

CHƯƠNG 1. TỔNG QUAN

1.1. Giới thiệu tổng quát:

Đề tài xây dựng Ứng dụng quản lý và bán phòng khách sạn, được phát triển theo mô hình ASP.NET Core MVC.

Hệ thống cho phép khách hàng tra cứu, đặt phòng theo ngày, quản lý đặt phòng và thanh toán. Ngoài ra, còn hỗ trợ lễ tân trong việc check-in/check-out, quản lý buồng phòng, cập nhật trạng thái phòng và hỗ trợ người quản lý theo dõi báo cáo doanh thu, tỷ lệ lấp đầy, cùng các gói giá (RatePlan) linh hoạt.

1.2. Mục đích của đề tài:

- Xây dựng hệ thống đặt phòng khách sạn trực tuyến giúp người dùng dễ dàng tìm, xem, và đặt phòng.
- Chuẩn hóa quy trình vận hành khách sạn (quản lý phòng, khách, lễ tân, buồng phòng).
- Cung cấp giao diện quản trị trực quan, hỗ trợ quản lý dễ dàng.
- Đảm bảo dữ liệu tập trung, an toàn và khả năng mở rộng trong tương lai.

1.3. Đối tượng và phạm vi nghiên cứu:

- Đối tượng:** Khách sạn nhỏ và vừa; khách du lịch sử dụng dịch vụ phòng.
- Phạm vi:** Quản lý **mảng phòng**, không bao gồm tour du lịch hay dịch vụ mở rộng.
- Frontend:** Giao diện web ASP.NET MVC.
- Backend:** ASP.NET Core + EF Core + SQL Server.
- Tính năng chính:** tìm phòng, đặt phòng, thanh toán mô phỏng, quản lý trạng thái phòng, báo cáo doanh thu.

1.4. Phương pháp nghiên cứu:

- **Phân tích yêu cầu:** sử dụng chuẩn IEEE 830/29148 cho SRS.
- **Thiết kế hệ thống:** use case, activity, sequence, ERD.
- **Phát triển:** ASP.NET Core MVC, EF Core (Code-first), Identity.
- **Kiểm thử:** test case chức năng, xác nhận đúng quy trình đặt phòng.

1.5. Nội dung thực hiện của đề tài:

1.5.1. Phân tích yêu cầu:

- Xác định các nhóm người dùng (Khách, Lễ tân, Quản lý, Buồng phòng, Admin) cùng các yêu cầu chức năng tương ứng.

1.5.2. Thiết kế hệ thống:

- Phân tích, thiết kế Use case, Activity, Sequence, ERD.
- Thiết kế cơ sở dữ liệu phục vụ các nghiệp vụ quản lý phòng, đặt phòng, báo cáo.

1.5.3. Phát triển ứng dụng:

Phát triển theo mô hình MVC với các module chính:

- Tìm và đặt phòng
- Quản lý phòng, buồng phòng
- Báo cáo và quản lý giá

1.5.4. Tối ưu và kiểm thử:

Kiểm thử đơn vị và tích hợp, tối ưu hiệu năng truy vấn.

1.5.5. Hướng dẫn sử dụng:

Tài khoản mẫu: admin/manager, lê tân, buồng phòng.

Kết nối CSDL thông qua chuỗi kết nối để tương tác với CSDL bên dưới.

1.6. Môi trường ứng dụng:

- **Ngôn ngữ:** C#, Razor, SQL,HTML,CSS,JS.
- **Framework:** ASP.NET Core MVC, Entity Framework Core.
- **Hệ quản trị CSDL:** MySql
- **Công cụ:** Visual Studio, MySql WorkBench
- **Hệ điều hành:** Windows 10/11.

1.7. Kết quả của thực hiện:

- Ứng dụng hoàn chỉnh hỗ trợ quy trình đặt – quản lý – báo cáo phòng khách sạn.
- Giao diện thân thiện, dễ mở rộng, hỗ trợ nhiều vai trò người dùng.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Giới thiệu ngôn ngữ và công nghệ. (Backend + Database – Fontend)

2.1. ASP.NET Core MVC

- ASP.NET Core MVC là framework mã nguồn mở của Microsoft, giúp xây dựng ứng dụng web theo mô hình **Model – View – Controller**.

- **Model:** Chứa lớp dữ liệu, ánh xạ tới CSDL bằng Entity Framework Core.
- **View:** Hiển thị thông tin (Razor View Engine).
- **Controller:** Xử lý yêu cầu, điều hướng dữ liệu giữa Model và View.

2.2. Entity Framework Core

ORM (Object Relational Mapping) giúp thao tác CSDL bằng đối tượng C#, hỗ trợ:

- **Code-first migration** để tạo bảng tự động.
- **LINQ query** để truy vấn dễ dàng.
- **Tracking & caching** tự động giúp tối ưu hiệu năng.

2.3. MySql:

- Giới thiệu:

MySQL là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở phổ biến, hỗ trợ tốt cho các ứng dụng web. Trong đề tài này, MySQL được sử dụng để lưu trữ và quản lý toàn bộ dữ liệu của hệ thống như thông tin phòng, loại phòng, đặt phòng, thanh toán và báo cáo.

- Đặc điểm nổi bật:

- Hỗ trợ ngôn ngữ truy vấn SQL chuẩn, dễ tích hợp với ASP.NET Core thông qua Entity Framework Core (Pomelo Provider).
- Có khả năng đảm bảo toàn vẹn dữ liệu thông qua các ràng buộc khóa chính – khóa ngoại.

- InnoDB Engine hỗ trợ giao dịch (Transaction) và tính năng ACID đảm bảo an toàn dữ liệu.
- Hỗ trợ kiểu dữ liệu JSON, ENUM, giúp lưu trữ linh hoạt cấu hình và trạng thái.
- Khả năng mở rộng, tốc độ truy vấn cao và tương thích với hầu hết các hệ điều hành.

- Kết nối với ASP.NET Core:

Ứng dụng kết nối MySQL thông qua Pomelo.EntityFrameworkCore.MySql

Các truy vấn, thêm–xóa–sửa được thực hiện thông qua **LINQ** và **migrations** giúp quản lý phiên bản CSDL.

- Kết luận:

Việc sử dụng MySQL giúp hệ thống đảm bảo tốc độ truy cập nhanh, độ ổn định cao, dữ liệu tập trung và dễ bảo trì. Đây là lựa chọn phù hợp cho ứng dụng web quản lý phòng khách sạn quy mô nhỏ và vừa.

2.4. Mô hình MVC trong dự án:

Tổng quan:

Trong ứng dụng ASP.NET Core MVC, MySQL được sử dụng làm hệ thống lưu trữ dữ liệu trung tâm. Mô hình MVC (Model – View – Controller) giúp chia tách rõ ràng giữa giao diện hiển thị, xử lý nghiệp vụ, và dữ liệu, qua đó tăng khả năng mở rộng, dễ bảo trì.

Mô hình MVC:

Controllers/

 |—— HomeController.cs

 |—— BookingController.cs

 |—— RoomController.cs

Models/

 |—— Room.cs

 |—— Booking.cs

Views/

 |—— Booking/Index.cshtml

 |—— Booking/Checkout.cshtml

Luồng dữ liệu trong mô hình được triển khai như sau:

- **Model:** đại diện cho cấu trúc dữ liệu và ánh xạ trực tiếp tới các bảng trong MySQL (sử dụng **Entity Framework Core** để ORM – Object Relational Mapping).
- **Controller:** tiếp nhận yêu cầu từ người dùng, gọi đến các **Service hoặc DbContext** để truy xuất MySQL.
- **View:** hiển thị dữ liệu trả về từ Controller lên giao diện người dùng (Razor View).

Mô hình hoạt động chi tiết:

1. **Người dùng (Client):** Gửi yêu cầu HTTP (ví dụ: xem danh sách phòng, đặt phòng).
2. **Controller:** Nhận yêu cầu và gọi tới **Model** hoặc **Service layer** để xử lý logic.
3. **Model (qua DbContext):** Tương tác trực tiếp với **MySQL** bằng các câu truy vấn LINQ (được EF Core chuyển thành SQL).
4. **Cơ sở dữ liệu MySQL:** Trả về kết quả (danh sách phòng, thông tin booking, trạng thái thanh toán...).
5. **Controller** Xử lý kết quả, có thể áp dụng thêm logic nghiệp vụ (tính giá, lọc khuyến mãi).
6. **View (Razor Page):** Hiển thị dữ liệu kết quả ra giao diện web cho người dùng.

Ví dụ thực tế trong hệ thống:

Ví dụ khi người dùng thực hiện **tìm phòng trống**:

1. Người dùng nhập ngày check-in/check-out → gửi yêu cầu đến **BookingController**.
2. **BookingController** gọi **RoomService** → **DbContext.Rooms** để truy vấn **MySQL**.
3. **EF Core** tự động chuyển truy vấn **LINQ** sang **SQL** và lấy kết quả.
4. Kết quả (danh sách phòng, giá, RatePlan) được trả lại **View** để hiển thị.

CHƯƠNG 3. PHÂN TÍCH, THIẾT KẾ HỆ THỐNG

3.1. Mô tả bài toán:

Hệ thống "Ứng dụng quản lý và bán phòng khách sạn" được xây dựng nhằm giải quyết các vấn đề trong quy trình vận hành khách sạn vừa và nhỏ, bao gồm hai mảng nghiệp vụ chính: **Bán phòng trực tuyến (Booking)** và **Quản lý vận hành (PMS)**.

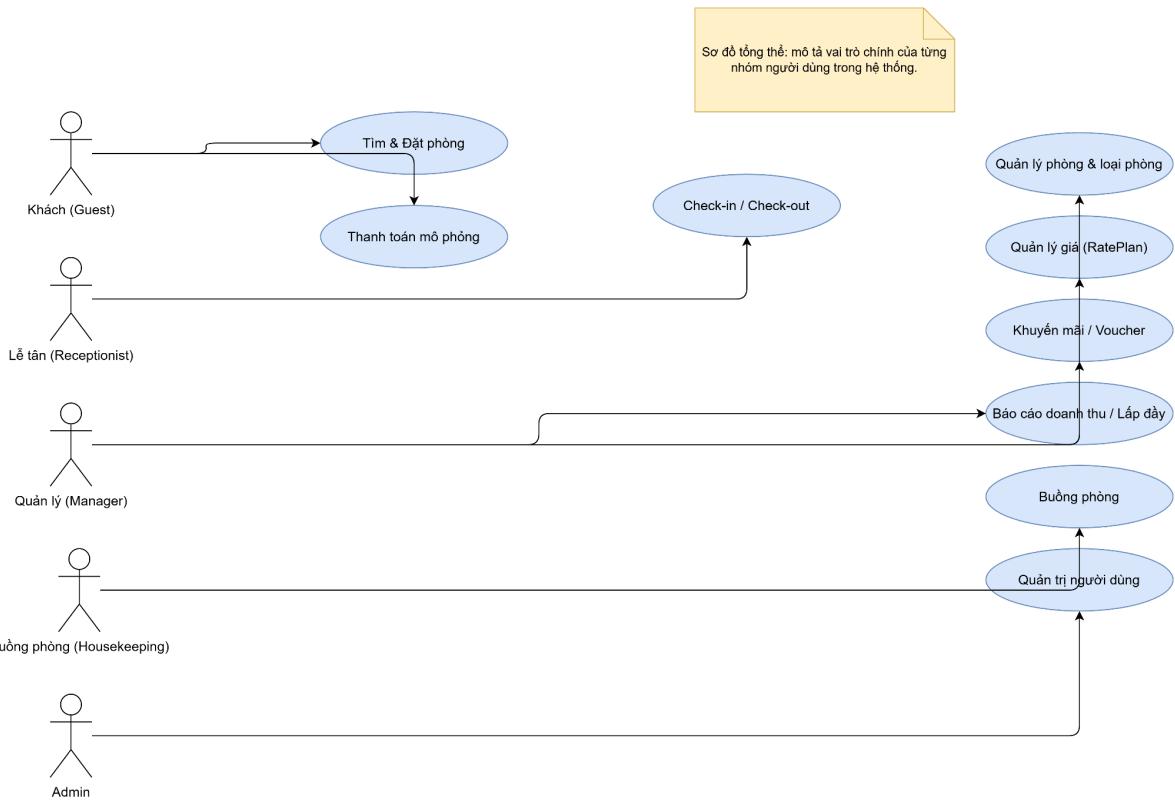
Bài toán được phân rã thành các nhóm chức năng dành cho các đối tượng cụ thể như sau:

- Khách hàng (Guest):** Có nhu cầu tìm kiếm phòng theo ngày, xem thông tin tiện nghi, giá cả và thực hiện đặt phòng trực tuyến. Hệ thống cần đảm bảo quy trình đặt phòng minh bạch, hỗ trợ áp dụng mã giảm giá (Voucher) và thanh toán mô phỏng.
- Lễ tân (Receptionist):** Chịu trách nhiệm thực hiện quy trình Check-in (nhận phòng) và Check-out (trả phòng) cho khách. Lễ tân cần theo dõi sơ đồ phòng thời gian thực để biết phòng nào đang trống, đang ở hoặc đang dọn dẹp.
- Quản lý (Manager):** Cần công cụ để quản lý danh sách phòng, thiết lập các gói giá (RatePlan) linh hoạt (theo ngày thường, cuối tuần, lễ Tết) và chính sách hủy. Ngoài ra, quản lý cần xem các báo cáo doanh thu và tỷ lệ lấp đầy (Occupancy) để đưa ra chiến lược kinh doanh.
- Buồng phòng (Housekeeping):** Cần danh sách các phòng cần dọn dẹp sau khi khách trả phòng và cập nhật trạng thái phòng thành "Sạch" để Lễ tân có thể xếp khách mới.
- Admin:** Quản trị toàn bộ người dùng và phân quyền hệ thống.

3.2. Sơ đồ Use Case:

3.2.1. Sơ đồ Use Case tổng quát:

Dựa trên yêu cầu hệ thống, sơ đồ Use Case tổng quát mô tả sự tương tác giữa các tác nhân (Actors) và các chức năng chính (Use Cases) của hệ thống.



Hình 3.1: Sơ đồ Use Case tổng quát hệ thống quản lý khách sạn.

Danh sách các Tác nhân (Actors):

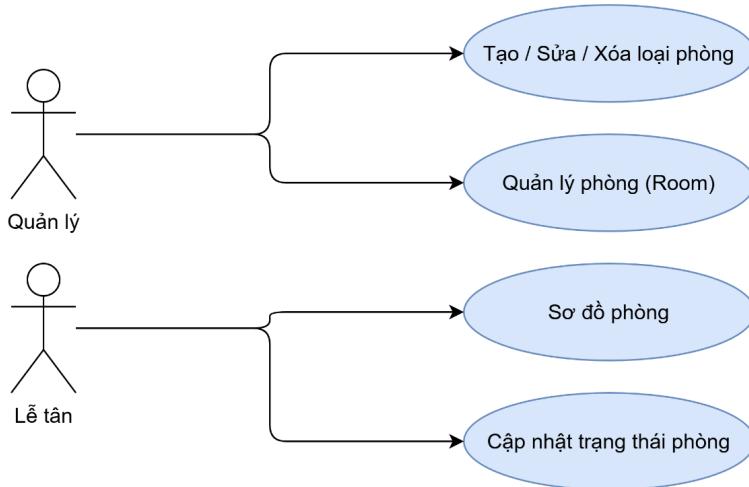
- **Guest (Khách):** Người tìm và đặt phòng.
- **Receptionist (Lễ tân):** Người xử lý check-in/check-out.
- **Manager (Quản lý):** Người cấu hình giá, phòng và xem báo cáo.
- **Housekeeping (Buồng phòng):** Người dọn dẹp và cập nhật trạng thái phòng.
- **Admin:** Quản trị hệ thống.

3.2.2. Đặc tả Use Case:

Phần này mô tả chi tiết các luồng sự kiện chính dựa trên các quy trình nghiệp vụ đã phân tích.

3.2.2.1. Use Case Đăng nhập (UC_01)

(Use case này dùng chung cho nhân viên khách sạn: Admin, Manager, Lễ tân, Buồng phòng)



Mỗi phòng có trạng thái: Trống, Đang ở, Vệ sinh, Bảo trì. Lễ tân chỉ thay đổi khi có khách thực tế.

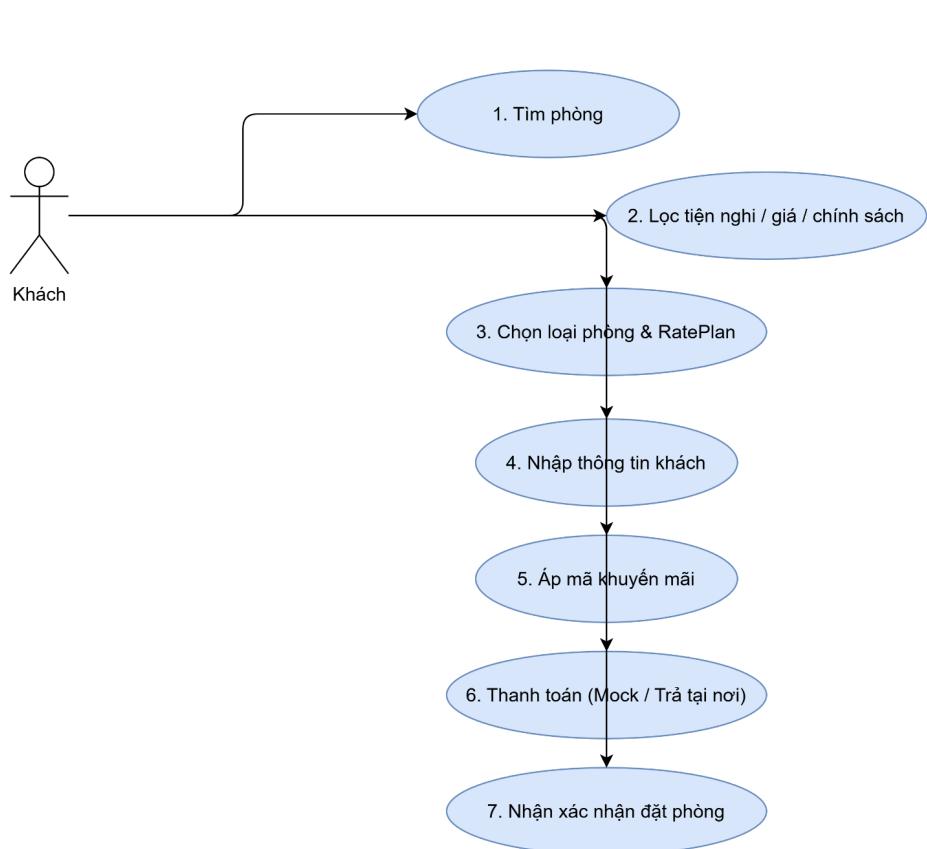
Mã Use Case	UC_01
Tên Use Case	Đăng nhập hệ thống
Tác nhân	Admin, Manager, Receptionist, Housekeeping
Mô tả ngắn	Người dùng đăng nhập vào hệ thống quản trị để thực hiện các chức năng theo phân quyền.
Tiền điều kiện	Người dùng đã có tài khoản được cấp và truy cập vào trang đăng nhập.
Kết quả	Hệ thống chuyển hướng người dùng đến trang Dashboard tương ứng với vai trò (Role).
Luồng sự kiện chính	1. Người dùng nhập Email/Username và Password. 2. Nhấn nút "Đăng nhập". 3. Hệ thống kiểm tra thông tin xác thực trong CSDL. 4. Nếu đúng, hệ thống cấp Token/Session và chuyển vào trang quản trị. 5. Hiển thị thông báo "Đăng nhập thành công".

Luồng sự kiện phụ

- **Sai mật khẩu/tài khoản:** Hệ thống báo lỗi "Thông tin đăng nhập không chính xác" và yêu cầu nhập lại.- **Tài khoản bị khóa:** Hệ thống báo "Tài khoản của bạn đã bị vô hiệu hóa".

3.2.2.2. Use Case Tìm và Đặt phòng (UC_02)

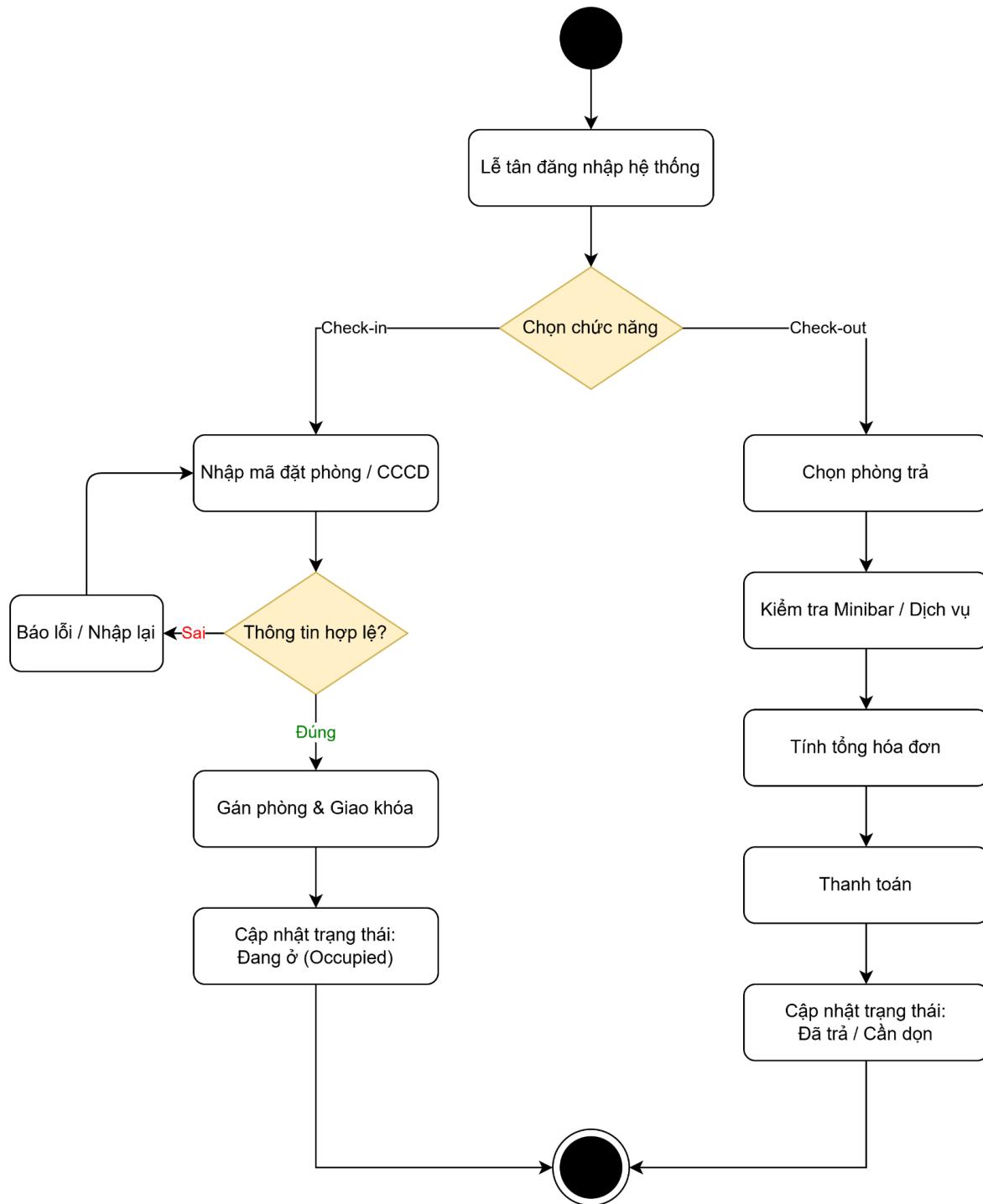
(Dựa trên sơ đồ: Quy trình 7 bước tìm -> đặt phòng)



Mã Use Case	UC_02
Tên Use Case	Tìm kiếm và Đặt phòng (Booking)
Tác nhân	Khách hàng (Guest)
Mô tả ngắn	Khách hàng tìm kiếm phòng trống theo ngày và thực hiện quy trình đặt phòng trực tuyến.
Tiền điều kiện	Khách hàng truy cập vào trang chủ website.
Luồng sự kiện chính	<p>1. Tìm phòng: Khách chọn ngày Check-in, Check-out và số lượng người.2. Lọc: Hệ thống hiển thị danh sách phòng trống kèm giá. Khách lọc theo tiện nghi, giá hoặc hạng phòng.3. Chọn phòng: Khách chọn loại phòng và gói giá (RatePlan) phù hợp (Ví dụ: Không hoàn tiền hoặc Hủy miễn phí).4. Nhập thông tin: Khách nhập họ tên, email, số điện thoại.5. Áp mã: Khách nhập mã Voucher (nếu có) để được giảm giá.6. Thanh toán: Thực hiện thanh toán mô phỏng (hoặc trả sau).7. Xác nhận: Hệ thống tạo Booking, lưu xuống CSDL và hiển thị thông báo "Đặt phòng thành công".</p>
Luồng sự kiện phụ	<ul style="list-style-type: none"> - Hết phòng: Nếu trong lúc chọn, phòng đã bị người khác đặt, hệ thống thông báo "Phòng đã hết" và yêu cầu chọn phòng khác. - Mã giảm giá không hợp lệ: Thông báo lỗi voucher và giữ nguyên giá gốc.

3.2.2.3. Use Case Check-in / Check-out (UC_03)

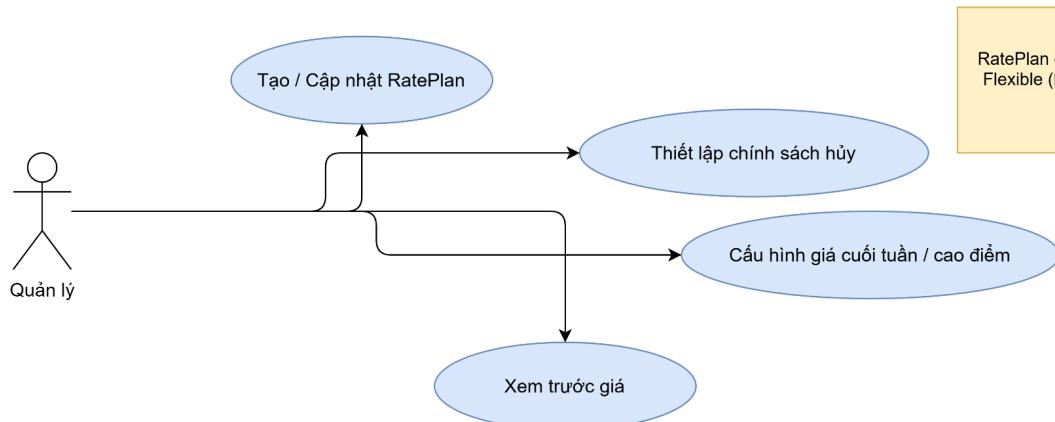
(Dựa trên sơ đồ của Lê Tân)



Mã Use Case	UC_03
Tên Use Case	Check-in / Check-out (Làm thủ tục nhận/trả phòng)
Tác nhân	Lễ tân (Receptionist)
Mô tả ngắn	Lễ tân xử lý việc khách đến nhận phòng hoặc khách rời đi và thanh toán.
Tiền điều kiện	Lễ tân đã đăng nhập hệ thống.
Luồng sự kiện chính	**Quy trình Check-in:** 1. Lễ tân tìm kiếm thông tin đặt phòng (theo tên hoặc mã booking). 2. Xác nhận giấy tờ tùy thân của khách. 3. Gán phòng cụ thể (nếu chưa gán) và giao chìa khóa. 4. Cập nhật trạng thái Booking thành Checked-in và trạng thái phòng thành Occupied (Đang ở). **Quy trình Check-out:** 1. Lễ tân chọn phòng khách đang ở. 2. Kiểm tra các dịch vụ phát sinh (minibar,...) để tính tổng bill. 3. Khách thanh toán số tiền còn lại. 4. Cập nhật trạng thái Booking thành Completed (Hoàn thành). 5. Hệ thống tự động chuyển trạng thái phòng sang Dirty (Cần dọn) để báo cho bộ phận buồng phòng.
Luồng sự kiện phụ	(Trống)

3.2.2.4. Use Case Quản lý Gói giá (RatePlan) (UC_04)

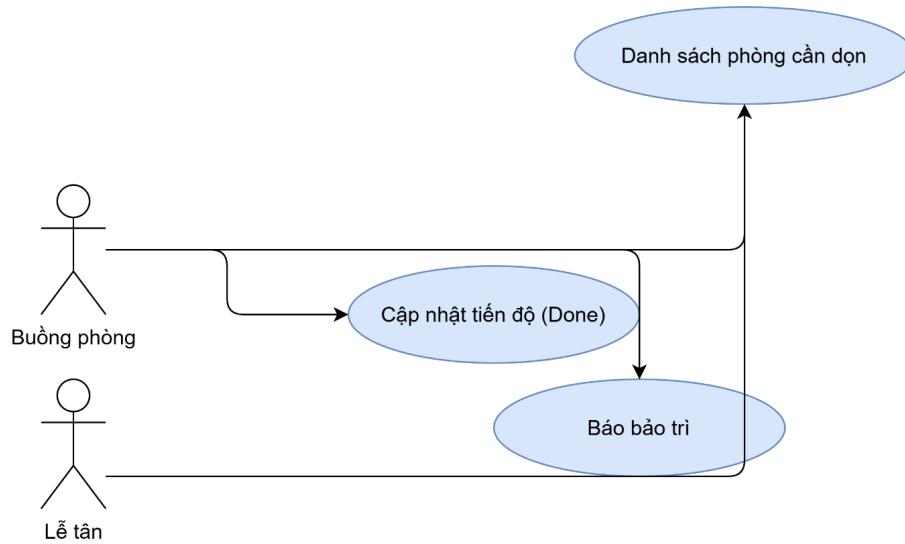
(Dựa trên sơ đồ Manager - RatePlan)



Mã Use Case	UC_04
Tên Use Case	Quản lý Gói giá (RatePlan)
Tác nhân	Quản lý (Manager)
Mô tả ngắn	Cấu hình các mức giá bán khác nhau và chính sách hủy phòng.
Luồng sự kiện chính	1. Quản lý truy cập danh sách RatePlan.2. Chọn chức năng "Tạo mới" hoặc "Cập nhật".3. Thiết lập chính sách hủy: Chọn loại Flexible (Hủy miễn phí trước X ngày) hoặc Non-refundable (Không hoàn tiền).4. Cấu hình giá: Nhập giá cơ bản, thiết lập phụ thu cho ngày cuối tuần hoặc mùa cao điểm.5. Xem trước: Hệ thống hiển thị giá mô phỏng theo lịch.6. Lưu lại cấu hình vào CSDL.
Luồng sự kiện phụ	(Trống)

3.2.2.5. Use Case Quản lý Buồng phòng (UC_05)

(Dựa trên sơ đồ Buồng phòng - Housekeeping)



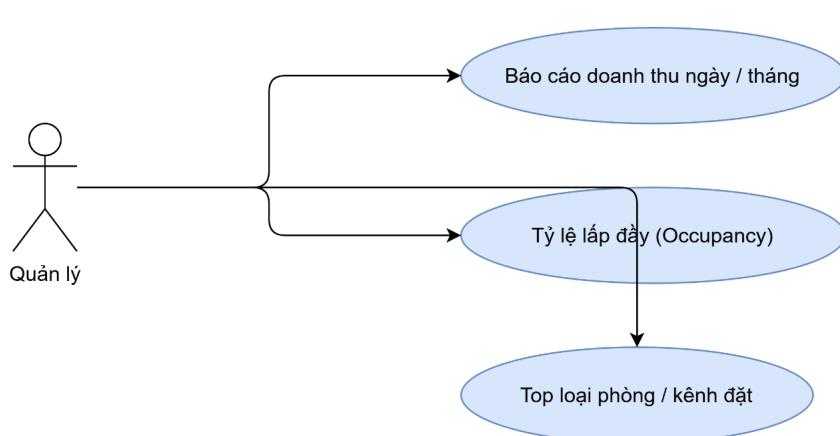
Khi khách Check-out → phòng sang trạng thái
'Vệ sinh'.
Sau khi dọn xong → chuyển lại 'Trống'.

Mã Use Case	UC_05
Tên Use Case	Cập nhật trạng thái dọn dẹp
Tác nhân	Buồng phòng (Housekeeping), Lễ tân
Mô tả ngắn	Theo dõi và cập nhật trạng thái vệ sinh của phòng sau khi khách trả phòng.
Tiền điều kiện	Phòng đang ở trạng thái Dirty (Bẩn) sau khi khách Check-out.
Luồng sự kiện chính	1. Nhân viên buồng phòng đăng nhập, xem "Danh sách phòng cần dọn". 2. Thực hiện dọn dẹp phòng thực tế. 3. Sau khi dọn xong, nhân viên chọn phòng trên hệ thống và nhấn "Cập nhật tiến độ (Done)". 4. Trạng thái phòng trên hệ thống chuyển từ Dirty sang Ready (Sẵn sàng/Trống) hoặc Inspecting (Chờ kiểm tra). 5. Lễ tân nhìn thấy phòng đã sạch và có thể xếp khách mới.

Luồng sự kiện phụ

Báo bảo trì: Nếu trong quá trình dọn phát hiện thiết bị hư hỏng, nhân viên chọn chức năng "Báo bảo trì". Trạng thái phòng chuyển sang Maintenance (Bảo trì) và không thể bán.

3.2.2.6. Use Case Báo cáo thống kê (UC_06) (Dựa trên sơ đồ Báo cáo doanh thu/Lắp đầy)



Mã Use Case	UC_06
Tên Use Case	Xem Báo cáo doanh thu & Tỷ lệ lắp đầy
Tác nhân	Quản lý (Manager)
Mô tả ngắn	Hỗ trợ người quản lý theo dõi hiệu quả kinh doanh qua các biểu đồ trực quan.

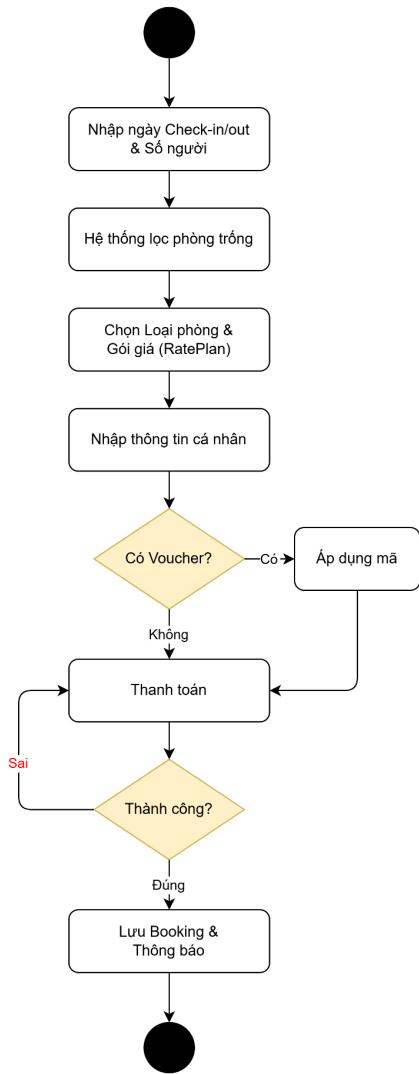
Luồng sự kiện chính	1. Quản lý vào menu "Báo cáo".2. Chọn loại báo cáo: "Doanh thu" hoặc "Tỷ lệ lấp đầy (Occupancy)".3. Chọn khoảng thời gian (Ngày/Tháng/Năm).4. Hệ thống truy vấn dữ liệu từ các hóa đơn và lịch sử đặt phòng.5. Hiển thị biểu đồ doanh thu theo thời gian và danh sách "Top loại phòng/kênh đặt" được ưa chuộng nhất.
Luồng sự kiện phụ	(Trống)

3.3. Sơ đồ hoạt động (Activity Diagram):

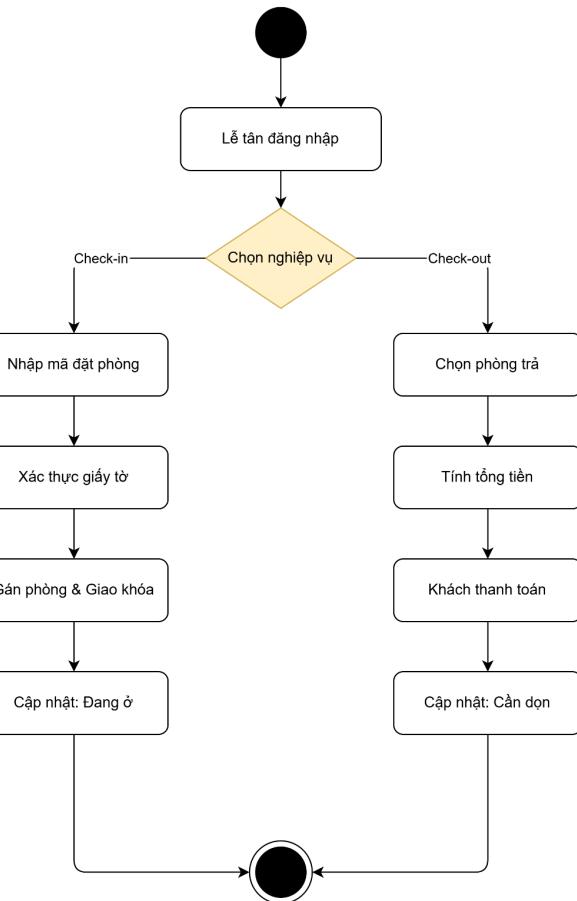
Sơ đồ hoạt động (Activity Diagram) được sử dụng để mô tả luồng công việc hoặc quá trình nghiệp vụ trong hệ thống, tập trung vào các bước thực hiện và quyết định. Dựa trên tính quan trọng và độ phức tạp của quy trình, chúng tôi sẽ xây dựng sơ đồ cho hai Use Case chính: **Tìm kiếm và Đặt phòng (UC_02)** và **Check-in / Check-out (UC_03)**.

- **Sơ đồ hoạt động cho UC_02: Tìm kiếm và Đặt phòng**
 - Sơ đồ này sẽ mô tả luồng 7 bước của khách hàng, bắt đầu từ việc chọn ngày check-in/check-out, tìm kiếm phòng trống, lọc kết quả, chọn gói giá (RatePlan), nhập thông tin cá nhân, áp dụng mã giảm giá (Voucher), và kết thúc bằng việc thanh toán và xác nhận Booking.
 - Các swimlane (phân làn) chính: Khách hàng (Guest), Hệ thống, Cơ sở dữ liệu (MySQL/EF Core).
- **Sơ đồ hoạt động cho UC_03: Check-in / Check-out**
 - Sơ đồ này bao gồm hai luồng chính là Check-in và Check-out.
 - **Luồng Check-in:** Mô tả quy trình Lễ tân tìm booking, xác minh thông tin, gán phòng và cập nhật trạng thái Booking sang *Checked-in* và trạng thái Phòng sang *Occupied*.
 - **Luồng Check-out:** Mô tả quy trình Lễ tân kiểm tra dịch vụ phát sinh, tính tổng bill, khách thanh toán, cập nhật Booking sang *Completed*, và hệ thống tự động chuyển trạng thái phòng sang *Dirty* (báo cho bộ phận Buồng phòng).

Hình 3.x: Quy trình Tìm & Đặt phòng (UC_02)



Hình 3.y: Quy trình Check-in / Check-out (UC_03)



3.4. Sơ đồ tuần tự (Sequence Diagram):

Sơ đồ tuần tự (Sequence Diagram) mô tả thứ tự các thông điệp được trao đổi giữa các đối tượng trong hệ thống theo thời gian để hoàn thành một chức năng. Chúng tôi sẽ tập trung vào các tương tác cốt lõi: **Đăng nhập (UC_01)** và **Tìm kiếm và Đặt phòng (UC_02)**.

- **Sơ đồ tuần tự cho UC_01: Đăng nhập**

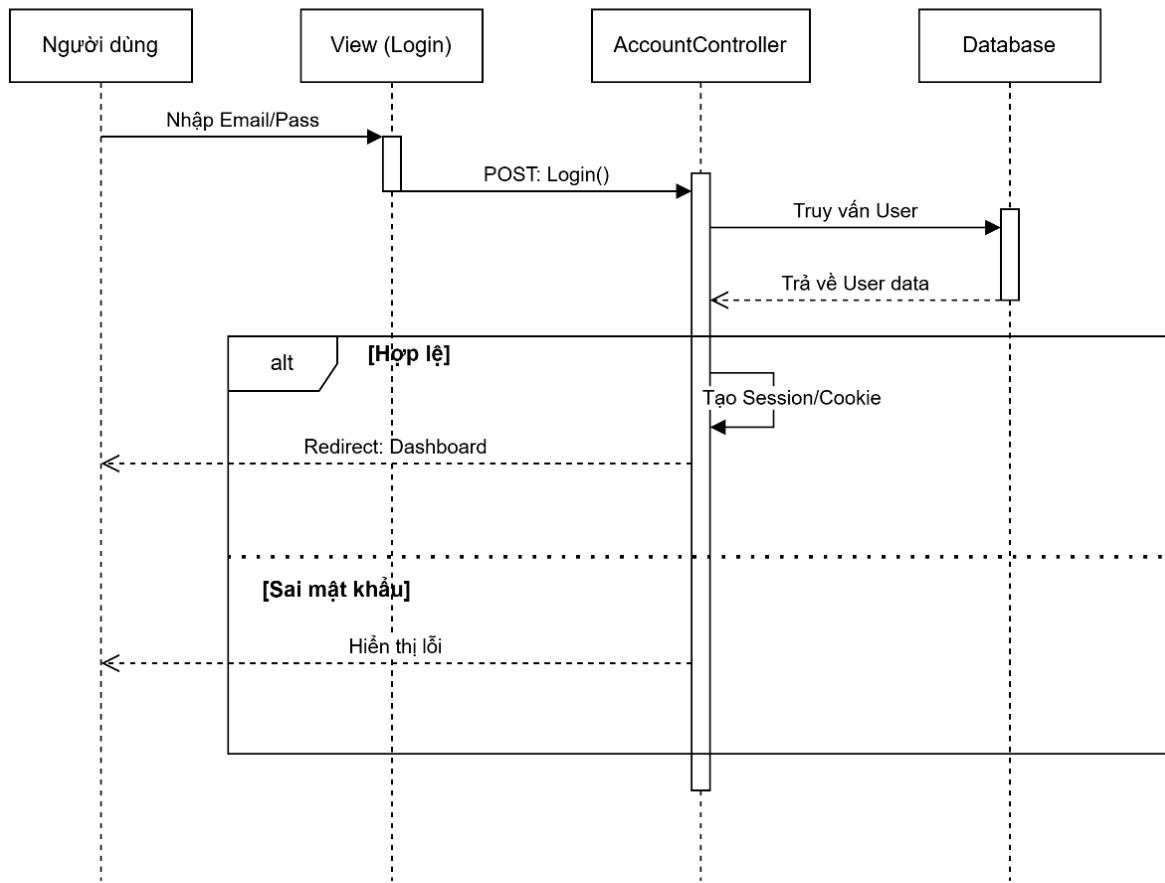
- Các đối tượng tham gia: Người dùng (Client), Controller (ví dụ: *AccountController*), Service/Logic Xử lý, DbContext (EF Core), Cơ sở dữ liệu (MySQL).
- Luồng thông điệp: Người dùng gửi yêu cầu Đăng nhập (Email/Pass) → Controller nhận và gọi đến Service → Service sử dụng DbContext để

truy vấn MySQL xác thực → MySQL trả về kết quả → Service trả về Token/Session → Controller chuyển hướng đến Dashboard.

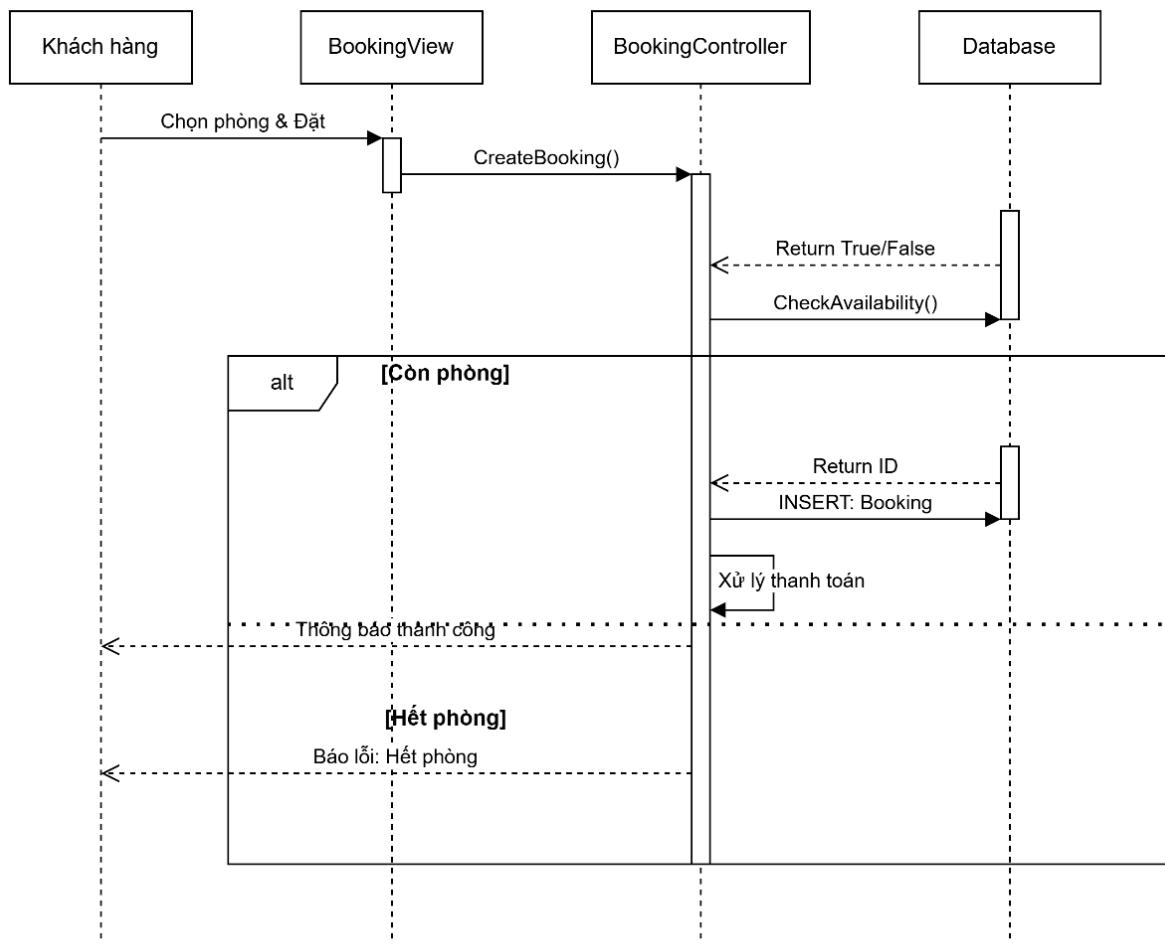
- **Sơ đồ tuần tự cho UC_02: Tìm kiếm và Đặt phòng**

- Các đối tượng tham gia: Khách hàng (Client), Controller (*BookingController*), Service (*RoomService*, *BookingService*), DbContext (EF Core), Cơ sở dữ liệu (MySQL).
- Luồng thông điệp: Khách gửi yêu cầu Tìm phòng (ngày/số lượng) → Controller gọi *RoomService* → *RoomService* truy vấn MySQL thông qua DbContext để kiểm tra phòng trống và gói giá (RatePlan) → Kết quả trả về cho Controller và hiển thị ra View → Khách tiến hành Đặt phòng → Controller gọi *BookingService* → *BookingService* thực hiện lưu thông tin Booking, áp dụng Voucher và lưu hóa đơn (*Invoice*) vào MySQL.

Hình 3.4.1: Sơ đồ tuần tự chức năng Đăng nhập

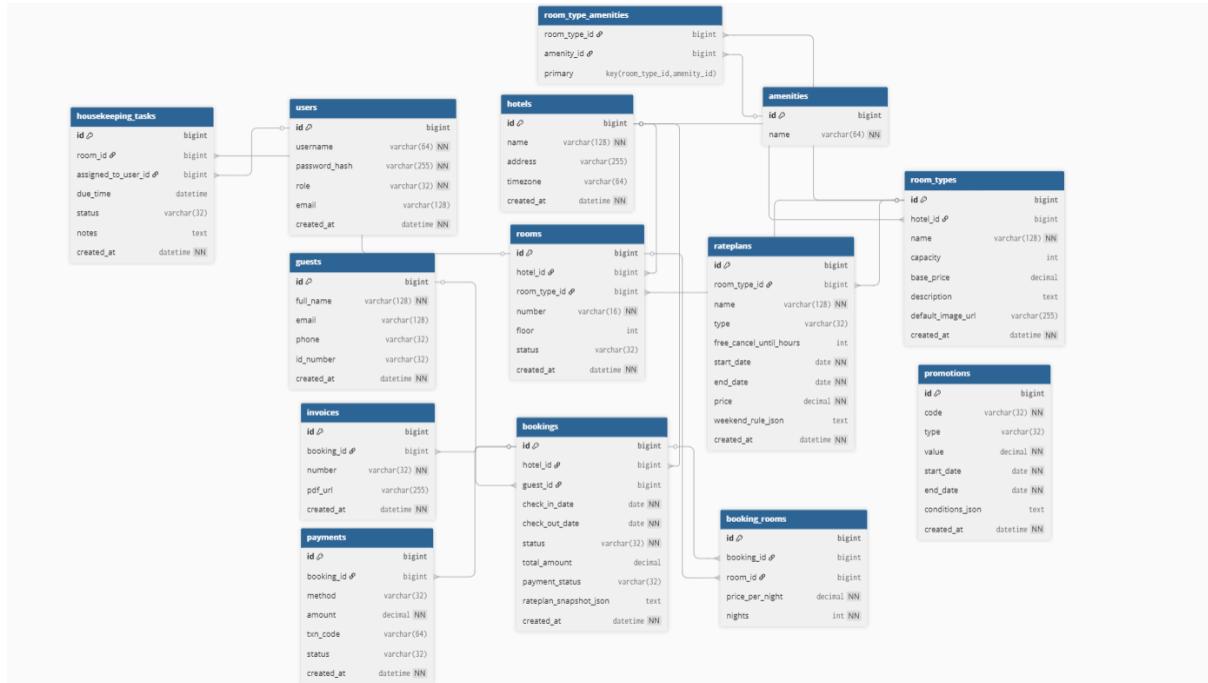


Hình 3.4.2: Sơ đồ tuần tự chức năng Đặt phòng



3.5. Thiết kế Cơ sở dữ liệu (Database Design):

Cơ sở dữ liệu (CSDL) được thiết kế dựa trên phân tích nghiệp vụ trong Chương 3.1 và các Use Case đã đặc tả. CSDL sử dụng hệ quản trị MySQL và được ánh xạ qua Entity Framework Core (Code-first).



Các bảng chính cần thiết để hỗ trợ toàn bộ nghiệp vụ của ứng dụng:

Tên Bảng (Entity)	Mô tả nghiệp vụ	Liên kết chính
Users	Lưu trữ thông tin tài khoản người dùng, phân biệt qua vai trò (Role): Admin, Manager, Receptionist, Housekeeping.	Liên kết với Bookings (cho Khách hàng).
RoomTypes	Lưu trữ danh sách các loại phòng (ví dụ: Standard, Deluxe, Suite) và các tiện	One-to-many với Rooms .

	nghi (Amenity) đi kèm.	
Rooms	Lưu trữ danh sách phòng vật lý của khách sạn (ví dụ: P101, P102). Ghi nhận trạng thái hiện tại của phòng (Ready, Occupied, Dirty, Maintenance).	One-to-one với RoomTypes .
RatePlans	Lưu trữ các gói giá bán và chính sách hủy (Flexible, Non-refundable) được thiết lập bởi Quản lý.	One-to-many với Bookings .
Bookings	Lưu trữ thông tin chi tiết về mỗi lượt đặt phòng của khách hàng (ngày check-in/check-out, tổng tiền, trạng thái Booking: Pending, Confirmed, Checked-in, Completed, Cancelled).	Foreign Keys: UserID , RoomID , RatePlanID .
Invoices	Lưu trữ thông tin hóa đơn thanh toán cho mỗi Booking, bao gồm cả dịch vụ phát sinh (nếu có).	One-to-one với Bookings .
Vouchers	Lưu trữ danh sách mã giảm giá và điều kiện áp dụng.	One-to-many với Bookings (qua mã Voucher được áp dụng).

Thiết kế CSDL sẽ tập trung vào việc đảm bảo toàn vẹn dữ liệu (Integrity) và hỗ trợ các truy vấn báo cáo doanh thu, tỷ lệ lấp đầy một cách hiệu quả.