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.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class LoginEditorUI extends UserBaseUI {

private UserLoginInfo userLoginInfo;

private String username;

private StyledButton saveBtn, deleteBtn;

private StyledLabel usernameLabel, passwordLabel, statusLabel, nameLabel;

private StyledTextField usernameTf, nameTf;

private StyledPasswordField passwordTf;

private JComboBox<String> statusCmb;

public LoginEditorUI(UserLoginInfo userLoginInfo, String username) {

super(userLoginInfo);

this.userLoginInfo = userLoginInfo;

this.username = username;

if (username != null) {

setPageTitle("Edit user account");

} else {

setPageTitle("Add new user account");

}

backButton.setVisible(true);

backButton.addActionListener(this);

initUI();

bind();

if (username != null) {

readFromDb();

}

}

private void initUI() {

int x = 200;

int y = 140;

usernameLabel = new StyledLabel("Username: ");

usernameLabel.setBounds(x, y, 100, 30);

usernameTf = new StyledTextField();

usernameTf.setBounds(x + 110, y, 200, 30);

if (username != null) {

usernameTf.setEditable(false);

}

passwordLabel = new StyledLabel("Password: ");

passwordLabel.setBounds(x, y + 40, 100, 30);

passwordTf = new StyledPasswordField();

passwordTf.setBounds(x + 110, y + 40, 200, 30);

statusLabel = new StyledLabel("Status: ");

statusLabel.setBounds(x, y + 40 \* 2, 100, 30);

statusCmb = new JComboBox<>(new String[]{"employee", "customer"});

statusCmb.setBounds(x + 110, y + 40 \* 2, 200, 30);

statusCmb.setEnabled(false);

nameLabel = new StyledLabel("Name: ");

nameLabel.setBounds(x, y + 40 \* 3, 100, 30);

nameTf = new StyledTextField();

nameTf.setBounds(x + 110, y + 40 \* 3, 200, 30);

saveBtn = new StyledButton("Save");

saveBtn.setBounds(x + 110, y + 40 \* 4, 200, 35);

deleteBtn = new StyledButton("Delete");

deleteBtn.setBackground(new Color(0xF35E5F));

deleteBtn.setBounds(x + 110, y + 40 \* 5, 200, 35);

if (username == null || username.isEmpty()) {

deleteBtn.setVisible(false);

}

mainPanel.add(usernameLabel);

mainPanel.add(usernameTf);

mainPanel.add(passwordLabel);

mainPanel.add(passwordTf);

mainPanel.add(statusLabel);

mainPanel.add(statusCmb);

mainPanel.add(nameLabel);

mainPanel.add(nameTf);

mainPanel.add(saveBtn);

mainPanel.add(deleteBtn);

}

private void bind() {

saveBtn.addActionListener(this);

deleteBtn.addActionListener(this);

}

private void readFromDb() {

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

try {

PreparedStatement ps = conn.prepareStatement(

"SELECT username, password, status, name FROM login WHERE username=?"

);

ps.setString(1, username);

ResultSet rs = ps.executeQuery();

while (rs.next()) {

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

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

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

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

usernameTf.setText(username);

passwordTf.setText(password);

statusCmb.setSelectedItem(status);

nameTf.setText(name);

}

} catch (SQLException e) {

e.printStackTrace();

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

new EmployeeUI(userLoginInfo).setVisible(true);

setVisible(false);

dispose();

}

}

public void actionPerformed(ActionEvent e) {

super.actionPerformed(e);

Object src = e.getSource();

if (src == backButton) {

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

new EmployeeUI(userLoginInfo).setVisible(true);

} else {

new CustomerUI(userLoginInfo).setVisible(true);

}

setVisible(false);

dispose();

} else if (src == saveBtn) {

save();

} else if (src == deleteBtn) {

delete();

}

}

private void save() {

// depending on whether the user is in edit mode, appropriate query will be used

String insertQuery = "INSERT INTO login VALUES(?, ?, ?, ?)";

String updateQuery = "UPDATE login SET username=?, password=?, status=?, name=? WHERE username=?";

String updateCustomerQuery = "UPDATE customer SET username=? WHERE username=?";

String updateEmployeeQuery = "UPDATE employee SET username=? WHERE username=?";

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

PreparedStatement ps, psCustomer = null, psEmployee = null;

try {

if (username == null || username.trim().isEmpty()) {

ps = conn.prepareStatement(insertQuery);

ps.setString(1, usernameTf.getText());

ps.setString(2, new String(passwordTf.getPassword()));

ps.setString(3, statusCmb.getSelectedItem().toString());

ps.setString(4, nameTf.getText());

} else {

ps = conn.prepareStatement(updateQuery);

psCustomer = conn.prepareStatement(updateCustomerQuery);

psEmployee = conn.prepareStatement(updateEmployeeQuery);

ps.setString(1, usernameTf.getText());

ps.setString(2, new String(passwordTf.getPassword()));

ps.setString(3, statusCmb.getSelectedItem().toString());

ps.setString(4, nameTf.getText());

ps.setString(5, username);

psCustomer.setString(1, usernameTf.getText());

psCustomer.setString(2, username);

psEmployee.setString(1, usernameTf.getText());

psEmployee.setString(2, username);

}

System.out.println(ps);

ps.execute();

if (psEmployee != null) {

System.out.println(psCustomer);

psEmployee.executeUpdate();

System.out.println(psEmployee);

psCustomer.executeUpdate();

}

JOptionPane.showMessageDialog(this, "Success!");

new EmployeeUI(userLoginInfo).setVisible(true);

setVisible(false);

dispose();

} catch (Exception e) {

e.printStackTrace();

JOptionPane.showMessageDialog(this, "Error! Failed to add/update customer.");

}

}

private void delete() {

// depending on whether the user is in edit mode, appropriate query will be used

String deleteQuery = "DELETE FROM login WHERE username=?";

String deleteCustomerQuery = "DELETE FROM customer WHERE username=?";

String deleteEmployeeQuery = "DELETE FROM employee WHERE username=?";

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

PreparedStatement ps, psCustomer, psEmployee;

try {

ps = conn.prepareStatement(deleteQuery);

psCustomer = conn.prepareStatement(deleteCustomerQuery);

psEmployee = conn.prepareStatement(deleteEmployeeQuery);

ps.setString(1, username);

psCustomer.setString(1, username);

psEmployee.setString(1, username);

System.out.println(ps);

int count = ps.executeUpdate();

psCustomer.executeUpdate();

psEmployee.executeUpdate();

if (count > 0) {

JOptionPane.showMessageDialog(this, "Success!");

new EmployeeUI(userLoginInfo).setVisible(true);

setVisible(false);

dispose();

} else {

JOptionPane.showMessageDialog(this, "Error! Wrong username.");

}

} catch (Exception e) {

e.printStackTrace();

JOptionPane.showMessageDialog(this, "Error! Failed to delete customer.");

}

}

}