**GUI: Diary App**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.File;

import java.io.FileWriter;

import java.io.IOException;

import java.util.ArrayList;

public class DiaryApp {

private JFrame frame;

private CardLayout cardLayout;

private JPanel mainPanel;

private ArrayList<String> diaryEntries;

// Predefined login credentials

private final String USERNAME = "user";

private final String PASSWORD = "password";

public DiaryApp() {

diaryEntries = new ArrayList<>();

frame = new JFrame("Diary Application");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setSize(400, 400);

cardLayout = new CardLayout();

mainPanel = new JPanel(cardLayout);

// Add panels for login, diary input, and saved entries

mainPanel.add(createLoginPage(), "Login");

mainPanel.add(createDiaryPage(), "Diary");

mainPanel.add(createSavedEntriesPage(), "SavedEntries");

frame.add(mainPanel);

frame.setVisible(true);

}

private JPanel createLoginPage() {

JPanel panel = new JPanel(new GridLayout(3, 2, 10, 10));

panel.setBorder(BorderFactory.createEmptyBorder(50, 50, 50, 50));

JLabel userLabel = new JLabel("Username:");

JLabel passLabel = new JLabel("Password:");

JTextField userField = new JTextField();

JPasswordField passField = new JPasswordField();

JButton loginButton = new JButton("Login");

loginButton.addActionListener(e -> {

String username = userField.getText();

String password = new String(passField.getPassword());

if (USERNAME.equals(username) && PASSWORD.equals(password)) {

JOptionPane.showMessageDialog(frame, "Login Successful!");

cardLayout.show(mainPanel, "Diary");

} else {

JOptionPane.showMessageDialog(frame, "Invalid credentials. Try again.");

}

});

panel.add(userLabel);

panel.add(userField);

panel.add(passLabel);

panel.add(passField);

panel.add(new JLabel());

panel.add(loginButton);

return panel;

}

private JPanel createDiaryPage() {

JPanel panel = new JPanel(new BorderLayout(10, 10));

panel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

JTextArea diaryText = new JTextArea(10, 30);

JButton saveButton = new JButton("Save Entry");

JButton viewEntriesButton = new JButton("View Saved Entries");

JButton saveToFileButton = new JButton("Save to File");

saveButton.addActionListener(e -> {

String entry = diaryText.getText().trim();

if (!entry.isEmpty()) {

diaryEntries.add(entry);

JOptionPane.showMessageDialog(frame, "Diary entry saved!");

diaryText.setText(""); // Clear text area

} else {

JOptionPane.showMessageDialog(frame, "Diary entry cannot be empty.");

}

});

viewEntriesButton.addActionListener(e -> cardLayout.show(mainPanel, "SavedEntries"));

// Save to File button logic

saveToFileButton.addActionListener(e -> saveDiaryToFile(diaryText.getText()));

panel.add(new JScrollPane(diaryText), BorderLayout.CENTER);

JPanel buttonPanel = new JPanel();

buttonPanel.add(saveButton);

buttonPanel.add(viewEntriesButton);

buttonPanel.add(saveToFileButton); // Add the new button

panel.add(buttonPanel, BorderLayout.SOUTH);

return panel;

}

// Method to save diary entry to a text file

private void saveDiaryToFile(String entry) {

if (entry.trim().isEmpty()) {

JOptionPane.showMessageDialog(frame, "Cannot save an empty entry.");

return;

}

try {

// Create a file chooser to select the location

JFileChooser fileChooser = new JFileChooser();

fileChooser.setDialogTitle("Save Diary Entry");

int userSelection = fileChooser.showSaveDialog(frame);

if (userSelection == JFileChooser.APPROVE\_OPTION) {

// Get the file selected by the user

File fileToSave = fileChooser.getSelectedFile();

// Ensure the file ends with ".txt"

if (!fileToSave.getName().endsWith(".txt")) {

fileToSave = new File(fileToSave.getAbsolutePath() + ".txt");

}

// Create a FileWriter and write the entry to the file

FileWriter fileWriter = new FileWriter(fileToSave, true); // true to append

fileWriter.write(entry + "\n\n");

fileWriter.close();

JOptionPane.showMessageDialog(frame, "Diary entry saved to file!");

}

} catch (IOException ex) {

JOptionPane.showMessageDialog(frame, "Error saving the file: " + ex.getMessage());

}

}

private JPanel createSavedEntriesPage() {

JPanel panel = new JPanel(new BorderLayout(10, 10));

panel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

DefaultListModel<String> listModel = new DefaultListModel<>();

JList<String> entryList = new JList<>(listModel);

entryList.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION);

JButton deleteButton = new JButton("Delete Entry");

JButton backButton = new JButton("Back");

deleteButton.addActionListener(e -> {

int selectedIndex = entryList.getSelectedIndex();

if (selectedIndex != -1) {

int confirm = JOptionPane.showConfirmDialog(frame, "Are you sure you want to delete this entry?", "Confirm Delete", JOptionPane.YES\_NO\_OPTION);

if (confirm == JOptionPane.YES\_OPTION) {

diaryEntries.remove(selectedIndex);

listModel.remove(selectedIndex);

JOptionPane.showMessageDialog(frame, "Entry deleted successfully.");

}

} else {

JOptionPane.showMessageDialog(frame, "Please select an entry to delete.");

}

});

backButton.addActionListener(e -> cardLayout.show(mainPanel, "Diary"));

panel.add(new JScrollPane(entryList), BorderLayout.CENTER);

JPanel buttonPanel = new JPanel();

buttonPanel.add(deleteButton);

buttonPanel.add(backButton);

panel.add(buttonPanel, BorderLayout.SOUTH);

// Update the list model when switching to this page

panel.addComponentListener(new java.awt.event.ComponentAdapter() {

@Override

public void componentShown(java.awt.event.ComponentEvent e) {

listModel.clear();

for (String entry : diaryEntries) {

listModel.addElement(entry);

}

}

});

return panel;

}

public static void main(String[] args) {

SwingUtilities.invokeLater(DiaryApp::new);

    }

}