import java.io.\*;

import java.util.\*;

class Room {

int roomNumber;

String category;

boolean isBooked;

public Room(int roomNumber, String category) {

this.roomNumber = roomNumber;

this.category = category;

this.isBooked = false;

}

}

class Reservation {

String customerName;

String category;

int roomNumber;

String reservationId;

public Reservation(String customerName, String category, int roomNumber) {

this.customerName = customerName;

this.category = category;

this.roomNumber = roomNumber;

this.reservationId = UUID.randomUUID().toString();

}

public String toString() {

return reservationId + "," + customerName + "," + category + "," + roomNumber;

}

public static Reservation fromString(String line) {

String[] parts = line.split(",");

Reservation r = new Reservation(parts[1], parts[2], Integer.parseInt(parts[3]));

r.reservationId = parts[0];

return r;

}

}

class Hotel {

ArrayList<Room> rooms = new ArrayList<>();

ArrayList<Reservation> reservations = new ArrayList<>();

final String FILE\_NAME = "bookings.txt";

public Hotel() {

// Initialize rooms

for (int i = 1; i <= 5; i++) rooms.add(new Room(i, "Standard"));

for (int i = 6; i <= 10; i++) rooms.add(new Room(i, "Deluxe"));

for (int i = 11; i <= 15; i++) rooms.add(new Room(i, "Suite"));

loadReservations();

}

public void showAvailableRooms(String category) {

System.out.println("Available " + category + " rooms:");

for (Room r : rooms) {

if (!r.isBooked && r.category.equalsIgnoreCase(category)) {

System.out.println("Room " + r.roomNumber);

}

}

}

public Reservation bookRoom(String name, String category) {

for (Room r : rooms) {

if (!r.isBooked && r.category.equalsIgnoreCase(category)) {

r.isBooked = true;

Reservation res = new Reservation(name, category, r.roomNumber);

reservations.add(res);

saveReservations();

System.out.println("Payment Successful! Booking Confirmed.");

return res;

}

}

System.out.println("No rooms available in " + category + " category.");

return null;

}

public boolean cancelReservation(String reservationId) {

Iterator<Reservation> iterator = reservations.iterator();

while (iterator.hasNext()) {

Reservation res = iterator.next();

if (res.reservationId.equals(reservationId)) {

for (Room r : rooms) {

if (r.roomNumber == res.roomNumber) {

r.isBooked = false;

break;

}

}

iterator.remove();

saveReservations();

System.out.println("Reservation cancelled.");

return true;

}

}

System.out.println("Reservation ID not found.");

return false;

}

public void viewReservations() {

if (reservations.isEmpty()) {

System.out.println("No reservations found.");

return;

}

for (Reservation r : reservations) {

System.out.println("Reservation ID: " + r.reservationId);

System.out.println("Name: " + r.customerName);

System.out.println("Category: " + r.category);

System.out.println("Room Number: " + r.roomNumber);

System.out.println("---------------------------");

}

}

public void saveReservations() {

try (PrintWriter out = new PrintWriter(FILE\_NAME)) {

for (Reservation r : reservations) {

out.println(r.toString());

}

} catch (IOException e) {

System.out.println("Error saving reservations.");

}

}

public void loadReservations() {

try (BufferedReader br = new BufferedReader(new FileReader(FILE\_NAME))) {

String line;

while ((line = br.readLine()) != null) {

Reservation res = Reservation.fromString(line);

reservations.add(res);

for (Room r : rooms) {

if (r.roomNumber == res.roomNumber) {

r.isBooked = true;

break;

}

}

}

} catch (IOException e) {

// file may not exist on first run

}

}

}

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

Hotel hotel = new Hotel();

while (true) {

System.out.println("\n--- Hotel Reservation System ---");

System.out.println("1. View Available Rooms");

System.out.println("2. Book Room");

System.out.println("3. Cancel Reservation");

System.out.println("4. View All Reservations");

System.out.println("5. Exit");

System.out.print("Choose an option: ");

int choice = sc.nextInt();

sc.nextLine(); // consume newline

switch (choice) {

case 1:

System.out.print("Enter room category (Standard/Deluxe/Suite): ");

String cat = sc.nextLine();

hotel.showAvailableRooms(cat);

break;

case 2:

System.out.print("Enter your name: ");

String name = sc.nextLine();

System.out.print("Enter room category (Standard/Deluxe/Suite): ");

String category = sc.nextLine();

Reservation res = hotel.bookRoom(name, category);

if (res != null)

System.out.println("Reservation ID: " + res.reservationId);

break;

case 3:

System.out.print("Enter reservation ID to cancel: ");

String id = sc.nextLine();

hotel.cancelReservation(id);

break;

case 4:

hotel.viewReservations();

break;

case 5:

System.out.println("Exiting...");

sc.close();

return;

default:

System.out.println("Invalid option.");

}

}

}

}