package epos;

// -> Gui

import java.nio.file.\*;

import java.util.Formatter;

import java.awt.BorderLayout;

import java.awt.EventQueue;

import javax.swing.BorderFactory;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.Border;

import javax.swing.border.EmptyBorder;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JComboBox;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

import javax.swing.JTextField;

import javax.swing.JButton;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

//

//Json ->

import java.io.FileWriter;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

public class addItem extends JFrame {

private JPanel contentPane;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

addItem frame = new addItem();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

eposGui epos = new eposGui();

private JTextField textFieldAdd;

String selectedJCB;

String newItem;

JSONArray jsnArr = new JSONArray();

public void addArrayToJson(String jsonArrKey, JSONArray jsnArray) throws JSONException {

JSONObject jsonObject = new JSONObject();//create json object to put values in the json file

jsonObject.put(jsonArrKey, jsnArray);// // add json array to json object (to json file \*basically\*) (KEY:VALUE)

}

public void writeToFile(JSONObject jsObj) {//write to a file

try {

FileWriter fw = new FileWriter("my\_jsonEpos.json");

fw.write(jsObj.toString());

fw.flush();

fw.close();

}

catch (Exception e) {

e.printStackTrace();

}

}

public void addArrayToJsonAndFile(String jsonArrKey, JSONArray jsnBurgerArray, JSONObject jsnObj) {

try {

addArrayToJson(jsonArrKey, jsnBurgerArray);

writeToFile(jsnObj);

}

catch (JSONException e) { e.printStackTrace(); }

}

public addItem() throws JSONException {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 531, 443);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

String title = "Add Item";

Border border = BorderFactory.createTitledBorder(title);

contentPane.setBorder(border);

contentPane.setLayout(null);

// - - JLabel

JLabel lblNewLabel = new JLabel("Add item to current Drop down");

lblNewLabel.setBounds(303, 39, 198, 42);

contentPane.add(lblNewLabel);

JLabel lblAddNewDrop = new JLabel("Add new Drop down menu?");

lblAddNewDrop.setBounds(12, 39, 182, 42);

contentPane.add(lblAddNewDrop);

JLabel lblEnterDrop = new JLabel("Select which drop down menu");

lblEnterDrop.setBounds(303, 94, 198, 42);

contentPane.add(lblEnterDrop);

// - - JtextField

textFieldAdd = new JTextField();

textFieldAdd.setToolTipText("Enter item to add");

textFieldAdd.setBounds(303, 232, 187, 22);

contentPane.add(textFieldAdd);

textFieldAdd.setColumns(10);

// - - JCombo Box

String allJCB[] = {"burgerNames", "burgerToppings", "burgerSauces"};//stores all combo boxes

JComboBox JcomBoxArrayList = new JComboBox(allJCB);

JcomBoxArrayList.setToolTipText("Add item");

JcomBoxArrayList.setBounds(303, 151, 187, 50);

contentPane.add(JcomBoxArrayList);

JcomBoxArrayList.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if (e.getSource() == JcomBoxArrayList) {

JComboBox cbObj = (JComboBox)e.getSource();

String selectedItem = (String) cbObj.getSelectedItem();

for (int i = 0; i < allJCB.length; i++) {

if (selectedItem == allJCB[i]) {

selectedJCB = allJCB[i];

}

}

}

}

});

// - - Jbutton

JButton btnAddItem = new JButton("Add item");

btnAddItem.addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent arg0) {

if (allJCB[0] == selectedJCB) {

jsonBurgerArray.put(textFieldAdd.getText());//stores the new item in the jsonArray

try {

jsonObject.put("burgerNames", jsonBurgerArray); }//stores the array to the Json key

catch (JSONException e) {e.printStackTrace();}

// addArrayToJsonAndFile("burgerNames", jsonBurgerArray, jsonObject);//stores the array to the json file, and updates file

}

if (allJCB[1] == selectedJCB) {

jsonToppingArray.put(textFieldAdd.getText());

try { jsonObject.put("burgerToppings", jsonToppingArray); }//stores the array to the Json key

catch (JSONException e) {e.printStackTrace();}

// addArrayToJsonAndFile("burgerToppings", jsonToppingArray, jsonObject);

}

else {

jsonSaucesArray.put(textFieldAdd.getText());

try { jsonObject.put("burgerSauces", jsonToppingArray); }//stores the array to the Json key

catch (JSONException e) {e.printStackTrace();}

// addArrayToJsonAndFile("burgerSauces", jsonSaucesArray, jsonObject);

}

JOptionPane.showMessageDialog(null, "Sucessefully added");

}

});

btnAddItem.setBounds(303, 284, 89, 23);

contentPane.add(btnAddItem);

}

}