package net.banking.dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.List;

import net.banking.bean.UserBean;

import net.banking.dbconnection.ConnectionProvider;

public class UserDao {

private Connection conn;

List<UserBean> listUsers = new ArrayList<UserBean>();

public UserDao() {

System.out.println("Got The Connnection...");

conn = ConnectionProvider.getConnection();

}

public void addUser(UserBean userBean) {

try {

System.out.println("In addUser()");

if(conn==null){

System.out.println("conn is null");

}

String sql = "INSERT INTO accountinfo(userID,name,Balnce)" +

" VALUES (?, ?, ? )";

PreparedStatement ps = conn.prepareStatement(sql);

ps.setInt(1, userBean.getId());

ps.setString(2, userBean.getfName());

ps.setString(3, userBean.getlName());

ps.executeUpdate();

System.out.println("after prepared Statement");

} catch (SQLException e) {

e.printStackTrace();

}

}

public void removeUser(int userId) {

try {

System.out.println("In removeUser()");

String sql = "DELETE FROM accountinfo WHERE userID=?";

PreparedStatement ps = conn

.prepareStatement(sql);

ps.setInt(1, userId);

ps.executeUpdate();

} catch (SQLException e) {

e.printStackTrace();

}

}

public void editUser(UserBean userBean) {

try {

System.out.println("In editUser()");

String sql = "UPDATE accountinfo SET name=?, balance=?" +

" WHERE userid=?";

PreparedStatement ps = conn

.prepareStatement(sql);

ps.setString(1, userBean.getfName());

ps.setString(2, userBean.getlName());

ps.setInt(3, userBean.getId());

ps.executeUpdate();

} catch (SQLException e) {

e.printStackTrace();

}

}

public String doTransaction(UserBean userBean) {

String msg="";

int Id;

String name;

int balance = 0;

System.out.println("in doTrasaction");

if(userBean.getfName()!=null){

//for deposite operation

try {

System.out.println("In doTrasaction()");

String sql = "SELECT \* FROM accountinfo where userID=?";

PreparedStatement ps = conn.prepareStatement(sql);

ps.setInt(1, userBean.getId());

ResultSet rs = ps.executeQuery();

while (rs.next()) {

Id=rs.getInt("userID");

name=rs.getString("name");

balance=rs.getInt("Balnce");

}

ps.close();

rs.close();

} catch (SQLException e) {

e.printStackTrace();

}

int preAmount=balance;

int lastAmount=preAmount;

String CurrAmount=userBean.getfName();

int updatedAmount=Integer.parseInt(CurrAmount);

Integer finalAmount=lastAmount+updatedAmount;

msg="Your Last Balance is "+lastAmount+" And Deposite amount is "+updatedAmount;

GetTransaction(finalAmount, userBean);

}else{

//for withdraw operation

try {

System.out.println("In doTrasaction()");

String sql = "SELECT \* FROM accountinfo where userID=?";

PreparedStatement ps = conn.prepareStatement(sql);

ps.setInt(1, userBean.getId());

ResultSet rs = ps.executeQuery();

while (rs.next()) {

Id=rs.getInt("userID");

name=rs.getString("name");

balance=rs.getInt("Balnce");

}

ps.close();

rs.close();

} catch (SQLException e) {

e.printStackTrace();

}

int preAmount=balance;

int lastAmount=preAmount;

String CurrAmount=userBean.getlName();

int updatedAmount=Integer.parseInt(CurrAmount);

if(updatedAmount>=lastAmount){

msg="There is not enough fund";

lastAmount=0;

}else{

Integer finalAmount=lastAmount-updatedAmount;

msg="Your Last Balance is "+lastAmount+" And Withdrow amount is "+updatedAmount;

GetTransaction(finalAmount,userBean);

}

}

return msg;

}

private void GetTransaction(Integer finalAmount, UserBean userBean) {

// TODO Auto-generated method stub

try {

System.out.println("in getTrasaction");

String sql = "UPDATE accountinfo SET Balnce=?" + " WHERE userID=?";

PreparedStatement ps = conn.prepareStatement(sql);

ps.setString(1, finalAmount.toString());

// ps.setString(2, userBean.getlName());

ps.setInt(2, userBean.getId());

int x=ps.executeUpdate();

System.out.println(x+" Row Updated ");

} catch (SQLException e) {

e.printStackTrace();

}

}

public String selectData(UserBean userBean){

String msg="";

// UserBean userBean = new UserBean();

try {

System.out.println("In selectData()");

String sql = "SELECT \* FROM accountinfo where userID=?";

PreparedStatement ps = conn.prepareStatement(sql);

ps.setInt(1, userBean.getId());

ResultSet rs = ps.executeQuery();

if(rs.next()){

msg="The Authentication is Successfull";

while (rs.next()) {

UserBean userBean1 = new UserBean();

userBean1.setId(rs.getInt("userID"));

userBean1.setfName(rs.getString("name"));

userBean1.setlName(rs.getString("Balnce"));

listUsers.add(userBean1);

}

}else{

msg="The Authentication Has Failed";

}

ps.close();

rs.close();

userBean.setMsg(msg);

} catch (SQLException e) {

e.printStackTrace();

}

return msg;

}

public List getAllUsers() {

List users = new ArrayList();

try {

System.out.println("In getAllUser()");

String sql = "SELECT \* FROM accountinfo";

PreparedStatement ps = conn.prepareStatement(sql);

ResultSet rs = ps.executeQuery();

while (rs.next()) {

UserBean userBean = new UserBean();

userBean.setId(rs.getInt("userID"));

userBean.setfName(rs.getString("name"));

userBean.setlName(rs.getString("Balnce"));

users.add(userBean);

}

} catch (SQLException e) {

e.printStackTrace();

}

return users;

}

public UserBean getUserById(int userId) {

UserBean userBean = new UserBean();

try {

System.out.println("In getUserById()");

String sql = "SELECT \* FROM accountinfo WHERE userID=?";

PreparedStatement ps = conn.

prepareStatement(sql);

ps.setInt(1, userId);

ResultSet rs = ps.executeQuery();

if (rs.next()) {

userBean.setId(rs.getInt("userID"));

userBean.setfName(rs.getString("name"));

userBean.setlName(rs.getString("Balnce"));

}

} catch (SQLException e) {

e.printStackTrace();

}

return userBean;

}

}