**New Script GUI**

import java.awt.BorderLayout;

import java.awt.FlowLayout;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.\*;

@SuppressWarnings("serial")

public class NewScriptGUI extends JFrame {

private JPanel container = new JPanel();

private JPanel generic = new JPanel();

private JPanel buttons = new JPanel();

private JPanel script = new JPanel();

@SuppressWarnings("rawtypes")

private JComboBox medsList;

private Customer cust;

private JRadioButton rYes = new JRadioButton("Yes");

private JRadioButton rNo = new JRadioButton("No");

private ButtonGroup radios = new ButtonGroup();

private JLabel l1 = new JLabel("Precription ID:");

private JLabel l2 = new JLabel("Medication:");

private JLabel l3 = new JLabel("Physician Name:");

private JLabel l4 = new JLabel("Physician Phone #");

private JLabel l5 = new JLabel("Date of Issue:");

private JLabel l6 = new JLabel("Expiration Date:");

private JLabel l7 = new JLabel("Number of Refills:");

private JLabel l8 = new JLabel("Units Per Refill:");

private JLabel l9 = new JLabel("Generics Allowed:");

private JTextField scriptIdText;

private JTextField medicationText;

private JTextField physicianText;

private JTextField physicianPhoneText;

private JTextField dateOfIssueText;

private JTextField expirationDateText;

private JTextField numRefillsText;

private JTextField unitRefillsText;

private JButton submit = new JButton("Submit");

private JButton customer = new JButton("Return to Customer");

private boolean isGeneric;

private GUIController control;

public NewScriptGUI() {

createButtons();

add(container, BorderLayout.NORTH);

add(buttons, BorderLayout.SOUTH);

setSize(300, 300);

setUndecorated(true);

}

public NewScriptGUI(GUIController c) {

this();

control = c;

createComboBox();

createTextFields();

createPanel();

}

private void createTextFields() {

scriptIdText = new JTextField();

medicationText = new JTextField();

physicianText = new JTextField();

physicianPhoneText = new JTextField();

physicianPhoneText.setText("Input as ###-###-####");

dateOfIssueText = new JTextField();

dateOfIssueText.setText("Input as mm/dd/yyyy");

expirationDateText = new JTextField();

expirationDateText.setText("Input as mm/dd/yyyy");

numRefillsText = new JTextField();

unitRefillsText = new JTextField();

}

@SuppressWarnings({ "rawtypes", "unchecked" })

private void createComboBox() {

Medications meds = control.getMedications();

String[] medName = new String[meds.size() - 1];

for (int i = 0; i < meds.size() - 1; i++) {

medName[i] = meds.getMedications().get(i).getMedicationName();

}

medsList = new JComboBox(medName);

}

private void createPanel() {

container.setLayout(new GridLayout(2, 1, 0, 0));

script.setLayout(new GridLayout(9, 2, 0, 0));

generic.setLayout(new FlowLayout());

buttons.setLayout(new GridLayout(1, 2, 0, 0));

generic.add(l9);

generic.add(rYes);

generic.add(rNo);

script.add(l1);

script.add(scriptIdText);

script.add(l2);

script.add(medsList);

script.add(l3);

script.add(physicianText);

script.add(l4);

script.add(physicianPhoneText);

script.add(l5);

script.add(dateOfIssueText);

script.add(l6);

script.add(expirationDateText);

script.add(l7);

script.add(numRefillsText);

script.add(l8);

script.add(unitRefillsText);

container.add(script);

container.add(generic);

buttons.add(submit);

buttons.add(customer);

}

public void scriptOwner(Customer c) {

cust = c;

}

private void createButtons() {

// Radio Buttons

rYes.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

isGeneric = true;

}

});

rNo.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

isGeneric = false;

}

});

radios.add(rYes);

radios.add(rNo);

// Solid Buttons

submit.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

cust.getPrescriptionHistoryList().add(

new Prescription(scriptIdText.getText(),

cust.getCustomerId(),

medicationText.getText(),

physicianText.getText(),

physicianPhoneText.getText(), dateOfIssueText.getText(),

expirationDateText.getText(),

Integer.parseInt(numRefillsText.getText()),

Integer.parseInt(unitRefillsText.getText()),

isGeneric));

control.setNewScriptRun(false);

control.setViewRun(true, cust);

// Adds the Script to the Customer's History

}

}); // Adds the exit button's actionListener

customer.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

control.setNewScriptRun(false);

control.setViewRun(true, cust);

}

}); // Adds the exit button's actionListener

}

}