

## **1.3Introduction**

### **1.3.1 Purpose**

The purpose of this document is to present a detailed description of the E-Banking System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

### **1.3.2 Scope**

This Document plays a vital role in the development life cycle (SDLC) as it describes the complete requirement of the system. It is meant for use by the developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

### **1.3.3 Overview of Document**

The Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The next chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

## 1.4 Overall Description

### 1.4.1 Contexts Diagram

A System Context Diagram is the highest level view of a system, similar to Block Diagram, showing a (normally software-based) system as a whole and its inputs and outputs from/to external factors. The Context Diagrams show the interactions between a system and other actors with which the system is designed to face. They are also typically drawn using labelled boxes to represent each of the external entities and another labelled box to represent the system being developed. The relationship is drawn as a line between the entities and the system being developed.

Context Diagram is a data flow diagram showing data flows between a generalized application within the domain and the other entities and abstractions with which it communicates.

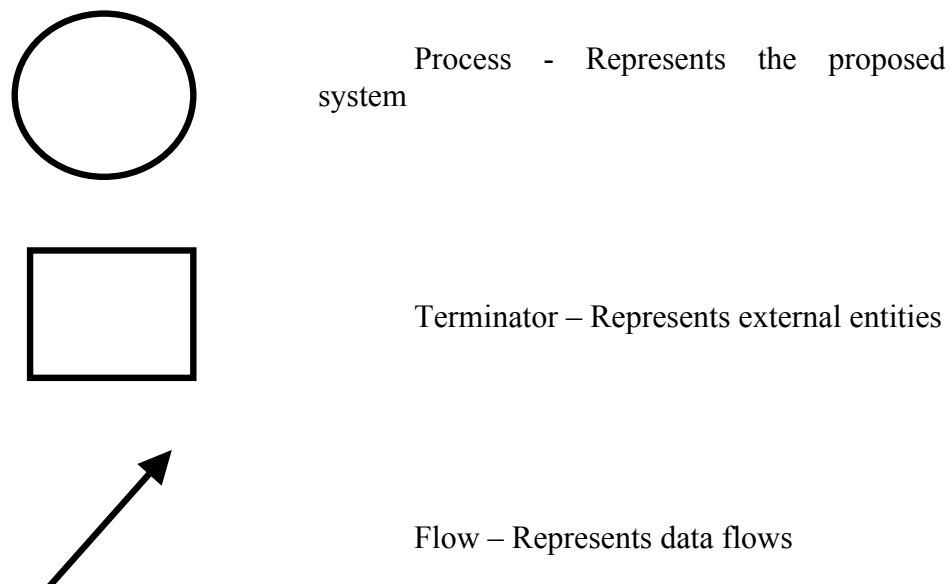


Fig 1 Notation for context Diagram

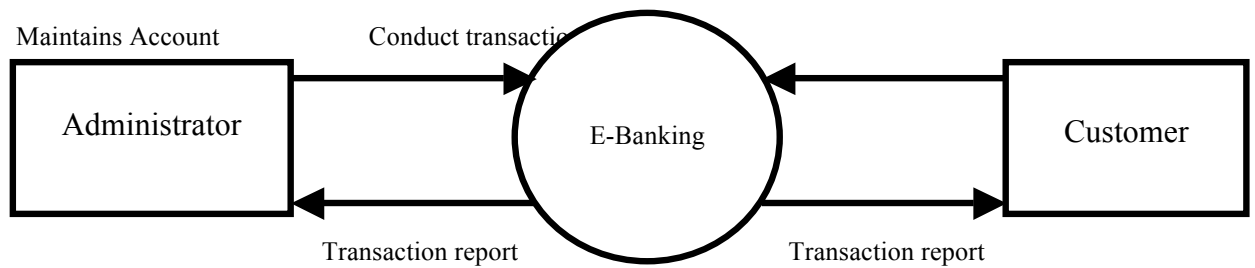


Fig 2. Context Diagram of internet banking system

### 1.4.2 User Interfaces

User Interface of the project will be simple and user friendly. It will have the following pages for the user interactions:

- A login screen in which user has to enter his id and password, so that authorized user can have an access to his account.
- After login the next screen helps the user to check his balance, transfer funds, update his account etc.
- Every above task has its own separate page. User can have access to these pages after logging in successfully.

### 1.4.3 Hardware Interfaces

#### Minimum Requirements

Processor : Pentium III Processor

RAM : At least 1 GB

Hard Disk : At least 10 GB

Change of Password: This module allows customers to change their password.

Cheque Book Request: This module allows customers to request for cheque book.

#### **1.4.6 User Characteristics**

The Customer is expected to be Internet literate and be able to use a search engine. The main screen of the E-Banking Website will have the login link. Customer should know his ID and Password which is mandatory for login.

### **1.5 Specific Requirements**

#### **1.5.1 External Interface Requirements**

The link to external system are the links to the Customer Database to verify the account of a Customer. The Customer Database fields of interest to E-Banking Systems are Customer name, Customer (ID) number, and email address.

The Customer's id and password is sent to Database and it validates returns status of his account. The admin can request for a list of customer names, requests (optional) email addresses for any alteration.

#### **1.5.2 Specific Requirements**

How the E-Banking System will interact with the environment, what will be the functional and non-functional requirements. These all the steps should be defined here for providing a powerful base to design phase. The design of the project will completely depend on the functional and non-functional requirements. So these should be defined clearly and accurately for the effectiveness.

### 1.5.3 Functional Requirements:

Following are the services which this system will provide. These are the facilities and functions required by the customer.

- a) Balance enquiry
- b) Updating customer information
- c) Balance transfer
- d) Cheque book request
- e) Cheque Status

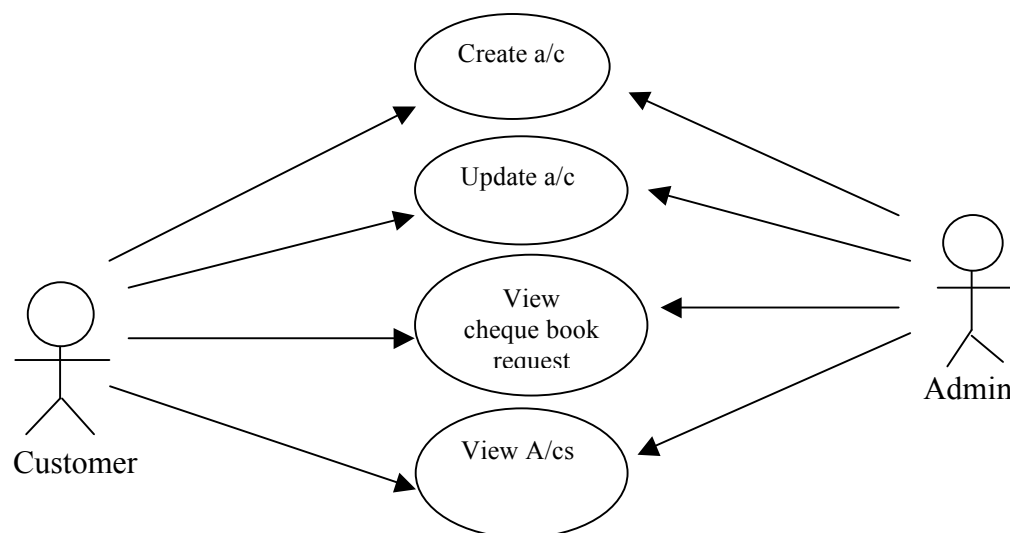


Figure 3 – Use case diagram

# **Chapter 2**

# **System Design**

## 2.1 Physical Design

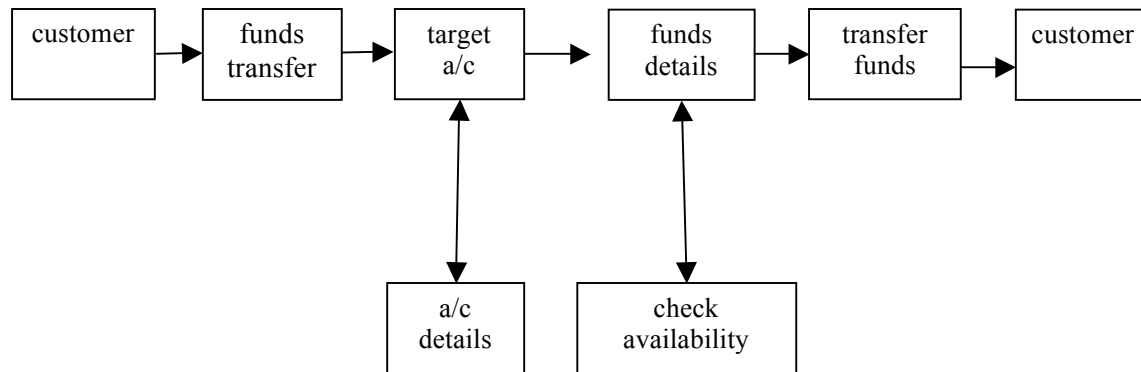


Fig 4: Block Diagram

### 2.1.1 PROJECT PLANNING

The organization can perform the its activities with the help of following modules

#### **Bank Master Module:**

This module mainly deals with creation of new Account i.e. Adding new Account Holders. Other activities such as viewing, Modifying, Deleting Account holder Details, Viewing Chequebook requests.

#### **Transfer Module:**

This module deals with transactions such as transferring funds. The account holder can transfer funds to another account by specifying his first name, last name, account number and amount to be transferred. If user's entered amount greater than his available balance the transaction will not happen.

#### **Reports Module:**

This module deals with generation of reports for customer and Bank such as Daily reports, Weakly reports and report for a particular period of time.

### 2.1.2 DFD

#### LEVEL -0 DFD

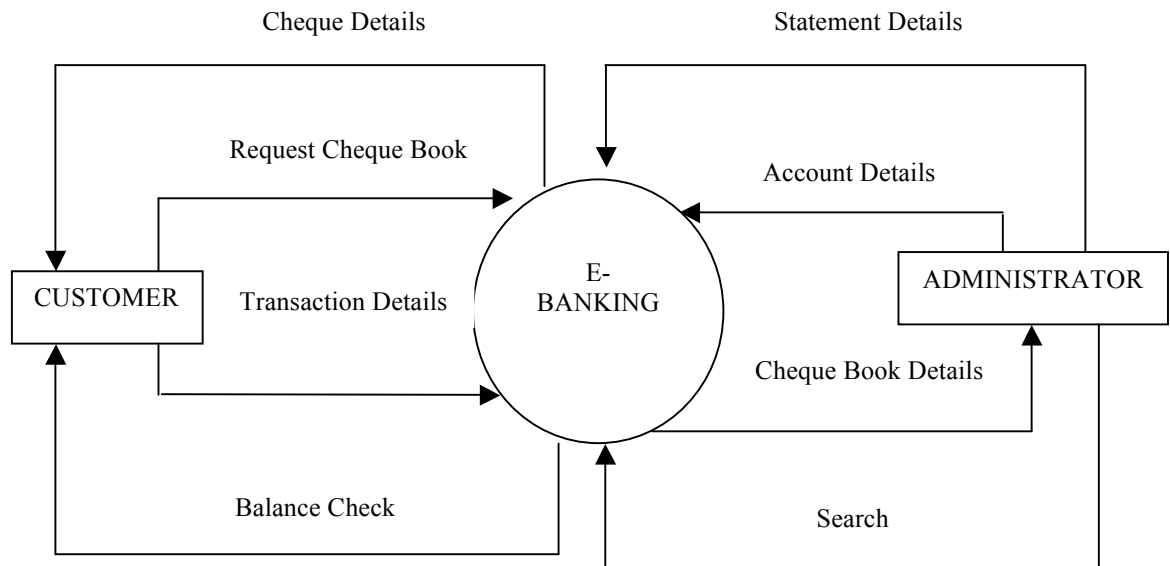


Fig 5.1: Level-0



## Level-1 DFD

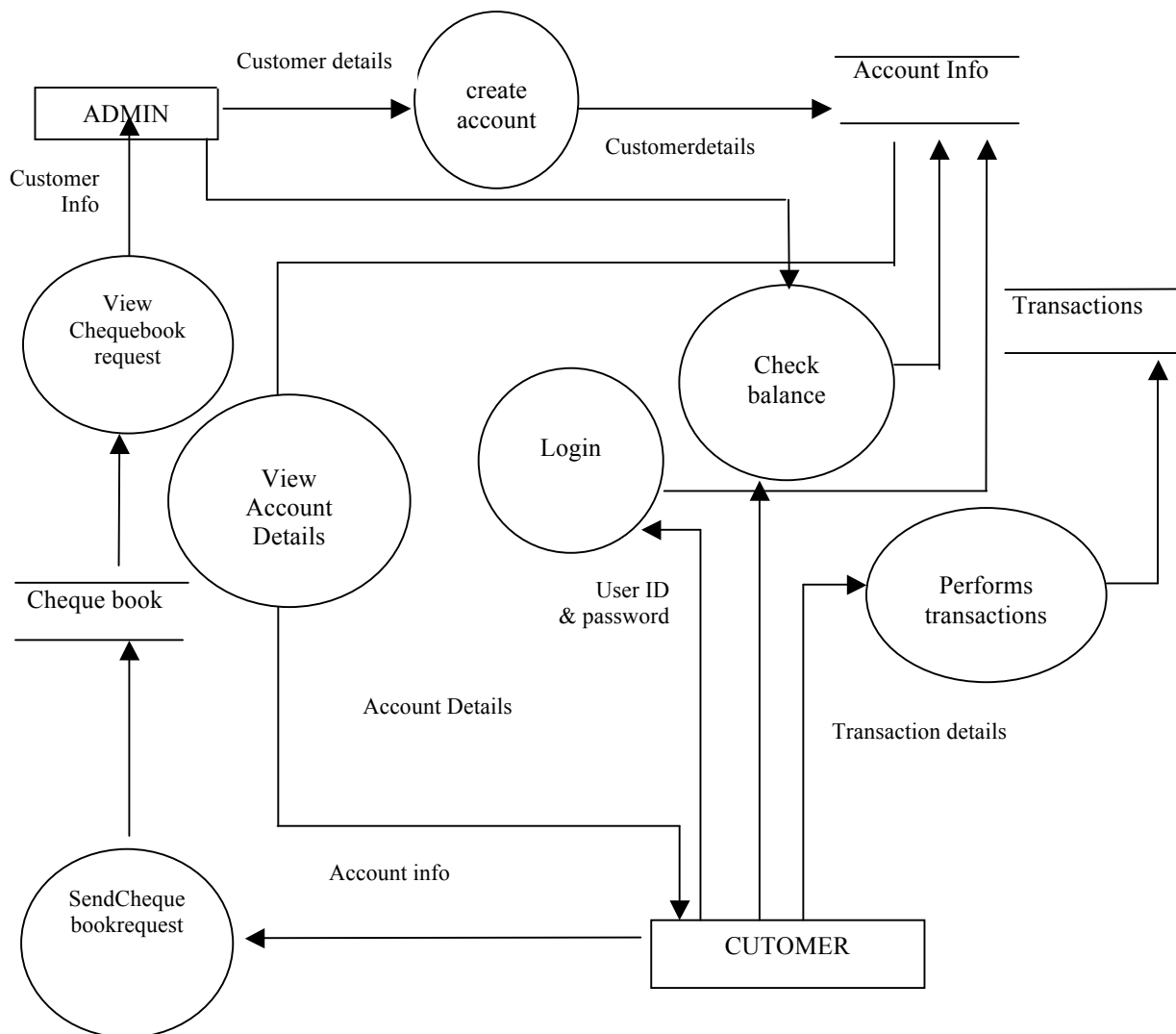
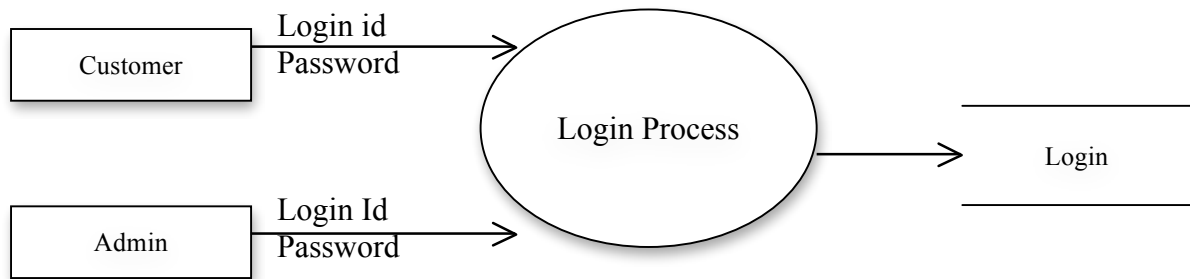


Fig 5.2: Level-1

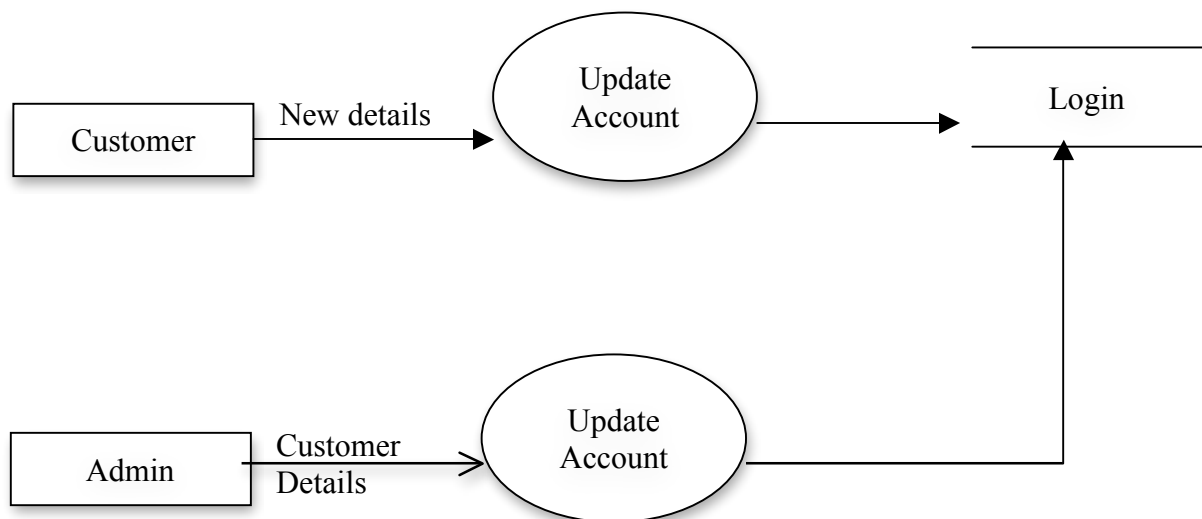
## Level 2 DFD

### 1. Login



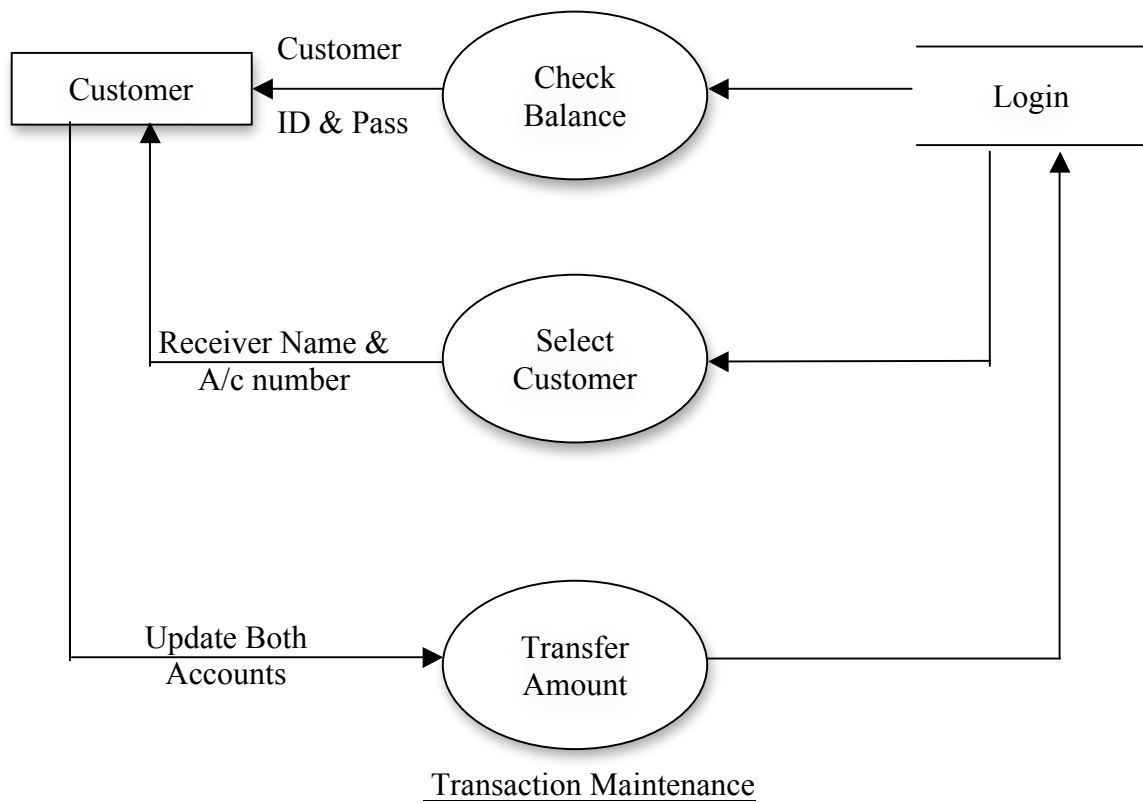
DFD Login

### 2. Account maintenance

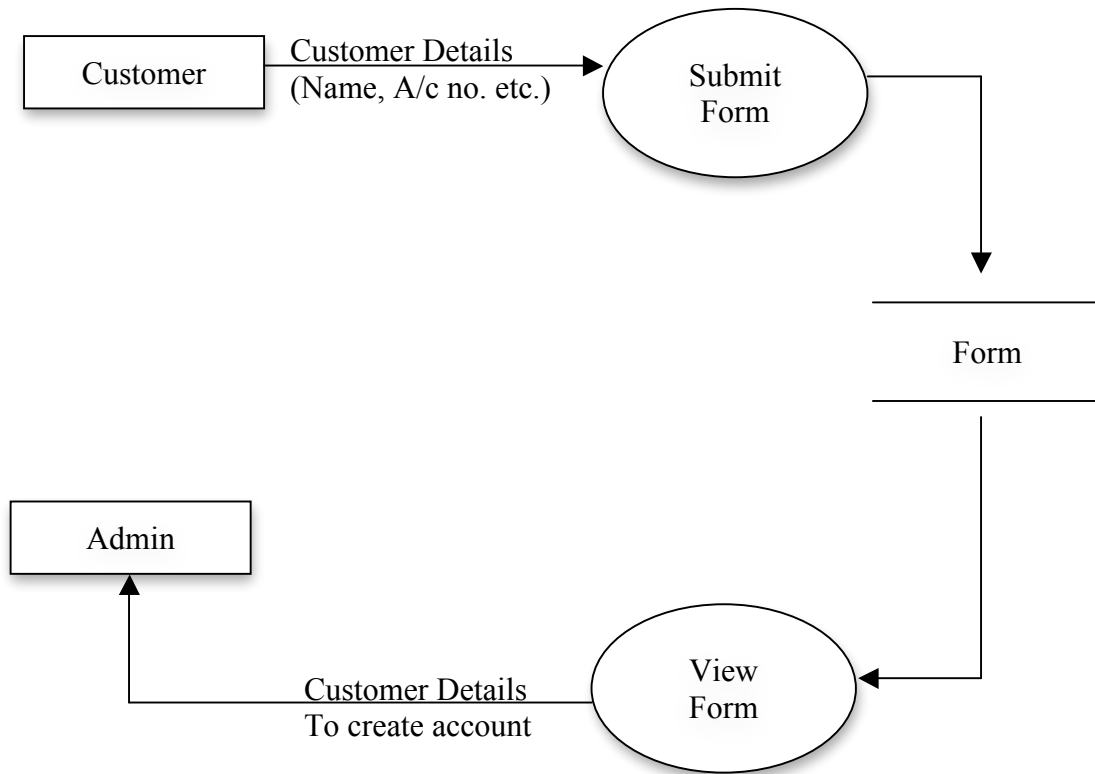


DFD Account maintenance

### 3. Transaction Maintenance



#### 4. Net Banking Forms maintenance



DFD Net banking form maintenance

## 2.2 Database Design:-

### 2.2.1 Tables

A) Form:-

Field name	Field type	size	Description
first_name	Varchar	10	First name of the customer
last_name	Varchar	10	Last name of the customer
father_name	Varchar	20	Father's name of the customer
gender	Varchar	6	Gender of the customer
Date	Varchar	2	Day of birth
Month	Varchar	2	Month of birth
Year	Varchar	4	Year of birth
Address1	Varchar	15	Address line 1
Address2	Varchar	15	Address line 2
Address3	Varchar	15	Address line 3
City	Varchar	10	City of residence
Pan	Varchar	10	PAN number of the customer
Ac_no	Varchar	11	Account number of customer

Table 1: Form

B) Login:-

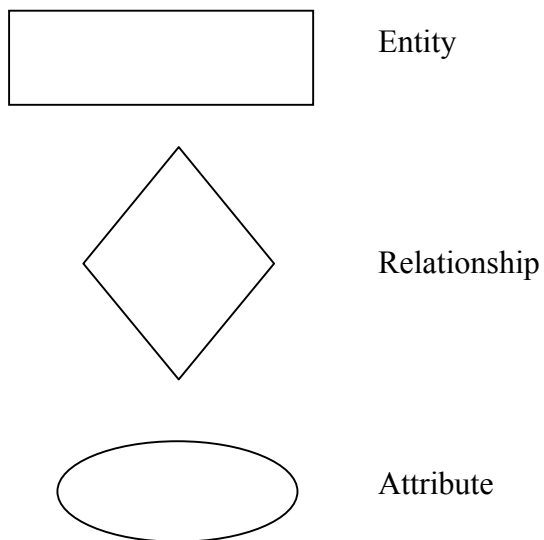
Field name	Field type	size	Description
User	Varchar	20	User id of customer
Password	Varchar	10	Password of customer
Account	Int	11	Account balance of customer
First_name	char	10	First name of the customer
Last_name	char	10	Last name of the customer
Father_name	Varchar	10	Father's name of the customer
Gender	Varchar	10	Gender
Date	Int	2	Day of birth
Month	Int	2	Month of birth
Year	Int	4	Year of birth
Address1	Varchar	10	Address line 1
Address2	Varchar	10	Address line 2
Address3	Varchar	10	Address line 2
City	Varchar	10	City of residence
PAN	Varchar	10	PAN number of the customer
Ac_no	Int	10	Account umber of the customer

Table 2: Login

### 2.2.2E-R Model

An entity-relationship model (ERM) is an abstract and conceptual representation of data. Entity-relationship modelling is a database modelling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion. Diagrams created by this process are called entity-relationship diagrams, ER diagrams or ERDs.

#### 2.2.2.1 COMPONENTS OF E-R DIAGRAM



Components of E-R Diagram

## A) ER Diagram

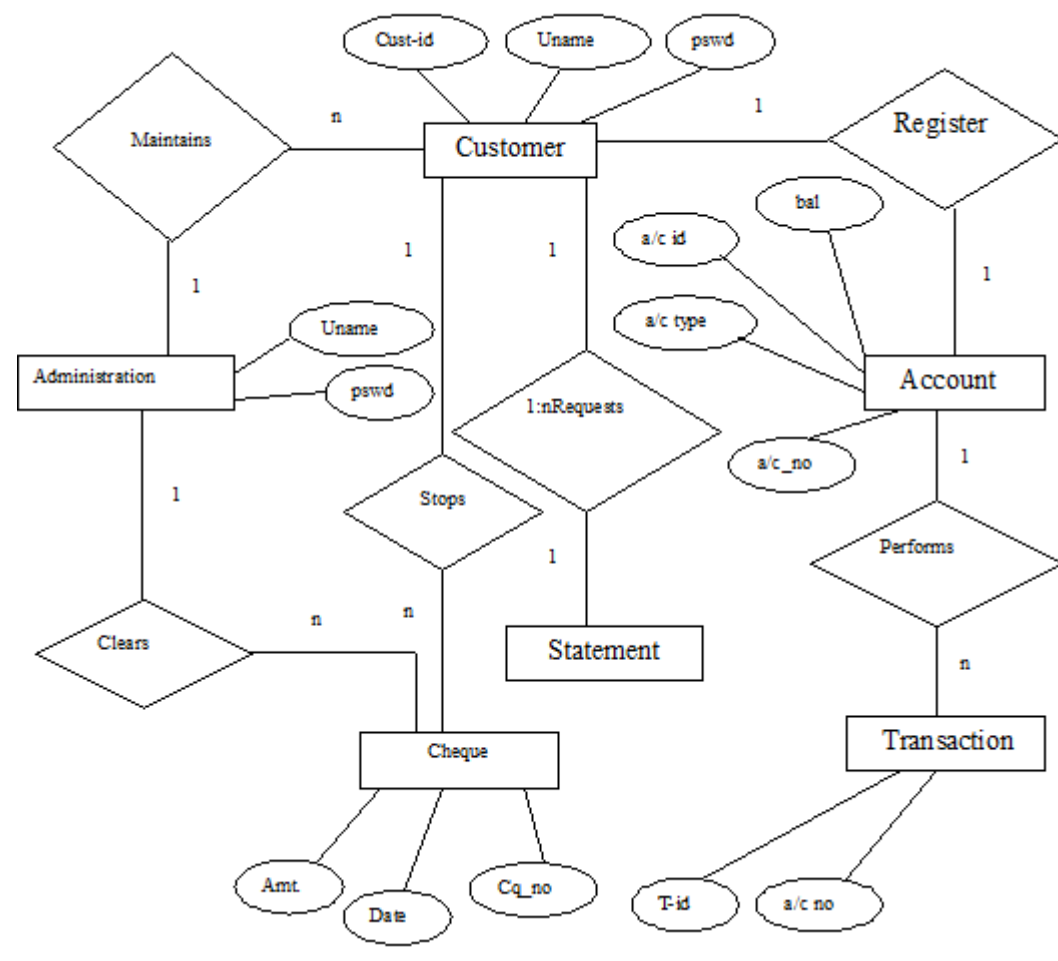


Fig 6: E-R Model



## 2.3 Interface design

### (a) Input design

#### Login Form

Login to access account	
User name	<input type="text"/>
Password	<input type="password"/>
<input type="button" value="Log in"/>	

#### Validation:-

1. Username and Password provided by the user must be correct if not invalid user will be printed on the screen.
2. Both the fields cannot be left blank.

## Request Net banking form

Please fill in your details

First name	<input type="text"/>
Last name	<input type="text"/>
Father's name	<input type="text"/>
Gender	<input type="text"/>
Date of birth	date <input type="text"/> month <input type="text"/> year <input type="text"/>
Address line 1	<input type="text"/>
Address line 2	<input type="text"/>
Address line 3	<input type="text"/>
City	<input type="text"/>
PAN	<input type="text"/>
Account number	<input type="text"/>
<input type="button" value="Submit"/>	

### Validations:-

1. Account number can only contain numeric values.
2. All the fields must be filled.
3. Age should be more than or equal to 18.

## Create Account Page

Please fill in the details to create Account

User ID	<input type="text"/>
Password	<input type="text"/>
Balance	<input type="text"/>
First name	<input type="text"/>
Last name	<input type="text"/>
Father's name	<input type="text"/>
Gender	<input type="text"/>
Date of birth	date <input type="text"/> month <input type="text"/> year <input type="text"/>
Address line 1	<input type="text"/>
Address line 2	<input type="text"/>
Address line 3	<input type="text"/>
City	<input type="text"/>
PAN	<input type="text"/>
Account number	<input type="text"/>
<input type="button" value="Submit"/>	

### Validations:-

1. Account number can only contain numeric values.
2. All the fields must be filled.
3. Age should be more than or equal to 18.

## Update Account

### Update Account Details

First Name\*

Last Name\*

Father's Name\*

Gender

Male▼

Date of Birth

Date

1▼

Month

1▼

Year

1920▼

Address Line 1\*

Address Line 2\*

Address Line 3

City

Faridabad▼

Submit

### Validations:-

1. All the fields must be filled.

## Transfer Money

Transfer money to another account

\* Fields must be filled

First name*	<input type="text"/>
Last name*	<input type="text"/>
Account number*	<input type="text"/>
Amount to be transferred*	<input type="text"/>

Proceed

### Validations:-

1. All the fields must be filled.
2. Amount to be transferred must be less than Available balance

(b) output design

Administrator's Page

Review Forms

First name	Last name	Father name	Gender	DD	MM	YY	Add 1	Add 2	Add 3	City	PAN	Ac. no.

Validation:-

No validations.

## Customer's Page

Welcome user's name

Your Account Balance is \_\_\_\_\_

Customer Details

Update Account

Transfer Money

## Account Details

User ID	Password	Ac. Bal.	First name	Last name	Father name	Gender	DD	MM	YY	Add 1	Add 2	Add 3	City	PAN	Ac. no.

### Validations:-

1. No two validations

# **Chapter 3**

## **Systems Development & Implementation**



## 3.1 Program Development

### A) Coding

#### 1) Login

```
<html>

<head>

<title>Login Page</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

</head>

<body><%@include file="home.jsp" %>

<script>

function check()

{

if(document.frm.name.value=="")

{

alert("User Name cannot be blank");

return false;      }

if(document.frm.pass.value=="")

{

alert("Password cannot be blank");

return false;      }

return true;

    }

</script><br><br><br>

<form action="logged.jsp" name="frm" method="get" onsubmit="return check()">

<p align="center">Login to access Account</p>

<table align="center">

<tr>
```

```

<th>User Name:</th>

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

</tr>

<tr>

<th>Password:</th>

<td><input type="password" name="pass"></td>

</tr>

<tr><td align="center">

<input type="submit" value="Login" align="center">

</td></tr>

</table>

</form>

</body>

</html>

```

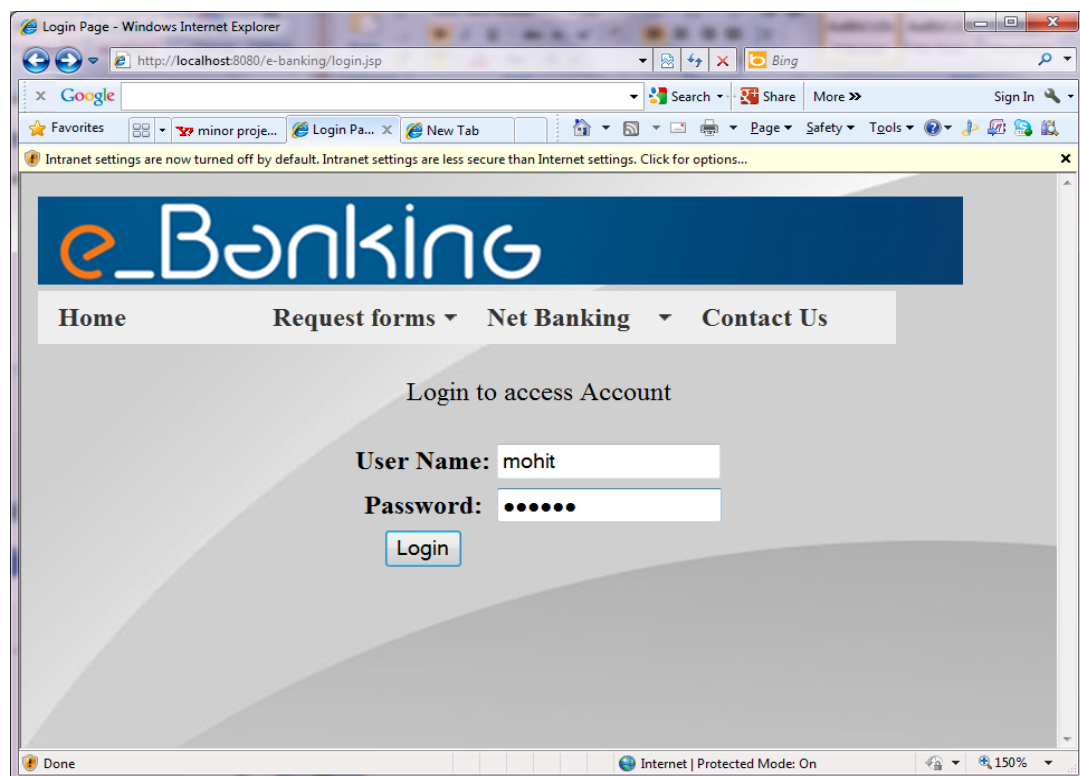


Fig 7: login form

## 2) Admin

```
<%@include file="home.jsp" %>

<form action="form_view.jsp">

<br><br><h1>Welcome Administrator</h1>

<input type="submit" value="Review Forms" align="center">

    <a href="view_acc.jsp">view Accounts</a>

    <a href="createAcc.jsp">create Account</a>

</form>
```

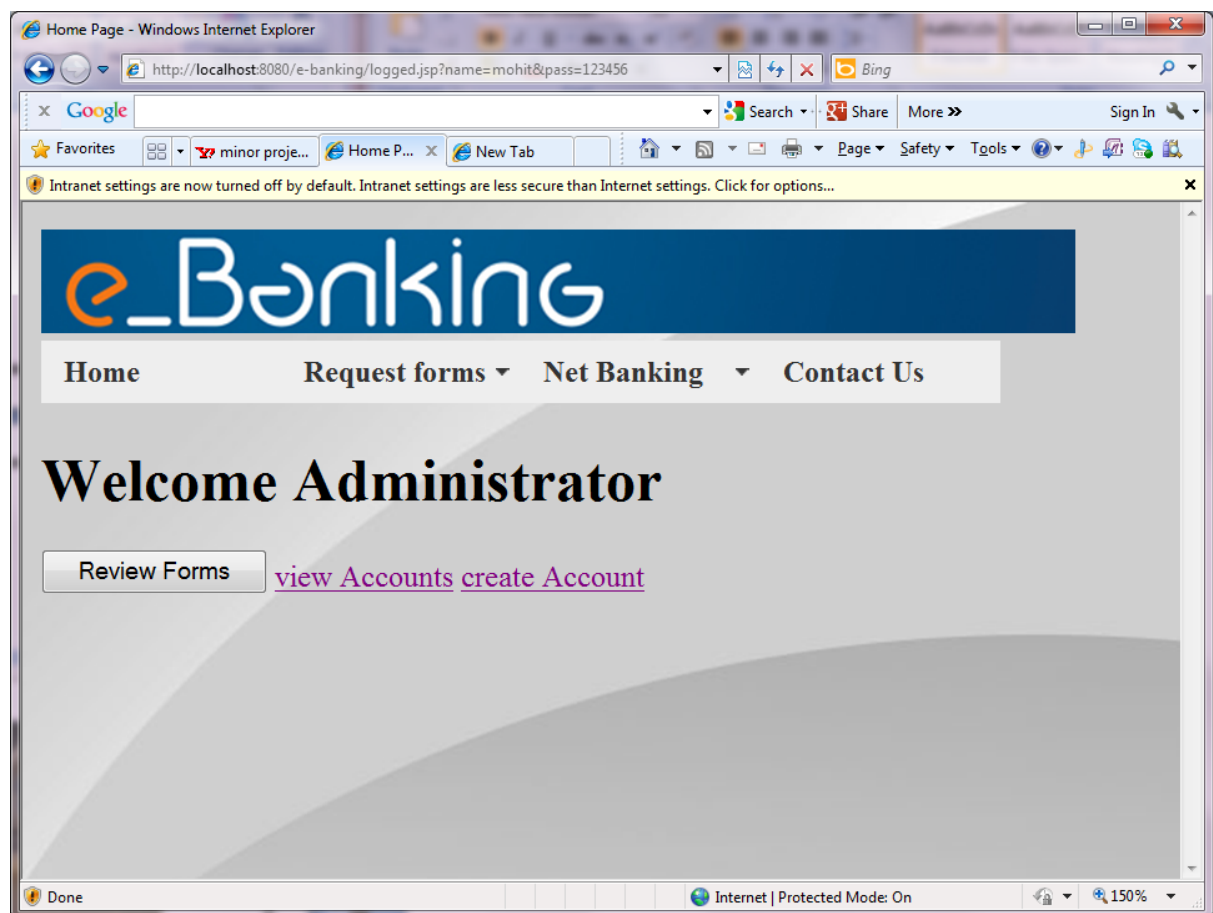


Fig 8: Administrator Page

```

<%
while(rs.next())
{   out.println(rs.getString(3));
}
%>
</p>
<a href="customer_details.jsp">Customer Details</a>
<a href="update_acc.jsp">Update Account Details</a>
<br>
<a href="transfer.jsp">Transfer Money</a>
<%} catch(Exception e)
{   out.println("exception is "+e);
}   %>

```



Fig 9: Customer page

```

<br><p align="center">Please fill the details to create account</p>

<form action="transfer_next.jsp" name="frm" onsubmit="return(check())">

<table align="center">

    <p><center>Transfer monney to another account</center> </p>

    <tr><td>First Name*</td><td><input type="text" name="first"></td></tr>

    <tr><td>Last Name*</td><td><input type="text" name="last"></td></tr>

    <tr><td>Account Number*</td><td><input type="text"
name="acc_no"></td></tr>

    <tr><td>Ammount to be Transferred*</td><td><input type="text"
name="amt"></td></tr>

    <tr><td colspan="2">* fields must be filled </td></tr>

    <tr><td colspan="2"><input type="submit" value="Proceed"></td></tr>

</table>    </form>

</html>

```

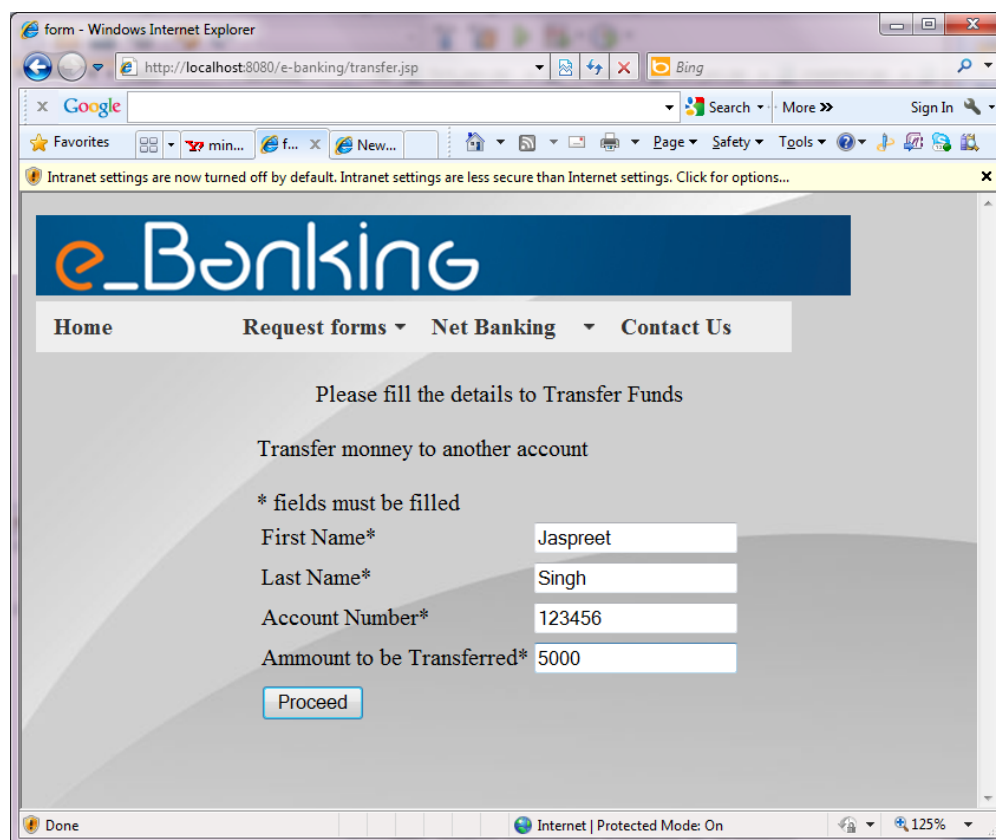


Fig 10: Transfer funds

## **3.2 Testing And Debugging**

*Abstract*—there is a continuously growing number of customers who use online banking because of its convenience. We consider two current online banking problems. First, we observe a lack of attention and research focusing on security issues relevant to the clients' side of online banking systems. Second, there are many security products used in online banking systems. However, security testing is still in its infancy and little is available to verify if those security products are working properly. We discuss the current security testing categories and standards, as well as common security testing approaches.

### **3.2.1 Types of Testing**

#### **A) Unit testing**

Unit testing is not a new concept. It's been there since the early days of the programming. Usually developers and sometime white box testers write unit tests to improve code quality by verifying each and every unit of the code used to implement functional requirements.

Most of us might know the classic definition of unit testing -

“Unit testing is the method of verifying smallest piece of testable code against its purpose.” If the purpose or requirement failed then unit test has failed.

## **B) Black-box testing**

It is a method of software testing that tests the functionality of an application as opposed to its internal structures or workings. This method of test can be applied to all levels of software testing: unit, integration, system and acceptance. It typically comprises most if not all testing at higher levels, but can also dominate unit testing as well.

## **C) System testing**

**System testing** of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

## **D) White box testing**

**White-box testing** (also known as **clear box testing**, **glass box testing**, **transparent box testing**, and **structural testing**) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit

### 3.2.2 Test Cases

#### Login Form

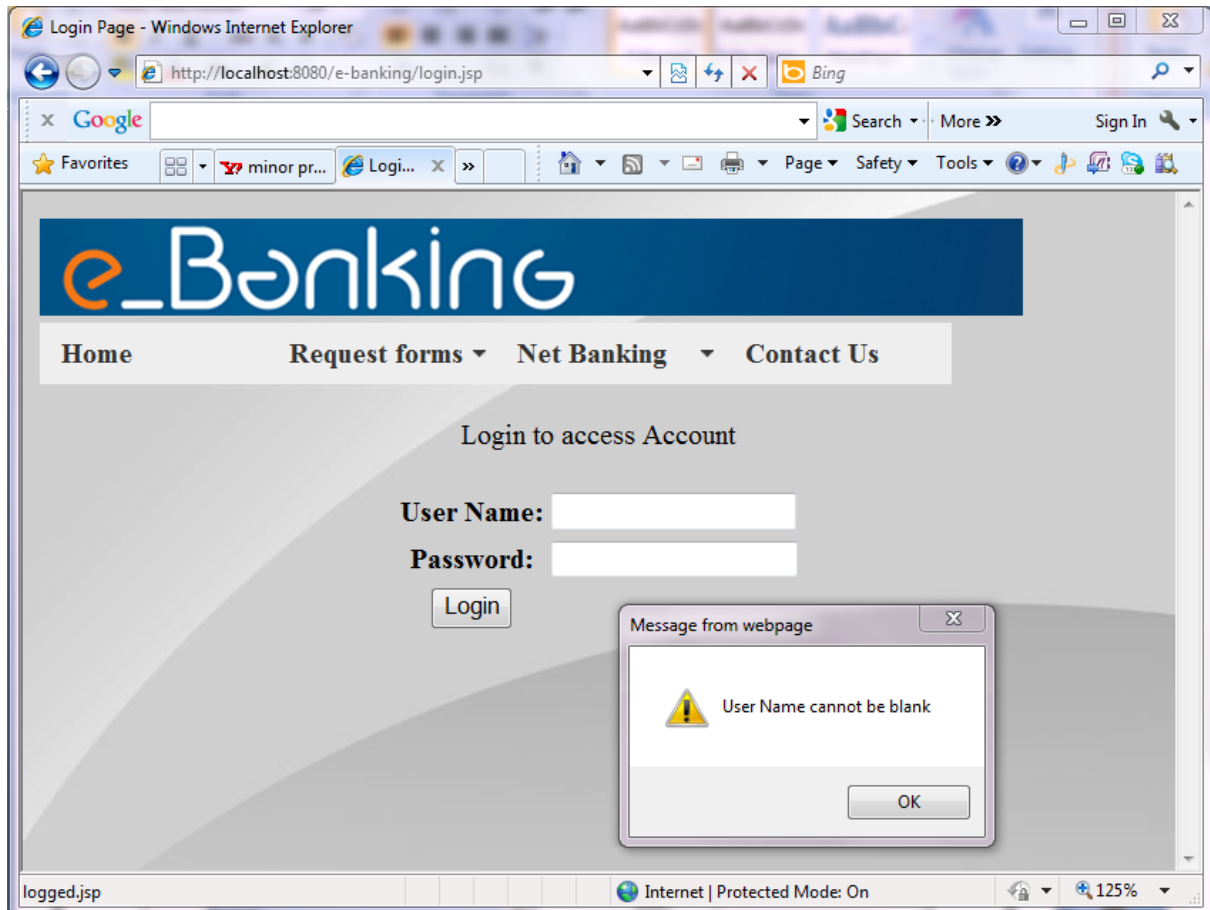


Fig 12: Testing login

1. User name cannot be blank
2. Password cannot be blank



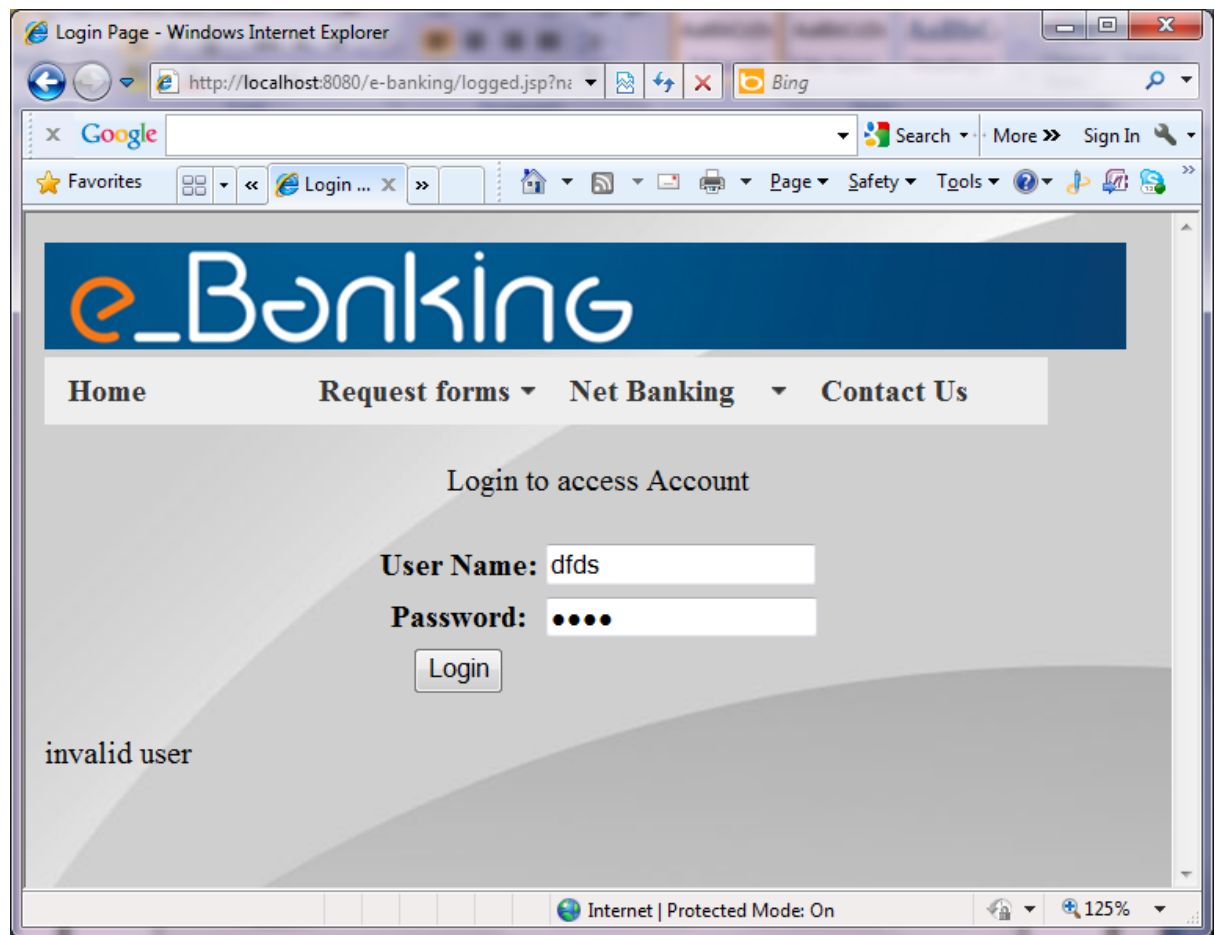


Fig 13: Testing login 2

If User name and Password is incorrect than invalid user will be printed

## Create Account

- a) All fields must be filled.
- b) Account number can only contain numeric values.

form - Windows Internet Explorer

http://localhost:8080/e-banking/createAcc.jsp

Google Search Sign In

Home Request forms Net Banking Contact Us

Please fill the details to create account

User ID	Rocky
Password	●●●●●●
Balance	23000
First Name	Rocky
Last Name	Stallone
Father's Name	hsdsdf
Gender	Male
Date of Birth	Date 28 Month 7 Year 1971
Address Line 1	usa
Address Line 2	china
Address Line 3	india
City	Faridabad
PAN	DGJY689G54
Account Number	DFGT123

Submit

Message from webpage

Account Number can only contains numeric values

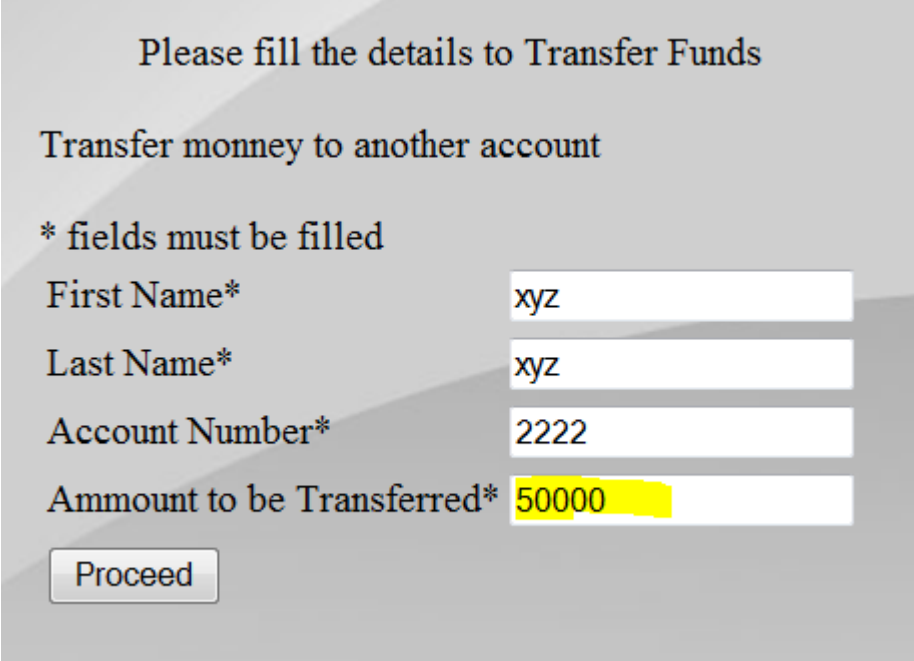
OK

Done Internet | Protected Mode: On 125%

Fig 14: Testing Create account

## Transfer funds

If amount transferred is greater than available balance than not enough balance will be printed



Please fill the details to Transfer Funds

Transfer monney to another account

\* fields must be filled

First Name*	xyz
Last Name*	xyz
Account Number*	2222
Ammount to be Transferred*	50000

Proceed

Fig 15: Testing Transfer funds

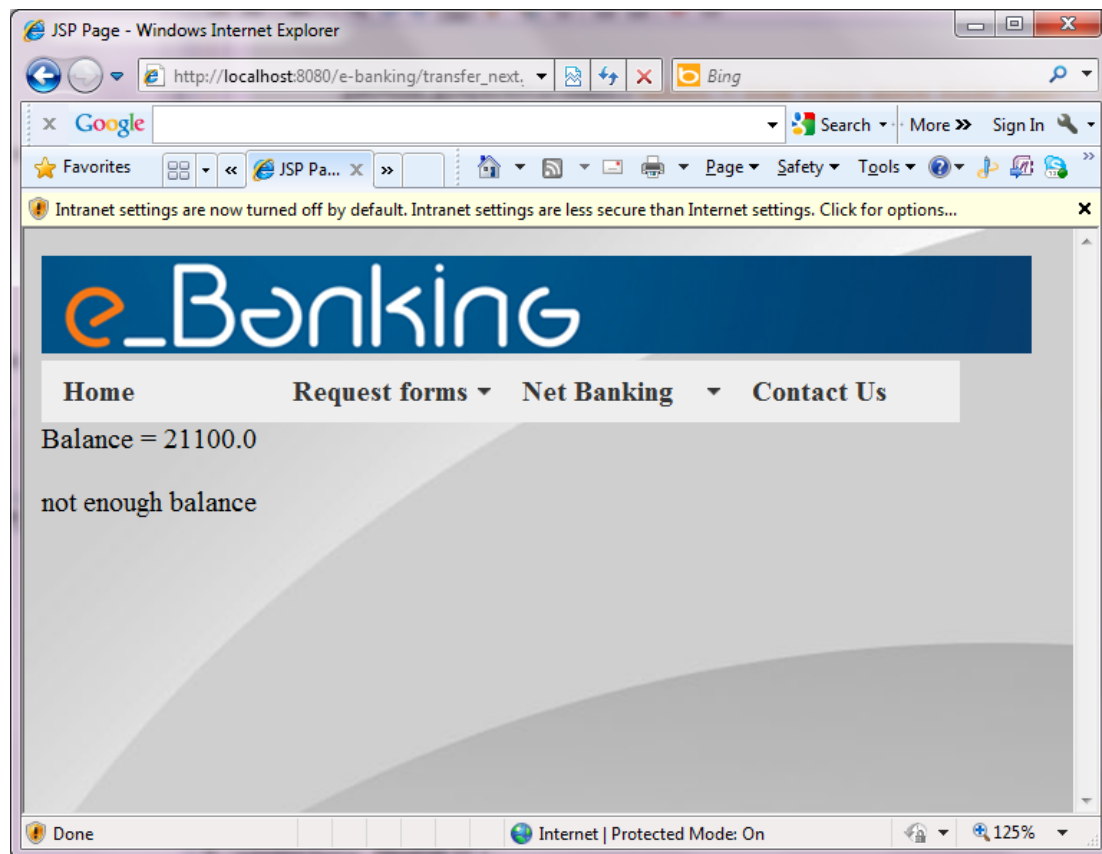


Fig 16: Testing Transfer funds 2