INTERNET BANKING SYSTEM

Computer Science
Honours Documentation 2012
Author: Mohamed Hassan Ali
Student Number: 3270152

Supervisor: Mr. Michael Norman
Department of Computer Science

A mini-thesis submitted in partial fulfillment of the requirements for the degree of
B.Sc. Honours.
ABSTRACT

The adoption of Electronic Banking by commercial enterprise has been in existence since the mid 90s, much greater in number due to lower operating costs associated with it. Electronic banking has been in the form of automatic teller machines and telephone transactions.

Internet banking system services can include: Open an account, Balance enquiry, Request for Cheque book, Beneficiary payments (EFT), Viewing monthly.

More recently, it has been transformed by the Internet, a new delivery channel for banking services that benefits both customers and banks.

Furthermore, customer’s application for electronic banking facilities is expanding as the cost savings on Transactions over the Internet are significant.
I, Mohamed Hassan Ali, certify that this project is my own work. I understand what plagiarism is and I have used quotations and references to fully acknowledge all the words and ideas of others, which we have used in our project. I have not copied anyone else's project. I have also not permitted anyone to copy my project.

Signature: _ _Mohamed Hassan Ali_ _
ACKNOWLEDGEMENTS

First and foremost I am ever grateful to my Allah to whom I owe my life. I would also like to thank my parents for giving me the opportunity to study at university.

I am wholeheartedly grateful to my supervisor Mr. Michael Norman for guiding me to reach my initial milestones in the first semester.
# TABLE OF CONTENT

INTERNET BANKING SYSTEM ........................................................................................................... i
ABSTRACT ............................................................................................................................................ i
PLAGIARISM DECLARATION ........................................................................................................... ii
ACKNOWLEDGEMENTS ................................................................................................................... iii
TABLE OF CONTENT ....................................................................................................................... iv
LIST OF FIGURES ............................................................................................................................ vi
LIST OF TABLES ................................................................................................................................ vii
LIST OF ACRONYMES ..................................................................................................................... viii

Chapter 1 ........................................................................................................................................... 1

  user's requirements document ......................................................................................................... 1
    1.1 Background ................................................................................................................................ 1
    1.2 Problem Statement .................................................................................................................... 1
    1.3 Scope of the Study ..................................................................................................................... 1
    1.4 Limitations of the Internet Banking System ............................................................................. 2

Chapter 2 ........................................................................................................................................... 3

  requirements analysis document ....................................................................................................... 3
    2.1 Functional Requirements ......................................................................................................... 3
    2.2 Non-functional Requirements .................................................................................................. 3
    2.3 Class Diagram .......................................................................................................................... 3
    2.4 System Requirements .............................................................................................................. 4

Chapter 3 ........................................................................................................................................... 5

  User interface specification .............................................................................................................. 5
    3.1 Description of the complete user interface ............................................................................. 5
    3.2 What the user interface looks like to the user ......................................................................... 5
    3.3 How the user interface behaves .............................................................................................. 8
    3.4 How the user interacts with the system .................................................................................... 8
    3.5 How the admin interface looks like ........................................................................................... 9

Chapter 4 ........................................................................................................................................... 11

  High Level Design (Object Oriented Analysis) .............................................................................. 11
4.1 Data Dictionary ........................................................................................................................................... 11
4.2 Entity-Relationship Model ......................................................................................................................... 13

Chapter 5 .......................................................................................................................................................... 14

LOW LEVEL DESIGN (OBJECT ORIENTED DESIGN) ..................................................................................... 14
5.1 Event Diagram ........................................................................................................................................... 14
5.2 Algorithmic Description .............................................................................................................................. 15

Chapter 6.......................................................................................................................................................... 17

IMPLEMENTATION .......................................................................................................................................... 17

BIBLIOGRAPHY ................................................................................................................................................ 35
### LIST OF FIGURES

- **Figure 1**: Use-case Diagram of Internet Banking ........................................ 2
- **Figure-2** Class Diagram of Internet banking system ................................. 4
- **Figure-3** Registration form and opening a new account ............................. 6
- **Figure-4** Home Page .............................................................................. 7
- **Figure-5** Transaction Page ....................................................................... 7
- **Figure-6** Create a new account .................................................................. 8
- **Figure-7** Login Process ............................................................................ 9
- **Figure-8** Admin Page ................................................................................ 9
- **Figure-9** Event Diagram .......................................................................... 13
LIST OF TABLES

Table 1: System Requirements. ........................................... 3
Table 2: Login Table. ......................................................10
Table 3: Clients Table. .................................................... 10
Table 4: Account Table. ...................................................11
Table 5: Transaction Table. ..............................................11
LIST OF ACRONYMES

CSS- Cascading Style Sheets

EFT- Electronic Funds Transfer

HTML- Hypertext Mark-up Language

Internet Banking System- A system allowing individuals to perform banking activities at home, via the internet.

MYSQL- is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases.

RAD- Requirements Analysis Document

SSL- Secure Socket Layer

PHP- Hypertext Pre-processor

URD- User Requirements Document
Chapter 1

USER'S REQUIREMENTS DOCUMENT

1.1 Background
Internet Banking System refers to systems that enable bank customers to Access accounts and general Information on bank products and services through a personal computer or other intelligent device. The chances and threats that the internet symbolizes is no longer news to the present day banking sector. No traditional bank would dare face investment analysts without an Internet strategy. The main intention behind the commencement of electronic banking services is to provide the customers with an alternative that is more responsive and with less expensive options. With options just a click away, customers have more control than ever. Their expectations are usability and real-time answers. They also want personal attention and highly customized products and services.

1.2 Problem Statement
We got user requirements from some Computer Science students, Zukilla Roro, Micheal Motlhabi, Yasser Buchana, and friends Allen Mwangonde, Ismail, from which we formulated the document analysis in February 2012.

Internet banking identifies a particular set of technological solutions for the development and the distribution of financial services, which rely upon the open architecture of the Internet. With the implementation of internet banking system, it maintain a direct relationship with the end users via the web and are able to provide a personal characterization to the interface, by offering additional customized services.

1.3 Scope of the Study
The scope of this project is limited to the activities of the operations unit of the banking system which includes opening of Account, Deposit of funds, Electronic funds transfer, Cheque balance and Monthly statement.
In the figure below, is the use-case diagram of the Internet banking system that the customer can expect all those functions with the bank manager acceptance.

Figure-1 Use-case Diagram of showing user requirements

1.4 Limitations of the Internet Banking System

- **Problems of security**: Various sites are not properly locked at to ensure whether the customer’s money is safe in cyber world or not.
- **Wrong assumption**: Many people are afraid using Internet Banking because of the assumption that it is more expensive than the traditional method of dealing with bank transactions. They still prefer going to bank to perform transactions.
- **Lack of awareness**: Another great hindrance is lack of awareness because effective and wide media efforts in publishing Internet Banking need to be emphasized.
Chapter 2

Requirements Analysis Document

2.1 Functional Requirements

- Customer can request details of the last ‘n’ number of transactions he has performed on any account.
- Customer can make a funds transfer to another account in the same bank.
- Customer can request for cheque book
- Customer can view his monthly statement. She/he can also take print out of the same.
- Customer can make EFT’s to accounts at their and other banks.
- The system is providing balance enquiry facility.

2.2 Non-functional Requirements

Those requirements which are not the functionalities of a system but are the characteristics of a system are called the non-functionalities.

- Secure access of confidential data. SSL can be used.
- 24X7 availability
- Better component design to get better performance at peak time
- Flexible service based architecture will be highly desirable for future extensions.

2.3 Class Diagram

The below class diagram shows that the customer can have more than one account and that relationship goes to one-many relationship.

The transaction functions always depends on the web service, which means it’s a web based.
2.4 System Requirements

<table>
<thead>
<tr>
<th>Software Requirements</th>
<th>Hardware Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System: Windows or linux or MAC</td>
<td>Processor: any</td>
</tr>
<tr>
<td>User Interface: HTML, CSS</td>
<td>Hard Disk: 10 GB minimum</td>
</tr>
<tr>
<td>Programming Language: PHP</td>
<td>RAM: 256MB or more</td>
</tr>
<tr>
<td>Database: MYSQL</td>
<td>Any Screen</td>
</tr>
</tbody>
</table>

Table 1: System Requirements
Chapter 3

USER INTERFACE SPECIFICATION

The purpose of this document is to provide a detailed specification of the Internet Banking System user interface. These requirements will detail the outwardly observable behavior of the program. The user interface provides the means for the user, to interact with the program. This User Interface Specification is intended to convey the general idea for the user interface design and the operational concept for the software. This document will be updated with additional detail as our analysis and design activities progress.

Section 2.5 gives a description of the complete user interface, section 2.6 shows what the user interface looks like to the user, section 2.7 tells how the interface behaves and section 2.8 tells how the user interacts with the system.

3.1 Description of the complete user interface
The User Interface Specification (UIS) consists of one main graphical user interface (GUI), which consists with different operations enlisted in the options.

3.2 What the user interface looks like to the user
The customer must first register then only the customer can open a new account in the system. He/She must fill all the details in the below form, as well as choose a password in order to login after the registration.
Figure-3 Registration form and opening a new account
The Login page consists of two text boxes, namely Account No and Password, and a login command button allowing the customer to log into the system. The login page helps the customers to login as a user who visualizes and analyze data contained in the database.

![Figure-4 Home Page](image)

Once logged on, the customer is ready to perform the transactions.

![Figure-5 Transaction Page](image)
3.3 How the user interface behaves
The system verifies customers input in the field of account no and password, and displays an error message if the customer enters incorrect information. Thus, if the customer provides an appropriate data, then he will be allowed to logged in.

3.4 How the user interacts with the system
The sequence diagram shows how the customer can open an account as well as how to register the internet banking system in order to login the system. When the customer submit all the details in the form then the system automatically gives an account and sends to the database.

Create a new Account as well as registration

![Sequence Diagram]

Figure-6 Create a new account
Then the login process is shown below, the customer enters a valid account number and password then the system checks if it is correct input or not, if it is correct then it allows to access for the transactions, if it is not correct it will remain the home page.

**Login Process**

![Login Process Diagram]

Figure-7 Login Process

### 3.5 How the admin interface looks like

The administrator should login to perform the transactions like to approve or disapprove a loan and so on.
Figure-8 AdminPage
Chapter 4

HIGH LEVEL DESIGN (OBJECT ORIENTED ANALYSIS)

This chapter presents the object oriented view of the system, analysis of the high level design and describes the objects needed to implement the system. Each one of these objects is described and documented, and a data dictionary providing details of each object is provided.

4.1 Data Dictionary

Table Name: LOGIN

**Description:** This table is used to store Login details.

<table>
<thead>
<tr>
<th>Key</th>
<th>Field Name</th>
<th>Data Type</th>
<th>Length</th>
<th>Nullable</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
<td>ACCOUNTNO</td>
<td>VARCHAR</td>
<td>12</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>PASSWORD</td>
<td>VARCHAR</td>
<td>45</td>
<td>NO</td>
</tr>
</tbody>
</table>

Table 2: Login Table

Table Name: CLIENTS

**Description:** This table is used to store customer details.

<table>
<thead>
<tr>
<th>Key</th>
<th>Field Name</th>
<th>Data Type</th>
<th>Length</th>
<th>Nullable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NAME</td>
<td>VARCHAR</td>
<td>45</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>SURNAME</td>
<td>VARCHAR</td>
<td>45</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>INITIAL</td>
<td>VARCHAR</td>
<td>10</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>ACCOUNTTYPE</td>
<td>VARCHAR</td>
<td>45</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>SEX</td>
<td>VARCHAR</td>
<td>6</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>D.O.B</td>
<td>DATE</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>ADDRESS</td>
<td>VARCHAR</td>
<td>200</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>MOBILENO</td>
<td>VARCHAR</td>
<td>10</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>TELEPHONENO</td>
<td>VARCHAR</td>
<td>10</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>EMAIL</td>
<td>VARCHAR</td>
<td>45</td>
<td>NO</td>
</tr>
<tr>
<td>PK</td>
<td>ID_PASSPORT</td>
<td>VARCHAR</td>
<td>45</td>
<td>YES</td>
</tr>
</tbody>
</table>

Table 3: Clients Table
Table Name: ACCOUNT

Description: This table is used to store account details.

<table>
<thead>
<tr>
<th>Key</th>
<th>Field Name</th>
<th>Data Type</th>
<th>Length</th>
<th>Nullable</th>
</tr>
</thead>
<tbody>
<tr>
<td>FK</td>
<td>ACCOUNTNO</td>
<td>VARCHAR</td>
<td>12</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>ACCOUNTTYPE</td>
<td>VARCHAR</td>
<td>45</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>ACCOUNTHOLDER</td>
<td>VARCHAR</td>
<td>45</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>DATEOPENED</td>
<td>DATE</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>BRANCHCODE</td>
<td>INT</td>
<td>5</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>DATEAPPROVED</td>
<td>DATE</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>ACCOUNTBALANCE</td>
<td>DECIMAL</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>APPROVED</td>
<td>VARCHAR</td>
<td>6</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>DISAPPROVED</td>
<td>VARCHAR</td>
<td>6</td>
<td>NO</td>
</tr>
</tbody>
</table>

Table 4: Account Table

Table Name: TRANSACTION

Description: This table is used to store the transaction details.

<table>
<thead>
<tr>
<th>Key</th>
<th>Field Name</th>
<th>Data Type</th>
<th>Length</th>
<th>Nullable</th>
</tr>
</thead>
<tbody>
<tr>
<td>FK</td>
<td>ACCOUNTNO</td>
<td>VARCHAR</td>
<td>12</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>TRANSACTIONID</td>
<td>INT</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>TYPEOFTRANSACTION</td>
<td>VARCHAR</td>
<td>45</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>TRANSACTIONDATE</td>
<td>DATETIME</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>REFERENCE</td>
<td>VARCHAR</td>
<td>45</td>
<td>NO</td>
</tr>
</tbody>
</table>

Table 5: Transaction Table
4.2 Entity-Relationship Model
Chapter 5

LOW LEVEL DESIGN (OBJECT ORIENTED DESIGN)

This chapter presents the object oriented design of the system, analysis of the low level design and provides details for the object oriented analysis of the system.

5.1 Event Diagram
The diagram below indicates the customer connects to the internet to perform all the transactions after he logged in successfully then the information will receive the server to maintain the requirements, and it will send a copy of the data to the database and vice versa.

Figure-8 Event Diagram
5.2 Algrothmic Description

Registration and opening new account:

Function register()

{
GetCustomer_information(name,surname....)
Valid = CheckInformation()
If (Valid) then {
    Accountnumber= Generate_AccountNum()
    Insert(Accountnumber,name,...)
    Display (success)
}
Else
{
    Display_error (message)
}

Login_process()

Get_CustomerAuthentication(Accountnumber && password)
If (Accountnumber && password=correct) then
{
    Display (transactions)
}
Elseif (Accountno && password=wrong) then
{
    Display (Account Number or password are mismatched)
}
} 
Else 
{
    Display (Register now)
}

Viewing_Balance() 
Login_process() 
Display (AccountBalance)

Beneficiary() 
Beneficiary_process() 
If (AccountBalance=sufficient) 
{
    Make (payment) 
    Display (Update_Account_balance)
}
Else 
{
    Display (insufficient)
}
Chapter 6

IMPLEMENTATION

This code is for transaction page, it contains html, php, and javascript.

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="stylesheet" href="style.css" type="text/css" media="all" />
<link href="css.css" rel="stylesheet" type="text/css" />
<link rel="stylesheet" href="css/validationEngine.jquery.css" type="text/css" media="screen" title="no title" charset="utf-8" />
<link rel="stylesheet" href="css/template.css" type="text/css" media="screen" title="no title" charset="utf-8" />
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.3/jquery.min.js" type="text/javascript"></script>
<script src="js/jquery.validationEngine-en.js" type="text/javascript"></script>
<script src="js/jquery.validationEngine.js" type="text/javascript"></script>
<title>Internet Banking</title>
<link href="css.css" rel="stylesheet" type="text/css" />
<script type="text/javascript">

function postTodatabase(){

if (window.XMLHttpRequest)
{
   // code for IE7+, Firefox, Chrome, Opera, Safari
   xmlhttp=new XMLHttpRequest();
}
else
{
   // code for IE6, IE5
   xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}

var fsend = "clientname=";

var tosend = document.getElementById('name').value;

fsend += tosend;

fsend += "&surname=";

tosend = document.getElementById('surname').value;

fsend += tosend;

fsend += "&initial=";

tosend = document.getElementById('initial').value;

fsend += tosend;

fsend += "&id_passport=";

tosend = document.getElementById('id_passport').value;
fsend += tosend;

/*
 fsend += "&mobileno=";
tosend = document.getElementById('mobileno').value;
fsend += tosend;

 fsend += "&telephoneno=";
tosend = document.getElementById('telephoneno').value;
fsend += tosend;

 fsend += "&address=";
tosend = document.getElementById('address').value;
fsend += tosend;

 fsend += "&email=";
tosend = document.getElementById('email').value;
fsend += tosend;

 fsend += "&accounttype=";
tosend = document.getElementById('accounttype').value;
fsend += tosend;

 fsend += "&sex=";
tosend = document.getElementById('sex').value;
fsend += tosend;

fsend += ";

tosend = document.getElementById('dob').value;

fsend += tosend;

fsend += ";

tosend = document.getElementById('password').value;

fsend += tosend;/*

xmlhttp.open("GET","register.php"+fsend,false);

xmlhttp.send();

document.getElementById("maincont").innerHTML=xmlhttp.responseText;

}

function getBalance(acctNo){
if (window.XMLHttpRequest)

{// code for IE7+, Firefox, Chrome, Opera, Safari

xmlhttp=new XMLHttpRequest();

}

else
var fsend = "accNo=";

var tosend = acctNo;

fsend += tosend;

xmlhttp.open("GET","getbalance.php?"+fsend,false);

xmlhttp.send();

document.getElementById("detailst").innerHTML=xmlhttp.responseText;

function logout(){
    if (window.XMLHttpRequest)
        {  // code for IE7+, Firefox, Chrome, Opera, Safari
            xmlhttp=new XMLHttpRequest();
        }
    }

else
```javascript
// code for IE6, IE5

xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

xmlhttp.open("GET","logout.php?",false);

xmlhttp.send();

window.location = "home.html";
}

function getCookie(c_name)
{
    var i,x,y,ARRcookies=document.cookie.split(";");
    for (i=0;i<ARRcookies.length;i++)
    {
        x=ARRcookies[i].substr(0,ARRcookies[i].indexOf("="));
        y=ARRcookies[i].substr(ARRcookies[i].indexOf(="")+1);
        x=x.replace(/\s+\|\s+$/g,"");
        if (x==c_name)
        {
            return unescape(y);
        }
    }
};

function checkLogin(){
```
var accountNo = getCookie("accountNo");

if (accountNo!=null & accountNo!='') {
    getBalance(accountNo);
}
else {
    alert("Please login in-order to perform a transaction");
    window.location = "home.html";
}
}

/*
$(document).ready(function() { 

    // SUCCESS AJAX CALL, replace "success: false," by: success : function() {
    callSuccessFunction() },

    $('#form1').validationEngine({
        ajaxSubmit: true,
        ajaxSubmitFile: "ajaxSubmit.php",
        ajaxSubmitMessage: "Thank you, We will contact you soon !",
        success : false,
        failure : function() {} 
    })

})
function beneficiarypayment() {

    if (window.XMLHttpRequest) {
        // code for IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp = new XMLHttpRequest();
    }
    else {
        // code for IE6, IE5
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    }
    var accountNo = getCookie("accountNo");

    var fsend = "benaccountno=";
    var tosend = document.getElementById('benaccountno').value;
    fsend += tosend;

    fsend += "&amount=";
    tosend = document.getElementById('amount').value;
    fsend += tosend;

    fsend += "&accountNo=";
}
tosend = accountNo;
fsend += tosend;

xmlhttp.open("GET","beneficiary.php?"+fsend,false);

xmlhttp.send();

//document.getElementById("detailst").innerHTML=xmlhttp.responseText;
var resp=xmlhttp.responseText;

if(resp == "paid"){
    alert('Your Transaction has been Successfully Completed.');
    getBalance(accountNo);
}
else{
    alert("You current balance is less than the amount you want to pay");
}

function getBalance(acctNo){
    if (window.XMLHttpRequest)
        {// code for IE7+, Firefox, Chrome, Opera, Safari
xmlhttp=new XMLHttpRequest();
}
else
{// code for IE6, IE5
    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}

var fsend = "accNo=";

var tosend = acctNo;

fsend += tosend;

xmlhttp.open("GET","getbalance.php?"+fsend,false);

xmlhttp.send();

document.getElementById("detailst").innerHTML=xmlhttp.responseText;

function transferamount(){
if (window.XMLHttpRequest)
    {// code for IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp=new XMLHttpRequest();
    }
else
    {// code for IE6, IE5
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
var accountNo = getCookie("accountNo");

var fsend = "benaccountno=";
var tosend = document.getElementById('benaccountnos').value;
fsend += tosend;

fsend += "&company=";
tosend = document.getElementById('company').value;
fsend += tosend;

fsend += "&amount=";
tosend = document.getElementById('amounts').value;
fsend += tosend;

fsend += "&accountNo=";
tosend = accountNo;
fsend += tosend;

xmlhttp.open("GET","test.php?"+fsend,false);

xmlhttp.send();

//document.getElementById("detailst").innerHTML=xmlhttp.responseText;
var resp=xmlhttp.responseText;

if(resp == "done"){
    alert('You have paid the Beneficiary.');

        getBalance(accountNo);

} else{
    alert("You current balance is less than the amount you want to pay");
}

</script>

</head>

<body onload="checkLogin()"
<div id="maincont">

<div id="wrapperdetail">

<div id="form-divdetail">

<form class="form" id="form1" action='register.php' method='GET'>

<h1 align="center"><label >DETAILS</label></h1>

<div>
<h3>
<label>
<span class="acctholder">Account Holder</span>
</label>

</h3>

<br/>
<hr/>

<label>
<div id ="detailst">
<span class="acctype">Account Type</span>
</div>
</label>

<label>
<div id ="detailst">
<span class="accNo">Account Number</span>
</div>
</label>

<label>
<div id ="detailst">
<span class="balance">Available Balance</span>
</div>
</label>

</div>

<br/>
<br/>

</form>
</div>
</div>
</div>
<br/>
<br/>
<hr/>
<p class="submit">
<input type="button" value="Logout" onclick="logout()"/>
</p>
</form>
</div id="form-div">
</div>
</div>
<div id="AASDFASDFASDF" class="container">
<div align="right">
<h2><a href="loan.php">Apply For A Loan</a></h2>
</div>
<br/>
<br/>
<p>
  <div id="wrapper2">
  <div id="form-div2">
  <form class="form" id="form2">
    <h3 align="center">EFT Transaction</h3>
    <hr />
    <p class="name">
      <input name="benaccountno" type="text" class="validate[required,custom[onlyLetter],length[0,100]] text-input" id="benaccountno" />
      <label for="name">To Account No.</label>
    </p>
    <p class="name">
      <input name="amount" type="text" class="validate[required,custom[onlyLetter],length[0,100]] text-input" id="amount" />
      <label for="name">Amount</label>
    </p>
    <p class="submit">
      <input type="button" value="Transfer" onclick="beneficiarypayment()" />
    </p>
  </form>
  </div>
</div>
</p>
Beneficiary Payment

Bank Name

Beneficiary Account No.

Telkom

Edgars
<option value="UWC">UWC</option>

</select>


Company</label>

</p>

<p class="name">

<input name="amount" type="text" class="validate[required,custom[onlyLetter],length[0,100]] text-input" id="amounts" />

<label for="name">Amount</label>

</p>

<p class="submit">

<input type="button" value="Pay" onclick="transferamount()"/>

</p>

</form>

</div>

</div>

<div id="wrapper8">

<div id="form-div8">


<id="wrapper8">

<id="form-div8">


Enter your Acc number<br><br>
<input name="AccountPin" type="text" id="AccountPin" />
<p class="submit">
  <input type="submit" name="submit" value="View" />
</p>
BIBLIOGRAPHY

- Seminarprojects.com/Thread-insurance-on-internet—9437
- www.scribd.com/doc/53669376/36/Limitations-of-E-Banking
- http://www.programmingportal.in/2010/05/online-banking-system-sequence-diagram.html