IQueryMapper

package com.cg.wallet.dao;

public interface IQueryMapper {

public String insert="insert into account(name,email,mobileno,balance) values(?,?,?,?)";

public String getBal="select balance from account where mobileno=?";

public String getacc="select \* from account where mobileno=?";

public String update="update account set balance=?,date1=? where mobileno=?";

}

WalletDaoImpl

package com.cg.wallet.dao;

import java.sql.Connection;

import java.sql.Date;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.time.LocalDate;

import java.time.LocalDateTime;

import com.cg.wallet.beans.Account;

import com.cg.wallet.db.DBUtil;

import com.cg.wallet.exception.WalletException;

public class WalletDaoImpl implements IWalletDao {

@Override

public String createAccount(Account acc) throws WalletException {

Connection con=DBUtil.getConnection();

PreparedStatement stat;

try{

con.setAutoCommit(false);

stat = con.prepareStatement(IQueryMapper.insert);

stat.setString(1, acc.getName());

stat.setString(2, acc.getEmail());

stat.setString(3, acc.getMobileno());

stat.setDouble(4, acc.getBalance());

int res=stat.executeUpdate();

if(res==1){

con.commit();

return acc.getMobileno();

}else{

throw new WalletException("Could not create account");

}

} catch (SQLException e) {

// TODO Auto-generated catch block

throw new WalletException(e.getMessage());

}

}

@Override

public double showBalance(String mobileNo) throws WalletException {

Connection con=DBUtil.getConnection();

PreparedStatement stat;

try{

stat=con.prepareStatement(IQueryMapper.getBal);

stat.setString(1, mobileNo);

ResultSet rs=stat.executeQuery();

con.commit();

if(rs!=null){

rs.next();

return rs.getDouble("balance");

}else{

throw new WalletException("mobile no does not exists");

}

}catch (SQLException e) {

// TODO Auto-generated catch block

throw new WalletException(e.getMessage());

}

}

@Override

public Account deposit(String mobileNo,double depositAmt) throws WalletException {

Connection con=DBUtil.getConnection();

PreparedStatement stat;

PreparedStatement stat1;

try{

stat=con.prepareStatement(IQueryMapper.getacc);

stat.setString(1, mobileNo);

ResultSet rs=stat.executeQuery();

if(rs!=null){

rs.next();

Account acc=new Account();

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

acc.setName(rs.getString("name"));

acc.setMobileno(rs.getString("mobileno"));

acc.setBalance(balance);

acc.setEmail(rs.getString("email"));

acc.setDate(Date.valueOf(LocalDate.now()));

//acc.setDate(LocalDateTime.now());

//Account acc1=new Account();

stat1=con.prepareStatement(IQueryMapper.update);

stat1.setDouble(1, acc.getBalance());

stat1.setDate(2, acc.getDate());

stat1.setString(3, acc.getMobileno());

int rs1=stat1.executeUpdate();

if(rs1==1){

con.commit();

return acc;

}else{

throw new WalletException("balance is not updated");

}

}

else{

throw new WalletException("mobile no does not exists");

}

}catch (SQLException e) {

// TODO Auto-generated catch block

throw new WalletException(e.getMessage());

}

}

@Override

public Account withdraw(String mobileNo,double withdrawAmt) throws WalletException {

Connection con=DBUtil.getConnection();

PreparedStatement stat;

PreparedStatement stat1;

try{

stat=con.prepareStatement(IQueryMapper.getacc);

stat.setString(1, mobileNo);

ResultSet rs=stat.executeQuery();

if(rs!=null){

rs.next();

Account acc=new Account();

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

acc.setName(rs.getString("name"));

acc.setMobileno(rs.getString("mobileno"));

acc.setBalance(balance);

acc.setEmail(rs.getString("email"));

acc.setDate(Date.valueOf(LocalDate.now()));

//acc.setDate(LocalDateTime.now());

//Account acc1=new Account();

stat1=con.prepareStatement(IQueryMapper.update);

stat1.setDouble(1, acc.getBalance());

stat1.setDate(2, acc.getDate());

stat1.setString(3, acc.getMobileno());

int rs1=stat1.executeUpdate();

if(rs1==1){

//rs1.next();

con.commit();

//bal1=acc.getBalance();

return acc;

}else{

throw new WalletException("mobile no does not exists");

}

//return bal1;

}

else{

throw new WalletException("mobile no does not exists");

}

}catch (SQLException e) {

// TODO Auto-generated catch block

throw new WalletException(e.getMessage());

}

}

@Override

public Account printTransactionDetails(String mobileNo) throws WalletException{

Connection con=DBUtil.getConnection();

PreparedStatement stat;

try{

stat=con.prepareStatement(IQueryMapper.getacc);

stat.setString(1, mobileNo);

ResultSet rs=stat.executeQuery();

con.commit();

if(rs!=null){

rs.next();

Account ac=new Account();

ac.setName(rs.getString("name"));

ac.setMobileno(rs.getString("mobileno"));

ac.setEmail(rs.getString("email"));

ac.setBalance(rs.getDouble("balance"));

ac.setDate(rs.getDate("date1"));

return ac;

}else{

throw new WalletException("mobile no does not exists");

}

}catch (SQLException e) {

// TODO Auto-generated catch block

throw new WalletException(e.getMessage());

}

}

}

WalletServiceImp

package com.cg.wallet.service;

import java.time.LocalDateTime;

import com.cg.wallet.beans.Account;

import com.cg.wallet.dao.IWalletDao;

import com.cg.wallet.dao.WalletDaoImpl;

import com.cg.wallet.exception.WalletException;

public class WalletServiceImpl implements IWalletService {

IWalletDao dao = new WalletDaoImpl();

@Override

public String createAccount(Account acc) throws WalletException {

if (!acc.getMobileno().matches("\\d{10}")) {

throw new WalletException("Mobile number should contain 10 digits");

}

if (acc.getName().isEmpty() || acc.getName() == null) {

throw new WalletException("Name cannot be empty");

} else {

if (!acc.getName().matches("[A-Z][A-Za-z]{3,}")) {

throw new WalletException(

"Name should start with capital letter and should contain only alphabets");

}

}

if (acc.getBalance() < 0) {

throw new WalletException("Balance should be greater than zero");

}

if (!acc.getEmail().matches("[a-z]+@[a-z]+\\.com")) {

throw new WalletException("Enter valid emailid");

}

return dao.createAccount(acc);

}

@Override

public double showBalance(String mobileNo) throws WalletException {

if (!mobileNo.matches("\\d{10}")) {

throw new WalletException("Mobile number should contain 10 digits");

}

return dao.showBalance(mobileNo);

}

@Override

public Account deposit(String mobileNo, double depositAmt)

throws WalletException {

if (!mobileNo.matches("\\d{10}")) {

throw new WalletException("Mobile number should contain 10 digits");

}

//Account acc=dao.findOne(mobileNo);

if (depositAmt <= 0) {

throw new WalletException(

"Deposit amount must be greater than zero");

}

return dao.deposit(mobileNo,depositAmt);

//acc1.getBalance();

}

@Override

public Account withdraw(String mobileNo, double withdrawAmt)

throws WalletException {

if (!mobileNo.matches("\\d{10}")) {

throw new WalletException("Mobile number should contain 10 digits");

}

//Account acc=dao.findOne(mobileNo);

//if (withdrawAmt > acc.getBalance() || withdrawAmt <= 0) {

if ( withdrawAmt <= 0) {

throw new WalletException(

"The amount to be withdrawn should be greater than available balance and greater than zero");

}

//acc.setBalance(acc.getBalance() - withdrawAmt);

// acc.setDate(LocalDateTime.now());

Account acc1 = dao.withdraw(mobileNo,withdrawAmt);

return acc1;

}

@Override

public boolean fundTransfer(String sourceMobileNo, String destMobileNo,

double transferAmt) throws WalletException {

if (!sourceMobileNo.matches("\\d{10}")) {

throw new WalletException("Mobile number should contain 10 digits");

}

if (!destMobileNo.matches("\\d{10}")) {

throw new WalletException("Mobile number should contain 10 digits");

}

IWalletService service = new WalletServiceImpl();

Account acc1 = service.withdraw(sourceMobileNo, transferAmt);

Account d2 = service.deposit(destMobileNo, transferAmt);

return true;

}

@Override

public Account printTransactionDetails(String mobileNo)

throws WalletException {

return dao.printTransactionDetails(mobileNo);

}

}

DBUtil

package com.cg.wallet.db;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import com.cg.wallet.exception.WalletException;

public class DBUtil {

public static Connection getConnection() throws WalletException{

String url="jdbc:oracle:thin:@localhost:1521:xe";

try{

Class.forName("oracle.jdbc.driver.OracleDriver");

return DriverManager.getConnection(url,"system","root");

}catch(ClassNotFoundException e){

throw new WalletException(e.getMessage());

}catch(SQLException e1){

throw new WalletException(e1.getMessage());

}

}

}