Dr. Babasaheb Ambedkar Marathwada University,

Aurangabad

Have Satisfactorily carried out and completed the project work entitled

A

Project On

"Institute Management System"

This work is being submitted for the award of the degree in B.C.A. 2018-2019

Submitted by

Mr. Akshay K. Wankhede

Guided by

Prof. D. E. Suradkar

SAGAR COLLEGE OF MANAGEMENT, JALNA

INDEX

ACKNOWLEDGEMENT

With our great pleasure, we wish to express our knowledge under the great guidance of Prof. D. E. Suradkar who help us with her graceful support and other infrastructure with personal attention.

We are also thankful to our project guide Prof. D.E. Suradkar who herself a knowledgeable person with the great brilliance. We thanking her, for her immense interest, valuable guidance, kindly suggestion and co-operation thought out the period of undertaken which have been instrument in the success of our project.

It is matter of honour to express our special thanks to all the staff members who support us I completion of our project and provide us their own interest.
We also thankful to all our friends who have directly or indirectly supported us by morally.

Mr. Akshay K. Wankhede

B.C.A. (III rd Year)

Sagar College of Management, Jalna

CERTIFICATE

This is to certify that the seminar report entitled,

"Institute Management System"

Submitted by **Mr. Akshay K. Wankhede**, as per the requirement of *Dr. Babasaheb Ambedkar Marathwada University* In the partial fulfilment of bachelor of computer application third year for the academic year 2018-19.

Seat No.

PRINCIPAL
Sagar College, Naina

Guided by **Principle**

Prof. *D. E. Suradkar*

12-4-19

DECLARATION

I have undersign that I have completed the project work on the topic Software Development from [redacted] in partial fulfilment of B.C.A course as related to the [redacted] as per the syllabus of B.C.A. degree.

I hereby declare that this project is genuine and origin and never been submitted previously by me for the award of any other degree or any other university.

Mr. Akshay K. Wankhede

INTRODUCTION TO PROJECT

The aim of this software is to manage the Institute of Management System events. This software is developed using VISUAL BASICS 6.0 as the front end tool and MS-Access 2000 as the back end tool.

In the Jetking all the process was carried out with the help of the registers and files to store. All the calculations and information related to Customers are done and there is no possibility of miscalculation and misplacements of files or the information. To keep records with reference to Customers. its purchasing details, types of vehicles bill amounts. etc. So to solve the problem we have developed this software, which reduce the large extent of problems to store information related to Customers details and other details.

INTRODUCTION TO VISUAL BASIC

We know that computer involves writing coded Character sequence that contains combination of language element. These lines of code taken in the entity, as referred to as some program. In order to execute a source program we must put in under the control of an interpreter computer program such as VB as to convert to an executable program.

Language element: The VB language consist of large number of language element such as command button, text box, label box etc. these languages elements causes the computer to perform action. VB provides value we can write lines of code that causes an object property.

The GUI (Graphical User Interface) is a device employed by VB to make it easy for us to install and define the visual object form the frame works for us program.

It also provides us with code window and extends that make our programming codes easier.

Data access in VB consists of performing operation on physical database. We are making tables in Ms-Access and reports forms with view coding in MS VB.

INTRODUCTION TO MS-ACCESS

As we know that all relational DB is a collection of database. Ms-Access from Microsoft Corporation is also a RDBMS and it has got the following qualities.

Relational Data Base Management

Ms-Access is a relational to manage the stood in its uses only capabilities information database.

Information Representation

All the information is stored in Access is represented by only data item. Values which are stored in the table that makes up a database. Associations between data items are not logically represented in any other way.

Logical Accessibility

Every data item values stood in Access is accessible from the table in which it is stored, the name of the column of the same table and the valve of primary key that defines the row in which it is stood.

FEASIBILITY STUDY

Feasibility is the determination of whether or not a project is worth doing. The process followed in making this determination is called Feasibility Study. This type of study determines if project can and should be taken.

Once it has been determined that the project is flexible, the analyst can go ahead and prepare the project specification which finalize project requirements.

Generally, feasibility studies are taken within tight time constraints and normally culminate in a written MS ACCESS feasibility report. The content and recommendation of such study will be used as sound basis for deciding whether to proceed or to cancel the project.

Thus the feasibility study may lead to commitment to large resources it becomes necessary that it should be conducted completely and that no fundamental errors of judgments are made.

SYSTEM REQUIREMENTS

Hardware:-

- Processor : **Minimum** Intel Pentium II
- RAM : 64MB
- Hard disk : 10GB

Softwares:- (Minimum Requirements)

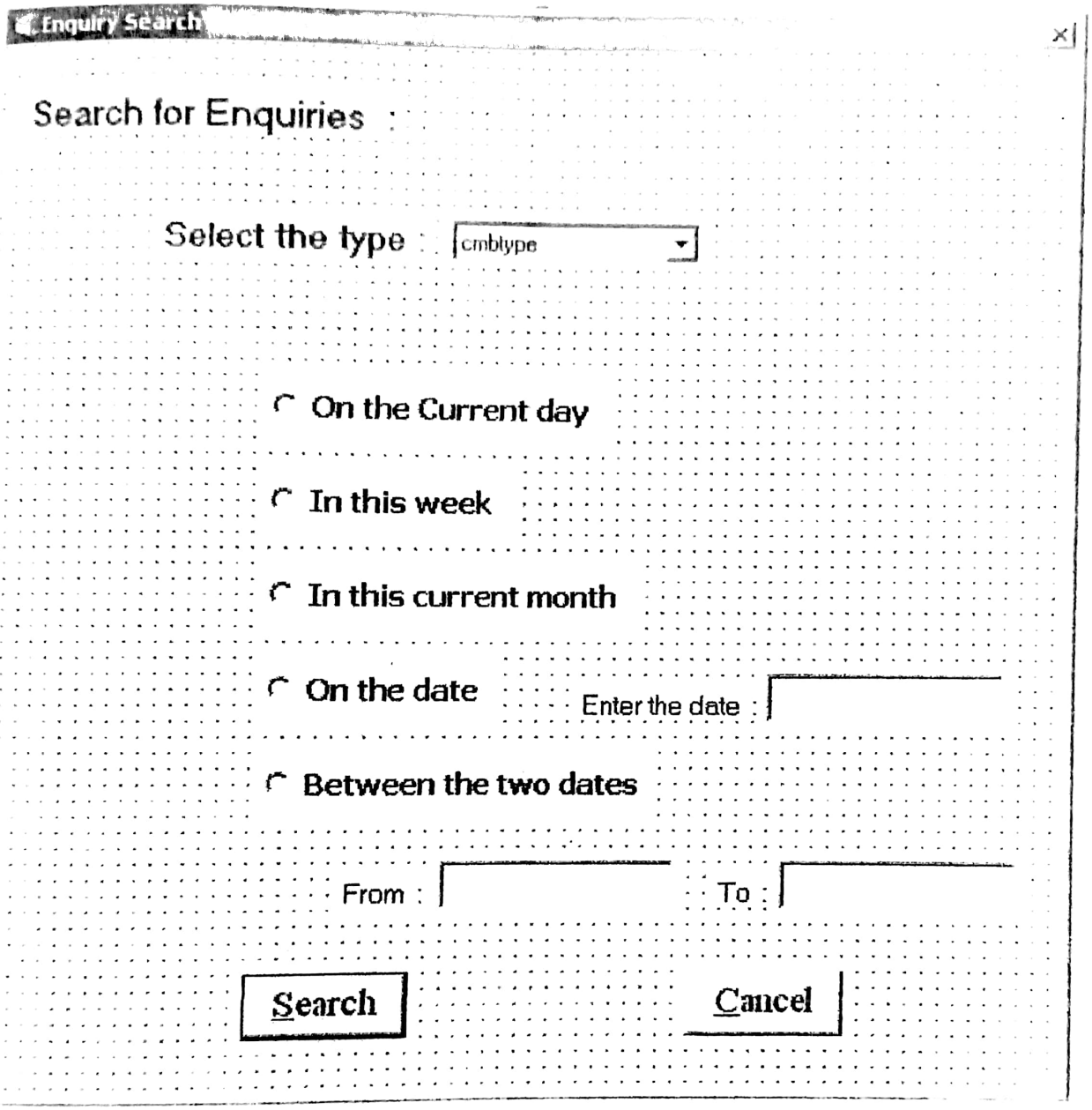
- Windows 98 Operating System
- Visual Basics 6.0
- MS-Access 2000

FORMS ADMISSION:-

```
Dim rsadm As New ADODB.Recordset
Dim rsreg As New ADODB.Recordset
Dim rsenq As New ADODB.Recordset
Dim rsint As New ADODB.Recordset
Dim rsatd As New ADODB.Recordset
Dim rslea As New ADODB.Recordset
Dim rsis As New ADODB.Recordset
Dim rstp As New ADODB.Recordset
Dim rspla As New ADODB.Recordset

Dim cn As New ADODB.Connection
Dim edit As Integer
Dim add As Integer

Private Sub chkother_Click(Index As Integer)
If chkother(1).Value = 1 Then
   txtother.Visible = True
ElseIf chkother(1).Value = 0 Then
   txtother.Visible = False
End If
End Sub

Private Sub cmdadd_Click()
Call clear
rsadm.Close
rsadm.Open "select * from admission", cn, adOpenDynamic, adLockOptimistic
txtrollno.Enabled = True
dtstartdate.Enabled = True
txtmodule.Enabled = True
txtbatch.Enabled = True
txtcentre.Enabled = True
txtcentre1.Enabled = True
txtcourse.Enabled = True
txtfrom.Enabled = True
txtto.Enabled = True
frapersonal.Enabled = True
fraeducation.Enabled = True

If rsadm.BOF = True Then
   txtsrno.Text = "1"
Else
   While rsadm.EOF = False
      rsadm.MoveNext
```

```
Private Sub cmdedit_Click()
edfname = InputBox("Enter the Student's First Name ", "First Name")
edsurname = InputBox("Enter the Student's Last Name ", "Last Name")
rsadm.Close
rsadm.Open "select * from admission where fname ='" &UCase(edfname) & "'and surname ='"
& UCase(edsurname) & "'", cn, adOpenDynamic, adLockOptimistic

If rsadm.BOF = True And rsadm.EOF = True Then
  MsgBox "No such person has taken Admission."
  rsadm.Close
  rsadm.Open "select * from admission", cn, adOpenDynamic, adLockOptimistic
  rsadm.Requery
  Exit Sub
Else
  Call display
  txtrollno.Enabled = True
  dtstartdate.Enabled = True
  txtmodule.Enabled = True
  txtbatch.Enabled = True
  txtcentre.Enabled = True
  txtcentre1.Enabled = True
  txtcourse.Enabled = True
  txtfrom.Enabled = True
  txtto.Enabled = True
  frapersonal.Enabled = True
  fraeducation.Enabled = True
  cmdsave.Enabled = True
  cmddel.Enabled = False
  cmdedit.Enabled = False
  cmdadd.Enabled = False
  cmdnext.Enabled = True
  cmdcancel.Visible = True
  cmddel.Visible = False
 edit = 1
End If
End Sub

Private Sub cmdnext_Click()
if frapersonal.Visible = True And fraeducation.Visible = True Then
  frapersonal.Visible = False
  fraeducation.Visible = False
  fraexports.Visible = True
  cmdfinish.Enabled = True
```

```
    cmdnext.Enabled = True
    cmdsave.Enabled = True
    txtgames.SetFocus
ElseIf frasports.Visible = True Then
    frasports.Visible = False
    frareg.Visible = True
    cmdback.Enabled = True
    cmdsave.Enabled = True
    cmdnext.Enabled = False
    txtfeetype.SetFocus
End If
End Sub

Private Sub cmdsave_Click()
If optyes.Value = True Or optno.Value = True Then
    GoTo b
Else
msg = MsgBox("Please select the option from the Money Back.", vbCritical, "Selection
Error")
    Exit Sub
End If
b:
If txtmodule.Text = "" Or txtcourse.Text = "" Or txtbatch.Text = "" _
    Or txtrollno.Text = "" Or txtfrom.Text = "" Or txtto.Text = "" _
    Or txtcentre1.Text = "" Or dtstartdate.Value = "" _
    Or txtsurname.Text = "" Or txtfname.Text = "" Or txtmname.Text = "" _
    Or txtaddress.Text = "" Or txtarea.Text = "" Or txtcity.Text = "" _
    Or txtpin.Text = "" Or txtresphone.Text = "" Or txtoffphone.Text = "" _
    Or txtdob.Text = "" Or txtdegree.Text = "" Or txtschool.Text = "" _
    Or txtboard.Text = "" Or txtyear.Text = "" Or txtdivision.Text = "" _
    Or txtgames.Text = "" Or txtsocialact.Text = "" Or txthobbies.Text = "" _
    Or txtspeak.Text = "" Or txtread.Text = "" Or txtwrite.Text = "" _
    Or txtplace.Text = "" Or txtdate.Text = "" Or txtserial.Text = "" _
    Or txtrollno.Text = "" Or txtfeecode.Text = "" Or txtfeetype.Text = "" Or txttotalfee.Text =
""
    Or txtdp.Text = "" Or txtdprec.Text = "" Or txtdpdate.Text = "" _
    Or txtinstamt.Text = "" Or txtinstno.Text = "" Or txt1stinstdate.Text = "" _
    Or txtlib.Text = "" Or txtlibdate.Text = "" _
    Or txtcaution.Text = "" Or txtcautionrec.Text = "" Or txtcautiondate.Text = "" _
    Or txtmb.Text = "" Or txtmbdate.Text = "" _
    Or txtscheme.Text = "" Or txtaccepted.Text = "" Or txtapproved.Text = "" _
    Or txtcounsellor.Text = "" Or txtregdate.Text = "" Or txtenqno.Text = "" Then
msg = MsgBox("You forgot to enter one of the required fields.", vbCritical + vbOKOnly,
"Input Error")
    Exit Sub
Else
```



Or

```
chkhoarding.Value = 1 Or chkbus.Value = 1 Or chkpresentation.Value = 1 _
      Or chkinq.Value = 1 Or chkdirect.Value = 1 Or chkother(1).Value = 1 Then
   Else
msg = MsgBox("Please select only one option" + vbCrLf + "from the Source options", vbCritical,
"Selection Error")
      Exit Sub
   End If
   If edit = 1 Then
     GoTo a
   End If
   While rsadm.EOF = False
     rsadm.MoveNext
   Wend
   If rsadm.EOF = True Then
     'rsadm.MoveLast
     rsadm.addnew
```

```
        'rsadm.MoveLast
        rsadm.addnew
      End If
      While rsreg.EOF = False
        rsreg.MoveNext
      Wend
      If rsreg.EOF = True Then
        'rsreg.MoveLast
        rsreg.addnew
      End If
    a:
    rsadm!rollno = txtrollno.Text
    rsadm!moduleno = txtmodule.Text
    rsadm!course = txtcourse.Text
    rsadm!batch = txtbatch.Text
      'rsadm!rollno = txtrollno.Text
    rsadm!timingfrom = txtfrom.Text
    rsadm!timingto = txtto.Text
    rsadm!centre = txtcentre1.Text
    rsadm!batchstartdate = dtstartdate.Value
    rsadm!surname = Trim(txtsurname.Text)
    rsadm!fname = Trim(txtfname.Text)
    rsadm!mname = Trim(txtmname.Text)
    rsadm!address = txtaddress.Text
    rsadm!area = txtarea.Text
    rsadm!city = txtcity.Text
    rsadm!pincode = txtpin.Text
    rsadm!rescontactno = txtresphone.Text
    rsadm!offcontactno = txtoffphone.Text
    rsadm!dob = txtdob.Text
    rsadm!degree = txtdegree.Text
    rsadm!school = txtschool.Text
    rsadm!board = txtboard.Text
    rsadm!yearofpassing = txtyear.Text
    rsadm!marks = txtdivision.Text
    rsadm!games = txtgames.Text
    rsadm!socialactivities = txtsocialact.Text
    rsadm!hobbies = txthobbies.Text
    rsadm!langspeak = txtspeak.Text
    rsadm!langread = txtread.Text
    rsadm!langwrite = txtwrite.Text
    End If
    If chklocalnews.Value = 1 Then
    rsadm!Source = "Local Newspaper"
    ElseIf chknationalnews.Value = 1 Then
    rsadm!Source = "National Newspaper"
```

```
ElseIf chkword.Value = 1 Then
rsadm!Source = "Word of Mouth"
ElseIf chkbanner.Value = 1 Then
rsadm!Source = "Banner"
ElseIf chkhoarding.Value = 1 Then
rsadm!Source = "Hoarding"
ElseIf chkpresentation.Value = 1 Then
rsadm!Source = "Presentation"
ElseIf chkposter.Value = 1 Then
rsadm!Source = "Poster"
ElseIf chkbus.Value = 1 Then
rsadm!Source = "Bus Shelter"
ElseIf chkinq.Value = 1 Then
rsadm!Source = "Inquiry Lab"
ElseIf chkdirect.Value = 1 Then
rsadm!Source = "Direct Mail"
ElseIf chkjetking.Value = 1 Then
rsadm!Source = "Jetking Student"
ElseIf chkother(1).Value = 1 Then
  If txtother.Text = "" Then
    MsgBox "Please write the information of the Source."
    rsadm.Requery
    rsreg.Requery
    Exit Sub
  Else
rsadm!Source = txtother.Text
  End If
Else
msg = MsgBox("Please select only one option from the + vbcrlf + Source options given",
vbCritical, "Selection Error")
  Exit Sub
End If

If optyes.Value = True Then
rsadm!moneyback = "Yes"
Else
rsadm!moneyback = "No"
End If

rsadm!place = txtplace.Text
rsadm!Date = txtdate.Text

rsreg!smo = txtserial.Text
rsreg!rollno = txtrollno.Text
rsreg!feecode = txtfeecode.Text
rsreg!feetype = txtfeetype.Text
```

```
rsreg!totalfee = txttotalfee.Text
rsreg!dp = txtdp.Text
rsreg!dprec = txtdprec.Text
rsreg!dpdate = txtdpdate.Text
rsreg!instamount = txtinstamt.Text
rsreg!instno = txtinstno.Text
rsreg!instdate = txtfstinstdate.Text
If txtlibrec.Text = "" Then
rsreg!libdeposit = "0"
rsreg!librec = "0"
rsreg!libdate = Date
Else
rsreg!libdeposit = txtlib.Text
rsreg!librec = txtlibrec.Text
rsreg!libdate = txtlibdate.Text
End If
rsreg!caution = txtcaution.Text
rsreg!cautionrec = txtcautionrec.Text
rsreg!cautiondate = txtcautiondate.Text
If txtmbrec.Text = "" Then
rsreg!mb = "0"
rsreg!mbrec = "0"
rsreg!mbdate = Date
Else
rsreg!mb = txtmb.Text
rsreg!mbrec = txtmbrec.Text
rsreg!mbdate = txtmbdate.Text
End If
rsreg!scheme = txtscheme.Text
rsreg!acceptedby = txtaccepted.Text
rsreg!approvedby = txtapproved.Text
rsreg!counsellor = txtcounsellor.Text
rsreg!regdate = txtregdate.Text
rsreg!centre = txtcentre.Text
rsreg!enquiryno = txtenqno.Text

rsadm!Update
rsreg.Update
MsgBox "Your record has been saved"
frareg.Visible = False
frasports.Visible = False
frapersonal.Visible = True
fraeducation.Visible = True

cmdnext.Enabled = False
cmdsave.Enabled = False
```

```
    cmdback.Enabled = False
    cmdedit.Enabled = True
    cmdadd.Enabled = True
    cmddel.Enabled = True
    If cmdcancel.Visible = True Then
      cmdcancel.Visible = False
      cmddel.Visible = True
      cmddel.Enabled = True
    End If

    'rsenq.Close
    'rsenq.Open "select * from enquiry", cn, adOpenDynamic, adLockOptimistic
    'rsadm.Close
    'rsadm.Open "select * from admission", cn, adOpenDynamic, adLockOptimistic
    'rsadm.Close
    'rsadm.Open "select * from admission_reg", cn, adOpenDynamic, adLockOptimistic

    rsenq.Requery
    rsadm.Requery
    rsreg.Requery

    rsadm.MoveFirst
    rsreg.MoveFirst
    Call display
    Call Form_Activate
    End Sub

    Private Sub Form_Activate()
    txtrollno.Enabled = False
    dtstartdate.Enabled = False
    txtmodule.Enabled = False
    txtbatch.Enabled = False
    txtcentre.Enabled = False
    txtcentre1.Enabled = False
    txtcourse.Enabled = False
    txtfrom.Enabled = False
    txtto.Enabled = False
    frapersonal.Enabled = False
    fraeducation.Enabled = False
    End Sub

    Private Sub Form_Load()
    jetking.Enabled = False

    Set cn = New ADODB.Connection
```

```
cn.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\institute\cms.mdb;Persist Security Info=False"
cn.CursorLocation = adUseClient
cn.Open
Set rsadm = New ADODB.Recordset
Set rsint = New ADODB.Recordset
Set rsatd = New ADODB.Recordset
Set rslea = New ADODB.Recordset
Set rsfs = New ADODB.Recordset
Set rsfp = New ADODB.Recordset
Set rspla = New ADODB.Recordset

rspla.Open "select * from placement", cn, adOpenDynamic, adLockOptimistic
rsfs.Open "select * from fee_schedule", cn, adOpenDynamic, adLockOptimistic
rsfp.Open "select * from fee_paid", cn, adOpenDynamic, adLockOptimistic
rsint.Open "select * from interview", cn, adOpenDynamic, adLockOptimistic
rsatd.Open "select * from attendence", cn, adOpenDynamic, adLockOptimistic
rslea.Open "select * from leave", cn, adOpenKeyset, adLockOptimistic
rsadm.Open "select * from admission", cn, adOpenKeyset, adLockOptimistic
rsreg.Open "select * from admission_reg", cn, adOpenKeyset, adLockOptimistic
rsenq.Open "select * from enquiry", cn, adOpenKeyset, adLockOptimistic

txtserial.Enabled = False
frasports.Visible = False
frareg.Visible = False
frapersonal.Visible = True
fraeducation.Visible = True

cmdnext.Enabled = False
cmdback.Enabled = False
cmdsave.Enabled = False
cmdedit.Enabled = True
cmdadd.Enabled = True
txtbatch.Text = rsadm!batch
txtfrom.Text = rsadm!timingfrom
txtto.Text = rsadm!timingto
txtcentre1.Text = rsadm!centre
dtpstartdate.Value = rsadm!batchstartdate
txtsurname.Text = rsadm!surname
txtfname.Text = rsadm!fname
txtmname.Text = rsadm!mname
txtaddress.Text = rsadm!address
txtarea.Text = rsadm!area
txtcity.Text = rsadm!city
txtpin.Text = rsadm!pincode
txttelephone.Text = rsadm!residenceno
```

FORMS ABOUT:-

```
Option Explicit

'Reg Key Security Options...
Const READ_CONTROL = &H20000
Const KEY_QUERY_VALUE = &H1
Const KEY_SET_VALUE = &H2
Const KEY_CREATE_SUB_KEY = &H4
Const KEY_ENUMERATE_SUB_KEYS = &H8
Const KEY_NOTIFY = &H10
Const KEY_CREATE_LINK = &H20
Const KEY_ALL_ACCESS = KEY_QUERY_VALUE + KEY_SET_VALUE + _
                       KEY_CREATE_SUB_KEY + KEY_ENUMERATE_SUB_KEYS + _
                       KEY_NOTIFY + KEY_CREATE_LINK + READ_CONTROL

'Reg Key ROOT Types...
Const HKEY_LOCAL_MACHINE = &H80000002
Const ERROR_SUCCESS = 0
Const REG_SZ = 1                        ' Unicode nul terminated string
Const REG_DWORD = 4                     ' 32-bit number

Const gREGKEYSYSINFOLOC = "SOFTWARE\Microsoft\Shared Tools Location"
Const gREGVALSYSINFOLOC = "MSINFO"
Const gREGKEYSYSINFO = "SOFTWARE\Microsoft\Shared Tools\MSINFO"
Const gREGVALSYSINFO = "PATH"

Private Declare Function RegOpenKeyEx Lib "advapi32" Alias "RegOpenKeyExA" (ByVal
hKey As Long, ByVal lpSubKey As String, ByVal ulOptions As Long, ByVal samDesired As
Long, ByRef phkResult As Long) As Long
Private Declare Function RegQueryValueEx Lib "advapi32" Alias "RegQueryValueExA"
(ByVal hKey As Long, ByVal lpValueName As String, ByVal lpReserved As Long, ByRef
lpType As Long, ByVal lpData As String, ByRef lpcbData As Long) As Long
Private Declare Function RegCloseKey Lib "advapi32" (ByVal hKey As Long) As Long


Private Sub cmdSysInfo_Click()
    Call StartSysInfo
End Sub

Private Sub cmdOK_Click()
    Unload Me
End Sub

Private Sub Form_Load()
```

FORMS ABOUT:

```
Option Explicit

' Reg Key Security Options...
Const READ_CONTROL = &H20000
Const KEY_QUERY_VALUE = &H1
Const KEY_SET_VALUE = &H2
Const KEY_CREATE_SUB_KEY = &H4
Const KEY_ENUMERATE_SUB_KEYS = &H8
Const KEY_NOTIFY = &H10
Const KEY_CREATE_LINK = &H20
Const KEY_ALL_ACCESS = KEY_QUERY_VALUE + KEY_SET_VALUE + _
            KEY_CREATE_SUB_KEY + KEY_ENUMERATE_SUB_KEYS + _
            KEY_NOTIFY + KEY_CREATE_LINK + READ_CONTROL

' Reg Key ROOT Types...
Const HKEY_LOCAL_MACHINE = &H80000002
Const ERROR_SUCCESS = 0
Const REG_SZ = 1                    ' Unicode nul terminated string
Const REG_DWORD = 4                 ' 32-bit number

Const gREGKEYSYSINFOLOC = "SOFTWARE\Microsoft\Shared Tools Location"
Const gREGVALSYSINFOLOC = "MSINFO"
Const gREGKEYSYSINFO = "SOFTWARE\Microsoft\Shared Tools\MSINFO"
Const gREGVALSYSINFO = "PATH"

Private Declare Function RegOpenKeyEx Lib "advapi32" Alias "RegOpenKeyExA" (ByVal hKey As Long, ByVal lpSubKey As String, ByVal ulOptions As Long, ByVal samDesired As Long, ByRef phkResult As Long) As Long
Private Declare Function RegQueryValueEx Lib "advapi32" Alias "RegQueryValueExA" (ByVal hKey As Long, ByVal lpValueName As String, ByVal lpReserved As Long, ByRef lpType As Long, ByVal lpData As String, ByRef lpcbData As Long) As Long
Private Declare Function RegCloseKey Lib "advapi32" (ByVal hKey As Long) As Long


Private Sub cmdSysInfo_Click()
  Call StartSysInfo
End Sub

Private Sub cmdOK_Click()
  Unload Me
End Sub

Private Sub Form_Load()
```

```
    Me.caption = "About "& App.Title
    lblVersion.caption = "Version "& App.Major & "." & App.Minor & "." & App.Revision
End Sub

Public Sub StartSysInfo()
    On Error GoTo SysInfoErr

    Dim rc As Long
    Dim SysInfoPath As String

' Try To Get System Info Program Path\Name From Registry...
    If GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFO,
gREGVALSYSINFO, SysInfoPath) Then
' Try To Get System Info Program Path Only From Registry...
    ElseIf GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFOLOC,
gREGVALSYSINFOLOC, SysInfoPath) Then
' Validate Existance Of Known 32 Bit File Version
        If (Dir(SysInfoPath & "\MSINFO32.EXE") <> "") Then
            SysInfoPath = SysInfoPath & "\MSINFO32.EXE"

' Error - File Can Not Be Found...
        Else
            GoTo SysInfoErr
        End If
' Error - Registry Entry Can Not Be Found...
    Else
        GoTo SysInfoErr
    End If

    Call Shell(SysInfoPath, vbNormalFocus)

    Exit Sub
SysInfoErr:
    MsgBox "System Information Is Unavailable At This Time", vbOKOnly
End Sub

Public Function GetKeyValue(KeyRoot As Long, KeyName As String, SubKeyRef As
String, ByRet KeyVal As String) As Boolean
    Dim i As Long                           ' Loop Counter
    Dim rc As Long                          ' Return Code
    Dim hKey As Long                        ' Handle To An Open Registry Key
    Dim hDepth As Long                      '
    Dim KeyValType As Long                  ' Data Type Of A Registry Key
    Dim tmpVal As String                    ' Temporary Storage For A Registry Key Value
    Dim KeyValSize As Long                  ' Size Of Registry Key Variable
```

```
    '------------------------------------------------------------
' Open RegKey Under KeyRoot {HKEY_LOCAL_MACHINE...}
    '------------------------------------------------------------
rc = RegOpenKeyEx(KeyRoot, KeyName, 0, KEY_ALL_ACCESS, hKey) ' Open Registry
Key

    If (rc<> ERROR_SUCCESS) Then GoTo GetKeyError          ' Handle Error ..

tmpVal = String$(1024, 0)                    ' Allocate Variable Space
    KeyValSize = 1024                         ' Mark Variable Size

    '------------------------------------------------------------
' Retrieve Registry Key Value...
    '------------------------------------------------------------
rc = RegQueryValueEx(hKey, SubKeyRef, 0, _
                KeyValType, tmpVal, KeyValSize)    ' Get/Create Key Value

    If (rc<> ERROR_SUCCESS) Then GoTo GetKeyError          ' Handle Errors

    If (Asc(Mid(tmpVal, KeyValSize, 1)) = 0) Then          ' Win95 Adds Null Terminated
String...
tmpVal = Left(tmpVal, KeyValSize - 1)            ' Null Found, Extract From String
                                          ' WinNT Does NOT Null Terminate String...
    Else
tmpVal = Left(tmpVal, KeyValSize)                ' Null Not Found, Extract String Only
    End If
    '------------------------------------------------------------
' Determine Key Value Type For Conversion...
    '------------------------------------------------------------
    Select Case KeyValType                          ' Search Data Types...
    Case REG_SZ                                  ' String Registry Key Data Type
      KeyVal = tmpVal                            ' Copy String Value
    Case REG_DWORD                                  ' Double Word Registry Key Data Type
      For i = Len(tmpVal) To 1 Step -1              ' Convert Each Bit
        KeyVal = KeyVal + Hex(Asc(Mid(tmpVal, i, 1)))  ' Build Value Char. By Char.
      Next
      KeyVal = Format$("&h" + KeyVal)               ' Convert Double Word To String
    End Select

    GetKeyValue = True                              ' Return Success
rc = RegCloseKey(hKey)                           ' Close Registry Key
    Exit Function                                ' Exit

GetKeyError:      ' Cleanup After An Error Has Occured...
    KeyVal = ""                                  ' Set KeyVal To Empty String
    GetKeyValue = False                          ' Return Failure
rc = RegCloseKey(hKey)                           ' Close Registry Key
End Function
```

About MyApp

Application Title

Version

Institute Management System is an application which manages the Institute info. by keeping the details of Student. It manages student info. from the day He/She gets registered to the day they are placed in the company for the job at good post.

Warning: ...

OK

System Info...

FORMS FOR ADMISSION REPORT

```
Dim cn As New ADODB.Connection
Dim rsadm As New ADODB.Recordset

Private Sub formatlabel(lblx As RptLabel, caption)
With lblx
   .caption = caption
   .CanGrow = True
End With
End Sub

Private Sub cmdcancel_Click()
Unload Me
jetking.Enabled = True
jetking.Visible = True
End Sub

Private Sub cmdOK_Click()
If cmbtype.Text = "" Then
   MsgBox "Please select the type from the list."
   cmbtype.SetFocus
   Exit Sub
End If

admissionreport.Show

If opt = 1 Then
   rsadm.Close
   rsadm.Open "select * from admission where area='" & cmbtype.Text & "'", cn,
adOpenDynamic, adLockOptimistic
   While Not rsadm.EOF = True
cnt = cnt + 1
      rsadm.MoveNext
   Wend
formatlabel admissionreport.Sections(5).Controls(1), _
   "Total Admissions " + optarea.caption + vbCrLf + "(" + cmbtype.Text + ")"
End If
If opt = 2 Then
   rsadm.Close
   rsadm.Open "select * from admission where degree='" & cmbtype.Text & "'", cn,
adOpenDynamic, adLockOptimistic
```

```
cnt = cnt + 1
      rsadm.MoveNext
   Wend
formatlabel admissionreport.Sections(5).Controls(1), _
   "Total Admissions " + optedu.caption + vbCrLf + "(" + cmbtype.Text + ")"
End If
If opt = 3 Then
   rsadm.Close
   rsadm.Open "select * from admission where moneyback='" & cmbtype.Text & "'", cn,
adOpenDynamic, adLockOptimistic
   While Not rsadm.EOF = True
cnt = cnt + 1
      rsadm.MoveNext
   Wend
formatlabel admissionreport.Sections(5).Controls(1), _
   "Total Admissions " + optmb.caption + vbCrLf + "(" + cmbtype.Text + ")"
End If
If opt = 4 Then
   rsadm.Close
   rsadm.Open "select * from admission where source='" & cmbtype.Text & "'", cn,
adOpenDynamic, adLockOptimistic
   While Not rsadm.EOF = True
cnt = cnt + 1
      rsadm.MoveNext
   Wend
formatlabel admissionreport.Sections(5).Controls(1), _
   "Total Admissions " + optsource.caption + vbCrLf + "(" + cmbtype.Text + ")"
End If

formatlabel admissionreport.Sections(5).Controls(2), _
cnt
End Sub

Private Sub Form_Load()
opt = 0
cnt = 0
jetkimg.Enabled = False
Set cn = New ADODB.Connection


cn.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\institute\cms1.mdb;Persist Security Info=False"
cn.CursorLocation = adUseClient
cn.Open
Set rsadm = New ADODB.Recordset
```

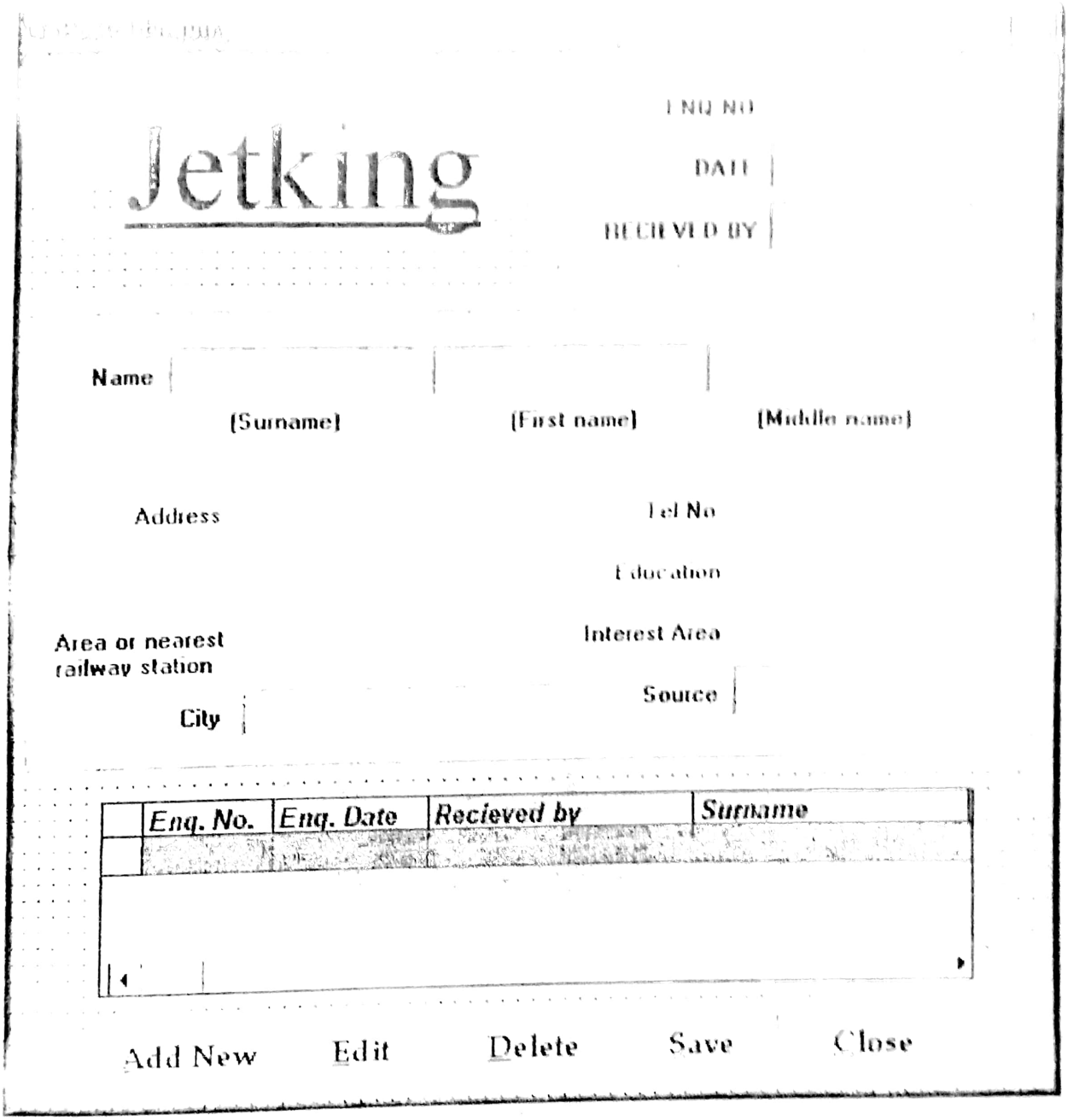
```
rsadm.Open "select * from admission", cn, adOpenDynamic, adLockOptimistic

cmbtype.clear

End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
Unload Me
jetking.Enabled = True
jetking.Visible = True
End Sub

Private Sub optarea_Click()
opt = 1
If optarea.Value = True Then
   rsadm.Close
   rsadm.Open "select distinct area from admission", cn, adOpenDynamic, adLockOptimistic
   cmbtype.clear
   While Not rsadm.EOF = True
cmbtype.AddItem rsadm!area
      rsadm.MoveNext
   Wend
   'formatlabel admissionreport.Sections(5).Controls(1), _
   ""Total Admissions " + optarea.caption
   cmbtype.SetFocus
 Else
   rsadm.Requery
 End If
 End Sub

 Private Sub optedu_Click()
 opt = 2
 If optedu.Value = True Then
    rsadm.Close
    rsadm.Open "select distinct degree from admission", cn, adOpenDynamic,
 adLockOptimistic
    cmbtype.clear
    While Not rsadm.EOF = True
 cmbtype.AddItem rsadm!degree
      rsadm.MoveNext
    Wend
    cmbtype.SetFocus
    'formatlabel admissionreport.Sections(2).Controls(1), _
    ""Total Admissions " + optedu.caption
```

```
Else
  rsadm.Requery
End If
End Sub

Private Sub optmb_Click()
opt = 3
If optmb.Value = True Then
  rsadm.Close
  rsadm.Open "select distinct moneyback from admission", cn, adOpenDynamic,
adLockOptimistic
  cmbtype.clear
  While Not rsadm.EOF = True
cmbtype.AddItem rsadm!moneyback
    rsadm.MoveNext
  Wend
  'formatlabel admissionreport.Sections(2).Controls(1), _
  '"Total Admissions " + optint.caption
  cmbtype.SetFocus
Else
  rsadm.Requery
End If
End Sub

Private Sub optsource_Click()
opt = 4
If optsource.Value = True Then
  rsadm.Close
  rsadm.Open "select distinct source from admission", cn, adOpenDynamic,
adLockOptimistic
  cmbtype.clear
  While Not rsadm.EOF = True
cmbtype.AddItem rsadm!Source
    rsadm.MoveNext
  Wend
  'formatlabel admissionreport.Sections(2).Controls(1),
  '"Total Admissions " + optsource.caption
  cmbtype.SetFocus
Else
  rsadm.Requery
End If
End Sub
```

FORMS FOR ATTENDANCE

```
Dim cn As New ADODB.Connection
Dim rsadm As New ADODB.Recordset
Dim rsatd As New ADODB.Recordset
Dim rslea As New ADODB.Recordset

Private Sub cmdadd_Click()
Call clear
txtrollno.Enabled = True
If frmattend.caption = "Leave" Then
   fraattend.Visible = False
   fraleave.Visible = True
   fraleave.Enabled = True
Else
   fraattend.Visible = True
   fraleave.Visible = False
   fraattend.Enabled = True
End If

txtrollno.SetFocus
cmddel.Enabled = False
cmdedit.Enabled = False
cmdadd.Enabled = False
cmdsave.Enabled = True
addnew = 1
End Sub

Private Sub cmdcancel_Click()
If frmattend.caption = "Leave" Then
   rslea.MoveFirst
   Call display
   Call Form_Activate
Else
   rsatd.MoveFirst
   Call display
   Call Form_Activate
End If
cmdcancel.Visible = False
cmddel.Visible = True
End Sub

Private Sub cmdclose_Click()
Unload Me
jetking.Enabled = True
jetking.Visible = True
```

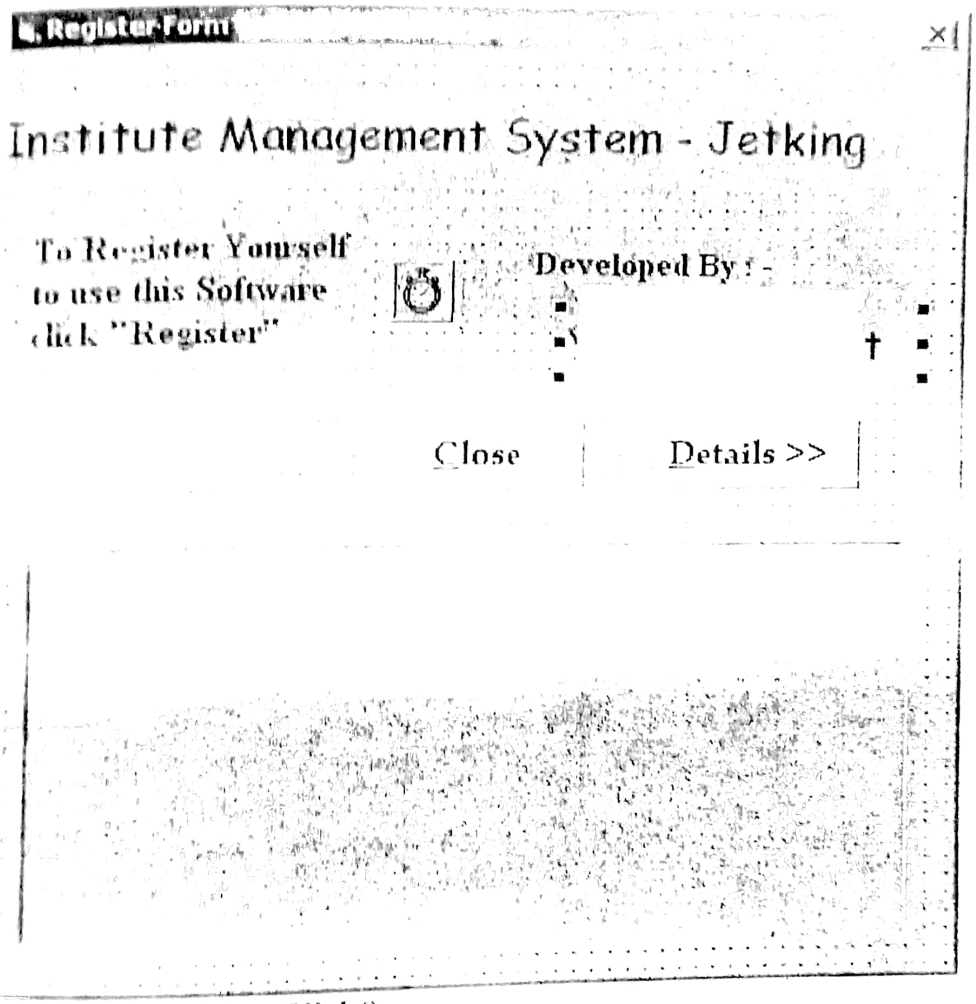
```
End Sub
Private Sub clear()
txtrollno.Text = ""
txtcourse.Text = ""
txtbatchno.Text = ""
txtsurname.Text = ""
txtfname.Text = ""
txtmname.Text = ""

If frmattend.caption = "Leave" Then
   txtlmodule.Text = ""
   dtfrom.Value = Date
   dtto.Value = Date
   txtdays.Text = ""
   txtreason.Text = ""
   txtupdation.Text = ""
Else
   txtmodule.Text = ""
   dtstdate.Value = Date
   txtfaculty.Text = ""
   dtenddate.Value = Date
   txtattend.Text = ""
   txttotaldays.Text = ""
   txtperformance.Text = ""
   txtremarks.Text = ""
End If
End Sub

Private Sub display()
txtrollno.Text = rsadm!rollno
txtcourse.Text = rsadm!course
txtbatchno.Text = rsadm!batch
txtsurname.Text = rsadm!surname
txtfname.Text = rsadm!fname
txtmname.Text = rsadm!mname
```




```
If frmattend.caption = "Leave" Then
   txtmodule.Text = rslea!module
   dtfrom.Value = rslea!From
   dtto.Value = rslea!To
   txtdays.Text = rslea!days
   txtreason.Text = rslea!reason
   txtupdation.Text = rslea!updation
Else
   txtmodule.Text = rsatd!module
   dtstdate.Value = rsatd!stdate
   txtfaculty.Text = rsatd!faculty
   dtenddate.Value = rsatd!enddate
   txtattend.Text = rsatd!attend
   txttotaldays.Text = rsatd!totaldays
```

```
    txtperformance.Text = rsatd!performance
    txtremarks.Text = rsatd!remarks
End If
End Sub

'Private Sub cmddel_Click()
'edfname = InputBox("Enter the Student's First Name ", "First Name")
'edsurname = InputBox("Enter the Student's Last Name ", "Last Name")
'rsadm.Close
'rsadm.Open "select * from admission where fname='" &UCase(edfname) & "'and surname='"
& UCase(edsurname) & "'", cn, adOpenDynamic, adLockOptimistic
'rollno = rsadm!rollno
'If frmattend.caption = "Leave" Then
'   rslea.Close
'   rslea.Open "select * from leave where rollno = " & rollno & "", cn, adOpenDynamic,
adLockOptimistic
'   If rsadm.BOF = True And rsadm.EOF = True Then
'      MsgBox "Search not found."
'      Exit Sub
'   Else
'      Call display
'      cmdsave.Enabled = False
'      cmddel.Enabled = True
'      cmdedit.Enabled = False
'      cmdadd.Enabled = False
'      cmdnext.Enabled = True
'      msg = MsgBox("Are you sure you want to delete the Record of " + vbCrLf + "
'" & txtfname.Text & " " & txtmname.Text & " " & txtsurname.Text & "'", vbYesNo +
vbQuestion, "Delete")
'      If msg = vbYes Then
'         msg = MsgBox("If you delete the Record of'" & txtfname.Text & " " &
txtmname.Text & " " & txtsurname.Text & "'," + vbCrLf + "then all the Records of'" &
txtfname.Text & "' will be deleted." + vbCrLf + vbCrLf + "To Delete select 'OK' else select
'CANCEL' to Exit", vbQuestion + vbOKCancel, "Delete")
'         If msg = vbYes Then
'            rollno = rsadm!rollno
'            rsadm.Delete
'            rsreg.Requery
'            While Not rsreg.EOF = True
'               If rsreg!rollno = rollno Then
'                  rsreg.Delete
'               End If
'               rsreg.MoveNext
'            Wend
'            rsatd.Requery
'            While Not rsatd.EOF = True
```

cmdedit.Enabled = True
cmddel.Enabled = True
cmdadv.Enabled = True
cmdback.Enabled = False
End Sub



Private Sub cmdedit_Click()
On Error Resume Next
Dim edrollno As Integer
edrollno = InputBox("Enter the Student's Roll No ", "Roll No.")

cmddel.Visible = False
cmdadv.Visible = True
If frmattend.caption = "Leave" Then
rsleave.Close
rsleave.Open "select * from leave where rollno = " & edrollno & "", cn, adOpenDynamic, adLockOptimistic
If rsleave.BOF = True And rsleave.EOF = True Then
MsgBox "Search not found."
Exit Sub
Else
Call display

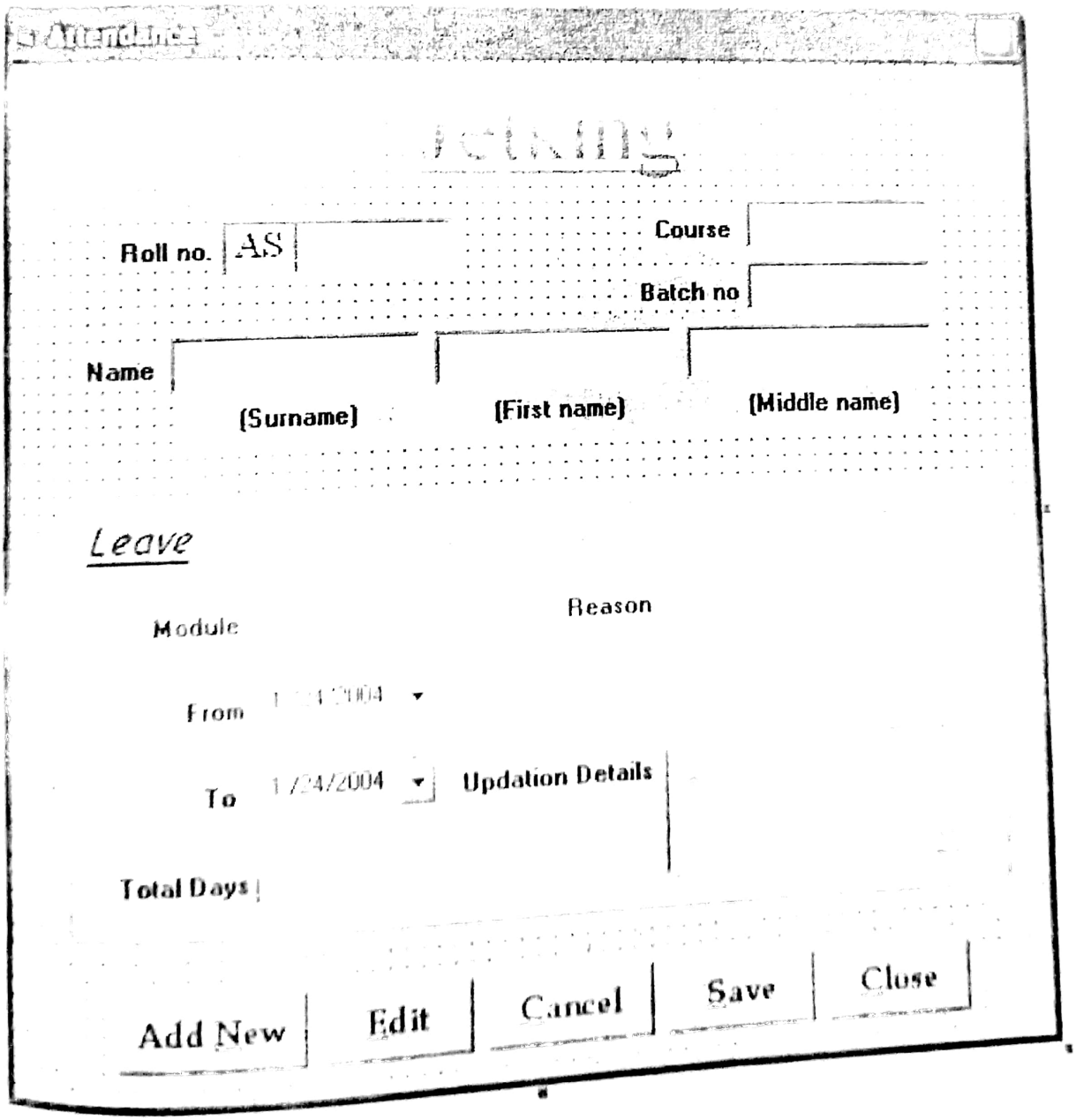
```
        txtrollno.Enabled = True
        fraattend.Visible = False
        fraleave.Enabled = True
        cmddel.Enabled = False
        cmdsave.Enabled = True
        cmdedit.Enabled = False
        cmdadd.Enabled = False
edit = 1
   End If
Else
   rsatd.Close
   rsatd.Open "select * from attendence where rollno = " & edrollno & "", cn,
adOpenDynamic, adLockOptimistic
   If rsatd.BOF = True And rsatd.EOF = True Then
      MsgBox "Search not found."
      Exit Sub
   Else
      Call display
      txtrollno.Enabled = True
      fraleave.Visible = False
      fraattend.Enabled = True
      cmddel.Enabled = False
      cmdsave.Enabled = True
      cmdedit.Enabled = False
      cmdadd.Enabled = False
edit = 1
   End If
End If
End Sub

Private Sub cmdsave_Click()
If frmattend.caption = "Leave" Then
   If txtlmodule.Text = "" Or txtdays.Text = "" Or txtreason.Text = "" Or txtupdation.Text =
"" Then
      MsgBox "You forgot to enter one of the required fields."
   Else
      If addnew = 1 Then
addnew = 0
         If rslea.BOF = True And rslea.EOF = True Then
            rslea.addnew
         Else
            rslea.MoveLast
            rslea.addnew
         End If
      ElseIf edit = 1 Then
edit = 0
```

```
        End If
    rslea!rollno = txtrollno.Text
    rslea!module = txtmodule.Text
    rslea!From = dtfrom.Value
    rslea!To = dtto.Value
    rslea!days = txtdays.Text
    rslea!reason = txtreason.Text
    rslea!updation = txtupdation.Text
        'rslea.Save
        rslea.Update
        MsgBox "Your record is saved."

        cmdadd.Enabled = True
        cmdedit.Enabled = True
        cmddel.Enabled = True
        cmdsave.Enabled = False

        fraleave.Enabled = False
        rslea.MoveFirst
        Call display
        Call Form_Activate
      End If
    Else
      If txtmodule.Text = "" Or txtfaculty.Text = "" Or _
        txtattend.Text = "" Or txtperformance.Text = "" Or txtremarks.Text = "" Then
          MsgBox "You forgot to enter one of the required fields."
      Else
        If addnew = 1 Then
    addnew = 0
          If rsatd.BOF = True And rsatd.EOF = True Then
            rsatd.addnew
          Else
            rsatd.MoveLast
            rsatd.addnew
          End If
        ElseIf edit = 1 Then
    edit = 0
        End If
    rsatd!rollno = txtrollno.Text
    rsatd!module = txtmodule.Text
    rsatd!stdate = dtstdate.Value
    rsatd!faculty = txtfaculty.Text
    rsatd!enddate = dtenddate.Value
    rsatd!attend = txtattend.Text
    rsatd!totaldays = txttotaldays.Text
    rsatd!performance = txtperformance.Text
```

```
rsatd!remarks = txtremarks.Text
        'rsfca.Save
        rsatd.Update
        MsgBox "Your record is saved "

        cmdadd.Enabled = True
        cmdedit.Enabled = True
        cmddel.Enabled = True
        cmdsave.Enabled = False

        fraattend.Enabled = False
        rsatd.MoveFirst
        Call display
        Call Form_Activate
    End If
End If
End Sub

Private Sub dtenddate_LostFocus()
txttotaldays.Text = DateDiff("d", dtstdate.Value, dtenddate.Value)
End Sub

Private Sub dtto_LostFocus()
txtdays.Text = DateDiff("d", dtfrom.Value, dtto.Value)
End Sub

Private Sub Form_Activate()
txtrollno.Enabled = False
fraattend.Enabled = False
fraleave.Enabled = False

txtcourse.Enabled = False
txtbatchno.Enabled = False
txtsurname.Enabled = False
txtfname.Enabled = False
txtmname.Enabled = False

cmdadd.Enabled = True
cmddel.Enabled = True
cmdedit.Enabled = False
cmdsave.Enabled = False
End Sub

Private Sub Form_Load()
Dking.Enabled = False
Set cn = New ADODB.Connection
```

```
cn.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\institute\emst.mdb;Persist Security Info=False"
cn.CursorLocation = adUseClient
cn.Open

txtrollno.Enabled = False

Set rsadm = New ADODB.Recordset
Set rsatd = New ADODB.Recordset
Set rslea = New ADODB.Recordset

rsatd.Open "select * from attendence", cn, adOpenDynamic, adLockOptimistic
rsadm.Open "select * from admission", cn, adOpenKeyset, adLockOptimistic
rslea.Open "select * from leave", cn, adOpenKeyset, adLockOptimistic

txttotaldays.Enabled = False
txtdays.Enabled = False
End Sub

Private Sub Form_Paint()
If frmattend.caption = "Leave" Then
  If rslea.BOF = True And rslea.EOF = True Then
    GoTo a
  Else
    rslea.MoveFirst
                rsatd.Requery

                cmdedit.Enabled = True
                Exit Sub
              End If
              rsatd.MoveNext
            Wend
            If rsatd.EOF = True Then
              MsgBox "Attendence sheet of '" & txtfname.Text & txtsurname.Text & "' has not
been updated."
              rsatd.Requery
              Exit Sub
            End If
        End If
        Exit Sub
  End If
  rsadm.MoveNext
Wend
If rsadm.EOF = True Then
  MsgBox "Please enter correct Roll No."
  txtcourse.Text = ""
```

```
    txtbatchno.Text = ""
    txtsurname.Text = ""
    txtfname.Text = ""
    txtmname.Text = ""
    txtrollno.SetFocus
    txtrollno.Text = ""
    rsadm.MoveFirst
End If
End Sub
```



FORMS FOR FEE SHEDULE

```
Dim rs.adm As New ADODB.Recordset
Dim rsfs As New ADODB.Recordset
Dim rsfp As New ADODB.Recordset
Dim rsreg As New ADODB.Recordset
Dim refs As New ADODB.Recordset
Dim cn As New ADODB.Connection
Dim total As Double
Dim add As Integer
Dim edit As Integer

Private Sub cmdadd_Click()
Call clear
rsfs.Requery
txtrollno.Enabled = True
frafee.Enabled = True
txtrollno.SetFocus
cmddel.Enabled = False
cmdadd.Enabled = False
cmdedit.Enabled = False
cmdsave.Enabled = True
add = 1
End Sub

Private Sub cmdcancel_Click()
Call display
Call Form_Activate
cmddel.Visible = True
cmdcancel.Visible = False
cmdadd.Enabled = True
cmddel.Enabled = True
cmdedit.Enabled = True
cmdsave.Enabled = False
End Sub

Private Sub cmdclose_Click()
Unload Me
jetking.Enabled = True
jetking.Visible = True
End Sub

Private Sub cmddel_Click()
rdrollno = InputBox("Enter the Student's Roll No.", "Roll No.")
rs.Close
rs.Open "Select * from feesched.fs where rollno = " & (rdrollno) & "", cn
rs.OpenDynamic, adLockOptimistic
```

```
rsfs.BOF = True And rsfs.EOF = True Then
  MsgBox "Search not found"
  Exit Sub
Else
  Call display
  txtrollno.Enabled = True
  frafee.Enabled = True
msg = MsgBox("Are you sure you want to delete the Record of " + vbCrLf + "          " &
txtfname.Text & " " & txtmname.Text & " " & txtsurname.Text & "", vbYesNo +
vbQuestion, "Delete")
  If msg = vbYes Then
    rsfs.Delete
    rsfs.Close
    rsfs.Open "select * from feeschedule", cn, adOpenDynamic, adLockOptimistic
    rsfs.Requery
    If rsfs.BOF = True And rsfs.EOF = True Then
      MsgBox "Currently there are no Records."
    ElseIf rsfs.EOF = True Then
      rsfs.MovePrevious
      Call display
    ElseIf rsfs.BOF = True Then
      rsgfs.MoveNext
      Call display
    Else
      rsfs.MoveFirst
      Call display
    End If
    MsgBox "Your Record has been deleted."
  End If
End If
Call Form_Activate
cmdsave.Enabled = False
cmddel.Enabled = True
cmdedit.Enabled = True
cmdadd.Enabled = True
End Sub

Private Sub cmdedit_Click()
Dim edrollno As Integer
edrollno = InputBox("Enter the Student's Roll No.", "Roll No.")

rsfs.Close
rsfs.Open "select * from feeschedule where rollno = " & edrollno & "", cn, adOpenDynamic,
adLockOptimistic
If rsfs.BOF = True And rsfs.EOF = True Then
  MsgBox "Search not found"
```

```
    Exit Sub
Else
    Call Form_
    txtrollno.Enabled = True
    frafee.Enabled = True
    cmddel.Enabled = False
    cmdsave.Enabled = True
    cmdedit.Enabled = False
    cmdadd.Enabled = False
    cmddel.Visible = False
    cmdcancel.Visible = True
edit = 1
End If
End Sub
Private Sub cmdsave_Click()

If txtamt1.Text = "" Then
    MsgBox ("Please enter the correct information.")
    Exit Sub
End If

total = Val(txtamt1.Text) + Val(txtamt2.Text) + Val(txtamt3.Text) _
    + Val(txtamt4.Text) + Val(txtamt5.Text) + Val(txtamt6.Text) _
    + Val(txtamt7.Text) + Val(txtamt8.Text) + Val(txtamt9.Text) _
    + Val(txtamt10.Text) + Val(txtamt11.Text) + Val(txtamt12.Text) _
    + Val(txtamt13.Text) + Val(txtamt14.Text)
rsreg.Close
rsreg.Open "select * from Admission_Reg", cn, adOpenDynamic, adLockOptimistic

While Not rsreg.EOF = True
  If rsreg!rollno = txtrollno.Text Then
     If rsreg!totalfee> total Then
msg = MsgBox(" The total fee of the Student is given as '" & rsreg!totalfee & "'" + vbCrLf +
"and the amount you entered differs by '" & Val(rsreg!totalfee - total) & "'." + vbCrLf +
vbCrLf + "Select 'OK' to correct it and 'CANCEL' to Exit.", vbCritical + vbOKCancel,
"Logical Error")
        If msg = vbOK Then
          Exit Sub
        Else
          rsfs.MoveFirst
          Call display
          Call Form_Activate
          Exit Sub
        End If
    ElseIf rsreg!totalfee< total Then
```

[illegible]

```
        [illegible] Then
            [illegible]
        [illegible]
            rds.MoveFirst
            Call display
            Call Form_Activate
            Exit Sub
        End If
      End If
    End If
    rsreg.MoveNext
Wend
[illegible] Then
    [illegible]
End If
[illegible]

rds!rollno = txtrollno.Text
rds!course = txtcourse.Text
rds!batchno = txtbatchno.Text
rds!date1 = dtdate1.Value
rds!amt1 = txtamt1.Text
If txtamt2.Text = "" Then
rds!date2 = Date
rds!amt2 = "0"
Else
rds!date2 = dtdate2.Value
rds!amt2 = txtamt2.Text
[illegible]
[illegible] Text
[illegible] Date
[illegible] "0"
[illegible]
[illegible] Value
[illegible] Text
[illegible]
[illegible] Text = "" Then
[illegible] Date
[illegible] "0"
[illegible]
[illegible] Value
[illegible] Text
```

```
End If
If txtamt5.Text = "" Then
rsfs!date5 = Date
rsfs!amt5 = "0"
Else
rsfs!date5 = dtdate5.Value
rsfs!amt5 = txtamt5.Text
End If
If txtamt6.Text = "" Then
rsfs!date6 = Date
rsfs!amt6 = "0"
Else
rsfs!date6 = dtdate6.Value
rsfs!amt6 = txtamt6.Text
End If
If txtamt7.Text = "" Then
rsfs!date7 = Date
rsfs!amt7 = "0"
Else
rsfs!date7 = dtdate7.Value
rsfs!amt7 = txtamt7.Text
End If
If txtamt8.Text = "" Then
rsfs!date8 = Date
rsfs!amt8 = "0"
Else
rsfs!date8 = dtdate8.Value
rsfs!amt8 = txtamt8.Text
End If
If txtamt9.Text = "" Then
rsfs!date9 = Date
rsfs!amt9 = "0"
Else
rsfs!date9 = dtdate9.Value
rsfs!amt9 = txtamt9.Text
End If
If txtamt10.Text = "" Then
rsfs!date10 = Date
rsfs!amt10 = "0"
Else
rsfs!date10 = dtdate10.Value
rsfs!amt10 = txtamt10.Text
End If
If txtamt11.Text = "" Then
rsfs!date11 = Date
rsfs!amt11 = "0"
```

```
        Else
        rsfs!date11 = dtdate11.Value
        rsfs!amt11 = txtamt11.Text
        End If
        If txtamt12.Text = "" Then
        rsfs!date12 = Date
        rsfs!amt12 = "0"
        Else
        rsfs!date12 = dtdate12.Value
        rsfs!amt12 = txtamt12.Text
        End If
        If txtamt13.Text = "" Then
        rsfs!date13 = Date
        rsfs!amt13 = "0"
        Else
        rsfs!date13 = dtdate13.Value
        rsfs!amt13 = txtamt13.Text
        End If
        If txtamt14.Text = "" Then
        rsfs!date14 = Date
        rsfs!amt14 = "0"
        Else
        rsfs!date14 = dtdate14.Value
        rsfs!amt14 = txtamt14.Text
        End If
        rsfs.Update
        rsfs.Requery
        If txtamt1.Text <> "" Then
          refs.addnew
        refs!rollno = txtrollno.Text
        refs!course = txtcourse.Text
        refs!batchno = txtbatchno.Text
        refs!fshdate = dtdate1.Value
        refs!fshamt = txtamt1.Text
        refs!fshpaid = txtamt1.Text
        refs!fshleft = "0"
        End If
        If txtamt2.Text <> "" Then
          refs.addnew
        refs!rollno = txtrollno.Text
        refs!course = txtcourse.Text
        refs!batchno = txtbatchno.Text
        refs!fshdate = dtdate2.Value
        refs!fshamt = txtamt2.Text
        refs!fshpaid = "0"
        refs!fshleft = txtamt2.Text
```

```
End If
If txtamt3.Text <> "" Then
  refs.addnew
refs!rollno = txtrollno.Text
refs!course = txtcourse.Text
refs!batchno = txtbatchno.Text
refs!fshdate = dtdate3.Value
refs!fshamt = txtamt3.Text
refs!fshpaid = "0"
refs!fshleft = txtamt3.Text
End If
If txtamt4.Text <> "" Then
  refs.addnew
refs!rollno = txtrollno.Text
refs!course = txtcourse.Text
refs!batchno = txtbatchno.Text
refs!fshdate = dtdate4.Value
refs!fshamt = txtamt4.Text
refs!fshpaid = "0"
refs!fshleft = txtamt4.Text
End If
If txtamt5.Text <> "" Then
  refs.addnew
refs!rollno = txtrollno.Text
refs!course = txtcourse.Text
refs!batchno = txtbatchno.Text
refs!fshdate = dtdate5.Value
refs!fshamt = txtamt5.Text
refs!fshpaid = "0"
refs!fshleft = txtamt5.Text
End If
If txtamt6.Text <> "" Then
  refs.addnew
refs!rollno = txtrollno.Text
refs!course = txtcourse.Text
refs!batchno = txtbatchno.Text
refs!fshdate = dtdate6.Value
refs!fshamt = txtamt6.Text
refs!fshpaid = "0"
refs!fshleft = txtamt6.Text
End If
If txtamt7.Text <> "" Then
  refs.addnew
refs!rollno = txtrollno.Text
refs!course = txtcourse.Text
refs!batchno = txtbatchno.Text
```

```
    add = 0
End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
Unload Me
jetking.Enabled = True
jetking.Visible = True
End Sub
Private Sub display()
txtrollno.Text = rsfs!rollno
txtcourse.Text = rsadm!course
txtbatchno.Text = rsadm!batch
txtsurname.Text = rsadm!surname
txtfname.Text = rsadm!fname
txtmname.Text = rsadm!mname

dtdate1.Value = rsfs!date1
dtdate2.Value = rsfs!date2
dtdate3.Value = rsfs!date3
dtdate4.Value = rsfs!date4
dtdate5.Value = rsfs!date5
dtdate6.Value = rsfs!date6
dtdate7.Value = rsfs!date7
dtdate8.Value = rsfs!date8
dtdate9.Value = rsfs!date9
dtdate10.Value = rsfs!date10
dtdate11.Value = rsfs!date11
dtdate12.Value = rsfs!date12
dtdate13.Value = rsfs!date13
dtdate14.Value = rsfs!date14

If rsfs!amt1 = "0" Then
  txtamt1.Text = ""
Else
  txtamt1.Text = rsfs!amt1
End If
If rsfs!amt2 = "0" Then
  txtamt2.Text = ""
Else
  txtamt2.Text = rsfs!amt2
End If
If rsfs!amt3 = "0" Then
  txtamt3.Text = ""
Else
```

```
End If
End Sub

Private Sub clear()
txtrollno.Text = ""
txtcourse.Text = ""
txtbatchno.Text = ""
txtsurname.Text = ""
txtfname.Text = ""
txtmname.Text = ""

txtdate1.Value = Date
txtdate2.Value = Date
txtdate3.Value = Date
txtdate4.Value = Date
txtdate5.Value = Date
txtdate6.Value = Date
txtdate7.Value = Date
txtdate8.Value = Date
txtdate9.Value = Date
txtdate10.Value = Date
txtdate11.Value = Date
txtdate12.Value = Date
txtdate13.Value = Date
txtdate14.Value = Date

txtamt1.Text = ""
txtamt2.Text = ""
txtamt3.Text = ""
txtamt4.Text = ""
txtamt5.Text = ""
txtamt6.Text = ""
txtamt7.Text = ""
txtamt8.Text = ""
txtamt9.Text = ""
```

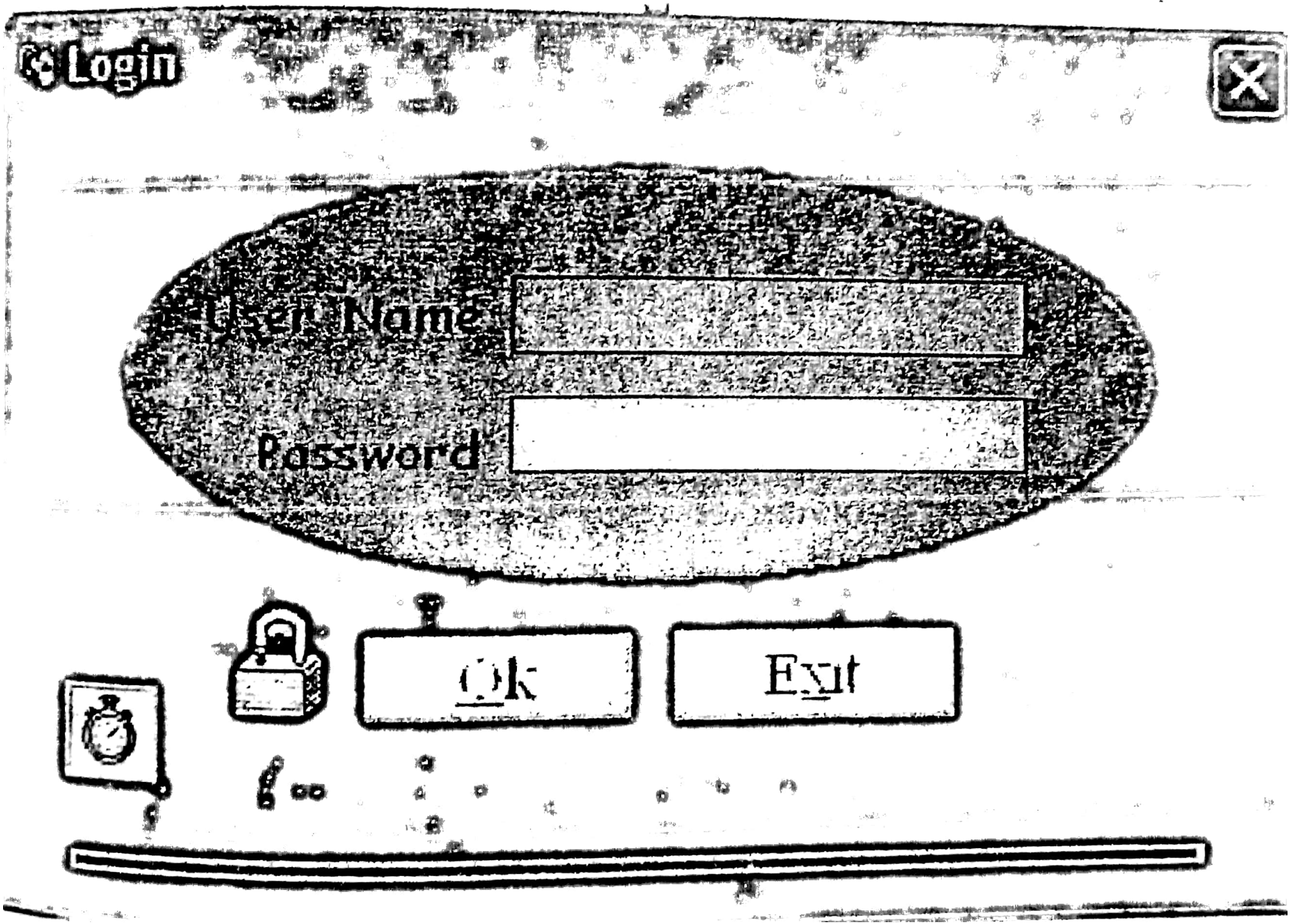
```
    txtamt11.Text = ""
        KeyAscii = KeyAscii
        Exit Sub
    ElseIf KeyAscii = 13 Then
      dtdate11.SetFocus
    End If
    MsgBox "Please enter only Numbers."
    KeyAscii = 0
  Else
    KeyAscii = KeyAscii
  End If
End Sub
Private Sub txtamt11_KeyPress(KeyAscii As Integer)
If KeyAscii < 48 Or KeyAscii > 57 Or KeyAscii = 8 Or KeyAscii = 13 Then
  If KeyAscii = 8 Then
    KeyAscii = KeyAscii
    Exit Sub
  ElseIf KeyAscii = 13 Then
    dtdate12.SetFocus
  End If
  MsgBox "Please enter only Numbers."
  KeyAscii = 0
Else
  KeyAscii = KeyAscii
End If
End Sub

Private Sub txtamt12_KeyPress(KeyAscii As Integer)
If KeyAscii < 48 Or KeyAscii > 57 Or KeyAscii = 8 Or KeyAscii = 13 Then
  If KeyAscii = 8 Then
    KeyAscii = KeyAscii
    Exit Sub
  ElseIf KeyAscii = 13 Then
    dtdate13.SetFocus
  End If
  MsgBox "Please enter only Numbers."
  KeyAscii = 0
Else
  KeyAscii = KeyAscii
End If
End Sub
Private Sub txtamt13_KeyPress(KeyAscii As Integer)
If KeyAscii < 48 Or KeyAscii > 57 Or KeyAscii = 8 Or KeyAscii = 13 Then
  If KeyAscii = 8 Then
    KeyAscii = KeyAscii
    Exit Sub
```

```
    ElseIf KeyAscii = 13 Then
      dtdate14.SetFocus
    End If
    MsgBox "Please enter only Numbers."
    KeyAscii = 0
  Else
    KeyAscii = KeyAscii
  End If
  End Sub
  Private Sub txtamt14_KeyPress(KeyAscii As Integer)
  If KeyAscii < 48 Or KeyAscii > 57 Or KeyAscii = 8 Or KeyAscii = 13 Then
    If KeyAscii = 8 Then
      KeyAscii = KeyAscii
      Exit Sub
    ElseIf KeyAscii = 13 Then
      cmdsave.SetFocus
    End If
    MsgBox "Please enter only Numbers."
    KeyAscii = 0
  Else
    KeyAscii = KeyAscii

  End If
  End Sub

  Private Sub txtrollno_LostFocus()
  rsfs.Requery
  rsfp.Requery
  While Not rsfp.EOF = True
    If rsfp!rollno = txtrollno.Text Then
      If rsfp!totalfee = rsfp!feepaid Then
        rsadm.Requery
        While Not rsadm.EOF = True
          If rsadm!rollno = txtrollno.Text Then
msg = MsgBox("" & rsadm!fname & " " & rsadm!surname & "' has paid the full fee.",
vbInformation, "Fees Payment")
          End If
          rsadm.MoveNext
        Wend
        rsadm.Requery
        rsfp.Requery
        rsfs.MoveFirst
        Call display
        Call Form_Activate
        cmdadd.Enabled = True
        cmdedit.Enabled = True
```

```
            cmddel.Enabled = True

               txtamt2.Text = rsreg!instamount
               dtdate2.SetFocus
               GoTo a
            End If
            rsreg.MoveNext
         Wend

a:
         rsadm.Requery
         Exit Sub
      End If
      rsadm.MoveNext
   Wend
   If rsadm.EOF = True Then
      MsgBox "Please enter correct Roll No."
      txtcourse.Text = ""
      txtbatchno.Text = ""
      txtsurname.Text = ""
      txtfname.Text = ""
      txtmname.Text = ""
      txtrollno.SetFocus
      txtrollno.Text = ""
      rsadm.MoveFirst
   End If
End Sub
```

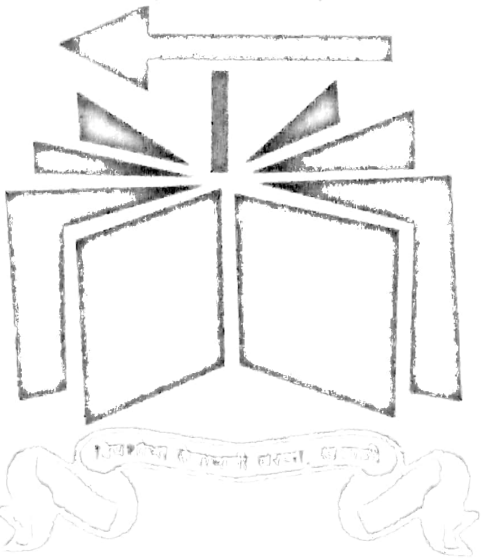
FORMS FOR LOGIN

```
Dim cn As New ADODB.Connection
Dim rslogin As New ADODB.Recordset

Private Sub cmdcancel_Click()
End
End Sub

Private Sub cmdOK_Click()
cmdOK.SetFocus
If rslogin!UserName = Trim(txtUserName.Text) And rslogin!Password =
Trim(txtPassword.Text) Then
  lblload.Visible = False
  pblogin.Visible = True
  Timer1.Enabled = True
  Timer2.Enabled = True
  lclose.Visible = False
  lopen.Visible = True
Else
  'msg = MsgBox("Please enter correct User Name and Password", vbCritical, "Login Error")
  frmLogin.Height = 3840
  lblload.caption = "Invalid Username or password."
  txtUserName.Text = ""
  txtPassword.Text = ""
  txtUserName.SetFocus
End If
End Sub

Private Sub Form_Load()
frmLogin.Height = 3195

Set cn = New ADODB.Connection
cn.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\institute\cms1.mdb;Persist Security Info=False"
cn.CursorLocation = adUseClient
cn.Open
Set rslogin = New ADODB.Recordset
rslogin.Open "select * from login", cn, adOpenKeyset, adLockOptimistic
rslogin.MoveFirst

lclose.Visible = True
lopen.Visible = False

Timer1.Enabled = False
Timer2.Enabled = False
```



DR.BABASAHEB AMBEDKAR MARATHWADA
UNIVERSITY, AURANGABAD

SAGAR COLLEGE, JALNA.
DEPARTMENT OF COMPUTER MANAGMENT
(2018-19)



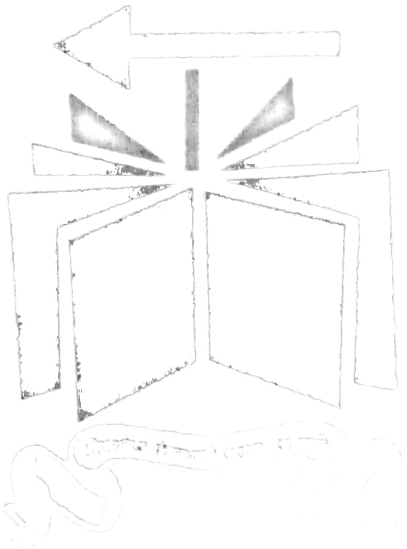
A
Project Report
On

[redacted]

Submitted By :
Mr. AVINASH ASHOK JADHAV

DR.BABASAHEB AMBEDKAR MARTHWADA UNIVERSITY AURANGABAD

SAGAR COLLEGE, JALNA.
DEPARTMENT OF COMPUTER MANAGEMENT
(2018-19)



CERTIFICATE

This is to certify that, Miss. AVINASH ASHOK JADHAV Of B.C.A Third year VI[th] Semester Project Report has Completed the Project towards the project fulfilment of the Subject on "**WEB-APPLICATION SYSTEM**" During the academic Year 2018-2019.

Internal Examiner

External Examiner

PRINCIPAL
Sagar College, Jalna
Principal

DECLARATION

We hereby declare that the project intitled **"WEB-APPLICATION SYSTEM"** which is being submitted as Mini project in Computer Managment Department to **SAGAR COLLEGE, JALNA** is an authentic record of our genuine work done under the guidance of **Prof. N D Najardhane**, Dept. Of Computer Managment, Sagar College, Jalna.

Mr. AVINASH ASHOK JADHAV

B.C.A VI th Sem

ACKNOWLEDGEMENT

We wish to express our sincere gratitude to **Mr. D Suradkar Sir** principle of *Sagar College Jalna* for providing me an opportunity to do our project work in **"WEB-APPLICATION SYSTEM."**

I sincerely thanks to **Prof. N D Najardhane** for their guidance and encouragement in carrying out this project work. I also wish to express my gratitude to the officials and other staff member *of Sagar. College Jalna* who rendered their help during the periods of my project work.

I also thanks to the **Mr. D Suradkar Sir** Secretary of Deepshobha Sevabhavi Sanstha, Khandali for providing me the opportunity to emmark on this project.

Thank you!

Mr. AVINASH ASHOK JADHAV

B.C.A IIInd year

2018-2019

Index

1. Introduction

This is a "web-application" project which introduce itself in the sector of Computer Science.First we have add the student Is with teacher name number and subject too. To facilitate the Attendance System in educational institute it uses the most reliable of uniquely identifing the student's through website this will enable the easy way of maintaining class attendance the system will also generate the report.

Here the most import role is played by parent &admin where we show the separate form to see the report of students. There is a addition of class student teacher and also the feedback sector where parents can give the feedback to teacher. Report are announced in the form of graph where graph are generated in the form of this attendance system is most technical to arrange the sequence of student in the educational life.

It will help the institute to keep the record online by uploading the attendance immediately.

1.1 Existing System & Need for System:-

The system must be evaluated from the technical point of view first. The project should be developed such that the necessary function and performance are achieved within the constrains. The project is developed within latest technology that is Android Studio.

- Android Attendance application development is the process, by which the application for institute/school.

1. Existing System:-

- Reduce the time
- Insert the record related to student is present or absent
- Shows the information related to class student and their contact as well as report related to studies and present student

1.2 Scope of work:-

It will help in collecting the data related to subject accordingly , the data manual arrangement of present, absent and sick and late student.

It will also help the parent to check the response and performance of their child in the institute / school. It will be online process of collecting data and strong in data and storing it into a database.

The android Attendance system will reduce the time of arranging the data technically. It will be easy to understand to user, easy to operate using ID and change and update.

Operating Environment – Hardware & Software:-

Hardware Information:-

Processor – Intel core i3 2.1 GHz.

Ram-4Gb.

HDD- 1TB.

Software Information:-

Windows 10,

Xampp control panel v3 2.2

Microsoft office word 2007

MySQl database.

2.Purpose:-

The purpose of this document is to specify the requirements and preview some elements of the analysis model of the program 'Improver', improver is a simple program which consist of two programs:

- ImproverServer –question and answer database editor and exam server
- ImproverServer - the program to install unto each client computer

2.2 Objective of the system:-

Android attendance system consist of details where are added by Admin, it will provide class details, student details, teacher details too.

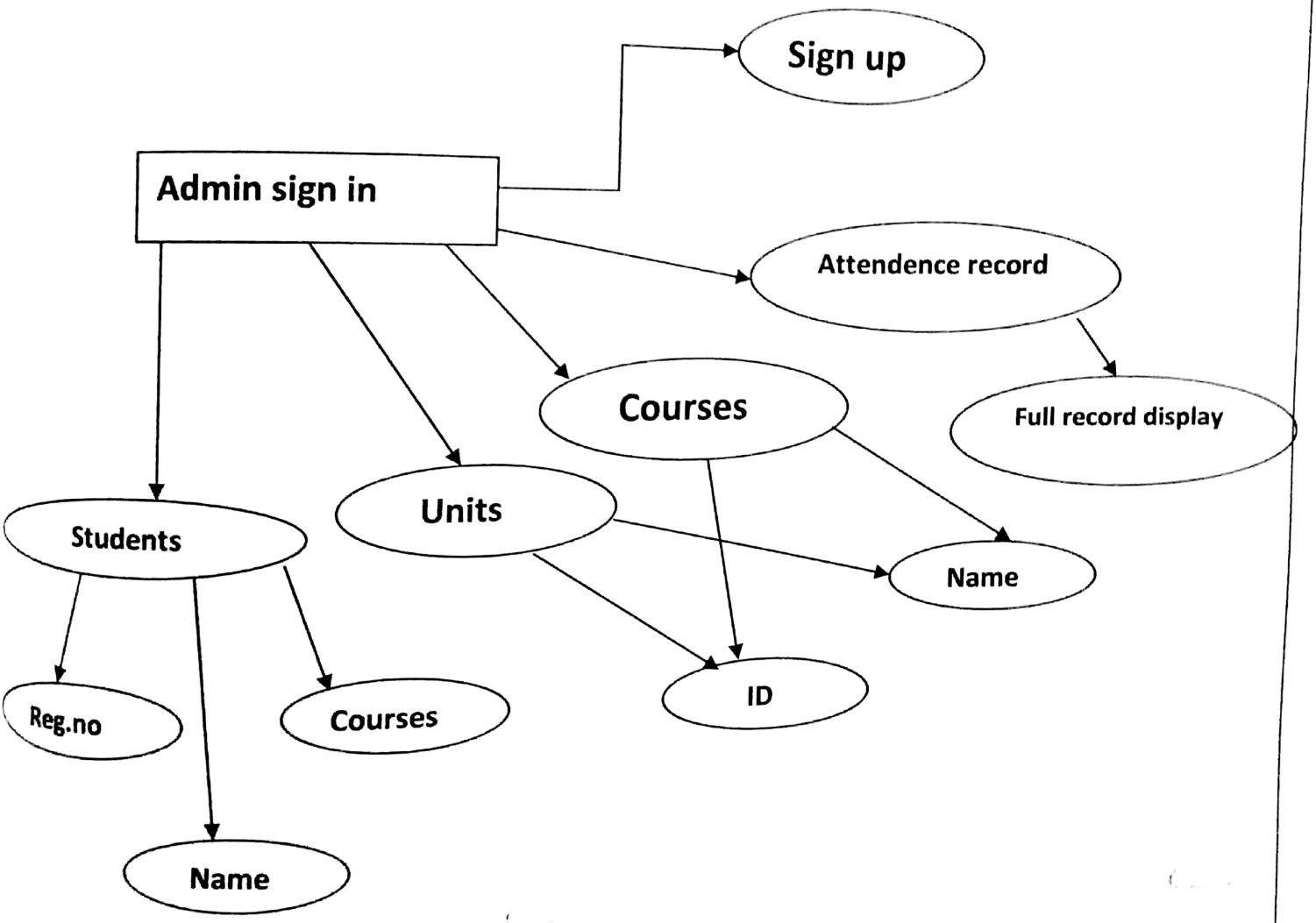
Where reports have sections which shows student reports and attendance. The new details of students and class are added with teacher too.

The details are also edited by admin and updated. It delets the previous record, it main wrk is to reduce manual work for tasking.

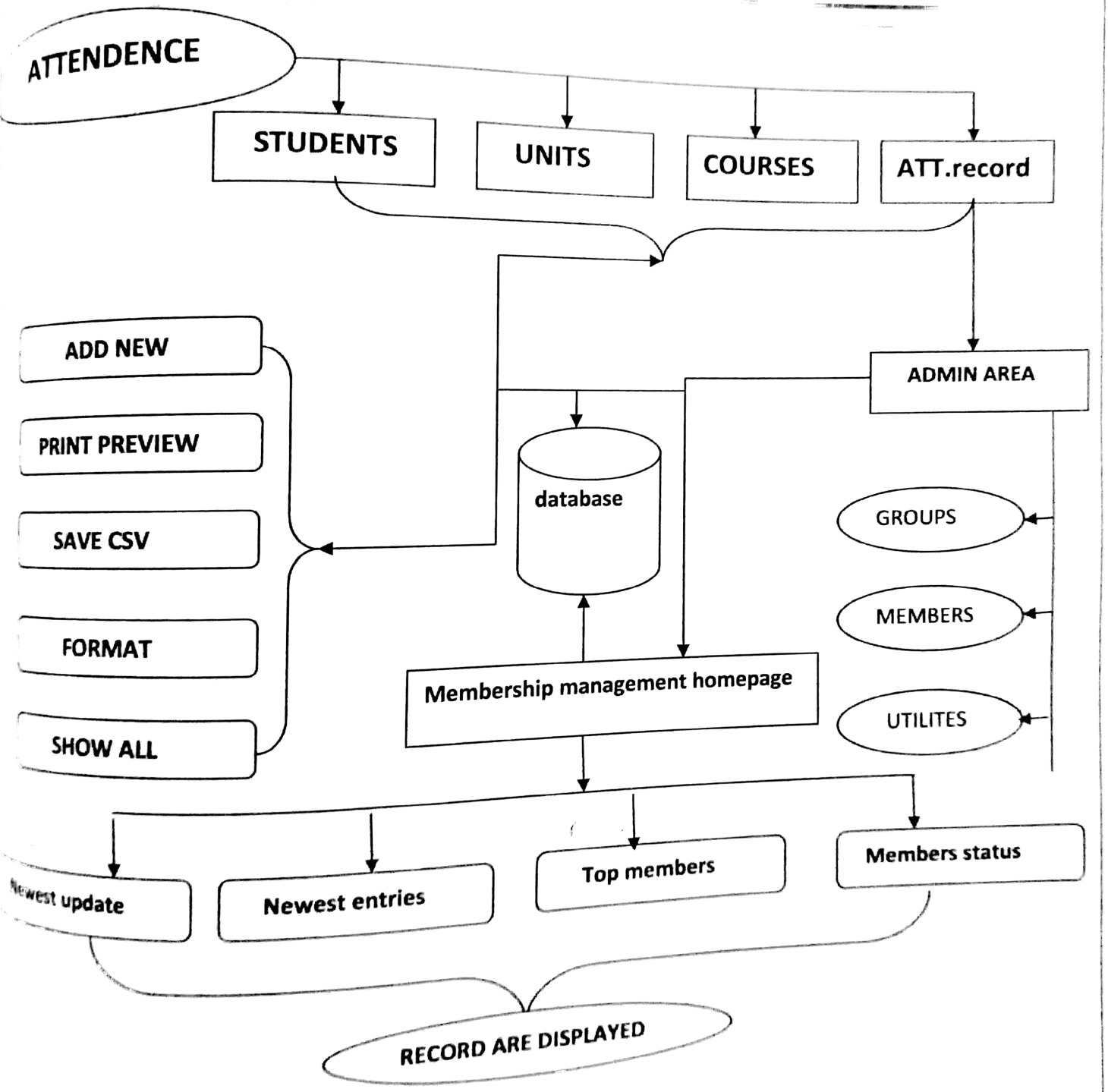
2.3 User Requirments:-

- Managing students details
- Data will be sorted in mannor
- Reliables information will be displayed
- Easily add new information

3.1 E R DIAGRAM:-



3.2 DATA FLOW:-



4.Source code:

Insert the record

```
<?php
function students_insert(){
global $Translation;
// mm: can member insert record?
$arrPerm=getTablePermissions('students');
if(!$arrPerm[1]){
return false;
}
$data['regno'] = makeSafe($_REQUEST['regno']);
if($data['regno'] == empty_lookup_value){ $data['regno'] = ''; }
$data['name'] = makeSafe($_REQUEST['name']);
if($data['name'] == empty_lookup_value){ $data['name'] = ''; }
$data['course'] = makeSafe($_REQUEST['course']);
if($data['course'] == empty_lookup_value){ $data['course'] = ''; }
if($data['regno'] == '') {echo StyleSheet() . "\n\n<div class=\"alert alert-danger\">" .
$Translation['error:'] . " 'Regno': " . $Translation['pkfield empty'] . '</div>'; exit;}
if($data['name']== ''){
echo StyleSheet() . "\n\n<div class=\"alert alert-danger\">" . $Translation['error:'] . " 'Name':
" . $Translation['field not null'] . '<br><br>';
echo '<a href="" onclick="history.go(-1); return false;">'.$Translation['< back'].'</a></div>';
exit;
}
// hook: students_before_insert
if(function_exists('students_before_insert')){
```

```
$args=array();

if(!students_before_insert($data, getMemberInfo(), $args)){ return false; }

$o = array('silentErrors' => true);

sql('insert into `students` set       `regno`=' . (($data['regno'] !== '' && $data['regno'] !==
NULL) ? "'{$data['regno']}'" : 'NULL') . ', `name`=' . (($data['name'] !== '' && $data['name']
!== NULL) ? "'{$data['name']}'" : 'NULL') . ', `course`=' . (($data['course'] !== '' &&
$data['course'] !== NULL) ? "'{$data['course']}'" : 'NULL'), $o);

if($o['error']!=''){

echo $o['error'];

echo "<a href=\"students_view.php?addNew_x=1\">{$Translation['< back']}</a>";

exit;

}$recID = $data['regno'];

// hook: students_after_insert

if(function_exists('students_after_insert')){

$res = sql("select * from `students` where `regno`='" . makeSafe($recID, false) . "' limit 1",
$eo);

if($row = db_fetch_assoc($res)){

$data = array_map('makeSafe', $row);

}

$data['selectedID'] = makeSafe($recID, false);

$args=array();

if(!students_after_insert($data, getMemberInfo(), $args)){ return $recID; }

}// mm: save ownership data

set_record_owner('students', $recID, getLoggedMemberID());

return $recID;

}
```

Membership coding:

```
// save member data

$needsApproval = sqlValue("select needsApproval from membership_groups where
groupID='$groupID'");

sql("INSERT INTO `membership_users` set memberID='$memberID',
passMD5='".md5($password)."', email='$email', signupDate='".@date('Y-m-d')."',
groupID='$groupID', isBanned='0', isApproved='".($needsApproval==1 ? '0' : '1')."',
custom1='$custom1', custom2='$custom2', custom3='$custom3', custom4='$custom4',
comments='member signed up through the registration form.'", $eo);

// admin mail notification

/* --- application name as provided in AppGini is used here ---- */

$message = nl2br("A new member has signed up for {$app_name}.\n\n" ."Member name:
{$memberID}\n" ."Member group: " . sqlValue("select name from membership_groups
where groupID='{$groupID}'") . "\n" ."Member email: {$email}\n" .

"IP address: {$_SERVER['REMOTE_ADDR']}\n" .

"Custom fields:\n" .($adminConfig['custom1'] ? "{$adminConfig['custom1']}: {$custom1}\n" :
'') .($adminConfig['custom2'] ? "{$adminConfig['custom2']}: {$custom2}\n" : '') .

($adminConfig['custom3'] ? "{$adminConfig['custom3']}: {$custom3}\n" : '') .

($adminConfig['custom4'] ? "{$adminConfig['custom4']}: {$custom4}\n" : '')

if($adminConfig['notifyAdminNewMembers'] == 2 && !$needsApproval){

sendmail(array('to' => $adminConfig['senderEmail'],'subject' => "{{$app_name}} New
member signup",'message' => $message

if($adminConfig['notifyAdminNewMembers'] >= 1 &&
$needsApproval){sendmail(array('to' => $adminConfig['senderEmail'],

'subject' => "{{$app_name}} New member awaiting approval",'message' => $message

// hook: member_activity

if(function_exists('member_activity')){

$args=array(); member_activity(getMemberInfo($memberID), ($needsApproval ? 'pending' :
'approved'), $args);}

// redirect to thanks page
```

membership_thankyou.php$redirect");

exit;

Update coding:

```
function students_update($selected_id){

global $Translation;

// mm: can member edit record?$arrPerm=getTablePermissions('students');

$ownerGroupID=sqlValue("select groupID from membership_userrecords where
tableName='students' and pkValue='".makeSafe($selected_id)."'");

 $ownerMemberID=sqlValue("select lcase(memberID) from membership_userrecords where
tableName='students' and pkValue='".makeSafe($selected_id)."'");

 if(($arrPerm[3]==1 && $ownerMemberID==getLoggedMemberID()) || ($arrPerm[3]==2 &&
$ownerGroupID==getLoggedGroupID()) || $arrPerm[3]==3){ // allow update?

// update allowed, so continue ...

}else{

return false;

$data['regno'] = makeSafe($_REQUEST['regno']);if($data['regno'] == empty_lookup_value){
$data['regno'] = ''; }

$data['name'] = makeSafe($_REQUEST['name']);

if($data['name'] == empty_lookup_value){ $data['name'] = ''; }

if($data['name']==''){

echo StyleSheet() . "\n\n<div class=\"alert alert-danger\">{$Translation['error:']} 'Name':
{$Translation['field not null']}<br><br>";echo '<a href="" onclick="history.go(-1); return
false;">'.$Translation['< back'].'</a></div>';

exit;}$data['course'] = makeSafe($_REQUEST['course']);     if($data['course'] ==
empty_lookup_value){ $data['course'] = ''; }

$data['selectedID']=makeSafe($selected_id);

// hook: students_before_update

if(function_exists('students_before_update')){
```

```
$args=array();

if(!students_before_update($data, getMemberInfo(), $args)){ return false; }}

$o=array('silentErrors' => true);

sql('update `students` set `regno`=' . (($data['regno'] !== '' && $data['regno'] !== NULL) ?
"'{$data['regno']}'" : 'NULL') . ', `name`=' . (($data['name'] !== '' && $data['name'] !== NULL)
? "'{$data['name']}'" : 'NULL') . ', `course`=' . (($data['course'] !== '' && $data['course'] !==
NULL) ? "'{$data['course']}'" : 'NULL') . " where `regno`='".makeSafe($selected_id)."'", $o);

if($o['error']!=''){echo $o['error'];

echo '<a
href="students_view.php?SelectedID='.urlencode($selected_id)."\">{$Translation['<
back']}</a>";exit;}

$data['selectedID'] = $data['regno'];

// hook: students_after_update

if(function_exists('students_after_update')){

        $res = sql("SELECT * FROM `students` WHERE `regno`='{$data['selectedID']}' LIMIT 1",
$eo)if($row = db_fetch_assoc($res))$data = array_map('makeSafe', $row}

$data['selectedID'] = $data['regno'];

$args = array();        if(!students_after_update($data, getMemberInfo(), $args)){ return; }

}// mm: update ownership data

sql("update membership_userrecords set dateUpdated='".time()."',
pkValue='{$data['regno']}' where tableName='students' and
pkValue='".makeSafe($selected_id)."'", $eo);

}
```

Delete coding:

```
function students_delete($selected_id, $AllowDeleteOfParents=false, $skipChecks=false){
// insure referential integrity ...
global $Translation; $selected_id=makeSafe($selected_id);
// mm: can member delete record?
$arrPerm=getTablePermissions('students');
$ownerGroupID=sqlValue("select groupID from membership_userrecords where
tableName='students' and pkValue='$selected_id'");
        $ownerMemberID=sqlValue("select lcase(memberID) from membership_userrecords where
tableName='students' and pkValue='$selected_id'");
        if(($arrPerm[4]==1 && $ownerMemberID==getLoggedMemberID()) || ($arrPerm[4]==2 &&
$ownerGroupID==getLoggedGroupID()) || $arrPerm[4]==3){ // allow delete?
// delete allowed, so continue ...
}else{ return $Translation['You don\'t have enough permissions to delete this record'];
} // hook: students_before_delete
if(function_exists('students_before_delete')){ $args=array();
if(!students_before_delete($selected_id, $skipChecks, getMemberInfo(), $args))
return $Translation['Couldn\'t delete this record'];
} // child table: attendance
$res = sql("select `regno` from `students` where `regno`='$selected_id'", $eo);
$regno = db_fetch_row($res);      $rires = sql("select count(1) from `attendance` where
`student`='".addslashes($regno[0])."'", $eo);  $rirow = db_fetch_row($rires);
if($rirow[0] && !$AllowDeleteOfParents && !$skipChecks){
$RetMsg = $Translation["couldn't delete"];  $RetMsg = str_replace("<RelatedRecords>", $rirow[0],
$RetMsg);  $RetMsg = str_replace("<TableName>", "attendance", $RetMsg);
return $RetMsg;  }elseif($rirow[0] && $AllowDeleteOfParents && !$skipChecks){
$RetMsg = $Translation["confirm delete"];  $RetMsg = str_replace("<RelatedRecords>", $rirow[0],
$RetMsg);  $RetMsg = str_replace("<TableName>", "attendance", $RetMsg);
$RetMsg = str_replace("<Delete>", "<input type=\"button\" class=\"button\"
value=\"".$Translation['yes']."\"
```

```
onClick=\"window.location='students_view.php?SelectedID=".urlencode($selected_id)."&delete_x=
1&confirmed=1';\">", $RetMsg); $RetMsg = str_replace("<Cancel>", "<input type=\"button\"
class=\"button\" value=\"".$Translation['no']."\"
onClick=\"window.location='students_view.php?SelectedID=".urlencode($selected_id)."';\">",
$RetMsg); return $RetMsg; }

sql("delete from `students` where `regno`='$selected_id'", $eo);

// hook: students_after_delete

if(function_exists('students_after_delete')){  $args=array();

students_after_delete($selected_id, getMemberInfo(), $args);

} // mm: delete ownership data

sql("delete from membership_userrecords where tableName='students' and
pkValue='$selected_id'", $eo);

}
```

Database coding:

```
<?php

/*You can implement a new db driver in this file by uncommenting the line below

and replacing 'mssql' with the desired driver name, then adding cases for the

that driver in each function below ...*/

// define('DATABASE', 'mssql');

if(!defined('DATABASE')){

if(function_exists('mysqli_connect')) { define('DATABASE', 'mysqli');

}else{ define('DATABASE', 'mysql');     }

}  define('mysql_charset', 'utf8');

function db_link($link = NULL){   static $db_link;

if($link) $db_link = $link;  return $db_link;}

function db_close($link = NULL){

db_link(false); /* close db link */  switch(DATABASE){

case 'mysql':return mysql_close($link);case 'mysqli':
```

```
return mysqli_close($link);}  }function db_select_db($dbname, $link = NULL){
if(!$link) $link = db_link(); switch(DATABASE){  case 'mysql':
return mysql_select_db($dbname, $link);    case 'mysqli':
return mysqli_select_db($link, $dbname);
}} function db_fetch_array($res){
switch(DATABASE){    case 'mysql':
return @mysql_fetch_array($res);    case 'mysqli':
return @mysqli_fetch_array($res, MYSQLI_BOTH);}
}  function db_fetch_assoc($res){
switch(DATABASE){
case 'mysql':  return @mysql_fetch_assoc($res);  case 'mysqli':
return @mysqli_fetch_assoc($res)}}
function db_fetch_row($res){  switch(DATABASE){ case 'mysql':
return @mysql_fetch_row($res);    case 'mysqli':
return @mysqli_fetch_row($res);}  }
function db_num_fields($res){    switch(DATABASE){    case 'mysql':
return @mysql_num_fields($res);
case 'mysqli':  return @mysqli_num_fields($res);}
}function db_num_rows($res){  switch(DATABASE){  case 'mysql':
return @mysql_num_rows($res);
case 'mysqli':  return @mysqli_num_rows($res); } } function db_affected_rows($link = NULL){
if(!$link) $link = db_link();  switch(DATABASE){ case 'mysql': return mysql_affected_rows($link);
case 'mysqli':    return mysqli_affected_rows($link);  } }
function db_query($query, $link = NULL){if(!$link) $link = db_link();
switch(DATABASE){ case 'mysql': return @mysql_query($query, $link);
case 'mysqli':  return @mysqli_query($link, $query);}  }
function db_insert_id($link = NULL){  if(!$link) $link = db_link(); switch(DATABASE){
```

```
case 'mysql': return mysql_insert_id($link); case 'mysqli': return mysqli_insert_id($link);}
} function db_field_name($res, $field_offset){ switch(DATABASE){ case 'mysql':
return @mysql_field_name($res, $field_offset); case 'mysqli':
$fo = @mysqli_fetch_field_direct($res, $field_offset); if($fo === false) return false;
return $fo->name; } } function db_field_type($res, $field_offset){
switch(DATABASE){m case 'mysql': return @mysql_field_type($res, $field_offset);
case 'mysqli': $fo = @mysqli_fetch_field_direct($res, $field_offset);
if($fo === false) return false; return $fo->type;} }
function db_escape($str = NULL, $link = NULL){ if(!$link) $link = db_link();
switch(DATABASE){ case 'mysql':
if(function_exists('mysql_real_escape_string')) return mysql_real_escape_string($str, $link);
return mysql_escape_string($str); case 'mysqli': return mysqli_real_escape_string($link, $str);}
} function db_connect($host = NULL, $username = NULL, $passwd = NULL, $dbname = NULL, $port =
NULL, $socket = NULL){ switch(DATABASE){ case 'mysql':
if($host === NULL) $host = ini_get("mysql.default_host");
if($username === NULL) $username = ini_get("mysql.default_user");
if($passwd === NULL) $passwd = ini_get("mysql.default_password");
$link = mysql_connect($host, $username, $passwd);
if(!$link) return false; db_link($link); /* db_link() can now be used to retrieve the db link from
anywhere */@mysql_query("SET NAMES '" . mysql_charset . "'");if($dbname){
if(!mysql_select_db($dbname, $link)) return false; }return $link;case 'mysqli':
if($host === NULL) $host = ini_get("mysqli.default_host");
if($username === NULL) $username = ini_get("mysqli.default_user");
if($passwd === NULL) $passwd = ini_get("mysqli.default_pw");
if($dbname === NULL) $dbname = "";  if($port === NULL) $port = ini_get("mysqli.default_port");
if($socket === NULL) $socket = ini_get("mysqli.default_socket");
$link = mysqli_connect($host, $username, $passwd, $dbname, $port, $socket); if(!$link) return false;
db_link($link); /* db_link() can now be used to retrieve the db link from anywhere
*/mysqli_set_charset($link, mysql_charset);
```

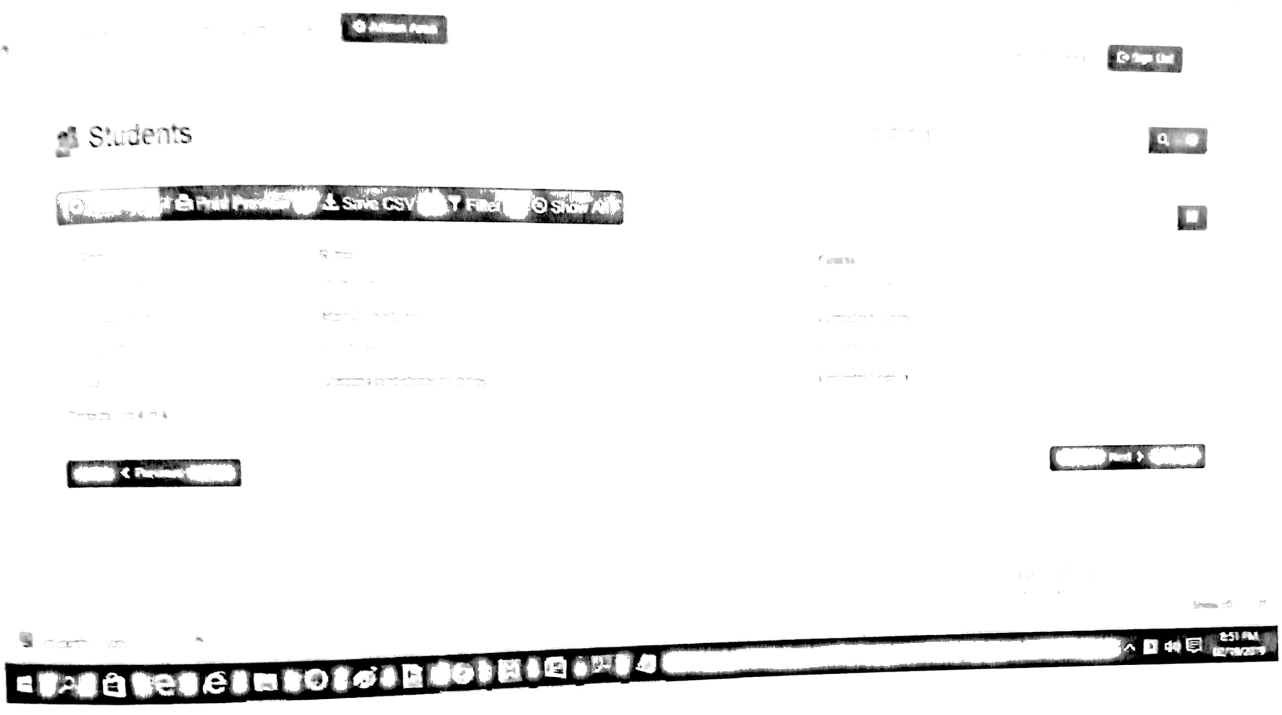
```
return $link;} }
function db_errno($link = NULL, $mysqli_connect = false){ if(!$link) $link = db_link();
switch(DATABASE){ case 'mysql': return mysql_errno($link); case 'mysqli':
if($mysqli_connect) return mysqli_connect_errno(); return mysqli_errno($link);}
} function db_error($link = NULL, $mysqli_connect = false){
if(!$link) $link = db_link(); switch(DATABASE){ case 'mysql':
return mysql_error($link); case 'mysqli': if($mysqli_connect) return mysqli_connect_error();
return mysqli_error($link);} }
```

Ouput screen:

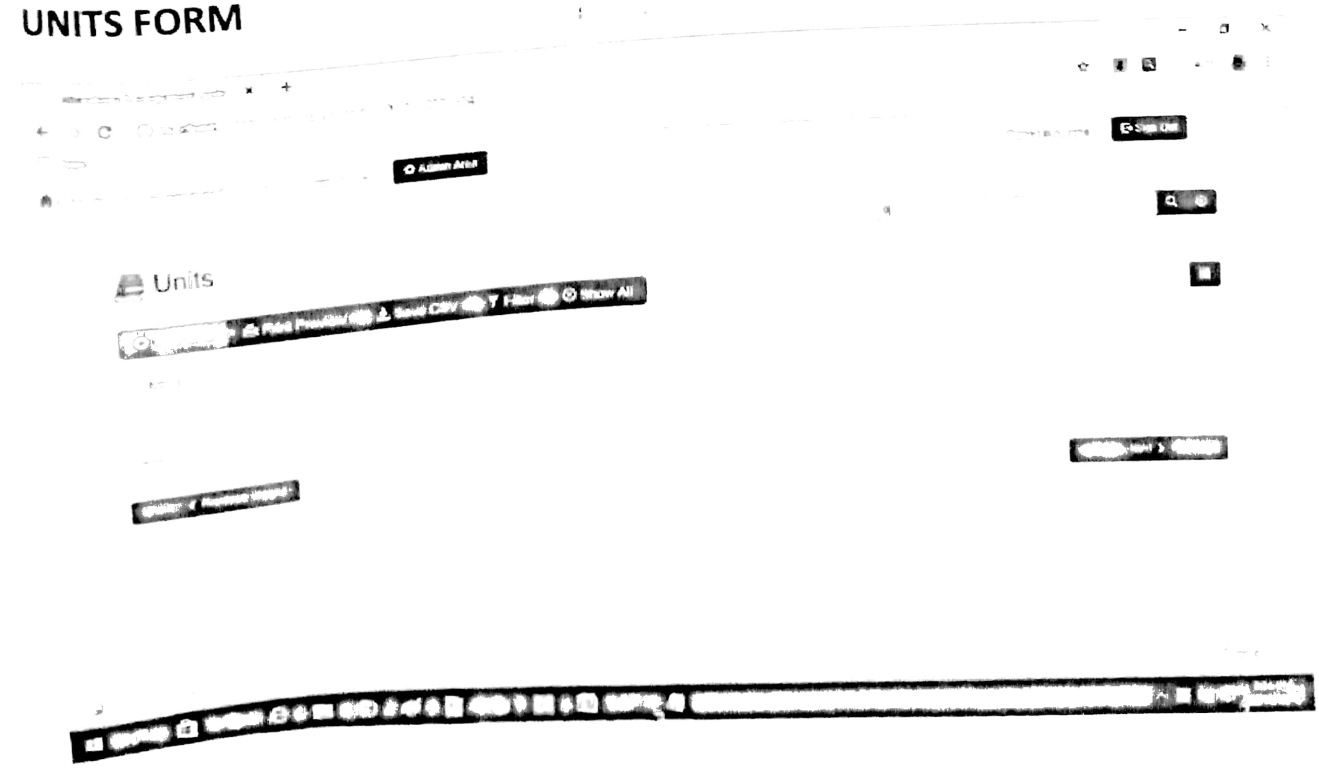
Sign up & Sign in

Main Menu

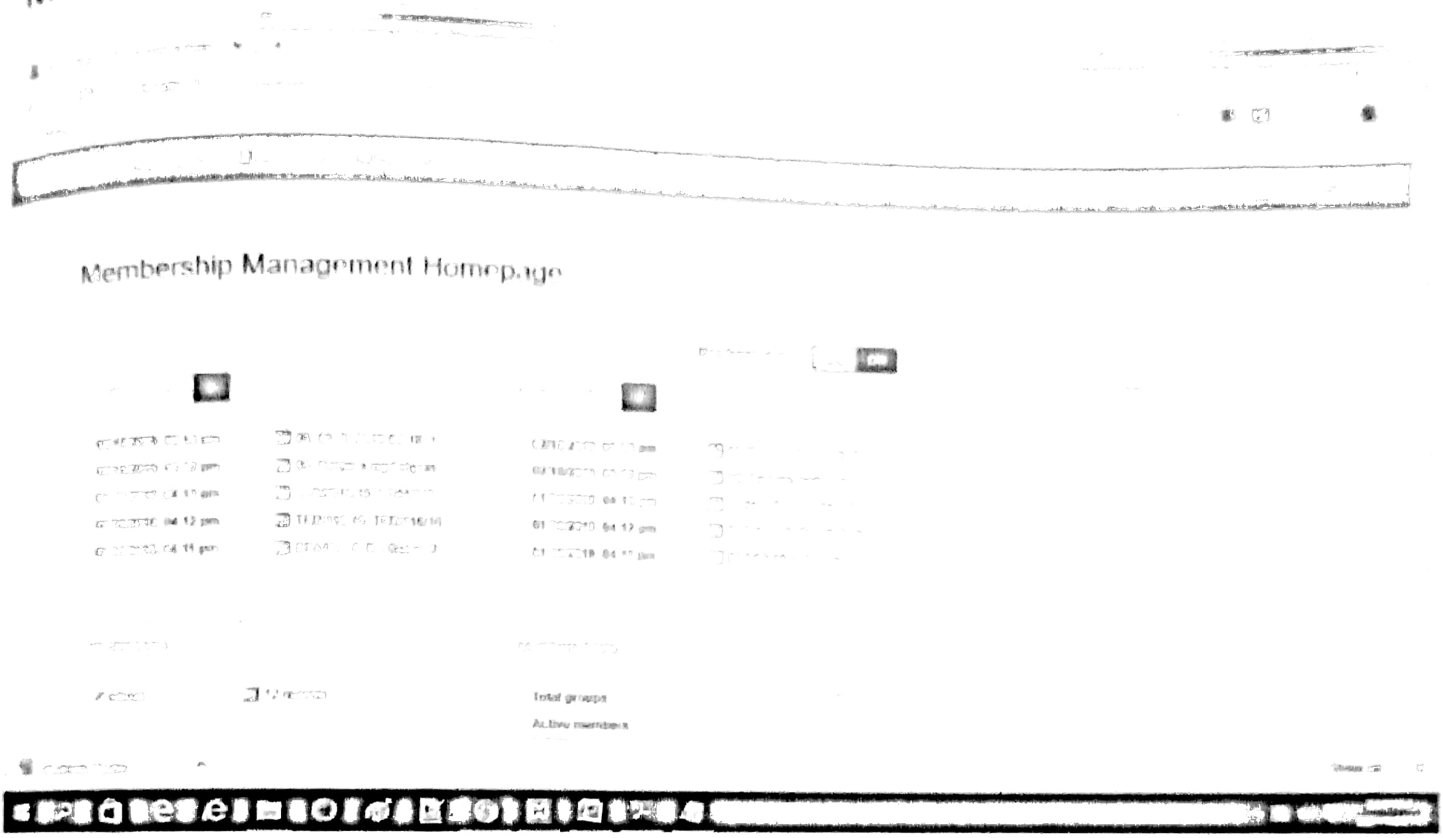
STUDENT FORM



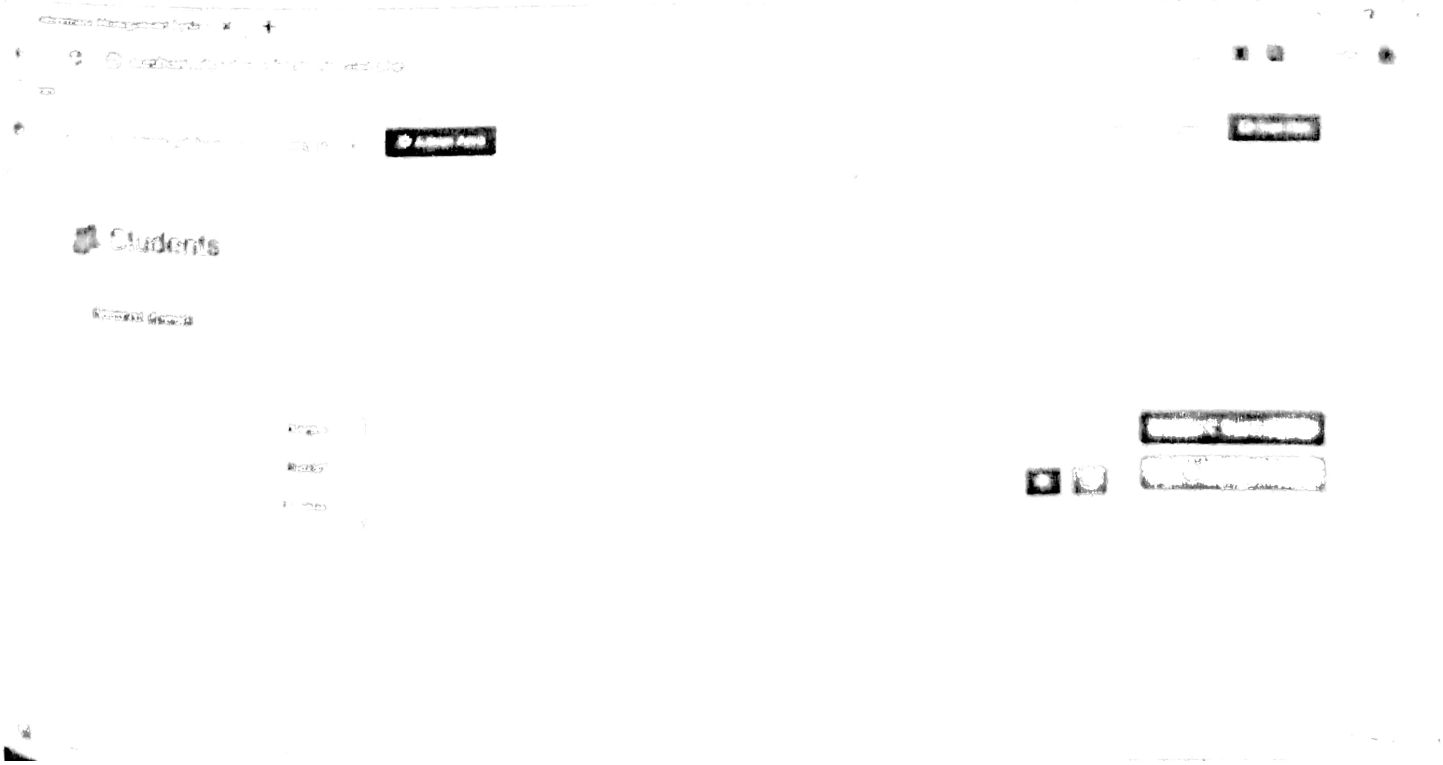
UNITS FORM



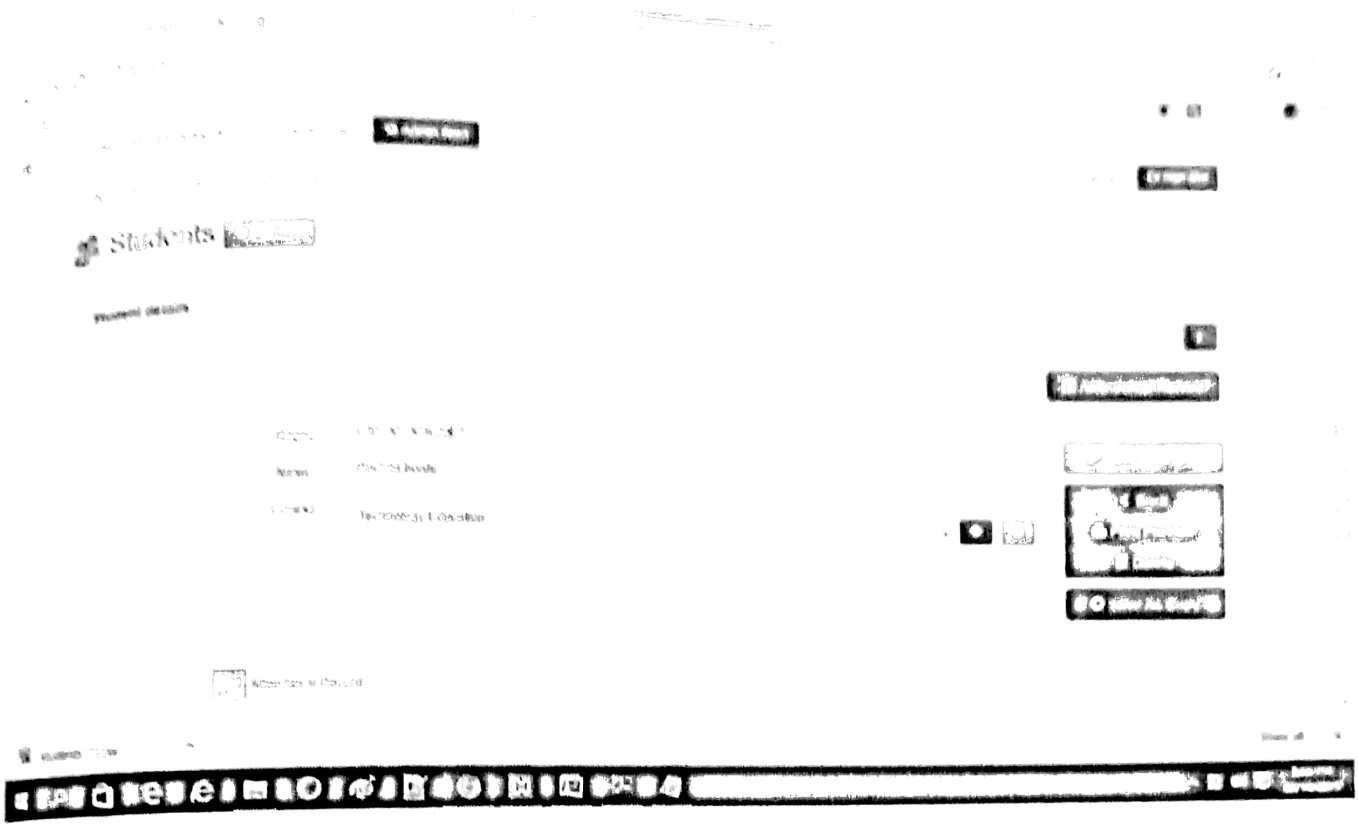
MEMBERSHIP FORM



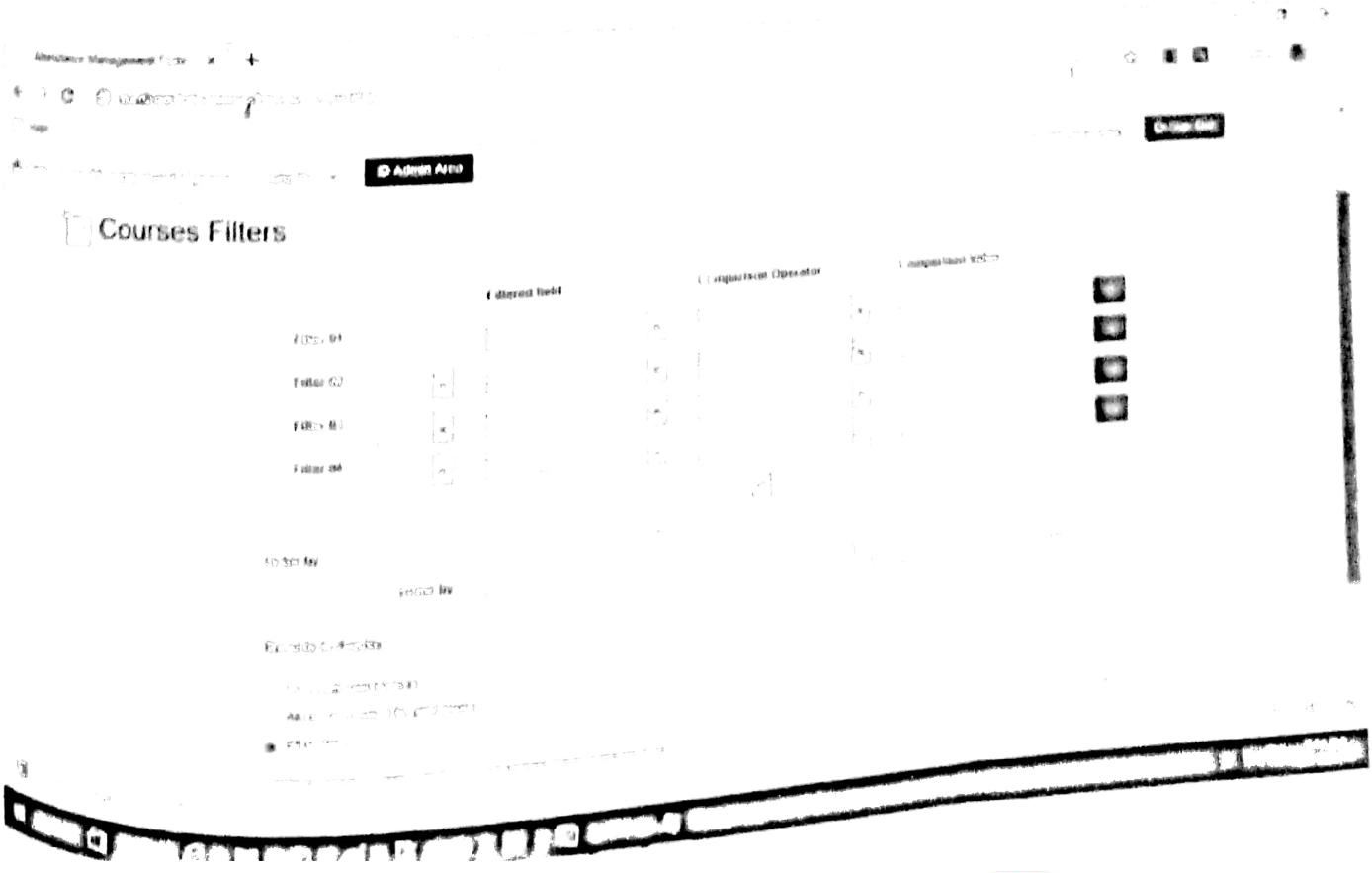
STUDENDS LOGIN



RECORD SAVE



COURSES FILTERS

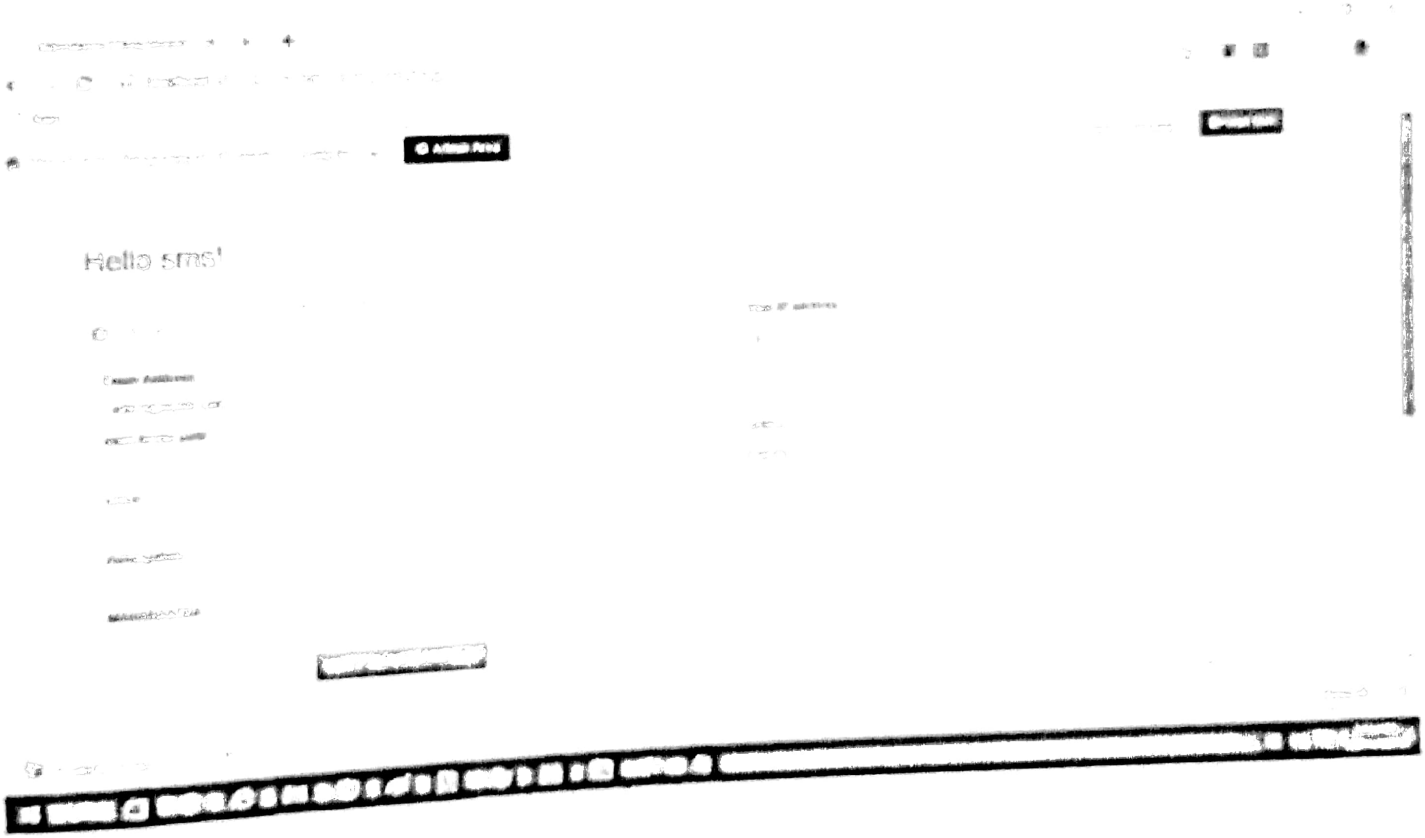


ATTENDENCE RECORD

Attendance Record

SIGN UP

ADMIN INFO



6.Conclusion:-

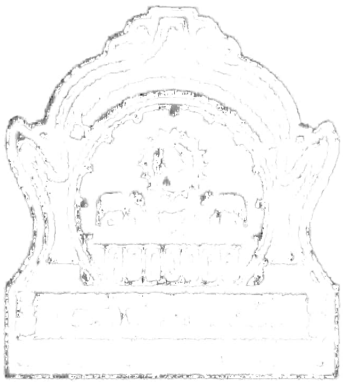
Our project is only a humble venture to satisfy the needs to manage their project work. Several user friendly coding have also adopted. This package shall prove to be a powerful package to satisfying all the requirement of the college. The objective of the software planning is to be provide a framework that enables the manager to make reasonable estimate made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

7. Bibliography:-

1. Professional Android 4 Application Development
2. Programming Android ; java programming for the new generation of the mobile devices.
3. Android Cookbook; problems and solutions for android developers. First edition.
4. www.youtube.com
5. www.phptutorials.com

Dr. Babasaheb Ambedkar Marathwada University,

Aurangabad



Have Satisfactorily carried out and completed the project work entitled

A

Project On

"Banking System"

This work is being submitted for the award of the degree in B.C.A. 2018-2019

Submitted by

Mr. Satish S.Borde

Guided by

Prof. D. E. Suradkar

CERTIFICATE

This is to certify that the seminar report entitled,
" "Banking System" I"Submitted by
Mr. Satish S. Borde, as per the requirement of *Dr. Babasaheb Ambedkar Marathwada University* In the partial fulfilment of bachelor of computer application third year for the academic year 2018-19.

Seat No. BCA-060421

PRINCIPAL
Sagar College, Jalna
Principle

Guided by

Prof. *D. E. Suradkar*

External Examiner

ACKNOWLEDGEMENT

With our great pleasure, we wish to express our knowledge under the great guidance of [redacted] who help us with her graceful support and other infrastructure with personal attention.

We are also thankful to our project guide [redacted] who herself a knowledgeable person with the great brilliance. We thanking her, for her immense interest, valuable guidance, kindly suggestion and co-operation thought out the period of undertaken which have been instrument in the success of our project.

It is matter of honour to express our special thanks to all the staff members who support us I completion of our project and provide us their own interest.
We also thankful to all our friends who have directly or indirectly supported us by morally.

Mr. Satish S. Borde

B.C.A. (IIIrd Year)

Sagar College of Management, Jalna

DECLARATION

I have undersign that I have completed the project work on the topic Software Developmentfrom INSTITUTE MANAGEMENT SYSTEMinpartial fulfilment of B.C.A. course as related to the Dr. Babasaheb Ambedkar Marathwada University, Aurangabad as per the syllabus of B.C.A. degree.

I hereby declare that this project is genuine and origin and never been submitted previously by me for the award of any other degree or any other university.

Mr. Satish S. Borde

INTRODUCTION TO VISUAL BASIC 6.0

Microsoft Visual Basic, is the fastest and easiest way to create applications for Microsoft windows. Visual basic provides a complete set of tools to simplify rapids application development.

So what is Visual Basic? The "virtual" part refers to the method used to create the graphical user interface (GUI). Rather than writing numerous lines of codes to describe the appearance and location of interface elements, you simply add rebuilt objects into place on screen. If you have ever used a drawing program such as Paint, you already have most of the skills necessary to create an effective user interface.

The "Basic" part refers to the BASIC language, a language used by more programmers then any other language in the history of computing. VB has evolved from the original BASIC language and now contains several hundreds statements, functions, and keywords, many of which relate directly to the Windows GUI. Beginners can create useful applications by learning just a few of the keywords, yet the power of the language allows professionals to accomplished using any other Windows programming language.

The visual Basic programming language is not unique to Visual Basic. The Visual Basic programming system, Applications Edition included in MS Excel, MS Access, and many other windows applications uses the same language. The Visual Basic scripting (VB Script) is a widely used scripting language and a subset of the Visual Basic language. The investment you make in learning Visual Basic will carry to these areas.

MODULES :

In Visual Basic we can create a procedure or function or a variable or a function global for that project using a useful feature of VB called module. Modules help us in reducing codes & faster execution. As in any programming language such as C++ or JAVA in which access specifiers are used we can use same type of public access specifier for the module

MDI FORM :

The multiple document interface (MDI) allows us to create an application that maintains multiple forms within a single container. An MDI application allows the user to display documents at the same time with each document displayed in its own [illegible] child windows are contained in a parent window [illegible] all child windows of different types [illegible] MDI child [illegible]

[illegible]

With menu editor, we can add new commands to existing menus, replace existing menu commands with our own commands, create new menus and menu bar, and change and delete existing menus and menu bars. The main advantage of menu editor is its ease of use. We can customize menu in a completely manner that involves very little programming.

DATA REPORTS :

VB's data report designer is a versatile data report generator that features the ability to created-banded hierarchical reports. Used in conjunction with a data source such as the Data Environment designer, we can create reports drom several different relational tables. In data report drags and drop functionality works very effectively for fields. VB automatically creates a text box control on the data report and sets the data member and data field properties of the dropped field. We can also drag a command object from the data environment designer to the data report designer.

VB'S STANDARD CONTROL :

COMMAND BUTTON CONTROL :

We can use a command button control to begin, interrupt, or end a process. When chosen, a Command Button appears pushed in and so is sometimes called a push button. To display text on a Command Button control, set its Caption property. A user can always choose a Command Button by clicking it. To allow the user to choose it by pressing ENTER, set the Default property to true. To allow the user to choose the button by pressing ESC, set the Cancel property of the Command Button to True.

LABEL CONTROLS :

A Label tool display text on a form that user cannot change. It is mainly used to give the labels or messages to the Text Box.

TEXT BOX :

A Text Box command is used to accept the data in it. We can dispiay the data, change the data etc. that is available in this box.

CHECK & OPTION BOX :

Check Box is used to select one or more choices from the given choices. Option Button is used to provide an option to the user and user can select only one of them.

LIST & DRIVE LIST BOX :

List Box is similar to Combo box, used to display lost of item. Drive List Boxis used to display all the drives, which are connected to your computer.

TIMER :

A Timer control can execute code at regular intervals by causing a Timer event to occur. The Timer control is invisible to user & is useful for background processing.

FILE LIST & DIRECTORY LIST BOX :

File List Box is used to display list of all the [illegible] Directory List Box [illegible] to display all the [illegible]

SHAPE BOX :

Shape Box is used to draw graphical elements, such as boxes, lines, circles, etc.

OLE :

Object Linking and Embedding is called as OLE. OLE is used to insert word, excel or other documents in your form.

INTRODUCTION TO MS-ACCESS

A database is a collection of related and ordered information, organized in such a way that information can be accessed quickly and easily. A database can consist of one or more tables of information that are related in some way. A Database Management System(DBMS) is a computer based system to manage a database, or a collection of databases or files.

Microsoft Access 2000 is a DBMS package from Microsoft. It is one of the products in the Microsoft Office 2000 suite. MS-ACCESS supports RDBMS features like setting Primary key, relations between more than one table, designing queries, reports.

- A primary key is to be set to a table, which uniquely identifies each record.
- A query that displays specified columns and data from a table(s).

MS-ACCESS supports SQL – Structured Query Language which is used to manipulate data, retrieve data from more than one tables or for query designing.

The three basic data functions provided by SQL are:

Data Definition Language (DDL):

Consists of commands to create the objects such as tables, views, indexes etc.

Data Manipulation Language (DML):

Uses for query , insertion , deletion and updation of data stored in the database.

Data Control Language (DCL):

this is used for controlling data and their access to the database.

PROJECT OVERVIEW

The Domain "Banking System" keeps the day by day tally record as a complete banking. It can keep the information of Account type, account opening form, Deposit, Withdrawal, and Searching the transaction, Transaction report, Individual account opening form, Group Account. The exciting part of this project is; it displays Transaction reports, Statistical Summary of Account type and Interest Information.

SYNOPSIS

Banking System" keeps the day by day tally record as a complete banking. It can keep the information of Account type, account opening form, Deposit, Withdrawal, and Searching the transaction, Transaction reports, Individual account opening form, Group Account. The exciting part of this project is; it displays Transaction reports, Statistical Summary of Account type and Interest Information.

AIM

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.
They are:
Administrative Module

ADMINISTRATIVE MODULE

This module is the main module which performs all the main operations in the system. The major operations in the system are

- Account
- Deposit
- Withdrawal

- Account type
- Searching Transaction
- Transaction report

SYSTEM STUDY AND ANALISYS

SYSTEM ANALYSIS

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minutest detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

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.

2.1.1 PROBLEMS WITH EXISTING SYSTEM

- [illegible]
- [illegible]

- 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.

PROPOSED SYSTEM

The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces the manual work.

2. 2. 1 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.
- Reduce the damages of the machines.
- Minimize manual data entry.
- Minimum time needed for the various processing.
- Greater efficiency.
- Better service.
- User friendliness and interactive.
- Minimum time required.

2.3. 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 [illegible]. Thus when a new application is proposed it normally goes through a feasibility study before it is approved for development.

[illegible] and provide the [illegible] of the project that is being designed and [illegible] that were [illegible] carefully during the feasibility study [illegible] and Operational feasibilities. The [illegible]

2.3.1. TECHNICAL FEASIBILITY

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.

2.3.2. ECONOMIC FEASIBILITY

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 give an indication of the system is economically possible for development

2.3.3. BEHAVIORAL FEASIBILITY

This includes the following questions:

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

SYSTEM DESIGN

[illegible]

[illegible] 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, the resultant has undergone the best possible design phase for the highest performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. The design is viewed through two phases of development: Logical and Physical

LOGICAL DESIGN:

[illegible] the system and define the boundaries of a system. It includes the following:

- [illegible] the current physical system – its data flows, file content, volumes, [illegible]
- [illegible] specifications – that is, determines the format, content and [illegible]
- [illegible] specifications – format, content and most of the input functions.
- [illegible] and control specifications
- [illegible] implementation plan
- [illegible] walk through of the information flow, output specifications, [illegible] plan
- [illegible]

[illegible]

- [illegible]
- [illegible]

- 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 resource hardware/software.
- Update benefits , costs , conversion date and system constraints.

Design/Specification activities:

- Concept formulation.
- Problem understanding.
- High level requirements proposals.
- Feasibility study.
- Requirements engineering.
- Architectural design.

MODULE DESIGN

Admin

The Administrator logs in using the admin login. In this module two operations are done. During login the Login and Password is verified with that in the database

INPUT DESIGN

The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting accurate information from the computerized system.

It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to

from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record navigation facilities. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in a maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should :

- Identify the specific output that is needed to meet the requirements.
- Select methods for presenting information.
- Create document, report, or other formats that contain information produced by the system.

3.3 DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

- To structure the data so that there is no repetition of data, thus helps to overcome
- To permit simple retrieval of data in response to query and report request
- To simplify the maintenance of the data through updates, insertions, deletions
- To reduce the need to restructure or reorganize data when new application requirements arise.

RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS):

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a tale represents a set of related values.

RELATIONS, DOMAINS & ATTRIBUTES:

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values. Every value in a relation is atomic, that is not decomposable.

RELATIONSHIPS:

Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.Entity Integrity enforces that no Primary Key can have null values.Referential Integrity enforces that no Primary Key can have null values.

Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity.

NORMALIZATION:

As the name implies, it denoted putting things in the normal form. The application developer uses normalization tries to achieve a sensible organization of data into proper tables and columns, and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups in data and thereby avoids data redundancy which puts excessive burden on the computer resources. There methods

First Normal Form:

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows "relations within relations" or "relations as attribute values within tuples". The only attribute values permitted by 1NF are single atomic or indivisible values.

The first step is to put the data into First Normal Form. This can be donor by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each nonatomic attribute or nested relation. This eliminated repeating groups of data.

A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

Second Normal Form:

According to Second Normal Form, For relations where primary key contains multiple attributes, no nonkey attribute should be functionally dependent on a part of the primary key.

In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependant on apart of the key.

A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attributes of the relation is fully dependent on its primary key alone.

Third Normal Form:

According to Third Normal Form, Relation should not have a nonkey attribute functionally determined by another nonkey attribute or by a set of nonkey attributes. That is, there should be no transitive dependency on the primary key.

In this we decompose and set up relation that includes the nonkey attributes that functionally determines other nonkey attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key.

A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non key attributes.

TABLES STRUCTURE

Table: bank_table
Primary bank id

Field	Data Type	Constraints	Description
bank id	Number(9)	Primary key	
name	char(30)		
type	char(30)		
date	date		
Address	char(30)		
total	number		
Acc_no	number		

Table: bankwithdraw
Primary Key: b id

Field	Data Type	Constraints	Description
emp id	Number(9)	Primary key	
name	char(30)		
Acc_no	Number		
date	date		
amount	Number		

Table: DailyTrans
Primary Key: Account no:

Field	Data Type	Constraints	Description
Account no	Number(9)	Primary key	
firstname	char(30)		
surname	char(30)		
date	date		
TransactionName	char(30)		
Transaction amount	Number(9)		
Previous Balance	Number(9)		
Current Balance	Number(9)		
Branch Name	char(30)		

Table: current
Primary Key: account no

Field	Data Type	Constraints	Description
Account no	Number(9)	Primary key	
Surname	char(30)		
Firstname	char(30)		
Account type	Char(30)		
Current balance	char(30)		

Table: Account type
Primary account no

Field	Data Type	Constraints	Description
Account no	Number(9)	Primary key	
Surname	char(30)		
Firstname	char(30)		
DateOfOpening	date		
Address	char(30)		
Phone no	Number(9)		
Occupation	char(30)		
Nextofkin	char(30)		
Account Type	char(30)		
Account status	char(30)		
Opening amount	Number(9)		

SYSTEM DEVELOPEMENT

4.1 SYSTEM SPECIFICATION

HARDWARE REQUIREMENTS

Processor	:	X86 Compatible processor with 1.7 GHz Clock speed
RAM	:	512 MB or more
Hard disk	:	20 GB or more
Monitor	:	VGA/SVGA
Keyboard	:	104 Keys
Mouse	:	2 buttons/ 3 buttons

SOFTWARE REQUIREMENTS

Operating System	: Windows 2000/XP
Front end	: Visual Basic 6.0
Back end	: MS Access

4.2 SOFTWARE ENVIRONMENT

The Control Properties

Before writing an event procedure for the control to response to a user's input, you have to set certain properties for the control to determine its appearance and how it will work with the event procedure. You can set the properties of the controls in the properties window or at runtime.

Handling some of the common controls

3.2.1 The Text Box

[illegible] the standard control for accepting input from the user as well as to [illegible] the output. It can handle string [illegible] and numeric data but not images [illegible] [illegible] text box can be convert[illegible] [illegible] data by [illegible] [illegible] example [illegible] program that processes the input [illegible]

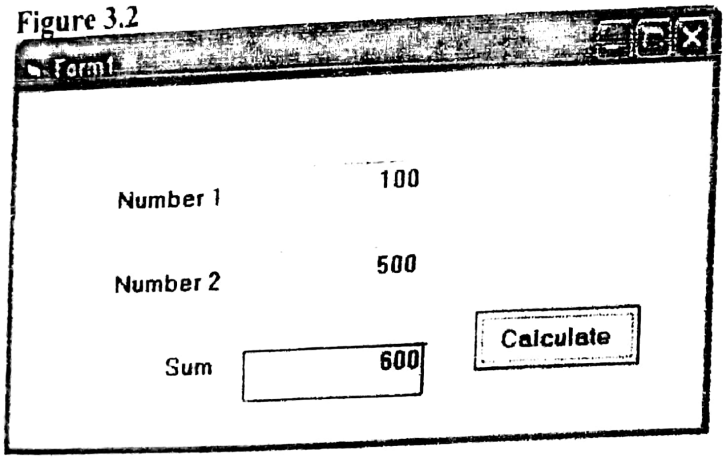
pictures. String in a text box can be converted to a numeric data by using the function Val(text). The following example illustrates a simple program that processes the input from the user.

Example 3.1

In this program, two text boxes are inserted into the form together with a few labels. The two text boxes are used to accept inputs from the user and one of the labels will be used to display the sum of two numbers that are entered into the two text boxes. Besides, a command button is also programmed to calculate the sum of the two numbers using the plus operator. The program use creates a variable sum to accept the summation of values from text box 1 and text box 2.The procedure to calculate and to display the output on the label is shown below. The output is shown in Figure 3.2

```
Private Sub Command1_Click()
'To add the values in text box 1 and text box 2
Sum = Val(Text1.Text) + Val(Text2.Text)
'To display the answer on label 1
Label1.Caption = Sum
End Sub
```

Figure 3.2



3.2.2 The Label

The label is a very useful control for Visual Basic, as it is not only used to provide instructions and guides to the users, it can also be used to display outputs. One of its most important properties is **Caption**. Using the syntax **label.Caption**, it can display text and numeric data . You can change its caption in the properties window and also at runtime. Please refer to Example 3.1 and Figure 3.1 for the usage of label.

3.2.3 The Command Button

The command button is one of the most important controls as it is used to execute commands. It displays an illusion that the button is pressed when the user click on it. The most common event associated with the command button is the Click event and the syntax for the procedure is

```
Private Sub Command1_Click()
Statements
End Sub
```

The Picture Box is one of the controls that is used to handle graphics. You can load a picture at design phase by clicking on the picture item in the properties window and select the picture from the selected folder. You can also load the picture at runtime using the **LoadPicture** method. For example, the statement will load the picture grape.gif into the picture box.

Picture1.Picture=LoadPicture ("C:\VB program\Images\grape.gif")

You will learn more about the picture box in future lessons. The image in the picture box is not resizable.

3.2.5 The Image Box

The Image Box is another control that handles images and pictures. It functions almost identically to the picture box. However, there is one major difference, the image in an Image Box is stretchable, which means it can be resized. This feature is not available in the Picture Box. Similar to the Picture Box, it can also use the LoadPicture method to load the picture. For example, the statement loads the picture grape.gif into the image box.

Image1.Picture=LoadPicture ("C:\VB program\Images\grape.gif")

3.2.6 The List Box

The function of the List Box is to present a list of items where the user can click and select the items from the list. In order to add items to the list, we can use the **AddItem method**. For example, if you wish to add a number of items to list box 1, you can key in the following statements

Example 3.2

```
Private Sub Form_Load ( )

List1.AddItem "Lesson1"
List1.AddItem "Lesson2"
List1.AddItem "Lesson3"
List1.AddItem "Lesson4"

End Sub
```

The items in the list box can be identified by the **ListIndex** property, the value of the ListIndex for the first item is 0, the second item has a ListIndex 1, and the second item has a ListIndex 2 and so on

3.2.7 The Combo Box

The function of the Combo Box is also to present a list of items where the user can click and select the items from the list. However, the user needs to click on the small arrowhead on the right of the combo box to see the items which are presented in a drop-down list. In order to add items to the list, you can also use the **AddItem method**. For example, if you wish to add a number of items to a combo box 1, you can key in the following statements

Example 3.3

```
Private Sub Form_Load ( )
Combo1.AddItem "Item1"
Combo1.AddItem "Item2"
```

```
Combo1.AddItem "Item3"
Combo1.AddItem "Item4"
End Sub
```

3.2.8 The Check Box

The Check Box control lets the user selects or unselects an option. When the Check Box is checked, its value is set to 1 and when it is unchecked, the value is set to 0. You can include the statements Check1.Value=1 to mark the Check Box and Check1.Value=0 to unmark the Check Box, as well as use them to initiate certain actions. For example, the program will change the background color of the form to red when the check box is unchecked and it will change to blue when the check box is checked. You will learn about the conditional statement If....Then....Elesif in later lesson. VbRed and vbBlue are color constants and BackColor is the background color property of the form.

3.2.9 The Option Box

The Option Box control also lets the user selects one of the choices. However, two or more Option Boxes must work together because as one of the Option Boxes is selected, the other Option Boxes will be unselected. In fact, only one Option Box can be selected at one time. When an option box is selected, its value is set to "True" and when it is unselected; its value is set to "False". In the following example, the shape control is placed in the form together with six Option Boxes. When the user clicks on different option boxes, different shapes will appear. The values of the shape control are 0, 1, and 2,3,4,5 which will make it appear as a rectangle, a square, an oval shape, a rounded rectangle and a rounded square respectively.

Example 3.4

```
Private Sub Option1_Click ( )
Shape1.Shape = 0
End Sub
Private Sub Option2_Click()
Shape1.Shape = 1
End Sub
Private Sub Option3_Click()
Shape1.Shape = 2
End Sub
Private Sub Option4_Click()
Shape1.Shape = 3
End Sub
Private Sub Option5_Click()
Shape1.Shape = 4
End Sub
Private Sub Option6_Click()
Shape1.Shape = 5
End Sub
```

3.2.10 The Drive List Box

The Drive ListBox is for displaying a list of drives available in your computer. When you place this control into the form and run the program, you will be able to select different drives from your computer as shown in Figure 3.3

3.2.11 The Directory List Box

The Directory List Box is for displaying the list of directories or folders in a selected drive. When you place this control into the form and run the program, you will be able to select different directories from a selected drive in your computer as shown in Figure 3.4

3.2.12 The File List Box

The File List Box is for displaying the list of files in a selected directory or folder. When you place this control into the form and run the program, you will be able to shown the list of files in a selected directory as shown in Figure 3.5
You can coordinate the Drive List Box, the Directory List Box and the File List Box to search for the files you want. The procedure will be discussed in later lessons.

CODING

5.1 CODING

Main Form Coding

```
Private Sub cmdQuit_Click()
End
End Sub

Private Sub Command1_Click()
End
End Sub

Private Sub MDIForm_Load()
Call connectDatabase
End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Index
Case 1:
frmCustomers.Show
Case 4:
frmDeposits.Show
Case 7:
frmWithdrawal.Show
Case 10:
frmTransactions.Show
End Select
End Sub

Private Sub Toolbar1_ButtonMenuClick(ByVal ButtonMenu As
MSComctlLib.ButtonMenu)
Select Case ButtonMenu.Key
Case "Exit"
Unload Me
End Select
End Sub
```

```
Set rptDeposits.DataSource = Nothing
Set rptDeposits.DataSource = rsDeposit
rptDeposits.Show

Case "withdraw"
Set rptWithdrawals.DataSource = Nothing
Set rptWithdrawals.DataSource = rsWithdrawal
rptWithdrawals.Show


End Select
End Sub


Private Sub cmdAdd_Click()
NewRecord = True

cmdAdd.Enabled = False
cmdSave.Enabled = True

cmdCancel.Enabled = True
cmdEdit.Enabled = False
cmdQuit.Enabled = False
Call UnLock_Form_Controls(Me)
MsgBox ("Are Ypu sure")
Call clear_Form_Controls(Me)
Call GenerateNewAccountCode
txtAccountID.Locked = True
txtAccountName.SetFocus

End Sub

Private Sub cmdCancel_Click()
cmdAdd.Enabled = True
cmdSave.Enabled = False

cmdCancel.Enabled = False
cmdEdit.Enabled = True
cmdQuit.Enabled = True
With rsAccTypes
  If NewRecord = True Then
     CancelUpdate
     NewRecord = False
  Else
     CancelUpdate
  End If
  [illegible]
End Sub
MsgBox ("Are You Sure")
Call Lock_Form_Controls(Me)
```

```
End Sub

Private Sub cmdEdit_Click()
NewRecord = False
cmdAdd.Enabled = False
cmdSave.Enabled = True
cmdCancel.Enabled = False
cmdEdit.Enabled = True
cmdQuit.Enabled = True
MsgBox ("Are you Sure to Edit")
Call UnLock_Form_Controls(Me)

End Sub

Private Sub cmdFirst_Click()
Call MoveToFirst(rsAccTypes)
Call DisplayaccTypes(rsAccTypes)
lblStatus.Caption = CStr("Record :" & rsAccTypes.AbsolutePosition & " of " &
rsAccTypes.RecordCount)
End Sub

Private Sub cmdLast_Click()
Call MoveToLast(rsAccTypes)
Call DisplayaccTypes(rsAccTypes)
lblStatus.Caption = CStr("Record :" & rsAccTypes.AbsolutePosition & " of " &
rsAccTypes.RecordCount)
End Sub

Private Sub cmdNext_Click()
Call MoveToNext(rsAccTypes)
Call DisplayaccTypes(rsAccTypes)
lblStatus.Caption = CStr("Record :" & rsAccTypes.AbsolutePosition & " of " &
rsAccTypes.RecordCount)

End Sub

Private Sub cmdPrevious_Click()
Call MoveToPrev(rsAccTypes)
Call DisplayaccTypes(rsAccTypes)
lblStatus.Caption = CStr("Record :" & rsAccTypes.AbsolutePosition & " of " &
rsAccTypes.RecordCount)
End Sub

Private Sub cmdQuit_Click()
If MsgBox("Are you Sure to Exit", vbYesNo) = vbYes Then
End
End If
End Sub
```

```
End Sub

Private Sub cmdOk_Click()
With deBankUba
  .conBankUba.Open  "PROVIDER=Microsoft.Jet.OLEDB.4.0;Data  Source="  +
App.Path + "\BankUba.mdb;"
  .rscmdStatement.Open "Select * From DailyTrans where [AccountNumber]='" &
txtAcctNo & "'", deBankUba.conBankUba, adOpenDynamic, adLockOptimistic
  rptStatement.Show vbModal
  .conBankUba.Close
End With
Unload Me
End Sub


Private Sub cmdInquery_Click()
fraAcctType.Visible = True
frmStatement.Caption = "Enter your Account Number"
frmStatement.cmdSearch.Caption = "&Inquire"
frmWel.fraTransaction.Visible = False
End Sub

Private Sub cmdNewAcct_Click()
frmNewAcct.Show vbModal
End Sub

Private Sub cmdPin_Click()
fraTransaction.Visible = False
frmChangePin.Show vbModal
End Sub

Private Sub cmdProceed_Click()
fraAcctType.Visible = False
frmStatement.Show vbModal
End Sub

Private Sub cmdStatement_Click()
fraTransaction.Visible = False
fraAcctType.Visible = True
End Sub

Private Sub cmdTranster_Click()
strMessage = "This Service is Un Available at the moment, Please bear with us"
MsgBox strMessage, vbInformation, "ATM SERVICE"
fraTransaction.Visible = True
End Sub

Private Sub cmdWithD_Click()
fraAcctType.Visible = True
frmStatement.Caption = "Enter your Account Number"
```

```
    frmStatement.cmdSearch.Caption = "&Process"
    frmWithdrawTransaction.Visible = False
    End Sub

    Private Sub digits_Click(Index As Integer)
    If cleardisplay Then
      txtDisplay.Text = ""
      cleardisplay = False
    End If
      txtDisplay.Text = txtDisplay.Text + digits(Index).Caption
    End Sub

    Private Sub Form_Activate()
      'ShockwaveFlash1.Movie = App.Path & "\" & "BANKING.swf"
      ShockwaveFlash1.Play
      ShockwaveFlash1.Loop = True
    End Sub

    Private Sub Form_Load()
    Call OpenDB
    fraSecretCode.Visible = True
    End Sub

    Private Sub mnuDeposit_Click()
    frmDeposit.Show vbModal
    End Sub

    Dim curBalance As Currency

    Private Sub cboCustomerNo_Click()

    Set rsTemp = New ADODB.Recordset
    rsTemp.Open "Select * FROM tblCustomers WHERE CustomerID = '" &
    cboCustomerNo.Text & "'", cnBank, adOpenKeyset, adLockOptimistic
    With rsTemp

    If .RecordCount > 0 Then
      txtAccountNo.Text = !AccountNo

      txtNarration.SetFocus
    Else
    MsgBox "Invalid Customer Code", vbInformation

    txtAccountNo.Text = ""
    Exit Sub
    End If
    '
    Dim Button As Button
    Private Sub cbo_Click()
    Set rsTemp = New ADODB.Recordset
```

```
rsTemp.Open "Select * from tblCustomer Where AccountNo = " & cboAccNo.Text
& "", cnBank, adOpenKeyset, adLockOptimistic
With rsTemp
If .RecordCount > 0 Then
cboAccNo = !AccountNo
cboCustomerID = !CustomerID
cboFirst = !FirstName
Else
MsgBox "Invalid customer ID/Name/Account NO. Please Try Again", vbInformation
Exit Sub
End If
.Close
End With


Set rsTemp = New ADODB.Recordset
rsTemp.Open "Select * from tblTransactions Where AccountNo='" & cboAccNo.Text
& "", cnBank, adOpenKeyset, adLockOptimistic
With rsTemp
If .RecordCount > 0 Then
lvwTransactions.ListItems.Clear
Call LoadListView(rsTemp)
'cboAccNo = !AccountNo
'cboCustomerID = !CustomerID
'cboFirst = !FirstName
Else
MsgBox "Invalid customer ID/Name/Account NO. Please Try Again", vbInformation
Exit Sub
End If
.Close
End With


End Sub

Private Sub cboCustomerID_Click()

Set rsTemp = New ADODB.Recordset
rsTemp.Open "Select * from tblCustomers Where customerID='" &
cboCustomerID.Text & "", cnBank, adOpenKeyset, adLockOptimistic
With rsTemp
If .RecordCount > 0 Then
cboAccNo = !AccountNo
cboCustomerID = !CustomerID
cboFirst = !FirstName
Else
MsgBox "Invalid customer ID/Name/Account NO. Please Try Again", vbInformation
Exit Sub
End If
.Close
End With
```

```
End With

Set rsTemp = New ADODB.Recordset
rsTemp.Open "Select * from tblTransactions Where CustomerID='" &
cboCustomerID.Text & "'", cnBank, adOpenKeyset, adLockOptimistic
With rsTemp
If .RecordCount > 0 Then
lvwTransactions.ListItems.Clear
Call LoadListView(rsTemp)
Else
MsgBox "No Transactions bearing this customer ID. Please Try Again",
vbInformation
Exit Sub
End If
.Close
End With
End Sub

Private Sub cboFirst_Click()
Set rsTemp = New ADODB.Recordset
rsTemp.Open "Select * from tblCustomers Where FirstName='" & cboFirst.Text &
"'", cnBank, adOpenKeyset, adLockOptimistic
With rsTemp
If .RecordCount > 0 Then
cboAccNo = !AccountNo
cboCustomerID = !CustomerID
cboFirst = !FirstName
Else
MsgBox "Invalid customer ID/Name/Account NO. Please Try Again", vbInformation
Exit Sub
End If
.Close
End With


Set rsTemp = New ADODB.Recordset
rsTemp.Open "Select * from tblTransactions Where CustomerID='" &
cboCustomerID.Text & "'", cnBank, adOpenKeyset, adLockOptimistic
With rsTemp
If .RecordCount > 0 Then
lvwTransactions.ListItems.Clear
Call LoadListView(rsTemp)
Else
MsgBox "No Transactions bearing this customers' first name. Please Try Again",
vbInformation
Exit Sub
End If
.Close
End With
End Sub
```

```
Private Sub cmdEdit_Click()
With rsTransactions
.MoveFirst
While Not .EOF

If lvwTransactions.SelectedItem.ListSubItems(9) = !Code Then
   frmTransaction.txtCustomerID.Text = !CustomerID
   frmTransaction.txtAccountNo.Text = !AccountNo
   frmTransaction.txtNarration.Text = !Narration
   frmTransaction.txtCheckNo.Text = !CheckNO
   frmTransaction.txtDated.Value = !Dated
   frmTransaction.txtDebit.Text = !Debit
   frmTransaction.txtMode.Text = !Mode
   frmTransaction.txtCredit.Text = !Credit
   frmTransaction.txtBalance.Text = !Balance
   frmTransaction.txtCode.Text = !Code
   .MoveLast
   .MoveNext
   Else
   .MoveNext
   End If
Wend
frmTransaction.Show
MsgBox ("Are You Sure")
End With
End Sub

Private Sub cmdOk_Click()
Set rsTemp = New ADODB.Recordset
rsTemp.Open "Select * from tblTransactions Where Dated BETWEEN #" &
dtFrom.Value & "# AND #" & dtTo.Value & "#", cnBank, adOpenKeyset,
adLockOptimistic
With rsTemp
If .RecordCount > 0 Then
lvwTransactions.ListItems.Clear
Call LoadListView(rsTemp)
Else
MsgBox "No Transactions Were carried out between these Dates. Please Try Again",
vbInformation
Exit Sub
End If
Close
End With
End Sub
```

```
        .cmdStatement_Grouping Val(lvwTransactions.SelectedItem.Text)
        rptStatement.Show
    End With
    "Dim strSql As String
    "'strSql
    "SELECT
    tblCustomers.FirstName,tblCustomers.LastName,tblCustomers.Address,tblCustomers
    .PostalCode,tblCustomers.Location,tblCustomers.OpeningBalance,tblCustomers.Cust
    omerID,tblTransactions.AccountNo,tblTransactions.Debit,tblTransactions.Credit,tblT
    ransactions.Dated,tblTransactions.Mode,tblTransactions.CheckNo,tblTransactions.Co
    de          from          tblCustomers,tblTransactions          where
    tblCustomers.CustomerID=tblTransactions.CustomerID                AND
    tblTransactions.Code=4'"  "&  lvwTransactions.SelectedItem.ListSubItems(9).Text  &
    "'"
    "strSql = " SELECT tblCustomers.AccountNo, tblCustomers.Address,"
    "strSql = strSql & "tblCustomers.LastName, tblCustomers.FirstName,"
    "strSql = strSql & "tblCustomers.CustomerID,"
    "strSql = strSql & "tblCustomers.Location, tblCustomers.OpeningBalance, "
    "strSql = strSql & "tblCustomers.PostalCode, tblTransactions.Code, "
    "strSql = strSql & "tblTransactions.CheckNo, tblTransactions.Credit, "
    "strSql = strSql & "tblTransactions.Debit, tblTransactions.Mode "
    "
    "strSql = strSql & "From tblCustomers, tblTransactions "
    "strSql = strSql & "Where tblCustomers.CustomerID = tblTransactions.CustomerID"
    "
    "Set rsTemp = New ADODB.Recordset
    "rsTemp.Open strSql, cnBank, adOpenKeyset, adLockOptimistic
    "With deBank
    "'.Commands("cmdStatement").Parameters = lvwTransactions.SelectedItem.Text
    "'.cmdStatement_Grouping( ,lvwTransactions.SelectedItem.ListSubItems(9))=
    ".Commands("cmdStatement_Grouping").Parameters("tblCustomersCode") = "1"
    "End With

    'With rsTemp
    'If .RecordCount > 0 Then
    'Set rptStatement.DataSource = Nothing
    'Set rptStatement.DataSource = rsTemp
    'rptStatement.Show
    'Else
    'MsgBox "few"
    'End If
    'End With

End Sub

Private Sub cmdPrint_Click()
rptStatement.Show
End Sub
```

```
End Sub


Private Sub DTPicker1_Click()

End Sub

Private Sub cmdRefresh_Click()
lvwTransactions.Refresh
End Sub


Private Sub Command1_Click()

End Sub

'Text1.Text = lvwTransactions.SelectedItem.Text


Private Sub Form_Load()

Call connectDatabase
Call LoadListView(rsTransactions)
With rsCustomers
.MoveFirst
For X = 1 To .RecordCount
  cboCustomerID.AddItem !CustomerID
  cboFirst.AddItem !FirstName
  cboAccNo.AddItem !AccountNo
  .MoveNext
Next X
End With
Frame1.Enabled = False
End Sub


Private Sub lvwTransactions_ColumnClick(ByVal ColumnHeader As _
  MSComctlLib.ColumnHeader)
  ' Sort according to data in this column
  If lvwTransactions.Sorted And _
    ColumnHeader.Index - 1 = lvwTransactions.SortKey Then
    ' Already sorted on this column, just invert the sort order
    lvwTransactions.SortOrder = 1 - lvwTransactions.SortOrder
  Else
    lvwTransactions.SortOrder = lvwAscending
    lvwTransactions.SortKey = ColumnHeader.Index - 1
  End If
  lvwTransactions.Sorted = True

End Sub
```

```
Public Sub LoadListView(myRs As Recordset)
With myRs
While Not .EOF
'lvwTransactions.ListItems.Add , , !CustomerID & " " & !AccountNo ' & " " &
!Narration & " " & !Dated & " " & !Debit & " " & !Credit
Set lstItem = lvwTransactions.ListItems.Add(, , !CustomerID)
lstItem.SubItems(1) = !AccountNo
lstItem.SubItems(2) = !Narration
lstItem.SubItems(3) = !Dated
lstItem.SubItems(4) = Format(!Debit, "#,###,##00.00")
lstItem.SubItems(5) = Format(!Credit, "#,###,##00.00")
lstItem.SubItems(6) = !Mode
lstItem.SubItems(7) = !CheckNO
lstItem.SubItems(8) = !Balance
lstItem.SubItems(9) = !Code
.MoveNext
Wend
End With
End Sub

Private Sub lvwTransactions_DblClick()
Call cmdEdit_Click
End Sub

Private Sub Option1_Click()
Frame1.Enabled = True
End Sub

Private Sub Option2_Click()
Frame1.Enabled = False
Set rsTemp = New ADODB.Recordset
rsTemp.Open "Select * from tblTransactions", cnBank, adOpenKeyset,
adLockOptimistic
With rsTemp
If .RecordCount > 0 Then
lvwTransactions.ListItems.Clear
Call LoadListView(rsTemp)
Else
MsgBox "Database Empty..", vbInformation
Exit Sub
End If
Close
End With
End Sub
```

```
!Narration = txtNarration.Text
!CheckNO = "N/A"
!Dated = txtDated.Value
!Debit = "00"
!Mode = "N/A"
!Credit = txtAmountWithdrawn.Text
!Balance = currBalance
.Update
End With

Set rsTemp = New ADODB.Recordset
rsTemp.Open "Select * FROM tblBalances WHERE CustomerID '" &
cboCustomerNo.Text & "'", cnBank, adOpenKeyset, adLockOptimistic
With rsTemp
!Balance = currBalance
.Update
.Requery
.Close
End With

'rsBalances.Open "UPDATE tblBalances SET Balance = '" & currBalance & "'
WHERE CustomerID='" & cboCustomerNo.Text & "'", cnBank, adOpenKeyset,
adLockOptimistic

End Sub

Private Sub cmdWithdraw_Click()
NewRecord = True
Call clear_Form_Controls(Me)
Call GenerateNewTransactCode
cboCustomerNo.SetFocus
MsgBox ("You are successfully Withdrawed,plz check your current balance,Thank
You....")
End Sub

Private Sub Form_Load()
Call connectDatabase

With rsCustomers
For X = 1 To RecordCount
cboCustomerNo.AddItem !CustomerID
MoveNext
Next X
End With
txtDated.Value = Date
End Sub
Public Sub GenerateNewTransactCode()
Dim LastNumber As Long, newNumber As Long
```

```
        If BOF = True And EOF = True Then
          lastnumber = 1000
        Else
          .MoveLast
          lastnumber = !TransactionID
        End If
        'Generate New Number
        newnumber = lastnumber + 1
        txtTransactionID.Text = newnumber
        End With
    End Sub

    Private Sub txtAccountNo_Change()
    txtNarration.SetFocus
    End Sub


    Private Sub txtAmountwithdrawn_KeyPress(KeyAscii As Integer)
    Call ValidNumeric(KeyAscii)

    End Sub

    Private Sub txtNarration_KeyPress(KeyAscii As Integer)
    KeyAscii = Asc(UCase$(Chr$(KeyAscii)))
    If KeyAscii = 13 Then txtAmountWithdrawn.SetFocus
    End Sub
    End With

    Set rsTemp = New ADODB.Recordset
    rsTemp.Open "Select * FROM tblBalances, WHERE CustomerID='" &
    cbo CustomerNo.Text & "'", cnBank, adOpenKeyset, adLockOptimistic
    With rsTemp
    If RecordCount > 0 Then
    lblBalance.Caption = !Balance
    Else
    Exit Sub
    End If
    End With
```

```
    Exit Sub
    End If

    If txtCheckNo.Text = "" Then
    MsgBox "Please Enter the Check No.", vbInformation
    txtCheckNo.SetFocus
    Exit Sub
    End If


    With rsDeposit
    If NewRecord = True Then .AddNew
    !TransactionID = txtTransactionID.Text
    !CustomerID = cboCustomerNo.Text
    !AccountNo = txtAccountNo.Text
    !Narration = txtNarration.Text
    !AmountDeposited = txtAmountDeposited.Text
    !Mode = txtMode.Text
    !CheckNO = txtCheckNo.Text
    !Dated = txtDated.Value
    .Update
    End With

    currBalance = (CCur(lblBalance.Caption) + CCur(txtAmountDeposited.Text))

    With rsTransactions
    .AddNew
    !CustomerID = cboCustomerNo.Text
    !AccountNo = txtAccountNo.Text
    !Narration = txtNarration.Text
    !CheckNO = txtCheckNo.Text
    !Dated = txtDated.Value
    !Debit = txtAmountDeposited.Text
    !Mode = txtMode.Text
    !Credit = "00"
    !Balance = currBalance
    Update
    End With

    Set rsTemp = New ADODB.Recordset
    rsTemp.Open "Select * FROM tblBalances WHERE CustomerID = '" &
    cboCustomerNo.Text & "'", cnBank, adOpenKeyset, adLockOptimistic
    With rsTemp
    !Balance = currBalance
    .Update
    .Close
    End With
```

```
rsBalances.Open "UPDATE tblBalances SET Balance = " & curBalance & _
WHERE CustomerID='" & cboCustomerNo.Text & "'", cnBank, adOpenKeyset, _
adLockOptimistic

End Sub

Private Sub Form_Load()
Call connectDatabase

With rsCustomers
For X = 1 To .RecordCount
cboCustomerNo.AddItem !CustomerID
.MoveNext
Next X
End With
txtDated.Value = Date
End Sub
Public Sub GenerateNewTransactCode()
  Dim lastnumber As Long, newnumber As Long
  'Check if there are records in the file
  With rsDeposit
  If .BOF = True And .EOF = True Then
    lastnumber = 1000
  Else
    .MoveLast
    lastnumber = !TransactionID
  End If
  'Generate New Number
  newnumber = lastnumber + 1
  txtTransactionID.Text = newnumber
  End With
End Sub

Private Sub optCash_Click()
txtCheckNo.Text = "N/A"
txtCheckNo.Locked = True
txtMode.Text = "CASH"

End Sub

Private Sub optCheque_Click()
txtMode.Text = "CHEQUE"
txtCheckNo.Text = ""
txtCheckNo.Locked = False
txtCheckNo.SetFocus
```

```
txtCheckNo.Text = "N/A"
txtCheckNo.Locked = True
txtOther.SetFocus
End Sub
Private Sub txtAccountNo_Change()
txtNarration.SetFocus
End Sub


Private Sub txtAmountDeposited_KeyPress(KeyAscii As Integer)
Call ValidNumeric(KeyAscii)
If KeyAscii = 13 Then optCash.SetFocus
End Sub

Private Sub txtCheckNo_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then cmdSave.SetFocus
End Sub

Private Sub txtMode_LostFocus()
If txtMode.Text = "" Then
MsgBox "Select the Mode of Transaction", vbInformation
End If
End Sub

Private Sub txtNarration_KeyPress(KeyAscii As Integer)
KeyAscii = Asc(UCase$(Chr$(KeyAscii)))
If KeyAscii = 13 Then txtAmountDeposited.SetFocus
End Sub

Private Sub txtOther_LostFocus()
txtMode.Text = txtOther.Text
End Sub

            .Update
            MsgBox "Pin Has been Successfull changed!", vbInformation
            Unload Me
            frmWel.fraTransaction.Visible = True
          Else
          MsgBox "Confirm Pin does not march with New Pin", vbInformation
            txtOldPin = ""
            txtNewPin = ""
            txtConfNewPin = ""
            txtOldPin.SetFocus
        End If
    End With
End Sub

Private Sub cmdExit_Click()
Unload Me
frmWel.fraTransaction.Visible = True
End Sub
```

```
End Sub

Private Sub Form_Load()
Call tblSecretCode
Call OpenDB
End Sub
```

Transaction

```
Private Sub cmd1000_Click()
txtWAmtDpt = "1000"
WithD = txtWAmtDpt
Call WithDraw
End Sub

Private Sub cmd2000_Click()
txtWAmtDpt = "2000"
WithD = txtWAmtDpt
Call WithDraw
End Sub

Private Sub cmd3000_Click()
txtWAmtDpt = "3000"
WithD = txtWAmtDpt
Call WithDraw
End Sub

Private Sub cmd4000_Click()
txtWAmtDpt = "4000"
WithD = txtWAmtDpt
Call WithDraw
End Sub

Private Sub cmd5000_Click()
txtWAmtDpt = "5000"
WithD = txtWAmtDpt
Call WithDraw
End Sub

Private Sub cmdOthers_Click()
MsgBox ("ok You can enter your choice")
fraWamt.Visible = False
fraAmt.Visible = False
lblAmt.Visible = True
cmdOthers.Visible = False
txtWAmtDpt.Visible = True
txtSecCode.Visible = True
End Sub
```

```
Private Sub cmdWClose_Click()
Unload Me
Unload frmStatement
frmStatement.Hide
frmWel.fraTransaction.Visible = True
End Sub
Private Sub WithDraw()
With rsT
    .AddNew
    .Fields(0) = txtWDate
    .Fields(1) = ActVariable
    .Fields(5) = "Withdrawal"
    .Fields(6) = WithD
    .Fields(7) = PrevBal
    .Fields(8) = txtWCrrntBal
    .Update
End With
  With rsC
     .Fields(4) = txtWCrrntBal
     .Update
     Unload Me
     Me.Hide
  End With
Unload frmWithD
frmWithD.Hide
frmWaitWithD.Show vbModal
MsgBox "The Sum of: " & WithD & " Has been Deducted from your account",
vbInformation
frmCash.cmdCancel.Visible = False
frmCash.lblDisplay.Visible = False
Load frmCash
frmCash.Show vbModal
End Sub

Private Sub cmdWDraw_Click()
Dim strMessage As String
Dim strWithAmt As String
With rsT
    AddNew
    Fields(0) = txtWDate
    Fields(1) = ActVariable
    Fields(5) = "Withdrawal"
    Fields(6) = txtWAmtDpt
    Fields(7) = PrevBal
    Fields(8) = txtWCrrntBal
    Update
```

```
            Unload Me
            Me.Hide
        End With
        frmWaitWithD.Show vbModal
        Unload Me
        MsgBox " Your Toatl amount  Has been Deducted from your account,plz wait...",
        vbInformation
        frmCash.cmdCancel.Visible = False
        frmCash.lblDisplay.Visible = False
        Load frmCash
        frmCash.Show vbModal
        End Sub

        Private Sub Form_Load()
        Call Master
        Call Trans
        Call Crent
        txtWDate = Date
        With rsC
          .Find "AccountNumber =" & ActVariable & ""
          If .EOF Then
            MsgBox "Account does not Exist! Please contact customer services".
      vbInformation, "A T M Service....."
          Else
            PrevBal = rsC.Fields(4)
            txtWAcctTyp = rsC.Fields(1)
            txtWSname = rsC.Fields(2)
            txtWFname = rsC.Fields(3)
            txtWPrevBal = PrevBal
          End If
        End With
        End Sub

        Private Sub txtWAmtDpt_Change()
        txtWCrntBal = Val(txtWPrevBal) - Val(txtWAmtDpt)
        End Sub
        Private Sub ClearWBoxes()
          txtWDate = ""
          txtWAcctTyp = ""
          txtWAcctNum = ""
          txtWSname = ""
          txtWFname = ""
          txtWAmtDpt = ""
          txtWPrevBal = ""
          txtWCrntBal = ""
        End Sub
```

```
Private Sub cmdSearch_Click()
If cmdSearch.Caption = "&View Statement" Then
   ActVariable = txtStAcctNum.Text
   Unload Me
   Load frmWait
   frmWait.Show
   frmWelfraTransaction.Visible = False
   Exit Sub
 ElseIf cmdSearch.Caption = "&Process" Then
   ActVariable = txtStAcctNum
   Call Crent
   With rsC
     .Find "AccountNumber =" & ActVariable & ""
   If .EOF Then
     MsgBox "Account does not Exist! Please contact customer services",
vbInformation, "A T M Service....."
     txtStAcctNum = ""
     txtStAcctNum.SetFocus
     cmdSearch.Caption = "&Process"
   Else
     Unload Me
     Me.Hide
     Load frmWithD
     frmWithD.Show vbModal
   End If
End With
Else
  ActVariable = txtStAcctNum
  Call Crent
  With rsC
    .Find "AccountNumber =" & ActVariable & ""
  If .EOF Then
    MsgBox "Account does not Exist! Please contact customer services",
vbInformation, "A T M Service....."
    txtStAcctNum = ""
    txtStAcctNum.SetFocus
    cmdSearch.Caption = "&Inquire"
  Else
    Unload Me
    Me.Hide
    [illegible].Visible = False
    [illegible].cmdOK.Visible = False
    [illegible]
    [illegible]
  [illegible]
  [illegible]
[illegible]
```

```
frmWel.fraTransaction.Visible = True
frmWel.fraAcctType.Visible = False
frmWel.fraSecretCode.Visible = False
End Sub
```

5.2 SYSTEM IMPLEMENTATION AND TESTING

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

5.2.1. SYSTEM TESTING

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified?. Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Validation : Are we doing the right job?
Verification : Are we doing the job right?

Software testing should not be confused with debugging. Debugging is the process of analyzing and localizing bugs when software does not behave as expected. Although the identification of some bugs will be obvious from playing with the software, a methodical approach to software testing is a much more thorough means for identifying bugs. Debugging is therefore an activity which supports testing, but cannot replace testing.
Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis

looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work toward the integration of the entire computer-based system. Nothing is complete without testing. Testing objectives include the following rules that can serve well as testing objectives. They are:

Testing is a process of executing a program with the intend of finding an error A good test case is one that has high possibility of finding an undiscovered error A successful test is one that uncovers an undiscovered error

If a testing is conducted successfully according to the objectives as stated above it would uncovered errors in the software also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- Unit testing
- Integration Testing
- Data validation Testing
- Output Testing

UNIT TESTING

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered errors is limited by the constrained scope established for unit testing. Unit testing is white-box oriented, and the step can be conducted in parallel

examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified.

INTEGRATION TESTING

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop.

After unit testing in Sell-Soft System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover differences in program structures were removed and a unique program structure was evolved.

VALIDATION TESTING OR SYSTEM TESTING

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System testing.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

OUTPUT TESTING OR USER ACCEPTANCE TESTING

[illegible] user acceptance [illegible] in touch with [illegible] whenever required. [illegible] the following [illegible]

- Input Screen Designs.
- Output Screen Designs.
- Online message to guide the user and the like.

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which the errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

5.3. TRAINING

Once the system is successfully developed the next important step is to ensure that the administrators are well trained to handle the system. This is because the success of a system invariably depends on how they are operated and used. The implementation depends upon the right people being at the right place at the right time. Education involves creating the right atmosphere and motivating the user. The administrators are familiarized with the run procedures of the system, working through the sequence of activities on an ongoing basis.

Implementation is the state in the project where the theoretical design is turned into a working system. By this, the users get the confidence that the system will work effectively. The system can be implemented only after through testing.

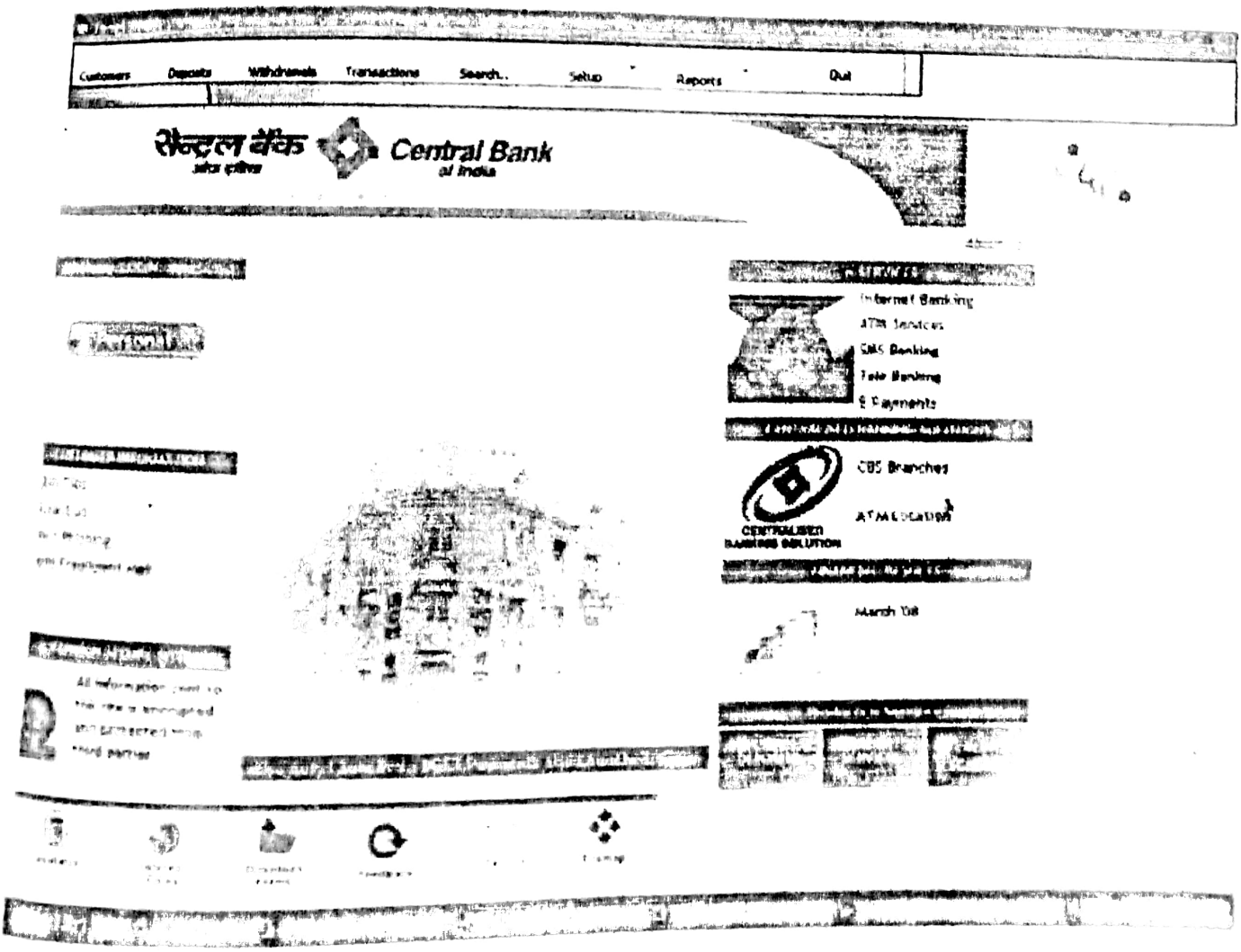
The systems personnel check the feasibility of the system. The actual data were inputted to the system and the working of the system was closely monitored. The master option was selected from the main menu and the actual data were input through the corresponding input screens. The data movement was studied and found to be correct queries option was then selected and this contains various reports. Utilities provide various data needed for inventory was input and the module was test run. Satisfactory results were obtained. Reports related to these processes were also successfully generated. Various input screen formats are listed in the appendix.

Implementation walkthroughs ensure that the completed system actually solves the original problem. This walkthrough occurs just before the system goes into use and it should include careful review of all manuals, training materials and system documentation. Again, users, the analyst and the members of the computer services staff may attend this meeting.

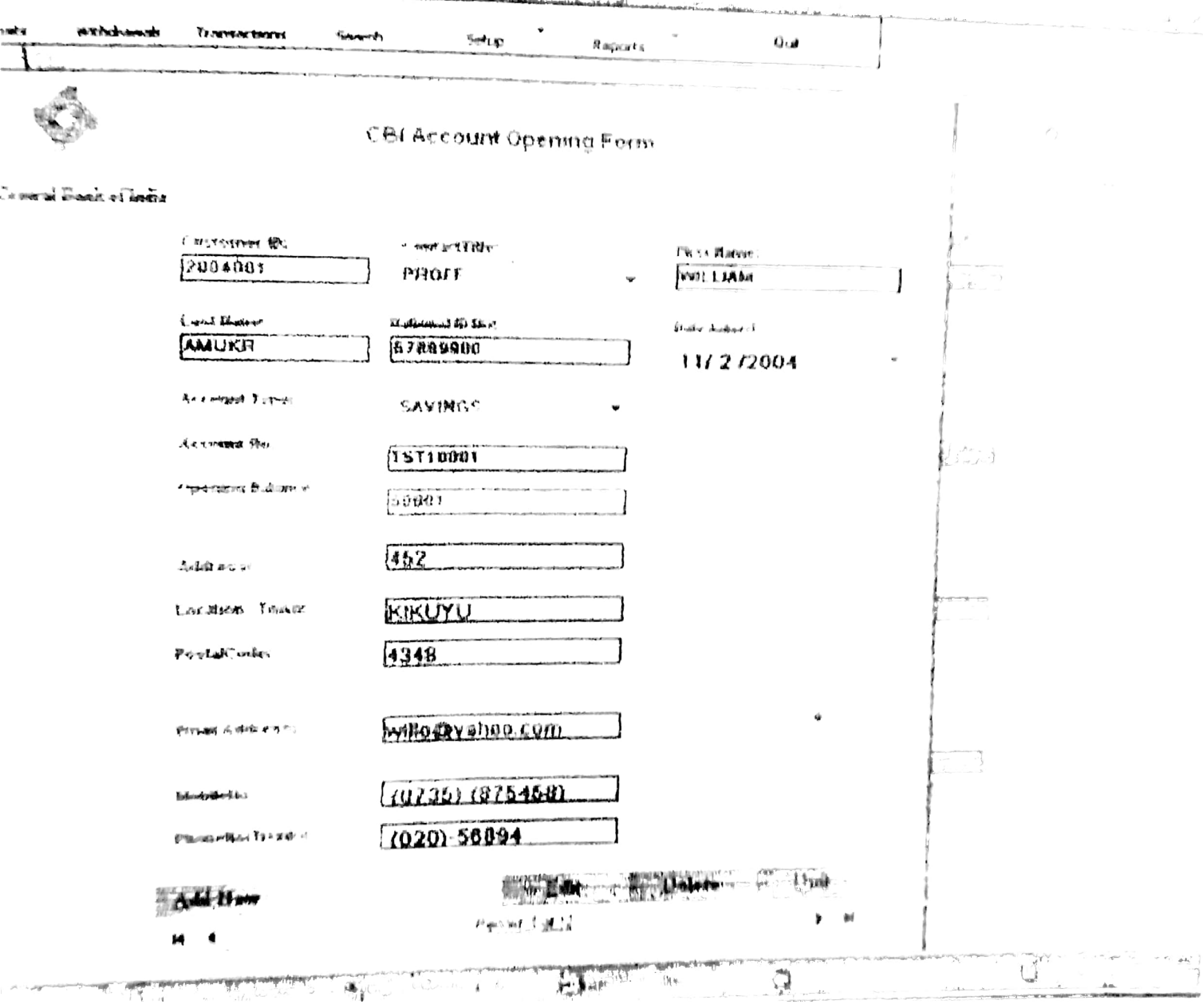
APPENDIX

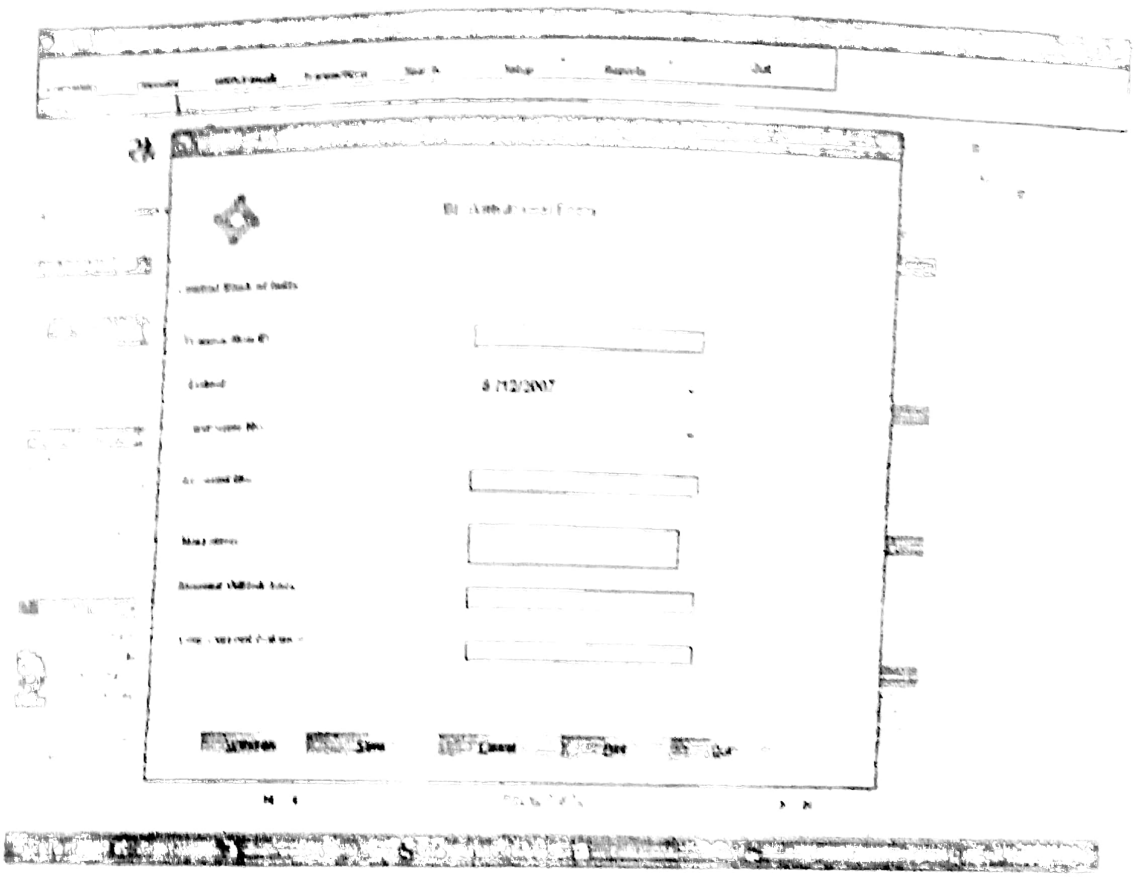
SCREEN SHOTS

Login

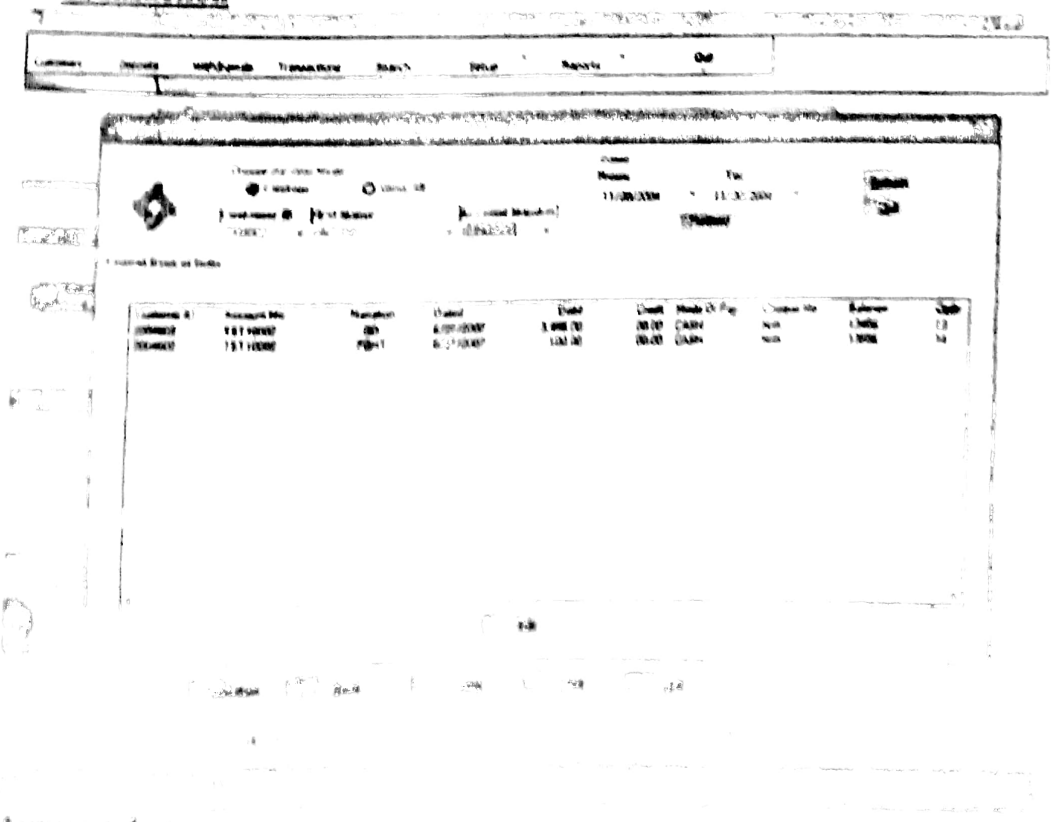


Opening Form

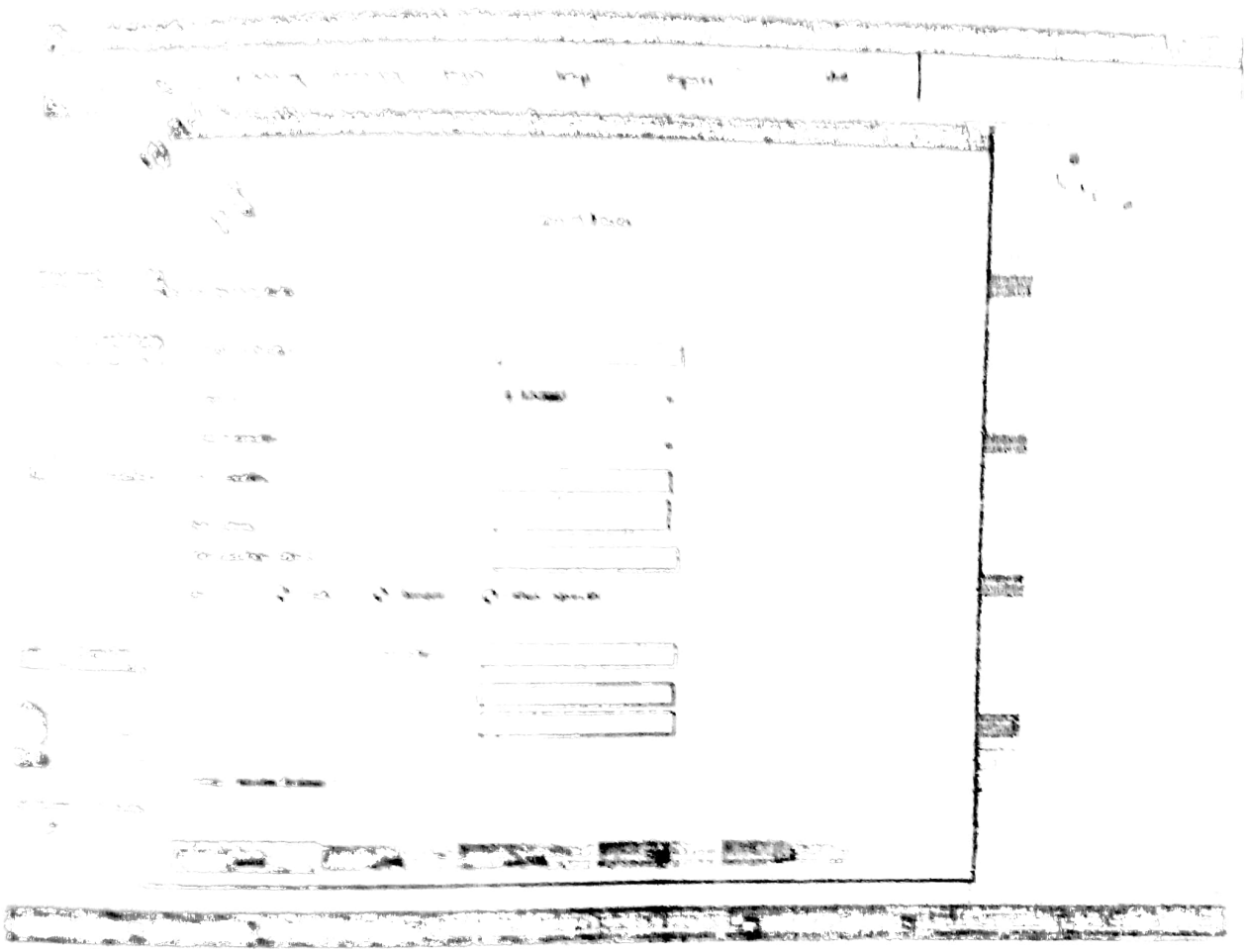




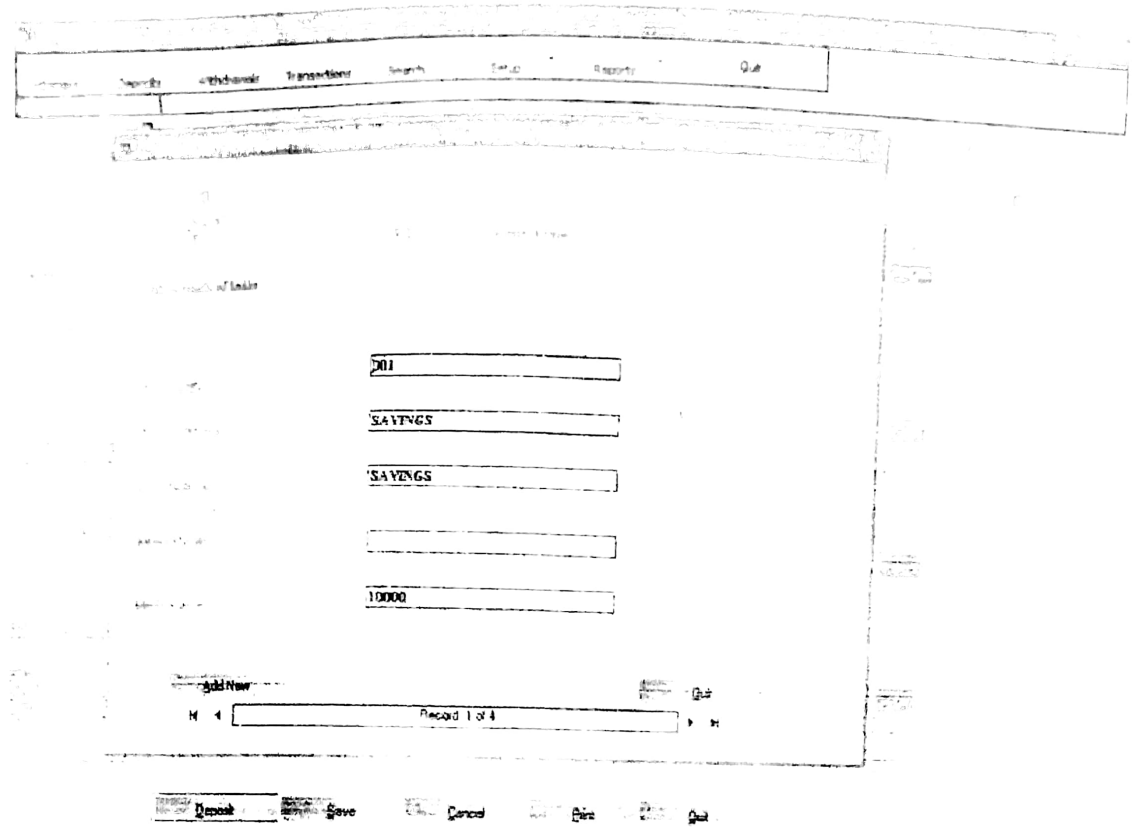
Transaction



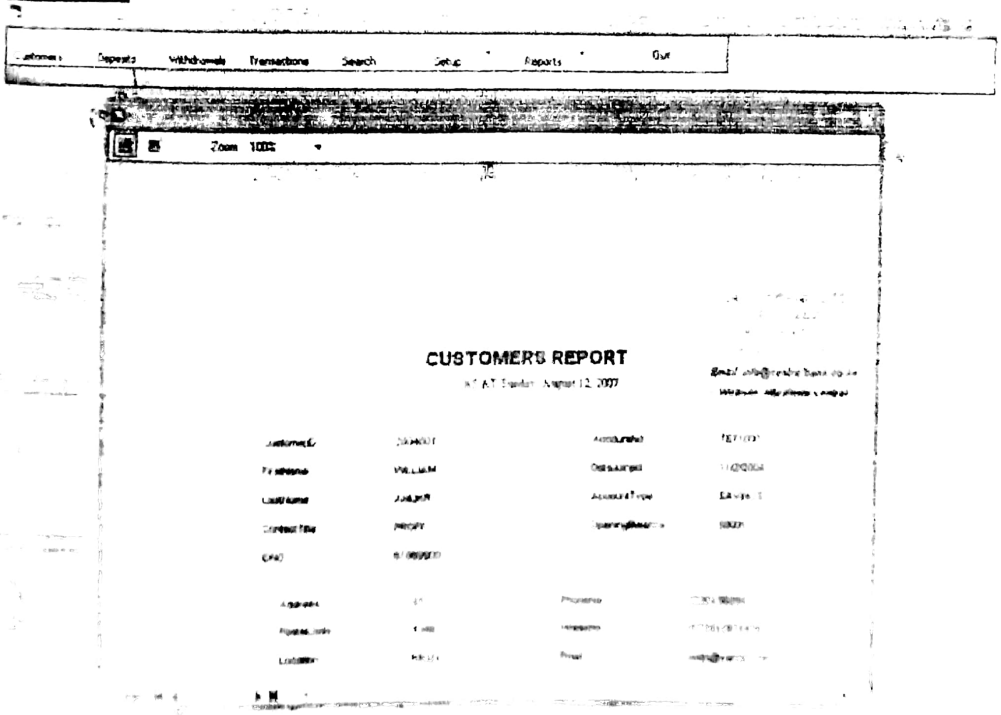
Account Type



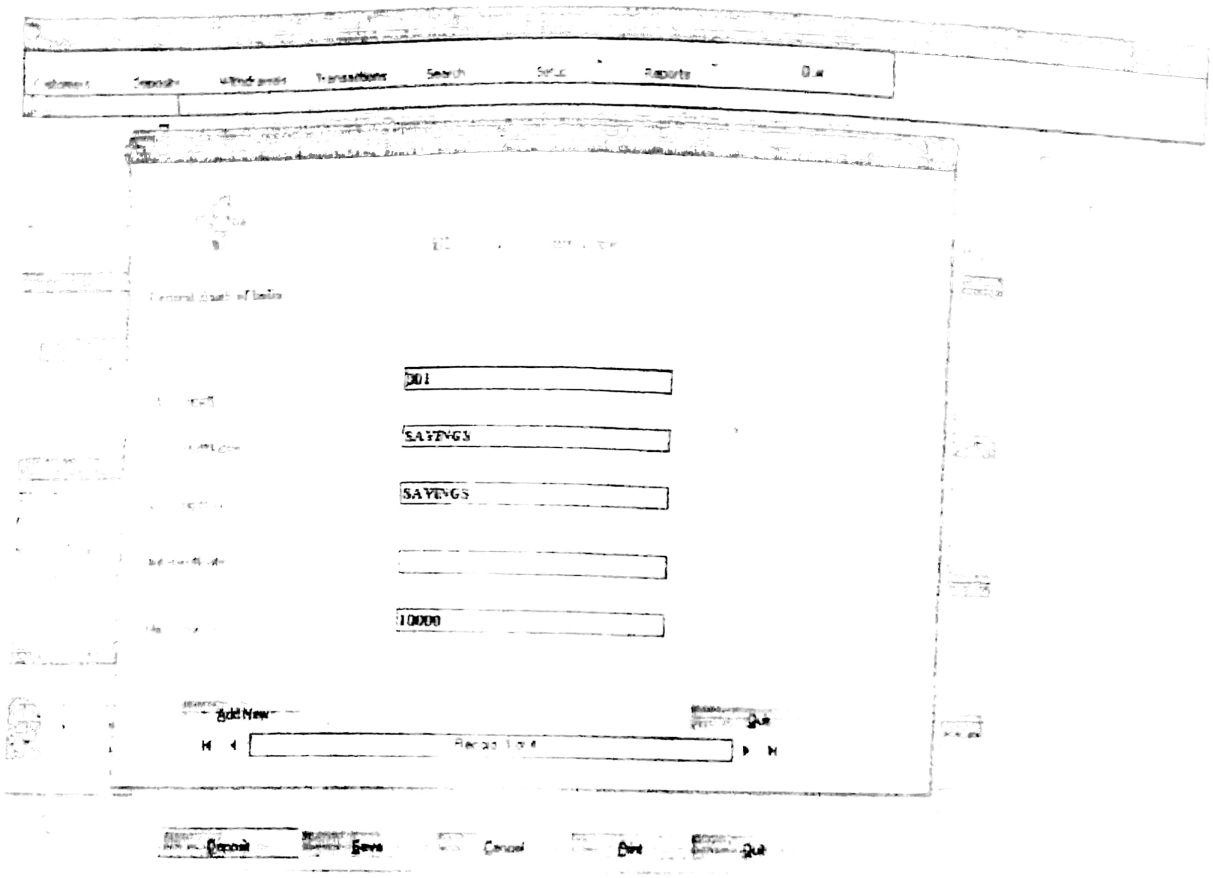
Correct!



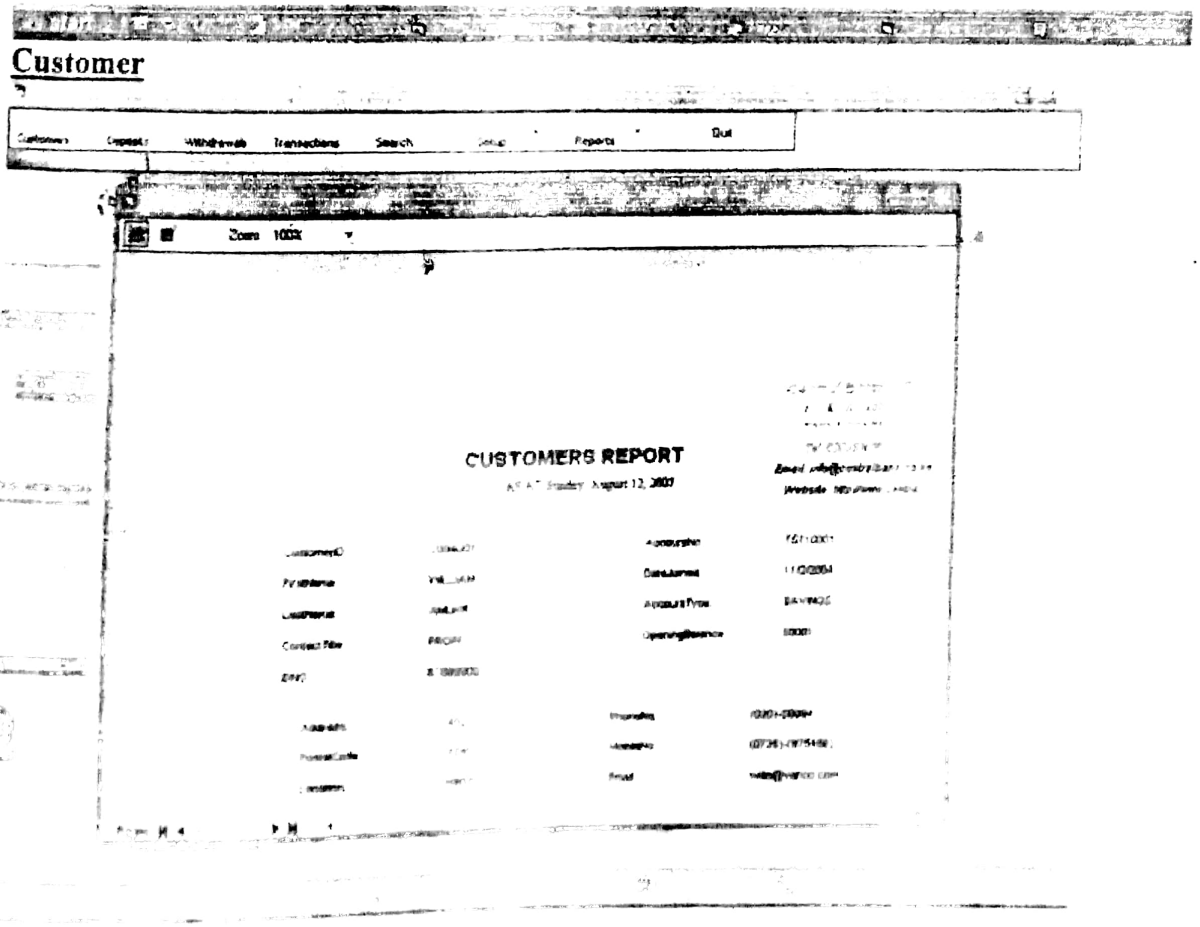
Customer

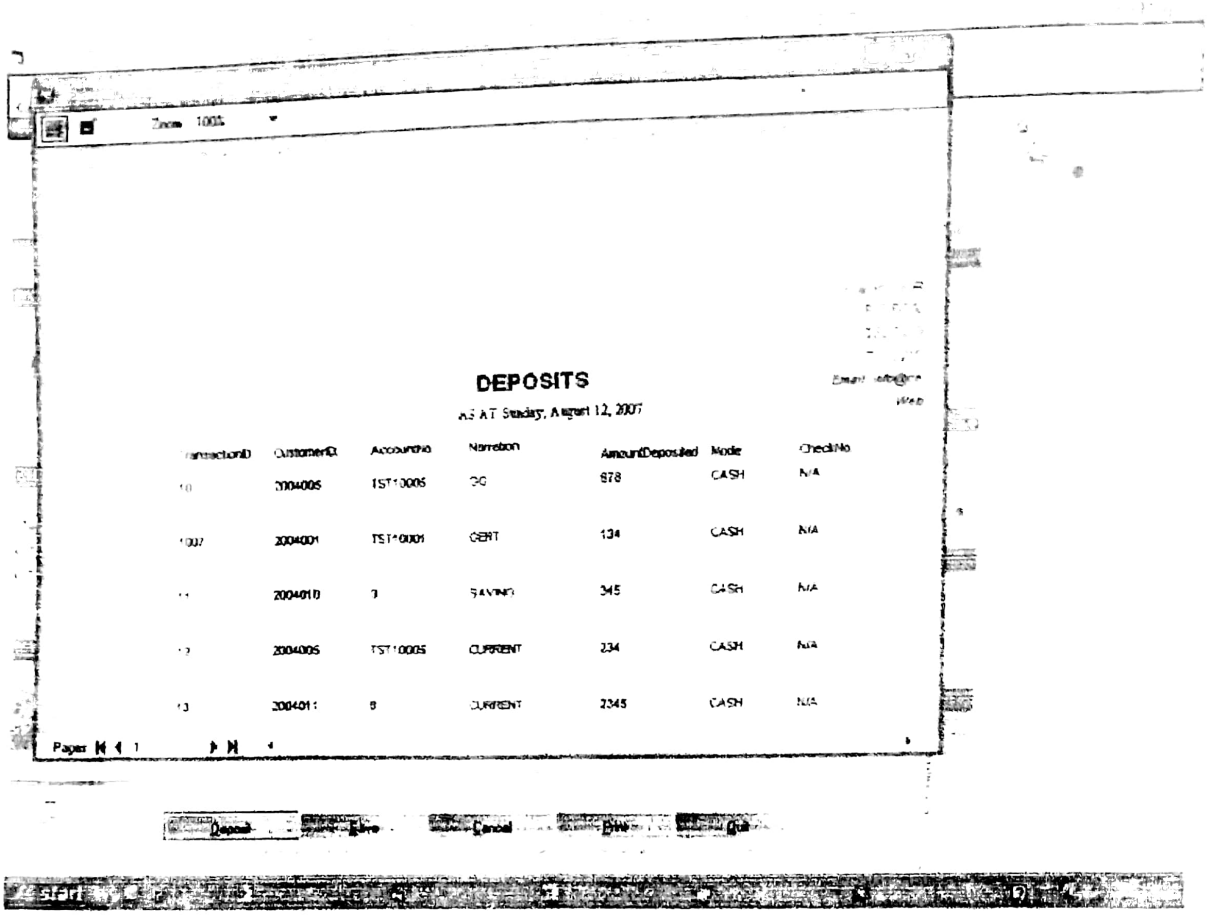


Deposit Transaction

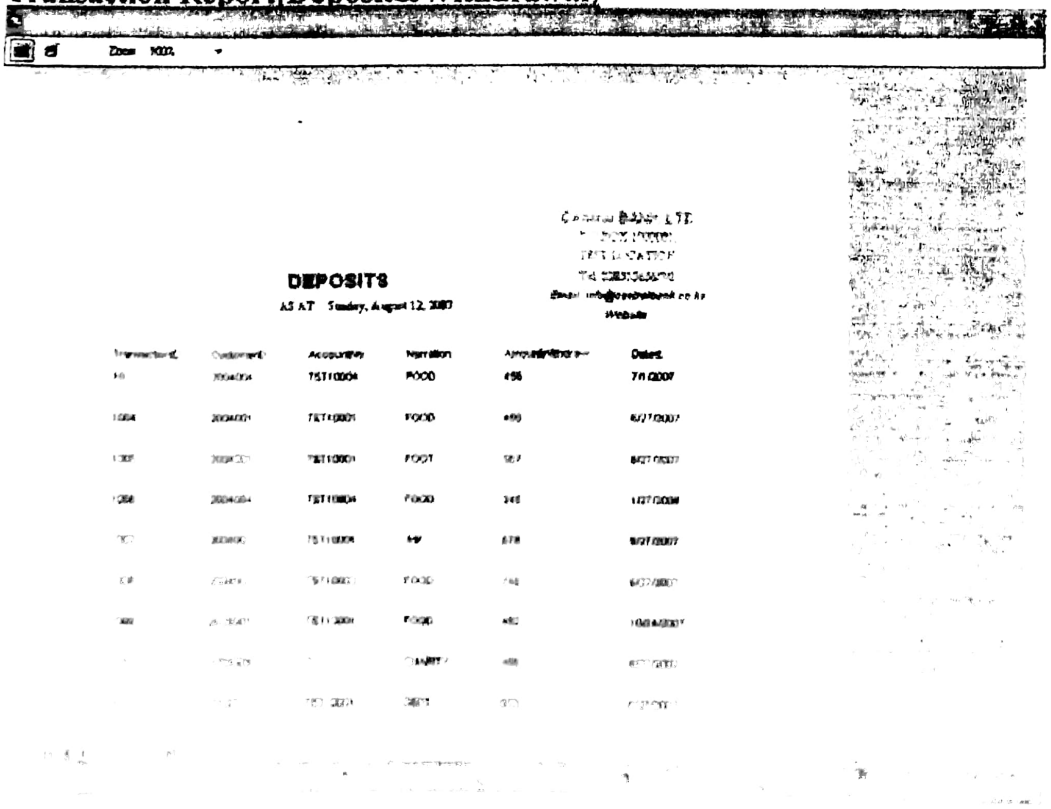


Customer





Transaction Report(Deposit&Withdrawal)



CONCLUSION

BOOKS:

1. Charles Hampfed (2000) 'Visual Basic' University of Toronto
2. Herbert Schildt (2000) 'Visual Basic 6.0' Tata McGraw Hill
3. John Zukowski (2000) 'Visual Basic 6.0' 'BPB Publications
4. Jamie Jaworsky 'Visual Basic 6.0'Techmedia
5. Stefen Denninger 'Visual Basic 6.0'Author's Press
6. Ian Somerville 'Software engincering'
7. Rajeev mall 'Software engineering'
8. Elmasri Navathe 'Fundamentals of database systems'

ONLINE REFERENCE:

www.w3schools.com
www.theserverside.com
www.visual.com