

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

**Bài tập lớn**

**Xây dựng Website Quản lý khách sạn bằng Spring MVC**

|  |  |
| --- | --- |
| Họ và tên: | Nguyễn Huy Quang |
| Mã Sinh Viên: | B20DCAT144 |
| Nhóm học phần: | Nhóm 03-03 |
| Hệ: | Chính quy |
| Giảng viên | Đỗ Quang Hưng |

**Hà Nội, 2023**

**LỜI MỞ ĐẦU**

Trong thời đại hiện nay, việc đặt phòng khách sạn không còn chỉ là một phần quan trọng trong lịch trình du lịch mà còn trở thành một trải nghiệm kỹ thuật số thú vị. Với sự bùng nổ của công nghệ và Internet, khả năng tìm kiếm và đặt phòng khách sạn trực tuyến đã trở nên dễ dàng và tiện lợi hơn bao giờ hết. Tuy nhiên, trong thế giới số hóa ngày càng phức tạp, việc lựa chọn một trang web đặt phòng khách sạn phù hợp có thể là một thách thức đối với du khách. Và tại thời điểm này, sự tiện lợi của việc tìm kiếm và đặt phòng trực tuyến trở nên quan trọng hơn bao giờ hết. Em hiểu rằng việc lựa chọn một trang web đặt phòng khách sạn phù hợp có thể đầy khó khăn, với hàng trăm tùy chọn và một loạt các yếu tố cần xem xét như vị trí, tiện ích, giá cả, và đánh giá của người dùng khác. Đó chính là lý do tại sao em đã quyết định nghiên cứu và phát triển một trang web đặt phòng khách sạn. Trang web này sẽ giúp du khách dễ dàng tìm kiếm, so sánh và đặt phòng tại các khách sạn trên khắp thế giới.

Em cam kết xây dựng trang web này với mục tiêu mang đến trải nghiệm tốt nhất cho người dùng. Trang web của em sẽ là một cầu nối giữa du khách và các khách sạn, giúp họ tìm thấy nơi lưu trú lý tưởng cho chuyến đi của mình. Em hi vọng rằng trang web này sẽ giúp du khách có những kỷ niệm đáng nhớ và thoải mái tại bất kỳ địa điểm nào họ muốn khám phá.

Bởi vậy trước nhu cầu và lợi thế đó, em quyết định nghiên cứu và xây dựngtrang web: **Quản lý khách sạn**

**LỜI CẢM ƠN**

Em xin gửi lời cảm ơn đặc biệt tới thầy Đỗ Quang Hưng, người hướng dẫn và giúp đỡ em trong quá trình nghiên cứu và thực hiện dự án. Sự kiên nhẫn, kiến thức và sự động viên của cô đã giúp em vượt qua mọi khó khăn và hoàn thành dự án một cách xuất sắc. Em rất biết ơn những bài học quý báu về phương pháp luận nghiên cứu khoa học mà cô đã truyền đạt cho em.

Em xin gửi lời cảm ơn chân thành tới gia đình và bạn bè, những người đã luôn ở bên cạnh và động viên em trong mọi thử thách. Những lời khích lệ và sự tin tưởng của quý vị đã thúc đẩy em vượt qua khó khăn và đạt được những thành công này.

Em cũng muốn bày tỏ lòng biết ơn đến đối tác, đồng nghiệp, và tất cả những ai đã đóng góp ý kiến, thời gian, và nỗ lực để làm cho dự án này trở nên thành công.

Mục Lục

[Danh sách các thành viên và phân công công việc. 4](#_Toc155047271)

[Chương 1. Mở đầu và mô tả hệ thống 5](#_Toc155047272)

[Website quản lý đặt phòng khách sạn 5](#_Toc155047273)

[1.1. Giới thiệu website 5](#_Toc155047274)

[1.2. Phân tích yêu cầu hệ thống. 5](#_Toc155047275)

[1.3. Thiết kế tổng quan hệ thống 5](#_Toc155047276)

[1.4 Thiết kế chi tiết hệ thống 7](#_Toc155047277)

[1.4.1. Sơ đồ khối các chức năng hệ thống 7](#_Toc155047278)

[1.4.2. Biểu đồ lớp – class diagram 9](#_Toc155047279)

[1.4.3. Sơ đồ thực thể quan hệ ERD 14](#_Toc155047280)

[1.4.4. Lược đồ quan hệ 16](#_Toc155047281)

[1.5. Đặc tả Usecase. 18](#_Toc155047282)

[1.5.1. Quản lý tài khoản. 18](#_Toc155047283)

[1.5.2. Quản lý phòng. 19](#_Toc155047284)

[1.5.3. Quản lý đặt phòng. 21](#_Toc155047285)

[1.5.4. Xem báo cáo. 23](#_Toc155047286)

[1.5.5:Usecase tổng quát: 23](#_Toc155047287)

[1.5.6 Usecase tổng quát của khách hàng 24](#_Toc155047288)

[1.5.7 Usecase tổng quát của manager 24](#_Toc155047289)

[1.5.8 Usecase tổng quát của admin 25](#_Toc155047290)

[1.5.9 Biểu đồ tuần tự của Manager 25](#_Toc155047291)

[1.5.10 Biểu đồ tuần tự của admin 27](#_Toc155047292)

[1.5.11 Biểu đồ tuần tự của khách hàng 28](#_Toc155047293)

[1.5.12 Lớp thực thế 28](#_Toc155047294)

[Chương 2. Các giao diện. 29](#_Toc155047295)

[2.1. Giao diện Home. 29](#_Toc155047296)

[2.2. Đăng ký. 32](#_Toc155047297)

[2.3. Thêm phòng. 33](#_Toc155047298)

[2.4. Chỉnh sửa phòng. 34](#_Toc155047299)

[2.5. Danh sách phòng. 35](#_Toc155047300)

[2.6. Thông tin phòng. 37](#_Toc155047301)

[2.7. Đặt phòng. 38](#_Toc155047302)

[2.8. Danh sách đặt phòng. 38](#_Toc155047303)

[2.9. Quản lý tài khoản. 39](#_Toc155047304)

[2.10. Xem báo cáo. 39](#_Toc155047305)

[Chương 3. Kết luận, hạn chế. 40](#_Toc155047306)

[3.1. Kết luận. 40](#_Toc155047307)

[3.2. Hạn chế. 41](#_Toc155047308)

# Danh sách các thành viên và phân công công việc.

|  |  |
| --- | --- |
| Họ và tên | Công việc |
| Nguyễn Huy Quang – B20DCAT144 | * Phân chia công việc * Thiết kế modul “Quản lý phòng” * Thiết kế modul “Quản lý đặt phòng” * Viết báo cáo * Thực hiện code sau khi đã có phân tích thiết kế * Bổ sung css |
| Vũ Anh Thao - B20DCAT177 | * Thiết kế modul “Quản lý tài khoản” |
| Trần Quốc Huy - B20DCAT086 | * Thiết kế modul “Xem báo cáo” |

# Chương 1. Mở đầu và mô tả hệ thống

# Website quản lý đặt phòng khách sạn

## Giới thiệu website

Trước sự phát triển như vũ bão của CNTT cùng cuộc cách mạng công nghệ 4.0, việc đặt phòng khách sạn không chỉ là thực hiện tại chỗ mà còn có thể thực hiện ở bất cứ nơi đâu. Và website quản lý đặt phòng khách sạn đã trở thành một lựa chọn thông minh và tiện lợi cho việc quản lý khách sạn

Với sự hỗ trợ của java và spring framework cùng cơ sở dữ liệu mysql, website cung cấp cho khách hàng những trải nghiệm đặt phòng tiện lợi và nhanh chóng.

Khách hàng có thể dễ dàng xem xét, so sánh và đặt phòng khách sạn yêu thích từ bất kỳ đâu và bất kỳ lúc nào.

Bài tập lớn nhóm của tụi em đảm nhiệm về chủ đề xây dựng trang web quản lý đặt phòng khách sạn. Hệ thống sử dụng framework Spring của Java và cơ sở dữ liệu MySQL.

Spring Framework nhằm giúp cho việc phát triển ứng dụng Java trở nên dễ dàng hơn, giảm thiểu các công việc lặp lại, tăng tốc độ phát triển và giảm thiểu các lỗi phát sinh trong quá trình phát triển ứng dụng.

## Phân tích yêu cầu hệ thống.

Dự án là một trang web quản lý đặt phòng khách sạn, nó cung cấp một danh sách các phòng khác nhau có sẵn cho người dùng đặt phòng theo hình thức online. Mỗi người dùng sẽ có thể xem danh sách phòng bao gồm các mô tả để lựa chọn xem có phù hợp không. Hệ thống được triển khai bằng Spring framework của Java.

## Thiết kế tổng quan hệ thống

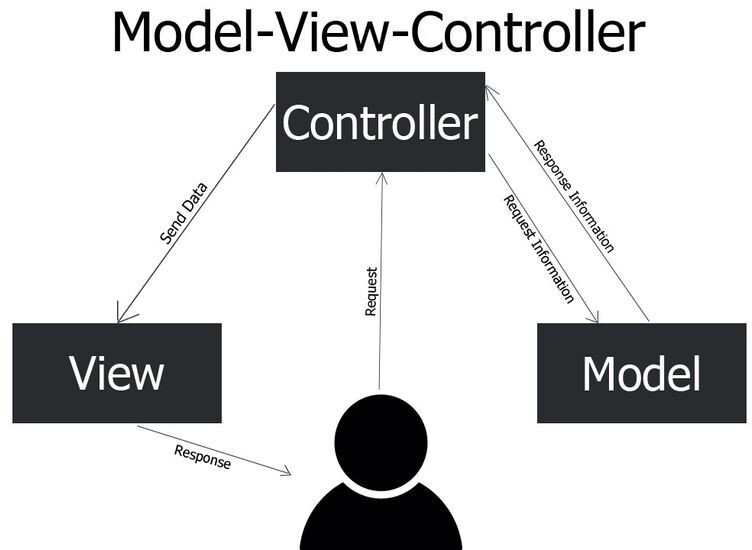
Website được xây dựng với mô hình Spring MVC. Website được xây dựng bằng HTML và framework Spring và database MySQL.

Mô hình MVC (Model-View-Controller) là một mô hình thiết kế phần mềm được sử dụng rộng rãi trong phát triển ứng dụng web. Nó tách biệt các thành phần

khác nhau của ứng dụng và định nghĩa vai trò riêng cho chúng, giúp cho việc phát triển và bảo trì ứng dụng dễ dàng hơn.

Java Spring là một framework phát triển ứng dụng web phổ biến trong cộng đồng phát triển Java. Nó cung cấp một cách tiếp cận dựa trên mô hình MVC cho phát triển ứng dụng web.

Trong Java Spring, Model đại diện cho các lớp đối tượng và dữ liệu trong ứng dụng, View là phần giao diện người dùng và Controller là phần điều khiển hoạt động của ứng dụng.



*Hình 1.1. Mô hình MVC.*

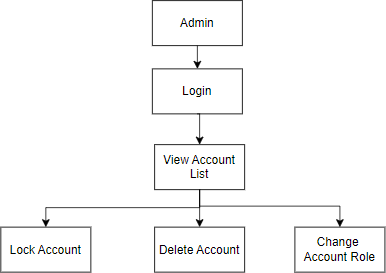
Các thành phần trong mô hình MVC của Java Spring hoạt động như sau:

* + - Model: Đây là nơi chứa dữ liệu và các phương thức để truy xuất và cập nhật dữ liệu. Model được định nghĩa bởi các lớp đối tượng trong ứng dụng. Spring cung cấp các công cụ hỗ trợ để truy xuất dữ liệu từ các nguồn khác nhau như cơ sở dữ liệu, tệp tin hoặc web service.
    - View: Đây là phần giao diện người dùng, nơi hiển thị dữ liệu cho người dùng. Spring hỗ trợ nhiều công nghệ view khác nhau như JSP, Thymeleaf hoặc AngularJS.
    - Controller: Đây là nơi điều khiển hoạt động của ứng dụng. Nó xử lý các yêu cầu từ người dùng và truy xuất và cập nhật dữ liệu trong Model. Sau đó, nó chuyển dữ liệu sang View để hiển thị kết quả cho người dùng. Spring cung cấp các công cụ hỗ trợ để xử lý các yêu cầu từ người dùng và định tuyến chúng đến các Controller tương ứng.

Tổng quan, Java Spring MVC giúp cho phát triển ứng dụng web trở nên dễ dàng hơn với việc phân tách logic của ứng dụng vào các thành phần riêng biệt, giúp cho việc phát triển, bảo trì và mở rộng ứng dụng trở nên dễ dàng hơn.

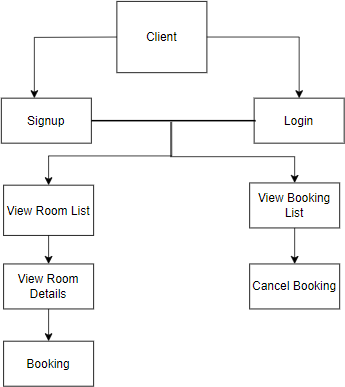
## Thiết kế chi tiết hệ thống

### Sơ đồ khối các chức năng hệ thống

* + - 1. Về phía Admin: Quản trị viên đăng nhập vào hệ thống, có thể xem danh sách các tài khoản của hệ thống, khóa tài khoản, xóa tài khoản, đổi vai trò giữa quản lý và khách hàng

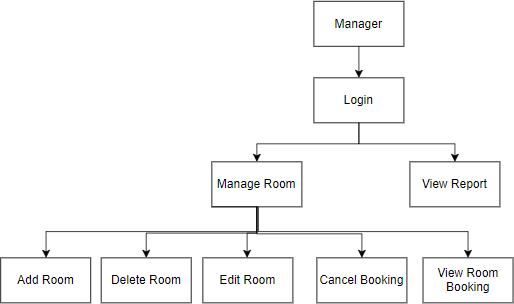
*Hình 1.2. Sơ đồ khối phía admin.*

* + - 1. Về phía Client: Khách hàng có thể xem danh sách các phòng, đặt phòng, hủy đặt phòng, xem danh sách đặt phòng, thanh toán



*Hình 1.3. Sơ đồ khối phía Client.*

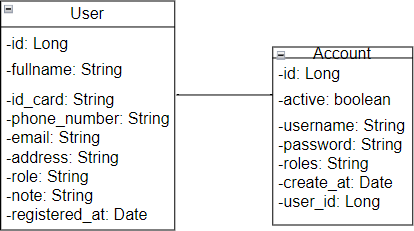
* + - 1. Về phía Manager: Quản lý đăng nhập vào hệ thống, có thể quản lý phòng (xem chi tiết phòng, thêm, sửa, xóa thông tin phòng, hủy đặt phòng), xem thống kê đặt phòng



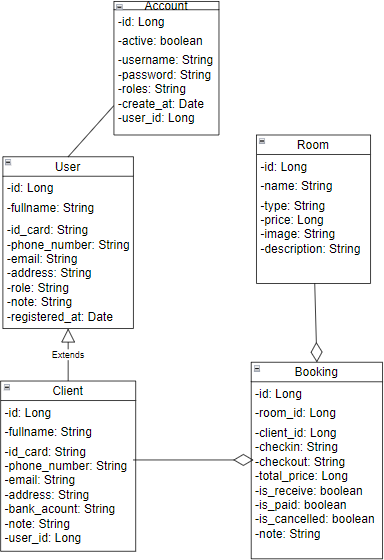
*Hình 1.4. Sơ đồ khối phía manager.*

### Biểu đồ lớp – class diagram

* + - 1. Về phía Admin: Gồm 2 class: User, Account Class diagram:



*Hình 1.5. Biểu đồ lớp phía admin*

* + - 1. Về phía Client: Gồm 5 class: User, Client, Booking, Room, Account Class diagram:

*Hình 1.6. Biểu đồ lớp phía client.*

Mô tả chi tiết các thuộc tính:

Bảng 1.1. Bảng thể hiện các thuộc tính của class User

|  |  |  |
| --- | --- | --- |
|  | Tên | Mô tả |
| Class | User | Thể hiện thông tin người dùng hệ thống |
| Thuộc tính | id | Thể hiện mã người dùng, mã này hệ thống tự tạo |
| fullname | Thể hiện tên đầy đủ của người dùng |
| id\_card | Mã số cccd của người dùng |
| phone\_number | Số điện thoại của người dùng |
| email | Địa chỉ email của người dùng |
| address | Địa chỉ của người dùng |
| role | Vai trò của người dùng |
| note | Ghi chú nếu có |
| registered\_at | Thời gian tạo người dùng |

Bảng 1.2. Bảng thể hiện các thuộc tính của class Client

|  |  |  |
| --- | --- | --- |
|  | Tên | Mô tả |
| Class | Client | Thể hiện thông tin khách hàng |
| Thuộc tính | id | Thể hiện mã khách hàng, mã này hệ thống tự tạo |
| fullname | Thể hiện tên đầy đủ của khách hàng |
| id\_card | Mã số cccd của khách hàng |
| phone\_number | Số điện thoại của khách hàng |
| email | Địa chỉ email của khách hàng |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  | address | Địa chỉ của khách hàng |
| bank\_account | Tài khoản ngân hàng của khách hàng |
| note | Ghi chú nếu có |
| user\_id | Mã người dùng của khách hàng |

Bảng 1.3. Bảng thể hiện các thuộc tính của class Booking

|  |  |  |
| --- | --- | --- |
|  | Tên | Mô tả |
| Class | Booking | Thể hiện thông tin đặt phòng của khách hàng trên hệ thống |
| Thuộc tính | id | Thể hiện mã đặt phòng, được hệ thống tự tạo |
| checkin | Thể hiện ngày nhận phòng |
| checkout | Ngày trả phòng |
| total\_price | Tổng tiền |
| is\_receive | Thể hiện trạng thái đã nhận phòng hay chưa |
| is\_paid | Thể hiện trạng thái khách đã thanh toán chưa |
| is\_cancelled | Thể hiện trạng thái đơn đặt phòng đã bị hủy hay chưa |
| note | Ghi chú về đơn đặt phòng |
| room\_id | Mã phòng của đơn |
| client\_id | Mã khách hàng đặt đơn |

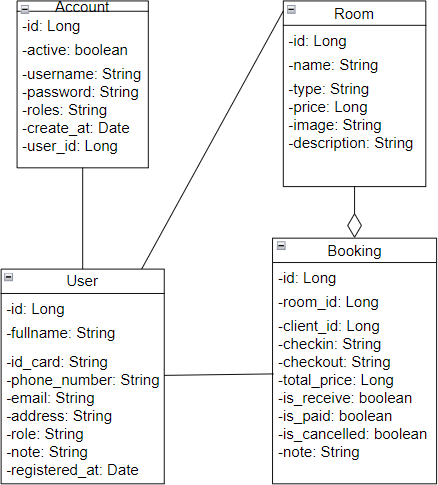
Bảng 1.4. Bảng thể hiện các thuộc tính của class Room

|  |  |  |
| --- | --- | --- |
|  | Tên | Mô tả |
| Class | Room | Thể hiện thông tin của phòng |
| Thuộc tính | id | Mã của phòng |
| name | Tên phòng |
| type | Loại phòng |
| price | Giá của phòng |
| image | Ảnh phòng |
| description | Mô tả |
| floor | Tầng |

Bảng 1.5. Bảng thể hiện các thuộc tính của Account

|  |  |  |
| --- | --- | --- |
|  | Tên | Mô tả |
| Class | Account | Thể hiện thông tin tài khoản của người dùng hệ thống |
| Thuộc tính | id | Mã của tài khoản |
| username | Tên đăng nhập của người dùng |
| password | Mật khẩu đăng nhập của người dùng |
| active | Thể hiện trạng thái hoạt động của tài khoản |
| roles | Thể hiện vai trò của tài khoản |
| create\_at | Thời gian tài khoản được tạo |
| user\_id | Mã của người dùng |

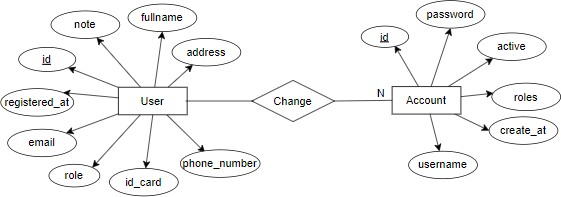
* + - 1. Về phía Manager: gồm 4 class: User, Account, Room, Booking Class diagram:



*Hình 1.7. Biểu đồ lớp phía manager.*

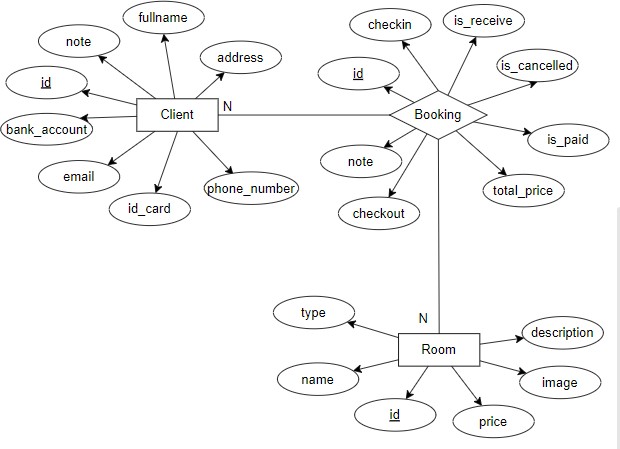
### Sơ đồ thực thể quan hệ ERD

1. Về phía Admin
   * Mỗi Admin có thể thay đổi nhiều tài khoản: khóa tài khoản, xóa tài khoản, đổi vai trò giữa khách hàng và quản lý
   * Mỗi Admin là 1 User có role (vai trò) là Admin. User còn có các thuộc tính khác là tên đầy đủ, địa chỉ, ghi chú, mã người dùng, thời gian tạo người dùng, địa chỉ email, số cccd, số điện thoại



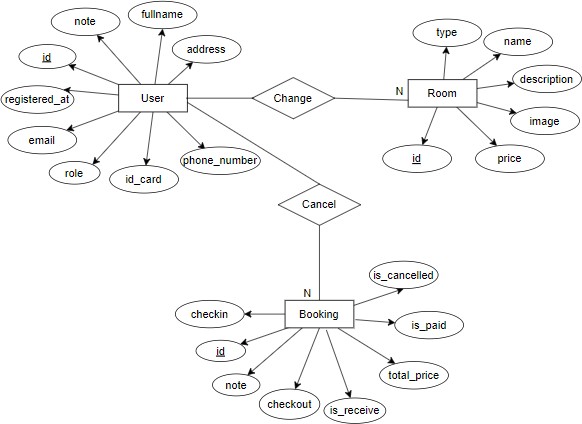
*Hình 1.8. Sơ đồ thực thể phía admin.*

1. Về phía Client



*Hình 1.9 Sơ đồ thực thể phía client.*

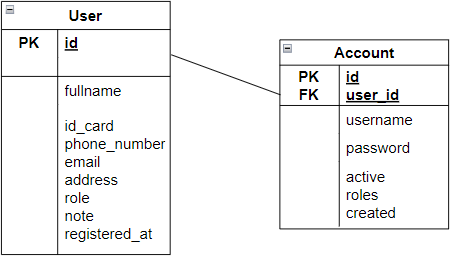
1. Về phía Manager



*Hình 1.10. Sơ đồ thực thể phía manager.*

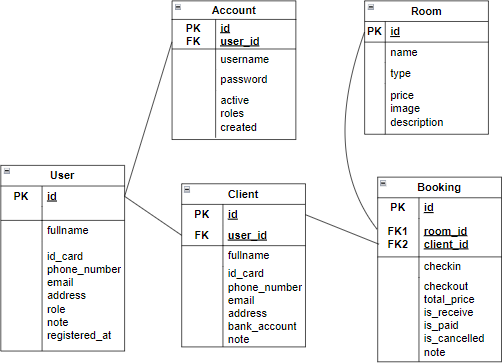
### Lược đồ quan hệ

1. Về phía Admin



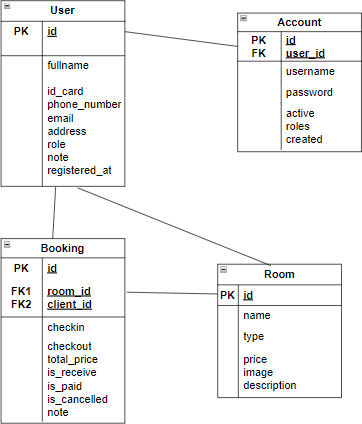
*Hình 1.11. Lược đồ quan hệ phía admin.*

1. Về phía Client



*Hình 1.12. Lược đồ quan hệ phía client.*

1. Về phía Manager



*Hình 1.13. Lược đồ quan hệ phía manager.*

## Đặc tả Usecase.

### Quản lý tài khoản.

|  |
| --- |
| Tên use case: Quản lý tài khoản |
| Actor: Quản trị viên |
| Mô tả: Các bước quản trị viên quản lý tài khoản |
| Điều kiện: Đã đăng nhập vào hệ thống |
| * Tại giao diện Home, Quản trị viên nhấn vào “Quản lý tài khoản”   =>hiển thị giao diện Danh sách tài khoản   * Tại giao diện này, ở mỗi tài khoản muốn thay đổi, quản trị viên có thể:   + Nhấn “ Khóa” để khóa tài khoản người dùng => khóa thành công. Nhấn “Khôi phục” để bỏ khóa tài khoản => khôi phục thành công  + Nhấn “Đổi vai trò quản lý-khách” để đổi lại vai trò của tài khoản từ quản lý sang khách hoặc từ khách sang quản lý => đổi |
| thành công  + Nhấn “Xóa tài khoản” => xóa tài khoản thành công |

*Bảng 1.1. Đặc tả usecase “Quản lý tài khoản”.*

### Quản lý phòng.

1. Thêm phòng

|  |
| --- |
| Tên use case: Thêm phòng |
| Actor: Quản lý |
| Mô tả: Các bước thêm phòng |
| Điều kiện: Đã đăng nhập vào hệ thống |
| * Tại giao diện Home, Quản lý nhấn vào “Quản lý phòng” =>hiển thị giao diện Danh sách phòng * Quản lý nhấn vào “Thêm phòng” =>hiển thị giao diện Thêm phòng * Quản lý nhập tên phòng, tick chọn loại phòng, nhập giá phòng, nhập chi tiết phòng và nhập hình ảnh. Sau đó nhấn “Xác nhận” => Thêm phòng thành công |

*Bảng 1.2. Đặc tả usecase “Thêm phòng”.*

1. Sửa phòng

|  |
| --- |
| Tên use case: Sửa phòng |
| Actor: Quản lý |
| Mô tả: Các bước sửa phòng |
| Điều kiện: Đã đăng nhập vào hệ thống |
| * Tại giao diện Home, Quản lý nhấn vào “Quản lý phòng” =>hiển thị giao diện Danh sách phòng * Quản lý nhấn vào “Chi tiết” phòng muốn sửa=>hiển thị giao diện Thông tin phòng kèm tên phòng * Quản lý chọn “Chỉnh sửa” => hiển thị giao diện Chỉnh sửa thông tin phòng kèm tên phòng * Quản lý tick chọn loại phòng, nhập giá phòng, nhập chi tiết phòng và nhập hình ảnh. Sau đó nhấn “Xác nhận” => Sửa phòng thành công |

*Bảng 1.3. Đặc tả usecase “Sửa phòng”.*

1. Xóa phòng

|  |
| --- |
| Tên use case: Xóa phòng |
| Actor: Quản lý |
| Mô tả: Các bước xóa phòng |
| Điều kiện: Đã đăng nhập vào hệ thống và phòng muốn xóa không được đặt |
| * Tại giao diện Home, Quản lý nhấn vào “Quản lý phòng” =>hiển thị giao diện Danh sách phòng * Quản lý nhấn vào “Chi tiết” phòng muốn sửa=>hiển thị giao diện Thông tin phòng kèm tên phòng * Quản lý chọn “Xóa phòng: <tên phòng>” => Xóa phòng thành công |

*Bảng 1.4. Đặc tả usecase “Xoá phòng”.*

### Quản lý đặt phòng.

1. Đặt phòng.

|  |
| --- |
| Tên usecase: Đặt phòng |
| Actor: Khách hàng đã đăng ký với hệ thống |
| Mô tả: Các bước đặt phòng. |
| Điều kiện: Đã đăng nhập |
| * Tại giao diện Home, khách hàng nhấn nút “Xem danh sách các phòng”   => Hiển thị giao diện danh sách phòng.   * Tại đây người dùng xem thông tin các phòng rồi nhấn đặt phòng, hoặc xem chi tiết phòng => Hiển thị giao diện thông tin chi tiết phòng rồi nhấn đặt phòng.   => Hiển thị giao diện đặt phòng.   * Khách hàng chọn ngày bắt đầu, ngày kết thúc rồi xác nhận.   => Đặt phòng thành công. |

*Bảng 1.5. Đặc tả usecase “Đặt phòng”.*

1. Huỷ phòng bởi khách hàng.

|  |
| --- |
| Tên usecase: Huỷ phòng bởi khách hàng |
| Actor: Khách hàng đã đăng ký với hệ thống |
| Mô tả: Các bước huỷ phòng. |
| Điều kiện: Đã đăng nhập và đã đặt phòng. |
| * Tại giao diện Home, khách hàng nhấn nút “Xem danh sách đặt phòng”   => Hiển thị giao diện danh sách đặt phòng đối với mỗi người dùng.   * Tại đây người dùng nhấn vào nút huỷ đặt phòng.   => Huỷ phòng thành công. |

*Bảng 1.6. Đặc tả usecase “Huỷ phòng bởi khách hàng”.*

1. Huỷ phòng bởi quản lý.

|  |
| --- |
| Tên usecase: Đặt phòng |
| Actor: Quản lý |
| Mô tả: Các bước huỷ phòng. |
| Điều kiện: Đã đăng nhập và khách hàng đã đặt phòng. |
| * Tại giao diện Home, quản lý nhấn nút “Quản lý phòng”   => Hiển thị giao diện danh sách phòng.   * Quản lý chọn nút “Yêu cầu đặt phòng” cho phòng muốn huỷ đặt phòng.   => Hiển thị giao diện danh sách đặt phòng theo mỗi phòng   * Tại đây quản lý nhấn vào nút huỷ đặt phòng cho khách hàng muốn huỷ.   => Huỷ phòng thành công. |

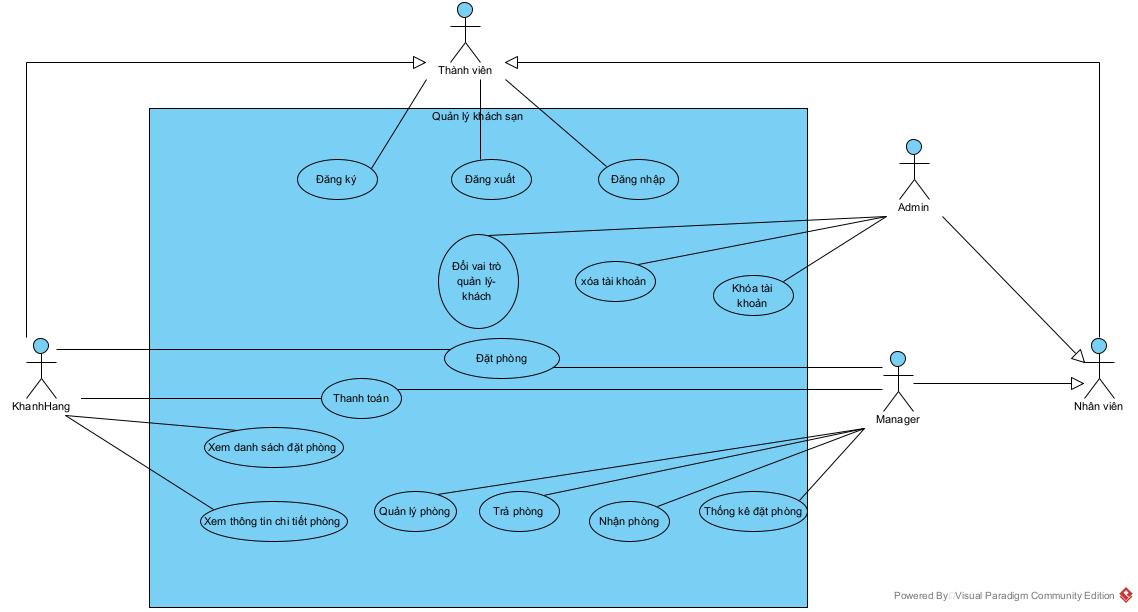
*Bảng 1.7. Đặc tả usecase “Huỷ phòng bởi quản lý”.*

### Xem báo cáo.

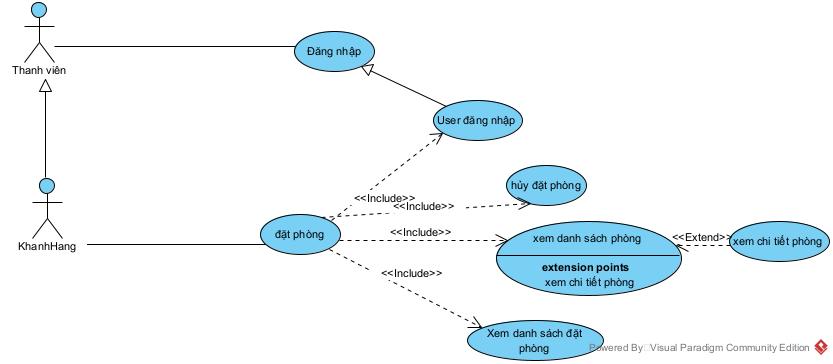
|  |
| --- |
| Tên usecase: Xem báo cáo |
| Actor: Quản lý |
| Mô tả: Các bước xem báo cáo |
| Điều kiện: Đã đăng nhập và khách hàng đã đặt phòng. |
| - Tại giao diện Home, khách hàng nhấn nút “Xem, báo cáo”  => Hiển thị giao diện xem báo cáo. |
| - Quản lý nhập ngày bắt đầu và ngày kết thúc sau đó nhấn “Thống kê”  => Hiển thị bảng đặt phòng được lọc theo ngày tháng trên và tổng doanh thu trong thời gian này. |

*Bảng 1.8. Đặc tả usecase “Xem thống kê”.*

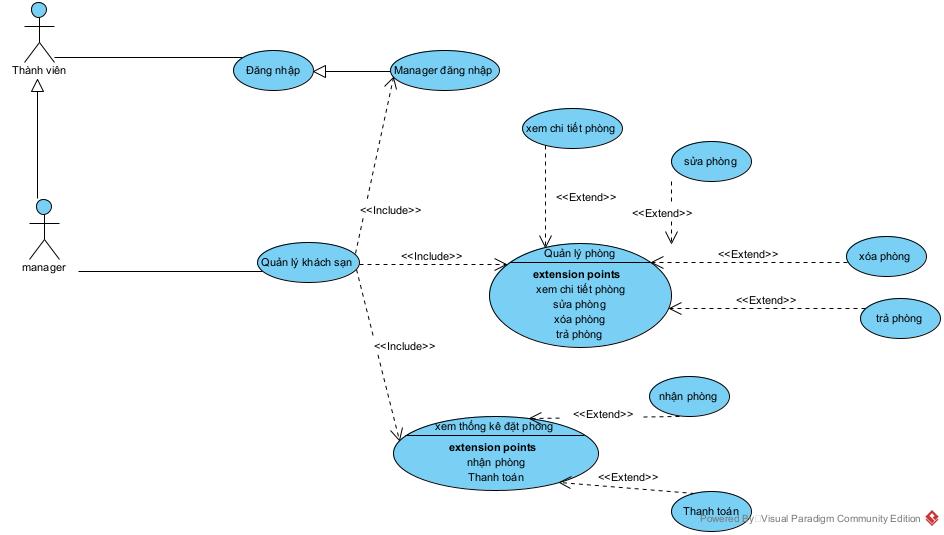
### 1.5.5:Usecase tổng quát:



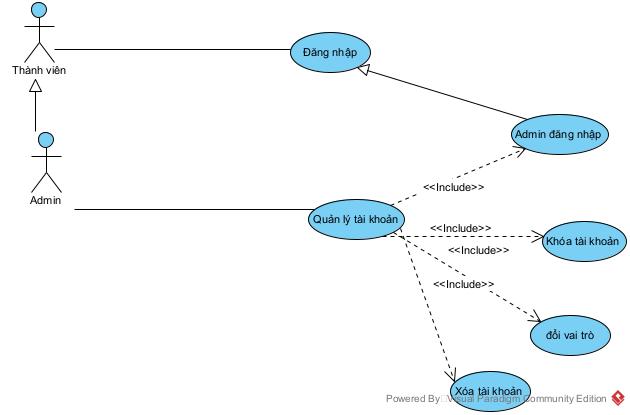
**1.5.6 Usecase tổng quát của khách hàng**



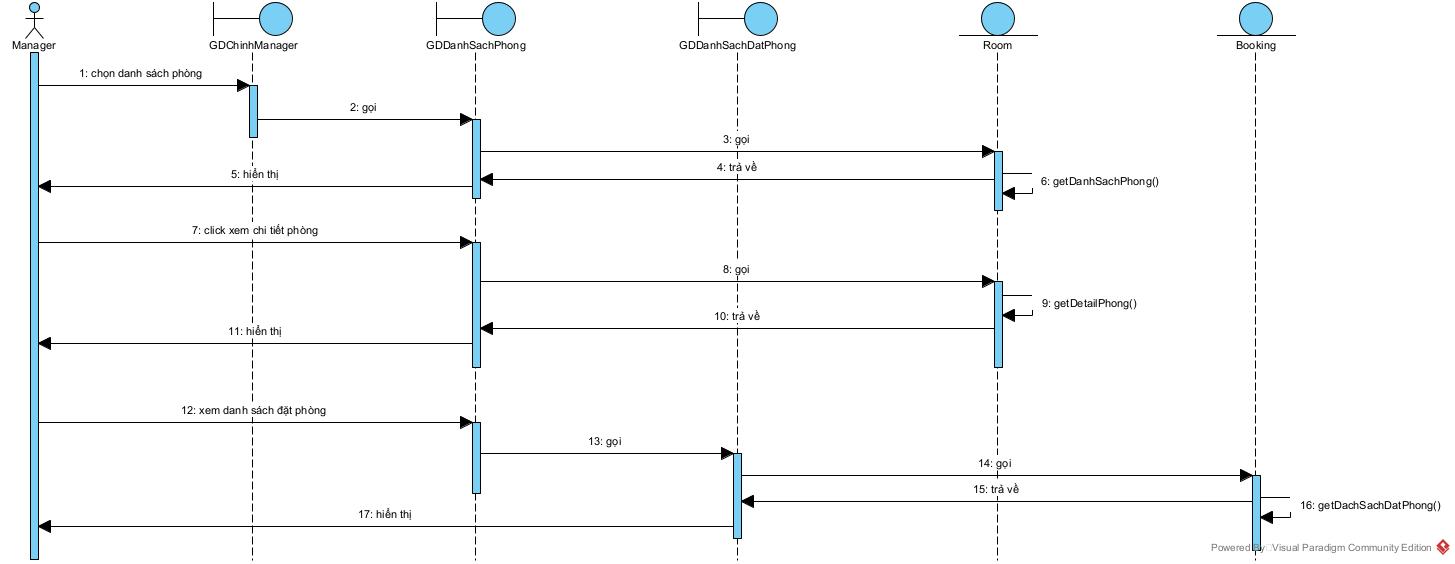
**1.5.7 Usecase tổng quát của manager**



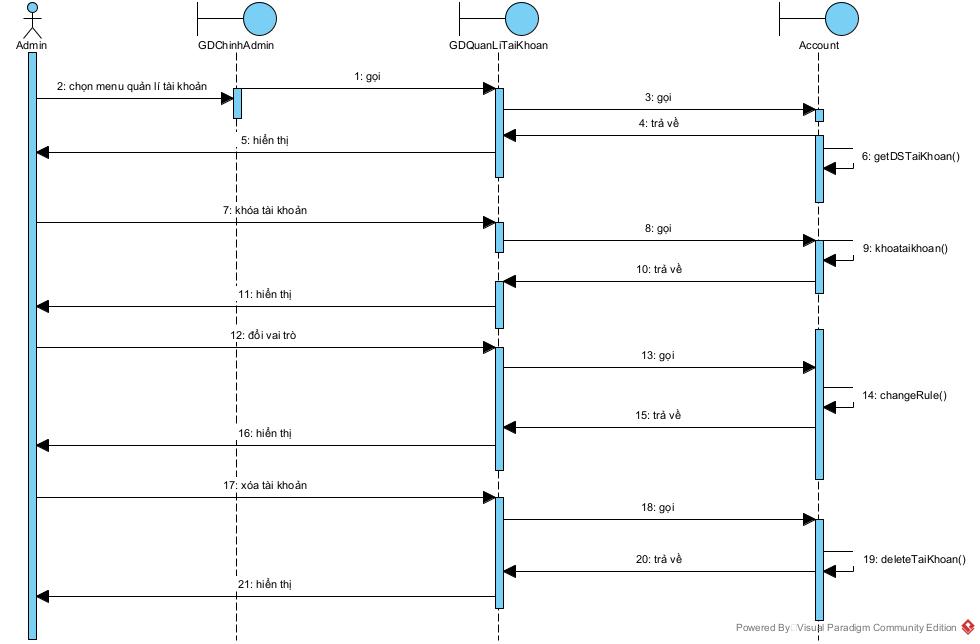
**1.5.8 Usecase tổng quát của admin**



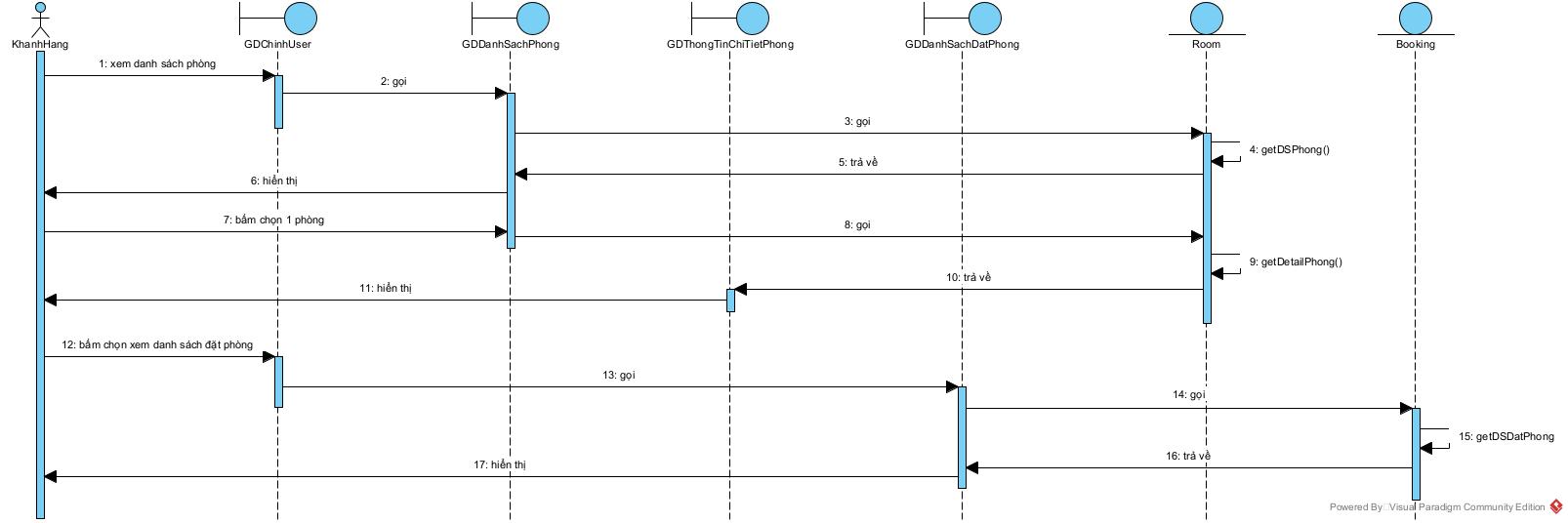
**1.5.9 Biểu đồ tuần tự của Manager**



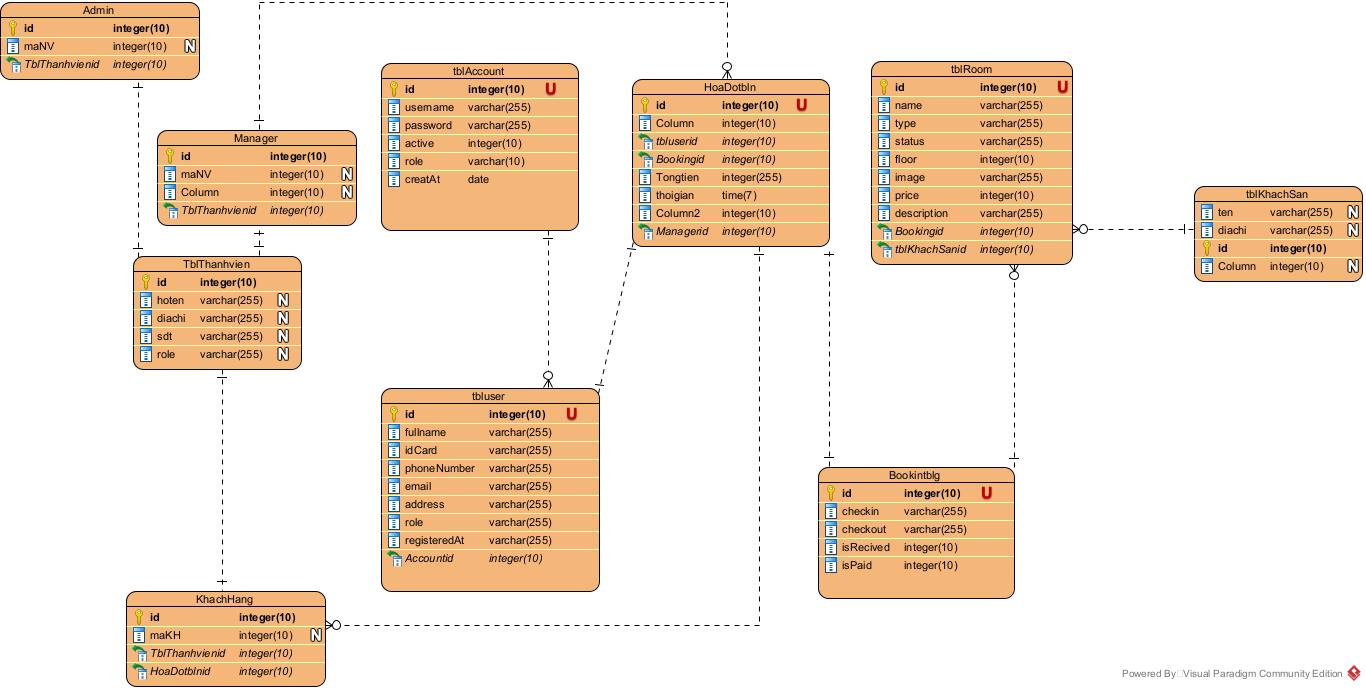
**1.5.10 Biểu đồ tuần tự của admin**



**1.5.11 Biểu đồ tuần tự của khách hàng**



**1.5.12 Lớp thực thế**



# Chương 2. Các giao diện.

## Giao diện Home.

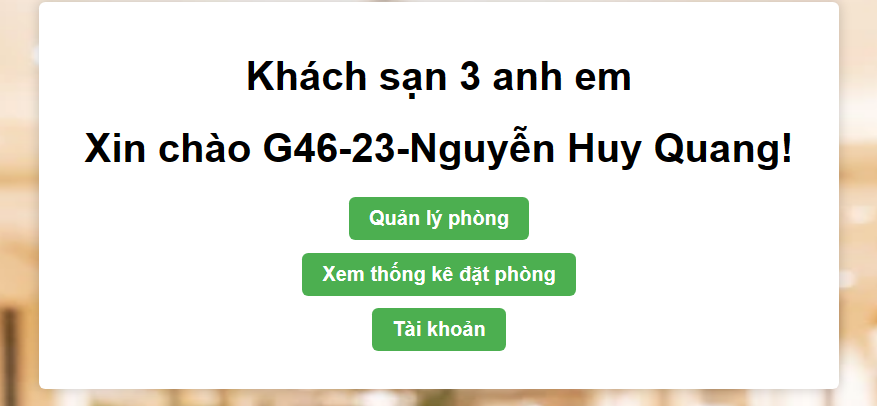
* + - Đây là giao diện bắt đầu khi chương trình được chạy lên.
    - Người dùng chọn chức năng tương ứng với vai trò.

*Hình 2.1. Giao diện Home chưa đăng nhập*

*.*

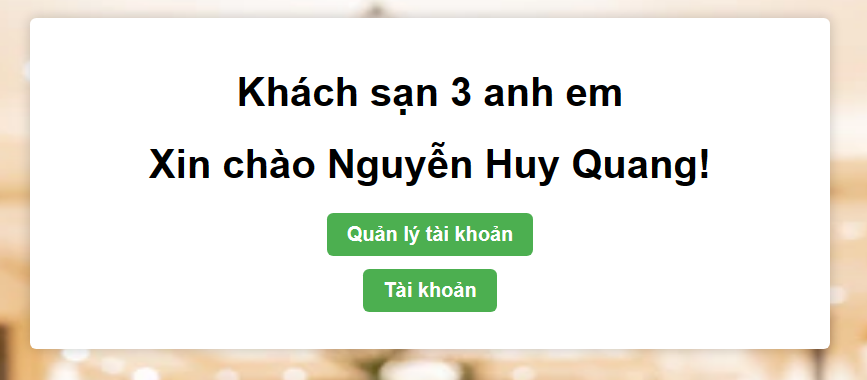
*Hình 2.2. Giao diện Home của quản lý.*

* + - Quản lý có thể nhấn quản lý phòng hoặc xem thống kê hoặc xem thông tin tài khoản.



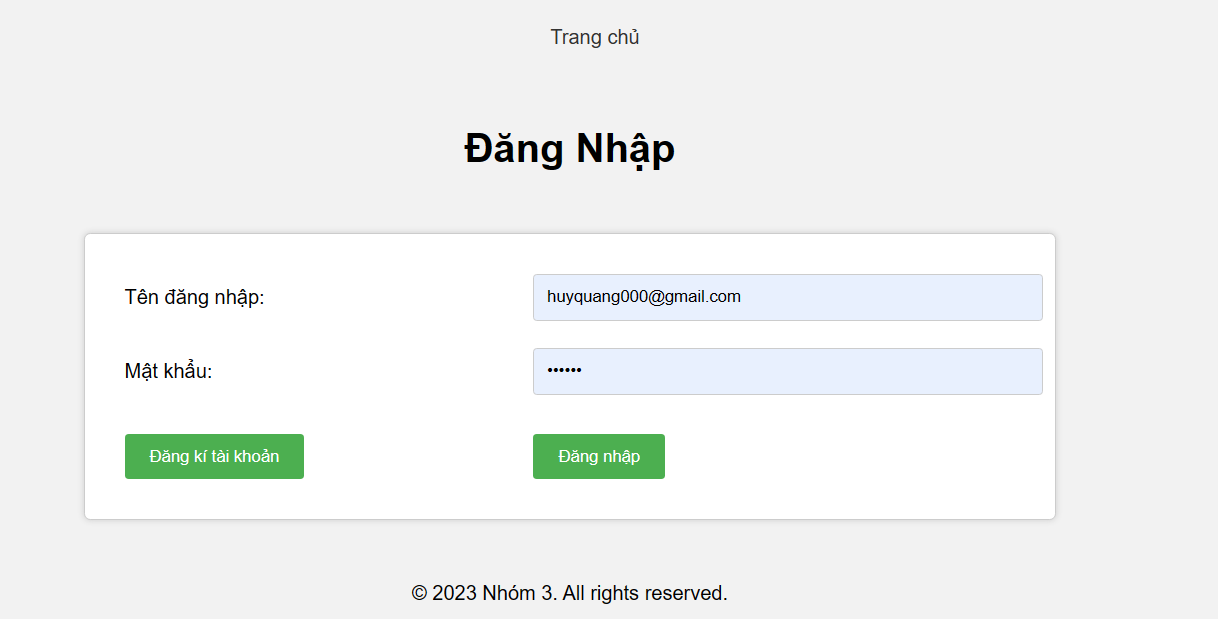
*Hình 2.3. Giao diện Home của khách hàng.*

* + - Khách hàng có thể xem danh sách đặt phòng cũng như là danh sách phòng mà mình đã đặt.
    - Ngoài ra khách hàng có thể xem thông tin tài khoản của mình



*Hình 2.4. Giao diện Home của Admin.*

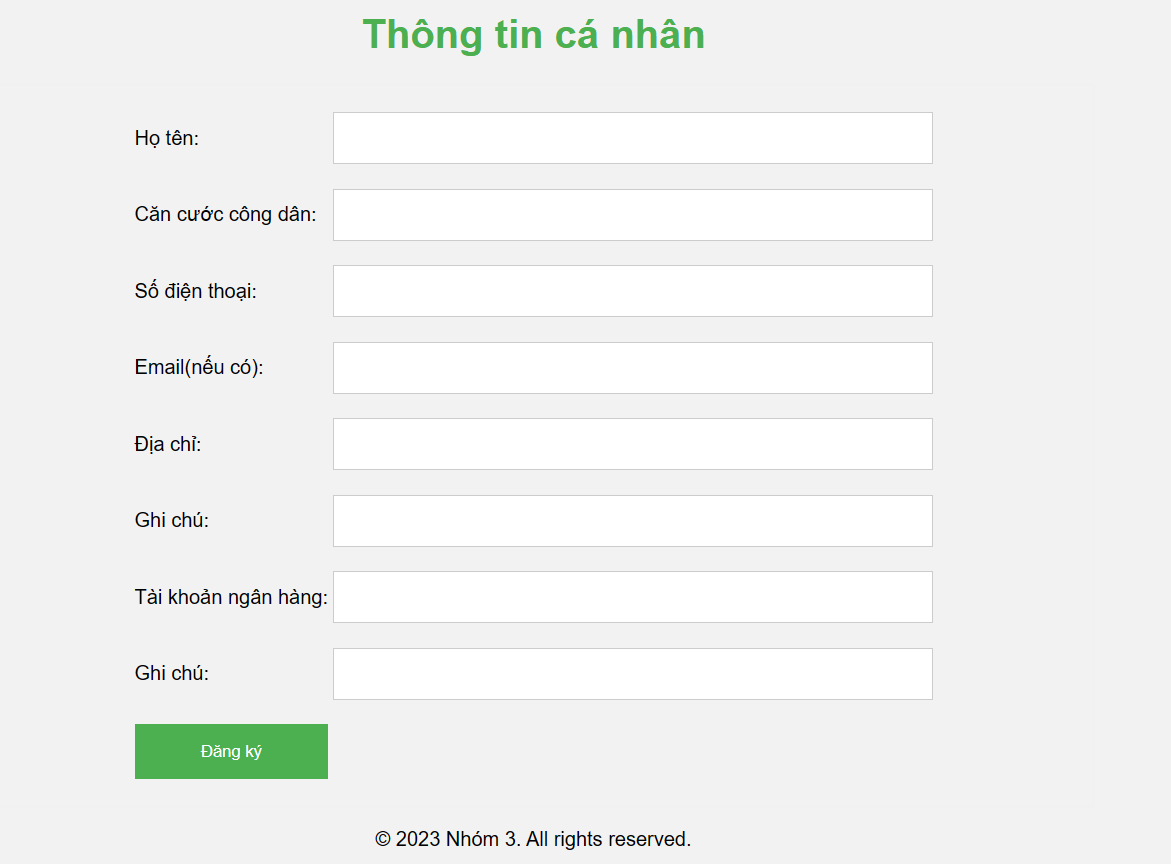
* + - Quản trị viên có thể nhấn quản lý tài khoản để quản lý thông tin các tài khoản.
    - Ngoài ra còn có thể xem thông tin tài khoản của mình.
  1. Đăng nhập.



*Hình 2.5. Giao diện đăng nhập.*

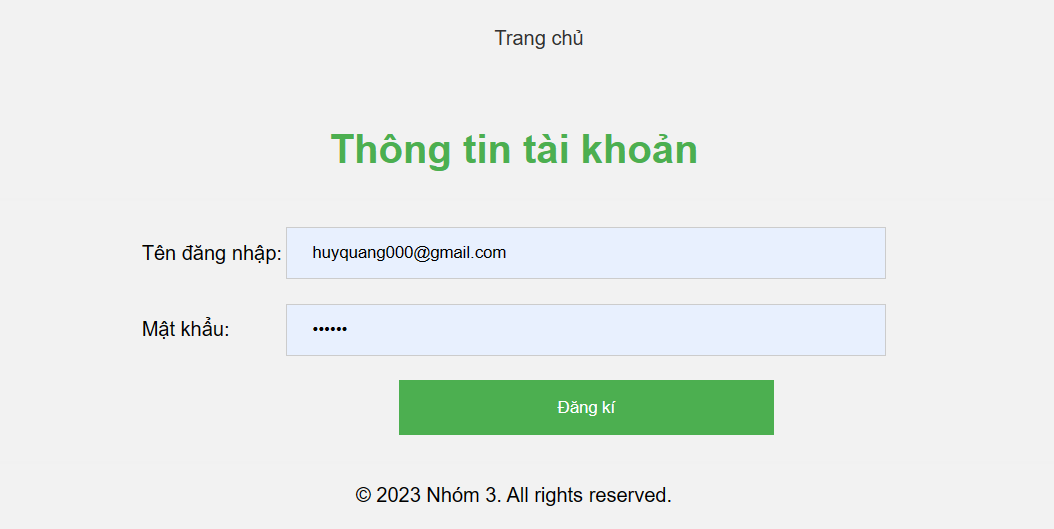
* + - Người dùng nhập tên đăng nhập và mật khẩu rồi nhấn nút đăng nhập để đăng nhập.
    - Nếu chưa có tài khoản thì nhấn đăng ký để tạo tài khoản mới với vai trò khách hàng.

## Đăng ký.

****

*Hình 2.6. Giao diện đăng ký thông tin khách hàng.*

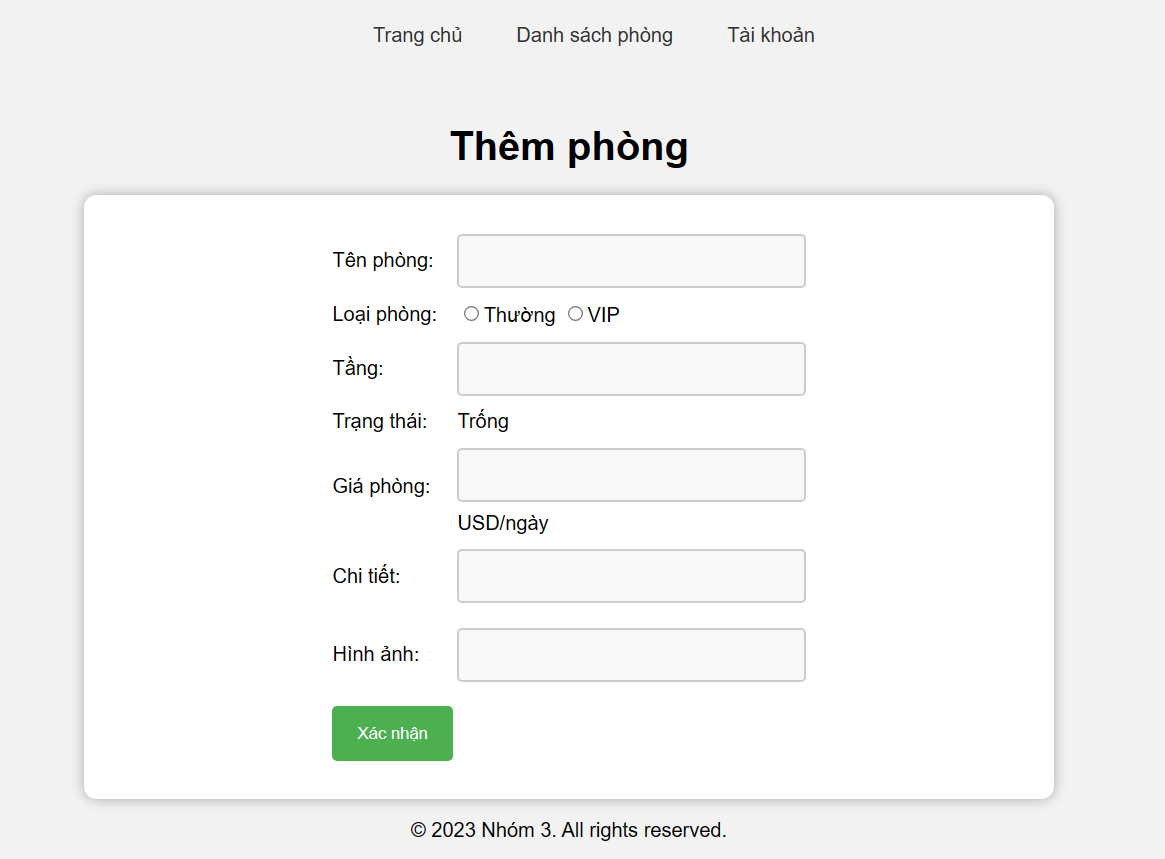
* + - Người dùng nhập các thông tin rồi nhấn vào nút đăng ký để chuyển tới giao diện tạo thông tin người dùng mới.



*Hình 2.7. Giao diện đăng ký người dùng.*

* + - Người dùng nhập tên đăng nhập và mật khẩu để đăng ký tài khoản mới.

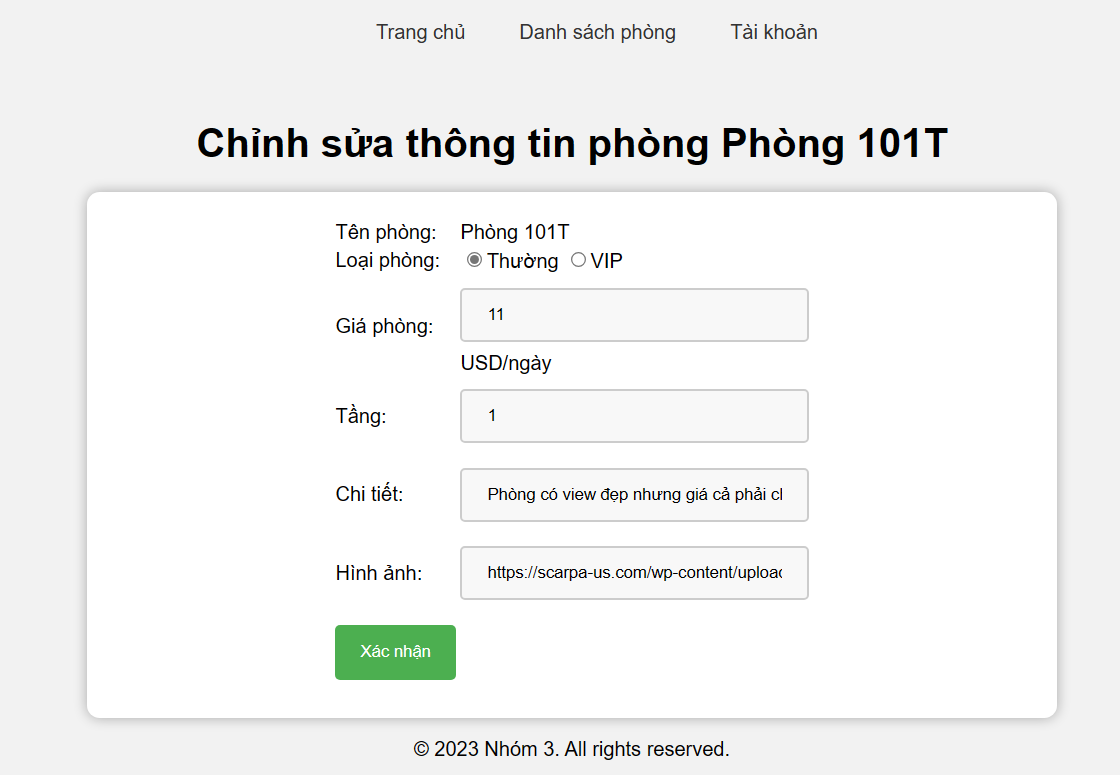
## Thêm phòng.

****

*Hình 2.8. Giao diện thêm phòng.*

* + - Chỉ quản lý mới có thể dùng chức năng này.
    - Quản lý nhập các trường thông tin của phòng để thêm phòng vào hệ thống.

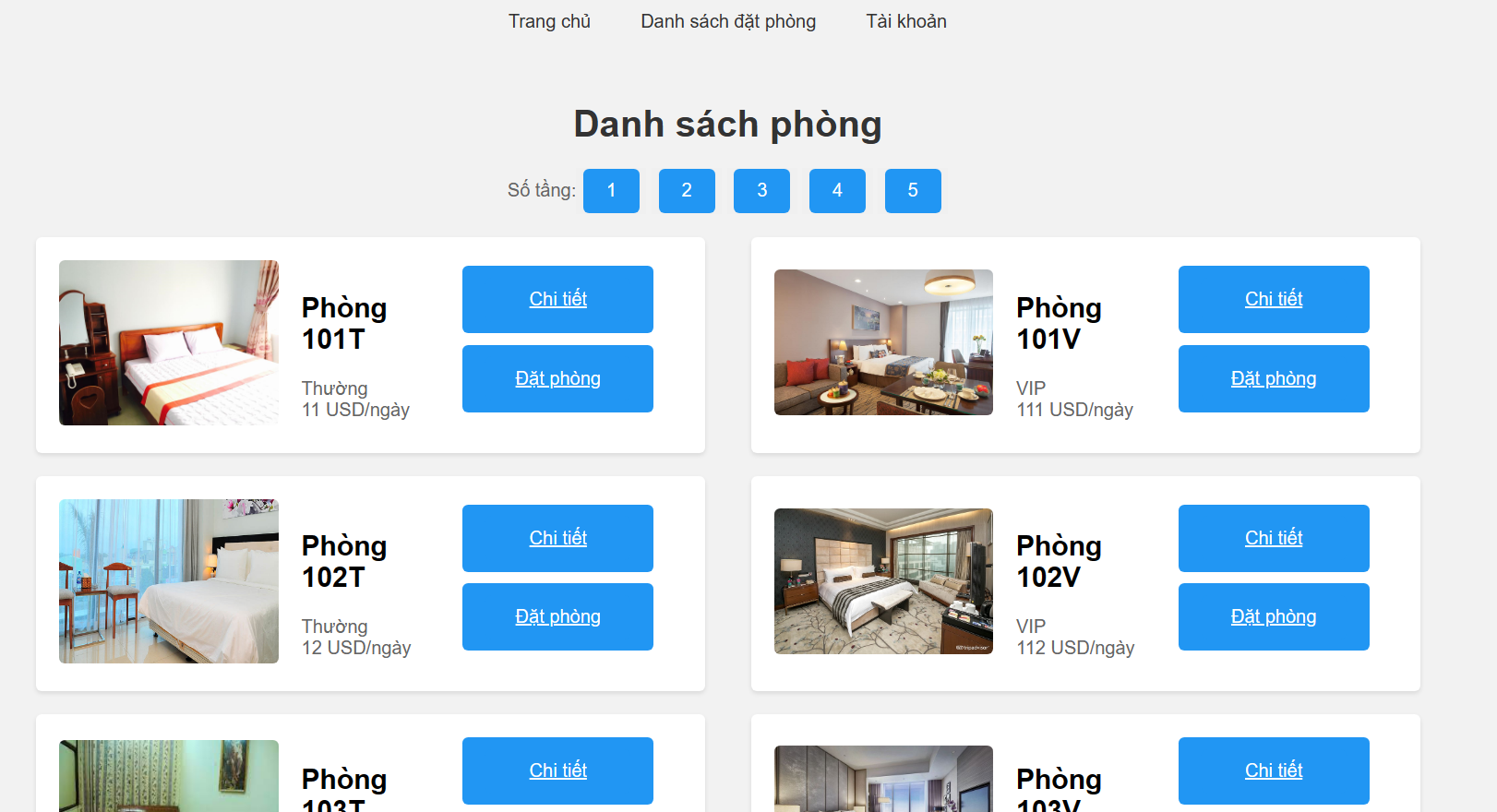
## Chỉnh sửa phòng.

****

*Hình 2.9. Giao diện chỉnh sửa phòng.*

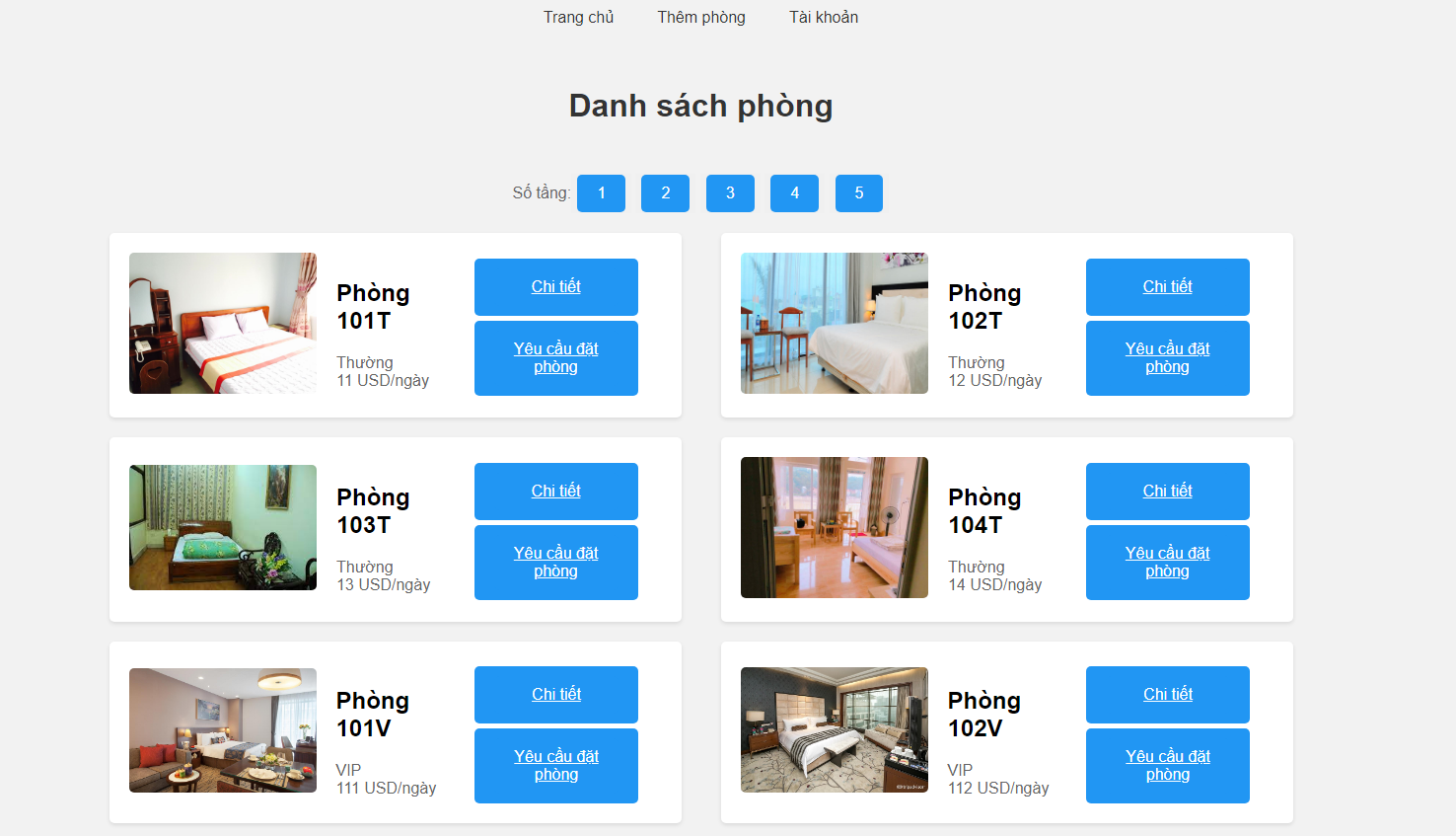
* + - Chỉ quản lý mới có thể xem giao diện này.
    - Quản lý nhập lại thông tin vào phòng rồi lưu lại vào hệ thống.

## Danh sách phòng.

****

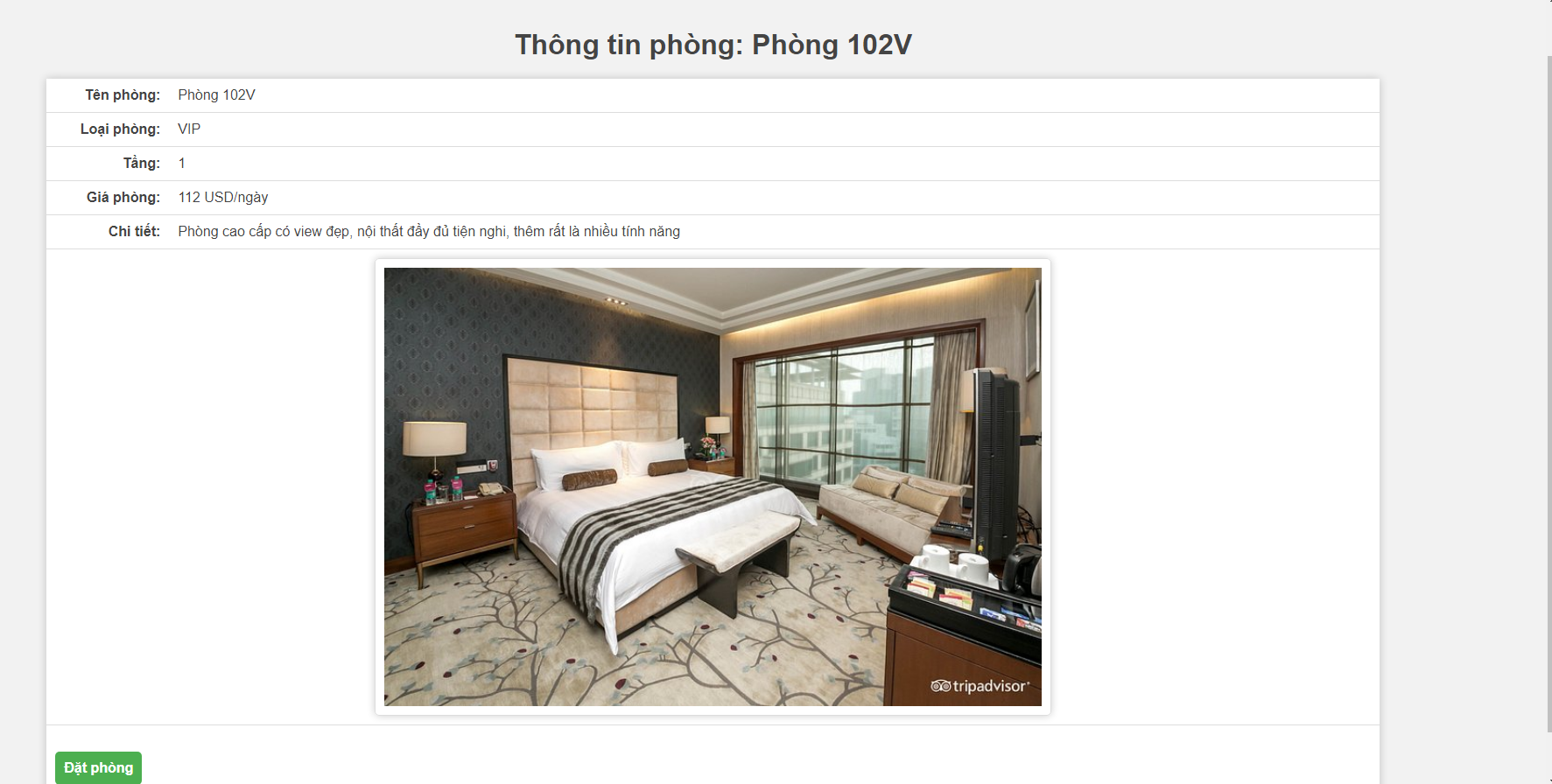
*Hình 2.10. Giao diện danh sách phòng cho khách hàng.*

* + - Khách hàng có thể xem thông tin các phòng.
    - Khách hàng có thể nhấn đặt phòng để xem chuyển tới giao diện đặt phòng.

*Hình 2.11. Giao diện danh sách phòng cho quản lý.*

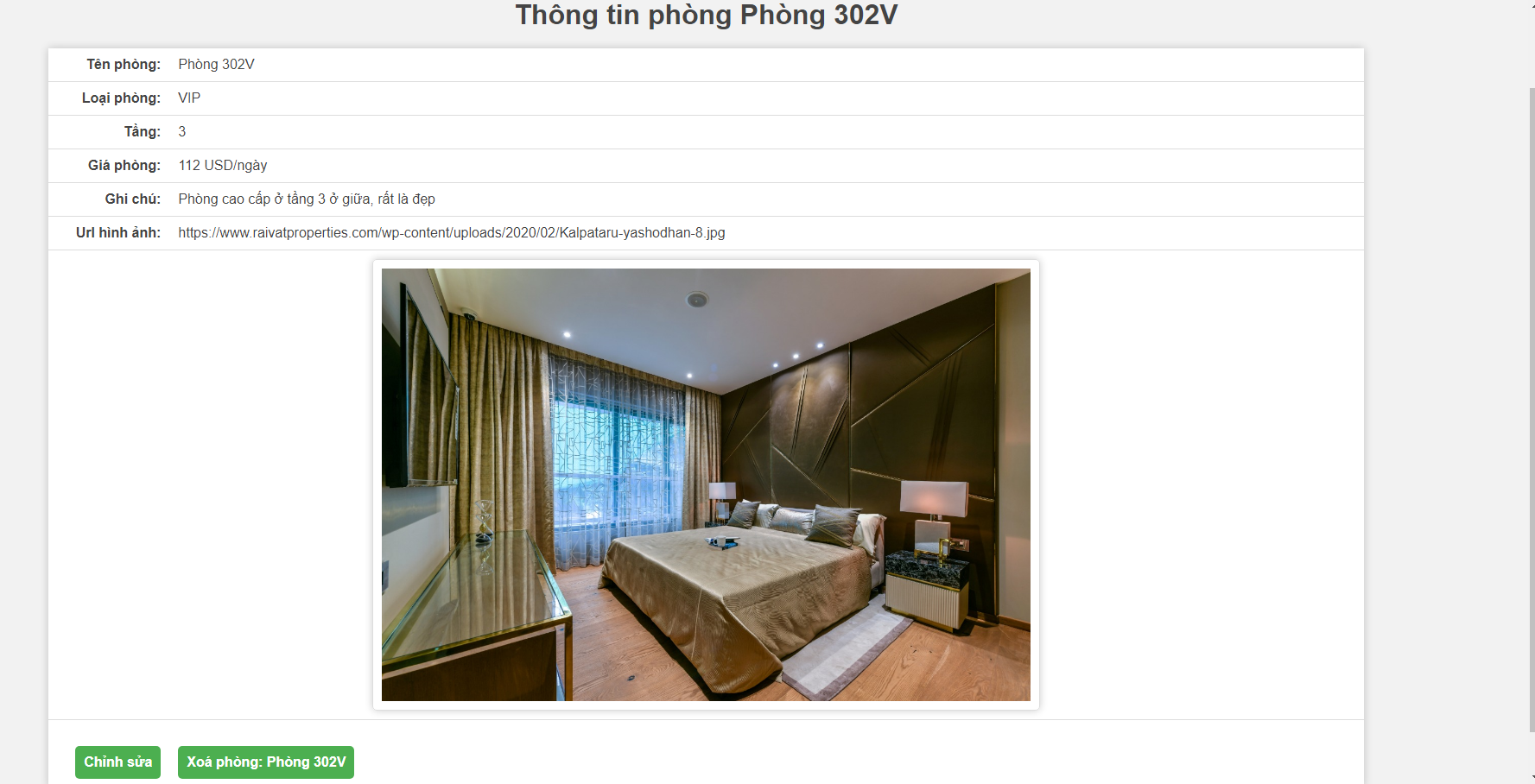
* + - Quản lý có thể xem thông tin các phòng.
    - Quản lý có thể thêm phòng, hoặc xem danh sách yêu cầu đặt phòng cho mỗi phòng.

## Thông tin phòng.

****

*Hình 2.12. Giao diện thông tin chi tiết phòng cho khách hàng.*

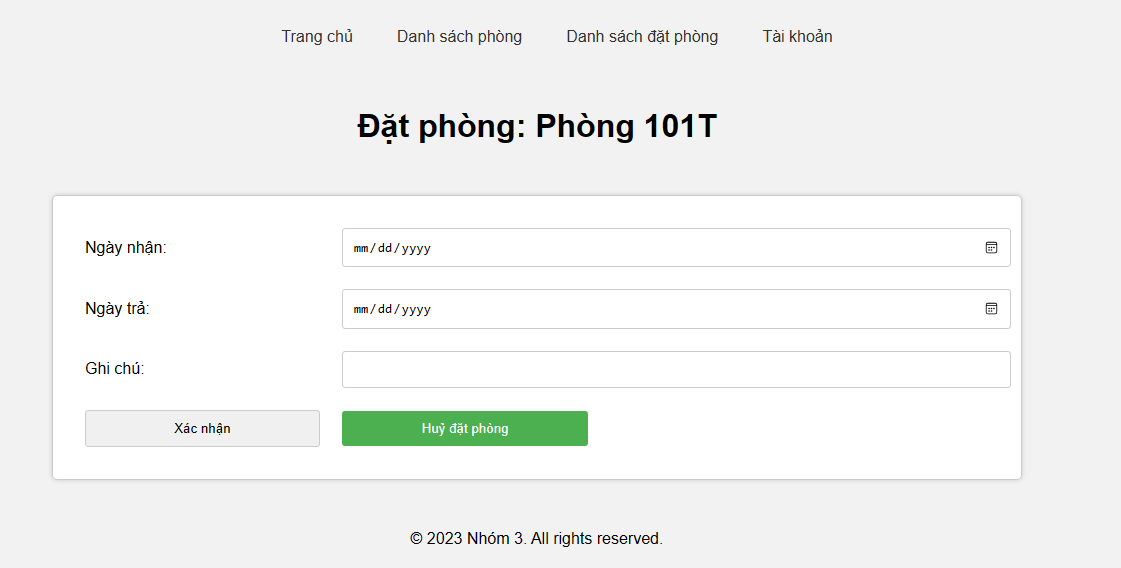
* + - Khách hàng có thể xem cụ thể hơn thông tin mỗi phòng
    - Khách hàng cũng có thể nhấn đặt phòng để chuyển đến giao diện đặt phòng.



*Hình 2.13. Giao diện thông tin chi tiết phòng cho quản lý.*

* + - Quản lý có thể xem thông tin phòng, chỉnh sửa lại, hoặc xóa phòng.

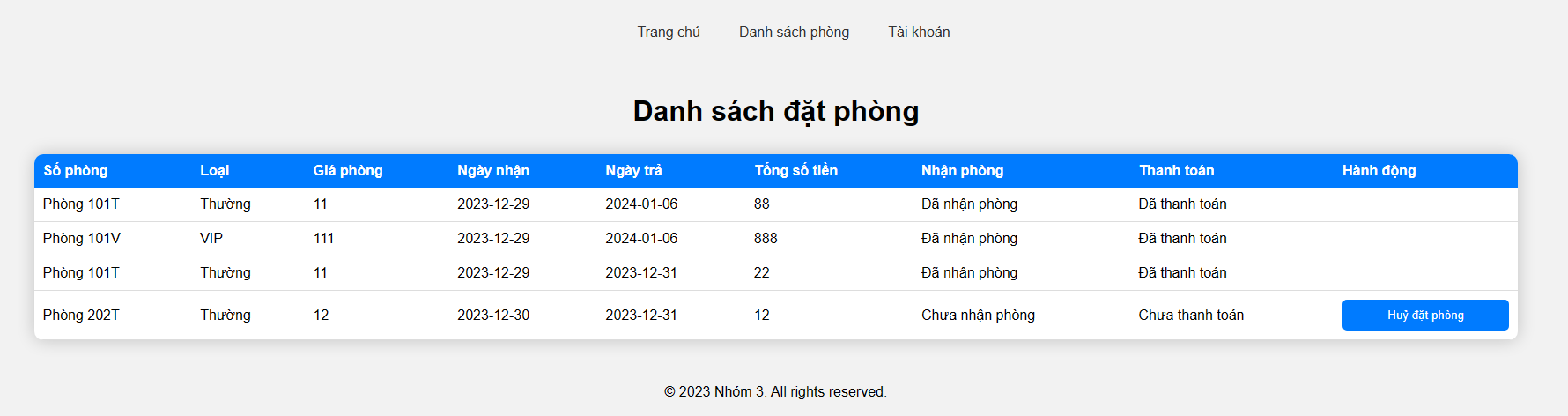
## Đặt phòng.

****

*Hình 2.14. Giao diện đặt phòng.*

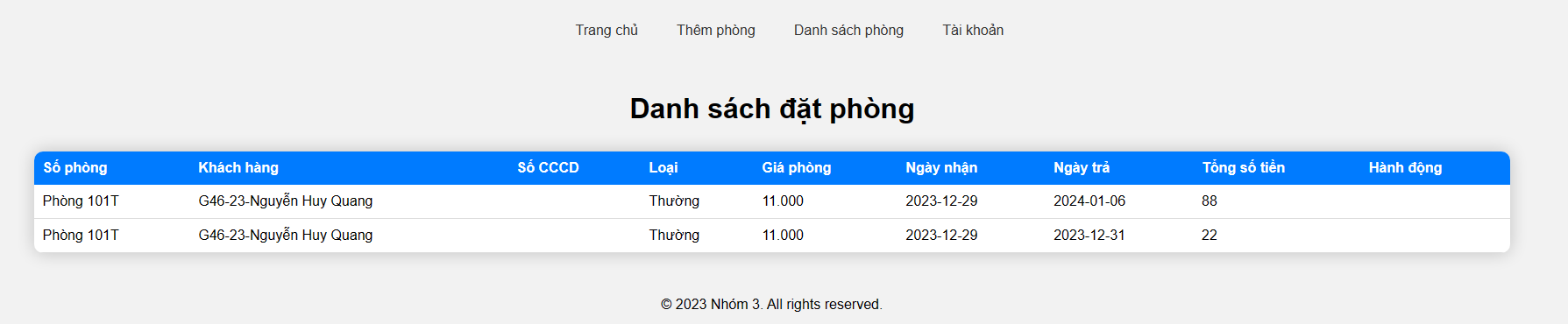
* + - Khách hàng nhập ngày bắt đầu, ngày kết thúc, ghi chú(nếu có) để xác nhận đặt phòng.

## Danh sách đặt phòng.

****

*Hình 2.15. Giao diện danh sách đặt phòng cho mỗi khách hàng.*

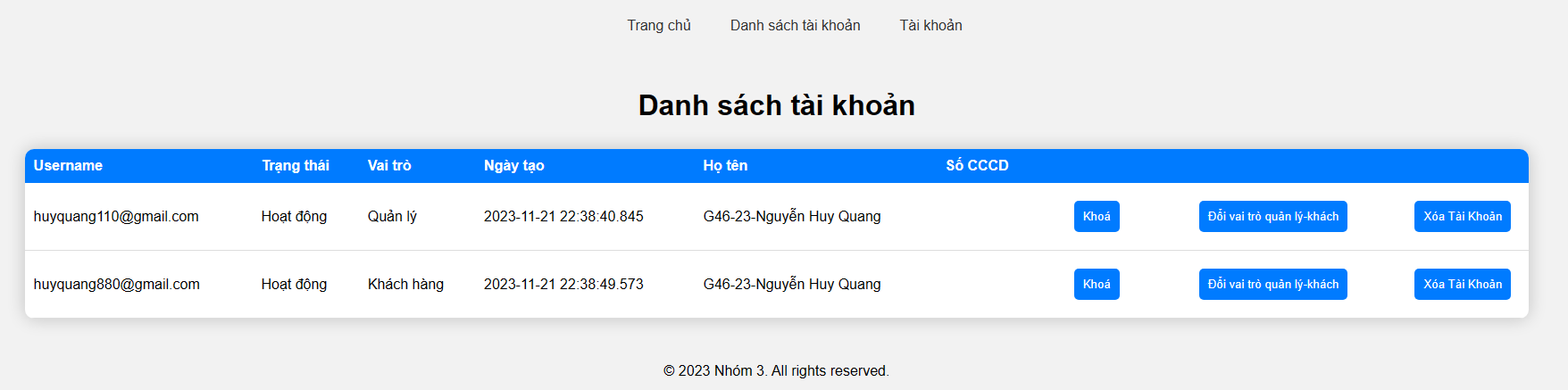
Khách hàng có thể huỷ đặt phòng.



*Hình 2.16. Giao diện danh sách đặt phòng cho mỗi phòng.*

* + - Quản lý mới xem giao diện này.
    - Quản lý có thể huỷ đặt phòng theo yêu cầu khách hàng.

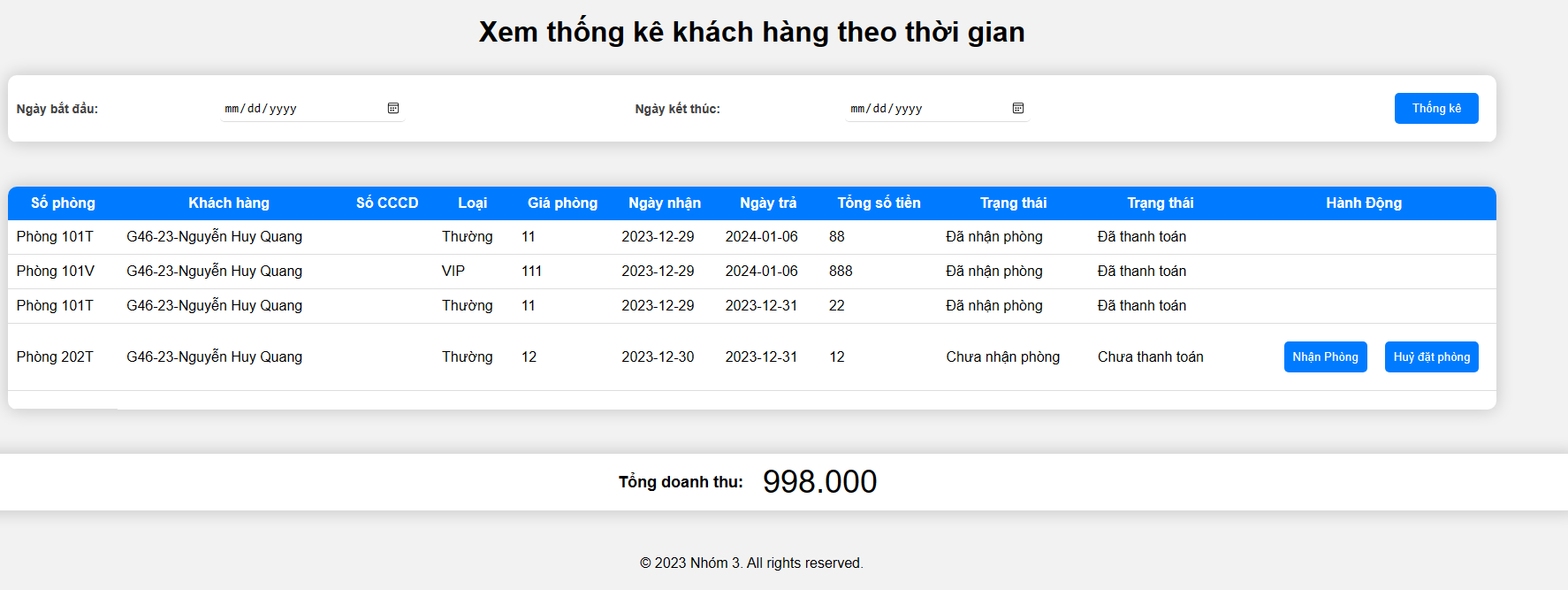
## Quản lý tài khoản.

****

*Hình 2.17. Giao diện danh sách tài khoản.*

* + - Quản trị có thể khoá, thay đổi vai trò hoặc xoá tài khoản.

## Xem báo cáo.

****

*Hình 2.18. Giao diện xem báo cáo.*

* + - Quản lý mới có thể xem giao diện này.
    - Nhập ngày bắt đầu, ngày kết thúc để lọc ra danh sách đặt phòng và tính tổng doanh thu trong thời gian nhập.

# Chương 3. Kết luận, hạn chế.

## Kết luận.

Spring là một trong những framework phổ biến nhất để phát triển các ứng dụng web trên nền tảng Java. Nó được thiết kế để cung cấp một cách tiếp cận linh hoạt và hiệu quả cho việc xây dựng các ứng dụng web và dịch vụ web.

Spring là một trong những framework tốt nhất để lựa chọn. Nó có nhiều tính năng hữu ích để giúp phát triển ứng dụng web, bao gồm cả quản lý session, xử lý các yêu cầu HTTP, bảo mật, quản lý nghiệp vụ, và quản lý các đối tượng trong ứng dụng.

Spring hỗ trợ xây dựng các tính năng và giải pháp hữu ích như thực hiện các truy vấn đến cơ sở dữ liệu, xử lý các tác vụ bất đồng bộ, tạo và quản lý các API RESTful, tương tác với các dịch vụ web khác, và triển khai các ứng dụng trên các nền tảng khác nhau.

Dự án là một trang web quản lý khách sạn, cung cấp danh sách các phòng khác nhau có sẵn cho người dùng đặt phòng theo hình thức online. Dự án sử dụng framework Spring của Java và cơ sở dữ liệu MySQL để quản lý dữ liệu sản phẩm và đơn hàng của khách hàng.

Dự án có 3 giao diện: một dành cho Client, hai dành cho Admin, ba dành cho quản lý.

Bên phía Client, khách hàng có thể lựa chọn xem phòng và đặt phòng cũng như huỷ phòng.

Bên phía Admin, có thể chỉnh sửa thông tin các tài khoản.

Bên phía Quản lý, có thể thêm, sửa, xóa phòng, huỷ đặt phòng và xem báo cáo.

## Hạn chế.

Do thời gian nghiên cứu còn hạn chế nên việc nghiên cứu các vấn đề còn chưa đầy đủ.

Kiến thức thực tế và kiến thức lập trình còn hạn chế nên tính chuyên nghiệp của chương trình chưa cao, bắt lỗi chưa kỹ ở một số chỗ và một số tính năng chưa được tối ưu.

Chương trình chưa thực sự đầy đủ các tính năng như mong muốn.

Slide:

**Tài liệu tham khảo**

+ Session1\_Spring Basic.pptx

+ Session2\_Spring Data JDBC.pptx

+ Session3\_Sping Data JPA.pptx Tìm hiểu về Spring JPA:

+ <https://docs.spring.io/spring-data/jpa/docs/current/reference/html/>

+ <https://docs.spring.io/spring-data/jpa/docs/3.1.0-RC1/reference/html/>