PROJECT REPORT

MUHAMMAD ANSHAD P A
MCA2336

Topic : Hotel Management System

1.Introduction :

The Hotel Management System is designed to automate and streamline various operations involved in managing hotel reservations and room occupancy. The system offers functionalities such as making reservations, canceling reservations, finding rooms for guests, viewing all reservations, and displaying available rooms. Implemented in Java with a MySQL database backend, the system provides a user-friendly graphical interface for seamless interaction.

2. Aim :

The aim of this project is to create a Hotel Management System that facilitates various operations such as making reservations, canceling reservations, finding rooms for guests, viewing all reservations, and displaying available rooms. The system aims to streamline the management of hotel reservations and room occupancy efficiently.

3. Technologies used :

- Java: For developing the application logic and graphical user interface.
- Swing: Java GUI toolkit for creating windows, buttons, text fields, and other UI components.
- MySQL: Relational database management system for storing reservation and room information.
- JDBC: Java Database Connectivity API for integrating Java applications with MySQL database.

4. Functionalities :

✧ Make Reservation: Allows users to make a reservation by providing guest name and the number of guests. Assigns an available room to the reservation and provides a unique reservation ID upon successful booking.

✧ Cancel Reservation: Enables users to cancel a reservation by providing the reservation ID. Frees up the room assigned to the reservation.

✧ Find Room: Allows users to find the room number associated with a guest's name. Searches through existing reservations to locate the room for the specified guest.

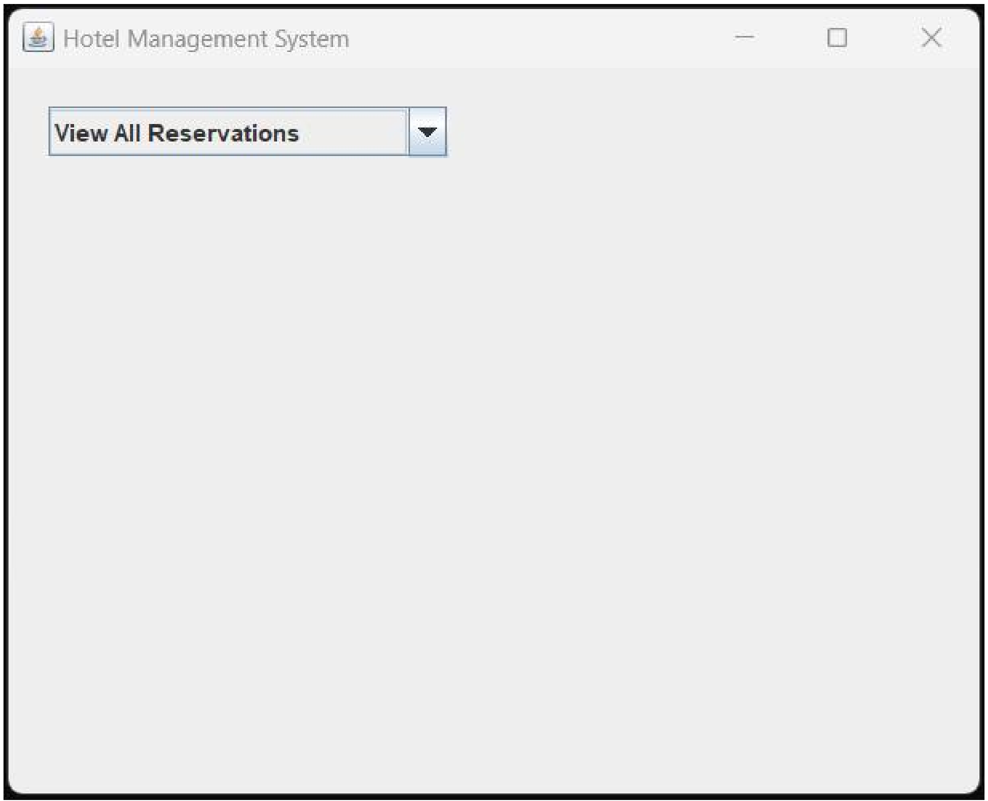
✧ View All Reservations: Displays all reservations made in the system, including reservation ID, guest name, room number, and number of guests.

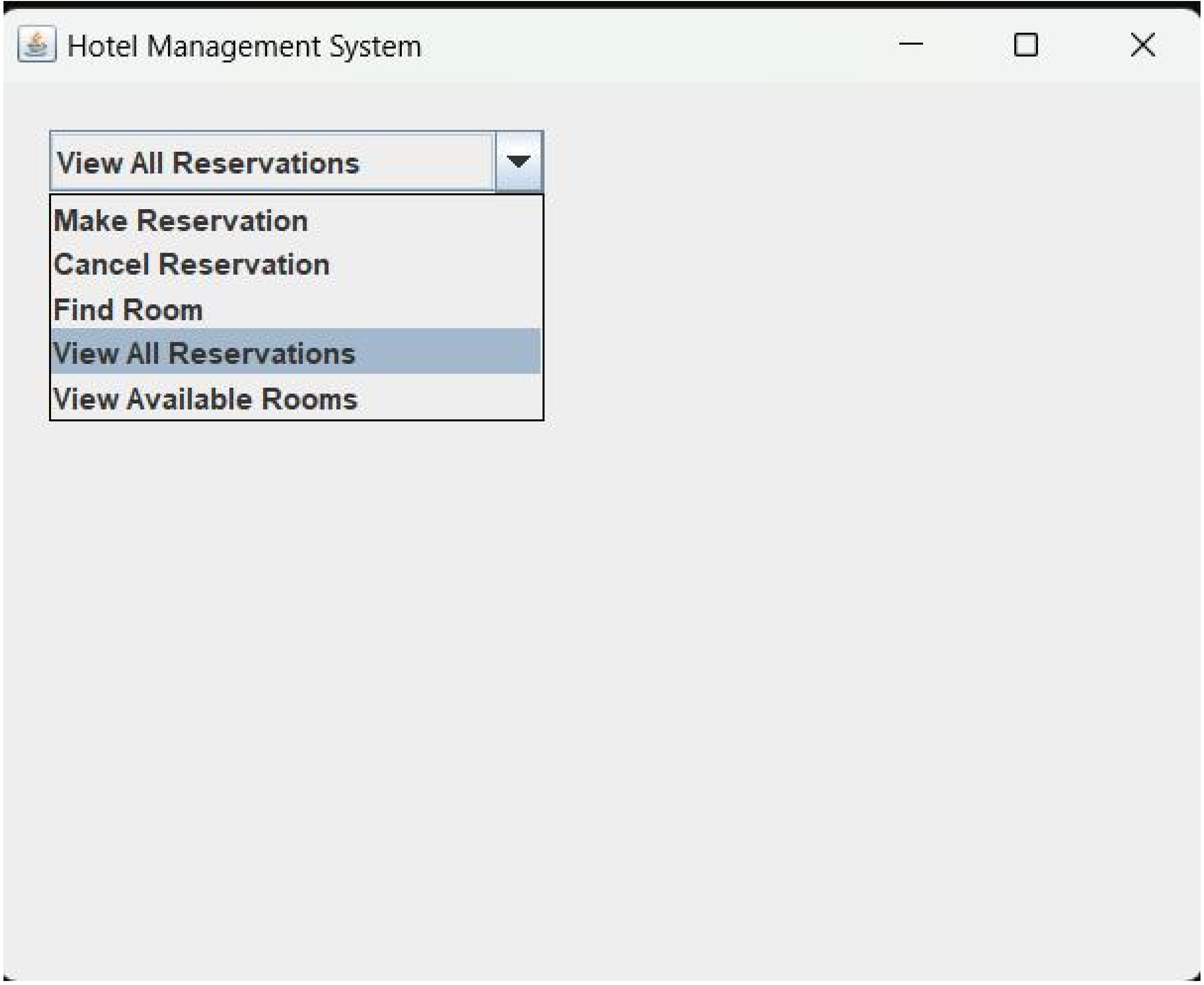
✧ View Available Rooms: Displays all available rooms that are not currently booked, helping users identify vacant rooms for reservation.

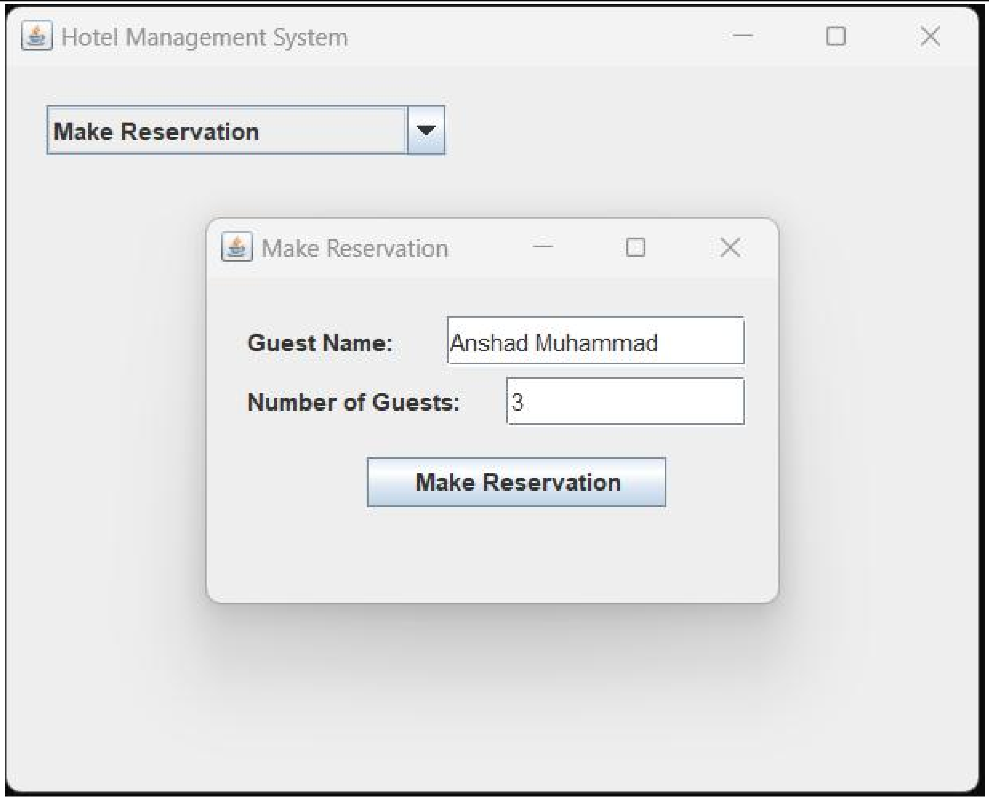
SOURCE CODE :

```
import javax.swing.*;
import java.awt.event.*;
import java.util.*;
import java.sql.*;

public class Project_Hotel_anshad {
    private HotelManagementService hotelService = new
HotelManagementService();
    private JTextField guestNameField, numOfGuestsField, reservationIdField,
guestNameSearchField;
    private JTextArea reservationTextArea,availableRoomsTextArea;

    public Project_Hotel_anshad() {
        // GUI initialization code
        JFrame frame = new JFrame("Hotel Management System");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(500, 400);

        JPanel panel = new JPanel();
        panel.setLayout(null);

        JComboBox<String> mainMenu = new JComboBox<>(new String[]{"Make
Reservation", "Cancel Reservation", "Find Room", "View All
Reservations","View Available Rooms"});
        mainMenu.setBounds(20, 20, 200, 25);
        mainMenu.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String selectedOption = (String) mainMenu.getSelectedItem();
                switch (selectedOption) {
                    case "Make Reservation":
                        createMakeReservationPanel();
                        break;
                    case "Cancel Reservation":
                        createCancelReservationPanel();
                        break;
                    case "Find Room":
                        createFindRoomPanel();
                        break;
                    case "View All Reservations":
                        displayAllReservationsPanel();
                        break;
                    case "View Available Rooms":
                        displayAvailableRoomsPanel();
                        break;
                }
            }
        });
        panel.add(mainMenu);

        frame.add(panel);
        frame.setVisible(true);
    }
    private void createMakeReservationPanel() {
        JFrame makeReservationFrame = new JFrame("Make Reservation");
        makeReservationFrame.setSize(300, 200);

makeReservationFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

        JPanel panel = new JPanel();
        panel.setLayout(null);

        JLabel guestNameLabel = new JLabel("Guest Name:");
        guestNameLabel.setBounds(20, 20, 100, 25);
        panel.add(guestNameLabel);

        guestNameField = new JTextField();
        guestNameField.setBounds(120, 20, 150, 25);
        panel.add(guestNameField);

        JLabel numOfGuestsLabel = new JLabel("Number of Guests:");
        numOfGuestsLabel.setBounds(20, 50, 150, 25);
        panel.add(numOfGuestsLabel);

        numOfGuestsField = new JTextField();
        numOfGuestsField.setBounds(150, 50, 120, 25);
        panel.add(numOfGuestsField);

        JButton makeReservationButton = new JButton("Make Reservation");
        makeReservationButton.setBounds(80, 90, 150, 25);
        makeReservationButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String guestName = guestNameField.getText();
                int numOfGuests =
Integer.parseInt(numOfGuestsField.getText());
                int resId = hotelService.addReservation(guestName,
numOfGuests);
                if (resId == -2 || resId == -1) {
                    JOptionPane.showMessageDialog(null, "Sorry, no available
rooms for reservation.");
                }
                else{
                    JOptionPane.showMessageDialog(null, "Reservation
successful! Reservation ID is : " + resId);
                }
            }
        });
        panel.add(makeReservationButton);

        makeReservationFrame.add(panel);
        makeReservationFrame.setVisible(true);
    }

    private void createCancelReservationPanel() {
        JFrame cancelReservationFrame = new JFrame("Cancel Reservation");
        cancelReservationFrame.setSize(300, 150);

cancelReservationFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

        JPanel panel = new JPanel();
        panel.setLayout(null);

        JLabel reservationIdLabel = new JLabel("Reservation ID:");
        reservationIdLabel.setBounds(20, 20, 100, 25);
        panel.add(reservationIdLabel);

        reservationIdField = new JTextField();
        reservationIdField.setBounds(120, 20, 150, 25);
        panel.add(reservationIdField);

        JButton cancelReservationButton = new JButton("Cancel Reservation");
        cancelReservationButton.setBounds(80, 60, 150, 25);
        cancelReservationButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                int                      reservationId                      =
Integer.parseInt(reservationIdField.getText());
                boolean                      cancelRes                       =
hotelService.cancelReservation(reservationId);
                if (cancelRes == false) {
                    JOptionPane.showMessageDialog(null,  "Reservation  with  ID
" + reservationId + " not found.");
                }
                else{
                    JOptionPane.showMessageDialog(null,  "Reservation  with  ID
" + reservationId + " is Cancelled!");
                }
            }
        });
        panel.add(cancelReservationButton);

        cancelReservationFrame.add(panel);
        cancelReservationFrame.setVisible(true);
    }
    private void createFindRoomPanel() {
        JFrame findRoomFrame = new JFrame("Find Room");
        findRoomFrame.setSize(300, 200);
        findRoomFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

        JPanel panel = new JPanel();
        panel.setLayout(null);

        JLabel guestNameSearchLabel = new JLabel("Guest Name:");
        guestNameSearchLabel.setBounds(20, 20, 100, 25);
        panel.add(guestNameSearchLabel);

        guestNameSearchField = new JTextField();
        guestNameSearchField.setBounds(120, 20, 150, 25);
        panel.add(guestNameSearchField);
        JButton findRoomButton = new JButton("Find Room");
        findRoomButton.setBounds(80, 60, 150, 25);
        findRoomButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String guestName = guestNameSearchField.getText();
                int guestRoom = hotelService.findRoom(guestName);
                if (guestRoom == -2) {
                    JOptionPane.showMessageDialog(null, "Sorry, there is no
Room Reserved for the given Customer");
                }
                else{
                    JOptionPane.showMessageDialog(null, "Guest Found! in Room
number :" + guestRoom);
                }
            }
        });

        panel.add(findRoomButton);


        findRoomFrame.add(panel);
        findRoomFrame.setVisible(true);
    }
    private void displayAllReservationsPanel() {
        JFrame allReservationsFrame = new JFrame("All Reservations");
        allReservationsFrame.setSize(500, 300);

allReservationsFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

        JPanel panel = new JPanel();
        panel.setLayout(null);

        reservationTextArea = new JTextArea();
        reservationTextArea.setEditable(false);
        JScrollPane scrollPane = new JScrollPane(reservationTextArea);
        scrollPane.setBounds(20, 20, 460, 230);
        panel.add(scrollPane);

        int chk =
hotelService.displayAllReservationsInTextArea(reservationTextArea); //
Display reservations in JTextArea
        if (chk == -2) {
            JOptionPane.showMessageDialog(null, "There are No
reservations!");
        }

        allReservationsFrame.add(panel);
        allReservationsFrame.setVisible(true);
    }
    private void displayAvailableRoomsPanel() {
        JFrame availableRoomsFrame = new JFrame("Available Rooms");
        availableRoomsFrame.setSize(300, 200);

availableRoomsFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

        JPanel panel = new JPanel();
        panel.setLayout(null);

        availableRoomsTextArea = new JTextArea();
        availableRoomsTextArea.setEditable(false);
        JScrollPane scrollPane = new JScrollPane(availableRoomsTextArea);
        scrollPane.setBounds(20, 20, 260, 130);
        panel.add(scrollPane);

        int chk =
hotelService.displayAvailableRoomsInTextArea(availableRoomsTextArea); //
Display available rooms in JTextArea
        if (chk == -2) {
            JOptionPane.showMessageDialog(null, "There are No Rooms
Available!");
        }
        availableRoomsFrame.add(panel);
        availableRoomsFrame.setVisible(true);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            new Project_Hotel_anshad();
        });
    }
}

class HotelManagementService {
    final String JDBC_URL =
"jdbc:mysql://localhost:3306/db_hotel?characterEncoding=utf8";
    final String USERNAME = "root";
    final String PASSWORD = "";

    public int addReservation(String guestName, int numberOfGuests) {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con = DriverManager.getConnection(JDBC_URL, USERNAME,
PASSWORD);
            Statement st = con.createStatement();
            ResultSet rs;

            String avl_room = "SELECT * FROM `rooms` WHERE is_booked = 0
LIMIT 1;";
            rs = st.executeQuery(avl_room);
            rs.next();
            int rs_chk = 0;
            rs_chk = rs.getInt("room_number");

            if (rs_chk != 0) {
                String str = "insert into
reservations(room_number,guest_name,number_of_guests) values (";
                str = str + rs_chk + ",'";
                str = str + guestName + "',";
                str = str + numberOfGuests + ")";
                st.executeUpdate(str);

                str = "SELECT * FROM reservations WHERE room_number = " +
rs_chk;
                rs = st.executeQuery(str);
                rs.next();
                int res_id = rs.getInt("reservation_id");

                str = "update rooms set is_booked = 1 where room_number = " +
rs_chk + ";";
                st.executeUpdate(str);
                return res_id;

            } else {
                return -2;
            }
        } catch (Exception e) {
            System.out.println("\nError : " + e);
        }
        return -1;
    }

    public boolean cancelReservation(int reservationId) {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con = DriverManager.getConnection(JDBC_URL, USERNAME,
PASSWORD);
            Statement st = con.createStatement();
            ResultSet rs;
            String str = "select * from reservations";
            rs = st.executeQuery(str);
            boolean found = false;
            while (rs.next()) {

                int rs_chk = rs.getInt("reservation_id");
                int rm_no = rs.getInt("room_number");
                if (rs_chk == reservationId) {
                    str = "DELETE FROM reservations WHERE reservation_id = "
+ rs_chk;
                    st.executeUpdate(str);

                    str = "update rooms set is_booked = 0 where room_number =
" + rm_no + ";";
                    st.executeUpdate(str);
                    found = true;
                    return true;
                }
            }
            if (!found) {
                return false;
            }
        } catch (Exception e) {
            System.out.println("\nError : " + e);
        }
        return false;

    }


    public int displayAllReservationsInTextArea(JTextArea textArea) {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con = DriverManager.getConnection(JDBC_URL, USERNAME,
PASSWORD);

            Statement st = con.createStatement();
            String str = "SELECT * FROM reservations";
            ResultSet rs = st.executeQuery(str);

            StringBuilder sb = new StringBuilder();
            boolean hasReservations = false; // Flag to check if there are
reservations
            sb.append("Displaying All Reservations --------->\n");
            while (rs.next()) {
                hasReservations = true; // Mark that there are reservations
                int reservationId = rs.getInt("reservation_id");
                int roomNumber = rs.getInt("room_number");
                String guestName = rs.getString("guest_name");
                int numberOfGuests = rs.getInt("number_of_guests");
                    sb.append("\nReservation ID: ").append(reservationId)
                        .append("\nGuest Name: ").append(guestName)
                        .append("\nRoom Number: ").append(roomNumber)
                        .append("\nNumber of Guests: 
").append(numberOfGuests)
                        .append("\n-----------------\n");

            }
            if (!hasReservations) {
                return -2;
            }
            textArea.setText(sb.toString());
        } catch (Exception e) {
            System.out.println("\nError : " + e);
        }
        return -1;
    }

    public int displayAvailableRoomsInTextArea(JTextArea textArea) {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con = DriverManager.getConnection(JDBC_URL, USERNAME, 
PASSWORD);

            Statement st = con.createStatement();
            String str = "SELECT * FROM rooms WHERE is_booked = 0";
            ResultSet rs = st.executeQuery(str);

            StringBuilder sb = new StringBuilder();
            boolean hasRooms = false; // Flag to check if there are Rooms
            sb.append("Available Rooms ---------->\n");
            while (rs.next()) {
                hasRooms = true; // Mark that there are Rooms
                int roomNumber = rs.getInt("room_number");
                sb.append("Room Number : ").append(roomNumber).append("\n");
                sb.append("-------------------------------\n");
            }
            if (!hasRooms) {
                return -2;
            }
            textArea.setText(sb.toString());
        } catch (Exception e) {
            System.out.println("\nError : " + e);
        }
        return -1;
    }
    public int findRoom(String gname) {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con = DriverManager.getConnection(JDBC_URL, USERNAME,
PASSWORD);
            Statement st = con.createStatement();
            String str = "SELECT * FROM reservations";
            ResultSet rs = st.executeQuery(str);

            boolean found = false;
            while (rs.next()) {
                int roomNumber = rs.getInt("room_number");
                String guestName = rs.getString("guest_name");
                if (guestName.equals(gname)) {
                    found = true;
                    return roomNumber;
                }
            }

            if (!found) {
                return -2;
            }
        } catch (Exception e) {
            System.out.println("\nError : " + e);
        }
        return -1;
    }
}
```

ScreenShots :

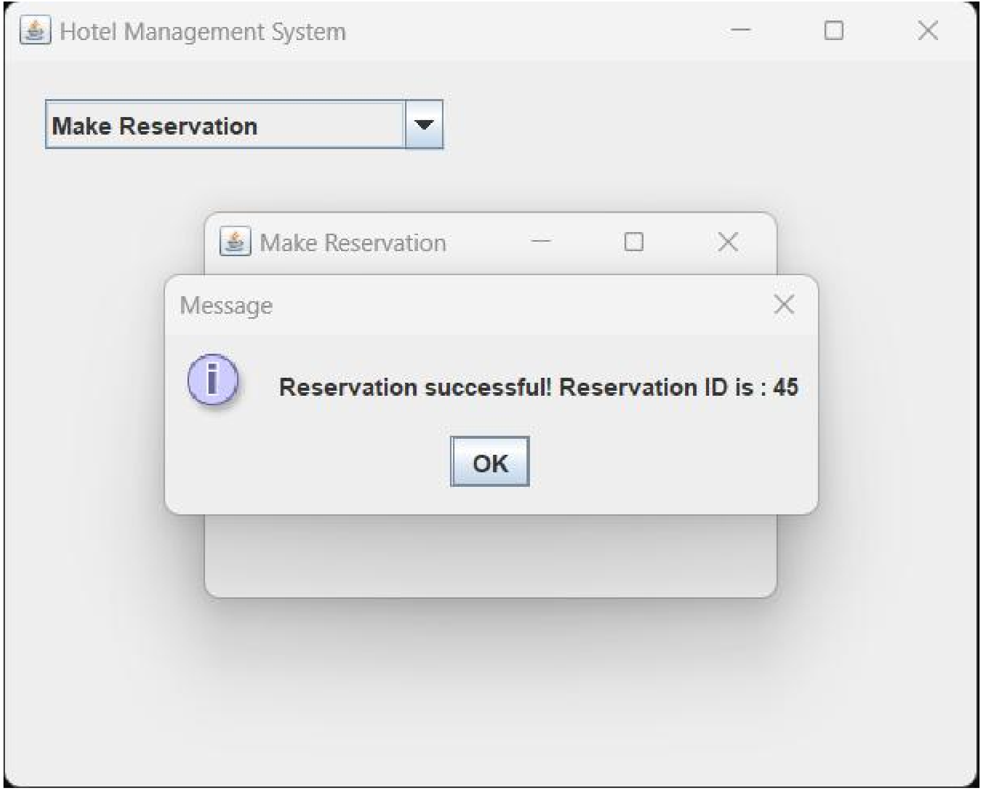




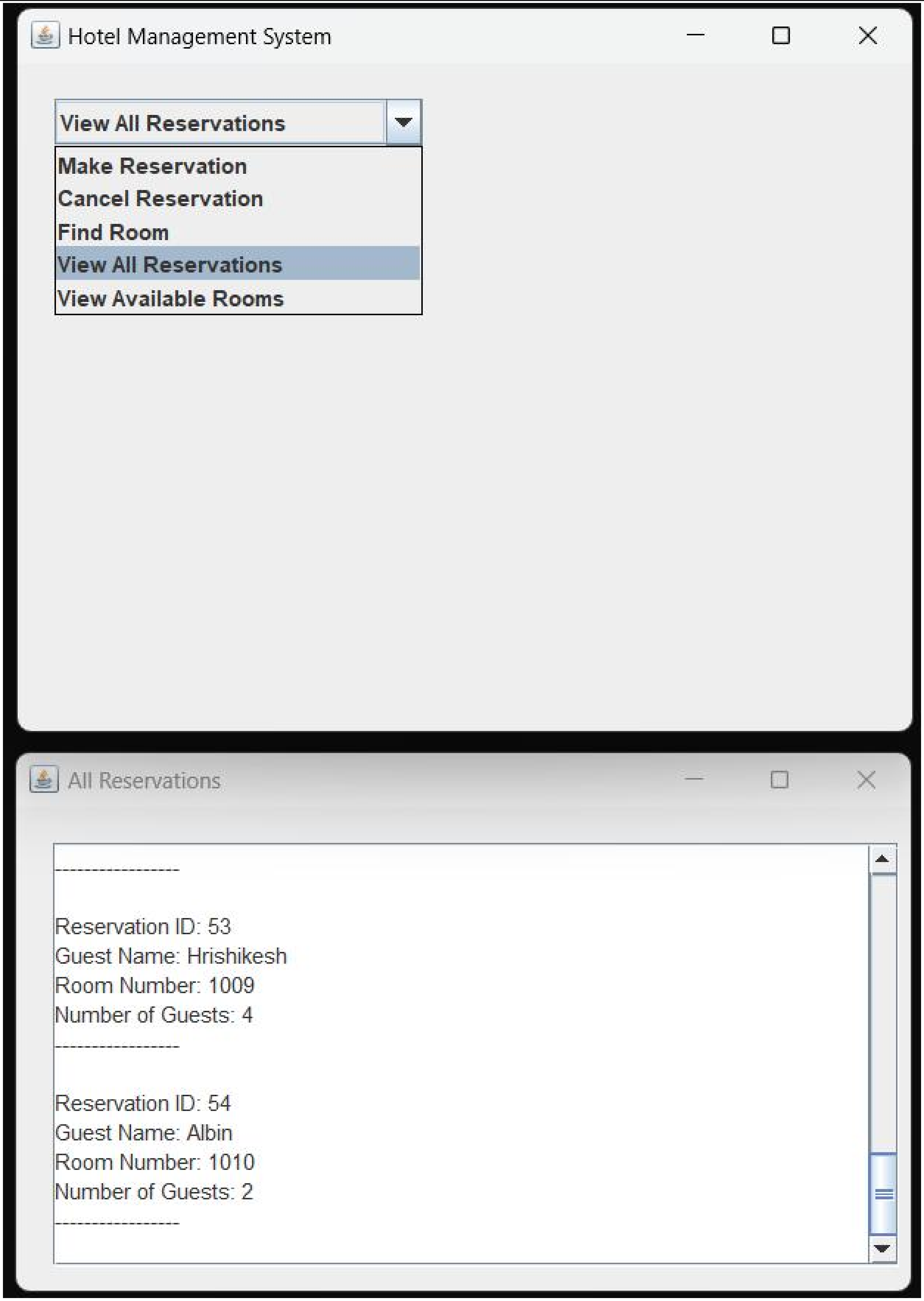


Guest Name: Anshad Muhammad

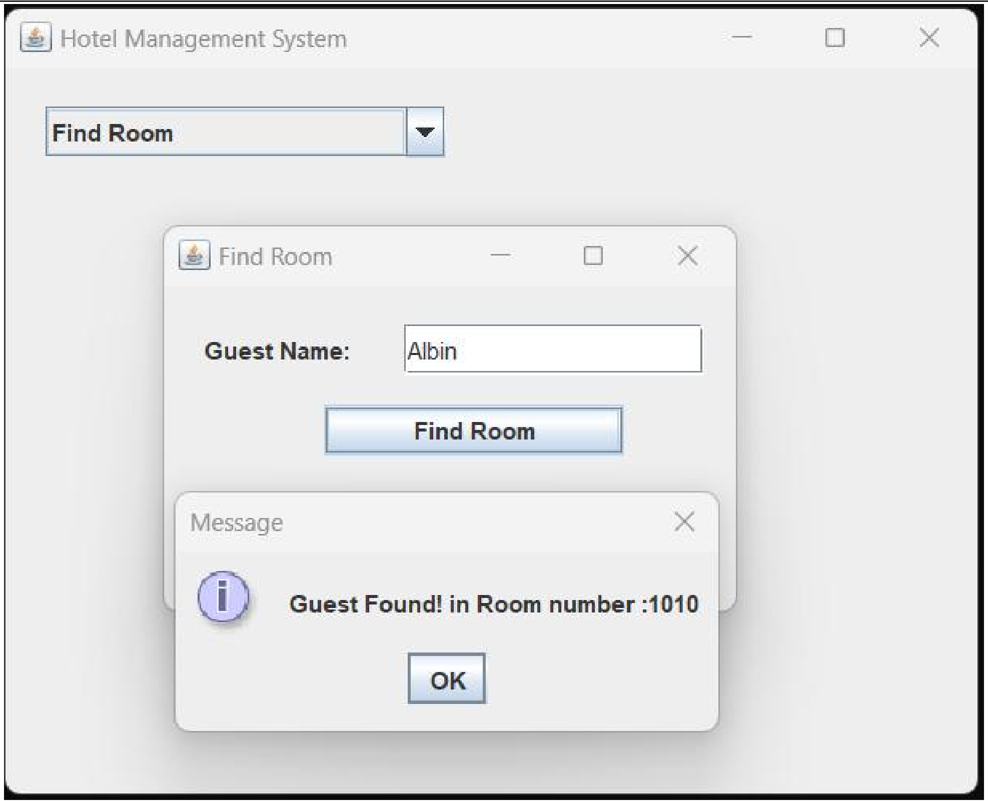
Number of Guests: 3



Reservation successful! Reservation ID is : 45

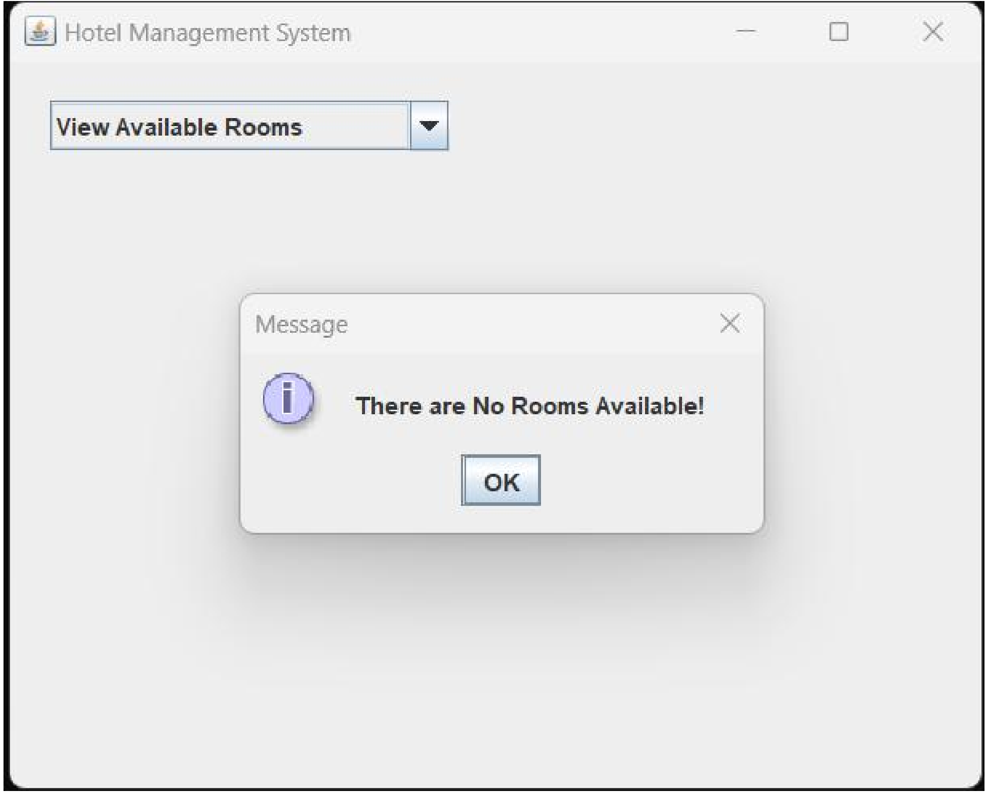


```
-----------------

Reservation ID: 53
Guest Name: Hrishikesh
Room Number: 1009
Number of Guests: 4
-----------------

Reservation ID: 54
Guest Name: Albin
Room Number: 1010
Number of Guests: 2
-----------------
```

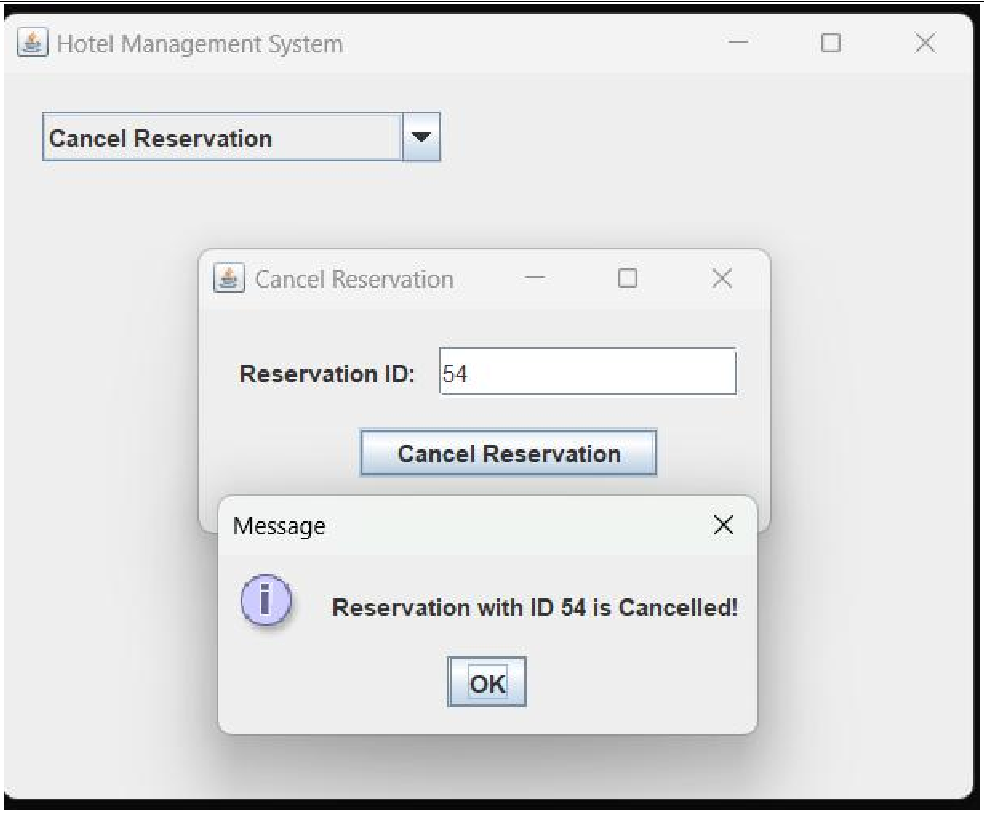


Guest Name: Albin

Guest Found! in Room number :1010



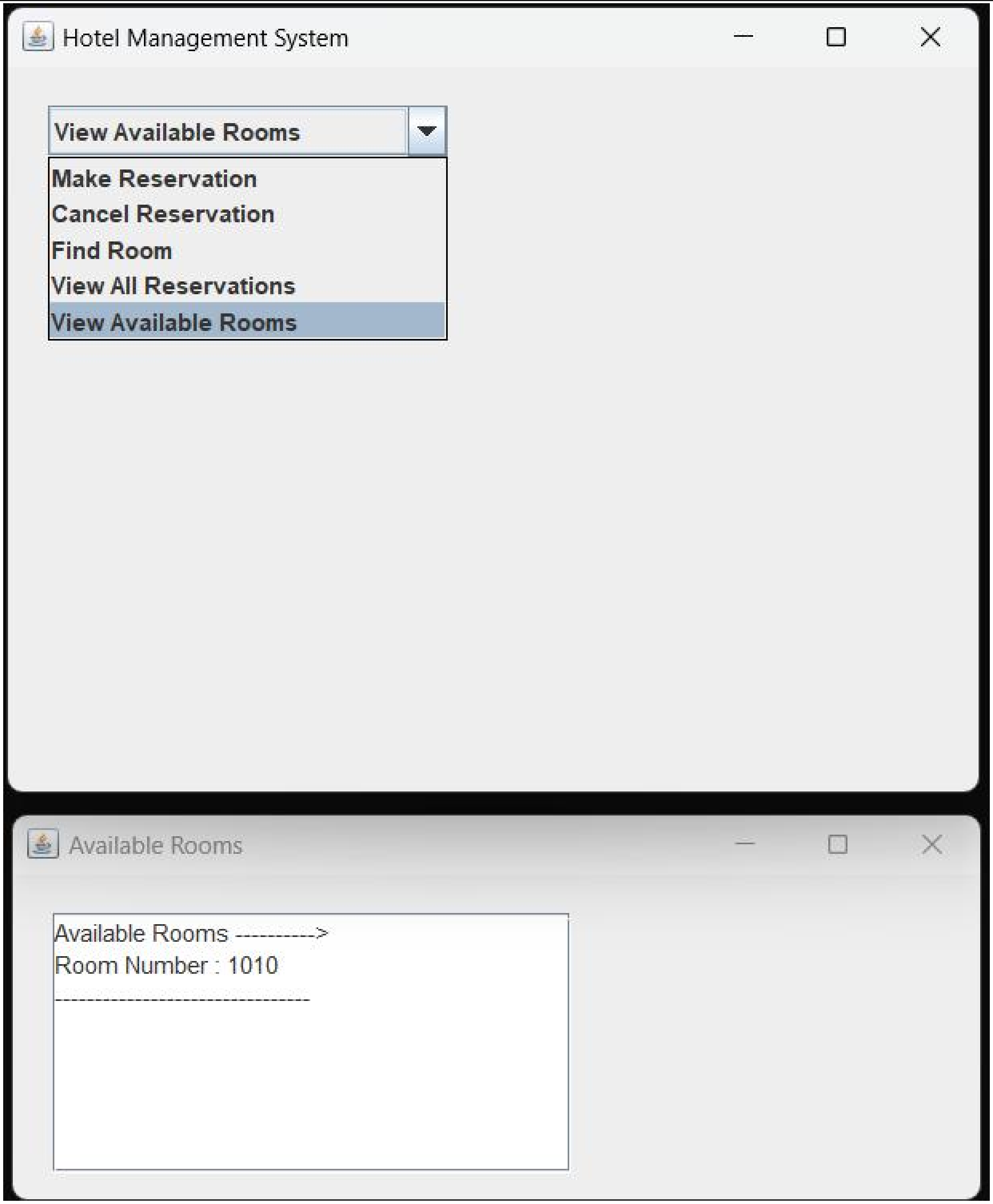
There are No Rooms Available!



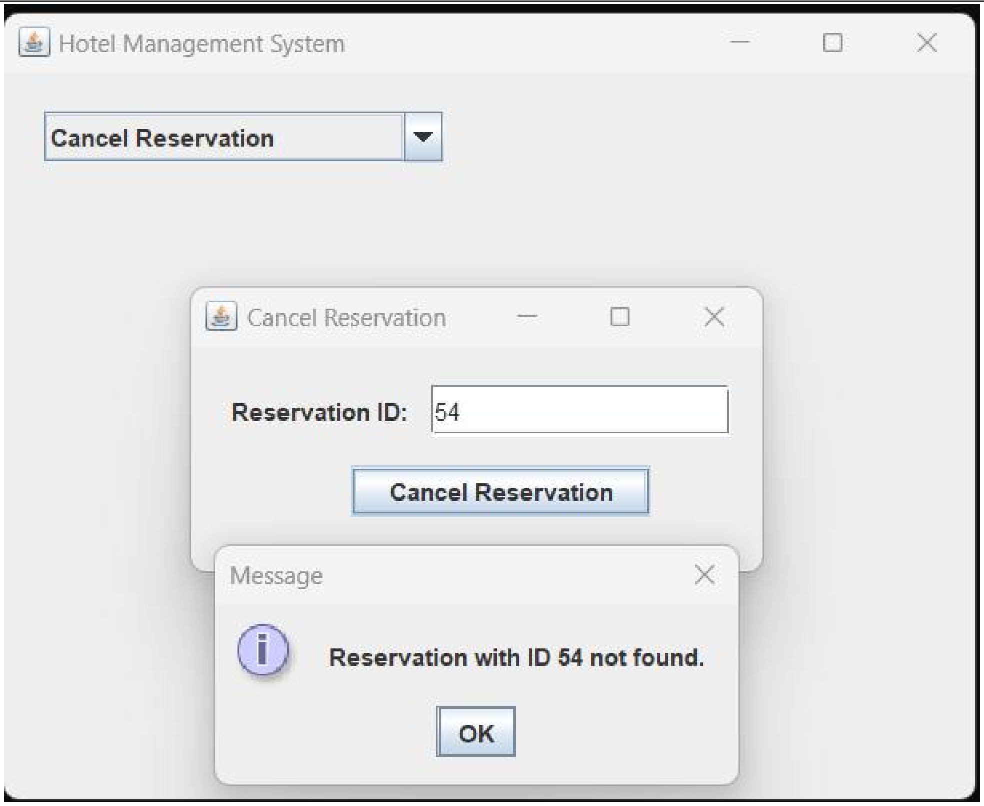
Cancel Reservation

Reservation ID: 54

Reservation with ID 54 is Cancelled!



```
Available Rooms ---------->
Room Number : 1010
---------------------------------
```



Reservation ID: 54

Reservation with ID 54 not found.