import m6.ConnectionManager;

import m6.UserLoginInfo;

import m6.components.StyledButton;

import m6.components.StyledLabel;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

/\*\*

\* Home page for employees. Based on whether the employee is a manager or not, it'll automatically

\* show the relevant actions which he can perform.

\*/

public class CustomerUI extends UserBaseUI {

private UserLoginInfo userLoginInfo;

private StyledLabel balanceLabel, balanceInfoLabel;

private StyledButton viewTransactionButton, myInfoBtn;

public CustomerUI(UserLoginInfo userLoginInfo) {

super(userLoginInfo);

this.userLoginInfo = userLoginInfo;

setPageTitle("Customer | Home");

initUI();

bind();

readFromDb();

}

private void initUI() {

final int x = 190;

balanceLabel = new StyledLabel("Balance: ");

balanceLabel.setFont(new Font("Calibri", Font.BOLD, 24));

balanceLabel.setBounds(x, 200, 120, 30);

balanceInfoLabel = new StyledLabel();

balanceInfoLabel.setFont(new Font("Calibri", Font.BOLD, 24));

balanceInfoLabel.setBounds(x + 140, 200, 120, 30);

// Accounts

viewTransactionButton = new StyledButton("View Transactions");

viewTransactionButton.setBounds(x, 250, 400, 40);

myInfoBtn = new StyledButton("My Information");

myInfoBtn.setBounds(x, 300, 400, 40);

mainPanel.add(viewTransactionButton);

mainPanel.add(balanceLabel);

mainPanel.add(balanceInfoLabel);

mainPanel.add(myInfoBtn);

}

private void bind() {

viewTransactionButton.addActionListener(this);

myInfoBtn.addActionListener(this);

}

public void actionPerformed(ActionEvent e) {

super.actionPerformed(e);

Object src = e.getSource();

if (src == viewTransactionButton) {

new TransactionViewerUI(userLoginInfo, userLoginInfo.username).setVisible(true);

setVisible(false);

dispose();

} else if (src == myInfoBtn) {

new LoginEditorUI(userLoginInfo, userLoginInfo.username).setVisible(true);

setVisible(false);

dispose();

}

}

private void readFromDb() {

try {

Connection conn = ConnectionManager.getInstance().getConnection();

PreparedStatement ps = conn.prepareStatement(

"SELECT account.balance FROM account, customer WHERE customer.username=? AND account.accountNumber=customer.accountNumber"

);

ps.setString(1, userLoginInfo.username);

ResultSet rs = ps.executeQuery();

if (rs.next()) {

double balance = rs.getDouble("account.balance");

balanceInfoLabel.setText(String.valueOf(balance));

}

} catch (SQLException e) {

e.printStackTrace();

JOptionPane.showMessageDialog(this, "Error! Failed to fetch data.");

}

}

}