo 	<ul style="list-style-type: none"> Các công nghệ bảo mật hiện đại được áp dụng như mã hoá mật khẩu hoặc những thông tin nhạy cảm của người dùng. Khả năng khắc phục sự cố nhanh chóng với đội ngũ kỹ thuật luôn sẵn sàng hỗ trợ.
Tính thân thiện	<ul style="list-style-type: none"> Khả năng tương tác với người dùng còn hạn chế. Giao diện tuy dễ sử dụng nhưng chưa có sự tối ưu riêng theo nhu cầu của các doanh nghiệp khác nhau 	<ul style="list-style-type: none"> Tương tác tốt với người dùng. Có khả năng tùy chỉnh theo phong cách của doanh nghiệp.
Khả năng đáp ứng nhu cầu	<ul style="list-style-type: none"> Tương đối ổn định tuy nhiên vẫn còn khá sơ sài so với tốc độ phát triển và tốc độ thay đổi nhu cầu của người dùng hiện nay. 	<ul style="list-style-type: none"> Luôn cập nhật theo xu hướng công nghệ mới và dễ dàng đáp ứng được nhu cầu cho người dùng.

2.3. Công nghệ ràng buộc.

2.3.1. Kỹ thuật phát triển hệ thống.

- Nền tảng ứng dụng Web.
- Ngôn ngữ lập trình: C#, ASP.NET, API, BOOSTRAP, ANGULAR
- Cơ sở dữ liệu: SQL SERVER.
- Công nghệ đi kèm: API Momo / VN Pay for Developer, Mailtrap, Firebase.

2.3.2. Môi trường phát triển.

- Máy người dùng phải được kết nối Internet, có trình duyệt Web (IE, Google Chrome, Fire Fox v.v...).
- Phần mềm phát triển dự án: Visual Studio Code, SQL Server, VSCode.

- Công cụ quản lý mã nguồn: Github.

2.3.3. Các ràng buộc khác.

- Nguồn lực: 1 người.
- Kinh phí: Hạn chế.
- Thời gian: Dự kiến hoàn thành trong 2 tháng.
- Công nghệ: ASP.NET, API, ANGULAR.

2.4. Phân tích SWOT (Strengths – Weakness – Opportunities – Threats)

Bảng 2. 2: Phân tích SWOT

<p><u>Điểm mạnh:</u></p> <ul style="list-style-type: none"> ➤ Đã được học tập và trải nghiệm thực tế với dự án sử dụng Asp.net Core Web API và Angular. ➤ Có đủ thời gian để nghiên cứu và phát triển thêm các kỹ năng cần thiết cho dự án. 	<p><u>Điểm yếu:</u></p> <ul style="list-style-type: none"> ➤ Còn chưa có nhiều kinh nghiệm trong việc phát triển một dự án với đầy đủ các quy trình. ➤ Còn nhiều công nghệ mới chưa thể vận dụng được vào dự án.
<p><u>Cơ hội</u></p> <ul style="list-style-type: none"> ➤ Được giáo viên hướng dẫn có kinh nghiệm và nhiệt tình chỉ dẫn trong quá trình hoàn thành đồ án. ➤ Khám phá ra được nhiều công nghệ hay và thú vị trong quá trình xây dựng dự án. 	<p><u>Bất lợi</u></p> <ul style="list-style-type: none"> ➤ Phải cân bằng giữa thời gian xây dựng dự án và thời gian dành cho công việc cá nhân. ➤ Giới hạn về mặt tài nguyên nên chưa thể triển khai được ứng dụng trên đa nền tảng.

[3]

2.5. Đặc tả yêu cầu hệ thống.

2.5.1. Yêu cầu chức năng.

- Quản lý người dùng:
 - Hệ thống phải cho phép người dùng đăng ký tài khoản mới bằng email và mật khẩu.

- Hệ thống phải cho phép người dùng đăng nhập và đăng xuất.
- Hệ thống phải cho phép người dùng cập nhật thông tin cá nhân. [4]
- Quản lý phòng:
 - Hệ thống phải cho phép quản trị viên thêm, sửa, xóa thông tin phòng/khách hàng/nhân viên.
 - Hệ thống phải cho phép người dùng xem danh sách phòng/khách hàng/nhân viên.
 - Hệ thống phải cho phép người dùng tìm kiếm phòng/khách hàng/nhân viên theo tên hoặc danh mục.
- Thanh toán:
 - Hệ thống phải cho phép người dùng đặt phòng/quản lý phòng.
 - Hệ thống phải tính toán chi phí đặt phòng của khách hàng.
 - Hệ thống phải hỗ trợ các phương thức thanh toán như thẻ tín dụng, PayPal, Momo, Vn Pay, ...
- Quản lý đặt phòng:
 - Hệ thống phải cho phép người dùng xem lịch sử đặt phòng.
 - Hệ thống phải gửi email xác nhận đặt phòng sau khi thanh toán thành công.
 - Hệ thống phải cho phép quản trị viên theo dõi và quản lý trạng thái phòng.

2.5.2. Yêu cầu phi chức năng.

- Hiệu suất:
 - Hệ thống phải có khả năng xử lý cùng lúc nhiều yêu cầu khác nhau.
 - Thời gian phản hồi của hệ thống không được quá chậm để người dùng được trải nghiệm hiệu quả nhất. [4]
- Bảo mật:
 - Hệ thống phải sử dụng mã hóa cho tất cả các giao dịch trực tuyến.
 - Hệ thống phải yêu cầu xác thực hai yếu tố cho các hoạt động nhạy cảm như thay đổi mật khẩu hoặc thực hiện thanh toán. [4]
- Khả năng mở rộng:
 - Hệ thống phải có khả năng mở rộng để hỗ trợ nhiều người dùng hơn.
 - Hệ thống phải có khả năng thêm các máy chủ mới mà không cần dừng hoạt động. [4]

- Khả năng sử dụng:
 - Giao diện người dùng phải thân thiện và dễ sử dụng.
 - Hệ thống phải cung cấp tài liệu hướng dẫn chi tiết cho người dùng mới.
- Độ tin cậy:
 - Hệ thống phải có thời gian hoạt động đáp ứng được song song với thời gian của khách sạn.
 - Hệ thống phải có khả năng phục hồi sau sự cố trong thời gian ngắn nhất có thể.
- Bảo trì:
 - Mã nguồn phải được viết rõ ràng và có chú thích đầy đủ để dễ dàng bảo trì.
 - Hệ thống phải cho phép triển khai các bản cập nhật mà không ảnh hưởng đến hoạt động của người dùng.

2.5.3. Hạn chế và giả định.

Trong quá trình phát triển một ứng dụng website quản lý đặt phòng khách sạn, việc xác định rõ ràng các hạn chế và giả định là rất quan trọng. Điều này giúp đội ngũ phát triển và các bên liên quan hiểu rõ các giới hạn và điều kiện tiên quyết của hệ thống, từ đó có thể lập kế hoạch và triển khai dự án một cách hiệu quả.

2.5.3.1. Hạn chế

Ngân sách:

Ngân sách hạn chế có thể ảnh hưởng đến phạm vi và chất lượng của dự án, bao gồm số lượng tính năng và công nghệ được sử dụng cũng như khả năng triển khai dự án trên các nền tảng cloud cần trả phí.

Thời gian:

Thời gian phát triển bị giới hạn có thể yêu cầu đội ngũ phát triển phải ưu tiên các tính năng quan trọng và có thể phải cắt giảm một số tính năng không thiết yếu.

Công nghệ:

Ứng dụng có thể bị giới hạn bởi công nghệ hiện có hoặc các nền tảng mà nó phải hỗ trợ, chẳng hạn như yêu cầu hỗ trợ cả trên desktop và mobile.

Nguồn lực nhân sự:

Số lượng và kỹ năng của cá nhân em có thể là một yếu tố hạn chế, đặc biệt đây là dự án cá nhân nên cần tham khảo thêm các chia sẻ từ các chuyên gia trong các lĩnh vực cụ thể.

Quy định và luật pháp:

Phải tuân thủ các quy định pháp luật liên quan đến bảo vệ dữ liệu cá nhân (ví dụ: GDPR), thuế, và các quy định của ngành du lịch.

Tích hợp với hệ thống bên ngoài:

Ứng dụng có thể cần tích hợp với các hệ thống bên ngoài như hệ thống thanh toán, hệ thống quản lý khách sạn (PMS), và các nền tảng OTA (Online Travel Agency). Việc tích hợp này có thể bị hạn chế bởi API và các dịch vụ của bên thứ ba.

2.5.3.2. Giả định

Giả định là những điều kiện hoặc hoàn cảnh được coi là đúng hoặc có thể xảy ra trong quá trình phát triển dự án. Dưới đây là một số giả định phổ biến trong ứng dụng quản lý đặt phòng khách sạn:

Người dùng có kết nối internet:

Giả định rằng người dùng sẽ có kết nối internet ổn định để truy cập và sử dụng ứng dụng.

Thiết bị người dùng hỗ trợ ứng dụng:

Giả định rằng người dùng sẽ sử dụng các thiết bị (desktop, laptop, smartphone) có khả năng chạy các trình duyệt web hiện đại hỗ trợ các công nghệ như HTML5, CSS3 và JavaScript.

Sẵn sàng dữ liệu của khách sạn:

Giả định rằng thông tin về phòng, giá cả, hình ảnh và các dịch vụ khác của khách sạn sẽ được cung cấp đầy đủ và chính xác từ phía khách sạn.

Thường xuyên bảo trì và cập nhật hệ thống:

Giả định rằng hệ thống sẽ được bảo trì và cập nhật định kỳ để đảm bảo hiệu suất và bảo mật.

Người dùng có kỹ năng cơ bản về công nghệ:

Giả định rằng người dùng có khả năng cơ bản về sử dụng internet và các ứng dụng web.

Quy trình kinh doanh khách sạn:

Giả định rằng quy trình kinh doanh của khách sạn (như quy trình đặt phòng, hủy phòng, thanh toán) sẽ không thay đổi đột ngột và sẽ tuân theo các quy trình chuẩn.

Khách sạn sẽ cung cấp dịch vụ hỗ trợ khách hàng:

Giả định rằng khách sạn sẽ cung cấp dịch vụ hỗ trợ khách hàng để giải quyết các vấn đề liên quan đến đặt phòng.

2.5.4. Tiêu chí chấp nhận.

Tiêu chí chấp nhận là các điều kiện mà một phòng phần mềm phải đáp ứng để được coi là hoàn thành và có thể được chấp nhận bởi người dùng hoặc khách hàng. Dưới đây là một danh sách các tiêu chí chấp nhận cụ thể cho một website quản lý đặt phòng khách sạn:

Đăng ký và Đăng nhập Người dùng

- Đăng ký tài khoản mới:
 - Người dùng phải có khả năng đăng ký tài khoản mới bằng cách cung cấp các thông tin bắt buộc như tên, email, và mật khẩu.
 - Hệ thống phải gửi email xác nhận đăng ký tài khoản mới.
 - Mật khẩu phải đáp ứng các tiêu chuẩn bảo mật (ít nhất 8 ký tự, bao gồm ít nhất một chữ cái viết hoa và một chữ số).
- Đăng nhập:
 - Người dùng đã đăng ký phải có khả năng đăng nhập bằng email và mật khẩu hợp lệ.
 - Hệ thống phải cung cấp tùy chọn "Quên mật khẩu" để người dùng khôi phục mật khẩu qua email.

Quản lý Hồ sơ Người dùng

- Chỉnh sửa thông tin cá nhân:
 - Người dùng phải có khả năng cập nhật thông tin cá nhân của mình như tên, email, số điện thoại và mật khẩu.
 - Hệ thống phải thông báo cho người dùng khi thông tin được cập nhật thành công.

Tìm kiếm và Đặt phòng

- Tìm kiếm phòng:

- Người dùng phải có khả năng tìm kiếm phòng theo các tiêu chí như ngày nhận phòng, ngày trả phòng, số lượng khách, và loại phòng.
- Kết quả tìm kiếm phải hiển thị các thông tin cần thiết về phòng như hình ảnh, giá, mô tả và tình trạng sẵn có.
- Xem chi tiết phòng:
 - Người dùng phải có khả năng xem chi tiết từng phòng, bao gồm hình ảnh, mô tả, tiện nghi, và đánh giá của khách hàng trước.
- Đặt phòng:
 - Người dùng phải có khả năng đặt phòng bằng cách cung cấp thông tin cá nhân và thông tin thanh toán.
 - Hệ thống phải xác nhận đặt phòng thành công và gửi email xác nhận đặt phòng cho người dùng.

Quản lý Đặt phòng

- Xem và quản lý đặt phòng:
 - Người dùng phải có khả năng xem lịch sử đặt phòng của mình.
 - Người dùng phải có khả năng hủy đặt phòng trước một thời hạn nhất định và nhận xác nhận hủy phòng qua email.

Thanh toán

- Thanh toán trực tuyến:
 - Hệ thống phải hỗ trợ các phương thức thanh toán trực tuyến phổ biến như thẻ tín dụng, PayPal.
 - Thanh toán phải được xử lý an toàn và bảo mật.

Đánh giá và Phản hồi

- Đánh giá phòng:
 - Người dùng phải có khả năng để lại đánh giá và bình luận về phòng đã sử dụng.
 - Hệ thống phải hiển thị đánh giá và bình luận của người dùng trên trang chi tiết phòng.

Tính năng bổ sung

- Hỗ trợ khách hàng:
 - Hệ thống phải cung cấp thông tin liên hệ để hỗ trợ khách hàng và có thể bao gồm tính năng chat trực tuyến hoặc biểu mẫu liên hệ.

- Thông báo:
 - Hệ thống phải gửi thông báo qua email hoặc SMS cho các hoạt động quan trọng như xác nhận đặt phòng, cập nhật trạng thái đặt phòng, và thông tin khuyến mãi.

Tiêu chí phi chức năng

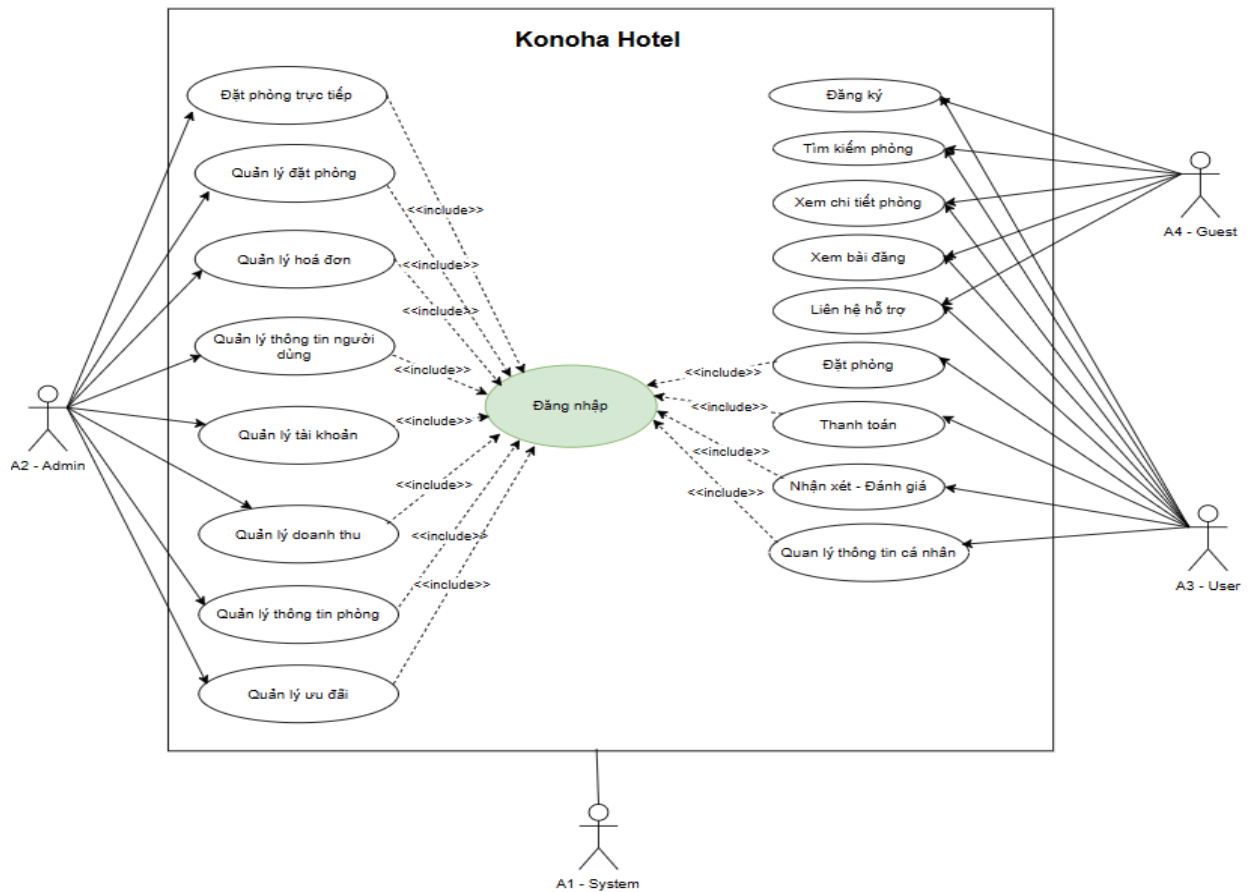
- Hiệu suất:
 - Hệ thống phải có thời gian phản hồi nhanh.
 - Hệ thống phải có khả năng xử lý khi nhiều người dùng cùng thao tác.
- Bảo mật:
 - Tất cả dữ liệu nhạy cảm phải được mã hóa.

Khả năng mở rộng:

- Hệ thống phải có khả năng mở rộng để hỗ trợ số lượng người dùng tăng lên mà không làm gián đoạn dịch vụ.
- Khả năng sử dụng:
 - Giao diện người dùng phải thân thiện, dễ sử dụng và có hướng dẫn rõ ràng.
 - Hệ thống phải hỗ trợ đa ngôn ngữ để phục vụ người dùng từ nhiều quốc gia khác nhau.

2.6. Thiết kế usecase hệ thống.

2.6.1. Biểu đồ usecase tổng quát hệ thống.



Sơ đồ 2. 2: Usecase tổng quát hệ thống.

Bảng 2. 3: Mô tả các tác nhân hệ thống.

Actor ID	Actor	Mô tả
A01	System	Tác nhân hệ thống là cơ sở dữ liệu, giúp truy xuất dữ liệu khi cần hiển thị trên giao diện cho tác nhân khác.
A02	Admin	Tác nhân quản trị viên quản lý toàn bộ hệ thống, có mọi quyền quản lý bao gồm quản lý tài khoản, quản lý người dùng, quản lý vó, quản lý hoá đơn, quản lý blog, quản lý thông kê, quản lý khách hàng.
A03	User	Tác nhân thành viên khi đã có tài khoản, đăng nhập vào hệ thống sẽ được dùng các tính năng gồm xem thông tin cá nhân, đặt phòng, thanh toán.

A04	Guest	Tác nhân người dùng khách được xem giới thiệu về hệ thống, xem các tính năng hệ thống cung cấp như xem danh sách phòng và sử dụng các tính năng khác mà không cần đăng nhập.
------------	-------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Danh sách các usecase

Bảng 2. 4: Danh sách các usecase trong hệ thống.

STT	ID	Use-case	Ý nghĩa	Mô tả chức năng
1	UC01	Quản lý tài khoản	Actor có thể thực hiện được các tính năng khi làm việc với hệ thống bao gồm việc phải có tài khoản mới được truy cập vào hệ thống, và giúp người dùng lấy lại tài khoản khi quên mật khẩu	<ul style="list-style-type: none"> – Đăng ký – Đăng nhập – Đăng xuất – Quên mật khẩu – Xem thông tin tài khoản
2	UC02	Quản lý người dùng	Admin quản lý thành viên như xem thông tin thành viên, thêm người dùng vào hệ thống, chỉnh sửa và nâng quyền thành viên	<ul style="list-style-type: none"> – Xem thông tin người dùng – Xem danh sách người dùng – Thêm người dùng – Chỉnh sửa thông tin – Chỉnh sửa quyền
3	UC03	Quản lý phòng	Actor được xem danh sách phòng, xem chi tiết mô tả, bình luận. Admin có quyền đăng thông tin phòng (dịch vụ) mới, xóa và sửa phòng (dịch vụ) đã có	<ul style="list-style-type: none"> – Xem danh sách các loại phòng (dịch vụ) – Xem chi tiết phòng (dịch vụ) – Thêm phòng (dịch vụ) – Sửa thông tin phòng (dịch vụ) – Xóa phòng (dịch vụ)

4	UC04	Quản lý hoá đơn	<p>Actor sau khi đăng nhập vào hệ thống có thể đặt phòng xem đơn đặt phòng, kiểm tra lịch sử đặt phòng</p> <p>Admin có quyền sửa thông tin nếu người dùng nhập sai thông tin</p>	<ul style="list-style-type: none"> – Tạo đặt phòng – Xem đặt phòng – Chỉnh sửa đặt phòng – Hủy đặt phòng – Xem danh sách đặt phòng – Kiểm tra đặt phòng – Thanh toán đặt phòng – Nhập mã ưu đãi cho đặt phòng
5	UC05	Quản lý phản hồi, liên hệ	Admin quản lý liên hệ như xem danh sách các liên hệ của khách hàng, xem thông tin chi tiết của một liên hệ và chỉnh sửa liên hệ bao gồm: xoá hoặc đánh dấu spam phản hồi, liên hệ	<ul style="list-style-type: none"> – Xem danh sách phản hồi, liên hệ – Xem thông tin chi tiết của một liên hệ – Xoá hoặc đánh dấu spam phản hồi, liên hệ
6	UC06	Quản lý thống kê	Admin thống kê lập báo cáo số lượng sách đã bán, doanh thu và số người dùng	<ul style="list-style-type: none"> – Thống kê người dùng – Thống kê hoá đơn – Thống kê doanh thu
7	UC07	Quản lý blog	Admin quản lý bài viết như xem danh sách bài viết, xem thông tin chi tiết của một bài viết – xoá bài viết	<ul style="list-style-type: none"> – Xem danh sách bài viết – Xem thông tin chi tiết một bài viết. – Xoá bài viết.

Đặc tả usecase:

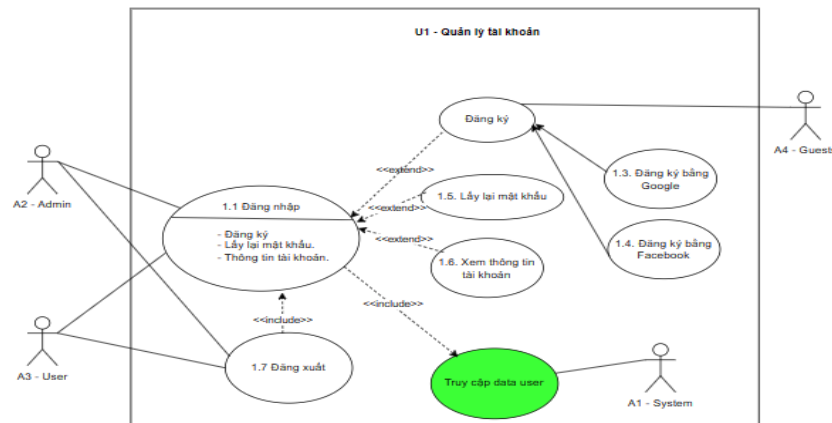
Bảng 2. 5: Đặc tả các thành phần trong usecase

Các thành phần		Mô tả
Use Case Name	Tên của use-case, bắt đầu bằng một động từ	
	Ví dụ: Quản lý tài khoản	

Use Case ID	Định danh của use-case (ID: Identification) Ví dụ: UC01
Description	Mô tả ngắn gọn về use-case. Mô tả này nằm trong khoảng từ một dòng đến một đoạn văn mô tả mục đích và cách sử dụng nó
Stereotype & Package	Mô tả khuôn mẫu và gói mà trường hợp sử dụng này thuộc về. Đây là thông tin tùy chọn và có thể không phải lúc nào cũng được ghi lại, mặc dù nó sẽ được nhập dễ dàng trong một công cụ mô hình hóa
Actors	Danh sách các actor có liên quan đến use-case
Preconditions	Điều kiện tiên quyết là những điều kiện cần được thỏa mãn trước khi thực hiện được mô tả khi use-case có thể bắt đầu
Postconditions	Điều kiện sau là điều kiện phải được đáp ứng khi kết thúc use-case
Use Case Relationships	Relationships của use-case (gồm 3 loại: Include, Extend và Inherited), các actor có quan hệ giữa use-case với use-case
Basic Flow	Luồng thành công chính, trường hợp mọi bước trong use-case đều được thực hiện thành công. Ví dụ như: 1. <mô tả> 2. <mô tả> (A1, E1, E2) 3. <mô tả> (A2, E3) <<include>> <mô tả của use-case included> <<extends>> <mô tả của use-case extended>
Alternative Flow	A1 – Các mô tả tùy chọn ở đây là các luồng thay thế trong các điều kiện được chỉ định trong các bước trong basic flow
Exceptions	E1 – Các mô tả tùy chọn ở đây chỉ định các hành động được thực hiện trong các điều kiện ngoại lệ gặp phải trong basic flow

2.6.2. Biểu đồ usecase cho từng chức năng.

2.6.2.1. Quản lý tài khoản.



Sơ đồ 2. 3: Usecase Quản lý tài khoản.

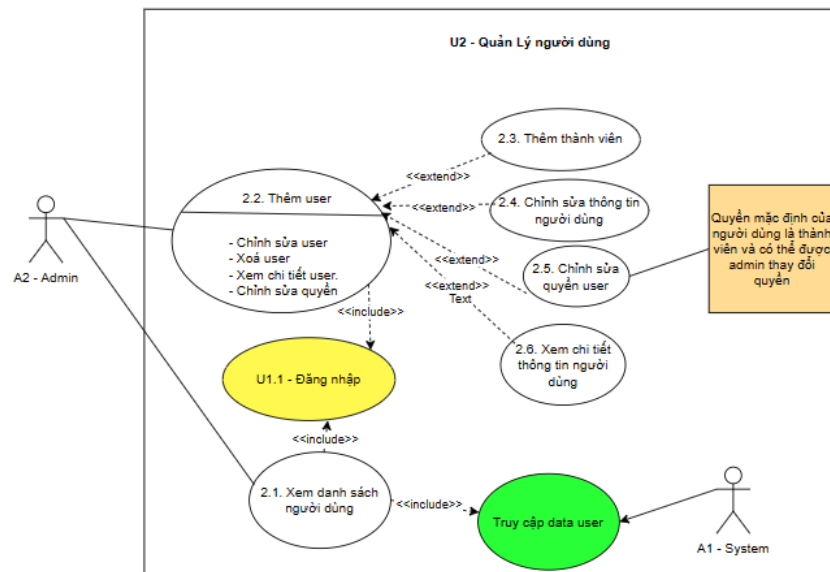
Bảng 2. 6: Đặc tả usecase quản lý tài khoản.

Các thành phần	Mô tả
Use Case Name	Đăng nhập
Use Case ID	U1.1
Description	Các tác nhân muốn đăng nhập vào hệ thống
Actors	A1-System, A2-Admin, A3-User, A4-Guest
Preconditions	Actors đã đăng ký tài khoản hoặc liên kết với tài khoản Google, Facebook. Thiết bị của actor đã được kết nối Internet khi đăng nhập
Postconditions	Actor đăng nhập thành công Hệ thống ghi nhận actor đã đăng nhập Actor có thể sử dụng các tính năng trên hệ thống
Use Case Relationships	Liên kết với A1-System, A2-Admin, A3-User, A4-Guest
Basic Flow	1. Tác nhân truy cập vào trang web 2. Tác nhân đăng nhập tài khoản, chọn lệnh đăng nhập (A1, E1)

	<p>3. Tác nhân đăng nhập thành công (E2)</p> <p>4. Hệ thống ghi nhận hoạt động đăng nhập</p> <p>5. Tác nhân sử dụng các tính năng của hệ thống (A2, A3)</p> <p><<include>> use-case U1.7 – Đăng xuất</p> <p><<extends>> use-case U1.2 – Đăng ký tài khoản</p> <p><<extends>> use-case U1.5 – Lấy lại mật khẩu</p> <p><<extends>> use-case U1.6 – Xem thông tin tài khoản</p>
Alternative Flow	<p>< A1. Tác nhân đăng nhập bằng tài khoản Google, Facebook</p> <p>A1-1. Hệ thống chuyển sang màn hình đăng nhập của Google, Facebook</p> <p>A1-2. Google, Facebook xác thực thông tin đăng nhập thành công và cho phép truy cập hệ thống</p> <p>Use Case tiếp tục bước 3 ></p> <p>< A2. Tác nhân muốn xem thông tin tài khoản, chọn avatar và lệnh xem tài khoản.</p> <p>Use Case tiếp tục Use Case U1.6 ></p> <p>< A2. Tác nhân muốn đăng xuất khỏi hệ thống, chọn avatar và lệnh đăng xuất.</p> <p>Use Case tiếp tục Use Case U1.7 ></p>
Exceptions	<p>< E1. Người dùng chưa có tài khoản, chọn lệnh đăng ký</p> <p>Use Case tiếp tục Use Case U1.2 ></p> <p>< E2. Hệ thống xác thực thông tin đăng nhập không thành công và hiển thị thông báo</p> <p>E2-1. Người dùng thoát khỏi hệ thống</p> <p>Use Case dừng lại</p> <p>E2-2. Người dùng quên mật khẩu, chọn lệnh lấy lại mật khẩu</p> <p>Use Case tiếp tục Use Case U1.5 ></p>
Constraints	Tác nhân đăng nhập không được bỏ trống trường dữ liệu Username và Password
User Interface Specifications	U1-Home, U9.1-Login/Register
Metrics	Đơn giản
Priority	<p>Cao</p> <p>Ban đầu: Đăng ký, đăng nhập</p>

Status Author & History	Giữa: Bổ sung phương thức đăng nhập với Facebook, Google Cuối: Bổ sung đặc tả
	Nguyễn Trọng Hiếu.

2.6.2.2. Quản lý người dùng.



Sơ đồ 2. 4: Usecase quản lý người dùng

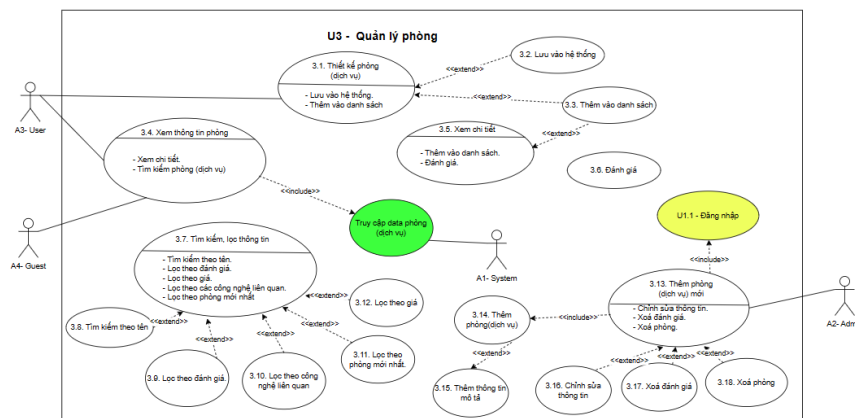
Bảng 2. 7: Đặc tả usecase quản lý người dùng.

Các thành phần	Mô tả
Use Case Name	Thêm người dùng
Use Case ID	UC02.2
Description	Admin thêm người dùng vào hệ thống
Actors	A01-System, A02-Admin
Preconditions	Admin đăng nhập vào hệ thống hoặc đăng nhập vào cơ sở dữ liệu
Postcondition	Admin thêm người dùng thành công Hệ thống ghi nhận thành viên vào cơ sở dữ liệu

<p>S</p> <p>Use Case Relationships</p>	<p>Thành viên được thêm vào có thể đăng nhập.</p> <p>Liên kết với A01-System, A02-Admin, A04-Guest</p>
<p>Basic Flow</p>	<ol style="list-style-type: none"> Admin truy cập vào trang web Konoha Hotel Admin đăng nhập tài khoản Konoha Hotel, chọn lệnh đăng nhập (A1, E1) Admin đăng nhập thành công (E2) Hệ thống ghi nhận hoạt động đăng nhập, admin truy cập trang quản lý Hệ thống mặc định trả về dữ liệu danh sách thành viên Admin chọn thêm thành viên mới, nhập các trường dữ liệu và nhấn chọn Thêm thành viên Hệ thống thêm thông tin thành viên vào cơ sở dữ liệu thành công, thành viên đã có tài khoản (A2, A3, A4, A5, E3) <p><<include>> use-case UC01.1 – Đăng nhập</p> <p><<include>> use-case UC02.3 – Thêm thành viên</p> <p><<extends>> use-case UC02.4 – Chỉnh sửa thông tin người dùng</p> <p><<extends>> use-case UC02.5 – Chỉnh sửa quyền</p> <p><<extends>> use-case UC02.6 – Xem chi tiết thông tin người dùng</p> <p><<extends>> use-case UC02.7 – Xóa người dùng</p>
<p>Alternative Flow</p>	<p>< A1. Admin đăng nhập bằng tài khoản Google, Facebook (trùng tự đặc tả A1 ở UC01.1 – Đăng nhập)</p> <p>Use Case tiếp tục bước 2 ></p> <p>< A2. Admin có thể chỉnh sửa thông tin người dùng</p> <p>Use Case tiếp tục Use Case UC02.4 ></p> <p>< A3. Admin có thể chỉnh sửa quyền người dùng</p> <p>Use Case tiếp tục Use Case UC02.5 ></p>
	<p>< A4. Admin có thể xem thông tin thành viên</p> <p>Use Case tiếp tục Use Case UC02.6 ></p> <p>< A5. Admin có thể xóa người dùng</p>

Exceptions	<p>Use Case tiếp tục Use Case UC02.7 ></p> <p>< E1. Người dùng chưa có tài khoản, chọn lệnh đăng ký</p> <p>Use Case tiếp tục Use Case UC01.2 ></p> <p>< E2. Hệ thống xác thực thông tin đăng nhập không thành công và hiển thị thông báo</p> <p>E2-1. Người dùng thoát khỏi hệ thống</p> <p>Use Case dừng lại</p> <p>E2-2. Người dùng quên mật khẩu, chọn lệnh lấy lại mật khẩu</p> <p>Use Case tiếp tục Use Case UC01.5</p> <p>E2-3. Người dùng không phải admin, không có quyền quản lý hệ thống</p> <p>Use Case dừng lại ></p> <p>< E3. Hệ thống hiện thông báo thêm người dùng thất bại</p> <p>Use Case dừng lại ></p>
Constraints	Bắt buộc phải là admin đăng nhập vào hệ thống mới có thể quản lý thành viên
Metrics	Trung bình
Priority	Trung bình
Status	<p>Ban đầu: Thêm use-case</p> <p>Chính: Bổ sung chỉnh sửa quyền Cuối: Bổ sung ghi chú</p>
Author & History	Nguyễn Trọng Hiếu.

2.6.2.3. Quản lý phòng.



Sơ đồ 2. 5: Usecase quản lý phòng.

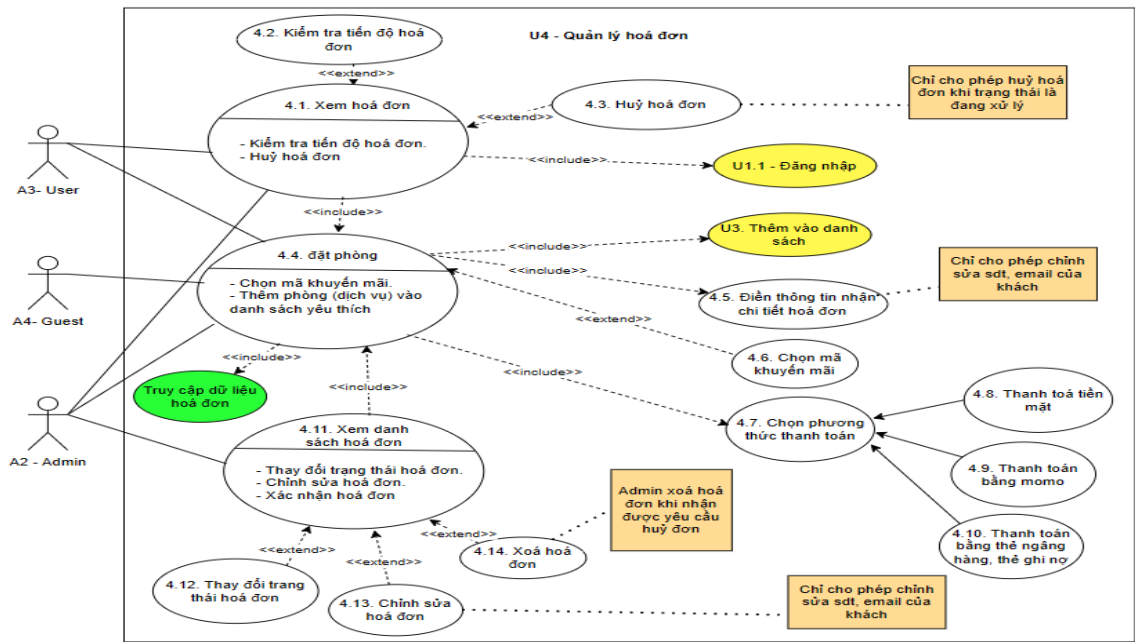
Bảng 2. 8: Đặc tả usecase quản lý phòng.

Các thành phần	Mô tả
Use Case Name	Thêm phòng mới
Use Case ID	UC03.13
Description	Admin đăng phòng mới lên trang web
Actors	A01-System, A02-Admin
Preconditions	Admin đăng nhập thành công vào hệ thống và cơ sở dữ liệu
Postconditions	Admin đăng phòng mới thành công Thông tin phòng mới được được lưu trữ trên cơ sở dữ liệu Các actor khác có thể nhìn thấy phòng mới trên giao diện
Use Case Relationships	Liên kết với A01-System, A02-Admin, A03-Guest, A04-User 1. Admin truy cập vào trang web 2. Admin đăng nhập tài khoản, chọn lệnh đăng nhập (A1, E1) 3. Admin đăng nhập thành công (E2) 4. Hệ thống ghi nhận hoạt động đăng nhập, admin truy cập vào trang quản lý vào mục quản lý phòng 5. Hệ thống mặc định trả về dữ liệu danh sách các phòng hiện có

<p>Basic Flow</p>	<p>trên cơ sở dữ liệu</p> <p>6. Admin chọn thêm phòng mới, nhập các trường dữ liệu và nhấn chọn Gửi</p> <p>7. Hệ thống thêm thông tin phòng mới vào cơ sở dữ liệu.</p> <p>8. Màn hình xuất hiện thông báo thêm phòng thành công.</p> <p><<include>> use-case UC01.1 – Đăng nhập</p> <p><<include>> use-case UC03.14 – Thêm phòng (dịch vụ) mới</p> <p><<include>> use-case UC03.15 – Thêm thông tin mô tả</p> <p><<extends>> use-case UC03.16 – Chỉnh sửa thông tin vớ</p> <p><<extends>> use-case UC03.17 – Xóa đánh giá</p> <p><<extends>> use-case UC03.18 – Xóa phòng</p>
<p>Alternative Flow</p>	<p>< A1. Admin đăng nhập bằng tài khoản Google, Facebook (tương tự đặc tả A1 ở UC01.1 – Đăng nhập)</p> <p>Use Case tiếp tục bước 4 ></p> <p>< A2. Admin muốn tiếp tục thêm phòng mới, nhấn chọn nút Thêm phòng mới</p> <p>Use Case quay lại bước 6 ></p> <p>< A4. Admin có thể chỉnh sửa phòng</p> <p>Use Case tiếp tục use-case UC03.16 ></p> <p>< A4. Admin có thể xóa đánh giá từ người dùng</p> <p>Use Case tiếp tục use-case UC03.17 ></p> <p>< A5. Admin có thể xóa phòng đã tạo</p> <p>Use Case tiếp tục Use use-case UC03.18 ></p> <p>< E1. Người dùng chưa có tài khoản, chọn lệnh đăng ký</p> <p>Use Case tiếp tục Use Case UC01.2 ></p> <p>< E2. Hệ thống xác thực thông tin đăng nhập không thành công và hiển thị thông báo</p> <p>E2-1. Người dùng thoát khỏi hệ thống Use Case dừng lại</p> <p>E2-2. Người dùng quên mật khẩu, chọn lệnh lấy lại mật khẩu</p>

Exceptions	Use Case tiếp tục Use Case UC01.3
	E2-3. Người dùng không phải admin, không có quyền quản lý hệ thống
	Use Case dừng lại >
	< E3. Phòng không được thêm vào cơ sở dữ liệu
Constraints	Use Case dừng lại >
	< E4. Hệ thống hiện thông báo thêm phòng mới thất bại
Metrics	Use Case dừng lại >
	Bắt buộc phải là admin mới có thể thêm phòng mới
Priority	Phức tạp
Status	Cao
Author & History	Ban đầu: thêm use-case
	Chính: bổ sung chỉnh sửa xem chi tiết phòng (dịch vụ) Cuối: bổ sung tìm kiếm, lọc
	Nguyễn Trọng Hiếu.

2.6.2.4. Quản lý hoá đơn.



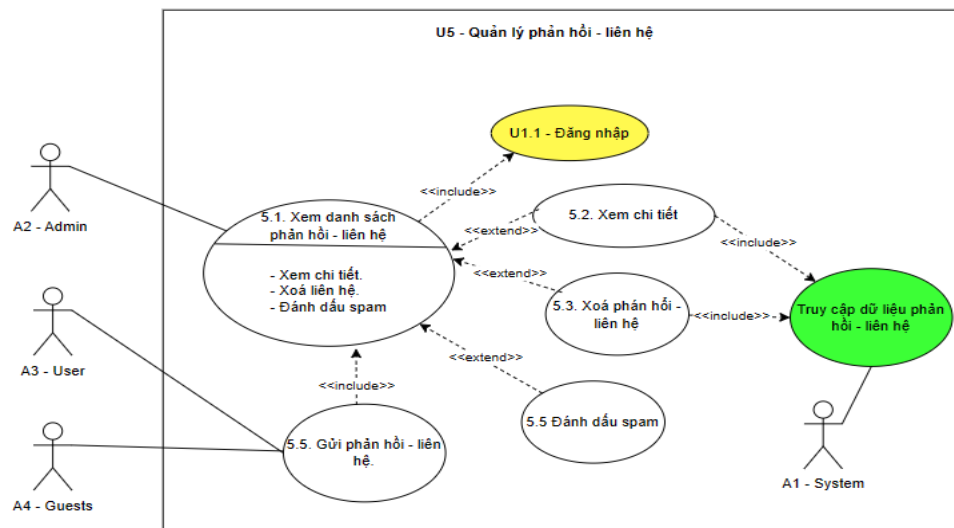
Sơ đồ 2. 6: Usecase quản lý hoá đơn.

Bảng 2. 9: Đặc tả usecase quản lý hoá đơn.

Các thành	
Phần	Mô tả
Use Case Name	Đặt phòng
Use Case ID	UC04.4
Description	Actor tiến hành đặt phòng trên hệ thống
Actors	A02-Admin, A03-User, A04-Guest, A05-WooCommerce
Preconditions	Actor phải thêm phòng vào danh sách.
Postconditions	Actor đặt phòng thành công Hệ thống đơn đặt phòng với trạng thái đang xử lý vào cơ sở dữ liệu Admin có thể tiến hành chỉnh sửa và xử lý đặt phòng
Use Case Relationships	Liên kết với A01-System, A02-Admin, A04-User

Basic Flow	<ol style="list-style-type: none"> 1. Actor truy cập vào trang web Konoha Hotel 2. Actor tiến hành thêm các phòng vào danh sách. 3. Hệ thống cập nhật dữ liệu cho danh sách của actor 4. Actor tiến hành xem danh sách 5. Hệ thống hiển thị giao diện xem danh sách 6. Actor tiến hành chỉnh sửa thông tin các phòng đã được hệ thống thêm vào danh sách. Nhấn nút Đặt phòng 7. Actor tiến hành lựa chọn phương thức thanh toán (A2) 8. Actor kiểm tra lại đặt phòng và nhấn Đặt phòng 9. Đơn đặt phòng được thêm vào cơ sở dữ liệu thành công. Màn hình hiện ra giao diện xác nhận đặt phòng (E1) <p> <<include>> use-case UC03.3 – Thêm vào danh sách <<include>> use-case UC04.5 – Điền thông tin phòng <<include>> use-case UC04.7 – Chọn phương thức thanh toán <<extends>> use-case UC04.6 – Chọn mã khuyến mãi </p>
Alternative Flow	< A1. Actor có thể chọn mã khuyến mãi Use Case tiếp tục Use Case UC04.6 >
Exceptions	< E1. Hệ thống hiện thông báo đặt phòng không thành công Use Case dừng lại >
Metrics	Phức tạp
Priority	Cao
Status	Ban đầu: thêm use-case Chính: bổ sung ghi chú, chỉnh sửa inherit thanh toán Cuối: hoàn thiện sơ đồ
Author & History	Nguyễn Trọng Hiếu

2.6.2.5. Quản lý phản hồi



Sơ đồ 2. 7: Usecase quản lý phản hồi – liên hệ.

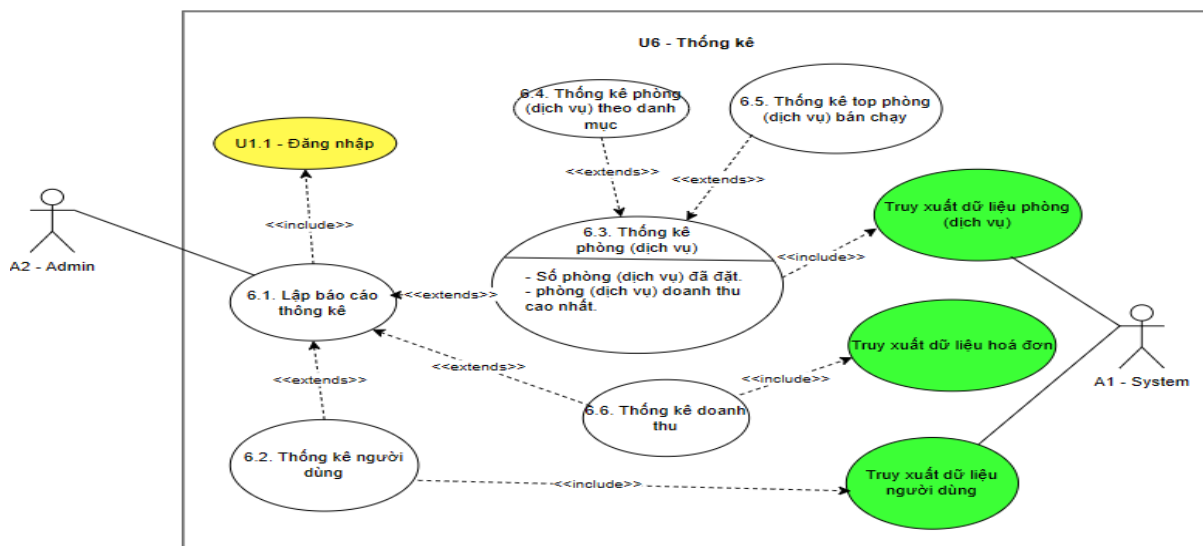
Bảng 2. 10: Đặc tả usecase quản lý phản hồi – liên hệ.

Các thành	
Phần	Mô tả
Use Case Name	Xem danh sách phản hồi, liên hệ
Use Case ID	U5.1
Description	Admin muốn xem danh sách phản hồi, liên hệ
Actors	A1-System, A2-Admin
Preconditions	Admin đã đăng nhập thành công vào hệ thống Admin truy cập thành công vào cơ sở dữ liệu trước đó
Postconditions	Admin lấy thông tin về các liên hệ, phản hồi thành công
Use Case Relationships	Liên kết với A01-System, A02-Admin

Basic Flow	<ol style="list-style-type: none"> 1. Admin truy cập vào trang web Konoha Hotel 2. Admin đăng nhập tài khoản Konoha Hotel, chọn lệnh đăng nhập (A1, E1) 3. Admin đăng nhập thành công (E2) 4. Hệ thống ghi nhận hoạt động đăng nhập, admin truy cập trang quản lý, vào mục quản lý phản hồi, liên hệ. 5. Hệ thống trả về dữ liệu danh sách các phản hồi, liên hệ đã gửi về cho hệ thống (A2, A3, A4, E3). <p><<include>> use-case UC01.1 – Đăng nhập</p> <p><<extends>> use-case UC06.2 – Xem chi tiết phản hồi, liên hệ</p> <p><<extends>> use-case UC06.3 – Xóa phản hồi, liên hệ</p> <p><<extends>> use-case UC06.4 – Đánh dấu spam</p>
Alternative Flow	<p>< A1. Tác nhân đăng nhập bằng tài khoản (tương tự đặc tả A1 ở UC01.1 – Đăng nhập) Use Case tiếp tục bước 3 ></p> <p>< A2. Tác nhân muốn xem chi tiết phản hồi, liên hệ Use Case tiếp tục Use Case UC06.2 ></p> <p>< A3. Tác nhân muốn xóa phản hồi, liên hệ Use Case tiếp tục Use Case UC06.3 ></p> <p>< A4. Tác nhân muốn đánh dấu spam phản hồi, liên hệ Use Case tiếp tục Use Case UC06.4 ></p>
Exceptions	<p>< E1. Người dùng chưa có tài khoản, chọn lệnh đăng ký Use Case tiếp tục Use Case UC01.2 ></p> <p>< E2. Hệ thống xác thực đăng nhập không thành công và hiển thị thông báo (tương tự đặc tả E2 ở U1.1 – Đăng nhập) Use Case dừng lại ></p> <p>< E3. Hệ thống không truy xuất được vào dữ liệu phản hồi, liên hệ Use Case dừng lại ></p>

Constraints	Bắt buộc admin phải đăng nhập vào hệ thống
Metrics	Trung bình
Priority	Trung bình
Status	Ban đầu: chưa có Chính: thêm use-case Cuối: bổ sung hoàn chỉnh
Author & History	Nguyễn Trọng Hiếu.

2.6.2.6. Thống kê.



Sơ đồ 2. 8: Usecase thống kê.

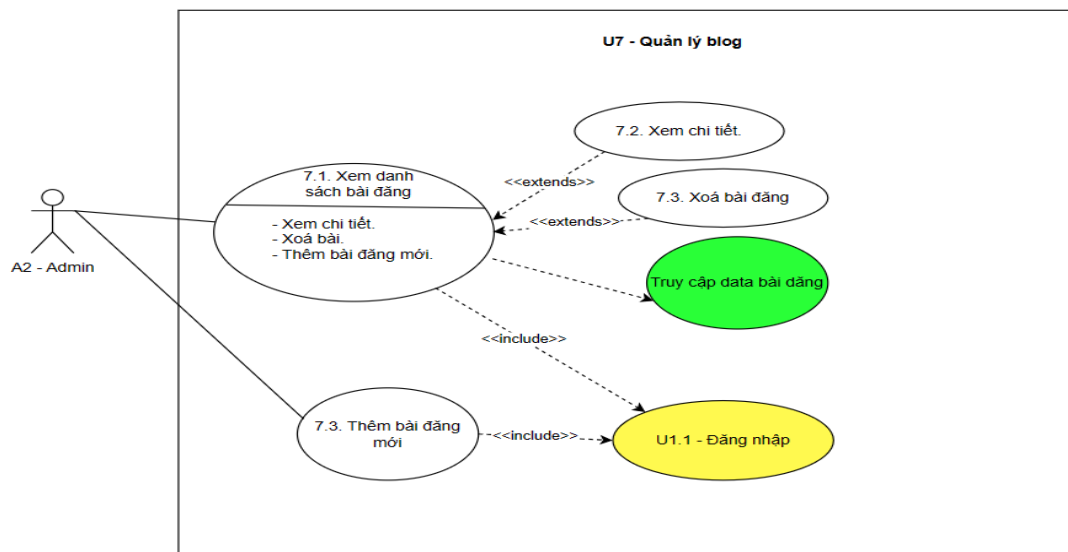
Bảng 2. 11: Đặc tả usecase thống kê.

Các thành	
Phần	Mô tả
Use Case Name	Lập báo cáo thống kê
Use Case ID	U6.1

Description	Admin có thể lập báo cáo thống kê doanh thu và người dùng tham gia theo từng tháng, quý hoặc năm
Actors	A1-System, A2-Admin
Preconditions	Admin đăng nhập vào hệ thống và đã đăng nhập vào cơ sở dữ liệu trước đó
Postconditions	Admin lập báo cáo thành công
Use Case Relationships	Liên kết với A1-System, A2-Admin
Basic Flow	<ol style="list-style-type: none"> 1. Admin truy cập vào trang web Konoha Hotel 2. Admin đăng nhập tài khoản Konoha Hotel, chọn lệnh đăng nhập (A1, E1) 3. Admin đăng nhập thành công (E2) 4. Hệ thống ghi nhận hoạt động đăng nhập, admin truy cập trang quản lý, vào mục thống kê 5. Hệ thống mặc định trả về dữ liệu tổng số người dùng, số lượng phòng đã bán và doanh thu trong 1 tháng 6. Admin chọn trang quản lý thống kê, lựa chọn các loại thống kê (như thống kê phòng, người dùng, doanh thu), lựa chọn khoảng thời gian (tháng, quý hoặc năm) và nhấn nút lập báo cáo 7. Hệ thống trả về dữ liệu báo cáo mà admin cần (E3)
	<<include>> use-case U1.1 – Đăng nhập
	<<extends>> use-case U6.2 – Thống kê người dùng
	<<extends>> use-case U6.3 – Thống kê dịch vụ - phòng
	<<extends>> use-case U6.4 – Thống kê phòng (dịch vụ) đã thu theo danh mục
	<<extends>> use-case U6.5 – Thống kê top phòng (dịch vụ) được doanh thu cao nhất.
	<<extends>> use-case U6.6 – Thống kê doanh thu

Alternative Flow	<p>< A1. Admin đăng nhập bằng tài khoản Google, Facebook (tương tự đặc tả A1 ở U1.1 – Đăng nhập)</p> <p>Use Case tiếp tục bước 3 ></p> <p>< E1. Người dùng chưa có tài khoản, chọn lệnh đăng ký</p> <p>Use Case tiếp tục Use Case U1.2 ></p> <p>< E2. Hệ thống xác thực thông tin đăng nhập không thành công và hiển thị thông báo</p> <p>E2-1. Người dùng thoát khỏi hệ thống</p> <p>Use Case dừng lại</p> <p>E2-2. Người dùng quên mật khẩu, chọn lệnh lấy lại mật khẩu</p> <p>Use Case tiếp tục Use Case U1.3</p> <p>E2-3. Người dùng không phải admin, không có quyền quản lý hệ thống</p>
Exceptions	<p>Use Case dừng lại ></p> <p>< E3. Hệ thống hiện thông báo lập báo cáo thất bại</p> <p>Use Case dừng lại ></p>
Constraints	Bắt buộc phải đăng nhập vào hệ thống
Metrics	Phức tạp
Priority	Trung bình
Status	Ban đầu: thêm use-case

2.6.2.7. Quản lý bài đăng.



Sơ đồ 2. 9: Usecase quản lý blog.

Bảng 2. 12: Đặc tả usecase quản lý blog.

Các thành phần	
Phần	Mô tả
Use Case Name	Xem thông tin chi tiết bài viết
Use Case ID	U7.3
Description	Admin xem thông tin chi bài viết
Actors	A2-Admin
Preconditions	Admin đăng nhập vào hệ thống và đã đăng nhập vào cơ sở dữ liệu trước đó
Postconditions	Admin lấy thông tin khách hàng thành công
Use Case Relationships	Liên kết với A2-Admin
	1. Admin truy cập vào trang web Konoha Hotel 2. Admin đăng nhập tài khoản Konoha Hotel, chọn lệnh đăng nhập (A1, E1) 3. Admin đăng nhập thành công (E2)

Basic Flow	<p>4. Hệ thống ghi nhận hoạt động đăng nhập, admin truy cập trang quản lý, vào mục quản lý bài viết</p> <p>5. Hệ thống mặc định trả về danh sách bài viết trên trang web.</p> <p>6. Admin chọn vào một bài viết và xem thông tin chi tiết bài viết</p> <p><<include>> use-case U1.1 – Đăng nhập</p>
	<p><<extends>> use-case U7.1 – Xem danh sách bài viết <<extends>> use-case U7.2 – Xóa bài viết</p> <p>< A1. Admin đăng nhập bằng tài khoản (tương tự đặc tả A1 ở U1.1 – Đăng nhập)</p> <p>Use Case tiếp tục bước 3 ></p>
Alternative Flow	
Exceptions	<p>< E1. Người dùng chưa có tài khoản, chọn lệnh đăng ký</p> <p>Use Case tiếp tục Use Case U1.2 ></p>
	<p>< E2. Hệ thống xác thực thông tin đăng nhập không thành công và hiển thị thông báo.</p> <p>E2-1. Người dùng thoát khỏi hệ thống</p> <p>Use Case dừng lại</p>
	<p>E2-2. Người dùng quên mật khẩu, chọn lệnh lấy lại mật khẩu</p> <p>Use Case tiếp tục Use Case U1.3</p>
	<p>E2-3. Người dùng không phải admin, không có quyền quản lý hệ thống</p> <p>Use Case dừng lại ></p> <p>< E3. Hệ thống hiện thông báo xem thông tin thất bại</p> <p>Use Case dừng lại ></p>
Constraints	Bắt buộc phải đăng nhập vào hệ thống
Metrics	Phức tạp
Priority	Trung bình
Status	Ban đầu: thêm use-case
	Giữa: thêm quản lý khách hàng
Author & History	Cuối: bổ sung hoàn chỉnh
	Nguyễn Trọng Hiếu

2.6.3. Bảng chức năng và độ ưu tiên.

(Thứ tự độ ưu tiên được đánh giá tăng dần từ 1 đến 3)

Bảng 2. 13: Bảng chi tiết chức năng và độ ưu tiên.

ID	Chức năng	Đối tượng	Mong muốn	Mục đích	Ưu tiên
CN0 1	Đăng ký tài khoản	Khách hàng /Nhân viên /Quản lý	Thêm mới tài khoản vào hệ thống	Để Người dùng có thể đăng kí tài khoản mới	1
CN0 2	Đăng nhập	Khách hàng /Nhân viên /Quản lý /Admin	Truy cập vào hệ thống	Để Người dùng có thể đăng nhập vào hệ thống trang web	1
CN0 3	Tìm kiếm phòng	Khách hàng	Tìm thông tin phòng của khách sạn	Để Khách hàng có thể tìm thông tin về phòng có trong website khách sạn	1
CN0 4	Đặt phòng trực tuyến	Khách hàng	Đặt phòng trực tuyến qua Website hoặc trực tiếp tại quầy khách sạn	Để khách hàng có thể đặt phòng trực tuyến qua Website hoặc trực tiếp tại quầy khách sạn	1
CN0 5	Thanh toán trực tuyến	Khách hàng	Thanh toán thông qua hình thức trực tiếp khi khách hàng đặt phòng ở website	Để Khách hàng tiến hành thanh toán trực tuyến khi đặt phòng trên website của khách sạn khách sạn	1
CN0 6	Liên hệ hỗ trợ trực tuyến	Khách hàng	Liên hệ được đến nhân viên hoặc quản lý để được hỗ trợ	Để Khách hàng có thể được nhân viên hỗ trợ trực tuyến về các vấn đề thắc mắc	1

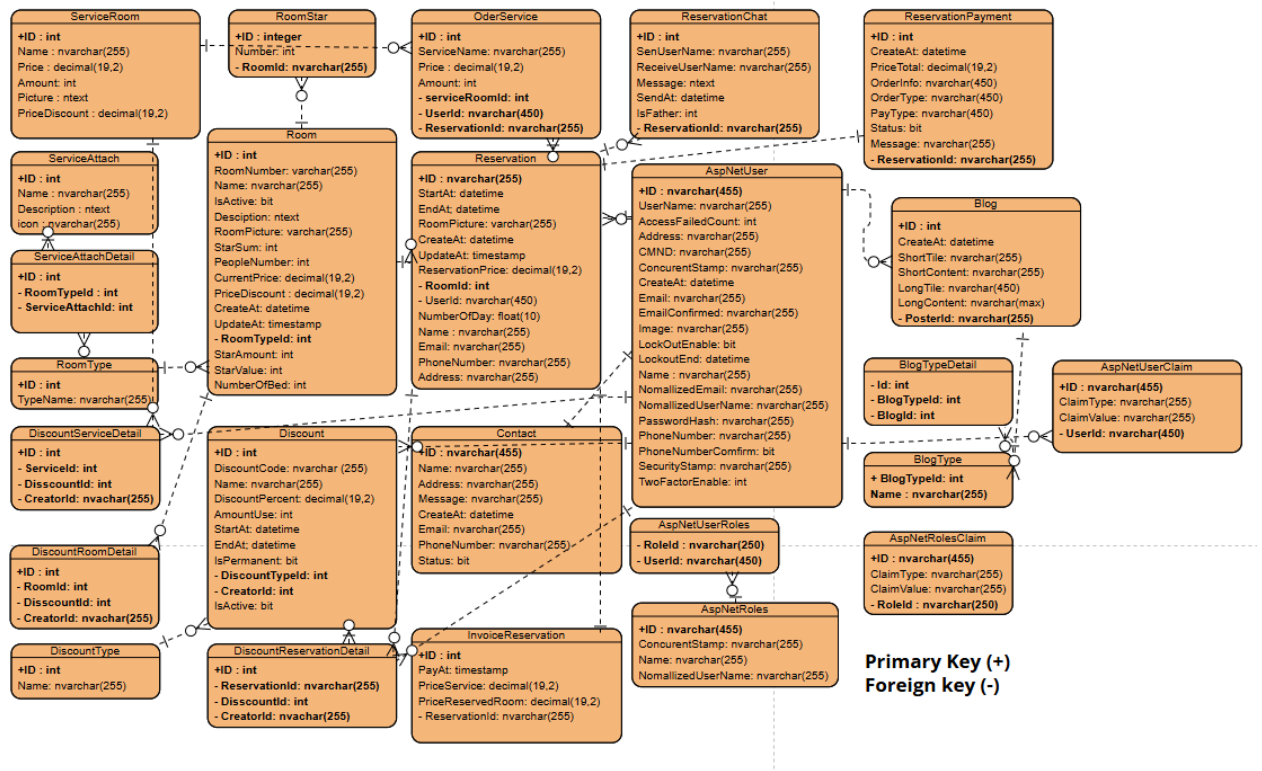
CN0 7	Quản lý đặt phòng	Nhân viên	Quản lý các phòng đã được đặt tại khách sạn	Để Nhân viên có thể xem thông tin về phòng đã được đặt	1
CN0 8	Quản lý dịch vụ đính kèm	Quản lý	Quản lý dịch vụ đính kèm được thêm vào các phòng hiển thị trên trang web của khách sạn	Để Quản lý có thể xem thông tin hoặc thực hiện thêm mới, sửa, xóa các dịch vụ, danh mục dịch vụ đính kèm theo phòng của khách sạn	1
CN0 9	Quản lý phòng	Quản lý	Quản lý phòng trong khách sạn	Để Quản lý có thể xem thông tin hoặc thực hiện thêm mới, sửa, xóa các phòng, danh mục phòng của khách sạn	1
CN1 0	Quản lý dịch vụ	Quản lý	Quản lý dịch vụ của khách sạn	Để Quản lý có thể xem thông tin hoặc thực hiện thêm mới, sửa, xóa các dịch vụ, danh mục dịch vụ của khách sạn	1
CN1 1	Quản lý thống kê doanh thu	Quản lý	Quản lý, xem thống kê, báo cáo doanh thu của phòng và dịch vụ	Quản lý có thể xem các bộ lọc và biểu đồ thống kê, báo cáo số liệu phòng và dịch vụ được đặt và thanh toán từ dữ liệu hệ thống theo ngày, tháng, năm, ...	1

CN1 2	Xem chi tiết phòng	Khách hàng /Nhân viên /Quản lý	Xem chi tiết phòng của khách sạn	Để người dùng có thể xem thông tin chi tiết về phòng mà khách sạn cung cấp	2
CN1 3	Đăng xuất	Khách hàng /Nhân viên /Quản lý /Admin	Đăng xuất khỏi hệ thống	Để Người dùng có thể thoát khỏi hệ thống của mình	2
CN1 4	Quản lý lịch sử đặt phòng	Khách hàng	Xem lịch sử các giao dịch đã thực hiện	Để Người dùng có thể quản lý được các lịch sử về phòng đã đặt trực tuyến trước đó	2
CN1 5	Đặt phòng trực tiếp	Nhân viên	Nhân viên sử dụng máy tính tại lễ tân để đặt phòng trực tiếp cho khách hàng	Để Nhân viên có thể tiến hành đặt phòng cho khách hàng tới trực tiếp tại khách sạn	2
CN1 6	Đặt dịch vụ trực tiếp	Nhân viên	Nhân viên sử dụng máy tính tại lễ tân để đặt dịch vụ trực tiếp cho khách hàng	Để Nhân viên có thể tiến hành đặt dịch vụ cho khách hàng đã đặt phòng tại khách sạn	2
CN1 7	Quản lý hoá đơn	Nhân viên/ Quản lý	Xem hoá đơn đã được thanh toán	Để Nhân viên có thể xem chi tiết về hoá đơn đã thanh toán hoặc xuất hoá đơn lại cho khách hàng	2
CN1 8	Hỗ trợ chăm sóc khách hàng	Nhân viên/ Quản lý	Nhắn tin hỗ trợ giải quyết vấn đề cho khách hàng	Để Nhân viên có thể hỗ trợ trực tuyến về các vấn đề thắc mắc cho khách hàng	2

CN1 9	Quản lý đặt dịch vụ	Nhân viên	Quản lý các dịch vụ đã được đặt tại khách sạn	Để Nhân viên có thể quản lý các dịch vụ đã được đặt theo số phòng trước đó	2
CN2 0	Quản lý ưu đãi giảm giá	Quản lý	Quản lý thông tin các dịch vụ và phòng được giảm giá	Để Quản lý có thể tạo các mã khuyến mãi về dịch vụ và phòng tại khách sạn	2
CN2 1	Quản lý tài khoản	Quản lý/ Admin	Quản lý tất cả các tài khoản được đăng ký vào trang web và hệ thống quản lý của khách sạn	Để có thể quản lý, chỉnh sửa, hoặc xem chi tiết về thông tin của các tài khoản người dùng.	2
CN2 2	Quản lý phân quyền	Admin	Quản lý và phân quyền các vai trò của tài khoản người dùng	Để Admin có thể xem thông tin về quyền truy cập của người dùng và tiến hành phân quyền lại hoặc cấp thêm quyền cho tài khoản người dùng.	2
CN2 3	Thay đổi mật khẩu	Khách hàng /Nhân viên /Quản lý /Admin	Thay đổi mật khẩu của tài khoản	Để Người dùng có thể thay đổi mật khẩu hiện tại của tài khoản	3
CN2 4	Quên mật khẩu	Khách hàng /Nhân viên /Quản lý	Lấy lại mật khẩu	Để Người dùng có thể lấy lại mật	3
CN2 5	Xem bài đăng	Khách hàng	Xem các bài viết, bài blog, các bài giới thiệu về khách sạn	Để Khách hàng có thể tìm hiểu thêm về khách sạn thông qua cá bài đăng	3

CN2 6	Xem ưu đãi giảm giá	Khách hàng	Xem ưu đãi giảm giá về phòng và dịch vụ	Để Khách hàng có thể xem ưu đãi về phòng và dịch vụ của khách sạn	3
CN2 7	Nhận xét và đánh giá	Khách hàng	Viết nhận xét và đánh giá về các dịch vụ tại khách sạn và phòng tại khách sạn	Để Khách hàng thực hiện đánh giá về phòng ốc và các dịch vụ tại khách sạn	3
CN2 8	Quản lý thông tin cá nhân	Khách hàng /Nhân viên /Quản lý /Admin	Quản lý thông tin cá nhân	Để Người dùng có thể xem và cập nhật thay đổi thông tin của mình	3
CN2 9	Quản lý bài đăng	Nhân viên	Nhân viên tiến hành đăng các bài viết giới thiệu về phòng, dịch vụ, tiện nghi, ... có tại khách sạn	Để có thể giới thiệu thêm thông tin khách sạn cho khách hàng	3

2.6.4. Thiết kế sơ đồ thực thể liên kết.



Sơ đồ 2. 10: Sơ đồ thực thể liên kết. [7]

2.7. Thiết kế database.

2.7.1. Lựa chọn cơ sở dữ liệu.

Cơ sở dữ liệu trong ứng dụng SQL SERVER.

Lý do lựa chọn:

- Tính linh hoạt
- Tính thực thi cao
- Có thể sử dụng ngay
- Dễ dàng quản lý
- Tổng chi phí thấp

2.7.2. Lựa chọn cơ sở dữ liệu.

- ServiceRoom (**Id**, Name, Price, Amount, Picture, PriceDiscount): Lưu trữ thông tin của dịch vụ phòng.
- ServiceAttach (**Id**, Name, Description, icon): Lưu trữ Dịch vụ đính kèm
- ServiceAttachDetail (Id, RoomTypeId, ServiceAttachId): Lưu trữ thông tin chi tiết của dịch vụ đính kèm.
- RoomStar (**Id**, Number, RoomId): Lưu trữ số sao đánh giá của phòng.

- Room (**Id**, RoomNumber, Name, IsActive, Description, RoomPicture, StarSum, PeopleNumber, CurrentPrice, DiscountPrice, CreatedAt, UpdateAt, RoomTypeId, StarAmount, StarValue, NumberOfBed): Lưu trữ thông tin của phòng.
- OderService (**Id**, ServiceName, Price, Amount, ServiceRoomId, UserId, ReservationId): Lưu trữ thông tin của đặt dịch vụ
- Reservation (**Id**, StarDate, EndDate, RoomPrice, CreatedAt, UpdateAt, ReservationPrice, Roomid, UserId, NumberOfDay, Name, Email, PhoneNumber, Address): Lưu trữ thông tin đặt phòng
- ReservationChat (**Id**, SendUsername, ReceiveUsername, Message, SendAt, IdFather, ReservationId): Lưu trữ tin nhắn gửi tới người đặt phòng
- Discount (**Id**, DiscountCode, Name, DiscountPercent, AmountUser, StartAt, EndAt, IsPermanent, DiscountTypeid, Creatorid, Active): Lưu trữ mã giảm giá
- DiscountReservationDetail (**Id**, Reservationid, Discountid, Creatorid): Lưu trữ thông tin về giảm giá của đặt phòng.
- DiscountType (**Id**, Name): Lưu trữ các loại mã giảm giá
- ReservationPayment (**Id**, CreateAt, PriceTotal, OderInfo, OderType, PayType, Status, Message, ReservationId): Lưu trữ thông tin hoá đơn đặt phòng
- Blog (**Id**, CreateAt, ShortTitle, ShortContent, LongTitle, LongContent, PosterId): Lưu trữ bài đăng
- BlogTypeDetail (**Id**, BlogTypeId, BlogId): Lưu trữ thông tin loại bài đăng
- BlogType (**Id**, Name): Lưu trữ loại bài đăng
- InvoiceReservation (**Id**, PayAt, PriceService, PriceReserdRoom, ReservationId): Lưu trữ thông tin hoá đơn phòng đã thanh toán
- DiscountServiceDetail (**Id**, ServiceId, DiscountId, CreatorId): Lưu trữ thông tin về giảm giá của dịch vụ.
- DiscountRoomDetail (**Id**, RoomId, DiscountId, Creatorid): Lưu trữ thông tin về giảm giá phòng.
- RoomType (**Id**, TypeName): Lưu trữ thông tin về loại phòng.
- Contact (**Id**, Name, Phone, Email, Message, Address, CreatedAt, Status): Lưu trữ thông tin liên lạc
- ApsNetUser (**Id**, AccessFailedCount, Address, CMND, CouncurrencyStamp, CreatedAt, Email, EmailConfirmed, Image, LockoutEnable, LockoutEnd, Name,

NormalizedEmail, NormalizedUserName, PasswordHash, PhoneNumber, PhoneNumberConfirmed, SecurityStamp, TwoFactorenabled, UserName): Lưu trữ thông tin người dùng ApsNet

- ApsNetUserTokens(LoginProvider, Name, UserId, Value): Lưu mã token của người dùng
- ApsNetUserRoles(RoleId, UserId): Lưu thông tin vai trò người dùng
- ApsNetUserLogins(LoginProvider, ProviderKey, ProviderDisplayName, UserId): Lưu thông tin đăng nhập
- ApsNetUserClaims(Id, ClaimType, ClaimValue, UserId):
- Notification(Id, CreateAt, Description, Link, NotificationType, Status, Title, UserId): Lưu trữ Thông báo.

2.7.3. Định nghĩa bảng (table).

2.7.3.1. Bảng dịch vụ phòng.

Tên bảng: **ServiceRoom**.

Mục đích: Lưu trữ thông tin dịch vụ phòng.

Bảng 2. 14: Table ServiceRoom.

Column	Data type	Size	Description
Id	int		ID dịch vụ phòng
Name	nvarchar	255	Tên dịch vụ
Price	decimal	19,2	Giá của dịch vụ
Amount	int		Số lượng dịch vụ
Picture	ntext		Hình ảnh
PriceDiscount	decimal	19,2	Giá khi được giảm

2.7.3.2. Bảng dịch vụ đính kèm

Tên bảng: **ServiceAttach**

Mục đích: Lưu dịch vụ đính kèm

Bảng 2. 15: Table ServiceAttach

Column	Data type	Size	Description
Id	int		ID dịch vụ đính kèm
Name	nvarchar	255	Tên dịch vụ đính kèm
Description	ntext		Mô tả dịch vụ
Icon	nvarchar	255	Icon

2.7.3.3. Bảng Thông tin dịch vụ đính kèm

Tên bảng: **ServiceAttachDetail**

Mục đích: Lưu trữ thông tin chi tiết của dịch vụ đính kèm.

Bảng 2. 16: Table ServiceAttachDetail

Column	Data type	Size	Description
Id	int		ID dịch vụ

RoomTyprId	int		Khoá ngoại
ServiceAttach	int		Khoá ngoại

2.7.3.4. Bảng Số sao của phòng

Tên bảng: **RoomStar**

Mục đích: Lưu trữ số sao đánh giá của phòng

Bảng 2. 17: Table RoomStar

Column	Data type	Size	Description
Id	int		ID
Number	int		Số sao
RoomId	nvarchar	255	Id phòng

2.7.3.5. Bảng Phòng

Tên bảng: **Room**

Mục đích: Lưu trữ thông tin của phòng

Bảng 2. 18: Table Room

Column	Data type	Size	Description
Id	nvarchar	255	ID phòng
RoomNumber	varchar	255	Số phòng
Name	nvarchar	255	Tên phòng
IsActive	bit		Trạng thái
Description	ntext		Mô tả
RoomPicture	varchar	255	Hình ảnh
StarSum	int		Tổng sao đánh giá
PeopleNumber	int		Số lượng người
CurrentPrice	decimal	12,2	Giá phòng hiện tại
DiscountPrice	decimal	19,2	Giá phòng giảm giá
CreatedAt	datetime		Phòng được khởi tạo lúc
UpdateAt	timestamp		Phòng được thay đổi lúc
RoomTypeId	int		ID Loại phòng
StarAmount	int		Số lượng sao
StarValue	int		
NumberOfBed	int		Số giường

2.7.3.6. Bảng đặt dịch vụ

Tên bảng: **OderService**

Mục đích: Lưu trữ thông tin đặt dịch vụ

Bảng 2. 19: Table OderService

Column	Data type	Size	Description
Id	int		ID đặt dịch vụ
ServiceName	nvarchar	255	Tên dịch vụ
Price	decimal	19,2	Giá tiền
Amount	int		Số lượng dịch vụ
ServiceRoomId	int		Khoá ngoại bảng ServiceRoom
UserId	nvarchar	450	Khoá ngoại của bảng User
ReservationId	nvarchar	255	Khoá ngoại của bảng Reservation

2.7.3.7. Bảng đặt phòng

Tên bảng: **Reservation**

Mục đích: Lưu trữ thông tin đặt phòng

Bảng 2. 20: Table Reservation

Column	Data type	Size	Description
Id	nvarchar	255	ID đặt phòng
StarDate	datetime		Ngày bắt đầu
EndDate	datetime		Ngày kết thúc
RoomPrice	decimal	19,2	Giá phòng
CreatedAt	timestamp		Ngày tạo
UpdateAt	datetime		Ngày thay đổi
ReservationPrice	decimal	19,2	Giá đặt phòng
RoomId	nvarchar	255	Id phòng
UserId	nvarchar	450	ID người dùng
NumberOfDay	float	10	Số ngày ở
Name	nvarchar	255	Họ tên người đặt phòng
Email	nvarchar	255	Địa chỉ email
PhoneNumber	nvarchar	255	Số điện thoại
Address	nvarchar	255	Địa chỉ

2.7.3.8. Bảng tin nhắn đặt phòng.

Tên bảng: **ReservationChat**

Mục đích: Lưu trữ tin nhắn gửi tới người đặt phòng

Bảng 2. 21: Table ReservationChat

Column	Data type	Size	Description
Id	int		ID của tin nhắn
SendUsername	nvarchar	255	Tên người gửi
ReceiveUsername	nvarchar	255	Tên người nhận
Message	ntext		Tin nhắn
SendAt	datetime		Ngày gửi
IdFather	int		Mã của người gửi
ReservationId	nvarchar	255	Khoá ngoại của bảng Reservation

2.7.3.9. Bảng Ưu đãi giảm giá

Tên bảng: **Discount**

Mục đích: Lưu trữ thông tin về các ưu đãi giảm giá

Bảng 2. 22: Table Discount

Column	Data type	Size	Description
Id	int		ID giảm giá
DiscountCode	nvarchar	255	Mã code giảm giá
Name	nvarchar	255	Tên giảm giá
DiscountPercent	decimal	19,2	Phần trăm giảm
AmountUser	int		Số lượng mã giảm giá
StartAt	datetime		Ngày bắt đầu

EndAt	datetime		Ngày kết thúc
IsPermanent	bit		Hạn dùng
DiscountTypeId	int		ID Loại mã giảm giá
CreatorId	nvarchar	450	ID người tạo
Active	bit		Trạng thái

2.7.3.10. Bảng thông tin giảm giá đặt phòng:

Tên bảng: **DiscountReservationDetail**

Mục đích: Lưu trữ thông tin về giảm giá của đặt phòng

Bảng 2. 23: Table DiscountReservationDetail

Column	Data type	Size	Description
Id	int		ID giảm giá
Reservationid	nvarchar	255	Khoá ngoại của bảng Reservation
Discountid	int		Khoá ngoại của bảng Discount
Creatorid	nvarchar	450	Id người tạo

2.7.3.11. Bảng người dùng APSNET

Tên bảng: **ApsNetUser**

Mục đích: Lưu trữ thông tin về người dùng.

Bảng 2. 24: Table ApsNetUser

Column	Data type	Size	Description
Id	nvarchar	450	ID
AccessFailedCount	int	256	Truy cập không thành công
Address	nvarchar	500	Địa chỉ
CMND	nvarchar	500	CMND
CouncurrencyStamp	nvarchar	500	Tem tiền tệ
CreatedAt	datetime		Người tạo
Email	nvarchar	256	Địa chỉ email
EmailConfirmed	bit		Xác nhận Email
Image	nvarchar	450	Ảnh
LockoutEnable	bit		Khóa kích hoạt
LockoutEnd	datetimeoffset		Khóa kết thúc
Name	nvarchar	500	Tên
NormalizedEmail	nvarchar	256	Email chuẩn hóa
NormalizedUserName	nvarchar	256	Tên chuẩn hóa
PasswordHash	nvarchar	500	Mật khẩu
PhoneNumber	nvarchar	500	SĐT
PhoneNumberConfirmed	bit		SĐT xác nhận
SecurityStamp	nvarchar	500	Tem bảo mật
TwoFactorenabled	bit		Bảo mật hai lớp
UserName	nvarchar	256	Tên người dùng

2.7.3.12. Bảng loại mã giảm giá

Tên bảng: **DiscountType**

Mục đích: Lưu trữ thông tin về các loại mã giảm giá

Bảng 2. 25: Table DiscountType

Column	Data type	Size	Description
Id	int		ID loại mã giảm giá
Name	nvarchar	255	Tên của loại mã giảm giá

2.7.3.13. Bảng hoá đơn đặt phòng

Tên bảng: **ReservationPayment**

Mục đích: Lưu trữ thông tin hoá đơn đặt phòng

Bảng 2. 26: Table ReservationPayment

Column	Data type	Size	Description
Id	int		ID hoá đơn
CreateAt	datetime		Thời gian tạo
PriceTotal	decimal	19,0	Tổng tiền
OderInfo	nvarchar	500	Thông tin người đặt
OderType	nvarchar	255	Loại đặt
PayType	varchar	255	Loại hình thức thanh toán
Status	int		Trạng thái
Message	nvarchar	255	Tin nhắn
ReservationId	nvarchar	255	Khoá ngoại của bảng Reservation

2.7.3.14. Bảng bài đăng

Tên bảng: **Blog**

Mục đích: Lưu trữ bài đăng

Bảng 2. 27: Table Blog

Column	Data type	Size	Description
Id	int		ID bài đăng
CreateAt	datetime		Thời gian tạo
ShortTitle	nvarchar	255	Tiêu đề ngắn
ShortContent	nvarchar	255	Nội dung tham khảo
LongTitle	nvarchar	450	Tiêu đề chính
LongContent	nvarchar	450	Nội dung chính
PosterId	nvarchar	450	ID bài đăng

2.7.3.15. Bảng chi tiết loại bài đăng

Tên bảng: **BlogTypeDetail**

Mục đích: Lưu trữ thông tin chi tiết loại bài đăng.

Bảng 2. 28: Table BlogTypeDetail

Column	Data type	Size	Description
Id	int	20	ID loại bài đăng
BlogTypeId	int	255	ID loại bài đăng
BlogId	int		ID bài đăng

2.7.3.16. Bảng loại bài đăng

Tên bảng: **BlogType**

Mục đích: Lưu trữ loại bài đăng

Bảng 2. 29: Table BlogType

Column	Data type	Size	Description
Id	int	20	ID loại bài đăng
Name	int	20	Mã vai trò

2.7.3.17. Bảng hóa đơn đặt trước

Tên bảng: **InvoiceReservation**

Mục đích: Lưu trữ thông tin loại hoá đơn đặt trước.

Bảng 2. 30: Table InvoiceReservation

Column	Data type	Size	Description
Id	int		ID hóa đơn
PayAt	timestamp		Thời gian thanh toán
PriceService	decimal	19,2	Giá dịch vụ
PriceReserdRoom	decimal	19,2	Giá đặt phòng
ReservationId	nvarchar	255	Khoá ngoại của bảng Reservation

2.7.3.18. Bảng chi tiết giảm giá dịch vụ

Tên bảng: **DiscountServiceDetail**

Mục đích: Lưu trữ mã giảm giá chi tiết của dịch vụ.

Bảng 2. 31: Table DiscountServiceDetail

Column	Data type	Size	Description
Id	int		ID giảm giá
ServiceId	int	450	ID dịch vụ
ReservationId	int		Khoá ngoại của bảng Reservation
DiscountId	int	255	ID giảm giá
CreatorId	nvarchar		ID người tạo

2.7.3.19. Bảng chi tiết giảm giá phòng

Tên bảng: **DiscountRoomDetail**

Mục đích: Lưu trữ mã giảm giá chi tiết của phòng.

Bảng 2. 32: Table DiscountRoomDetail

Column	Data type	Size	Description
Id	int		ID giảm giá
RoomId	int	450	ID
DiscountId	int	255	ID giảm giá
CreatorId	nvarchar		ID người tạo

2.7.3.20. Bảng loại phòng

Tên bảng: **RoomType**

Mục đích: Lưu trữ thông tin loại phòng.

Bảng 2. 33: Table RoomType

Column	Data type	Size	Description
Id	int		Mã giảm giá
RoomId	int	450	ID
DiscountId	int	255	ID giảm giá

CreatorId	nvarchar		ID người tạo
-----------	----------	--	--------------

2.7.3.21. Bảng liên hệ

Tên bảng: **Contact**

Mục đích: Lưu trữ thông tin liên lạc.

Bảng 2. 34: Table Contact

Column	Data type	Size	Description
Id	int		ID liên hệ
Name	nvarchar	450	Họ tên
Phone	varchar	255	SDT
Email	nvarchar		Địa chỉ email
Message	nvarchar		Tin nhắn
Address	nvarchar		Địa chỉ
CreatedAt	datetime		Thời gian tạo
Status	Bit		Trạng thái

2.7.3.22. Bảng mã thông báo người dùng APSNET

Tên bảng: **ApsNetUserTokens**

Mục đích: Lưu trữ mã thông báo người dùng.

Bảng 2. 35: Table ApsNetUserTokens

Column	Data type	Size	Description
LoginProvider	nvarchar	450	Nhà cung cấp đăng nhập
Name	nvarchar	450	Tên người dùng
UserId	nvarchar	500	ID người dùng
Value	nvarchar	450	Giá trị

2.7.3.23. Bảng vai trò người dùng APSNET

Tên bảng: **ApsNetUserRoles**

Mục đích: Lưu trữ vai trò người dùng.

Bảng 2. 36: Table ApsNetUserRoles

Column	Data type	Size	Description
RoleId	nvarchar	450	ID vai trò
UserId	nvarchar	450	ID người dùng

2.7.3.24. Bảng đăng nhập người dùng APSNET

Tên bảng: **ApsNetUserLogins**

Mục đích: Lưu trữ thông tin đăng nhập người dùng.

Bảng 2.37: Table ApsNetUserLogins

Column	Data type	Size	Description
LoginProvider	nvarchar	450	Nhà cung cấp đăng nhập
ProviderKey	nvarchar	450	Khóa nhà cung cấp
ProviderDisplayName	nvarchar	500	Tên hiển thị của nhà cung cấp
UserId	nvarchar	450	ID người dùng

2.7.3.25. Bảng yêu cầu người dùng APSNET

Tên bảng: **ApsNetUserClaims**

Mục đích: Lưu trữ mã thông báo người dùng.

Bảng 2.38: Table ApsNetUserClaims

Column	Data type	Size	Description
Id	nvarchar	450	Nhà cung cấp đăng nhập
ClaimType	nvarchar	450	Loại yêu cầu
ClaimValue	nvarchar	500	Loại giá trị
UserId	nvarchar	450	ID người dùng

2.7.3.26. Bảng thông báo

Tên bảng: **Notification**

Mục đích: Lưu trữ mã thông báo người dùng.

Bảng 2.39: Table Notification

Column	Data type	Size	Description
Id	int		Nhà cung cấp đăng nhập
CreateAt	datetime		Người tạo
Description	nvarchar	500	Mô tả
Link	nvarchar	500	Liên kết
NotificationType	int		Loại thông báo
Status	bit		Trạng thái
Title	nvarchar	500	Tiêu đề
UserId	nvarchar	450	ID người dùng

2.8. Thiết kế giao diện.

2.8.1. Giao diện đăng nhập.

The image shows a web browser window with a single tab labeled "Page 1". The address bar displays the URL "https://www.KonohaHotel.io". The main content area of the browser contains a login form layout. On the left, there is a placeholder for a "Logo" and a large rectangular area labeled "IMG". To the right of these placeholders, the text "Welcome!" is displayed. Below this, there are two input fields: one for "Email" with the placeholder text "Type Email", and another for "Password" with the placeholder text "Type your Password". A blue button labeled "Sign In" is positioned below the password field. At the bottom of the login section, there is a link that says "Don't have an account? SignUp Now" followed by a link for "Forgot Password".

Hình 2. 2: Layout giao diện đăng nhập

2.8.2. Giao diện đăng ký.

The wireframe shows a browser window with the address bar containing 'https://www.KonohaHotel.io'. The page layout includes a 'Logo' placeholder on the left. On the right, there is a 'Sign Up' section with the following fields: 'User Name' (with placeholder 'Type user Name'), 'Email' (with placeholder 'Email'), 'Full Name' (with placeholder 'Type Full Name'), 'Phone Number' (with placeholder 'Enter 10 digit phone num'), 'Create Password' (with placeholder 'Type password'), and 'Confirm Password' (with placeholder 'Confirm Password'). Below these fields is a checkbox with the text 'By creating an account, you agree to the terms of service and Conditions , and privacy policy'. A blue button labeled 'Sign Up Now' is positioned below the checkbox. At the bottom of the sign-up section, there is a link that says 'Already have an account? Login'.

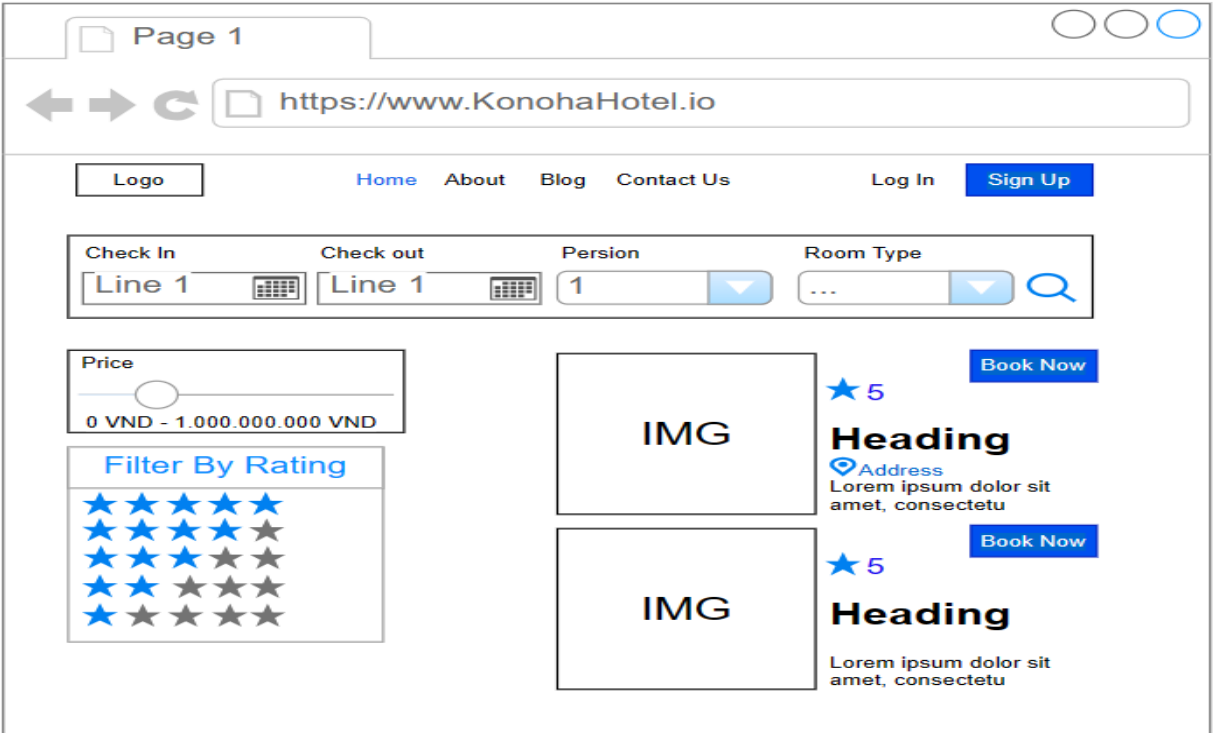
Hình 2. 3: Layout giao diện đăng ký.

2.8.3. Giao diện trang chủ user.

The wireframe shows a browser window with the address bar containing 'https://www.KonohaHotel.io'. The page layout includes a 'Logo' placeholder on the left. To the right of the logo is a navigation menu with links: 'Home', 'About', 'Blog', 'Contact Us', 'Log In', and a blue 'Sign Up' button. Below the navigation menu, there is a 'Heading' section with a paragraph of placeholder text: 'Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.' To the right of the heading is a large 'IMG' placeholder. At the bottom of the page, there is a search bar with the following fields: 'Check In' (with placeholder 'Line 1' and a calendar icon), 'Check out' (with placeholder 'Line 1' and a calendar icon), 'Persion' (with placeholder '1' and a dropdown arrow), 'Room Type' (with placeholder '...' and a dropdown arrow), and a search icon (magnifying glass).

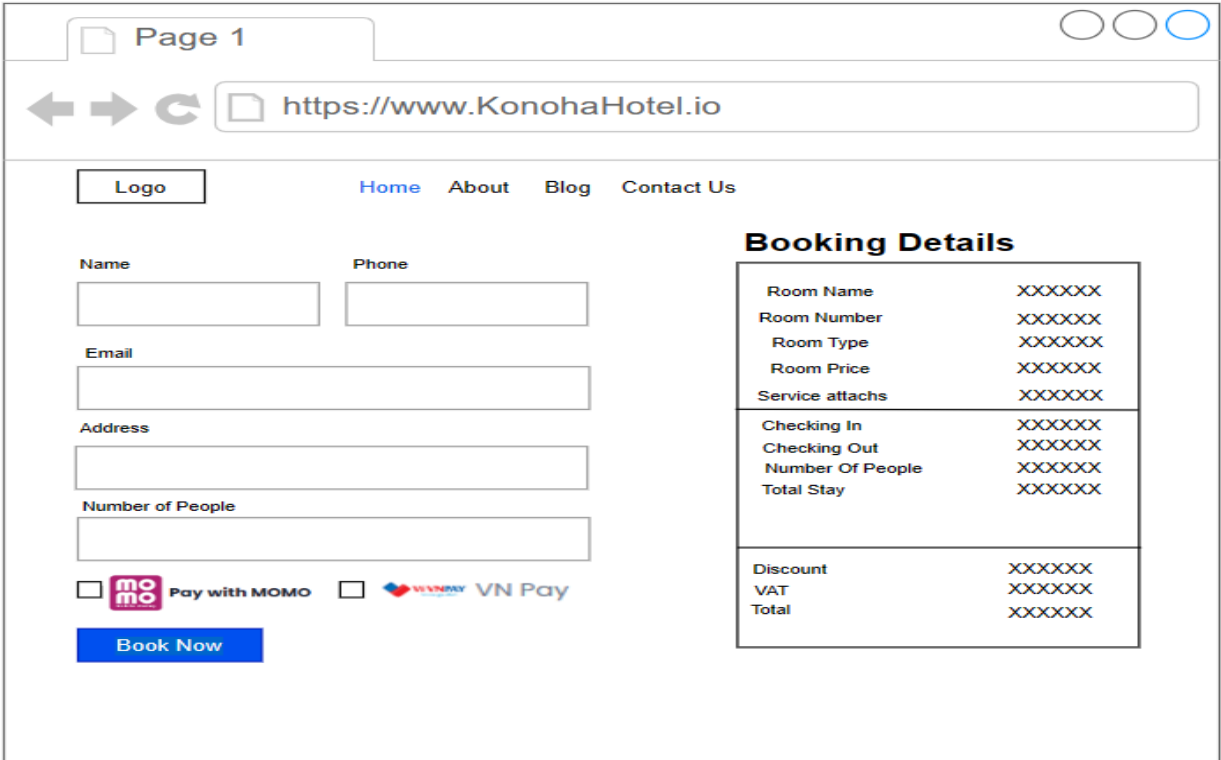
Hình 2. 4: Layout giao diện trang chủ user.

2.8.4. Giao diện danh sách phòng user.



Hình 2. 5: Layout giao diện danh sách phòng user

2.8.5. Giao diện đặt phòng user.



Hình 2. 6: Layout giao diện đặt phòng user.

2.8.6. Giao diện thanh toán user.

Page 1

https://www.KonohaHotel.io

Booking Details

Room Name	XXXXXX
Room Number	XXXXXX
Room Type	XXXXXX
Room Price	XXXXXX
Service attaches	XXXXXX
Checking In	XXXXXX
Checking Out	XXXXXX
Number Of People	XXXXXX
Total Stay	XXXXXX
Discount	XXXXXX
VAT	XXXXXX
Total	XXXXXX

Quét mã QR để thanh toán

Hướng dẫn

Hình 2. 7: Layout giao diện thanh toán user.

2.8.7. Giao diện quản lý tài khoản user.

Page 1

https://www.KonohaHotel.io

Logo Home About Blog Contact Us user name

Payment

Profile

Log Out

user name

Address abc@gmail.com 123444222

Edit

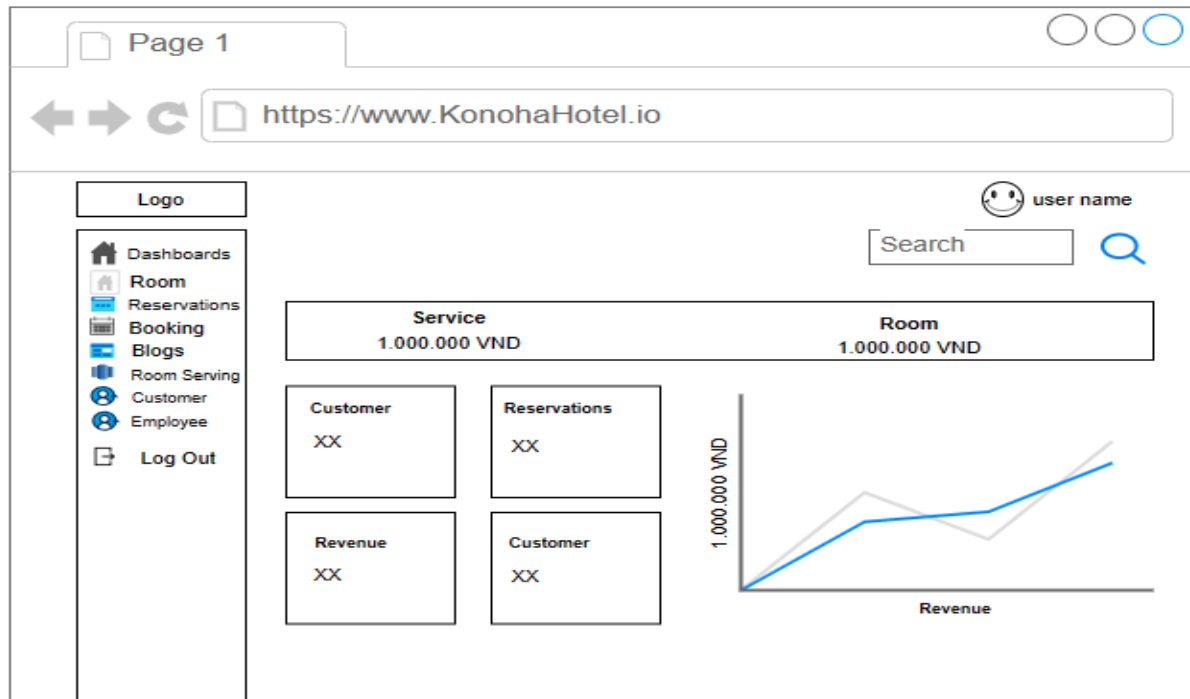
Password

Last change 2 days ago

Change Password

Hình 2. 8: Layout giao diện quản lý tài khoản user.

2.8.8. Giao diện trang chủ admin.



Hình 2. 9: Layout giao diện trang chủ admin.

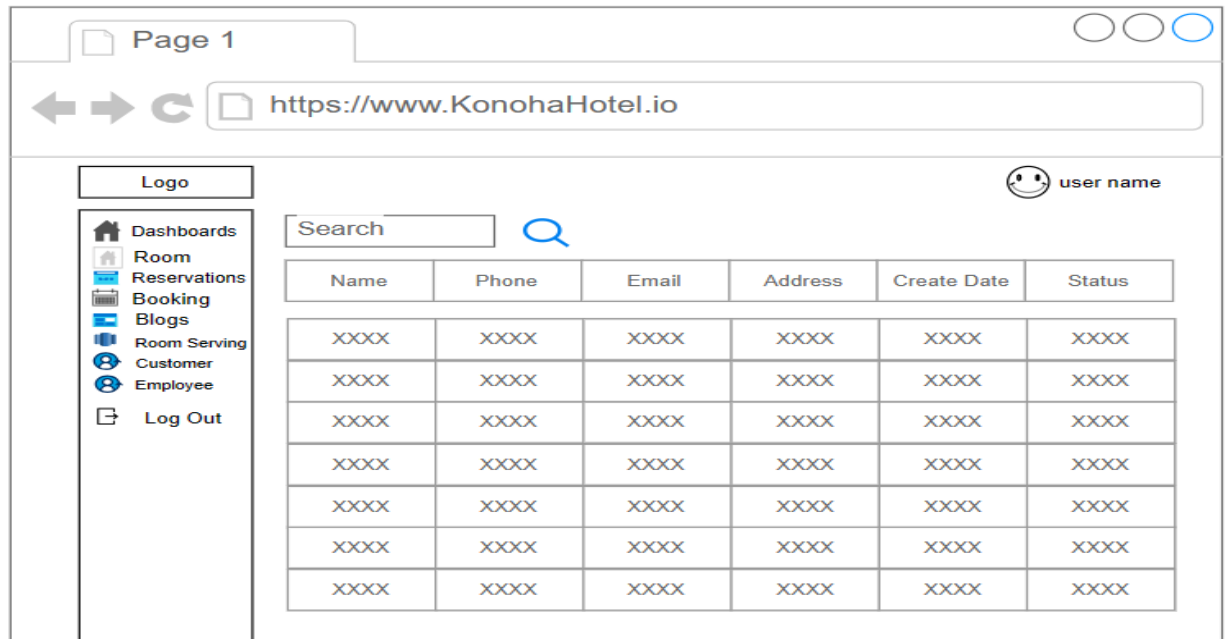
2.8.9. Giao diện quản lý phòng admin.

The screenshot shows the Admin Room Management interface for KonohaHotel.io. The interface includes a sidebar with navigation links: Dashboards, Room, Reservations, Booking, Blogs, Room Serving, Customer, Employee, and Log Out. The main content area features a table with columns: Name, Room Number, Type, Price, Discount, Simple Bed, Double bed, People number, Service, Status, and Action. The table contains 8 rows of placeholder data (XXXX). An 'Add Room' button is located in the top right corner.

Name	Room Number	Type	Price	Discount	Simple Bed	Double bed	People number	Service	Status	Action
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX

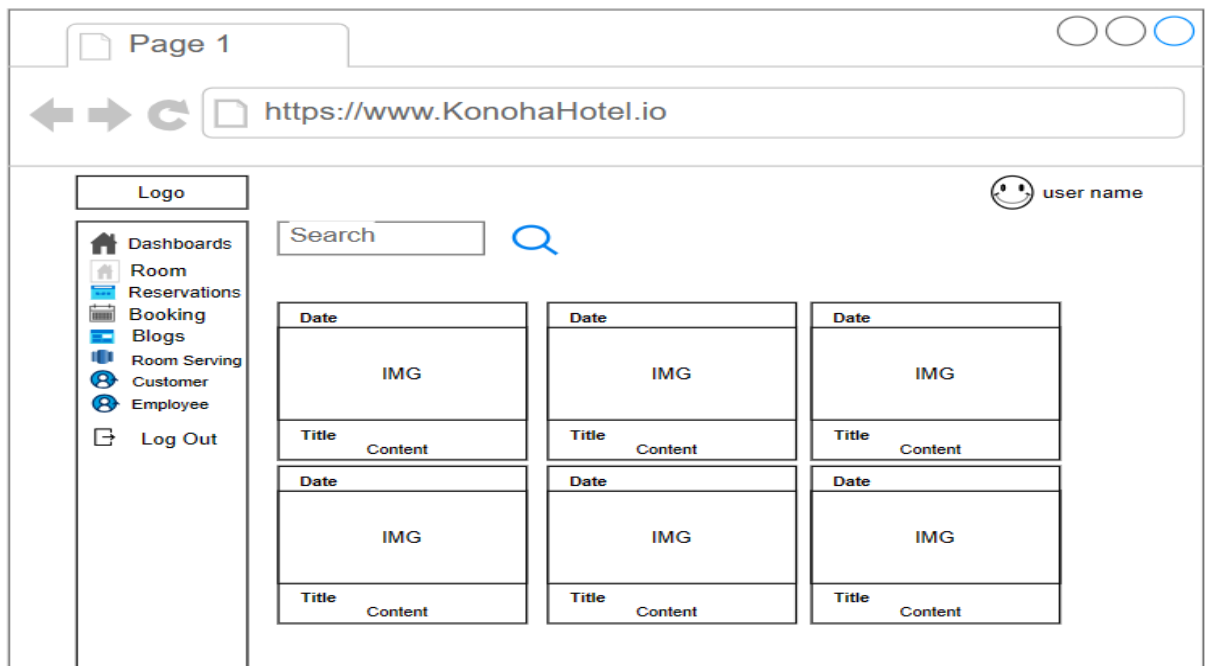
Hình 2. 10: Layout giao diện quản lý phòng admin.

2.8.10. Giao diện quản lý user cho admin.



Hình 2. 11: Layout giao diện quản lý user cho admin.

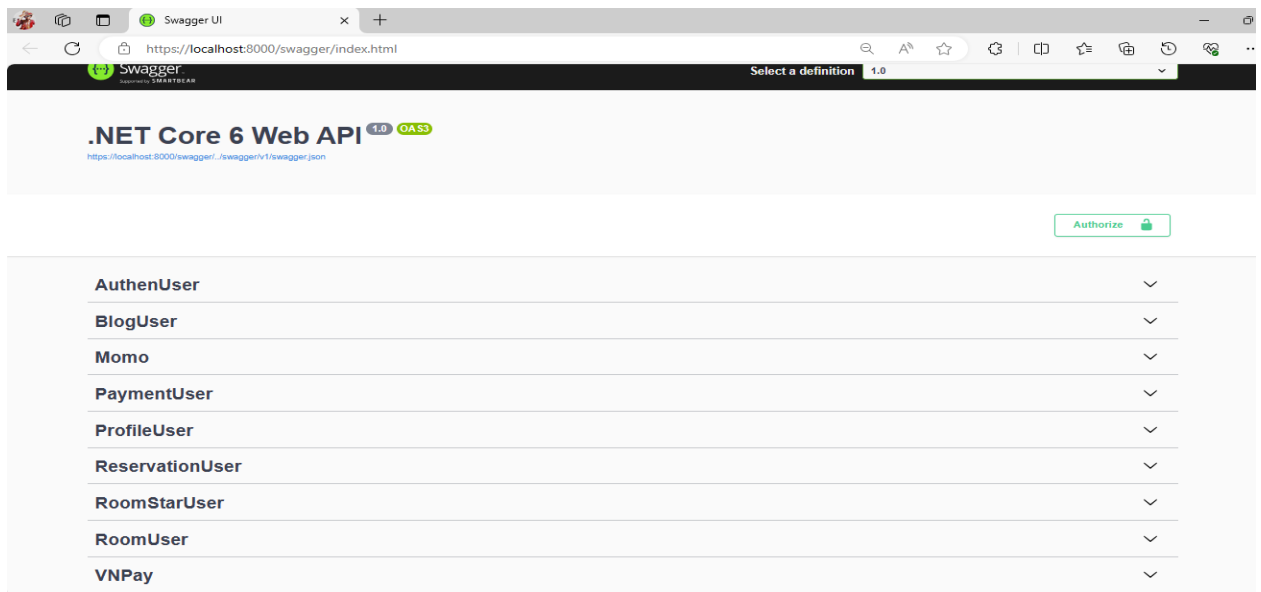
2.8.11. Giao diện quản lý Blog cho admin.



Hình 2. 12: Layout giao diện quản lý bài viết cho admin.

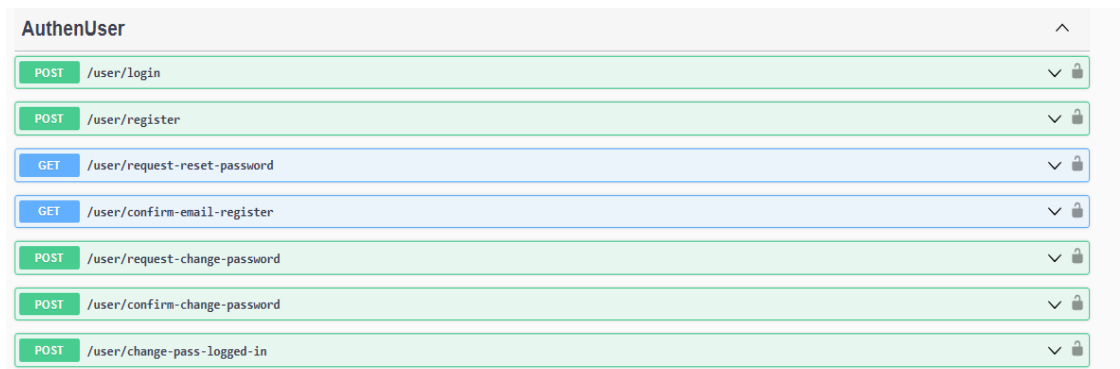
CHƯƠNG III: CÀI ĐẶT HỆ THỐNG VÀ KIỂM THỬ

3.1. Các Api chức năng của hệ thống.



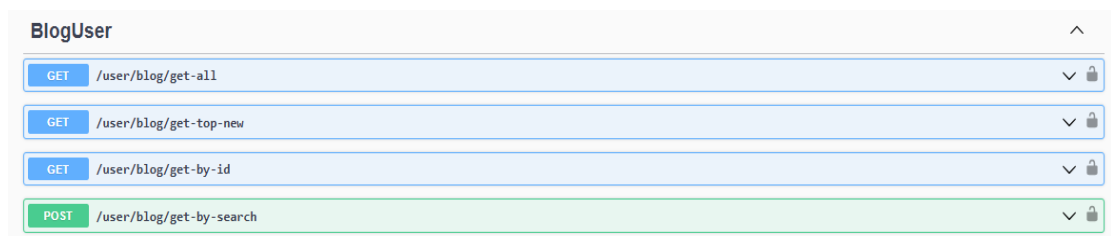
Hình 3. 1: Giao diện Swagger API definition 1.0 cho user.

- Controller xác thực user: **AuthenUser**.



Hình 3. 2: Các api dùng để xác thực người dùng và thay đổi mật khẩu

- Controller xem blog cho user: **BlogUser**.



Hình 3. 3: Các api dùng để người dùng xem và tìm kiếm blog

- Controller Thanh toán bằng Momo và VnPay: **Momo + VNPay**.

Momo			^
POST	/user/payment/momoQR		✓ 🔒
PaymentUser			✓
ProfileUser			✓
ReservationUser			✓
RoomStarUser			✓
RoomUser			✓
VNPay			^
POST	/user/vn-pay/create		✓ 🔒
GET	/user/vn-pay/response		✓ 🔒

Hình 3. 4: Các api thanh toán bằng Momo và VNPay

- Controller quản lý phòng cho user: **RoomUser**.

RoomUser			^
GET	/user/room/get-all		✓ 🔒
GET	/user/room/get-top-new		✓ 🔒
GET	/user/room/get-top-on-sale		✓ 🔒
GET	/user/room/get-by-id		✓ 🔒
POST	/user/room/get-all-by		✓ 🔒
GET	/user/search-room		✓ 🔒

Hình 3. 5: API dùng cho người dùng quản lý thông tin phòng.

- Controller quản lý thông tin cho user: **PaymentUser** + **ProfileUser**.

PaymentUser			^
POST	/user/invoice/create		✓ 🔒
GET	/user/invoice/get-all		✓ 🔒
ProfileUser			^
POST	/user/send-file		✓ 🔒
GET	/user/delete-file		✓ 🔒
GET	/user/get-file		✓ 🔒
POST	/user/user-profile/update		✓ 🔒
GET	/user/user-profile/get		✓ 🔒

Hình 3. 6: API quản lý thông tin cá nhân và thông tin thanh toán cho người dùng.

- Controller quản lý thanh toán cho user: **ReservaionUser**.

ReservationUser			^
POST	/user/reservation/create		✓ 🔒
GET	/user/reservation/delete-if-failed		✓ 🔒
GET	/user/reservation/get-successful		✓ 🔒
POST	/user/reservation/edit-info		✓ 🔒
GET	/user/reservation/get-by-id		✓ 🔒

Hình 3. 7: API quản lý thông tin hoá đơn cho người dùng.

Swagger UI		+
https://localhost:8000/swagger/index.html?urls.primaryName=2.0		🔍 A ☆ ⚙️ 📄 📌 🔄 🛑
.NET Core 6 Web API 2.0 <small>CRS5</small>		
		Authorize 🔒
AccountAdmin	✓	
AuthenAdmin	✓	
BlogAdmin	✓	
DiscountAdmin	✓	
DiscountReservationDetailAdmin	✓	
DiscountRoomDetailAdmin	✓	
DiscountServiceDetailAdmin	✓	
DiscountTypeAdmin	✓	
InvoiceAdmin	✓	
OrderServiceAdmin	✓	
ReservationAdmin	✓	
RevenueAdmin	✓	
RoleAdmin	✓	

Hình 3. 8: Giao diện Swagger API definition 2.0 cho admin

- Controller xác thực và quản lý tài khoản cho admin: **AuthenAdmin** + **AccountAdmin**.

AccountAdmin			^
GET	/v2/admin/account/get-all		✓ 🔒
GET	/v2/admin/account/get-employee-salary		✓ 🔒
GET	/v2/admin/account/get-employee		✓ 🔒
GET	/v2/admin/account/get-manager		✓ 🔒
GET	/v2/admin/account/get-user		✓ 🔒
GET	/v2/admin/account/inactive-user		✓ 🔒
AuthenAdmin			^
POST	/v2/admin/authen/login		✓ 🔒
POST	/v2/admin/authen/register-admin		✓ 🔒
POST	/v2/admin/authen/register-employee		✓ 🔒
POST	/v2/admin/authen/request-change-password		✓ 🔒
POST	/v2/admin/authen/confirm-change-password		✓ 🔒

Hình 3. 9: Các Api quản lý xác thực và quản lý tài khoản cho admin

- Controller quản lý phòng cho Admin: **RoomAdmin**.

RoomAdmin			^
POST	/v2/admin/room/create		✓ 🔒
GET	/v2/admin/room/get-all		✓ 🔒
GET	/v2/admin/room/get-by-id		✓ 🔒
GET	/v2/admin/room/get-all-by		✓ 🔒
POST	/v2/admin/room/update		✓ 🔒
GET	/v2/admin/room/delete		✓ 🔒

Hình 3. 10: Api quản lý phòng dành cho admin

- Controller quản lý giảm giá và dịch vụ: **DiscountAdmin** + **DiscountRoomDetailAdmin** + **DiscountTypeAdmin** + **DiscountServiceDetailAdmin**.

DiscountAdmin			^
POST	/v2/admin/discount/create		✓ 🔒
GET	/v2/admin/discount/get-all		✓ 🔒
GET	/v2/admin/discount/get-by-id		✓ 🔒
GET	/v2/admin/discount/get-by-search		✓ 🔒
POST	/v2/admin/discount/update		✓ 🔒
GET	/v2/admin/discount/delete		✓ 🔒
DiscountReservationDetailAdmin			^
POST	/v2/admin/discount-reservation-detail/create		✓ 🔒
DiscountRoomDetailAdmin			^
POST	/v2/admin/discount-room-detail/create		✓ 🔒
DiscountServiceDetailAdmin			^
POST	/v2/admin/discount-service-detail/create		✓ 🔒

Hình 3. 11: Các Api quản lý giảm giá và quản lý dịch vụ phòng cho admin

- Controller quản lý phân quyền: **RoleAdmin**.

RoleAdmin			^
POST	/v2/admin/role/create		✓ 🔒
GET	/v2/admin/role/get-all		✓ 🔒
POST	/v2/admin/role/add-role		✓ 🔒

Hình 3. 12: Api phân quyền cho admin

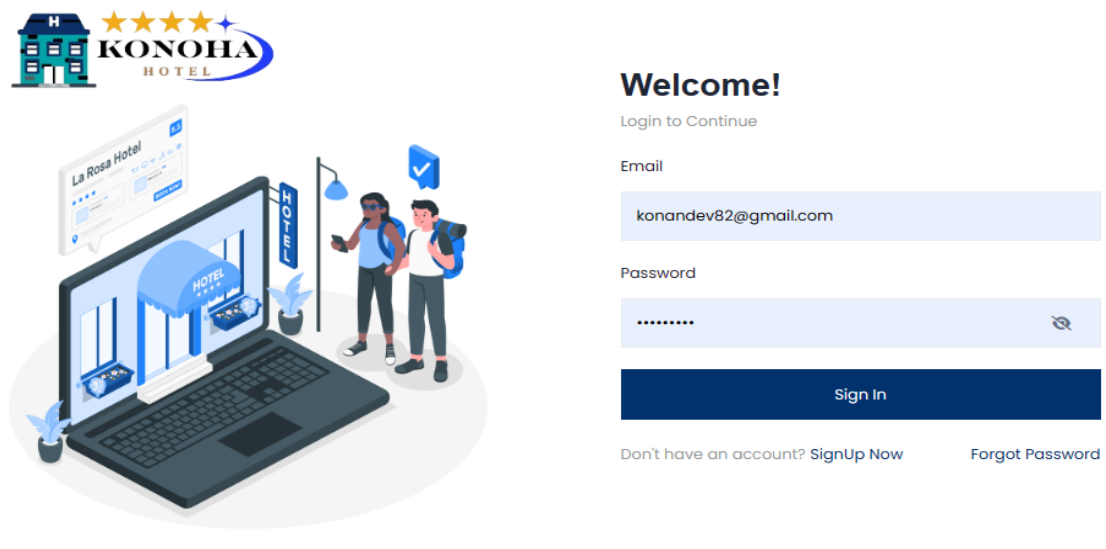
- Controller thống kê doanh thu: **RevenueAdmin**.

RevenueAdmin			^
GET	/v2/admin/revenue/get-sum		✓
GET	/v2/admin/revenue/revenue-by-year		✓
GET	/v2/admin/revenue/revenue-by-month		✓
GET	/v2/admin/revenue/get-for-type-room		✓
GET	/v2/admin/revenue/get-room-and-service		✓
GET	/v2/admin/revenue/get-room-top-booked		✓
GET	/v2/admin/revenue/get-user-top-booked		✓
GET	/v2/admin/revenue/get-employee-top-booked		✓
GET	/v2/admin/revenue/get-service-room-top-booked		✓

Hình 3. 13: Api thống kê doanh thu cho admin

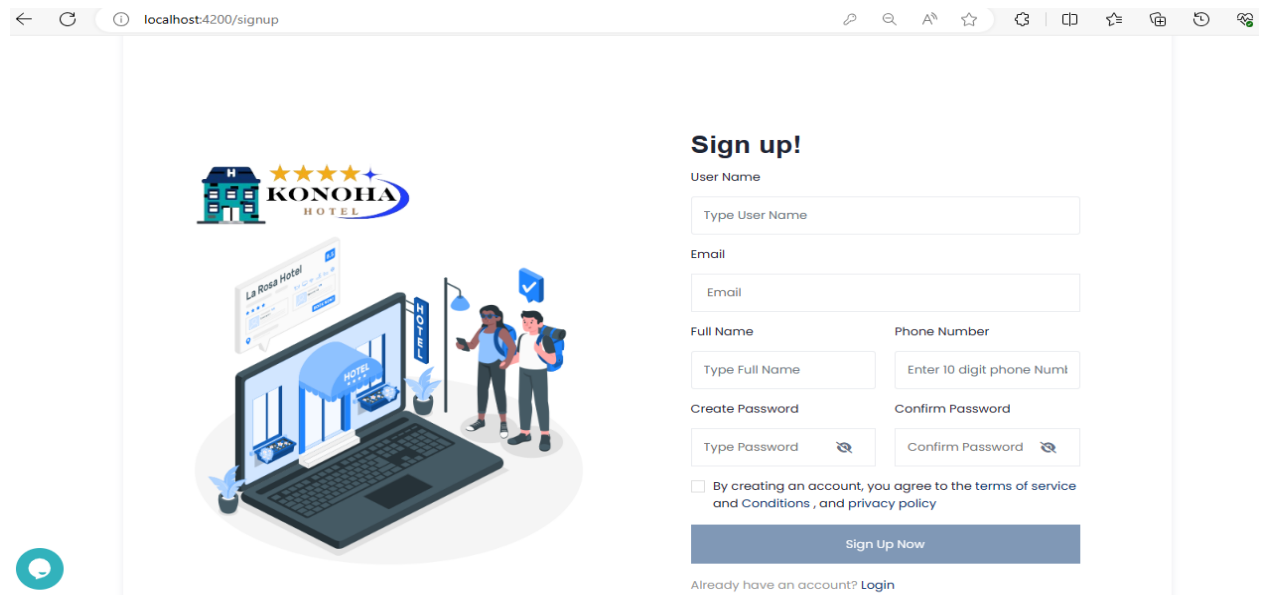
3.2. Giao diện hệ thống.

3.2.1. Giao diện đăng nhập.



Hình 3. 14: Giao diện đăng nhập bằng email và mật khẩu.

3.2.2. Giao diện đăng ký.



Sign up!

User Name
Type User Name

Email
Email

Full Name
Type Full Name

Phone Number
Enter 10 digit phone Num1

Create Password
Type Password

Confirm Password
Confirm Password

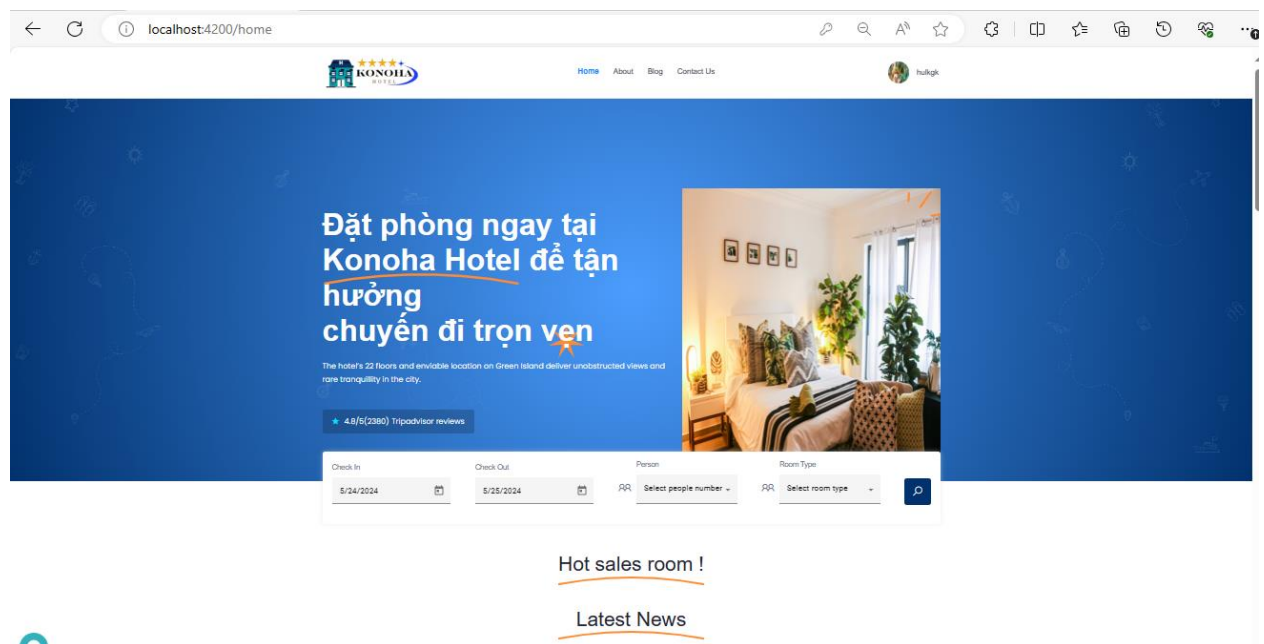
☐ By creating an account, you agree to the terms of service and Conditions , and privacy policy

Sign Up Now

Already have an account? [Login](#)

Hình 3. 15: Giao diện đăng ký bằng tên – email – số điện thoại - và mật khẩu.

3.2.3. Giao diện trang chủ user.



Đặt phòng ngay tại Konoha Hotel để tận hưởng chuyến đi trọn vẹn

The hotel's 22 floors and enviable location on Green Island deliver unobstructed views and rare tranquility in the city.

★ 4.8/5(2380) Tripadvisor reviews

Check In: 5/24/2024

Check Out: 5/25/2024

Person: Select people number

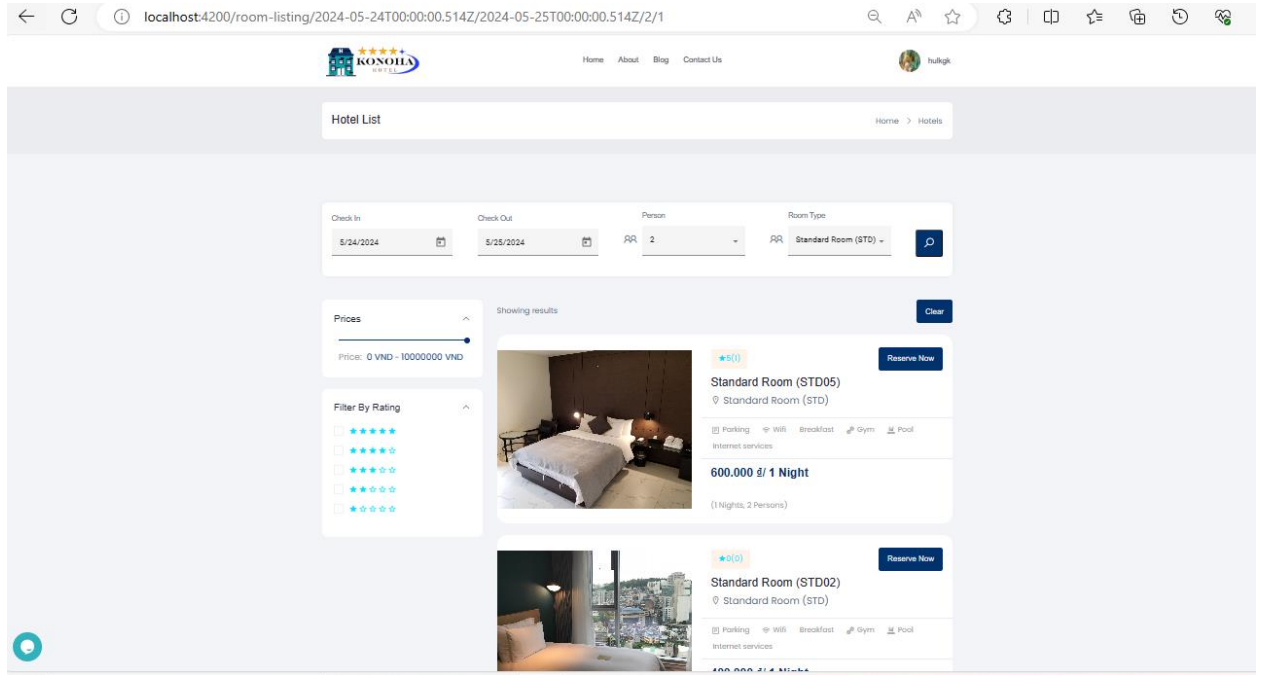
Room Type: Select room type

Hot sales room !

Latest News

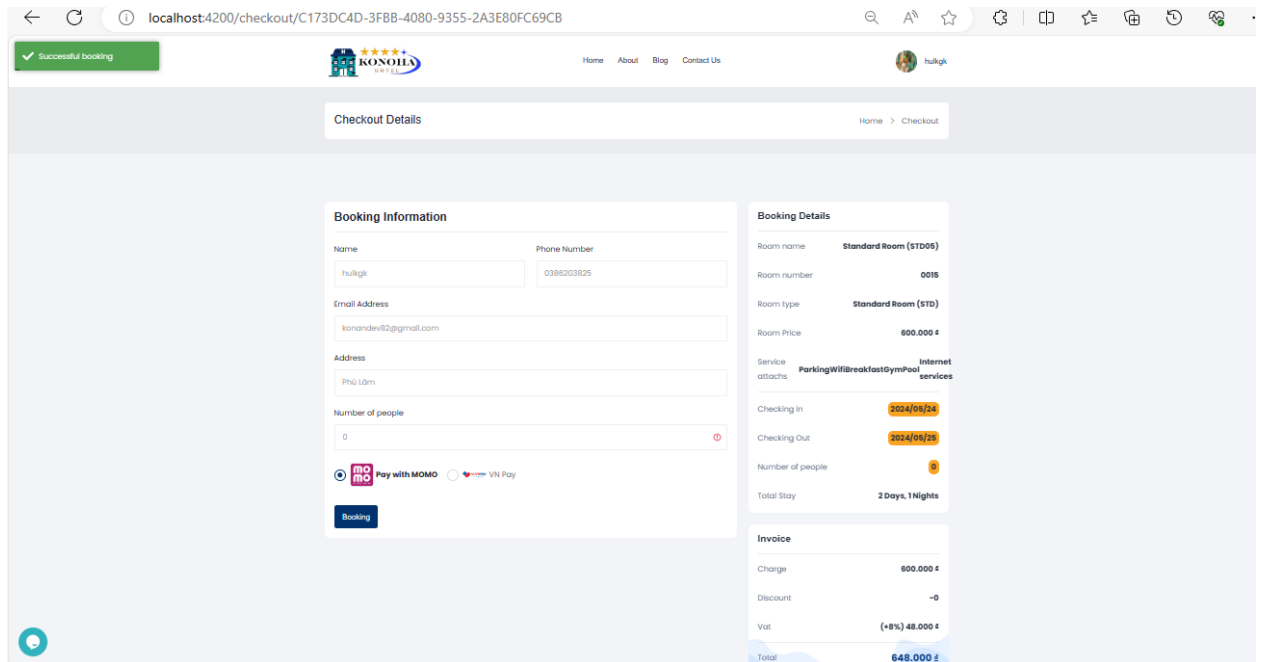
Hình 3. 16: Giao diện trang chủ user sau khi đăng nhập thành công.

3.2.4. Giao diện danh sách phòng user.



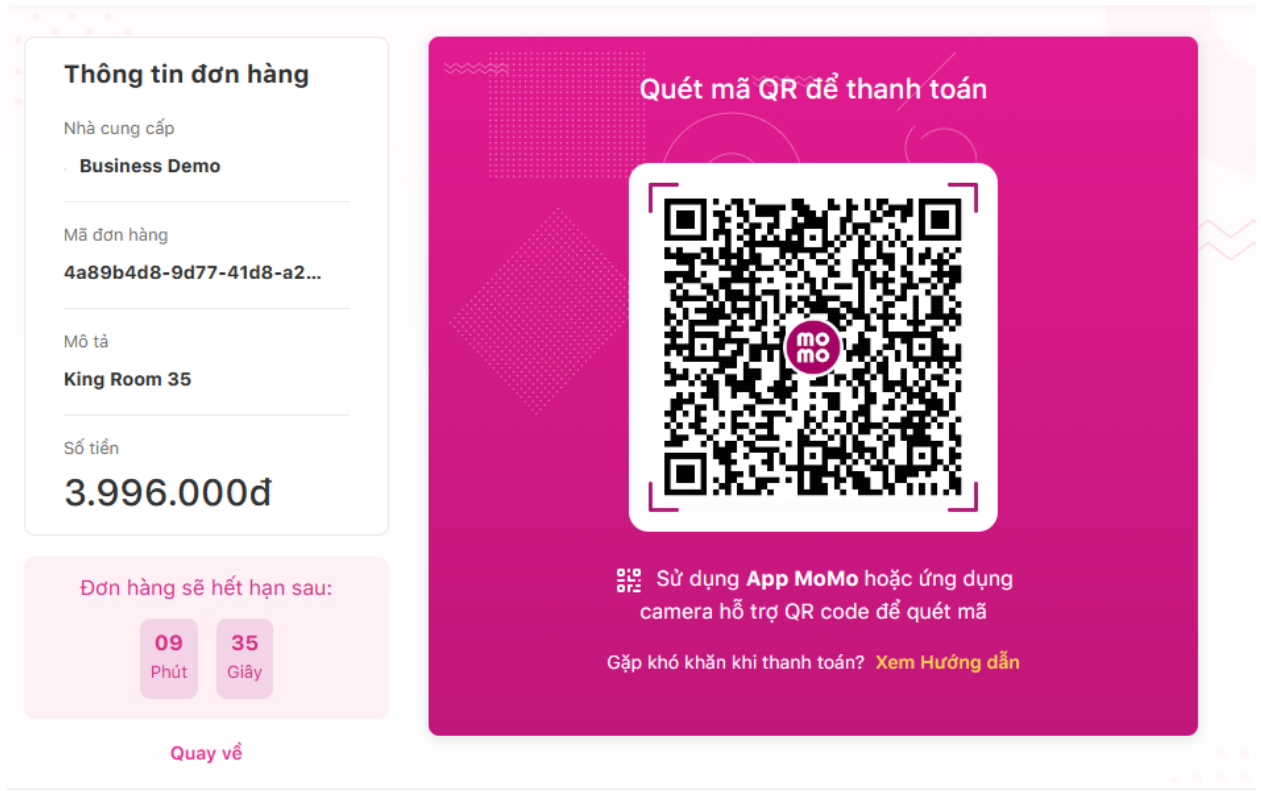
Hình 3. 17: Giao diện danh sách phòng cho user sau khi lựa chọn tìm kiếm.

3.2.5. Giao diện đặt phòng user.

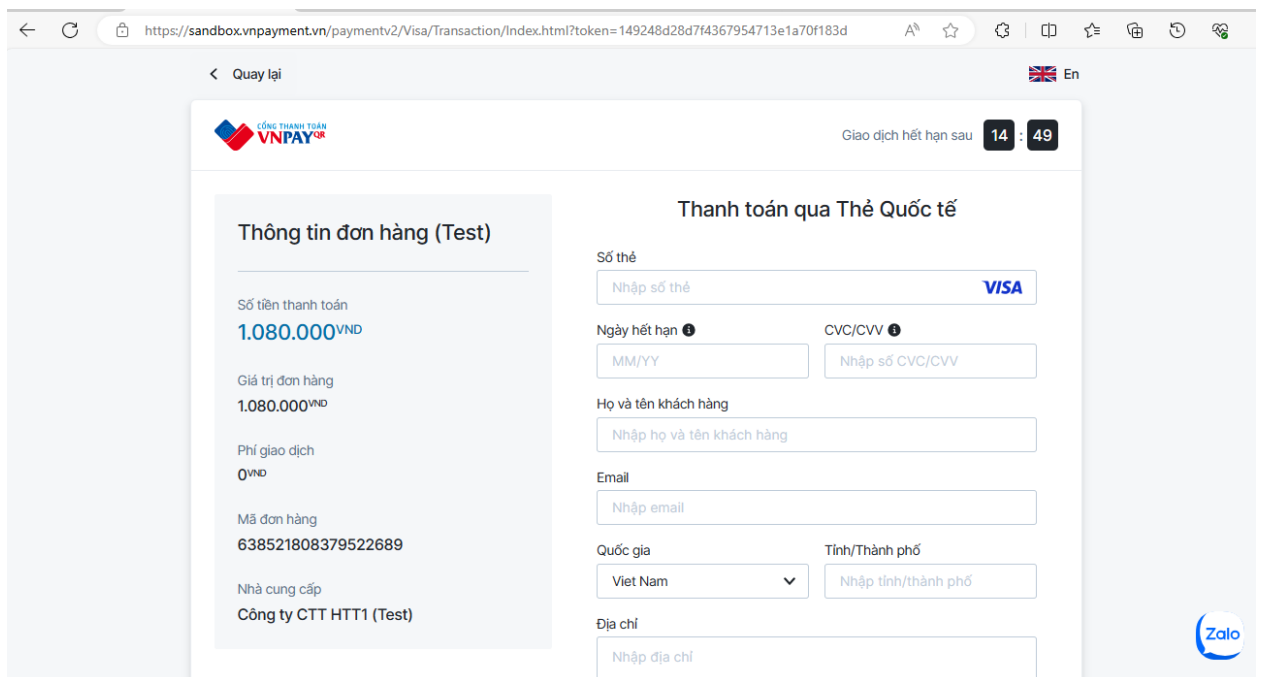


Hình 3. 18: Giao diện đặt phòng cho user sau khi lựa chọn booking phòng.

3.2.6. Giao diện thanh toán user.

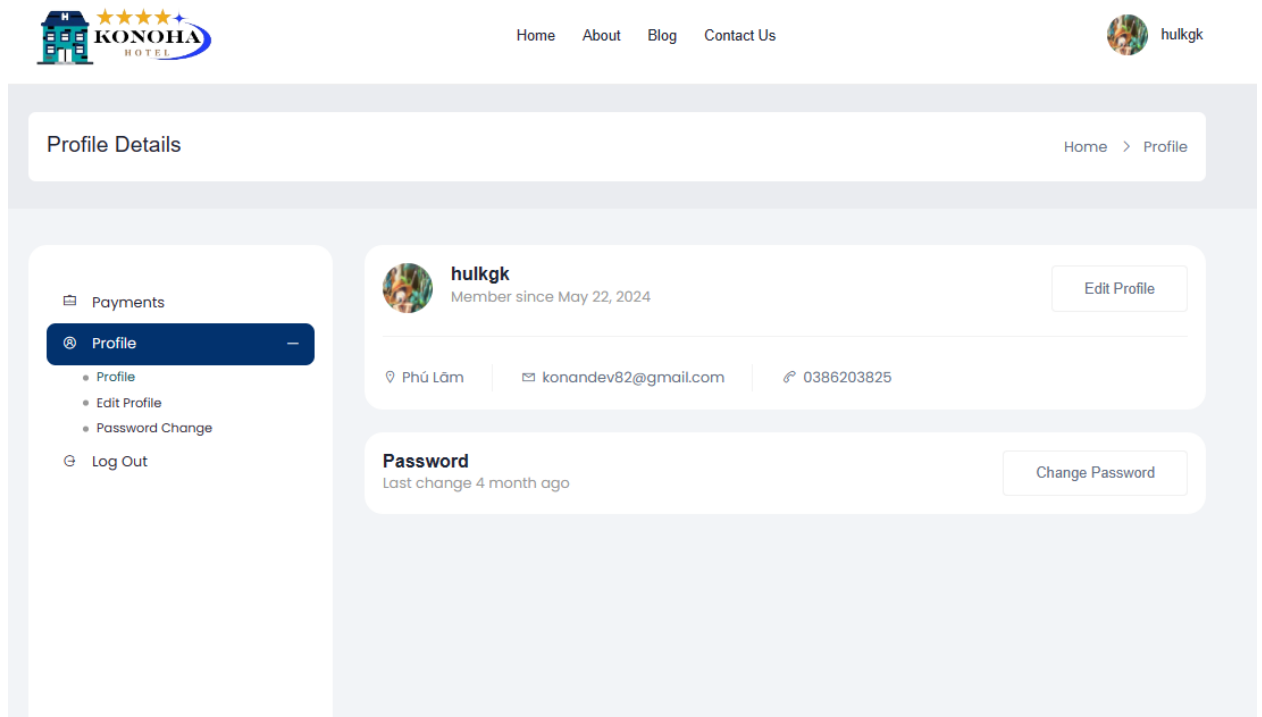


Hình 3. 19: Giao diện thanh toán Momo cho user khi chọn phương thức thanh toán Momo. [5]



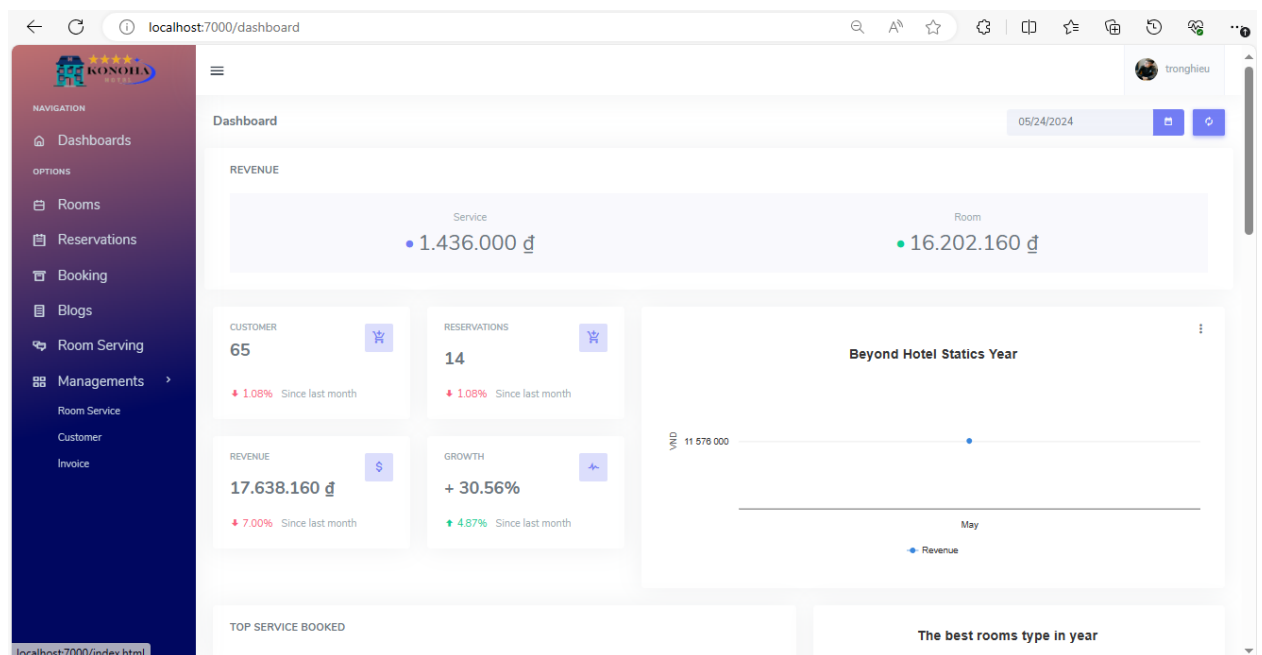
Hình 3. 20: Giao diện thanh toán bằng VN Pay cho user khi chọn phương thức thanh toán VnPay. [5]

3.2.7. Giao diện quản lý tài khoản user.



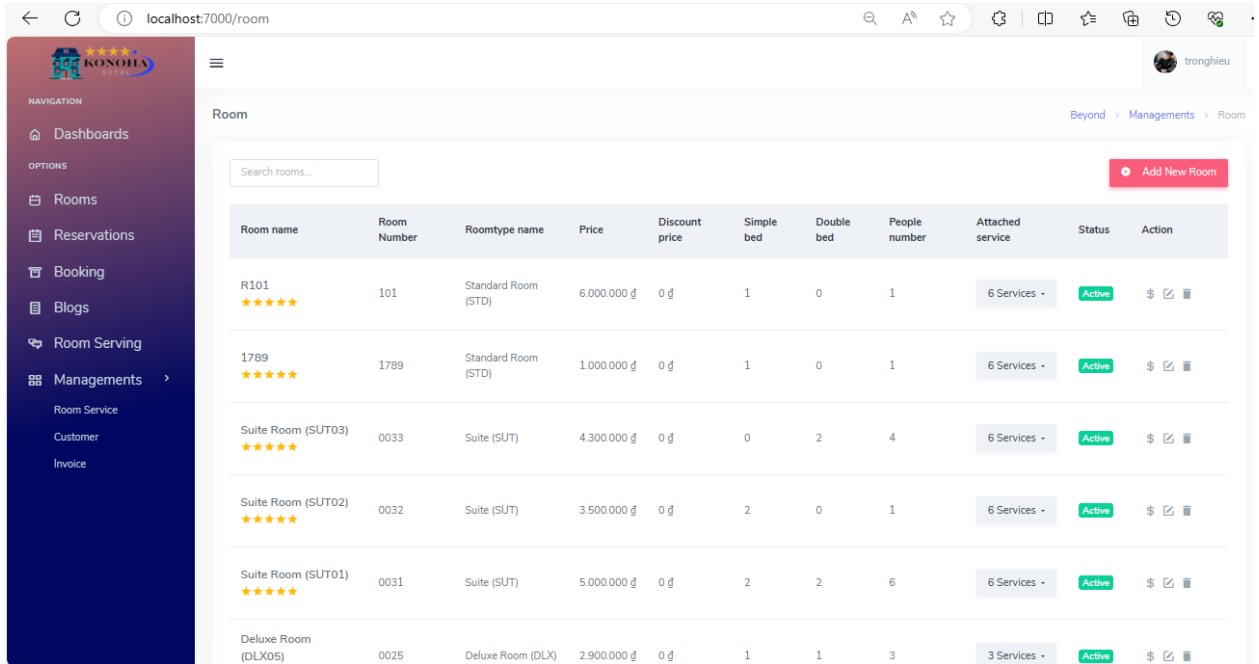
Hình 3. 21: Giao diện quản lý tài khoản cho user khi muốn xem lại thông tin của mình.

3.2.8. Giao diện trang chủ admin.



Hình 3. 22: Giao diện trang chủ admin sau khi đăng nhập bằng tài khoản Admin.

3.2.9. Giao diện quản lý phòng admin.

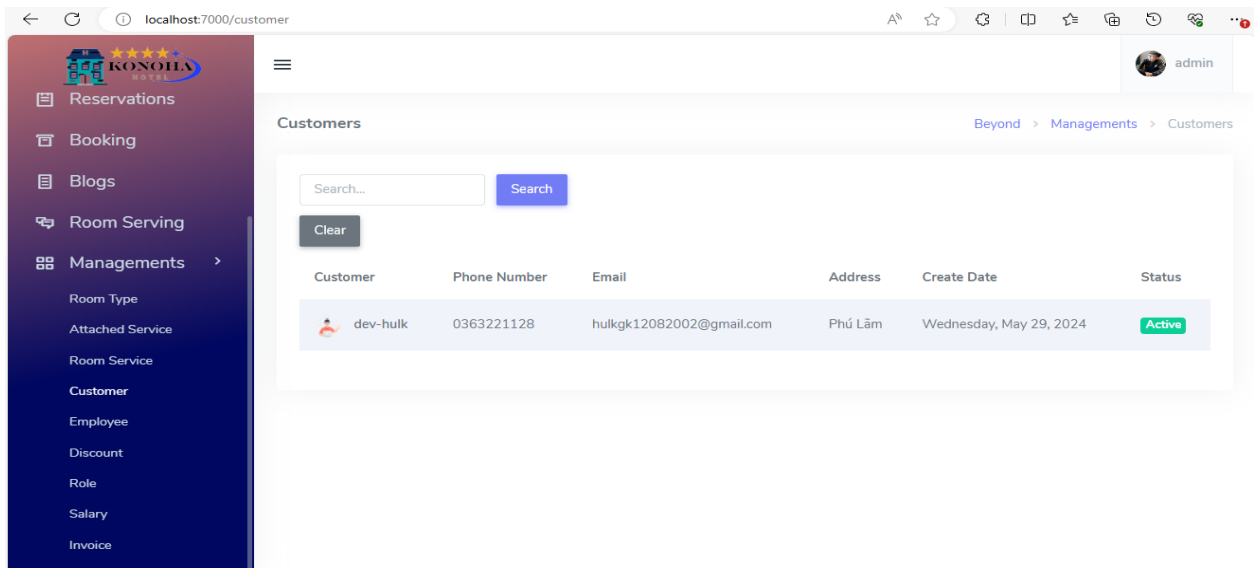


The screenshot shows the 'Room' management interface in an admin dashboard. The left sidebar contains navigation links: Dashboards, Rooms, Reservations, Booking, Blogs, Room Serving, and Managements (expanded). The main content area displays a table of rooms with columns: Room name, Room Number, Roomtype name, Price, Discount price, Simple bed, Double bed, People number, Attached service, Status, and Action. The table lists six room types: R101, 1789, Suite Room (SUT03), Suite Room (SUT02), Suite Room (SUT01), and Deluxe Room (DLX05). Each row includes a star rating, a price in Vietnamese Dong (₫), and an 'Active' status. An 'Add New Room' button is located in the top right corner of the table area.

Room name	Room Number	Roomtype name	Price	Discount price	Simple bed	Double bed	People number	Attached service	Status	Action
R101 ★★★★★	101	Standard Room (STD)	6.000.000 ₫	0 ₫	1	0	1	6 Services -	Active	\$ [icon] [icon]
1789 ★★★★★	1789	Standard Room (STD)	1.000.000 ₫	0 ₫	1	0	1	6 Services -	Active	\$ [icon] [icon]
Suite Room (SUT03) ★★★★★	0033	Suite (SUT)	4.300.000 ₫	0 ₫	0	2	4	6 Services -	Active	\$ [icon] [icon]
Suite Room (SUT02) ★★★★★	0032	Suite (SUT)	3.500.000 ₫	0 ₫	2	0	1	6 Services -	Active	\$ [icon] [icon]
Suite Room (SUT01) ★★★★★	0031	Suite (SUT)	5.000.000 ₫	0 ₫	2	2	6	6 Services -	Active	\$ [icon] [icon]
Deluxe Room (DLX05)	0025	Deluxe Room (DLX)	2.900.000 ₫	0 ₫	1	1	3	3 Services -	Active	\$ [icon] [icon]

Hình 3. 23: Giao diện quản lý phòng admin khi muốn xem danh sách và thông tin phòng.

3.2.10. Giao diện quản lý user cho admin.

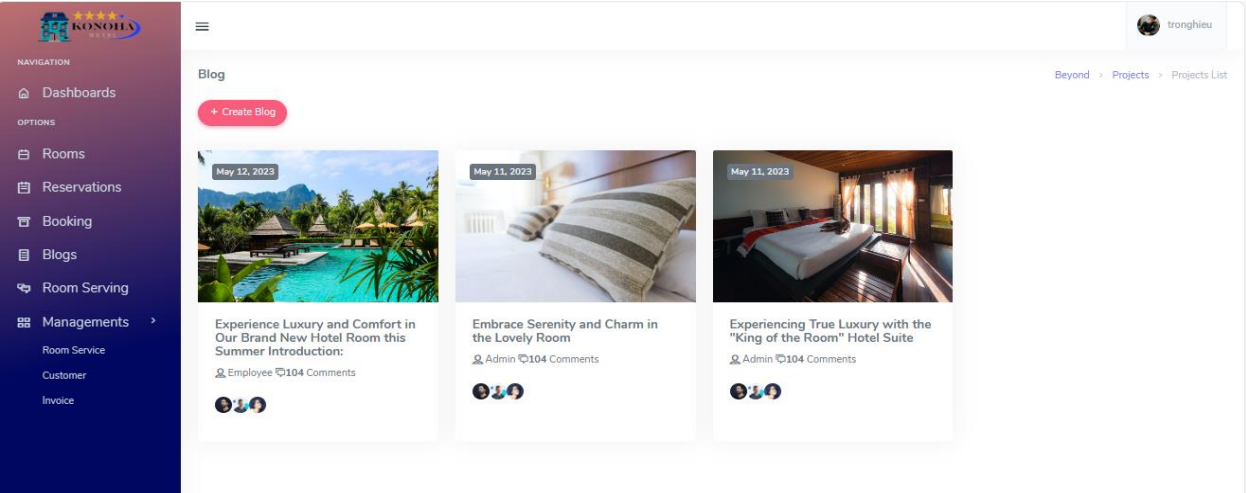


The screenshot shows the 'Customers' management interface in an admin dashboard. The left sidebar contains navigation links: Reservations, Booking, Blogs, Room Serving, and Managements (expanded). The main content area displays a table of customers with columns: Customer, Phone Number, Email, Address, Create Date, and Status. The table lists one customer: dev-hulk, with phone number 0363221128, email hulkgk12082002@gmail.com, address Phú Lâm, and create date Wednesday, May 29, 2024. The status is 'Active'. A search bar and a 'Search' button are located above the table.

Customer	Phone Number	Email	Address	Create Date	Status
dev-hulk	0363221128	hulkgk12082002@gmail.com	Phú Lâm	Wednesday, May 29, 2024	Active

Hình 3. 24: Giao diện quản lý user cho admin khi muốn xem danh sách người dùng.

3.2.11. Giao diện quản lý Blog cho admin.



Hình 3. 25: Giao diện quản lý blog cho admin.

3.3. Kiểm thử hệ thống.

3.3.1. Kiểm thử chức năng tìm kiếm phòng.

Bảng 3. 1: Bảng mô tả kiểm thử tìm kiếm phòng

ID	Mô tả	Hành động	Mong đợi	Thực tế	Trạng thái	Ngày test	Tester
Test Giao diện - GUI							
GUI - 01	[Check In] Calender	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	10/05/2024	Hiếu
GUI – 02	[Check Out] Calender	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	10/05/2024	Hiếu
GUI – 03	[Person] Select	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	10/05/2024	Hiếu
GUI – 04	[Room Type] Select	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	10/05/2024	Hiếu
GUI - 05	[Search] Search Bar	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	10/05/2024	Hiếu
Test chức năng - Function							

FUNC_01	Xác thực rằng trang "Tìm kiếm phòng" có trong ứng dụng	1. Khởi động Web 2. Chọn Tìm kiếm phòng trên trang Home	Giao diện đúng như bản thiết kế	Giao diện đúng như bản thiết kế	Passed	10/05/2024	Hiếu
FUNC_02	Tìm kiếm các phòng theo ngày tháng năm	1.Chọn ngày checkin 2.Chọn ngày checkout 3.Chọn số lượng person 4.Chọn room type 5.Click vào icon seach	Hiện thị được danh sách các phòng theo đúng yêu cầu	Hiện thị được danh sách các phòng theo đúng yêu cầu	Passed	10/05/2024	Hiếu
FUNC_03	Kiểm tra khi không nhập dữ liệu và chọn Search	Tại giao diện Tìm kiếm phòng 1. Không nhập dữ liệu 2. Nhấn vào icon Search	Hiện thị thông báo "You have to select to Checkout day!"	Hiện thị thông báo "You have to select to Checkout day!"	Passed	10/05/2024	Hiếu

3.3.2. Kiểm thử chức năng đặt phòng trực tuyến.

Bảng 3. 2: Bảng mô tả kiểm thử đặt phòng online

ID	Mô tả	Hành động	Mong đợi	Thực tế	Trạng thái	Ngày test	Tester
Test Giao diện - GUI							
GUI - 01	[Name] TextBox	Check hiển thị	- Status = enable	- Status = enable	Passed	11/05/2024	Hiếu

		giao diện					
GUI – 02	[Phone Number] TextBox	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	11/05/2024	Hiếu
GUI – 03	[Email Address] TextBox	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	11/05/2024	Hiếu
GUI – 04	[Address] TextBox	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	11/05/2024	Hiếu
GUI - 05	[Pay with Momo] CheckBox	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	10/05/2024	Hiếu
Test chức năng - Function							
FUNC_01	Để trống tất cả các trường	1. Để trống tất cả các trường 2. Chọn Booking	2. Hiện thị thông báo "One or more validation errors occurred"	2. Hiện thị thông báo "One or more validation errors occurred"	Passed	11/05/2024	Hiếu
FUNC_02	Để trống trường Name	1. Để trống trường Name 2. Nhập đúng và đủ các trường còn lại 3. Chọn Booking	2. Hiện thị thông báo "One or more validation errors occurred"	2. Hiện thị thông báo "One or more validation errors occurred"	Passed	11/05/2024	Hiếu
FUNC_03	Để trống trường Phone Number	1. Để trống trường Phone 2. Nhập đúng và đủ các trường còn lại	3. Hiện thị thông báo "One or more validation errors occurred"	3. Hiện thị thông báo "One or more validation errors occurred"	Passed	11/05/2024	Hiếu

		3. Chọn Booking					
--	--	-----------------	--	--	--	--	--

3.3.3. Kiểm thử chức năng thanh toán online.

Bảng 3. 3: Bảng mô tả kiểm thử thanh toán online.

ID	Mô tả	Hành động	Mong đợi	Thực tế	Trạng thái	Ngày test	Tester
Test Giao diện - GUI							
GUI - 01	[Ứng dụng thanh toán hỗ trợ VNPAY] Button	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	15/05/2024	Hiếu
GUI – 02	[Thẻ nội địa và tài khoản ngân hàng] Button	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	15/05/2024	Hiếu
GUI – 03	[Thẻ thanh toán quốc tế] Button	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	15/05/2024	Hiếu
GUI – 04	[Ví điện tử VNPAY] Button	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	15/05/2024	Hiếu
GUI - 05	[1900.5555.77] Button	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	15/05/2024	Hiếu
GUI – 06	[Nhà cung cấp] TextBox	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	15/05/2024	Hiếu
GUI – 07	[Mã đơn hàng] TextBox	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	15/05/2024	Hiếu
GUI – 08	[Mô tả] TextBox	Check hiển thị	- Status = enable	- Status = enable	Passed	15/05/2024	Hiếu

		giao diện					
GUI – 09	[Số tiền] TextBox	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	15/05/2024	Hiếu
GUI - 10	[Đơn hàng sẽ hết hạn sau:] TimeBox	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	15/05/2024	Hiếu
Test chức năng - Function							
FUNC_01	Xác thực rằng mục "Thanh toán Momo" có trong website	1. Khởi động Web 2. Chọn phòng 3. Chọn Pay with MOMO 4. Chọn Booking	Giao diện thanh toán MOMO hiển thị đúng như bản thiết kế	Giao diện thanh toán MOMO hiển thị đúng như bản thiết kế	Passed	15/05/2024	Hiếu
FUNC_02	Xác thực rằng mục "Thanh toán VNPAY" có trong website	1. Khởi động Web 2. Chọn phòng 3. Chọn VNPay 4. Chọn Booking	Giao diện thanh toán VNPAY hiển thị đúng như bản thiết kế	Giao diện thanh toán VNPAY hiển thị đúng như bản thiết kế	Passed	15/05/2024	Hiếu
FUNC_03	Thanh toán thành công trong trường hợp nhập đúng và đủ các thủ tục có trên giao diện MOMO	1. Nhập đúng và đủ các thủ tục có trên giao diện 2. Chọn Thanh toán	2. Hiển thị thông báo - "Successful" - Button "Back to Home" và "Download Invoice"	2. Hiển thị thông báo - "Successful" - Button "Back to Home" và "Download Invoice"	Passed	15/05/2024	Hiếu
	Thanh toán	1. Nhập đúng và	2. Hiển thị thông báo	2. Hiển thị thông báo	Passed	15/05/2024	Hiếu

FUNC_04	thành công trong trường hợp nhập đúng và đủ các thủ tục có trên giao diện VNPAY	đủ các thủ tục có trên giao diện 2. Chọn Thanh toán	- "Successful" - Button "Back to Home" và "Download Invoice"	- "Successful" - Button "Back to Home" và "Download Invoice"			
FUNC_05	Làm thiếu các thủ tục thanh toán có trên giao diện	1. Nhập thiếu không đúng các thủ tục có trên giao diện 2. Chọn Thanh toán	2. Hiện thị thông báo - "Transaction denied by user. Your payment is denied!" - Button "Back to Home" và "Download Invoice"	2. Hiện thị thông báo - "Successful" - Button "Back to Home" và "Download Invoice"	Passed	15/05/2024	Hiếu
FUNC_06	Chọn vào Button Xem hướng dẫn	Chọn Button Xem hướng dẫn	Hiện thị thông báo - " Hướng dẫn quét mã " - " Button Đóng "	Hiện thị thông báo - " Hướng dẫn quét mã " - " Button Đóng "	Passed	15/05/2024	Hiếu
FUNC_07	Chọn vào Button Quay về	Chọn Button Quay về	Hiện thị thông báo - "Hủy giao dịch thanh toán" - "Bạn chắc chắn muốn hủy giao dịch thanh toán với Business Demo?" - Button "ĐÓNG" và "HỦY	Hiện thị thông báo - "Hủy giao dịch thanh toán" - "Bạn chắc chắn muốn hủy giao dịch thanh toán với Business Demo?" - Button "ĐÓNG" và "HỦY	Passed	15/05/2024	Hiếu

			GIAO DỊCH"	GIAO DỊCH"			
FUNC_08	Chọn vào Buuton Quay lại	Chọn Button Quay lại	Hiển thị thông báo - "Hủy thanh toán" - "Quý khách có chắc chắn muốn hủy thanh toán giao dịch này?" - Button "Đóng" và "Xác nhận hủy"	Hiển thị thông báo - "Hủy thanh toán" - "Quý khách có chắc chắn muốn hủy thanh toán giao dịch này?" - Button "Đóng" và "Xác nhận hủy"	Passed	15/05/2024	Hiếu

3.3.4. Kiểm thử chức năng quản lý phòng.

Bảng 3. 4: Bảng mô tả kiểm thử quản lý phòng.

ID	Mô tả	Hành động	Mong đợi	Thực tế	Trạng thái	Ngày test	Tester
Test Giao diện - GUI							
GUI - 01	[Room name, Roomtype name, Price, Discount price, Simple bed, Double bed, People number, Status, Action] GridView	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	10/05/2024	Hiếu
GUI – 02	[Add Room] Button	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	10/05/2024	Hiếu

GUI – 03	[Room Number] TextBox	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	10/05/2024	Hiếu
GUI – 04	[Room Name] TextBox	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	10/05/2024	Hiếu
GUI - 05	[Status] Select	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	10/05/2024	Hiếu
Test chức năng - Function							
FUNC_01	Xác thực rằng mục "Quản lý phòng" có trong website	1. Khởi động Web 2. Chọn Room	Giao diện hiển thị đúng như bản thiết kế	Giao diện hiển thị đúng như bản thiết kế	Passed	10/05/2024	Hiếu
FUNC_02	Xác thực rằng mục "Add Room" có trong website	1. Khởi động Web 2. Chọn Room 3. Chọn Add New	3 .Hiển thị đúng như bản thiết kế	4 .Hiển thị đúng như bản thiết kế	Passed	10/05/2024	Hiếu
FUNC_03	Thêm phòng thành công trong trường hợp nhập đúng và đủ các trường trên giao diện	1. Nhập dữ liệu đúng và đầy đủ vào tất cả các trường có trên giao	2. Hiển thị thông báo "Room created successfully! "	2. Hiển thị thông báo "Room created successfully! "	Passed	10/05/2024	Hiếu

		diện 2. Chọn Button Add					
FUNC_04	Thêm mới phòng thành công trong trường hợp nhập đúng và đủ chỉ các trường bắt buộc trên giao diện	1. Nhập dữ liệu đúng và đầy đủ vào tất cả các trường có trên giao diện. Các trường không bắt buộc để trống. 2. Click button Add	2. Hiện thị thông báo "Room created successfully!"	2. Hiện thị thông báo "Room created successfully!"	Passed	10/05/2024	Hiếu
FUNC_05	Để trống tất cả các trường	1. Để trống tất cả các trường 2. Chọn button Add	2. Hiện thị thông báo "...is required " cho các trường bắt buộc	2. Hiện thị thông báo "...is required " cho các trường bắt buộc	Passed	10/05/2024	Hiếu

3.3.5. Kiểm thử chức năng quản lý doanh thu.

Bảng 3. 5: Bảng mô tả kiểm thử quản lý doanh thu.

ID	Mô tả	Hành động	Mong đợi	Thực tế	Trạng thái	Ngày test	Tester
Test Giao diện - GUI							

GUI - 01	[UI – Responsive Web] Menu	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	12/05/2024	Hiếu
GUI – 02	[UI – Responsive Web] Header	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	12/05/2024	Hiếu
GUI – 03	[UI – Responsive Web] Column chart	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	12/05/2024	Hiếu
GUI – 04	[Dashboard] Button, Font, Icon	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	12/05/2024	Hiếu
GUI - 05	[Header] Button, Font, Icon, Picture	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	12/05/2024	Hiếu
Test chức năng - Function							
FUNC_01	Xác thực rằng trang "Dashboard" trong ứng dụng	1. Khởi động Web 2. Chọn Home trên Menu	Giao diện đúng như bản thiết kế	Giao diện đúng như bản thiết kế	Passed	12/05/2024	Hiếu
FUNC_02	Chuyển hướng các trang khác trên thanh Header	Click vào các button trên thanh header để chuyển trang	Có thể chuyển hướng các sang trang khác	Có thể chuyển hướng các sang trang khác	Passed	12/05/2024	Hiếu
FUNC_03	Chuyển hướng các trang khác trên thanh Header	Click vào các button trên thanh header để chuyển trang	Có thể chuyển hướng các sang trang khác	Có thể chuyển hướng các sang trang khác	Passed	12/05/2024	Hiếu

3.3.6. Kiểm thử chức năng quản lý hoá đơn.

Bảng 3. 6: Bảng mô tả kiểm thử quản lý hoá đơn.

ID	Mô tả	Hành động	Mong đợi	Thực tế	Trạng thái	Ngày test	Tester
Test Giao diện - GUI							
GUI - 01	[UI – Responsive Web] Menu	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	12/05/2024	Hiếu
GUI – 02	[UI – Responsive Web] header	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	12/05/2024	Hiếu
GUI – 03	[Menu] Button, Font, Icon	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	12/05/2024	Hiếu
GUI – 04	[Header] Button, Font, Icon	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	12/05/2024	Hiếu
GUI - 05	[Search Booking] TextBox, Button	Check hiển thị giao diện	- Status = enable	- Status = enable	Passed	12/05/2024	Hiếu
Test chức năng - Function							
FUNC_01	Xác thực rằng mục "Quản lý hóa đơn" có trong website	1. Khởi động Web 2. Chọn Managment 3. Chọn Invoice	Giao diện hiển thị đúng như bản thiết kế	Giao diện hiển thị đúng như bản thiết kế	Passed	12/05/2024	Hiếu
FUNC_02	Điền trống search bar	Click chuột vào ô search bar	Hiện thị tất cả bản ghi có trong hệ thống	Hiện thị tất cả bản ghi có trong hệ thống	Passed	12/05/2024	Hiếu
FUNC_03	Nhập từ khóa tìm kiếm chỉ toàn chữ thường	1. Click chuột vào ô search bar 2. Nhập từ khóa tìm kiếm chỉ toàn chữ thường 3. Chọn Btn Search	3. Hiện thị danh sách tìm kiếm có chứa từ khóa tìm kiếm ở gridview	3. Hiện thị danh sách tìm kiếm có chứa từ khóa tìm kiếm ở gridview	Passed	12/05/2024	Hiếu

KẾT LUẬN

Tổng kết:

- *Kết quả đạt được:*
 - Xây dựng được đa phần các chức năng cần thiết của hệ thống đã mô tả
 - Giao diện website thân thiện, đẹp mắt cho người sử dụng.
 - Thành công trong việc nghiên cứu sử dụng công nghệ mới để phát triển dự án.
 - Hoàn thành dự án đúng thời gian quy định.
 - Nắm được công nghệ phát triển web.
 - Hoạt động ổn định trên nhiều trình duyệt và hiệu năng nhanh.
- *Những điều chưa đạt được:*
 - Do thiếu kinh nghiệm thực tế nên việc phân tích cơ sở dữ liệu gặp nhiều khó khăn, vẫn chưa tối ưu nhất đối với CSDL của hệ thống.
 - Chưa phát triển hệ thống phù hợp với các thiết bị di động.
 - Chiều sâu về công nghệ cốt lõi còn chưa hoàn chỉnh đặc biệt là logic để liên kết các dữ liệu từ nhiều luồng với nhau.

Hướng phát triển:

- *Tối ưu hóa hiệu suất và tải trang:*

Để đảm bảo ứng dụng luôn hoạt động mượt mà và nhanh chóng, chúng ta cần tiếp tục tối ưu hóa hiệu suất cả từ phía frontend và backend. Việc này bao gồm việc giảm độ trễ tải trang và cải thiện thời gian đáp ứng từ server.

- *Tăng cường an ninh thông tin:*

Với sự gia tăng của môi trường trực tuyến, bảo vệ thông tin người dùng trở thành một ưu tiên. Chúng ta có thể cải thiện các biện pháp an ninh như thêm lớp mã hóa và chống tấn công để ngăn chặn mọi nguy cơ tiềm ẩn.

- *Phát triển ứng dụng di động:*

Để mở rộng phạm vi sử dụng, việc phát triển ứng dụng di động sẽ là một bước quan trọng. Ứng dụng di động giúp người dùng dễ dàng truy cập và sử dụng dịch vụ bất kể ở đâu, tăng cường khả năng tiện lợi và sự linh hoạt.

- *Mở rộng quy mô và dịch vụ:*

Cuối cùng, một trong những hướng phát triển quan trọng nhất là mở rộng quy mô dịch vụ. Nâng cấp và mở rộng các tính năng, dịch vụ mới sẽ làm cho ứng dụng ngày càng phù hợp với nhu cầu đa dạng của người sử dụng.

TÀI LIỆU THAM KHẢO

Tiếng Việt

- [1] [Xây dựng website quản lý khách sạn \(webhotel.vn\)](#)
- [2] <https://tuhocict.com/web-api-trong-aspnet-core/>
- [3] Lê Thị Thanh Hương, Kỹ Năng Quản Lý - Phân Tích SWOT (Ebook)
- [4] Phạm Minh Tuấn 2023, Giáo trình phân tích thiết kế hệ thống thông tin quản lý (NXB Hà Nội)
- [5] [Hướng Dẫn Test | MoMo Developers](#)

Tiếng Anh

- [6] <https://dotnet.microsoft.com/en-us/apps/aspnet/apis>
- [7] Sikha Saha Bagui, Richard Walsh Earp 9/2022, Database Design Using Entity-Relationship Diagrams.
- [8] Andrew Lock 3/2021, ASP.NET Core in Action, Second Edition

Công cụ hỗ trợ

- <https://chatgpt.com/>
- <https://online.visual-paradigm.com/>
- <https://app.diagrams.net/>
- <https://firebase.google.com/>
- <https://github.com/>