import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.ArrayList;

class Event {

private String eventName;

private String eventDate;

private String eventTime;

public Event(String eventName, String eventDate, String eventTime) {

this.eventName = eventName;

this.eventDate = eventDate;

this.eventTime = eventTime;

}

public String getEventName() {

return eventName;

}

public String getEventDate() {

return eventDate;

}

public String getEventTime() {

return eventTime;

}

@Override

public String toString() {

return "Event Name: " + eventName + ", Date: " + eventDate + ", Time: " + eventTime;

}

}

public class EventSchedulerSwing {

private static ArrayList<Event> eventList = new ArrayList<>();

public static void main(String[] args) {

// Create the main frame

JFrame frame = new JFrame("Event Scheduler");

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

frame.setSize(400, 300);

frame.setLayout(new GridLayout(4, 1));

// Create buttons for the main menu

JButton addEventButton = new JButton("Add Event");

JButton viewEventsButton = new JButton("View Events");

JButton searchEventButton = new JButton("Search Event by Date");

JButton exitButton = new JButton("Exit");

// Add buttons to the frame

frame.add(addEventButton);

frame.add(viewEventsButton);

frame.add(searchEventButton);

frame.add(exitButton);

// Add event listeners for each button

addEventButton.addActionListener(e -> showAddEventDialog(frame));

viewEventsButton.addActionListener(e -> showViewEventsDialog(frame));

searchEventButton.addActionListener(e -> showSearchEventDialog(frame));

exitButton.addActionListener(e -> System.exit(0));

// Display the frame

frame.setVisible(true);

}

private static void showAddEventDialog(JFrame frame) {

// Create a panel for the input fields

JPanel panel = new JPanel(new GridLayout(4, 2));

JTextField nameField = new JTextField();

JTextField dateField = new JTextField();

JTextField timeField = new JTextField();

// Add input fields to the panel

panel.add(new JLabel("Event Name:"));

panel.add(nameField);

panel.add(new JLabel("Event Date (YYYY-MM-DD):"));

panel.add(dateField);

panel.add(new JLabel("Event Time (HH:MM):"));

panel.add(timeField);

// Show the dialog and get user input

int result = JOptionPane.showConfirmDialog(frame, panel, "Add Event", JOptionPane.OK\_CANCEL\_OPTION);

if (result == JOptionPane.OK\_OPTION) {

String name = nameField.getText();

String date = dateField.getText();

String time = timeField.getText();

if (!name.isEmpty() && !date.isEmpty() && !time.isEmpty()) {

eventList.add(new Event(name, date, time));

JOptionPane.showMessageDialog(frame, "Event added successfully!");

} else {

JOptionPane.showMessageDialog(frame, "All fields are required.", "Error", JOptionPane.ERROR\_MESSAGE);

}

}

}

private static void showViewEventsDialog(JFrame frame) {

if (eventList.isEmpty()) {

JOptionPane.showMessageDialog(frame, "No events scheduled.");

} else {

StringBuilder events = new StringBuilder("Scheduled Events:\n");

for (Event event : eventList) {

events.append(event).append("\n");

}

JOptionPane.showMessageDialog(frame, events.toString());

}

}

private static void showSearchEventDialog(JFrame frame) {

String date = JOptionPane.showInputDialog(frame, "Enter date to search (YYYY-MM-DD):");

if (date != null && !date.isEmpty()) {

StringBuilder results = new StringBuilder("Events on " + date + ":\n");

boolean found = false;

for (Event event : eventList) {

if (event.getEventDate().equals(date)) {

results.append(event).append("\n");

found = true;

}

}

if (found) {

JOptionPane.showMessageDialog(frame, results.toString());

} else {

JOptionPane.showMessageDialog(frame, "No events found on this date.");

}

}

}

}