PROJECT ON

“Bank Management System”

Certificate from the Supervisor

This is to certify that the dissertation is the record of Final Year Project, entitled "**Bank Management System**" undergone **Sammilani Mahavidyalaya** carried out by **Suman Debnath** bearing **Roll No. 183513-21-0026** of the Department of Computer Science under my supervision and guidance. To the best of my knowledge, the results embodied in this report, are original in nature and worthy of incorporation in the present version of the report for B.Sc. programme in Computer Science.

This report has not been submitted to any other university or institution for the award of any degree.

Guide / Supervisor :

Signed: ....., Date: .....

Smt. Swagata Saha Sau

(Assistant Professor, Head of the Department)

Acknowledgement

I would like to express my deepest sense of gratitude to my teacher and supervisor **Smt. Swagata Saha Sau**, Head of Department Computer Science, Sammilani Mahavidyalaya, and, for her untiring guidance, constant supervision, enthusiastic encouragement, sagacious advice and an effective surveillance throughout the entire period of my project and preparation of the manuscript. I greatly say thank you. Wish to express my heart full thanks to all of my honorable teachers of the Department of Computer , Sammilani Mahavidyalaya.

I am thankful to **Smt. Swagata Saha Sau**, who is having a vast knowledge of System Analysis and Design & Programming concepts which are the building block of project.

I also want to express my appreciation to my classmates and friends who helped me in one way or another during the course of developing this project. They endured the long hours of my absence during the development of this project.

I am thankful to my father **Sri. Swapan Debnath**, who himself being an employee of Sate Bank of India, has helped me provide information on the inner working of a banking system.

I deeply express my respect to my parent and my teachers for their blessing and constant inspiration in every step of my education. I am very thankful to my all friends for their help and company during the project and for giving me the encouragement to carry out the work.

July, 2021

Suman Debnath

Dept. of Computer Science

Sammilani Mahavidyalaya

Table of Contents

Abstract

The Bank 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 banking system and cover the basic functionality of a Bank 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.

The Bank Management System undertaken as a project is based on relevant technologies. The main aim of this project is to develop software for Bank Management System. This project has been developed to carry out the processes easily and quickly, which is not possible with the manuals systems, which are overcome by this software. This project is developed using PHP, HTML, CSS, JavaScript language and MYSQL use for database connection. Creating and managing requirements is a challenge of IT, systems and product development projects or indeed for any activity where you have to manage a contractual relationship. Organization need to effectively define and manage requirements to ensure they are meeting needs of the customer, while proving compliance and staying on the schedule and within budget. The impact of a poorly expressed requirement can bring a business out of compliance or even cause injury or death. Requirements definition and management is an activity that can deliver a high, fast return on investment.

The project analyzes the system requirements and then comes up with the requirements specifications. The system is then designed in accordance with specifications to satisfy the requirements. The system design is then implemented with MYSQL, PHP, HTML, CSS, JavaScript. The system is designed as an interactive and content management system. The content management system deals with data entry, validation confirm and updating whiles the interactive system deals with system interaction with the administration and users.

Thus, above features of this project will save transaction time and therefore increase the efficiency of the system.

1. Introduction

The “Bank Management System” 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. E-banking facilitates banking transactions by customers round the clock globally.

The primary aim of this “Bank Management System” is to provide an improved design methodology, which envisages the future expansion, and modification, which is necessary for a core sector like banking. This necessitates the design to be expandable and modifiable and so a modular approach is used in developing the application software. Anybody who is an Account holder in this bank can become a member of Bank Management System. He has to fill a form with his personal details and Account Number.

Bank is the place where customers feel the sense of safety for their property. In the bank, customers deposit and withdraw their money. Transaction of money also is a part where customer takes shelter of the bank. Smooth and efficient management affects the satisfaction of the customers and staff members, indirectly. And of course, it encourages management committee in taking some needed decision for future enhancement of the bank.

Now a day’s, managing a bank is tedious job up to certain limit. So software that reduces the work is essential. Also today’s world is a genuine computer world and is getting faster and faster day-by-day. Thus, considering above necessities, the software for bank management has became necessary which would be useful in managing the bank more efficiently.

All transactions are carried out online by transferring from accounts in the same Bank or national bank. The software is meant to overcome the drawbacks of the manual system.

The software has been developed using the most powerful and secure backend MYSQL database and the most widely accepted web oriented as well as application oriented.

1.1.Domain Description

The main aim of designing and developing this Internet banking System PHP primarily based 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 email id and password. Users will have all options and features in that application like money transfer to others, and send cash or money to inter banking as well as other banking customers by simply adding them as payees.

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database.

Data within the most common types of databases in operation today is typically modeled in rows and columns in a series of tables to make processing and data querying efficient. The data can then be easily accessed, managed, modified, updated, controlled, and organized. Most databases use structured query language (SQL) for writing and querying data.

Web page is a document available on world wide web. Web Pages are stored on web server and can be viewed using a web browser. A web page can cotain huge information including text, graphics, audio, video and hyper links. These hyper links are the link to other web pages. Collection of linked web pages on a web server is known as website. There is unique Uniform Resource Locator (URL) is associated with each web page.

Static Web page

Static web pages are also known as flat or stationary web page. They are loaded on the client’s browser as exactly they are stored on the web server. Such web pages contain only static information. User can only read the information but can’t do any modification or interact with the

information. Static web pages are created using only HTML. Static web pages are only used when the information is no more required to be modified.

1.2.Motivation

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 hard-hitting 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.

During the past several decades personnel function has been transformed from a relatively obscure record keeping staff to central and top level management function. There are many factors that have influenced this transformation like technological advances, professionalism, and general recognition of human beings as most important resources.

A computer based management system is designed to handle all the primary information required to calculate monthly statements of customer account which include monthly statement of any month. Separate database is maintained to handle all the details required for the correct statement calculation and generation.

This project intends to introduce more user friendliness in the various activities such as record updation, maintenance, and searching. The searching of record has been made quite simple as all the details of the customer can be obtained by simply keying in the identification or account number of that customer. Similarly, record maintenance and updation can also be accomplished by using the account number with all the details being automatically generated. These details are also being promptly automatically updated in the master file thus keeping the record absolutely up-to-date.

The entire information has maintained in the database or Files and whoever wants to retrieve can't retrieve, only authorization user can retrieve the necessary information which can be easily be accessible from the file.

1.3.Scope of Work

Bank Management System 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. The existing part of this project is; it displays Transaction reports, Statistical Summary of Account type.

Some of the features available through online banking are to **View balances**: Firstly login your account with your account number and password. Then checking your balance doesn't require much work. You simply select Account balances and take a look at your balance and past transactions. If you have more than one account, you can also do transfers between accounts. **Transfer funds**: When you select Transfer Funds, you'll be asked where to transfer the money to and from, and the amount.

The main objective of our project is to develop a software program for managing the entire bank process related to Administration accounts customer accounts and to keep each every track about their property and their various transaction processes efficiently.

- Hereby, our main objective is the customer's satisfaction considering today's faster in the world.
- Client can do his operations comfortably without any risk or losing of his privacy.
- Our software will perform and fulfill all the tasks that any customer would desire.
- Client doesn't need to go to the bank to do small operation.
- It helps the customer to be satisfied and comfortable in his choices, this protection contains customer's account, money and his privacy.
- Help client transferring money to/or another bank or country.

This project includes the entire upgraded feature required for the computerization banking system. This system is very easy to use, so that any user can use without getting pre-knowledge about this.

Its very much user friendly and meet almost all daily working process requirements. This system is completely GUI based and can be use by mouse and as well as keyboard. This system is melded in such a way that has got all features to upgrade without making much change in existing components.

2. Review of Related Work

Existing System

In the existing system the transactions are done only manually but in proposed system we have to computerize all the banking transaction using the software Banking system.

Disadvantages of Existing System

- Lack of security of data.
- More man power.
- Time consuming.
- Consumes large volume of pare work.
- Needs manual calculations.
- No direct role for the higher officials.
- Damage of machines due to lack of attention
- To avoid all these limitations and make the working more accurately the system needs to be computerized.

Advantages of the Proposed System

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features :

- Security of data.
- Ensure data accuracy's.
- Proper control of the higher officials.
- Minimize manual data entry.
- Minimum time needed for the various processing.
- Greater efficiency.
- Better service.
- User friendliness and interactive.
- Minimum time required.

3. Methodology

3.1. Problem Statement

- Keeping track the all activities and their record on paper and error.
- It is also very efficient and a time consuming process of observing continious increase in number of client visiting the bank.
- Recording and maintaining all the client record highly unreliable, inefficient and error prone.
- The problem facing the current manual system is difficult to update and maintain, inconsistent data, insecurity, difficult to impose different various data files and difficult to data backup.
- It is against this backdrop this Bank Management System is being developed to addressed the problem.

3.2. Method Description

Step 1: Read Choice
Step 2: If Choice = Client Login then
Call clientLogin(ClientId)
Else If Choice = Employee Login then
Call employeeLogin(EmployeeId)
Else If Choice = Manger Login then
Call managerLogin(ManagerId)
Default : Home Page
Step 3: Exit

Method clientLogin()
Step1: Read Choice
Step 2: If Choice = View Details then
Call clientDetails(accountNo)
Else If Choice = View Transaction History then

Call transactionHistory(ClientAccountNo)

Else If Choice = Transfer Funds then

Call transferFunds(SenderAccountNo, ReceiverAccountNo, Amount)

Else If Choice = Reset Password then

Call resetPassword (ClientId)

Else If Choice = Logout then

Call logout(ClientId)

Default : Client Dashboard

Step 3: Exist

Method employeeLogin()

Step1: Read Choice

Step 2: If Choice = View Client Details then

Call clientDetails(ClientAccountNo)

Else If Choice = View Client transaction History then

Call transactionHistory(ClientAccountNo)

Else If Choice = Add Client then

Call addClient()

Else If Choice = Edit Client then

Call editClient(ClientAccountNo)

Else If Choice = Remove Client then

Call removeClient(ClientAccountNo)

Else If Choice = Debit Funds then

Call debitFunds(ClientAccountNo, Amount)

Else If Choice = Credit Funds then

Call creditFunds(ClientAccountNo, Amount)

Else If Choice = Transfer Funds then

Call transferFunds(SenderAccountNo, ReceiverAccountNo, Amount)

Else If Choice = Reset Password then

Call resetPassword (EmployeeId)

Else If Choice = Logout then

Call Logout(EmployeeId)

Default : Employee Dashboard

Step 3: Exist

Method managerLogin()

Step1: Read Choice

Step 2: If Choice = View Client Details then

Call clientDetails(ClientAccountNo)

Else If Choice = Add Client then

Call addClient()

Else If Choice = Edit Client then

Call editClient(ClientAccountNo)

Else If Choice = Remove Client then

Call removeClient(ClientAccountNo)

Else If Choice = View Employee Details then

Call employeeDetails(EmployeeId)

Else If Choice = Add Client then

Call addEmployee()

Else If Choice = Edit Client then

Call editEmployee(EmployeeId)

Else If Choice = Remove Client then

Call removeEmployee(EmployeeId)

Else If Choice = Reset Password then

Call resetPassword (ManagerId)

Else If Choice = Logout then

Call Logout(ManagerId)

Default : Manager Dashboard

Step 3: Exist

Method clientDetails()

Step 1: If ClientAccountNo = clientAccounts(accountNo) then

Return(Details from Database)

Else

Return(No record found.)

Step 2: Exit

Method transactionHistory()

Step 1: If ClientId = transctionHistory(id) then

Return(Transaction History from Database)

Else

Return(No record found.)

Step 2: Exit

Method transferFunds()

Step 1: If ReceiverAccountNo = clientAccounts(accountNo) then

ReceiverAccountNo.balance = balance + Amount

SenderAccountNo.balance = balance - Amount

Else

Return(No record found.)

Step 2: Exit

Method resetPassword()

Step 1: Read CurrentPassword

Step 2: Read NewPassword

Step 3: If ClientId = clientAccounts(id) then

If password = CurrentPassword then

password = NewPassword

Step 4: Exit

Method logout()

Step 1: Set session(id) = NULL

Step 2: Exit

Method addClient()

Step 1: Read information from user

Step 2: Add to database table clientAccounts()

Step 3: Exit

Method editClient()

Step 1: If ClientAccountNo = clientAccounts(accountNo) then

Read information from user

Store them in database table clientAccounts(accountNo)

Else

Return(No record found.)

Step 2: Exit

Method removeClient()

Step 1: If ClientAccountNo = clientAccounts(accountNo) then

Delete from database table clientAccounts(accountNo)

Else

Return(No record found.)

Step 2: Exit

Method debitFunds ()

Step 1: If ClientAccountNo = clientAccounts(accountNo) then

balance = balance - amount

Else

Return(No record found.)

Step 2: Exit

Method creditFunds ()

Step 1: If ClientAccountNo = clientAccounts(accountNo) then

balance = balance + amount

Else

Return(No record found.)

Step 2: Exit

Method employeeDetails()

Step 1: If EmployeeId = employeeAccounts(empId) then

Return(Details from Database)

Else

Return(No record found.)

Step 2: Exit

Method addEmployee()

Step 1: Read information from user

Step 2: Add to database table employeeAccounts()

Step 3: Exit

Method editEmployee()

Step 1: If EmployeeId = employeeAccounts(empId) then

Read information from user

Store them in database table employeeAccounts(empId)

Else

Return(No record found.)

Step 2: Exit

Method removeEmployee()

Step 1: If EmployeeId = employeeAccounts(empId) then

Delete from database table employeeAccounts(empId)

Else

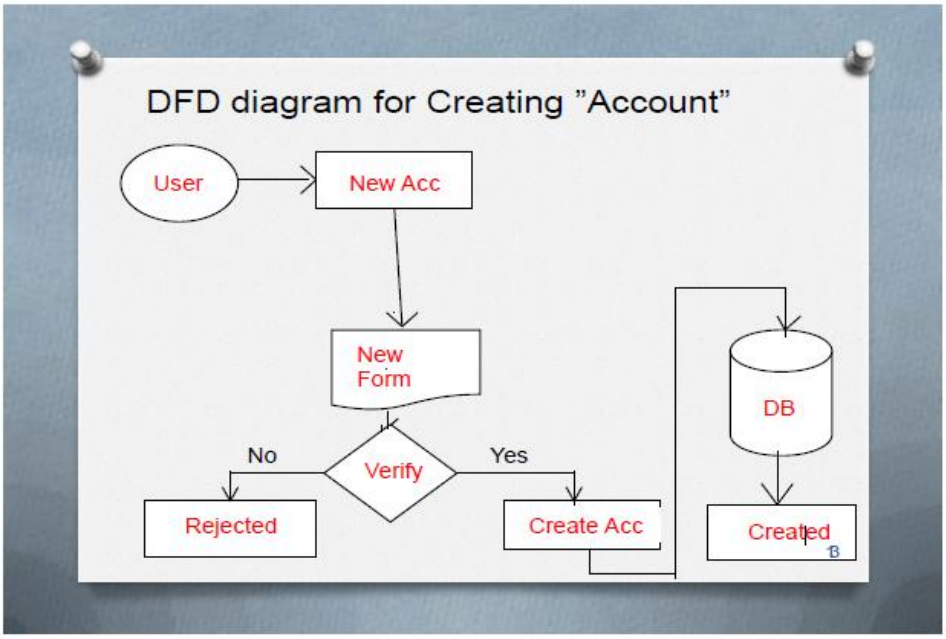
Return(No record found.)

Step 2: Exit

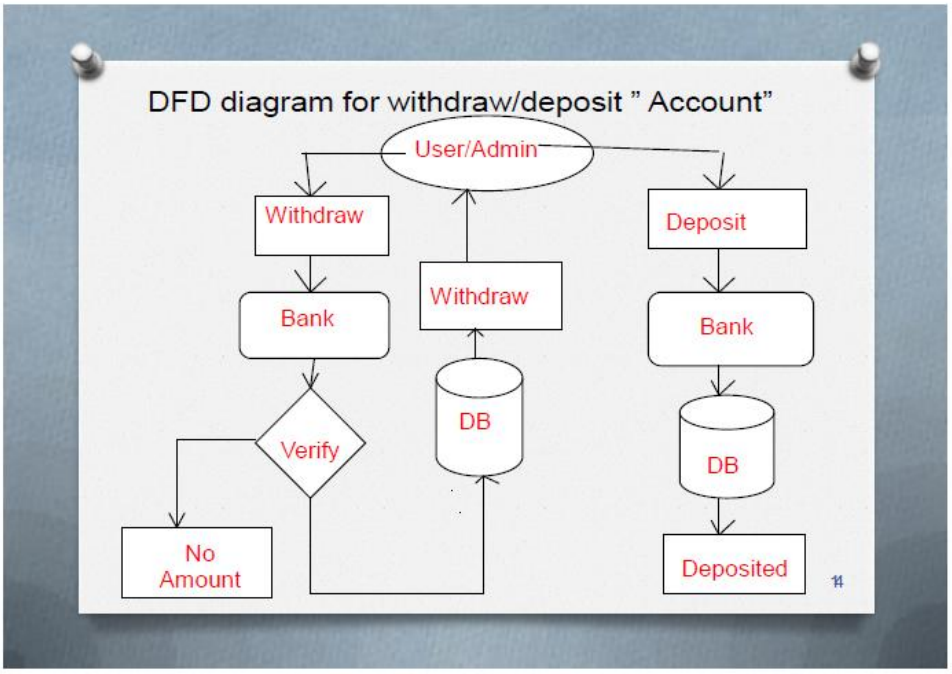
3.3. Design Description

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel.

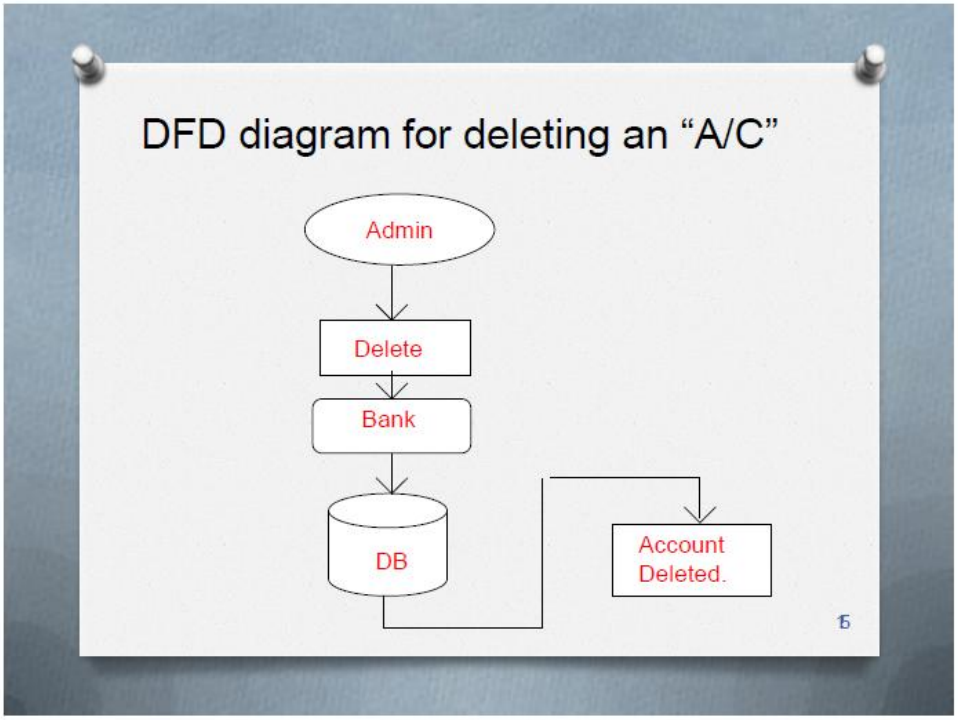
Data flow diagram



Create new account DFD



Withdraw/deposit account DFD



Deleting an account DFD

4. Implementation

Connection To Database :

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "bank_management";

$con = new mysqli($servername, $username, $password, $dbname);

if($con -> connect_error)
{
        echo "Failed to connect!";
        exit();
}
else
        // echo "Connection is built successfully.";
?>
```

Add / edit client OR employee :

```
<?php

   // very important to get the session variables
   session_start();

   // if employee OR manager is logged off then redirect to login page
   if (!isset($_SESSION['employeeId']) && !isset($_SESSION['managerId']))
      header('location:../login.php');

   // connection to database
   include "../connect.php";
```

```
// ------------------------------ start of add form -------------------------------//

// form request to add a client to db by employee OR manager
if (isset($_POST['add_client']))
{
    // convert the POST variables to uppercase
    $_POST['first_name'] = strtoupper($_POST['first_name']);
    $_POST['last_name'] = strtoupper($_POST['last_name']);
    $_POST['gender'] = strtoupper($_POST['gender']);
    $_POST['source'] = strtoupper($_POST['source']);
    $_POST['account_type'] = strtoupper($_POST['account_type']);
    $_POST['district'] = strtoupper($_POST['district']);
    $_POST['city'] = strtoupper($_POST['city']);
    $_POST['address'] = strtoupper($_POST['address']);

    // check if the email id. is already registered or not
    $ar = $con->query("SELECT * FROM `clientAccounts` WHERE
`clientAccounts`.`email` = '$_POST[email]'");

    if ($ar->num_rows > 0)
    {
        if ($_GET['user'] == 'employee')
            echo "<script>alert('Email Id. Already Present.\\nFailed to Add
Client.');window.location.href='../employee/employee-dashboard.php'</script>";
        else
            echo "<script>alert('Email Id. Already Present.\\nFailed to Add
Client.');window.location.href='../manager/manager-dashboard.php'</script>";
    }
    else
    {
        // check if password and confirm password fields match
        if ($_POST['password'] == $_POST['cpassword'])
        {
            // convert password to hash
            $hash = password_hash($_POST['password'], PASSWORD_DEFAULT);

            if ($con->query("INSERT INTO `clientAccounts` (`first_name`, `last_name`,
`gender`, `dob`, `balance`, `aadhaar_no`, `email`, `password`, `contact_no`, `source`,
`account_no`, `account_type`, `state`, `district`, `city`, `address`, `pincode`, `branch_id`,
`date`) values
('$_POST[first_name]','$_POST[last_name]','$_POST[gender]','$_POST[dob]','$_POST[
```

```
deposit]','$_POST[aadhaar_no]','$_POST[email]','$hash','$_POST[contact_no]','$_POST[
source]','$_POST[account_no]','$_POST[account_type]','$_POST[state]','$_POST[district]','$_POST[city]','$_POST[address]','$_POST[pincode]','$_POST[branch_id]',
CURRENT_TIMESTAMP)"))
            {
               // check if user is employee OR manager and redirect to dashbord
accordingly
               if ($_GET['user'] == 'employee')
                  echo "<script>alert('Client added
Successfully!');window.location.href='../employee/employee-dashboard.php'</script>";
               else
                  echo "<script>alert('Client added
Successfully!');window.location.href='../manager/manager-dashboard.php'</script>";
            }
            else
            {
               // check if user is employee OR manager and redirect to dashbord
accordingly
               if ($_GET['user'] == 'employee')
                  echo "<script>alert('Failed to Add
Client.');window.location.href='../employee/employee-dashboard.php'</script>";
               else
                  echo "<script>alert('Failed to Add
Client.');window.location.href='../manager/manager-dashboard.php'</script>";
            }
         }
         else
         {
            // check if user is employee OR manager and redirect to dashbord accordingly
            if ($_GET['user'] == 'employee')
               echo "<script>alert('Confirmed Password Did Not Match.\\nFailed to Add
Client.');window.location.href='../employee/employee-dashboard.php'</script>";
            else
               echo "<script>alert('Confirmed Password Did Not Match.\\nFailed to Add
Client.');window.location.href='../manager/manager-dashboard.php'</script>";
         }
      }
   }

   // form request to add an employee to db by manager
   if (isset($_POST['add_employee']))
   {
      // convert the POST variables to uppercase
```

```
        $_POST['first_name'] = strtoupper($_POST['first_name']);
        $_POST['last_name'] = strtoupper($_POST['last_name']);
        $_POST['district'] = strtoupper($_POST['district']);
        $_POST['city'] = strtoupper($_POST['city']);
        $_POST['address'] = strtoupper($_POST['address']);

        // check if the email id. is already registered or not
        $ar = $con->query("SELECT * FROM `employeeAccounts` WHERE
`employeeAccounts`.`email` = '$_POST[email]'");

        if ($ar->num_rows > 0)
        {
            echo "<script>alert('Email Id. Already Present.\\nFailed to Add
Client.');window.location.href='../manager/manager-dashboard.php'</script>";
        }
        else
        {
            // check if password and confirm password fields match
            if ($_POST['password'] == $_POST['cpassword'])
            {
                // convert password to hash
                $hash = password_hash($_POST['password'], PASSWORD_DEFAULT);

                if ($con->query("INSERT INTO `employeeAccounts` (`first_name`,
`last_name`, `dob`, `aadhaar_no`, `email`, `password`, `contact_no`, `emp_id`, `state`,
`district`, `city`, `address`, `date`) VALUES
('$_POST[first_name]','$_POST[last_name]','$_POST[dob]','$_POST[aadhaar_no]','$_P
OST[email]','$hash','$_POST[contact_no]','$_POST[emp_id]','$_POST[state]','$_POST[d
istrict]','$_POST[city]','$_POST[address]', CURRENT_TIMESTAMP)"))
                {
                    echo "<script>alert('Employee added
Successfully!');window.location.href='../manager/manager-dashboard.php'</script>";
                }
                else
                    echo "<script>alert('Failed to Add
Employee.');window.location.href='../manager/manager-dashboard.php'</script>";
            }
            else
                echo "<script>alert('Confirmed Password Did Not Match.\\nFailed to Add
Employee.');window.location.href='../manager/manager-dashboard.php'</script>";
        }
    }
```

```
// ------------------------------ start of edit form -------------------------------//

// form request to edit a client in db by employee OR manager
if (isset($_POST['edit_client']))
{
   // convert the POST variables to uppercase
   $_POST['source'] = strtoupper($_POST['source']);
   $_POST['account_type'] = strtoupper($_POST['account_type']);
   $_POST['district'] = strtoupper($_POST['district']);
   $_POST['city'] = strtoupper($_POST['city']);
   $_POST['address'] = strtoupper($_POST['address']);

   // check if the email id. is already registered or not except his current email id.
   $ar = $con->query("SELECT * FROM `clientAccounts` WHERE
`clientAccounts`.`email` = '$_POST[email]' AND `clientAccounts`.`id` <>
'$_GET[editId]'");

   if ($ar->num_rows > 0)
   {
      if ($_GET['user'] == 'employee')
         echo "<script>alert('Email Id. Already Present.\\nFailed to Edit Client
Details.');window.location.href='../employee/client-details-edit.php'</script>";
      else
         echo "<script>alert('Email Id. Already Present.\\nFailed to Edit Client
Details.');window.location.href='../manager/client-details-edit.php'</script>";
   }
   else
   {
      if ($con->query("UPDATE `clientAccounts` SET `email` = '$_POST[email]',
`contact_no` = '$_POST[contact_no]', `source` = '$_POST[source]', `account_type` =
'$_POST[account_type]', `state` = '$_POST[state]', `district` = '$_POST[district]', `city` =
'$_POST[city]', `address` = '$_POST[address]', `pincode` = '$_POST[pincode]',
`branch_id` = '$_POST[branch_id]' WHERE `id` = '$_GET[editId]'"))
         {
            // check if user is employee OR manager and redirect to dashbord
accordingly
            if ($_GET['user'] == 'employee')
               echo "<script>alert('Client Details Edited
Successfully!');window.location.href='../employee/client-details-edit.php'</script>";
            else
```

```
                echo "<script>alert('Client Details Edited
Successfully!');window.location.href='../manager/client-details-edit.php'</script>";
            }
            else
            {
                // check if user is employee OR manager and redirect to dashbord
accordingly
                if ($_GET['user'] == 'employee')
                    echo "<script>alert('Failed to Edit Client Details.')</script>";
                else
                    echo "<script>alert('Failed to Edit Client
Details.');window.location.href='../manager/client-details-edit.php'</script>";
            }
        }
    }


    // form request to edit a client in db by manager
    if (isset($_POST['edit_employee']))
    {
        // convert the POST variables to uppercase
        $_POST['district'] = strtoupper($_POST['district']);
        $_POST['city'] = strtoupper($_POST['city']);
        $_POST['address'] = strtoupper($_POST['address']);

        // check if the email id. is already registered or not except his current email id.
        $ar = $con->query("SELECT * FROM `employeeAccounts` WHERE
`employeeAccounts`.`email` = '$_POST[email]' AND `employeeAccounts`.`id` <>
'$_SESSION[editId]'");

        if ($ar->num_rows > 0)
        {
            echo "<script>alert('Email Id. Already Present.\\nFailed to Edit Employee
Details.');window.location.href='../manager/employee-details-edit.php'</script>";
        }
        else
        {
            if ($con->query("UPDATE `employeeAccounts` SET `email` = '$_POST[email]',
`contact_no` = '$_POST[contact_no]', `state` = '$_POST[state]', `district` =
'$_POST[district]', `city` = '$_POST[city]', `address` = '$_POST[address]' WHERE `id` =
'$_SESSION[editId]'"))
                {
```

```
                echo "<script>alert('Employee Details Edited
Successfully!');window.location.href='../manager/employee-details-edit.php'</script>";
            }
            else
            {
                echo "<script>alert('Failed to Edit Employee
Details.');window.location.href='../manager/employee-details-edit.php'</script>";
            }
        }
    }
?>
```

Transfer Funds :

```
<?php

    // very important to get the session variables
        session_start();

    // if client OR employee is logged off then redirect to login page
        if (!isset($_SESSION['employeeId']) && !isset($_SESSION['clientId']))
            header('location:../login.php');

    // connection to database
    include "../connect.php";



    // ------------------------------ start of employee fund transfer ---------------------------------
//

    if (isset($_GET['employee']))
    {
        if (isset($_POST['transfer-fund']))
```

```
    {
        $from_acc = $con->query("SELECT * FROM `clientaccounts` WHERE
account_no = '$_POST[from_account]' LIMIT 1");
        $userData_from = $from_acc->fetch_assoc();

        // check if the from client exists or not
        if ($from_acc->num_rows >0)
        {
            if ($userData_from['balance'] >= $_POST['amount'])
            {
                $to_acc = $con->query("SELECT * FROM `clientaccounts` WHERE
account_no = '$_POST[to_account]' LIMIT 1");
                $userData_to = $to_acc->fetch_assoc();

                // check if the to client exists or not
                if ($to_acc->num_rows >0)
                {
                    // check if transfering money to same account or not
                    if ($userData_from['account_no'] == $userData_to['account_no'])
                    {
                        echo "<script>alert('Cannot Transfer To Self.\\nFund Transfer
Failed.');window.location.href='../employee/employee-dashboard.php'</script>";
                    }
                    else
                    {
                        $added_amount = $userData_to['balance'] + $_POST['amount'];
                        $result_added = $con->query("UPDATE `clientaccounts` SET balance
= $added_amount WHERE account_no = '$_POST[to_account]' LIMIT 1");

                        $subtracted_amount = $userData_from['balance'] - $_POST['amount'];
```

```
                $result_subtracted = $con->query("UPDATE `clientaccounts` SET
balance = $subtracted_amount WHERE account_no = '$_POST[from_account]' LIMIT
1");

                // check if client balance updated or not
                if ($con->affected_rows > 0)
                {
                   $tr = $con->query("INSERT INTO `transclient`
(`action`,`from_account`,`to_account`,`trans_amount`) VALUES
('transfer','$_POST[from_account]','$_POST[to_account]','$_POST[amount]')");

                   // check if transaction details recorded or not
                   if ($con->affected_rows > 0)
                   {
                      echo "<script>alert('Fund Transfered
Successfully!');window.location.href='../employee/employee-dashboard.php'</script>";
                   }
                   else
                      echo "<script>alert('Fund Transfered but Transactions Not
Updated.');window.location.href='../employee/employee-dashboard.php'</script>";
                }
                else
                   echo "<script>alert('Fund Not
Debited.');window.location.href='../employee/employee-dashboard.php'</script>";
             }
          }
          else
             echo "<script>alert('Client Not Found To whom Funds are to be
Transfered.');window.location.href='../employee/employee-dashboard.php'</script>";
       }
       else
```

```
                echo "<script>alert('Insufficient
Balance.');window.location.href='../employee/employee-dashboard.php'</script>";
            }
            else
                echo "<script>alert('Client Not Found From whom Funds are to be
Transfered.');window.location.href='../employee/employee-dashboard.php'</script>";
        }
    }



    // ------------------------------ start of client fund transfer ---------------------------------//

    if (isset($_GET['client']))
    {
        if (isset($_POST['transfer-fund']) || isset($_POST['transfer-fund-nav']))
        {
            $_POST['ifsc'] = strtoupper($_POST['ifsc']);

            $IFSC_bank_name = substr($_POST['ifsc'], 0, 4);
            $IFSC_control_no = substr($_POST['ifsc'], 4, 1);
            $IFSC_branch_code = substr($_POST['ifsc'], 5);

            if (ctype_alpha($IFSC_bank_name) && $IFSC_control_no == '0' &&
ctype_alnum($IFSC_branch_code))
            {
                if ($IFSC_bank_name == 'RTHB')
                {
                    $branch = $con->query("SELECT * FROM `branch` WHERE
`branch_code` = '$IFSC_branch_code'");
```

```
            if ($branch->num_rows >0)
            {
                $from_acc = $con->query("SELECT * FROM `clientaccounts` WHERE
`id` = '$_SESSION[clientId]'");
                $userData_from = $from_acc->fetch_assoc();

                $to_acc = $con->query("SELECT * FROM `clientaccounts` WHERE
`account_no` = '$_POST[to_account]' LIMIT 1");
                $userData_to = $to_acc->fetch_assoc();

                // check if the 'to' account exists or not
                if ($to_acc->num_rows >0)
                {
                    // check if transfering money to same account or not
                    if ($userData_from['account_no'] == $_POST['to_account'])
                    {
                        if (isset($_POST['transfer-fund']))
                            echo "<script>alert('Cannot Transfer To Self.\\nFund Transfer
Failed.');window.location.href='../client/client-dashboard.php'</script>";
                        else
                            echo "<script>alert('Cannot Transfer To Self.\\nFund Transfer
Failed.');window.location.href='../client/client-transfer.php'</script>";
                    }
                    // if transfering money to different account
                    else
                    {
                        // check if the user has insufficient balance for transfer
                        if ($userData_from['balance'] >= $_POST['amount'])
                        {
                            // add fund to the receivier account
                            $added_amount = $userData_to['balance'] + $_POST['amount'];
```

```
                    $result_added = $con->query("UPDATE `clientaccounts` SET
balance = $added_amount WHERE account_no = '$_POST[to_account]' LIMIT 1");

                    // subtract fund from the sender account
                    $subtracted_amount = $userData_from['balance'] -
$_POST['amount'];
                    $result_subtracted = $con->query("UPDATE `clientaccounts` SET
balance = $subtracted_amount WHERE id = '$_SESSION[clientId]'");

                    // check if client balance updated or not
                    if ($con->affected_rows > 0)
                    {
                       $tr = $con->query("INSERT INTO `transclient`
(`action`,`from_account`,`to_account`,`trans_amount`) VALUES
('transfer','$userData_from[account_no]','$_POST[to_account]','$_POST[amount]')");

                       // check if transaction details recorded or not
                       if ($con->affected_rows > 0)
                       {
                          if (isset($_POST['transfer-fund']))
                             echo "<script>alert('Fund Transfered
Successfully!');window.location.href='../client/client-dashboard.php'</script>";
                          else
                             echo "<script>alert('Fund Transfered
Successfully!');window.location.href='../client/client-transfer.php'</script>";
                       }
                       // if transaction details not recorded
                       else
                       {
                          if (isset($_POST['transfer-fund']))
```

```
                                    echo "<script>alert('Fund Transfered but Transactions Not
Updated.');window.location.href='../client/client-dashboard.php'</script>";
                                else
                                    echo "<script>alert('Fund Transfered but Transactions Not
Updated.');window.location.href='../client/client-transfer.php'</script>";
                            }
                        }
                        // if client balance not updated
                        else
                        {
                            if (isset($_POST['transfer-fund']))
                                echo "<script>alert('Fund Transfered
Failed.');window.location.href='../client/client-dashboard.php'</script>";
                            else
                                echo "<script>alert('Fund Transfered
Failed.');window.location.href='../client/client-transfer.php'</script>";
                            }
                    }
                    // if the user has insufficient balance
                    else
                    {
                        if (isset($_POST['transfer-fund']))
                            echo "<script>alert('Insufficient
Balance.');window.location.href='../client/client-dashboard.php'</script>";
                        else
                            echo "<script>alert('Insufficient
Balance.');window.location.href='../client/client-transfer.php'</script>";
                    }
                }
            }
            // if the 'to' account not found
```

```
                else
                {
                    if (isset($_POST['transfer-fund']))
                        echo "<script>alert('Account Not Found To whom Funds are to be Transfered.');window.location.href='../client/client-dashboard.php'</script>";
                    else
                        echo "<script>alert('Account Not Found To whom Funds are to be Transfered.');window.location.href='../client/client-transfer.php'</script>";
                }
            }
            else
            {
                if (isset($_POST['transfer-fund']))
                    echo "<script>alert('No Branch.');window.location.href='../client/client-dashboard.php'</script>";
                else
                    echo "<script>alert('No Branch.');window.location.href='../client/client-transfer.php'</script>";
                }
            }
            else
            {
                if (isset($_POST['transfer-fund']))
                    echo "<script>alert('Different Bank.');window.location.href='../client/client-dashboard.php'</script>";
                else
                    echo "<script>alert('Different Bank.');window.location.href='../client/client-transfer.php'</script>";
                }
            }
            else
```

```
        {
            if (isset($_POST['transfer-fund']))
                echo "<script>alert('Please Enter a Proper IFSC
Code.');window.location.href='../client/client-dashboard.php'</script>";
            else
                echo "<script>alert('Please Enter a Proper IFSC
Code.');window.location.href='../client/client-transfer.php'</script>";
        }
    }
  }
?>
```

Debit Funds :

```
<?php

  // very important to get the session variables
        session_start();

  // if employee is logged off then redirect to login page
        if (!isset($_SESSION['employeeId']))
            header('location:login.php');

  // connection to database
  include "../connect.php";



  // ------------------------------ start of debit form ----------------------------------//

  if (isset($_POST['debit-fund']))
  {
     $ar = $con->query("SELECT * FROM `clientaccounts` WHERE account_no =
'$_POST[account_no]' LIMIT 1");
     $userData = $ar->fetch_assoc();

     // check if client exists or not
     if ($ar->num_rows >0)
```

```
    {
        if ($userData['balance'] >= $_POST['amount'])
        {
            $amount = $userData['balance'] - $_POST['amount'];
            $result = $con->query("UPDATE `clientaccounts` SET balance = $amount
WHERE account_no = '$_POST[account_no]' LIMIT 1");

            // check if client balance updated or not
            if ($con->affected_rows > 0)
            {
                $tr = $con->query("INSERT INTO `transclient`
(`action`,`from_account`,`trans_amount`) VALUES
('withdraw','$_POST[account_no]','$_POST[amount]')");

                // check if transaction details recorded or not
                if ($con->affected_rows > 0)
                {
                    echo "<script>alert('Fund Debited
Successfully!');window.location.href='../employee/employee-dashboard.php'</script>";
                }
                else
                    echo "<script>alert('Fund Debited but Transactions Not
Updated.');window.location.href='../employee/employee-dashboard.php'</script>";
            }
            else
                echo "<script>alert('Fund Not
Debited.');window.location.href='../employee/employee-dashboard.php'</script>";
        }
        else
            echo "<script>alert('Insufficient
Balance.');window.location.href='../employee/employee-dashboard.php'</script>";
    }
    else
        echo "<script>alert('Client Not
Found.');window.location.href='../employee/employee-dashboard.php'</script>";
  }
?>
```

Credit Funds :

```
<?php

   // very important to get the session variables
         session_start();

   // if employee is logged off then redirect to login page
         if (!isset($_SESSION['employeeId']))
            header('location:login.php');

   // connection to database
   include "../connect.php";



   // ------------------------------ start of credit form ----------------------------------//

   if (isset($_POST['credit-fund']))
   {
      $ar = $con->query("SELECT * FROM `clientaccounts` WHERE account_no =
'$_POST[account_no]' LIMIT 1");
      $userData = $ar->fetch_assoc();

      // check if client exists or not
      if ($ar->num_rows >0)
      {
         $amount = $userData['balance'] + $_POST['amount'];
         $result = $con->query("UPDATE `clientaccounts` SET balance = $amount
WHERE account_no = '$_POST[account_no]' LIMIT 1");

         // check if client balance updated or not
         if ($con->affected_rows > 0)
         {
            $tr = $con->query("INSERT INTO `transclient`
(`action`,`to_account`,`trans_amount`) VALUES
('deposit','$_POST[account_no]','$_POST[amount]')");

            // check if transaction details recorded or not
            if ($con->affected_rows > 0)
            {
               echo "<script>alert('Fund Credited
Successfully!');window.location.href='../employee/employee-dashboard.php'</script>";
            }
```

```
            else
                echo "<script>alert('Fund Credited but Transactions Not
Updated.');window.location.href='../employee/employee-dashboard.php'</script>";
        }
        else
            echo "<script>alert('Fund Not
Credited.');window.location.href='../employee/employee-dashboard.php'</script>";
      }
      else
        echo "<script>alert('Client Not
Found.');window.location.href='../employee/employee-dashboard.php'</script>";
   }
?>
```

5. Results and Discussion

General Information:

1. You should register for RITCH bank with the branch where you maintain the account.
2. If you maintain accounts at more than one branch, you need to register at each branch separately.
3. Normally RITCH Bank services will be open to the customer only after he/she acknowledges the receipt of password.
4. We invite you to visit your account on the site frequently for transacting business or viewing account balances. If you believe that any information relating to your account has a discrepancy, please bring it to the notice of the branch by e-mail or letter.
5. All accounts at the branch whether or not listed in the registration form, will be available on the RITCH Bank. However the applicant has the option to selectively view the accounts on the RITCH Bank.

Security terms:

1. The Branch where the customer maintains his/her account will assign:
    a) User Account Number &
    b) Password
2. The Password given by the branch must be replaced by Password of customer's choice at the time of first log-on. This is mandatory.
3. Bank will make reasonable use of available technology to ensure security and to prevent unauthorized access to any of these services.
4. You are welcome to access RITCH Bank from anywhere anytime. However, as a matter of precaution, customers may avoid using PCs with public access.
5. There is no way to retrieve a password from the system. Therefore if a customer forgets his/her password, he/she must approach the branch for re-registration.

6. 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 a bank. Future version of this project will still be much enhanced than the current version. Writing and depositing checks are perhaps the most fundamental ways to move money in and out of a checking account, but advancements in technology have added ATM and debit card transactions. All banks have rules about how long it takes to access your deposits, how many debit card transactions you're allowed in a day, and how much cash you can withdraw from an ATM. Access to the balance in your checking account can also be limited by businesses that place holds on your funds. Banks are providing internet banking services also so that the customers can be attracted. Online banking is an innovative tool that is fast becoming a necessity. It is a successful strategic weapon for banks to remain profitable in a volatile and competitive marketplace of today. 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.

Thus the Bank Management System it is developed and executed successfully.

7. References

1. Fundamentals of database systems by (**Elmasri Navathe, 2000**),
   Website:https://archive.org/stream/FundamentalsOfDatabaseSystemselmasrinavathe#page/n51/mode/2up, Page: From **52** to more.
2. Article: **Online banking**,
   Website: https://en.wikipedia.org/wiki/Online_banking
3. Online Bank Account Management System
   Website: http://www.slideshare.net
4. Learning **MYSQL**, JavaScript, **PHP**, **HTML**, CSS3,
   Website: http://www.w3schools.com,
   https://www.youtube.com/watch?v=hKB-YGF14SY&list=PLp6c3tqokHOvaOA13vwUK8OQZmV_trJeJ&index=20&t=10775s,
   https://www.youtube.com/watch?v=u5-K_ua9sOw&list=PLp6c3tqokHOvaOA13vwUK8OQZmV_trJeJ&index=18&t=4700s,
   https://www.youtube.com/watch?v=BsDoLVMnmZs&list=PLp6c3tqokHOvaOA13vwUK8OQZmV_trJeJ&index=17,
   https://www.youtube.com/watch?v=1SnPKhCdlsU&list=PLp6c3tqokHOvaOA13vwUK8OQZmV_trJeJ&index=16&t=6410s
5. Veneeva, V. (2006), “E-Banking (Online Banking) and Its Role in Today's Society”, Ezine articles