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 CustomerEditorUI extends UserBaseUI {

private UserLoginInfo userLoginInfo;

private String username;

private StyledButton saveBtn, deleteBtn, transactionBtn;

private StyledLabel usernameLabel, phoneNumberLabel, accountNumberLabel, passwordLabel, nameLabel;

private StyledTextField usernameTf, phoneNumberTf, accountNumberTf, nameTf;

private StyledPasswordField passwordTf;

public CustomerEditorUI(UserLoginInfo userLoginInfo, String username) {

super(userLoginInfo);

this.userLoginInfo = userLoginInfo;

this.username = username;

if (username != null) {

setPageTitle("Edit Customer");

} else {

setPageTitle("Add Customer");

}

backButton.setVisible(true);

backButton.addActionListener(this);

initUI();

bind();

if (username != null) {

readFromDb();

}

}

private void initUI() {

int x = 200;

int y = 110;

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);

}

phoneNumberLabel = new StyledLabel("Phone Number: ");

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

phoneNumberTf = new StyledTextField();

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

accountNumberLabel = new StyledLabel("Account Number: ");

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

accountNumberTf = new StyledTextField();

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

passwordLabel = new StyledLabel("Password: ");

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

passwordTf = new StyledPasswordField();

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

nameLabel = new StyledLabel("Name: ");

nameLabel.setBounds(x, y + 40 \* 4, 100, 30);

nameTf = new StyledTextField();

nameTf.setBounds(x + 110, y + 40 \* 4, 200, 30);

saveBtn = new StyledButton("Save");

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

deleteBtn = new StyledButton("Delete");

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

deleteBtn.setBounds(x + 110, y + 40 \* 6, 200, 35);

transactionBtn = new StyledButton("Transactions");

transactionBtn.setBounds(x + 110, y + 40 \* 7, 200, 35);

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

deleteBtn.setVisible(false);

transactionBtn.setVisible(false);

}

mainPanel.add(usernameLabel);

mainPanel.add(usernameTf);

mainPanel.add(phoneNumberLabel);

mainPanel.add(phoneNumberTf);

mainPanel.add(accountNumberLabel);

mainPanel.add(accountNumberTf);

mainPanel.add(nameLabel);

mainPanel.add(nameTf);

mainPanel.add(passwordLabel);

mainPanel.add(passwordTf);

mainPanel.add(saveBtn);

mainPanel.add(deleteBtn);

mainPanel.add(transactionBtn);

}

private void bind() {

saveBtn.addActionListener(this);

deleteBtn.addActionListener(this);

transactionBtn.addActionListener(this);

}

private void readFromDb() {

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

try {

PreparedStatement ps = conn.prepareStatement(

"SELECT login.username, login.name, login.password, customer.phoneNumber, customer.accountNumber " +

"FROM customer, login " +

"WHERE customer.username=? AND login.username=customer.username"

);

ps.setString(1, username);

ResultSet rs = ps.executeQuery();

while (rs.next()) {

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

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

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

String phoneNumber = rs.getString("customer.phoneNumber");

String accountNumber = rs.getString("customer.accountNumber");

nameTf.setText(name);

passwordTf.setText(password);

usernameTf.setText(username);

phoneNumberTf.setText(phoneNumber);

accountNumberTf.setText(accountNumber);

}

} catch (SQLException e) {

e.printStackTrace();

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

new CustomerViewerUI(userLoginInfo).setVisible(true);

setVisible(false);

dispose();

}

}

public void actionPerformed(ActionEvent e) {

super.actionPerformed(e);

Object src = e.getSource();

if (src == backButton) {

new CustomerViewerUI(userLoginInfo).setVisible(true);

setVisible(false);

dispose();

} else if (src == saveBtn) {

save();

} else if (src == deleteBtn) {

delete();

} else if (src == transactionBtn) {

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

setVisible(false);

dispose();

}

}

private boolean verifyUsername(String username) {

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

try {

PreparedStatement ps = conn.prepareStatement(

"SELECT username FROM login WHERE username=?"

);

ps.setString(1, username);

System.out.println(ps);

ResultSet rs = ps.executeQuery();

if (rs.next()) {

rs.close();

ps.close();

return true;

}

} catch (SQLException e) {

e.printStackTrace();

}

return false;

}

private void save() {

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

String insertUserQuery = "INSERT INTO login(username, password, status, name) VALUES(?, ?, ?, ?)";

String insertQuery = "INSERT INTO customer(username, phoneNumber, accountNumber) VALUES(?, ?, ?)";

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

String updateCustomerQuery = "UPDATE customer SET phoneNumber=?, accountNumber=? WHERE username=?";

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

PreparedStatement ps;

try {

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

if (verifyUsername(usernameTf.getText())) {

JOptionPane.showMessageDialog(this,

"A user with the given name already exists");

return;

}

// insert the user

ps = conn.prepareStatement(insertUserQuery);

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

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

ps.setString(3, "customer");

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

System.out.println(ps);

ps.executeUpdate();

// insert the customer

ps = conn.prepareStatement(insertQuery);

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

ps.setString(2, phoneNumberTf.getText().trim());

ps.setString(3, accountNumberTf.getText().trim());

System.out.println(ps);

ps.execute();

} else {

// update user

ps = conn.prepareStatement(updateUserQuery);

ps.setString(1, new String(passwordTf.getPassword()));

ps.setString(2, nameTf.getText().trim());

ps.setString(3, username);

System.out.println(ps);

ps.executeUpdate();

// update customer

ps = conn.prepareStatement(updateCustomerQuery);

ps.setString(1, phoneNumberTf.getText().trim());

ps.setString(2, accountNumberTf.getText().trim());

ps.setString(3, usernameTf.getText());

System.out.println(ps);

ps.executeUpdate();

}

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

new CustomerViewerUI(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 updateQuery = "DELETE FROM customer WHERE username=?";

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

PreparedStatement ps;

try {

ps = conn.prepareStatement(updateQuery);

ps.setString(1, username);

System.out.println(ps);

int count = ps.executeUpdate();

if (count > 0) {

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

new CustomerViewerUI(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.");

}

}

}