Project Report on

Banking Management System
in JAVA

Submitted by:

Name	Rajesh	***Satya***	SaiSree
Roll no	51	40	72
Reg. no.	12115363	12114535	12114133
Section	K21QT	K21QT	K21QT

Contents

- Introduction
- Project requirements
- System modules
- User interface specification
- Use case and data flow diagram's

1. An Internet Banking System

"Internet banking" refers to systems that enable bank customers to access accounts and general information on bank products and services through a personal computer (PC) or other intelligent device.

- ✓ Our aims at creation of secure banking system.
- ✓ this will be accessible to all users who have a valid user Id and password.
- ✓ A cusomer can access his account from anywhere.

1.1 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.

- Secure access of require data.
- 24X7 availability
- user friendly
- Intelligent user interfaces

Software Requirements	Hardware Requirements
Operating System: Windows or linux or MAC	Processor: any
User Interface: HTML, APPLET	Hard Disk: 10 GB minimum
Programming Language: JAVA	RAM: 256MB or more
Database: ORACLE	Any Screen
Web Server Tomcat-Apache	

Technologies to be used:

•**J2EE** applications are made up of components. A *J2EE component* is a self-contained functional software unit that is assembled into a J2EE application with its related classes and files and that communicates with other components. The J2EE specification defines the following J2EE components:

• Application clients and applets are components that run on the client.

• **A *servlet*** is a Java programming language class that is used to extend the Capabilities of servers that host applications access via a request-response Programming model. Although servlet can respond to any type of request, they are commonly used to extend the applications hosted by web servers.

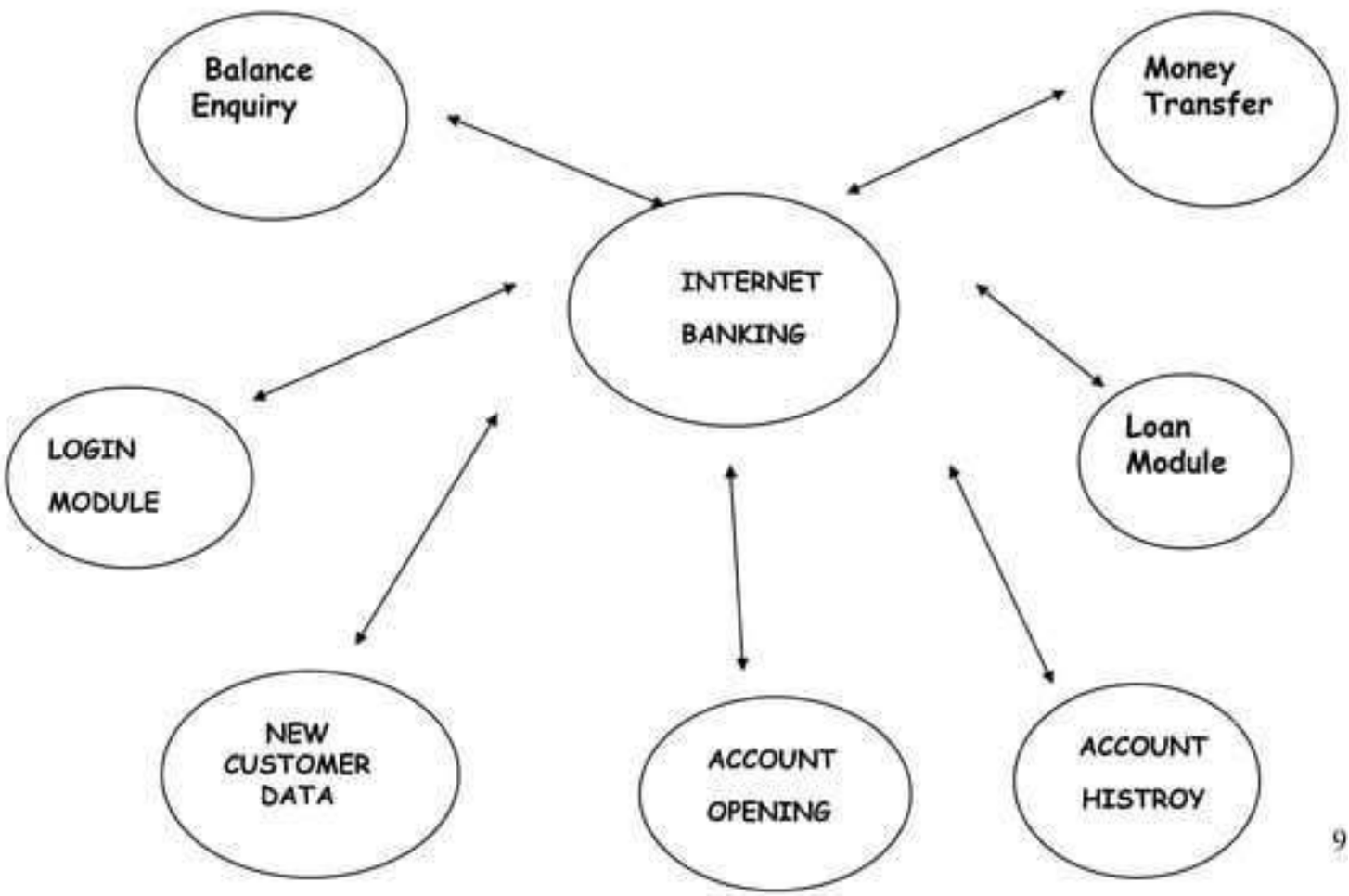
•**JAVASERVER Pages** (JSP) technology allows you to easily create Web content that has both static and dynamic components. JSP technology Makes available all the dynamic capabilities of Java Servlet technology but provides a more natural approach to creating static content. The main Features of JSP technology are as follows:

2.0 System Modules:

Presented by vishnu

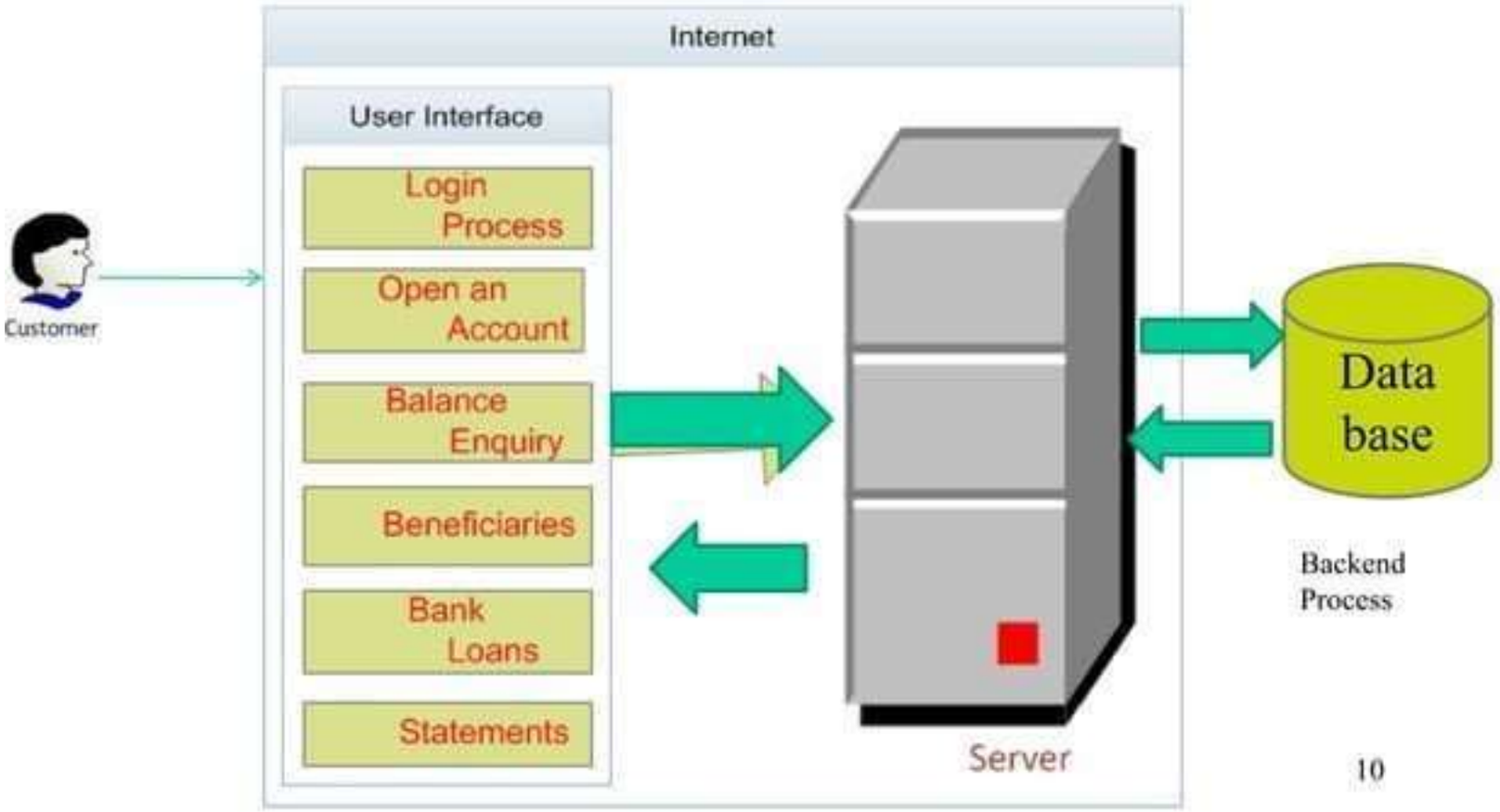
System Modules	Description
(a) Login Module	A user can login with id & password
(b)Open Account	New user can add a account
(c)Balance Enquiry	User can check his balance
(d)Account History	A user can check his account history
(e)Admin Module	Admin can change in function or other action
(f)Loan Module	User can apply for a loan(car,education..)
(g)Money Transfer	User can transfer money

2.1 DATA FLOW DIAGRM OF SYSTEM MODULES



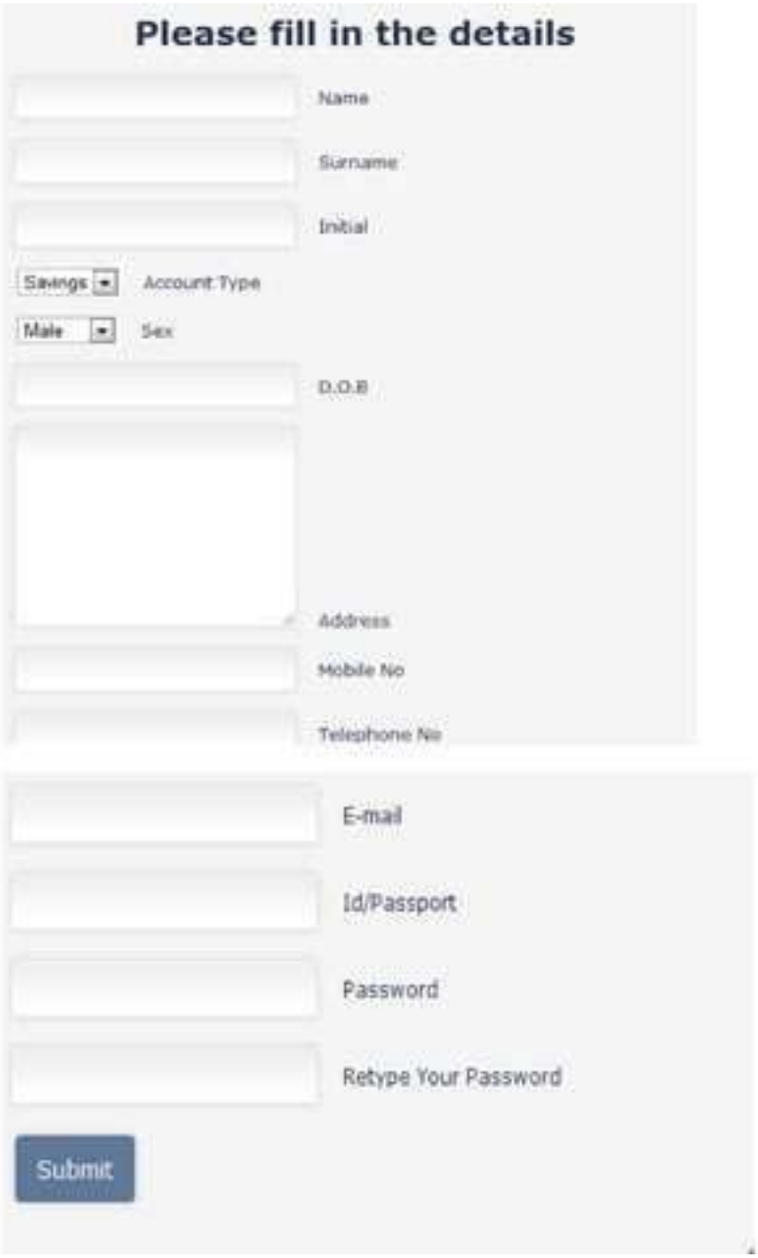
Event Diagram

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.



1.0 Home Page

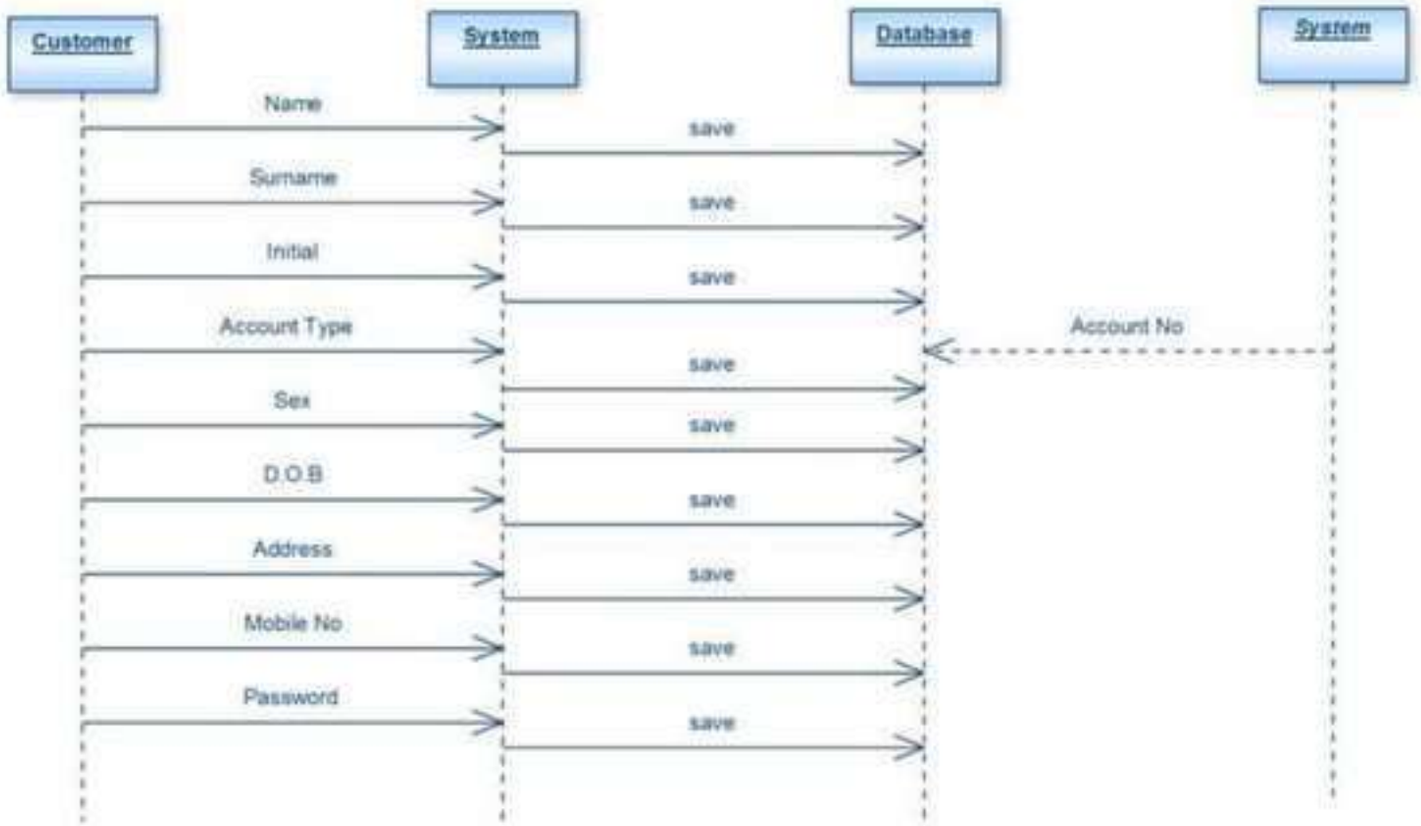

Online Banking.com
Home Page
About Us
Register Now
Products
Contacts
Updated News!
Create your own account
Manage your Financial Life
Client's area
Account No:
Password:
Login
Staff Area



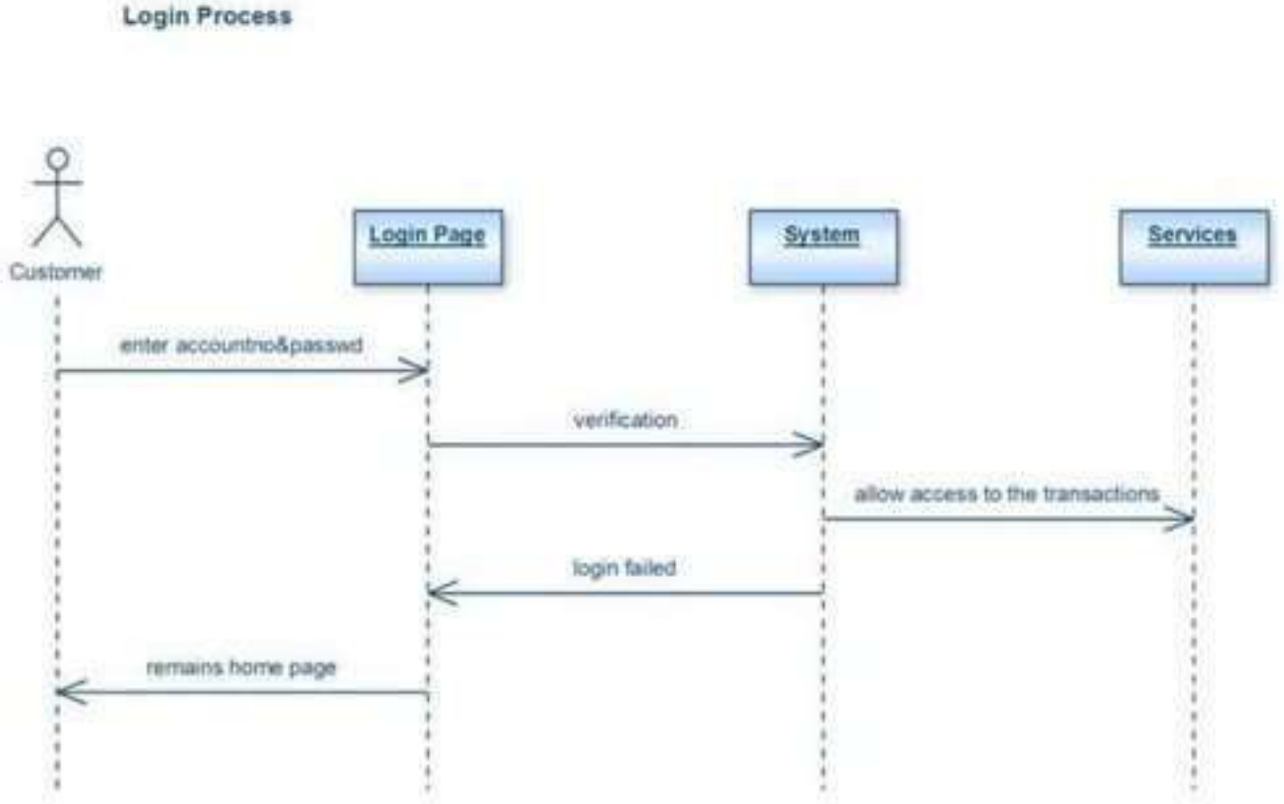
1.1 Registration form and opening a new account

1.2 How the user interacts with the system

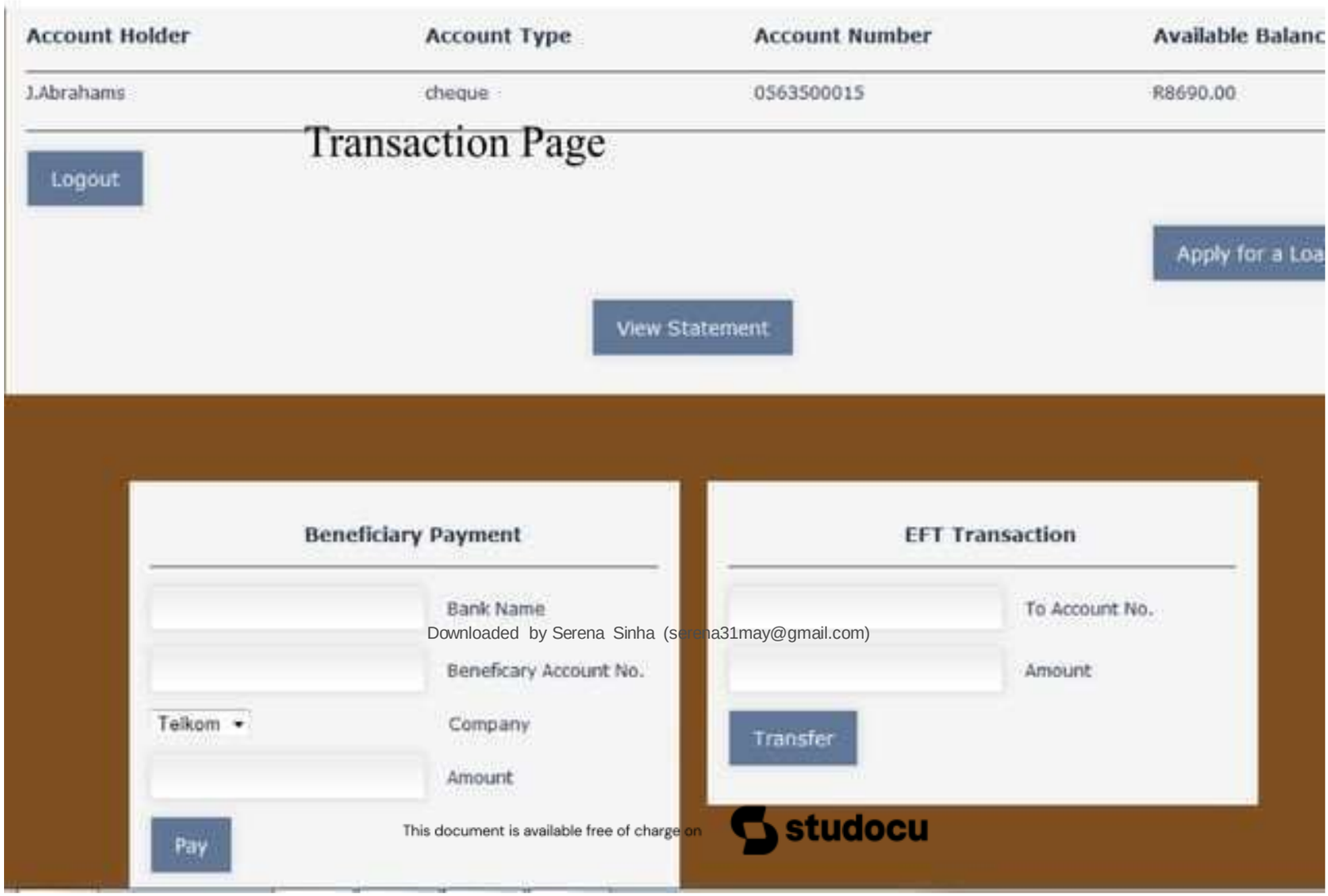
Create a new Account as well as registeration



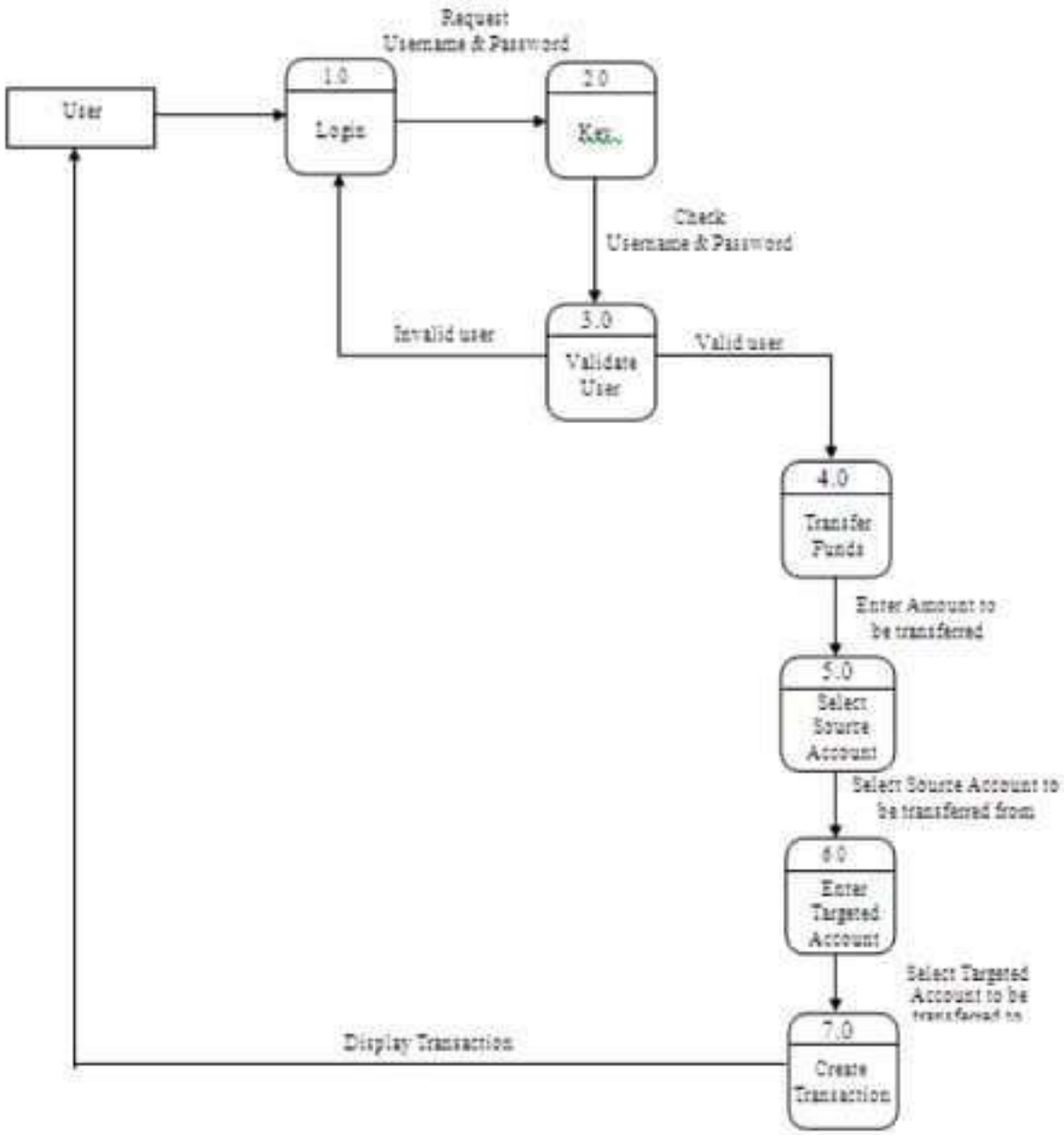
1.3 Login Process



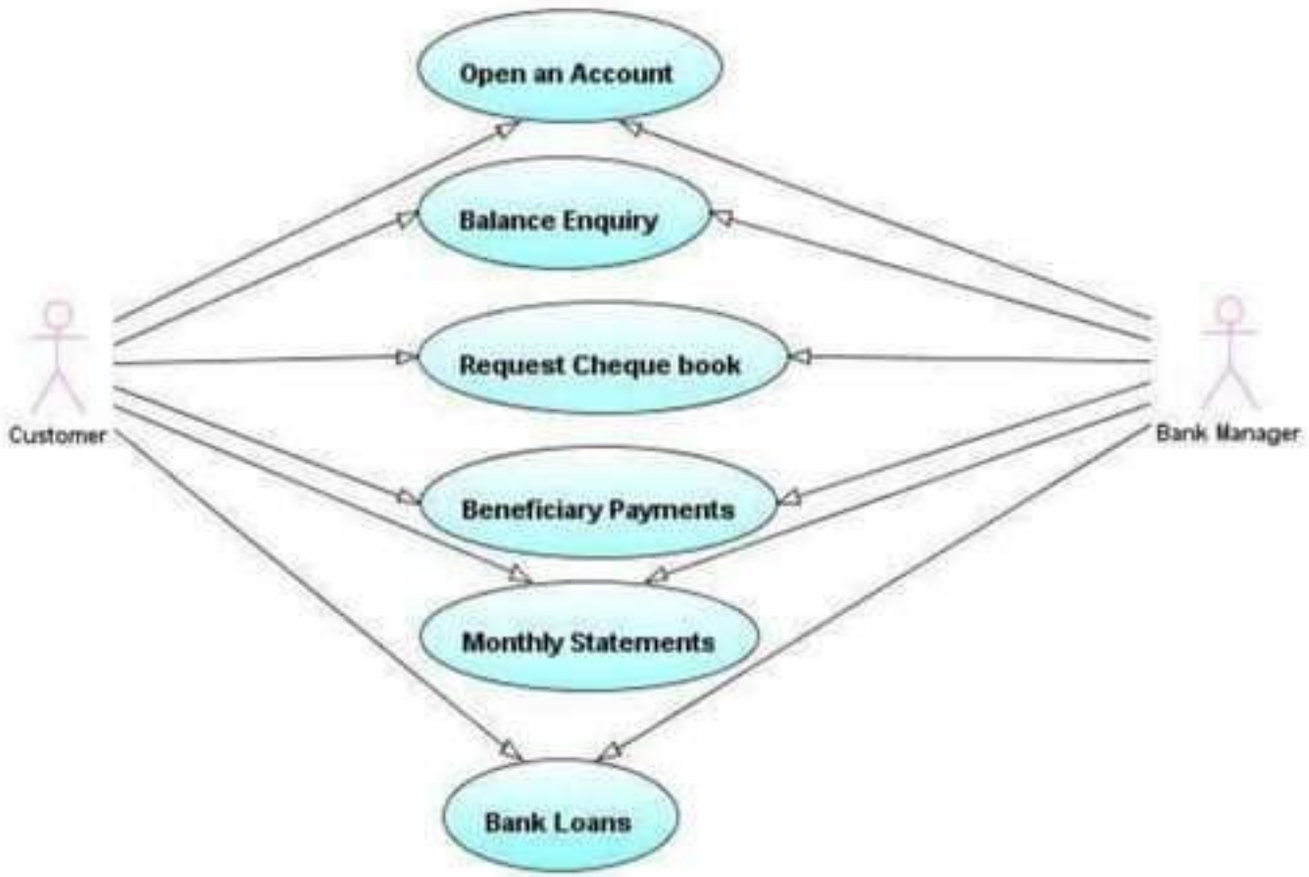
1.4 Transaction page



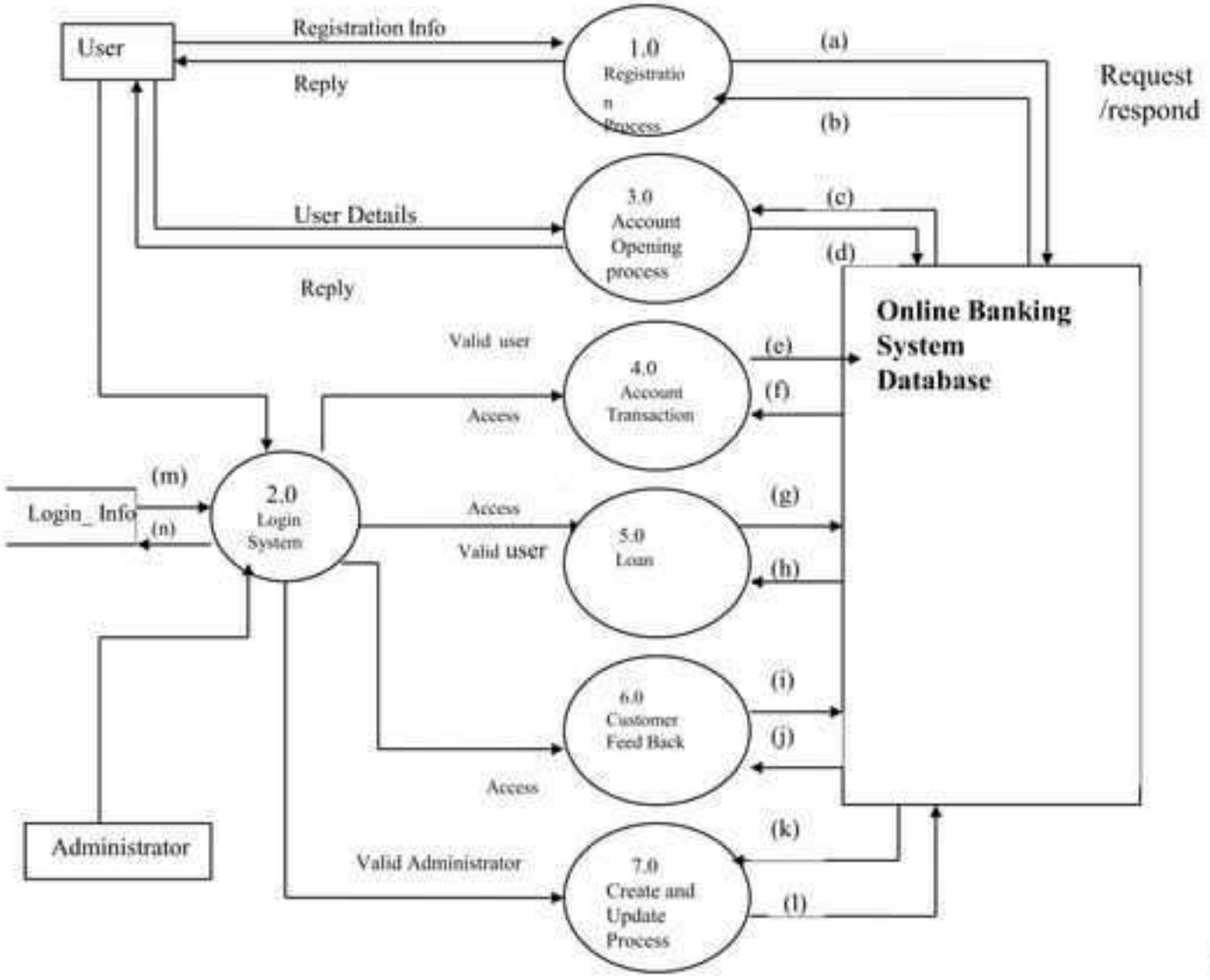
1.5 fund transfer



Use Case diagram of customer requirements



Over all working of Internet Banking



Banking Management System Code: in java

```
mport java.util.Scanner;

public class BankApplication {


    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter your 'Name' and 'CustomerId' to access
your Bank account:");
        String name=sc.nextLine();
        String customerId=sc.nextLine();
        BankAccount obj1=new BankAccount(name,customerId);
        obj1.menu();
    }
}

class BankAccount{
    double bal;
    double prevTrans;
    String customerName;
    String customerId;

    BankAccount(String customerName,String customerId){
        this.customerName=customerName;
        this.customerId=customerId;
    }
```

```
void deposit(double amount){
        if(amount!=0){
            bal+=amount;
            prevTrans=amount;
        }
    }

    void withdraw(double amt){
        if(amt!=0 && bal>=amt){
            bal-=amt;
            prevTrans=-amt;
        }
        else if(bal<amt){
            System.out.println("Bank balance insufficient");
        }
    }

    void getPreviousTrans(){
        if(prevTrans>0){
            System.out.println("Deposited: "+prevTrans);
        }
        else if(prevTrans<0){
            System.out.println("Withdrawn: "+Math.abs(prevTrans));
        }
        else{
            System.out.println("No transaction occured");
        }
    }
```

```
void menu(){
        char option;
        Scanner sc=new Scanner(System.in);
        System.out.println("Welcome "+customerName);
        System.out.println("Your ID:"+customerId);
        System.out.println("\n");
        System.out.println("a) Check Balance");
        System.out.println("b) Deposit Amount");
        System.out.println("c) Withdraw Amount");
        System.out.println("d) Previous Transaction");
        System.out.println("e) Exit");

        do{
           System.out.println("*****************");
           System.out.println("Choose an option");
           option=sc.next().charAt(0);
           System.out.println("\n");

           switch (option){
              case 'a':
                 System.out.println(".....................");
                 System.out.println("Balance ="+bal);
                 System.out.println(".....................");
                 System.out.println("\n");
                 break;
              case 'b':
                 System.out.println(".....................");
                 System.out.println("Enter a amount to deposit :");
                 System.out.println(".....................");
                 double amt=sc.nextDouble();
                 deposit(amt);
                 System.out.println("\n");
                 break;
```

```
case 'c':
                System.out.println(".....");
                System.out.println("Enter a amount to Withdraw :");
                System.out.println(".....");
                double amtW=sc.nextDouble();
                withdraw(amtW);
                System.out.println("\n");
                break;
            case 'd':
                System.out.println(".....");
                System.out.println("Previous Transaction:");
                getPreviousTrans();
                System.out.println(".....");
                System.out.println("\n");
                break;

            case 'e':
                System.out.println(".....");
                break;
            default:
                System.out.println("Choose a correct option to proceed");
                break;
          }

      }while(option!='e');

      System.out.println("Thank you for using our banking services");
   }

}
```

conclusion

The application demonstrate the way to develop an online banking system by using interactive web client by using JSP, Servlet with more secure way to access.
This means the application server easily deployable and accessible.