**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Java Script Assignments\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<HTML>

<HEAD>

<META NAME="GENERATOR" Content="Microsoft Visual Studio 8.0">

<TITLE>Registration Details...</TITLE>

<script language="javascript" type="text/javascript">

var country\_arr = new Array("Afghanistan","India");

var s\_a = new Array();

s\_a[0]="";

s\_a[1]="Oruzgan|Paktia|Paktika|Parvan|Samangan|Sar-e Pol|Takhar|Vardak|Zabol";

s\_a[2]="Goa|Maharashtra|AP|UP|Bihar|Delhi";

function print\_country(country\_id)

{

// given the id of the <select> tag as function argument,

//it inserts <option> tags

var option\_str =

document.getElementById(country\_id);

option\_str.length=0;

option\_str.options[0] = new Option('Select Country','');

option\_str.selectedIndex = 0;

for (var i=0; i<country\_arr.length; i++)

{

option\_str.options[option\_str.length] = new Option(country\_arr[i],country\_arr[i]);

}

}

function print\_state(state\_id, state\_index)

{

var option\_str = document.getElementById(state\_id);

option\_str.length=0; // Fixed by Julian Woods

option\_str.options[0] = new Option('Select State','');

option\_str.selectedIndex = 0;

var state\_arr = s\_a[state\_index].split("|");

for (var i=0; i<state\_arr.length; i++)

{

option\_str.options[option\_str.length] =

new Option(state\_arr[i],state\_arr[i]);

alert(option\_str.length)

}

}

function chkAllValidation()

{

var digitPattern=/[0-9]/;

var mobilePattern=/[0-9]{10}/;

var personalEmailPtrn =

"^[\_A-Za-z0-9-]+(\\.[\_A-Za-z0-9-]+)\*@[A-Za-z0-9]+(\\.[A-Za-z0-9]+)\*(\\.[A-Za-z]{2,})$";

var compEmailPattern1=/[a-zA-Z][a-zA-Z0-9\\_]\*\@igate\.com/;

var compEmailPattern2=/[a-zA-Z][a-zA-Z0-9\\_]\*\@igate\.in/;

var compEmailPattern3=/[a-zA-Z][a-zA-Z0-9\\_]\*\@igate\.co.\in/;

var fnamedata=window.document.form1.fnTxt.value;

var mobilenodata=window.document.form1.mobileTxt.value;

var passdata=window.document.form1.passTxt.value;

var conpassdata=window.document.form1.conPassTxt.value;

var emaildata=window.document.form1.emailTxt.value;

var personalemaildata=window.document.form1.personalEmailTxt.value;

if((fnamedata=="" )|| (fnamedata==null))

{

// alert("Mandatory Field");

var fnSection=document.getElementById("fnDiv");

fnSection.innerHTML="Mandatory Field";

return false;

}

if(fnamedata.match( digitPattern))

{

//alert (" Digits Not allowed in First Name ");

var fnSection=document.getElementById("fnDiv");

fnSection.innerHTML="Digits Not allowed in First Name";

return false;

}

if(!mobilenodata.match(mobilePattern))

{

// alert( "mobilenodata no should be digits and Max length should be 10");

var mobileSection=document.getElementById("mobileDiv");

mobileSection.innerHTML="mobilenodata no should be digits and Max length should be 10";

return false;

}

if(passdata!= conpassdata)

{

// alert(" Password and Confirm Password should match");

var conPassSection=document.getElementById("conPassDiv");

conPassSection.innerHTML="Password and Confirm Password should match";

return false;

}

if (compEmailPattern1.exec(emaildata)<=0)

{

if (compEmailPattern2.exec(emaildata)<=0)

{

if (compEmailPattern3.exec(emaildata)<=0)

{

//alert(" "+re1.exec(emaildata));

var emailSection=document.getElementById("emailDiv");

emailSection.innerHTML=" Only IGate Email Id";

// return false;

}

// return false;

}

return false;

}

if(!personalemaildata.match(personalEmailPtrn))

{

var personalEmailSection=document.getElementById("personalEmailDiv");

personalEmailSection.innerHTML=" Email Address Should have underscore digits . & @";

return false;

}

/\* if(personalemaildata.match(personalEmailPtrn))

{

var personalEmailSection=document.getElementById("personalEmailDiv");

personalEmailSection.innerHTML=" Email Address Should have underscore digits . & @";

return false;

} \*/

var extraEmailPattern = new RegExp("@[a-z]+");

var extraEmaildata=window.document.form1.extraEmailTxt.value;

if (!(extraEmaildata.match(extraEmailPattern)))

{

var extraEmailSection=document.getElementById("extraEmailDiv");

extraEmailSection.innerHTML="Not a Valid Email Address ";

return false;

}

return true;

}

</script>

</HEAD>

<BODY bgcolor="cyan">

<h3> Enter UR Registration Details....</h3>

<form name="form1" action="success.html" onsubmit="return chkAllValidation();">

<table border="1">

<tr><td><font color="red">\*</font>First Name:</td><td><input type ="text" name="fnTxt" value="" />

</td><td><div id="fnDiv"></div></td> </tr>

<tr><td>Last Name :</td><td><input type ="text" name="lnTxt" value="" />

</td><td><div id="lnDiv"></div> </td></tr>

<tr>

<td>Mobile No:</td>

<td> <input type ="text" name="mobileTxt" value="" /> </td>

<td><div id="mobileDiv"></div> </td>

</tr>

<tr><td>Password: </td>

<td><input type ="password" name="passTxt" value="" /> </td>

<td><div id="passDiv"></div></td> </tr>

<tr>

<td>Confirm Password:</td>

<td> <input type ="password" name="conPassTxt" value="" /> </td>

<td><div id="conPassDiv"></div></td>

</tr>

<tr><td> Comp Email :</td>

<td><input type="text" name="emailTxt" value=" " /></td>

<td><div id="emailDiv"></div></td>

</tr>

<tr><td> Personal Email :</td>

<td><input type="text" name="personalEmailTxt" value=" " /></td>

<td><div id="personalEmailDiv"></div></td>

</tr>

<tr>

<td>Select Country:</td>

<td><select onchange="print\_state('state', this.selectedIndex);" id="country"

name ="country"></select>

</td>

</tr>

<tr>

<td> City/District/State:</td>

<td><select name ="state" id ="state">

</select></td>

</tr>

<tr>

<td><input type="submit" name="regButton" value="REGISTER" /></td>

</tr>

</table>

</form>

<script language="javascript">print\_country("country");</script>

</BODY>

</HTML>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ANT Demo\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**package** oata;

**public** **class** HelloWorld {

**public** **static** **void** main(String[] args) {

System.*out*.println("Hello World Vaishali............"); }

}

JavaProject/build.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<project default=*"main"* >

<property

name=*"src.dir"*

value=*"src"* />

<property

name=*"build.dir"*

value=*"build"* />

<property

name=*"classes.dir"*

value=*"${build.dir}/classes"* />

<property

name=*"jar.dir"*

value=*"${build.dir}/jar"* />

<property

name=*"main-class"*

value=*"oata.HelloWorld"* />

<target name=*"clean"* >

<delete dir=*"${build.dir}"* />

</target>

<target name=*"compile"* >

<mkdir dir=*"${classes.dir}"* />

<javac

destdir=*"${classes.dir}"*

srcdir=*"${src.dir}"* />

</target>

<target

name=*"jar"*

depends=*"compile"* >

<mkdir dir=*"${jar.dir}"* />

<jar

basedir=*"${classes.dir}"*

destfile=*"${jar.dir}/${ant.project.name}.jar"* >

<manifest>

<attribute

name=*"Main-Class"*

value=*"${main-class}"* />

</manifest>

</jar>

</target>

<target

name=*"run"*

depends=*"jar"* >

<java

fork=*"true"*

jar=*"${jar.dir}/${ant.project.name}.jar"* />

</target>

<target

name=*"clean-build"*

depends=*"clean,jar"* />

<target

name=*"main"*

depends=*"clean,run"* />

</project>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Core Java Security\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Sender Steps\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Step-1) C:\ex>javac LineNumberReaderDemo.java

Step-2) C:\ex>java LineNumberReaderDemo

1 Hello

2 Welcome

Step-3) C:\ex>java -Djava.security.manager LineNumberReaderDemo

Exception in thread "main" java.security.AccessControlException: access denied (

java.io.FilePermission c:\ex\Output.txt read)

at java.security.AccessControlContext.checkPermission(Unknown Source)

at java.security.AccessController.checkPermission(Unknown Source)

at java.lang.SecurityManager.checkPermission(Unknown Source)

at java.lang.SecurityManager.checkRead(Unknown Source)

at java.io.FileInputStream.<init>(Unknown Source)

at java.io.FileInputStream.<init>(Unknown Source)

at java.io.FileReader.<init>(Unknown Source)

at LineNumberReaderDemo.main(LineNumberReaderDemo.java:10)

Step-4) C:\ex>jar cvf linenum.jar LineNumberReaderDemo.class

added manifest

adding: LineNumberReaderDemo.class(in = 1112) (out= 641)(deflated 42%)

**Step-5)**

C:\ex>keytool -genkey -alias signFiles -keypass igatepatni123 -keystore igatepatnistore -storepass store123

What is your first and last name?

[Unknown]: Vaishali S

What is the name of your organizational unit?

[Unknown]: iLearn

What is the name of your organization?

[Unknown]: iGate

What is the name of your City or Locality?

[Unknown]: Pune

What is the name of your State or Province?

[Unknown]: MS

What is the two-letter country code for this unit?

[Unknown]: IN

Is CN=Vaishali S, OU=iLearn, O=iGate, L=Pune, ST=MS, C=IN correct?

[no]: yes

**Step-6)**

C:\ex>jarsigner -keystore igatepatnistore -signedjar slinenum.jar linenum.jar signFiles

Enter Passphrase for keystore: store123

Enter key password for signFiles :igatepatni123

Warning:

The signer certificate will expire within six months.

**Step-7)**

C:\ex>keytool -export -keystore igatepatnistore -alias signFiles -file iGatePatni.cer

Enter keystore password: store123

Certificate stored in file <iGatePatni.cer>

**Step-8)**  # Share the Signed Jar and Certificate with receiver.

Cut Pase Signed Jar file in Receiver Folder

**Receiver Steps\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Step-1)**

C:\ex1>java -cp slinenum.jar LineNumberReaderDemo

1 Hello

2 Welcome

**Step-2)**

C:\ex1>java -Djava.security.manager -cp slinenum.jar LineNumberReaderDemo

Exception in thread "main" java.security.AccessControlException: access denied (

java.io.FilePermission c:\ex\Output.txt read)

at java.security.AccessControlContext.checkPermission(Unknown Source)

at java.security.AccessController.checkPermission(Unknown Source)

at java.lang.SecurityManager.checkPermission(Unknown Source)

at java.lang.SecurityManager.checkRead(Unknown Source)

at java.io.FileInputStream.<init>(Unknown Source)

at java.io.FileInputStream.<init>(Unknown Source)

at java.io.FileReader.<init>(Unknown Source)

at LineNumberReaderDemo.main(LineNumberReaderDemo.java:10)

**Step-3)** Copy Pastefile iGatePatni.cer in receiver folder[ex1]

**Step-4)**

C:\ex1>keytool -import -alias igatePatniReceiver -file iGatePatni.cer -keystore iLearnReceiverStore

Enter keystore password: vaishali123

Re-enter new password: vaishali123

#New Password- Any Password can be typed

Enter keystore password:

Re-enter new password:

Owner: CN=Vaishali S, OU=iLearn, O=iGate, L=Pune, ST=MS, C=IN

Issuer: CN=Vaishali S, OU=iLearn, O=iGate, L=Pune, ST=MS, C=IN

Serial number: 51804fa1

Valid from: Tue Jul 22 15:12:32 IST 2014 until: Mon Oct 20 15:12:32 IST 2014

Certificate fingerprints:

MD5: 2D:01:59:7A:77:EA:8F:77:A7:27:A6:78:2E:C7:9E:79

SHA1: 90:4F:30:B1:E2:41:CD:74:F8:47:87:EF:B0:D2:16:FC:00:A8:7E:F7

SHA256: 91:EC:BD:FE:7B:E5:E7:B1:FE:82:94:A5:44:66:E8:A0:3B:3B:E6:CC:F6:

27:7C:B3:66:CF:80:BB:54:D0:2B:CA

Signature algorithm name: SHA1withDSA

Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false

SubjectKeyIdentifier [

KeyIdentifier [

0000: F1 18 85 A3 5E CA 50 67 BF D4 AB 6A 84 6E D3 29 ....^.Pg...j.n.)

0010: 98 A8 0E 6D ...m

]

]

Trust this certificate? [no]: yes

Certificate was added to keystore

**Step-5)** Open Policy Tool

C:\ex1>policytool

**Step-6)** Create a policy file called mypolicy.policy using create new policy option . Give a Read permission to all files.

**Step-7)**

C:\ex1>java -Djava.security.manager -cp slinenum.jar LineNumberReaderDemo

Exception in thread "main" java.security.AccessControlException: access denied (

java.io.FilePermission c:\ex\Output.txt read)

at java.security.AccessControlContext.checkPermission(Unknown Source)

at java.security.AccessController.checkPermission(Unknown Source)

at java.lang.SecurityManager.checkPermission(Unknown Source)

at java.lang.SecurityManager.checkRead(Unknown Source)

at java.io.FileInputStream.<init>(Unknown Source)

at java.io.FileInputStream.<init>(Unknown Source)

at java.io.FileReader.<init>(Unknown Source)

at LineNumberReaderDemo.main(LineNumberReaderDemo.java:10)

**Step-8)**

C:\ex1>java -Djava.security.manager -Djava.security.policy=mypolicy.policy -cp slinenum.jar LineNumberReaderDemo

1 Hello

2 Welcome

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**\*\*\*\*\*\*\*\*\*Java-Lab 9.1\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\* MobileBean.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.bean;

public class MobileBean {

long mobileId ;

String mobileName;

double mobilePrice ;

int mobileQuantity ;

public long getMobileId() { return mobileId; }

public void setMobileId(long mobileId) { this.mobileId = mobileId; }

public String getMobileName() { return mobileName; }

public void setMobileName(String mobileName) {this.mobileName = mobileName;}

public double getMobilePrice() { return mobilePrice; }

public void setMobilePrice(double mobilePrice) {this.mobilePrice = mobilePrice; }

public int getMobileQuantity() { return mobileQuantity; }

public void setMobileQuantity(int mobileQuantity) {this.mobileQuantity = obileQuantity;

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* PurchaseDetailsBean.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.bean;

import java.util.Date;

public class PurchaseDetailsBean {

int purchaseId;

String custName ;

String mailId;

long phoneNo;

Date purchaseDate;

int mobileId;

int qty;

public int getQty() {return qty; }

public void setQty(int qty) {this.qty = qty;}

public int getPurchaseId() { return purchaseId; }

public String getCustName() { return custName; }

public void setCustName(String custName) { this.custName = custName; }

public String getMailId() { return mailId; }

public void setMailId(String mailId) { this.mailId = mailId; }

public Date getPurchaseDate() { return purchaseDate; }

public void setPurchaseDate(Date purchaseDate) {this.purchaseDate = purchaseDate;}

public int getMobileId() {return mobileId; }

public void setMobileId(int mobileId) { this.mobileId = mobileId; }

public long getPhoneNo() {return phoneNo; }

public void setPhoneNo(long phoneNo) {this.phoneNo = phoneNo; }

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DAO\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*MobileDAOImpl\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.dao;

import java.sql.Connection;

import java.sql.Date;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.Collection;

import com.igate.dto.MobileDTO;

import com.igate.exception.InvalidDataException;

import com.igate.util.DBUtil;

public class MobileDAOImpl implements MobileDAO

{

Connection con=DBUtil.getConnection();

PreparedStatement updatePst,pst=null;

ResultSet rs=null;

Statement st;

@Override

public void addMobile() // TODO Auto-generated method stub }

@Override

public void deleteMobile(int mobileId) { // TODO Auto-generated method stub }

@Override

public ArrayList<MobileDTO> getMobiles() { // TODO Auto-generated method stub return null;

}

@Override

public MobileDTO getMobileByPriceRange(int lowRange, int highRange) {

return null; }

@Override

public void updateMobileByQty(int newQty,int mobileId)throws InvalidDataException

{

updatePst=null;

int oldQty=0;

try

{

con.setAutoCommit(false);

oldQty=getMobileQty(mobileId);

System.out.println(" Old qty is .............."+ oldQty);

updatePst=con.prepareStatement("update mobiles set quantity=?" +

"Where mobileid=?");

updatePst.setInt(1,(oldQty-newQty));

updatePst.setInt(2,mobileId);

int kk=updatePst.executeUpdate();

con.commit();

System.out.println("Data is updated in table : " +kk);

}

catch (SQLException e)

{

try

{

con.rollback();

}

catch (SQLException e1)

{

throw new InvalidDataException(e.getMessage());

}

}

finally

{

try

{

con.close();

updatePst.close();

} catch (SQLException e)

{

throw new InvalidDataException(e.getMessage());

}

}

}

@Override

public ArrayList<Integer> getMobileIds()throws InvalidDataException

{

ArrayList<Integer> mobilIdL=new ArrayList<Integer>();

try

{

st=con.createStatement();

rs=st.executeQuery("select mobileid FROM mobiles");

while(rs.next())

{

mobilIdL.add(rs.getInt(1));

}

}

catch (Exception e)

{

throw new InvalidDataException(e.getMessage());

}

finally

{

try

{

con.close();

st.close();

rs.close();

} catch (SQLException e)

{

throw new InvalidDataException(e.getMessage());

}

}

return mobilIdL;

}

@Override

public int getMobileQty(int mobileId) throws InvalidDataException

{

int noOfQty=0;

try

{

pst=con.prepareStatement("select quantity FROM mobiles where mobileid=?");

pst.setInt(1, mobileId);

rs=pst.executeQuery();

rs.next();

noOfQty=rs.getInt(1);

}

catch (Exception e)

{

throw new InvalidDataException(e.getMessage());

}

finally

{

try

{

con.close();

pst.close();

rs.close();

} catch (SQLException e)

{

throw new InvalidDataException(e.getMessage());

}

}

return noOfQty;

}}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* PurchaseDetailsDAOImpl .java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.dao;

import java.sql.Connection;

import java.sql.Date;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.Calendar;

import java.util.Collection;

import com.igate.dto.MobileDTO;

import com.igate.dto.PurchaseDetailsDTO;

import com.igate.exception.InvalidDataException;

import com.igate.util.DBUtil;

public class PurchaseDetailsDAOImpl implements PurchaseDetailsDAO

{

Connection con=DBUtil.getConnection();

PreparedStatement updatePst=null;

PreparedStatement pst;

ResultSet rs=null;

@Override

public void addPurchaseDetails(PurchaseDetailsDTO purchaseDetailsDto)throws InvalidDataException

{

/\*\* create sequence Seq\_Purchase\_ID

start with 100

INCREMENT By 1

maxvalue 1000

nocycle

\*/

Calendar currenttime = Calendar.getInstance();

Date sqlCurrDate = new Date((currenttime.getTime()).getTime());

try

{

pst= con.prepareStatement

("insert into purchasedetails values(Seq\_Purchase\_ID.NEXTVAL,?,?,?,?,?)");

pst.setString(1, purchaseDetailsDto.getCustName());

pst.setString (2,purchaseDetailsDto.getMailId());

pst.setLong(3,purchaseDetailsDto.getPhoneNo());

pst.setDate(4, sqlCurrDate);

pst.setInt(5,purchaseDetailsDto.getMobileId());

int ii=pst.executeUpdate();

System.out.println("Data is inserted and table :"+ii + " ");

con.commit();

}

//

catch (SQLException e)

{

throw new InvalidDataException(e.getMessage());

}

finally

{

try

{

con.close();

rs.close();

} catch (SQLException e)

{

throw new InvalidDataException(e.getMessage());

}

}

}

@Override

public void deletePurchaseDetails() { // TODO Auto-generated method stub }

@Override

public ArrayList<PurchaseDetailsDTO> getPurchaseDetails() { // TODO Auto-generated method stub

return null;

}

@Override

public MobileDTO getPurchaseDetailsById() {

// TODO Auto-generated method stub

return null; }

@Override

public void updatePurchaseDetails() {

// TODO Auto-generated method stub

}}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* InvalidDataException..java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.exception;

public class InvalidDataException extends Exception

{

public InvalidDataException(String msg)

{

super(msg);

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* MobileService.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.service;

import java.util.Collection;

import com.igate.dao.MobileDAO;

import com.igate.dao.MobileDAOImpl;

import com.igate.exception.InvalidDataException;

public class MobileService

{

// public void addMobile();

//public void deleteMobile(int mobileId);

//public Collection<MobileDTO> getMobiles();

//public MobileDTO getMobileByPriceRange(int lowRange,int highRange);

public int getMobileQty(int mobileId) throws InvalidDataException

{

MobileDAO md=new MobileDAOImpl();

return md.getMobileQty(mobileId);

}

public void updateMobileByQty(int qty,int mobileId)throws InvalidDataException

{

MobileDAO md=new MobileDAOImpl();

md.updateMobileByQty(qty, mobileId);

}

// public int getMobileQty(int mobileId);

public Collection<Integer> getMobileIds() throws InvalidDataException

{

MobileDAO md=new MobileDAOImpl();

return md.getMobileIds();

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* PurchaseDetailsService.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.service;

import java.util.Collection;

import java.util.regex.Pattern;

import com.igate.dao.MobileDAO;

import com.igate.dao.MobileDAOImpl;

import com.igate.dao.PurchaseDetailsDAO;

import com.igate.dao.PurchaseDetailsDAOImpl;

import com.igate.dto.PurchaseDetailsDTO;

import com.igate.exception.InvalidDataException;

public class PurchaseDetailsService

{

public void addPurchaseDetails(PurchaseDetailsDTO purchaseDetailsDto)throws InvalidDataException

{

boolean custFlag = validateName(purchaseDetailsDto.getCustName());

if (custFlag == true)

{

boolean emailFlag = validateEmail(purchaseDetailsDto.getMailId());

if (emailFlag == true)

{

boolean phoneFlag = validatePhone(purchaseDetailsDto.getPhoneNo());

if (phoneFlag == true)

{

boolean mobileFlag = validateMobileId(purchaseDetailsDto.getMobileId());

if (mobileFlag == true)

{

boolean qtyFlag = validateQty(purchaseDetailsDto.getQty(),purchaseDetailsDto.getMobileId());

if (qtyFlag == true)

{

PurchaseDetailsDAO pddao=new PurchaseDetailsDAOImpl();

MobileDAO mobileDao=new MobileDAOImpl();

pddao.addPurchaseDetails(purchaseDetailsDto); mobileDao.updateMobileByQty(purchaseDetailsDto.getQty(),

purchaseDetailsDto.getMobileId());

//break;//break Qty loop }

//break;// Mobile loop

}

//break;//break Phone loop

}

//break;//break email loop }

//break ;//break customerfalg loop

}

/\* public void deletePurchaseDetails()

{

}

public Collection<PurchaseDetailsDTO> getPurchaseDetails() { }

public MobileDTO getPurchaseDetailsById() { }

public void updatePurchaseDetails() { }\*/

private static boolean validatePhone(long phoneNo)

{

String phonepattern="[\\d]{10}";

String no=new Long(phoneNo).toString();

if(Pattern.matches(phonepattern, no))

{

return true;

}

else

{

System.out.println(" Phone No maximum 10 digits");

return false;

}

}

private static boolean validateEmail(String emailId)

{

String emailNamePtrn =

"^[\_A-Za-z0-9-]+(\\.[\_A-Za-z0-9-]+)\*@[A-Za-z0-9]+(\\.[A-Za-z0-9]+)\*(\\.[A-Za-z]{2,})$";

if(Pattern.matches(emailNamePtrn, emailId))

{

return true;

}

else

{

System.out.println(" Give a valid email address");

return false;

}

}

private static boolean validateName(String custName)

{

//First Cap max 20;

String pattern="[A-Z][a-z]{0,19}";

if(Pattern.matches(pattern, custName))

{

return true;

}

else

{

System.out.println(" Customer name should start with capital and max length is 20");

return false;

}

}

private static boolean validateMobileId(int mobileId) throws InvalidDataException

{

String pattern="[0-9]{4}";

String mId=new Integer(mobileId).toString();

if(Pattern.matches(pattern,mId ))

{

MobileService ms=new MobileService();

Collection<Integer> midLi=ms.getMobileIds();

if(midLi.contains(mobileId))

{

return true;

}

else

{

System.out.println(" Mobile Id do not exist It should be from following values");

for(int ii:midLi)

{

System.out.print (" "+ii);

}

return false;

}

}

else

{

System.out.println(" Mobile Id Should be only 4 digits");

return false;

}

}

public static boolean validateQty(int purchaseQty,int mobileId) throws InvalidDataException

{

MobileService mobileService=new MobileService();

int qtyInHand=mobileService.getMobileQty(mobileId);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"+qtyInHand);

if((purchaseQty>0)&&(purchaseQty<qtyInHand))

{

return true;

}

else

{

System.out.println(" UR Quantity shouls be greater than zero ");

System.out.println(" In hand Quantity is : "+qtyInHand);

return false;

}

}}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*db.properties\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#Propert File To Test The application

driver=oracle.jdbc.driver.OracleDriver

url=jdbc:oracle:thin:@192.168.224.26:1521:trgdb

password=user1

username=user1

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* DBUtil.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.util;

import java.io.FileInputStream;

import java.io.IOException;

import java.io.InputStream;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.Properties;

import oracle.jdbc.pool.OracleDataSource;

public class DBUtil

{

private static Properties loadProperties(String fileName)

{

Properties dbProp = new Properties();

try

{

InputStream propsFile = new FileInputStream(fileName);

dbProp .load(propsFile);

// System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"+dbProp .getProperty("driver"));

// System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"+dbProp .getProperty("url"));

//System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"+dbProp .getProperty("password"));

//System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"+dbProp .getProperty("username"));

propsFile.close();

}

catch (IOException ioe)

{

ioe.printStackTrace();

}

return dbProp ;

}

public static Connection getConnection()

{

String fn="db.properties";

Properties dbProp=DBUtil.loadProperties(fn);

OracleDataSource ods;

Connection con=null;

try

{

ods = new OracleDataSource();

ods.setDriverType(dbProp .getProperty("driver"));

ods.setURL(dbProp .getProperty("url"));

ods.setUser(dbProp .getProperty("username"));

ods.setPassword(dbProp .getProperty("password"));

con = ods.getConnection();

con.setAutoCommit(false);

}

catch (Exception e)

{

e.printStackTrace();

try

{

con.close();

} catch (SQLException e1)

{

e1.printStackTrace();

}

return null;

}

return con;

}}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* TestMobilePurchaseDemo.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

import java.util.Collection;

import java.util.Scanner;

import java.util.regex.Pattern;

import com.igate.bean.MobileBean;

import com.igate.bean.PurchaseDetailsBean;

import com.igate.dao.PurchaseDetailsDAO;

import com.igate.dao.PurchaseDetailsDAOImpl;

import com.igate.dto.PurchaseDetailsDTO;

import com.igate.exception.InvalidDataException;

import com.igate.service.MobileService;

import com.igate.service.PurchaseDetailsService;

public class TestMobilePurchaseDemo

{

static Scanner sc=new Scanner(System.in);

public static void main(String[] args)

{

int choice;

while(true)

{

System.out.println("What Do U Want To Do?");

System.out.println("\t\t\t 1: INSERT CUSTOMER & Purchase DEATAILS");

System.out.println("\t\t\t 2: View MOBILEs DEATAILS ");

System.out.println("\t\t\t 3: Delete MOBILE DEATAILS ");

System.out.println("\t\t\t 4: SEARCH MOBILE USING RANGE");

System.out.println("\t\t\t 0: EXIT");

System.out.print(" Please Enter Your Choice: ");

choice=sc.nextInt();

switch(choice)

{

case 1:enterCustomerDetails();

break;

default:System.exit(0);

}

}

}

private static void enterCustomerDetails()

{

String emailId;

String custName;

long phoneNo;

int purchaseId;

int mobileId;

int quantity;

boolean testCustFlag = true;

boolean testEmailFlag = true;

boolean testPhoneFlag = true;

boolean testMobileIdFlag=true;

boolean testQtyFlag=true;

Scanner sc = new Scanner(System.in);

//System.out.print("\t\tEnter Purchase Id:");

//purchaseId=sc.nextInt();

while (testCustFlag)

{

System.out.print("\t\t Enter custName :");

custName = sc.next();

System.out.print("\t\t Enter Email Address :");

emailId = sc.next();

System.out.print("\t\t Enter Phone No :");

phoneNo = sc.nextLong();

System.out.print("\t\tEnter Mobile Id :");

mobileId = sc.nextInt();

System.out.print("\t\tEnter Qty:");

quantity = sc.nextInt();

PurchaseDetailsDTO purchaseDetailsDto=new PurchaseDetailsDTO();

purchaseDetailsDto.setCustName(custName);

purchaseDetailsDto.setMailId(emailId);

purchaseDetailsDto.setPhoneNo(phoneNo);

purchaseDetailsDto.setMobileId(mobileId);

purchaseDetailsDto.setQty(quantity);

PurchaseDetailsService purchaseDetailsService=

new PurchaseDetailsService();

try

{

purchaseDetailsService.addPurchaseDetails(purchaseDetailsDto);

}

catch (InvalidDataException e)

{

System.out.println("Some data error");

// e.printStackTrace()); } }}}