**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**ĐỀ TÀI: XÂY DỰNG HỆ THỐNG ĐẶT PHÒNG KHÁCH SẠN VÀ QUẢN LÝ THÔNG TIN KHÁCH HÀNG**

**Giáo viên hướng dẫn: ThS. Kiều Tuấn Dũng**

**Sinh viên thực hiện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Mã sv** | **Họ và tên** | **Lớp** |
| 01 | 177102334 | Nguyễn Hữu Hưng | CNTT-1704 |

**Hà Nội, năm 2024**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**ĐỀ TÀI: XÂY DỰNG HỆ THỐNG ĐẶT PHÒNG KHÁCH SẠN VÀ QUẢN LÝ THÔNG TIN KHÁCH HÀNG**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1771020334 | Nguyễn Hữu Hưng | 07/09/2005 |  |  |

### 

### CÁN BỘ CHẤM THI

**Hà Nội, năm 2024**

# LỜI NÓI ĐẦU

Trong bối cảnh phát triển nhanh chóng của ngành du lịch và dịch vụ lưu trú, việc xây dựng một hệ thống đặt phòng khách sạn hiệu quả và quản lý thông tin khách hàng trở thành một yêu cầu thiết yếu. Ngành khách sạn không chỉ cần cung cấp dịch vụ lưu trú chất lượng mà còn phải tối ưu hóa quy trình đặt phòng, nâng cao trải nghiệm của khách hàng và quản lý thông tin một cách chính xác và an toàn.

Bài tập lớn này với chủ đề "Xây dựng hệ thống đặt phòng khách sạn và quản lý thông tin khách hàng" được thực hiện nhằm áp dụng kiến thức lập trình hướng đối tượng Java vào việc phát triển một ứng dụng thực tiễn. Hệ thống này sẽ bao gồm các chức năng chính như đặt phòng, hủy phòng, kiểm tra tình trạng phòng trống, và quản lý thông tin khách hàng.

Ứng dụng này có tác dụng quan trọng trong việc nâng cao hiệu quả hoạt động và trải nghiệm của khách hàng trong thực tiễn. Ứng dụng được sử dụng bởi các quản lý và lễ tân, giúp quá trình làm việc trở nên nhẹ nhàng và hiệu quả hơn.

Thông qua việc nghiên cứu và phát triển hệ thống, chúng em không chỉ mong muốn nâng cao kỹ năng lập trình mà còn hiểu sâu hơn về quy trình hoạt động của ngành khách sạn. Bên cạnh đó, bài tập này cũng nhằm tạo ra một nền tảng cho việc triển khai các tính năng phức tạp hơn trong tương lai, góp phần vào sự phát triển bền vững của ngành du lịch.

**MỤC LỤC**

**[LỜI NÓI ĐẦU 3](#_Toc18913)**

**[CHƯƠNG 1. MÔ TẢ BÀI TOÁN 4](#_Toc9431)**

**[1.1 Giới thiệu 4](#_Toc30330)**

**[1.2 Chức năng chính 5](#_Toc3761)**

**[1.3 Yêu cầu phi chức năng 7](#_Toc16541)**

**[CHƯƠNG 2. PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG 9](#_Toc5462)**

**[2.1. Phân tích yêu cầu 9](#_Toc2336)**

**[2.2. Thiết kế hệ thống 13](#_Toc910)**

**[2.2.1. Xác định các lớp 13](#_Toc20437)**

**[Class Customer 13](#_Toc16765)**

**[2.2.2. Thiết kế biểu đồ lớp 16](#_Toc3450)**

**[2.3. Triển Khai 18](#_Toc6485)**

**[2.3.1. Xây dựng các lớp theo mô hình MVC: 18](#_Toc28225)**

**[2.3.2 Xử lý đọc/ghi file: 19](#_Toc28214)**

**[2.3.3 Giao diện console: 19](#_Toc31127)**

**[2.3.4. Kiểm thử: 20](#_Toc24424)**

**[2.3.5. Tối ưu hóa: 20](#_Toc22794)**

**[CHƯƠNG 3. KẾT QUẢ THỰC HIỆN 21](#_Toc25112)**

**[3.1. Công nghệ đã sử dụng 21](#_Toc16257)**

**[3.2. Tiến độ thực hiện 21](#_Toc28216)**

**[3.1.2 Chức năng quản lý phòng 21](#_Toc15147)**

**[2.3.2. Quản lý thống kê 34](#_Toc32571)**

**[KẾT LUẬN 37](#_Toc20418)**

# CHƯƠNG 1. MÔ TẢ BÀI TOÁN

## 1.1 Giới thiệu

Ứng dụng “hệ thống đặt phòng khách sạn và quản lý thông tin khách hàng” được xây dựng nhằm tối ưu hóa quy trình đặt phòng và quản lý khách hàng trong lĩnh vực khách sạn. Sử dụng ngôn ngữ lập trình Java và các nguyên tắc lập trình hướng đối tượng, hệ thống này giúp quản lý các thao tác như đặt phòng, hủy phòng, kiểm tra tình trạng phòng trống, cũng như lưu trữ và cập nhật thông tin khách hàng một cách hiệu quả.

Hệ thống không chỉ đáp ứng nhu cầu cơ bản của một khách sạn trong việc quản lý phòng và khách hàng, mà còn hỗ trợ các chức năng mở rộng như quản lý dịch vụ bổ sung, tạo hóa đơn, và thống kê doanh thu. Ứng dụng sẽ có giao diện console, cung cấp trải nghiệm mượt mà cho người quản lý khách sạn và nhân viên lễ tân, đồng thời đảm bảo tính bảo mật và toàn vẹn dữ liệu.

## 1.2 Chức năng chính

Ứng dụng cần có các chức năng sau:

* **Quản lý phòng:**
* Số phòng: Số thứ tự phòng, thường hiển thị trực tiếp trên cửa phòng. Cho phép người dùng nhập một số nguyên biểu thị số phòng.
* **Thêm thông tin phòng:** Cho phép thêm mới các phòng vào danh sách khách sạn với các thông tin như mã phòng, loại phòng (Standard, Deluxe), giá phòng, số lượng, và dịch vụ đi kèm.
* **Sửa thông tin phòng:** Cập nhật thông tin phòng hiện có, bao gồm thay đổi giá, loại phòng, hoặc trạng thái (trống, đang được đặt, cần bảo trì).
* **Xóa phòng:** Xóa một hoặc nhiều phòng không còn sử dụng khỏi hệ thống.
* **Xem danh sách các phòng còn trống:** Hiển thị danh sách các phòng sẵn sàng cho khách hàng đặt, có thể lọc theo loại phòng hoặc mức giá.
* **Quản lý khách hàng**
* **Thêm thông tin khách hàng:** Thêm mới thông tin khách hàng, bao gồm tên, số điện thoại, email, và địa chỉ.
* **Sửa thông tin khách hàng:** Cho phép chỉnh sửa các thông tin của khách hàng đã có, chẳng hạn như cập nhật số điện thoại hoặc email.
* **Xóa thông tin khách hàng:** Xóa thông tin của khách hàng khi không còn cần thiết hoặc do yêu cầu từ khách hàng.
* **Tìm kiếm thông tin khách hàng:** Tìm kiếm khách hàng theo các tiêu chí như tên, số điện thoại, hoặc mã khách hàng.
* **Quản lý đặt phòng**
* **Thực hiện đặt phòng:** Cho phép khách hàng đặt phòng bằng cách chọn phòng còn trống, nhập ngày check-in, ngày check-out, và thông tin khách hàng.
* **Hủy đặt phòng:** Hủy một đơn đặt phòng đã tạo trước đó và cập nhật trạng thái của phòng về "Trống".
* **Cập nhật thông tin đặt phòng:** Thay đổi thông tin của một đơn đặt phòng như ngày check-in, ngày check-out, hoặc thay đổi phòng khác.
* **Chức năng quản lý lịch sử đặt phòng**
* **Tự động ghi nhận lịch sử sau mỗi lần đặt phòng:** Bao gồm thông tin khách hàng, phòng, ngày nhận/trả phòng, trạng thái giao dịch.
* **Xem lịch sử đặt phòng:** Theo từng khách hàng hoặc theo phòng.
* **Lọc lịch sử:** Theo thời gian (ngày, tháng, năm) hoặc trạng thái (đã hoàn thành, hủy).
* **Chức năng quản lý hóa đơn**
* **Tạo hóa đơn sau khi thanh toán thành công**:
  + - Khi khách hàng thanh toán giao dịch đặt phòng, hệ thống sẽ tự động tạo hóa đơn.
    - Thông tin hóa đơn bao gồm:
      * Mã hóa đơn.
      * Giao dịch đặt phòng liên quan.
      * Ngày lập hóa đơn.
      * Tổng số tiền.
      * Tình trạng: "Đã thanh toán".
* **In hóa đơn**:
  + - Hóa đơn chỉ được in sau khi giao dịch đã được thanh toán.
    - Định dạng hóa đơn (in ra file txt hoặc Excel) sẽ gồm:
      * Thông tin khách hàng.
      * Thông tin phòng đã đặt.
      * Ngày nhận/trả phòng.
      * Tổng tiền.
      * Ngày thanh toán.
* **Theo dõi tình trạng thanh toán**:
  + - Hóa đơn sẽ có hai trạng thái:
      * **"Chưa thanh toán"**: Nếu khách hàng chưa thanh toán, hóa đơn không được phép in.
      * **"Đã thanh toán"**: Khi khách hàng hoàn thành thanh toán, hóa đơn chuyển sang trạng thái này và có thể in.
* **Kiểm tra danh sách hóa đơn**:
  + - Xem danh sách các hóa đơn đã thanh toán và chưa thanh toán.
    - Cho phép lọc theo thời gian, khách hàng, hoặc trạng thái thanh toán.
* **Chức năng thống kê:**
* **Thống kê số lượng phòng đã đặt:** Theo ngày, tháng, hoặc năm.
* **Thống kê doanh thu:** Tổng doanh thu theo khoảng thời gian cụ thể.
* **Thống kê hoạt động khách hàng:** Danh sách khách hàng đặt phòng nhiều nhất.

## 1.3 Yêu cầu phi chức năng

* **Dễ sử dụng:** Giao diện console cần rõ ràng và dễ sử dụng.
* **Hiệu năng:** Ứng dụng cần hoạt động nhanh chóng và hiệu quả.
* **Độ tin cậy:** Dữ liệu công việc cần được lưu trữ an toàn và không bị mất mát.
* **Khả năng mở rộng đơn giản:**
* Hệ thống phải dễ dàng mở rộng để thêm nhiều khách hàng hoặc phòng mà không cần thay đổi cấu trúc chính của ứng dụng.
* Các chức năng mới (như báo cáo doanh thu, quản lý dịch vụ bổ sung) có thể được thêm vào mà không ảnh hưởng đến các chức năng hiện tại.
* Khả năng bảo trì:
* Mã nguồn cần được viết rõ ràng, dễ đọc và dễ bảo trì, tuân thủ các nguyên tắc chuẩn hoá chung
* Cấu trúc mã nguồn phải dễ hiểu để có thể mở rộng thêm chức năng hoặc sửa lỗi một cách nhanh chóng.
* Hệ thống cần có tài liệu hoặc chú thích trong mã để hướng dẫn các lập trình viên khác khi bảo trì.
* Tính thân thiện với người dùng:
* Giao diện console phải thân thiện, dễ sử dụng, các hướng dẫn và lệnh được hiển thị rõ ràng.
* Hệ thống phải cung cấp thông báo lỗi dễ hiểu khi người dùng nhập sai hoặc thực hiện thao tác không hợp lệ.
* Người dùng cần có khả năng điều hướng các chức năng chính dễ dàng qua các menu hoặc dòng lệnh đơn giản.
* Tính tương thích: Hệ thống phải chạy được trên các thiết bị khác nhau (điều chỉnh các liên kết để tương thích khi chạy trên thiết bị khác).

# CHƯƠNG 2. PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG

## 2.1. Phân tích yêu cầu

**Xác định người dùng:**

* Người dùng cuối có thể là các quản lý trong khách sạn hay các lễ tân.
* Người dùng có thể có các mức độ am hiểu về máy tính khác nhau, do đó ứng dụng cần dễ sử dụng và thân thiện với người dùng.

**Thu thập yêu cầu:**

* Dựa trên mô tả bài toán, đã xác định được các chức năng chính và các yêu cầu phi chức năng của ứng dụng.

**Phân tích yêu cầu:**

* **Thêm phòng mới:** yêu cầu người dùng nhập:
* Mã phòng: Là mã duy nhất để xác định phòng với kiểu dữ liệu String, khi người dùng nhập yêu cầu không trùng lặp với các phòng đã có
* Số phòng: Là số định danh cho phòng với kiểu dữ liệu String ( có thể là số hoặc chữ tùy người nhập)
* Loại phòng: Kiểu phòng “Standard” và “Deluxe” kiểu dữ liệu String, khi nhập chỉ cho phép nhập các giá trị hợp lệ như “Standard” và “Deluxe”
* Trạng thái: Tình trạng của phòng (trống, đã đặt, đang bảo trì) với kiểu dữ String, khi nhập yêu cầu giá trị hợp lệ là "Trống", "Đã đặt", "Bảo trì"
* Gía phòng: Giá thuê phòng niêm yết của khách sạn( theo đêm) với kiểu dữ liệu Float hoặc Integer, khi nhập giá trị của phòng phải lớn hơn 0
* Mô tả : Hiển thị thông tin phòng(view, nội thất,...)
* Dịch vụ đi kem: Danh sách các các dịch vụ (ăn sáng, dọn phòng,...) kiểu dữ liệu List hoặc String
* **Hiển thị danh sách các phòng:** In ra thông tin các phòng bao gồm: Mã phòng, số phòng, loại phòng, trạng thái, giá phòng , mô tả, dịch vụ đi kèm
* **Hiển thị danh sách các phòng đang trống:** In ra thông tin của các phòng như chức năng hiển thị danh sách các phòng nhưng chỉ in ra các phòng có phần trạng thái là “Trống”.
* **Chỉnh sửa thông tin của phòng:** Cho người dùng nhập vào mã phòng (xác minh xem phòng đó có tồn tại hay không, nếu không thì cho nhập lại), nếu tồn tại thì cho người dùng nhập vào các thông tin:
* Mã Phòng
* Số phòng
* Loại phòng (Lựa chọn giữa “Standard” và “Deluxe”)
* Gía phòng ( giá phòng phải lớn hơn 0)
* Mô tả
* Dịch vụ đi kèm
* **Xóa phòng:** Cho phép người dùng nhập vào mã phòng (xác minh xem phòng đó có tồn tại hay không, nếu không thì cho nhập lại) nếu tồn tại thì sẽ hỏi người dùng có chắc chắn muốn xóa phòng này không, nếu người dùng chọn ‘y’ thì xóa phòng đó khỏi danh sách.
* **Lọc danh sách phòng:** Cho phép người dùng lọc ra các phòng “**Standard**” và “**Deluxe**” (phải thỏa mãn tiêu chí lọc, nếu không có phòng nào phù hợp thì thông báo “Không tìm thấy phòng phù hợp”)
* **Tìm kiếm phòng:** Cho phép người dùng nhập vào mã phòng muốn tìm (xác minh xem phòng đó có tồn tại hay không, nếu không thì thông báo “Không có phòng này“), nếu có thì in ra tất cả thông tin của phòng đó
* **Thêm khách hàng:** Cho phép người dùng thêm 1 khách hàng vào danh sách, yêu cầu người dùng nhập các thông tin:
* Mã khách hàng: Mã định danh duy nhất của khách hàng. Dùng để tránh trùng lặp và hỗ trợ tìm kiếm và quản lý. Yêu cầu không trùng với các mã đã có, nếu trùng thì cho người dùng nhập lại
* Tên khách hàng: Nhập tên đầy đủ của khách hàng
* Số điện thoại: Nhập đầy dủ số điện thoại khách hàng
* Email khách hàng: Địa chỉ email để gửi thông báo hoặc hóa đơn.
* CCCD: Số căn cước công dân để xác minh danh tính (cần đảm bảo duy nhất). Có thể lấy CCCD để tìm khách hàng
* Ghi chú: Khách VIP hoặc yêu cầu đặc biệt (Phòng không hút thuốc, view biển...)
* **Hiển thị danh sách khách hàng**: In ra thông tin của khách hàng
* Sửa thông tin khách hàng: Nhập mã khách hàng cần sửa ( xác minh mã người dùng nhập,nếu không có thì hiện thị “Không có khách hàng này trong danh sách”) nếu có thì cho người dùng sửa thông tin khách hàng như Mã khách hàng, tên khách hàng, số điện thoại, Email, CCCD, ghi chú
* **Xóa khách hàng:** Xóa khách hàng ra khỏi danh sách, Nhập mã khách hàng cần xóa (kiểm tra mã khách hàng, nếu không có thì thông báo “không có khách hàng này”) nếu có thì xóa khách hàng đó khỏi danh sách
* **Tìm kiếm khách hàng:** Tìm kiếm bằng mã khách hàng hoặc CCCD ( xác minh xem có mã hoặc CCCD không, nếu không có thì hiện thi “Không có khách hàng này”). Nếu có thì hiện thị thông tin khách hàng ra
* **Đặt phòng:** Chức năng này sẽ cho phép người dùng nhập vào thông tin bao gồm:
* Mã đặt phòng: Mã đặt phòng là mã duy nhất để xác định mỗi khi giao dịch phòng (kiểm tra trùng lặp mã nếu đã có thì thông báo “ đã có mã đặt phòng”, cho người dùng nhập lại), nếu có thì lưu vào
* Nhập CCCD: Nếu khách hàng cũ thì sẽ có mã khách hàng đã được lưu còn không thì quay lại phần thêm khách hàng sau đó tới bước tiếp theo
* Mã số phòng: Liên kết với thông tin khách hàng đã lưu trong hệ thống.
* Ngày đặt phòng: Ngày khách hàng đặt phòng (thời điểm giao dịch).
* Ngày checkin: Ngày khách hàng sẽ đến nhận phòng.
* Ngày checkout: Ngày khách hàng sẽ trả phòng.
* Ghi chú

Khi đã nhập xong toàn bộ thông tin, sẽ chuyển tiếp các thông tin cần thiết sẽ tự động cập nhật trạng thái phòng và thêm khách hàng vào phần quản lý khách hàng (Kiểm tra nếu chưa có khách hàng này thì sẽ thêm khách hàng mới, ngược lại thì cập nhật thông tin khách hàng).

* **Hủy phòng:** Cho phép khách hàng hoặc nhân viên hủy đơn đặt phòng trước khi khách hàng đến.
* **Chỉnh sửa thông tin đặt phòng:** Cập nhật thông tin về phòng, ngày giờ, hoặc yêu cầu đặc biệt sau khi đặt.
* **Thanh toán tiền phòng:** Thực hiện thanh toán qua các hình thức như thẻ tín dụng, chuyển khoản hoặc thanh toán trực tiếp.
* **Kiểm tra:** Kiểm tra trạng thái thanh toán
* **Hiển thị:** Hiện thị thông tin đặt phòng
* **Tạo hóa đơn:** Tạo và in hóa đơn cho khách sau khi họ đã hoàn tất thanh toán.
* **Lịch Sử đặt phòng:** Nơi lưu trữ lại thông tin quan trọng:
* **Thêm lịch sử đặt phòng:** Tự động thêm thông tin mới vào danh sách lịch sử khi có giao dịch đặt phòng xảy ra.
* **Xem lịch sử đặt phòng:** Hiển thị danh sách các lịch sử đặt phòng của một khách hàng cụ thể.
* **Lọc lịch sử:** Cung cấp bộ lọc để tìm kiếm các lịch sử đặt phòng
* **Thống kê:** Thông kê tất cả doanh thu, số lượng phòng...
* Tính doanh thu: Chức năng này tính tổng doanh thu từ các đặt phòng đã hoàn thành trong một khoảng thời gian nhất định.
* Số lượng phòng đang được sử dụng: Chức năng này đếm tổng số phòng đang được khách hàng sử dụng (trạng thái: "Đang sử dụng").
* Số lượng phòng đang trống: Chức năng này đếm tổng số phòng đang trống, sẵn sàng để đặt (trạng thái: "Còn trống").
* Số lượng phòng đã được đặt: Chức năng này đếm tổng số phòng đã được đặt trước nhưng chưa sử dụng (trạng thái: "Đã đặt").

## 2.2. Thiết kế hệ thống

### 2.2.1. Xác định các lớp

### ****Class**** Customer

* **Attributes:**
* customerId (String): Mã khách hàng (duy nhất).
* customerName (String): Tên khách hàng.
* phoneNumber (String): Số điện thoại.
* email (String): Email.
* idCard (String): Chứng minh nhân dân hoặc căn cước công dân.
* **Methods:**
* addCustomer(): Thêm khách hàng mới.
* updateCustomerInfo(): Chỉnh sửa thông tin khách hàng.
* deleteCustomer(): Xóa khách hàng khỏi hệ thống.
* searchCustomer(): Tìm kiếm khách hàng theo cccd, số điện thoại.
* toString(): Hiện thi thông tin khách hàng

**Class** Room

* **Attributes:**
* roomId (String): Mã phòng.
* roomType (String): Loại phòng (Standard, Deluxe...).
* roomPrice (double): Giá phòng.
* roomStatus (String): Trạng thái phòng (Còn trống, Đã đặt, Đang sử dụng).
* description (String): Mô tả về phòng (tiện nghi, vị trí, đặc điểm phòng...).
* **Methods:**
* addRoom(): Thêm phòng mới.
* updateRoom(): Chỉnh sửa thông tin phòng.
* deleteRoom(): Xóa phòng khỏi hệ thống.
* checkRoomStatus(): Kiểm tra trạng thái phòng.
* toString(): Hiện thị thông tin phòng

**Class** Booking

* **Attributes:**
* bookingId (String): Mã đặt phòng (duy nhất).
* customer (Customer): Khách hàng liên quan đến đặt phòng.
* room (Room): Phòng đã đặt.
* checkInDate (LocalDate): Ngày nhận phòng.
* checkOutDate (LocalDate): Ngày trả phòng.
* totalAmount (double): Tổng tiền (số đêm \* giá phòng).
* paymentStatus (String): Trạng thái thanh toán (Chưa thanh toán, Đã thanh toán).
* bookingDate (LocalDate): Ngày đặt phòng.
* **Methods:**
* bookRoom(): Thực hiện đặt phòng.
* cancelBooking(): Hủy đặt phòng.
* makePayment(): Thực hiện thanh toán cho đặt phòng.
* checkPaymentStatus(): Kiểm tra trạng thái thanh toán.
* toString(): Hiện thị thông tin đặt phòng

**Class** Invoice

* **Attributes:**
* invoiceId (String): Mã hóa đơn.
* booking (Booking): Đặt phòng liên quan đến hóa đơn.
* issueDate (LocalDate): Ngày lập hóa đơn.
* totalAmount (double): Tổng tiền.
* paymentStatus (String): Trạng thái thanh toán ("Đã thanh toán", "Chưa thanh toán").
* **Methods:**
* createInvoice(): Tạo hóa đơn sau khi thanh toán.
* printInvoice(): In hóa đơn sau khi thanh toán.
* checkPaymentStatus(): Kiểm tra trạng thái thanh toán của hóa đơn.
* toString(): Hiện thị thông tin hóa đơn

**Class** BookingHistory

* **Attributes:**
* historyId (String): Mã lịch sử.
* customer (Customer): Khách hàng đã thực hiện đặt phòng.
* booking (Booking): Thông tin đặt phòng.
* bookingDate (LocalDate): Ngày đặt phòng.
* status (String): Trạng thái đặt phòng (Hoàn thành, Hủy).
* **Methods:**
* addHistory(): Thêm lịch sử đặt phòng.
* viewHistory(): Xem lịch sử đặt phòng của khách hàng.
* filterHistory(): Lọc lịch sử theo thời gian, trạng thái.

**Class** Statistics

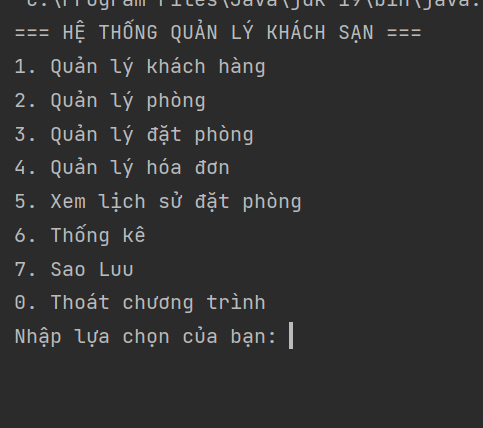
* **Methods:**
* calculateRevenue (double): Tính doanh thu trong một khoảng thời gian.
* countRoomsInUse (int): Số lượng phòng đang được sử dụng.
* countAvailableRooms (int): Số lượng phòng đang trống.
* countBookedRooms (int): Số lượng phòng đã được đặt.

**Thiết kế giao diện**

* Hiện thị menu chính với các lựa chọn:

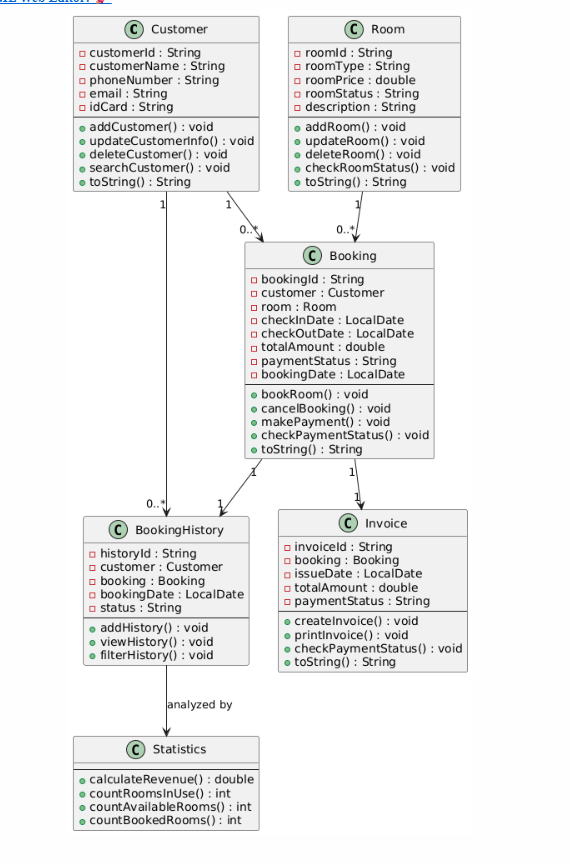
1. Quản lý khách hàng
2. Quản lý phòng
3. Quẩn lý đặt phòng
4. Quản lý hóa đơn
5. Xem lịch sử đặt phòng
6. Thống kê
7. Sao lưu
   1. Thoát chương trình

* Sử dụng hiển thị các thông báo rõ ràng để nhập
* Dưới đây là ảnh minh họa:



### 2.2.2. Thiết kế biểu đồ lớp

Biểu đồ:



Biểu đồ lớp trên thể hiện cấu trúc và mối quan hệ giữa các thành phần chính trong hệ thống quản lý đặt phòng khách sạn.

Hệ thống gồm các lớp chính: **Customer**, **Room**, **Booking**, **BookingHistory**, **Invoice**, và **Statistics**, mỗi lớp đại diện cho một phần chức năng cụ thể.

* **Customer** chịu trách nhiệm quản lý thông tin khách hàng như họ tên, số điện thoại, email, và mã định danh. Lớp này cho phép thêm, cập nhật, tìm kiếm và xóa thông tin khách hàng.
* **Room** quản lý thông tin về phòng như mã phòng, loại phòng, giá phòng, và trạng thái hiện tại. Nó cung cấp các phương thức để thêm, cập nhật và kiểm tra trạng thái phòng.
* **Booking** xử lý việc đặt phòng với thông tin về ngày nhận phòng, trả phòng, số tiền, và trạng thái thanh toán. Lớp này cũng đảm nhận việc đặt phòng, hủy đặt phòng, thanh toán và kiểm tra trạng thái thanh toán.
* **BookingHistory** lưu trữ lịch sử đặt phòng của khách, bao gồm ngày đặt, trạng thái và cung cấp chức năng xem, lọc, và thêm lịch sử.
* **Invoice** đại diện cho hóa đơn, ghi nhận ngày xuất hóa đơn, số tiền thanh toán, và cung cấp phương thức để tạo, in, và kiểm tra hóa đơn.
* **Statistics** phân tích dữ liệu hệ thống, bao gồm doanh thu, số lượng phòng đã được đặt, và danh sách khách hàng thường xuyên. Nó hỗ trợ tạo các báo cáo thống kê như doanh thu, tình trạng đặt phòng, và khách hàng thường xuyên.

Các mối quan hệ giữa các lớp được thiết kế để đảm bảo tính liên kết và phân chia rõ ràng trách nhiệm. Ví dụ, **Customer** có thể tạo **Booking**, **Booking** sinh ra **Invoice**, và thông tin từ **BookingHistory** được sử dụng để phân tích trong **Statistics**.

## 2.3. Triển Khai

2.3.1. Xây dựng các lớp theo mô hình MVC:

* Model:
* Tạo các lớp đại diện cho dữ liệu, ví dụ:
* Customer: Lưu thông tin khách hàng như mã khách hàng, họ tên, số điện thoại, email.
* Room: Lưu thông tin phòng như mã phòng, loại phòng, giá, trạng thái.
* Booking: Quản lý thông tin đặt phòng như mã đặt phòng, khách hàng, phòng, ngày đặt, số ngày thuê.
* Thực hiện các phương thức để đọc dữ liệu từ file .txt và lưu dữ liệu vào file.
* View:
* Hiển thị giao diện người dùng qua console:
* Menu tương tác với các tùy chọn như thêm khách hàng, đặt phòng, xem danh sách phòng, thống kê doanh thu,...
* Hiển thị danh sách dữ liệu như khách hàng, phòng, hoặc thông báo trạng thái khi thao tác thành công/thất bại.
* Controller:
* Xử lý các yêu cầu từ người dùng, ví dụ:
* Thêm khách hàng vào danh sách.
* Kiểm tra trạng thái phòng trước khi đặt.
* Tính toán doanh thu dựa trên thông tin đặt phòng.
* Gọi các phương thức của Model để lưu hoặc đọc dữ liệu khi cần.

### 2.3.2 Xử lý đọc/ghi file:

* Đọc dữ liệu:
* Khi khởi động hệ thống, đọc dữ liệu từ các file .txt để tải thông tin khách hàng, phòng, và đặt phòng vào hệ thống.
* Tách các file theo loại dữ liệu để dễ quản lý, ví dụ:
* customers.txt lưu thông tin khách hàng.
* rooms.txt lưu thông tin phòng.
* bookings.txt lưu thông tin đặt phòng.
* Ghi dữ liệu:
* Khi có thao tác thêm, sửa, hoặc xóa dữ liệu, ghi lại toàn bộ dữ liệu vào file để đảm bảo thông tin được lưu trữ lâu dài.

### 2.3.3 Giao diện console:

* Xây dựng menu chính cho người dùng với các chức năng như:
* Thêm khách hàng.
* Đặt phòng.
* Xem danh sách phòng.
* Thống kê doanh thu.
* Thoát chương trình.
* Hiển thị thông tin dưới dạng bảng để dễ đọc.
* Xử lý các đầu vào sai hoặc ngoại lệ (ví dụ: nhập dữ liệu không hợp lệ, file không tồn tại).

### 2.3.4. Kiểm thử:

* Chức năng kiểm thử:
* Đọc dữ liệu từ file .txt.
* Xử lý các logic nghiệp vụ:
* Đặt phòng (kiểm tra trạng thái phòng, cập nhật trạng thái).
* Tính doanh thu (dựa trên số ngày thuê và giá phòng).
* Ghi dữ liệu vào file.
* Các trường hợp ngoại lệ:
* File dữ liệu không tồn tại hoặc không đọc được.
* Dữ liệu đầu vào không hợp lệ (ví dụ: giá phòng âm, mã khách hàng trùng lặp).

### 2.3.5. Tối ưu hóa:

* Dữ liệu:
* Tách biệt dữ liệu theo từng loại để dễ quản lý.
* Tạo các phương thức chuyển đổi dữ liệu từ chuỗi sang đối tượng và ngược lại.
* Giao diện:
* Thiết kế menu rõ ràng, dễ thao tác.
* Hiển thị thông báo trạng thái cho người dùng (thành công/thất bại).

# CHƯƠNG 3. KẾT QUẢ THỰC HIỆN

## 3.1. Công nghệ đã sử dụng

* Ngôn ngữ lập trình: Java
* Công cụ: IntelliJ IDEA
* Thư viện: (tùy chọn) có thể sử dụng các thư viện hỗ trợ đọc/ghi file, hỗ trợ xử lý văn bản. Lớp SimpleDateFormat trong gói java.text được dùng để định dạng và phân tích ngày tháng từ chuỗi theo một mẫu cụ thể, trong khi ParseException hỗ trợ xử lý các lỗi xảy ra nếu chuỗi ngày tháng không khớp với định dạng mong muốn. Lớp Date trong java.util biểu diễn ngày tháng trong Java, còn ArrayList và List được sử dụng để tạo và quản lý các danh sách động. Cuối cùng, Scanner trong java.util được dùng để nhận dữ liệu đầu vào từ người dùng thông qua console.

## 3.2. Tiến độ thực hiện

### 3.1.2 Chức năng quản lý phòng

* **Xây dựng class Room**

**Mục đích thiết kế:**  
 Class Room được xây dựng nhằm quản lý thông tin cơ bản về phòng trong khách sạn, như mã phòng, loại phòng, giá phòng, và trạng thái. Nó giúp tổ chức dữ liệu một cách logic và hỗ trợ các thao tác như kiểm tra, cập nhật trạng thái phòng và tính toán giá.

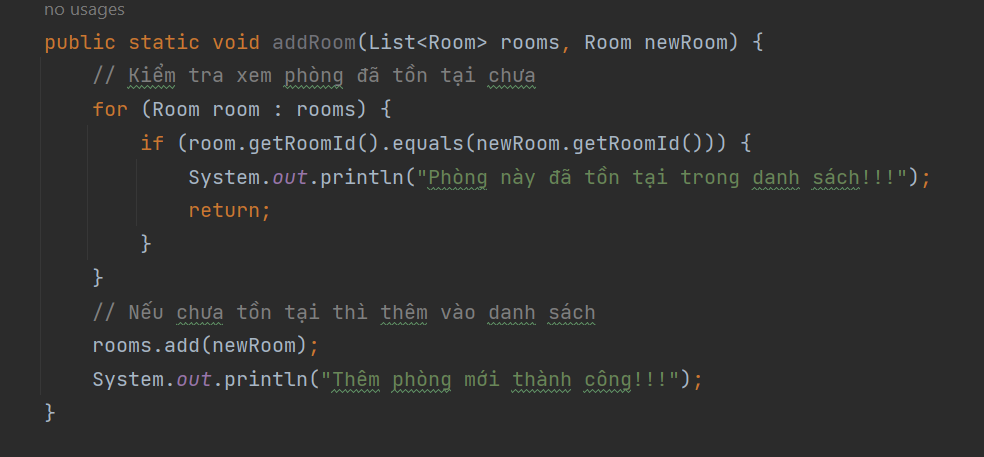
**Chi tiết thiết kế:**

**Thuộc tính (Attributes):**

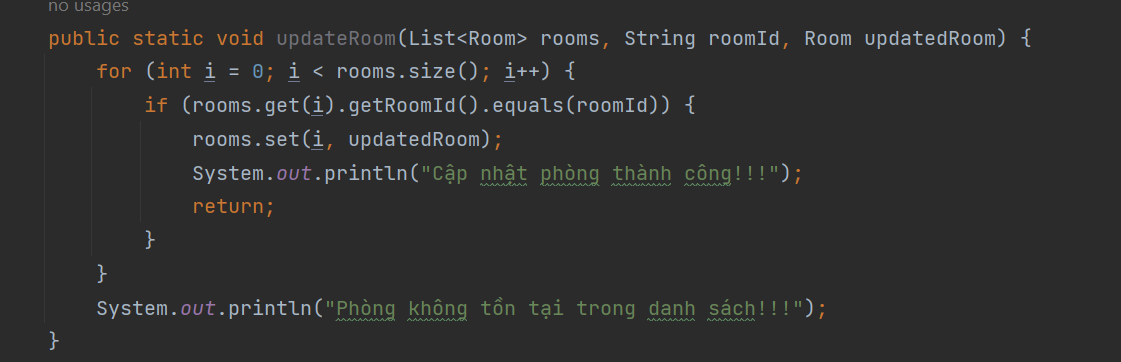
* + roomId (String): Mã định danh duy nhất của phòng.
  + roomType (String): Loại phòng (ví dụ: Standard, Deluxe, Suite).
  + price (double): Giá phòng.
  + status (String): Trạng thái phòng (Available, Booked).

**Phương thức (Methods):**

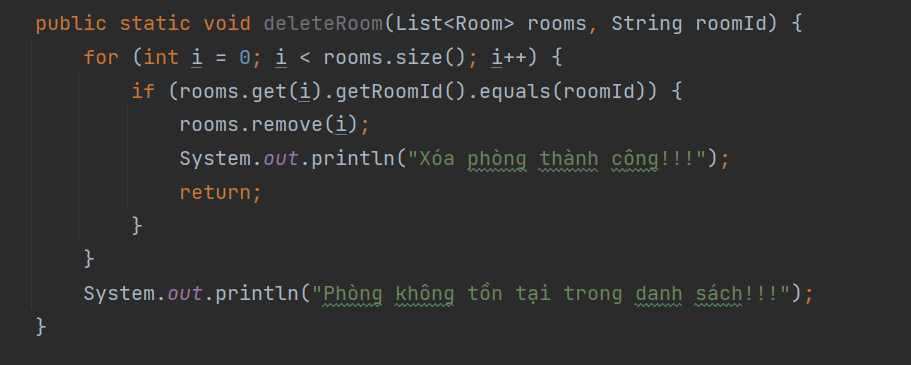
* + **Constructor:** Khởi tạo một đối tượng Room với các thông tin cần thiết.
  + **Getter và Setter:** Hỗ trợ truy cập và cập nhật thông tin thuộc tính.
  + **addRoom:** Chức năng addRoom được thiết kế để thêm phòng mới vào danh sách quản lý, giúp hệ thống mở rộng dữ liệu một cách tự động và thuận tiện. Trong quá trình thực hiện, chức năng đảm bảo kiểm tra mã phòng (roomId) để tránh trường hợp trùng lặp với các phòng đã tồn tại, từ đó duy trì tính chính xác và toàn vẹn dữ liệu. Đồng thời, việc sử dụng chức năng này giúp giảm thiểu các lỗi phát sinh từ thao tác thủ công, hỗ trợ người dùng dễ dàng quản lý và cập nhật thông tin phòng một cách nhất quán và hiệu quả.



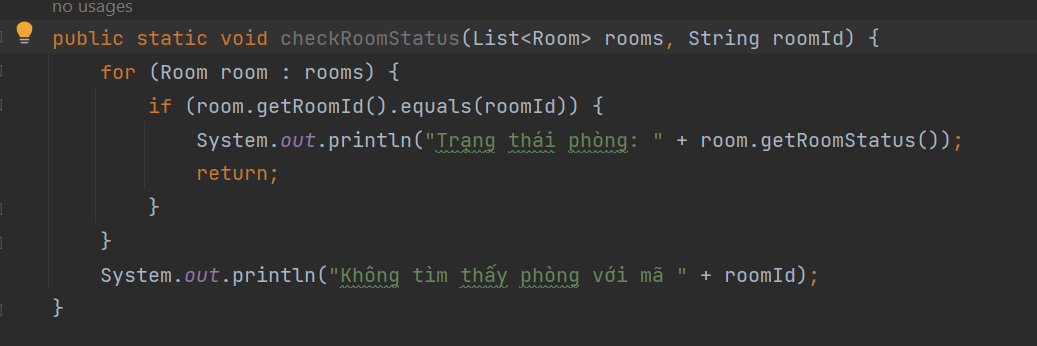
* + **updateRoom:** Chức năng updateRoom được sử dụng để cập nhật thông tin của một phòng trong danh sách quản lý dựa trên mã phòng (roomId). Mục đích chính là đảm bảo hệ thống luôn phản ánh thông tin phòng mới nhất khi có sự thay đổi. Chức năng này giúp người dùng dễ dàng thay thế thông tin cũ của một phòng bằng thông tin mới một cách chính xác và hiệu quả. Bên cạnh đó, việc kiểm tra sự tồn tại của mã phòng trước khi cập nhật cũng giúp ngăn ngừa lỗi truy cập dữ liệu, từ đó duy trì tính toàn vẹn và độ tin cậy của danh sách phòng.



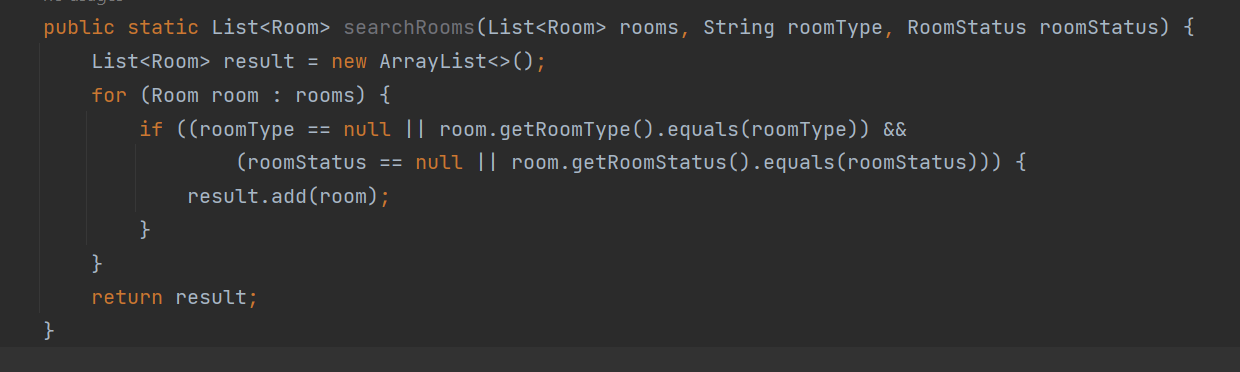
* + **deleteRoom:** Chức năng deleteRoom được sử dụng để xóa một phòng khỏi danh sách quản lý dựa trên mã phòng (roomId). Mục đích chính là loại bỏ các phòng không còn tồn tại hoặc cần xóa bỏ do yêu cầu cập nhật dữ liệu. Chức năng này giúp duy trì danh sách phòng luôn gọn gàng, chính xác và phù hợp với thực tế. Ngoài ra, việc kiểm tra sự tồn tại của mã phòng trước khi thực hiện xóa giúp ngăn ngừa lỗi thao tác trên dữ liệu không hợp lệ, đảm bảo hệ thống hoạt động ổn định và đáng tin cậy.



* + **checkRoomstatus:** Chức năng checkRoomStatus được thiết kế để kiểm tra trạng thái của một phòng trong danh sách dựa trên mã phòng (roomId). Mục đích chính là cung cấp thông tin về tình trạng hiện tại của phòng, như còn trống, đã đặt, hay đang sử dụng, nhằm hỗ trợ quản lý và xử lý yêu cầu từ khách hàng hoặc nhân viên. Chức năng này giúp tăng cường tính linh hoạt và hiệu quả trong việc quản lý phòng, đồng thời đảm bảo thông tin được cung cấp kịp thời và chính xác. Việc kiểm tra mã phòng trước khi lấy trạng thái cũng giúp giảm thiểu lỗi truy cập vào dữ liệu không hợp lệ, duy trì tính ổn định cho hệ thống.



* + **searchRoom:** Chức năng searchRooms được sử dụng để tìm kiếm các phòng trong danh sách dựa trên loại phòng (roomType) và trạng thái phòng (roomStatus). Mục đích chính là hỗ trợ người dùng lọc và truy xuất nhanh chóng những phòng đáp ứng các tiêu chí cụ thể, như tìm tất cả phòng trống hoặc phòng thuộc một loại nhất định. Chức năng này giúp tăng cường hiệu quả trong việc quản lý và xử lý yêu cầu đặt phòng, đồng thời mang lại tính linh hoạt nhờ khả năng tìm kiếm theo tiêu chí tùy chọn (cho phép một hoặc cả hai tiêu chí là null). Việc trả về danh sách kết quả phù hợp cũng giúp hệ thống dễ dàng hiển thị hoặc xử lý thêm các thông tin liên quan.

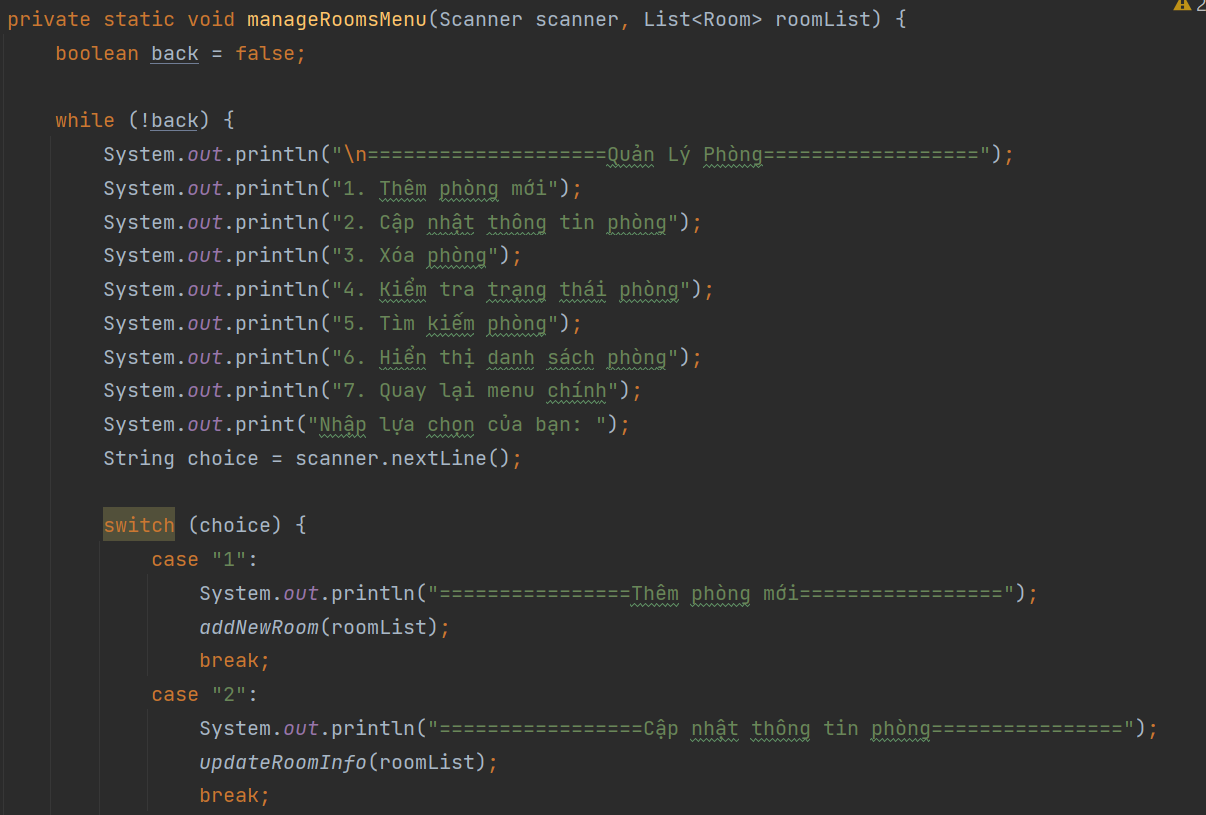


* **Logic code main**

1. **Menu con quản lý phòng**

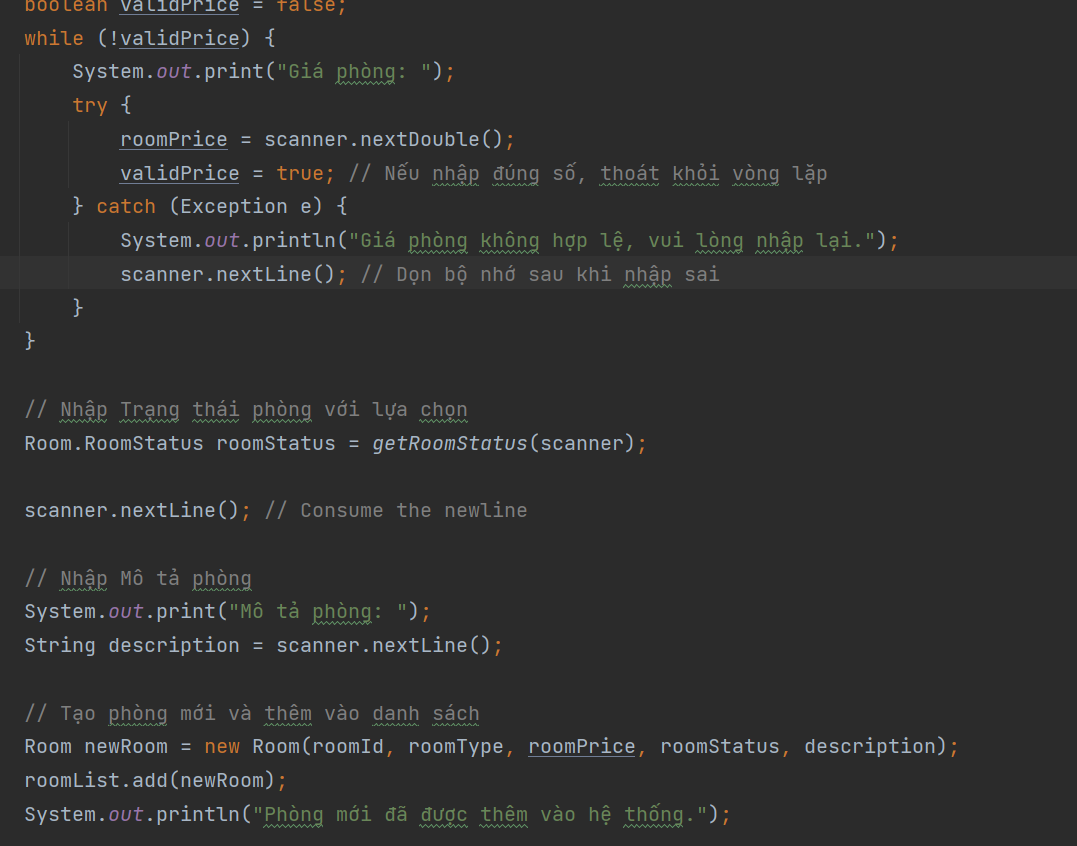
Phương thức manageRoomsMenu được thiết kế để quản lý các hoạt động liên quan đến phòng trong hệ thống, cho phép người dùng thực hiện nhiều thao tác khác nhau trên danh sách phòng. Nó cung cấp một menu tương tác với các tùy chọn như thêm, cập nhật, xóa, kiểm tra trạng thái, tìm kiếm, và hiển thị danh sách phòng. Phương thức này cho phép người dùng thực hiện các thao tác quản lý phòng một cách dễ dàng và linh hoạt thông qua một vòng lặp while cho đến khi người dùng chọn quay lại menu chính.

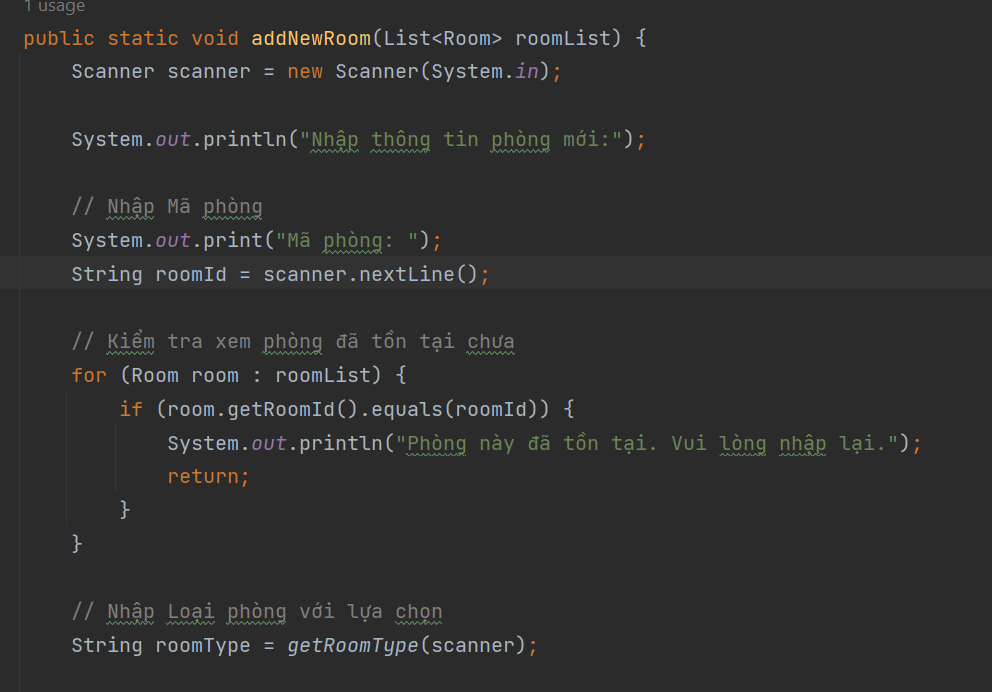
Mục đích của phương thức là cung cấp giao diện quản lý phòng, nơi người quản trị có thể thực hiện các thao tác cần thiết để quản lý thông tin phòng trong hệ thống, giúp duy trì và cập nhật dữ liệu phòng một cách hiệu quả.



1. **Các chức năng trong menu.**

* **Thêm phòng mới:**





Phương thức addNewRoom có mục đích thêm một phòng mới vào hệ thống quản lý phòng. Quá trình thực hiện bao gồm các bước sau:

**- Nhập thông tin phòng mới:** Người dùng được yêu cầu nhập mã phòng, loại phòng, giá phòng, trạng thái phòng và mô tả phòng. Mỗi thông tin này đều có kiểm tra hợp lệ để đảm bảo nhập đúng dữ liệu.

**- Kiểm tra mã phòng trùng lặp:** Trước khi thêm phòng mới, hệ thống sẽ kiểm tra xem mã phòng đã tồn tại trong danh sách phòng hay chưa. Nếu phòng đã tồn tại, người dùng sẽ được thông báo và yêu cầu nhập lại mã phòng.

**- Nhập loại phòng và trạng thái phòng:** Phương thức getRoomType và getRoomStatus được sử dụng để cho phép người dùng chọn loại phòng và trạng thái của phòng từ một danh sách lựa chọn.

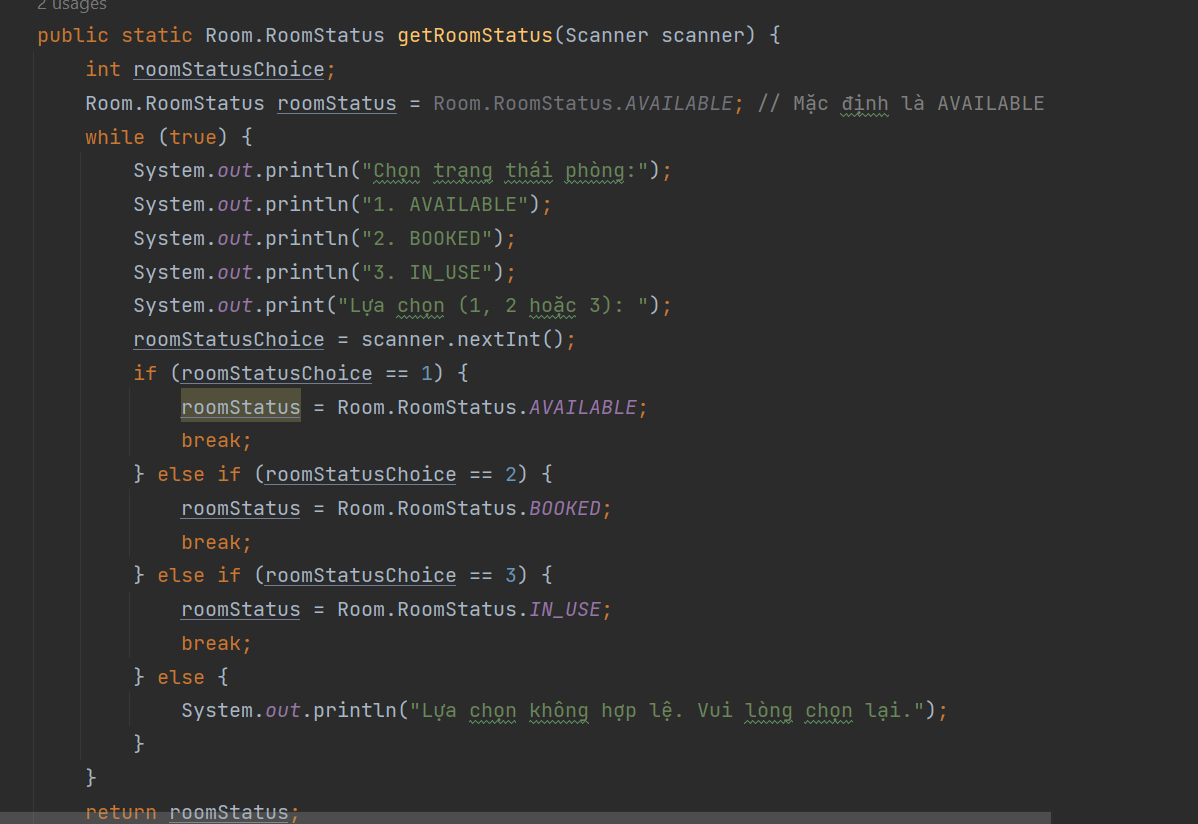
**- Nhập giá phòng:** Hệ thống yêu cầu người dùng nhập giá phòng và sẽ kiểm tra xem giá phòng có hợp lệ hay không (phải là một số dương). Nếu nhập sai, hệ thống sẽ yêu cầu người dùng nhập lại.

**- Nhập mô tả phòng:** Cuối cùng, người dùng được yêu cầu nhập mô tả cho phòng mới.

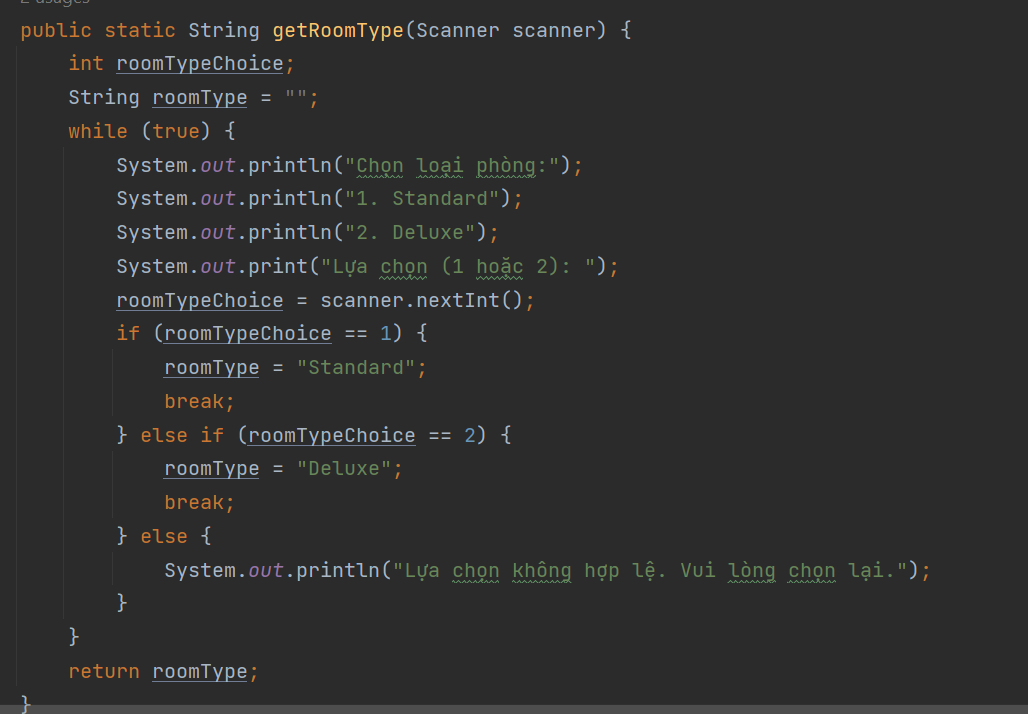
**- Thêm phòng vào hệ thống:** Sau khi tất cả các thông tin đã được nhập đầy đủ và hợp lệ, một đối tượng Room mới sẽ được tạo và thêm vào danh sách phòng roomList.

Mục đích của phương thức này là cung cấp một chức năng cho phép người quản trị thêm phòng mới vào hệ thống, đảm bảo các thông tin được nhập đầy đủ và hợp lệ trước khi phòng được lưu trữ trong danh sách phòng.

**2 phương thức được sử dụng:**



Mục đích của phương thức getRoomStatus là để yêu cầu người dùng chọn trạng thái hiện tại của phòng khi thêm hoặc cập nhật thông tin phòng trong hệ thống quản lý. Phương thức này giúp đảm bảo rằng thông tin trạng thái phòng được nhập vào một cách chính xác và hợp lệ. Cụ thể, phương thức cung cấp ba lựa chọn trạng thái phòng là: AVAILABLE (Phòng còn trống), BOOKED (Phòng đã được đặt), và IN\_USE (Phòng đang được sử dụng). Việc nhập đúng trạng thái giúp hệ thống theo dõi và quản lý các phòng hiệu quả hơn, từ đó đảm bảo sự chính xác trong việc sắp xếp, đặt phòng và phục vụ khách hàng. Thêm vào đó, phương thức này giúp ngăn ngừa các lỗi nhập liệu không hợp lệ bằng cách yêu cầu người dùng nhập lại khi lựa chọn không đúng.



Mục đích của phương thức getRoomType là yêu cầu người dùng chọn loại phòng khi thêm hoặc cập nhật thông tin phòng trong hệ thống. Phương thức này giúp đảm bảo rằng thông tin loại phòng được nhập chính xác, tránh sự nhầm lẫn khi nhập liệu. Cụ thể, phương thức cung cấp hai lựa chọn loại phòng là Standard (phòng tiêu chuẩn) và Deluxe (phòng cao cấp). Người dùng sẽ nhập lựa chọn của mình (1 hoặc 2), và nếu lựa chọn không hợp lệ, hệ thống sẽ yêu cầu nhập lại cho đến khi có lựa chọn đúng. Việc xác định chính xác loại phòng sẽ giúp hệ thống quản lý các phòng và các mức giá tương ứng một cách hiệu quả, đồng thời cải thiện trải nghiệm của người dùng trong việc quản lý và đặt phòng.

* **Chức năng cập nhật thông tin phòng**

Hình ảnh code chức năng chinh sửa phòng



Phương thức updateRoomInfo cập nhật thông tin của phòng trong danh sách roomList. Dưới đây là phân tích ngắn gọn:

* **Nhập mã phòng**: Phương thức yêu cầu người dùng nhập mã phòng cần cập nhật.
* **Tìm phòng theo mã**: Duyệt qua danh sách roomList để tìm phòng có mã khớp. Nếu không tìm thấy, thông báo lỗi và dừng lại.
* **Hiển thị thông tin cũ**: In thông tin hiện tại của phòng để người dùng dễ dàng theo dõi.
* **Cập nhật thông tin**:
  1. Cập nhật loại phòng (gọi phương thức getRoomType).
  2. Cập nhật trạng thái phòng (gọi phương thức getRoomStatus).
  3. Cập nhật giá phòng, xử lý lỗi nhập liệu nếu giá không hợp lệ.
  4. Cập nhật mô tả phòng.
* **Thông báo cập nhật thành công**: In thông tin phòng sau khi cập nhật thành công.

Phương thức này sử dụng các phương thức hỗ trợ đã viết trước đó và kiểm tra dữ liệu nhập vào để đảm bảo tính chính xác.

* **Chức năng xóa phòng**

Hình ảnh code chức năng xóa phòng



Phương thức deleteRoom thực hiện xóa một phòng khỏi danh sách roomList với các bước chính như sau:

1. **Nhập mã phòng**: Người dùng nhập mã phòng cần xóa.
2. **Tìm phòng theo mã**: Duyệt qua danh sách roomList để tìm phòng có mã khớp. Nếu không tìm thấy, thông báo lỗi và dừng lại.
3. **Hiển thị thông tin phòng**: In thông tin phòng cần xóa để người dùng xác nhận.
4. **Xác nhận xóa**: Yêu cầu người dùng xác nhận việc xóa. Nếu người dùng đồng ý (nhấn "y"), phòng sẽ bị xóa khỏi danh sách. Nếu không, việc xóa sẽ bị hủy.
5. **Thông báo kết quả**: Thông báo về việc xóa thành công hoặc hủy bỏ thao tác xóa.

Phương thức này giúp người dùng xóa phòng một cách chắc chắn, với xác nhận trước khi xóa để tránh thao tác sai lầm.

* **Kiểm tra trạng thái phòng**

Mục đích của phương thức checkRoomStatus là cho phép người dùng kiểm tra trạng thái của một phòng cụ thể trong danh sách phòng của hệ thống. Phương thức này thực hiện các bước sau:

1. **Nhập mã phòng**: Người dùng nhập mã phòng mà họ muốn kiểm tra trạng thái.
2. **Tìm phòng trong danh sách**: Hệ thống tìm kiếm phòng theo mã phòng nhập vào trong danh sách các phòng hiện có (roomList).
3. **Kiểm tra phòng**: Nếu phòng không được tìm thấy, hệ thống sẽ thông báo cho người dùng biết rằng không có phòng với mã phòng đó. Nếu tìm thấy phòng, hệ thống sẽ hiển thị thông tin về phòng đó.
4. **Hiển thị trạng thái phòng**: Thông tin bao gồm mã phòng, loại phòng và trạng thái phòng (có thể là AVAILABLE, BOOKED hoặc IN\_USE).

Mục đích cuối cùng của phương thức này là giúp người dùng nhanh chóng kiểm tra trạng thái hiện tại của phòng để có thể quản lý và cập nhật thông tin phòng hiệu quả.



* **Tìm kiếm phòng**

Mục đích của phương thức searchRooms là cung cấp chức năng tìm kiếm các phòng trong hệ thống theo các tiêu chí cụ thể như loại phòng, trạng thái phòng, và giá. Phương thức này thực hiện các bước sau:

**Nhập thông tin tìm kiếm**:

* **Loại phòng**: Người dùng có thể chọn loại phòng (Standard hoặc Deluxe) hoặc bỏ qua nếu muốn tìm tất cả các loại phòng.
* **Trạng thái phòng**: Người dùng có thể chọn trạng thái phòng (AVAILABLE, BOOKED, IN\_USE) hoặc bỏ qua nếu muốn tìm tất cả trạng thái phòng.

**Tìm kiếm phòng**:

* Hệ thống sẽ duyệt qua danh sách phòng (roomList) và kiểm tra xem mỗi phòng có phù hợp với các tiêu chí tìm kiếm (loại phòng và trạng thái) hay không.
* Nếu phòng thỏa mãn tất cả các điều kiện, phòng đó sẽ được thêm vào danh sách kết quả (searchResults).

**Hiển thị kết quả tìm kiếm**:

* Nếu không tìm thấy phòng nào phù hợp, hệ thống sẽ thông báo cho người dùng.
* Nếu có phòng thỏa mãn yêu cầu tìm kiếm, hệ thống sẽ hiển thị danh sách các phòng tìm được, bao gồm thông tin về mã phòng, loại phòng, giá, trạng thái và mô tả.
* **Hiển thị danh sách phòng**

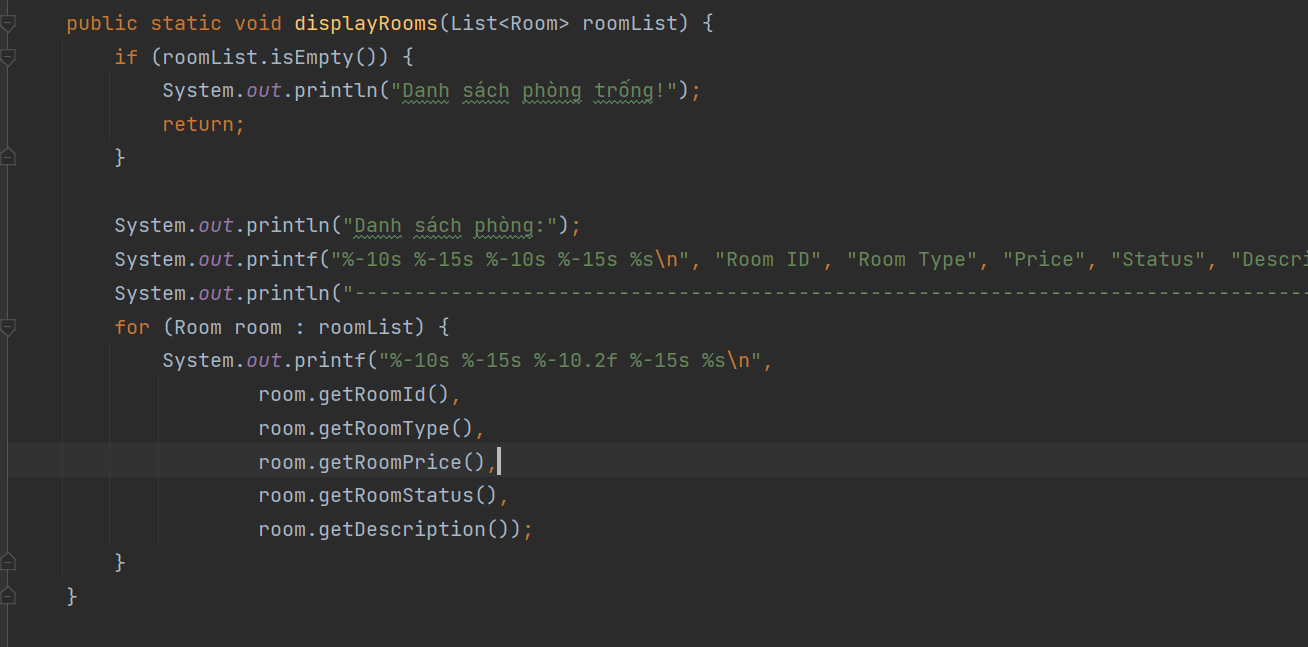
Mục đích của phương thức displayRooms là hiển thị danh sách các phòng trong hệ thống. Cụ thể, phương thức này thực hiện các bước sau:

**Kiểm tra danh sách phòng**:

* Nếu danh sách phòng (roomList) rỗng, hệ thống sẽ thông báo rằng "Danh sách phòng trống!" và không thực hiện bất kỳ thao tác nào khác.

**Hiển thị danh sách phòng**:

* Nếu danh sách phòng không trống, hệ thống sẽ in ra tiêu đề cột cho bảng danh sách phòng, bao gồm các thông tin: "Room ID", "Room Type", "Price", "Status", và "Description".
* Sau đó, phương thức sẽ duyệt qua từng phòng trong roomList và hiển thị các thông tin tương ứng của mỗi phòng như mã phòng, loại phòng, giá phòng, trạng thái phòng và mô tả phòng.



### 2.3.2. Quản lý thống kê

**Phương thức:**

**revenueStatisticsByDateRange**:

- Mục đích của phương thức revenueStatisticsByDateRange là tính toán và hiển thị tổng doanh thu trong một khoảng thời gian cụ thể, dựa trên danh sách hóa đơn (invoiceList). Các bước thực hiện của phương thức này bao gồm:

**Nhập dữ liệu từ người dùng**:

* Phương thức yêu cầu người dùng nhập ngày bắt đầu và ngày kết thúc dưới định dạng dd/MM/yyyy để xác định khoảng thời gian cần thống kê.

**Chuyển đổi định dạng ngày**:

* Ngày bắt đầu và ngày kết thúc được chuyển thành đối tượng LocalDate để dễ dàng so sánh với ngày thanh toán của các hóa đơn.

**Tính toán doanh thu**:

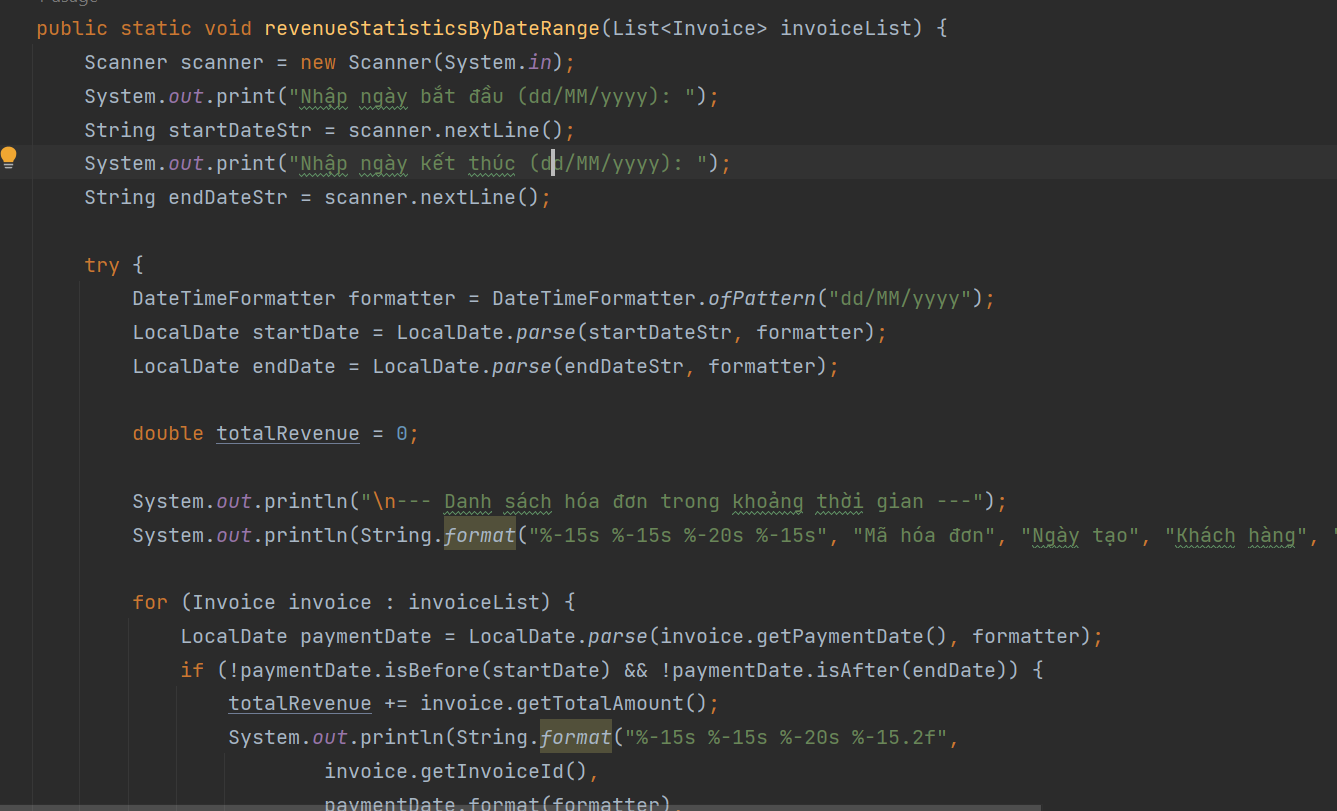
* Phương thức duyệt qua tất cả các hóa đơn trong danh sách và kiểm tra ngày thanh toán của mỗi hóa đơn xem có nằm trong khoảng thời gian đã nhập hay không.
* Nếu ngày thanh toán nằm trong khoảng thời gian, phương thức sẽ cộng tổng số tiền của hóa đơn vào tổng doanh thu.

**Hiển thị thông tin**:

* Sau khi tính toán doanh thu, hệ thống sẽ hiển thị danh sách các hóa đơn trong khoảng thời gian cùng với thông tin chi tiết của mỗi hóa đơn: Mã hóa đơn, ngày thanh toán, tên khách hàng và tổng tiền.
* Cuối cùng, phương thức sẽ hiển thị tổng doanh thu của tất cả các hóa đơn trong khoảng thời gian.

**Xử lý lỗi**:

* Nếu người dùng nhập sai định dạng ngày hoặc có lỗi trong quá trình xử lý, hệ thống sẽ thông báo lỗi và yêu cầu người dùng thử lại.



Phương thức roomBookingStatistics có mục đích thống kê và hiển thị thông tin về các phòng đã được đặt trong hệ thống, giúp người quản lý dễ dàng theo dõi các phòng đã được đặt và tình trạng của chúng. Các bước thực hiện của phương thức này bao gồm:

**Tính tổng số lượng phòng đã đặt**:

* Đầu tiên, phương thức tính toán tổng số lượng phòng đã được đặt bằng cách lấy kích thước của danh sách bookingList, đại diện cho tất cả các giao dịch đặt phòng.

**Hiển thị thông tin về các phòng đã đặt**:

* Phương thức sau đó in ra thông tin chi tiết về mỗi phòng đã đặt, bao gồm:
  + 1. **Mã phòng**: ID của phòng được đặt.
    2. **Tên khách hàng**: Tên của khách hàng đã đặt phòng.
    3. **Ngày nhận phòng**: Ngày khách hàng đến nhận phòng.
    4. **Ngày trả phòng**: Ngày khách hàng trả phòng.
    5. **Trạng thái phòng**: Trạng thái của phòng (còn trống, đã đặt, đang sử dụng).

**Hiển thị tổng số lượng phòng đã đặt**:

* Cuối cùng, tổng số lượng phòng đã đặt được hiển thị, giúp người quản lý biết được tổng số phòng đã được khách hàng đặt.

Phương thức này hữu ích trong việc theo dõi tình trạng đặt phòng, phục vụ cho công tác quản lý và điều hành của khách sạn. Nó giúp người quản lý có cái nhìn tổng quan về các phòng đã được đặt và có thể dự báo được tình hình sử dụng phòng trong thời gian tới.

# KẾT LUẬN

Hệ thống quản lý khách sạn được thiết kế với các lớp rõ ràng như Customer, Room, Booking, Invoice, BookingHistory, và Statistics, giúp quản lý dễ dàng các hoạt động trong khách sạn. Các lớp này không chỉ giúp tổ chức thông tin khách hàng, phòng, đặt phòng và hóa đơn một cách có hệ thống mà còn đảm bảo tính mở rộng và bảo trì dễ dàng. Hệ thống cũng hỗ trợ các chức năng như tìm kiếm, thêm mới, chỉnh sửa và xóa, giúp người dùng quản lý dữ liệu hiệu quả và chính xác. Các phương thức tính toán thống kê doanh thu và trạng thái phòng cũng mang lại cái nhìn tổng quan về hoạt động kinh doanh của khách sạn.

Một trong những nhược điểm của hệ thống là khả năng mở rộng hạn chế nếu như không được thiết kế tốt ngay từ đầu. Ví dụ, khi cần bổ sung tính năng mới hoặc tích hợp các công nghệ khác như thanh toán trực tuyến hoặc quản lý khách sạn thông qua ứng dụng di động, hệ thống có thể gặp phải sự phức tạp trong việc tích hợp với các hệ thống bên ngoài. Hơn nữa, nếu không có sự tối ưu hóa và quản lý tài nguyên hợp lý, hệ thống có thể gặp phải vấn đề về hiệu suất khi xử lý nhiều yêu cầu đồng thời.

Để phát triển hệ thống trong tương lai, cần tập trung vào việc mở rộng các tính năng tích hợp với các công nghệ mới như thanh toán trực tuyến, quản lý phòng qua ứng dụng di động, và tính năng phân tích dựa trên dữ liệu khách hàng để đưa ra các chiến lược kinh doanh hiệu quả hơn. Ngoài ra, có thể tích hợp trí tuệ nhân tạo để dự đoán nhu cầu đặt phòng, tối ưu hóa việc sử dụng phòng, và tự động hóa các tác vụ như nhắc nhở khách hàng về việc thanh toán hoặc nhận phòng. Việc cải thiện giao diện người dùng (UI) và trải nghiệm người dùng (UX) cũng sẽ là yếu tố quan trọng để nâng cao sự hài lòng của khách hàng và nhân viên.