```
public class hotelms2 {
    private final List<Hotel> hotels = new ArrayList<>();
    private static final int MAX_HOTELS = 50;

    public boolean addHotel(Hotel hotel) {
        if (hotels.size() < MAX_HOTELS) {
            hotels.add(hotel);
            return true;
        }
        return false;
    }

    public List<Hotel> getHotels() {
        return Collections.unmodifiableList(hotels);
    }

    public static void main(String[] args) {
        hotelms2 hms = new hotelms2();
        Admin admin = new Admin();

        // Create 5 hotels with rooms
        for (int i = 1; i <= 5; i++) {
            Hotel hotel = new Hotel("Hotel " + i, i);
            admin.addRoom(hotel, new Room("10" + i, RoomStyle.STANDARD, RoomStatus.AVAILABLE, 
100.0));
            admin.addRoom(hotel, new Room("20" + i, RoomStyle.DELUXE, RoomStatus.AVAILABLE, 150.0));
            hms.addHotel(hotel);
        }

        // Guest searches for a standard room
        SearchService searchService = new SearchService(hms);
        List<Room> availableRooms = searchService.findAvailableRooms(RoomStyle.STANDARD, new Date(), 
2);

        if (!availableRooms.isEmpty()) {
            Room roomToBook = availableRooms.get(0);
            Guest guest = new Guest("John Doe", "john@example.com");

            // Create booking
            BookingService bookingService = new BookingService();
            RoomBooking booking = bookingService.createBooking(
                guest,
                Collections.singletonList(roomToBook),
                new Date(),
                2
            );

            // Check-in
            Receptionist receptionist = new Receptionist();
            receptionist.checkInGuest(guest, booking);

            // Simulate additional charges
            booking.applyServiceCharge(20.0);
            booking.applyPurchaseCharge(30.0);

            // Check-out
            receptionist.checkOutGuest(guest, booking);

            // Generate bill
            BillGeneration billGenerator = new BillGeneration();
            billGenerator.printBill(booking);
        } else {
            System.out.println("No available rooms found");
        }
    }
}
```

```java
// Updated Hotel class
class Hotel {
    private String name;
    private int id;
    private List<Room> rooms = new ArrayList<>();

    public Hotel(String name, int id) {
        this.name = name;
        this.id = id;
    }

    public void addRoom(Room room) {
        rooms.add(room);
    }

    public List<Room> getRooms() {
        return Collections.unmodifiableList(rooms);
    }
}

class Room {
    private final String roomId;      // Unique across all hotels
    private final String roomNumber;  // Room number within the hotel
    private final int hotelId;        // To identify which hotel this room belongs to
    private RoomStyle roomStyle;
    private RoomStatus roomStatus;
    private double bookingPrice;

    public Room(String roomId, String roomNumber, int hotelId, RoomStyle style, RoomStatus status, 
double price) {
        this.roomId = roomId;
        this.roomNumber = roomNumber;
        this.hotelId = hotelId;
        this.roomStyle = style;
        this.roomStatus = status;
        this.bookingPrice = price;
    }

    public String getRoomId() { return roomId; }
    public String getRoomNumber() { return roomNumber; }
    public int getHotelId() { return hotelId; }
    public RoomStyle getRoomStyle() { return roomStyle; }
    public RoomStatus getRoomStatus() { return roomStatus; }
    public double getBookingPrice() { return bookingPrice; }

    public void setRoomStatus(RoomStatus status) { this.roomStatus = status; }

    @Override
    public String toString() {
        return "Room{" +
                "roomId='" + roomId + '\'' +
                ", roomNumber='" + roomNumber + '\'' +
                ", hotelId=" + hotelId +
                ", roomStyle=" + roomStyle +
                ", roomStatus=" + roomStatus +
                ", bookingPrice=" + bookingPrice +
                '}';
    }
}

// Updated Admin class
class Admin extends Person {
    public void addRoom(Hotel hotel, Room room) {
        hotel.addRoom(room);
    }
}
```

```java
class SearchService {
    private final hotelms2 hms;

    public SearchService(hotelms2 hms) {
        this.hms = hms;
    }

    public List<Room> findAvailableRooms(RoomStyle style, Date date, int duration) {
        List<Room> availableRooms = new ArrayList<>();
        for (Hotel hotel : hms.getHotels()) {
            for (Room room : hotel.getRooms()) {
                if (room.getRoomStatus() == RoomStatus.AVAILABLE && room.getRoomStyle() == style) {
                    availableRooms.add(room);
                }
            }
        }
        return availableRooms;
    }
}



// Updated RoomBooking class
class RoomBooking {
    // ... existing fields ...

    public void applyServiceCharge(double amount) {
        this.totalRoomCharges = new RoomServiceCharge(totalRoomCharges, amount);
    }

    public void applyPurchaseCharge(double amount) {
        this.totalRoomCharges = new InRoomPurchaseCharges(totalRoomCharges, amount);
    }
}

// Updated Decorators (immutable)
class RoomServiceCharge extends RoomChargeDecorator {
    private final double serviceCost;

    public RoomServiceCharge(BaseRoomCharge decoratedCharge, double serviceCost) {
        super(decoratedCharge);
        this.serviceCost = serviceCost;
    }

    @Override
    public Double getCost() {
        return super.decoratedCharge.getCost() + serviceCost;
    }
}

class InRoomPurchaseCharges extends RoomChargeDecorator {
    private final double purchaseCost;

    public InRoomPurchaseCharges(BaseRoomCharge decoratedCharge, double purchaseCost) {
        super(decoratedCharge);
        this.purchaseCost = purchaseCost;
    }

    @Override
    public Double getCost() {
        return super.decoratedCharge.getCost() + purchaseCost;
    }
}

// Updated BillGeneration
class BillGeneration {
    public void printBill(RoomBooking booking) {
        double total = booking.totalRoomCharges.getCost();
        System.out.println("\n========= FINAL BILL =========");
        System.out.println("Booking ID: " + booking.bookingId);
        System.out.println("Guest: " + booking.guestList.get(0).name);
        System.out.println("Room: " + booking.roomInfo.get(0).roomNumber);
        System.out.println("Total Charges: $" + total);
        System.out.println("==============================\n");
    }
}
```