|  |
| --- |
|  |
| ONLINE BANKING SYSTEM |
| **Name:** Swarvanu Sengupta  **College:** Seacom Engg. College  **University:** West Bengal University Of Technology  **Department:** Computer Science & Engg. (Btech)   * GROUP MEMBERS: |
|  |
|  |
| ***NAME COLLEGE REG NO*** |

* SWARVANU SENGUPTA (PL) SEACOM ENGINEERING COLLEGE 9904
* SAYANTAN SAMANTA SEACOM ENGINEERING COLLEGE 9983
* SWARNALI BANERJEE SEACOM ENGINEERING COLLEGE 9906
* SK ASFAQUE AHAMED SEACOM EMGINEERING COLLEGE 9989
* SUDIPTA MONDAL SEACOM ENGINEERING COLLEGE 9905

|  |
| --- |
|  |

* ***INDEX***

**ITEM PAGE-NO**

**1>Requirement Specification**

* **Domain Description 4**
* **Problem Definition 5**
* **Functional Requirements 6**
* **Hardware/Software Requirements 7**

**2>Database Design**

* + - **ER Diagram 8**
    - **Table Description 9**

**3>Page Flow Diagram 11**

**4>Screen Shots 12**

**5>Future Scope Of Improvements 21**

**6>Code 22**

* ***ACKNOWLEDGEMENT***

This training and project in GLOBSYN, KOLKATA has been a truly remarkable experience for us. At the outset we would like to express our gratitude to all the members of the GLOBSYN for providing us an opportunity to be a part of the organization and pursue this project.

The preparation of this report would not have been possible without the valuable support of our project guide **A. Velli Raj**. We wish to express our sincere gratitude to our project guide for his guidance and encouragement in carrying out this project work. We also wish to express our gratitude to the other officials and staff members of GLOBSYN who rendered their help during the period of my project work and for their kind co-operation to the successful completion of my project work.

Being a BTECH student of SEACOM ENGINEERING COLLEGE, WEST BENGAL we are grateful to our principal, head of the department and other teaching staffs for their kind co-operation and encouragement in this regard.

Last but not the least we wish to avail ourselves to this opportunity, express a sense of gratitude and love to my beloved parents for their manual support, strength and help.

THANKS TO ALL

* **SWARVANU SENGUPTA (COMPUTER SCIENCE AND ENGINEERING)**
* **SAYANTAN SAMANTA BTECH (INFORMATION TECHNOLOGY)**
* **SWARNALI BANERJEE BTECH (COMPUTER SCIENCE AND ENGINEERING)**
* **SK.ASFAQUE AHAMED BTECH (INFORMATION TECHNOLOGY)**
* **SUDIPTA MONDAL BTECH (COMPUTER SCIENCE AND ENGINEERING)**

*SEACOM ENGINEERING COLLEGE, JALADHULAGORI, SANKRAIL, HOWRAH,WEST BENGAL*

* ***REQUIREMENT SPECIFICATION***

Requirement Analysis is a Software Engineering task that bridges the gap between System level requirement Engineering and Software design. Any computer based system.Development phase will begin with preliminary analysis of the system scope and feasibility. This phase is also known as Survey phase. This phase determines whether or not significant resources to be committed to the other phases of the life cycle. During the survey phase, we will define the scope of the system, including all end-users perceived the problems and opportunities, business and technical constraints, system goals and possible situations.

* **Domain Description:**
* *Banking Domain:*

A bank is a business; banks sell financial services such as vehicle loans, home mortgage loans, business loans, checking accounts, credit card services, certificate of deposits and individual retirement accounts etc.

Some people go to banks in search of a safe place to keep their money. Other are seeking to borrow money to buy a house or a car, start a business, expand a farm, pay for college etc.

* Types of bank:

a>savings bank

b>commercial bank

c>industrial bank

d>land mortgage bank

e>federal/central/national bank

f>co-operative bank

g>consumers bank

* Types of bank account:

a>Savings account

b>Certificates of deposit

c>Checking accounts or current accounts

d>money market deposit accounts

* Types of loans:

a>personal loans

b>home loans

c>tax benefits on home loans

d>bank loans against property

e>business loans

f>car loans

g>educational loans

* Types of deposits:

a>savings deposit

b>current deposit

c>fixed deposit

d>recurring deposit

* **Problem Definition:**

The aim of the proposed system is to address the limitations of the current system .The requirements for the system has been gathered from the defects recorded in the past and also based on the feedback users of the previous metric tools.

***Manual process*:**

Customer----->physically visits the bank----->inquire for an existing service or specific information----->The in charge clerk checks the specification and answers the query----->Associated and integrates the information as needed ----->leaves the bank

***Online process:***

*customer----->physically visits the bank----->praises a request for new checkbook by filling in the prescribed form---->the in charge clerk accepts the request and prepares the cheque book with respect to given specification---->The cheque book is sent for manages initials---->Customer makes a customer sign and receives the checkbook*

* **Functional Requirements:**

The development of the new system contains the following activities which try to automate the entire process keeping in view of the database integration approach.

Following are the Objectives of the proposed system:

**MODULE - 1 : ADMINISTRATOR**

* ***REGISTER AN ACCOUNT AND CUSTOMER :***

• Administrator registers the details of an account and start services. It provide a valid User Id and password to customers. The user id & password are included in the welcome kit given at the time of opening bank account.

• When customer forgets his password and asks for it. Upon successful authentication, the passwords will be sent to the communication address.

• When user login for the first time, to ensure the security of the account, customers are requested to change the password.

* ***CLOSE ACCOUNT :***

• This section allows administrator to block a customer account.

• Then after no transaction is possible with that account.

* ***VIEW DETAILS OF AN ACCOUNT:***

• It provides the facility to view the details of a customer using his account number.

* ***SEARCH:***

• It provides the administrator to search the customer with their name and branch name

**MODULE - 2: CUSTOMERS:**

* **USER ID & PASSWORD:**

• To login to this online system, Customers must have a valid User Id and password.

• Customer can change his password. For this he can click the 'Change Password' link.

* ***ACCOUNT DETAILS:***

• Customer can see his balance in the bank account.

• Customer can see account statement from today to a specified no. of days.

* ***FUND TRANSFER:***

• Customer can transfer the funds to other bank accounts all over India

• He has to request by using account no. and name of the customer in which account transfer will perform.

* **Hardware/software Requirements:**
* **Hardware requirements:**

Client side

|  |  |  |
| --- | --- | --- |
| *Processor* | *RAM* | *Disc Space* |
| Pentium II at 500Mhz | 64 MB | 500 MB |

Server side

|  |  |  |
| --- | --- | --- |
| *Processor* | *RAM* | *Disc Space* |
| Pentium III at 1Ghz | 512MB | 1GB |
| Pentium III at 1Ghz | 512MB | 512MB |

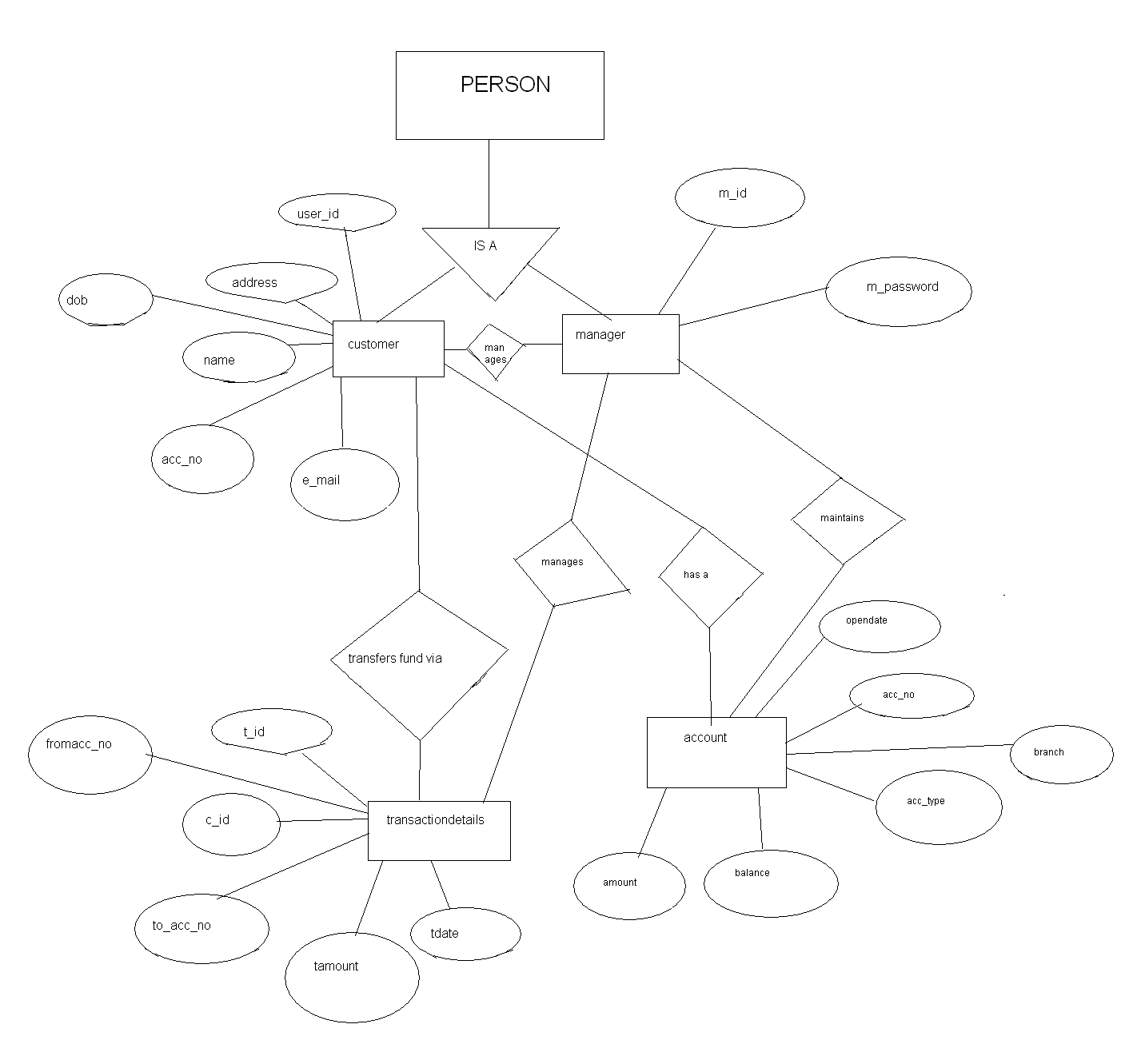
* **Software requirements:**

WebServer- JBOSS 4.2

Front End -JEE,JSP Back End – Oracle 10g

***DATABASE DESIGN***

* **ER Diagram**

****

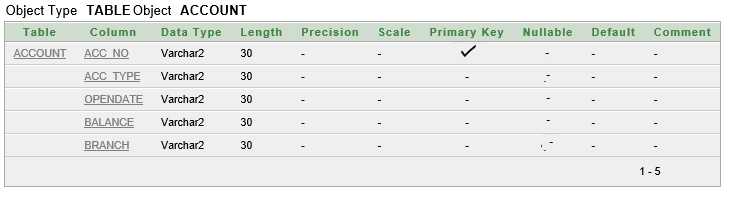
* ***Table Description:***

We have maintained four tables.

We haven’t used any primary and foreign key for better freedom in order to modify, delete the table individually. Although results much more complexity and coding.

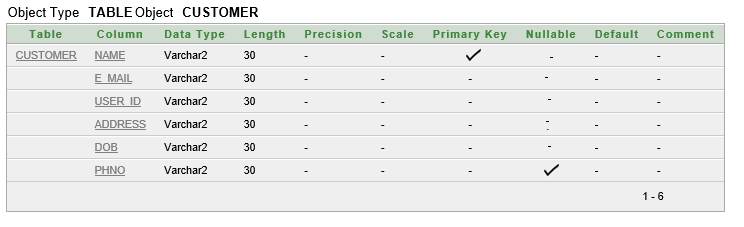
**Account Table**

Account table holding information on an account identified by the unique account no.

****

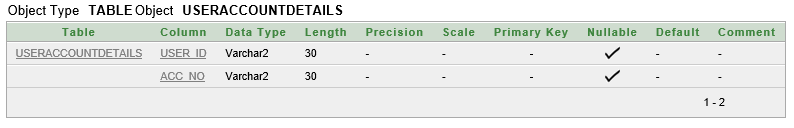
**Customer Table**

Customer table holding information of a customer identified by the unique customer no.

****

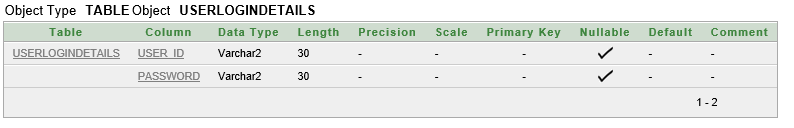
User Account Details Table

Mapping between ‘customer no’ and ‘acc. No.’ Used to identify which customer has which account.

****

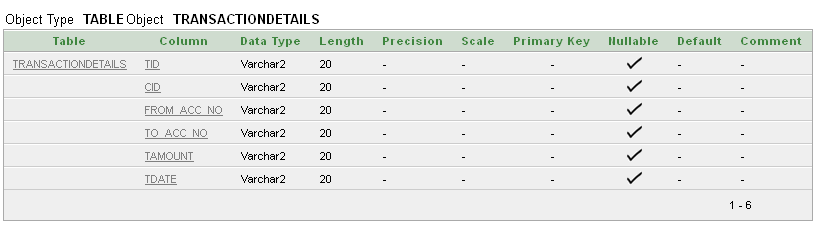
**User Login Details Table**

**Mapping between ‘customer no’ and ‘password’. Used to store customer’s password in a more secure way.**

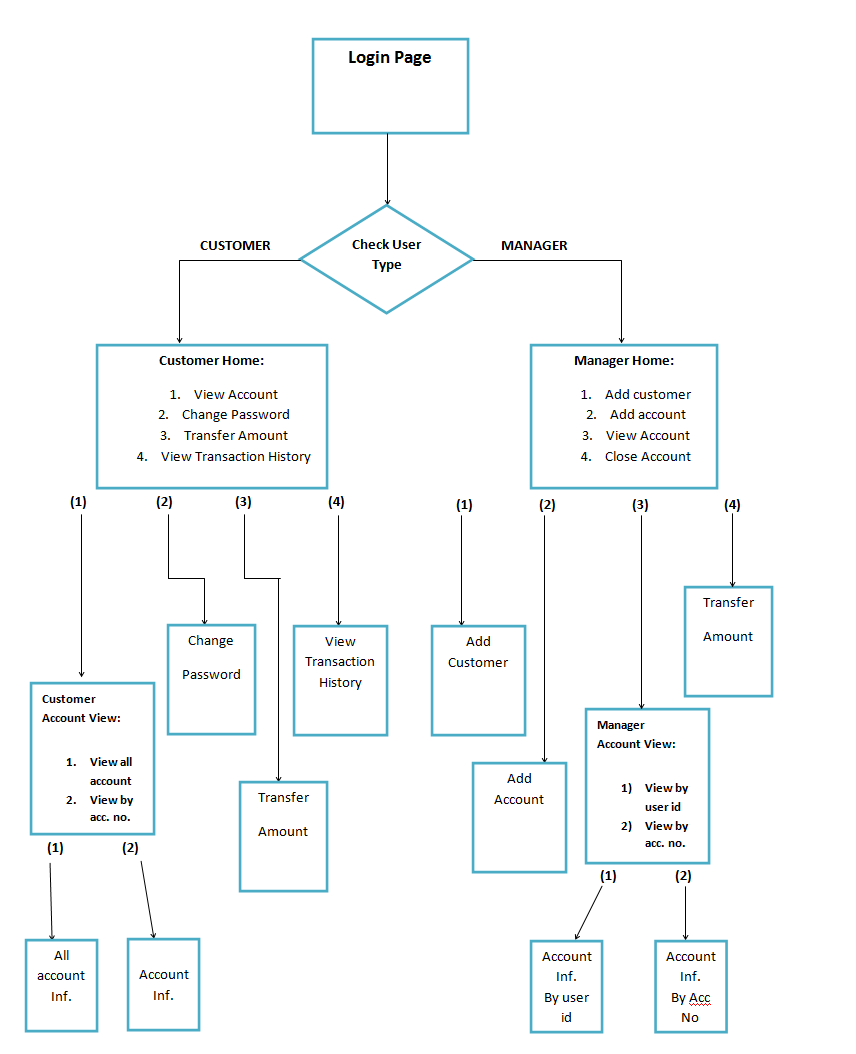
****

**Transaction Details Table**

**Storing the information about every transaction and mapping it with customer id and account no.**

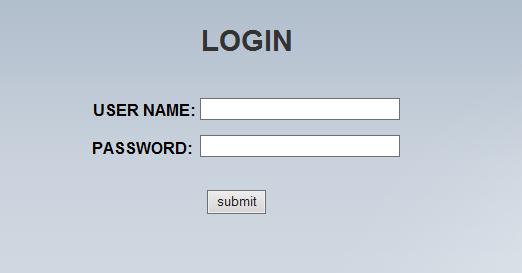
****

* ***PAGE FLOW DIAGRAMS***

****

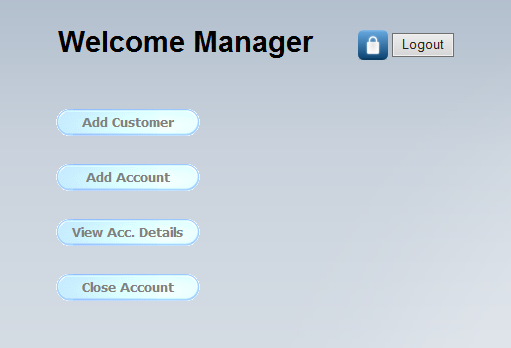
* ***SCREEN SHOTS***

This is the login page(Start page) for both the manager and customer

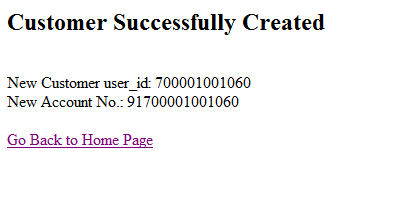
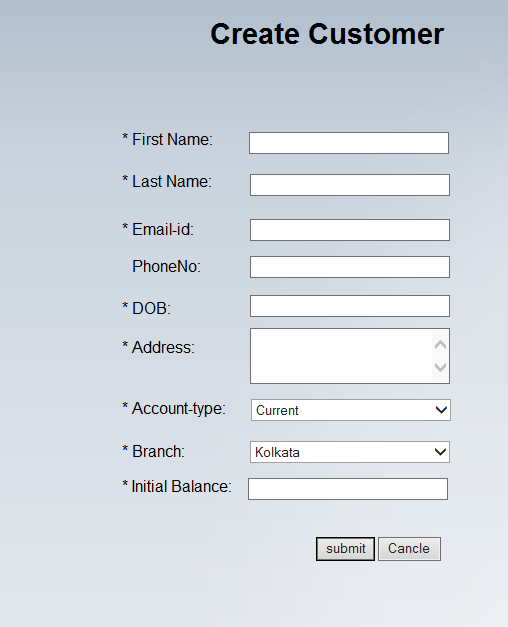


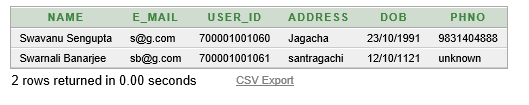
* ***Manager:***

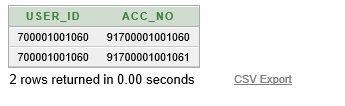
*If we give the manager username and password then the following page will appear*

**

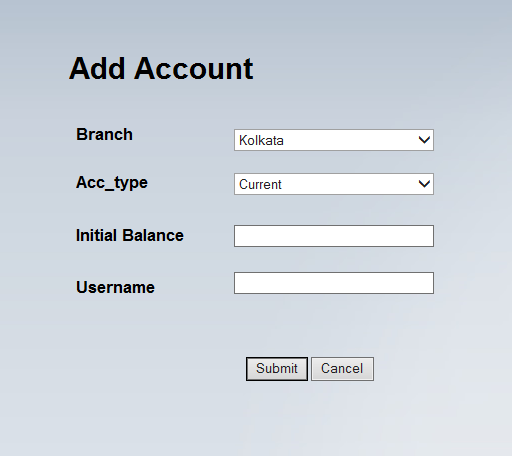
* *On clicking the* ***add customer*** *the following page will appear*

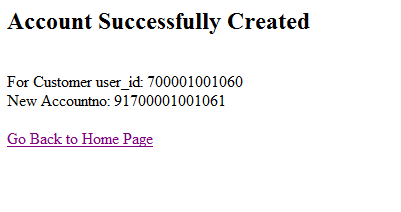
**

*The Modified tables are: *

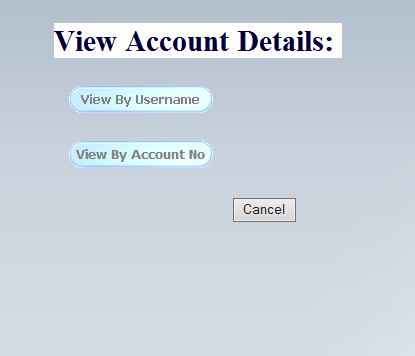
**

* *On clicking the add account option the following page will appear*

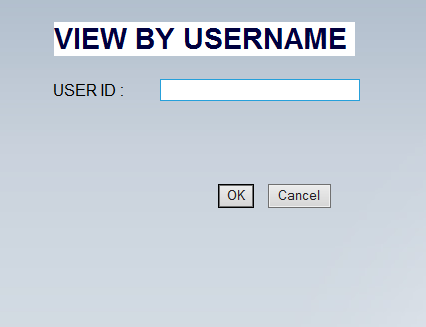
**

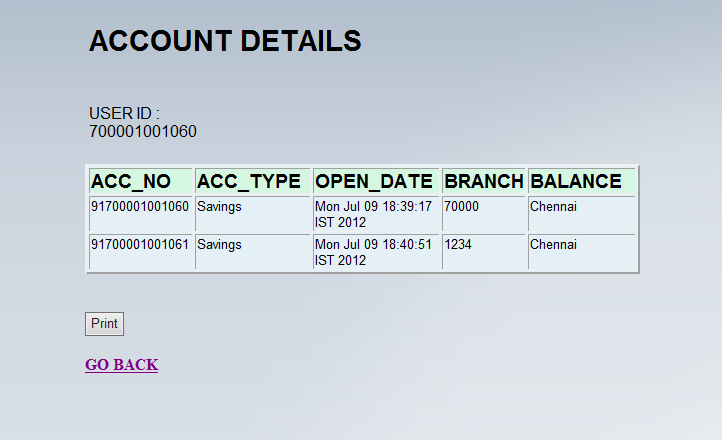
******

* *On clicking the view account details the following page will appear*

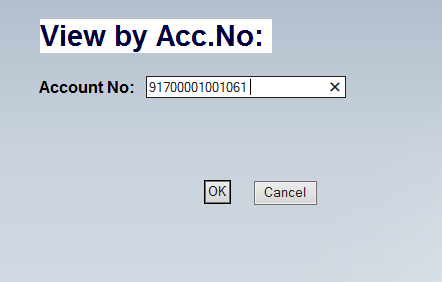
**

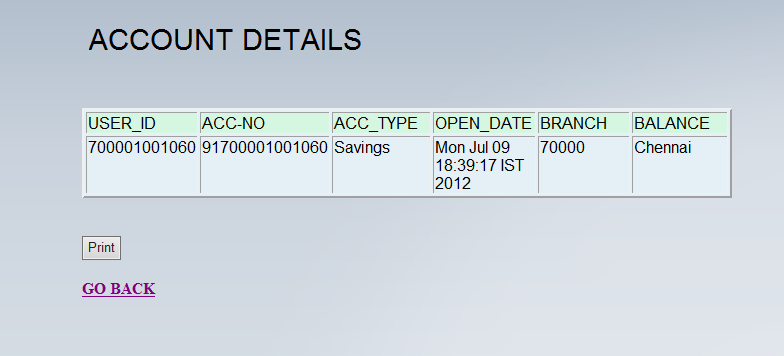
* *On clicking the view by username the following page will appear:*

**

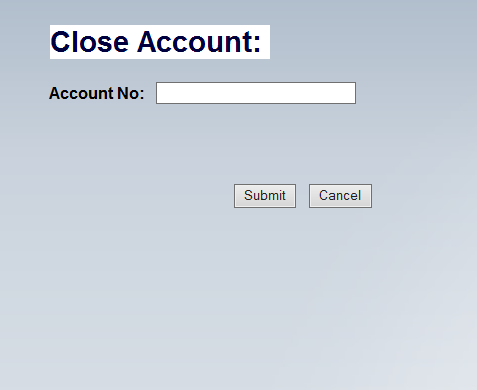
**

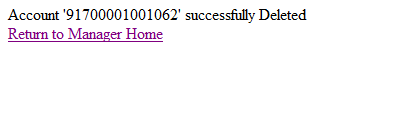
* *On clicking the view by account no the page will appear*

**

**

* *On clicking the close account option the following page will appear*

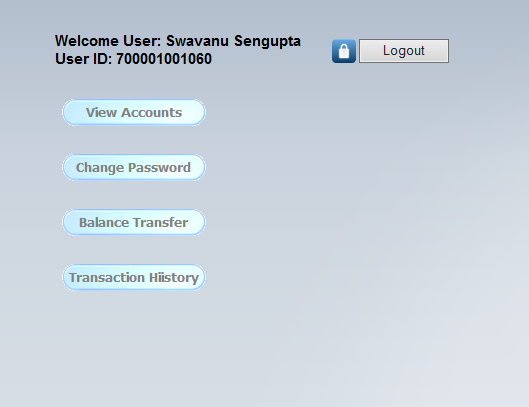
**

**

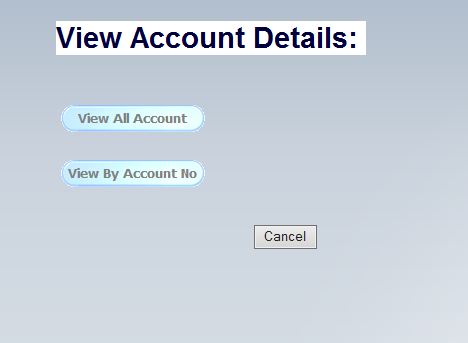
**

* **Customer:**

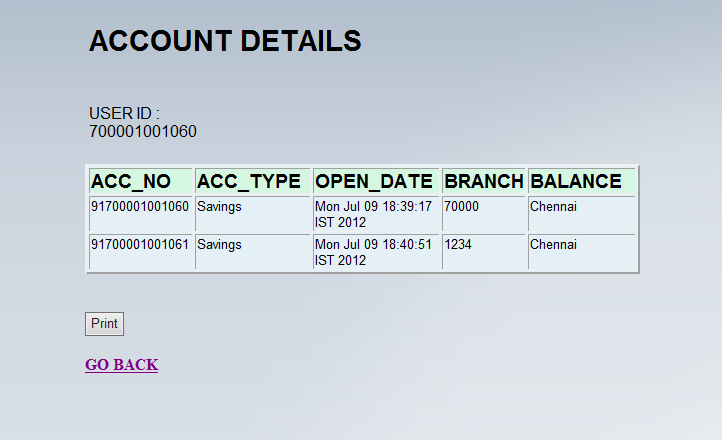
*If we give the Customer username and password then the following page will appear*

**

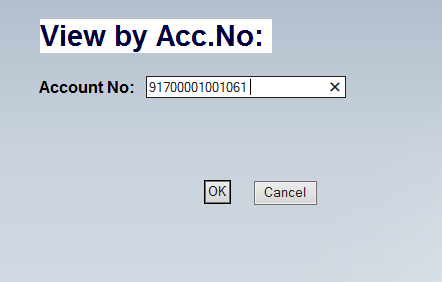
* *On clicking the view accounts option the following page will appear*

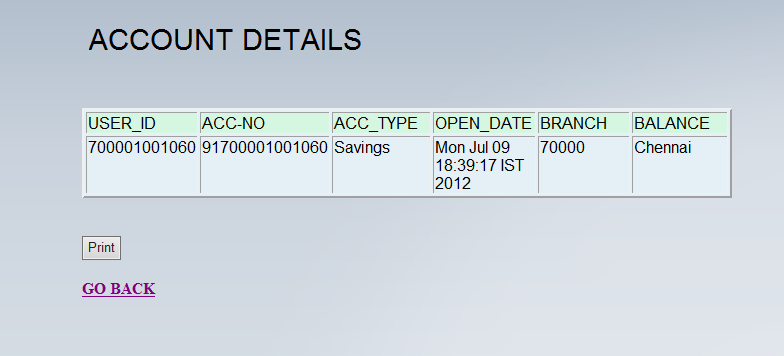
**

* *On clicking the view all account option the following page will appear*

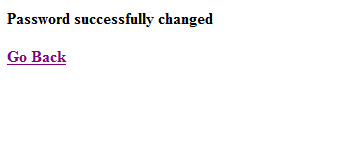
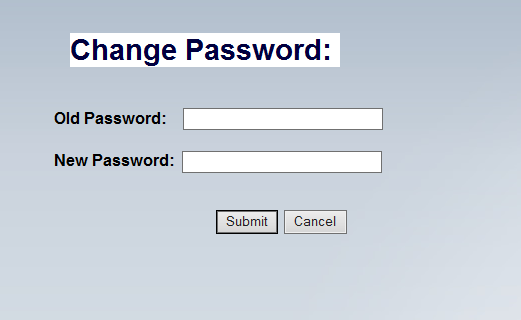
**

* *On clicking the view by account no option the following page will appear*

**

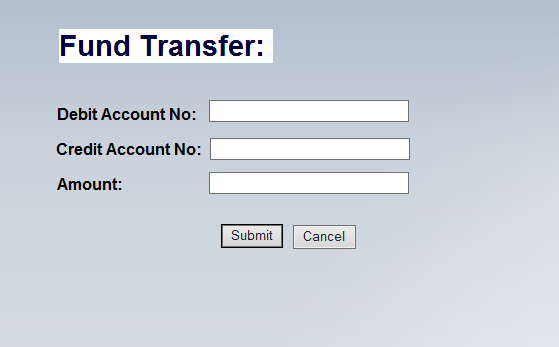
**

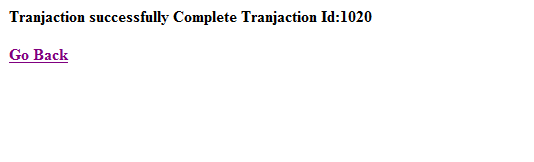
* *On clicking the change password the following page will appear*

**

|  |  |
| --- | --- |
| *Before using this functionality* | *After using this functionality* |
| *H:\login\before change.PNG* | *H:\login\afterpasschange.PNG* |

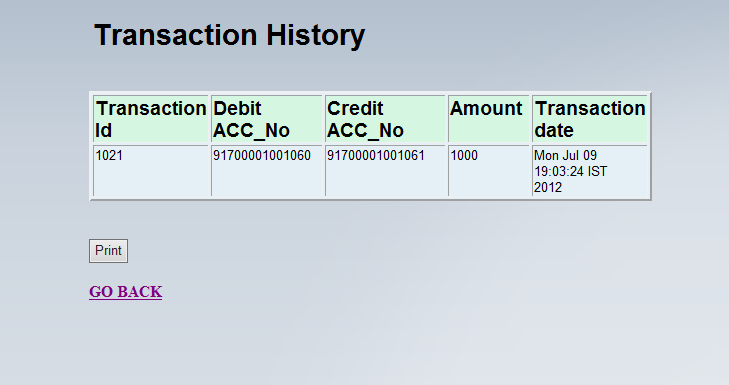
* On clicking the balance transfer option the following page will appear



**

|  |  |
| --- | --- |
| *Before transaction* | *After transaction* |
| *H:\login\beforetranjuction.PNG* | *H:\Capture.PNG* |
| *H:\login\transtable.PNG* | |

* *On clicking the transaction history the following page will appear*

**

* ***FUTURE SCOPE OF IMPROVEMENTS***

***Feature Oriented:***

* In future it will be multiuser. That is multiple user can login at a time.
* Cash withdrawal record can be maintained.
* Record of loans can also be kept.
* Online bill payment
* Forgot password features

***Technical Oriented:***

* Use of SQL keys
* Ultimately the software may lead to a complete automation system.
* We are looking forwarding to provide much security.
* Better UI using modern technology.
* ***CODE***

In the coding section of the project we followed some general rules and condition to make the code more readable and user-friendly.

We also tried to keep it small, simple and tried hardly to reuse the code as much as possible.

* **Some General Views we followed:**

Servlet

JSP

Browser

Browser

CONNECTION

POJO

DAO

Servlet

* **Design:**
* We used CSS (Cascading Style Sheet) for better Look and Feel.
* Interactive User Interface By java Script.
* Alternative rollover images etc.

**Style Sheet (Inline & Embedded)**:

* **Embedded:**

<style type="text/css">

.ws6 {font-size: 8px;}

.ws7 {font-size: 9.3px;}

.ws8 {font-size: 11px;}

.ws9 {font-size: 12px;}

.ws10 {font-size: 13px;}

.ws11 {font-size: 15px;}

.ws12 {font-size: 16px;}

.ws14 {font-size: 19px;}

.ws16 {font-size: 21px;}

.ws18 {font-size: 24px;}

.wpmd {font-size: 13px;font-family: Arial,Helvetica,Sans-Serif;font-style: normal;font-weight: normal;}

/\*----------Para Styles----------\*/

DIV,UL,OL /\* Left \*/

{

margin-top: 0px;

margin-bottom: 0px;

}

</style>

* **Inline:**

<div id="text1" style="position:absolute; overflow:hidden; left:83px; top:43px; width:284px; height:48px; z-index:0">

<div class="wpmd">

<div><font class="ws11"><B><% out.println(heading); out.println(CheckMethods.getnamebyuserid(uid)); %>

</br>

Name :

</B></font></div>

</div></div>

**Rollover Images:**

<div id="nav10d" style="position:absolute; left:90px; top:110px; z-index:1"><a onMouseOut="MM\_swapImgRestore()" onMouseOver="MM\_swapImage('nav10','','resource/images/customerhome/nav24716330a.gif',1)" href="cviewaccount.html"><img name="nav10" alt="" border=0 src="resource/images/customerhome/nav24716330i.gif"></a></div>

**Interactive User Interface by JavaScript**:

<script type="text/javascript">

functiongoBack()

{

window.location="managerhome.html";

}

**Validation Checking:**

function validate()

{

var x = document.getElementById("ib").value;

if(x==null || x=="")

{

alert("Initial Balence must be filled out");

return false;

}

x = document.getElementById("un").value;

if(x==null || x=="")

{

alert("User-Name/User-Id must be filled out");

return false;

}

}

</script>

**Auto generated jsp Design:**

while(rs1.next())

{%>

<trvalign=top>

<TD width=100><div class="wpmd">

<div><font class="ws12"><% out.println(rs1.getString("user\_id")); %></font></div>

</div>

</TD>

<% while(rs2.next()){ %>

<TD width=100><div class="wpmd">

<div><font class="ws12"><% out.println(acno); %></font></div>

</div>

</TD>

<TD width=100><div class="wpmd">

<div><font class="ws12"><% out.println(rs2.getString("opendate")); %></font></div>

</div>

</TD>

<TD width=100><div class="wpmd">

<div><font class="ws12"><% out.println(rs2.getString("branch")); %></font></div>

</div>

</TD>

<%} %>

</tr>

<% }%>

* **Implementation of backend Coding:**

Let us describe the process of Account Add

**Controller:**

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

boolean status = CheckMethods.checkunincustomer(request.getParameter("username"));

PrintWriter pw = response.getWriter();

if(status==true)

{

AccountInf account = new AccountInf();

account.setBranch(request.getParameter("branch"));

account.setAcc\_type(request.getParameter("acctype"));

account.setBalance(request.getParameter("initbalance"));

account.setUser\_id(request.getParameter("username"));

account.setOpendate(Calendar.getInstance().getTime().toString());

String accNo = InsertMethods.addAccount(account);

if(accNo == null) {pw.println("Error in AddAccount with user\_id:" + request.getParameter("username"));

}

else

pw.println("<h2>Account Successfully Created</h2> </br> For Customer user\_id: " + request.getParameter("username") + "</br>" + "New Accountno: " + accNo);

}else{pw.println("<h2>Error: This customer (" + request.getParameter("username") +") does'nt exist</h2>");}

pw.println("</br></br><a href=\"managerhome.html\">Go Back to Home Page</a>");

}

Check whether Customer id is valid or not.

**POJO class:**

package entity\_pojo;

public class AccountInf {

private String acc\_no, acc\_type, opendate, balance, branch, user\_id;

public String getAcc\_no() {

return acc\_no;

}

public void setAcc\_no(String acc\_no) {

this.acc\_no = acc\_no;

}

public String getAcc\_type() {

return acc\_type;

}

public void setAcc\_type(String acc\_type) {

this.acc\_type = acc\_type;

}

public String getOpendate() {

return opendate;

} ……….

**Database Access Object class:**

public static String addAccount(AccountInf account)

{

try{

Connection conn = InitConnection.getConnection();

PreparedStatement pst=conn.prepareStatement(AllStatement.insertainf); //acc\_type,opendate,balance,branch

pst.setString(1, account.getAcc\_type());

pst.setString(2, account.getOpendate());

pst.setString(3, account.getBalance());

pst.setString(4, account.getBranch());

int updtcnt = pst.executeUpdate();

pst.close();

if(updtcnt!=0)

{

//Getting the accno

pst = conn.prepareStatement(AllStatement.getan);

ResultSet rs = pst.executeQuery();

while(rs.next())

{

account.setAcc\_no(rs.getString("acc\_no"));

}

pst = conn.prepareStatement(AllStatement.insertunacno);

pst.setString(1, account.getUser\_id());

pst.setString(2, account.getAcc\_no());

updtcnt = pst.executeUpdate();

if(updtcnt!=0)

return account.getAcc\_no();

else

throw new Exception("updateCount to table useraccountdetails is '0'");

}

else{throw new Exception("updateCount to table account is '0'");}

}catch(Exception e){System.out.println("Error in InsertMethods-addAccount: " + e.getMessage());return null;}

}

* **Extra code for Reusability and Modeling:**
* **Hidden field and dynamic checking:**

<form id="hiddenform" method="get" action="Logout" style="margin:0px">

<input type="hidden" name="sname" id="sn" value="Name"></input>

</form>

---------------------------------------------------------------------------------------------------------------

function logOut()

{

document.forms["hiddenform"].submit();

}

-----------------------------------------------------------------------------------------------------------------

<% String uid = request.getParameter("uid");

int pagestat = 0;

if(uid==null || uid=="")

{

uid = session.getAttribute("Name").toString();

pagestat = 1;

}

String query1 = "select acc\_no from useraccountdetails where user\_id =?";

String query2 = "select \* from account where acc\_no =?";

%>

* **Synchronized transaction :**

**Synchronized** {

System.out.println("Before");

boolean status1 = InsertMethods.transferAmount(dacno,cacno,Integer.parseInt(amount));

System.out.println("middle");

String status2 = InsertMethods.addfundTransferdetailsintable(uid,dacno,cacno,amount);

System.out.println("after");

if(status1==true && status2!=null)

{

pw.println("<b>Tranjaction successfully Complete</b>");

pw.println("<b>Tranjaction Id:" + status2 +"</b>");

}

else

pw.println("<b>Problem in : InsertMethods.transferAmount</b>");}

* **All types of checking :**

if(dacno.equals(cacno) != true)

{

if(CheckMethods.checkuidacnopair(uid, dacno))

{

System.out.println("Enter");

if(CheckMethods.checkacnoinaccount(cacno))

{

Checking if both acno is same

Debit account has enough balance

Credit account does exist and belongs to whome

* **Single class Query Definition:**

package dao;

class AllStatement {

//All check statement

//Customer inf

static String checkunpass= "select \* from userlogindetails where user\_id=? and password=?";

static String checkunacno= "select user\_id from useraccountdetails where acc\_no =?";

static String checkacnoun= "select \* from useraccountdetails where user\_id =?";

static String getun = "select user\_id from customer";

static String getan = "select acc\_no from account";

static String checkunincustomer = "select \* from customer where user\_id =?";

static String checkacnoinaccount = "select \* from account where acc\_no =?";

static String getpassbyuid = "select password from userlogindetails where user\_id=?";

//Account inf:

static String getacbyuid = "select acc\_no from account where user\_id =?";

static String checkuidacnopair = "select \* from useraccountdetails where user\_id=? and acc\_no=?";

static String getamountbyacno = "select balance from account where acc\_no =?";

//All insertion statement

//Customer inf

static String insertcinf = "insert into customer(name,e\_mail,user\_id,address,dob,phno) values(?,?,unseq.nextval,?,?,?)";

static String insertunpass = "insert into userlogindetails(user\_id,password) values(?,?)";

//Account inf

static String insertainf = "insert into account(acc\_no,acc\_type,opendate,balance,branch) values(anseq.nextval,?,?,?,?)";

static String insertunacno = "insert into useraccountdetails(user\_id,acc\_no) values(?,?)";

//Transaction inf

static String insertninf = "insert into transactiondetails values(tnseq.nextval,?,?,?,?,?)";

static String gettid = "select tid from transactiondetails";

//All delete statements

//Account inf

static String deleteainf = "delete from account where acc\_no =?";

static String deleteunacno = "delete from useraccountdetails where acc\_no =?";

//Customer inf

static String deletecinf = "delete from customer where user\_id =?";

static String deleteunpass = "delete from userlogindetails where user\_id =?";

//All update statements

//Customer inf

static String updatepass = "update userlogindetails set password =? where user\_id =?";

//Account inf

static String updatebalance = "update account set balance =? where acc\_no =?";

}

* **Use of some Typical Function:**
* **Account Open Date:**

account.setOpendate(Calendar.getInstance().getTime().toString());

* **Checking The Init parameter for Manager Username and Pass:**

public void init(ServletConfig config) throws ServletException {

// TODO Auto-generated method stub

super.init(config);

adminname=this.getInitParameter("adminuname");

adminpass=this.getInitParameter("adminpass");

}

String senduid = request.getParameter("uname");

String sendpass = request.getParameter("pwrd");

HttpSession session = request.getSession(true);

if(senduid.equals(adminname) && sendpass.equals(adminpass))//Checking if manager

{

response.sendRedirect("managerhome.html");

}

* **Request forwording:**

if(CheckMethods.checkacnoinaccount(acno))

{

request.getRequestDispatcher("accountdetailsbyacno.jsp").forward(request, response);

}

* ***CONCLUSION:***

As the world becomes more of a global village being runned by paperless systems, the idea of a cashless society is the hope of the future. Thus more innovations will still evolve which will make cashless transactions easily accessible and affordable.

****

**Certificate**

This is to certify that **Mr Swarvanu Sengupta** of **Seacom Engineering College**, registration no **9904** has successfully completed a project on **Online Banking System** using JEE under the guidance of **A.Velli Raj**.

*A.Velli Raj*

Globsyn Finishing School