Cofetaria

Pentru crearea unei aplicatii care tine evidenta torturilor preparate si comercializare dintr-un magazin am folosit Platforma NetBeans.

Aplicatia gestioneaza torturile dintr-o cofetarie,astfel incat cofetarul poate sa vizualizeze torturile aflate in magazine, dar si sa adauge noi torturi, stearga produse sau sa actualizeze informatiile unui tort, cum ar fi: numele ,pretul, imaginea si gramajul. Aplicatia are conexiune la o baza de date realizata in Acces, unde se afla doua tabele. Primul tabel contine datele despre login si al doilea tabel contine datele despre torurile magazinului.

Pentru a realiza separarea clientilor care intra pe aplicatie cu administratorul aplicatiei, am creat doua JPanel-uri diferite.Clientii putand doar sa vizualizeze torturile si sa le cumpere ceea ce doresc ,iar cofetarul avand drepturi depline pe aplicatie dar si putand sa intre ca si client pentru a verifica sau chiar sa cumpere torture.

Login-ul este creat tot intr-un JPanel pentru a creea mai usor si mai eficient designe-ul, iar pentru a creea legatura cu cele doua JPanel-uri pentru utilizatori, respective administrator, am folosit clasa CardLayout.

Pe JPanel-ul pentru login, am creeat trei butoane:unul pentru a inchide aplicatia si inca doua butoane pentru a login, unul pentru client si unul pentru cofetar. Cand este apasat butonul de Login Cofetar, aplicatia verifica in baza de date daca user-ul are dreptul de acces la acest JPanel.

In aplicatie am creat o functie ,public boolean checkInputs(),care verifica daca campurile:name,price,greutate , sunt completate si returneaza false sau true.

Am mai creat o functie pentru conexiune :

public Connection getConnection() {

try {

Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");

con = DriverManager.getConnection("jdbc:ucanaccess://torturi\_db.accdb");

return con;

} catch (Exception ex) {

Logger.getLogger(Main\_Window.class.getName()).log(Level.SEVERE, null, ex);

JOptionPane.showMessageDialog(null, "Not Connected");

}

return null

}

Functia realizeaza conexiunea la baza de date din acces ,iar daca conexiunea nu este realizata v-a aparea pe ecran mesajul “Not Connected” la deschiderea aplicatiei.

Pentru a redimensiona automat imaginile importate am creat functia :

public ImageIcon ResizeImage(String ImagePath, byte[] pic) {

ImageIcon myImage = null;

if (ImagePath != null) {

myImage = new ImageIcon(ImagePath);

} else {

myImage = new ImageIcon(pic);

}

Image img = myImage.getImage();

Image img2 = img.getScaledInstance(lbl\_image.getWidth(), lbl\_image.getHeight(), Image.SCALE\_SMOOTH);

ImageIcon image = new ImageIcon(img2);

return image;

}

Functiile Show\_Produs\_In\_Table() ,respectic Show\_Produs\_In\_Table2() sunt pentru a introduce in tabelurile din JPanel-ul pentru cofetar,respective client ,informatiile despre torture luate din baza de date,pentru a avea mereu o privire a tuturor produselor comercializate .

public void Show\_Produs\_In\_JTable() {

ArrayList<Produs> list = getProdusList();

DefaultTableModel model = (DefaultTableModel) JTable\_Produs.getModel();

model.setRowCount(0);

Object[] row = new Object[5];

for (int i = 0; i < list.size(); i++) {

row[0] = list.get(i).getId();

row[1] = list.get(i).getName();

row[2] = list.get(i).getPrice();

row[3] = list.get(i).getGreutate();

model.addRow(row);

}

}

Pentru a vizualiza detaliile unui tort fara a folosi tabelul cu torture,am creat o functie care afiseaza detaliile tortului selectat in JTextField-urile alaturate atat in JPanel-ul cofetarului cat si al clientului.

public void ShowItem(int index) {

txt\_id.setText(Integer.toString(getProdusList().get(index).getId()));

txt\_name.setText(getProdusList().get(index).getName());

txt\_price.setText(Float.toString(getProdusList().get(index).getPrice()));

txt\_greutate.setText(Float.toString(getProdusList().get(index).getGreutate()));

lbl\_image.setIcon(ResizeImage(null, getProdusList().get(index).getImage()));

}

Butonul Login Client verifica daca username-ul si password-ul scris de utilizatorul aplicatiei in JTextField-urile respective ,coincid cu cele din tabelul Login din baza de date.Daca coincide JPanel-ul se va modifica si va aparea JPanelul pentru Client.

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

PreparedStatement pst = null;

ResultSet rs = null;

String sql;

sql = "select \* from Login WHERE Username=? and Password=?";

try {

pst = con.prepareStatement(sql);

pst.setString(1, username\_text.getText());

pst.setString(2, password\_text.getText());

rs = pst.executeQuery();

if (rs.next()) {

CardLayout a;

a = (CardLayout) (jPanel1.getLayout());

a.show(jPanel1, "card3");

Show\_Produs\_In\_JTable2();

}

} catch (Exception e) {

JOptionPane.showConfirmDialog(null, e);

}

}

Butonul Login Cofetar l-am facut la fel doar ca mai verifica si daca contul respectiv are drep de acces asupra JPanel-ului pentru administrator.Informatie luata tot din tabelul Login din baza de date.

Butoanele First,Previous,Next si Last realizeaza deplasarea in tabelul afisat pe JPanel si afiseaza pe JTextField-urile alaturate detaliile despre produsul selectat din tabel.

private void Btn\_FirstActionPerformed(java.awt.event.ActionEvent evt) {

pos = 0;

ShowItem(pos);

}

private void Btn\_PreviousActionPerformed(java.awt.event.ActionEvent evt) {

pos--;

if (pos <= 0) {

pos = 0;

}

ShowItem(pos);

}

private void Btn\_NextActionPerformed(java.awt.event.ActionEvent evt) {

pos++;

if (pos >= getProdusList().size()) {

pos = getProdusList().size() - 1;

}

ShowItem(pos);

}

private void Btn\_LastActionPerformed(java.awt.event.ActionEvent evt) {

pos = getProdusList().size() - 1;

ShowItem(pos);

}

Surse de informatie:tutoriale de pe youtube,Oracle.