**Software Requirements Specification**

**ONLINE HOTEL BOOKING SYSTEM**

1. **Mô tả vấn đề:**

**ONLINE HOTEL BOOKING SYSTEM** là website đặt phòng online giúp việc đặt phòng thuận tiện và nhanh chống, phần mềm hỗ trợ việc quản lý đặt phòng khách sạn chính xác nhất. Hệ thống giúp việc đặt phòng không tốn quá nhiều thời gian

1. **Các yêu cầu hệ thống:**
2. **Yêu cầu chức năng:**

* Khách sạn có các loại phòng cần quản lý là: single (cho một người), double (cho 2 người), và family (cho nhiều người).
* Khách hàng (Customer) có thể vào trang web của khách sạn để kiểm tra yêu cầu đặt phòng của mình (số lượng từng loại phòng muốn đặt, ngày nhận phòng, số ngày ở) khách sạn có đáp ứng được hay không.
* Khách hàng (Customer) có thể yêu cầu đặt phòng trước (reservation) trực tiếp với khách sạn. Thông tin mà khách hàng phải cung cấp khi đăng ký đặt phòng để nhân viên điều hành nhập vào hệ thống bao gồm: Tên khách hàng, số điện thoại liên lạc, số lượng từng loại phòng khách hàng muốn đặt, ngày nhận phòng, số ngày ở. Hệ thống sẽ kiểm tra các phòng có sẵn có đáp ứng nhu cầu của khách không để thông báo cho khách hàng.Ngòai ra, khách hàng phải cung cấp thông tin chi tiết của các khách hàng sẽ nghỉ ở khách sạn bao gồm: tên, địa chỉ, số điện thoại, ...
* Khi khách hàng đến nhận phòng, nhân viên điều hành phải thực hiện xử lý nhận phòng (check in): ghi nhận ngày nhận phòng thực sự cho từng khách vào hệ thống để làm cơ sở tính hóa đơn thanh toán cho khách hàng sau này.
* Khi khách hàng đến trả phòng, nhân viên điều hành phải thực hiện xử lý trả phòng (check out): ghi nhận ngày trả phòng, xử lý thanh toán và in hóa đơn thanh toán cho khách hàng.

1. **Yêu cầu phi chức năng:**

* Hệ thống phải có khả năng xử lý nhiều giao dịch một lần.
* Hệ thống phải cung cấp cho khách hàng dịch vụ đặt phòng trực tuyến 24 / 7 giờ.
* Hệ thống phải hỗ trợ hầu hết tất cả các trình duyệt (Internet Explorer, Safari, Chrome và Firefox).
* Hệ thống sẽ có thể chuyển đổi giá từ R.S sang USD.
* Hệ thống có chức năng bảo mật và phân quyền.
* Người sử dụng chương trình: được cấp username và password và được phân quyền sử dụng chương trình, giúp người quản lý có thể theo dõi, kiểm soát được chương trình, tránh các hợp sửa đổi thông tin không thuộc phạm vi quyền hạn
* Password của người sử dụng được mã hóa trước khi ghi vào dữ liệu và người sử dụng có thể đổi password của mình

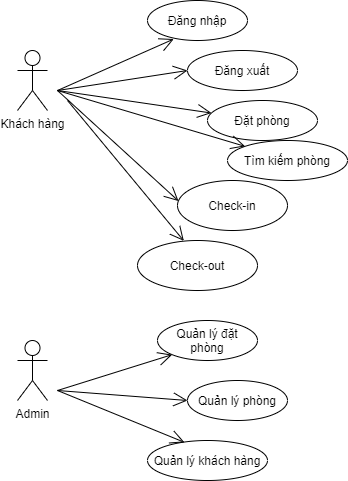
1. **Công nghệ sử dụng:**

**Fontend:** Bootstrap, JavaScript, HTML,CSS

**Backend:** Spring MVC

**Database:** MySql

1. **Use Case Diagram:**



1. **Đặc tả yêu cầu:**

1. Use Case: Tìm kiếm phòng:

|  |  |
| --- | --- |
| **Use Case** | Tìm kiếm phòng |
| **Actor** | Khách hàng |
| **Description** | Khách hàng tìm kiếm phòng |
| **Precondition** | Khách hàng truy cập vào website đặt phòng khách sạn |
| **Basic Path** | **1** Khách hàng truy cập vào website |
| **2** Chọn ngày check-in, check-out Khách hàng chọn loại phòng  Loại phòng: phòng đơn, phòng đôi, phòng gia đình |
| **3** Khách hàng chọn loại phòng  Loại phòng: phòng đơn, phòng đôi, phòng gia đình |
| **3** Click button Tìm kiếm |
| **4** Hệ thống hiển thị danh sách phòng |
| **Alternative Paths** | Ở bước 4 nếu chọn button “Thoát” thì quay lại bước 2 |
| **Post condition** | Hệ thống hiện thị danh sách phòng trống với ngày check-in, check-out |

2 . Use case: Đặt phòng:

|  |  |
| --- | --- |
| **Use Case** | Đặt phòng |
| **Actor** | Khách hàng |
| **Description** | Khách hàng đặt phòng khách sạn |
| **Trigger** | Khách hàng truy cập website đặt phòng khách sạn và tìm kiếm phòng |
| **Precondition** | Khách hàng phải đăng ký là thành viên và đăng nhập vào hệ thống |
| **Basic Path** | **1**  Click button “Đặt phòng” trong danh sách phòng trống |
| **2**  Hệ thống hiện thị form Đặt phòng gồm thông tin tên khách hàng,số điện thoại, địa chỉ,ngày check-in, check-out |
| **3**  Người dùng nhập thông tin vào form |
| **4** Hệ thống kiểm tra tính hợp lệ của thông tin |
|  | **5** Nếu hợp lệ thì chấp nhận đặt phòng và hiển thị thông báo đặt thành công |
| **Alternative Paths** | Ở bước 3, người dùng chưa đăng nhập vào hệ thống thì báo lỗi và yêu cầu đăng nhập hoặc đăng ký làm thành viên nếu chưa có tài khoản  Ở bước 5, thông tin không hợp lệ thì báo lỗi và yêu cầu nhập lại |
| **Post-Conditions** | Thành công: thông tin đặt phòng được lưu vào cơ sở dữ liệu, số phòng trống bị trừ và hiển thị thông báo thành công  Thất bại: Trạng thái của hệ thống không thay đổi |

3. Use case: Quản lý phòng

|  |  |
| --- | --- |
| **Use Case Name** | Quản lý phòng |
| **Actor** | Toàn bộ hệ thống |
| **Description** | Tìm kiếm thông tin phòng của khách sạn |
| **Trigger** | Khách hàng truy cập website đặt phòng khách sạn |
| **Precondition** | Website đã hiển thị danh sách tìm kiếm |
| **Basic Path** | 1. Khách hàng chọn cách tìm kiếm trên trang web.Những lựa chọn có thể là về giá, loại, vị trí và từ khóa |
|  | 1. Nếu được tìm kiếm bởi từ khóa hệ thống sẽ tạo ra và hiện thị danh sách những thông tin liên quan đến từ khóa trong cơ sở dữ liệu. Trong trường hợp khách sạn có nhiều phòng, mỗi phòng đã có trong danh sách 2. Khách hàng chọn phòng 3. Hệ thống tạo và hiển thị danh sách bao gồm tất cả các phòng trong cơ sở dữ liệu 4. Khách hàng chọn phòng 5. Hệ thống hiển thị chi tiết về phòng 6. Khách hàng chọn đặt phòng hoặc trả về danh sách phòng hoặc danh sách trước đó. |
| **Alternative Paths** | Ở bước 2, nếu khách hàng chọn tìm kiếm theo loại phòng thì hệ thống sẽ tạo và hiện thị một danh sách tất cả các loại phòng trong cơ sở dữ liệu   1. Khách hàng chọn một loại 2. Hệ thống tạo và hiển thị một danh sách tất cả phòng theo loại phòng đã chọn trong cơ sở dữ liệu. Quay về bước 5   Ở bước 2, nếu khách hàng chọn tìm kiếm theo từ khóa, hệ thống sẽ hiện thị một hộp thoại để nhập từ khóa hoặc câu   1. Khách hàng nhập một từ khóa 2. Hệ thống tìm kiếm chi tiết của phòng với từ khóa đã nhập, sau đó tạo và hiển thị một danh sách tất cả các phòng trong cơ sở dữ liệu.Quay lại bước 5 |
| **Post-Conditions** | Thành công: Hiển thị danh sách theo yêu cầu  Thất bại: Hiện thị toàn bộ danh sách trong cơ sở dữ liệu |
| **Exception Paths** | Khách hàng có thể từ bỏ tìm kiếm bất cứ lúc nào. |

4. Use case : Quản lý khách hàng

|  |  |
| --- | --- |
| **Use Case Name** | Quản lý phòng |
| **Actor** | Toàn bộ hệ thống |
| **Description** | Use case này cho phép actor(nhân viên, quản lý, giám đốc) duy trì thông tin khách hàng trong hệ thống. Bao gồm thêm, hiệu chỉnh và xóa khách hàng ra khỏi hệ thống. |
| **Trigger** | Actor(nhân viên, quản lý, giám đốc) truy cập hệ thống |
| **Precondition** | Use case này bắt đầu khi actor  muốn thêm, thay đổi, và/hoặc xóa thông tin giáo sư trong hệ thống. |
| **Basic Path** | * + 1. Hệ thống yêu cầu actor chọn chức năng muốn thực hiện (Thêm thông tin khách hàng, Cập nhật thông tin khách hàng, hoặc Xóa thông tin khách hàng).     2. Sau khi actor cung cấp thông tin được yêu cầu, một trong các luồng phụ sau được thực hiện.   Nếu actor chọn “Thêm thông tin khách hàng”, luồng phụ **Thêm khách hàng** được thực hiện.  Nếu cán bộ đào tạo chọn “Cập nhật thông tin khách hàng”, luồng phụ **Cập nhât thông tin khách hàng** được thực hiện.  Nếu cán bộ đào tạo chọn “Xóa thông tin khách hàng”, luồng phụ **Xóa khách hàng** được thực hiện. |
|  | Ở bước 2, nếu actor chọn “Thêm thông tin khách hàng”, luồng phụ **Thêm khách hàng** được thực hiện.Khách hàng chọn một loại   1. Hệ thống yêu cầu actor nhập vào các thông tin của khách hàng. Bao gồm: Họ tên CMND Địa chỉ Điện thoại   Email   1. Sau khi actor cung cấp thông tin được yêu cầu, hệ thống sẽ phát sinh và gán một số ID độc nhất cho khách hàng này. Khách hàng này được thêm vào hệ thống. 2. Hệ thống cung cấp cho actor số ID của khách hàng mới.   Ở bước 2, nếu actor chọn “Cập nhật thông tin khách hàng”, luồng phụ **Cập nhât thông tin khách hàng** được thực hiện.   1. Hệ thống yêu cầu actor nhập vào số ID của khách hàng hoặc tìm kiếm khách hàng. 2. Actor nhập số ID khách hàng. Hệ thống truy xuất và hiển thị thông tin của khách hàng này. 3. Actor thay đổi một số thông tin của khách hàng. Gồm bất cứ thông tin nào được chỉ ra trong luồng phụ **Thêm khách hàng**. 4. Sau khi actor cập nhật xong các thông tin cần thiết, hệ thống cập nhật mẩu tin của khách hàng này.   Ở bước 2, nếu actor chọn “Xóa thông tin khách hàng”, luồng phụ **Xóa khách hàng** được thực hiện   1. Hệ thống yêu cầu actor nhập vào số ID của khách hàng hoặc tìm kiếm khách hàng. 2. Actor nhập số ID khách hàng. Hệ thống truy xuất và hiển thị thông tin của khách hàng này: 3. Hệ thống nhắc người dùng xác nhận thao tác xóa khách hàng. 4. Actor xác nhận xóa. 5. Hệ thống xóa thông tin của khách hàng này ra khỏi hệ thống. |
| **Alternative Paths** | **Không tìm thấy khách hàng**Nếu trong luồng phụ **Cập nhật thông tin khách hàng** hoặc **Xóa khách hàng** không tồn tại khách hàng nào có số ID được nhập vào thì hệ thống sẽ hiển thị một thông báo lỗi. Actor có thể nhập một số ID khác hoặc hủy bỏ thao tác, lúc này use case kết thúc.**Thao tác xóa bị hủy**Nếu trong luồng phụ **Xóa khách hàng**actor quyết đinh không xóa khách hàng này nữa, thao tác xóa bị hủy và **Dòng sự kiện chính** được bắt đầu lại từ đầu. |
| **Post-Conditions** | Nếu use case thành công, thông tin khách hàng được thêm, cập nhật hoặc xóa khỏi hệ thống. Ngược lại, trạng thái của hệ thống không thay đổi. |
| **Exception Paths** | Không có. |

**5. Use case Quản lý đặt phòng**

|  |  |
| --- | --- |
| **Use Case Name** | Quản lý thuê phòng |
| **Actor** | Khách hàng,nhân viên lễ tân |
| **Description** | Đặt phòng trực tuyến |
| **Trigger** | Khách hàng, nhân viên truy cập website của khách sạn |
| **Precondition** | Khách hàng, nhân viên phải có tài khoản của khách sạn |
| **Basic Path** | 1. Khách hàng chọn đặt phòng từ hệ thống  2. Hiển thị danh sách các phòng còn trống  3. Chọn phòng muốn thuê  4. Chọn lập phiếu thuê  5. Người dùng cung cấp đầy đủ thông tin vào phiếu thuê: Họ tên KH, địa chỉ, cmnd, sđt..  6. Nếu đồng ý thì lưu, nếu không có thể đổi phòng hoặc không thuê nữa. |
| **Alternative Paths** | 1. Nếu có sai sót về thông tin ngày thuê hệ thống sẽ thông báo. vd: ngày trả phòng trước ngày thuê… 2. Nếu thông tin cập nhật chính xác, thông tin phiếu thuê sẽ được lưu vào CSDL. |
| **Postcondition** |  |

**6. Use case: Login ( Đăng nhập):**

Use case này mô tả cách một người dùng đăng nhập vào Hệ thống quản lý khách sạn.

|  |  |
| --- | --- |
| **Use Case Name** | Login ( Đăng nhập) |
| **Actor** | Khách hàng, admin |
| **Description** | Đăng nhập vào hệ thống |
| **Trigger** | Actor truy cập vào trang web và thực hiện đăng nhập |
| **Precondition** | Use case này bắt đầu khi một actor muốn đăng nhập vào hệ thống |
| **Basic Path** | 1. Hệ thống yêu cầu actor nhập tên và mật khẩu. 2. Hệ thống kiểm tra tên và mật khẩu được nhập. 3. Thông báo thành công và cho phép actor đăng nhập hệ thống. |
| **Alternative Paths** | Ở bước 2, nếu hệ thống kiểm tra tên đăng nhập hay mật khẩu sai, hệ thống sẽ hiển thị thông báo lỗi, người dùng có thể:   * Chọn quay trở về bước 1 và đăng nhập lại * Hủy bỏ việc đăng nhập   Nếu người dùng chọn đăng nhập lại:  hệ thống chỉ cho phép nhập sai tên hoặc mật khẩu tối đa 3 lần, sau đó tài khoản sẽ bị tạm khóa, người dùng có thể liên hệ người quản trị để lấy lại tài khoản.  Nếu người dùng hủy bỏ việc đăng nhập, use case kết thúc. |
| **Postcondition** | Nếu use case thành công, actor lúc này đã đăng nhập vào hệ thống. Nếu không trạng thái hệ thống không thay đổi |

**7. Use case Logout( Đăng xuất)**

Use case này mô tả cách một người dùng đăng xuất khỏi Hệ thống quản lý khách sạn.

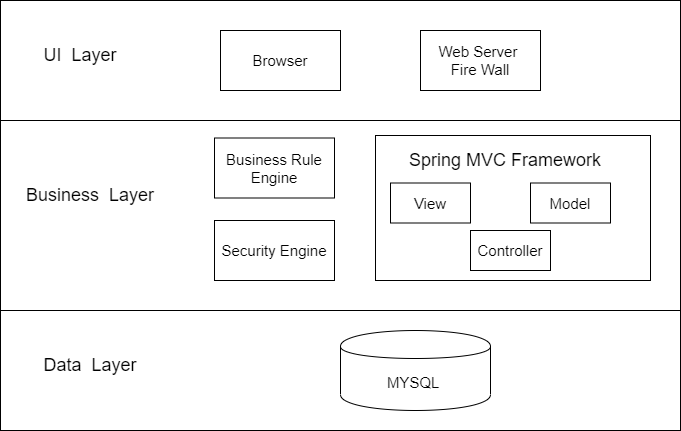
|  |  |
| --- | --- |
| **Use Case Name** | Logout ( Đăng xuất) |
| **Actor** | Khách hàng, admin |
| **Description** | Đăng xuất khỏi hệ thống |
| **Trigger** | Actor đã truy cập vào trang web và thực hiện đăng xuất |
| **Precondition** | Actor đã đăng nhập vào hệ thống |
| **Basic Path** | 1.Actor thực hiện chức năng đăng xuất khỏi hệ thống.  2. Hệ thống hiển thị yêu cầu xác nhận từ actor  3. Actor dùng xác nhận đăng xuất  4. Hệ thống đăng xuất tài khoản actor khỏi hệ thống. |
| **Alternative Paths** | Ở bước 3, nếu actor không xác nhận đăng xuất thì hệ thống sẽ giữ nguyên hiện trạng và người dùng tiếp tục ở lại trang web |
| **Postcondition** |  |

**8. Use case Thanh toán**

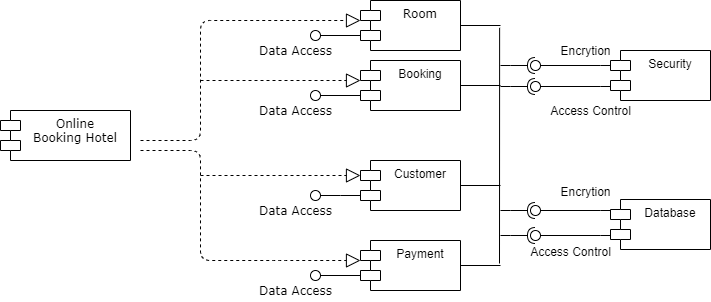
Use case này cho phép nhân viên lập hóa đơn tính tiền khi khách hàng trả phòng.

|  |  |
| --- | --- |
| **Use Case Name** | Thanh toán |
| **Actor** | Nhân viên |
| **Description** | Thanh toán cho khách hàng |
| **Trigger** | Khách hàng đã nhận phòng |
| **Precondition** | Nhân viên phải đăng nhập vào hệ thống trước khi thực hiện thanh toán |
| **Basic Path** | 1.Nhân viên tiếp tân sẽ ghi nhận lại thông tin về số phòng, CMND của khách hàng.  2. Dựa vào đó sẽ tra cứu( số ngày đã ở, các loại dịch vụ khách hàng yêu cầu).  3. Hệ thống sẽ tính tiền tự động và in kết quả vào hóa đơn cho khách hàng kiểm tra lại. Nếu không có gì sai sót thì nhân viên sẽ lưu dữ liệu xuống database. |
| **Alternative Paths** |  |
| **Postcondition** | Nếu usecase thực hiện thành công hóa đơn sẽ được lưu xuống database nếu không hệ thống không thay đổi. |

1. **Kiến trúc hệ thống:**

****

1. **Kiến trúc phần mềm:**

****

1. **Activity Diagram:**

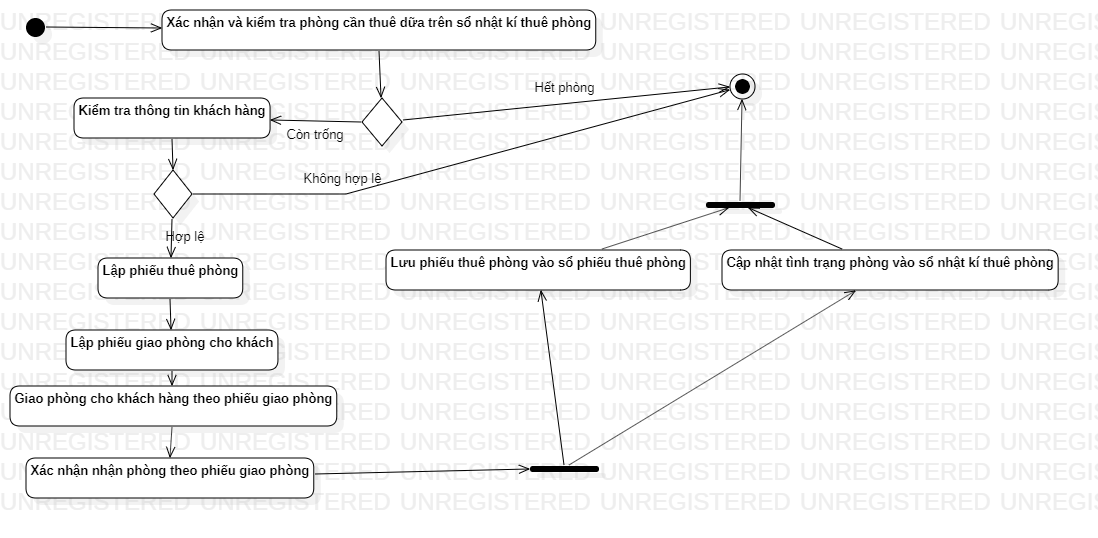
a. Tìm kiếm phòng:



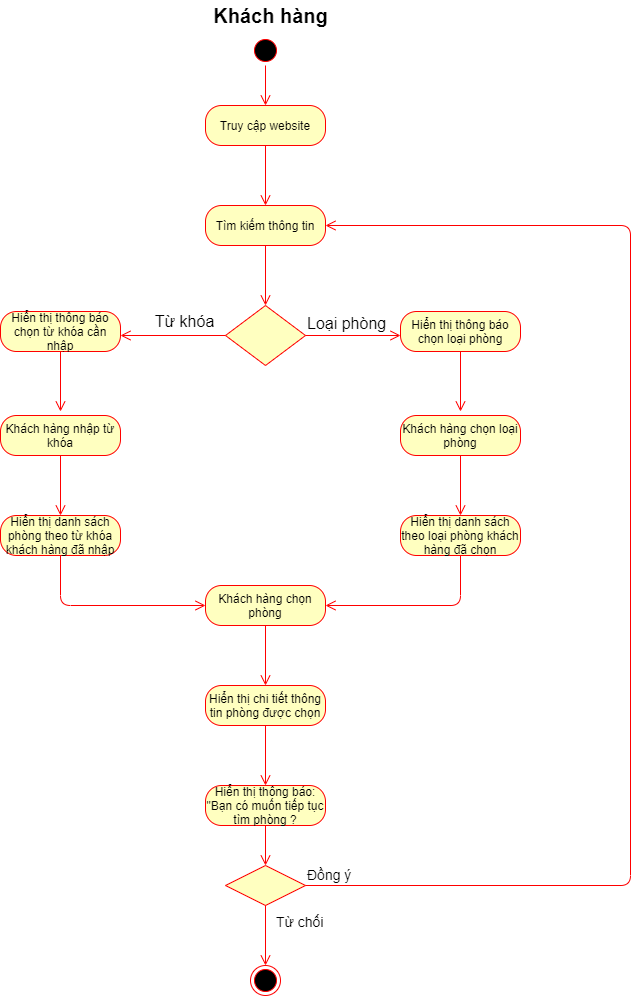
b. Đặt phòng:



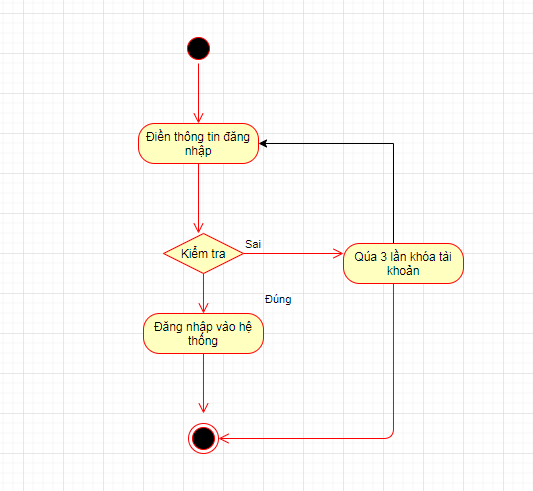
c. Quản lý đặt phòng:



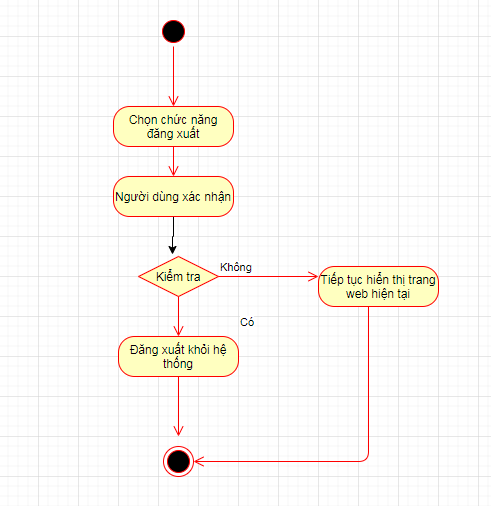
d. Quản lý phòng



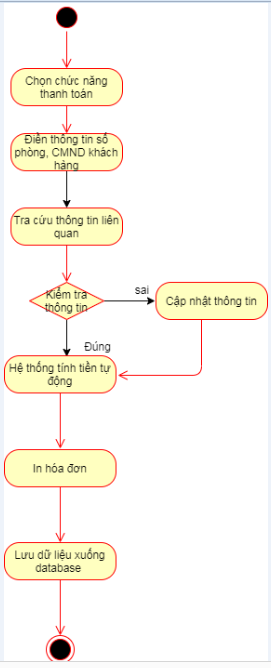
e.Đăng nhập:



f. Đăng xuất:



g.Thanh toán:



1. **Sequence Diagram:**

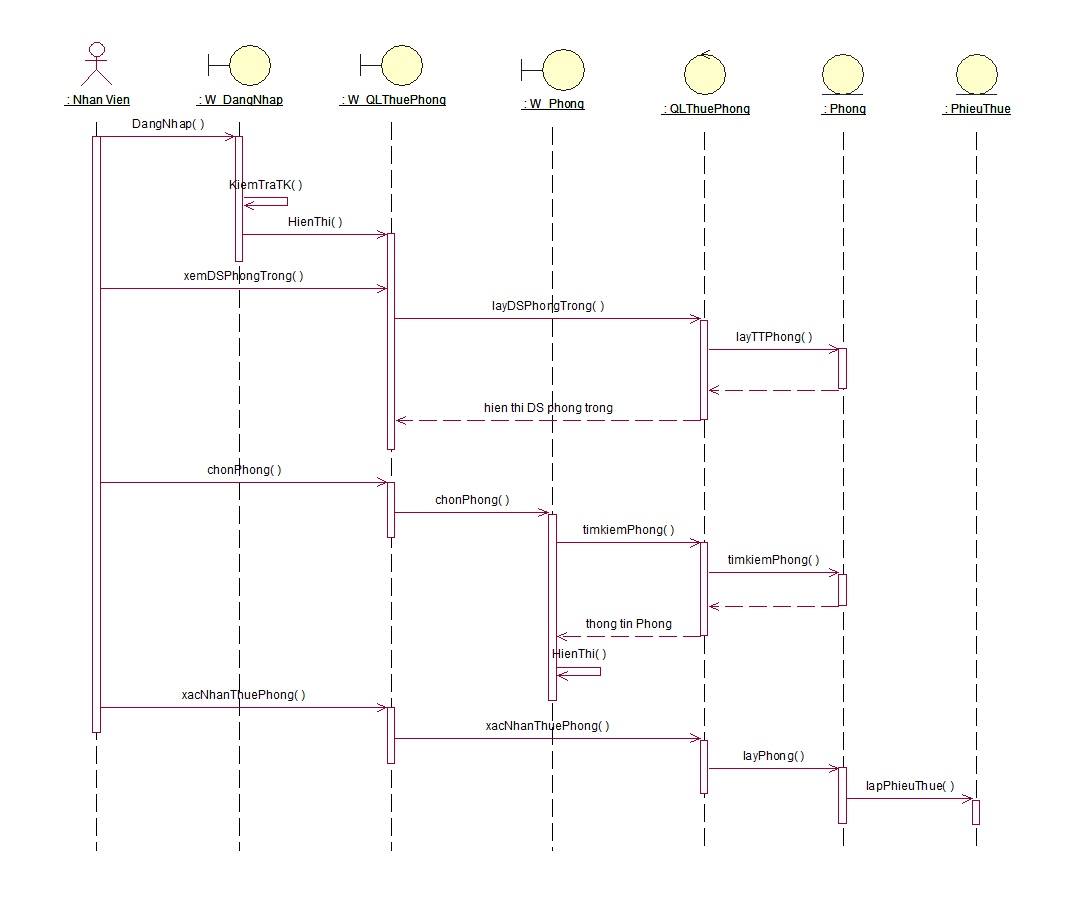
a. Tìm kiếm phòng:



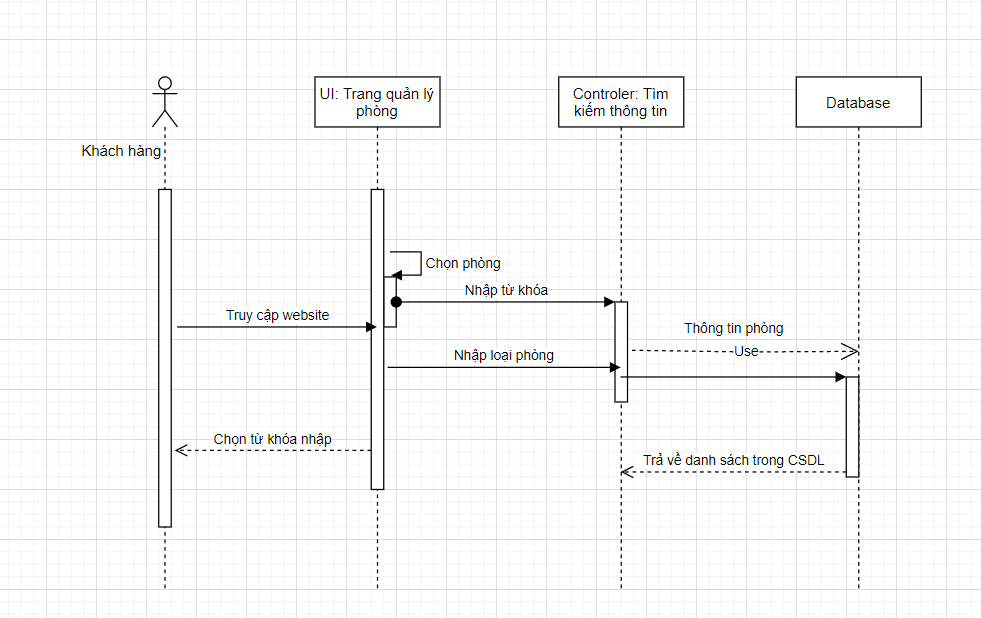
b. Đặt phòng:



c. Quản lý đặt phòng:

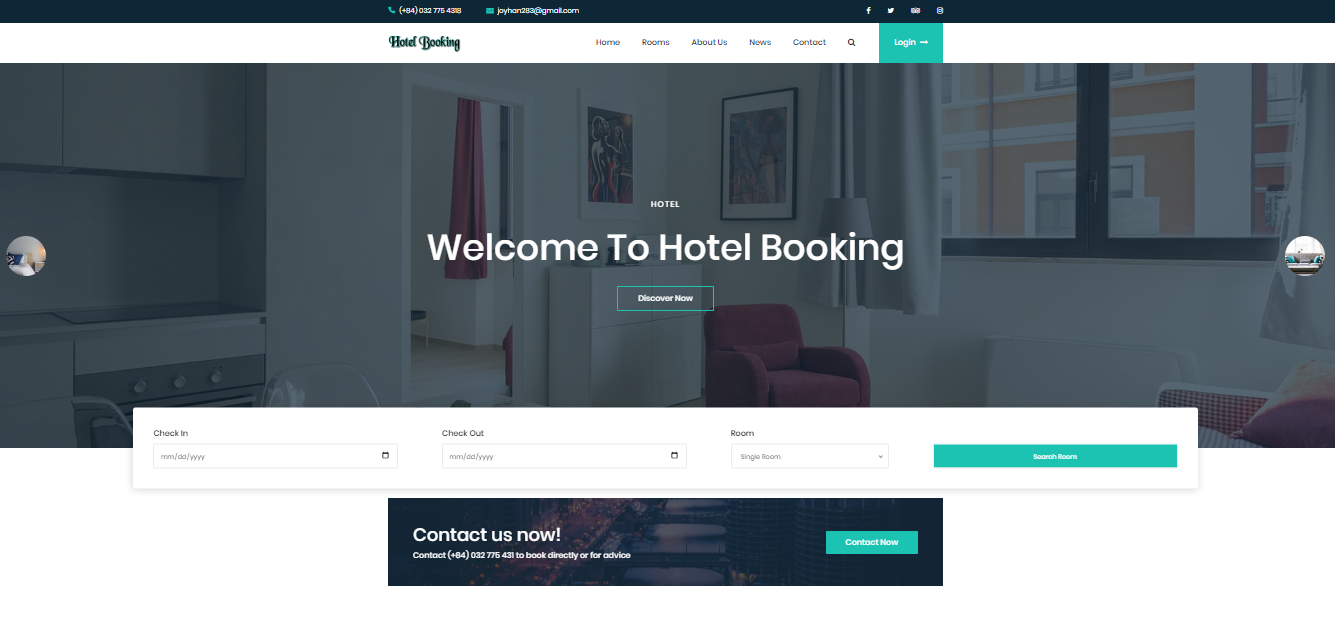


d. Quản lý Phòng:

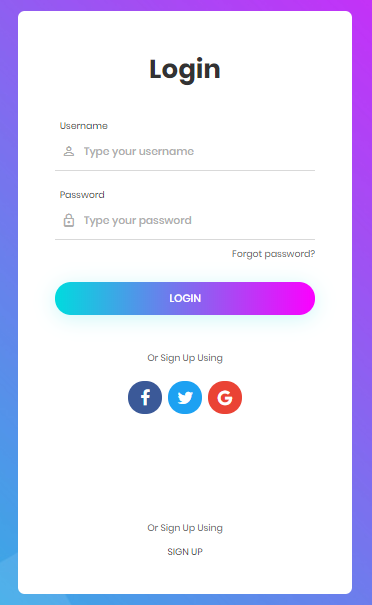


1. **Giao diện:**

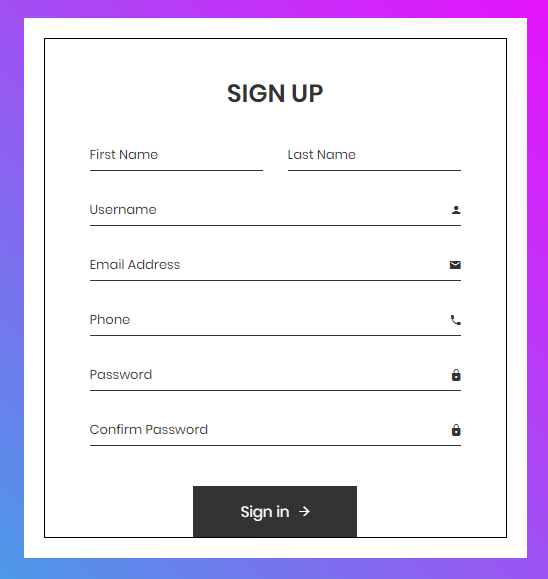
a. Trang chủ:



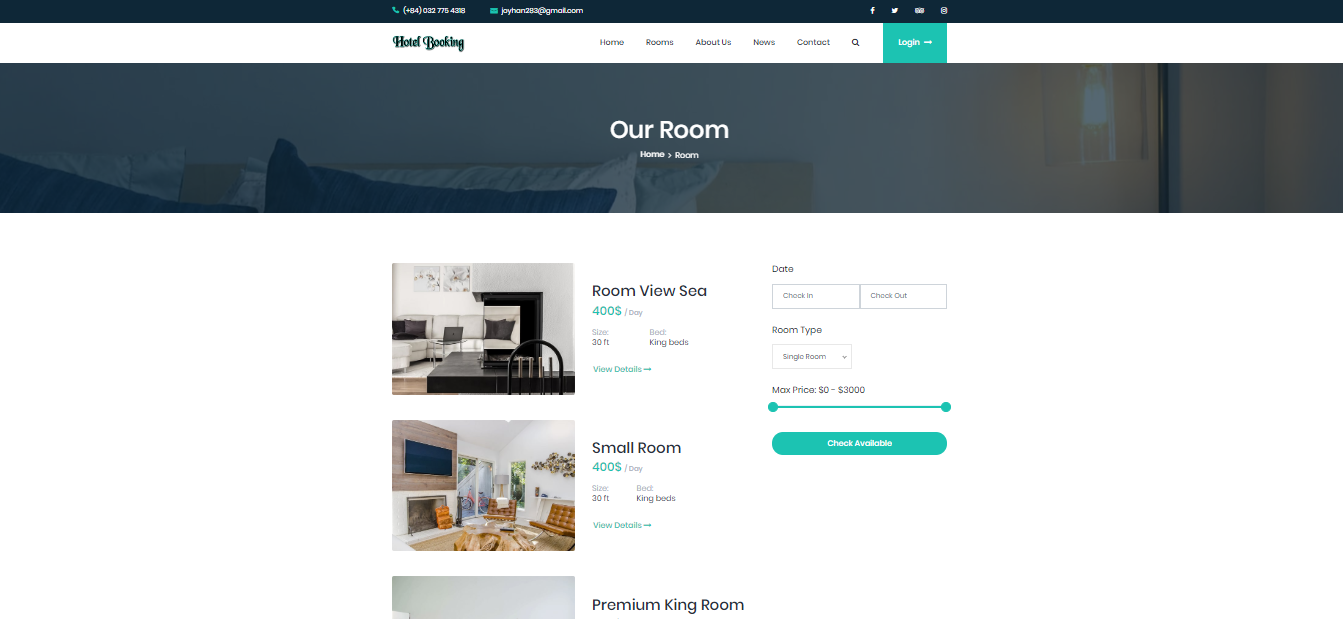
b. Login:



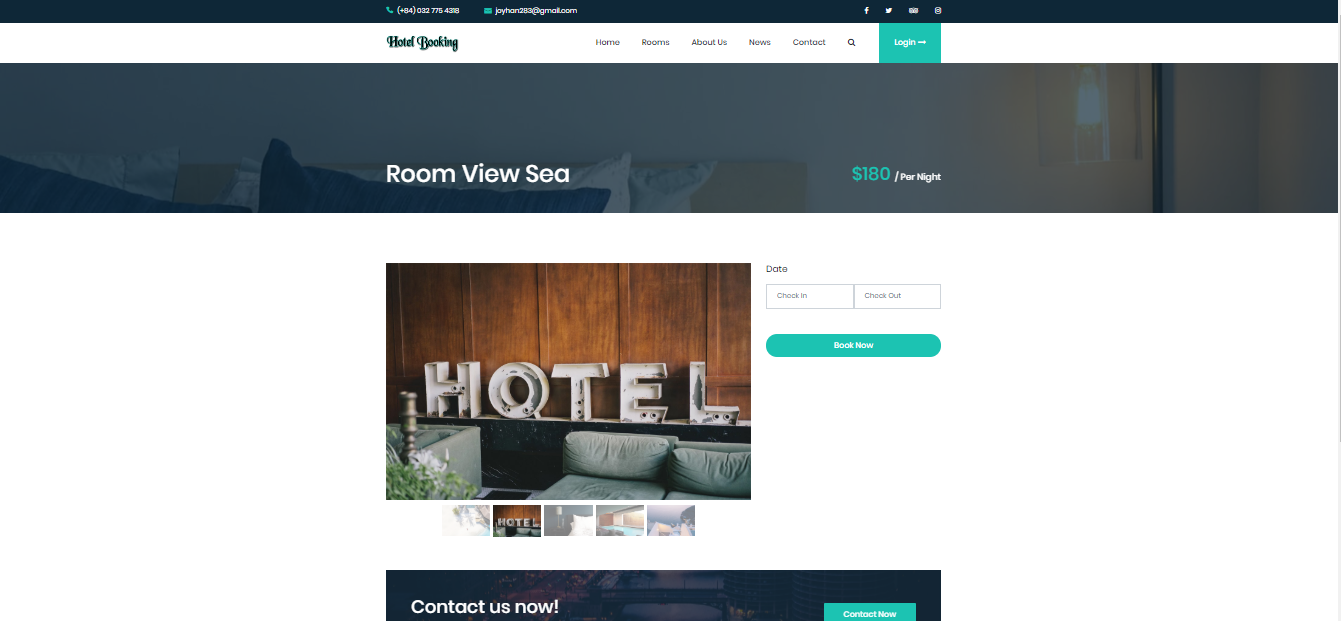
c. Sign up:



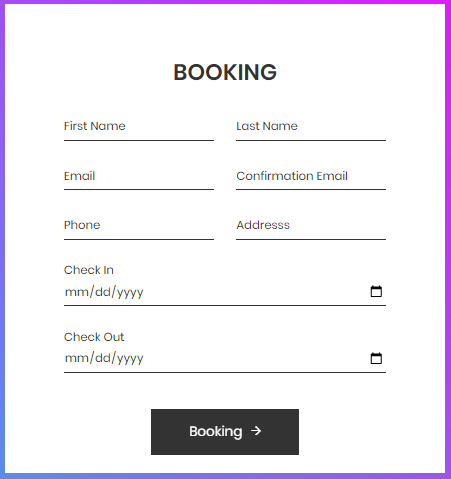
d. Search Room:



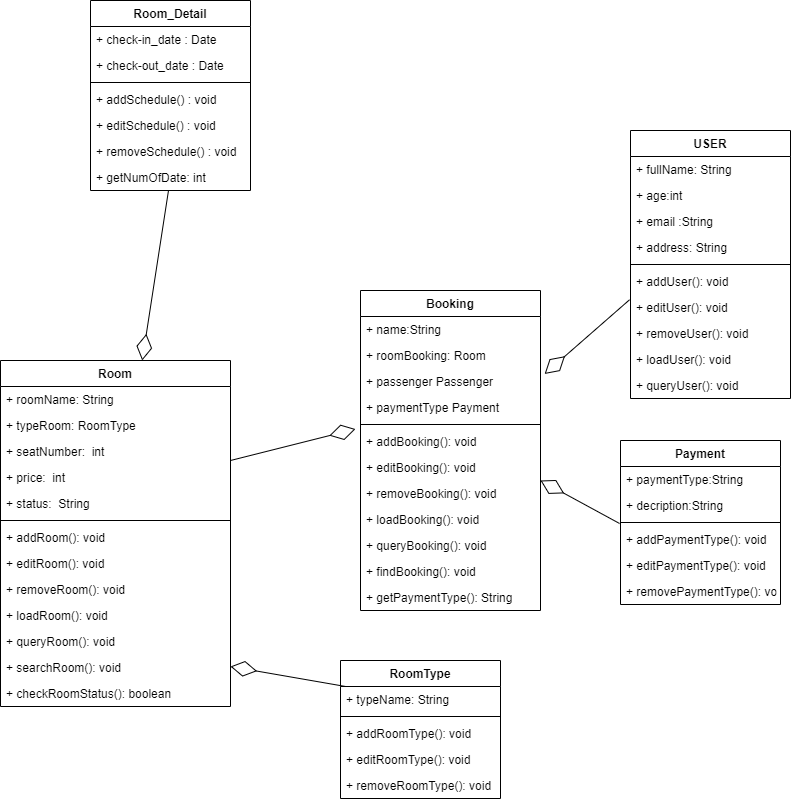
e. Room Detail:



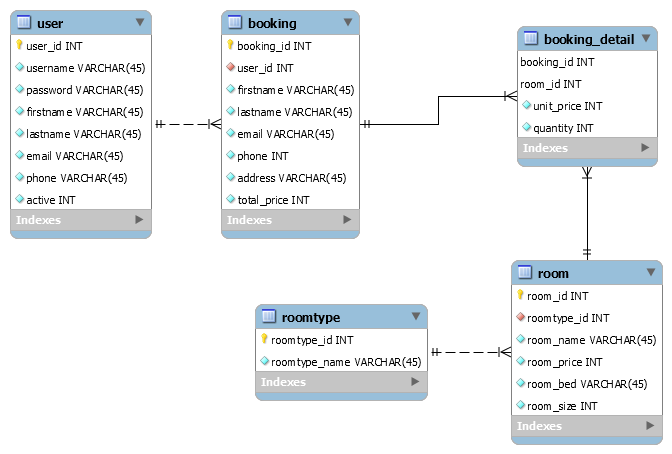
f. Booking Form:



1. **Class Diagram:**

****

1. **Database:**

****