E-BILLING AND INVOICE SYSTEM

A PROJECT REPORT

Submitted by

V.SURYA INDIRA **103678137**

In partial fulfillment for the award of the degree of

B.Tech
In

COMPUTER SCIENCE

Under the esteemed Guidance of

Prof.Ch. Raja Ramesh
M.Tech, [PhD]

REGENCY INSTITUTE OF TECHNOLOGY
ADAVIPOLAM, YANAM-533464

Department of Computer Science and Engineering
March-2012

DECLARATION BY THE CANDIDATE

I here by declare that the project report entitled “E-Billing and Invoice System“submitted for the partial fulfillment of mini project in B.Tech Third year Computer Science & Engineering Department. I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Yanam Signature of the Candidate

Date:

DEPARTEMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

I here by declare that the project report entitled " E-Billing and Invoice System " submitted for the partial fulfillment of mini project in B.Tech Third year Computer Science & Engineering Department bonafide work carried out by him/her under my guidance. The project fulfills the requirements as per the regulations of this Institute and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Ch. Raja Ramesh	**S.T.V.V.S Kumar**
M.Tech [PhD]	**M.Tech,[PhD]**
Lab in charge	**Head of the Department**
Date:	Date:

Internal Examiner (s) **External Examiner (s)**

ACKNOWLEDGEMENT

"Task successful" makes everyone happy. But the happiness will be gold without glitter if we didn't state the persons who have supported us to make it a success. Success will be crowned to people who made it a reality but the people whose constant guidance and encouragement made it possible will be crowned first on the eve of success.

This acknowledgment transcends the reality of formality when we would like to express deep gratitude and respect to all those people behind the screen who guided, inspired and helped me for the completion of our project work. We consider ourselves lucky enough to get such a good project. This project would add as an asset to my academic profile.

We express our sincere gratitude to our respectful Principal **Dr. A. Ramakrishna Rao** for enabling us to make use of laboratory and library facilities liberally, that helped us a long way in carrying out our project work successfully.

We express our gratitude to the help of the Head of the Department of Computer Science and Engineering, **Mr. S. T. V. S. Kumar, M.Tech, (Ph. D)**, for his constant supervision, guidance and co-operation throughout the project and we would like to express our thankfulness to our project guide, **Mr.Ch. Raja Ramesh, M.Tech** for his constant motivation and valuable help through the project work.

We extend our sincere gratitude to our parents who have encouraged us with their blessings to do this project successfully. Finally we would like to thank to all our friends, all the teaching and non-teaching staff members of the CSE Department, for all the timely help, ideas and encouragement which helped throughout in the completion of project.

INDEX

Objective

This section presents a conceptual overview of the solution, and then provides an introduction to its requirements.

Solution Concept

The E- Billing and Invoice System consist of:

- **Customer, Products, Billing Generation**: Automate the current manual bill generation system and maintain the searchable customer, products database and product invoice, maintain the data security, user rights.
- **Report Generation**: A Report Generation system will be developed for the user and management of E- Billing and Invoicing System. This MIS system will have both details and summary type reports for analysis the sales volume, sales trend, available stock

Part of the solution will involve converting data from Microsoft Excel to the Database, system can import existing customer, product data from predefined MS Excel, CSV format. It reduces the data entry work and quick implementation of the system.

List of Tables:

List of Figures:

1. Introduction

1.1. Overview of Proposed System

1.1.1 Existing System:

The client uses MS Excel, and maintains their product list, customer list, and prints the invoice, however it is not possible them to share the data from multiple system in multi user environment, there is lot of duplicate work, and chance of mistake. When the product price are changed they need to update each and every excel file. There is no option to find and print previous saved invoice. There is no security; any body can access any report and sensitive data, also no reports to find out the sales volume, stock list, and summary report

1.1.2 Proposed System:

. This E-Billing and invoicing system is used to overcome the entire problem which they are facing currently, and making complete atomization of manual billing and invoicing system

2. System Analysis

The following preliminary lists are based on initial interviews

2.1 Business Requirements:

The business goal for the application is to support an increase the productivity and complete automation of existing manual bill and invoice generation process. Business requirements are discussed in the Scope section, with the following additional detail:

- Sales representatives need a method to store and access sales opportunity data, and when a sale is generated, convert some or all of the information into a sales order without re-entering information.
- Each sales representative should receive customer and sales data pertinent only to them.
- The accountant should be able to enter or update product information in one interface only, with all necessary product information being received by sales staff.
- Manager must receive his or her customer and appointment data plus detailed and rollup information for each sales representative on his or her team.
- The application should support the capability to use multi user environment.
- The MIS Executive should able to generate all type of reports as and when required by the management.

The sales staff wants to improve their current ability to analyze their customers. In particular, they want to focus on identifying their best customers and building long-term relationships within that base. To enable them to accomplish this goal, they want to extract meaningful data that easily answers the following questions:

- What are the early warning signs of problems?
- Who are my best customers across product lines?
- With whom do I focus my efforts for building a long-term relationship?

- What are my customers' issues as groups?
- Geographically, where are my best customers?
- What products are my customers buying and at what rate?

2.2 User Requirements:

User requirements are categorized by user type.

Sales Staff (Representatives and Managers)

- View the data in various ways, for example:
    - Customers who are the top buyers of specific items
    - Best customers based on criteria to be determined
    - Best customers based on geographical analysis
    - Drops in a customer's sales
- Store multilingual and multiregional information in the database rather than relying on the sales staff to translate the information
- Identify which product prices have been modified, especially on current orders in progress
- Use opportunity rules, which are statements that help the sales representative convert an opportunity into a sale
- Add third-party data sources and financial evaluation tools
- Identify where promotions and programs would be the most beneficial
- Apply discounts to customer orders:
    - Sales representatives can offer discounts up to 15 percent, or up to 20 percent with authorization.
    - Sales managers can offer discounts up to 20 percent
- Enable capture, analysis, and sharing of data about a customer across the company
- Use forecasts to establish sales goals

- Allow all staff to view all contacts, but allow each contact to be assigned to a sales representative

Accountant

- Update product details, including price, photo, and description
- Add, delete, and update product specifications
- Generate stock reports and check the availability, order the item as and when needed.

MIS Executive

- Generate Reports as per requirement.
- Share the data, and convert reports to different format like Text, CSV.

2.3 Operational Requirements:

The following requirements provide a high-level view of how the system will run:

- Processor usage should not exceed 80 percent during concurrent uses.
- Backups will occur incrementally throughout the day.
- A full weekly backup is required to WORM drives.
- Ensure that information is easy to access either, and meaningful for the sales representative and the company.
- Minimize the technical knowledge that sales and marketing staff need to access the data, generate ad hoc queries, track promotions, and view customer segmentation information.
- Any change to information must be reflected immediately, and the changes must be propagated to the search engine so that employees that perform searches see this new information.

- The application should work with the existing communications and networking infrastructure.
- The application should deploy with a minimum of additional operational processes, manual or otherwise.

2.4 System Requirements:

These are additional constraints from a system perspective:

- Previous data of customer, product details must be imported in the new system.
- The administrator must be able to monitor everything from the IT department.
- The information must be accessible by everyone in the company as per the rights specify.

Success Criteria

To determine the success of this project, the following metrics can be quantified and used to analyze success factors.

Sales Automation

- Problem customer identification. Ability to identify top 10 developing problem customers by sales representative.
- Identification of best customers across product lines and regions. All customers can be sorted by the factors (to be determined) that rank them qualitatively. Variable sorts and rankings can be chosen by the sales representative.
- Identification of issues across groups. All customers can be sorted by the issues (to be determined) within group categories that can be updated. Variable sorts and rankings can be chosen by the sales representative or other user.
- Analysis of quantities and rates of products. Products can be sorted by the quantity and dates sold. Variable sorts and rankings can be chosen by the user.

2.5 Software and Hardware Requirement:

General Requirement for Server/Client:

Type	Software	Hardware
Work Station/ Node	1. Windows XP 2. VB Run time 3. MS Office	1. P-4 2. RAM -256 MB
Database Server	1.Win2000 Advance Server	1.P-4 2.RAM- 2GB 3.Hard Disk-160GB 4.RAID
Application Server	1.Win 2000 Advance Server	1.P-4 2.RAM- 1GB 3.Hard Disk-40GB

Development Tools and Technologies

Front End

1. VB.Net

   Why VB.Net?

   VB.Net is built on the Windows Server System to take major advantage of the OS and which comes with a host of different servers which allows for building, deploying, managing and maintaining Windows Based solutions. The Windows Server System is designed with performance as priority and it provides scalability, reliability.

   The advantage of VB.NET includes

   - Rapid Application Development (RAD) Support.
   - Powerful Windows-based Applications
   - Simplified Deployment
   - Powerful, Flexible, Simplified Data Access
   - Improved Coding
   - Direct Access to the OS Platform
   - Object-Oriented Constructs
   - COM Interoperability

Back End

1. MS Access

   Why MS Access?

   MS Access is desktop RDMS support small application with all features like relational query, different data types, joins, and query. Includes

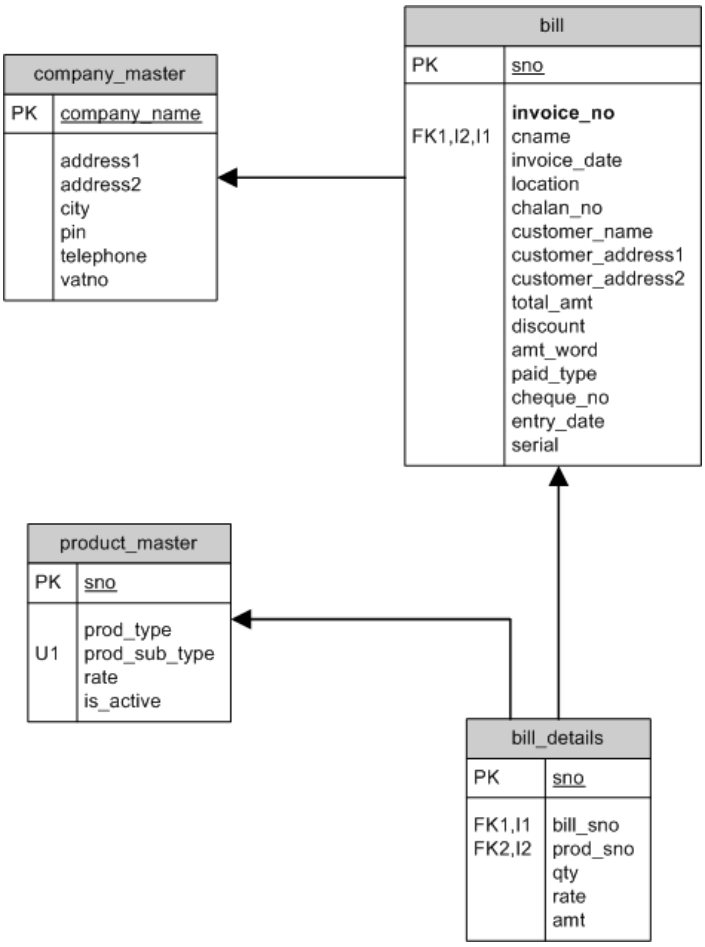
   - Easy to use and easy to deployment.

- Integration with Windows OS
- Scalability
- Import and Export of data in all major database system.
- Centralized Management
- Reliability
- Automating tasks

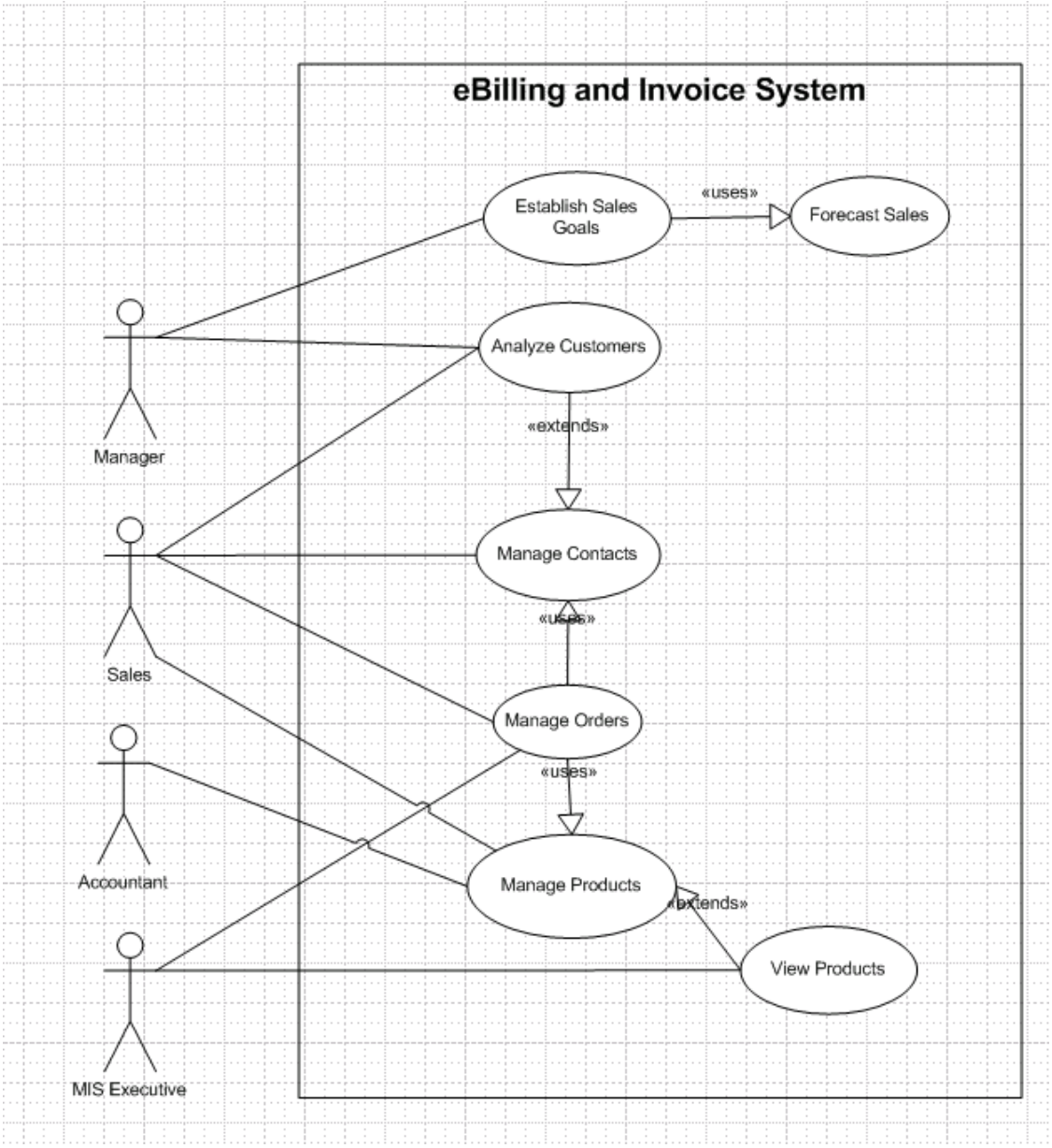
3. System Design

3.1 UML Diagrams

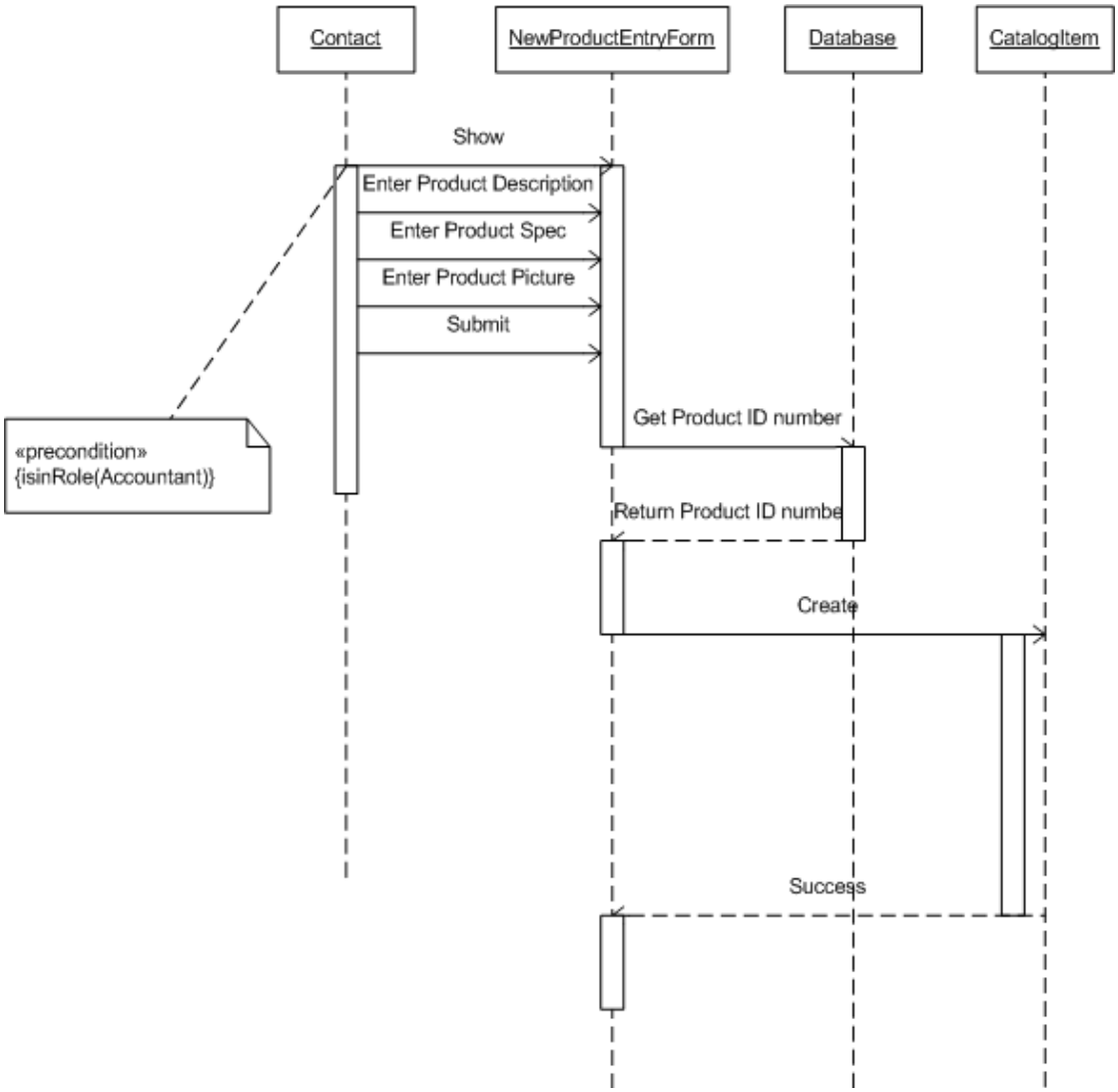
3.1.1 Class Diagrams



3.1.2 Activity Diagrams

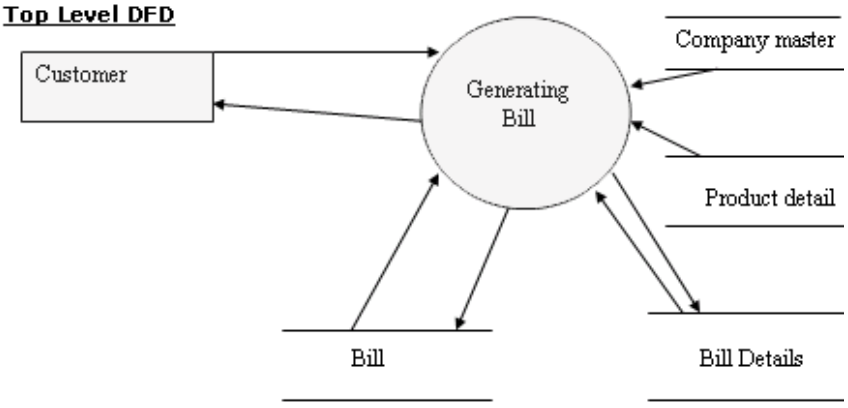


3.1.3 Sequence Diagrams

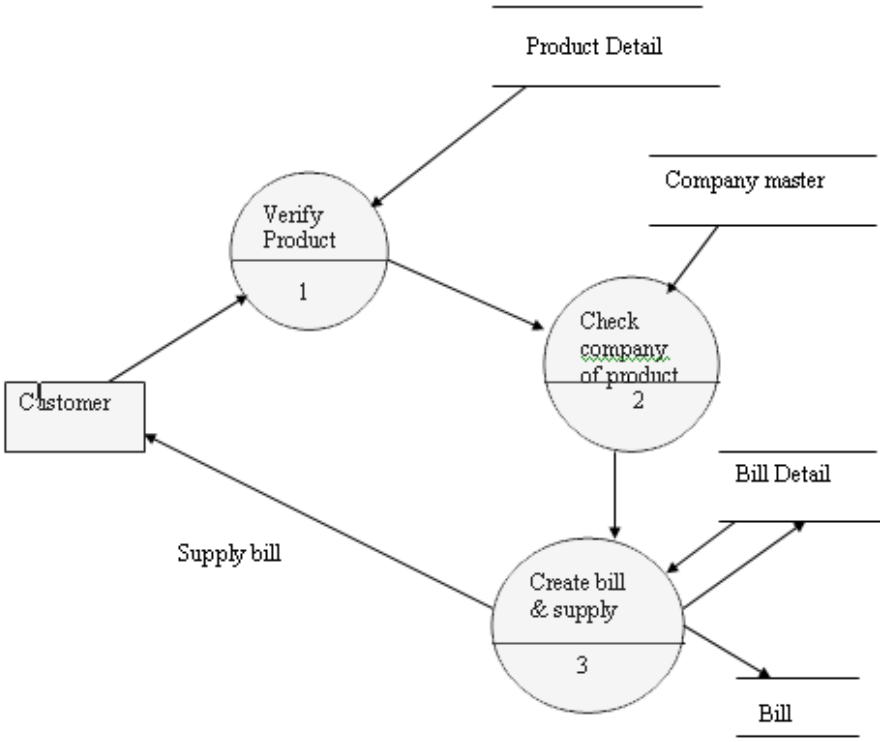


3.1.4 Data Flow Diagrams

Top Level DFD



First Level DFD



3.2 Data Dictionary

Table : product_master

Owner:	dbo
Destination DB name:	ebill
Number of columns:	5
Number of indexes:	2
Number of foreign keys:	0
Extended attributes:	
OnFileGroup	PRIMARY
Clustered PK	Yes

Columns	Data type	Allow NULLs	Value/range
Sno	int	Not allowed	
ProdType	nvarchar(60)	Not allowed	
ProdSubType	nvarchar(60)	Not allowed	
Rate	int	Not allowed	
isactive	nvarchar(30)	Not allowed	

Indexes	Type	Columns
PK_ProductMaster_sno	clustered, unique, primary key located on PRIMARY	sno

Table : bill

Owner:dbo

Destination DB name:	ebill
Number of columns:	15
Number of indexes:	3
Number of foreign keys:	0
Extended attributes:	
OnFileGroup	PRIMARY
Clustered PK	Yes

Columns	Data type	Allow NULLs	Value/range

sno	int	Not allowed
invoiceno	nvarchar(60)	Not allowed
canme	nvarchar(60)	Not allowed
Invoice_date	datetime	Not allowed
location	nvarchar(60)	Not allowed
chalan_no	int	Not allowed
customer_name	nvarchar(100)	Not allowed
Customer_address1	nvarchar(100)	Not allowed
Customer_address2	nvarchar(100)	Allowed
Total_amt	int	Not allowed
discount	int	Not allowed
Amt_words	nvarchar(100)	Not allowed
Paid_type	nvarchar(60)	Not allowed
Cheque_no	nvarchar(60)	Allowed
Entry_date	datetime	Not allowed

Indexes	Type	Columns
PK_Bill_SNO	clustered, unique, primary key located on PRIMARY	sno
IX_bill_cname	nonclustered located on PRIMARY	cname
IX_bill_invoiceno	nonclustered, unique located on PRIMARY	invoiceno

Table : bill_details

Owner:	dbo
Destination DB name:	ebill
Number of columns:	6
Number of indexes:	3
Number of foreign keys:	2
Extended attributes:	
OnFileGroup	PRIMARY
Clustered PK	Yes

Columns	Data type	Allow	Value/ran

		NULLs	ge
sno	int	Not allowed	
Bill_sno	int	Not allowed	
Prod_sno	nvarchar(60)	Not allowed	
qty	datetime	Not allowed	
rate	nvarchar(60)	Not allowed	
amt	int	Not allowed	

Indexes	Type	Columns
PK_Bill_details_SNO	clustered, unique, primary key located on PRIMARY	sno
IX_bill_details_bill_sno	nonclustered located on PRIMARY	Bill_sno
IX_bill_deatils_prod_sno	nonclustered, unique located on PRIMARY	Prod_sno

Table : Company_master

Owner:	dbo
Destination DB name:	ebill
Number of columns:	7
Number of indexes:	2
Number of foreign keys:	0
Extended attributes:	
OnFileGroup	PRIMARY
Clustered PK	Yes

Columns	Data type	Allow NULLs	Value/range
Company_name	nvarchar(60)	Not allowed	
Address1	nvarchar(100)	Not allowed	
Address2	nvarchar(100)	Allowed	
city	nvarchar(60)	Not allowed	
pin	int	Not allowed	
telephone	nvarchar(60)	Not allowed	
vatno	nvarchar(60)	Not allowed	

Indexes	Type	Columns
PK_comapny_master_company_name	clustered, unique, primary key located on PRIMARY	Comapnay_name
IX_ comapny_master_pin	nonclustered located on PRIMARY	pin

Table : user_master

Owner:	dbo
Destination DB name:	ebill
Number of columns:	3
Number of indexes:	2
Number of foreign keys:	0
Extended attributes:	
OnFileGroup	PRIMARY
Clustered PK	Yes

Columns	Data type	Allow NULLs	Value/range
user_name	nvarchar(60)	Not allowed	
User_password	nvarchar(60)	Not allowed	
User_type	nvarchar(60)	Allowed	

Indexes	Type	Columns
PK_user_master_user_name	clustered, unique, primary key located on PRIMARY	user_name
IX_ user_master_user_type	nonclustered located on PRIMARY	User_type

4. Implementation

FrmLogin

```
Dim Rs As New ADODB.Recordset
   Private Sub Command1_Click()
      '>>> check wheather user name and password are blank
      '>>> if its is blan warn user to enter
      If TxtUserName.Text = "" Or TxtPassword.Text = "" Then
         MsgBox("Enter user name and password ...", vbExclamation)
         TxtUserName.SetFocus()
         Exit Sub
      End If

      '>>> check for entered company
      '>>> query to database and if no record found warn user to select company from the
list.
      If Rs.State = adStateOpen Then Rs.Close()
      Rs.Open("select * from company_master where company_name='" &
CmbCompanyName.Text & "'", Cn, adOpenStatic, adLockReadOnly)
      If Rs.RecordCount > 0 Then
         CompanyName = CmbCompanyName.Text
      Else
```

```
            MsgBox("Select company name from the list", vbExclamation)
            CmbCompanyName.SetFocus()
            Exit Sub
        End If

        '>>> check for username and password
        '>>> query to user_master with user_name and password
        '>>> if no record found check warn user for enter valid user namne and password
        '>>> if record found store user_nmae, user_type in global variable for future use.
        If Rs.State = adStateOpen Then Rs.Close()
        Rs.Open("select * from user_master where USER_name ='" & TxtUserName.Text
& "' and user_password ='" & TxtPassword & "'", Cn, adOpenStatic, adLockReadOnly)
        If Rs.RecordCount > 0 Then
            CheckLogin = True
            UserName = IIf(IsNull(Rs("USER_name").Value) = True, "NA",
Rs("USER_name").Value)
            UserType = IIf(IsNull(Rs("user_type").Value) = True, "NA",
Rs("user_type").Value)
            Unload(Me)
        Else
            MsgBox("Invalid User Name and Password ... ", vbExclamation, "Login Error ")
            TxtPassword.Text = ""
            TxtUserName.SetFocus()
            Exit Sub
        End If
    End Sub

    Private Sub Command2_Click()
        '>>> close the application
        End
        FrmLogin = Nothing
    End Sub
```

```
Private Sub Form_Load()
    '>>> open the global connection
    If Cn.State = 1 Then Cn.Close()
    OpenCon()
    '>>> center the form
Me.Move (Screen.Width - Width) / 2, (Screen.Height - Height) / 2

    '>>> fill the combo box with all company_name from company master
    If Rs.State = adStateOpen Then Rs.Close()
    Rs.Open("select * from company_master ", Cn, adOpenStatic, adLockReadOnly)
    CmbCompanyName.Clear()
    If Rs.RecordCount > 0 Then
        While Rs.EOF = False
            CmbCompanyName.AddItem(Rs("company_name"))
            Rs.MoveNext()
        End While
    End If
    If Rs.State = adStateOpen Then Rs.Close()
End Sub

Private Sub Form_QueryUnload(ByVal Cancel As Integer, ByVal UnloadMode As
Integer)
    '>>> release all the object variable used by form
    FrmLogin = Nothing
End Sub
```

FrmBillSummary

(Show bill summary for particular date range)

```
'************************************
    '          eBilling System
```

```
'           Version 1.0.0
'************************************
'       Show summery of bill
'     Used Table : bill
'                : bill_details
'show bill summary for seleted date
'range, show report in crystal report
'move data into temp table and show
'report from temp table
'************************************


Option Explicit

   Private Sub CmdClose_Click()
      '>>> close the form
      Unload(Me)
   End Sub

   Private Sub CmdGetBill_Click()
      'NOTE : it is not the right solution to call crystal report by temp using temp table
      'some time it is a good practice for complecated databse relation table
      'This may not run properly in multi user environment
      'Better approch is passing value by SelectionFormula in crystal report
      'but anyway it is a working solution
      '>>> find the bill sno from seleted invoice no
      '>>> if record found
      '>>> delete temp bill na dbill_details
      '>>> insert from bill,bill_details to temp_bill, teemp_bill_details

      Cn.Execute("delete from temp_bill_details")
      Cn.Execute("delete from temp_bill")
```

```
    Cn.Execute("insert into temp_bill select * from bill  where invoice_date>=#" &
Format(DTPicker1.Value, "dd-mmm-yy") & "# and invoice_date<=#" &
Format(DTPicker2.Value, "dd-mmm-yy") & "# and cname='" & CompanyName & "' ")
    Cn.Execute("insert into temp_bill_details select * from bill_details where bill_sno in
(  select sno from bill  where  invoice_date >=#" & Format(DTPicker1.Value, "dd-mmm-
yy") & "# and invoice_date<=#" & Format(DTPicker2.Value, "dd-mmm-yy") & "# and
cname='" & CompanyName & "')")
    Call OpenCon()
    '>>> open crystal report
    Cr1.DataFiles(0) = App.Path & "\data.mdb"
    Cr1.WindowState = crptMaximized
    Cr1.ReportFileName = App.Path & "\reports\billsummary.rpt"
    Cr1.Action = 1
End Sub
Private Sub Form_Load()
    '>>> cnter the form
    Me.Left = (Screen.Width - Me.Width) / 2
    Me.Top = (Screen.Height - Me.Height) / 2
    '>>> show the current date
DTPicker1.Value = Date
DTPicker2.Value = Date
End Sub
```

FrmExportData

(Export product Data in FlexGrid and MS-Excel with formatting)

```
'************************************
'          eBilling System
'           Version 1.0.0
'************************************
' Export product_master in grid and ms-excel
'     Used Table : product_master
'open the product_master in grid
```

```
'display record in flex grid with gropu by product type
'************************************
Option Explicit

Private Sub CmdClose_Click()
'>>> close the form
Unload(Me)
End Sub

Private Sub CmdExcel_Click()
'>>>export data into ms excel from grid with formatting
'>>> check the grid
If Mf1.TextMatrix(0, 0) = "" Then
MsgBox("No Records Available for Exporting ... ", vbExclamation)
Exit Sub
End If
Label1.Caption = "WAIT ... Generate Excel "
Label1.Refresh()

'>>> creating excel object variable
Dim ex As New Excel.Application
Dim wb As New Workbook
Dim Es As New Worksheet
wb = ex.Workbooks.Add
Es = wb.Worksheets(1)
Dim i As Integer
Dim j As Integer
'>>> set excel columns width as per flex grid columns width
For i = 0 To Mf1.Cols - 1
Mf1.Row = 1
Mf1.Col = i
```

```
        Es.Columns(ReturnAlphabet(i + 1) & ":" & ReturnAlphabet(i + 1)).ColumnWidth
= Mf1.CellWidth / 110
    Next
    '>>> set data from grid to excel row, column wise
    Dim K As Integer
    For i = 0 To Mf1.Rows - 1
       For j = 0 To Mf1.Cols - 1
          ex.Cells(i + 1, j + 1) = Mf1.TextMatrix(i, j)
       Next
    Next

    Dim R1 As String
    Dim R2 As String
    R2 = ReturnAlphabet(Mf1.Cols) & "1"

    '>>> formatting excel
    Dim x As Range
    '>>>head
    x = Es.Range("A1:" & R2)
    x.Font.Bold = True
    x.Font.ColorIndex = 40
    x.Interior.ColorIndex = 9
    x.Interior.Pattern = xlSolid
    x.HorizontalAlignment = xlCenter
    x.VerticalAlignment = xlBottom
    '>>>border
    R2 = ReturnAlphabet(Mf1.Cols) & Mf1.Rows - 1
    x = Es.Range("A1:" & R2)

    With x.Borders(xlEdgeLeft)
       .LineStyle = xlContinuous
       .Weight = xlThin
```

```
    .ColorIndex = xlAutomatic
End With
With x.Borders(xlEdgeTop)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With x.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With x.Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With x.Borders(xlInsideVertical)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With x.Borders(xlInsideHorizontal)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
'>>> FILL DETAILS
x = Es.Range("A2:" & R2)
x.Interior.ColorIndex = 40
'>>TOTAL
R2 = ReturnAlphabet(Mf1.Cols) & Mf1.Rows
```

```
    x = Es.Range("A" & Mf1.Rows & ":" & R2)
    x.Font.Bold = True
    x.Font.ColorIndex = 9

   Es.Name = "Report"
    ex.Visible = True

    ex.Quit()
    wb = Nothing
    Es = Nothing
    ex= Nothing
    '>>> process complete
    Label1.Caption = "Ready"
    Label1.Refresh()

End Sub

Private Sub CmdProductMaster_Click()
    '>>> reset the grid

    Mf1.Rows = 2
    Mf1.Cols = 3
    Mf1.Clear()
    Mf1.Refresh()
    Mf1.Row = 0
    Mf1.Col = 0
    Mf1.ColWidth(0) = 800
    Mf1.Text = "Sr"
    Mf1.CellAlignment = 4
    Mf1.CellFontName = "Arial"
    Mf1.Font.Size = 10
    Mf1.Font.Bold = True
```

```
        Mf1.CellForeColor = vbBlue
        Mf1.CellBackColor = vbCyan
        Mf1.Col = 1
        Mf1.ColWidth(1) = 2500
        Mf1.Text = "Product Type"
        Mf1.CellAlignment = 4
        Mf1.CellFontName = "Arial"
        Mf1.Font.Size = 10
        Mf1.Font.Bold = True
        Mf1.CellForeColor = vbBlue
        Mf1.CellBackColor = vbCyan
        Mf1.Col = 2
        Mf1.ColWidth(2) = 5000
        Mf1.Text = "Product"
        Mf1.CellAlignment = 4
        Mf1.CellFontName = "Arial"
        Mf1.Font.Size = 10
        Mf1.Font.Bold = True
        Mf1.CellForeColor = vbBlue
        Mf1.CellBackColor = vbCyan

        '>>> find distinct product type from product master
        '>>> loop all product type
        Dim RS1 As New ADODB.Recordset
        Dim Rs2 As New ADODB.Recordset
        RS1.Open("select distinct prod_type from product_master", Cn, adOpenStatic,
adLockReadOnly)
        Dim i As Integer
        Dim j As Integer
        For i = 0 To RS1.RecordCount - 1
            Me.Caption = i + 1
            Mf1.Row = Mf1.Rows - 1
```

```
        Mf1.Col = 0
        Mf1.Text = i + 1
        Mf1.Col = 1
        Mf1.Text = RS1("prod_type")
        '>>> query product master for each prod type from outer loop
        If Rs2.State = adStateOpen Then Rs2.Close()
        Rs2.Open("select prod_sub_type from product_master where prod_type ='" &
RS1("prod_type") & "' order by prod_sub_type", Cn, adOpenStatic, adLockReadOnly)
        For j = 0 To Rs2.RecordCount - 1
          Mf1.Row = Mf1.Rows - 1
          Mf1.Col = 2
          Mf1.Text = Rs2(0)
          Mf1.Rows = Mf1.Rows + 1
          Rs2.MoveNext()
        Next
        RS1.MoveNext()
      Next
  End Sub
  Private Sub Form_Load()
    '>>> cnter the form
    Me.Left = (Screen.Width - Me.Width)
    Me.Top = (Screen.Height - Me.Height)

  End Sub
```

FrmCompanyInfo

(Add/Modify Company Info)

```
  '************************************
  '          eBilling System
  '           Version 1.0.0
  '************************************
  ' add/edit/delete company details
  '     Used Table : company_master
```

```
    'open the company_master
    'display first record in form load
    'add edit save delete and navigation
    '************************************

Option Explicit
    Dim RS1 As New ADODB.Recordset
    Dim AddEdit As String

    Private Sub Command1_Click()
        '>>> close the form
        Unload(Me)
    End Sub

    Private Sub Command11_Click()
        '>>> delete the record
        If RS1.State = adStateClosed Then Exit Sub
        If RS1.RecordCount <= 0 Then Exit Sub

        On Error GoTo myer1
        '>>> confirm before delete
        If MsgBox("Delete the Record ? ", vbCritical + vbYesNo) = vbYes Then
            RS1.Delete()
            Call ClearText()
            Command4_Click()
        End If
        Exit Sub
myer1:
        MsgBox("Error Occured : " & Err.Description, vbCritical)
    End Sub

    Private Sub Command2_Click()
```

```
    '>>> move record ponter to first record
    '>>> display first record
    On Error Resume Next
    If RS1.State = adStateClosed Then Exit Sub
    If RS1.RecordCount <= 0 Then Exit Sub
    RS1.MoveFirst()
    Call DisplayRecord()
End Sub

Private Sub Command3_Click()
    '>>> move back the record pointer and display current record
    On Error Resume Next
    If RS1.State = adStateClosed Then Exit Sub
    If RS1.RecordCount <= 0 Then Exit Sub
    If RS1.AbsolutePosition > 1 Then
        RS1.MovePrevious()
    Else
        MsgBox("First Record ..", vbInformation)

        RS1.MoveFirst()
    End If
    Call DisplayRecord()

End Sub

Private Sub Command4_Click()
    '>>> move next the record pointer and display current record
    On Error Resume Next
    If RS1.State = adStateClosed Then Exit Sub
    If RS1.RecordCount <= 0 Then Exit Sub
    If RS1.AbsolutePosition < RS1.RecordCount Then
        RS1.MoveNext()
```

```
    Else
        MsgBox("Last Record ..", vbInformation)

        RS1.MoveLast()
    End If
    Call DisplayRecord()

End Sub

Private Sub Command5_Click()
    '>>> move last the record pointer and display current record
    On Error Resume Next
    If RS1.State = adStateClosed Then Exit Sub
    If RS1.RecordCount <= 0 Then Exit Sub
    RS1.MoveLast()
    Call DisplayRecord()

End Sub

Private Sub Command6_Click()
    '>>> prepare for add record, clear all text box, set flag to ADD
    If RS1.State = adStateClosed Then Exit Sub
    If RS1.RecordCount <= 0 Then Exit Sub
    AddEdit = "ADD"
    Call ClearText()
    DE(False, True)
    TxtCompanyName.SetFocus()
End Sub

Private Sub Command7_Click()
    '>>> prepare for edit record,  set flag to EDIT
    If RS1.State = adStateClosed Then Exit Sub
```

```
    If RS1.RecordCount <= 0 Then Exit Sub

    AddEdit = "EDIT"
    DE(False, True)
    TxtCompanyName.SetFocus()
End Sub

Private Sub Command8_Click()
    '>>> save the record
    '>>> check for validation
    '>>> check the flag for ADD/Edit
    If RS1.State = adStateClosed Then Exit Sub
    If RS1.RecordCount <= 0 Then Exit Sub

    On Error GoTo myer1
    If Trim(TxtCompanyName.Text) = "" Then
        MsgBox("Enter Company Name  ", vbCritical)
        TxtCompanyName.SetFocus()
        Exit Sub
    End If
    If AddEdit = "ADD" Then
        RS1.AddNew()
        RS1("company_name") = TxtCompanyName.Text
        RS1("Address1") = TxtAddress1.Text
        RS1("Address2") = TxtAddress2.Text
        RS1("city") = TxtCity.Text
        RS1("pin") = TxtPin.Text
        RS1("telephone") = TxtTelephone.Text
        RS1("vatno") = TxtVatNo.Text

        RS1.Update()
        RS1.MoveLast()
```

```
            Call DisplayRecord()
        Else
            RS1("Address1") = TxtAddress1.Text
            RS1("Address2") = TxtAddress2.Text
            RS1("city") = TxtCity.Text
            RS1("pin") = TxtPin.Text
            RS1("telephone") = TxtTelephone.Text
            RS1("vatno") = TxtVatNo.Text
            RS1.Update()

            '>>> if it is edit after requery show the edited record

            Dim p As Integer
            p = RS1.AbsolutePosition
            RS1.Requery()
            RS1.MoveFirst()
            RS1.Move(p - 1)
            Call DisplayRecord()
        End If
        DE(True, False)
        Exit Sub
myer1:
        MsgBox("Error Occured : " & Err.Description, vbCritical)
    End Sub

    Private Sub Command9_Click()
        '>>> cancel save
        DE(True, False)
    End Sub

    Private Sub Form_Load()
        '>>> center the form
```

```
    Me.Move (Screen.Width - Width) / 2, (Screen.Height - Height) / 2
        '>>> reset connection
        '>>> clear all text
        ClearText()
        OpenCon()
        '>>> load alreday saved clent data
        '>>> and show the first record

        If RS1.State = adStateOpen Then RS1.Close()
        RS1.Open("select * from company_master order by company_name ", Cn,
adOpenDynamic, adLockOptimistic)
        If RS1.RecordCount > 0 Then
            RS1.MoveFirst()
            Call DisplayRecord()
        End If
        DE(True, False)
    End Sub

    Private Sub ClearText()
        '>>> clear all text box in the form
        Dim Ctl As Control
        For Each Ctl In Me.Controls
            If TypeOf Ctl Is TextBox Then
                Ctl.Text = ""
            End If
        Next
    End Sub

    Private Sub DisplayRecord()
        '>>> display current record
        On Error Resume Next
        Call ClearText()
```

```
    TxtCompanyName.Text = IIf(IsNull(RS1("company_name")) = True, "",
RS1("company_name"))
    TxtAddress1.Text = IIf(IsNull(RS1("Address1")) = True, "", RS1("Address1"))
    TxtAddress2.Text = IIf(IsNull(RS1("Address2")) = True, "", RS1("Address2"))
    TxtCity.Text = IIf(IsNull(RS1("city")) = True, "", RS1("city"))
    TxtPin.Text = IIf(IsNull(RS1("pin")) = True, "", RS1("pin"))
    TxtTelephone.Text = IIf(IsNull(RS1("telephone")) = True, "", RS1("telephone"))
    TxtVatNo.Text = IIf(IsNull(RS1("vatno")) = True, "", RS1("vatno"))

    Label17.Caption = RS1.AbsolutePosition & "/" & RS1.RecordCount
  End Sub

  Private Sub DE(ByVal T1 As Boolean, ByVal T2 As Boolean)
    '>>> enable disable buttons
    Command2.Enabled = T1
    Command3.Enabled = T1
    Command4.Enabled = T1
    Command5.Enabled = T1
    Command6.Enabled = T1
    Command7.Enabled = T1
    Command11.Enabled = T1
    Command8.Enabled = T2
    Command9.Enabled = T2
```

5. Testing

System Development and Testing:

System Development is a process of conceiving the specification specified in the designing stage into source code. Careful study and observation about system designing were made and accordingly the system was coded to convert from designing to source code, where visual Basic as the front end and OracleXE as the backend. The System was developed such that it should be used for the future enhancement.

All the module of the system is combined and is put to the operational use. This means that the new and old system are run in the parallel for sometimes, errors are identified and the corresponding errors are to be concerned to get the required output.

The set of working programs and initialized tables are also provided for the easy start of the user, in addition, system documentation is also provided, all users have been trained to use the system.

This creates two problems,

- The time lag between the cause and appearance of the problem.
- The effect of system errors on files and records within the system.

Types of testing done:

5.1 Unit Testing:

Unit test is designed to ensure that the purpose for which it was designed for which it was designed for is fulfilled. Each and every module was tested individually with the test data and error messages were displayed for incorrect and sufficient for entry works. All validation was tested to correctness. Test data were fed in and results were checked for the maintenance module, to ensure that all tables created contained nothing but valid data. Reverential integrity constraints specified as part of the table definition was also tested.

5.2 Recovery Testing:

Many computer based systems must recover from faults and resume processing within a pre-specified time. In some cases a system must be fault tolerant.ie processing faults must not cause overall system function to cease. In the casers a system failure must be corrected within a specified period of time or severe economic damage will occur.

5.3 Security Testing:

Any computer-based system the manages sensitive information or cause action that can improperly harm individual is a tablet for improper or illegal penetration Security testing attempts to verify that protection mechanism built into a system will, in fact protect it from improper penetration . During security testing, the tester plays the role of the individual who desire to penetrate the system. The tester may attempt to acquire passwords through external clerical means; may attack the system with custom software designed to break down any defenses that have been constructed; may overwhelm the System.

5.4 Performance Testing:

For real time and embedded system, software that provides required functions but not confirm to performance requirements is unacceptable. Performance testing is designed to test the run time performance of software within the context of an integrated system. Performance testing occurs throughout all steps in the testing process. Even at unit level, the performance of an individual module may be accessed as white box test recon ducted. However, it is not until all system elements are fully integrated that true performance of a system can be ascertained.

Performance Tests are sometimes coupled with stress testing and often required other hardware and software implementation. It is often necessary to measure Resource utilization .By incrementing a system the tester can uncover situations that lead to degradation and possible system failure.

5.5 Integration Testing

In integration testing a system consisting of different modules is tested for problems arising from component interaction. Integration testing should be developed from the system specification. Firstly, a minimum configuration must be integrated and tested. In my project I have done integration testing in a bottom up fashion i.e. in this project I have started construction and testing with atomic modules. After unit testing the modules are integrated one by one and then tested the system for problems arising from component interaction.

5.6 White Box Testing

In white box testing knowing the internal working of the base, test can be conducted to ensure that internal operations are performed according to specification and all internal components have been adequately exercised. In white box testing logical path through the software are tested by providing test cases that exercise specific set of conditions and loops. Using white-box testing software developer can derive test case that

- Guarantee that all independent paths within a module have been exercised at least

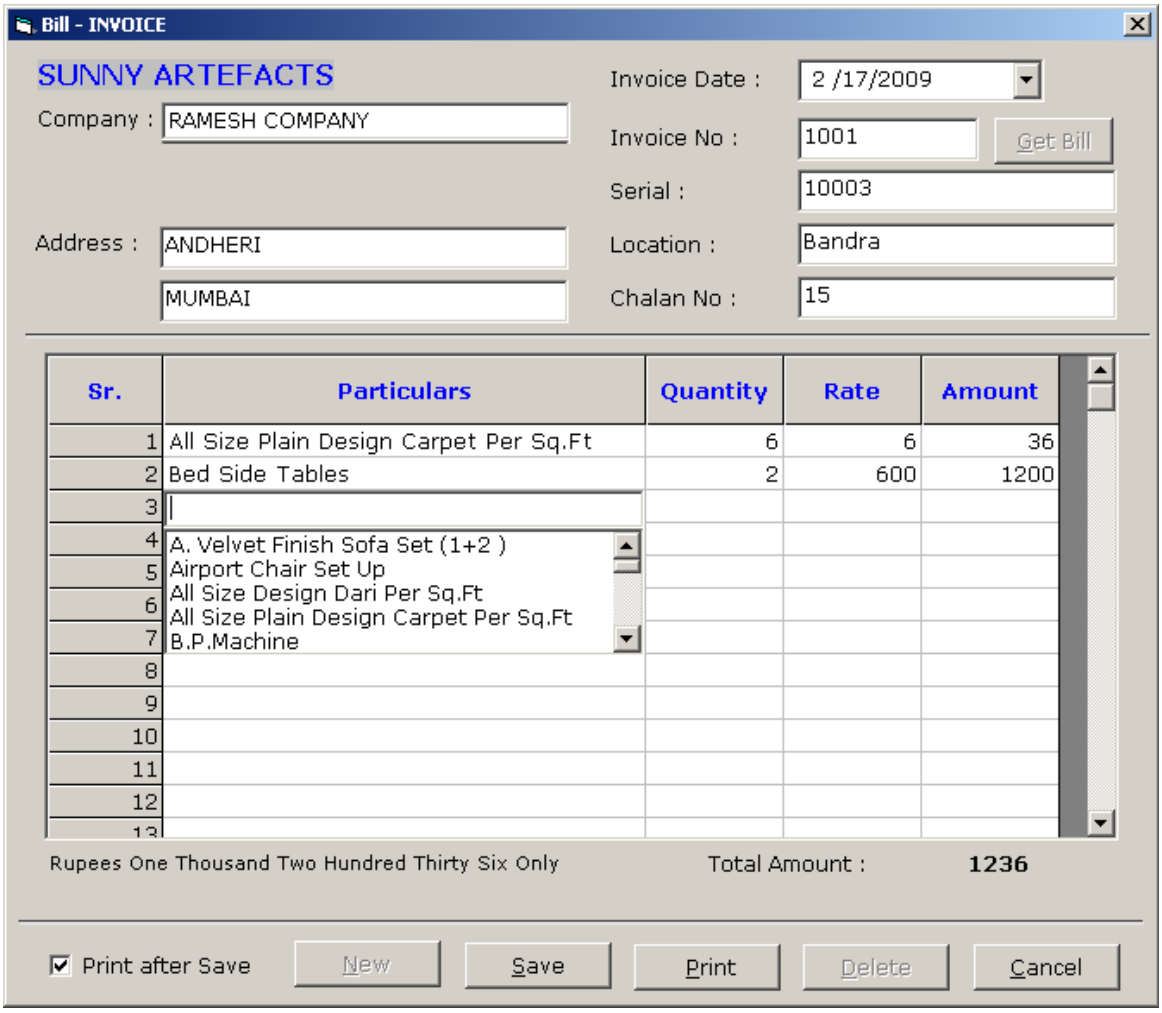
once.

- Exercise all logical decisions on their true and false side.
- Exercise all loops at their boundaries and within their operational bound.
- Exercise internal data structure to ensure their validity.

6. Experimental Results

FrmLogin

(Module to Authenticate User)

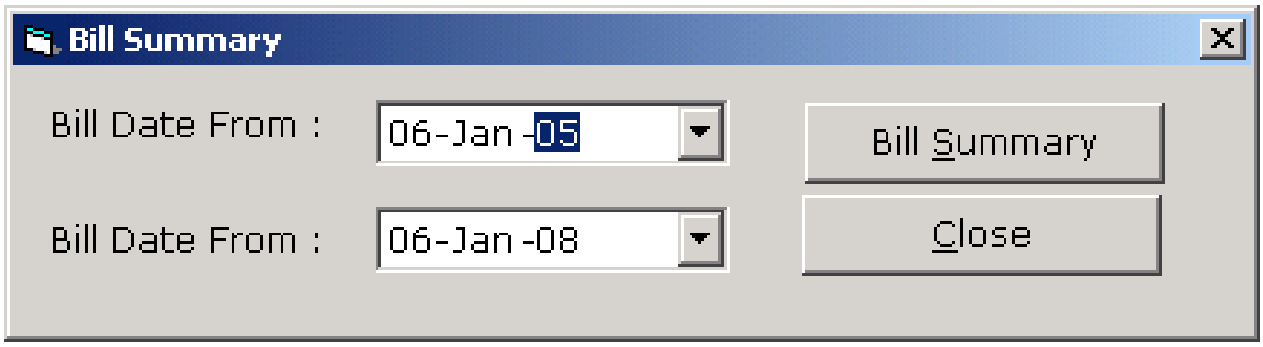


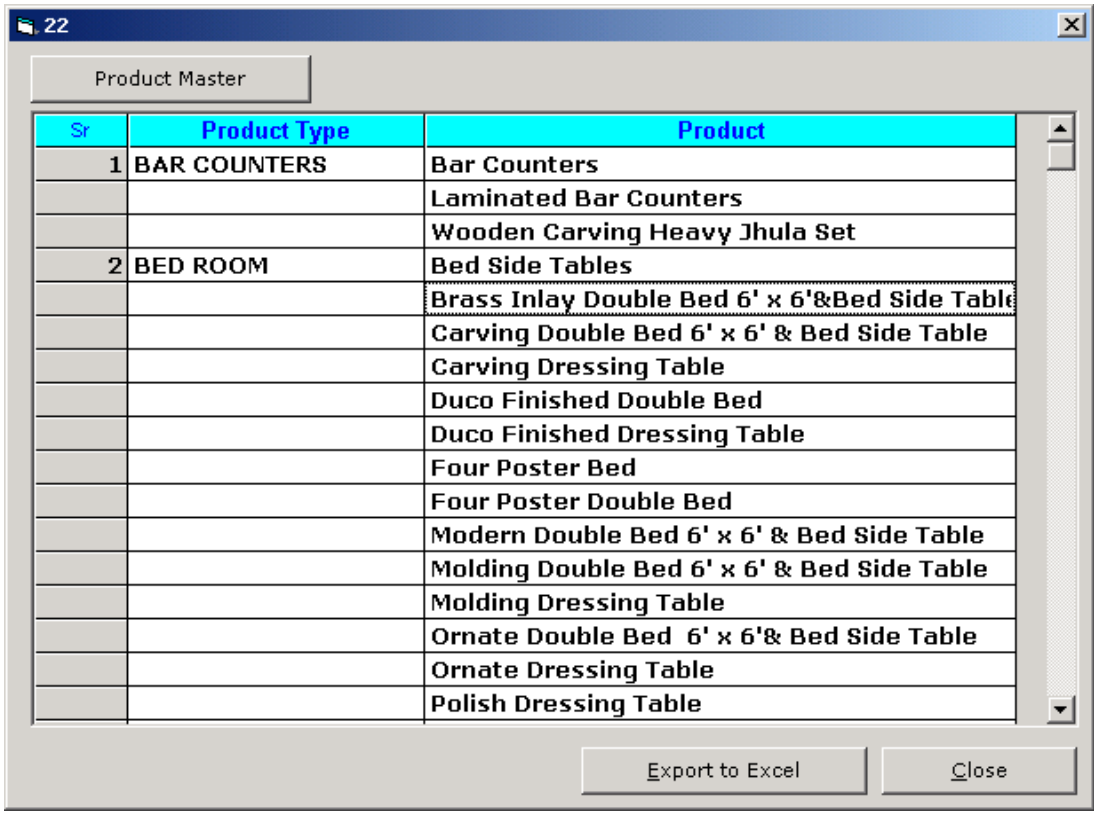
FrmBill

(Create and modify bill and print the saved bill)

FrmBillSummary

(Show bill summary for particular date range)



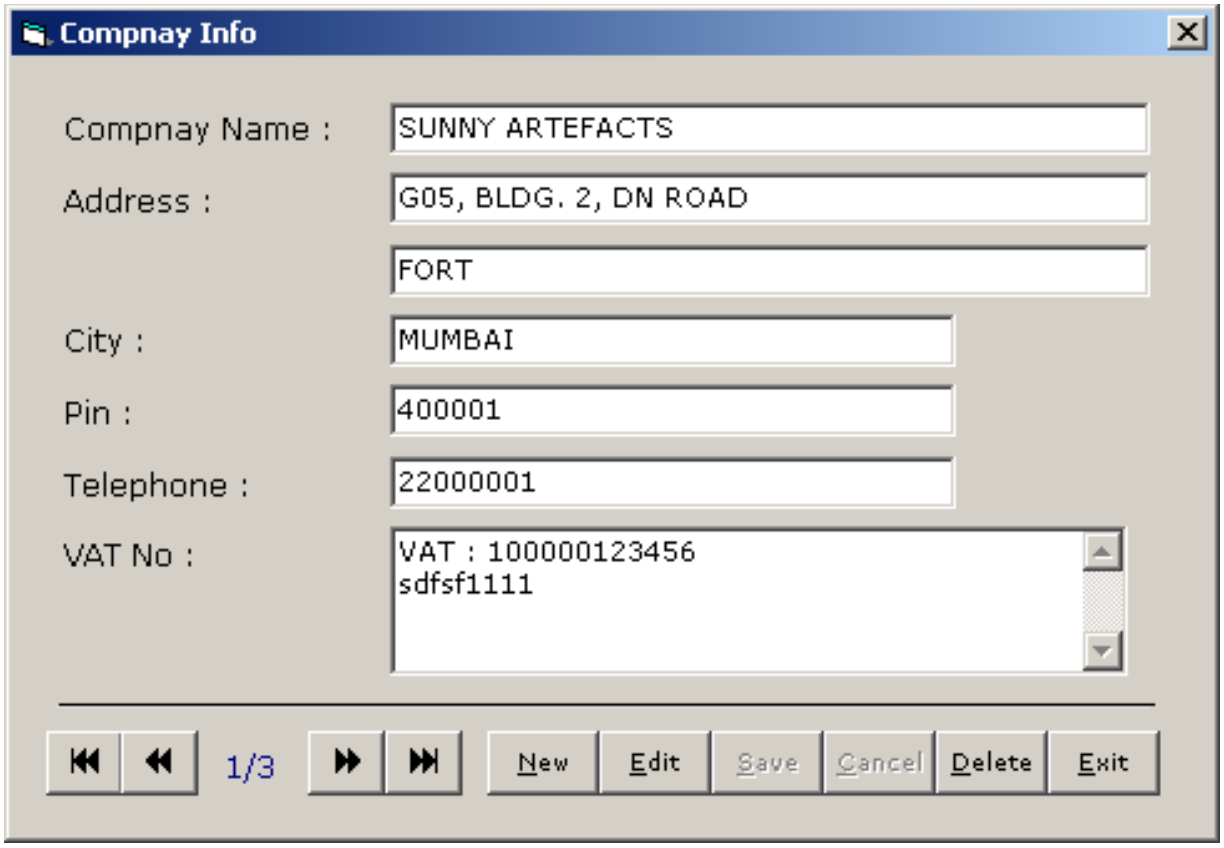


FrmExportData

(Export product Data in FlexGrid and MS-Excel with formatting)

FrmCompanyInfo

(Add/Modify Company Info)



7. Conclusion

This project was developed to fulfill user and business requirement; however there are lots of scope to improve the performance of the eBilling and Invoice System in the area of user interface, database performance, and query processing time. Etc.

So there are many things for future enhancement of this project. The future enhancements that are possible in the project are as follows.

- Linking and integration of any legacy system for accounting.
- Integration with travel agent through Web Services
- Connection to third-party OLAP applications
- Electronic Data Interchange (EDI) system between banks, other credit verification agency and their vendors
- In the area of data security and system security.
- Provide more online tips and help.
- To optimize the query which is embedded in the system

8. References

Websites

- http://www.google.com
- http://www.microsoft.com
- http://www.programmer2programmer.net
- http://www.codeproject.com
- http://www.msdn.com.
- http://www.vb123.com
- http://www.vbcode.com
- http://www.sqltuner.com

Books

- Mastering Visual Basic 6 (Paperback)
- Mastering Visual Basic .NET (Paperback)
- Visual Basic Black Book (Paperback)
- SQL Bible, 2nd Edition (Paperback)
- Database Development in Visual Basic