|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **KHOA CÔNG NGHỆ THÔNG TIN**  **======\*\*\*======**  **E:\dh-cn-hanoi.png**  **BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**  **NGÀNH: HỆ THỐNG THÔNG TIN**  **ĐỀ TÀI: XÂY DỰNG HỆ THỐNG ĐẶT PHÒNG KHÁCH SẠN SỬ DỤNG STRING BOOT VÀ REACTJS**   |  |  | | --- | --- | | Cán bộ hướng dẫn | **: ThS. Nguyễn Lan Anh** | | Sinh viên thực hiện | **: Nguyễn Thanh Hương** | | Mã số sinh viên  Lớp | **: 2019605677**  **: HTTT02-K14** |   **Hà Nội – Năm 2023** |

# DANH MỤC BẢNG BIỂU

[Bảng 2.1. Các use case hệ thống 11](#_Toc102943702)

# DANH MỤC HÌNH VẼ

# MỤC LỤC

[DANH MỤC BẢNG BIỂU i](#_Toc130883535)

[DANH MỤC HÌNH VẼ ii](#_Toc130883536)

[MỤC LỤC iii](#_Toc130883537)

[MỞ ĐẦU 1](#_Toc130883538)

[CHƯƠNG 1. TỔNG QUAN VỀ CÔNG NGHỆ VÀ CƠ SỞ LÝ THUYẾT 3](#_Toc130883539)

[1.1. Spring Boot 3](#_Toc130883540)

[1.1.1. Giới thiệu về Spring Boot 3](#_Toc130883541)

[1.1.2. Lịch sử phát triển của Spring Boot 4](#_Toc130883542)

[1.1.3. Ưu điểm của Spring Boot 4](#_Toc130883543)

[1.2. MySQL 5](#_Toc130883544)

[1.2.1. Giới thiệu về MySQL 5](#_Toc130883545)

[1.2.2. Lịch sử phát triển của MySQL 5](#_Toc130883546)

[1.2.3. Ưu điểm của MySQL 6](#_Toc130883547)

[1.3. ReactJS 6](#_Toc130883548)

[1.3.1. Giới thiệu ReactJs 6](#_Toc130883549)

[1.3.2. Lịch sử phát triển 7](#_Toc130883550)

[1.3.3. Ưu điểm của ReactJS 7](#_Toc130883551)

[1.4. JDK (Java Development Kit) 8](#_Toc130883552)

[1.4.1. Giới thiệu về JDK 8](#_Toc130883553)

[1.4.2. Phiên bản sử dụng: JDK 11 9](#_Toc130883554)

[1.5. Restful API 9](#_Toc130883555)

[1.5.1. Khái niệm 9](#_Toc130883556)

[1.5.2. Các thành phần của Restfull API 10](#_Toc130883557)

[1.6. Một số thư viện nổi bật sử dụng trong sản phẩm 10](#_Toc130883558)

[CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG VÀ THIẾT KẾ CƠ SỞ DỮ LIỆU 12](#_Toc130883559)

[2.1. Xác định các yêu cầu chính của hệ thống 12](#_Toc130883560)

[2.1.1. Phân tích các yêu cầu 12](#_Toc130883561)

[2.1.2. Các yêu cầu phi chức năng 13](#_Toc130883562)

[2.2. Xây dựng biểu đồ use case 13](#_Toc130883563)

[2.2.1. Các tác nhân 13](#_Toc130883564)

[2.2.2. Các use case của hệ thống 14](#_Toc130883565)

[2.2.3. Biểu đồ use case 15](#_Toc130883566)

[2.2.4. Mô tả chi tiết các use case 17](#_Toc130883567)

[2.2.5. Phân tích các use case 17](#_Toc130883568)

[2.3. Xây dựng cơ cở dữ liệu 19](#_Toc130883569)

[2.3.1. Biểu đồ Entity Relationship Diagram 19](#_Toc130883570)

[2.3.2. Các bảng trong cơ sở dữ liệu 19](#_Toc130883571)

[2.4. Thiết kế giao diện 25](#_Toc130883572)

[CHƯƠNG 3. KẾT QUẢ ĐẠT ĐƯỢC 26](#_Toc130883573)

[3.1. Giao diện chương trình 26](#_Toc130883574)

[3.1.1. Giao diện phía khách hàng 26](#_Toc130883575)

[3.1.2. Màn hình trang quản lý 26](#_Toc130883576)

[KẾT LUẬN 27](#_Toc130883577)

[TÀI LIỆU THAM KHẢO 28](#_Toc130883578)

# MỞ ĐẦU

1. **Lý do chọn đề tài**

Hiện nay trong khi đời sống con người nâng cao, nhu cầu du lịch ngày càng phát triển thì nhu cầu tìm những khách sạn có phòng nghỉ chất lượng và dịch vụ tốt để nghỉ ngơi thư giãn ngày càng phổ biến. Bằng cách truy cập vào các website tìm kiếm thông tin phòng tại khách sạn giúp khách hàng chọn được phòng theo ý thích của mình.

Trước kia khi chưa ứng dụng tin học vào công tác đặt phòng thì các công việc như cập nhật thông tin khách hàng, xem các thông tin về dịch vụ của khách sạn, thông tin về số phòng, hoá đơn thanh toán,thông tin phòng, dịch vụ đi kèm… tất cả đều làm thủ công nên người quản trị phải mất rất nhiều thời gian mới tổng hợp được những thông tin cần thiết để truyền đạt đến khách hàng.

Mặt khác hiện nay trong các khách sạn, người quản trị xử lý thông tin của khách sạn phải quản lý một lượng lớn thông tin về thông tin phòng,dịch vụ, tình trạng phòng… Việc này mất rất nhiều thời gian và công sức của con người nên việc ứng dụng tin học vào công tác xây dựng hệ thống đặt phòng khách sạn theo yêu cầu của người dùng ngày càng trở nên hết sức cần thiết, quan trọng.

Với mong muốn củng cố và nâng cao kiến thức về lập trình website, áp dụng kiến thức để xây dựng một hệ thống giúp người dùng có thể dễ dàng sử dụng, đặt phòng , tham khảo các loại phòng khách sạn với mức giá hợp lý , sử dụng hệ thống với công nghệ mới nhanh và mượt mà hơn em đã chọn đề tài “Xây dựng hệ thống đặt phòng khách sạn sử dụng Spring Boot và ReactJS”.

Quá trình thực hiện đề tài của em có thể còn nhiều thiếu xót. Em rất mong nhận được sự cảm thông và chia sẻ của quý thầy cô để có thể đưa đề tài phát triển thêm mạnh mẽ hơn nữa. Em xin chân thành cảm ơn!

1. **Mục đích của đề tài**

Đồ án Xây dựng hệ thống đặt phòng khách sạn với mục đích cung cấp cho khách hàng các thông tin chính xác về các phòng tại khách sạn đang có và cách thức đặt phòng qua mạng. Các thông tin được cập nhật thường xuyên và nhanh chóng. Vì vậy, rút ngắn được khoảng cách giữa khách hàng và khách sạn cung cấp dịch vụ, đưa thông tin về các phòng tại các khách sạn nhanh chóng đến cho khách hàng.

Xây dựng hệ thống website hoàn thiện gồm các module khách hàng tham gia đặt phòng và các module quản trị hệ thống. Rèn luyện các kiến thức và kỹ năng phát triển hệ thống phần mềm và thái độ làm việc chuyên nghiệp.

1. **Nội dung nghiên cứu**

* Khảo sát và phân tích yêu cầu quy trình quản lý đặt phòng khách sạn.
* Phân tích thiết kế hệ thống.
* Thiết kế cơ sở dữ liệu.
* Lập trình backend API.
* Lập trình giao diện và kết nối tới API.
* Có kế hoạch và đánh giá kiểm thử hệ thống.

1. **Đối tượng và phạm vi nghiên cứu**

* Chatbot tự động tư vấn cho khách hàng .
* Gửi email cảm ơn tự động sau khi thanh toán xong.

1. **Ý nghĩa khoa học và thực tiễn của đề tài**
2. **Bố cục đề tài**

Ngoài các phần Mở đầu, Kết luận và Tài liệu tham khảo, báo cáo đồ án được bố cục thành ba chương chính sau.

* Chương 1: Trình bày tổng quan về công nghệ và cơ sở lý thuyết.
* Chương 2: Phân tích thiết kế hệ thống và thiết kế cơ sở dữ liệu.
* Chương 3: Cài đặt phần mềm và kết quả đạt được.

# TỔNG QUAN VỀ CÔNG NGHỆ VÀ CƠ SỞ LÝ THUYẾT

## Spring Boot

### Giới thiệu về Spring Boot

Spring Boot là một extension của Spring Framework giúp các lập trình viên loại bỏ các bước cấu hình phức tạp mà Spring bắt buộc. Spring Boot là dự án phát triển bởi ngôn ngữ Java (JAV) trong hệ sinh thái Spring framework..

* Đặc điểm nổi bật của Spring Boot:
* Được phát triển tối ưu sao cho cấu hình XML trở nên đơn giản nhanh chóng và dễ dàng nhất trong Spring.
* Gia tăng được năng suất trong quá trình lập trình
* Giảm ở mức tối thiểu thời gian lập trình
* Giúp người dùng mặc dù không có nhiều kiến thức lập trình vẫn có thể xây dựng được một ứng dụng đơn giản.
* Một số tính năng quan trọng của Spring Boot:
* SpringApplication: Khi lập trình xong và bạn chỉ muốn chạy thử nghiệm thì nên làm thế nào? Bạn chỉ cần gọi run() là được, vì Spring Boot được thiết kế theo dạng “just run”. Giúp cho các lập trình viên chỉ cần cấu hình ít Spring nhất, phần còn lại Spring Boot sẽ lo liệu.
* Externalized Configuration: Spring Boot giúp config cấu hình từ ngoài và ứng dụng của bạn sẽ có thể chạy thoải mái.
* Profiles: Nếu có nhiều config khác nhau, ta có thể sử dụng Profile để phân chia từng loại cho từng môi trường để dễ dàng quản lý hơn.
* Logging: Tính năng này được sử dụng cho toàn bộ chức năng log trong phạm vi nội bộ và nó được quản lý mặc định.
* Bên cạnh những tính năng quan trọng nói trên, còn một số tính năng tương tự khác như: Security, Messaging, Developing web Applications, Working with SQL Technologies, Caching, Sending Email, Validation, Calling rest Services with RestTemplate/WebClient,…

### Lịch sử phát triển của Spring Boot

Ở thời điểm hiện tại Spring Boot có lẽ đang là framework nổi tiếng và được sử dụng nhiều bậc nhất trong phát triển các ứng dụng sử dụng ngôn ngữ Java.

Được xây dựng trên nền tảng Spring framework, nó có tất cả các tính năng của Spring cộng thêm những tiện ích mà nó mang lại như giảm thiểu các bước cấu hình phức tạp, nhúng server container (Tomcat, Jetty hoặc Undertow) tự động vào ứng dụng giúp chúng ta có thể khởi chạy một ứng dụng ngay lập tức, quản lý dependence thông minh,…

### Ưu điểm của Spring Boot

* Phát triển các ứng dụng dựa trên Spring một cách tiết kiệm thời gian và dễ dàng.
* Tự động cấu hình tất cả các components cho một ứng dụng Spring cấp sản xuất.
* Các máy chủ nhúng được tạo sẵn (Tomcat, Jetty và Undertow), dẫn đến việc triển khai ứng dụng được tăng tốc và hiệu quả hơn.
* Điểm cuối HTTP, cho phép nhập các tính năng bên trong ứng dụng như chỉ số, tình trạng sức khỏe, v.v.
* Không có cấu hình XML.
* Nhiều lựa chọn bổ sung, hỗ trợ nhà phát triển làm việc với cơ sở dữ liệu được nhúng và trong bộ nhớ.
* Dễ dàng truy cập cơ sở dữ liệu và các dịch vụ hàng đợi như MySQL, Oracle, MongoDB, Redis, ActiveMQ và các dịch vụ khác.
* Tích hợp trơn tru với hệ sinh thái Spring.
* Cộng đồng lớn và rất nhiều hướng dẫn, tạo điều kiện cho giai đoạn làm quen.
* Giảm thiểu thời gian phát triển code, tăng hiệu suất phát triển chung của cả dự án.
* Dễ dàng tích hợp các mô-đun liên quan như Sping-MVC, Spring Data, Spring Sercurity, Spring Cloud,v.v…
* Cung cấp công cụ CLI(Command Line Interface) cho việc phát triển và test ứng dụng nhanh chóng từ command line
* Ngoài ra còn có nhiều plugins để phát triển nhanh chóng bằng các công cụ như Build như Maven hoặc Gradle.

## MySQL

### Giới thiệu về MySQL

MySQL là hệ quản trị cơ sở dữ liệu mã nguồn mở miễn phí nằm trong nhóm LAMP (Linux – Apache -MySQL – PHP) >< Microsoft (Windows, IIS, SQL Server, ASP/ASP.NET),vì MySQL được tích hợp sử dụng chung với apache, PHP nên nó phổ biến nhất thế giới. Vì MySQL ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh và Mysql cũng có cùng một cách truy xuất và mã lệnh tương tự với ngôn ngữ SQL chính vì thế nên MySQL được sử dụng và hỗ trợ của những lập trình viên yêu thích mã nguồn mở.

### Lịch sử phát triển của MySQL

MySQL là một sản phẩm cơ sở dữ liệu nguồn mở được tạo bởi MySQL AB, một công ty được thành lập năm 1995 tại Thụy Điển. Năm 2008, Sun microsystems đã mua lại MySQL AB với giá xấp xỉ 1 tỷ đô la.

* Dự án của MySQL được bắt đầu vào năm 1979, khi nhà phát minh của MySQL, Michael Widenius phát triển một công cụ cơ sở dữ liệu nội bộ có tên UNIREG để quản lý cơ sở dữ liệu.
* Sau đó, UNIREG đã được viết lại bằng nhiều ngôn ngữ khác nhau và được mở rộng để xử lý các cơ sở dữ liệu lớn.
* Sau một thời gian Michael Widenius đã liên lạc với David Hughes, tác giả của mQuery, để xem liệu Hughes có quan tâm đến việc kết nối mQuery với trình xử lý B + ISAM của UNIREG để cung cấp lập chỉ mục cho mQuery hay không. Đó là cách MySQL ra đời.

MySQL được đặt theo tên con gái của Michael Widenius có tên là "My".

### Ưu điểm của MySQL

* Linh hoạt và dễ dùng: Quá trình cài đặt tương đối đơn giản và không mất quá 30 phút và bạn có thể dễ dàng chỉnh sửa source code mà không phải thanh toán thêm tiền.
* Hiệu năng cao: Dù dữ liệu của bạn lớn như thế nào thì MySQL cũng đáp ứng với tốc độ cao, mượt mà kể cả big data của các trang thương mại điện tử hoặc những hoạt động kinh doanh nặng nề liên quan đến công nghệ thông tin.
* Tiêu chuẩn trong ngành: Bất cứ ai đã dấn thân vào ngành công nghệ và dữ liệu thì đều đã sử dụng MySQL và người dùng cũng có thể triển khai dự án nhanh và thuê các chuyên gia dữ liệu.
* An toàn: Vấn đề an toàn luôn là vấn đề cực kì quan trọng trong ngành dữ liệu và MySQL đảm bảo được tiêu chuẩn bảo mật rất cao.

## ReactJS

### Giới thiệu ReactJs

ReactJS là thư viện mã nguồn mở của Facebook được viết bằng JavaScript với mục đích xây dựng giao diện người dùng.

Reactjs là một thư viện Javascript mã nguồn mở hỗ trợ xây dựng các thành phần giao diện nhanh gọn và tiện lợi. Bình thường các lập trình viên sẽ nhúng javascript vào code HTML thông qua các attribute như AngularJS nhưng với Reactjs làm việc như một thư viện cho phép nhúng HTML vào javascript thông qua JSX

Trong ReactJS, giao diện được chia thành các phần nhỏ gọi là component và có thể dễ dàng tái sử dụng.

ReactJS giúp giao diện của ứng dụng web có thể phản ứng nhanh nhẹn và mượt mà hơn.

### Lịch sử phát triển

ReactJS được phát triển bởi Jordan Walke – một kỹ sư phần mềm tại Facebook vào năm 2012. Lúc đầu Facebook chỉ áp dụng vào các sản phẩm của họ nhưng sau đó không lâu, vào tháng 5 năm 2013 ReactJS đã được mã nguồn mở tại JSConf US.

### Ưu điểm của ReactJS

* Tốc độ phát triển ứng dụng nhanh.
* Phù hợp với đa dạng các thể loại website: ReactJS khiến cho việc khởi tạo website dễ dàng hơn bởi vì không cần phải code nhiều như khi tạo trang web thuần chỉ dùng JavaScript, HTML và nó đã cung cấp đủ loại công cụ để có thể dùng cho nhiều trường hợp.
* Dễ dàng bảo trì và mở rộng.
* Tái sử dụng các Component: Nếu xây dựng các Component đủ tốt, đủ flexible để có thể thoả các “yêu cầu” của nhiều dự án khác nhau, ta chỉ tốn thời gian xây dựng ban đầu và sử dụng lại hầu như toàn bộ ở các dự án sau.
* ReactJS được thiết kế cho các ứng dụng yêu cầu hiệu suất cao nhờ việc cung cấp DOM ảo và kết xuất phía máy chủ.
* Dễ dàng sử dụng cho người có kiến thức cơ bản về JavaScript.
* Bên cạnh đó ReactNative là một phiên bản khác của ReactJS giúp xây dựng ứng dụng trên cả nền tảng Android và iOS.
* Có thể sử dụng cho cả Mobile application.
* Thân thiện với SEO.
* Debug dễ dàng
* Công cụ phát triển web hot nhất hiện nay.

## JDK (Java Development Kit)

### Giới thiệu về JDK

Java Development Kit (JDK) là một trong ba gói công nghệ cốt lõi được sử dụng trong lập trình Java, cùng với JVM (Máy ảo Java - Java Virtual Machine) và JRE (Java Runtime Environment - Môi trường Java Runtime). Việc phân biệt giữa ba công nghệ này, cũng như hiểu được cách chúng kết nối với nhau là rất quan trọng.

JVM là thành phần Java nền tảng để chạy các chương trình.

JRE là phần on-disk của Java, phần tạo ra JVM.

JDK cho phép các developer tạo các chương trình Java, trong đó các chương trình có thể được JVM và JRE xử lý và chạy.

Các developer khi mới sử dụng Java thường dễ nhầm lẫn Java Development Kit với Java Runtime Environment. Để phân biệt, ta cần nhớ rằng JDK là một gói các công cụ phát triển phần mềm dựa trên Java, trong khi JRE là gói công cụ sử dụng để chạy Java code.

JRE có thể được sử dụng như một thành phần độc lập để chạy các chương trình Java, dù vậy, JRE vẫn là một phần của JDK. JDK đòi hỏi bộ công cụ JRE vì chạy các chương trình Java là một phần của việc phát triển chúng.

Chia sẻ JDK theo 2 cách như sau:

Định nghĩa chuyên ngành: JDK là một hệ tiêu chuẩn trong việc triển khai nền tảng Java, bao gồm các trình thông dịch dịch và thư viện lớp.

Định nghĩa thông thường: JDK là gói phần mềm bạn tải xuống để tạo các ứng dụng dựa trên Java.

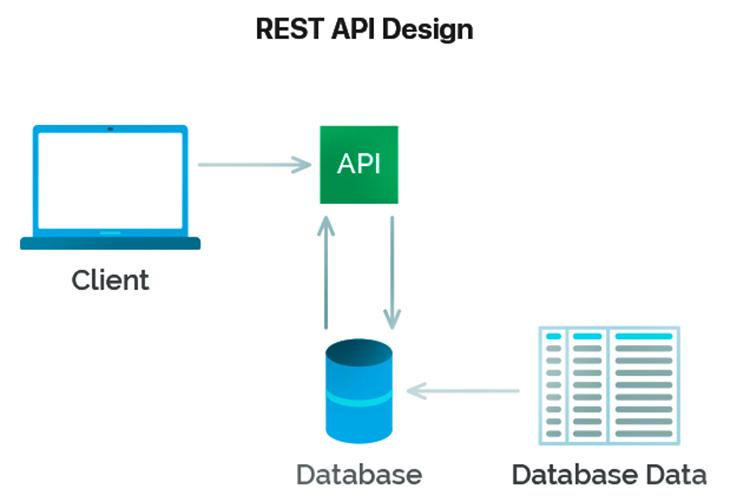
### Phiên bản sử dụng: JDK 11

Khái niệm Long Term Support (LTS) có lẽ cũng đã không còn xa lạ với các lập trình viên trên các nền tảng khác ví dụ như NodeJS. Nhưng trong thế giới của Java thì đó mới là khái niệm đáng để quan tâm từ Java 10. Câu chuyện bắt đầu từ Java 10 khi Oracle thay đổi cách để release các phiên bản Java không phải dựa theo các tính năng (feature-based release) nữa mà dựa theo khoảng thời gian (Time-based release). Để đáp ứng cho việc triển khai ứng dụng Java lên môi trường thật, chúng ta cần một phiên bản Java thật hoàn chỉnh và đã được kiểm duyệt cẩn thận, và đó chính là các phiên bản Java được hỗ trợ dài hạn (Long Term Support). JDK 11 là một trong số những bản Java như thế và chúng ta có thể tự tin triển khai Java 11 lên trên hệ thống ở môi trường thật của chúng ta với một sự yên tâm về tỉ lệ rủi ro xuất hiện các bug là thấp. Xin nói thêm là chu kì ra đời cho một phiên bản Java Long Term Support sẽ tính theo đơn vị năm.

## Restful API

### Khái niệm

Rest API là một ứng dụng co phép chuyển đổi cấu trúc dữ liệu có những phương thức kết nối với các thư viện và ứng dụng khác. Đây là một giải pháp giúp xây dựng lên những ứng dụng web chuyên dụng thay thế cho những giải pháp cũ.



Hình 1.1. Rest API

Restful API là một trong những tiêu chuẩn được sử dụng để thiết kế API để tiện hơn cho việc quản lý các resource. Dữ liệu được truyền tải thông qua HTTP.

### Các thành phần của Restfull API

* API (Application Programming Interface) là giao diện lập trình ứng dụng giúp đưa ra các phương thức kết nối với các thư viện và ứng dụng khác.
* REST (Reprosemtational State Transfer) là một đại diện cho sự chuyển đổi dữ liệu. Nghĩa là client và server hoàn toàn độc lập, các yêu cầu gửi tới Rest API đều không mang theo trạng thái của yêu cầu trước đó.

## Một số thư viện nổi bật sử dụng trong sản phẩm

* Reacttrap: Thư viện UI cho React được xây dựng dựa trên bootstrap (Áp dụng cho trang khách).
* Spring: là một hệ sinh thái, trong đó có nhiều các framework và thư viện khác : Spring Data Jpa để tương tác với Database, Spring Security để xác thực và phân quyền ứng dụng, Mail Sender của Spring để gửi mail, Spring AOP lập trình hướng khía cạnh để xử lý các Exception.

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG VÀ THIẾT KẾ CƠ SỞ DỮ LIỆU

## Xác định các yêu cầu chính của hệ thống

### Phân tích các yêu cầu

**Những quyền lợi đối với khách hàng**

* Xem trang tĩnh: Khách hàng có quyền xem nội dung trang tĩnh trên website.
* Xem phòng còn trống : Khách hàng có quyền xem danh sách các phòng còn trống .
* Đăng ký: Khách hàng có thể đăng kí tài khoản người dùng ở trên hệ thống .
* Đăng nhập: Khách hàng có thể đăng nhập vào hệ thống sau khi đã đăng ký thành công.
* Cập nhật thông tin cá nhân: Khách hàng có quyền thay đổi thông tin cá nhân.
* Đặt phòng: Khách hàng có quyền xem thông tin phòng và tiến hành đặt phòng nếu phòng còn trống.
* Hủy phòng : Khách hàng có quyền hủy phòng trước một ngày check-in và cần có lý do hủy phòng
* Xem thông tin khách sạn:Khách hàng có quyền xem thông tin khách sạn
* Xem thông tin khuyến mãi: Khách hàng có quyền xem thông tin khuyến mãi của phòng.
* Sử dụng chatbot tự động:Khách hàng có thể sử dụng chatbot để nhận tư vấn tự động từ hệ thống
* Đánh giá : Khách hàng có thể đánh giá về chất lượng sử dụng sau khi đặt phòng tại hệ thống
* Đặt dịch vụ: Khách hàng có thể đặt dịch vụ đi kèm khi đặt phòng (ở mỗi dịch vụ sẽ đi kèm các sản phẩm mặc định sẵn)

**Những quyền lợi đối với quản trị viên**

* Đăng nhập: Quản trị viên cần đăng nhập vào trang quản lý để thực hiện các chứ năng quản trị hệ thống.
* Thay đổi thông tin cá nhân: Quản trị viên có thể thay đổi thông tin cá nhân.
* Bảo trì bài viết : xem, thêm sửa, xoá bài viết trên hệ thống.
* Bảo trì phòng : xem, thêm, sửa, xoá phòng trên hệ thống.
* Bảo trì dịch vụ: xem, thêm sửa xoá dịch vụ trên hệ thống.
* Thống kê đặt phòng: Quản trị có thể thống kê số lượng đặt phòng
* Quản lý người dùng/khách hàng: Quản trị có thể quản lý người dùng/khách hàng đã có tài khoản trên hệ thống đặt phòng.
* Thống kê khách đặt phòng nhiều lần nhất (top 10)
* Cài đặt phòng và giá: Quản trị có thể cài đặt lại các phòng và giá của các phòng.
* Cài đặt dịch vụ của phòng
* Cài đặt dịch vụ đi kèm
* Tính tiền phòng đặt
* Gửi email cảm ơn khách (tự động sau khi thanh toán )

### Các yêu cầu phi chức năng

* Hiệu năng cao, trải nghiệm mượt mà, ổn định.
* Giao diện thân thiện,dễ sử dụng.
* Đáp ứng được lượng lớn truy cập đồng thời.
* Tối ưu hoá SEO.
* Dễ dàng bảo trì và mở rộng.

### Mô tả nghiệp vụ bài toán

* **Mô tả nghiệp vụ đặt phòng/ đặt dịch vụ (Khi đặt phòng sẽ tiếp tục đặt dịch vụ luôn)**
* Khách hàng sẽ chọn phòng cần đặt, chọn thời gian nhận phòng và trả phòng. Tiếp theo khách hàng có thể chọn các dịch vụ đi kèm như đồ ăn, đồ uống, bơi…Sau khi khách hàng xác nhận đặt phòng. Booking sẽ chuyển status sang trạng thái Pending (Ở trạng thái Pending khách hàng khác vẫn có thể booking phòng này nhưng chỉ có thể booking với những thời điểm khác với những khách hàng đã booking phòng này).
* Khi khách hàng đến nhận phòng và check in thành công, thì status sẽ chuyển sang Checked-In, và những người khác sẽ không thể booking được phòng này được nửa cũng như k thể tìm thấy được phòng này.

* **Mô tả nghiệp vụ hủy phòng**
* Khi khách hàng muốn hủy phòng thì sẽ cần phải có lý do hủy, sẽ có 1 list các option cho khách hàng chọn. Và sau đó sẽ chuyển trạng thái booking sang Cancel (Hủy đặt) . Khách hàng chỉ được phép hủy đặt phòng trước ngày nhận phòng lớn hơn tối thiểu là 1 ngày. Nếu không đạt yêu cầu hủy phòng thì sẽ phải thanh toán toàn bộ số tiền.

* **Mô tả nghiệp vụ tính tiền - trả phòng**
* Phòng được tính tiền theo ngày, được tính bắt đầu từ 14h chiều ngày hôm nay đến 12h trưa ngày hôm sau. Nếu khách hàng trả phòng sau 14h (ví dụ 15h, 16h) thì vẫn tính là 1 ngày.
* Khi trả phòng, nhân viên sẽ xác nhận check out, và hệ thống sẽ tổng số tiền cần thanh toán (bao gồm tiền phòng, tiền dịch vụ và các phí phát sinh khác). Tính tiền sẽ dựa trên thời gian của Booking, giá phòng của bảng BookingRoomDetail và tổng tiền dịch vụ của bảng BookingServiceDetail.
  + Sẽ lấy checkout - checkin = T
  + Thời gian \* tiền phòng = tổng tiền phòng
  + tổng tiền phòng + tổng tiền các dịch vụ = tổng tiền cần thanh toán
* Khi thực hiện tính tiền và in bill thành công => chuyển status booking sang trạng thái Checked-Out

* Phí phát sinh:
* Thời gian check in sẽ là >= 14h, check out là <= 12h trưa ngày hôm sau. Vào thời điểm đó sẽ được tính là 1 ngày tiền phòng.
* Nếu check in trước 14h và check out sau 12h thì sẽ phải chịu thêm phí
* Cụ thể:
* Đối với trường hợp nhận phòng sớm
* Check in từ 5h-9h: Tính thêm 50% giá mỗi phòng
* Check in từ 9h-14h: Tính thêm 30% giá mỗi phòng
* Đối với trường hợp trả phòng muộn
* Check out từ 12h-15h: Tính theo 30% giá mỗi phòng
* Check out từ 15h-18h: Tính thêm 50% giá mỗi phòng
* Sau 18h: Tính thêm 100% giá mỗi phòng

## Xây dựng biểu đồ use case

### Các tác nhân

* Người quản trị: Quản lý nội dung website, xử lý các yêu cầu từ khách hàng.
* Khách hàng: Đăng nhập, xem thông tin phòng, đặt phòng…

### Các use case của hệ thống

Bảng 2.1. Các use case hệ thống

|  |  |
| --- | --- |
| **Tác nhân** | **Các ca sử dụng** |
| Khách hàng | Đăng nhập  Đăng ký  Cập nhật thông tin  Xem phòng còn trống  Đặt phòng  Hủy phòng  Xem thông tin khách sạn  Xem thông tin khuyến mãi  Sử dụng chatbot tự động  Đặt dịch vụ |
| Người quản trị | Đăng nhập Admin  Thay đổi thông tin cá nhân  Bảo trì bài viết  Bảo trì phòng  Bảo trì dịch vụ  Bảo trì sản phẩm  Bảo trì khuyến mãi  Thống kê đặt phòng  Quản lý người dùng/khách hàng  Thống kê top 10 khách hàng đặt phòng nhiều nhất  Tính tiền phòng đặt  Gửi mail cảm ơn tự động (sau khi thanh toán) |

### Biểu đồ use case



Hình 2.1. Các use case chính



Hình 2.2. Các use case thứ cấp

### Mô tả chi tiết các use case

#### Use case Đăng nhập

* **Mô tả vắn tắt**: Use case này cho phép khách hàng đăng nhập vào hệ thống để đặt phòng.
* **Luồng sự kiện**:
  + **Luồng cơ bản**:

1. Use ase này bắt đầu khi khách hàng truy cập vào trang chủ của hệ thống. Hệ thống sẽ hiện màn hình đăng nhập ,khách hàng tiến hành điền tên đăng nhập (email/số điện thoại) và mật khẩu .
2. Khách hàng nhập thông tin đăng nhập và click vào “Submit”.Hệ thống sẽ kiểm tra thông tin tài khoản của khách hàng đã đăng ký bao gồm tên đăng nhập:số điện thoại hoặc email và password của khách hàng từ bảng USER trong CSDL và cho phép khách hàng đăng nhập vào giao diện chính của hệ thống.

Use case kết thúc.

* **Luồng rẽ nhánh**

1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
2. Tại bước 2 trong luồng cơ bản, khi người dùng nhập thiếu thông tin hoặc thông tin không hợp lệ hoặc không tìm thấy bản ghi trong bảng USER. Hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại. Người dùng có thể nhập lại để tiếp tục hoặc kết thúc thao tác.

* **Các yêu cầu đặc biệt**: Khách hàng cần nhập đúng thông tin để yêu cầu đăng nhập.
* **Các yêu cầu đặc biệt:**

Không có.

* **Tiền điều kiện:**

Khách hàng cần đăng ký tài khoản .

* **Hậu điều kiện:**

Không có.

* **Điểm mở rộng:**

Không có.

#### Use case Đăng ký

* **Mô tả vắn tắt**: Use case này cho phép khách hàng đăng ký tài khoản để đăng nhập vào hệ thống để đặt phòng.
* **Luồng sự kiện**:
  + **Luồng cơ bản**:

1. Use ase này bắt đầu khi khách hàng truy cập vào trang chủ của hệ thống.Khách hàng click chọn “Sign Up” trên giao diện hệ thống.
2. Khách hàng tiến hành nhập các thông tin cần đăng ký bao gồm: Email, phone number (số điện thoại) , password( mật khẩu), first name, last name,gender( giới tính) ,ngày tháng năm sinh và address( địa chỉ) sau đó lựa chọn “Submit”.Hệ thống sẽ lưu thông tin tài khoản của khách hàng đã đăng ký vào CSDL trong bảng USER sau đó gửi mã xác nhận về email người dùng dùng để đăng ký. Mã xác nhận được lưu ở bảng VERIFICATION TOKEN trong CSDL .
3. Hệ thống hiển thị form nhập mã xác nhận , người dùng điền mã nhận được ở email đăng ký rồi click chọn “Submit”.Hệ thống hiển thị thông báo đăng ký thành công.

Use case kết thúc.

* **Luồng rẽ nhánh**

1. Tại bước 2 trong luồng cơ bản, khi người dùng nhập thiếu thông tin hoặc sai định dạng cơ bản của thông tin cần điền. Hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại. Người dùng có thể nhập lại để tiếp tục hoặc kết thúc thao tác.
2. Tại bất kỳ thời điểm nào trong quá trình thực hiện Use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và Use case kết thúc.

* **Các yêu cầu đặc biệt:**

Không có.

* **Tiền điều kiện:**

Không có.

* **Hậu điều kiện:**

Không có.

* **Điểm mở rộng:**

Khách hàng có thể đăng nhập được vào hệ thống.

#### Use case Cập nhật thông tin(Phía người dùng)

* **Mô tả vắn tắt:** Use case này cho phép người dùng thay đổi thông tin cá nhân của mình.
* **Luồng sự kiện**
  + **Luồng cơ bản:**

1. Use case này bắt đầu khi khách hàng vào trang cá nhân sau khi đã đăng nhập và click chuột vào tài khoản của mình ở góc phải màn hình và lựa chọn “Thay đổi thông tin cá nhân/My account”. Hệ thống sẽ điều hướng sang màn hình và hiển thị thông tin cá nhân của khách hàng bao gồm: email, tên, số điện thoại, password( mật khẩu) ,giới tính ,địa chỉ, ngày tháng năm sinh từ bảng USER sau đó hiển thị lên màn hình.
2. Khách hàng nhập thông tin muốn thay đổi và nhấn vào nút “Update profile”. Hệ thống cập nhật thông tin vừa thay đổi vào bảng USER trong CSDL, sau đó lấy thông tin cá nhân của người dùng đang đăng nhập và hiển thị lên màn hình.

Use case kết thúc.

* + **Luồng rẽ nhánh:**

1. Tại bước 2 trong luồng cơ bản, khi người dùng nhập thiếu thông tin hoặc sai định dạng cơ bản của thông tin cần điền. Hệ thống sẽ hiển thị thông báo lỗi và yêu cầu nhập lại. Người dùng có thể nhập lại để tiếp tục hoặc kết thúc thao tác.
2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi.

Use case kết thúc.

* **Các yêu cầu đặc biệt:**

Không có.

* **Tiền điều kiện:**

Không có.

* **Hậu điều kiện:**

Sau khi use case kết thúc thành công thì các thông tin cá nhân của người dùng sẽ được cập nhật ở bảng USER.

* **Điểm mở rộng:**

Không có.

#### Use case Xem phòng

* **Mô tả vắn tắt:** Use case này cho phép người dùng xem danh sách các phòng trên hệ thống đặt phòng .
* **Luồng sự kiện**
  + **Luồng cơ bản:**

1. Use case này bắt đầu khi khách hàng click chuột vào “Rooms” trên giao diện trang chủ của hệ thống. Hệ thống sẽ lấy thông tin danh sách phòng còn trống từ bảng ROOM trong CSDL và hiển thị thông tin của các phòng còn trống bao gồm: tiêu đề phòng , giá phòng, size phòng, dịch vụ đi kèm và hiển thị lên màn hình.
2. Khách hàng tiến hành click chọn phòng muốn xem chi tiết và chọn “More Details” ,hệ thống sẽ hiển thị các thông tin chi tiết của phòng từ bảng ROOM trong CSDL bao gồm tiêu đề phòng , giá phòng, size phòng, dịch vụ đi kèm, mô tả chi tiết .

Use case kết thúc

* + **Luồng rẽ nhánh:**

1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi.

Use case kết thúc.

* **Các yêu cầu đặc biệt:**

Không có.

* **Tiền điều kiện:**

Không có.

* **Hậu điều kiện:**

Khách hàng có thể đặt phòng khi xem chi tiết phòng.

* **Điểm mở rộng:**

Không có.

#### Use case Xem chi tiết phòng

* **Mô tả vắn tắt:** Use case này cho phép người dùng xem chi tiết phòng trên hệ thống.
* **Luồng sự kiện**
  + **Luồng cơ bản:**

1. Use case này bắt đầu khi khách hàng click chuột vào “Rooms” trên giao diện trang chủ của hệ thống. Hệ thống sẽ lấy thông tin danh sách phòng còn trống từ bảng ROOM trong CSDL và hiển thị thông tin của các phòng còn trống bao gồm: tiêu đề phòng , giá phòng, size phòng, dịch vụ đi kèm và hiển thị lên màn hình.
2. Khách hàng tiến hành click chọn phòng muốn xem chi tiết ,hệ thống sẽ hiển thị các thông tin chi tiết của phòng từ bảng ROOM trong CSDL bao gồm tiêu đề phòng , giá phòng, size phòng, dịch vụ đi kèm, mô tả chi tiết .

Use case kết thúc

* + **Luồng rẽ nhánh:**

1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi.

Use case kết thúc.

* **Các yêu cầu đặc biệt:**

Không có.

* **Tiền điều kiện:**

Không có.

* **Hậu điều kiện:**

Khách hàng có thể đặt phòng khi xem chi tiết phòng.

* **Điểm mở rộng:**

Không có.

#### Use case Xem bài viết

* **Mô tả vắn tắt:** Use case này cho phép khách hàng xem các bài viết blog trên hệ thống .
* **Luồng sự kiện**
  + **Luồng cơ bản:**

1. Use case này bắt đầu khi khách hàng click chuột vào “Blogs” trên giao diện trang chủ của hệ thống.
2. Hệ thống sẽ hiển thị danh sách các bài viết bao gồm tiêu đề bài viết, hình ảnh từ bảng POST và MEDIA trong CSDL lên màn hình.Khách hàng lựa chọn bài viết muốn xem ,hệ thống sẽ hiện thông tin chi tiết bài viết lên màn hình hệ thống .

Use case kết thúc.

* + **Luồng rẽ nhánh:**

1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi.

Use case kết thúc.

* **Các yêu cầu đặc biệt:**

Không có.

* **Tiền điều kiện:**

Không có.

* **Hậu điều kiện:**

Không có.

* **Điểm mở rộng:**

Không có.

#### Use case Đánh giá

* **Mô tả vắn tắt:** Use case này cho phép khách hàng đánh giá chất lượng phòng sau khi trải nghiệm đặt phòng.
* **Luồng sự kiện**
  + **Luồng cơ bản:**

1. Use case này bắt đầu khi khách hàng click chuột vào “Rooms” trên giao diện trang chủ của hệ thống và lựa chọn phòng đã đặt.
2. Hệ thống sẽ hiển thị giao diện text box đánh giá của khách hàng ở bên dưới mỗi chi tiết phòng. Khách hàng tiến hành đánh giá trải nghiệm sau khi đã đặt phòng tại hệ thống và click chọn “Submit Now”. Thông tin đánh giá của mỗi khách hàng sẽ được lưu vào CSDL trong bảng ROOMRATING. Use case kết thúc.
   * **Luồng rẽ nhánh:**
3. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi.

Use case kết thúc.

* **Các yêu cầu đặc biệt:**

Không có.

* **Tiền điều kiện:**

Khách hàng đã đăng nhập vào hệ thống.

* **Hậu điều kiện:**

Không có.

* **Điểm mở rộng:**

Không có.

#### Use case Đặt phòng

* **Mô tả vắn tắt:** Use case này cho phép khách hàng tiến hành đặt phòng tại hệ thống.
* **Luồng sự kiện**
  + **Luồng cơ bản:**

1. Use case này bắt đầu khi khách hàng click chuột vào “Booking Now” trên giao diện trang chủ của hệ thống.Khách hàng tiến hành chọn thời gian Check-In và Check-Out , chọn số người cần thuê phòng và loại phòng sau đó nhấn chọn “Check available”.Hệ thống sẽ lấy thông tin danh sách phòng theo loại phòng, số người theo yêu cầu trong khoảng thời gian khách hàng chọn và khuyến mãi ở mỗi phòng từ bảng ROOM, SALE trong CSDL và hiển thị thông tin của các phòng còn trống bao gồm: tiêu đề phòng , giá phòng, size phòng và hiển thị lên màn hình.
2. Khách hàng tiến hành chọn phòng muốn đặt và click vào “Add to my booking”. Sau đó khách hàng tiến hành chọn dịch vụ theo ý muốn rồi click vào “Add Service”, thông tin dịch vụ được lấy từ bảng SERVICE trong CSDL hiển thị lên màn hình bao gồm: tên dịch vụ, mô tả và giá dịch vụ…. Khách hàng xác nhận đặt phòng bằng cách nhấn chọn “Countinue”. Booking sẽ chuyển status sang trạng thái Pending (Ở trạng thái Pending khách hàng khác vẫn có thể booking phòng này nhưng chỉ có thể booking với những thời điểm khác với những khách hàng đã booking phòng này). Hệ thống sẽ hiển thị thông báo “Đặt phòng thành công !”
3. Khi khách hàng đến nhận phòng và check in thành công, thì trạng thái sẽ chuyển sang Checked-In thông tin sẽ được cập nhật vào bảng BOOKING, BOOKINGROOMDETAIL, BOOKINGSERVICEDETAIL và ROOM trong CSDL và những người khác sẽ không thể booking được phòng này được nữa.

Use case kết thúc.

* + **Luồng rẽ nhánh:**

1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi.

Use case kết thúc.

* **Các yêu cầu đặc biệt:**

Không có.

* **Tiền điều kiện:**

Khách hàng đăng nhập thành công vào hệ thống.

* **Hậu điều kiện:**

Khách hàng có thể đặt phòng khi xem chi tiết phòng.

* **Điểm mở rộng:**

Xem chi tiết đơn đặt phòng và có thể hủy phòng nếu thỏa mãn yêu cầu hủy.

#### Use case Hủy phòng

* **Mô tả vắn tắt:** Use case này cho phép khách hàng hủy phòng mà mình đã đặt trước đó.
* **Luồng sự kiện**
  + **Luồng cơ bản:**

1. Use case này bắt đầu khi khách hàng click chuột vào “Booking Cart” trên giao diện trang chủ của hệ thống. Giao diện sẽ hiển thị danh sách phòng mà khách hàng đã đặt.
2. Khách hàng tiến hành click để hủy phòng đã đặt.Hệ thống sẽ đưa ra màn hình danh sách các lựa chọn lý do hủy phòng .Khách hàng tiến hành chọn lý do hủy và lựa chọn “ Cancel”. Trạng thái phòng được hủy sẽ chuyển sang Cancel và dữ liệu sẽ được lưu vào bảng BOOKING trong CSDL.
3. Khách hàng chỉ được phép hủy đặt phòng trước ngày nhận phòng lớn hơn một ngày nếu không thì hệ thống sẽ không cho phép khách hàng hủy phòng.

Use case kết thúc

* + **Luồng rẽ nhánh:**

1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi.

Use case kết thúc.

* **Các yêu cầu đặc biệt:**

Không có.

* **Tiền điều kiện:**

Không có.

* **Hậu điều kiện:**

Khách hàng có thể đặt phòng khi xem chi tiết phòng.

* **Điểm mở rộng:**

Không có.

#### Use case Sử dụng chatbot tự động

#### Use case Đặt dịch vụ

* **Mô tả vắn tắt:** Use case này cho phép khách hàng đặt thêm dịch vụ ở mỗi lần đặt phòng.
* **Luồng sự kiện**
  + **Luồng cơ bản:**

1. Use case này bắt đầu khi khách hàng click chuột vào “Rooms” trên giao diện trang chủ của hệ thống. Hệ thống sẽ lấy thông tin danh sách phòng còn trống từ bảng ROOM trong CSDL và hiển thị thông tin của các phòng còn trống bao gồm: tiêu đề phòng , giá phòng, size phòng, dịch vụ đi kèm và hiển thị lên màn hình.
2. Khách hàng tiến hành click chọn phòng muốn xem chi tiết ,hệ thống sẽ hiển thị các thông tin chi tiết của phòng từ bảng ROOM trong CSDL bao gồm tiêu đề phòng , giá phòng, size phòng, dịch vụ đi kèm, mô tả chi tiết .

Use case kết thúc

* + **Luồng rẽ nhánh:**

1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case, nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi.

Use case kết thúc.

* **Các yêu cầu đặc biệt:**

Không có.

* **Tiền điều kiện:**

Không có.

* **Hậu điều kiện:**

Khách hàng có thể đặt phòng khi xem chi tiết phòng.

* **Điểm mở rộng:**

Không có.

#### Use case Bảo trì phòng

* **Mô tả vắn tắt:** Use case này cho phép quản trị viên xem, thêm, sửa, xoá phòng trên hệ thống.
* **Luồng sự kiện**
  + **Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng nhấn vào nút “Manage Rooms” trên thanh menu quản trị. Hệ thống sẽ lấy thông tin trong bảng ROOM từ CSDL bao gồm các thông tin: id, title, price( giá), type (Loại phòng) , maxNum( số lượng người ), Roor, Description( Giới thiệu) và hiển thị lên màn hình.
2. Thêm phòng mới:
   1. Người quản trị nhấn vào nút “Add rooms” trong màn hình danh sách phòng . Hệ thống sẽ hiển thị modal yêu cầu nhập thông tin cho phòng mới cần tạo.
   2. Người quản trị nhập thông tin cho phòng mới và nhấn vào nút “Submit”. Hệ thống sẽ thêm một bản ghi mới vào bảng ROOM và MEDIA sau đó lấy thông tin về danh sách phòng mới gồm id, title, price( giá), type (Loại phòng) , maxNum( số lượng người ), Roor, Description( Giới thiệu)và hiển thị lên màn hình.
3. Sửa thông tin phòng :
   1. Người quản trị nhấn vào biểu tưởng chỉnh sửa trên một dòng của từng phòng. Hệ thống sẽ lấy ra thông tin của phòng đó từ bảng ROOM trong CSDL .
   2. Người quản trị sửa thông tin cần sửa cho phòng được chọn sau đó nhấn vào nút “Cập nhật”. Hệ thống cập nhật thông tin phòng vào bảng ROOM sau đó lấy ra thông tin danh sách phòng số gồm id, title, price( giá), type (Loại phòng) , maxNum( số lượng người ), Roor, Description( Giới thiệu) và hiển thị lên màn hình.
4. Xoá phòng:
   1. Người quản trị nhấn vào biểu tượng xoá trên một dòng chỉ phòng cần xóa. Hệ thống sẽ hiển thị lên popup để xác nhận xoá.
   2. Người quản trị nhấn vào nút “Yes, delete it!”, hệ thống sẽ thực hiện xoá phòng đã chọn khỏi danh sách phòng trên hệ thông và thông tin sẽ được cập nhật vào bảng ROOM và các bảng liên quan, sau đó lấy thông tin danh sách thông số gồm id, title, price( giá), type (Loại phòng) , maxNum( số lượng người ), Floor, Description( Giới thiệu) và hiển thị lên màn hình.

Use case kết thúc.

* **Luồng rẽ nhánh:**

1. Tại thời điểm bất kỳ trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
2. Tại bước 1 trong luồng cơ bản, nếu không có bản ghi nào trong bảng ROOM thì hệ thống sẽ hiển thị thông báo“No Data” và use case kết thúc.
3. Tại bước 2b và 3b trong luồng cơ bản, khi quản trị viên nhập thông tin phòng không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi yêu cầu người quản trị nhập lại, quản trị viên có thể tiếp tục nhập lại hoặc nhấn “Cancel” để kết thúc.
4. Tại bước 4b trong luồng cơ bản, nếu người quản trị nhấn vào nút “Cancel” thì hệ thống sẽ bỏ qua thao tác xoá và hiển thị danh sách các phòng trong bảng ROOM lên màn hình và use case kết thúc.

* **Các yêu cầu đặc biệt**

Cần kiểm soát quyền thực hiện use case này để đảm bảo tính an toàn và bảo mật.

* **Tiền điều kiện**

Người quản trị cần đăng nhập với quyền quản trị trước khi thực hiện use case.

* **Hậu điều kiện**

Sau khi use case kết thúc thành công thì lưu các thông tin vào bảng ROOM,MEDIA

* **Điểm mở rộng**

Không có

#### Use case Bảo trì bài viết

* **Mô tả vắn tắt:** Use case này cho phép quản trị viên xem, thêm, sửa, xoá bài viết trên hệ thống.
* **Luồng sự kiện**
  + **Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng nhấn vào nút “Manage Postss” trên thanh menu quản trị. Hệ thống sẽ lấy trong bảng POST gồm các thông tin: PostId, Content, và hiển thị lên màn hình.
2. Thêm bài viết mới:
   1. Người quản trị nhấn vào nút “Add posts” trong màn hình danh sách thông số. Hệ thống sẽ hiển thị modal yêu cầu nhập thông tin cho bài viết gồm PostId, Content, CreatedBy, LastModifiedBy cho bài viết .
   2. Người quản trị nhập thông tin cho thông số mới và nhấn vào nút “Submit”. Hệ thống sẽ thêm một bản ghi mới vào bảng POST và MEDIA sau đó lấy thông tin về danh sách bài viế mới gồm PostId, Content, CreatedBy, LastModifiedBy và hiển thị lên màn hình.
3. Sửa bài viết :
   1. Người quản trị nhấn vào biểu tưởng chỉnh sửa trên một dòng bài viết chần chỉnh sửa. Hệ thống sẽ lấy thông tin của bài viết được chọn từ bảng POST.
   2. Người quản trị sẽ sửa các thông tin của bài viết bào gồm: PostId, Content sau đó nhấn vào nút “Submit”. Hệ thống cập nhật thông tin bài viết vào bảng POST sau đó lấy thông tin danh bài viết bao gồm PostId, Content, CreatedBy, LastModifiedBy và hiển thị lên màn hình.
4. Xoá bài viết:
   1. Người quản trị nhấn vào biểu tượng xoá trên một dòng bài viết. Hệ thống sẽ hiển thị lên popup để xác nhận xoá.
   2. Người quản trị nhấn vào nút “Yes,delete it!”, hệ thống sẽ thực hiện xoá bài viết đã chọn khỏi bảng POST sau đó và cập nhật lại thông tin của các bảng liên quan, lấy thông tin danh sách bài viết gồm PostId, Content, CreatedBy, LastModifiedBy và hiển thị lên màn hình.

Use case kết thúc.

* **Luồng rẽ nhánh:**

1. Tại thời điểm bất kỳ trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
2. Tại bước 1 trong luồng cơ bản, nếu không có bản ghi nào trong bảng POST thì hệ thống sẽ hiển thị “No Data” và use case kết thúc.
3. Tại bước 2b và 3b trong luồng cơ bản, khi quản trị viên nhập thông tin bài viết không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi yêu cầu người quản trị nhập lại, quản trị viên có thể tiếp tục nhập lại hoặc nhấn “Cancel” để kết thúc.
4. Tại bước 4b trong luồng cơ bản, nếu người quản trị nhấn vào nút “Cancel” thì hệ thống sẽ bỏ qua thao tác xoá và hiển thị danh sách các bài viết trong bảng POST lên màn hình và use case kết thúc.

* **Các yêu cầu đặc biệt**

Cần kiểm soát quyền thực hiện use case này để đảm bảo tính an toàn và bảo mật.

* **Tiền điều kiện**

Người quản trị cần đăng nhập với quyền quản trị trước khi thực hiện use case.

* **Hậu điều kiện**

Sau khi use case kết thúc thành công thì lưu các thông tin vào bảng POST và MEDIA.

* **Điểm mở rộng**

Không có

#### Use case Bảo trì dịch vụ

* **Mô tả vắn tắt:** Use case này cho phép quản trị viên xem, thêm, sửa, xoá dịch vụ ở mỗi phòng trên hệ thống.
* **Luồng sự kiện**
  + **Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng nhấn vào nút “Manage Services” trên thanh menu quản trị. Hệ thống sẽ lấy thông tin trong bảng SERVICE từ CSDL gồm các thông tin: id, title (tiêu đề), price (Giá), Description và hiển thị lên màn hình.
2. Thêm dịch vụ mới:
   1. Người quản trị nhấn vào nút “Add services” trong màn hình danh sách dịch vụ. Hệ thống sẽ hiển thị modal yêu cầu nhập thông tin cho dịch vụ bao gồm title (tiêu đề), price (Giá), Description và hình ảnh cho dịch vụ.
   2. Người quản trị nhập thông tin cho dịch vụ mới và nhấn vào nút “Submit”. Hệ thống sẽ thêm một bản ghi mới vào bảng SERVICE sau đó lấy thông tin về danh sách dịch vụ mới gồm : id, title (tiêu đề), price (Giá), Description và hiển thị lên màn hình.
3. Sửa dịch vụ :
   1. Người quản trị nhấn vào biểu tưởng chỉnh sửa trên một dòng dịch vụ . Hệ thống sẽ lấy tên của dịch vụ được chọn từ bảng SERVICE và hiển thị lên màn hình thông tin dịch vụ bao gồm .
   2. Người quản trị sẽ sửa thông tin của dịch vụ sau đó nhấn vào nút “Submit”. Hệ thống cập nhật thông tin dịch vụ vào bảng SERVICE sau đó lấy thông tin danh sách dịch vụ gồm id, title (tiêu đề), price (Giá), Description và hiển thị lên màn hình.
4. Xoá dịch vụ :
   1. Người quản trị nhấn vào biểu tượng xoá trên một dòng dịch vụ. Hệ thống sẽ hiển thị lên popup để xác nhận xoá.
   2. Người quản trị nhấn vào nút “Yes,delete it!”, hệ thống sẽ thực hiện xoá dịch vụ đã chọn khỏi bảng SERVICE sau đó lấy thông tin danh sách dịch vụ gồm id, title (tiêu đề), price (Giá), Description và hiển thị lên màn hình.

Use case kết thúc.

* **Luồng rẽ nhánh:**

1. Tại thời điểm bất kỳ trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
2. Tại bước 1 trong luồng cơ bản, nếu không có bản ghi nào trong bảng SERVICE thì hệ thống sẽ hiển thị “No Data” và use case kết thúc.
3. Tại bước 2b và 3b trong luồng cơ bản, khi quản trị viên nhập thông tin dịch vụ không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi yêu cầu người quản trị nhập lại, quản trị viên có thể tiếp tục nhập lại hoặc nhấn chọn “Cancel” để kết thúc.
4. Tại bước 4b trong luồng cơ bản, nếu người quản trị nhấn vào nút “Cancel” thì hệ thống sẽ bỏ qua thao tác xoá và hiển thị danh sách các dịch vụ trong bảng SERVICE lên màn hình và use case kết thúc.

* **Các yêu cầu đặc biệt**

Cần kiểm soát quyền thực hiện use case này để đảm bảo tính an toàn và bảo mật.

* **Tiền điều kiện**

Người quản trị cần đăng nhập với quyền quản trị trước khi thực hiện use case.

* **Hậu điều kiện**

Sau khi use case kết thúc thành công thì lưu các thông tin vào bảng SERVICE

* **Điểm mở rộng**

Không có

#### Use case Bảo trì khuyến mãi

* **Mô tả vắn tắt:** Use case này cho phép quản trị viên xem, thêm, sửa, xoá khuyến mãi ở mỗi phòng trên hệ thống.
* **Luồng sự kiện**
  + **Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng nhấn vào nút “Manage Sales” trên thanh menu quản trị. Hệ thống sẽ lấy thông tin trong bảng SALE từ CSDL gồm các thông tin: day start (ngày bắt đầu), day end( ngày kết thúc) , sale percent( giá trị giảm giá) và hiển thị lên màn hình.
2. Thêm khuyến mãi mới:
   1. Người quản trị nhấn vào nút “Add sales” trong màn hình danh sách khuyến mãi. Hệ thống sẽ hiển thị modal yêu cầu nhập thông tin cho khuyến mãi bao gồm day start (ngày bắt đầu), day end( ngày kết thúc) , sale percent( giá trị giảm giá).
   2. Người quản trị nhập thông tin cho khuyến mãi mới và nhấn vào nút “Submit”. Hệ thống sẽ thêm một bản ghi mới vào bảng SALE sau đó lấy thông tin về danh sách khuyến mãi mới gồm : day start (ngày bắt đầu), day end( ngày kết thúc) , sale percent( giá trị giảm giá) và hiển thị lên màn hình.
3. Sửa khuyến mãi :
   1. Người quản trị nhấn vào biểu tưởng chỉnh sửa trên một dòng khuyến mãi. Hệ thống sẽ lấy thông tin của khuyến mãi được chọn từ bảng SALE và hiển thị lên màn hình thông tin khuyến mãi bao gồm: day start (ngày bắt đầu), day end( ngày kết thúc) , sale percent( giá trị giảm giá) .
   2. Người quản trị sẽ sửa thông tin của khuyến mãi sau đó nhấn vào nút “Submit”. Hệ thống cập nhật thông tin dịch vụ vào bảng SALE sau đó lấy thông tin danh sách khuyến mãi gồm day start (ngày bắt đầu), day end( ngày kết thúc) , sale percent( giá trị giảm giá) và hiển thị lên màn hình.
4. Xoá khuyến mãi :
   1. Người quản trị nhấn vào biểu tượng xoá trên một dòng khuyến mãi. Hệ thống sẽ hiển thị lên popup để xác nhận xoá.
   2. Người quản trị nhấn vào nút “Yes,delete it!”, hệ thống sẽ thực hiện xoá dịch vụ đã chọn khỏi bảng SALE sau đó lấy thông tin danh sách khuyến mãi gồm day start (ngày bắt đầu), day end( ngày kết thúc) , sale percent( giá trị giảm giá) và hiển thị lên màn hình.

Use case kết thúc.

* **Luồng rẽ nhánh:**

1. Tại thời điểm bất kỳ trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
2. Tại bước 1 trong luồng cơ bản, nếu không có bản ghi nào trong bảng SERVICE thì hệ thống sẽ hiển thị “No Data” và use case kết thúc.
3. Tại bước 2b và 3b trong luồng cơ bản, khi quản trị viên nhập thông tin dịch vụ không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi yêu cầu người quản trị nhập lại, quản trị viên có thể tiếp tục nhập lại hoặc nhấn chọn “Cancel” để kết thúc.
4. Tại bước 4b trong luồng cơ bản, nếu người quản trị nhấn vào nút “Cancel” thì hệ thống sẽ bỏ qua thao tác xoá và hiển thị danh sách các dịch vụ trong bảng SERVICE lên màn hình và use case kết thúc.

* **Các yêu cầu đặc biệt**

Cần kiểm soát quyền thực hiện use case này để đảm bảo tính an toàn và bảo mật.

* **Tiền điều kiện**

Người quản trị cần đăng nhập với quyền quản trị trước khi thực hiện use case.

* **Hậu điều kiện**

Sau khi use case kết thúc thành công thì lưu các thông tin vào bảng SALE

* **Điểm mở rộng**

Không có

#### Use cas Bảo trì sản phẩm đi kèm dịch vụ

* **Mô tả vắn tắt:** Use case này cho phép quản trị viên xem, thêm, sửa, xoá sản phẩm đi kèm dịch vụ tại mỗi phòng trên hệ thống.
* **Luồng sự kiện**
  + **Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng nhấn vào nút “Manage Products” trên thanh menu quản trị. Hệ thống sẽ lấy thông tin trong bảng PRODUCT,SERVICE từ CSDL gồm các thông tin: ProductId,tên sản phẩm , Description (Giới thiệu) , ServiceId và hiển thị lên màn hình.
2. Thêm sản phẩm mới cho dịch vụ:
   1. Người quản trị nhấn vào nút “Add products” trong màn hình danh sách sản phẩm đi kèm từng dịch vụ. Hệ thống sẽ hiển thị modal yêu cầu nhập thông tin cho sản phẩm bao gồm day ProductId,tên sản phẩm , Description (Giới thiệu) , ServiceId.
   2. Người quản trị nhập thông tin cho sản phẩm mới bao gồm: ProductId,tên sản phẩm , Description (Giới thiệu) , ServiceId, Thumbnai (ảnh sản phẩm) và nhấn vào nút “Submit”. Hệ thống sẽ thêm một bản ghi mới vào bảng PRODUCT sau đó lấy thông tin về danh sách sản phẩm mới gồm : ProductId,tên sản phẩm , Description (Giới thiệu) , ServiceId và hiển thị lên màn hình.
3. Sửa thông tin sản phẩm:
   1. Người quản trị nhấn vào biểu tưởng chỉnh sửa trên một dòng sản phẩm . Hệ thống sẽ lấy thông tin của sản phẩm được chọn từ bảng PRODUCT và hiển thị lên màn hình thông tin sản phẩm bao gồm: ProductId,tên sản phẩm , Description (Giới thiệu) , ServiceId. Người quản trị sẽ sửa thông tin của sản phẩm sau đó nhấn vào nút “Submit”. Hệ thống cập nhật thông tin sản phẩm vào bảng PRODUCT sau đó lấy thông tin danh sách sản phẩm bao gồm: ProductId,tên sản phẩm , Description (Giới thiệu) , ServiceId và hiển thị lên màn hình.
4. Xoá sản phẩm :
   1. Người quản trị nhấn vào biểu tượng xoá trên một dòng sản phẩm. Hệ thống sẽ hiển thị lên popup để xác nhận xoá.
   2. Người quản trị nhấn vào nút “Yes,delete it!”, hệ thống sẽ thực hiện xoá sản phẩm đã chọn khỏi bảng PRODUCT sau đó lấy thông tin danh sách sản phẩm gồm ProductId,tên sản phẩm , Description (Giới thiệu) , ServiceId và hiển thị lên màn hình.

Use case kết thúc.

* **Luồng rẽ nhánh:**

1. Tại thời điểm bất kỳ trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
2. Tại bước 1 trong luồng cơ bản, nếu không có bản ghi nào trong bảng PRODUCT thì hệ thống sẽ hiển thị “No Data” và use case kết thúc.
3. Tại bước 2b và 3b trong luồng cơ bản, khi quản trị viên nhập thông tin sản phẩm không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi yêu cầu người quản trị nhập lại, quản trị viên có thể tiếp tục nhập lại hoặc nhấn chọn “Cancel” để kết thúc.
4. Tại bước 4b trong luồng cơ bản, nếu người quản trị nhấn vào nút “Cancel” thì hệ thống sẽ bỏ qua thao tác xoá và hiển thị danh sách các sản phẩm trong bảng PRODUCT lên màn hình và use case kết thúc.

* **Các yêu cầu đặc biệt**

Cần kiểm soát quyền thực hiện use case này để đảm bảo tính an toàn và bảo mật.

* **Tiền điều kiện**

Người quản trị cần đăng nhập với quyền quản trị trước khi thực hiện use case.

* **Hậu điều kiện**

Sau khi use case kết thúc thành công thì lưu các thông tin vào bảng PRODUCT

* **Điểm mở rộng**

Không có

#### Use case Quản lý người dùng

* **Mô tả vắn tắt:** Use case này cho phép quản trị viên xem, thêm, sửa, xoá người dùng trên hệ thống.
* **Luồng sự kiện**
  + **Luồng cơ bản:**

1. Use case này bắt đầu khi người dùng nhấn vào nút “Manage Users” trên thanh menu quản trị. Hệ thống sẽ lấy thông tin trong bảng USER từ CSDL gồm các thông tin: UserId,email,phone number, password,first name, last name, gender,birthday, address và hiển thị lên màn hình.
2. Thêm người dùng:
   1. Người quản trị nhấn vào nút “Add users” trong màn hình danh sách người dùng. Hệ thống sẽ hiển thị modal yêu cầu nhập thông tin cho người dùng bao gồm day UserId,email,phone number, password,first name, last name, gender,birthday, address.
   2. Người quản trị nhập thông tin cho người dùng mới bao gồm: UserId,email,phone number, password,first name, last name, gender,birthday, address và nhấn vào nút “Submit”. Hệ thống sẽ thêm một bản ghi mới vào bảng USER sau đó lấy thông tin về danh sách người dùng mới gồm : UserId,email,phone number, password,first name, last name, gender,birthday, address và hiển thị lên màn hình.
3. Sửa thông tin người dùng:
   1. Người quản trị nhấn vào biểu tưởng chỉnh sửa trên một dòng người dùng. Hệ thống sẽ lấy thông tin của người dùng được chọn từ bảng USER và hiển thị lên màn hình thông tin người dùng bao gồm: UserId,email,phone number, password,first name, last name, gender,birthday, address. Người quản trị sẽ sửa thông tin của người dùng cần sửa sau đó nhấn vào nút “Submit”. Hệ thống cập nhật thông tin người dùng vào bảng USER sau đó lấy thông tin danh sách người dùng bao gồm: UserId,email,phone number, password,first name, last name, gender,birthday, address và hiển thị lên màn hình.
4. Xoá người dùng :
   1. Người quản trị nhấn vào biểu tượng xoá trên một dòng người dùng. Hệ thống sẽ hiển thị lên popup để xác nhận xoá.
   2. Người quản trị nhấn vào nút “Yes,delete it!”, hệ thống sẽ thực hiện xoá người dùng đã chọn khỏi bảng USER sau đó lấy thông tin danh sách người dùng gồm UserId,email,phone number, password,first name, last name, gender,birthday, address và hiển thị lên màn hình.

Use case kết thúc.

* **Luồng rẽ nhánh:**

1. Tại thời điểm bất kỳ trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
2. Tại bước 1 trong luồng cơ bản, nếu không có bản ghi nào trong bảng USER thì hệ thống sẽ hiển thị “No Data” và use case kết thúc.
3. Tại bước 2b và 3b trong luồng cơ bản, khi quản trị viên nhập thông tin người dùng không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi yêu cầu người quản trị nhập lại, quản trị viên có thể tiếp tục nhập lại hoặc nhấn chọn “Cancel” để kết thúc.
4. Tại bước 4b trong luồng cơ bản, nếu người quản trị nhấn vào nút “Cancel” thì hệ thống sẽ bỏ qua thao tác xoá và hiển thị danh sách người dùng trong bảng USER lên màn hình và use case kết thúc.

* **Các yêu cầu đặc biệt**

Cần kiểm soát quyền thực hiện use case này để đảm bảo tính an toàn và bảo mật.

* **Tiền điều kiện**

Người quản trị cần đăng nhập với quyền quản trị trước khi thực hiện use case.

* **Hậu điều kiện**

Sau khi use case kết thúc thành công thì lưu các thông tin vào bảng USER

* **Điểm mở rộng**

Không có

#### Use case thống kê đặt phòng

#### Use case thống kê đặt phòng nhiều nhất

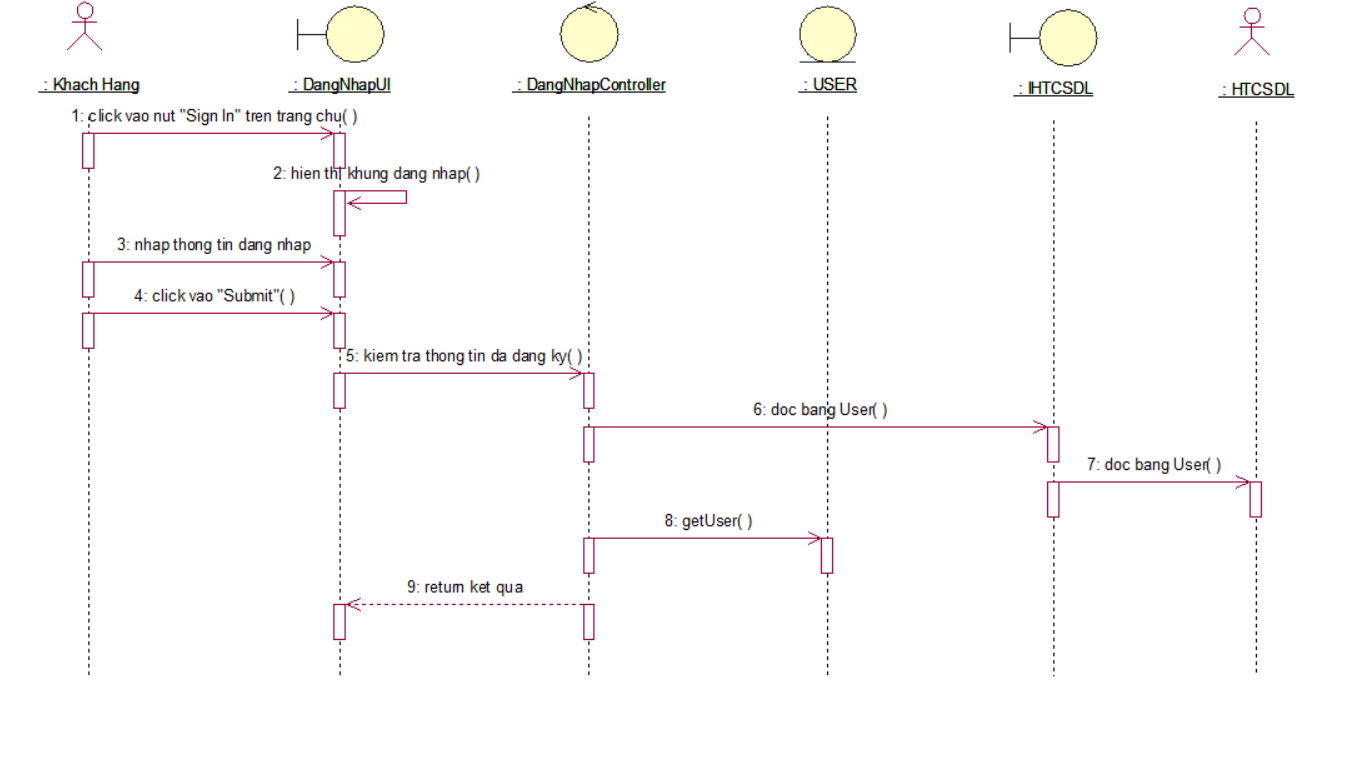
### Phân tích các use case

#### Use case Đăng Nhập

* **Biểu đồ lớp phân tích**



* **Biểu đồ trình tự**



#### Use case Đăng ký

* ***Biểu đồ lớp phân tích VOPC***



* ***Biểu đồ trình tự***



#### Use case Cập nhật thông tin cá nhân – Phía User

* ***Biểu đồ lớp phân tích VOPC:***



* ***Biểu đồ trình tự :***



#### Use case Xem phòng

* ***Biểu đồ lớp phân tích :***



* ***Biểu đồ trình tự:***

#### Use case Xem thông tin khuyến mãi

#### Use case Đánh giá

* **Biểu đồ lớp phân tích :**



* **Biều đồ lớp trình tự:**



#### Use case Đặt phòng

* **Biểu đồ lớp phân tích VOPC:**



* **Biểu đồ trình tự:**



#### Use case Hủy phòng

* ***Biểu đồ lớp phân tích VOPC:***

******

* ***Biểu đồ trình tự:***

#### Use case Sử dụng chat bot tự động

* ***Biểu đồ lớp phân tích VOPC:***



* ***Biều đồ trình tự :***

#### Use case Bảo trì phòng

* ***Biểu đồ lớp phân tích VOPC :***

******

* ***Biểu đồ trình tự :***

******

#### Use case Bảo trì bài viết

* ***Biểu đồ lớp phân tích VOPC:***

******

* ***Biểu đồ trình tự:***

******

#### Use case Bảo trì dịch vụ

* ***Biểu đồ VOPC:***

******

* ***Biểu đồ trình tự:***



#### Use case Bảo trì khuyến mãi

* ***Biểu đồ VOPC:***

******

* ***Biểu đồ trình tự:***

******

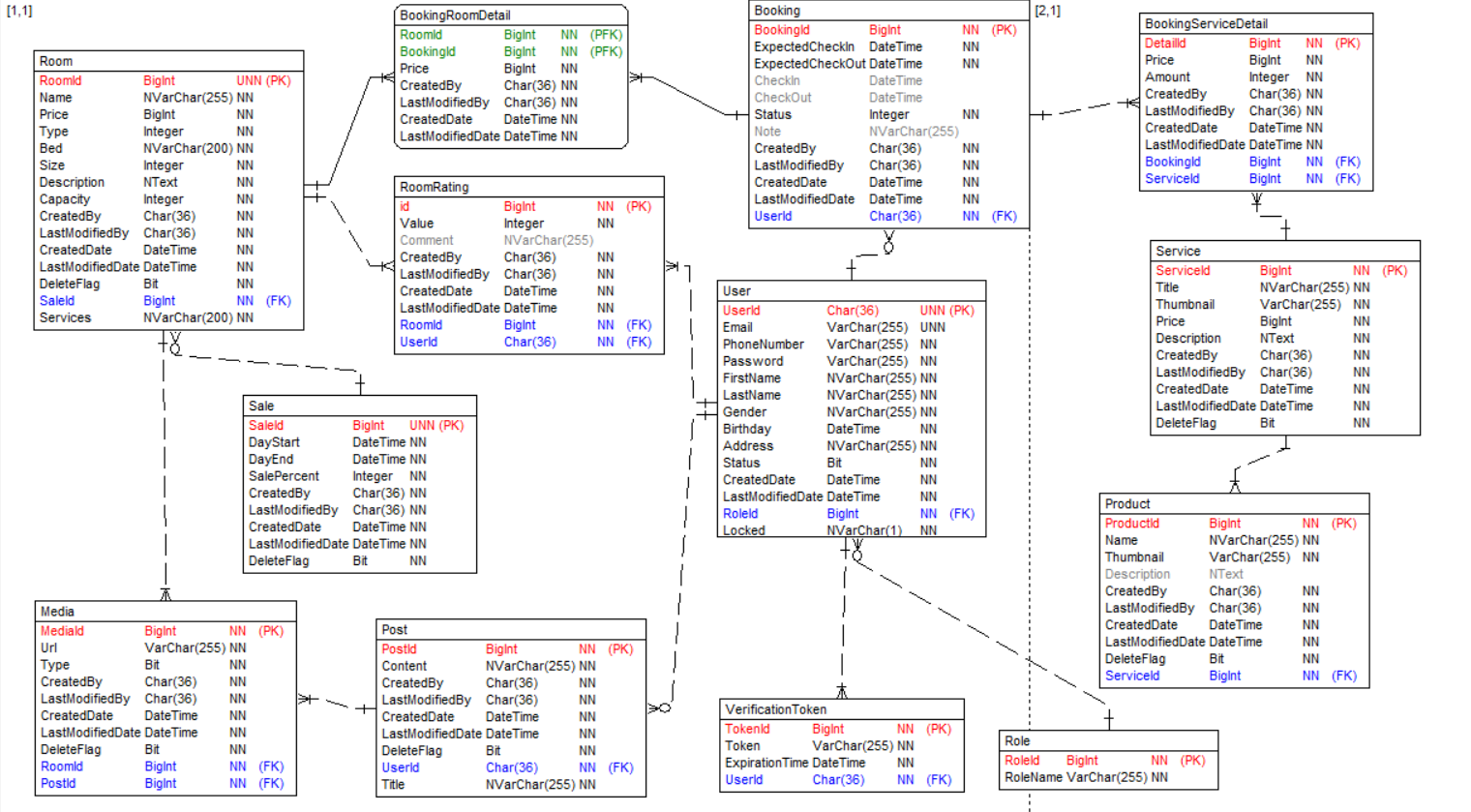
#### Use case Quản lý người dùng/khách hàng

#### Use case thống kê số lượng đặt phòng

#### Use case thống kê khách hàng đặt phòng nhiều nhất

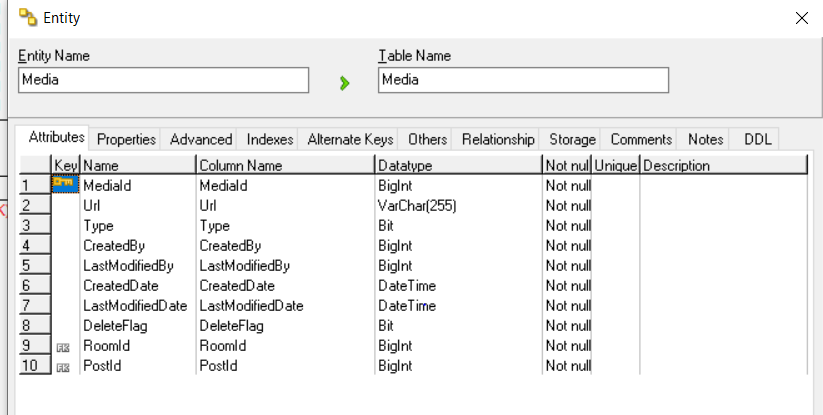
## Xây dựng cơ cở dữ liệu

### Biểu đồ Entity Relationship Diagram



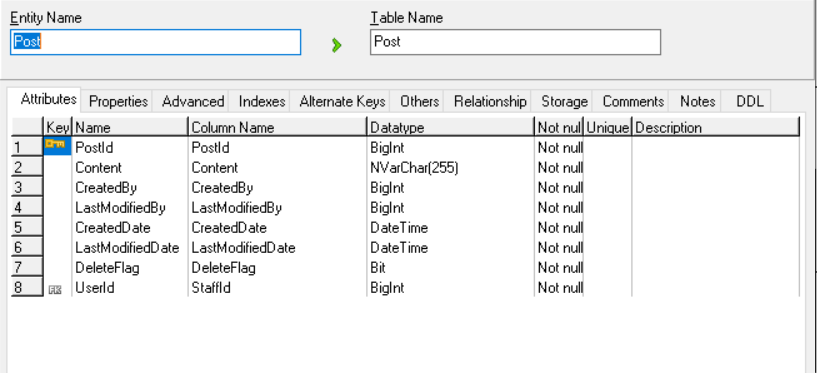
### Các bảng trong cơ sở dữ liệu

* **Bảng Media** (Lưu ảnh phòng)



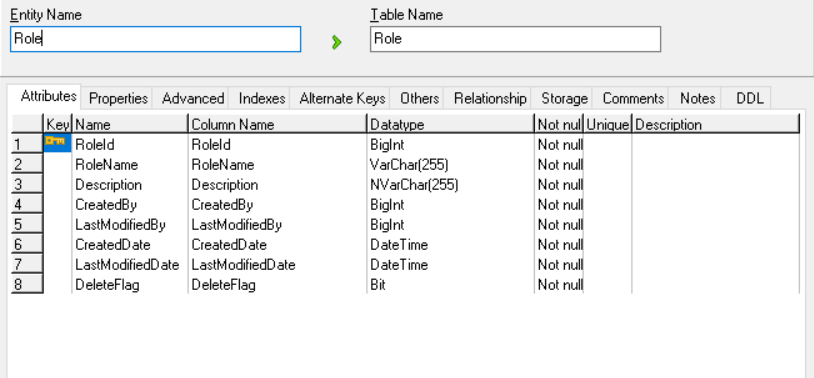
Media: Bảng lưu thông tin ảnh/video

* MediaId: Khóa chính, dùng để xác định bài viết
* Url: đường dẫn ảnh/video
* Type: loại ảnh/video
* RoomId: Khóa ngoại liên kết với bảng Room để xác định ảnh/video của Room
* PostId: Khóa ngoại liên kết với bảng Post để xác định ảnh/video của Post
* **Bảng Post** (Lưu bài viết trên trang)



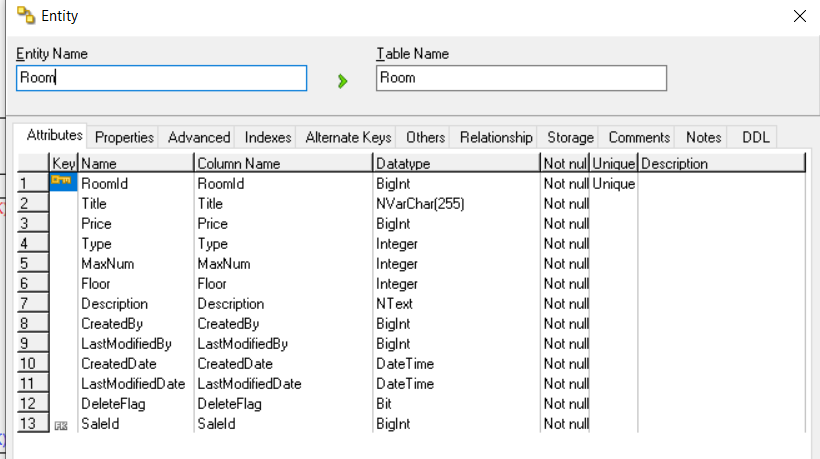
Post: Bảng lưu thông tin bài viết

* PostId: Khóa chính, dùng để xác định bài viết
* Content: Nội dung
* UserId: Khóa ngoại liên kết với bảng User để xác định người đăng bài
* **Bảng Role** (Lưu thông tin quyền của người dùng)



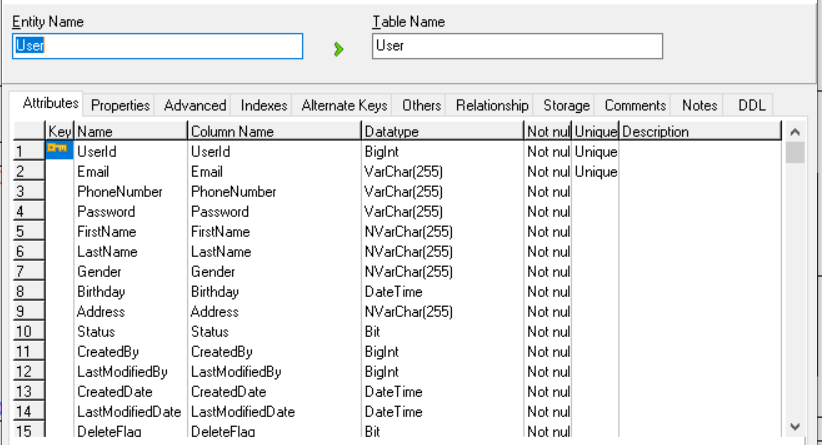
Role: Bảng lưu thông tin các quyền truy cập của người dùng

* RoleId: Khóa chính, dùng để xác định các quyền
* RoleName: Tên quyền
* Description: Mô tả quyền
* CreatedBy: Được tạo bởi
* LastModifiedBy:
* CreatedDate: Được tạo vào ngày nào
* LastModifiedDate:
* DeleteFlag:Sử dụng để xóa mềm
* **Bảng Room** (Lưu thông tin phòng)



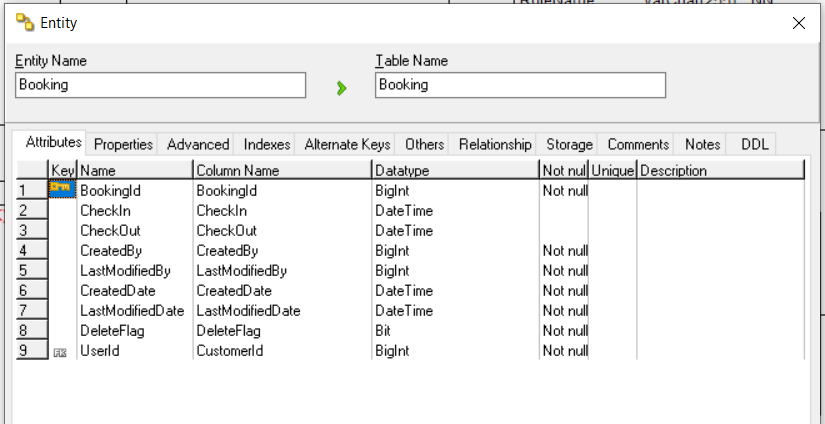
Room: Bảng lưu thông tin các phòng

* RoomId: Khóa chính, dùng để xác định các phòng
* Title: Tiêu đề phòng
* Price: Giá phòng
* Type: Loại phòng (Phòng vip, phòng thường, …)
* MaxNum: Số người tối đa
* Floor: Vị trí tầng của phòng
* Description: Mô tả phòng
* Status: Trạng thái phòng còn trống hay đã hết
* CreatedBy: Người tạo phòng
* LastModifiedBy: Người sửa đổi phòng cuối cùng
* CreatedDate: Ngày tạo phòng
* LastModifiedDate: Ngày sửa đổi cuối cùng
* DeleteFlag: Sử dụng để xóa mềm các phòng
* **Bảng User** (Lưu thông tin người dùng)



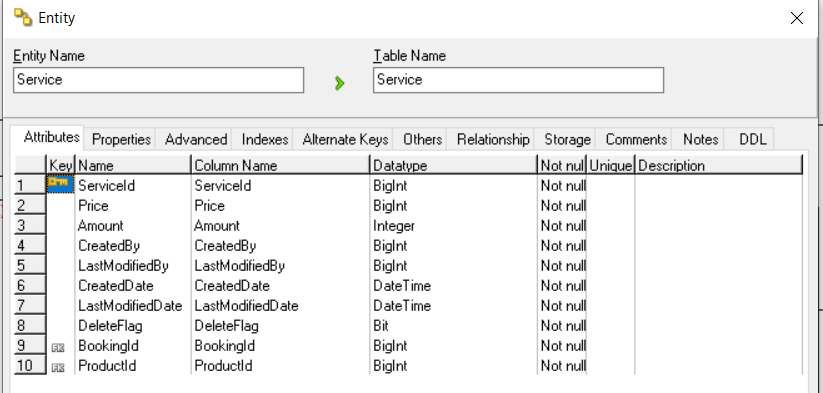
User: Bảng lưu thông tin người dùng

* UserId: Khóa chính, dùng để xác định các user
* Email: Email của user dùng để đăng nhập
* PhoneNumber: Số điện thoại dùng để đăng nhập
* Password: Mật khẩu dùng để đăng nhập
* FirstName: Tên
* LastName: Họ
* Gender: Giới tính
* Birthday: Ngày sinh
* Address: Địa chỉ
* Status: Trạng thái xác thực tài khoản
* CreatedDate: Ngày tạo user
* LastModifiedDate: Ngày sửa đổi cuối cùng
* RoleId: Khóa ngoại liên kết với bảng Role để xác định quyền của người dùng
* **Bảng Booking** (Lưu thông tin đặt phòng)



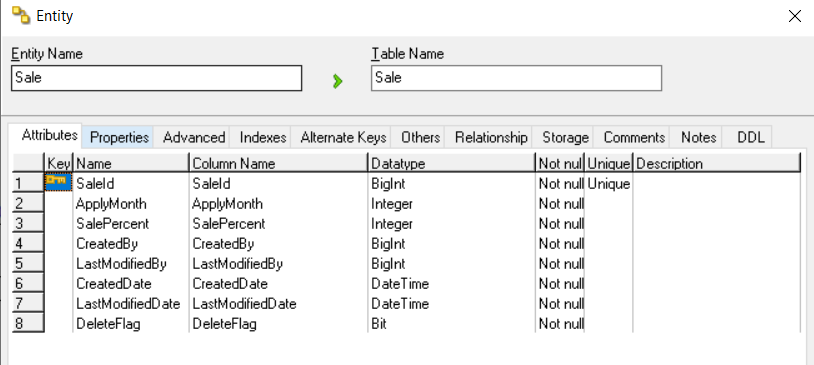
Booking: Bảng lưu thông tin các lần đặt phòng

* BookingId : Khóa chính, dùng để xác định đơn đặt phòng của khách hàng
* CheckIn: Ngày nhận phòng
* CheckOut: Ngày trả phòng
* Status: trạng thái booking (pending, checked in, checked out, cacel)
* Note: để lưu lý do nếu khách hàng hủy đặt phòng
* UserId: Khóa ngoại liên kết với bảng User để xác định người đặt phòng
* **Bảng Service** (Lưu thông tin dịch vụ theo phòng)



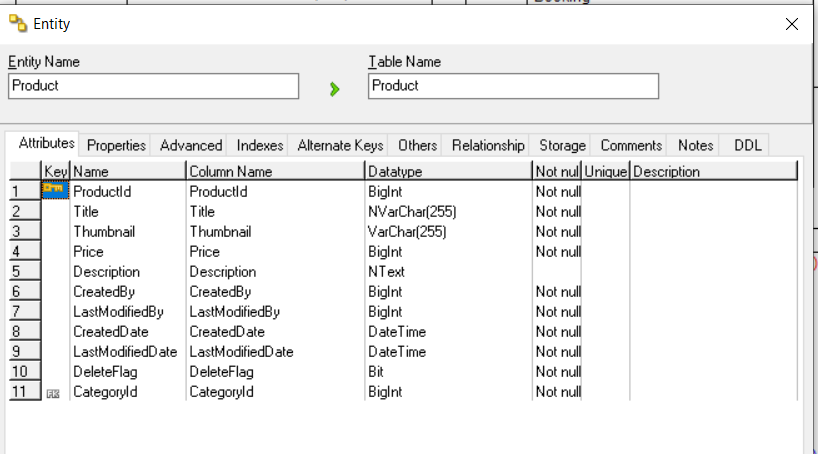
Service: Bảng lưu thông tin các dịch vụ của khách sạn

* ServiceId: id
* Title tiêu đề dịch vụ
* Thumbnail: ảnh mô tả dịch vụ
* Price: giá dịch vụ kèm theo
* **Bảng Sale** (Lưu thông tin khuyến mãi)



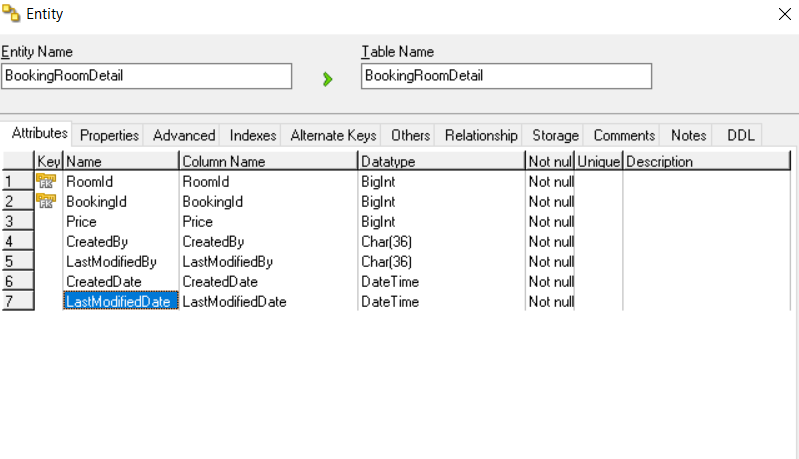
Sale: Bảng lưu thông tin khuyến mại tương ứng với mỗi phòng thuộc khách sạn

* SaleId: Khóa chính, dùng để xác định các khuyến mãi
* DayStart: Ngày áp dụng
* DayEnd: Ngày hết hạn
* SalePercent: Phần trăm giảm giá
* CreatedBy: Người tạo khuyến mãi
* LastModifiedBy: Người sửa đổi khuyến mãi cuối cùng
* CreatedDate: Ngày tạo khuyến mãi
* LastModifiedDate: Ngày sửa đổi cuối cùng
* DeleteFlag: Sử dụng để xóa mềm các khuyến mãi
* **Bảng Product** (Lưu thông tin sản phẩm)



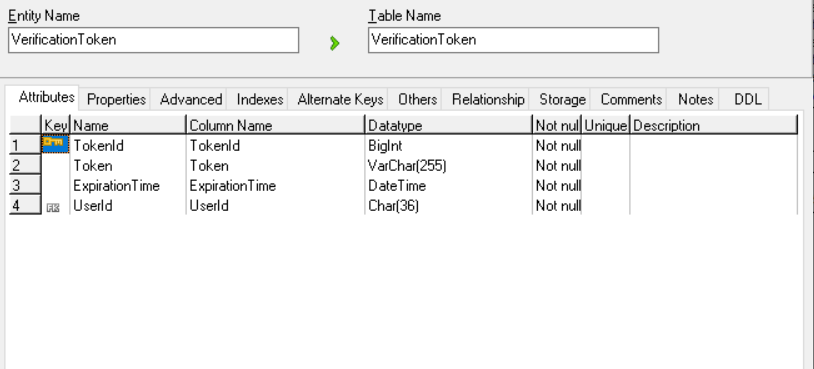
Product: Bảng lưu thông tin các sản phẩm của dịch vụ

* ProductId : Mã sản phẩm
* Name: Tên sản phẩm
* Thumbnail: ảnh sản phẩm
* Price: giá sp
* Description: Mô tả sản phẩm.
* **Bảng BookingRoomDetail ( Lưu thông tin chi tiết đặt phòng)**



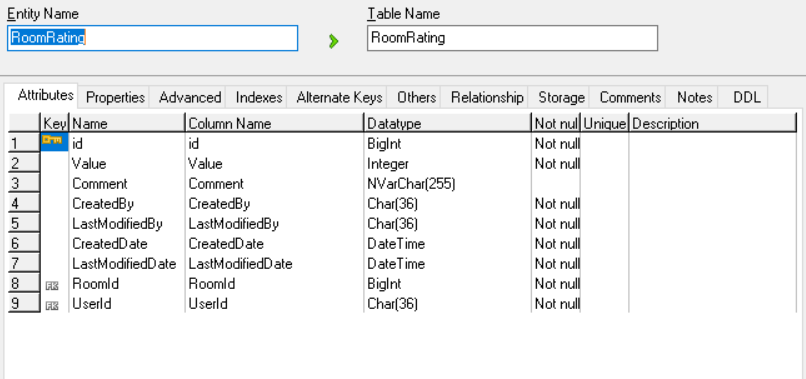
BookingServiceDetail: Bảng lưu các dịch vụ mà phòng đó đã dùng

* ServiceId…
* Price: Giá dịch vụ (Tương tự phòng, lưu giá tại thời điểm booking, giá dịch vụ có thể thay đổi theo tgian)
* Amount: Số lượng sử dụng
* **Bảng Verification Token : Lưu token để xác thực tài khoản or để xác nhận khi nhấn quên mật khẩu**



VerificationToken: Lưu token để xác thực tài khoản hoặc để xác nhận khi nhấn quên mật khẩu

* TokenId: mã id của mã xác thực tài khoản hoặc quên mật khẩu
* Token: mã xác thực tài khoản hoặc quên mật khẩu
* Expiration Time: Thời gian hết hạn (now + 3 phút)
* **Bảng RoomRating : Lưu đánh giá của khách hàng khi đặt phòng tại hệ thống**



## Thiết kế giao diện

# KẾT QUẢ ĐẠT ĐƯỢC

## Giao diện chương trình

### Giao diện phía khách hàng

#### Giao diện trang chủ

#### Màn hình đăng nhập

### Màn hình trang quản lý

#### Màn hình đăng nhập

# KẾT LUẬN

|  |  |
| --- | --- |
|  | Hà Nội, ngày tháng 5 năm 2023  Sinh viên thực hiện  **Nguyễn Thanh Hương** |

# TÀI LIỆU THAM KHẢO

* Giáo trình môn học Phân tích thiết kế hệ thống.
* Giáo trình môn học Thiết kế cơ sở dữ liệu.