**ACKNOWLEDGEMENT**

I, MohammadSajid. A. Ansari student of Rizvi College Of ASC, have done this project as a part of T.Y.BSc (CS) syllabus, Semester 6th. This project uses the technology studied by us throughout the course offered by the University of Mumbai.

First, I would like to express my gratitude and sincere thanks to Mr. Qasim who helped me throughout the project development.

I would also like to acknowledge our Head Of Department Mr. Mohammed Razi for his support and help throughout the project.

I would also like to express my sincere thanks to all the staff members of BSc (CS) for making sure we have the support needed to complete the project on time.

Last but not the least; I would like to thank my family and specially my mom and my friends who have helped me during the course of the project by providing their precious time and undivided moral support.

**INDEX**

|  |  |  |
| --- | --- | --- |
| Sr.No | Topic | Page No |
| 1 | Introduction | 03 |
| 2 | Description Of Current System | 04 |
| 3 | Limitation Of Current System | 04 |
| 4 | Proposed System | 05 |
| 5 | Scope Of The System | 05 |
| 6 | Technology Used | 06 |
| 7 | Entity Relationship Diagram(ERD) | 06 |
| 8 | Activity Diagram | 07 |
| 9 | Class Diagram | 08 |
| 10 | State Diagram | 09 |
| 11 | Use Case Diagram | 10 |
| 12 | Event Table | 11 |
| 13 | ERD to Tables | 12-16 |
| 14 | Data Flow Diagram | 17 |
| 15 | Database Design | 18-19 |
| 16 | Forms Layout | 20-33 |
| 17 | Validating Forms/Applications | 34-37 |
| 18 | Coding | 39-59 |
| 19 | Modules Design | 60 |
| 20 | Software Testing | 61-62 |
| 21 | Future Enhancement | 62 |
| 22 | Conclusion | 62 |
| 23 | Nomenclature | 63-64 |
| 24 | Gantt Chart | 65 |
| 25 | Bibliography | 66 |

**INTRODUCTION**

**‘BANANA CO’ is the visual organization for which I have chosen to build a software because basically there are many problems in present system.**

**‘BANANA CO’ is an organization which provides customer with cable networks with different channels.**

**In this System there is an Admin who can add co-admins, Employees as well as Customer’s. Admin will manage the entire System, Admin has the full privilege to add, edit and delete the above mentioned.**

**System provides a number of packages, packages contains number of channels. Each Package has some mutual and different channels. So User can select the appropriate package that he/she wants.**

**System is applicable for generating Customer Bills and keeping track of it. System can generate Bills of maximum 12 months.**

**DISCRIPTION OF SYSTEM**

**This organization does all their work i.e. storing client’s data and activated packages in a book which are done manually.**

**In this system cable operator often complaints that it takes more time to enter the data of all customers.**

**But even after all the trouble the cable operator is managing all the records from past many years.**

**This organization provides channels and packages of channels to customer and according to which their monthly bills are generated and even the generation of bills are done manually which takes a long time for generating bill of all customers.**

**The record of monthly expenses of customer are also stored manually that whether customers have paid or not paid their fees.**

**LIMITATIONS OF CURRENT SYSTEM**

**In present system all the work done are manually stored which is the main problem due to which present system has certain limitations.**

1. **TIME CONSUMPTION: As all the work done i.e. data of customers are manually stored it takes a lot of time to enter all the data which may cause wastage of time.**
2. **DATA LOSS: There is possibility of the data loss i.e. sometimes the data book may be misplaced that may cause data loss which were stored in it.**
3. **MAINTAINING LARGE AMOUNT OF RECORDS IS DIFFICULT: A cable operator faces severe difficulties in maintaining records of clients and even entering the data because it is very difficult for an operator to hold and enter records manually.**
4. **MAKING BILLING PROCESS IS DIFFICULT: Cable operators often complain that making bills of many customers at a time is difficult and may even cause a lot of confusion.**

**PROPOSED STSYEM AND IT’S ADVANTAGES**

**As all work are manually done i.e. generating bill and clients records in present system I have decided to develop a software so it can be easy for operator to use and stored data in it.New system will contain customer’s records which channels or packages they have activated and according due to which bills will be generated. This system is less time consuming and easy to understand.**

**ADVANTAGES:-**

1. **MAINTAINING LARGE AMOUNT OF DATA IS POSSIBLE: In this new system we can maintain a large amount of data of customer which can be done in less time.**
2. **LESS TIME CONSUMPTION: This new system is less time consuming because no manual work is done all data are stored and secured in the system.**
3. **DATA SECURITY: In this system data are secured and no possibility of data loss will arrive.**
4. **EASY TO OPERATE AND UNDERSTAND: This new system is easy for an operator to operate and even understand the basic functions of the system.**
5. **MANAGING CUSTOMER DETAILS WITH BILLING IS EASY: This system will manage customer details and billing so that no confusion can occur with different customer bills.**

**SCOPE OF THE SYSTEM**

**To maintain entire operations of the cable operator to be automated**

1. **To maintain all of the major operations of the cable operator to be automated**
2. **This system is interactive with the database. As all the data is stored in database. And the operation done in frontend also effects the data in the backend.**

**TECHNOLOGY USED**

* **Front End : Vb.net**
* **Back End : MS-SQL Server 2008**

**ENTITY RELATIONSHIP DIAGRAM (ERD)**

DELIVERS

RECIEPT

ASKS FOR CONNECTION

BILL

IN

CONFIRMS

SUGGESTS

CHANNEL LISTS

ADMIN

CUSTOMER

WORKS WITH CONNECTION

PAYS

GENERATES

**ACTIVITY DIAGRAM**

**CUSTOMEREMPLOYEE SYSTEM**

CUSTOMER MAKES PAYMENT

ENTER DETAILS IN DATABASE

RECEIVE PAYMENT

DELIVER BILL TO CUSTOMER

GENERATE BILL

CONNECTION DETAILS

PACKAGE CONFIRMATION DETAILS

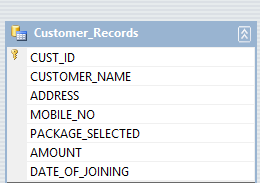
CHOOSES PACKAGES

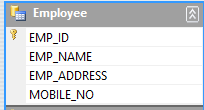
ASKS FOR CONNECTION

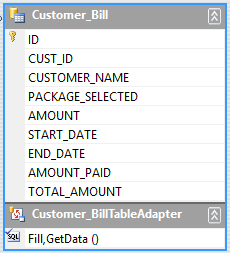
IS VALID

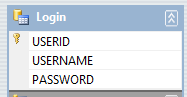
ADMIN LOGIN

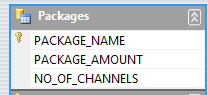
**Class Diagram**











**STATE DIAGRAM**

**INITIAL STATE OF OBJECT INTERMEDIATE STATE TRANSACTION STATE**

**IDLE**

**CUSTOMER CHOOSES PACKAGES**

**CUSTOMER ASKS FOR CONNECTION**

**EMPLOYEE GIVES CONNECTION DETAILS**

**INITIAL STATE**

**EMPLOYEE GIVES CONNECTION CONFIRMATION TO CUSTOMER**

**ABNORMAL**

**EXIT(FAILURE)**

**FINAL STATE**

SESSION

**PAYMENT DETAILS**

COMPLETE

**USE-CASE DIAGRAM**

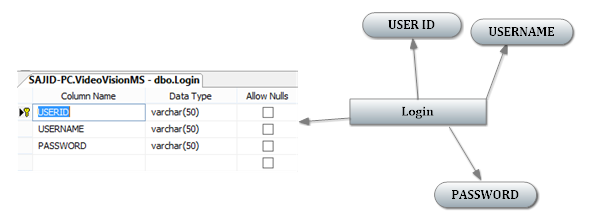
CUSTOMER

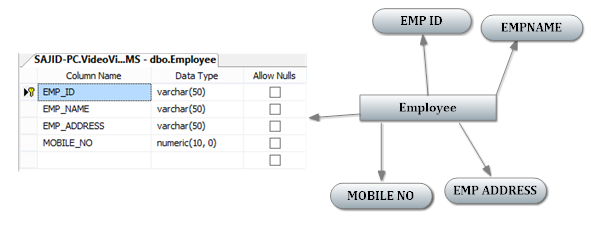
OPERATOR

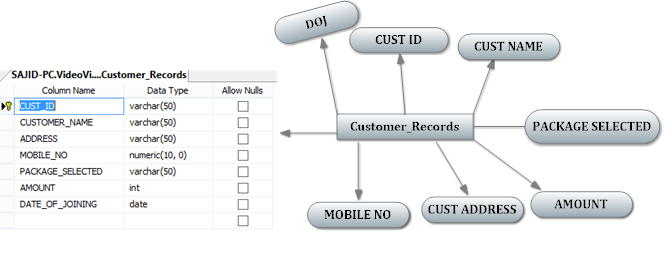
**EVENT TABLE**

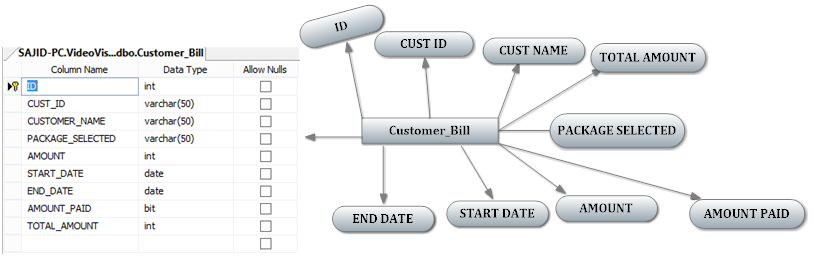
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EVENT** | **TRIGGER** | **SOURCE** | **ACTIVITY** | **RESPONSE** | **DESTINATION** |
| **1) PLACE NEW**  **CONNECTION** | **ENTER**  **CONNECTION**  **DETAILS** | **CUSTOMER** | **CONNECTION**  **PLACING** | **CONNECTION**  **PLACED** | **ADMIN** |
| **2) CUSTOMER**  **CHOOSES**  **PACKAGES** | **PACKAGES**  **DETAILS** | **CUSTOMER** | **PACKAGE**  **SELECTION** | **PACKAGE**  **SELECTED** | **ADMIN** |
| **3)CONNECTION**  **CONFIRMATION** | **CONNECTION**  **DETAILS** | **EMPLOYEE** | **CONNECTION**  **DONE** | **CONNECTION**  **CONFIRMATION** | **CUSTOMER** |
| **4) BILL**  **GENERATION** | **CREATING**  **BILL** | **EMPLOYEE** | **CONNECTION**  **DONE** | **CONNECTION**  **CONFIRMATION** | **CUSTOMER** |
| **5) IF A**  **CUSTOMER**  **WANTS TO**  **CHANGE THE**  **PACKAGES** | **PACKAGE**  **DETAILS** | **CUSTOMER** | **PACKAGE SELECTION WHICH CUSTOMER WANTS TO HAVE** | **PACKAGE**  **SELECTED** | **ADMIN** |

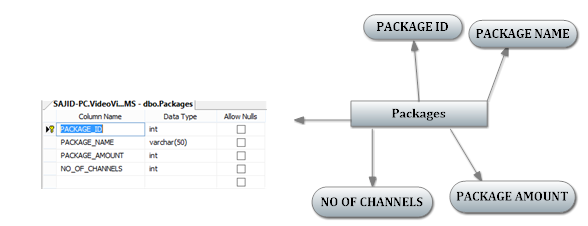
**ERD INTO TABLES**



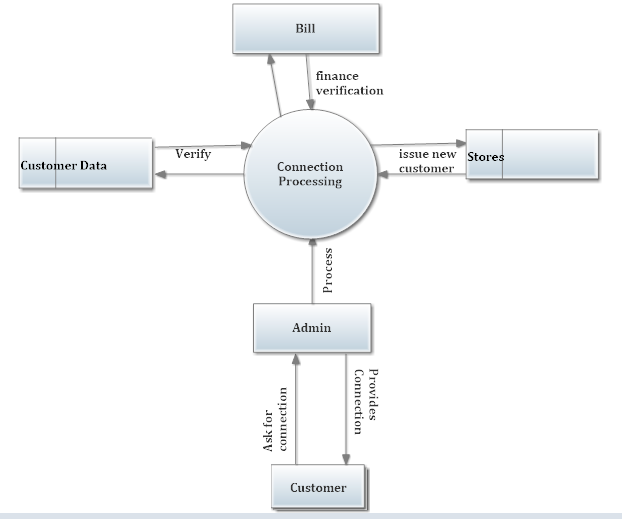






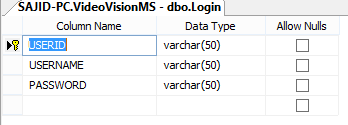


**DFD Diagram**

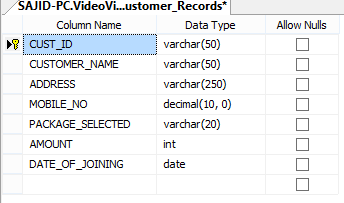


**DATABASE DESIGN**

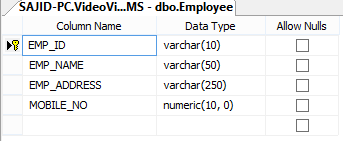
**Login**



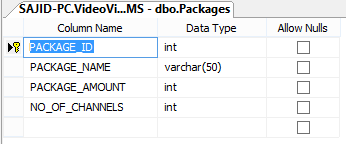
**Customer\_Records**



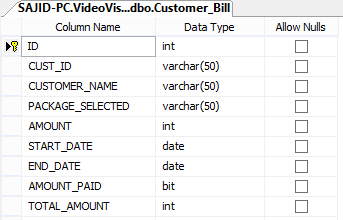
**Employee**



**Packages**

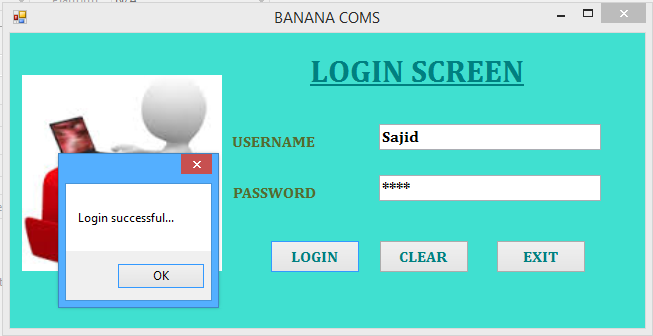


**Customer\_Bill**

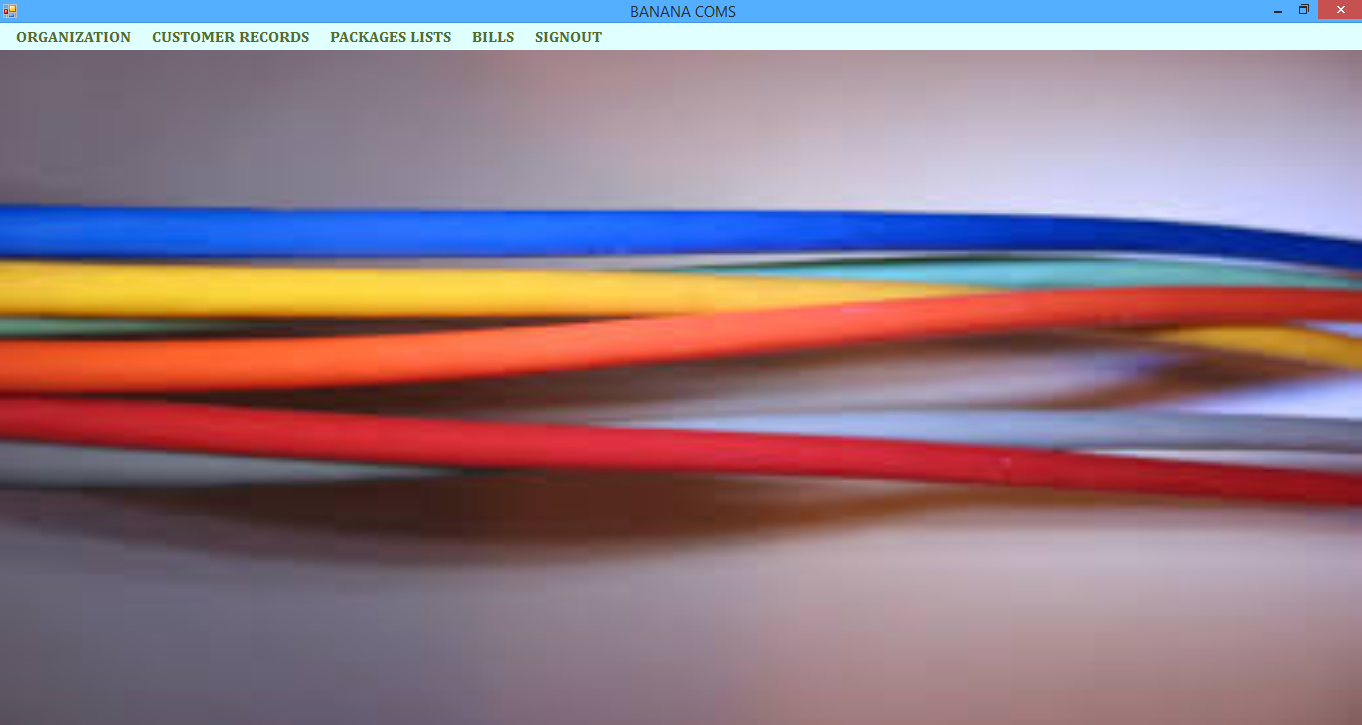


**FORMS LAYOUT**

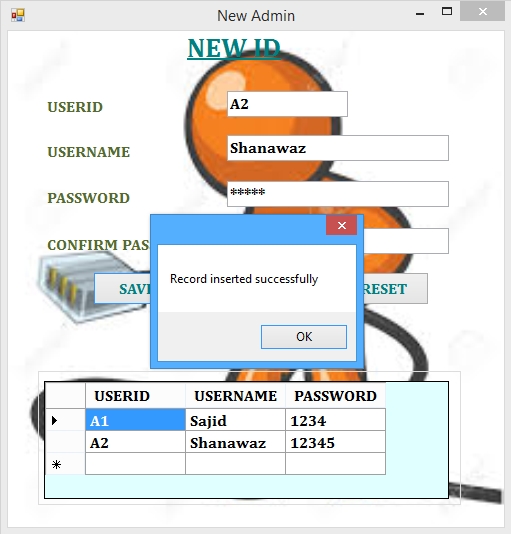
**LOGIN SCREEN**



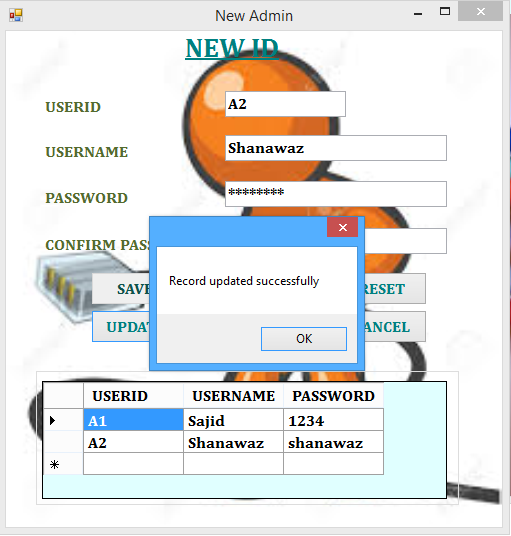
**MDI FORM (Main Page)**



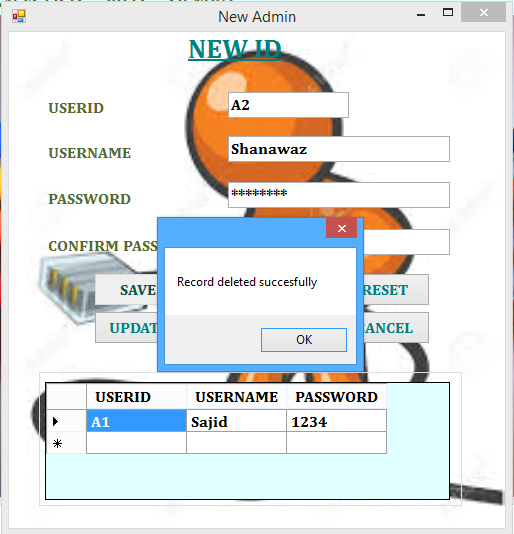
**NEW ADMIN ID**



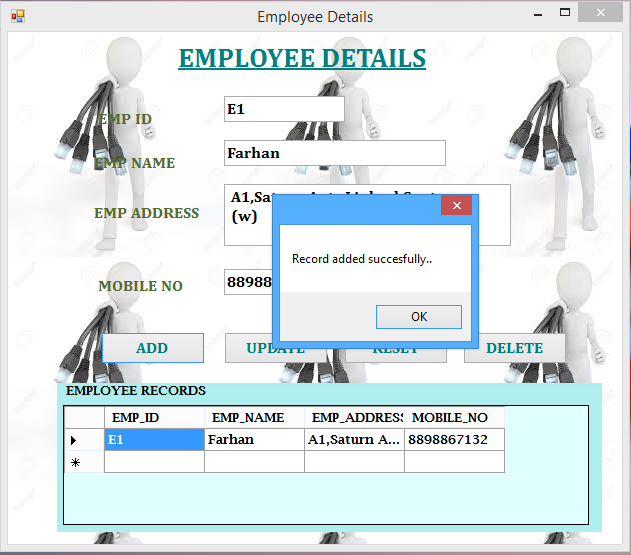
**ADMIN (update)**



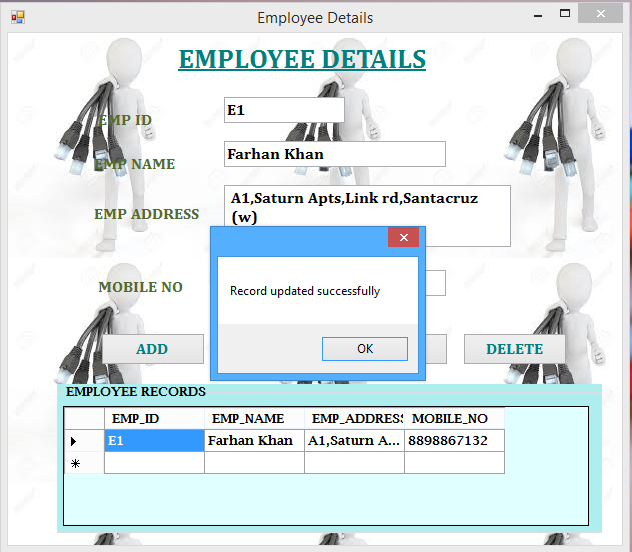
**ADMIN (delete)**



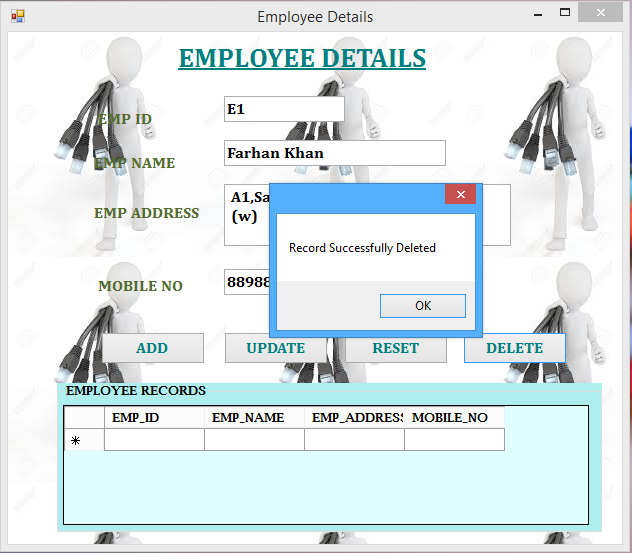
**NEW EMPLOYEE ID**



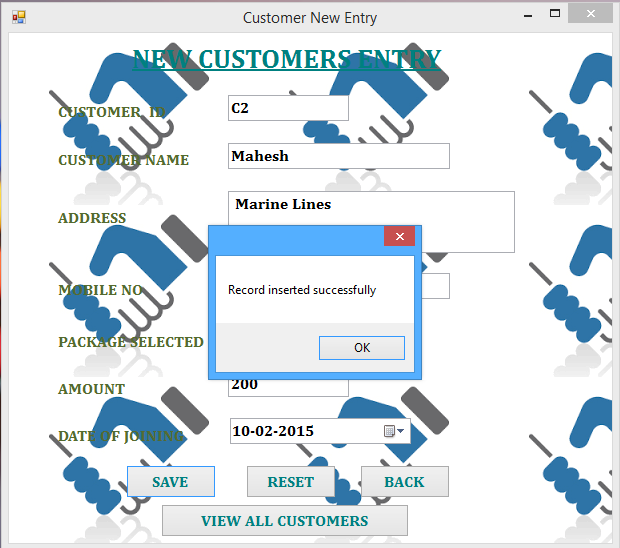
**EMPLOYEE (update)**



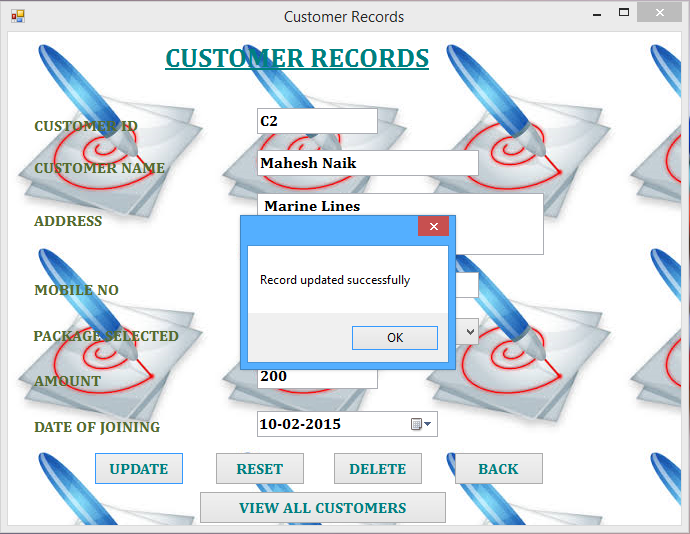
**EMPLOYEE (delete)**



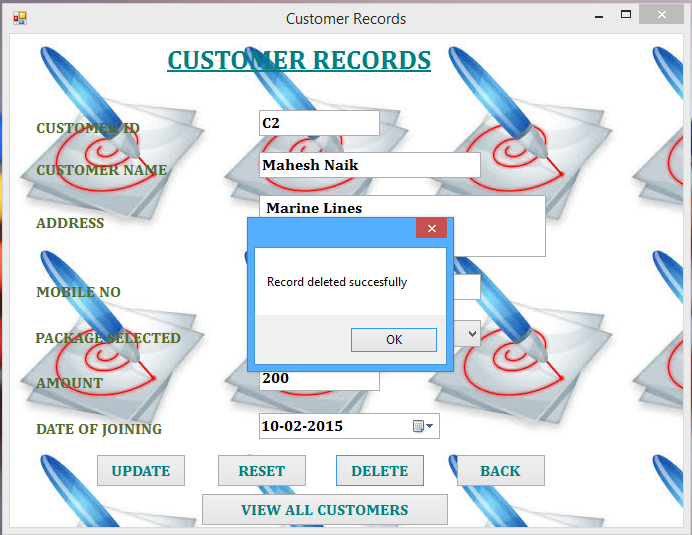
**NEW CUSTOMER ENTRY**



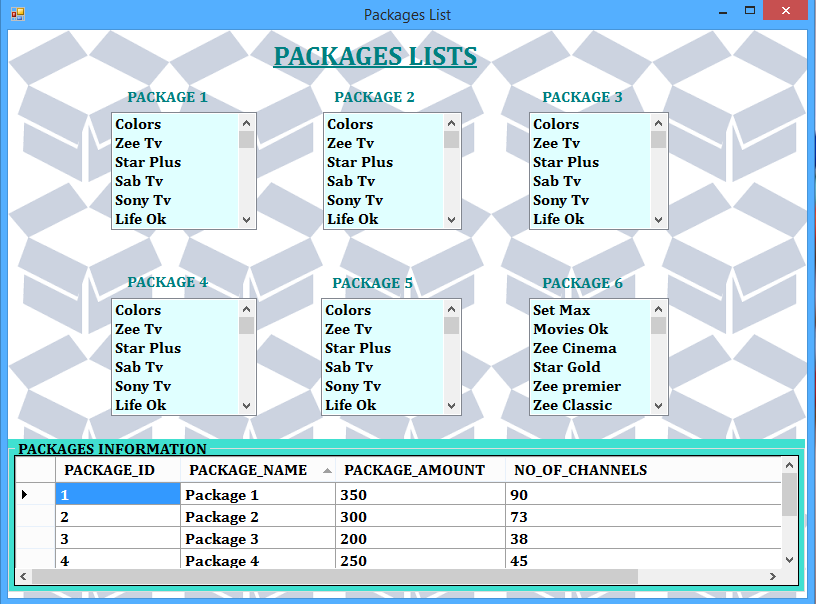
**CUSTOMER (update)**



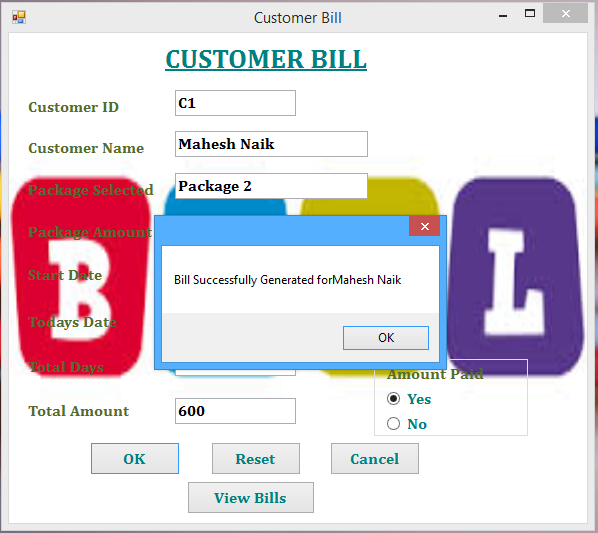
**CUSTOMER (delete)**

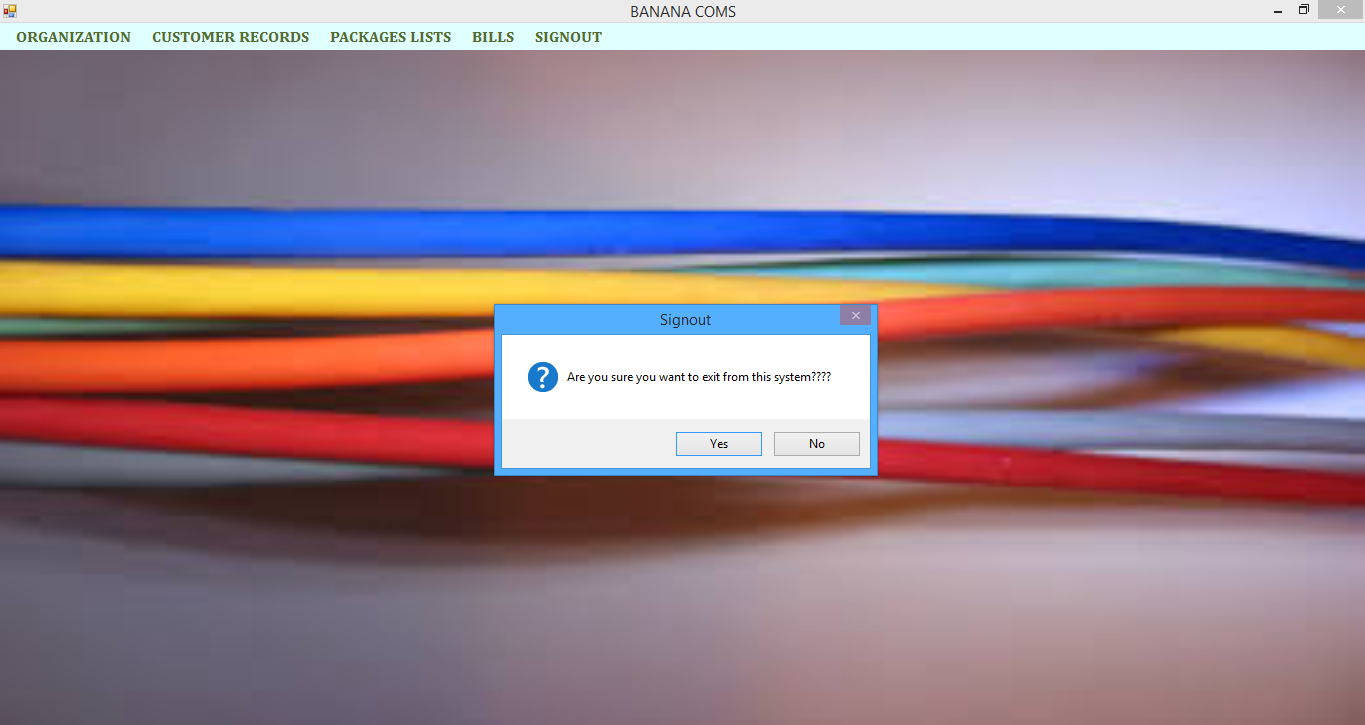


**PACKAGES**



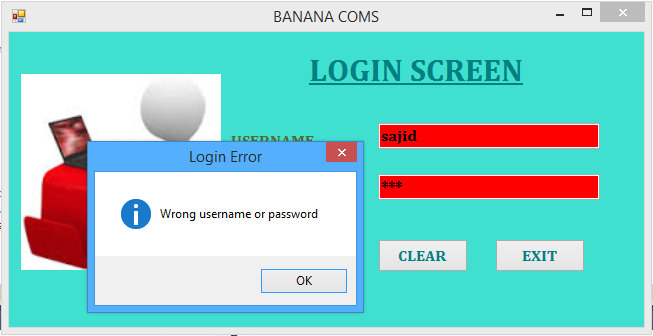
**CUSTOMER BILL**

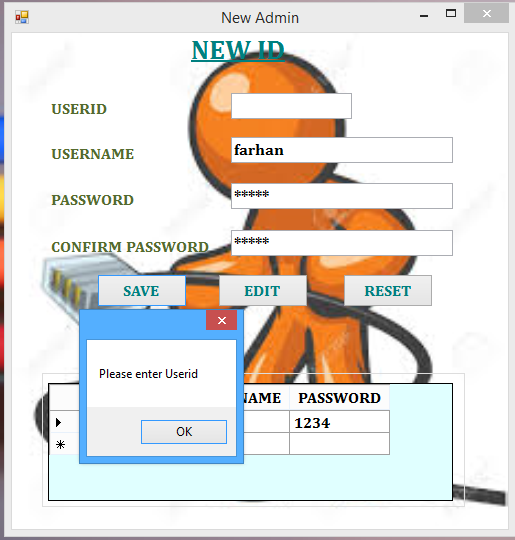


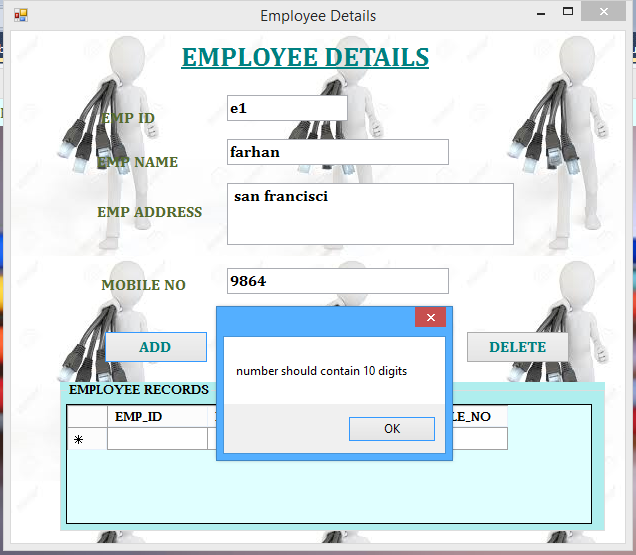
**LOGOUT**

**VALIDATING FORMS**

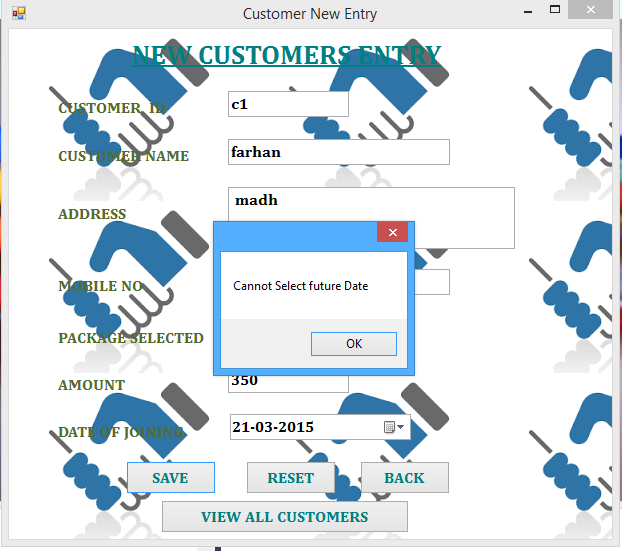
**Login:**



**Admin:**

**Employee:**

**Customers:**



**Coding**

**Login:**

Imports System.Data

Imports System.Data.SqlClient

Public Class Login

Dim constr As String = "Data Source=SAJID-PC;Initial Catalog=VideoVisionMS;Integrated Security=True"

Public con As New SqlConnection(constr)

Public cmd As New SqlCommand

Public dr As SqlDataReader

Public adp As SqlDataAdapter

Public ds As New DataSet

Private Sub btnLogin\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnLogin.Click

If tbUsername.Text = "" Or tbPass.Text = "" Then

MessageBox.Show("Please enter username & password", "Login Error", MessageBoxButtons.OK, MessageBoxIcon.Information)

ElseIf tbUsername.Text = "" Then

MessageBox.Show("Please enter username", "Login Error", MessageBoxButtons.OK, MessageBoxIcon.Information)

tbUsername.Focus()

ElseIf tbPass.Text = "" Then

MessageBox.Show("Please enter password", "Login Error", MessageBoxButtons.OK, MessageBoxIcon.Information)

Else

Dim cmd As New SqlCommand

cmd.Connection = con

cmd.CommandText = "select Username,Password from Login where Username='" & tbUsername.Text & "' and Password='" & tbPass.Text & "'"

Try

con.Open()

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

dr = cmd.ExecuteReader

Try

If dr.Read = False Then

MessageBox.Show("Wrong username or password", "Login Error", MessageBoxButtons.OK, MessageBoxIcon.Information)

tbUsername.Text = ""

tbPass.Text = ""

tbUsername.BackColor = Color.Red

tbPass.BackColor = Color.Red

tbUsername.Focus()

Else

MessageBox.Show("Login successful...")

tbUsername.BackColor = Color.White

tbPass.BackColor = Color.White

Me.Hide()

MDI.Show()

End If

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

con.Close()

End If

End Sub

Private Sub btnClear\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCLear.Click

tbUsername.Text = ""

tbPass.Text = ""

End Sub

Private Sub btnExit\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnExit.Click

Dim s = MessageBox.Show("Are you Sure you want to EXIT", "EXIT APPLICATION", MessageBoxButtons.YesNo, MessageBoxIcon.Question)

If s = Windows.Forms.DialogResult.Yes Then

Me.Close()

Else

Me.Refresh()

End If

End Sub

End Class

**MDI:**

Public Class MDI

Private Sub ABOUTORGANIZATIONToolStripMenuItem1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ABOUTORGANIZATIONToolStripMenuItem1.Click

AboutUs.Focus()

AboutUs.Show()

End Sub

Private Sub NEWENTRYToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NEWENTRYToolStripMenuItem.Click

CustomerNewEntry.Focus()

CustomerNewEntry.Show()

End Sub

Private Sub EXISTINGCUSTOMERSToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles EXISTINGCUSTOMERSToolStripMenuItem.Click

CustomerRecords.Focus()

CustomerRecords.Show()

End Sub

Private Sub PACKAGESLISTSToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PACKAGESLISTSToolStripMenuItem.Click

PackagesList.Focus()

PackagesList.Show()

End Sub

Private Sub NEWIDToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NEWIDToolStripMenuItem.Click

NewAdmin.Focus()

NewAdmin.Show()

End Sub

Private Sub DATABASEGRIDVIEWToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DATABASEGRIDVIEWToolStripMenuItem.Click

CustomerDatabase.Focus()

CustomerDatabase.Show()

End Sub

Private Sub EMPLOYEEDETAILSToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles EMPLOYEEDETAILSToolStripMenuItem.Click

EmployeeDetails.Focus()

EmployeeDetails.Show()

End Sub

Private Sub CUSTOMERBILLToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CUSTOMERBILLToolStripMenuItem.Click

CustomerBill.Focus()

CustomerBill.Show()

End Sub

Private Sub ALLBILLSToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ALLBILLSToolStripMenuItem.Click

ALLBILLS.Focus()

ALLBILLS.Show()

End Sub

Private Sub SIGNOUTToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SIGNOUTToolStripMenuItem.Click

Dim ans As Integer

ans = MsgBox("Are you sure you want to exit from this system????", vbYesNo + vbQuestion, "Signout")

If ans = MsgBoxResult.No Then

Exit Sub

Else

End

End If

End Sub

End Class

**New Amin:**

Imports System.Data

Imports System.Data.SqlClient

Public Class NewAdmin

Sub fillDatagrid()

Dim da As New SqlDataAdapter("select \* from Login", con)

Dim dt As New DataTable

da.Fill(dt)

DataGridView1.DataSource = dt

End Sub

Private Sub Form8\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

fillDatagrid()

tbUserID.Focus()

btnUpdate.Visible = False

btnDelete.Visible = False

btnCancel.Visible = False

tbUserID.Focus()

End Sub

Private Sub btnsave\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnsave.Click

OpenCon()

Dim tel As String

tel = tbpass.Text

If tel.Length < 5 Then

MessageBox.Show("Password is too weak please enter new password...!!")

tbpass.Focus()

ElseIf tbUserID.Text = "" Then

MessageBox.Show("Please enter Userid")

tbUserID.Focus()

ElseIf tbusername.Text = "" Then

MessageBox.Show("Please enter Username")

tbusername.Focus()

ElseIf tbpass.Text = "" Then

MessageBox.Show("Please enter password")

tbpass.Focus()

ElseIf tbpass.Text <> tbconfmpass.Text Then

MessageBox.Show("Password mismatched")

tbpass.Clear()

tbconfmpass.Clear()

tbpass.Focus()

Else

Try

Dim str As String = "Insert into Login(USERID,USERNAME,PASSWORD)values('" & tbUserID.Text & "','" & tbusername.Text & "','" & tbpass.Text & "')"

If (ExecuteQuery(str) = True) Then

fillDatagrid()

MessageBox.Show("Record inserted successfully")

clearcontrols()

End If

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

End If

End Sub

Private Sub btnUpdate\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnUpdate.Click

If tbUserID.Text = "" Then

MessageBox.Show("Please enter Userid")

tbUserID.Focus()

ElseIf tbusername.Text = "" Then

MessageBox.Show("Please enter Username")

tbusername.Focus()

ElseIf tbpass.Text = "" Then

MessageBox.Show("Please enter password")

tbpass.Focus()

ElseIf tbpass.Text <> tbconfmpass.Text Then

MessageBox.Show("Password Mismatched")

tbpass.Clear()

tbconfmpass.Clear()

tbpass.Focus()

Else

Try

Dim str As String = "update login set USERNAME = '" & tbusername.Text & "',PASSWORD='" & tbpass.Text & "' where USERID ='" & tbUserID.Text & "'"

If (ExecuteQuery(str) = True) Then

fillDatagrid()

MessageBox.Show("Record updated successfully")

clearcontrols()

End If

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

End If

End Sub

Private Sub btnDelete\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDelete.Click

If tbUserID.Text = "" Or tbusername.Text = "" Or tbpass.Text = "" Or tbconfmpass.Text = "" Then

MessageBox.Show("Error in deleting....")

tbUserID.Focus()

ElseIf tbpass.Text <> tbconfmpass.Text Then

MessageBox.Show("Password Mismatched")

tbconfmpass.Clear()

tbconfmpass.Focus()

Else

Try

Dim str As String = "delete from Login where USERID ='" & tbUserID.Text & "'"

If (ExecuteQuery(str) = True) Then

fillDatagrid()

MessageBox.Show("Record deleted succesfully")

clearcontrols()

End If

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

End If

End Sub

Private Sub btnedit\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnedit.Click

tbUserID.Focus()

btnCancel.Visible = True

btnsave.Enabled = False

btnedit.Enabled = False

btnUpdate.Visible = True

btnDelete.Visible = True

End Sub

Private Sub btnreset\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnreset.Click

clearcontrols()

End Sub

Private Sub btnCancel\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCancel.Click

tbUserID.Focus()

btnsave.Enabled = True

btnedit.Enabled = True

btnUpdate.Visible = False

btnDelete.Visible = False

btnCancel.Visible = False

End Sub

Private Sub clearcontrols()

tbUserID.Text = ""

tbusername.Text = ""

tbpass.Text = ""

tbconfmpass.Text = ""

tbUserID.Focus()

End Sub

Private Sub tbUserID\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tbUserID.TextChanged

Dim ID, USERNAME, PASSWORD As String

Dim userid As String = tbUserID.Text

Dim str As String = "select \* from Login where USERID = '" & userid & "' "

Try

dr = RetrieveQuery(str)

Do While dr.Read

ID = dr(0)

USERNAME = dr(1)

PASSWORD = dr(2)

tbUserID.Text = ID

tbusername.Text = USERNAME

tbpass.Text = PASSWORD

con.Close()

Exit Sub

Loop

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

con.Close()

If tbUserID.Text = "" Then

tbusername.Text = ""

tbpass.Text = ""

tbconfmpass.Text = ""

End If

End Sub

End Class

**Employee:**

Imports System.Data

Imports System.Data.SqlClient

Public Class EmployeeDetails

Sub fillDatagrid()

Dim da As New SqlDataAdapter("select \* from Employee", con)

Dim dt As New DataTable

da.Fill(dt)

DataGridView1.DataSource = dt

End Sub

Private Sub Form10\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

fillDatagrid()

End Sub

Private Sub btnADD\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnADD.Click

If tbEmpid.Text = "" Or tbEmpName.Text = "" Or tbempAddress.Text = "" Or tbMobileNo.Text = "" Then

MessageBox.Show("please insert all the required information....")

tbEmpid.Focus()

ElseIf tbEmpName.Text = "" Then

MessageBox.Show("please insert Employee Name")

tbEmpName.Focus()

ElseIf tbempAddress.Text = "" Then

MessageBox.Show("please insert Employee Address")

tbempAddress.Focus()

ElseIf tbMobileNo.Text = "" Then

MessageBox.Show("please insert Mobile Number")

tbMobileNo.Focus()

ElseIf tbMobileNo.Text.Length < 5 Then

MessageBox.Show("number should contain more than 5 digits...")

tbMobileNo.Focus()

ElseIf tbMobileNo.Text.Length > 10 Then

MessageBox.Show("number should contain less than or equal to 10 digits...")

tbMobileNo.Focus()

Else

Try

Dim str As String = "Insert into Employee(EMP\_ID,EMP\_NAME,EMP\_ADDRESS,MOBILE\_NO)values('" & tbEmpid.Text & "','" & tbEmpName.Text & "','" & tbempAddress.Text & "','" & tbMobileNo.Text & "')"

If (ExecuteQuery(str) = True) Then

fillDatagrid()

MessageBox.Show("Record added succesfully..")

clearrecords()

End If

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

End If

End Sub

Private Sub btnUpdate\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnUpdate.Click

If tbEmpid.Text = "" Or tbEmpName.Text = "" Or tbempAddress.Text = "" Or tbMobileNo.Text = "" Then

MessageBox.Show("Error in updating....")

tbEmpName.Focus()

Exit Sub

End If

Try

Dim str As String = "update employee set EMP\_NAME ='" & tbEmpName.Text & "',EMP\_ADDRESS='" & tbempAddress.Text & "',MOBILE\_NO='" & tbMobileNo.Text & "' where EMP\_ID = '" & tbEmpid.Text & "'"

If (ExecuteQuery(str) = True) Then

fillDatagrid()

MessageBox.Show("Record updated successfully")

clearrecords()

End If

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

End Sub

Private Sub btnDelete\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDelete.Click

If tbEmpid.Text = "" Or tbEmpName.Text = "" Or tbempAddress.Text = "" Or tbMobileNo.Text = "" Then

MessageBox.Show("Error in deleting....")

tbEmpid.Focus()

Exit Sub

End If

Try

Dim str As String = "delete from Employee where EMP\_ID ='" & tbEmpid.Text & "'"

Dim s = MessageBox.Show("Are you sure to delete", "Delete Records", MessageBoxButtons.YesNo, MessageBoxIcon.Question)

If s = vbYes Then

If (ExecuteQuery(str) = True) Then

fillDatagrid()

clearrecords()

MessageBox.Show("Record Successfully Deleted")

End If

Else

clearrecords()

End If

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

End Sub

Private Sub btnReset\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnReset.Click

clearrecords()

End Sub

Private Sub tbEmpid\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tbEmpid.TextChanged

Dim ID, EMPADDRESS, EMPNAME, EMPMOBILE As String

Dim empid As String = tbEmpid.Text

Dim str As String = "select \* from Employee where EMP\_ID= '" & empid & "' "

Try

dr = RetrieveQuery(str)

Do While dr.Read

ID = dr(0)

EMPNAME = dr(1)

EMPADDRESS = dr(2)

EMPMOBILE = dr(3)

tbEmpid.Text = ID

tbEmpName.Text = EMPNAME

tbempAddress.Text = EMPADDRESS

tbMobileNo.Text = EMPMOBILE

con.Close()

Exit Sub

Loop

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

con.Close()

If tbEmpid.Text = "" Then

clearrecords()

End If

End Sub

Private Sub tbMobileNo\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles tbMobileNo.KeyPress

Dim keychar = e.KeyChar

If Not Char.IsDigit(e.KeyChar) And (AscW(keychar) <> 8) Then

e.Handled = True

End If

End Sub

Private Sub clearrecords()

tbEmpid.Clear()

tbEmpName.Clear()

tbempAddress.Clear()

tbMobileNo.Clear()

End Sub

End Class

**New Customer:**

Imports System.Data

Imports System.Data.SqlClient

Public Class CustomerNewEntry

Dim sda As SqlDataAdapter

Private Sub CustomerNewEntry\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

AddtoCombo()

End Sub

Private Sub btnSave\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSave.Click

If tbCustomerID.Text = "" Or tbCustomerName.Text = "" Or tbAddress.Text = "" Or tbMobileno.Text = "" Or tbAmount.Text = "" Or cbPackageSelected.SelectedIndex = -1 Then

MessageBox.Show("please insert all the required information....")

tbCustomerID.Focus()

ElseIf tbCustomerName.Text = "" Then

MessageBox.Show("please insert Customer Name")

tbCustomerName.Focus()

ElseIf tbAddress.Text = "" Then

MessageBox.Show("please insert Customer Address")

tbAddress.Focus()

ElseIf tbMobileno.Text = "" Then

MessageBox.Show("please insert Mobile Number")

tbMobileno.Focus()

ElseIf tbMobileno.Text.Length < 5 Then

MessageBox.Show("number should contain more than 5 digits...")

tbMobileno.Focus()

ElseIf tbMobileno.Text.Length > 10 Then

MessageBox.Show("number should contain less than or equal to 10 digits...")

tbMobileno.Focus()

Else

Try

Dim str As String = "Insert into Customer\_Records(CUST\_ID,CUSTOMER\_NAME,ADDRESS,MOBILE\_NO,PACKAGE\_SELECTED,AMOUNT,DATE\_OF\_JOINING) values('" & tbCustomerID.Text & "','" & tbCustomerName.Text & "','" & tbAddress.Text & "','" & tbMobileno.Text & "','" & cbPackageSelected.SelectedItem & "','" & tbAmount.Text & "','" & DateTimePicker1.Value.Date.ToString("dd-MMM-yyyy") & "')"

If (ExecuteQuery(str) = True) Then

MessageBox.Show("Record inserted successfully")

Me.Hide()

clearrecords()

MDI.Show()

End If

Catch ex As Exception

'MessageBox.Show(ex.Message)

End Try

End If

End Sub

Private Sub btnBack\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnBack.Click

Me.Hide()

End Sub

Private Sub btnReset\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnReset.Click

clearrecords()

tbCustomerID.Focus()

End Sub

Private Sub cbPackageSelected\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cbPackageSelected.SelectedIndexChanged

retrievetocb()

End Sub

Private Sub clearrecords()

tbCustomerID.Text = ""

tbCustomerName.Text = ""

tbAddress.Text = ""

tbMobileno.Text = ""

cbPackageSelected.SelectedIndex = -1

tbAmount.Text = ""

End Sub

Private Sub AddtoCombo()

Dim x As Integer

Dim max As String

Dim constr = "Data Source=SAJID-PC;Initial Catalog=VideoVisionMS;Integrated Security=True"

Dim sqlstr = "Select PACKAGE\_NAME,PACKAGE\_AMOUNT from PAckages"

Dim con = New SqlConnection(constr)

Dim cmd = New SqlCommand(sqlstr, con)

Dim adp = New SqlDataAdapter(cmd)

adp.Fill(ds, "Packages")

max = ds.Tables("Packages").Rows.Count

For x = 0 To max - 1

cbPackageSelected.Items.Add(ds.Tables("Packages").Rows(x).Item(0))

Next

End Sub

Private Sub tbMobileno\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles tbMobileno.KeyPress

Dim keychar = e.KeyChar

If Not Char.IsDigit(e.KeyChar) And (AscW(keychar) <> 8) Then

e.Handled = True

End If

End Sub

Private Sub btnVIEWALLCUSTOMERS\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnVIEWALLCUSTOMERS.Click

CustomerDatabase.Show()

End Sub

Private Function retrievetocb() As Boolean

Dim query As String

Dim datarow As DataTableReader

query = "SELECT PACKAGE\_AMOUNT FROM Packages WHERE PACKAGE\_NAME = '" & cbPackageSelected.Text & "'"

Try

Module1.OpenCon()

sda = New SqlDataAdapter(query, con)

sda.Fill(ds, "Packages")

datarow = ds.Tables("Packages").CreateDataReader()

If datarow.HasRows() = True Then

While datarow.Read()

tbAmount.Text = datarow.Item("PACKAGE\_AMOUNT")

End While

End If

Catch ex As Exception

MessageBox.Show(ex.Message)

Finally

Module1.CloseCon()

End Try

Return True

End Function

End Class

**Customer Records:**

Imports System.Data

Imports System.Data.SqlClient

Public Class CustomerRecords

Dim sda As SqlDataAdapter

Private Sub CustomerRecords\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

FillComboPackage()

End Sub

Private Sub btnupdate\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnupdate.Click

If tbCustomerID.Text = "" Or tbCustomerName.Text = "" Or tbAddress.Text = "" Or tbMobileNo.Text = "" Or cbPackageSelected.SelectedIndex = -1 Or tbAmount.Text = "" Or DateTimePicker1.Value.Date > System.DateTime.Now Then

MessageBox.Show("Error in updating....")

tbCustomerID.Focus()

ElseIf tbCustomerName.Text = "" Then

MessageBox.Show("Please enter Customer Name")

'tbCustomerName.Focus()

ElseIf tbAddress.Text = "" Then

MessageBox.Show("Please enter Customer's Address")

tbAddress.Focus()

ElseIf tbMobileNo.Text = "" Then

MessageBox.Show("please insert Mobile Number")

tbMobileNo.Focus()

ElseIf tbMobileNo.Text.Length < 5 Then

MessageBox.Show("number should contain more than 5 digits...")

tbMobileNo.Focus()

ElseIf tbMobileNo.Text.Length > 10 Then

MessageBox.Show("number should contain less than or equal to 10 digits...")

tbMobileNo.Focus()

Else

Try

Dim str As String = "update Customer\_Records set CUSTOMER\_NAME ='" & tbCustomerName.Text & "', ADDRESS='" & tbAddress.Text & "',MOBILE\_NO='" & tbMobileNo.Text & "',PACKAGE\_SELECTED='" & cbPackageSelected.SelectedItem & "',AMOUNT='" & tbAmount.Text & "',DATE\_OF\_JOINING='" & DateTimePicker1.Value.Date.ToString("dd-MMM-yyyy") & "' where CUST\_ID = '" & tbCustomerID.Text & "'"

If (ExecuteQuery(str) = True) Then

MessageBox.Show("Record updated successfully")

Me.Hide()

clearrecords()

End If

Catch ex As Exception

'MessageBox.Show(ex.Message)

End Try

End If

End Sub

Private Sub btndelete\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btndelete.Click

If tbCustomerID.Text = "" Or tbCustomerName.Text = "" Or tbAddress.Text = "" Or tbMobileNo.Text = "" Or tbAmount.Text = "" Or DateTimePicker1.Value.Date > System.DateTime.Now Then

MessageBox.Show("Error in deleting....")

tbCustomerID.Focus()

Exit Sub

End If

Try

Dim str As String = "delete from Customer\_Records where CUST\_ID='" & tbCustomerID.Text & "'"

Dim s = MessageBox.Show("Are you sure to delete", "Delete Records", MessageBoxButtons.YesNo, MessageBoxIcon.Question)

If s = vbYes Then

If (ExecuteQuery(str) = True) Then

MessageBox.Show("Record deleted succesfully")

Me.Hide()

clearrecords()

CustomerDatabase.Show()

DateTimePicker1.Text = CStr(Date.Now)

End If

End If

Catch ex As Exception

'MessageBox.Show(ex.Message)

End Try

End Sub

Private Sub tbCustomerID\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tbCustomerID.TextChanged

Dim CUST\_ID, CUSTOMERNAME, ADDRESS, PHONENO, PACKAGESELECTED, AMOUNT, DATEOFJOINING As String

Dim id As String = tbCustomerID.Text

Dim str As String = "select \* from Customer\_Records where CUST\_ID= '" & id & "' "

Try

dr = RetrieveQuery(str)

Do While dr.Read

CUST\_ID = dr(0)

CUSTOMERNAME = dr(1)

ADDRESS = dr(2)

PHONENO = dr(3)

PACKAGESELECTED = dr(4)

AMOUNT = dr(5)

DATEOFJOINING = dr(6)

tbCustomerID.Text = CUST\_ID

tbCustomerName.Text = CUSTOMERNAME

tbAddress.Text = ADDRESS

tbMobileNo.Text = PHONENO

cbPackageSelected.Text = PACKAGESELECTED

tbAmount.Text = AMOUNT

DateTimePicker1.Value = DATEOFJOINING

con.Close()

Exit Sub

Loop

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

con.Close()

If tbCustomerID.Text = "" Then

tbCustomerName.Text = ""

tbAddress.Text = ""

tbMobileNo.Text = ""

cbPackageSelected.SelectedIndex = -1

tbAmount.Text = ""

End If

End Sub

Private Sub btnback\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnback.Click

Me.Hide()

End Sub

Private Sub cbPackageSelected\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cbPackageSelected.SelectedIndexChanged

retrievetocb()

End Sub

Private Sub btnreset\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnreset.Click

clearrecords()

tbCustomerID.Focus()

End Sub

Private Sub btnVIEWALLCUSTOMERS\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnVIEWALLCUSTOMERS.Click

CustomerDatabase.Show()

End Sub

Private Sub FillComboPackage()

Dim x As Integer

Dim max As String

'Dim constr As String = "Data Source=RAZI;Initial Catalog=VideoVisionMS;Integrated Security=True"

Dim constr = "Data Source=SAJID-PC;Initial Catalog=VideoVisionMS;Integrated Security=True"

Dim sqlstr = "Select PACKAGE\_NAME,PACKAGE\_AMOUNT from Packages"

Dim con = New SqlConnection(constr)

Dim cmd = New SqlCommand(sqlstr, con)

Dim adp = New SqlDataAdapter(cmd)

adp.Fill(ds, "Packages")

max = ds.Tables("Packages").Rows.Count

For x = 0 To max - 1

cbPackageSelected.Items.Add(ds.Tables("Packages").Rows(x).Item(0))

Next

End Sub

Private Sub clearrecords()

tbCustomerID.Text = ""

tbCustomerName.Text = ""

tbAddress.Text = ""

tbMobileNo.Text = ""

cbPackageSelected.SelectedIndex = -1

tbAmount.Text = ""

End Sub

Private Sub tbMobileNo\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles tbMobileNo.KeyPress

Dim keychar = e.KeyChar

If Not Char.IsDigit(e.KeyChar) And (AscW(keychar) <> 8) Then

e.Handled = True

End If

End Sub

Private Function retrievetocb() As Boolean

Dim query As String

Dim datarow As DataTableReader

query = "SELECT PACKAGE\_AMOUNT FROM Packages WHERE PACKAGE\_NAME = '" & cbPackageSelected.Text & "'"

Try

Module1.OpenCon()

sda = New SqlDataAdapter(query, con)

sda.Fill(ds, "Packages")

datarow = ds.Tables("Packages").CreateDataReader()

If datarow.HasRows() = True Then

While datarow.Read()

tbAmount.Text = datarow.Item("PACKAGE\_AMOUNT")

End While

End If

Catch ex As Exception

MessageBox.Show(ex.Message)

Finally

Module1.CloseCon()

End Try

Return True

End Function

End Class

**Customer Data grid:**

Imports System.Data

Imports System.Data.SqlClient

Public Class CustomerDatabase

Sub fillDatagrid()

Dim da As New SqlDataAdapter("select \* from Customer\_Records", con)

Dim dt As New DataTable

da.Fill(dt)

DataGridView1.DataSource = dt

End Sub

Private Sub Database\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

fillDatagrid()

End Sub

End Class

**Packages:**

Imports System.Data

Imports System.Data.SqlClient

Public Class PackagesList

Sub fillDatagrid()

Dim da As New SqlDataAdapter("select \* from Packages", con)

Dim dt As New DataTable

da.Fill(dt)

DataGridView1.DataSource = dt

End Sub

Private Sub PackagesList\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

fillDatagrid()

End Sub

End Class

**Customer Bill:**

Imports System.Data

Imports System.Data.SqlClient

Public Class CustomerBill

Dim AmountPaid As Boolean

Private Sub CustomerBill\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

rbNo.Checked = True

btnOk.Enabled = False

End Sub

Private Sub tbCustomerID\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tbCustomerID.TextChanged

Dim CUST\_ID, CUSTOMERNAME, PACKAGESELECTED, AMOUNT, STARTDATE As String

Dim id As String = tbCustomerID.Text

Dim str As String = "select \* from Customer\_Records where CUST\_ID= '" & id & "' "

Try

dr = RetrieveQuery(str)

Do While dr.Read

CUST\_ID = dr(0)

CUSTOMERNAME = dr(1)

PACKAGESELECTED = dr(4)

AMOUNT = dr(5)

STARTDATE = dr(6)

tbCustomerID.Text = CUST\_ID

tbCustomerName.Text = CUSTOMERNAME

tbPackageSelected.Text = PACKAGESELECTED

tbAmount.Text = AMOUNT

dtpStartDAte.Value = STARTDATE

dtpTodaysdate.Value = Today.Date

TotalDays()

TotalAmount()

con.Close()

Exit Sub

Loop

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

con.Close()

If tbCustomerID.Text = "" Then

tbCustomerName.Text = ""

tbPackageSelected.Text = ""

tbAmount.Text = ""

End If

End Sub

Private Sub rbYes\_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles rbYes.CheckedChanged

AmountPaid = 1

btnOk.Enabled = True

End Sub

Private Sub btnOk\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnOk.Click

If tbCustomerID.Text = "" Or tbCustomerName.Text = "" Or tbAmount.Text = "" Then

MessageBox.Show("please insert all the required information....")

tbCustomerID.Focus()

ElseIf tbCustomerName.Text = "" Then

MessageBox.Show("please insert Customer Name")

tbCustomerName.Focus()

Else

Try

Dim str As String = "Insert into Customer\_Bill(CUST\_ID,CUSTOMER\_NAME,PACKAGE\_SELECTED,AMOUNT,START\_DATE,END\_DATE,AMOUNT\_PAID,TOTAL\_AMOUNT) values('" & tbCustomerID.Text & "','" & tbCustomerName.Text & "','" & tbPackageSelected.Text & "','" & tbAmount.Text & "','" & dtpStartDAte.Value.Date.ToString("dd-MMM-yyyy") & "','" & dtpTodaysdate.Value.Date.ToString("dd-MMM-yyyy") & "','" & AmountPaid & "','" & tbTotalAmount.Text & "')"

If (ExecuteQuery(str) = True) Then

UpdateStartDate()

clearcontrols()

rbNo.Checked = True

btnOk.Enabled = False

End If

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

End If

End Sub

Private Sub rbNo\_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles rbNo.CheckedChanged

AmountPaid = 0

End Sub

Private Sub btnReset\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnReset.Click

clearcontrols()

End Sub

Private Sub btnCancel\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCancel.Click

Dim s = MessageBox.Show("Are you sur you want to cancel", "Bill", MessageBoxButtons.YesNo, MessageBoxIcon.Information)

If s = Windows.Forms.DialogResult.Yes Then

Me.Hide()

MDI.Show()

ElseIf s = Windows.Forms.DialogResult.No Then

Me.Show()

End If

End Sub

Private Sub tbTotalDays\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles tbTotalDays.KeyPress

If Not Char.IsDigit(e.KeyChar) Then

e.Handled = True

End If

End Sub

Private Sub TotalDays()

Try

Dim sd As Date = dtpStartDAte.Value

Dim ed As Date = dtpTodaysdate.Value

Dim totaldate As Integer = ed.Subtract(sd).TotalDays

tbTotalDays.Text = totaldate

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

End Sub

Private Sub TotalAmount()

Try

If Val(tbTotalDays.Text) = 0 Then

tbTotalAmount.Text = CInt(tbAmount.Text) \* 0

ElseIf Val(tbTotalDays.Text) > 0 And Val(tbTotalDays.Text) <= 30 Then

tbTotalAmount.Text = CInt(tbAmount.Text) \* 1

ElseIf Val(tbTotalDays.Text) > 30 And Val(tbTotalDays.Text) <= 60 Then

tbTotalAmount.Text = CInt(tbAmount.Text) \* 2

ElseIf Val(tbTotalDays.Text) > 60 And Val(tbTotalDays.Text) <= 90 Then

tbTotalAmount.Text = CInt(tbAmount.Text) \* 3

ElseIf Val(tbTotalDays.Text) > 90 And Val(tbTotalDays.Text) <= 120 Then

tbTotalAmount.Text = CInt(tbAmount.Text) \* 4

ElseIf Val(tbTotalDays.Text) > 120 And Val(tbTotalDays.Text) <= 150 Then

tbTotalAmount.Text = CInt(tbAmount.Text) \* 5

ElseIf Val(tbTotalDays.Text) > 150 And Val(tbTotalDays.Text) <= 180 Then

tbTotalAmount.Text = CInt(tbAmount.Text) \* 6

ElseIf Val(tbTotalDays.Text) > 180 And Val(tbTotalDays.Text) <= 210 Then

tbTotalAmount.Text = CInt(tbAmount.Text) \* 7

ElseIf Val(tbTotalDays.Text) > 210 And Val(tbTotalDays.Text) <= 240 Then

tbTotalAmount.Text = CInt(tbAmount.Text) \* 8

ElseIf Val(tbTotalDays.Text) > 240 And Val(tbTotalDays.Text) <= 270 Then

tbTotalAmount.Text = CInt(tbAmount.Text) \* 8

ElseIf Val(tbTotalDays.Text) > 270 And Val(tbTotalDays.Text) <= 300 Then

tbTotalAmount.Text = CInt(tbAmount.Text) \* 9

ElseIf Val(tbTotalDays.Text) > 300 And Val(tbTotalDays.Text) <= 330 Then

tbTotalAmount.Text = CInt(tbAmount.Text) \* 10

ElseIf Val(tbTotalDays.Text) > 330 And Val(tbTotalDays.Text) <= 360 Then

tbTotalAmount.Text = CInt(tbAmount.Text) \* 11

ElseIf Val(tbTotalDays.Text) > 360 And Val(tbTotalDays.Text) <= 390 Then

tbTotalAmount.Text = CInt(tbAmount.Text) \* 12

End If

Catch ex As Exception

End Try

End Sub

Private Sub clearcontrols()

tbCustomerID.Clear()

tbCustomerName.Clear()

tbPackageSelected.Clear()

tbAmount.Clear()

tbTotalDays.Clear()

tbTotalAmount.Clear()

rbNo.Checked = True

End Sub

Private Sub UpdateStartDate()

Dim str = "Update Customer\_Records set DATE\_OF\_JOINING = '" & dtpTodaysdate.Value.Date.ToString("dd-MMM-yyyy") & "' where CUST\_ID = '" & tbCustomerID.Text & "' "

If (ExecuteQuery(str) = True) Then

MessageBox.Show("Bill Successfully Generated for " + tbCustomerName.Text)

End If

End Sub

Private Sub btnViewBills\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnViewBills.Click

ALLBILLS.Show()

End Sub

End Class

**Customer Bill Data Grid:**

Imports System.Data.SqlClient

Imports System.Data

Public Class ALLBILLS

Dim adp As SqlDataAdapter

Dim dt As DataTable

Dim ds As DataSet

Dim delete As String

Private Sub filldatagrid()

Dim adp As New SqlDataAdapter("select \* from Customer\_Bill", con)

dt = New DataTable

adp.Fill(dt)

DataGridView1.DataSource = dt

End Sub

Private Sub ALLBILLS\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

filldatagrid()

End Sub

Private Sub btnDelete\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDelete.Click

Try

Module1.OpenCon()

For i As Integer = Me.DataGridView1.SelectedRows.Count - 1 To 0 Step -1

delete = "delete Customer\_Bill where CUST\_ID ='" & Me.DataGridView1.SelectedRows(i).Cells(1).Value.ToString & "'"

adp = New SqlDataAdapter(delete, con)

ds = New DataSet

adp.Fill(ds)

Me.DataGridView1.Rows.Remove(Me.DataGridView1.SelectedRows(i))

Next

filldatagrid()

MessageBox.Show("Deleted Successfully", "Informed", MessageBoxButtons.OK, MessageBoxIcon.Information)

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

End Sub

Private Sub Back\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Back.Click

Me.Hide()

End Sub

End Class

**Module1(Connection):**

Imports System.Data

Imports System.Data.SqlClient

Module Module1

'Data Source=RAZI;Initial Catalog=VideoVisionMS;Integrated Security=True

Dim constr As String = "Data Source=SAJID-PC;Initial Catalog=VideoVisionMS;Integrated Security=True"

'Dim constr As String = "Data Source=RAZI;Initial Catalog=VideoVisionMS;Integrated Security=True"

Public con As New SqlConnection(constr)

Public cmd As New SqlCommand

Public dr As SqlDataReader

Public adp As SqlDataAdapter

Public ds As New DataSet

Public dv As New DataView

Public cnumber As String

Public Sub OpenCon()

Try

If con.State = ConnectionState.Open Then

con.Close()

End If

con.Open()

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

End Sub

Public Sub CloseCon()

Try

con.Close()

Catch ex As Exception

MessageBox.Show(ex.Message)

End Try

End Sub

Public Function RetrieveQuery(ByVal strQuery As String) As SqlDataReader

Try

OpenCon()

cmd.Connection = con

cmd.CommandText = strQuery

dr = cmd.ExecuteReader

RetrieveQuery = dr

Catch ex As Exception

MessageBox.Show(ex.Message)

Return dr

End Try

End Function

Public Function ExecuteQuery(ByVal strQuery As String) As Boolean

Try

OpenCon()

cmd.Connection = con

cmd.CommandText = strQuery

cmd.ExecuteNonQuery()

CloseCon()

ExecuteQuery = True

Catch ex As Exception

ExecuteQuery = False

MessageBox.Show(ex.Message)

End Try

End Function

End Module

**Module Design**

**BANANA COM System (B-COMS) is an integrated software deals with Cable Operators. Helps the Cable Operators manage their customers. It makes easy to all operations of the Cable Operators and accurate.**

**The whole system is only accessible by the admin.**

**Admin:**

**In the Admin module an existing as a full power to add a new admin.**

**As well he can view, edit & delete the other admin.**

**Employee:**

**Admin will add the staff**

**All the staff members cannot access all the modules. He can add or delete the staff accounts.**

**Customer:**

**Admin will add the new customers those who take new connection. Admin can add, update or delete the customers and he can view the entire customers list.**

**Packages:**

**Packages are combination of different channels. Currently in this system there are 6 packages and for now they can be considered as static. User have achoice to select the package he/she interested in.**

**Every package has amount and number of channels associated with it. Admin can view the packages list. Which will consist of number of channels, amount & package name.**

**Customer Bill:**

**Admin will generate the customer bill. Bill can be generated by customer at any date after connection. Maximum of 12th months bill can be generated by the current system at a time.**

**SOFTWARE TESTING**

**black box testing**

**In clearing house across various modules this testing was performed to check the following.**

1. **Establishing communication with the database for handling request and response.**
2. **Verification of SQL providers in functionality**
3. **Parameters passing and report generation used from the application with crystal report.**

**White box testing**

**All the statements included in the code across various modules were tested to find none of the statements where overlooked or skipped from execution. This enabled isolating of errors that would have otherwise occurred and would have resulted in abnormal terminal or exceptions Thrown.**

**The test was corely tested in patient and responsibility, Insured party, ailments, procedures and applied payment modules**

**In addition, numeric inputs were tested for invalid characters, invalid data format, size of the input data and the data type being handled.**

**INTEGRATED TESTING**

**All the units were combined from a menu driven application which then provided for integration with other modules the following well tested.**

**1. Message passing and communication between the modules**

**2. Data usage and synchronization**

**3. Flow of control using top-down testing confirming appropriate return of control as well as associated usability features.**

**MUTATION TESTING**

**All fields across every module were tested rigorously with inputs that were intentionally provided with wrong data. This testing resolves bugs and errors through exception handling. That was a result of any kind of invalid data.**

**FUTURE ENHANCEMENT**

**This project can be further enhanced to provide greater flexibility and**

**performance with certain modifications whenever necessary such as:**

* **Addition of printing facility of customer bill.**
* **Keeping salary records of Employees.**
* **Allowing access to some Employees based on their designation.**
* **Records of Customer Bill on Monthly basis.**

**CONCLUSION**

**The primary goal of this project to keep Customer records and provide them bills. This software revolves around the customer. As Customer is one of the most important entity in this software.**

**Later on addition can be made to make Employees as another important entity and some other changes can also be made which I had already prescribed above in ‘Future Enhancement’.**

**For Now my conclusion is that this system works fine in respect to Customer’s aspects.**

**NOMENCLATURE**

**ER Diagram**

**The above rectangle represents an entity in ER diagram.**

**The above eclipse shape represents an attribute in ER Diagram**

**The Diamond repents a relationship between two entities in ER diagram**

**DFD Diagram**

**It represents an entity in DFD Diagram**

**The circle represents the process.**



**It represents the database.**

**The arrows represents the flow of Data.**

**GANTT CHART**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Month  Phases | November | December | January | February | March |
| Preliminary investigation |  |  |  |  |  |
| System analysis |  |  |  |  |  |
| System design |  |  |  |  |  |
| System coding |  |  |  |  |  |
| Maintenance and evaluation |  |  |  |  |  |
| Project |  |  |  |  |  |

**Planning**

**Actual**

**BIBLOGRAPHY**

* **Master in VB.NET 2008**
* **VB 2008 - Murach**
* **SQL 2 – The Complete Reference.**
* [**www.w3schools.com**](http://www.w3schools.com)
* [**www.tutorialpoints.com**](http://www.tutorialpoints.com)