package m6.ui;

import m6.ConnectionManager;

import m6.UserLoginInfo;

import m6.components.StyledButton;

import m6.components.StyledLabel;

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 AccountEditorUI extends UserBaseUI {

private UserLoginInfo userLoginInfo;

private String accountNumber;

private StyledButton saveBtn, deleteBtn;

private StyledLabel accountNumberLabel, balanceLabel;

private StyledTextField accountNumberTf, balanceTf;

public AccountEditorUI(UserLoginInfo userLoginInfo, String accountNumber) {

super(userLoginInfo);

this.userLoginInfo = userLoginInfo;

this.accountNumber = accountNumber;

if (accountNumber != null) {

setPageTitle("Edit Account");

} else {

setPageTitle("Add Account");

}

backButton.setVisible(true);

backButton.addActionListener(this);

initUI();

bind();

if (accountNumber != null) {

readFromDb();

}

}

private void initUI() {

int x = 180;

int y = 180;

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

accountNumberLabel.setBounds(x, y, 120, 30);

accountNumberTf = new StyledTextField();

accountNumberTf.setBounds(x + 130, y, 200, 30);

balanceLabel = new StyledLabel("Balance: ");

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

balanceTf = new StyledTextField();

balanceTf.setBounds(x + 130, y + 40, 200, 30);

saveBtn = new StyledButton("Save");

saveBtn.setBounds(x + 130, y + 40 + 40, 200, 35);

deleteBtn = new StyledButton("Delete");

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

deleteBtn.setBounds(x + 130, y + 40 + 40 + 40, 200, 35);

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

deleteBtn.setVisible(false);

}

mainPanel.add(accountNumberLabel);

mainPanel.add(balanceLabel);

mainPanel.add(accountNumberTf);

mainPanel.add(balanceTf);

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 accountNumber, balance FROM account WHERE accountNumber=?"

);

ps.setString(1, accountNumber);

ResultSet rs = ps.executeQuery();

if (rs.next()) {

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

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

accountNumberTf.setText(accountNumber);

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

}

} catch (SQLException e) {

e.printStackTrace();

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

new AccountsViewerUI(userLoginInfo).setVisible(true);

setVisible(false);

dispose();

}

}

public void actionPerformed(ActionEvent e) {

super.actionPerformed(e);

Object src = e.getSource();

if (src == backButton) {

new AccountsViewerUI(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 account VALUES(?, ?)";

String updateQuery = "UPDATE account SET accountNumber=?, balance=? WHERE accountNumber=?";

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

String updateTransactionQuery = "UPDATE transaction SET accountNumber=? WHERE accountNumber=?";

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

PreparedStatement ps, psCustomer = null, psTransaction = null;

try {

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

ps = conn.prepareStatement(insertQuery);

} else {

ps = conn.prepareStatement(updateQuery);

psCustomer = conn.prepareStatement(updateCustomerQuery);

psTransaction = conn.prepareStatement(updateTransactionQuery);

psCustomer.setString(1, accountNumberTf.getText().trim());

psCustomer.setString(2, accountNumber);

psTransaction.setString(1, accountNumberTf.getText().trim());

psTransaction.setString(2, accountNumber);

ps.setString(3, accountNumber);

}

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

ps.setDouble(2, Double.parseDouble(balanceTf.getText()));

System.out.println(ps);

ps.execute();

if (psCustomer != null) {

System.out.println(psCustomer);

psCustomer.execute();

System.out.println(psTransaction);

psTransaction.execute();

}

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

new AccountsViewerUI(userLoginInfo).setVisible(true);

setVisible(false);

dispose();

} catch (Exception e) {

e.printStackTrace();

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

}

}

private void delete() {

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

String deleteQuery = "DELETE FROM account WHERE accountNumber=?";

String deleteCustomerQuery = "UPDATE customer SET accountNumber='' WHERE accountNumber=?";

String deleteTransactionQuery = "DELETE FROM transaction WHERE accountNumber=?";

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

PreparedStatement ps, psCustomer, psTransaction;

try {

ps = conn.prepareStatement(deleteQuery);

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

psCustomer = conn.prepareStatement(deleteCustomerQuery);

psCustomer.setString(1, accountNumberTf.getText().trim());

psTransaction = conn.prepareStatement(deleteTransactionQuery);

psTransaction.setString(1, accountNumberTf.getText().trim());

System.out.println(ps);

System.out.println(psCustomer);

System.out.println(psTransaction);

int count = ps.executeUpdate();

psCustomer.execute();

psTransaction.execute();

if (count > 0) {

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

new AccountsViewerUI(userLoginInfo).setVisible(true);

setVisible(false);

dispose();

} else {

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

}

} catch (Exception e) {

e.printStackTrace();

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

}

}

}