

ONLINE BANKING APPLICATION

Submitted in partial fulfillment of the required for the award of

Bachelors of Computer Science

By

PRADEEP.D

(REGISTER NO 40290068)



DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF SCIENCE AND HUMANITIES

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC | 12B Status by UGC| Approved by AICTE

JEPPIAR NAGAR, RAJIV GANDHI SALAI, CHENNAI 600119

MAY 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **PRADEEP.D(40290068)** who carried out the project entitled "**ONLINE BANKING APPLICATION**" under our supervision from AUGUST 2022 TO NOVEMBER 2022.

6/5/2023
Internal Guide

Dr. ULAGAMUTHALVI.V M.E., Ph.D.

Rekha 5/9/23
DEAN AND HEAD OF THE DEPARTMENT

Dr. REKHA CHAKRAVARTHI M.E., Ph.D.

Submitted for Viva voce examination held on 09/06/2023

6/9/2023
Internal Examiner

B. Nishu Jay 9/5/23
External Examiner

DECLARATION

I am, PRADEEP.D (40290068) here by declare that the Project Report entitled "ONLINE BANKING APPLICATION" done by us under the guidance of Dr.ULAGAMUTHALVI.V M.E.,Ph.D. Department of Computer Science at SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY, Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai-600 119 is submitted in partial fulfillment of the requirements for the award of Master of Science degree in Computer Science.

DATE:

PLACE: CHENNAI


PRADEEP.D

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. REKHA CHAKRAVARTHI M.E., Ph.D. Dean of Arts and Science** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my project guide **Dr.ULAGAMUTHALVI.V M.E., Ph.D.** of Arts and Science and Head of the Department of Computer science for their valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff member of the Department of Computer Science who were helpful in many ways for the completion of the project.

PRADEEP.D

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
1	INTRODUCTION	1
	1.1 OVERVIEW OF THE PROJECT	1
	1.2 LITERATURE SURVEY	2
2	SYSTEM ANALYSIS	4
	2.1 EXISTING SYSTEM	4
	2.2 PROPOSED SYSTEM	4
	2.3 ADVANTAGES OF PROPOSED SYSTEM	4
	2.4 FEASIBILITY STUDY	5
	2.5 ECONOMIC FEASIBILITY	5
	2.6 OPERATIONAL FEASIBILITY	5
3	SYSTEM REQUIREMENTS	6
	3.1 REQUIREMENT ANALYSIS	6
	3.2 ABOUT JDK	6
	3.3 SOFTWARE REQUIREMENTS	7
	3.4 HARDWARE REQUIREMENTS	7

CHAPTER NO.	TITLE	PAGE NO.
4	SYSTEM DESIGN	8
	4.1 DATA FLOW DIAGRAM	8
	4.2 DATABASE TABLE	14
5	SYSTEM IMPLEMENTATIONS	15
	5.1 CIPHERTEXT	15
	5.2 ADVANCED ENCRYPTION STANDARD(AES)	15
6	TESTING	17
	6.1 INTRODUCTION	17
	6.2 SOFTWARE TESTING	17
	6.3 UNIT TESTING	18
	6.4 WHITE BOX TESTING	19
	6.5 CONDITIONAL TESTING	19
	6.6 DATA FLOW TESTING	19
7	CONCLUSION	20

8	APPENDIX	21
	8.1 SOURCE CODE	21
	8.2 OUTPUT	45
9	REFERENCE	50

LIST OF FIGURES

NO.	TITLE	PAGE NO.
4.1	Level 1 DFD	8
4.2	Level 2 DFD process-1	9
4.3	Level 2 DFD process-2	10
4.4	Level 2 DFD process-3	11
4.5	Level 2 DFD process-4	12
4.6	Level 3 DFD	13
6.1	Types of Testing	18
8.1	Home Page	45
8.2	Home Page	45
8.3	Crypto Value	46
8.4	Registration	47

8.5	Login Page	47
8.6	Money Transfer	47
8.7	Create New Account	48
8.8	Overall Transaction	48
8.9	Contact Us	49

LIST OF TABLES

NO.	TITLE	PAGE NO.
4.1	DataBase	14

ABSTRACT

The Bank Account Management System is an application for maintaining a person's account in a bank. In this project I tried to show the working of a online banking application and cover the basic functionality of a Bank Account Management System. To develop a project for solving financial applications of a customer in banking environment in order to nurture the needs of an end banking user by providing various ways to perform banking tasks. Also to enable the user's workspace to have additional functionalities which are not provided under a conventional banking project. This project is developed using PHP, HTML language and MYSQL use for database connection. This project allows the banks to customize their services which improves the quality of the customer service and experience. The main idea of this project is to develop an app which allows the user to login to their account and view their records without stepping out of their home. This app also saves all the sensitive data of the customer in an encrypted form. This is to ensure the customers that all of their datas are safe and secured. This app also serves as a platform where the bank can advertise their services to each customer based on the customer's preference. The system is then designed in accordance with specifications to satisfy the requirements. The system design is then implemented with MYSQL, PHP and HTML.

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

The "ONLINE BANKING APPLICATION" project is a model Internet Banking Site. This site enables the customers to perform the basic banking transactions by sitting at their office or at homes through PC or laptop. The system provides the access to the customer to create an account, deposit/withdraw the cash from his account, also to view reports of all accounts present. The customers can access the banks website for viewing their account details and perform the transactions on account as per their requirements. With Internet Banking, the brick and mortar structure of the traditional banking gets converted into a click and portal model, thereby giving a concept of virtual banking a real shape. Thus today's banking is no longer confined to branches.

ONLINE BANKING APPLICATION keeps the day by day tally record as a complete banking system. It can keep the information of Account type, account opening form, Deposit fund, Withdrawal, and Searching the transaction, Transaction reports, Individual account opening form, Group Account, it displays Transaction reports, Statistical Summary of Account type and Interest Information.

The main aim of designing and developing this Internet banking System PHP primarily based Engineering project is to provide secure and efficient net banking facilities to the banking customers over the internet. Apache Server Pages, MYSQL database used to develop this bank application where all banking customers can login through the secured web page by their account login id and password. Users will have all options and features in that application like get money from western union, money transfer to others, and send cash or money to inter banking as well as other banking customers by simply adding them as payees.

The main aim of designing and developing this Internet banking System PHP primarily based Engineering project is to provide secure and efficient net banking facilities to the

banking customers over the internet. Apache Server Pages, MYSQL database used to develop this bank application where all banking customers can login through the secured web page by their account login id and password. Users will have all options and features in that application like get money from western union, money transfer to others, and send cash or money to inter banking as well as other banking customers by simply adding them as payees.

The Traditional way of maintaining details of a user in a bank was to enter the details and record them. Every time the user needs to perform some transactions he has to go to bank and perform the necessary actions, which may not be so feasible all the time. It may be a hardhitting task for the users and the bankers too. The project gives real life understanding of Online Banking System and activities performed by various roles in the supply chain. Here, we provide automation for banking system through Internet. Online Banking System project captures activities performed by different roles in real life banking which provides enhanced techniques for maintaining the required information up-to-date, which results in efficiency. The project gives real life understanding of Online Banking System and activities performed by various roles in the supply chain.

1.2 LITERATURE SURVEY

The advent of internet technology has revolutionised the way business and services are provided by companies and businesses all over the world. It provided innumerable service innovations for the consumer (Sabi, 2014).

The technological change has brought the rapid transmission of information, the easy way for marketing banking products and enhanced the customer's access and awareness (George & Gireeshkumar, 2012).

This model views Internet Banking adoption as a social construct and it moves through the population over time. Individuals have different degrees of willingness to adopt an innovation as Internet Banking (Hanafizadeh et al., 2014)

In the theoretical framework from Rogers (1995), the rate of Internet Banking adoption is impacted by five characteristics. The characteristics are relative advantage, compatibility, complexity, trialability and observability. This model is often used in information systems research to explain user adoption of new technologies (Sinha & Mukherjee, 2016).

The relative advantage indicates the degree of how an innovation is perceived and to which degree it provides more benefits than its forerunner (Moore & Benbasat, 1991). Relative advantage has a positive influence on the adoption of new innovations (Holak & Lehmann, 1990). In the field of Internet Banking the relative advantage is the individual's belief that Internet Banking provides more economic benefits than the traditional ways of banking (Yousafzai, 2012). Internet Banking allows the customer to access their banking accounts at any time and from any location. It provides tremendous advantage to the customer. The relative advantage has been identified as a major determinant of a customer's intention to use Internet Banking. (Lichtenstein & Williamson, 2006; Nerme et al., 2013; Tan & Teo, 2000). Important for the financial institutions is the communication of the relative advantage to their customers. The customers need to be aware of the advantages (Lichtenstein & Williamson, 2006).

CHAPTER 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

The existing system is having many problems such as security problems, more human involvement which is a time consuming process with many manual calculations. It even includes the machine damage and signature verification process for secured transactions which allows the customers and banks to waste their valuable time and resources. The major problem in online banking system is unauthorized user access with fake passwords. The hackers are trying to hack the user accounts and are performing different unauthorized transactions.

2.2 PROPOSED SYSTEM

The proposed system is highly computerized in which the data related to user accounts will be secured high with high accuracy that even reduces the machine damage and human made errors and this existing system is highly efficient to offer best services to the customers as well as banks because it has user friendly access that consumes less time when compared with normal banking system performing different unauthorized transactions.

2.3 ADVANTAGES OF PROPOSED SYSTEM

- Customer Satisfaction
- Saving Customer's time
- Protecting the data
- Data's are encrypted

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

2.5 ECONOMIC FEASIBILITY

The online banking system is economically feasible as it builds on an already existing system and the cost of hardware resources needed is relatively low, software applications needed are readily available making the project budget to be manageable.

2.6 OPERATIONAL FEASIBILITY

The system provides very necessary information needed by user and provides them with efficient guide as to what they are to enter into the system's database. In addition to this, this application is expected to reduce the many security risks and much workloads currently being experienced at the bank.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 REQUIREMENT ANALYSIS

Requirements are a feature of a system or description of something that the system is capable of doing in order to fulfil the system's purpose. It provides the appropriate mechanism for understanding what the customer wants, analyzing the needs assessing feasibility, negotiating a reasonable solution, specifying the solution unambiguously, validating the specification and managing the requirements as they are translated into an operational system.

3.2 ABOUT JDK

The **Java Development Kit (JDK)** is a Sun Microsystems product aimed at Java developers. Since the introduction of Java, it has been by far the most widely used Java SDK. On 17 November 2006, Sun announced that it would be released under the GNU General Public License (GPL), thus making it free software. This happened in large part on 8 May 2007⁽¹⁾ and the source code was contributed to the OpenJDK.

The primary components of the JDK are a selection of programming tools, including:

- java – The loader for Java applications. This tool is an interpreter and can interpret the class files generated by the javac compiler. Now a single launcher is used for both development and deployment. The old deployment launcher, jre, is no longer provided with Sun JDK.
- javac – The compiler, which converts source code into Java bytecode
- jar – The archiver, which packages related class libraries into a single JAR file. This tool also helps manage JAR files.
- avadoc – The documentation generator, which automatically generates documentation from source code comments
- jdb – The debugger
- javap – The class file disassembler
- appletviewer – This tool can be used to run and debug Java applets without a web browser.
- javah – The C header and stub generator, used to write native methods
- extcheck – This utility can detect JAR-file conflicts.
- apt – The annotation processing tool
- jhat – (Experimental) Java heap analysis tool
- jstack – (Experimental) This utility prints Java stack traces of Java threads.

- jstat – (Experimental) Java Virtual Machine statistics monitoring tool
- jstatd – (Experimental) jstat daemon
- jinfo – (Experimental) This utility gets configuration information from a running Java process or crash dump.
- jmap – (Experimental) This utility outputs the memory map for Java and can print shared object memory maps or heap memory details of a given process or core dump.
- idlj – The IDL-to-Java compiler. This utility generates Java bindings from a given IDL file.
- policy tool – The policy creation and management tool, which can determine policy for a Java runtime, specifying which permissions are available for code from various sources

VisualVM – visual tool integrating several command line JDK tools and lightweight performance and memory profiling capabilities

3.3 SOFTWARE REQUIREMENTS

- Coding Language : JAVA
- Database : SQLyog community
- Tool : NetBeans IDE 8.2
- Operating System : Windows 7, 8, 9, 10, XP

3.4 HARDWARE REQUIREMENTS

- Processor : Intel
- Main Memory(RAM) : 256 MB
- Cache Memory : 512 KB
- Monitor : 14 inch Color Monitor
- Mouse : Optical Mouse

CHAPTER 4

SYSTEM DESIGN

4.1 DATAFLOW DIAGRAM

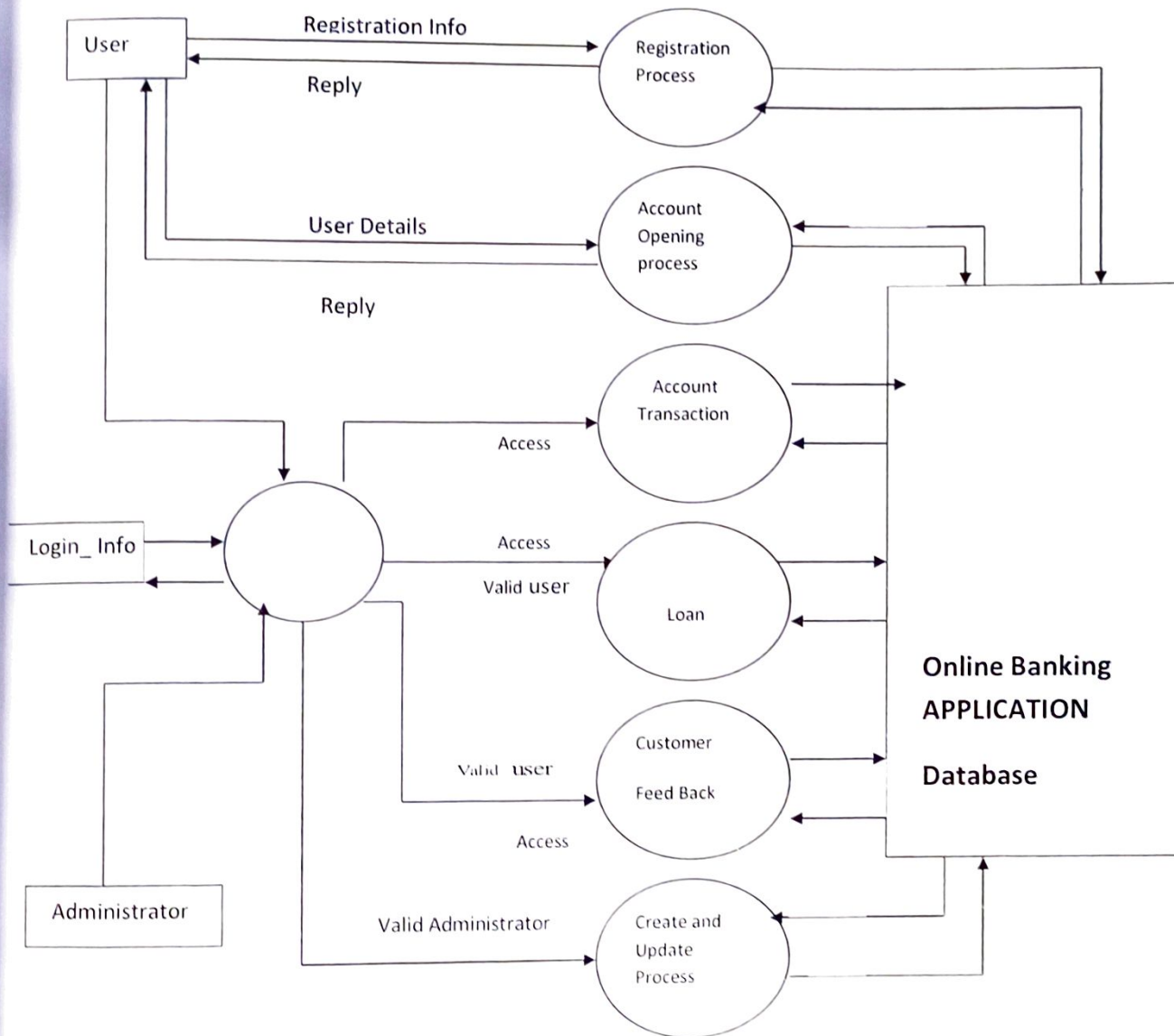


Fig4.1 Level 1 DFD

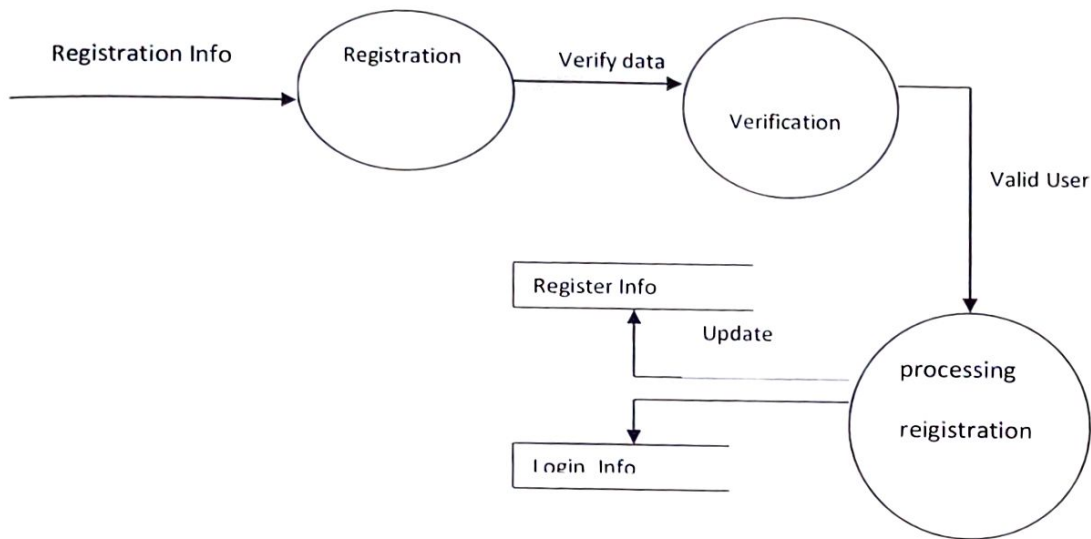


Fig4.2 Level 2 DFD process-1

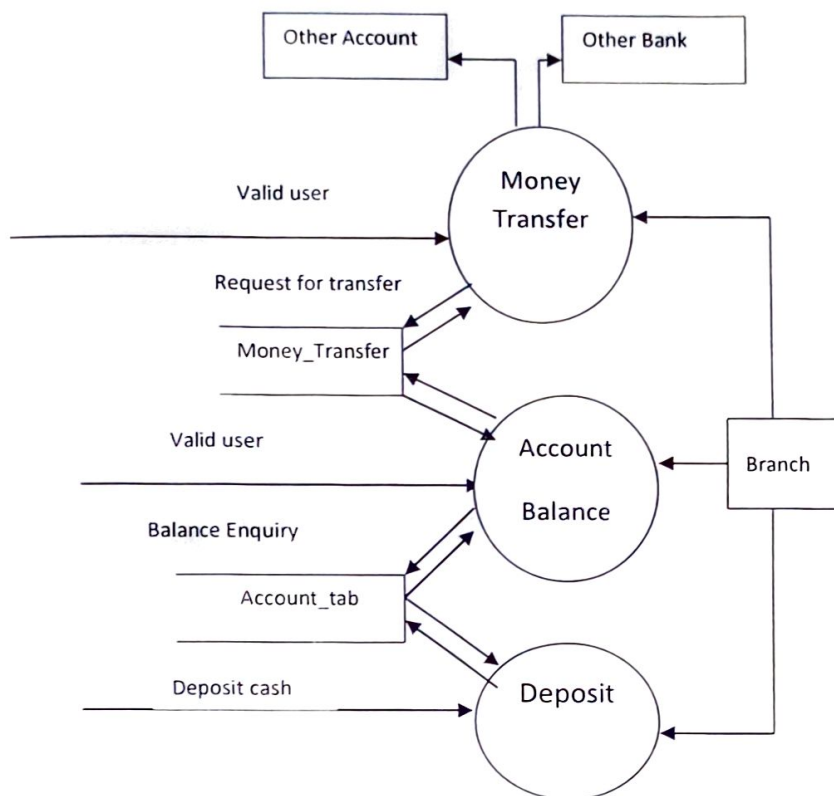


Fig4.3 Level 2 DFD process-2

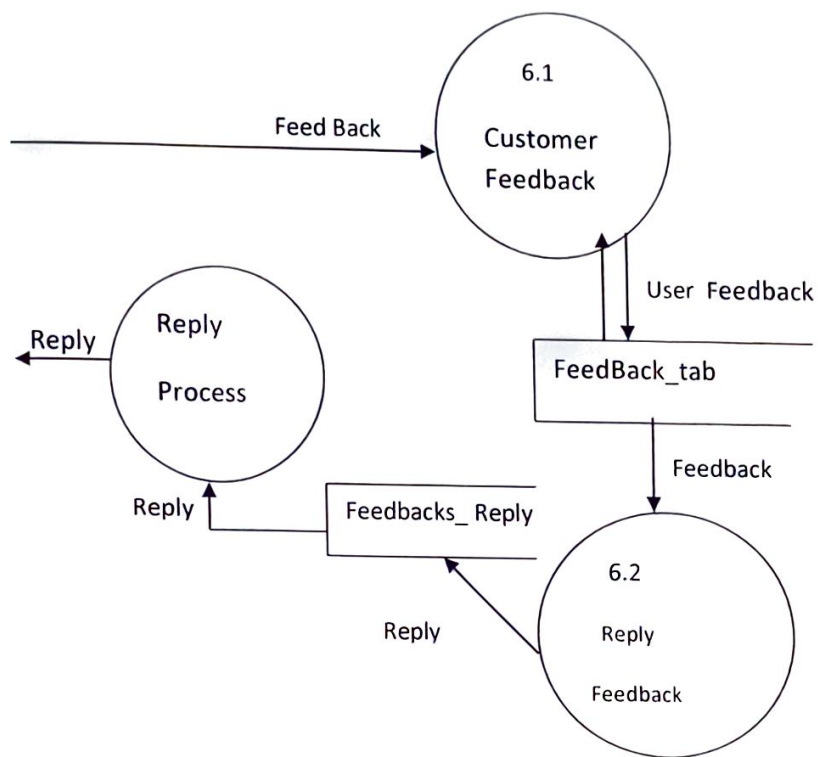


Fig4.4 Level 2 DFD process-3

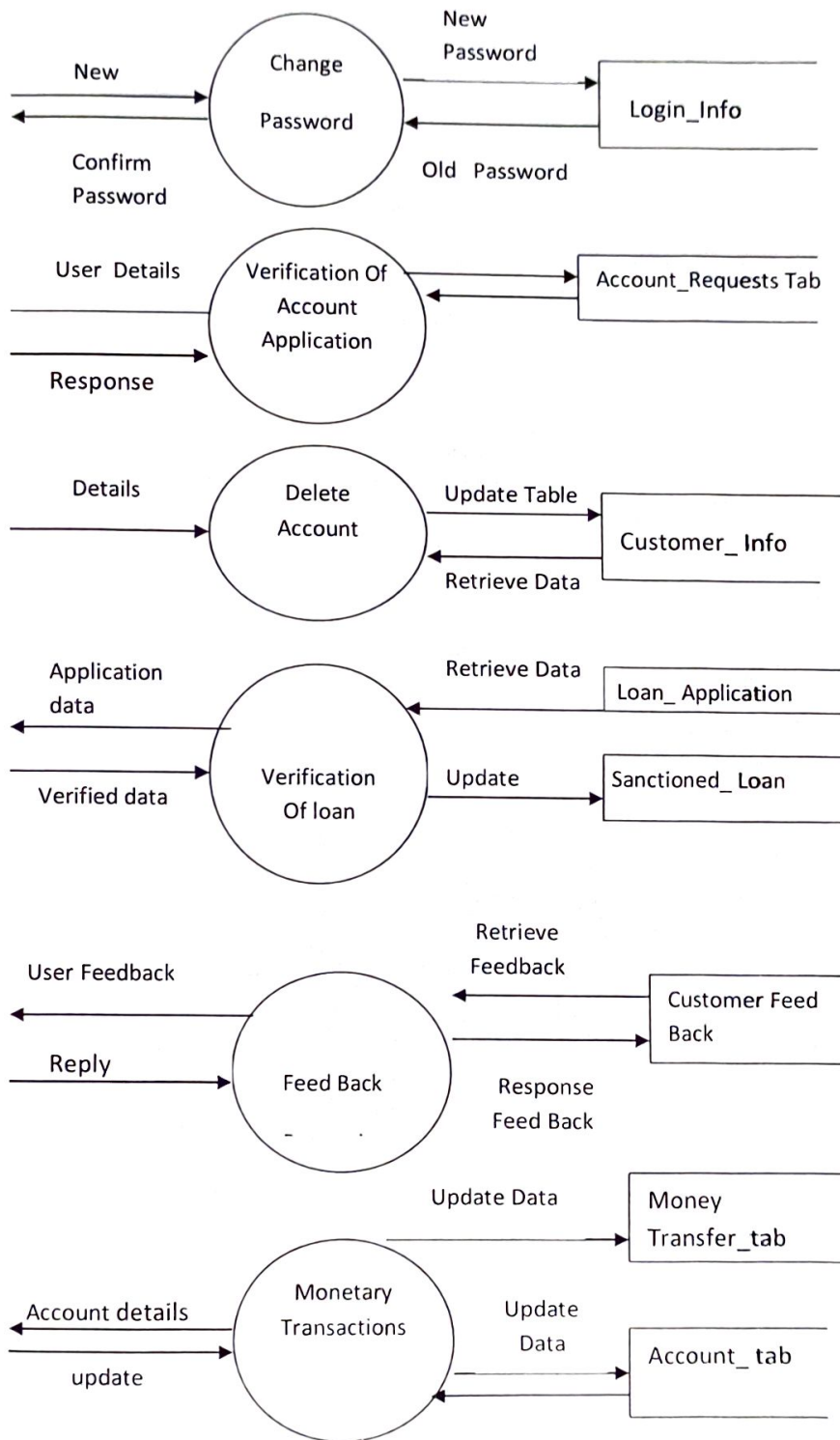


Fig4.5 Level 2 DFD process-4

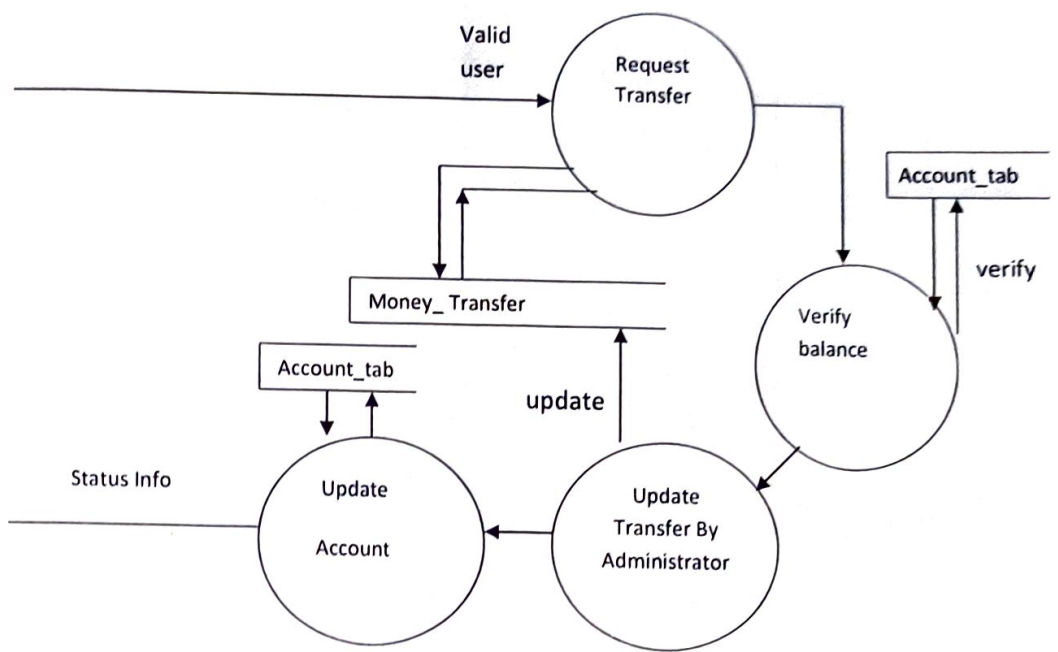


Fig4.6 Level 3 DFD

4.2 DATABASE TABLE

Column name	Data type	Nullable	Primary key
ACCOUNTINFO	Number	No	Yes
USERNAME	Varchar2	Yes	No
PASSWORD	Varchar2	Yes	No
AMOUNT	Varchar2	Yes	No
ADDRESS	Varchar2	Yes	No
PHONE	Varchar2	Yes	No

Table4.1 DataBase

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 CIPHERTEXT

Ciphertext is also known as encrypted or encoded information because it contains a version of the original plaintext that cannot be read by a person or computer without the proper cypher to decrypt it.

Decryption is the process of turning unreadable ciphertext into plaintext, which is the opposite of encryption. Symmetric ciphers, which are typically used to secure online communications, are incorporated into many different network protocols to be used to encrypt exchanges.

For example, Transport Layer Security uses ciphers to encrypt application layer data. Virtual private networks connecting remote workers or remote branches into corporate networks use protocols with symmetric ciphers to protect data communications. Symmetric ciphers protect data privacy in most Wi-Fi networks, online banking, ecommerce services and mobile telephony.

Other protocols, including secure shell, OpenPGP and Secure/Multipurpose Internet Mail Extensions use asymmetric cryptography to encrypt and authenticate endpoints but also to securely exchange the symmetric keys to encrypt session data. For performance reasons, protocols often rely on ciphers to encrypt session data.

5.2 AES(Advanced Encryption Standard)

Because the AES technique uses a symmetric key, the same the data is encrypted and decrypted using a key. To encrypt sensitive data, AES is utilised widely in hardware and software.

- AES-128 uses a 128-bit key length to encrypt and decrypt a block of messages.
- AES-192 uses a 192-bit key length to encrypt and decrypt a block of messages.
- AES-256 uses a 256-bit key length to encrypt and decrypt a block of messages.

The AES encryption algorithm defines numerous transformations that are to be performed on data stored in an array. The first step of the cipher is to put the data into an array, after which the cipher transformations are repeated over multiple encryption rounds. The first transformation in the AES encryption cipher is substitution of data using a substitution table. The second transformation shifts data rows. The third mixes columns. The last transformation is performed on each column using a different part of the encryption key. Longer keys need more rounds to complete.

Security-Competing algorithms were to be judged on their ability to resist attack as compared to other submitted ciphers. Security strength was to be considered the most important factor in the competition.

Cost-Intended to be released on a global, nonexclusive and royalty-free basis, the candidate algorithms were to be evaluated on computational and memory efficiency.

Implementation-Factors to be considered included the algorithm's flexibility, suitability for hardware or software implementation, and overall simplicity

CHAPTER 6

TESTING

6.1 INTRODUCTION

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a well planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

6.2 SOFTWARE TESTING

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture.

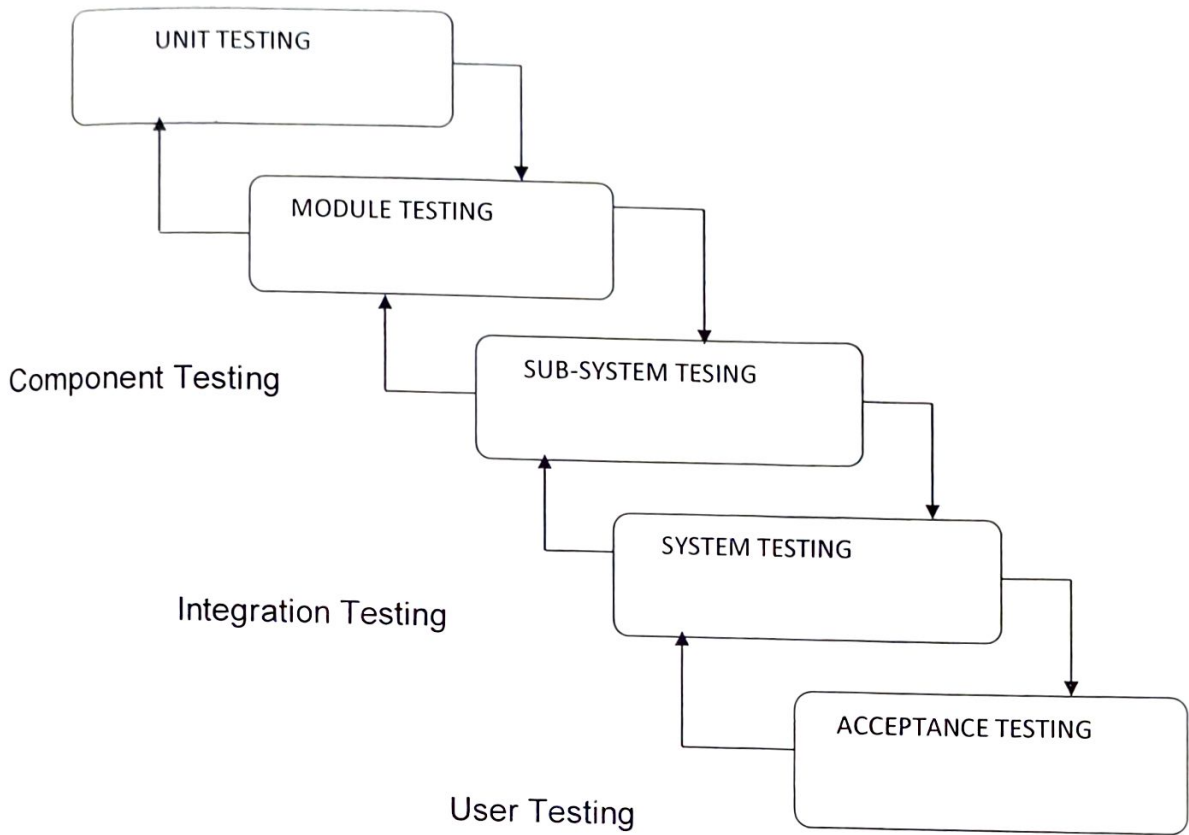


Fig6.1 Types of Testing

6.3 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

6.4 WHITE BOX TESTING

This type of testing ensures that

- All independent paths have been exercised at least once
- All logical decisions have been exercised on their true and false sides
- All loops are executed at their boundaries and within their operational bounds
- All internal data structures have been exercised to assure their validity.

To follow the concept of white box testing we have tested each form .we have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, All loops are executed on their boundaries.

6.5 CONDITIONAL TESTING

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generate on particular condition is traced to uncover any possible errors.

6.6 DATA FLOW TESTING

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variable were declared. The *definition-use chain* method was used in this type of testing. These were particularly useful in nested statements.

CHAPTER 7

CONCLUSION

This project is developed to nurture the needs of a user in a banking sector by embedding all the tasks of transactions taking place in bank. Future version of this project will still be much enhanced than the current version. If proper training should be given to customer by the bank employs to open an account will be beneficial secondly the website should be made friendlier from where the first time customers can directly make and access their accounts.

CHAPTER 8

APPENDIX

8.1 SOURCE CODE

```
<%@page import="java.sql.DriverManager"%>

<%@page import="java.sql.ResultSet"%>

<%@page import="java.sql.PreparedStatement"%>

<%@page import="java.sql.Connection"%>


<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

    <head>

        <!-- basic -->

        <meta charset="utf-8">

        <meta http-equiv="X-UA-Compatible" content="IE=edge">

        <!-- mobile metas -->

        <meta name="viewport" content="width=device-width, initial-scale=1">

        <meta name="viewport" content="initial-scale=1, maximum-scale=1">

        <!-- site metas -->
```

```

<title>bitcypo</title>

<meta name="keywords" content="">

<meta name="description" content="">

<meta name="author" content="">

<!-- bootstrap css -->

<link rel="stylesheet" href="css/bootstrap.min.css">

<!-- style css -->

<link rel="stylesheet" href="css/style.css">

<!-- responsive-->

<link rel="stylesheet" href="css/responsive.css">

<!-- awesome fontfamily -->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">

<!--[if lt IE 9]>

<script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>

<script
src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script><![endif]-->

</head>

<!-- body -->

<body class="main-layout">

<!-- loader -->

```

```
<div class="loader_bg">
```

```
  <div class="loader"></div>
```

```
</div>
```

```
<!-- end loader -->
```

```
<!-- header -->
```

```
<header>
```

```
  <!-- header inner -->
```

```
  <div class="head-top">
```

```
    <div class="container-fluid">
```

```
      <div class="row d_flex">
```

```
        <div class="col-sm-3">
```

```
          <div class="logo">
```

```
            <a href="index.html">Banking</a>
```

```
          </div>
```

```
        </div>
```

```
  <style>
```

```
    .nav{
```

```
color:white;

padding:10px;

margin-left:600px;
```

```
}
```

```
a{
```

```
color:white;
```

```
padding:10px; }
```

```
</style>
```

```
<div class="nav">
```

```
    <a href="user.jsp">Home</a>
```

```
    <a href="add.jsp" >Add bank</a>
```

```
    <a href="view.jsp" >View amount </a>
```

```
    <a href="send.jsp" >Send Money </a>
```

```
    <a href="trans.jsp" >Overall transaction</a>
```

```
    <a href="index.html">Logout</a>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</header>
```

```
<!-- end header -->
```

```
<!-- start slider section -->
```

```
<div id="top_section" class=" banner_main">
```

```
  <div class="container-fluid">
```

```
    <div class="contact">
```

```
  <div class="container">
```

```
    <div class="row">
```

```
      <div class="col-md-12">
```

```
    </div>
```

```
  </div>
```

```
<div class="row">
```

```
  <div class="col-md-12">
```

```
    <div class="titlepage text_align_center">
```

```
      <h2 style=" font-size:24px;">View Overall Details</h2>
```

```
    </div>
```

```
  </div>
```

```
<div class=" col-md-15 offset-md-1">
```

```
  <%
```

```
String user=session.getAttribute("user").toString();
```

%>

<style>

.contactusi {

border: #17a2b8 solid 1px;

padding: 0 15px;

margin-bottom: 25px;

width:30%;

height: 40px;

background: transparent;

color: #5e687d;

font-size: 12px;

font-weight: normal;

margin-left:330px;

display: Inline-box;

text-align: center;

}

label{

display: Inline-box;

max-width: 120px;

margin-left: 300px;

```
}  
table.tr.th,td{
```

```
border:2px solid #17a2b8;
```

```
color: #17a2b8;
```

```
margin-left:200px;
```

```
padding: 5px;
```

```
}
```

```
th{
```

```
color: #5e687d;
```

```
}
```

```
</style>
```

```
<div class="row">
```

```
<table>
```

```
<tr>
```

```
<th> User name</th>
```

```
<th>Card No</th>
```

```
<th>Account No</th>
```

```
<th>IFSC No</th>
```

<th>Branch</th>

<th>Amount</th>

</tr>

<%

```
Class.forName("com.mysql.jdbc.Driver");
```

```
Connection
```

```
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/banking","root","root");
```

```
PreparedStatement ps=con.prepareStatement("Select * from card where  
user='"+user+"' ");
```

```
ResultSet rs=ps.executeQuery();
```

```
System.out.println(rs);
```

```
while(rs.next())
```

```
{
```

```
String cno=rs.getString("cno");
```

```
String acno=rs.getString("acno");
```

```
String ifsc=rs.getString("ifsc");
```

```
String branch=rs.getString("branch");
```

```
String amount=rs.getString("amount");
```

```
%>
```

```
<tr>

    <td><%=user%></a></td>

    <td><%=cno%></td>

    <td><%=acno%></td>

    <td><%=ifsc%></td>

    <td><%=branch%></td>

    <td><%=amount%></td>

</tr><%}%>

</table>

</div>
```

```
</div>
```

```
<br>
```

```
<br>
```

```
<br>
```

```
<br>
```

```
<br>
```


</div>

</div>

</div>

</footer>

<!-- end footer -->

<!-- Javascript files-->

<script src="js/jquery.min.js "></script>

<script src="js/bootstrap.bundle.min.js "></script>

<script src="js/jquery-3.0.0.min.js "></script>

<script src="js/custom.js "></script>

</body>

</html>

CreateServlet.java

package g;

import java.io.IOException;

import java.io.PrintWriter;

```
import java.rmi.Naming;
```

```
import javax.servlet.RequestDispatcher;
```

```
import javax.servlet.ServletException;
```

```
import javax.servlet.http.HttpServlet;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.HttpServletResponse;
```

```
public class CreateServlet extends HttpServlet {
```

```
    public void doGet(HttpServletRequest request, HttpServletResponse response)
```

```
        throws ServletException, IOException {
```

```
        response.setContentType("text/html");
```

```
        PrintWriter out=response.getWriter();
```

```
        String username=request.getParameter("username");
```

```
        String password=request.getParameter("password");
```

```
        String repassword=request.getParameter("repassword");
```

```
        String amoun=request.getParameter("amount");
```

```
        double amount=Double.parseDouble(amoun);
```

```
String address=request.getParameter("address");
```

```
String ph=request.getParameter("phone");
```

```
double phone=Double.parseDouble(ph);
```

```
//double mname=Double.parseDouble(num);
```

```
//String country=request.getParameter("country");
```

```
int status=RegisterUser.register(username, password, repassword, amount,  
address,phone);
```

```
if(status>0){
```

```
    out.print("WELCOME! YOUR ACCOUNT HAS OPENED");
```

```
    RequestDispatcher rd=request.getRequestDispatcher("index.jsp");
```

```
    rd.include(request, response);
```

```
}
```

```
else{
```

```
    out.print("Sorry,Registration failed. please try later");
```

```
    RequestDispatcher
```

```
rd=request.getRequestDispatcher("MyHtml.html");
```

```
    rd.include(request, response);
```

```
}
```

```
out.close();
```

```
}}
```

DBInitializer.java

package g;

public interface DBInitializer {

String DRIVER="oracle.jdbc.driver.OracleDriver";

String CON_STRING="jdbc:oracle:thin:@localhost:1521:xe";

String USERNAME="system";

String PASSWORD="oracle";

}

Details.java

package g;

// THIS PROGRAM GBANK IS THE INTERFACE FOR THE MAIN PROGRAM

import java.rmi.*;

public interface Details extends Remote

{

public int open(String username,String password,double amount,String address,double phone) throws RemoteException;

public String withdraw(int acno,String uname,String pwd,int amt) throws RemoteException;

public String deposit(int acno,String uname,String pwd,int amt) throws RemoteException;

```

    public String transfer(int acno,String uname,String pwd,int tacno,int amt) throws
RemoteException;
    public String close(int acno,String uname,String pass) throws RemoteException;
    public String balance(int acno,String uname,String pass) throws RemoteException;
}

```

GetCon.java

```

package g;
import java.sql.*;

```

```

public class GetCon {
    private GetCon(){}

```

```

    public static Connection con;

```

```

    static{

```

```

        try {

```

```

            Class.forName(DBInitializer.DRIVER);

```

```

            con=DriverManager.getConnection(DBInitializer.CON_STRING,DBInitializer.USE
RNAME,DBInitializer.PASSWORD);

```

```

        } catch (ClassNotFoundException e) {

```

```

            e.printStackTrace();

```

```

        } catch (SQLException e) {

```

```

            System.out.println("Exception in GetCon");

```

```
}  
  
}   
public static Connection getCon(){  
    return con;  
}  
}
```

```
public static int getPrimaryKey(){  
    int nextvalue=0;  
    Connection con=GetCon.getCon();  
    PreparedStatement ps2;  
    try {  
  
        ps2=con.prepareStatement("select javatpointnewaccount.nextval from dual");  
  
        ResultSet rs=ps2.executeQuery();  
        rs.next();  
        nextvalue=rs.getInt(1);  
  
    } catch (SQLException e) {
```

```
e.printStackTrace();
```

```
}
```

```
return nextvalue;
```

```
}
```

```
}
```

MyListener.java

```
package g;
```

```
import javax.servlet.*;
```

```
import java.sql.*;
```

```
public class MyListener implements ServletContextListener{
```

```
    public void contextInitialized(ServletContextEvent arg0) {
```

```
        int status=0;
```

```
        Connection con=null;
```

```
        try{
```

```
            con=GetCon.getCon();
```

```
            PreparedStatement ps1=con.prepareStatement("Select * from  
NEWACCOUNT");
```

```

try{
    status=ps1.executeUpdate();
}

catch(Exception e)
{e.printStackTrace();
    status=2;
    System.out.println("my staus is1111111"+status);
}

if(status==0)
    {System.out.println("your table name already exist"+status);}

else if(status==2)

    {System.out.println("else if part table does not exist new table has
created"+status);

        PreparedStatement ps3=con.prepareStatement("CREATE SEQUENCE
javatpointnewaccount MINVALUE 1 MAXVALUE 999999999999 INCREMENT BY 1
START WITH 1 NOCACHE NOORDER NOCYCLE");

        ps3.executeUpdate();

        PreparedStatement ps=con.prepareStatement("CREATE TABLE
NEWACCOUNT(ACCOUNTNONUMBER,USERNAMEVARCHAR2(4000),PASSWORD

```

```

VARCHAR2(4000),REPASSWORDVARCHAR2(4000),AMOUNTNUMBER,ADDRESS
VARCHAR2(4000),PHONE NUMBER,PRIMARY KEY (ACCOUNTNO))");

```

```

    ps.executeUpdate(); }

```

```

else{System.out.println("else part "+status);

```

```

    }}

```

```

catch(Exception e){e.printStackTrace();}

```

```

}

```

```

public void contextDestroyed(ServletContextEvent arg0) {

```

```

    System.out.println("project undeployed");

```

```

}

```

```

}

```

RegisterUser.java

```

package g;

```

```

import java.sql.*;

```

```

import g.GetCon;

```

```

public class RegisterUser {

```

```

    static int status=0;

```

```

    //int accountno=1;

```

```

    public static int register(String username,String password,String repassword,double
    amount,String address,double phone){

```

```

        //public static int register(String email,String password,String gender,String
        country,String name){

```

```
Connection con=GetCon.getCon();
```

```
PreparedStatement ps;
```

```
try {
```

```
    ps = con.prepareStatement("Insert into NEWACCOUNT  
values(?,?,?,?,?,?,?)");
```

```
    int nextvalue1=GetCon.getPrimaryKey();
```

```
    ps.setInt(1,nextvalue1);
```

```
    ps.setString(2,username);
```

```
    ps.setString(3,password);
```

```
    ps.setString(4,repassword);
```

```
    ps.setDouble(5,amount);
```

```
    ps.setString(6,address);
```

```
    ps.setDouble(7,phone);
```

```
    status=ps.executeUpdate();
```

```
} catch (SQLException e) {
```

```
    e.printStackTrace();
```

```
}
```

```
return status;
```

```
}
```

```
}
```

verifyLogin1.java

package g;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class verifyLogin1 {

public static boolean checkLogin(int accountno,String username,String password){

boolean status=false;

Connection con=GetCon.getCon();

try {

PreparedStatement ps=con.prepareStatement("Select * from MAILCASTINGUSER where EMAILADD = ? and PASSWORD =?");

PreparedStatement ps=con.prepareStatement("Select * from NEWACCOUNT where accountno=? and username = ? and password =?");

ps.setInt(1,accountno);

ps.setString(2,username);

ps.setString(3,password);

ResultSet rs=ps.executeQuery();

status=rs.next();

} catch (SQLException e) {

e.printStackTrace();

}

```
return status;
```

```
}
```

```
}
```

WEB.XML

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

```
<web-app version="2.5"
```

```
  xmlns="http://java.sun.com/xml/ns/javaee"
```

```
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
```

```
    http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
```

```
  <welcome-file-list>
```

```
    <welcome-file>index.html</welcome-file>
```

```
  </welcome-file-list>
```

```
  <servlet>
```

```
    <description>This is the description of my J2EE component</description>
```

```
    <display-name>This is the display name of my J2EE component</display-name>
```

```
    <servlet-name>CreateServlet</servlet-name>
```

```
    <servlet-class>g.CreateServlet</servlet-class>
```

```
  </servlet>
```

```
  <servlet-mapping>
```

```
    <servlet-name>CreateServlet</servlet-name>
```

```

<url-pattern>/CreateServlet</url-pattern>
</servlet-mapping>

<servlet>
  <description>This is the description of my J2EE component</description>
  <display-name>This is the display name of my J2EE component</display-name>
  <servlet-name>registereduserServlet</servlet-name>
  <servlet-class>g.registereduserServlet</servlet-class>
</servlet>

<servlet>
  <description>This is the description of my J2EE component</description>
  <display-name>This is the display name of my J2EE component</display-name>
  <servlet-name>verifyLogin</servlet-name>
  <servlet-class>verifyLogin</servlet-class>
</servlet>

<servlet>
  <description>This is the description of my J2EE component</description>
  <display-name>This is the display name of my J2EE component</display-name>
  <servlet-name>VerifyLogin1</servlet-name>
  <servlet-class>g.VerifyLogin1</servlet-class>
</servlet>

<servlet>
  <description>This is the description of my J2EE component</description>
  <display-name>This is the display name of my J2EE component</display-name>
  <servlet-name>verifyLogin1</servlet-name>

```

```
<servlet-class>g.verifyLogin1</servlet-class>
</servlet>
<servlet>
  <description>This is the description of my J2EE component</description>
  <display-name>This is the display name of my J2EE component</display-name>
  <servlet-name>GetCon</servlet-name>
  <servlet-class>GetCon</servlet-class>
</servlet>
```

```
<servlet-mapping>
  <servlet-name>registereduserServlet</servlet-name>
  <url-pattern>/go</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>verifyLogin1</servlet-name>
  <url-pattern>/servlet/verifyLogin1</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>VerifyLogin1</servlet-name>
  <url-pattern>/servlet/VerifyLogin1</url-pattern>
</servlet-mapping>
<servlet-mapping>
```

```
<servlet-name>verifyLogin1</servlet-name>  
<url-pattern>/servlet/verifyLogin1</url-pattern>  
</servlet-mapping>  
<servlet-mapping>  
  <servlet-name>GetCon</servlet-name>  
  <url-pattern>/servlet/GetCon</url-pattern>  
</servlet-mapping>
```

```
<listener>  
<listener-class>g.MyListener</listener-class>  
</listener>
```

```
</web-app>
```

8.2 OUTPUT



Fig8.1 Home Page

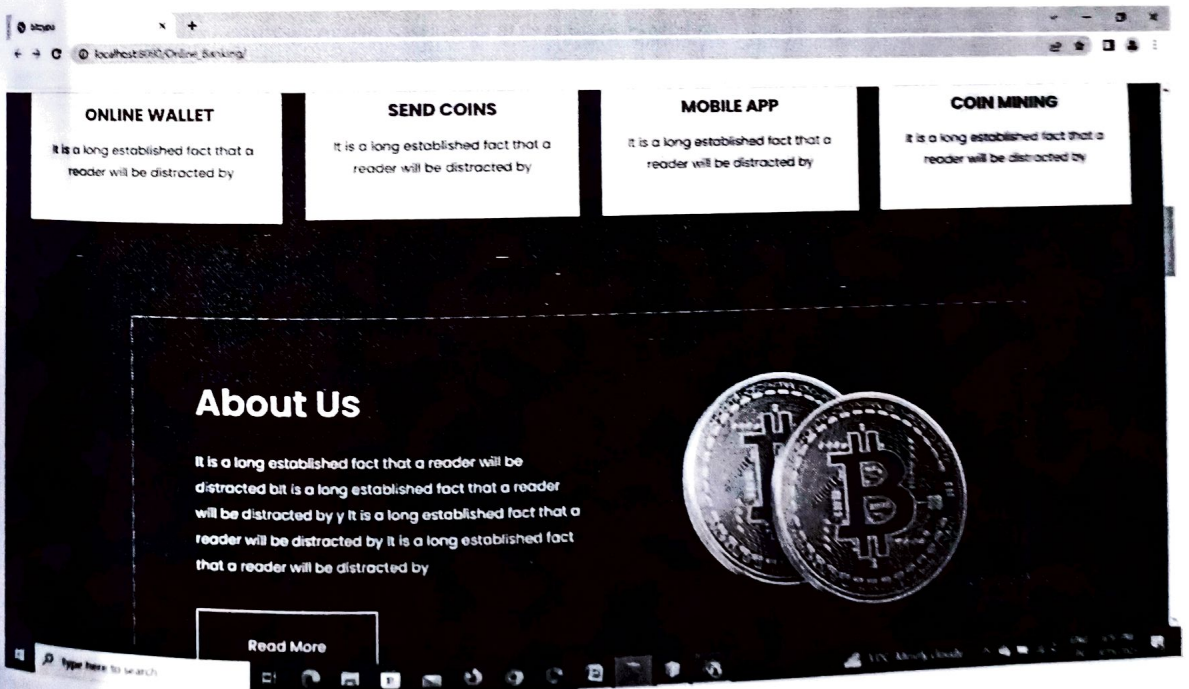


Fig8.2 Home Page



Fig8.3 Crypto Value

The screenshot shows a 'New Register' form on a web application. The form is set against a dark, textured background. It contains five input fields: 'User Name', 'Password', 'Email', 'Phone number', and 'Location'. A 'SUBMIT' button is located at the bottom of the form. The Windows taskbar is visible at the bottom.

Fig8.4 Registration

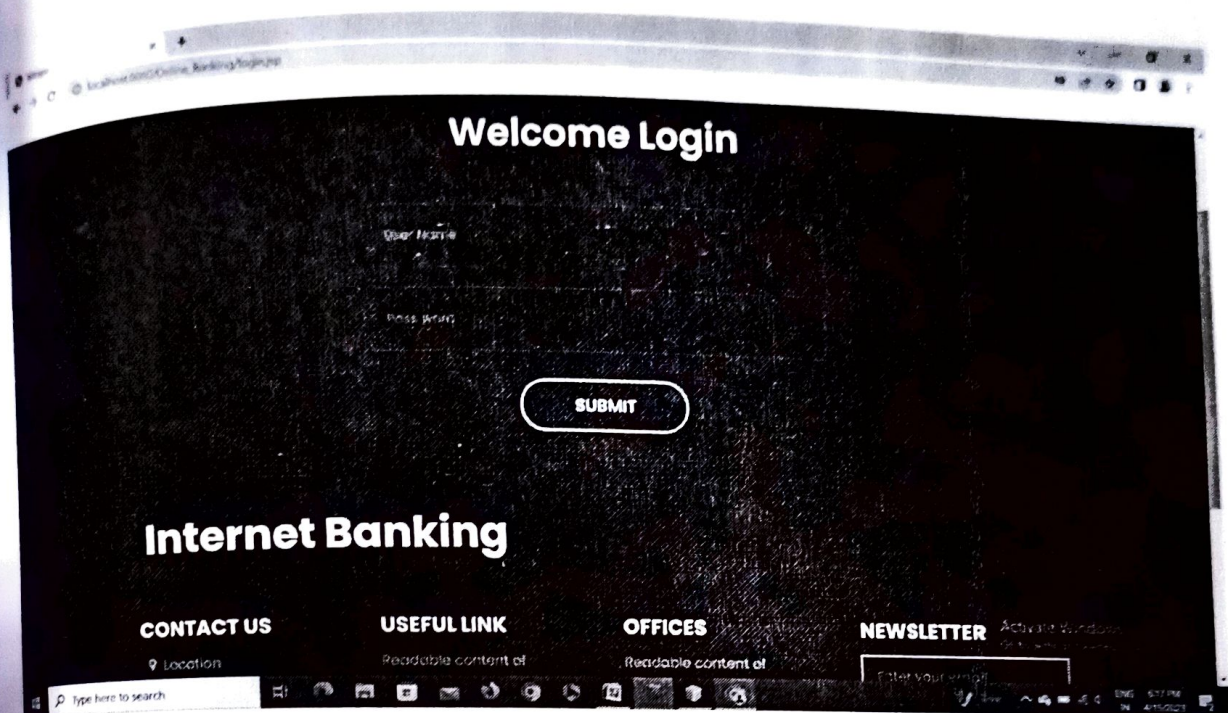


Fig8.5 Login Page

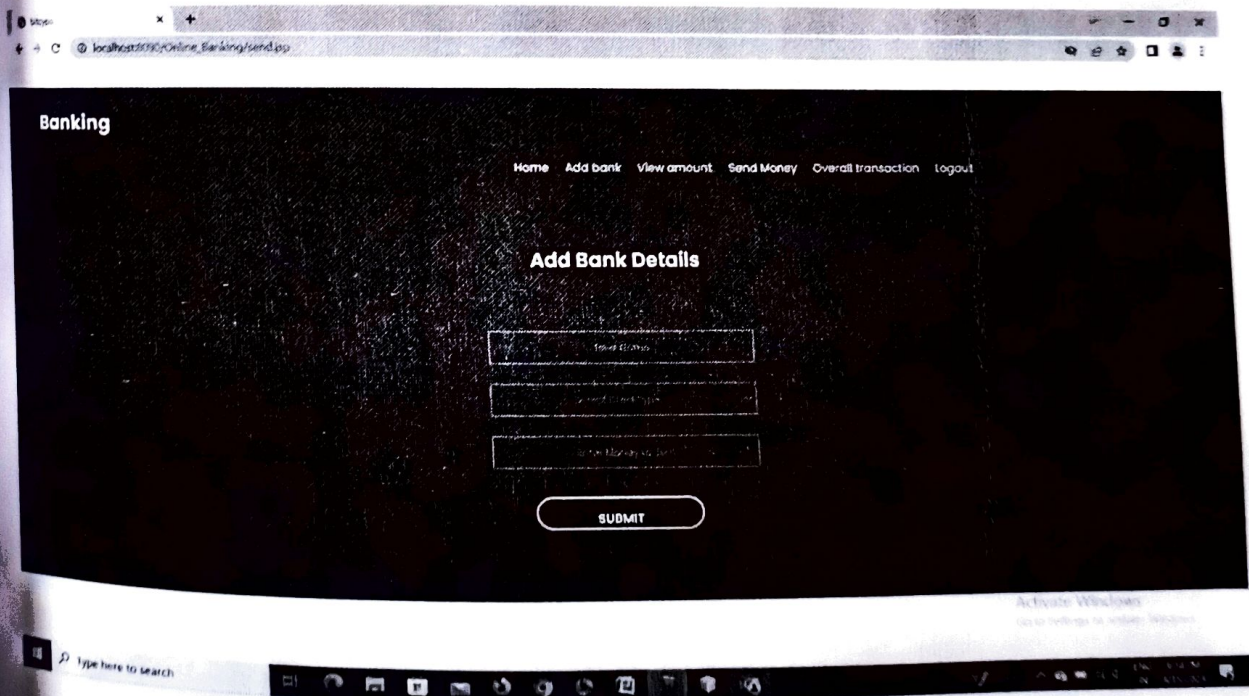


Fig8.6 Money Transfer

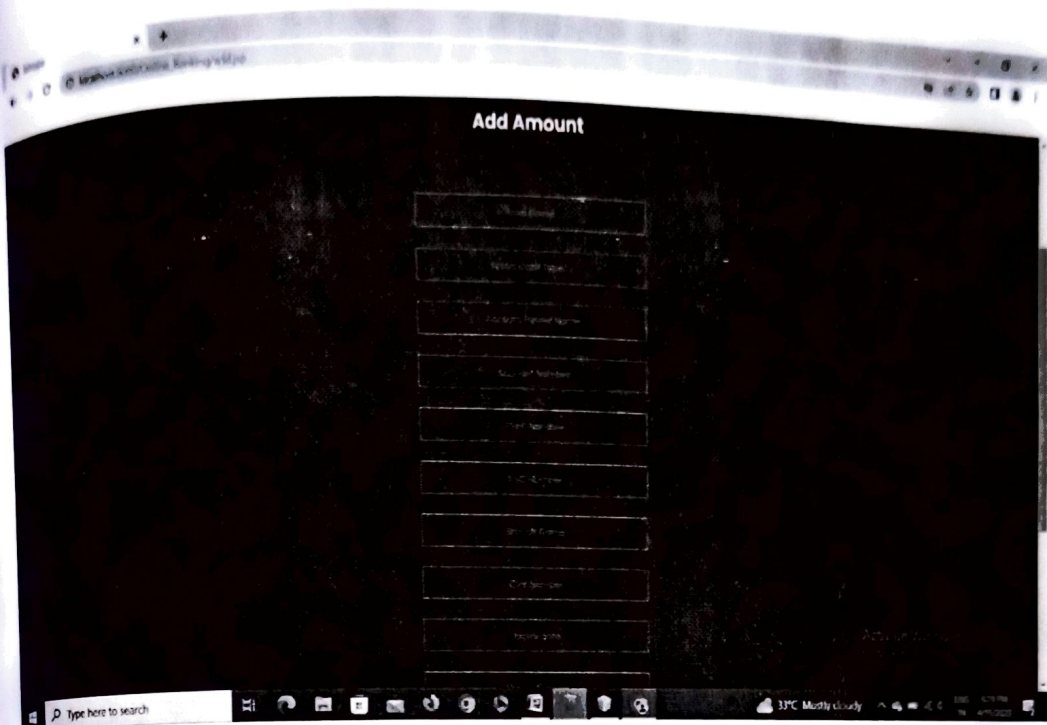


Fig8.7 Create New Account

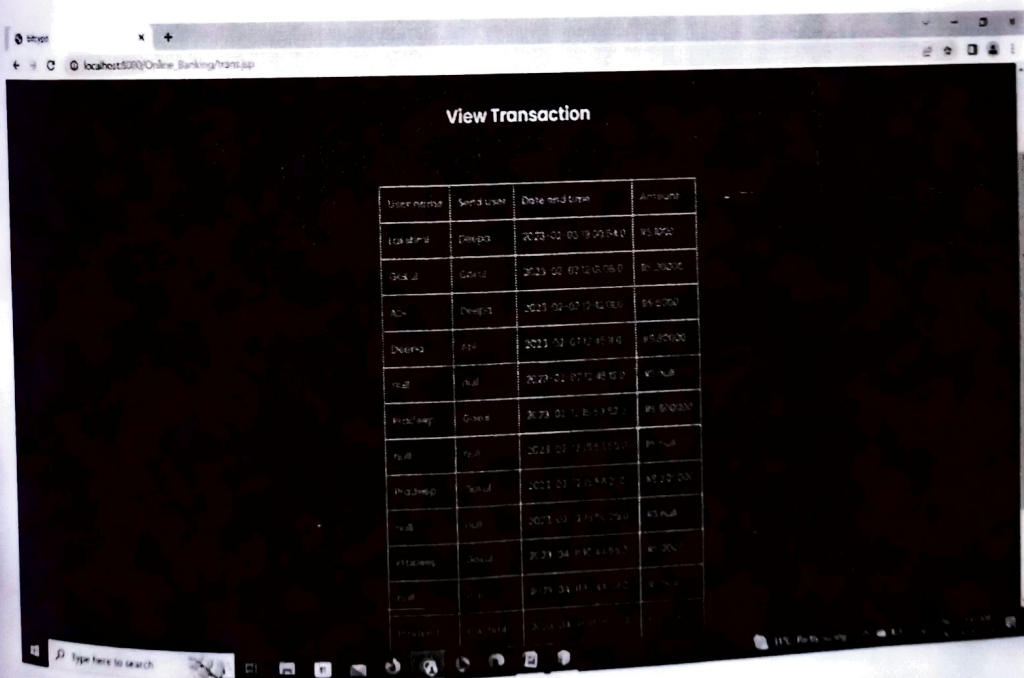


Fig8.8 Overall Transaction

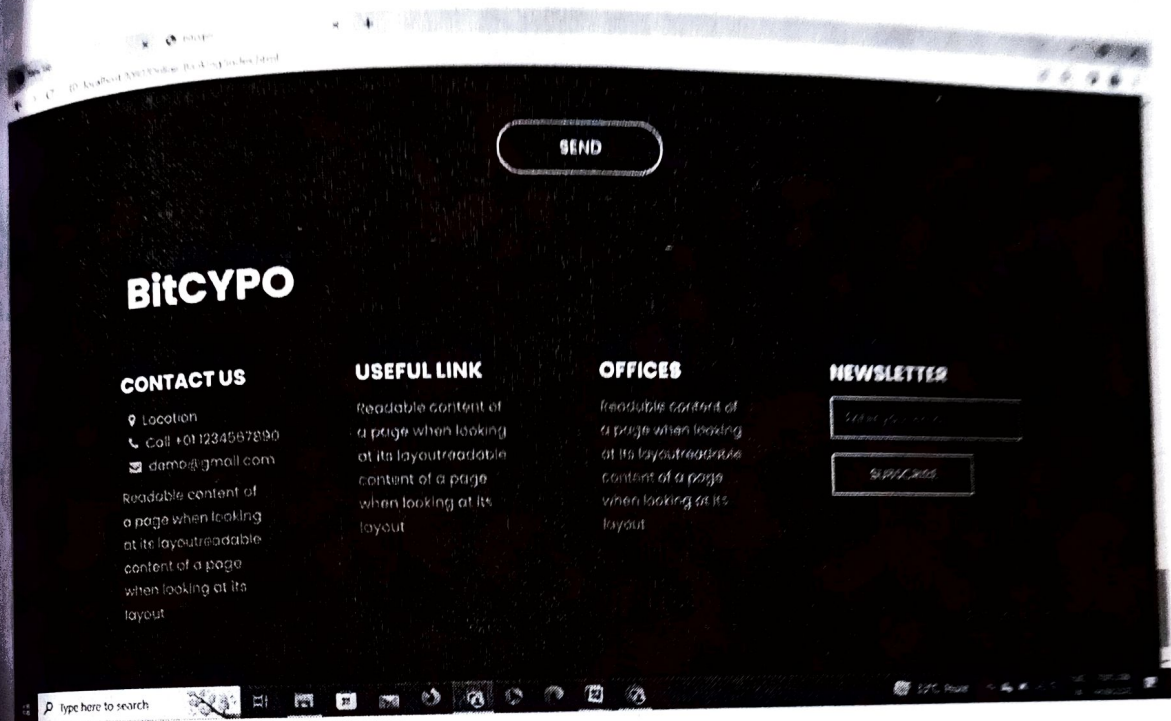


Fig8.9 Contact Us

CHAPTER 9

REFERENCE

Fundamentals of database systems by (Elmasri Navathe, 2000),
Website: <https://archive.org/stream/FundamentalsOfDatabaseSystemselmasrinavathe#page/n51/mode/2up>, Page: From 52 to more.

Article: Online banking, Website: https://en.wikipedia.org/wiki/Online_banking June 29, 2015.

Online Bank Account Management System Website: <http://www.slideshare.net> (Collect some info for report documents, 2014- 2015)

Learning MYSQL, JavaScript, jQuery, PHP, HTML, CSS3, Website: <http://www.w3schools.comm>, 2014-2015.

PHP and MySQL video tutorials, Oct 2014-2015 Website: <http://www.freebanglatutorial.com>, <http://www.youtube.com>

Veneeva, V. (2006), "E-Banking (Online Banking) and Its Role in Today's Society", Ezine articles, June 30, 2015.

JavaScript validation for empty input field, (May 10, 2015)
Website: <http://stackoverflow.com/questions/3937513/javascript-validation-for-empty-input-field>.

JavaScript form validation: Validate Password, Validate Email, Validate Phone Number, http://webcheatsheet.com/javascript/form_validation.php, (May 10, 2015)