BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----

****

**BÁO CÁO KẾT THÚC HỌC PHẦN**

**PHÁT TRIỂN ỨNG DỤNG WEB  
ĐỀ TÀI:**

**XÂY DỰNG PHẦN MỀM QUẢN LÝ KHÁCH SẠN**

|  |  |
| --- | --- |
| **Giáo viên hướng dẫn:** | **Th.S Bùi Chí Thành** |
| **Sinh viên thực hiện:** | **Nguyễn Trà My** |
| **Mã số sinh viên:** | **64131348** |

**Khánh Hòa – 2024**

**MỤC LỤC**

[MỤC LỤC i](#_Toc187097841)

[DANH MỤC HÌNH ẢNH iii](#_Toc187097842)

[CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI 1](#_Toc187097843)

[1.1 Giới thiệu chung về đề tài 1](#_Toc187097844)

[1.2 Mục tiêu của đề tài 1](#_Toc187097845)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 2](#_Toc187097846)

[2.1 Tổng quan về ASP.NET (Active Server Pages) 2](#_Toc187097847)

[2.1.1 Giới thiệu về ASP.NET 2](#_Toc187097848)

[2.1.2 Mô hình lập trình MVC (Model – View – Controller) 2](#_Toc187097849)

[2.2 Tổng quan về lập trình Web bằng ASP.NET MVC5 3](#_Toc187097850)

[2.2 Tổng quan về Microsoft SQL Server 5](#_Toc187097851)

[2.2.1 Khái niệm SQL Server 5](#_Toc187097852)

[2.2.2 Chức năng của SQL Server 5](#_Toc187097853)

[2.3 Tổng quan về HTML & CSS 5](#_Toc187097854)

[2.3.1 HTML là gì ? 5](#_Toc187097855)

[2.3.3 CSS là gì ? 6](#_Toc187097856)

[2.4 Bootstrap Framework 6](#_Toc187097857)

[2.5 Thư viện jQuery 7](#_Toc187097858)

[**2.5.1 jQuery là gì ?** 7](#_Toc187097859)

[CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG 8](#_Toc187097860)

[3.1 Yêu cầu chức năng 8](#_Toc187097861)

[CHƯƠNG 4. THIẾT KẾ HỆ THỐNG 10](#_Toc187097862)

[4.1 Thiết kế kiến trúc hệ thống 10](#_Toc187097863)

[4.1.1 Kiến trúc ba lớp 10](#_Toc187097864)

[4.1.2 Kiến trúc MVC 10](#_Toc187097865)

[4.2 Thiết kế cơ sở dữ liệu 11](#_Toc187097866)

[4.2.1 Mô hình dữ liệu 11](#_Toc187097867)

[4.2.2 Mô tả các bảng 11](#_Toc187097868)

[4.3 Thiết kế giao diện 16](#_Toc187097869)

[4.3.1 Chi tiết giao diện 16](#_Toc187097870)

[CHƯƠNG 5. KẾT LUẬN 21](#_Toc187097871)

[5.1 Ưu điểm của ứng dụng 21](#_Toc187097872)

[5.2 Nhược điểm của ứng dụng 21](#_Toc187097873)

[5.3 Đề xuất phát triển 21](#_Toc187097874)

[TÀI LIỆU THAM KHẢO 23](#_Toc187097875)

# DANH MỤC HÌNH ẢNH

[Hình 1. Giao diện trang chủ khách sạn 17](#_Toc187091619)

[Hình 2. Giao diện để khách hành liên hệ đặt phòng 17](#_Toc187091620)

[Hình 3. Giao diện đăng nhập 18](#_Toc187091621)

[Hình 4. Giao diện của nhân viên 18](#_Toc187091622)

[Hình 5. Giao diện danh sách phiếu đặt phòng 19](#_Toc187091623)

[Hình 6. Giao diện thêm mới thông tin đặt phòng 19](#_Toc187091624)

[Hình 7. Giao diện xem chi tiết thông tin đặt phòng 20](#_Toc187091625)

[Hình 8. Giao diện chỉnh sửa phiếu đặt phòng 20](#_Toc187091626)

[Hình 9. Giao diện xóa phiếu đặt phòng 21](#_Toc187091627)

[Hình 10. Giao diện tìm kiếm thông tin khách hàng 21](#_Toc187091628)

# CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

## 1.1 Giới thiệu chung về đề tài

Với sự phát triển của công nghệ số hiện nay, nó đang tạo ra những đột phá to lớn trong rất nhiều lĩnh vực, và đặc biệt là trong lĩnh vực kinh doanh khách sạn. Nếu như trước đây, việc đặt phòng khách sạn khá khó khăn, thì ngày nay, với sự hỗ trợ của các ứng dụng web, mọi thứ đã trở nên dễ dàng hơn rất nhiều. Khách hàng có thể dễ dàng tham khảo qua hàng loạt khách sạn, so sánh giá cả, xem hình ảnh chân thực và đọc những đánh giá khách quan trước khi đưa ra quyết định cuối cùng. Về phía nhà quản lý, các ứng dụng đóng vai trò quan trọng và gần như không thể thiếu, các ứng dụng giúp khách sạn quản lý mọi hoạt động một cách trơn tru và nâng cao hiệu quả kinh doanh.

Trong số các nền tảng phát triển web, ASP.NET MVC5 của Microsoft nổi bật với khả năng xây dựng các ứng dụng mạnh mẽ trên nền .NET Framework. Sự kết hợp giữa công nghệ LINQ và Entity Framework giúp đơn giản hóa việc tương tác với cơ sở dữ liệu, tạo điều kiện thuận lợi cho việc phát triển các ứng dụng quản lý phức tạp.

Xuất phát từ nhu cầu thực tiễn và tiềm năng phát triển của công nghệ, vì vậy em đã lựa chọn đề tài “Quản lý khách sạn” và sử dụng APS.NET MVC5 để phát triển một hệ thống quản lý hiệu quả và thân thiện với người dùng.

## 1.2 Mục tiêu của đề tài

Mục tiêu của đề tài là xây dựng một ứng dụng web quản lý khách sạn toàn diện, tạo sự hài hòa cho trải nghiệm người dùng và hiệu quả quản lý. Cụ thể, đề tài hướng tới việc tạo ấn tượng với khách hàng bằng thiết kế hiện đại, đồng thời cung cấp trải nghiệm trực quan và thân thiện, giúp khách hàng dễ tìm được những thông tin mình cần. Về phần quản lý, em xây dựng theo hướng tối giản nhưng vẫn hiện đại, bố trí chức năng hợp lý và hiển thị danh sách có tổ chức, giúp nhân viên tối ưu hóa quy trình nghiệp vụ, bao gồm quản lý phòng, khách hàng, quản lý đặt phòng, dịch vụ và báo cáo thống kê.

# CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

## 2.1 Tổng quan về ASP.NET (Active Server Pages)

### 2.1.1 Giới thiệu về ASP.NET

ASP.NET là một khung ứng dụng web (web application framework) được thiết kế và phát triển bởi Microsoft. ASP.NET là mã nguồn mở và là một tập hợp con của .NET Framework, đồng thời là phiên bản kế nhiệm của ASP cổ điển (Active Server Pages). Phiên bản 1.0 của .NET Framework được phát hành lần đầu vào tháng 1 năm 2002**.**

ASP.NET được xây dựng trên nền tảng CLR (Common Language Runtime) cho phép những lập trình viên thực thi mã bằng bất kỳ ngôn ngữ nào được hỗ trợ bới .NET language (ví dụ: C#, VB.NET, v.v.). Nó được thiết kế đặc biệt để làm việc với giao thức HTTP và hỗ trợ các nhà phát triển web tạo ra các trang web động (dynamic web pages), ứng dụng web, trang web, và dịch vụ web. ASP.NET cung cấp sự tích hợp tốt giữa HTML, CSS và JavaScript.

ASP.NET trở nên phổ biến trong giới lập trình viên vì ASP.NET là một phần mở rộng của .NET Framework, bổ sung các thư viện và công cụ hỗ trợ phát triển ứng dụng web. Nó cung cấp các thư viện cho các mẫu thiết kế web phổ biến như MVC, Editor Extensions và nền tảng xử lý yêu cầu web, cùng cú pháp mẫu trang web như Razor. ASP.NET nổi bật với hiệu suất cao, nhanh hơn nhiều nền tảng web khác trên thị trường. Với sự hỗ trợ của C#, lập trình viên có thể viết mã nền để truy cập dữ liệu và xử lý logic ứng dụng. Nhờ cú pháp Razor, ASP.NET cho phép phát triển các trang web động bằng cách kết hợp C# và HTML, đồng thời tích hợp dễ dàng với JavaScript và các framework như React và Angular để xây dựng ứng dụng trang đơn (SPA). Ngoài ra, ASP.NET hỗ trợ đa nền tảng, có thể phát triển và chạy trên Windows, Linux, Docker và macOS. Visual Studio đi kèm với các công cụ mạnh mẽ giúp xây dựng và triển khai ứng dụng .NET trên nhiều hệ điều hành khác nhau một cách linh hoạt và hiệu quả.

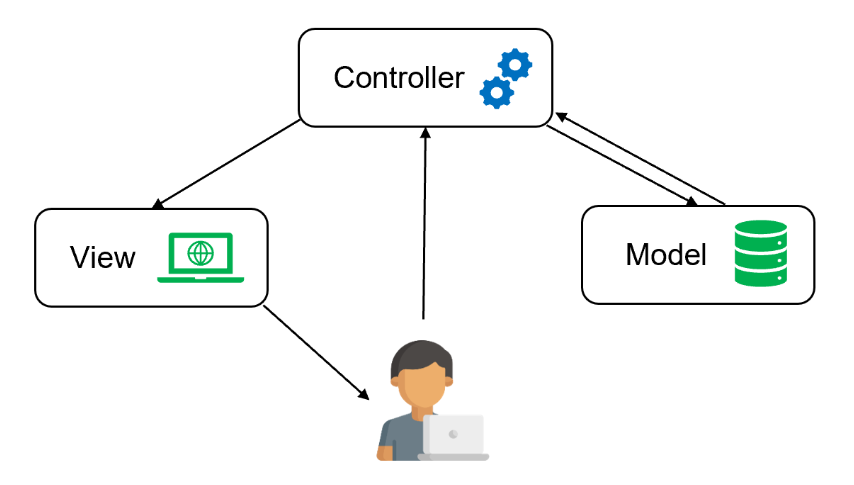
### 2.1.2 Mô hình lập trình MVC (Model – View – Controller)

MVC là mô hình thiết kế được sử dụng phổ biến trong kỹ thuật phần mềm (đặc biệt đối với phát triển ứng dụng web). Mô hình MVC thường được dùng để phát triển giao diện người dùng, cung cấp thành phần cơ bản để thiết lập chương trình điện thoại, máy tính hay các ứng dụng web. MVC được tách biệt thành 3 phần chính và có nhiệm vụ riêng biệt:

**Model:** là bộ phận có chức năng lưu trữ và xử lý toàn bộ dữ liệu của ứng dụng, là cầu nối giữa hai thành phần là View và Controller.

**View:** là nơi hiển thị dữ liệu và thu thập thông tin tương tác từ người dùng. Các View được tạo ra dựa trên dữ liệu thu thập từ model bằng cách yêu cầu thông tin từ Model, sau đó trả kết quả tới người dùng.

**Controller:** là các thành phần dùng để quản lý tương tác người dùng, làm việc với Model và chọn View để hiển thị giao diện người dùng. Trong một ứng dụng MVC, View chỉ được dùng để hiển thị thông tin, Controller chịu trách nhiệm quản lý và đáp trả nội dung người dùng nhập và tương tác với người dùng.



### 2.2 Tổng quan về lập trình Web bằng ASP.NET MVC5

ASP.NET MVC5 là một nền tảng phát triển ứng dụng web hiện đại, cho phép xây dựng các ứng dụng áp dụng mô hình MVC thay vì sử dụng mẫu thiết kế truyền thống dựa trên ASP.NET Web Forms. Một trong những ưu điểm nổi bật của ASP.NET MVC là tính nhẹ (lightweight), khả năng dễ dàng kiểm thử giao diện so với Web Forms và khả năng tích hợp các tính năng có sẵn của ASP.NET.

Một trong những lợi thế lớn nhất của ASP.NET MVC là khả năng phân tách rõ ràng các tác vụ trong ứng dụng, bao gồm logic nhập liệu, logic xử lý nghiệp vụ (business logic) và logic giao diện. Điều này giúp việc kiểm thử dễ dàng hơn và hỗ trợ phương pháp phát triển theo hướng kiểm thử (TDD - Test Driven Development). Các tính năng chính của mô hình MVC đều được triển khai dựa trên giao diện (Interface) và có thể kiểm thử thông qua các đối tượng mô phỏng (mock objects). Các đối tượng mock này giúp tái tạo hành vi của các đối tượng thực tế mà không cần thực thi chúng trong môi trường ASP.NET, hỗ trợ các bài kiểm thử đơn vị (unit test) nhanh chóng và hiệu quả.

ASP.NET MVC được thiết kế để có tính mở rộng (extensible) và khả năng nhúng (pluggable) cao. Các thành phần của MVC như view engine, bộ định tuyến URL, cơ chế kết xuất tham số của action-method và các thành phần khác đều có thể dễ dàng thay thế hoặc tùy chỉnh. Ngoài ra, ASP.NET MVC hỗ trợ các kỹ thuật lập trình hiện đại như Dependency Injection (DI) và Inversion of Control (IoC). Dependency Injection cho phép cung cấp các đối tượng cần thiết vào lớp sử dụng thay vì để lớp đó tự khởi tạo chúng, giúp tăng tính linh hoạt trong lập trình. Inversion of Control đảm bảo rằng khi một đối tượng yêu cầu một đối tượng khác, nó sẽ lấy đối tượng đó từ một nguồn bên ngoài, chẳng hạn như tập tin cấu hình, giúp đơn giản hóa quá trình kiểm thử và quản lý mã nguồn.

Hệ thống ánh xạ URL của ASP.NET MVC là một điểm nổi bật khác, cho phép tạo ra các địa chỉ URL thân thiện, dễ đọc và tối ưu hóa cho công cụ tìm kiếm. Các URL này không yêu cầu phần mở rộng của tên tập tin và được thiết kế để hỗ trợ định dạng RESTful, giúp tăng cường khả năng truy cập và sử dụng API.

ASP.NET MVC cũng cung cấp khả năng hỗ trợ các đặc tả quen thuộc từ các trang ASP.NET truyền thống như `.aspx`, điều khiển người dùng (.ascx) và trang chủ mẫu (master page). Nó cho phép lập trình viên sử dụng các tính năng sẵn có như nhúng trang chủ mẫu, biểu thức nội tuyến, điều khiển server controls, mẫu dữ liệu, liên kết dữ liệu (data-binding), hỗ trợ đa ngôn ngữ (localization) và nhiều tính năng khác.

Một điểm đặc biệt của ASP.NET MVC là việc bổ sung view engine Razor, giúp thiết kế giao diện người dùng nhanh chóng và thuận tiện hơn so với Web Forms view engine truyền thống. Razor cung cấp cú pháp đơn giản, rõ ràng và mạnh mẽ, giảm thiểu số lượng mã cần viết, đồng thời hỗ trợ việc tích hợp C# trực tiếp trong HTML để tạo các trang web động. Nhờ vậy, ASP.NET MVC trở thành một lựa chọn hàng đầu cho các lập trình viên khi phát triển các ứng dụng web hiện đại, linh hoạt và có khả năng mở rộng cao.

## 2.2 Tổng quan về Microsoft SQL Server

### 2.2.1 Khái niệm SQL Server

SQL Server hay Microsoft SQL Server là một hệ thống quản trị cơ sở dữ liệu quan hệ được phát triển bởi Microsoft vào năm 1988. SQL Server hoạt động dựa trên ngôn ngữ truy vấn cấu trúc (Structured Query Language – SQL), cho phép người dùng tạo và quản lý các cơ sở dữ liệu, bảng và các mối quan hệ giữa chúng. Phần mềm cũng cung cấp môi trường an toàn để lưu trữ thông tin quan trọng của doanh nghiệp, đồng thời hỗ trợ các công cụ để thực hiện các truy vấn phức tạp, phân tích dữ liệu và tạo báo cáo.

### 2.2.2 Chức năng của SQL Server

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ mạnh mẽ, cung cấp nhiều chức năng quan trọng để lưu trữ, quản lý và truy xuất dữ liệu. Về lưu trữ, SQL Server cho phép tổ chức dữ liệu một cách có cấu trúc và an toàn bằng cách sử dụng các bảng chứa hàng và cột. Mỗi hàng đại diện cho một bản ghi dữ liệu, còn mỗi cột biểu thị thuộc tính của bản ghi đó. Hệ thống hỗ trợ nhiều kiểu dữ liệu khác nhau như số nguyên, số thực, chuỗi ký tự, ngày giờ và giá trị logic, giúp quản lý dữ liệu linh hoạt và hiệu quả.

SQL Server còn cung cấp các công cụ quản lý tiên tiến như SQL Server Management Studio (SSMS), cho phép thực hiện các tác vụ như tạo cơ sở dữ liệu mới, định nghĩa bảng và cột, chèn hoặc cập nhật dữ liệu, sao lưu và khôi phục cơ sở dữ liệu, quản lý người dùng và quyền truy cập, cũng như giám sát và tối ưu hóa hiệu suất hệ thống.

Ngoài ra, SQL Server sử dụng ngôn ngữ truy vấn có cấu trúc (SQL) để truy xuất dữ liệu, hỗ trợ các câu lệnh phổ biến như SELECT, INSERT, UPDATE, DELETE cùng với các điều kiện lọc dữ liệu (WHERE), sắp xếp (ORDER BY), nhóm dữ liệu (GROUP BY) và kết hợp bảng (JOIN). Điều này cho phép người dùng truy vấn dữ liệu một cách linh hoạt và dễ dàng đáp ứng các nhu cầu phức tạp trong phân tích và xử lý dữ liệu.

## 2.3 Tổng quan về HTML & CSS

### 2.3.1 HTML là gì ?

HTML (viết tắt của HyperText Markup Language) là ngôn ngữ đánh dấu được sử dụng để tạo và cấu trúc nội dung trên các trang web. Nó cung cấp khung sườn cho một trang web bằng cách sử dụng các thẻ (tags) để định nghĩa các thành phần như tiêu đề, đoạn văn, hình ảnh, liên kết và biểu mẫu.

Một tài liệu HTML được hình thành bởi các phần tử HTML được quy định bằng các cặp thẻ (tag), các cặp thẻ này được bao bọc bởi một dấu ngoặc nhọn (ví dụ <html>) và thường là sẽ được khai báo thành một cặp, bao gồm thẻ mở và thẻ đóng (ví <strong> dụ </strong> và ). Các văn bản muốn được đánh dấu bằng. HTML sẽ được khai báo bên trong cặp thẻ (ví dụ <strong>Đây là chữ in đậm</strong>).

Khi trình duyệt web (như Chrome, Firefox hoặc Edge) tải một tệp HTML, nó sẽ đọc và phân tích mã HTML để hiển thị nội dung theo cấu trúc đã được chỉ định. Các thẻ HTML thường đi kèm với các thuộc tính (attributes) để cung cấp thông tin bổ sung hoặc kiểm soát cách hiển thị nội dung, chẳng hạn như thẻ <img> có thể sử dụng thuộc tính src để chỉ định nguồn hình ảnh.

### 2.3.3 CSS là gì ?

CSS (viết tắt của Cascading Style Sheets) là một ngôn ngữ thiết kế được sử dụng nhằm mục đích đơn giản hóa quá trình tạo nên một website. CSS được ra mắt vào năm 1996 bởi World Wide Web Consortium (W3C).

CSS xử lý một phần giao diện của trang web. Khi sử dụng CSS người dùng có thể kiểm soát màu sắc của văn bản, kiểu phông chữ, khoảng cách giữa các đoạn văn, cách các cột được đặt kích thước và bố cục, hình ảnh hoặc màu nền nào được sử dụng, thiết kế bố cục, các biến thể hiển thị cho các thiết bị và kích thước màn hình khác nhau cũng như hàng loạt các hiệu ứng khác.

Phương thức hoạt động của CSS là nó sẽ tìm dựa vào các vùng chọn, vùng chọn có thể là tên một thẻ HTML, tên một ID, class hay nhiều kiểu khác. Sau đó là nó sẽ áp dụng các thuộc tính cần thay đổi lên vùng chọn đó.

## 2.4 Bootstrap Framework

Bootstrap là một framework được Twitter phát triển và bootstrap là giao diện front-end phổ biến nhất, giúp tạo giao diện website đẹp, đồng bộ, và dễ bảo trì. Bootstrap sẽ quy định sẵn các thuộc tính về kích thích, màu sắc và các chiều dài, chiều rộng của các vùng website…Thông qua đó, designer có thể dễ dàng sáng tạo ra các website theo mong muốn nhưng vẫn đảm bảo tính thẩm mỹ. Đồng thời, sử dụng Bootstrap sẽ giúp tiết kiệm rất nhiều thời gian.

## 2.5 Thư viện jQuery

**2.5.1 jQuery là gì ?**

jQuery là một thư viện JavaScript phổ biến, được thiết kế để giúp đơn giản hóa việc viết mã JavaScript, đồng thời tối ưu hóa quá trình xử lý các sự kiện trên trang web. Thư viện này giúp bạn dễ dàng thực hiện các thao tác tương tác với người dùng, như thao tác DOM (Document Object Model), mà không cần phải viết mã JavaScript dài dòng và phức tạp. Một trong những tính năng đặc biệt của jQuery là khả năng tích hợp Ajax (Asynchronous JavaScript and XML), cho phép tải nội dung động mà không cần phải tải lại toàn bộ trang web.

Điều này có nghĩa là jQuery giúp tiết kiệm rất nhiều thời gian và công sức khi lập trình, vì chỉ cần viết vài dòng mã jQuery, bạn có thể thay thế cho hàng trăm dòng mã JavaScript truyền thống. Với jQuery, những thao tác phức tạp như thay đổi nội dung trang, xử lý sự kiện người dùng, hoặc hiệu ứng hoạt hình đều trở nên đơn giản và dễ thực hiện hơn. Bên cạnh đó, jQuery được thiết kế để tương thích tốt với mọi trình duyệt, giúp trang web của bạn hoạt động ổn định trên nhiều nền tảng khác nhau.

**2.5.2 jQuery mang lại lợi ích gì cho lập trình viên ?**

Một trong những ưu điểm lớn nhất của jQuery là khả năng dễ dàng truy cập và thao tác với các phần tử trong tài liệu HTML. Việc lựa chọn các phần tử trở nên đơn giản như cách bạn sử dụng CSS, thông qua các phương thức lựa chọn phần tử như $(selector). Nhờ đó, lập trình viên có thể nhanh chóng tìm và thay đổi các phần tử mà không cần phải viết mã DOM phức tạp.

jQuery không chỉ giúp thay đổi giao diện trang web một cách dễ dàng, mà còn tối ưu hóa khả năng hiển thị trên nhiều trình duyệt khác nhau. Khi sử dụng jQuery, bạn có thể thay đổi các thuộc tính CSS, thêm hoặc loại bỏ các lớp CSS, giúp việc điều chỉnh và làm mới giao diện của trang web trở nên linh hoạt và dễ dàng. Điều này rất hữu ích khi bạn muốn đảm bảo trang web của mình hiển thị chính xác trên các trình duyệt cũ và mới mà không gặp phải sự cố tương thích.

Thay vì phải sử dụng các vòng lặp phức tạp, xử lý sự kiện hay điều khiển mảng một cách thủ công, jQuery cung cấp các hàm và phương thức giúp giảm thiểu mã cần viết mà vẫn đảm bảo tính hiệu quả và khả năng mở rộng. Điều này giúp tiết kiệm thời gian và giảm thiểu sai sót trong quá trình phát triển web.

# CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG

## 3.1 Yêu cầu chức năng

Các yêu cầu chức năng được phân chia cụ thể như sau:

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên chức năng** | **Người dùng** |
| 1 | Xem trang chủ | Khách hàng, nhân viên, quản trị hệ thống |
| 2 | Xem danh sách loại phòng | Khách hàng, nhân viên, quản trị hệ thống |
| 3 | Gửi liên hệ đặt phòng | Khách hàng, nhân viên, quản trị hệ thống |
| 4 | Gửi phản hồi | Khách hàng, nhân viên, quản trị hệ thống |
| 5 | Đăng nhập | Nhân viên, quản trị hệ thống |
| 6 | Đổi mật khẩu | Nhân viên, quản trị hệ thống |
| 7 | Quản lý thông tin cá nhân | Nhân viên, quản trị hệ thống |
| 8 | Quản lý phòng | Nhân viên, quản trị hệ thống |
| 9 | Quản lý loại phòng | Nhân viên, quản trị hệ thống |
| 10 | Quản lý dịch vụ | Nhân viên, quản trị hệ thống |
| 11 | Quản lý phiếu đặt phòng | Nhân viên, quản trị hệ thống |
| 12 | Quản lý phiếu thuê phòng | Nhân viên, quản trị hệ thống |
| 13 | Quản lý hóa đơn | Nhân viên, quản trị hệ thống |
| 14 | Quản lý khách hàng | Nhân viên, quản trị hệ thống |
| 15 | Quản lý liên hệ | Nhân viên, quản trị hệ thống |
| 16 | Quản lý phản hồi | Nhân viên, quản trị hệ thống |
| 17 | Quản lý nhân viên | Quản tri hệ thống |
| 18 | Quản lý tài khoản nhân viên | Quản trị hệ thống |
| 19 | Quản lý nhóm nhân viên | Quản trị hệ thống |
| 20 | Phân quyền | Quản trị hệ thống |

**3.2 Yêu cầu phi chức năng**

|  |  |  |
| --- | --- | --- |
| **STT** | **Yêu cầu** | **Loại yêu cầu** |
| 1 | Phân quyền người dùng | Bảo mật |
| 2 | Giao diện thân thiện, trực quan, dễ sự dụng | Tiện dụng |
| 3 | Tốc độ tải trang website, chuyển tiếp giữa các trang nhanh, tốc độ phản hồi yêu cầu cao | Hiệu quả |
| 4 | Tương thích mọi loại trình duyệt web | Tương thích |

# CHƯƠNG 4. THIẾT KẾ HỆ THỐNG

## 4.1 Thiết kế kiến trúc hệ thống

### 4.1.1 Kiến trúc ba lớp

Lớp trình bày (Presentation Layer - View): Đảm nhận việc hiển thị giao diện người dùng và nhận các tương tác từ người dùng. Trong mô hình MVC, lớp trình bày tương ứng với phần View (ví dụ như các trang HTML hoặc Razor views).

Lớp ứng dụng (Application Layer - Controller): Lớp chứa các logic nghiệp vụ của ứng dụng. Trong mô hình MVC, lớp ứng dụng tương ứng với phần Controller, chịu trách nhiệm xử lý các yêu cầu từ người dùng, tương tác với lớp dữ liệu để truy xuất hoặc thay đổi thông tin, và chọn View thích hợp để trả về cho người dùng.

Lớp dữ liệu (Data Layer - Model): Lớp này quản lý và truy cập dữ liệu, đảm bảo việc lưu trữ và xử lý thông tin được thực hiện đúng cách. Trong mô hình MVC, phần Model chịu trách nhiệm biểu diễn dữ liệu và thực hiện các thao tác truy cập cơ sở dữ liệu, ví dụ như sử dụng Entity Framework Core.

### 4.1.2 Kiến trúc MVC

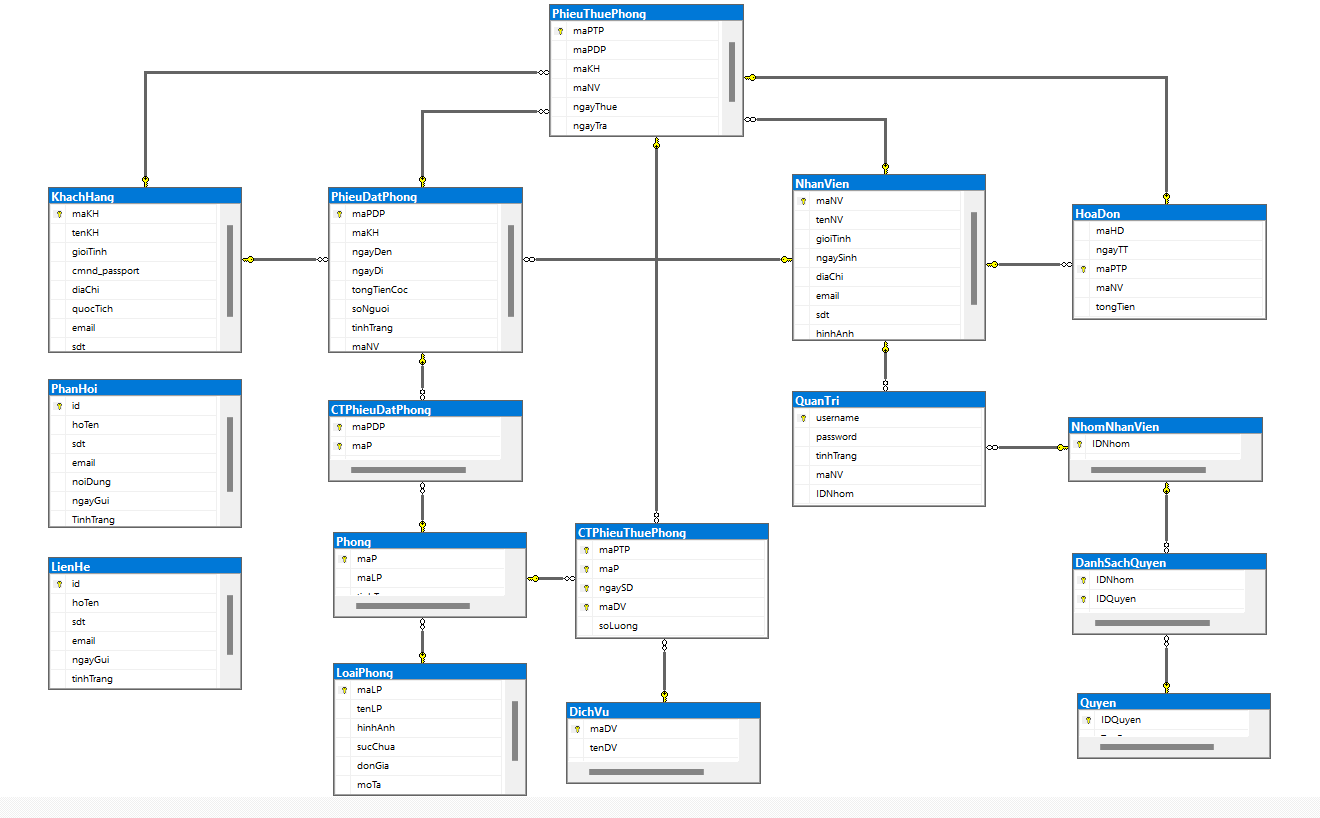
Model: Phần biểu diễn dữ liệu của ứng dụng. Thường là các lớp C# liên kết với các bảng trong cơ sở dữ liệu, chứa dữ liệu và logic xử lý dữ liệu.

View: Phần hiển thị dữ liệu cho người dùng. Nó thường là các tệp HTML, được tạo ra thông qua Razor engine trong ASP.NET MVC, giúp người dùng nhìn thấy giao diện và dữ liệu.

Controller: Bộ phận xử lý các yêu cầu từ người dùng. Nó tương tác với Model để truy xuất hoặc thay đổi dữ liệu, đồng thời chọn View phù hợp để trả về cho người dùng. Đồng thời là cầu nối giữa View và Model.

## 4.2 Thiết kế cơ sở dữ liệu

### 4.2.1 Mô hình dữ liệu



## 4.2.2 Mô tả các bảng

Bảng CTPhieuDatPhong

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô Tả |
| 1 | maPDP | varchar | <=10 | Mã Phiếu Đặt Phòng (Khóa chính) |
| 2 | maP | varchar | <=10 | Mã Phòng (Khóa chính) |
| 3 | tienCoc | decimal | (18, 0) | Tiền Cọc (Bắt buộc) |

Bảng CTPhieuThuePhong

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô Tả |
| 1 | maPTP | varchar | <=10 | Mã Phiếu Thuê Phòng (Khóa Chính) |
| 2 | maP | varchar | <=10 | Mã Phòng (Khóa Chính) |
| 3 | ngaySD | datetime | <=15 | Ngày Sử dụng (Khóa Chính) |
| 4 | maDV | varchar | <=10 | Mã Dịch vụ (Khóa Chính) |
| 5 | soLuong | int | <=50 | Số lượng (Bắt buộc) |

Bảng DanhSachQuyen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô Tả |
| 1 | IDNhom | varchar | <=20 | ID nhóm (Khóa Chính) |
| 2 | IDQuyen | varchar | <=50 | ID Quyền (Khóa Chính) |
| 3 | GhiChu | nvarchar | <=50 | Ghi Chú (Khóa Chính) |

Bảng DichVu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô Tả |
| 1 | maDV | varchar | <=10 | Mã Dịch vụ (Khóa Chính) |
| 2 | tenDV | nvarchar | <=50 | Tên dịch vụ (Bắt buộc) |
| 3 | gia | decimal | (18, 0) | Giá (Bắt buộc) |

Bảng HoaDon

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô Tả |
| 1 | maHD | varchar | <=10 | Mã hóa đơn (Khóa Chính) |
| 2 | ngayTT | varchar | <=10 | Ngày thanh toán (Bắt buộc) |
| 3 | maPTP | varchar | <=10 | Mã phiếu thuê phòng (Khóa ngoại) |
| 4 | maNV | varchar | <=10 | Mã nhân viên (Khóa ngoại) |
| 5 | tongTien | decimal | (18, 0) | Tổng tiền (Bắt buộc) |

Bảng KhachHang

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô Tả |
| 1 | maKH | varchar | <=10 | Mã khách hàng (Khóa chính) |
| 2 | tenKH | nvarchar | <=50 | Tên khách hàng (Bắt buộc) |
| 3 | gioiTinh | bit |  | Giới tính (Bắt buộc) |
| 4 | cmnd\_passport | varchar | <=15 | Chứng minh nhân dân (Bắt buộc) |
| 5 | diaChi | nvarchar | <=50 | Địa chỉ (Bắt buộc) |
| 6 | quocTich | nvarchar | <=50 | Quốc tịch (Bắt buộc) |
| 7 | email | varchar | <=50 | Email (Bắt buộc) |
| 8 | sdt | varchar | <=20 | Số điện thoại (Bắt buộc) |

Bảng LienHe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô Tả |
| 1 | Id | int | (1,1) | ID (Khóa chính) |
| 2 | hoTen | nvarchar | <=50 | Họ tên (Bắt buộc) |
| 3 | sdt | varchar | <=20 | Số điện thoại (Bắt buộc) |
| 4 | email | varchar | <=50 | Email (Bắt buộc) |
| 5 | ngayGui | date |  | Ngày gửi (Bắt buộc) |

Bảng LoaiPhong

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô Tả |
| 1 | maLP | varchar | <=10 | Mã Loại Phòng (Khóa chính) |
| 2 | tenLP | nvarchar | <=50 | Tên Loại Phòng (Bắt buộc) |
| 3 | hinhAnh | varchar | <=250 | Hình ảnh (Bắt buộc) |
| 4 | sucChua | int |  | Sửa chữa (Bắt buộc) |
| 5 | donGia | decimal | (18, 0) | Đơn giá (Bắt buộc) |
| 6 | moTa | nvarchar | <=500 | Mô tả (Bắt buộc) |

Bảng NhanVien

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô Tả |
| 1 | maNV | varchar | <=10 | Mã nhân viên (Khóa chính) |
| 2 | tenNV | nvarchar | <=50 | Tên nhân viên (Bắt buộc) |
| 3 | gioiTinh | bit | 0 | 1 | Giới tính (Bắt buộc) |
| 4 | ngaySinh | date |  | Ngày sinh (Bắt buộc) |
| 5 | diaChi | nvarchar | <=50 | Địa chỉ (Bắt buộc) |
| 6 | email | varchar | <=50 | Email (Bắt buộc) |
| 7 | sdt | varchar | <=20 | Số điện thoại (Bắt buộc) |
| 8 | hinhAnh | varchar | <=250 | Hình ảnh (Bắt buộc) |

Bảng NhomNhanVien

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô Tả |
| 1 | IDNhom | varchar | <=20 | ID nhóm (Khóa chính) |
| 2 | TenNhom | nvarchar | <=50 | Tên nhóm (Bắt buộc) |

Bảng PhanHoi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô Tả |
| 1 | Id | int |  | ID (Khóa chính) |
| 2 | hoTen | nvarchar | <=50 | Họ tên (Bắt buộc) |
| 3 | sdt | varchar | <=20 | Số điện thoại (Bắt buộc) |
| 4 | email | varchar | <=50 | Email (Bắt buộc) |
| 5 | noiDung | varchar | <=500 | Nội dung (Bắt buộc) |
| 6 | ngayGui | date |  | Ngày gửi (Bắt buộc) |

Bảng PhieuDatPhong

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô Tả |
| 1 | maPDP | varchar | <=10 | Mã phiếu đặt phòng (Khóa chính) |
| 2 | maKH | varchar | <=10 | Mã khách hàng (Khóa ngoại) |
| 3 | ngayDen | date |  | Ngày đến (Bắt buộc) |
| 4 | ngayDi | date | ngayDi >= ngayDen | Ngày đi (Bắt buộc) |
| 5 | tongTienCoc | decimal | (18,0) | Tổng tiền cọc (Bắt buộc) |
| 6 | soNguoi | int | >0 | Số người (Bắt buộc) |
| 7 | tinhTrang | bit | 0 | 1 | Tình trạng (Bắt buộc) |
| 8 | maNV | varchar | <=10 | Mã nhân viên (Khóa ngoại |

Bảng PhieuThuePhong

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô Tả |
| 1 | maPTP | varchar | <=10 | Mã phiếu thuê phòng (Khóa chính) |
| 2 | maPDP | varchar | <=10 | Mã phiếu đặt phòng (Khóa ngoại) |
| 3 | maKH | varchar | <=10 | Mã khách hàng (Khóa ngoại) |
| 4 | maNV | varchar | <=10 | Mã nhân viên (Khóa ngoại) |
| 5 | ngayThue | date |  | Ngày thuê (Bắt buộc) |
| 6 | ngayTra | date | ngayTra>=ngayThue | Ngày trả (Bắt buộc) |

Bảng Phong

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô Tả |
| 1 | maP | varchar | <=10 | Mã phòng (Khóa chính) |
| 2 | maLP | varchar | <=10 | Mã loại phòng (Khóa ngoại) |
| 3 | tinhTrang | nvarchar | <=20 | Tình trạng (Bắt buộc) |

Bảng QuanTri

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô Tả |
| 1 | username | varchar | <=20 | Tên đăng nhập (Khóa chính) |
| 2 | password | varchar | <=50 | Mật khẩu (Bắt buộc) |
| 3 | tinhTrang | bit | 0 | 1 | Tình trạng (Bắt buộc) |
| 4 | maNV | varchar | <=10 | Mã nhân viên (Khóa ngoại) |
| 5 | IDNhom | varchar | <=20 | ID nhóm (Khóa ngoại) |

Bảng Quyen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | IDQuyen | varchar | <=50 | ID quyền (Khóa chính) |
| 2 | TenQuyen | nvarchar | <=100 | Tên quyền (Bắt buộc) |

## 4.3 Thiết kế giao diện

### 4.3.1 Chi tiết giao diện

**Trang chủ của khách sạn:** Tại đây sẽ hiển thị tất cả thông tin tổng quan về khách sạn như : giới thiệu về khách sạn, hình ảnh và chi tiết giá các loại phòng, phần đặt phòng nhanh và thông tin liên hệ với khách sạn, đồng thời tích hợp phần phản hồi để nhận các góp ý từ khách hàng về khách sạn.

A screenshot of a hotel

Description automatically generated

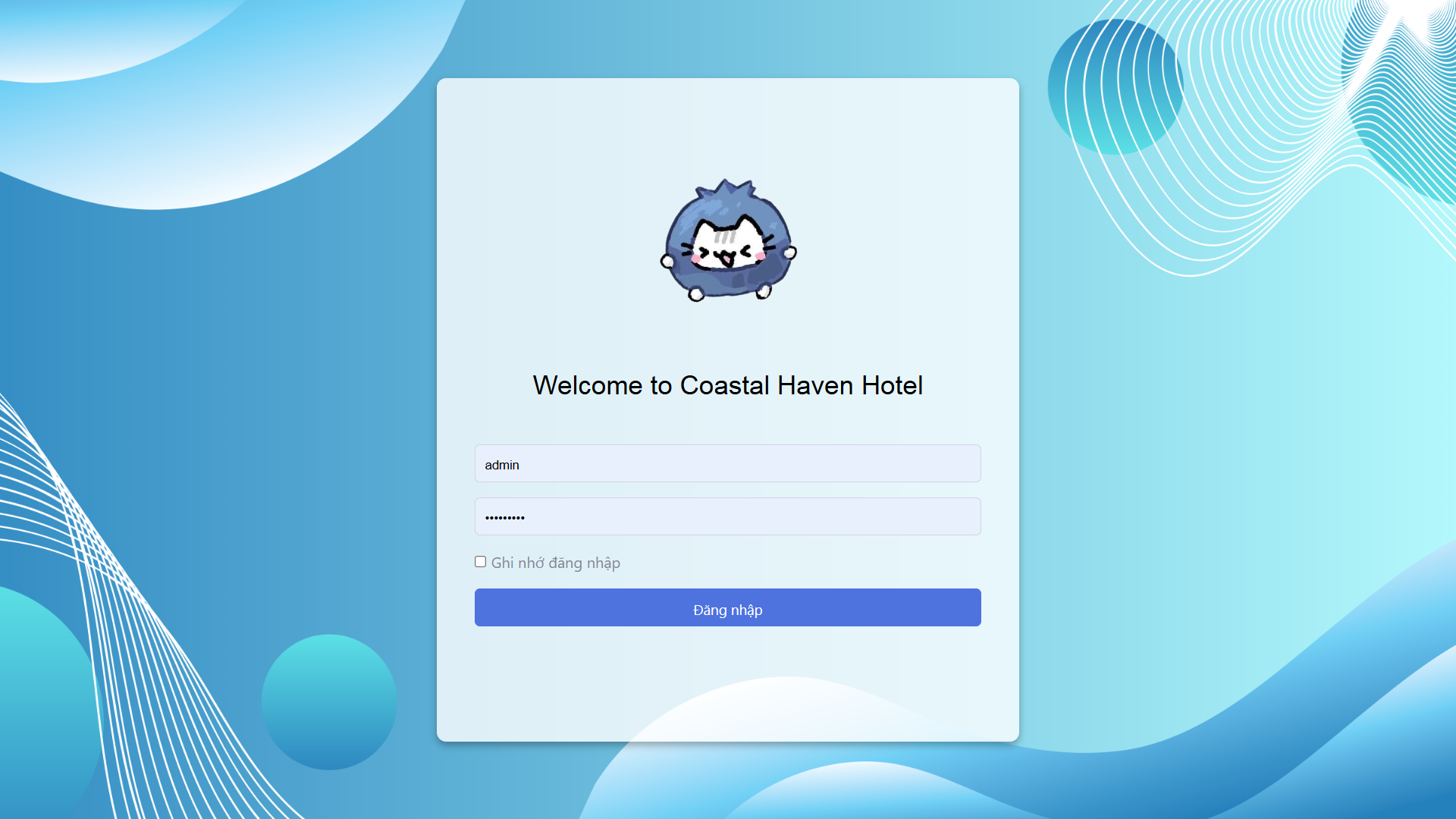
Hình . Giao diện trang chủ khách sạn

Screens screenshot of a computer

Description automatically generated

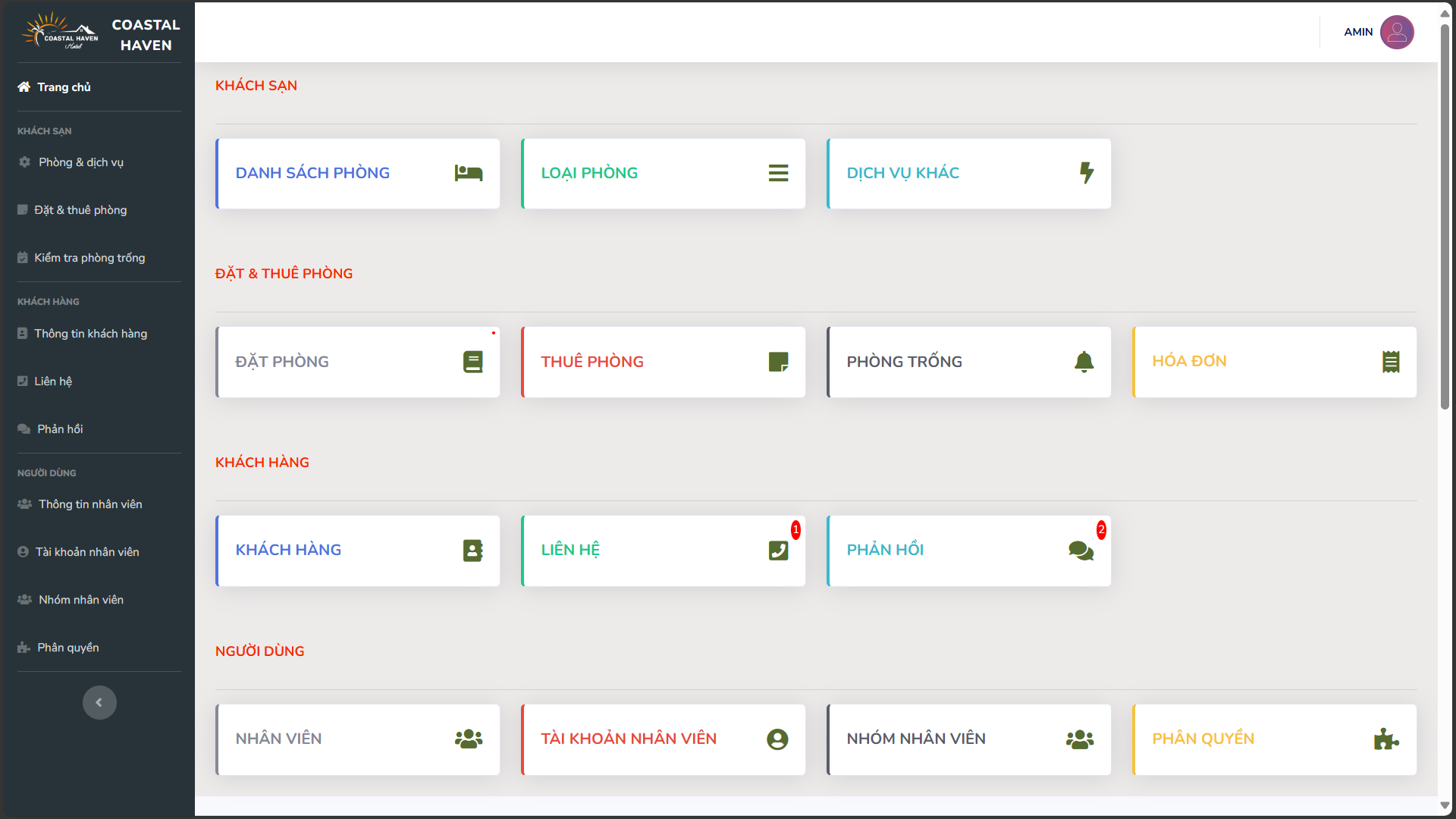
Hình . Giao diện để khách hành liên hệ đặt phòng

**Trang đăng nhập:** Giao diện đơn giản với các trường thông tin tài khoản và mật khẩu, có tính năng ghi nhớ mật khẩu và sẽ gợi ý thông tin đăng nhập cho những lần truy cập sau.



Hình . Giao diện đăng nhập

**Trang dành cho nhân viên:** Hiển thị đầy đủ thông tin về danh sách phiếu đặt/thuê phòng, kiểm tra phòng trống, danh sách dịch vụ và thông tin khách hàng….Các chức năng thêm, sửa và xóa thông tin để hỗ trợ việc quản lý hiệu quả.



Hình . Giao diện của nhân viên

**Trang danh sách phiếu đặt phòng:** Cung cấp thông tin chi tiết về các phiếu đặt phòng, đồng thời hỗ trợ các chức năng lọc, sắp xếp nhanh chóng.

A screenshot of a computer

Description automatically generated

Hình . Giao diện danh sách phiếu đặt phòng

**Chức năng thêm mới thông tin:** Cho phép nhập các thông tin phiếu đặt phòng mới. Hệ thống cũng có tích hợp tính năng kiểm tra dữ liệu đầu vào để đảm bảo tính chính xác trước khi lưu trữ.

A screenshot of a computer

Description automatically generated

Hình . Giao diện thêm mới thông tin đặt phòng

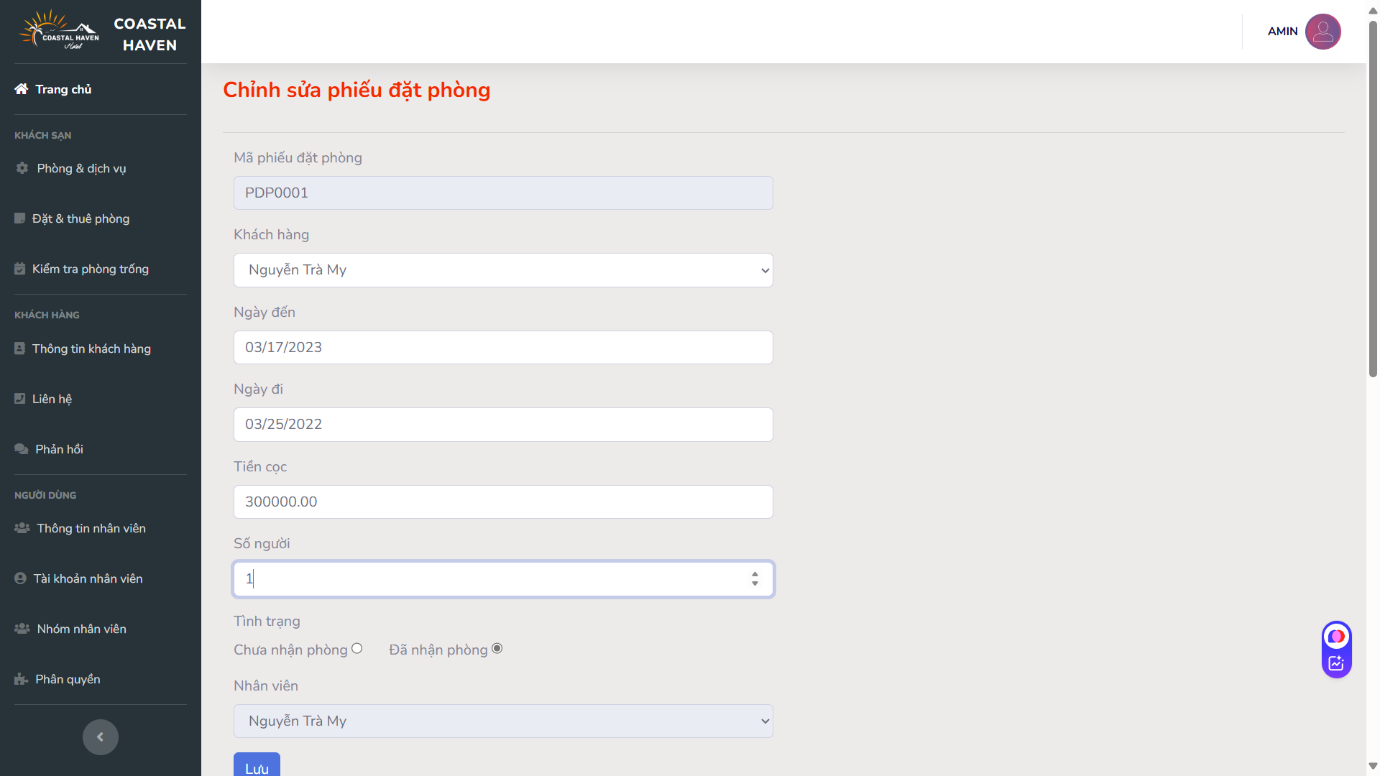
**Chức năng xem chi tiết thông:** Khi chọn vào xem chi tiết sẽ hiển thị các thông tin về: mã phiếu đặt phòng, loại phòng thông tin khách hàng, thời gian và giá cả. Tại đây, nhân viên có thể chỉnh sửa hoặc xóa dữ liệu trực tiếp khi cần thiết.

A screenshot of a computer

Description automatically generated

Hình . Giao diện xem chi tiết thông tin đặt phòng

**Chức năng sửa:** Khi chọn vào sửa, cho phép chỉnh sửa và cập nhật các dữ liệu đã nhập trước đó với giao diện thân thiện.



Hình . Giao diện chỉnh sửa phiếu đặt phòng

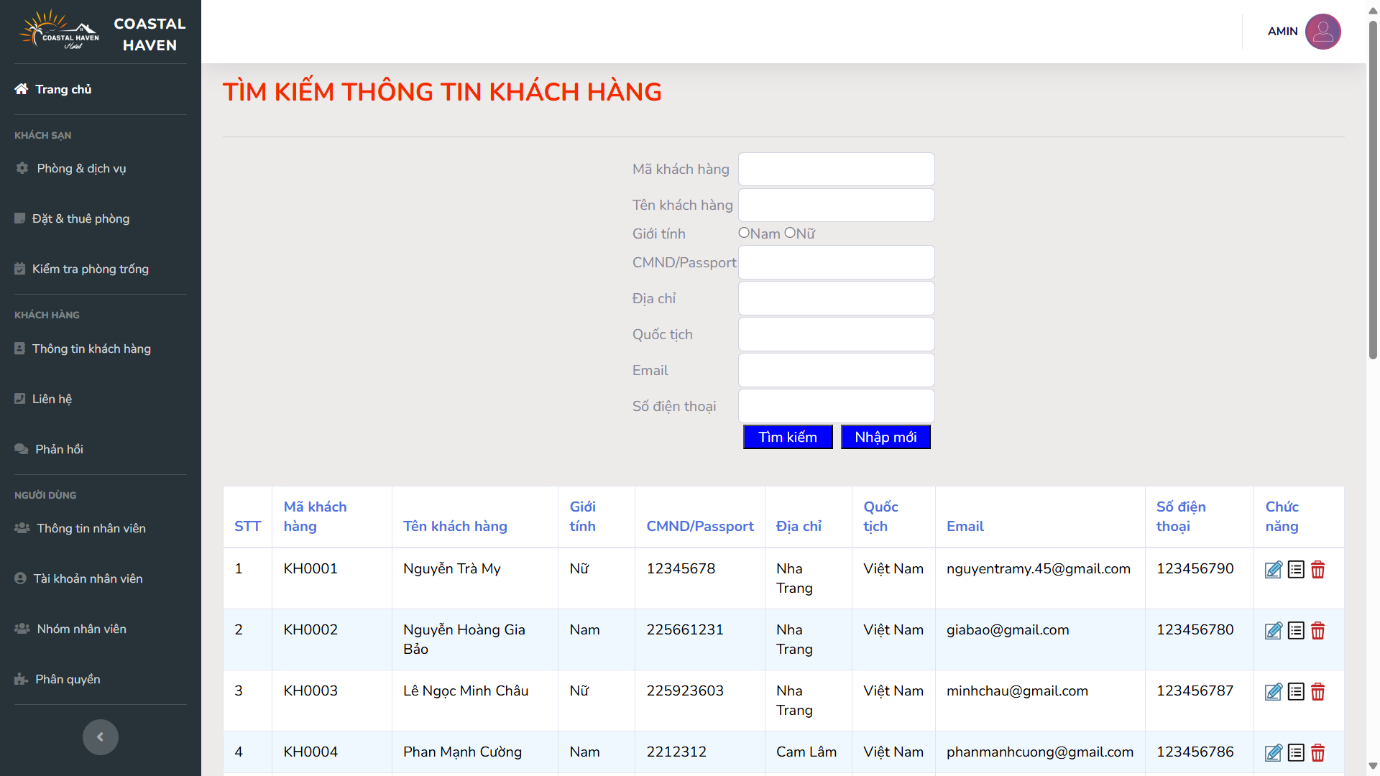
**Chức năng xóa:** Khi chọn vào xóa, hiển thị đầy đủ thông tin của mục cần xóa và cung cấp xác nhận trước khi thực hiện thao tác xóa để tránh nhầm lẫn. Sau khi hoàn tất, hệ thống sẽ hiển thị thông báo thành công.

A screenshot of a computer

Description automatically generated

Hình . Giao diện xóa phiếu đặt phòng

**Chức năng tìm kiếm:** Cho phép nhập từ khóa liên quan như tên hoặc mã khách hàng để nhanh chóng tìm kiếm thông tin. Kết quả tìm kiếm sẽ hiển thị danh sách khách hàng phù hợp kèm theo các thông tin liên quan.



Hình . Giao diện tìm kiếm thông tin khách hàng

# CHƯƠNG 5. KẾT LUẬN

## 5.1 Ưu điểm của ứng dụng

Ứng dụng quản lý khách sạn được thiết kế nhằm mang đến trải nghiệm thân thiện và dễ sử dụng cho người dùng. Với giao diện trực quan, các chức năng được bố trí hợp lý, giúp người dùng nhanh chóng nắm bắt và vận hành hệ thống hiệu quả.

Tính năng bảo mật của ứng dụng được chú trọng thông qua việc phân quyền người dùng chi tiết và cơ chế tự động đăng xuất khi không hoạt động trong thời gian dài, đảm bảo an toàn dữ liệu.

Trang chủ hiển thị đầy đủ thông tin về khách sạn, bao gồm loại phòng, giá cả, dịch vụ và hình ảnh minh họa, giúp khách hàng dễ dàng đưa ra lựa chọn. Điều này giúp khách hàng dễ dàng tiếp cận và nắm bắt thông tin quan trọng một cách thuận tiện.

Các tính năng như đặt phòng, tìm kiếm, quản lý khách hàng và hóa đơn được tối ưu hóa để giúp nhân viên vận hành dễ dàng hơn, tiết kiệm thời gian và chi phí. Đặc biệt, khả năng cập nhật và chỉnh sửa thông tin nhanh chóng giúp tăng tính linh hoạt khi xử lý yêu cầu của khách hàng.

## 5.2 Nhược điểm của ứng dụng

Mặc dù ứng dụng đã đạt được nhiều ưu điểm, nhưng vẫn còn một số hạn chế cần khắc phục. Do không thực hiện khảo sát thực tế nên dẫn đến một số chức năng có thể chưa hoàn toàn phù hợp với nhu cầu thực tế.

Hệ thống hiện tại chỉ phục vụ cho các khách sạn nhỏ và vừa, do đó chưa đáp ứng tốt các yêu cầu phức tạp của những khách sạn lớn với nhiều chi nhánh. Ngoài ra, tính năng thông báo tức thời khi có yêu cầu mới từ khách hàng chưa được tích hợp, làm giảm tốc độ phản hồi và hỗ trợ.

Một số chức năng như khuyến mãi hoặc cổng thanh toán trực tuyến vẫn chưa được phát triển, khiến hệ thống chưa thể đáp ứng đầy đủ các nhu cầu hiện đại của ngành khách sạn.

## 5.3 Đề xuất phát triển

Để nâng cao tính năng của ứng dụng, cần mở rộng các chức năng như thanh toán trực tuyến, xuất báo cáo chi tiết, và quản lý vật tư, trang thiết bị trong khách sạn. Và bổ sung chế độ cập nhật thông tin theo thời gian thực để giúp nhân viên và khách hàng tương tác nhanh hơn, nâng cao trải nghiệm người dùng.

Ngoài ra, nên phát triển thêm tính năng chat trực tuyến giữa nhân viên và khách hàng để hỗ trợ nhanh chóng các yêu cầu đặc biệt. Bên cạnh đó, việc phát triển một Chatbot thông minh để trả lời các câu hỏi phổ biến sẽ giúp giảm tải công việc cho nhân viên và tạo cảm giác chuyên nghiệp cho dịch vụ.

Cuối cùng, hệ thống cần tăng cường bảo mật dữ liệu và tối ưu hóa tốc độ xử lý để đáp ứng tốt hơn cho các khách sạn có quy mô lớn. Các tính năng đề xuất này sẽ giúp ứng dụng trở thành một giải pháp toàn diện và hiệu quả hơn trong quản lý khách sạn.

# TÀI LIỆU THAM KHẢO

**[1]** Bùi Chí Thành (2024). Bài giảng Phát triển ứng dụng Web. Bộ môn

Kỹ thuật phần mềm, Đại học Nha Trang, Khoa Công nghệ thông tin.

**[2]** Bạch Ngọc Toàn, Khóa học lập trình ASP.NET MVC, <https://tedu.com.vn/khoa-hoc-mien-phi/khoa-hoc-lap-trinh-aspnet-mvc-25.html>

**[3]** ColorLib, https://colorlib.com/wp/html5-and-css3-login-forms/

**[4]** W3.CSS, <https://www.w3schools.com/w3css/default.asp>