LoginPage

```
package com.mycompany.project;


import javax.swing.*;

import java.awt.*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.*;


public class LoginPage extends JFrame {


    private JLabel userLabel, passwordLabel;

    private JTextField userTextField;

    private JPasswordField passwordField;

    private JButton loginButton, cancelButton;


    public LoginPage(String title) {


        super(title);

        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        this.setSize(300, 200);

        this.setLayout(null);


        userLabel = new JLabel("Login:");

        userLabel.setBounds(10, 20, 80, 25);


        userTextField = new JTextField(20);

        userTextField.setBounds(100, 20, 165, 25);
    passwordLabel = new JLabel("Password:");
    passwordLabel.setBounds(10, 50, 80, 25);

    passwordField = new JPasswordField(20);
    passwordField.setBounds(100, 50, 165, 25);

    loginButton = new JButton("Enter");
    loginButton.setBounds(40, 80, 80, 25);

    cancelButton = new JButton("Cancel");
    cancelButton.setBounds(125, 80, 80, 25);

    this.add(userLabel);
    this.add(userTextField);
    this.add(passwordLabel);
    this.add(passwordField);
    this.add(loginButton);
    this.add(cancelButton);

    cancelButton.addActionListener(new CancelButtonListener());
    loginButton.addActionListener(new LoginButtonListener());
    this.setVisible(true);
}

private class CancelButtonListener implements ActionListener {

    @Override
    public void actionPerformed(ActionEvent e) {
            userTextField.setText("");
            passwordField.setText("");
        }
    }

    private class LoginButtonListener implements ActionListener {

    @Override
    public void actionPerformed(ActionEvent e) {

        String user = userTextField.getText();
        String password = new String(passwordField.getPassword());

        if (user.isEmpty() || password.isEmpty())
        {
            JOptionPane.showMessageDialog(null, "Check your login and password", "Input Error",
JOptionPane.ERROR_MESSAGE);
        } else {
            // Authenticate the user and get their role
            String userRole = authenticateAndGetUserRole(user, password);

            if (!userRole.isEmpty()) {
                // Determine if the user is an admin or a regular user
                if ("admin".equals(userRole.toLowerCase())) {
                    openAdminPage();
                } else {
                    openUserPage();
                }
                // Close the login page
                dispose();
            } else {
                JOptionPane.showMessageDialog(null, "Invalid login or password", "Authentication Failed",
JOptionPane.ERROR_MESSAGE);
            }
        }
    }

    private String authenticateAndGetUserRole(String user, String password) {

        String url =
"jdbc:ucanaccess://C://Users//shooc//Documents//NetBeansProjects//Project//src//main//java//com//
mycompany//project//Database1.accdb";
        String query = "SELECT Role FROM loginPage WHERE username = ? AND pass = ?";

        try (Connection connection = DriverManager.getConnection(url)) {
            try (PreparedStatement preparedStatement = connection.prepareStatement(query)) {
                preparedStatement.setString(1, user);
                preparedStatement.setString(2, password);

                try (ResultSet resultSet = preparedStatement.executeQuery()) {
                    if (resultSet.next()) {
                        return resultSet.getString("Role");
                    }
                }
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
        return "";  // Return an empty string if authentication fails or no role is found
    }

    private void openAdminPage() {
        AdminPage adminPage = new AdminPage("Admin Page");
        adminPage.setVisible(true);
    }

    private void openUserPage() {
        Visualproject userPage = new Visualproject("User Page");
        userPage.setVisible(true);
    }
}

    public static void main(String[] args) {
        LoginPage Login = new LoginPage("Login Page");
    }
}
```

Visualproject

```
package com.mycompany.project;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.*;

public class Visualproject extends JFrame implements ActionListener{
```

```
private JFrame mainFrame;
private JButton b1,b2,b3;
private JTextField tf1;
private JLabel l1;
private JTextArea ta1;

public Visualproject (String title){

    super(title);
    this.setLocation(300,300);
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    l1 = new JLabel ("Title");
    tf1 = new JTextField(25);
    JPanel p1 = new JPanel();
    p1.add(l1 );
    p1.add(tf1 );

    ta1 = new JTextArea(20,50);
    JPanel p2 = new JPanel();
    p2.add(ta1 );

    b1 = new JButton("Save");
    b1.addActionListener(this);
    b2 = new JButton("Search");
    b2.addActionListener(this);
```

```
        b3 = new JButton("Clear");
        b3.addActionListener(this);
        JPanel p3 = new JPanel ();
        p3.add(b1 );
        p3.add(b2 );
        p3.add(b3 );

        JPanel p=(JPanel)this.getContentPane();//to show them in the panel
        p.setLayout(new BorderLayout(3,1)); //we should add them in the same order we want to show them
        p.add(p1 , BorderLayout.NORTH);
        p.add(p2 , BorderLayout.CENTER);
        p.add(p3 ,BorderLayout.SOUTH);

        this.pack();
        this.setVisible(true);
        mainFrame = this;
    }

    @Override
    public void actionPerformed(ActionEvent e)
    {
        if (e.getActionCommand().equals("Save")) {
            saveToFile();
        } else if (e.getActionCommand().equals("Search")) {
            searchFrame();
            this.dispose();
        } else if (e.getActionCommand().equals("Clear")) {
            tf1.setText("");
            ta1.setText("");
        }
    }


    public void saveToFile(){
        String filename = tf1.getText();
        String content = ta1.getText();
        try {
            FileWriter writer = new FileWriter(filename + ".txt");
            writer.write(content);
            writer.close();
            JOptionPane.showMessageDialog(this,"File saved successfully!");
        } catch (IOException ex) {
            JOptionPane.showMessageDialog(this, "Error saving file: " + ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
        }
    }

    public void searchFrame() {
        SerachFrame searchFrame = new SerachFrame(this);
        searchFrame.setVisible(true);
        this.setVisible(false);
    }
    public void showMainFrame() {
        this.setVisible(true);
    }

    public static void main(String[] args) {
        Visualproject vb = new Visualproject("file");
    }
}
```

SerachFrame

```
package com.mycompany.project;



import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.*;

public class SerachFrame extends JFrame implements ActionListener {

        private Visualproject mainFrame;
        private JFrame searchFrame = new JFrame ("Search");
        private JLabel l1 , l2;
        private JTextField tf1 , tf2;
        private JButton b1,b2,b3;
        private JTextArea ta1;

        public SerachFrame (Visualproject mainFrame){

        this.mainFrame = mainFrame;
        this.setLocation(300,300);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        l1 = new JLabel("File to Search: ");
        tf1 = new JTextField(15);
        JPanel p1 = new JPanel();
        p1.add(l1);
        p1.add(tf1);

        ta1 = new JTextArea(20,50);
        JPanel p3 = new JPanel();
        p3.add(ta1);

        l2 = new JLabel("Keywords: ");
        tf2 = new JTextField(10);
        b1 = new JButton("Search");
        b1.addActionListener(this);
        b2 = new JButton("Clear");
        b2.addActionListener(this);
        b3 = new JButton("Cancel");
        b3.addActionListener(this);
```

```
    JPanel p2 = new JPanel();
    p2.add(l2);
    p2.add(tf2);
    p2.add(b1);
    p2.add(b2);
    p2.add(b3);


    JPanel p = (JPanel) this.getContentPane();
    p.setLayout(new BorderLayout(3,1)); //we should add them in the same order we want to show them
    p.add(p1 , BorderLayout.NORTH);
    p.add(p2 , BorderLayout.CENTER);
    p.add(p3 ,BorderLayout.SOUTH);

    this.pack();
    this.setVisible(true);
    setLocationRelativeTo(mainFrame);
    }

    public void actionPerformed(ActionEvent e) {
    if (e.getActionCommand().equals("Search")) {
      searchInFile();
    }
    else if(e.getActionCommand().equals("Clear")){
        tf1.setText("");
```

```
            tf2.setText("");
            ta1.setText("");

    }
    else if (e.getActionCommand().equals("Cancel")) {
        mainFrame.showMainFrame();
        this.dispose();
    }
}

void searchInFile(){
    String fileName = tf1.getText();
    String keyword = tf2.getText();

if (fileName != null && !fileName.isEmpty() && keyword != null && !keyword.isEmpty()) {
    try {
        BufferedReader reader = new BufferedReader(new FileReader(fileName + ".txt"));
        StringBuilder result = new StringBuilder();
        String line;
        boolean found = false;

        while ((line = reader.readLine()) != null) {
            if (line.contains(keyword)) {
                result.append(line);
                found = true;
            }
        }
```

```
            reader.close();


            if (found) {
                ta1.setText(result.toString());
            }
            else{
                JOptionPane.showMessageDialog(this, "No matches found for the keyword.");
            }
        } catch (IOException ex) {
            JOptionPane.showMessageDialog(this, "Error searching file: " + ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
        }
    } else {
        JOptionPane.showMessageDialog(this, "Please enter valid file name and keyword.");
    }
    }


}
```

AdminPage

```
package com.mycompany.project;


import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

public class AdminPage extends JFrame implements ActionListener {

    private JFrame mainFrame;
    private JLabel lab;
    private JTextField titlee, File;
    private JTextArea textArea;
    private JButton saveButton, searchButton, clearButton, statisticsButton;

    public AdminPage(String title) {

        super(title);
        this.setLocation(300, 300);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        lab = new JLabel("Title:");
        titlee = new JTextField(10);

        JPanel p1 = new JPanel();
        add(p1, BorderLayout.NORTH);
        p1.add(lab);
        p1.add(titlee);

        textArea = new JTextArea(20, 50);
        JPanel p2 = new JPanel();
        add(p2, BorderLayout.CENTER);
        p2.add(textArea);

        JPanel buttonPanel = new JPanel();
        add(buttonPanel, BorderLayout.SOUTH);

        saveButton = new JButton("Save");
        searchButton = new JButton("Search");
        clearButton = new JButton("Clear");
        statisticsButton = new JButton("Statistics");

        buttonPanel.add(saveButton);
        buttonPanel.add(searchButton);
        buttonPanel.add(clearButton);
        buttonPanel.add(statisticsButton);

        saveButton.addActionListener(this);
        searchButton.addActionListener(this);
        clearButton.addActionListener(this);
        statisticsButton.addActionListener(this);

        this.pack();
        this.setVisible(true);
        mainFrame = this;
    }

    @Override
```

```
public void actionPerformed(ActionEvent e)
{
    if (e.getSource() == saveButton)
    {
        saveToFile();
    }
    else if (e.getSource() == searchButton)
    {
        searchFrame();
        this.dispose();
    }
    else if (e.getSource() == clearButton)
    {
        titlee.setText("");
        textArea.setText("");
    }
    else if (e.getSource() == statisticsButton)
    {
        statisticsFrame();
    }
}

private void statisticsFrame()
{
    StatisticsFrame statFrame = new StatisticsFrame(this);
    statFrame.setVisible(true);
    this.setVisible(false);
    }

    private void saveToFile()
    {
        String title = titlee.getText();
        String content = textArea.getText();

        try {
            FileWriter writer = new FileWriter(title + ".txt");
            writer.write(content);
            writer.close();
            JOptionPane.showMessageDialog(this,"File saved successfully!");
        } catch (IOException ex) {
            JOptionPane.showMessageDialog(this, "Error saving file: " + ex.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
        }
    }

    private void searchFrame()
    {
        AdminSearch searchFrame = new AdminSearch(this);
        searchFrame.setVisible(true);
        this.setVisible(false);

    }

    public void showMainFrame() {
        this.setVisible(true);
    }

    public static void main(String[] args) {
        AdminPage adminPage = new AdminPage("Text Editor");
    }
}
```

StatisticsFrame

```
package com.mycompany.project;



import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class StatisticsFrame extends JFrame implements ActionListener {

    //private Visualproject mainFrame;
    private AdminPage mainf;
    private JFrame StatisticsFrame = new JFrame ("Statistics");
    private JTextField fileTextField;
    private JButton calculateButton;
```

```
public StatisticsFrame(AdminPage mainf) {

    this.mainf = mainf;
    this.setLocation(300,300);
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    fileTextField = new JTextField(20);
    calculateButton = new JButton("Calculate");

    JPanel panel = new JPanel();
    panel.add(new JLabel("File for Statistics:"));
    panel.add(fileTextField);
    panel.add(calculateButton);

    add(panel);

    calculateButton.addActionListener(this);

    this.pack();
    this.setVisible(true);
    setLocationRelativeTo(mainf);
}

@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == calculateButton) {
        calculateStatistics();
    }
}


private void calculateStatistics() {

    String fileName = fileTextField.getText();

    if (fileName.isEmpty())
    {
        JOptionPane.showMessageDialog(this, "Please enter a file name.", "Error", JOptionPane.ERROR_MESSAGE);
        return;
    }

    try (BufferedReader reader = new BufferedReader(new FileReader(fileName)))
    {
        int linesWithNumericValuesGreaterOrEqual10 = 0;
        int linesWithNumericValuesLessThan10 = 0;

        String line;
        while ((line = reader.readLine()) != null)
        {
            if (containsNumericValues(line))
            {
                double numericValue = extractNumericValue(line);
                if (numericValue >= 10)
                {
                    linesWithNumericValuesGreaterOrEqual10++;
                } else
                {
                    linesWithNumericValuesLessThan10++;
                }
            }
        }
        // Display the results in a dialog with an information icon
        String message = "Statistics:\n" +
                "Lines with numeric values >= 10: " + linesWithNumericValuesGreaterOrEqual10 +
"\n" +
                "Lines with numeric values < 10: " + linesWithNumericValuesLessThan10;

        JOptionPane.showMessageDialog(this, message, "Statistics",
JOptionPane.INFORMATION_MESSAGE);

    } catch (IOException | NumberFormatException ex) {
        JOptionPane.showMessageDialog(this, "Error occurred while reading the file.", "Error",
JOptionPane.ERROR_MESSAGE);
    }
}

private boolean containsNumericValues(String line) {
    return line.matches(".*\\d+.*"); //check if there any digital number in the line
}
//The goal of this method is to find the first word in the input string that represents a numeric value
//and return that value as a double. If no numeric value is found, the method returns 0.0 by default.
private double extractNumericValue(String line) {

    String[] words = line.split("\\s+");

    for (String word : words) {
        try {
            return Double.parseDouble(word);
        } catch (NumberFormatException ignored)
        {
        }
    }
    return 0; // Default value if no numeric value is found
}
}
```

AdminSearch

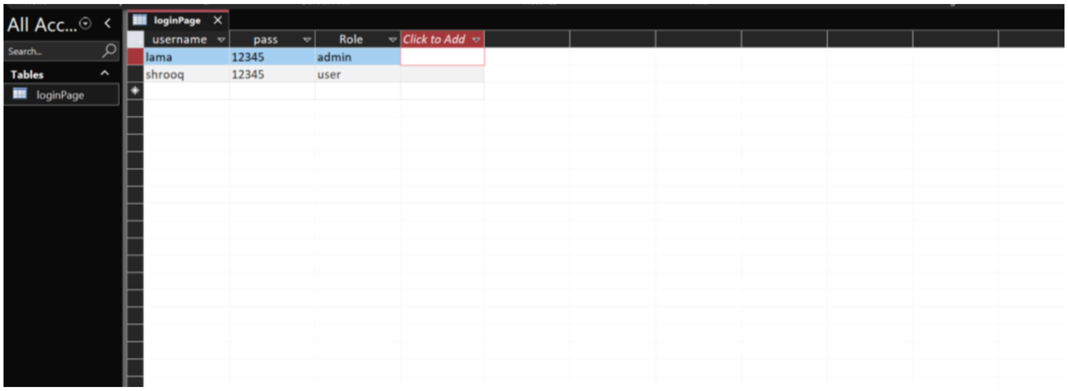
```
package com.mycompany.project;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.*;

public class AdminSearch extends JFrame implements ActionListener {

    private AdminPage mainFrame;
    private JFrame searchFrame = new JFrame ("Search");
    private JLabel l1 , l2;
    private JTextField tf1 , tf2;
    private JButton b1,b2,b3;
    private JTextArea ta1;

    public AdminSearch (AdminPage mF){
    this.mainFrame = mF;
    this.setLocation(300,300);
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    l1 = new JLabel("File to Search: ");
    tf1 = new JTextField(15);
    JPanel p1 = new JPanel();
    p1.add(l1);
    p1.add(tf1);

    ta1 = new JTextArea(20,50);
    JPanel p3 = new JPanel();
    p3.add(ta1);
```

```
        l2 = new JLabel("Keywords: ");
        tf2 = new JTextField(10);
        b1 = new JButton("Search");
        b1.addActionListener(this);
        b2 = new JButton("Clear");
        b2.addActionListener(this);
        b3 = new JButton("Cancel");
        b3.addActionListener(this);

        JPanel p2 = new JPanel();
        p2.add(l2);
        p2.add(tf2);
        p2.add(b1);
        p2.add(b2);
        p2.add(b3);


        JPanel p = (JPanel) this.getContentPane();
        p.setLayout(new BorderLayout(3,1)); //we should add them in the same order we want to
show them
        p.add(p1 , BorderLayout.NORTH);
        p.add(p2 , BorderLayout.CENTER);
        p.add(p3 ,BorderLayout.SOUTH);

        this.pack();
        this.setVisible(true);
        setLocationRelativeTo(mainFrame);
```

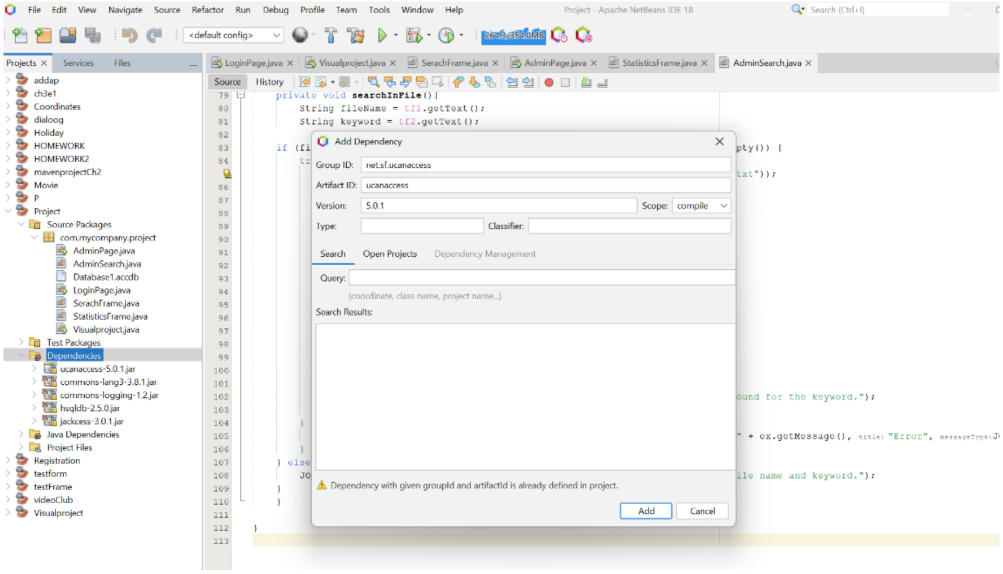
```
    }

    public void actionPerformed(ActionEvent e) {
    if (e.getActionCommand().equals("Search")) {
        searchInFile();
    }
    else if(e.getActionCommand().equals("Clear")){
            tf1.setText("");
            tf2.setText("");
            ta1.setText("");

    }
    else if (e.getActionCommand().equals("Cancel")) {
        System.out.print("here");
        mainFrame.showMainFrame();
        this.dispose();
    }
}

private void searchInFile(){
    String fileName = tf1.getText();
    String keyword = tf2.getText();

if (fileName != null && !fileName.isEmpty() && keyword != null && !keyword.isEmpty()) {
    try {
        BufferedReader reader = new BufferedReader(new FileReader(fileName + ".txt"));
        StringBuilder result = new StringBuilder();
```

```
      String line;
      boolean found = false;

      while ((line = reader.readLine()) != null) {
        if (line.contains(keyword)) {
          result.append(line);
          found = true;
        }
      }
      reader.close();

      if (found) {
        ta1.setText(result.toString());
      }
      else{
        JOptionPane.showMessageDialog(this, "No matches found for the keyword.");
      }
    } catch (IOException ex) {
      JOptionPane.showMessageDialog(this, "Error searching file: " + ex.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
    }
  } else {
    JOptionPane.showMessageDialog(this, "Please enter valid file name and keyword.");
  }
  }
```

```
}
```



username	pass	Role	Click to Add
lama	12345	admin	
shrooq	12345	user	



```
private void searchInFile(){
    String fileName = tf1.getText();
    String keyword = tf2.getText();

if (fi                                        pty()) {
    tr
                                              txt"));

                                              ound for the keyword.");

                                              " + ex.getMessage(), title: "Error", messageType:J
    }
    }
} else
    Jo                                        ile name and keyword.");
}
}

}
```

Add Dependency

Group ID: net.sf.ucanaccess

Artifact ID: ucanaccess

Version: 5.0.1 Scope: compile

Type: Classifier:

Search | Open Projects | Dependency Management

Query:

(coordinate, class name, project name...)

Search Results:

⚠ Dependency with given groupId and artifactId is already defined in project.

Add Cancel