HOTEL MANAGEMENT SYSTEM – JAVA PROJECT

Group Project By :-

Name: **APRAJITA (RA2311026030003)**

ANANYA RAJ (RA2311026030030)

COURSE: Bachelor of Technology

CLASS: CSE(AIML)

SECTION: A

SUBJECT: Advanced Programming Practice (21CSC203P)

YEAR: 2023-27

A Java Swing Application with Database Integration

- **Objective:** To develop a hotel management system that allows for customer management and room availability checks.

- **Overview:** This system uses a user-friendly GUI to interact with a database, enabling the addition of customers and checking room availability.

Technologies Used :

- **Programming Language:** Java
- **Libraries:**
  1. **Swing:** For creating the GUI.
  2. **Java SQL (JDBC):** For database connectivity.
- **Database:** MySQL

Database Design-

- **Database Name:** hotel_management

- **Tables:**

- **customers:**
  - Columns: customer_id, room_number, check_in_date
- **rooms:**
  - Columns: room_number, is_occupied (Boolean)

- **Functionality:**
- The customers table stores customer details.
- The rooms table keeps track of room availability.

Key Features

- **User-Friendly Interface:** Intuitive GUI for user interactions.

- **Add Customer Functionality:**

1. Input for customer name and room number.
2. Validates input before inserting into the database.

- **Check Room Availability:**

1. Input for room number.
2. Provides feedback on room availability status.

System Workflow

- **Adding a Customer:**

1. User clicks "Add Customer."
2. Input fields appear for name and room number.
3. User submits data.
4. Data is validated and stored in the database.

- **Checking Room Availability:**

1. User clicks "Check Room Availability."
2. Input field appears for room number.
3. System checks availability and displays the result.

Code Structure

- **Main Class:** HotelManagementGUI
- Sets up the main frame and buttons.

- **Methods:**
- showAddCustomerDialog(): Displays a dialog for adding customers.
- showCheckRoomAvailabilityDialog(): Displays a dialog for checking room availability.

Code Snippet - Adding a Customer

- **Key Code Elements:**
  - JDBC connection setup.
  - Input validation.
  - PreparedStatement for database insertion.

```
private void showAddCustomerDialog() {
    JDialog dialog = new JDialog(frame, title:"Add Customer");
    dialog.setSize(width:300, height:150);
    dialog.setLayout(new FlowLayout());

    JLabel nameLabel = new JLabel(text:"Name:");
    JTextField nameField = new JTextField(columns:20);

    JLabel roomLabel = new JLabel(text:"Room Number:");
    JTextField roomField = new JTextField(columns:5);

    JButton addButton = new JButton(text:"Add");
```

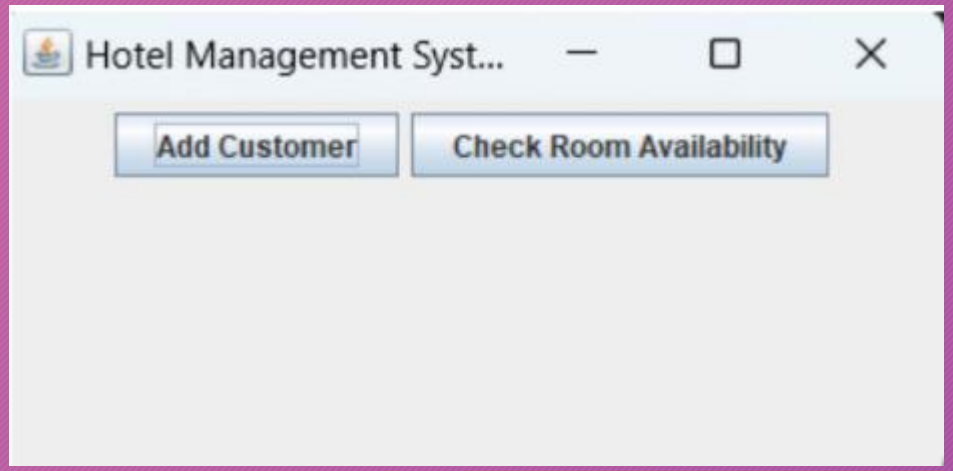
```
    addButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            String name = nameField.getText();
            int roomNumber = Integer.parseInt(roomField.getText());

            // Insert the customer into the database (similar to the previous code example)
            // You'll need to handle the database connection and query here

            // After inserting the customer, you can display a success message
            JOptionPane.showMessageDialog(dialog, message:"Customer added successfully!");

            dialog.dispose();
        }
    });

    dialog.add(nameLabel);
    dialog.add(nameField);
    dialog.add(roomLabel);
    dialog.add(roomField);
    dialog.add(addButton);

    dialog.setVisible(b:true);
}
```

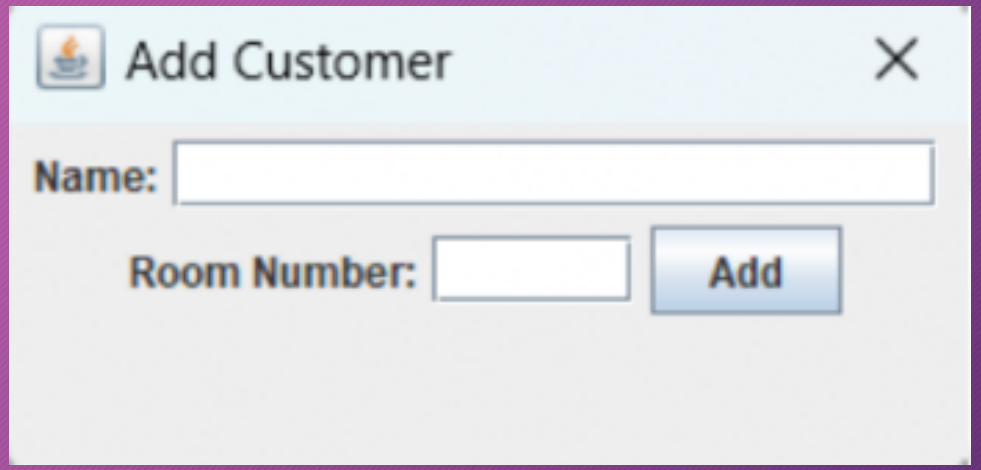
Code Snippet - Checking Room Availability

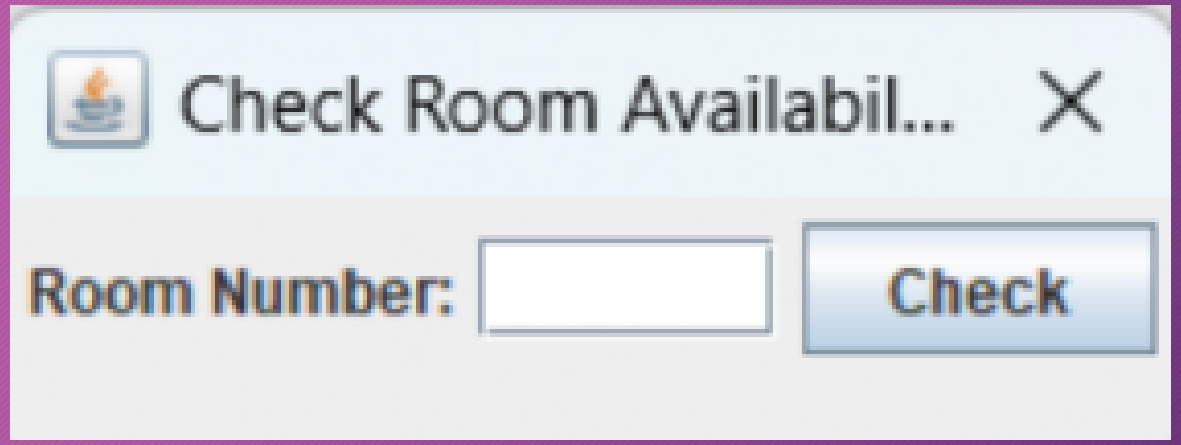
- **Key Code Elements:**
  - Room number input handling.
  - Placeholder for database query logic.

```
private void showCheckRoomAvailabilityDialog() {
    JDialog dialog = new JDialog(frame, title:"Check Room Availability");
    dialog.setSize(width:250, height:100);
    dialog.setLayout(new FlowLayout());

    JLabel roomLabel = new JLabel(text:"Room Number:");
    JTextField roomField = new JTextField(columns:5);

    JButton checkButton = new JButton(text:"Check");

    checkButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            int roomNumber = Integer.parseInt(roomField.getText());

            // Query the database to check room availability (similar to the previous code example)
            // You'll need to handle the database connection and query here

            // Display the result in a dialog
            // You can show a message like "Room is occupied by [customer name]" or "Room is available"

            dialog.dispose();
        }
    });

    dialog.add(roomLabel);
    dialog.add(roomField);
    dialog.add(checkButton);

        dialog.setVisible(b:true);
    }
```

User Interface Screenshots







Database Code Snippets

```sql
-- Create the hotel management database
CREATE DATABASE hotel_management;

-- Use the hotel management database
USE hotel_management;

-- Create a table for storing customer information
CREATE TABLE customers (
    customer_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    room_number INT NOT NULL,
    check_in_date DATE
);

-- Create a table for storing room information (option
CREATE TABLE rooms (
    room_number INT PRIMARY KEY,
    is_occupied BOOLEAN DEFAULT FALSE
);
```

Conclusion

- In conclusion, the Hotel Management System developed using Java Swing and database effectively streamlines customer management and room availability checks. By integrating a user-friendly interface with a relational database, we ensure efficient data handling and improved user experience.

THANK YOU