**View Customer GUI**

import java.awt.BorderLayout;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.\*;

@SuppressWarnings("serial")

public class ViewCustomerGUI extends JFrame

{

private JLabel name = new JLabel();

private JLabel phone =new JLabel();

private JLabel dateOfBirth =new JLabel();

private JLabel provider =new JLabel();

private JLabel policy =new JLabel();

private JButton newScript = new JButton("New Prescription");

private JButton drugHistory = new JButton("Prescription History");

private JButton fillScript = new JButton("Refill Prescription");

private JButton main = new JButton("Return to Main Menu");

private JPanel info = new JPanel();

private JPanel buttons = new JPanel();

private JPanel subInfo = new JPanel();

private Customer cust;

private GUIController control;

public ViewCustomerGUI()

{

createLabels();

createButtons();

createPanels();

add(info,BorderLayout.NORTH);

add(subInfo,BorderLayout.CENTER);

add(buttons,BorderLayout.SOUTH);

setSize(300,300);

setUndecorated(true);

}

The above code is initializing the format of the GUI.

public ViewCustomerGUI(GUIController gc)

{

this();

control = gc;

}

The above code retrieves information from the GUI Contorller.

private void createLabels()

{

name.setText("Customer Last Name,Customer First Name");

phone.setText("Phone: 555-555-5555");

dateOfBirth.setText("DOB: 01/01/1970");

provider.setText("Provider: Provider");

policy.setText("Policy Number: 12345");

}

The above code is setting the text fields to their default values.

private void createButtons()

{

newScript.addActionListener(

new ActionListener()

{

@Override

public void actionPerformed(ActionEvent ae)

{

control.setViewRun(false);

control.setScriptOwner(cust);

control.setNewScriptRun(true);

}

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

main.addActionListener(

new ActionListener()

{

@Override

public void actionPerformed(ActionEvent ae)

{

control.setViewRun(false);

control.setMainRun(true);

}

});

The above code adds the exit button's actionListener.

drugHistory.addActionListener(

new ActionListener()

{

@Override

public void actionPerformed(ActionEvent ae)

{

control.setHistoryDisplay(cust);

control.setHistoryRun(true);

control.setViewRun(false);

}

});

The above code adds the exit button's actionListener.

fillScript.addActionListener(

new ActionListener()

{

@Override

public void actionPerformed(ActionEvent ae)

{

//Not added

}

});

The above code adds the exit button's actionListener.

}

private void createPanels() {

subInfo.setLayout(new GridLayout(2,2,0,0));

info.add(name);

subInfo.add(phone);

subInfo.add(dateOfBirth);

subInfo.add(provider);

subInfo.add(policy);

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

buttons.add(newScript);

buttons.add(drugHistory);

buttons.add(fillScript);

buttons.add(main);

}

The above code is formatting the panel and adding the values that are to be returned.

public void display(Customer c) {

cust = c;

name.setText(c.getLastName()+ "," + cust.getFirstName());

phone.setText("Phone: " + cust.getTelephoneNumber());

dateOfBirth.setText("DOB: "+ cust.getDateOfBirth());

provider.setText("Provider: " + cust.getInsuranceProvider());

policy.setText("Policy Number: " + cust.getPolicyNumber());

}

}

The above code is formatting the data to be returned.