CRIMINAL RECORDS MANAGEMENT

SOURCE CODE:

```
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

public class CriminalManagementSystemUI extends JFrame {
    private CriminalManagement criminalManagement;
    private JTextField nameField, ageField, crimeField, dateOfCrimeField, searchField;
    private JTable criminalTable;
    private DefaultTableModel tableModel;

    public CriminalManagementSystemUI() {
        criminalManagement = new CriminalManagement();
        setTitle("Criminal Management System");
        setSize(600, 500);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        // Create tabs
        JTabbedPane tabbedPane = new JTabbedPane();

        // Create the Add Criminal panel
        JPanel addCriminalPanel = new JPanel(new GridLayout(5, 2));
        addCriminalPanel.add(new JLabel("Name:"));
        nameField = new JTextField();
        addCriminalPanel.add(nameField);

        addCriminalPanel.add(new JLabel("Age:"));
        ageField = new JTextField();
        addCriminalPanel.add(ageField);

        addCriminalPanel.add(new JLabel("Crime:"));
        crimeField = new JTextField();
        addCriminalPanel.add(crimeField);

        addCriminalPanel.add(new JLabel("Date of Crime:"));
        dateOfCrimeField = new JTextField();
        addCriminalPanel.add(dateOfCrimeField);
```

```
JButton addButton = new JButton("Add Criminal");
addCriminalPanel.add(addButton);

tabbedPane.addTab("Add Criminal", addCriminalPanel);

// Create the Display Criminal panel
String[] columnNames = {"Name", "Age", "Crime", "Date of Crime"};
tableModel = new DefaultTableModel(columnNames, 0);
criminalTable = new JTable(tableModel);
JScrollPane scrollPane = new JScrollPane(criminalTable);
tabbedPane.addTab("Display Criminals", scrollPane);

// Create the Delete Criminal panel
JPanel modifyDeletePanel = new JPanel(new GridLayout(2, 2));
modifyDeletePanel.add(new JLabel("Search Name:"));
searchField = new JTextField();
modifyDeletePanel.add(searchField);

JButton deleteButton = new JButton("Delete Criminal");
modifyDeletePanel.add(deleteButton);

tabbedPane.addTab("Delete Criminal", modifyDeletePanel);

// Add everything to the frame
add(tabbedPane);

// Button actions
addButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            // Get input values
            String name = nameField.getText();
            int age = Integer.parseInt(ageField.getText());
            String crime = crimeField.getText();
            String dateOfCrime = dateOfCrimeField.getText();

            // Create new criminal and add to the system
            Criminal criminal = new Criminal(name, age, crime, dateOfCrime);
            criminalManagement.addCriminal(criminal);

            // Clear form fields
            nameField.setText("");
            ageField.setText("");
            crimeField.setText("");
```

```
                    dateOfCrimeField.setText("");

                    // Update the table
                    updateCriminalTable();
                } catch (NumberFormatException ex) {
                    JOptionPane.showMessageDialog(null, "Please enter a valid age.");
                }
            }
        });

        deleteButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                // Get the name to delete
                String nameToDelete = searchField.getText().trim();
                if (!nameToDelete.isEmpty()) {
                    boolean deleted =
criminalManagement.deleteCriminalByName(nameToDelete);
                    if (deleted) {
                        JOptionPane.showMessageDialog(null, "Criminal deleted successfully.");
                    } else {
                        JOptionPane.showMessageDialog(null, "Criminal not found.");
                    }

                    // Clear the input field
                    searchField.setText("");

                    // Update the table
                    updateCriminalTable();
                } else {
                    JOptionPane.showMessageDialog(null, "Please enter a name to delete.");
                }
            }
        });
    }

    private void updateCriminalTable() {
        // Clear the table
        tableModel.setRowCount(0);
        List<Criminal> criminals = criminalManagement.getAllCriminals();
        for (Criminal criminal : criminals) {
            tableModel.addRow(new Object[]{criminal.getName(), criminal.getAge(),
criminal.getCrime(), criminal.getDateOfCrime()});
        }
    }
    public static void main(String[] args) {
        // Create and show the login screen
        JFrame loginFrame = new JFrame("Login");
        loginFrame.setSize(300, 200);
        loginFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        loginFrame.setLocationRelativeTo(null);

        JPanel loginPanel = new JPanel(new GridLayout(3, 2));
        loginPanel.add(new JLabel("Username:"));
        JTextField usernameField = new JTextField();
        loginPanel.add(usernameField);

        loginPanel.add(new JLabel("Password:"));
        JPasswordField passwordField = new JPasswordField();
        loginPanel.add(passwordField);

        JButton loginButton = new JButton("Login");
        loginPanel.add(loginButton);

        loginFrame.add(loginPanel);
        loginFrame.setVisible(true);

        loginButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String username = usernameField.getText();
                String password = new String(passwordField.getPassword());

                // Simple username/password check
                if (username.equals("admin") && password.equals("password")) {
                    loginFrame.dispose(); // Close login frame
                    SwingUtilities.invokeLater(() -> {
                        CriminalManagementSystemUI ui = new
CriminalManagementSystemUI();
                        ui.setVisible(true);
                    });
                } else {
                    JOptionPane.showMessageDialog(loginFrame, "Invalid credentials. Please
try again.");
                }
            }
        });
    }
}

class Criminal {
    private String name;
    private int age;
    private String crime;
    private String dateOfCrime;

    public Criminal(String name, int age, String crime, String dateOfCrime) {
        this.name = name;
        this.age = age;
        this.crime = crime;
        this.dateOfCrime = dateOfCrime;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }

    public String getCrime() {
        return crime;
    }

    public String getDateOfCrime() {
        return dateOfCrime;
    }
}

class CriminalManagement {
    private List<Criminal> criminals;

    public CriminalManagement() {
        criminals = new ArrayList<>();
    }

    public void addCriminal(Criminal criminal) {
        criminals.add(criminal);
    }

    public List<Criminal> getAllCriminals() {
        return criminals;
    }

    public boolean deleteCriminalByName(String name) {
        Iterator<Criminal> iterator = criminals.iterator(); // Use an iterator
        while (iterator.hasNext()) {
            Criminal criminal = iterator.next();
            if (criminal.getName().equalsIgnoreCase(name)) { // Case-insensitive search
                iterator.remove(); // Safe removal
                return true; // Criminal deleted
            }
        }
        return false; // Criminal not found
    }
}
```