import java.awt.\*;

import java.awt.event.\*;

import java.util.ArrayList;

class BloodDonor {

String name;

String bloodType;

String contact;

int donationUnits;

public BloodDonor(String name, String bloodType, String contact, int donationUnits) {

this.name = name;

this.bloodType = bloodType;

this.contact = contact;

this.donationUnits = donationUnits;

}

@Override

public String toString() {

return "Donor Name: " + name + ", Blood Type: " + bloodType +

", Contact: " + contact + ", Units Donated: " + donationUnits;

}

}

public class EnhancedBloodManagementSystem extends Frame {

private static ArrayList<BloodDonor> donorList = new ArrayList<>();

// UI Components

private TextField nameField, bloodTypeField, contactField, unitsField;

private TextArea displayArea;

public EnhancedBloodManagementSystem() {

// Frame settings

setTitle("Enhanced Blood Management System");

setSize(500, 400);

setLayout(new BorderLayout());

// Header Label

Label header = new Label("Blood Management System", Label.CENTER);

header.setFont(new Font("Arial", Font.BOLD, 18));

header.setBackground(Color.LIGHT\_GRAY);

add(header, BorderLayout.NORTH);

// Input Panel

Panel inputPanel = new Panel(new GridLayout(5, 2, 10, 10));

inputPanel.setBackground(Color.WHITE);

Label nameLabel = new Label("Name:");

nameField = new TextField();

inputPanel.add(nameLabel);

inputPanel.add(nameField);

Label bloodTypeLabel = new Label("Blood Type:");

bloodTypeField = new TextField();

inputPanel.add(bloodTypeLabel);

inputPanel.add(bloodTypeField);

Label contactLabel = new Label("Contact:");

contactField = new TextField();

inputPanel.add(contactLabel);

inputPanel.add(contactField);

Label unitsLabel = new Label("Units Donated:");

unitsField = new TextField();

inputPanel.add(unitsLabel);

inputPanel.add(unitsField);

add(inputPanel, BorderLayout.CENTER);

// Button Panel

Panel buttonPanel = new Panel(new FlowLayout());

Button addButton = new Button("Add Donor");

Button viewButton = new Button("View Donors");

Button exitButton = new Button("Exit");

buttonPanel.add(addButton);

buttonPanel.add(viewButton);

buttonPanel.add(exitButton);

add(buttonPanel, BorderLayout.SOUTH);

// Display Area

displayArea = new TextArea("", 10, 40, TextArea.SCROLLBARS\_VERTICAL\_ONLY);

displayArea.setEditable(false);

displayArea.setBackground(Color.LIGHT\_GRAY);

add(displayArea, BorderLayout.EAST);

// Action Listeners

addButton.addActionListener(e -> addDonor());

viewButton.addActionListener(e -> viewDonors());

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

// Window Listener for Closing

addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent we) {

System.exit(0);

}

});

// Set Frame Visible

setVisible(true);

}

// Method to Add Donor

private void addDonor() {

String name = nameField.getText();

String bloodType = bloodTypeField.getText();

String contact = contactField.getText();

String unitsText = unitsField.getText();

if (name.isEmpty() || bloodType.isEmpty() || contact.isEmpty() || unitsText.isEmpty()) {

displayArea.setText("Error: Please fill in all fields.");

return;

}

int units;

try {

units = Integer.parseInt(unitsText);

if (units <= 0) throw new NumberFormatException();

} catch (NumberFormatException e) {

displayArea.setText("Error: Units donated must be a positive integer.");

return;

}

BloodDonor donor = new BloodDonor(name, bloodType, contact, units);

donorList.add(donor);

// Clear fields and show success message

nameField.setText("");

bloodTypeField.setText("");

contactField.setText("");

unitsField.setText("");

displayArea.setText("Donor added successfully!");

}

// Method to View Donors

private void viewDonors() {

if (donorList.isEmpty()) {

displayArea.setText("No donors available.");

return;

}

StringBuilder donorInfo = new StringBuilder("List of Donors:\n");

for (BloodDonor donor : donorList) {

donorInfo.append(donor).append("\n");

}

displayArea.setText(donorInfo.toString());

}

// Main Method

public static void main(String[] args) {

new EnhancedBloodManagementSystem();

}

}