



OXFORD
COLLEGE OF ENGINEERING & MANAGEMENT



An ISO 9001:2015 Certified College

(AFFILIATED TO POKHARA UNIVERSITY)

PROJECT - II

Bank Management System by

--	--

**Under the Guidance
of
Sunil Kumar**

**Submitted to the Faculty of Science and Technology,
OCEM**

in partial fulfilment of the requirements

for the Fourth Semester

Bachelor of Computer Application (BCA)

2017

**Oxford College of Engineering and Management
Nepal**

ACKNOWLEDGEMENTS

It is our proud privilege to release of our feelings of our gratitude to several persons who helped us directly or indirectly to conduct this research project work in “Bank Management System”. We are most thankful to our college “**Oxford College of Engineering and Management**” and our respected principle **Er. Hari Bhandari** as well as our lecturer **Mr. Sunil Kumar** and our BCA HOD **Mr. Suresh Baral** for their coordination and cooperation and their kind guidance and encouragement to make this project more efficient and effective.

Many people, especially our classmates and team members itself, have made valuable comment suggestions on this proposal which gave us the inspiration to improve our assignment. We thank all the people for their help directly and indirectly to complete our assignment.

In spite of the great effort of all team members to make the project error-free and effective, there may occur errors in this project and it might not meet all the requirements due to our insufficient knowledge in this field. Any suggestion and comments from any individual or organization will be warmly considered.

ROLES AND RESPONSIBILITIES FORM

Name of the project: Bank Management System

Date of Completion: 30 November 2018

Name of the team Member	*ROLE	Tasks and Responsibilities
1. Prabhaw Soti	Overall Project Management With Coding and Designing	- Coding - Testing - Designing
2. Sandeep Bashyal	Coding and Designing	- Coding - Designing
3. Sagar Shrestha	Implementation, Testing and documentation	- Designing - Documentation

Name and Signature of the Project Team Member/s: Signature

Name: Prabhaw Soti

Name: Sandeep Bashyal

Name: Sagar Shrestha

Signature of the Project Guide: 30 November 2018

Table of Contents

1. INTRODUCTION.....	1
1.1 Background:	1
1.2 Objectives:	1
1.3 Time Plan:	2
1.4 Investigation of Problem:.....	4
2. REQUIREMENTS OF THE ANALYSIS OF THE SYSTEM:.....	5
2.1 Feasibility Study:	5
2.1.1 Technical Feasibility Study:	5
2.1.2 Economical Feasibility Study:.....	6
2.1.3 Behavior Feasibility Study:.....	6
2.2 Hardware and Software Requirements of The System:	7
2.2.1 Hardware Requirements:.....	7
2.2.2 Software Requirements:	7
3. SYSTEM DESIGN	8
3.1 Introduction:	8
3.1.1 Data Flow Diagram:.....	9
3.1.2 System Context Diagram:.....	11
3.1.3 Entity Relation Diagram:	12
3.1.4 System Flow Diagram:	13
4. CODING:	16
4.1 Database Connection:.....	19
4.2 Insert into Table:	20
4.3 Global Data:	21
4.4 Read Data from Database:.....	21
4.5 Update Data:	23
4.6 Delete Data:	25
4.7 Data Grid View:	26
5. TESTING:.....	29

5.1 Testing Strategies:.....	30
5.1.1 Unit Testing	31
5.1.2 Performance Testing:.....	32
5.1.3 System Testing:	33
5.2 Login Screen:.....	34
5.3 Input Output Screen of The Main Forms:	35
5.3.1 Customer Service form:	35
5.3.2 Deposit Form:.....	36
5.3.3 Withdrew Form	37
5.3.4 Loan Form:	38
5.3.5 Create Account Form:	39
5.3.6 Existing Account Form:	40
5.3.7 Loan Account Form:	41
5.3.8 Administration Form:.....	42
6. TEST REPORTS:	43
7. CONCLUSION:.....	45
7.1 Project Benefits:.....	46
7.2 Project Limitation:.....	46
8. REFERENCES:	46

1. INTRODUCTION

1.1 Background:

This project belongs to the banking management system which can be used by the simple in order to keep their data and further processing as desire by the customer on the basis of certain rules and policies.

It is able to cover the following feature

- It facilitates create account, display account, deposit account, withdraw account, bank amount, modify account, and at last delete account, account detail, loan operations.
- It keeps the day by day daily record as a complete banking system. It can keep the information of account type.
- It facilitates the customer to add, modify extra data as their needs some likes in practical.

In our project the main operations of that banking system are more emphasized to make those operations faster, reliable and more sophisticated. Those operations which are mainly focused on our project are: create account, withdraw, deposit, modify, delete transaction, create loan account, payment of installment, interest paid, and loan details and more.

1.2 Objectives:

The main object of this system is to provide a secure system. Our system is password protected and it only allows authorized user to access various functions available in the system.

It will Reduced manual work as most of the work done by computer. As all the manual work will be done automatically so it will increase work speed, reduce time consumption to complete any bank related work and reduced the manual workload and give information instantly.

Through this bank management system user can manage all bank account activity like deposit money, withdraw money, transfer money from one account to another account, user can check his account detail like balance in account, bank statement etc.

The Administrator can check bank account with a login can work out with A/C holders of the bank can withdraw/ deposit cash /transactions/ from their accounts.

In this project we have the bank process like Account Opening, Daily Transactions, Loan Sanctions, Account Maintenance. In this bank management system use can also search record of a particular Account Holder. Using this system user can manage following account type:

- Savings Account
- Current Account
- Fixed Deposit Account
- Loan Account
- Transactions
- Bank Detail
- Account Statement

1.3 Time Plan:

In this project “Bank Management System” we have used the Time plan of **Gantt Chart**. A Gantt Chart, commonly used in project management, is one of the most popular and useful ways of showing activities (tasks or events) displayed against time. On the left of the chart is a list of the activities and along the top is a suitable time scale. Each activity is representing by a bar; the position and length of the bar reflects the start date, duration and end date of the activity. This allows us to see at a glance:

- What the various activities are
- When each activity begins and ends, The start and end date of the whole project

1.4 Investigation of Problem:

We have seen the various problems in the bank during the banking transaction. The problem is that they keep the records in the book which is tedious task due to data of the banks in the very large size. So, it is very slow process to find any record from the file and calculation of daily transaction such as withdrawal, deposit, taking loan.

2. REQUIREMENTS OF THE ANALYSIS OF THE SYSTEM:

For the requirement of the analysis of the system there is a feasibility analysis in which we have used this, Feasibility Analysis in our project. The description of the feasibility analysis is given below.

2.1 Feasibility Study:

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features:

2.1.1 Technical Feasibility Study:

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed. Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology. Through the technology may become obsolete after some period of time, due to the fact that never version of same software supports older versions, the system may still be used. So, there are minimal constraints involved with this project. The system has been developed using Java the project is technically feasible for development.

2.1.2 Economical Feasibility Study:

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also, all the resources are already available, it gives an indication of the system is economically possible for development.

2.1.3 Behavior Feasibility Study:

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

2.2 Hardware and Software Requirements of The System:

2.2.1 Hardware Requirements:

Hardware requirement includes the following components:

1. Any computer (PC, Laptop etc.)
2. RAM (at list 2 GB)
3. Hard-disk (at list 5GB)
4. Visual Display Unit (CRT or LCD)
5. Processor (at list i3 2.23 GHZ)

2.2.2 Software Requirements:

It includes the following components:

1. Window Operating System Xp, 7 or Higher
2. Visual Basic 6.0 complier
3. Database (MySQL)
4. Microsoft .NET framework 4.5 and above

3. SYSTEM DESIGN

3.1 Introduction:

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. System design goes through two phases of development: Logical and Physical Design.

LOGICAL DESIGN:

The logical flow of a system and define the boundaries of a system. It includes:

- Reviews the current physical system – its data flows, file content, volumes, frequencies etc.
- Prepares output specifications – that is, determines the format, content and frequency of reports.
- Prepares input specifications – format, content and most of the input functions. Prepares edit, security and control specifications.
- Specifies the implementation plan.
- Prepares a logical design walk through of the information flow, output, input, controls and implementation plan.
- Reviews benefits, costs, target dates and system constraints.

PHYSICAL DESIGN:

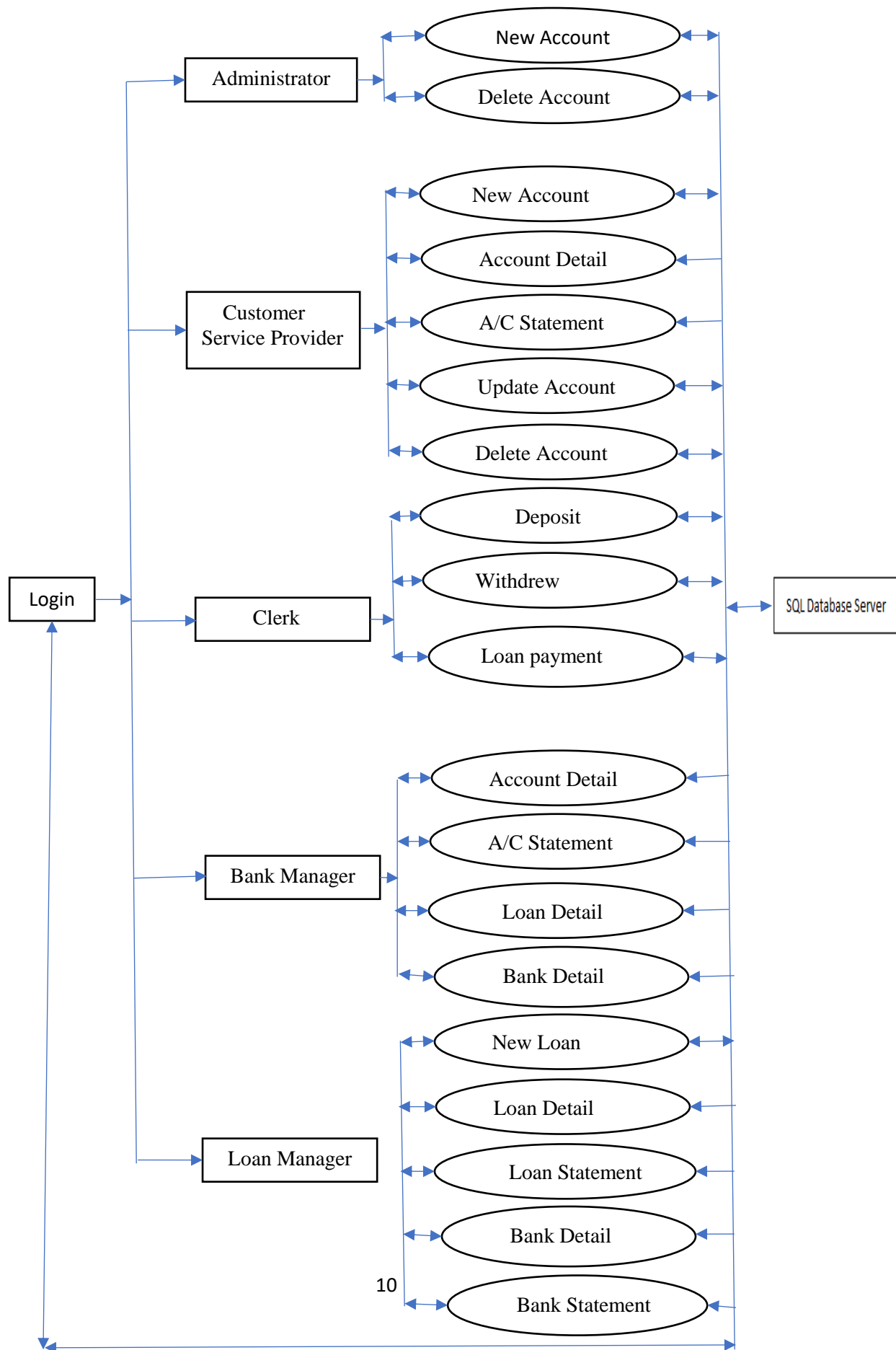
Physical system produces the working systems by define the design specifications that tell the programmers exactly what the candidate system must do. It includes the following steps.

- Design the physical system.
- Specify input and output media.
- Design the database and specify backup procedures.
- Design physical information flow through the system and a physical design Walk through.
- Plan system implementation.
- Prepare a conversion schedule and target date.
- Determine training procedures, courses and timetable.
- Devise a test and implementation plan and specify any new hardware/software.
- Update benefits, costs, conversion date and system constraints.

3.1.1 Data Flow Diagram:

The diagram that allows the model how data flow through the Banking Management System, the relationship among the data flows, and how data come to be store at specific bank data base is called as DFD (Data Flow Diagram). It also shows the processes that changes or transform the data, because DFD concentrate on the movement of the data processes, these are also called process model.

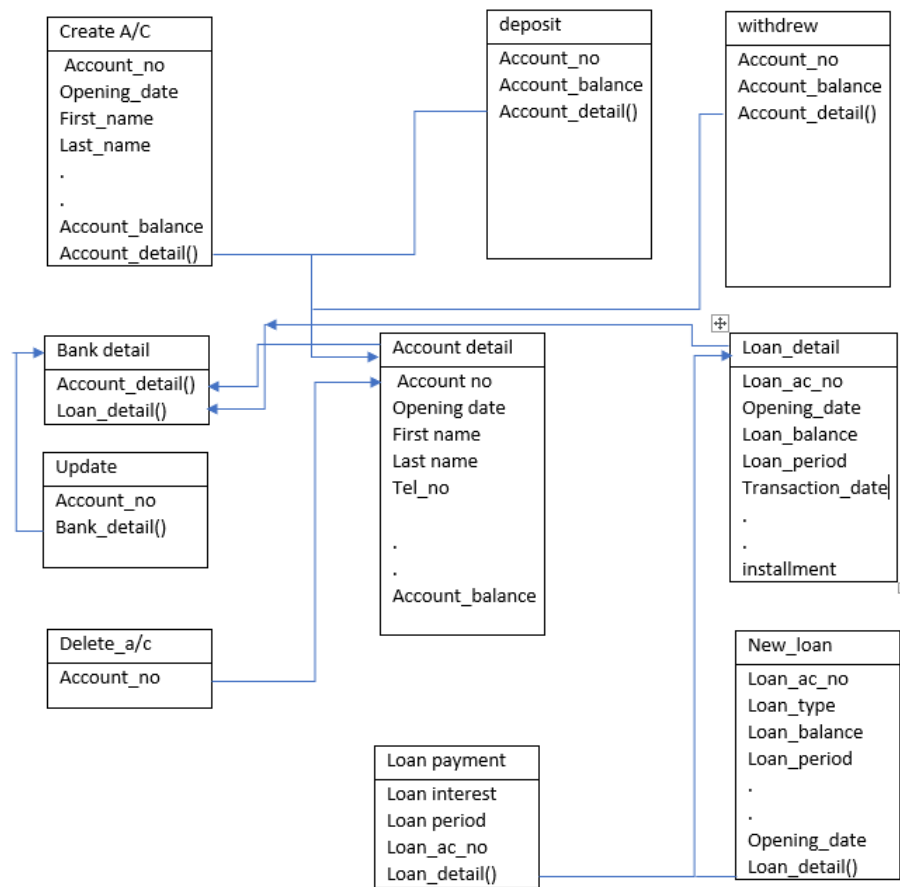
The following data flow diagram represents about Bank Management System.



3.1.2 System Context Diagram:

System Context Diagrams... represent all external entities that may interact with a system... Such a diagram pictures the system at the center, with no details of its interior structure, surrounded by all its interacting systems, environments and activities. The objective of the system context diagram is to focus attention on external factors and events that should be considered in developing a complete set of systems requirements and constraints.

The following system flow diagram represents about Bank Management System.

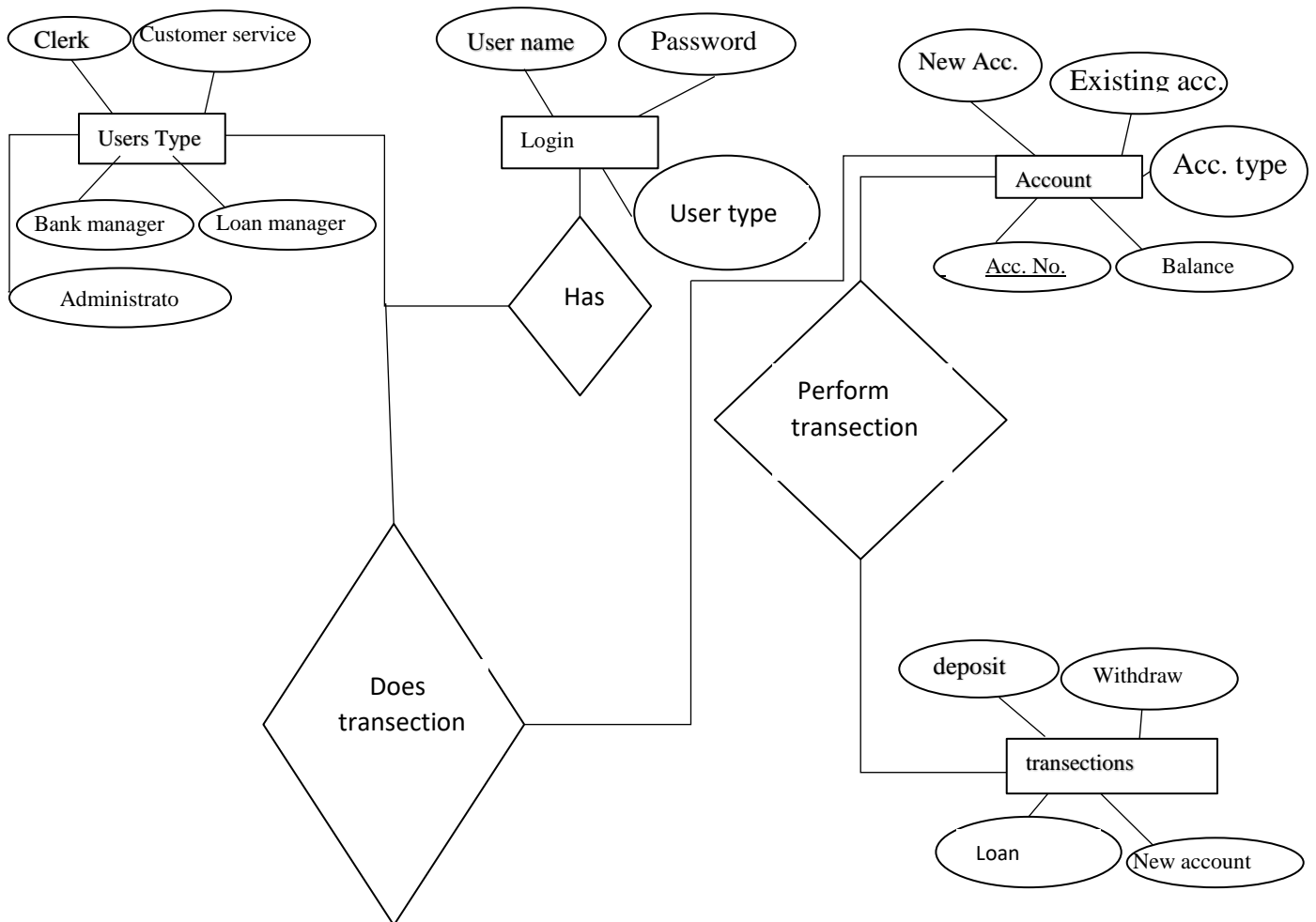


(fig. System Context Diagram)

3.1.3 Entity Relation Diagram:

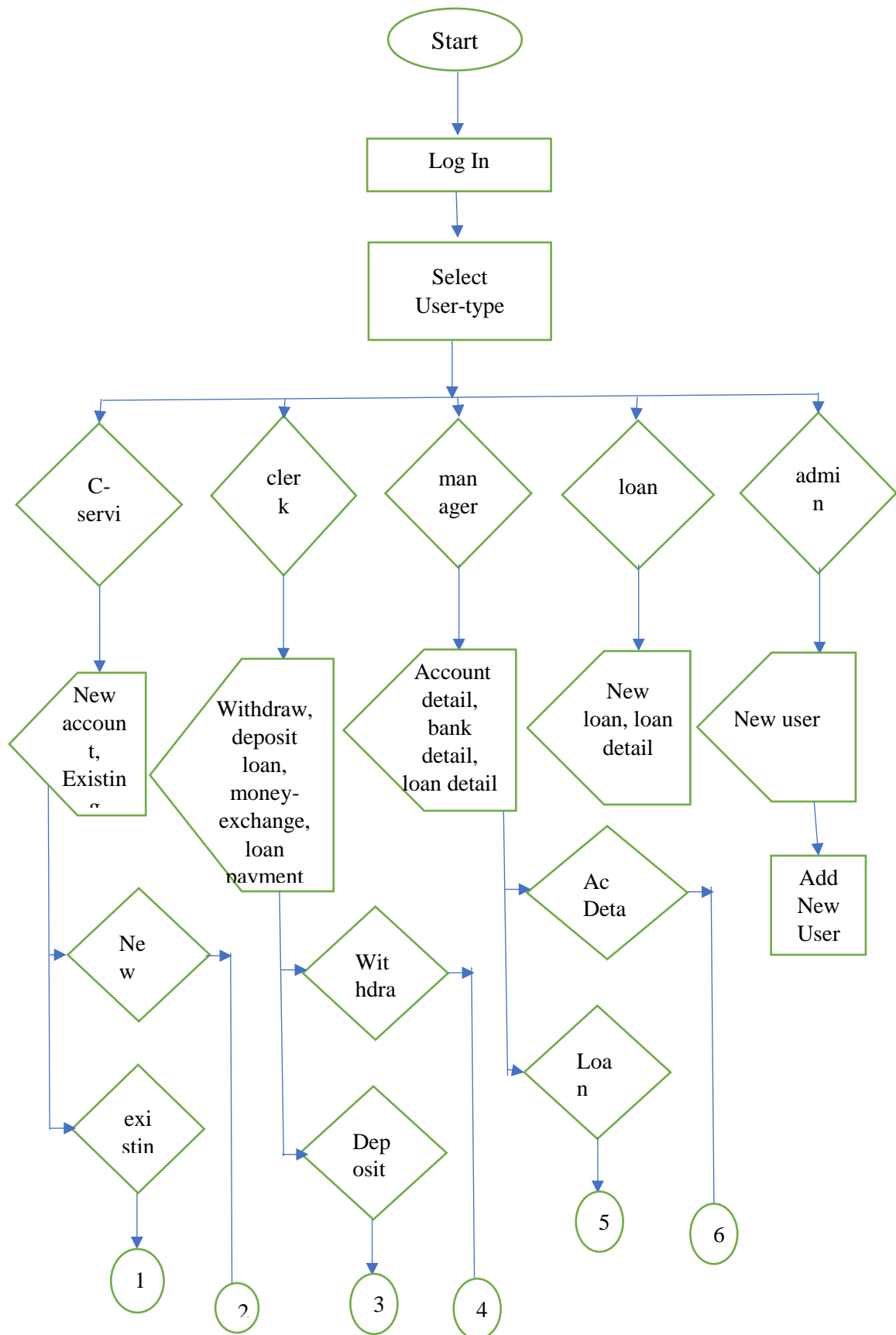
An entity–relationship model (ER model) describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types.

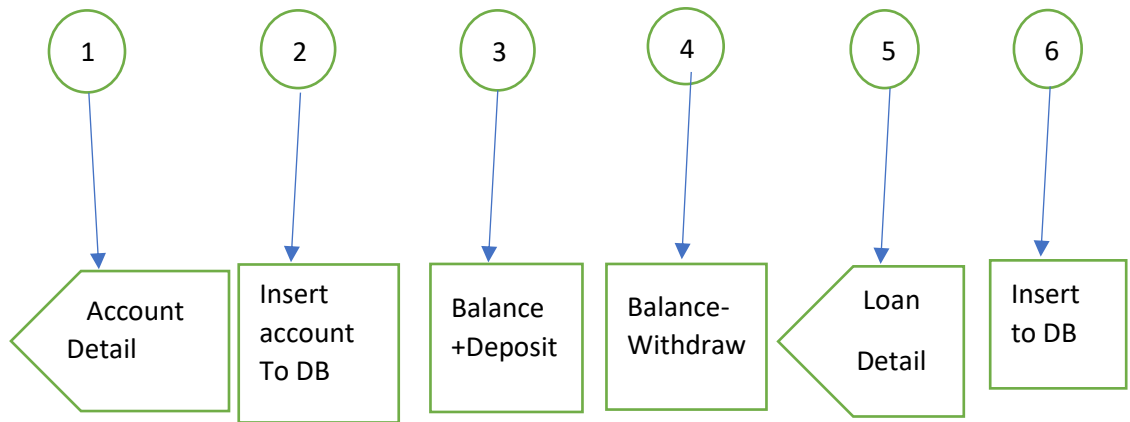
The following ER-diagram represents about Bank Management System.



(Fig. ER-Diagram)

3.1.4 System Flow Diagram:





4. CODING:

1. Creating Database Omega_bank

```
Dim Query As String
Dim cmd As New SqlCommand
Dim conn As New SqlConnection
Dim rdr As SqlDataReader

Try
    ' Dim rdr As SqlDataReader

    conn = New SqlConnection(" Data Source=.;Initial Catalog=;Integrated
Security=True ")

    conn.Open()

    Query = "if(select name from sys.databases where name = 'Omega_bank')is
null begin create database Omega_bank end "

    cmd = New SqlCommand(Query, conn)
    cmd.ExecuteNonQuery()

    Catch ex As Exception

        MessageBox.Show("Error 1: " & ex.Message)

    Finally

        conn.Close()

    End Try
```

2. Creating Table

```
Dim Query As String
```

```
Dim y As Int32 = 0
```

```
Dim r As SqlDataReader
```

```
Dim n As Int32
```

```
Try
```

```
    Call DatabaseConection.connect()
```

```
    Query = " select COUNT( *) as n from information_schema.tables "
```

```
    cmd = New SqlCommand(Query, con)
```

```
    r = cmd.ExecuteReader
```

```
    If r.Read = True Then
```

```
        y = r(n)
```

```
    End If
```

```
    r.Close()
```

```
Catch ex As Exception
```

```
    MessageBox.Show("Error1 : " & ex.Message)
```

```
Finally
```

```
    con.Close()
```

```
End Try
```

```
Try
```

```
    Call DatabaseConection.connect()
```

```
    con.ChangeDatabase("Omega_bank")
```

```
    cmd.CommandText = "create table account_detail( account_no  
varchar(20) NOT Null PRIMARY KEY,opening_date date Not Null,first_name  
varchar(30) Not Null,last_name varchar(30) Not Null , dob_bs date NOT Null,  
dob_ad date Not Null , account_type varchar(20) Not Null, ac_currency  
varchar(20), gender varchar(5) Not Null, citizenship_no varchar(20) Not Null,  
issue_place varchar(30) Not Null, issue_date date Not Null, nationality
```

```
varchar(20) Not Null, father_name varchar(30) Not Null, occupation
varchar(30) Not Null, address_ varchar(50) Not Null,intrest_rate float,
account_balance float, avaiable_balance float, mob_no varchar(15), tel_no
varchar(10), email varchar(50),photo varchar(500),intrest_date date)"
```

```
cmd.ExecuteNonQuery()
```

```
Catch ex As Exception
```

```
MsgBox("Error 3 : " + " " + ex.Message)
```

```
Finally
```

```
con.Close()
```

```
End Tr
```

```
Call DatabaseConection.connect()
```

```
con.ChangeDatabase("Omega_bank")
```

```
cmd.CommandText = "create table loan_detail( loan_acc_no
varchar(20) NOT Null PRIMARY KEY, opening_date date Not
Null,loan_first_name varchar(30) Not Null,loan_last_name varchar(30) Not
Null , loan_dob_bs date NOT Null, loan_dob_ad date Not Null , loan_type
varchar(20) Not Null, loan_gender varchar(5) Not Null, loan_citizen_no
varchar(20) Not Null, loan_issue_place varchar(30) Not Null, loan_issue_date
date Not Null, loan_nationality varchar(20) Not Null, loan_father_name
varchar(30) Not Null, loan_occupation varchar(30) Not Null, loan_address_
varchar(50) Not Null,loan_intrest_rate float, loan_balance float, intrest_paid
float, loan_mob_no varchar(15), loan_tel varchar(10), loan_email
varchar(50),installment float,installment_paid float,loan_period float,photo
varchar(500))"
```

```
cmd.ExecuteNonQuery()
```

```
Catch ex As Exception
```

```
MsgBox("Error 3 : " + " " + ex.Message)
```

```
Finally
```

```
        con.Close()  
    End Try
```

4.1 Database Connection:

```
Imports System.Data.SqlClient  
Module DatabaseConection  
    Public cmd As New SqlCommand  
    Public cmmd As New SqlCommand  
    Public con As New SqlConnection  
    Public Sub connect()  
        Try  
            con.ConnectionString = "Data  
Source=.;Initial Catalog=Omaga_bank;Integrated  
Security=True"  
            con.Open()  
            cmd.Connection = con  
            cmmd.Connection = con  
        Catch ex As Exception  
            MessageBox.Show("Error while inserting  
record on table..." & ex.Message, "Insert Records")  
        End Try  
    End Sub  
End Module
```



```

        End Try
    End Sub
End Module

```

4.2 Insert into Table:

```

Try
    Call DatabaseConection.connect()

    cmmd.CommandText = " update
accountno_generator set new_acc_no = '" &
AccountNoTextbox.Text & "'"

    cmmd.ExecuteNonQuery()

    cmd.CommandText = "INSERT INTO
account_detail(opening_date,account_no,first_name,
last_name, account_type, ac_currency, dob_bs,
dob_ad,gender, Citizenship_no, issue_date,
issue_place, nationality, father_name, occupation,
address_,
mob_no,tel_no,email,intrest_rate,account_balance,avali
able_balance) VALUES ('" & CDate(DateTextBox.Text) &
"', '" & AccountNoTextbox.Text & "', '" &
FNameTextbox.Text & "', '" & LnameTextbox.Text & "', '"
& AccountTypeCombo.Text & "', '" & CurrencyTextbox.Text
& "', '" & CDate(DobBsTextbox.Text) & "', '" &
CDate(DobAdTextbox.Text) & "', ' " & gen & "', '" &
CitizenNoTextbox.Text & "', '" &
CDate(CDateTextbox.Text) & "', '" & CPlaceTextbox.Text
& "', '" & NationalityTextbox.Text & "', '" &
FatherNameTextbox.Text & "', '" &
OccupationTextbox.Text & "', '" & AddressTextbox.Text &
"', '" & CInt(MobNoTextbox.Text) & "', '" &
CInt(TelTextbox.Text) & "', '" & EmailTextbox.Text &

```

```

"', '" & Convert.ToDouble(IntrestRateTextBox.Text) &
"', '" & CDb1(MinBalTextBox.Text) & "', '" & 0 & "'"
        cmd.ExecuteNonQuery()

        MsgBox("Account is saved sucessfully")

    Catch ex As Exception
        MsgBox("error" + " " + ex.Message)
    Finally
        con.Close()
    End Try

```

4.3 Global Data:

```

Module globaldata
    Public username As String
    Public account_no As String
    Public user_type As String
    Public loan_account_no As String
End Module

```

4.4 Read Data from Database:

```

Dim reader As SqlDataReader
    Dim Query As String
    Call DatabaseConection.connect()
    con.ChangeDatabase("Omaga_bank")

```

```

        Query = "select *from account_detail Where
account_no = '" & globaldata.loan_account_no & "'"
        cmd = New SqlCommand(Query, con)
        reader = cmd.ExecuteReader
        If reader.Read() = True Then
            ODateTextBox.Text = reader("opening_date")
            AccNOTextBox.Text = reader("account_no")
            FNameTextBox.Text = reader("first_name") +
" " + reader("last_name")
            AccountTypeTextBox.Text =
reader("account_type")
            DobBsTextBox.Text = reader("dob_bs")
            DobAdTextBox.Text = reader("dob_ad")
            GenderTextBox.Text = reader("gender")
            CitizenNoTextBox.Text =
reader("citizenship_no")
            CPlaceTextBox.Text = reader("issue_place")
            CDateTextBox.Text = reader("issue_date")
            NationalityTextBox.Text =
reader("nationality")
            FatherNameTextBox.Text =
reader("father_name")
            OccupationTextBox.Text =
reader("occupation")
            AddressTextBox.Text = reader("address_")
            MobNoTextBox.Text = CStr(reader("mob_no"))
            TelTextBox.Text = CStr(reader("tel_no"))

```

```

        EmailTextBox.Text = reader("email")

        AccCurrencyTextBox.Text =
reader("ac_currency")

        IntrestRateTextBox.Text =
reader("intrest_rate")

        AccBalanceTextBox.Text =
reader("account_balance")

        AvlBalTextBox.Text =
reader("avaliable_balance")

        ' PhotoBox.Image = reader("photo")
End If

reader.Close()

con.Close()

```

4.5 Update Data:

```

Try

        Call DatabaseConection.connect()

        cmmd.CommandText = " update
account_detail set occupation='" &
OccupationTextBox.Text & "' , address_='" &
AddressTextBox.Text & "' Where account_no = '" &
globaldata.account_no & "' "

        cmmd.ExecuteNonQuery()

Catch ex As Exception

        MsgBox("error" + " " + ex.Message)

Finally

```

```

        con.Close()

    End Try

End If

    If MobNoTextBox.Text = "" = False And
MobNoTextBox.Text = " " = False Then

        Try

            Call DatabaseConection.connect()

            cmmd.CommandText = " update
account_detail    set mob_no =' " &
CInt(MobNoTextBox.Text) & "'  Where account_no = ' " &
globaldata.account_no & "' "

            cmmd.ExecuteNonQuery()

            Catch ex As Exception

                MsgBox("error" + " " + ex.Message)

            Finally

                con.Close()

            End Try

        End If

        If TelTextBox.Text = "" = False And
TelTextBox.Text = " " = False Then

            Try

                Call DatabaseConection.connect()

                cmmd.CommandText = " update
account_detail    set mob_no =' " &
CInt(TelTextBox.Text) & "'  Where account_no = ' " &
globaldata.account_no & "' "

                cmmd.ExecuteNonQuery()

```

```

Catch ex As Exception
    MsgBox("error" + " " + ex.Message)
Finally
    con.Close()
End Try

```

4.6 Delete Data:

```

Try

    Call DatabaseConection.connect()

    cmmd.CommandText = " delete from
account_detail Where account_no = '" &
globaldata.account_no & "' "

    cmmd.ExecuteNonQuery()

Catch ex As Exception
    MsgBox("error" + " " + ex. Message)
Finally
    con.Close()
End Try

Try

    Call DatabaseConection.connect()

    cmmd.CommandText = " delete from
bank_statement Where account_no = '" &
globaldata.account_no & "' "

    cmmd.ExecuteNonQuery()

Catch ex As Exception
    MsgBox("error" + " " + ex.Message)

```

```

Finally
    con.Close()
End Try

Try
    Call DatabaseConection.connect()

    cmmd.CommandText = " delete from
account_statement Where account_no = '" &
globaldata.account_no & "' "

    cmmd.ExecuteNonQuery()

Catch ex As Exception
    MsgBox("error" + " " + ex.Message)
Finally
    con.Close()

    SearchAccountForm.Show()

    AccountDetail.Enabled = True

    Me.Hide()
End Try

```

4.7 Data Grid View:

```

Try

    Call DatabaseConection.connect()

    con.ChangeDatabase("Omaga_bank")

    Query = "select *from account_statement
Where account_no = '" & globaldata.account_no & "' and

```

```

tranc_date between '"' & startdate & '"' and '"' &
enddate & "';"

        cmd = New SqlCommand(Query, con)

        rdr = cmd.ExecuteReader

        Dim d As Integer =
StatementDataGridView.Rows.Add()

        While rdr.Read()

            ' If d = 0 = True Then

StatementDataGridView.Rows.Item(d).Cells(2).Value =
"Opening Balance"

StatementDataGridView.Rows.Item(d).Cells(6).Value =
rdr("opening_balance")

            ' d = d + 1

            ' End If

            Dim n As Integer =
StatementDataGridView.Rows.Add()

StatementDataGridView.Rows.Item(n).Cells(0).Value =
rdr("tranc_date")

StatementDataGridView.Rows.Item(n).Cells(1).Value =
rdr("tranc_id")

StatementDataGridView.Rows.Item(n).Cells(2).Value =
rdr("description_")

```



```

StatementDataGridView.Rows.Item(n).Cells(3).Value =
rdr("valu_date")

StatementDataGridView.Rows.Item(n).Cells(4).Value =
rdr("debit")

StatementDataGridView.Rows.Item(n).Cells(5).Value =
rdr("creadit")

StatementDataGridView.Rows.Item(n).Cells(6).Value =
rdr("balance")

End While

d = StatementDataGridView.Rows.Add()

StatementDataGridView.Rows.Item(d).Cells(2).Value =
"Closing Balance"

StatementDataGridView.Rows.Item(d).Cells(0).Value =
Date.Now.ToString("yyyy-MM-dd")

StatementDataGridView.Rows.Item(d).Cells(6).Value =
StatementDataGridView.Rows.Item(d - 1).Cells(6).Value

rdr.Close()

Catch ext As Exception

MsgBox(ext.Message)

Finally

con.Close()

```

End Try

5. TESTING:

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and to verify that the software product is fit for use.

Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

- meets the requirements that guided its design and development,
- responds correctly to all kinds of inputs,
- performs its functions within an acceptable time,
- is sufficiently usable,
- can be installed and run in its intended environments, and
- achieves the general result its stakeholders desire.

As the number of possible tests for even simple software components is practically infinite, all software testing uses some strategy to select tests that are feasible for the available time and resources. As a result, software testing typically (but not

exclusively) attempts to execute a program or application with the intent of finding software bugs (errors or other defects). The job of testing is an iterative process as when one bug is fixed; it can illuminate other, deeper bugs, or can even create new ones.

Software testing can be conducted as soon as executable software (even if partially complete) exists. The overall approach to software development often determines when and how testing is conducted. For example, in a phased process, most testing occurs after system requirements have been defined and then implemented in testable programs. In contrast, under an Agile approach, requirements, programming, and testing are often done concurrently.

Most of us have had an experience with software that did not work as expected. Software that does not work can have a large impact on an organization. It can lead to many problems including:

- **Loss of money** – This can include losing customers right through to financial penalties for non-compliance to legal requirements
- **Loss of time** – This can be caused by transactions taking a long time to process but can include staff not being able to work due to a fault or failure
- **Damage to business reputation** – If an organization is unable to provide service to their customers due to software problems then the customers will lose confidence or faith in this organization (and probably take their business elsewhere)

5.1 Testing Strategies:

Testing begins at the component level and works outward toward the integration of the entire computer-based system. Different testing techniques are appropriate at different points in time.

The developer of the software conducts testing and may be assisted by independent test groups for large projects. The role of the independent tester is to remove the conflict of interest inherent when the builder is testing his or her own product.

5.1.1 Unit Testing

Unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.

Unit testing is commonly automated, but may still be performed manually. The IEEE does not favor one over the other. The objective in unit testing is to isolate a unit and validate its correctness. A manual approach to unit testing may employ a step-by-step instructional document. However, automation is efficient for achieving this, and enables the many benefits listed in this article. Conversely, if not planned carefully, a careless manual unit test case may

execute as an integration test case that involves many software components, and thus preclude the achievement of most if not all of the goals established for unit testing.

To fully realize the effect of isolation while using an automated approach, the unit or code body under test is executed within a framework outside of its natural environment. In other words, it is executed outside of the product or calling context for which it was originally created. Testing in such an isolated manner reveals unnecessary dependencies between the code being tested and other units or data spaces in the product. These dependencies can then be eliminated.

As a consequence, unit testing is traditionally a motivator for programmers to create decoupled and cohesive code bodies. This practice promotes healthy habits in software development. Design patterns, unit testing, and refactoring often work together so that the best solution may emerge.

Advantages:

- Find problem early
- Facilitates change
- Simplifies Integration
- Documentation
- Design

5.1.2 Performance Testing:

Performance testing is a testing practice performed to determine how a system performs in terms of responsiveness and stability under a particular workload. It can also serve to investigate measure, validate or verify other quality attributes of the system, such as scalability, reliability and resource usage. Performance testing, a subset of performance engineering, is a computer science practice which strives to build performance standards into the implementation, design and architecture of a system.

Load testing: Load testing is the simplest form of performance testing. A load test is usually conducted to understand the behavior of the system under a specific expected load. This load can be the expected concurrent number of users on the application performing a specific number of transactions within the set duration. This test will give out the response

- times of all the important business critical transactions. The database, application server, etc. are also monitored during the test, this will assist in identifying bottlenecks in the application software and the hardware that the software is installed on.
- **Stress testing:** Stress testing is normally used to understand the upper limits of capacity within the system. This kind of test is done to determine the system's robustness in terms of extreme load and helps application administrators to determine if the system will perform sufficiently if the current load goes well above the expected maximum.
- **Soak testing:** Soak testing, also known as endurance testing, is usually done to determine if the system can sustain the continuous expected load. During soak tests, memory utilization is monitored to detect potential leaks. Also important, but often overlooked is performance degradation, i.e. to ensure that the throughput and/or response times after some long period of sustained activity are as good as or better than at the beginning of the test. It essentially involves applying a significant load to a system for an extended, significant period of time.
- **Spike testing:** Spike testing is done by suddenly increasing or decreasing the load generated by a very large number of users, and observing the

behavior of the system. The goal is to determine whether performance will suffer.

- **Isolation testing:** Isolation testing is not unique to performance testing but involves repeating a test execution that resulted in a system problem. Such testing can often isolate and confirm the fault domain.

5.1.3 System Testing:

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

As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

- Recovery testing – checks system's ability to recover from failures
- Security testing – varieties that system protection mechanism prevents improper penetration or data alteration
- Stress testing – program is checked to see how well it deals with abnormal resource demands
- Performance testing – tests the run-time performance of software

White Box Approach


This approach concentrates on the actual codes written during the development of the project. It checks every lines of code in all the function of the program.

Black Box Approach

This approach concentrates on the basic requirements of the project. It simply checks whether the project can be run without detail description of internal coding. It checks direct matching of records of particular book, after we select a book number of particular student.

5.2 Login Screen:

We have put the Log In features for the security purpose in this project. For the different users type to access to the different form of specific form.




Mega Bank

Customer Service

admin

#####

 Log In

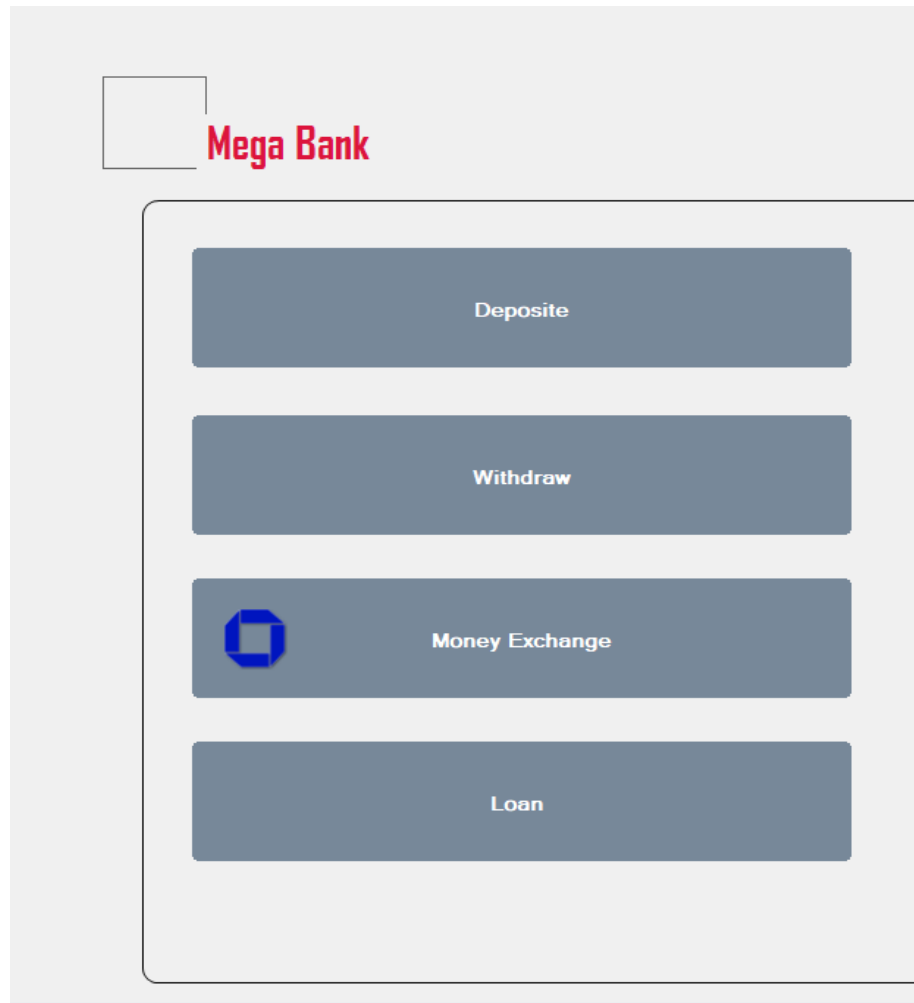
About Us

(Fig. Login Screen)

5.3 Input Output Screen of The Main Forms:

5.3.1 Customer Service form:

In this form we can obtain following form.



The image shows a digital interface for Mega Bank's customer service. At the top left, there is a square placeholder for a logo. To its right, the text "Mega Bank" is displayed in a red, sans-serif font. Below this header, a large, rounded rectangular container holds four dark blue buttons stacked vertically. The buttons are labeled "Deposit", "Withdraw", "Money Exchange", and "Loan" from top to bottom. The "Money Exchange" button is distinguished by a small blue icon of a square with a circle inside, positioned to the left of the text. The entire interface is set against a light gray background.

(Fig. Customer Service form)

5.3.2 Deposit Form:

In this form we can obtain the deposit form and the output. Following is the form of the deposit.

The screenshot shows a web application interface for a bank clerk. At the top, there is a blue header bar with window control icons on the right. Below the header, the interface includes a 'Mega Bank' logo on the left, a 'Clerk Desk' title in the center, and a 'Back' button on the right. A date and time stamp 'Date: 2018 Jan 26 12:35:57' is located in the top right corner. The main content area is titled 'Cash Deposit' and contains a form with the following fields: 'Date' (2018 01 26), 'Trac Id' (CD201801263507), 'Account No' (5), 'Account Holder Name' (ram adh), 'Opening Balance' (100), 'Currency' (NRP), 'Cr Amount' (500000), and 'Deposited By' (ram adh). A 'Save' button is positioned at the bottom of the form.

Date :	2018 01 26	Trac Id :	CD201801263507
Account No :	5		
Account Holder Name :	ram adh		
Opening Balance :	100		
Currency :	NRP		
Cr Amount :	500000		
Deposited By :	ram adh		

Save

(Fig. Deposit Form)

5.3.3 Withdrew Form

In this form we get the withdraw form in which a person can withdraw the money. Following is the form of the withdraw.


The screenshot shows a web application interface for a bank clerk. At the top, there is a blue header bar with navigation icons on the right. Below the header, the interface has a light gray background. On the left, there is a logo for 'Mega Bank' consisting of a square icon and the text 'Mega Bank'. In the center, the title 'Clerk Desk' is displayed. On the right, there is a 'Back' button and a date/time stamp 'Date: 2018 Jan 26 12:39:11'. The main content area is a large white rectangle with a thin black border. Inside this area, the title 'Withdraw' is centered at the top. Below the title, there is a form with several input fields and labels. The form is organized as follows: 'Date : 2018 01 26' and 'Trac Id : Wd201801263846' are at the top. Below them is 'Account No : 5'. Then 'Account Holder Name : ram adh'. Then 'Opening Balance : 500100'. Then 'Dr Amount : 5000'. Then 'Withdraw By : ram adh'. Then 'Closing Balance : 495100'. At the bottom of the form is an 'Ok' button.

Withdraw	
Date :	2018 01 26
Trac Id :	Wd201801263846
Account No :	5
Account Holder Name :	ram adh
Opening Balance :	500100
Dr Amount :	5000
Withdraw By :	ram adh
Closing Balance :	495100

(Fig. Withdraw Form)

5.3.4 Loan Form:

In this form we can obtain the loan form where we can get the loan of the any type. The form of the loan is shown below:



Loan manager Desk

Date: 2018 Jan 26 12:54:56

Back

Loan Account NO : 4

First Name : ram

Last Name : adh

Date Of Birth : 2053 02 13 BS 1997 05 30 AD

Loan Type : home loan

Interest rate : 24

Gender : ☒ Male ☐ Female

Citizenship / PassPort No : 5434

Issue Place : chitwan

Issue Date : 2018 01 25

Nationality : nepali

5.3.5 Create Account Form:

In this form we can create the account. The following form is the form of the create account.

Customer Services

Date :

Account No :

First Name :

Last Name :

Date Of Birth : BS AD

Account Type :

Intrest Rate : Account Currency :

Gender : ☒ Male ☐ Female

Citizenship / PassPort No :

Issue Place : Issue Date :

(Fig. Create Account Form)

5.3.6 Existing Account Form:

In this existing account form we can view the account holder of account that are already created. The form of existing account form is:

Customer Services

[Back](#)

Date: 2018 Jan 26 01:44:22

Customer Detail

Account Opening Date :
 Account Holder Name :
 Account No :
 Account Type :
 Date Of Birth : BS AD
 Gender :

Citizenship / PassPort No :

Issue Place :
 Nationality :

Issue Date :

(Fig. Existing Account Form)

5.3.7 Loan Account Form:

In this loan detail form we can see the loan detail.

Mega Bank

Loan manager Desk

Date: 2018 Jan 26 12:56:12

Back


Email : rem@gmail.com

Tel. : 987654567

Mobile No. : 9876543

Balance : 500000

Loan Period : 5

Photo :  Choose Photo

Save

OMegaBank

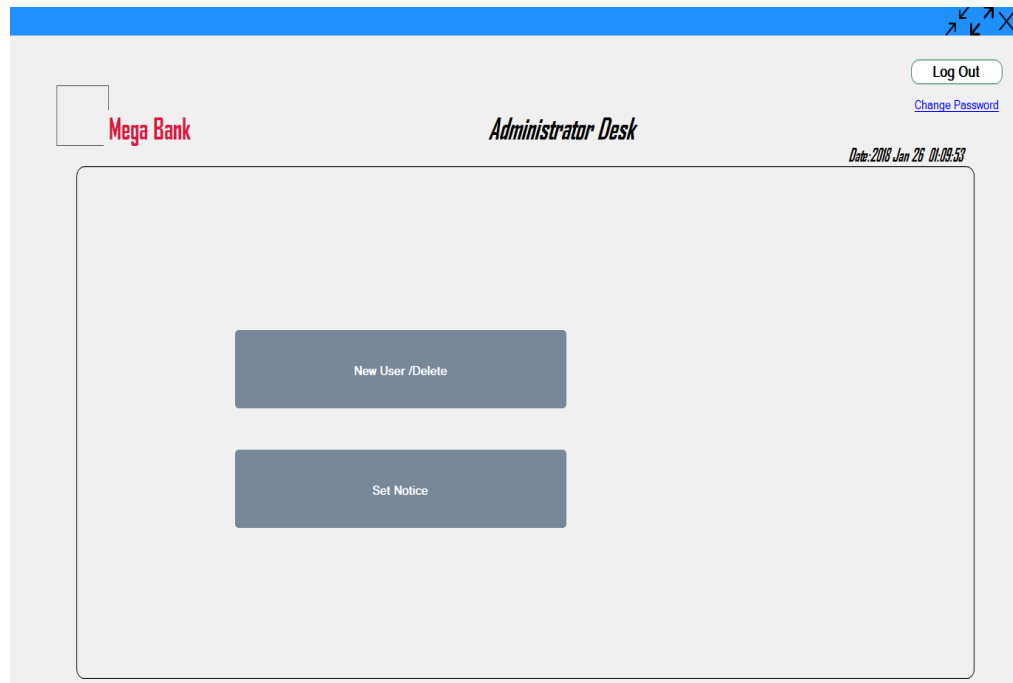
Loan Account is saved sucesfully

OK

(Fig. Loan Account Form)

5.3.8 Administration Form:

In this administration form we can modified the account.



(Fig. Administration Form)

6. TEST REPORTS:

Since this test concentrates on each unit of the software as implemented in source code, the test was conducted keeping every small module in consideration. The

following is the report of a few bugs that were overcome during the test and those remain unchecked.

S.N.	Bugs:	Solution
1	Different backup generated an execution time error Error description: Constraint Violation	Different, files and transaction log backups avoided
2	Interface command to the enterprise manager e=was able to executed with mere command of shell. Error description: File Not Executable.	Re-do the whole process of the project starting to end.
3	Size of files and response details could not be loaded during the display of search contents. Error description: Process Access Denied.	Permitted attributes displayed.
4	Unable to detect associated files by itself when called for. Error description: File Not Found.	Required association assembled. Detection remains unsolved.

Most of the errors caused by human because human develop system and manipulation is done by human themselves only, hence nobody can deny the possibility of having committing some or other mistake or the existence of bugs. Even though, extreme condition should be taken care for maintaining the data and respective entries of the system for the testing of reports. A set of test materials is nothing more than a list of possible problems in a program and set of procedure for determining whether the problems actually exist and are significant or not for determine whether the problems actually exist and are signified or not.

Unit Test: Unit testing, also known as component testing refers to tests that verify the functionality of a specific section of code, usually at the function level. These types of texts are usually written by developers as they work on code, to ensure that

the specific function is working as expected. One function might have multiple tests, to catch corner cases or other branches in the code. Unit testing alone cannot verify the functionality of a piece of software, but rather is used to assure that the building blocks the software uses work independently of each other.

Program Test: In this test we checked our program for the error like syntax error, logical error and so on.

String Test: We tested on program which is related to another program.

System test: System testing is the process of testing overall software system. Here in Electronic Airways Ticketing System all forms are integrated and tested.

User Acceptance Test: It is the test before the delivery of the system. This test identifies whether program works under the rules and regulation of the system specification or not. We can test our system entitled “flight Reservation System” by providing different inputs to it. Here we have to test the system on the basis of given action:

1. Add the record
2. Delete the record
3. Search the record
4. Update the record
5. Modified the record

7. CONCLUSION:

The experience was great. We got the chance to learn about how we can make the software for certain purpose. we got the chance to learn about the coding, better documentation and a good coordination of data. The exciting part of project was that how we can make the good communication in the banking organizations between customers and staffs and bank manager. How the work of the bank sector

is can be made easy and reliable. At last we got to know many things from our teachers from the discussion between the friends and professionals' hands out. We got the fair chance to share our knowledge.

7.1 Project Benefits:

- Access to privilege banking zone.
- Dedicated relationship manager.
- Facility to link with loan operation.
- User friendly.
- Reduce Maintained cost.
- It supports GUI (Graphical User Interface), operates on character commands.

7.2 Project Limitation:

- It does not have online operations.
- Only one user can access at a time.

8. REFERENCES:

For preparing this whole program, we took some help and guidance from different sources. Without help of these sources our program would not be complete our project report from the flowing sources. We took inspiration and guidance to complete our project report the following sources;

Books:

- *Visual Basic.Net programming (Black Book)*

- *Professional hands out*

Web sites:

- [https://msdn.microsoft.com/en-us/library/sxw2ez55\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/sxw2ez55(v=vs.85).aspx)

-THANK YOU-