

```
import java.awt.*;

import java.awt.event.*;

import java.util.*;

public class VisitorManagementApp {

    // A mock system and authentication

    static class VisitorManagementSystem {

        private Map<String, String> visitors = new HashMap<>();

        public void registerVisitor(String name, String contact, String checkInTime) {

            visitors.put(name, checkInTime);

            System.out.println("Visitor Registered: " + name);

        }

        public void checkOutVisitor(String name, String checkOutTime) {

            if (visitors.containsKey(name)) {

                visitors.remove(name);

                System.out.println("Visitor Checked Out: " + name);

            } else {

                System.out.println("Visitor not found!");

            }

        }

        public void displayCurrentVisitors() {

            if (visitors.isEmpty()) {

                System.out.println("No visitors currently.");

            } else {

                System.out.println("Current Visitors: " + visitors);

            }

        }

    }

}
```

```

public void displayVisitorHistory() {
    // Just displaying currently registered visitors as history for simplicity
    displayCurrentVisitors();
}
}

static class VisitorAuthentication {
    public void registerVisitorForSecurity(String name, String contact) {
        // Simple mock, can add actual security check logic
        System.out.println("Visitor " + name + " registered for security check.");
    }
}

// Main GUI Setup
public static void main(String[] args) {
    // System and Authentication objects
    VisitorManagementSystem system = new VisitorManagementSystem();
    VisitorAuthentication auth = new VisitorAuthentication();

    // Creating the frame
    Frame frame = new Frame("Visitor Management System");
    frame.setSize(400, 300);
    frame.setLayout(new FlowLayout());

    // Creating components
    Label label1 = new Label("Enter Visitor Name:");
    TextField nameField = new TextField(20);
    Label label2 = new Label("Enter Visitor Contact:");
    TextField contactField = new TextField(20);
    Label label3 = new Label("Enter Check-in Time:");
    TextField checkInField = new TextField(20);

```

```
Button registerButton = new Button("Register Visitor");
Button checkoutButton = new Button("Check-out Visitor");
Button viewCurrentButton = new Button("View Current Visitors");
Button viewHistoryButton = new Button("View Visitor History");
```

```
// Adding components to the frame
```

```
frame.add(label1);
frame.add(nameField);
frame.add(label2);
frame.add(contactField);
frame.add(label3);
frame.add(checkInField);
frame.add(registerButton);
frame.add(checkoutButton);
frame.add(viewCurrentButton);
frame.add(viewHistoryButton);
```

```
// Button actions
```

```
registerButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String name = nameField.getText();
        String contact = contactField.getText();
        String checkInTime = checkInField.getText();
        system.registerVisitor(name, contact, checkInTime);
        auth.registerVisitorForSecurity(name, contact);
    }
});
```

```
checkoutButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String name = nameField.getText();
```

```

        String checkOutTime = checkInField.getText();
        system.checkOutVisitor(name, checkOutTime);
    }
});

viewCurrentButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        system.displayCurrentVisitors();
    }
});

viewHistoryButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        system.displayVisitorHistory();
    }
});

// Window close operation
frame.addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent we) {
        System.exit(0);
    }
});

// Make the frame visible
frame.setVisible(true);
}
}

```