```
import java.time.LocalDate;
import java.time.temporal.ChronoUnit;
import java.util.*;

class Room {
   int roomId;
   String category;
   double pricePerNight;
   boolean isAvailable;

   Room(int roomId, String category, double pricePerNight) {
      this.roomId = roomId;
      this.category = category;
      this.pricePerNight = pricePerNight;
      this.isAvailable = true;
   }
}

class Reservation {
   int reservationId;
   String userName;
   int roomId;
   LocalDate checkInDate;
   LocalDate checkOutDate;
   double totalPrice;
```

```
    Reservation(int reservationId, String userName, int roomId, LocalDate checkInDate,
LocalDate checkOutDate, double totalPrice) {
        this.reservationId = reservationId;
        this.userName = userName;
        this.roomId = roomId;
        this.checkInDate = checkInDate;
        this.checkOutDate = checkOutDate;
        this.totalPrice = totalPrice;
    }
}

public class HotelReservationSystem {
    private List<Room> rooms = new ArrayList<>();
    private List<Reservation> reservations = new ArrayList<>();
    private Scanner scanner = new Scanner(System.in);

    public void addRoom(int roomId, String category, double pricePerNight) {
        rooms.add(new Room(roomId, category, pricePerNight));
    }

    public void searchAvailableRooms() {
        System.out.print("Enter room category to search (or press Enter to view all): ");
        String category = scanner.nextLine().trim();

        List<Room> availableRooms = new ArrayList<>();
        for (Room room : rooms) {
```

```
        if (room.isAvailable && (category.isEmpty() ||
room.category.equalsIgnoreCase(category))) {
            availableRooms.add(room);
        }
    }

    if (availableRooms.isEmpty()) {
        System.out.println("No available rooms found.");
    } else {
        System.out.println("Available rooms:");
        for (Room room : availableRooms) {
            System.out.println("Room ID: " + room.roomId + ", Category: " + room.category + ",
Price per Night: " + room.pricePerNight);
        }
    }
  }

  public void makeReservation() {
    System.out.print("Enter your name: ");
    String userName = scanner.nextLine();
    System.out.print("Enter Room ID to book: ");
    int roomId = scanner.nextInt();
    scanner.nextLine();
    System.out.print("Enter check-in date (YYYY-MM-DD): ");
    LocalDate checkInDate = LocalDate.parse(scanner.nextLine());
    System.out.print("Enter check-out date (YYYY-MM-DD): ");
    LocalDate checkOutDate = LocalDate.parse(scanner.nextLine());
```

```
    Optional<Room> roomOptional = rooms.stream().filter(r -> r.roomId ==
roomId).findFirst();
    if (!roomOptional.isPresent() || !roomOptional.get().isAvailable) {
      System.out.println("Room not available or does not exist.");
      return;
    }

    Room room = roomOptional.get();
    long days = ChronoUnit.DAYS.between(checkInDate, checkOutDate);
    double totalPrice = days * room.pricePerNight;
    Reservation reservation = new Reservation(reservations.size() + 1, userName, roomId,
checkInDate, checkOutDate, totalPrice);
    reservations.add(reservation);
    room.isAvailable = false;
    System.out.println("Reservation successful! Reservation ID: " +
reservation.reservationId);
  }

  public void viewBookingDetails() {
    System.out.print("Enter Reservation ID: ");
    int reservationId = scanner.nextInt();
    scanner.nextLine();

    Optional<Reservation> reservationOptional = reservations.stream().filter(r ->
r.reservationId == reservationId).findFirst();
    if (!reservationOptional.isPresent()) {
```

```
            System.out.println("Reservation not found.");
            return;
        }

        Reservation reservation = reservationOptional.get();
        System.out.println("Reservation Details:");
        System.out.println("User: " + reservation.userName);
        System.out.println("Room ID: " + reservation.roomId);
        System.out.println("Check-in: " + reservation.checkInDate);
        System.out.println("Check-out: " + reservation.checkOutDate);
        System.out.println("Total Price: " + reservation.totalPrice);
    }

    public void processPayment() {
        System.out.print("Enter Reservation ID for payment: ");
        int reservationId = scanner.nextInt();
        scanner.nextLine();

        Optional<Reservation> reservationOptional = reservations.stream().filter(r ->
r.reservationId == reservationId).findFirst();
        if (!reservationOptional.isPresent()) {
            System.out.println("Reservation not found.");
            return;
        }

        Reservation reservation = reservationOptional.get();
```

```
    System.out.print("Enter payment amount: ");
    double paymentAmount = scanner.nextDouble();
    scanner.nextLine();

    if (paymentAmount < reservation.totalPrice) {
        System.out.println("Insufficient payment. Total price is " + reservation.totalPrice);
    } else {
        System.out.println("Payment successful for Reservation ID " + reservationId);
    }
}

public void start() {
    while (true) {
        System.out.println("\nHotel Reservation System");
        System.out.println("1. Search Available Rooms");
        System.out.println("2. Make a Reservation");
        System.out.println("3. View Booking Details");
        System.out.println("4. Process Payment");
        System.out.println("5. Exit");
        System.out.print("Enter your choice: ");
        int choice = scanner.nextInt();
        scanner.nextLine();

        switch (choice) {
            case 1: searchAvailableRooms(); break;
            case 2: makeReservation(); break;
```

```
            case 3: viewBookingDetails(); break;
            case 4: processPayment(); break;
            case 5: System.out.println("Thank you for using the system!"); return;
            default: System.out.println("Invalid choice. Try again.");
          }
        }
    }

    public static void main(String[] args) {
        HotelReservationSystem system = new HotelReservationSystem();
        system.addRoom(101, "Single", 100);
        system.addRoom(102, "Double", 150);
        system.addRoom(103, "Suite", 300);
        system.start();
    }
}
```