1/

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package by.ny.client.ui;

import by.ny.client.ConnectionUtil;

import by.ny.server.entity.User;

import by.ny.server.entity.UserRole;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.net.Socket;

import javax.swing.JOptionPane;

import javax.swing.event.ListSelectionEvent;

import javax.swing.event.ListSelectionListener;

import javax.swing.table.DefaultTableModel;

import javax.swing.table.TableModel;

/\*\*

\*

\* @author Natallia Yurush

\*/

public class AdministratorWindow extends javax.swing.JFrame {

/\*\*

\* Creates new form ClientWindow

\*/

public AdministratorWindow() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

tabs = new javax.swing.JTabbedPane();

usersPanel1 = new by.ny.client.ui.UsersPanel();

creditsPanel1 = new by.ny.client.ui.CreditsPanel();

requestedCreditsPanel1 = new by.ny.client.ui.RequestedCreditsPanel();

grantedCreditsPanel1 = new by.ny.client.ui.GrantedCreditsPanel();

expertMethodPanel1 = new by.ny.client.ui.ExpertMethodPanel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Кредитная организация (Администратор)");

tabs.addChangeListener(new javax.swing.event.ChangeListener() {

public void stateChanged(javax.swing.event.ChangeEvent evt) {

tabsStateChanged(evt);

}

});

tabs.addTab("Пользователи", usersPanel1);

tabs.addTab("Кредиты", creditsPanel1);

tabs.addTab("Заявки на кредит", requestedCreditsPanel1);

tabs.addTab("Предоставленные кредиты", grantedCreditsPanel1);

tabs.addTab("Оценка целей кредитов", expertMethodPanel1);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(tabs)

.addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(tabs)

.addContainerGap())

);

pack();

}// </editor-fold>

private void tabsStateChanged(javax.swing.event.ChangeEvent evt) {

// TODO add your handling code here:

}

// Variables declaration - do not modify

private by.ny.client.ui.CreditsPanel creditsPanel1;

private by.ny.client.ui.ExpertMethodPanel expertMethodPanel1;

private by.ny.client.ui.GrantedCreditsPanel grantedCreditsPanel1;

private by.ny.client.ui.RequestedCreditsPanel requestedCreditsPanel1;

private javax.swing.JTabbedPane tabs;

private by.ny.client.ui.UsersPanel usersPanel1;

// End of variables declaration

}

2.

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package by.ny.client.ui;

import by.ny.client.ConnectionUtil;

import by.ny.client.CurrentUserUtil;

import by.ny.server.entity.UserRole;

import by.ny.server.entity.command.user.AuthorizationCommand;

import by.ny.server.entity.result.user.AuthorizationResult;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.net.Socket;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

/\*\*

\*

\* @author Natallia Yurush

\*/

public class AuthorizationDialog extends javax.swing.JFrame {

/\*\*

\* Creates new form AuthorizationDialog

\*/

public AuthorizationDialog() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jLabel1 = new javax.swing.JLabel();

jTextField1 = new javax.swing.JTextField();

jPasswordField1 = new javax.swing.JPasswordField();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Кредитная организация (Вход)");

jLabel1.setText("Авторизация пользователя");

jLabel2.setText("Номер паспорта:");

jLabel3.setText("Пароль:");

jButton1.setText("Войти");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addGap(0, 0, Short.MAX\_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel2, javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jLabel3, javax.swing.GroupLayout.Alignment.TRAILING))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jTextField1)

.addComponent(jPasswordField1, javax.swing.GroupLayout.DEFAULT\_SIZE, 232, Short.MAX\_VALUE)))

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel1)

.addGap(0, 218, Short.MAX\_VALUE))

.addComponent(jButton1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel1)

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel2))

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jPasswordField1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel3))

.addGap(40, 40, 40)

.addComponent(jButton1)

.addContainerGap(21, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

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

try {

// TODO add your handling code here:

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new AuthorizationCommand(jTextField1.getText(), String.valueOf(jPasswordField1.getPassword())));

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

AuthorizationResult result = (AuthorizationResult) inputStream.readObject();

if (result.isAuthorized()) {

this.setVisible(false);

CurrentUserUtil.setCurrentUser(result.getUser());

if (result.getUser().getRole() == UserRole.ADMIN) {

new AdministratorWindow().setVisible(true);

} else if (result.getUser().getRole() == UserRole.CLIENT) {

new ClientWindow().setVisible(true);

}

} else {

JOptionPane.showMessageDialog(this, "Авторизация не выполнена");

}

} catch (Exception ex) {

Logger.getLogger(AuthorizationDialog.class.getName()).log(Level.SEVERE, null, ex);

}

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JPasswordField jPasswordField1;

private javax.swing.JTextField jTextField1;

// End of variables declaration

}

3.

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package by.ny.client.ui;

/\*\*

\*

\* @author Natallia Yurush

\*/

public class ClientWindow extends javax.swing.JFrame {

/\*\*

\* Creates new form ClientWindow

\*/

public ClientWindow() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jTabbedPane1 = new javax.swing.JTabbedPane();

myCreditRequestsPanel1 = new by.ny.client.ui.MyCreditRequestsPanel();

myCreditsPanel2 = new by.ny.client.ui.MyCreditsPanel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Кредитная организация (Клиент)");

setPreferredSize(new java.awt.Dimension(894, 540));

setResizable(false);

setSize(new java.awt.Dimension(894, 540));

jTabbedPane1.addTab("Заявки на кредит", myCreditRequestsPanel1);

jTabbedPane1.addTab("Кредиты", myCreditsPanel2);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jTabbedPane1)

.addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jTabbedPane1)

.addContainerGap())

);

pack();

}// </editor-fold>

// Variables declaration - do not modify

private javax.swing.JTabbedPane jTabbedPane1;

private by.ny.client.ui.MyCreditRequestsPanel myCreditRequestsPanel1;

private by.ny.client.ui.MyCreditsPanel myCreditsPanel2;

// End of variables declaration

}

4/

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package by.ny.client.ui;

import by.ny.client.ConnectionUtil;

import by.ny.server.entity.Credit;

import by.ny.server.entity.command.credit.GetCreditCommand;

import by.ny.server.entity.result.credit.SingleCreditResult;

import java.io.BufferedReader;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.InputStreamReader;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.io.PrintWriter;

import java.net.Socket;

import java.util.ArrayList;

import java.util.Scanner;

import javax.swing.JOptionPane;

import javax.swing.event.ListSelectionEvent;

import javax.swing.event.ListSelectionListener;

import javax.swing.table.DefaultTableModel;

/\*\*

\*

\* @author Natallia Yurush

\*/

public class CreditObjectivesPanel extends javax.swing.JPanel {

//private String[] creditsObjectives;

private ArrayList<String> creditsObjectives;

/\*\*

\* Creates new form CreditObjectivesPanel

\*/

public CreditObjectivesPanel() {

creditsObjectives = new ArrayList<>();

initComponents();

readObjectivesFromFile();

reloadTable();

ObjectivesTable.getSelectionModel().addListSelectionListener(new ListSelectionListener() {

public void valueChanged(ListSelectionEvent event) {

int selectedRow = ObjectivesTable.getSelectedRow();

if (selectedRow < 0) {

return;

}

creditObjectiveField.setText(creditsObjectives.get(selectedRow));

deleteButton.setEnabled(true);

}

});

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jScrollPane1 = new javax.swing.JScrollPane();

ObjectivesTable = new javax.swing.JTable();

saveButton = new javax.swing.JButton();

deleteButton = new javax.swing.JButton();

jScrollPane2 = new javax.swing.JScrollPane();

creditObjectiveField = new javax.swing.JTextArea();

clearButton = new javax.swing.JButton();

ObjectivesTable.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

{null, null},

{null, null},

{null, null},

{null, null}

},

new String [] {

"Номер критерия", "Цель кредита"

}

) {

Class[] types = new Class [] {

java.lang.Integer.class, java.lang.String.class

};

boolean[] canEdit = new boolean [] {

false, false

};

public Class getColumnClass(int columnIndex) {

return types [columnIndex];

}

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

ObjectivesTable.setAutoResizeMode(javax.swing.JTable.AUTO\_RESIZE\_NEXT\_COLUMN);

jScrollPane1.setViewportView(ObjectivesTable);

if (ObjectivesTable.getColumnModel().getColumnCount() > 0) {

ObjectivesTable.getColumnModel().getColumn(1).setResizable(false);

}

saveButton.setText("Сохранить");

saveButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

saveButtonActionPerformed(evt);

}

});

deleteButton.setText("Удалить");

deleteButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

deleteButtonActionPerformed(evt);

}

});

creditObjectiveField.setColumns(20);

creditObjectiveField.setFont(new java.awt.Font("Arial", 0, 14)); // NOI18N

creditObjectiveField.setRows(5);

jScrollPane2.setViewportView(creditObjectiveField);

clearButton.setText("Очистить");

clearButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

clearButtonActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);

this.setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jScrollPane1)

.addContainerGap())

.addGroup(layout.createSequentialGroup()

.addGap(43, 43, 43)

.addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, 458, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 68, Short.MAX\_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(saveButton, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, 231, Short.MAX\_VALUE)

.addComponent(deleteButton, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(clearButton, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGap(55, 55, 55))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 231, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(39, 39, 39)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jScrollPane2)

.addGroup(layout.createSequentialGroup()

.addComponent(saveButton)

.addGap(32, 32, 32)

.addComponent(clearButton)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 31, Short.MAX\_VALUE)

.addComponent(deleteButton)))

.addContainerGap(36, Short.MAX\_VALUE))

);

}// </editor-fold>

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

if(creditsObjectives.size() == 4) {

clearCreditFields();

JOptionPane.showMessageDialog(this, "Невозможно добавить больше 4 целей");

return;

}

creditsObjectives.add((String)creditObjectiveField.getText());

readObjectivesToFile();

clearCreditFields();

reloadTable();

}

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

clearCreditFields();

}

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

String str = creditObjectiveField.getText();

creditsObjectives.remove(str);

clearCreditFields();

reloadTable();

}

private void clearCreditFields() {

creditObjectiveField.setText("");

try {

ObjectivesTable.clearSelection();

} catch (Exception e) {

}

deleteButton.setEnabled(false);

}

private void readObjectivesToFile() {

try {

PrintWriter writer = new PrintWriter("output.txt", "windows-1251");

for(String str : creditsObjectives) {

writer.write(str + System.getProperty("line.separator"));

}

writer.close();

} catch(Exception e) {

e.printStackTrace();

}

}

private void readObjectivesFromFile() {

try (BufferedReader reader = new BufferedReader(new InputStreamReader( new FileInputStream("output.txt") , "windows-1251"))) {

String line;

while ((line = reader.readLine()) != null) {

creditsObjectives.add(line);

}

}

catch (Exception e) {

e.printStackTrace();

}

}

private void reloadTable() {

int rowsCount = ((DefaultTableModel) ObjectivesTable.getModel()).getRowCount();

for (int i = 0; i < rowsCount; i ++) {

((DefaultTableModel) ObjectivesTable.getModel()).removeRow(0);

}

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

Object[] row = new Object[2];

row[0] = i+1;

row[1] = creditsObjectives.get(i);

((DefaultTableModel) ObjectivesTable.getModel()).addRow(row);

}

}

// Variables declaration - do not modify

private javax.swing.JTable ObjectivesTable;

private javax.swing.JButton clearButton;

private javax.swing.JTextArea creditObjectiveField;

private javax.swing.JButton deleteButton;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JButton saveButton;

// End of variables declaration

}

5/

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package by.ny.client.ui;

import by.ny.client.ConnectionUtil;

import by.ny.server.entity.Credit;

import by.ny.server.entity.User;

import by.ny.server.entity.command.credit.DeleteCreditCommand;

import by.ny.server.entity.command.credit.GetCreditCommand;

import by.ny.server.entity.command.credit.ListCreditsCommand;

import by.ny.server.entity.command.credit.SaveCreditCommand;

import by.ny.server.entity.command.user.ListUsersCommand;

import by.ny.server.entity.result.credit.CreditsResult;

import by.ny.server.entity.result.credit.SingleCreditResult;

import by.ny.server.entity.result.credit.UpdateCreditStatusResult;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.net.Socket;

import javax.swing.JOptionPane;

import javax.swing.event.ListSelectionEvent;

import javax.swing.event.ListSelectionListener;

import javax.swing.table.DefaultTableModel;

/\*\*

\*

\* @author Natallia Yurush

\*/

public class CreditsPanel extends javax.swing.JPanel {

/\*\*

\* Creates new form CreditsPanel

\*/

public CreditsPanel() {

initComponents();

reloadCreditsTableData();

creditsTable.getSelectionModel().addListSelectionListener(new ListSelectionListener() {

public void valueChanged(ListSelectionEvent event) {

int selectedRow = creditsTable.getSelectedRow();

if (selectedRow < 0) {

return;

}

Integer creditId = (Integer) creditsTable.getValueAt(selectedRow, 0);

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new GetCreditCommand(creditId));

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

SingleCreditResult result = (SingleCreditResult) inputStream.readObject();

Credit credit = result.getCredit();

idField.setText(String.valueOf(credit.getId()));

purposeField.setText(credit.getPurpose());

maxAmountField.setValue(credit.getMaxAmount());

percentField.setValue(credit.getPercent());

maxTermField.setValue(credit.getMaxTerm());

deleteButton.setEnabled(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jScrollPane1 = new javax.swing.JScrollPane();

creditsTable = new javax.swing.JTable();

jLabel1 = new javax.swing.JLabel();

idField = new javax.swing.JTextField();

jLabel2 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

saveButton = new javax.swing.JButton();

clearButton = new javax.swing.JButton();

deleteButton = new javax.swing.JButton();

jScrollPane2 = new javax.swing.JScrollPane();

purposeField = new javax.swing.JTextArea();

maxAmountField = new javax.swing.JSpinner();

percentField = new javax.swing.JSpinner();

maxTermField = new javax.swing.JSpinner();

creditsTable.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

{null, null, null, null, null},

{null, null, null, null, null},

{null, null, null, null, null},

{null, null, null, null, null}

},

new String [] {

"Идентификатор", "Цель выдачи", "Максимальная сумма", "Процент", "Максимальный срок (месяцы)"

}

) {

Class[] types = new Class [] {

java.lang.Integer.class, java.lang.String.class, java.lang.String.class, java.lang.String.class, java.lang.String.class

};

boolean[] canEdit = new boolean [] {

false, false, false, false, false

};

public Class getColumnClass(int columnIndex) {

return types [columnIndex];

}

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

jScrollPane1.setViewportView(creditsTable);

jLabel1.setText("Идентификатор:");

idField.setEnabled(false);

jLabel2.setText("Цель выдачи:");

jLabel4.setText("Максимальная сумма:");

jLabel5.setText("Процент:");

jLabel6.setText("Максимальный срок (месяцев):");

saveButton.setText("Сохранить");

saveButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

saveButtonActionPerformed(evt);

}

});

clearButton.setText("Очистить");

clearButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

clearButtonActionPerformed(evt);

}

});

deleteButton.setText("Удалить");

deleteButton.setEnabled(false);

deleteButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

deleteButtonActionPerformed(evt);

}

});

purposeField.setColumns(20);

purposeField.setRows(5);

jScrollPane2.setViewportView(purposeField);

maxAmountField.setModel(new javax.swing.SpinnerNumberModel(1.0d, 1.0d, 1000000.0d, 0.01d));

percentField.setModel(new javax.swing.SpinnerNumberModel(1, 1, 300, 1));

maxTermField.setModel(new javax.swing.SpinnerNumberModel(1, 1, 240, 1));

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane1)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jLabel4)

.addGap(18, 18, 18)

.addComponent(maxAmountField, javax.swing.GroupLayout.DEFAULT\_SIZE, 300, Short.MAX\_VALUE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel1)

.addComponent(jLabel2))

.addGap(51, 51, 51)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(idField)

.addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT\_SIZE, 300, Short.MAX\_VALUE))))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(26, 26, 26)

.addComponent(saveButton, javax.swing.GroupLayout.DEFAULT\_SIZE, 113, Short.MAX\_VALUE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(clearButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 120, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(deleteButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 120, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel6)

.addComponent(jLabel5))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(percentField)

.addComponent(maxTermField))))))

.addContainerGap())

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 298, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1)

.addComponent(idField, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel5)

.addComponent(percentField, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel2)

.addComponent(jLabel6)

.addComponent(maxTermField, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(9, 9, 9)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel4)

.addComponent(saveButton)

.addComponent(clearButton)

.addComponent(deleteButton)

.addComponent(maxAmountField, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);

this.setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addGap(0, 0, Short.MAX\_VALUE)

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

);

}// </editor-fold>

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

Integer creditId = Integer.valueOf(idField.getText());

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new DeleteCreditCommand(creditId));

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

UpdateCreditStatusResult result = (UpdateCreditStatusResult) inputStream.readObject();

if (result.isSuccess()) {

clearCreditFields();

reloadCreditsTableData();

JOptionPane.showMessageDialog(this, "Кредит успешно удален.");

} else {

JOptionPane.showMessageDialog(this, "Не удалось удалить кредит");

}

} catch(Exception e) {

e.printStackTrace();

}

}

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

clearCreditFields();

}

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

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

Integer id = "".equals(idField.getText()) ? null : Integer.valueOf(idField.getText());

outputStream.writeObject(new SaveCreditCommand(new Credit(id, purposeField.getText(), (Double) maxAmountField.getValue(),

(Integer) percentField.getValue(), (Integer) maxTermField.getValue())));

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

UpdateCreditStatusResult result = (UpdateCreditStatusResult) inputStream.readObject();

if (result.isSuccess()) {

clearCreditFields();

reloadCreditsTableData();

JOptionPane.showMessageDialog(this, "Кредит успешно сохранен.");

} else {

JOptionPane.showMessageDialog(this, "Не удалось сохранить кредит");

}

} catch (Exception e) {

e.printStackTrace();

}

}

private void clearCreditFields() {

idField.setText("");

purposeField.setText("");

maxAmountField.setValue(1.0);

percentField.setValue(1);

maxTermField.setValue(1);

try {

creditsTable.clearSelection();

} catch (Exception e) {

}

deleteButton.setEnabled(false);

}

private void reloadCreditsTableData() {

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new ListCreditsCommand());

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

CreditsResult result = (CreditsResult) inputStream.readObject();

int rowsCount = ((DefaultTableModel) creditsTable.getModel()).getRowCount();

for (int i = 0; i < rowsCount; i ++) {

((DefaultTableModel) creditsTable.getModel()).removeRow(0);

}

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

Credit credit = result.getCredits().get(i);

Object[] row = new Object[6];

row[0] = credit.getId();

row[1] = credit.getPurpose();

row[2] = credit.getMaxAmount();

row[3] = credit.getPercent();

row[4] = credit.getMaxTerm();

((DefaultTableModel) creditsTable.getModel()).addRow(row);

}

} catch (Exception e) {

e.printStackTrace();

}

}

// Variables declaration - do not modify

private javax.swing.JButton clearButton;

private javax.swing.JTable creditsTable;

private javax.swing.JButton deleteButton;

private javax.swing.JTextField idField;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JSpinner maxAmountField;

private javax.swing.JSpinner maxTermField;

private javax.swing.JSpinner percentField;

private javax.swing.JTextArea purposeField;

private javax.swing.JButton saveButton;

// End of variables declaration

}

6/

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package by.ny.client.ui;

/\*\*

\*

\* @author Natallia Yurush

\*/

public class ExpertMethodPanel extends javax.swing.JPanel {

/\*\*

\* Creates new form ExpertMethodPanel

\*/

public ExpertMethodPanel() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jTabbedPane1 = new javax.swing.JTabbedPane();

creditObjectivesPanel1 = new by.ny.client.ui.CreditObjectivesPanel();

pairComparisonMethodPanel1 = new by.ny.client.ui.PairComparisonMethodPanel();

jTabbedPane1.addTab("��������", creditObjectivesPanel1);

jTabbedPane1.addTab("����������� ������", pairComparisonMethodPanel1);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);

this.setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jTabbedPane1)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jTabbedPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 487, Short.MAX\_VALUE)

);

}// </editor-fold>

// Variables declaration - do not modify

private by.ny.client.ui.CreditObjectivesPanel creditObjectivesPanel1;

private javax.swing.JTabbedPane jTabbedPane1;

private by.ny.client.ui.PairComparisonMethodPanel pairComparisonMethodPanel1;

// End of variables declaration

}

7/

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package by.ny.client.ui;

import by.ny.client.ConnectionUtil;

import by.ny.server.entity.Credit;

import by.ny.server.entity.CreditStatus;

import by.ny.server.entity.User;

import by.ny.server.entity.UserCredit;

import by.ny.server.entity.UserRole;

import by.ny.server.entity.command.credit.GetCreditCommand;

import by.ny.server.entity.command.user.GetUserByIdCommand;

import by.ny.server.entity.command.user.ListUsersCommand;

import by.ny.server.entity.command.usercredit.ListUserCreditCommand;

import by.ny.server.entity.result.credit.SingleCreditResult;

import by.ny.server.entity.result.user.GetSingleUserResult;

import by.ny.server.entity.result.user.ListUsersResult;

import by.ny.server.entity.result.usercredit.ListUserCreditResult;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.net.Socket;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

import javax.swing.event.ListSelectionEvent;

import javax.swing.event.ListSelectionListener;

import javax.swing.table.AbstractTableModel;

/\*\*

\*

\* @author Natallia Yurush

\*/

public class GrantedCreditsPanel extends javax.swing.JPanel {

/\*\*

\* Creates new form GrantedCreditsPanel

\*/

public GrantedCreditsPanel() {

initComponents();

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new ListUsersCommand());

outputStream.flush();

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

ListUsersResult userListResult = (ListUsersResult) inputStream.readObject();

final List<UserCredit> foundUserCredits = new ArrayList();

for(User user : userListResult.getUsers()){

if( user.getRole() == UserRole.ADMIN){

continue;

}

socket = ConnectionUtil.getSocket();

outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new ListUserCreditCommand(user));

inputStream = new ObjectInputStream(socket.getInputStream());

ListUserCreditResult result = (ListUserCreditResult) inputStream.readObject();

outputStream.flush();

for (UserCredit userCredit : result.getUserCredits()) {

if (userCredit.getStatus() != CreditStatus.REQUESTED) {

foundUserCredits.add(userCredit);

}

}

}

socket.close();

reloadTable(foundUserCredits);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

lastNameField = new javax.swing.JTextField();

NumberPassportField = new javax.swing.JTextField();

jScrollPane1 = new javax.swing.JScrollPane();

grantedCreditsTable = new javax.swing.JTable();

searchButton = new javax.swing.JButton();

resetButton = new javax.swing.JButton();

jLabel2.setText("Фамилия:");

jLabel3.setText("Номер паспорта:");

lastNameField.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

lastNameFieldActionPerformed(evt);

}

});

NumberPassportField.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

NumberPassportFieldActionPerformed(evt);

}

});

grantedCreditsTable.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

{null, null, null, null, null, null},

{null, null, null, null, null, null},

{null, null, null, null, null, null},

{null, null, null, null, null, null}

},

new String [] {

"ФИО", "Номер паспорта", "Сумма кредита", "Дата выдачи", "Дата погашения", "Статус"

}

) {

Class[] types = new Class [] {

java.lang.String.class, java.lang.String.class, java.lang.Double.class, java.lang.String.class, java.lang.String.class, java.lang.String.class

};

boolean[] canEdit = new boolean [] {

false, false, false, false, false, false

};

public Class getColumnClass(int columnIndex) {

return types [columnIndex];

}

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

grantedCreditsTable.setEnabled(false);

jScrollPane1.setViewportView(grantedCreditsTable);

searchButton.setText("Поиск");

searchButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

searchButtonActionPerformed(evt);

}

});

resetButton.setText("Сброс");

resetButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

resetButtonActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);

this.setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane1)

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 68, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(lastNameField, javax.swing.GroupLayout.PREFERRED\_SIZE, 245, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jLabel3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(NumberPassportField, javax.swing.GroupLayout.PREFERRED\_SIZE, 245, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 38, Short.MAX\_VALUE)

.addComponent(searchButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 89, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addGap(0, 0, Short.MAX\_VALUE)

.addComponent(resetButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 88, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(lastNameField, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED\_SIZE, 28, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(NumberPassportField, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(searchButton))

.addGap(18, 18, 18)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 346, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(resetButton)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

}// </editor-fold>

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

// TODO add your handling code here:

}

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

if(lastNameField.getText().equals("") && NumberPassportField.getText().equals("")){

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new ListUsersCommand());

outputStream.flush();

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

ListUsersResult userListResult = (ListUsersResult) inputStream.readObject();

final List<UserCredit> foundUserCredits = new ArrayList();

for(User user : userListResult.getUsers()){

if( user.getRole() == UserRole.ADMIN){

continue;

}

socket = ConnectionUtil.getSocket();

outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new ListUserCreditCommand(user));

inputStream = new ObjectInputStream(socket.getInputStream());

ListUserCreditResult result = (ListUserCreditResult) inputStream.readObject();

outputStream.flush();

for (UserCredit userCredit : result.getUserCredits()) {

if (userCredit.getStatus() != CreditStatus.REQUESTED) {

foundUserCredits.add(userCredit);

}

}

}

socket.close();

reloadTable(foundUserCredits);

} catch (Exception e) {

e.printStackTrace();

}

} else if (!lastNameField.getText().equals("")) {

try{

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new ListUsersCommand());

outputStream.flush();

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

ListUsersResult userListResult = (ListUsersResult) inputStream.readObject();

final List<UserCredit> foundUserCredits = new ArrayList();

for(User user : userListResult.getUsers()){

if(!user.getLastName().contains(lastNameField.getText())) {

continue;

}

socket = ConnectionUtil.getSocket();

outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new ListUserCreditCommand(user));

inputStream = new ObjectInputStream(socket.getInputStream());

ListUserCreditResult result = (ListUserCreditResult) inputStream.readObject();

outputStream.flush();

for (UserCredit userCredit : result.getUserCredits()) {

if (userCredit.getStatus() != CreditStatus.REQUESTED) {

foundUserCredits.add(userCredit);

}

}

}

socket.close();

reloadTable(foundUserCredits);

} catch (Exception e) {

e.printStackTrace();

}

} else if(!NumberPassportField.getText().equals("")){

try{

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new ListUsersCommand());

outputStream.flush();

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

ListUsersResult userListResult = (ListUsersResult) inputStream.readObject();

final List<UserCredit> foundUserCredits = new ArrayList();

for(User user : userListResult.getUsers()){

if(!user.getPassportNumber().contains(NumberPassportField.getText())) {

continue;

}

socket = ConnectionUtil.getSocket();

outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new ListUserCreditCommand(user));

inputStream = new ObjectInputStream(socket.getInputStream());

ListUserCreditResult result = (ListUserCreditResult) inputStream.readObject();

outputStream.flush();

for (UserCredit userCredit : result.getUserCredits()) {

if (userCredit.getStatus() != CreditStatus.REQUESTED) {

foundUserCredits.add(userCredit);

}

}

}

socket.close();

reloadTable(foundUserCredits);

} catch (Exception e) {

e.printStackTrace();

}

} else {

try{

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new ListUsersCommand());

outputStream.flush();

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

ListUsersResult userListResult = (ListUsersResult) inputStream.readObject();

final List<UserCredit> foundUserCredits = new ArrayList();

for(User user : userListResult.getUsers()){

if(!(user.getPassportNumber().contains(NumberPassportField.getText()) && user.getLastName().contains(lastNameField.getText()))) {

continue;

}

socket = ConnectionUtil.getSocket();

outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new ListUserCreditCommand(user));

inputStream = new ObjectInputStream(socket.getInputStream());

ListUserCreditResult result = (ListUserCreditResult) inputStream.readObject();

outputStream.flush();

for (UserCredit userCredit : result.getUserCredits()) {

if (userCredit.getStatus() != CreditStatus.REQUESTED) {

foundUserCredits.add(userCredit);

}

}

}

socket.close();

reloadTable(foundUserCredits);

} catch (Exception e) {

e.printStackTrace();

}

}

}

private void clearSearchFields() {

lastNameField.setText("");

NumberPassportField.setText("");

//resetButton.setEnabled(false);

}

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

// TODO add your handling code here:

}

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

clearSearchFields();

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new ListUsersCommand());

outputStream.flush();

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

ListUsersResult userListResult = (ListUsersResult) inputStream.readObject();

final List<UserCredit> foundUserCredits = new ArrayList();

for(User user : userListResult.getUsers()){

if( user.getRole() == UserRole.ADMIN){

continue;

}

socket = ConnectionUtil.getSocket();

outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new ListUserCreditCommand(user));

inputStream = new ObjectInputStream(socket.getInputStream());

ListUserCreditResult result = (ListUserCreditResult) inputStream.readObject();

outputStream.flush();

for (UserCredit userCredit : result.getUserCredits()) {

if (userCredit.getStatus() != CreditStatus.REQUESTED) {

foundUserCredits.add(userCredit);

}

}

}

socket.close();

reloadTable(foundUserCredits);

} catch (Exception e) {

e.printStackTrace();

}

}

public void reloadTable(List<UserCredit> userCreditList) {

grantedCreditsTable.setModel(new AbstractTableModel() {

private String columns[] = new String [] { "ФИО", "Номер паспорта", "Сумма кредита", "Дата выдачи", "Дата погашения", "Статус" };

@Override

public int getRowCount() {

return userCreditList.size();

}

@Override

public int getColumnCount() {

return columns.length;

}

@Override

public Object getValueAt(int rowIndex, int columnIndex) {

UserCredit userCredit = userCreditList.get(rowIndex);

if (columnIndex == 0) {

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new GetUserByIdCommand(userCredit.getUserId()));

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

GetSingleUserResult result = (GetSingleUserResult) inputStream.readObject();

User user = result.getUser();

socket.close();

String string = user.getLastName() + " " + user.getFirstName() + " " + user.getMiddleName();

return string;

} catch (Exception e) {

e.printStackTrace();

}

} else if(columnIndex == 1) {

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new GetUserByIdCommand(userCredit.getUserId()));

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

GetSingleUserResult result = (GetSingleUserResult) inputStream.readObject();

User user = result.getUser();

socket.close();

return user.getPassportNumber();

} catch (Exception e) {

e.printStackTrace();

}

} else if (columnIndex == 2) {

return userCredit.getIssueAmount();

} else if (columnIndex == 3) {

SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");

return format.format(userCredit.getIssueDate());

} else if (columnIndex == 4) {

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new GetCreditCommand(userCredit.getCreditId()));

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

SingleCreditResult result = (SingleCreditResult) inputStream.readObject();

Credit credit = result.getCredit();

socket.close();

SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");

Date date = userCredit.getIssueDate();

date.setMonth(userCredit.getIssueDate().getMonth() + credit.getMaxTerm());

return format.format(date);

} catch (Exception e) {

e.printStackTrace();

}

} else if (columnIndex == 5) {

if(userCredit.getStatus() == CreditStatus.CLOSED)

return "погашен";

else if(userCredit.getStatus() == CreditStatus.ACTIVE)

return "активный";

else if(userCredit.getStatus() == CreditStatus.REQUESTED)

return "запрошенный";

}

return "";

}

// needed to show column names in JTable

public String getColumnName(int col) {

return columns[col];

}

public Class getColumnClass(int col) {

if (col == 2) {

return Double.class;

}

else {

return String.class;

}

}

});

grantedCreditsTable.getSelectionModel().addListSelectionListener(

new ListSelectionListener() {

public void valueChanged(ListSelectionEvent event) {

int selectedRow = grantedCreditsTable.getSelectedRow();

if (selectedRow < 0) {

return;

}

// userCreditId = requestedUserCredits.get(selectedRow).getId();

searchButton.setEnabled(true);

//confirmRequestedCreditButton.setEnabled(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JTextField NumberPassportField;

private javax.swing.JTable grantedCreditsTable;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTextField lastNameField;

private javax.swing.JButton resetButton;

private javax.swing.JButton searchButton;

// End of variables declaration

}

8.

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package by.ny.client.ui;

import by.ny.client.ConnectionUtil;

import by.ny.client.CurrentUserUtil;

import by.ny.server.entity.Credit;

import by.ny.server.entity.CreditStatus;

import by.ny.server.entity.UserCredit;

import by.ny.server.entity.command.credit.GetCreditCommand;

import by.ny.server.entity.command.usercredit.DeleteUserCreditCommand;

import by.ny.server.entity.command.usercredit.GetUserCreditByIdCommand;

import by.ny.server.entity.command.usercredit.ListUserCreditCommand;

import by.ny.server.entity.command.usercredit.PayUserCreditCommand;

import by.ny.server.entity.command.usercredit.SaveUserCreditCommand;

import by.ny.server.entity.result.credit.SingleCreditResult;

import by.ny.server.entity.result.usercredit.DeleteUserCreditResult;

import by.ny.server.entity.result.usercredit.ListUserCreditResult;

import by.ny.server.entity.result.usercredit.SingleUserCreditResult;

import by.ny.server.entity.result.usercredit.UpdateUserCreditStatusResult;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.net.Socket;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.List;

import javax.swing.DefaultComboBoxModel;

import javax.swing.JOptionPane;

import javax.swing.SpinnerNumberModel;

import javax.swing.event.ListSelectionEvent;

import javax.swing.event.ListSelectionListener;

import javax.swing.table.AbstractTableModel;

/\*\*

\*

\* @author Natallia Yurush

\*/

public class MyActiveCreditsPanel extends javax.swing.JPanel {

private Integer userCreditId;

public MyActiveCreditsPanel() {

initComponents();

reloadTableData();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jScrollPane1 = new javax.swing.JScrollPane();

activeCreditsTable = new javax.swing.JTable();

jLabel1 = new javax.swing.JLabel();

paymentSpinner = new javax.swing.JSpinner();

payButton = new javax.swing.JButton();

setPreferredSize(new java.awt.Dimension(836, 300));

activeCreditsTable.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

{null, null, null, null, null},

{null, null, null, null, null},

{null, null, null, null, null},

{null, null, null, null, null}

},

new String [] {

"Цель выдачи", "Сумма кредита", "Основной платеж (мес.)", "Процентный платеж (мес.)", "Остаток"

}

) {

Class[] types = new Class [] {

java.lang.String.class, java.lang.Double.class, java.lang.Double.class, java.lang.Double.class, java.lang.Double.class

};

boolean[] canEdit = new boolean [] {

false, false, false, false, false

};

public Class getColumnClass(int columnIndex) {

return types [columnIndex];

}

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

jScrollPane1.setViewportView(activeCreditsTable);

jLabel1.setText("Сумма оплаты:");

paymentSpinner.setModel(new javax.swing.SpinnerNumberModel(1.0d, 1.0d, null, 0.1d));

payButton.setText("Оплатить");

payButton.setEnabled(false);

payButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

payButtonActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);

this.setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 829, Short.MAX\_VALUE)

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel1)

.addGap(18, 18, 18)

.addComponent(paymentSpinner, javax.swing.GroupLayout.PREFERRED\_SIZE, 473, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(payButton, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)))

.addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 316, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1)

.addComponent(paymentSpinner, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(payButton))

.addGap(76, 76, 76))

);

}// </editor-fold>

public void reloadTableData() {

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new ListUserCreditCommand(CurrentUserUtil.getCurrentUser()));

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

ListUserCreditResult result = (ListUserCreditResult) inputStream.readObject();

final List<UserCredit> activeUserCredits = new ArrayList();

for (UserCredit userCredit : result.getUserCredits()) {

if (userCredit.getStatus() == CreditStatus.ACTIVE) {

activeUserCredits.add(userCredit);

}

}

activeCreditsTable.setModel(new AbstractTableModel() {

private String columns[] = new String [] { "Цель выдачи", "Сумма кредита", "Основной платеж (мес.)", "Процентный платеж (мес.)", "Остаток" };

@Override

public int getRowCount() {

return activeUserCredits.size();

}

@Override

public int getColumnCount() {

return columns.length;

}

@Override

public Object getValueAt(int rowIndex, int columnIndex) {

UserCredit userCredit = activeUserCredits.get(rowIndex);

//((SpinnerNumberModel) paymentSpinner.getModel()).setMaximum(userCredit.getIssueAmount());

if (columnIndex == 0) {

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new GetCreditCommand(userCredit.getCreditId()));

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

SingleCreditResult result = (SingleCreditResult) inputStream.readObject();

Credit credit = result.getCredit();

return credit.getPurpose();

} catch (Exception e) {

e.printStackTrace();

}

} else if (columnIndex == 1) {

return userCredit.getIssueAmount();

} else if (columnIndex == 2) {

return userCredit.getBasicMonthlyPayment();

} else if(columnIndex == 3) {

return userCredit.getInterestMonthlyPayment();

} else if(columnIndex == 4) {

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new GetCreditCommand(userCredit.getCreditId()));

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

SingleCreditResult result = (SingleCreditResult) inputStream.readObject();

Credit credit = result.getCredit();

return userCredit.getIssueAmount() + (userCredit.getIssueAmount()\*credit.getPercent()/1200) - userCredit.getContributed\_amount();

} catch (Exception e) {

e.printStackTrace();

}

}

return "";

}

// needed to show column names in JTable

public String getColumnName(int col) {

return columns[col];

}

public Class getColumnClass(int col) {

if (col == 0) {

return String.class;

}

else {

return Double.class;

}

}

});

activeCreditsTable.getSelectionModel().addListSelectionListener(

new ListSelectionListener() {

public void valueChanged(ListSelectionEvent event) {

int selectedRow = activeCreditsTable.getSelectedRow();

if (selectedRow < 0) {

return;

}

userCreditId = activeUserCredits.get(selectedRow).getId();

payButton.setEnabled(true);

//((SpinnerNumberModel) paymentSpinner.getModel()).setMaximum(100000);

//((SpinnerNumberModel) paymentSpinner.getModel()).setMaximum(userCredit.getIssueAmount());

}

});

} catch (Exception e) {

e.printStackTrace();

}

}

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

if (userCreditId == null) {

payButton.setEnabled(false);

return;

}

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new GetUserCreditByIdCommand(userCreditId));

outputStream.flush();

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

SingleUserCreditResult singleUserCreditResult = (SingleUserCreditResult) inputStream.readObject();

UserCredit userCredit = singleUserCreditResult.getUser();

// paymentSpinner.enableInputMethods(true);

// ((SpinnerNumberModel) paymentSpinner.getModel()).setMaximum(userCredit.getInterestMonthlyPayment() + userCredit.getBasicMonthlyPayment());

//CurrentUserUtil.getCurrentUser()

socket = ConnectionUtil.getSocket();

outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new PayUserCreditCommand(userCredit, (Double) paymentSpinner.getValue()));

inputStream = new ObjectInputStream(socket.getInputStream());

UpdateUserCreditStatusResult result = (UpdateUserCreditStatusResult) inputStream.readObject();

if (result.isSuccess()) {

payButton.setEnabled(false);

reloadTableData();

JOptionPane.showMessageDialog(this, "Оплата успешно произведена.");

} else {

JOptionPane.showMessageDialog(this, "Не удалось произвести оплату.");

}

} catch (Exception e) {

e.printStackTrace();

}

}

// Variables declaration - do not modify

private javax.swing.JTable activeCreditsTable;

private javax.swing.JLabel jLabel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JButton payButton;

private javax.swing.JSpinner paymentSpinner;

// End of variables declaration

}

9

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package by.ny.client.ui;

/\*\*

\*

\* @author Natallia Yurush

\*/

public class MyCreditRequestsPanel extends javax.swing.JPanel {

/\*\*

\* Creates new form MyCreditRequestsPanel

\*/

public MyCreditRequestsPanel() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jTabbedPane1 = new javax.swing.JTabbedPane();

requestCreditPanel1 = new by.ny.client.ui.RequestCreditPanel();

myRequestedCreditsPanel1 = new by.ny.client.ui.MyRequestedCreditsPanel();

setRequestFocusEnabled(false);

setVerifyInputWhenFocusTarget(false);

jTabbedPane1.addComponentListener(new java.awt.event.ComponentAdapter() {

public void componentShown(java.awt.event.ComponentEvent evt) {

jTabbedPane1ComponentShown(evt);

}

});

jTabbedPane1.addTab("РћСЃС‚Р°РІРёС‚СЊ Р·Р°СЏРІРєСѓ РЅР° РєСЂРµРґРёС‚", requestCreditPanel1);

myRequestedCreditsPanel1.addComponentListener(new java.awt.event.ComponentAdapter() {

public void componentShown(java.awt.event.ComponentEvent evt) {

myRequestedCreditsPanel1ComponentShown(evt);

}

});

jTabbedPane1.addTab("РњРѕРё Р·Р°СЏРІРєРё РЅР° РєСЂРµРґРёС‚", myRequestedCreditsPanel1);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);

this.setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jTabbedPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 836, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jTabbedPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 430, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap())

);

}// </editor-fold>

private void jTabbedPane1ComponentShown(java.awt.event.ComponentEvent evt) {

}

private void myRequestedCreditsPanel1ComponentShown(java.awt.event.ComponentEvent evt) {

myRequestedCreditsPanel1.reloadTableData();

}

// Variables declaration - do not modify

private javax.swing.JTabbedPane jTabbedPane1;

private by.ny.client.ui.MyRequestedCreditsPanel myRequestedCreditsPanel1;

private by.ny.client.ui.RequestCreditPanel requestCreditPanel1;

// End of variables declaration

}

10

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package by.ny.client.ui;

/\*\*

\*

\* @author Natallia Yurush

\*/

public class MyCreditsPanel extends javax.swing.JPanel {

/\*\*

\* Creates new form MyCreditsPanel

\*/

public MyCreditsPanel() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

myCreditsTab = new javax.swing.JTabbedPane();

myActiveCreditsPanel1 = new by.ny.client.ui.MyActiveCreditsPanel();

myFinishedCreditsPanel1 = new by.ny.client.ui.MyFinishedCreditsPanel();

myCreditsTab.addTab("РњРѕРё Р°РєС‚РёРІРЅС‹Рµ РєСЂРµРґРёС‚С‹", myActiveCreditsPanel1);

myCreditsTab.addTab("РњРѕРё Р·Р°РІРµСЂС€РµРЅРЅС‹Рµ РєСЂРµРґРёС‚С‹", myFinishedCreditsPanel1);

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addComponent(myCreditsTab, javax.swing.GroupLayout.PREFERRED\_SIZE, 836, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap())

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(myCreditsTab, javax.swing.GroupLayout.PREFERRED\_SIZE, 430, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap())

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);

this.setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addGap(0, 0, Short.MAX\_VALUE)

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

);

}// </editor-fold>

// Variables declaration - do not modify

private javax.swing.JPanel jPanel1;

private by.ny.client.ui.MyActiveCreditsPanel myActiveCreditsPanel1;

private javax.swing.JTabbedPane myCreditsTab;

private by.ny.client.ui.MyFinishedCreditsPanel myFinishedCreditsPanel1;

// End of variables declaration

}

11

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package by.ny.client.ui;

import by.ny.client.ConnectionUtil;

import by.ny.client.CurrentUserUtil;

import by.ny.server.entity.Credit;

import by.ny.server.entity.CreditStatus;

import by.ny.server.entity.UserCredit;

import by.ny.server.entity.command.credit.GetCreditCommand;

import by.ny.server.entity.command.usercredit.ListUserCreditCommand;

import by.ny.server.entity.result.credit.SingleCreditResult;

import by.ny.server.entity.result.usercredit.ListUserCreditResult;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.net.Socket;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.List;

import javax.swing.event.ListSelectionEvent;

import javax.swing.event.ListSelectionListener;

import javax.swing.table.AbstractTableModel;

/\*\*

\*

\* @author Natallia Yurush

\*/

public class MyFinishedCreditsPanel extends javax.swing.JPanel {

private Integer userCreditId;

public MyFinishedCreditsPanel() {

initComponents();

reloadTableData();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jScrollPane1 = new javax.swing.JScrollPane();

activeCreditsTable = new javax.swing.JTable();

activeCreditsTable.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

{null, null, null},

{null, null, null},

{null, null, null},

{null, null, null}

},

new String [] {

"Р¦РµР»СЊ РІС‹РґР°С‡Рё", "РЎСѓРјРјР° РєСЂРµРґРёС‚Р°", "РЎС‚Р°С‚СѓСЃ РєСЂРµРґРёС‚Р°"

}

) {

Class[] types = new Class [] {

java.lang.String.class, java.lang.Double.class, java.lang.String.class

};

boolean[] canEdit = new boolean [] {

false, false, false

};

public Class getColumnClass(int columnIndex) {

return types [columnIndex];

}

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

jScrollPane1.setViewportView(activeCreditsTable);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);

this.setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 812, Short.MAX\_VALUE)

.addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 379, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(50, 50, 50))

);

}// </editor-fold>

public void reloadTableData() {

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new ListUserCreditCommand(CurrentUserUtil.getCurrentUser()));

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

ListUserCreditResult result = (ListUserCreditResult) inputStream.readObject();

final List<UserCredit> requestedUserCredits = new ArrayList();

for (UserCredit userCredit : result.getUserCredits()) {

if (userCredit.getStatus() == CreditStatus.CLOSED) {

requestedUserCredits.add(userCredit);

}

}

activeCreditsTable.setModel(new AbstractTableModel() {

private String columns[] = new String [] { "Цель выдачи", "сумма кредита", "Статус кредита" };

@Override

public int getRowCount() {

return requestedUserCredits.size();

}

@Override

public int getColumnCount() {

return columns.length;

}

@Override

public Object getValueAt(int rowIndex, int columnIndex) {

UserCredit userCredit = requestedUserCredits.get(rowIndex);

if (columnIndex == 0) {

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new GetCreditCommand(userCredit.getCreditId()));

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

SingleCreditResult result = (SingleCreditResult) inputStream.readObject();

Credit credit = result.getCredit();

return credit.getPurpose();

} catch (Exception e) {

e.printStackTrace();

}

} else if (columnIndex == 1) {

return userCredit.getIssueAmount();

} else if (columnIndex == 2) {

if(userCredit.getStatus() == CreditStatus.CLOSED)

return "погашен";

else if(userCredit.getStatus() == CreditStatus.ACTIVE)

return "активен";

}

return "";

}

// needed to show column names in JTable

public String getColumnName(int col) {

return columns[col];

}

public Class getColumnClass(int col) {

if (col == 1) {

return Double.class;

}

else {

return String.class;

}

}

});

activeCreditsTable.getSelectionModel().addListSelectionListener(

new ListSelectionListener() {

public void valueChanged(ListSelectionEvent event) {

int selectedRow = activeCreditsTable.getSelectedRow();

if (selectedRow < 0) {

return;

}

userCreditId = requestedUserCredits.get(selectedRow).getId();

}

});

} catch (Exception e) {

e.printStackTrace();

}

}

// Variables declaration - do not modify

private javax.swing.JTable activeCreditsTable;

private javax.swing.JScrollPane jScrollPane1;

// End of variables declaration

}

12

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package by.ny.client.ui;

import by.ny.client.ConnectionUtil;

import by.ny.client.CurrentUserUtil;

import by.ny.server.entity.Credit;

import by.ny.server.entity.CreditStatus;

import by.ny.server.entity.UserCredit;

import by.ny.server.entity.command.credit.GetCreditCommand;

import by.ny.server.entity.command.credit.ListCreditsCommand;

import by.ny.server.entity.command.usercredit.DeleteUserCreditCommand;

import by.ny.server.entity.command.usercredit.ListUserCreditCommand;

import by.ny.server.entity.result.credit.CreditsResult;

import by.ny.server.entity.result.credit.SingleCreditResult;

import by.ny.server.entity.result.usercredit.DeleteUserCreditResult;

import by.ny.server.entity.result.usercredit.ListUserCreditResult;

import java.awt.event.FocusAdapter;

import java.awt.event.FocusEvent;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.net.Socket;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.List;

import javax.swing.JOptionPane;

import javax.swing.event.ListSelectionEvent;

import javax.swing.event.ListSelectionListener;

import javax.swing.table.AbstractTableModel;

import javax.swing.table.DefaultTableModel;

/\*\*

\*

\* @author Natallia Yurush

\*/

public class MyRequestedCreditsPanel extends javax.swing.JPanel {

private Integer userCreditId;

public MyRequestedCreditsPanel() {

initComponents();

reloadTableData();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jScrollPane1 = new javax.swing.JScrollPane();

requestedCreditsTable = new javax.swing.JTable();

declineCreditRequestButton = new javax.swing.JButton();

setPreferredSize(new java.awt.Dimension(836, 400));

requestedCreditsTable.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

{null, null, null},

{null, null, null},

{null, null, null},

{null, null, null}

},

new String [] {

"Р¦РµР»СЊ РІС‹РґР°С‡Рё", "РЎСѓРјРјР° РєСЂРµРґРёС‚Р°", "Р”Р°С‚Р° Р·Р°СЏРІРєРё"

}

) {

Class[] types = new Class [] {

java.lang.String.class, java.lang.Double.class, java.lang.String.class

};

boolean[] canEdit = new boolean [] {

false, false, false

};

public Class getColumnClass(int columnIndex) {

return types [columnIndex];

}

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

jScrollPane1.setViewportView(requestedCreditsTable);

declineCreditRequestButton.setText("РћС‚РѕР·РІР°С‚СЊ Р·Р°СЏРІРєСѓ");

declineCreditRequestButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

declineCreditRequestButtonActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);

this.setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 816, Short.MAX\_VALUE)

.addComponent(declineCreditRequestButton, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 330, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(declineCreditRequestButton)

.addGap(58, 58, 58))

);

}// </editor-fold>

public void reloadTableData() {

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new ListUserCreditCommand(CurrentUserUtil.getCurrentUser()));

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

ListUserCreditResult result = (ListUserCreditResult) inputStream.readObject();

final List<UserCredit> requestedUserCredits = new ArrayList();

for (UserCredit userCredit : result.getUserCredits()) {

if (userCredit.getStatus() == CreditStatus.REQUESTED) {

requestedUserCredits.add(userCredit);

}

}

requestedCreditsTable.setModel(new AbstractTableModel() {

private String columns[] = new String [] { "Цель выдачи", "Сумма кредита", "Дата заявки" };

@Override

public int getRowCount() {

return requestedUserCredits.size();

}

@Override

public int getColumnCount() {

return columns.length;

}

@Override

public Object getValueAt(int rowIndex, int columnIndex) {

UserCredit userCredit = requestedUserCredits.get(rowIndex);

if (columnIndex == 0) {

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new GetCreditCommand(userCredit.getCreditId()));

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

SingleCreditResult result = (SingleCreditResult) inputStream.readObject();

Credit credit = result.getCredit();

return credit.getPurpose();

} catch (Exception e) {

e.printStackTrace();

}

} else if (columnIndex == 1) {

return userCredit.getIssueAmount();

} else if (columnIndex == 2) {

SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");

return format.format(userCredit.getIssueDate());

}

return "";

}

// needed to show column names in JTable

public String getColumnName(int col) {

return columns[col];

}

public Class getColumnClass(int col) {

if (col == 1) {

return Double.class;

}

else {

return String.class;

}

}

});

requestedCreditsTable.getSelectionModel().addListSelectionListener(

new ListSelectionListener() {

public void valueChanged(ListSelectionEvent event) {

int selectedRow = requestedCreditsTable.getSelectedRow();

if (selectedRow < 0) {

return;

}

userCreditId = requestedUserCredits.get(selectedRow).getId();

declineCreditRequestButton.setEnabled(true);

}

});

} catch (Exception e) {

e.printStackTrace();

}

}

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

if (userCreditId == null) {

declineCreditRequestButton.setEnabled(false);

return;

}

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new DeleteUserCreditCommand(userCreditId));

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

DeleteUserCreditResult result = (DeleteUserCreditResult) inputStream.readObject();

if (result.isSuccess()) {

declineCreditRequestButton.setEnabled(false);

reloadTableData();

JOptionPane.showMessageDialog(this, "Заявка успешно отозвана.");

} else {

JOptionPane.showMessageDialog(this, "Не удалось отозвать заявку.");

}

} catch (Exception e) {

e.printStackTrace();

}

}

// Variables declaration - do not modify

private javax.swing.JButton declineCreditRequestButton;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTable requestedCreditsTable;

// End of variables declaration

}

13

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package by.ny.client.ui;

import java.io.BufferedReader;

import java.io.FileInputStream;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.Arrays;

import javax.swing.JOptionPane;

/\*\*

\*

\* @author Natallia Yurush

\*/

public class PairComparisonMethodPanel extends javax.swing.JPanel {

private ArrayList<String> creditsObjectives;

private Double[] targetWeights;

public PairComparisonMethodPanel() {

initComponents();

targetWeights = new Double[] {0.0, 0.0, 0.0, 0.0};

creditsObjectives = new ArrayList<>();

readObjectivesFromFile();

if(creditsObjectives.size() != 4) {

JOptionPane.showMessageDialog(this, "В закладке \"Критерии\" должны быть заполнены 4 цели");

CalculateButton.setEnabled(false);

RecordAnotherAnswerButton.setEnabled(false);

SaveButton.setEnabled(false);

ClearButton.setEnabled(false);

ScaleSizeField.setEnabled(false);

} else {

fillCriteriaField();

}

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jScrollPane1 = new javax.swing.JScrollPane();

CriteriaField = new javax.swing.JTextArea();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

jLabel7 = new javax.swing.JLabel();

jLabel8 = new javax.swing.JLabel();

field00 = new javax.swing.JTextField();

field01 = new javax.swing.JTextField();

field02 = new javax.swing.JTextField();

field03 = new javax.swing.JTextField();

field10 = new javax.swing.JTextField();

field11 = new javax.swing.JTextField();

field12 = new javax.swing.JTextField();

field13 = new javax.swing.JTextField();

field20 = new javax.swing.JTextField();

field21 = new javax.swing.JTextField();

field22 = new javax.swing.JTextField();

field23 = new javax.swing.JTextField();

field30 = new javax.swing.JTextField();

field31 = new javax.swing.JTextField();

field32 = new javax.swing.JTextField();

field33 = new javax.swing.JTextField();

RecordAnotherAnswerButton = new javax.swing.JButton();

CalculateButton = new javax.swing.JButton();

ClearButton = new javax.swing.JButton();

jScrollPane2 = new javax.swing.JScrollPane();

ResponseField = new javax.swing.JTextArea();

SaveButton = new javax.swing.JButton();

jLabel9 = new javax.swing.JLabel();

ScaleSizeField = new javax.swing.JTextField();

CriteriaField.setColumns(20);

CriteriaField.setFont(new java.awt.Font("Arial", 0, 16)); // NOI18N

CriteriaField.setRows(5);

jScrollPane1.setViewportView(CriteriaField);

jLabel1.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

jLabel1.setText("K1");

jLabel2.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

jLabel2.setText("K2");

jLabel3.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

jLabel3.setText("K3");

jLabel4.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

jLabel4.setText("K4");

jLabel5.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

jLabel5.setText("K1");

jLabel6.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

jLabel6.setText("K2");

jLabel7.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

jLabel7.setText("K3");

jLabel8.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

jLabel8.setText("K4");

field00.setEnabled(false);

field11.setEnabled(false);

field22.setEnabled(false);

field33.setEnabled(false);

RecordAnotherAnswerButton.setText("Записать еще один ответ");

RecordAnotherAnswerButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

RecordAnotherAnswerButtonActionPerformed(evt);

}

});

CalculateButton.setText("Рассчитать ");

CalculateButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

CalculateButtonActionPerformed(evt);

}

});

ClearButton.setText("Очистить");

ClearButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

ClearButtonActionPerformed(evt);

}

});

ResponseField.setColumns(20);

ResponseField.setFont(new java.awt.Font("Arial", 0, 12)); // NOI18N

ResponseField.setRows(5);

jScrollPane2.setViewportView(ResponseField);

SaveButton.setText("Сохранить");

SaveButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

SaveButtonActionPerformed(evt);

}

});

jLabel9.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

jLabel9.setText("Размер шкалы:");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);

this.setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(37, 37, 37)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

.addComponent(jScrollPane2, javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(CalculateButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 190, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(layout.createSequentialGroup()

.addGap(28, 28, 28)

.addComponent(jLabel9)

.addGap(18, 18, 18)

.addComponent(ScaleSizeField, javax.swing.GroupLayout.PREFERRED\_SIZE, 70, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGap(27, 27, 27)

.addComponent(RecordAnotherAnswerButton)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(SaveButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 150, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(26, 26, 26)

.addComponent(ClearButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 150, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, layout.createSequentialGroup()

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(55, 55, 55)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(66, 66, 66)

.addComponent(jLabel1)

.addGap(101, 101, 101)

.addComponent(jLabel2))

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel5)

.addComponent(jLabel6, javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jLabel7, javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jLabel8, javax.swing.GroupLayout.Alignment.TRAILING))

.addGap(23, 23, 23)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(field30, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, 76, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(field20, javax.swing.GroupLayout.PREFERRED\_SIZE, 76, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(field10)

.addComponent(field00, javax.swing.GroupLayout.PREFERRED\_SIZE, 76, javax.swing.GroupLayout.PREFERRED\_SIZE))))

.addGap(41, 41, 41)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(field11, javax.swing.GroupLayout.PREFERRED\_SIZE, 76, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(field21, javax.swing.GroupLayout.PREFERRED\_SIZE, 76, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(field31, javax.swing.GroupLayout.PREFERRED\_SIZE, 76, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(field01, javax.swing.GroupLayout.PREFERRED\_SIZE, 76, javax.swing.GroupLayout.PREFERRED\_SIZE))))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 51, Short.MAX\_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(field22, javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(field12, javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(field02, javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addComponent(jLabel3)

.addGap(31, 31, 31))

.addComponent(field32, javax.swing.GroupLayout.PREFERRED\_SIZE, 76, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(42, 42, 42)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(field03)

.addComponent(field13)

.addComponent(field23)

.addComponent(field33, javax.swing.GroupLayout.PREFERRED\_SIZE, 76, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGroup(layout.createSequentialGroup()

.addGap(72, 72, 72)

.addComponent(jLabel4)))

.addGap(35, 35, 35)))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1)

.addComponent(jLabel2)

.addComponent(jLabel9)

.addComponent(ScaleSizeField, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel5)

.addComponent(field00, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(field01, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(field02, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(field03, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(42, 42, 42)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel6)

.addComponent(field10, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(field11, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(field12, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(field13, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(44, 44, 44)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel7)

.addComponent(field20, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(field21, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(field22, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(field23, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(42, 42, 42)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel8)

.addComponent(field30, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(field31, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(field32, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(field33, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addComponent(jScrollPane1)))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel3)

.addComponent(jLabel4)))

.addGap(39, 39, 39)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(RecordAnotherAnswerButton)

.addComponent(CalculateButton)

.addComponent(ClearButton)

.addComponent(SaveButton))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT\_SIZE, 99, Short.MAX\_VALUE)

.addContainerGap())

);

}// </editor-fold>

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

clearFields();

}

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

clearFields();

}

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

if(field01.getText().equals("") || field02.getText().equals("") || field03.getText().equals("") ||

field10.getText().equals("") || field12.getText().equals("") || field13.getText().equals("") ||

field20.getText().equals("") || field21.getText().equals("") || field23.getText().equals("") || field30.getText().equals("")

|| field31.getText().equals("") || field32.getText().equals("")) {

JOptionPane.showMessageDialog(this, "Должны быть заполнены все ячейки матрицы");

return;

}

double temp1 = 0.0, temp2 = 0.0, temp3 = 0.0, temp4 = 0.0;

temp1 = (Double.parseDouble(field01.getText()) + Double.parseDouble(field02.getText())

+ Double.parseDouble(field03.getText())) / Integer.parseInt(ScaleSizeField.getText()) / 12;

temp2 = (Double.parseDouble(field10.getText()) + Double.parseDouble(field12.getText())

+ Double.parseDouble(field13.getText())) / Integer.parseInt(ScaleSizeField.getText()) / 12;

temp3 = (Double.parseDouble(field20.getText()) + Double.parseDouble(field21.getText())

+ Double.parseDouble(field23.getText())) / Integer.parseInt(ScaleSizeField.getText()) / 12;

temp4 = (Double.parseDouble(field30.getText()) + Double.parseDouble(field31.getText())

+ Double.parseDouble(field32.getText())) / Integer.parseInt(ScaleSizeField.getText()) / 12;

targetWeights[0] += temp1;

targetWeights[1] += temp2;

targetWeights[2] += temp3;

targetWeights[3] += temp4;

JOptionPane.showMessageDialog(this, "Матрица частот эксперта успешно сформирована");

//clearFields();

}

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

//сделать проверку на хоть 1 эксперта

String str = "Искомые веса целей:" + System.getProperty("line.separator") + "К1 = " + targetWeights[0].toString()

+ System.getProperty("line.separator") + "К2 = " + targetWeights[1].toString()

+ System.getProperty("line.separator") + "К3 = " + targetWeights[2].toString()

+ System.getProperty("line.separator") + "К4 = " + targetWeights[3].toString()

+ System.getProperty("line.separator") + "Наиболее весомая цель: " + creditsObjectives.get(max());

ResponseField.setText(str);

clearFields();

}

private void clearFields(){

field00.setText("");

field01.setText("");

field02.setText("");

field03.setText("");

field10.setText("");

field11.setText("");

field12.setText("");

field13.setText("");

field20.setText("");

field21.setText("");

field22.setText("");

field23.setText("");

field30.setText("");

field31.setText("");

field32.setText("");

field33.setText("");

}

private void readObjectivesFromFile() {

try (BufferedReader reader = new BufferedReader(new InputStreamReader( new FileInputStream("output.txt") , "windows-1251"))) {

String line;

while ((line = reader.readLine()) != null) {

creditsObjectives.add(line);

}

}

catch (Exception e) {

e.printStackTrace();

}

}

private void fillCriteriaField() {

//System.getProperty("line.separator");

CriteriaField.setText("Критерии:" + System.getProperty("line.separator") + "К1: " + creditsObjectives.get(0) + System.getProperty("line.separator")

+ "К2: " + creditsObjectives.get(1) + System.getProperty("line.separator") + "К3: " + creditsObjectives.get(2)

+ System.getProperty("line.separator") + "К4: " + creditsObjectives.get(3));

}

private Integer max() {

Integer maxIndex = 0;

Double currentNumber;

for (int i = 1; i < targetWeights.length; i++) {

currentNumber = targetWeights[i];

if ((currentNumber > targetWeights[maxIndex])){

maxIndex = i;

}

}

return maxIndex;

}

// Variables declaration - do not modify

private javax.swing.JButton CalculateButton;

private javax.swing.JButton ClearButton;

private javax.swing.JTextArea CriteriaField;

private javax.swing.JButton RecordAnotherAnswerButton;

private javax.swing.JTextArea ResponseField;

private javax.swing.JButton SaveButton;

private javax.swing.JTextField ScaleSizeField;

private javax.swing.JTextField field00;

private javax.swing.JTextField field01;

private javax.swing.JTextField field02;

private javax.swing.JTextField field03;

private javax.swing.JTextField field10;

private javax.swing.JTextField field11;

private javax.swing.JTextField field12;

private javax.swing.JTextField field13;

private javax.swing.JTextField field20;

private javax.swing.JTextField field21;

private javax.swing.JTextField field22;

private javax.swing.JTextField field23;

private javax.swing.JTextField field30;

private javax.swing.JTextField field31;

private javax.swing.JTextField field32;

private javax.swing.JTextField field33;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JLabel jLabel8;

private javax.swing.JLabel jLabel9;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane2;

// End of variables declaration

}

14

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package by.ny.client.ui;

import by.ny.client.ConnectionUtil;

import by.ny.client.CurrentUserUtil;

import by.ny.server.entity.command.usercredit.RequestUserCreditCommand;

import by.ny.server.entity.result.usercredit.RequestUserCreditResult;

import by.ny.server.entity.Credit;

import by.ny.server.entity.command.credit.ListCreditsCommand;

import by.ny.server.entity.result.credit.CreditsResult;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.net.Socket;

import javax.swing.ComboBoxModel;

import javax.swing.DefaultComboBoxModel;

import javax.swing.JOptionPane;

import javax.swing.SpinnerNumberModel;

/\*\*

\*

\* @author Natallia Yurush

\*/

public class RequestCreditPanel extends javax.swing.JPanel {

/\*\*

\* Creates new form RequestCreditPanel

\*/

public RequestCreditPanel() {

initComponents();

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new ListCreditsCommand());

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

CreditsResult result = (CreditsResult) inputStream.readObject();

creditsComboBox.setModel(new DefaultComboBoxModel<>(result.getCredits().toArray(new Credit[] {})));

creditsComboBox.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

Credit selectedCredit = (Credit) creditsComboBox.getSelectedItem();

((SpinnerNumberModel) requestedAmountSpinner.getModel()).setMaximum(selectedCredit.getMaxAmount());

}

});

} catch (Exception e) {

e.printStackTrace();

}

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

requestCreditButton = new javax.swing.JButton();

jLabel1 = new javax.swing.JLabel();

creditsComboBox = new javax.swing.JComboBox<>();

jLabel2 = new javax.swing.JLabel();

requestedAmountSpinner = new javax.swing.JSpinner();

requestCreditButton.setText("РћС„РѕСЂРјРёС‚СЊ Р·Р°СЏРІРєСѓ");

requestCreditButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

requestCreditButtonActionPerformed(evt);

}

});

jLabel1.setText("РљСЂРµРґРёС‚:");

jLabel2.setText("Р—Р°РїСЂР°С€РёРІР°РµРјР°СЏ СЃСѓРјРјР°:");

requestedAmountSpinner.setModel(new javax.swing.SpinnerNumberModel(1.0d, 1.0d, null, 0.1d));

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);

this.setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(requestCreditButton, javax.swing.GroupLayout.DEFAULT\_SIZE, 812, Short.MAX\_VALUE)

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel1)

.addGap(111, 111, 111)

.addComponent(creditsComboBox, 0, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel2)

.addGap(18, 18, 18)

.addComponent(requestedAmountSpinner)))

.addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1)

.addComponent(creditsComboBox, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel2)

.addComponent(requestedAmountSpinner, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 327, Short.MAX\_VALUE)

.addComponent(requestCreditButton)

.addGap(17, 17, 17))

);

}// </editor-fold>

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

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new RequestUserCreditCommand((Credit) creditsComboBox.getSelectedItem(), CurrentUserUtil.getCurrentUser(),

(Double) requestedAmountSpinner.getValue()));

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

RequestUserCreditResult result = (RequestUserCreditResult) inputStream.readObject();

if (result.isSuccess()) {

JOptionPane.showMessageDialog(this, "Заявка на кредит успешно оформлена.");

} else {

JOptionPane.showMessageDialog(this, "Не удалось оформить заявку на кредит.");

}

} catch (Exception e) {

e.printStackTrace();

}

}

// Variables declaration - do not modify

private javax.swing.JComboBox<Credit> creditsComboBox;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JButton requestCreditButton;

private javax.swing.JSpinner requestedAmountSpinner;

// End of variables declaration

}

Предпоследнее

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package by.ny.client.ui;

import by.ny.client.ConnectionUtil;

import by.ny.client.CurrentUserUtil;

import by.ny.server.entity.Credit;

import by.ny.server.entity.CreditStatus;

import by.ny.server.entity.User;

import by.ny.server.entity.UserRole;

import by.ny.server.entity.UserCredit;

import by.ny.server.entity.command.credit.GetCreditCommand;

import by.ny.server.entity.command.user.ListUsersCommand;

import by.ny.server.entity.command.usercredit.DeleteUserCreditCommand;

import by.ny.server.entity.command.usercredit.ListUserCreditCommand;

import by.ny.server.entity.result.credit.SingleCreditResult;

import by.ny.server.entity.result.user.GetSingleUserResult;

import by.ny.server.entity.result.user.ListUsersResult;

import by.ny.server.entity.command.user.GetUserByIdCommand;

import by.ny.server.entity.command.usercredit.GetUserCreditByIdCommand;

import by.ny.server.entity.command.usercredit.SaveUserCreditCommand;

import by.ny.server.entity.result.usercredit.DeleteUserCreditResult;

import by.ny.server.entity.result.usercredit.ListUserCreditResult;

import by.ny.server.entity.result.usercredit.RequestUserCreditResult;

import by.ny.server.entity.result.usercredit.SingleUserCreditResult;

import by.ny.server.entity.result.usercredit.UpdateUserCreditStatusResult;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.net.Socket;

import java.text.SimpleDateFormat;

import java.time.LocalDate;

import java.util.ArrayList;

import java.util.List;

import javax.swing.JOptionPane;

import javax.swing.event.ListSelectionEvent;

import javax.swing.event.ListSelectionListener;

import javax.swing.table.AbstractTableModel;

/\*\*

\*

\* @author Natallia Yurush

\*/

public class RequestedCreditsPanel extends javax.swing.JPanel {

private Integer userCreditId;

/\*\*

\* Creates new form RequestedCreditsPanel

\*/

public RequestedCreditsPanel() {

initComponents();

reloadTableData();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jScrollPane1 = new javax.swing.JScrollPane();

requestedCreditsTable = new javax.swing.JTable();

confirmRequestedCreditButton = new javax.swing.JButton();

rejectRequestedCreditButton = new javax.swing.JButton();

requestedCreditsTable.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

{null, null, null, null},

{null, null, null, null},

{null, null, null, null},

{null, null, null, null}

},

new String [] {

"Р¤Р�Рћ", "Р¦РµР»СЊ РІС‹РґР°С‡Рё РєСЂРµРґРёС‚Р°", "РЎСѓРјРјР° РєСЂРµРґРёС‚Р°", "Р”Р°С‚Р° Р·Р°СЏРІРєРё"

}

) {

Class[] types = new Class [] {

java.lang.String.class, java.lang.String.class, java.lang.Double.class, java.lang.String.class

};

boolean[] canEdit = new boolean [] {

false, false, false, false

};

public Class getColumnClass(int columnIndex) {

return types [columnIndex];

}

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

jScrollPane1.setViewportView(requestedCreditsTable);

confirmRequestedCreditButton.setText("РџРѕРґС‚РІРµСЂРґРёС‚СЊ Р·Р°СЏРІРєСѓ");

confirmRequestedCreditButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

confirmRequestedCreditButtonActionPerformed(evt);

}

});

rejectRequestedCreditButton.setText("РћС‚РєР»РѕРЅРёС‚СЊ Р·Р°СЏРІРєСѓ");

rejectRequestedCreditButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

rejectRequestedCreditButtonActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(rejectRequestedCreditButton, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(confirmRequestedCreditButton, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 841, Short.MAX\_VALUE))

.addContainerGap())

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 362, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(confirmRequestedCreditButton)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(rejectRequestedCreditButton)

.addContainerGap(21, Short.MAX\_VALUE))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);

this.setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 865, Short.MAX\_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(0, 0, Short.MAX\_VALUE)

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, Short.MAX\_VALUE)))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 472, Short.MAX\_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(0, 0, Short.MAX\_VALUE)

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, Short.MAX\_VALUE)))

);

}// </editor-fold>

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

if (userCreditId == null) {

confirmRequestedCreditButton.setEnabled(false);

rejectRequestedCreditButton.setEnabled(false);

return;

}

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new DeleteUserCreditCommand(userCreditId));

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

DeleteUserCreditResult result = (DeleteUserCreditResult) inputStream.readObject();

if (result.isSuccess()) {

rejectRequestedCreditButton.setEnabled(false);

reloadTableData();

JOptionPane.showMessageDialog(this, "Заявка успешно отклонена.");

} else {

JOptionPane.showMessageDialog(this, "Не удалось отклонить заявку.");

}

} catch (Exception e) {

e.printStackTrace();

}

}

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

if (userCreditId == null) {

confirmRequestedCreditButton.setEnabled(false);

rejectRequestedCreditButton.setEnabled(false);

return;

}

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new GetUserCreditByIdCommand(userCreditId));

outputStream.flush();

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

SingleUserCreditResult singleUserCreditResult = (SingleUserCreditResult) inputStream.readObject();

UserCredit userCredit = singleUserCreditResult.getUser();

//CurrentUserUtil.getCurrentUser()

socket = ConnectionUtil.getSocket();

outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new SaveUserCreditCommand(userCredit));

inputStream = new ObjectInputStream(socket.getInputStream());

UpdateUserCreditStatusResult result = (UpdateUserCreditStatusResult) inputStream.readObject();

if (result.isSuccess()) {

reloadTableData();

JOptionPane.showMessageDialog(this, "Подтверждение заявки прошло успешно.");

} else {

JOptionPane.showMessageDialog(this, "Не удалось подтвердить заявку.");

}

} catch (Exception e) {

e.printStackTrace();

}

}

public void reloadTableData() {

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new ListUsersCommand());

outputStream.flush();

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

ListUsersResult userListResult = (ListUsersResult) inputStream.readObject();

//inputStream.close();

//outputStream.close();

final List<UserCredit> requestedUserCredits = new ArrayList();

for(User user : userListResult.getUsers()){

if( user.getRole() == UserRole.ADMIN){

continue;

}

socket = ConnectionUtil.getSocket();

outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new ListUserCreditCommand(user));

inputStream = new ObjectInputStream(socket.getInputStream());

ListUserCreditResult result = (ListUserCreditResult) inputStream.readObject();

outputStream.flush();

for (UserCredit userCredit : result.getUserCredits()) {

if (userCredit.getStatus() == CreditStatus.REQUESTED) {

requestedUserCredits.add(userCredit);

}

}

}

socket.close();

requestedCreditsTable.setModel(new AbstractTableModel() {

private String columns[] = new String [] { "ФИО", "Цели выдачи кредита", "Сумма кредита", "Дата заявки" };

@Override

public int getRowCount() {

return requestedUserCredits.size();

}

@Override

public int getColumnCount() {

return columns.length;

}

@Override

public Object getValueAt(int rowIndex, int columnIndex) {

UserCredit userCredit = requestedUserCredits.get(rowIndex);

if (columnIndex == 0) {

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new GetUserByIdCommand(userCredit.getUserId()));

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

GetSingleUserResult result = (GetSingleUserResult) inputStream.readObject();

User user = result.getUser();

socket.close();

String string = user.getLastName() + " " + user.getFirstName() + " " + user.getMiddleName();

return string;

} catch (Exception e) {

e.printStackTrace();

}

} else if(columnIndex == 1) {

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new GetCreditCommand(userCredit.getCreditId()));

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

SingleCreditResult result = (SingleCreditResult) inputStream.readObject();

Credit credit = result.getCredit();

socket.close();

return credit.getPurpose();

} catch (Exception e) {

e.printStackTrace();

}

} else if (columnIndex == 2) {

return userCredit.getIssueAmount();

} else if (columnIndex == 3) {

SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");

return format.format(userCredit.getIssueDate());

}

return "";

}

// needed to show column names in JTable

public String getColumnName(int col) {

return columns[col];

}

public Class getColumnClass(int col) {

if (col == 2) {

return Double.class;

}

else {

return String.class;

}

}

});

requestedCreditsTable.getSelectionModel().addListSelectionListener(

new ListSelectionListener() {

public void valueChanged(ListSelectionEvent event) {

int selectedRow = requestedCreditsTable.getSelectedRow();

if (selectedRow < 0) {

return;

}

userCreditId = requestedUserCredits.get(selectedRow).getId();

rejectRequestedCreditButton.setEnabled(true);

confirmRequestedCreditButton.setEnabled(true);

}

});

} catch (Exception e) {

e.printStackTrace();

}

}

// Variables declaration - do not modify

private javax.swing.JButton confirmRequestedCreditButton;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JButton rejectRequestedCreditButton;

private javax.swing.JTable requestedCreditsTable;

// End of variables declaration

}

Послденее

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package by.ny.client.ui;

import by.ny.client.ConnectionUtil;

import by.ny.server.entity.User;

import by.ny.server.entity.UserRole;

import by.ny.server.entity.command.user.DeleteUserCommand;

import by.ny.server.entity.command.user.GetUserByIdCommand;

import by.ny.server.entity.command.user.ListUsersCommand;

import by.ny.server.entity.command.user.SaveUserCommand;

import by.ny.server.entity.result.user.GetSingleUserResult;

import by.ny.server.entity.result.user.ListUsersResult;

import by.ny.server.entity.result.user.UpdateUserStatusResult;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.net.Socket;

import javax.swing.JOptionPane;

import javax.swing.event.ListSelectionEvent;

import javax.swing.event.ListSelectionListener;

import javax.swing.table.DefaultTableModel;

import javax.swing.table.TableModel;

/\*\*

\*

\* @author Natallia Yurush

\*/

public class UsersPanel extends javax.swing.JPanel {

/\*\*

\* Creates new form UsersPanel

\*/

public UsersPanel() {

initComponents();

reloadUsersTableData();

usersTable.getSelectionModel().addListSelectionListener(

new ListSelectionListener() {

public void valueChanged(ListSelectionEvent event) {

int selectedRow = usersTable.getSelectedRow();

if (selectedRow < 0) {

return;

}

Integer userId = (Integer) usersTable.getValueAt(selectedRow, 0);

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new GetUserByIdCommand(userId));

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

GetSingleUserResult result = (GetSingleUserResult) inputStream.readObject();

User user = result.getUser();

idField.setText(String.valueOf(user.getId()));

firstNameField.setText(user.getFirstName());

lastNameField.setText(user.getLastName());

middleNameField.setText(user.getMiddleName());

passportNumberField.setText(user.getPassportNumber());

passwordField.setText(user.getPassword());

switch(user.getRole()) {

case ADMIN:

roleField.setSelectedIndex(0);

break;

case CLIENT:

roleField.setSelectedIndex(1);

break;

}

roleField.setEnabled(false);

deleteButton.setEnabled(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jScrollPane1 = new javax.swing.JScrollPane();

usersTable = new javax.swing.JTable();

jLabel1 = new javax.swing.JLabel();

idField = new javax.swing.JTextField();

jLabel2 = new javax.swing.JLabel();

firstNameField = new javax.swing.JTextField();

lastNameField = new javax.swing.JTextField();

jLabel3 = new javax.swing.JLabel();

middleNameField = new javax.swing.JTextField();

jLabel4 = new javax.swing.JLabel();

passportNumberField = new javax.swing.JTextField();

jLabel5 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

roleField = new javax.swing.JComboBox<>();

jLabel7 = new javax.swing.JLabel();

saveButton = new javax.swing.JButton();

clearButton = new javax.swing.JButton();

passwordField = new javax.swing.JPasswordField();

deleteButton = new javax.swing.JButton();

usersTable.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

{null, null, null, null, null, null},

{null, null, null, null, null, null},

{null, null, null, null, null, null},

{null, null, null, null, null, null}

},

new String [] {

"Идентификатор", "Фамилия", "Имя", "Отчество", "Номер паспорта", "Роль"

}

) {

Class[] types = new Class [] {

java.lang.Integer.class, java.lang.String.class, java.lang.String.class, java.lang.String.class, java.lang.String.class, java.lang.String.class

};

boolean[] canEdit = new boolean [] {

false, false, false, false, false, false

};

public Class getColumnClass(int columnIndex) {

return types [columnIndex];

}

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

jScrollPane1.setViewportView(usersTable);

jLabel1.setText("Идентификатор:");

idField.setEnabled(false);

jLabel2.setText("Имя:");

jLabel3.setText("Фамилия:");

jLabel4.setText("Отчество:");

passportNumberField.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

passportNumberFieldActionPerformed(evt);

}

});

jLabel5.setText("Номер паспорта:");

jLabel6.setText("Пароль:");

roleField.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Администратор", "Клиент" }));

roleField.setSelectedIndex(1);

jLabel7.setText("Роль:");

saveButton.setText("Сохранить");

saveButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

saveButtonActionPerformed(evt);

}

});

clearButton.setText("Очистить");

clearButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

clearButtonActionPerformed(evt);

}

});

deleteButton.setText("Удалить");

deleteButton.setEnabled(false);

deleteButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

deleteButtonActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jScrollPane1)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jLabel4)

.addGap(55, 55, 55)

.addComponent(middleNameField))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addComponent(jLabel3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(lastNameField, javax.swing.GroupLayout.PREFERRED\_SIZE, 316, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jLabel1)

.addGap(18, 18, 18)

.addComponent(idField, javax.swing.GroupLayout.PREFERRED\_SIZE, 316, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jLabel2)

.addGap(87, 87, 87)

.addComponent(firstNameField, javax.swing.GroupLayout.PREFERRED\_SIZE, 316, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGap(0, 0, Short.MAX\_VALUE)))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel5)

.addComponent(jLabel6)

.addComponent(jLabel7))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 21, Short.MAX\_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(passportNumberField, javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(roleField, javax.swing.GroupLayout.Alignment.TRAILING, 0, 260, Short.MAX\_VALUE)

.addComponent(passwordField)))

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(saveButton, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(clearButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 120, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(deleteButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 120, javax.swing.GroupLayout.PREFERRED\_SIZE)))))

.addContainerGap())

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 298, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1)

.addComponent(idField, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(passportNumberField, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel5))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel6)

.addComponent(passwordField, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(lastNameField, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel3))

.addGap(14, 14, 14)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(roleField, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(middleNameField, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel4)

.addComponent(saveButton)

.addComponent(clearButton)

.addComponent(deleteButton)))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel7)

.addComponent(firstNameField, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel2)))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);

this.setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 854, Short.MAX\_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(0, 2, Short.MAX\_VALUE)

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 2, Short.MAX\_VALUE)))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 472, Short.MAX\_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(0, 0, Short.MAX\_VALUE)

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, Short.MAX\_VALUE)))

);

}// </editor-fold>

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

// TODO add your handling code here:

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

Integer id = "".equals(idField.getText()) ? null : Integer.valueOf(idField.getText());

UserRole role = UserRole.CLIENT;

switch (roleField.getSelectedIndex()) {

case 1:

role = UserRole.CLIENT;

break;

case 2:

role = UserRole.ADMIN;

break;

}

outputStream.writeObject(new SaveUserCommand(new User(id, firstNameField.getText(), lastNameField.getText(),

middleNameField.getText(), passportNumberField.getText(), role, String.valueOf(passwordField.getPassword()))));

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

UpdateUserStatusResult result = (UpdateUserStatusResult) inputStream.readObject();

if (result.isSuccess()) {

clearUserFields();

reloadUsersTableData();

JOptionPane.showMessageDialog(this, "Пользователь успешно сохранен.");

} else {

JOptionPane.showMessageDialog(this, "Не удалось сохранить пользователя.");

}

} catch (Exception e) {

e.printStackTrace();

}

}

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

// TODO add your handling code here:

clearUserFields();

}

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

// TODO add your handling code here:

Integer userId = Integer.valueOf(idField.getText());

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new DeleteUserCommand(userId));

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

UpdateUserStatusResult result = (UpdateUserStatusResult) inputStream.readObject();

if (result.isSuccess()) {

clearUserFields();

reloadUsersTableData();

JOptionPane.showMessageDialog(this, "Пользователь успешно удален.");

} else {

JOptionPane.showMessageDialog(this, "Не удалось удалить пользователя.");

}

} catch(Exception e) {

e.printStackTrace();

}

}

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

// TODO add your handling code here:

}

private void clearUserFields() {

idField.setText("");

firstNameField.setText("");

lastNameField.setText("");

middleNameField.setText("");

passportNumberField.setText("");

passwordField.setText("");

roleField.setSelectedIndex(1);

roleField.setEnabled(true);

try {

usersTable.clearSelection();

} catch (Exception e) {

}

deleteButton.setEnabled(false);

}

private void reloadUsersTableData() {

try {

Socket socket = ConnectionUtil.getSocket();

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.writeObject(new ListUsersCommand());

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

ListUsersResult result = (ListUsersResult) inputStream.readObject();

int rowsCount = ((DefaultTableModel) usersTable.getModel()).getRowCount();

for (int i = 0; i < rowsCount; i ++) {

((DefaultTableModel) usersTable.getModel()).removeRow(0);

}

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

User user = result.getUsers().get(i);

Object[] row = new Object[6];

row[0] = user.getId();

row[1] = user.getFirstName();

row[2] = user.getLastName();

row[3] = user.getMiddleName();

row[4] = user.getPassportNumber();

String role = "";

switch (user.getRole()) {

case ADMIN:

role = "Администратор";

break;

case CLIENT:

role = "Клиент";

break;

}

row[5] = role;

((DefaultTableModel) usersTable.getModel()).addRow(row);

}

} catch (Exception e) {

e.printStackTrace();

}

}

// Variables declaration - do not modify

private javax.swing.JButton clearButton;

private javax.swing.JButton deleteButton;

private javax.swing.JTextField firstNameField;

private javax.swing.JTextField idField;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTextField lastNameField;

private javax.swing.JTextField middleNameField;

private javax.swing.JTextField passportNumberField;

private javax.swing.JPasswordField passwordField;

private javax.swing.JComboBox<String> roleField;

private javax.swing.JButton saveButton;

private javax.swing.JTable usersTable;

// End of variables declaration

}