package m6.ui;

import m6.ConnectionManager;

import m6.UserLoginInfo;

import m6.components.StyledButton;

import m6.components.StyledLabel;

import m6.components.StyledPasswordField;

import m6.components.StyledTextField;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class LoginUI extends JFrame implements ActionListener {

private JPanel panelLeft, panelRight;

private StyledLabel introLabel, usernameLabel, passwordLabel;

private StyledButton loginButton;

private StyledTextField usernameTf;

private StyledPasswordField passwordTf;

public LoginUI() {

super("Bank Management System | Login");

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

setResizable(false);

setSize(800, 500);

initUI();

bind();

setLocationRelativeTo(null);

}

private void initUI() {

panelLeft = new JPanel();

panelLeft.setBackground(new Color(0x2dce98));

panelLeft.setBounds(0, 0, 400, 500);

panelLeft.setLayout(null);

panelRight = new JPanel();

panelRight.setBackground(Color.white);

panelRight.setBounds(400, 0, 400, 500);

panelRight.setLayout(null);

introLabel = new StyledLabel("<html><body><strong>Bank<br>Management<br>System</body></html>");

introLabel.setForeground(Color.white);

introLabel.setFont(new Font("Calibri", Font.PLAIN, 38));

introLabel.setBounds(60, 20, 350, 400);

int x = 480;

usernameLabel = new StyledLabel("Username: ");

usernameLabel.setBounds(x - 25, 180, 200, 30);

usernameTf = new StyledTextField();

usernameTf.setBounds(x + 50, 180, 200, 30);

passwordLabel = new StyledLabel("Password: ");

passwordLabel.setBounds(x - 22, 220, 200, 30);

passwordTf = new StyledPasswordField();

passwordTf.setBounds(x + 50, 220, 200, 30);

loginButton = new StyledButton("Login");

loginButton.setBounds(x + 50, 260, 200, 40);

panelLeft.add(introLabel);

panelRight.add(usernameLabel);

panelRight.add(passwordLabel);

panelRight.add(usernameTf);

panelRight.add(passwordTf);

panelRight.add(loginButton);

add(panelLeft);

add(panelRight);

}

private void bind() {

loginButton.addActionListener(this);

passwordTf.addActionListener(this);

}

public void actionPerformed(ActionEvent e) {

Object src = e.getSource();

if (src == loginButton) {

verifyLogin();

} else if (src == passwordTf) {

verifyLogin();

}

}

private void verifyLogin() {

String username = usernameTf.getText();

String password = new String(passwordTf.getPassword());

Connection connection = ConnectionManager.getInstance()

.getConnection();

try {

PreparedStatement statement = connection

.prepareStatement("SELECT \* FROM login WHERE username=? AND password=?");

statement.setString(1, username);

statement.setString(2, password);

ResultSet rs = statement.executeQuery();

if (rs.next()) {

String status = rs.getString("status");

UserLoginInfo info = new UserLoginInfo();

info.name = rs.getString("name");

info.username = rs.getString("username");

info.status = status;

if (status.equals("employee")) {

new EmployeeUI(info).setVisible(true);

} else if (status.equals("customer")) {

new CustomerUI(info).setVisible(true);

}

setVisible(false);

dispose();

} else {

JOptionPane.showMessageDialog(this,

"Invalid username or password. Please try again.");

// clear the text fields

usernameTf.setText("");

passwordTf.setText("");

}

} catch (SQLException e) {

e.printStackTrace();

JOptionPane.showMessageDialog(this,

"Error! Failed to query database.");

}

}

}