**CHAPTER-1**

1. **INTRODUCTION**

**1.1 Overview of project**

The "**Medical Inventory**" has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and in some cases reduce the hardships faced by this existing system. Moreover this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus by this all it proves it is user-friendly. Medical Inventry System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help organization in better utilization of resources.

Every organization, whether big or small, has challenges to overcome and managing the information of Medicines, Pharmacy, Company, Sells, Inventory. Every medical inventrySystem has different medical needs, therefore we design exclusive employee management systems that are adapted to your managerial requirements. This is designed to assist in strategic planning, and will help you ensure that your organization is equipped with the right level of information and details for your future goals. Also, for those busy executive who are always on the go, our systems come with remote access features, which will allow you to manage your workforce anytime, at all times. These systems will ultimately allow you to better manage resources

**1.2 keywords**

Medical, sell, stock, customer, amount

**1.3 Objective of the project**

The main objective of the Project on Pharmacy Management System is to manage the details of Pharmacy, Medicines, Stocks, Company, Inventory. It manages all the information about Pharmacy, Sells, Inventory, Pharmacy. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the Pharmacy, Medicines, Sells and Stocks. It tracks all the details about the Stocks, Company and Inventory.

**1.4 Statement of the problem**

The Software Requirements Specification is produced at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by establishing a complete information description, a detailed functional and behavioral description, an indication of performance requirements and design constraints, appropriate validation criteria, and other data pertinent to requirements.

Documents and reports that must be provided by the new system: there can also be few reports, which can help management in decision-making and cost controlling, but since these reports do not get required attention, such kind of reports and information were also identified and given required attention.

* Details of the information needed for each document and report.
* The required frequency and distribution for each document.
* Probable sources of information for cach document and report.

With the implementation of computerized system, the task of keeping records in an organized manner will be solved. The greatest of all is the retrieval of information, which will be at the click of the mouse. So the proposed system helps in saving the time in different operations and making information flow easy giving valuable reports.

**1.5 Methodology**

by using this system user can easily store the data of the exicting stock and upcoming stock

so that it makes a clear of the data by it which the work easy of the and even stock supplier .

In the existing system the exams are done only manually but in proposed system we have to computerize the exams using this application

System needs store information about new entry of Pharmacy

System needs to help the internal staff to keep information of Medicines and find them as per various queries.

* System need to maintain quantity record.
* System need to keep the record of Stocks
* System need to update and delete the record.
* System also needs a search area
* It also needs a security system to prevent data.

**1.6 Limitations**

* Lack of security of data
* More man power
* Time consuming
* Consumes large volume of pare work.
* Needs manual calculations.
* No direct role for the higher officials

**CHAPTER-2**

**2.LITERATURE SURVEY**

It is the most creative and challenging phase of the system life cycle. The analysis phase is used to design the logical model of the system whereas the design phase is used to design the physical model

Many things are to be done in this phase we began the designing process by identifying forms, reports and the other outputs the system will produce. Then the specify data on each were pinpointed we sketched the forms or say, the displays, as expected to appear, on paper, so it serves as model for the project to began finally we design the form on computer display, using one of the automated system design tool, that is VISUAL BASIC 6.0.

After the forms were designed, the next step was to specify the data to be inputted, calculated and stored individual data items and calculation procedure were written in detail. File structure such as paper files were selected the procedures were written so as how to process the data and procedures the output during the programming phase. The documents were design ion the form of charts

Output design means what should be the format for presenting the results. It should be in most convenient and attractive format for the user. The input design deals with what should be the input to the system and thus prepare the input format. File design deals with how the data has to be stored on physical devices. Process design includes the description of the procedure for carrying out operations on the given data.

**CHAPTER 3**

**3.HARDWARE AND SOFTWARE REQUIREMENTS**

**HARDWARE REQUIREMENTS**

* System : Processor Intel(R) Core(TM) 13-3110M CPU @ 2.40GHz 2.40GHz.
* Hard ware : 40GB hard disk.
* Floppy disk : 1.44Mb.
* Monitor : 15 VGA colour.
* Mouse : Logitech.
* Ram : 128 Mb.

**SOFTWARE REQUIREMENTS**

* Operating system : - Windows 7 and above
* Coding Language : Microsoft Visual Studio 2010
* Data Base : Microsoft SQL Server Database

**CHAPTER 4**

**4. SYSTEM REQUIREMENT SPECIFICATIONS**

The system services and goals are established by consultation with system user. They are then defined in details and serve as a system specification. System requirement are those on which the system runs.

The Software Requirements Specification is produced at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by establishing a complete information description, a detailed functional and behavioral description, an indication of performance requirements and design constraints, appropriate validation criteria, and other data pertinent to requirements.

Documents and reports that must be provided by the new system: there can also be few reports, which can help management in decision-making and cost controlling, but since these reports do not get required attention, such kind of reports and information were also identified and given required attention.

* Details of the information needed for each document and report.
* The required frequency and distribution for each document.
* Probable sources of information for cach document and report.

With the implementation of computerized system, the task of keeping records in an organized manner will be solved. The greatest of all is the retrieval of information, which will be at the click of the mouse. So the proposed system helps in saving the time in different operations and making information flow easy giving valuable reports

**EXISITING SYSTEM**

In the existing system the exams are done only manually but in proposed system we have to

computerize the exams using this application

* Lack of security of data
* More man power
* Time consuming
* Consumes large volume of pare work.
* Needs manual calculations.
* No direct role for the higher officials

**PROPOSED SYSTEM**

System needs store information about new entry of Pharmacy

System needs to help the internal staff to keep information of Medicines and find them as per various queries.

* System need to maintain quantity record.
* System need to keep the record of Stocks
* System need to update and delete the record.
* System also needs a search area
* It also needs a security system to prevent data.

**3.1 Dataflow Diagram:**

A graphical tool used to describe and analysis the movement of data through a system is called dataflow diagram. Dataflow diagram is the central basis, from which other components are developed, the transformation of data from input to output, through process, maybe described logically and independent of the physical components associated with the system.

The DFD is also known as a dataflow, graph or a bubble chart.

Types of dataflow diagram

* Physical DFD
* Logical DFD

**3.1.1. Physical DFD:**

Structure analysis states that current system should first understand correctly. The physical DFD is the model of current system and is used to ensure that system has been clearly understood. Physical DFD's shows actual devices, departments and people etc. involved in the current system

**3.1.2. Logical DFD:**

Logical DFD are the model of proposed system. They clearly should show the requirement on which the new system should be built. Later during the design activity thus taken has basis for drawing the systern structure charts.

The basic notation used to create DFD's are:

**DATA FLOW**: Data move in a specific direction from an origin to a destination. The

physical component is not identified.

**SOURCE:** External sea or destination of data, which maybe people, organization other

**DATA STORAGE**: Here data are stored or referenced by a processor in the system.

Design is the first step in moving from domain to the solution domain Design is especially the bridge between requirement specification and the final solution. The goal of the design process is to produce a model for representation of a system, which can be used later to build that system. The produced model is called the "design of the system"

It is the solution for the system. In this level the focus is on dividing the module which is needed for the system and the specification of these models.

**3.2 Data flow diagram**

3.3 LEVEL 1 DFD

Manage supplier details

Manage customer details

Manage product details

Sales entry

Purchase entry

transaction

Manage stock reports

user

3.4 Module

Login:

|  |  |
| --- | --- |
| NAME | DATA TYPE |
| User name | Nvarchar(50) |
| password | Nvarchar(50) |

Category Entry:

|  |  |
| --- | --- |
| Name | Data type |
| Category name | Nvarchar(50) |
| Category description | Nvarchar(50) |

Group entry:

|  |  |
| --- | --- |
| Name | Data type |
| Group name | Nvarchar(50) |
| Group description | Nvarchar(50) |

|  |  |
| --- | --- |
| Name | Data type |
| Customer id | Nvarchar(50) |
| Customer name | Nvarchar(50) |
| Address | Nvarchar(50) |
| Pin code | Nvarchar(50) |
| Phone no | Nvarchar(50) |
| E mail | Nvarchar(50) |

Customer entry:

Supplier entry:

|  |  |
| --- | --- |
| Name | Data type |
| Supplier id | Nvarchar(50) |
| Supplier name | Nvarchar(50) |
| Address | Nvarchar(50) |
| Pin code | Nvarchar(50) |
| Phone no | Nvarchar(50) |
| E mail | Nvarchar(50) |

Product entry:

|  |  |
| --- | --- |
| Name | Data type |
| Item code | Nvarchar(50) |
| Item name | Nvarchar(50) |
| Category name | Nvarchar(50) |
| Group name | Nvarchar(50) |
| Unit name | Nvarchar(50) |
| Purchase price | Nvarchar(50) |
| Selling price | Nvarchar(50) |
| Tax percentage | Nvarchar(50) |

Purchase and sales entry:

Transaction table

|  |  |
| --- | --- |
| Name | Data type |
| Bill no | Nvarchar(50) |
| Bill date | Nvarchar(50) |
| Supplier code | Nvarchar(50) |
| Total amount | Nvarchar(50) |
| Other charges | Nvarchar(50) |
| Grand total | Nvarchar(50) |
| Btype | Nvarchar(50) |
| Remark | Nvarchar(50) |

3.5 source code

Welcome form

Public Class WelcomeForm

Dim I As Integer

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

I = 1

End Sub

Private Sub Timer1\_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick

I = I + 1

If I > 5 Then

Timer1.Enabled = False

Me.Hide()

LoginForm.Show()

End If

End Sub

End Class

3.5.1 Login

Imports System Data SqlClient

Public Class LoginForm

Private Sub TextBox1 Click 1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox1.Click

TextBox1.Clear())

End Sub

Private Sub TextBox2 Click 1(ByVal sender As System.Object, ByVal e As System EventArgs) Handles TextBox2. Click

TextBox2.Clear()

TextBox2.UseSystemPasswordChar- True

End Sub

Private Sub PictureBox5\_Click(ByVal sender As System.Object, ByVal e As

System.EventArgs) Handles PictureBox5.Click

If TextBox2.UseSystemPasswordChar- True Then

TextBox2.UseSystemPasswordChar = False

Else

TextBox2.UseSystemPasswordChar = True

End If

End Sub

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button Click

If Com State-ConnectionState Open Then Conn.Close()

Conn.Open()

Dim cmd0 As New SqlCommand("select from logintable where Username =" & UCase(TextBox1.Text) &”’ and Password="" & TextBox2.Text & “””, Conn)

Dim D1 As SqlDataReader-cmd0.ExecuteReader()

IF D1.Hasrows then

MDIParent1.Show()

Me.Hide()

If Conn State=ConnectionState.Open Then Conn.Close()

Else

MsgBox("user or password is not correct please check")

End If

End Sub

Private Sub Button2\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2. Click

TextBox1.Text=””

TextBox2.Text=””

End Sub

End Class

**MDI PARENT1 3.5.2**

Public Class MDIForm1

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

CustForm.MdiParent = Me

CustForm.Show()

End Sub

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

SupplierForm.MdiParent = Me

SupplierForm.Show()

End Sub

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

mCatForm.MdiParent = Me

mCatForm.Show()

End Sub

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

mGroupForm.MdiParent = Me

mGroupForm.Show()

End Sub

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

ItemForm.MdiParent = Me

ItemForm.Show()

End Sub

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

PurchaseForm.MdiParent = Me

PurchaseForm.Show()

End Sub

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

StockRep.MdiParent = Me

StockRep.Show()

End Sub

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

PurchaseRep.MdiParent = Me

PurchaseRep.Show()

End Sub

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

SalesRep.MdiParent = Me

SalesRep.Show()

End Sub

Private Sub RiMDI\_Disposed(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Disposed

End

End Sub

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

CounterSalesForm.MdiParent = Me

CounterSalesForm.Show()

End Sub

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

CustListReport.MdiParent = Me

CustListReport.Show()

End Sub

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

Me.Hide()

LoginForm.Show()

End Sub

Private Sub MDIForm1\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

End Sub

End Class

Customer form 3.5.3

Public Class CustForm

Dim pkVar As String

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

Me.WindowState = FormWindowState.Maximized

disRecords()

End Sub

Private Sub butNew\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

ClearTxtControls(Me, 1)

TextBox1.Focus()

End Sub

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

SaveRecord()

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

End Sub

Sub SaveRecord()

If TextBox1.Text = "" Then

MsgBox("Please enter the necessary details")

Exit Sub

End If

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select custCode from CustTab where custCode='" & UCase(TextBox1.Text) & "'", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

MsgBox("This record is allready present in the database")

If Conn.State = ConnectionState.Open Then Conn.Close()

Exit Sub

End If

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

q1Var = "insert into CustTab("

q2Var = " values("

q1Var = q1Var & "custCode" & ","

q2Var = q2Var & "'" & UCase(TextBox1.Text) & "',"

q1Var = q1Var & "custname" & ","

q2Var = q2Var & "'" & TextBox2.Text & "',"

q1Var = q1Var & "Add1" & ","

q2Var = q2Var & "'" & TextBox3.Text & "',"

q1Var = q1Var & "Add2" & ","

q2Var = q2Var & "'" & TextBox4.Text & "',"

q1Var = q1Var & "Add3" & ","

q2Var = q2Var & "'" & TextBox5.Text & "',"

q1Var = q1Var & "PinCode" & ","

q2Var = q2Var & "'" & TextBox6.Text & "',"

q1Var = q1Var & "PhoneNo" & ","

q2Var = q2Var & "'" & TextBox7.Text & "',"

q1Var = q1Var & "eMail" & ")"

q2Var = q2Var & "'" & TextBox8.Text & "')"

'MsgBox(q1Var & q2Var)

Dim cmd1 As New SqlCommand(q1Var & q2Var, Conn)

cmd1.ExecuteNonQuery()

MsgBox("details entered successfully")

If Conn.State = ConnectionState.Open Then Conn.Close()

'disRecords()

End Sub

Sub disRecords()

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim DS1 As New DataSet

Dim adp As New SqlDataAdapter("Select custCode,CustName From CustTab order by custCode", Conn)

adp.Fill(DS1)

DG1.DataSource = DS1.Tables(0)

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

Private Sub DG1\_CellContentClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DG1.CellContentClick

pkVar = DG1.CurrentRow.Cells(0).Value

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select \* from CustTab where custCode='" & pkVar & "'", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

D1.Read()

TextBox1.Text = D1(0).ToString

TextBox2.Text = D1(1).ToString

TextBox3.Text = D1(2).ToString

TextBox4.Text = D1(3).ToString

TextBox5.Text = D1(4).ToString

TextBox6.Text = D1(5).ToString

TextBox7.Text = D1(6).ToString

TextBox8.Text = D1(7).ToString

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = True

ButDelete.Enabled = True

Else

TextBox1.Text = ""

TextBox2.Text = ""

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

End If

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

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

Me.Close()

End Sub

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

If vbNo = MsgBox("Are you sure you want modify this record", MsgBoxStyle.YesNo, "Delete") Then Exit Sub

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim cmd1 As New SqlCommand("Delete from CustTab where custCode='" & pkVar & "'", Conn)

cmd1.ExecuteNonQuery()

If Conn.State = ConnectionState.Open Then Conn.Close()

SaveRecord()

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

End Sub

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

If vbNo = MsgBox("Are you sure you want delete this record", MsgBoxStyle.YesNo, "Delete") Then Exit Sub

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim cmd1 As New SqlCommand("Delete from CustTab where custCode='" & pkVar & "'", Conn)

cmd1.ExecuteNonQuery()

If Conn.State = ConnectionState.Open Then Conn.Close()

disRecords()

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

End Sub

Private Sub butNew\_Click1(ByVal sender As Object, ByVal e As System.EventArgs) Handles butNew.Click

ClearTxtControls(Me, 1)

TextBox1.Focus()

butNew.Enabled = False

butSave.Enabled = True

butModify.Enabled = False

ButDelete.Enabled = False

End Sub

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

disRecords()

End Sub

Private Sub TextBox1\_LostFocus(ByVal sender As Object, ByVal e As System.EventArgs) Handles TextBox1.LostFocus

TextBox1.Text = UCase(TextBox1.Text)

End Sub

End Class

Customer list form 3.5.4

Public Class CustListReport

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

Me.WindowState = FormWindowState.Maximized

End Sub

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim DS1 As New DataSet

Dim adp As New SqlDataAdapter("Select \* From CustTab order by custcode", Conn)

adp.Fill(DS1)

DG1.DataSource = DS1.Tables(0)

DG1.Columns(0).Width = 100

DG1.Columns(1).Width = 200

DG1.Columns(2).Width = 300

DG1.Columns(3).Width = 100

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

Private Sub PrintDocument1\_PrintPage(ByVal sender As System.Object, ByVal e As System.Drawing.Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage

Dim XPos, YPos As Long

YPos = 50

Dim MyFont As New Font("Arial", 18)

XPos = 10

e.Graphics.DrawString("Medical Inventory", MyFont, Brushes.Black, XPos, YPos)

YPos += 50

MyFont = New Font("Arial", 12)

e.Graphics.DrawString("No. 101, 4th floor, UB City, Bangalore - 560001", MyFont, Brushes.Black, XPos, YPos)

YPos += 100

XPos = 10

e.Graphics.DrawString("Customer List Report", MyFont, Brushes.Black, XPos, YPos)

YPos += 50

XPos = 10

MyFont = New Font("Arial", 12)

e.Graphics.DrawString("Code", MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString("Name", MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString("Address", MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 250

e.Graphics.DrawString("MobileNo", MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

YPos += 25

For Each r As DataGridViewRow In DG1.Rows

q1Var = r.Cells(1).Value & " : " & r.Cells(2).Value

XPos = 10

e.Graphics.DrawString(r.Cells(0).Value, MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString(r.Cells(1).Value, MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString(r.Cells(2).Value, MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 250

e.Graphics.DrawString(r.Cells(3).Value, MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

YPos += 25

Next

End Sub

Private Sub Button2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

PP1.ShowDialog()

End Sub

Private Sub Button3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click

Me.Close()

End Sub

End Class

Item form 3.5.5

Public Class ItemForm

Dim pkVar As String

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

Me.WindowState = FormWindowState.Maximized

disRecords()

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd1 As New SqlCommand("select catName from CatTab order by catName", Conn)

Dim D1 As SqlDataReader = Cmd1.ExecuteReader()

While D1.Read

ComboBox1.Items.Add(D1(0).ToString)

End While

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd2 As New SqlCommand("select groupName from groupTab order by groupName", Conn)

Dim D2 As SqlDataReader = Cmd2.ExecuteReader()

While D2.Read

ComboBox2.Items.Add(D2(0).ToString)

End While

End Sub

Private Sub butNew\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

ClearTxtControls(Me, 1)

TextBox1.Focus()

End Sub

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

SaveRecord()

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

End Sub

Sub SaveRecord()

If TextBox1.Text = "" Then

MsgBox("Please enter the necessary details")

Exit Sub

End If

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select ProdCode from ProductTab where ProdCode='" & UCase(TextBox1.Text) & "'", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

MsgBox("This record is allready present in the database")

If Conn.State = ConnectionState.Open Then Conn.Close()

Exit Sub

End If

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

q1Var = "insert into ProductTab("

q2Var = " values("

q1Var = q1Var & "ProdCode" & ","

q2Var = q2Var & "'" & UCase(TextBox1.Text) & "',"

q1Var = q1Var & "ProdName" & ","

q2Var = q2Var & "'" & TextBox2.Text & "',"

q1Var = q1Var & "CatName" & ","

q2Var = q2Var & "'" & ComboBox1.Text & "',"

q1Var = q1Var & "groupName" & ","

q2Var = q2Var & "'" & ComboBox2.Text & "',"

q1Var = q1Var & "UnitName" & ","

q2Var = q2Var & "'" & ComboBox5.Text & "',"

q1Var = q1Var & "PPrice" & ","

q2Var = q2Var & "" & Val(TextBox3.Text) & ","

q1Var = q1Var & "sPrice" & ","

q2Var = q2Var & "" & Val(TextBox4.Text) & ","

q1Var = q1Var & "taxPer" & ")"

q2Var = q2Var & "" & Val(TextBox5.Text) & ")"

'MsgBox(q1Var & q2Var)

Dim cmd1 As New SqlCommand(q1Var & q2Var, Conn)

cmd1.ExecuteNonQuery()

If Conn.State = ConnectionState.Open Then Conn.Close()

disRecords()

End Sub

Sub disRecords()

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim DS1 As New DataSet

Dim adp As New SqlDataAdapter("Select ProdCode,ProdName From ProductTab order by ProdCode", Conn)

adp.Fill(DS1)

DG1.DataSource = DS1.Tables(0)

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

Private Sub DG1\_CellContentClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DG1.CellContentClick

pkVar = DG1.CurrentRow.Cells(0).Value

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select \* from ProductTab where ProdCode='" & pkVar & "'", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

D1.Read()

TextBox1.Text = D1(0).ToString

TextBox2.Text = D1(1).ToString

ComboBox1.Text = D1(2).ToString

TextBox3.Text = D1(5).ToString

TextBox4.Text = D1(6).ToString

TextBox5.Text = D1(7).ToString

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = True

ButDelete.Enabled = True

Else

TextBox1.Text = ""

TextBox2.Text = ""

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

End If

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

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

Me.Close()

End Sub

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

If vbNo = MsgBox("Are you sure you want modify this record", MsgBoxStyle.YesNo, "Delete") Then Exit Sub

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim cmd1 As New SqlCommand("Delete from ProductTab where ProdCode='" & pkVar & "'", Conn)

cmd1.ExecuteNonQuery()

If Conn.State = ConnectionState.Open Then Conn.Close()

SaveRecord()

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

End Sub

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

If vbNo = MsgBox("Are you sure you want delete this record", MsgBoxStyle.YesNo, "Delete") Then Exit Sub

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim cmd1 As New SqlCommand("Delete from ProductTab where ProdCode='" & pkVar & "'", Conn)

cmd1.ExecuteNonQuery()

If Conn.State = ConnectionState.Open Then Conn.Close()

disRecords()

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

End Sub

Private Sub butNew\_Click1(ByVal sender As Object, ByVal e As System.EventArgs) Handles butNew.Click

ClearTxtControls(Me, 1)

TextBox1.Focus()

butNew.Enabled = False

butSave.Enabled = True

butModify.Enabled = False

ButDelete.Enabled = False

End Sub

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

disRecords()

End Sub

Private Sub TextBox1\_LostFocus(ByVal sender As Object, ByVal e As System.EventArgs) Handles TextBox1.LostFocus

TextBox1.Text = UCase(TextBox1.Text)

End Sub

End Class

Mcat form 3.5.6

Public Class mCatForm

Dim pkVar As String

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

Me.WindowState = FormWindowState.Maximized

disRecords()

End Sub

Private Sub butNew\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

ClearTxtControls(Me, 1)

TextBox1.Focus()

End Sub

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

SaveRecord()

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

End Sub

Sub SaveRecord()

If TextBox1.Text = "" Then

MsgBox("Please enter the necessary details")

Exit Sub

End If

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select CatName from CatTab where CatName='" & UCase(TextBox1.Text) & "'", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

MsgBox("This record is allready present in the database")

If Conn.State = ConnectionState.Open Then Conn.Close()

Exit Sub

End If

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

q1Var = "insert into CatTab("

q2Var = " values("

q1Var = q1Var & "CatName" & ","

q2Var = q2Var & "'" & UCase(TextBox1.Text) & "',"

q1Var = q1Var & "Details" & ")"

q2Var = q2Var & "'" & TextBox2.Text & "')"

' MsgBox(q1Var & q2Var)

Dim cmd1 As New SqlCommand(q1Var & q2Var, Conn)

cmd1.ExecuteNonQuery()

MsgBox("category saved successfully")

If Conn.State = ConnectionState.Open Then Conn.Close()

disRecords()

End Sub

Sub disRecords()

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim DS1 As New DataSet

Dim adp As New SqlDataAdapter("Select CatName,Details From CatTab order by CatName", Conn)

adp.Fill(DS1)

DG1.DataSource = DS1.Tables(0)

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

Private Sub DG1\_CellContentClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DG1.CellContentClick

pkVar = DG1.CurrentRow.Cells(0).Value

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select \* from CatTab where CatName='" & pkVar & "'", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

D1.Read()

TextBox1.Text = D1(0).ToString

TextBox2.Text = D1(1).ToString

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = True

ButDelete.Enabled = True

Else

TextBox1.Text = ""

TextBox2.Text = ""

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

End If

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

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

Me.Close()

End Sub

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

If vbNo = MsgBox("Are you sure you want modify this record", MsgBoxStyle.YesNo, "Delete") Then Exit Sub

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim cmd1 As New SqlCommand("Delete from CatTab where CatName='" & pkVar & "'", Conn)

cmd1.ExecuteNonQuery()

If Conn.State = ConnectionState.Open Then Conn.Close()

SaveRecord()

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

End Sub

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

If vbNo = MsgBox("Are you sure you want delete this record", MsgBoxStyle.YesNo, "Delete") Then Exit Sub

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim cmd1 As New SqlCommand("Delete from CatTab where CatName='" & pkVar & "'", Conn)

cmd1.ExecuteNonQuery()

If Conn.State = ConnectionState.Open Then Conn.Close()

disRecords()

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

End Sub

Private Sub butNew\_Click1(ByVal sender As Object, ByVal e As System.EventArgs) Handles butNew.Click

ClearTxtControls(Me, 1)

TextBox1.Focus()

butNew.Enabled = False

butSave.Enabled = True

butModify.Enabled = False

ButDelete.Enabled = False

End Sub

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

disRecords()

End Sub

Private Sub TextBox1\_LostFocus(ByVal sender As Object, ByVal e As System.EventArgs) Handles TextBox1.LostFocus

TextBox1.Text = UCase(TextBox1.Text)

End Sub

End Class

Mgroup form3.5.7

Public Class mGroupForm

Dim pkVar As String

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

Me.WindowState = FormWindowState.Maximized

disRecords()

End Sub

Private Sub butNew\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

ClearTxtControls(Me, 1)

TextBox1.Focus()

End Sub

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

SaveRecord()

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

End Sub

Sub SaveRecord()

If TextBox1.Text = "" Then

MsgBox("Please enter the necessary details")

Exit Sub

End If

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select GroupName from GroupTab where GroupName='" & UCase(TextBox1.Text) & "'", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

MsgBox("This record is allready present in the database")

If Conn.State = ConnectionState.Open Then Conn.Close()

Exit Sub

End If

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

q1Var = "insert into GroupTab("

q2Var = " values("

q1Var = q1Var & "GroupName" & ","

q2Var = q2Var & "'" & UCase(TextBox1.Text) & "',"

q1Var = q1Var & "Details" & ")"

q2Var = q2Var & "'" & TextBox2.Text & "')"

' MsgBox(q1Var & q2Var)

Dim cmd1 As New SqlCommand(q1Var & q2Var, Conn)

cmd1.ExecuteNonQuery()

MsgBox("group saved successfully")

If Conn.State = ConnectionState.Open Then Conn.Close()

disRecords()

End Sub

Sub disRecords()

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim DS1 As New DataSet

Dim adp As New SqlDataAdapter("Select GroupName,Details From GroupTab order by GroupName", Conn)

adp.Fill(DS1)

DG1.DataSource = DS1.Tables(0)

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

Private Sub DG1\_CellContentClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DG1.CellContentClick

pkVar = DG1.CurrentRow.Cells(0).Value

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select \* from GroupTab where GroupName='" & pkVar & "'", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

D1.Read()

TextBox1.Text = D1(0).ToString

TextBox2.Text = D1(1).ToString

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = True

ButDelete.Enabled = True

Else

TextBox1.Text = ""

TextBox2.Text = ""

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

End If

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

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

Me.Close()

End Sub

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

If vbNo = MsgBox("Are you sure you want modify this record", MsgBoxStyle.YesNo, "Delete") Then Exit Sub

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim cmd1 As New SqlCommand("Delete from GroupTab where GroupName='" & pkVar & "'", Conn)

cmd1.ExecuteNonQuery()

If Conn.State = ConnectionState.Open Then Conn.Close()

SaveRecord()

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

End Sub

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

If vbNo = MsgBox("Are you sure you want delete this record", MsgBoxStyle.YesNo, "Delete") Then Exit Sub

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim cmd1 As New SqlCommand("Delete from GroupTab where GroupName='" & pkVar & "'", Conn)

cmd1.ExecuteNonQuery()

If Conn.State = ConnectionState.Open Then Conn.Close()

disRecords()

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

End Sub

Private Sub butNew\_Click1(ByVal sender As Object, ByVal e As System.EventArgs) Handles butNew.Click

ClearTxtControls(Me, 1)

TextBox1.Focus()

butNew.Enabled = False

butSave.Enabled = True

butModify.Enabled = False

ButDelete.Enabled = False

End Sub

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

disRecords()

End Sub

Private Sub TextBox1\_LostFocus(ByVal sender As Object, ByVal e As System.EventArgs) Handles TextBox1.LostFocus

TextBox1.Text = UCase(TextBox1.Text)

End Sub

End Class

Purchase form3.5.8

Public Class PurchaseForm

Dim pkVar, BillNoVar As Long

Dim rowNo As Long

Dim totAmt, taxVar As Decimal

Dim DS2 As New DataSet

Dim Tab1 As DataTable

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

Me.WindowState = FormWindowState.Maximized

disRecords()

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select suppCode from SuppTab order by suppcode", Conn)

Dim D2 As SqlDataReader = Cmd0.ExecuteReader()

While D2.Read

ComboBox1.Items.Add(D2(0).ToString)

End While

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd1 As New SqlCommand("select ProdCode from ProductTab order by ProdCode", Conn)

Dim D1 As SqlDataReader = Cmd1.ExecuteReader()

While D1.Read

ComboBox2.Items.Add(D1(0).ToString)

End While

DG2Init()

End Sub

Sub DG2Init()

rowNo = 0

totAmt = 0

DS2 = New DataSet

Tab1 = New DataTable

Tab1 = DS2.Tables.Add("t1Tab")

Tab1.Columns.Add("ItemCode", GetType(String))

Tab1.Columns.Add("iRate", GetType(Long))

Tab1.Columns.Add("iQty", GetType(Long))

Tab1.Columns.Add("TaxAmt", GetType(Long))

Tab1.Columns.Add("TotAmt", GetType(Long))

DG2.DataSource = DS2.Tables(0)

End Sub

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

SaveRecord()

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

butPrint.Enabled = True

End Sub

Sub SaveRecord()

Dim cmd3 As New SqlCommand

'If TextBox1.Text = "" Then

' MsgBox("Please enter the necessary details")

' Exit Sub

'End If

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select max(Billno) from SaMain", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

D1.Read()

BillNoVar = Val(D1(0).ToString) + 1

If Conn.State = ConnectionState.Open Then Conn.Close()

End If

'TextBox1.Text = BillNoVar

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

q1Var = "insert into SaMain("

q2Var = " values("

q1Var = q1Var & "billNo" & ","

q2Var = q2Var & "" & BillNoVar & ","

q1Var = q1Var & "BDate" & ","

q2Var = q2Var & "'" & bDate.Value & "',"

q1Var = q1Var & "PartyCode" & ","

q2Var = q2Var & "'" & ComboBox1.Text & "',"

q1Var = q1Var & "RefNo" & ","

q2Var = q2Var & "'" & TextBox1.Text & "',"

q1Var = q1Var & "Btype" & ","

q2Var = q2Var & "2,"

q1Var = q1Var & "tAmt" & ","

q2Var = q2Var & "" & Val(TxtT1.Text) & ","

q1Var = q1Var & "Add1" & ","

q2Var = q2Var & "" & Val(TxtT2.Text) & ","

q1Var = q1Var & "gtAmt" & ","

q2Var = q2Var & "" & Val(TxtT3.Text) & ","

q1Var = q1Var & "Rem1" & ")"

q2Var = q2Var & "'" & txtRem.Text & "')"

'MsgBox(q1Var & q2Var)

Dim cmd1 As New SqlCommand(q1Var & q2Var, Conn)

cmd1.ExecuteNonQuery()

For Each r As DataGridViewRow In DG2.Rows

If Conn.State = ConnectionState.Open Then Conn.Close()

If Not r.Cells(0).Value = "" Then

q1Var = "insert into SaDet("

q2Var = " values("

q1Var = q1Var & "billNo" & ","

q2Var = q2Var & "" & BillNoVar & ","

q1Var = q1Var & "slNo" & ","

q2Var = q2Var & "" & I & ","

q1Var = q1Var & "ItemCode" & ","

q2Var = q2Var & "'" & r.Cells(0).Value & "',"

q1Var = q1Var & "iRate" & ","

q2Var = q2Var & "" & Val(r.Cells(1).Value) & ","

q1Var = q1Var & "iQty" & ","

q2Var = q2Var & "" & Val(r.Cells(2).Value) & ","

q1Var = q1Var & "TaxAmt" & ","

q2Var = q2Var & "" & Val(r.Cells(3).Value) & ","

q1Var = q1Var & "totAmt" & ","

q2Var = q2Var & "" & Val(r.Cells(4).Value) & ","

q1Var = q1Var & "Rem1" & ")"

q2Var = q2Var & "'')"

Conn.Open()

cmd3.Connection = Conn

cmd3.CommandText = (q1Var & q2Var)

cmd3.ExecuteNonQuery()

End If

Next

disRecords()

End Sub

Sub disRecords()

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim DS1 As New DataSet

Dim adp As New SqlDataAdapter("Select Billno,PartyCode From SaMain where btype =2 order by Billno", Conn)

adp.Fill(DS1)

DG1.DataSource = DS1.Tables(0)

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

Private Sub DG1\_CellContentClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DG1.CellContentClick

DG2Init()

ClearTxtControls(Me, 1)

pkVar = DG1.CurrentRow.Cells(0).Value

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select \* from SaMain where billNo=" & pkVar & "", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

D1.Read()

'TextBox1.Text = D1("BillNo").ToString

TextBox1.Text = D1("RefNo").ToString

bDate.Value = D1("bDate")

ComboBox1.Text = D1("PartyCode").ToString

TxtT1.Text = D1("tAmt").ToString

TxtT2.Text = D1("add1").ToString

TxtT3.Text = D1("gtAmt").ToString

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = True

ButDelete.Enabled = True

butPrint.Enabled = True

Else

TextBox1.Text = ""

ComboBox1.Text = ""

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

End If

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd4 As New SqlCommand("select \* from SaDet where billNo=" & pkVar & "", Conn)

Dim D2 As SqlDataReader = Cmd4.ExecuteReader()

rowNo = 1

totAmt = 0

While D2.Read

Dim dRow As DataRow = Tab1.NewRow

dRow(0) = D2(2)

dRow(1) = CLng(D2(3))

dRow(2) = CLng(D2(4))

dRow(3) = CLng(D2(5))

dRow(4) = CLng(D2(6))

Tab1.Rows.Add(dRow)

'Grid1.set\_TextMatrix(rowNo, 0, D2(2).ToString)

'Grid1.set\_TextMatrix(rowNo, 1, D2(3).ToString)

'Grid1.set\_TextMatrix(rowNo, 2, D2(4).ToString)

'Grid1.set\_TextMatrix(rowNo, 3, D2(5).ToString)

'Grid1.set\_TextMatrix(rowNo, 4, D2(6).ToString)

totAmt = totAmt + D2(6)

rowNo = rowNo + 1

End While

DG2.DataSource = DS2.Tables(0)

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

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

Me.Close()

End Sub

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

If vbNo = MsgBox("Are you sure you want modify this record", MsgBoxStyle.YesNo, "Delete") Then Exit Sub

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim cmd1 As New SqlCommand("Delete from SaMain where billNo=" & pkVar & "", Conn)

cmd1.ExecuteNonQuery()

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim cmd2 As New SqlCommand("Delete from Sadet where billNo=" & pkVar & "", Conn)

cmd2.ExecuteNonQuery()

If Conn.State = ConnectionState.Open Then Conn.Close()

SaveRecord()

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

End Sub

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

If vbNo = MsgBox("Are you sure you want delete this record", MsgBoxStyle.YesNo, "Delete") Then Exit Sub

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim cmd1 As New SqlCommand("Delete from SaMain where billNo=" & pkVar & "", Conn)

cmd1.ExecuteNonQuery()

If Conn.State = ConnectionState.Open Then Conn.Close()

disRecords()

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

End Sub

Private Sub butNew\_Click1(ByVal sender As Object, ByVal e As System.EventArgs) Handles butNew.Click

DG2Init()

ClearTxtControls(Me, 1)

TextBox1.Focus()

butNew.Enabled = False

butSave.Enabled = True

butModify.Enabled = False

ButDelete.Enabled = False

butPrint.Enabled = False

End Sub

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

disRecords()

End Sub

Private Sub TextBox1\_LostFocus(ByVal sender As Object, ByVal e As System.EventArgs) Handles TextBox1.LostFocus

TextBox1.Text = UCase(TextBox1.Text)

End Sub

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

Dim dRow As DataRow = Tab1.NewRow

dRow(0) = ComboBox2.Text

dRow(1) = CLng(TxtRate.Text)

dRow(2) = CLng(TxtQty.Text)

dRow(3) = CLng(TxtTax.Text)

dRow(4) = CLng(TxtTot.Text)

Tab1.Rows.Add(dRow)

DG2.DataSource = DS2.Tables(0)

rowNo = rowNo + 1

totAmt = totAmt + Val(TxtTot.Text)

TxtT1.Text = totAmt

TxtT3.Text = totAmt + Val(TxtT2.Text)

ComboBox2.Text = ""

TxtRate.Text = ""

TxtQty.Text = ""

TxtTax.Text = ""

TxtTot.Text = ""

ComboBox2.Focus()

End Sub

Private Sub ComboBox2\_Leave(ByVal sender As Object, ByVal e As System.EventArgs) Handles ComboBox2.Leave

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select pPrice,TaxPer from ProductTab where ProdCode='" & ComboBox2.Text & "'", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

D1.Read()

TxtRate.Text = D1(0).ToString

taxVar = D1(1).ToString

End If

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

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

TxtTax.Text = (Val(TxtQty.Text) \* Val(TxtRate.Text) \* Val(taxVar) \* 0.01).ToString

TxtTot.Text = (Val(TxtQty.Text) \* Val(TxtRate.Text)) + Val(TxtTax.Text)

End Sub

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

TxtTax.Text = (Val(TxtQty.Text) \* Val(TxtRate.Text) \* Val(taxVar) \* 0.01).ToString

TxtTot.Text = (Val(TxtQty.Text) \* Val(TxtRate.Text)) + Val(TxtTax.Text)

End Sub

Private Sub TxtT2\_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles TxtT2.TextChanged

TxtT3.Text = Val(TxtT1.Text) + Val(TxtT2.Text)

End Sub

Private Sub PrintDocument1\_PrintPage(ByVal sender As System.Object, ByVal e As System.Drawing.Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage

Dim XPos, YPos As Long

YPos += 50

XPos = 10

Dim MyFont As New Font("Arial", 20)

e.Graphics.DrawString("N1 Mobile Showroom", MyFont, Brushes.Black, XPos, YPos)

YPos += 50

MyFont = New Font("Arial", 14)

XPos = 10

e.Graphics.DrawString("Purchase Bill", MyFont, Brushes.Black, XPos, YPos)

XPos = 200

e.Graphics.DrawString("Date : ", MyFont, Brushes.Black, XPos, YPos)

XPos = 300

e.Graphics.DrawString(bDate.Value, MyFont, Brushes.Black, XPos, YPos)

YPos += 50

XPos = 200

e.Graphics.DrawString("BillNo : ", MyFont, Brushes.Black, XPos, YPos)

XPos = 300

e.Graphics.DrawString(TextBox1.Text, MyFont, Brushes.Black, XPos, YPos)

YPos += 50

XPos = 10

MyFont = New Font("Arial", 12)

e.Graphics.DrawString("From", MyFont, Brushes.Black, XPos, YPos)

YPos += 25

XPos = 10

e.Graphics.DrawString(ComboBox1.Text, MyFont, Brushes.Black, XPos, YPos)

YPos += 20

XPos = 10

e.Graphics.DrawString("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_", MyFont, Brushes.Black, XPos, YPos)

YPos += 30

XPos = 10

e.Graphics.DrawString("Product Name", MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 200

e.Graphics.DrawString("Qty", MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString("Rate", MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString("Tax Amt", MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString("Total Amt", MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

YPos += 10

XPos = 10

e.Graphics.DrawString("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_", MyFont, Brushes.Black, XPos, YPos)

YPos += 25

For Each r As DataGridViewRow In DG2.Rows

XPos = 10

e.Graphics.DrawString(r.Cells(0).Value, MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 200

e.Graphics.DrawString(r.Cells(1).Value, MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString(r.Cells(2).Value, MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString(r.Cells(3).Value, MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString(r.Cells(4).Value, MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

YPos += 25

Next

XPos = 10

e.Graphics.DrawString("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_", MyFont, Brushes.Black, XPos, YPos)

End Sub

Private Sub PP1\_Disposed(ByVal sender As Object, ByVal e As System.EventArgs) Handles PP1.Disposed

PP1.Close()

PrintDocument1.Dispose()

End Sub

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

PP1.ShowDialog()

End Sub

Private Sub DG2\_CellContentClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DG2.CellContentClick

End Sub

Private Sub TextBox1\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox1.TextChanged

End Sub

End Class

Purchase rep form

Public Class PurchaseRep

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

Me.WindowState = FormWindowState.Maximized

End Sub

Private Sub PrintDocument1\_PrintPage(ByVal sender As System.Object, ByVal e As System.Drawing.Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage

Dim XPos, YPos As Long

YPos = 50

Dim MyFont As New Font("Arial", 18)

XPos = 10

e.Graphics.DrawString("Medical Inventory", MyFont, Brushes.Black, XPos, YPos)

YPos += 50

MyFont = New Font("Arial", 12)

e.Graphics.DrawString("No 233, II Floor, UB City, Bangalore", MyFont, Brushes.Black, XPos, YPos)

YPos += 100

XPos = 10

e.Graphics.DrawString("Purchase Detail Report", MyFont, Brushes.Black, XPos, YPos)

YPos += 50

XPos = 10

MyFont = New Font("Arial", 12)

e.Graphics.DrawString("RefNo", MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 50

e.Graphics.DrawString("Bill Date", MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString("Party Code", MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString("Total Amount", MyFont, Brushes.Black, XPos, YPos)

YPos += 25

For Each r As DataGridViewRow In DG1.Rows

XPos = 10

e.Graphics.DrawString(r.Cells(0).Value, MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 50

e.Graphics.DrawString(Format(r.Cells(1).Value, "dd-MMM-yyyy"), MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString(r.Cells(2).Value, MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString(r.Cells(3).Value, MyFont, Brushes.Black, XPos, YPos)

YPos += 25

Next

End Sub

Private Sub Button2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

PP1.ShowDialog()

End Sub

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim DS1 As New DataSet

Dim adp As New SqlDataAdapter("Select RefNo,bDate,PartyCode,tamt From saMain where btype=2 order by refNo,bDate", Conn)

adp.Fill(DS1)

DG1.DataSource = DS1.Tables(0)

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

Private Sub Button3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click

Me.Close()

End Sub

End Class

Counter sales form3.5.9

public Class CounterSalesForm

Dim pkVar, BillNoVar As Long

Dim rowNo As Long

Dim totAmt, taxVar As Decimal

Dim DS2 As New DataSet

Dim Tab1 As DataTable

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

Me.WindowState = FormWindowState.Maximized

disRecords()

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select custCode from custTab order by custcode", Conn)

Dim D2 As SqlDataReader = Cmd0.ExecuteReader()

While D2.Read

ComboBox1.Items.Add(D2(0).ToString)

End While

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd1 As New SqlCommand("select ProdCode from ProductTab order by ProdCode", Conn)

Dim D1 As SqlDataReader = Cmd1.ExecuteReader()

While D1.Read

ComboBox2.Items.Add(D1(0).ToString)

End While

DG2Init()

End Sub

Sub DG2Init()

rowNo = 0

totAmt = 0

DS2 = New DataSet

Tab1 = New DataTable

Tab1 = DS2.Tables.Add("t1Tab")

Tab1.Columns.Add("ItemCode", GetType(String))

Tab1.Columns.Add("iRate", GetType(Long))

Tab1.Columns.Add("iQty", GetType(Long))

Tab1.Columns.Add("TaxAmt", GetType(Long))

Tab1.Columns.Add("TotAmt", GetType(Long))

DG2.DataSource = DS2.Tables(0)

End Sub

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

SaveRecord()

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

butPrint.Enabled = True

End Sub

Sub SaveRecord()

Dim cmd3 As New SqlCommand

'If TextBox1.Text = "" Then

' MsgBox("Please enter the necessary details")

' Exit Sub

'End If

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select max(Billno) from SaMain", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

D1.Read()

BillNoVar = Val(D1(0).ToString) + 1

If Conn.State = ConnectionState.Open Then Conn.Close()

End If

TextBox1.Text = BillNoVar

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

q1Var = "insert into SaMain("

q2Var = " values("

q1Var = q1Var & "billNo" & ","

q2Var = q2Var & "" & BillNoVar & ","

q1Var = q1Var & "BDate" & ","

q2Var = q2Var & "'" & bDate.Value & "',"

q1Var = q1Var & "PartyCode" & ","

q2Var = q2Var & "'" & ComboBox1.Text & "',"

q1Var = q1Var & "RefNo" & ","

q2Var = q2Var & "'" & TextBox1.Text & "',"

q1Var = q1Var & "Btype" & ","

q2Var = q2Var & "4,"

q1Var = q1Var & "tAmt" & ","

q2Var = q2Var & "" & Val(TxtT1.Text) & ","

q1Var = q1Var & "Add1" & ","

q2Var = q2Var & "" & Val(TxtT2.Text) & ","

q1Var = q1Var & "gtAmt" & ","

q2Var = q2Var & "" & Val(TxtT3.Text) & ","

q1Var = q1Var & "Rem1" & ")"

q2Var = q2Var & "'" & txtRem.Text & "')"

'MsgBox(q1Var & q2Var)

Dim cmd1 As New SqlCommand(q1Var & q2Var, Conn)

cmd1.ExecuteNonQuery()

For Each r As DataGridViewRow In DG2.Rows

If Conn.State = ConnectionState.Open Then Conn.Close()

If Not r.Cells(0).Value = "" Then

q1Var = "insert into SaDet("

q2Var = " values("

q1Var = q1Var & "billNo" & ","

q2Var = q2Var & "" & BillNoVar & ","

q1Var = q1Var & "slNo" & ","

q2Var = q2Var & "" & I & ","

q1Var = q1Var & "ItemCode" & ","

q2Var = q2Var & "'" & r.Cells(0).Value & "',"

q1Var = q1Var & "iRate" & ","

q2Var = q2Var & "" & Val(r.Cells(1).Value) & ","

q1Var = q1Var & "iQty" & ","

q2Var = q2Var & "" & Val(r.Cells(2).Value) & ","

q1Var = q1Var & "TaxAmt" & ","

q2Var = q2Var & "" & Val(r.Cells(3).Value) & ","

q1Var = q1Var & "totAmt" & ","

q2Var = q2Var & "" & Val(r.Cells(4).Value) & ","

q1Var = q1Var & "Rem1" & ")"

q2Var = q2Var & "'')"

Conn.Open()

cmd3.Connection = Conn

cmd3.CommandText = (q1Var & q2Var)

cmd3.ExecuteNonQuery()

End If

Next

disRecords()

End Sub

Sub disRecords()

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim DS1 As New DataSet

Dim adp As New SqlDataAdapter("Select Billno,PartyCode From SaMain where btype = 4 order by Billno", Conn)

adp.Fill(DS1)

DG1.DataSource = DS1.Tables(0)

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

Private Sub DG1\_CellContentClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DG1.CellContentClick

DG2Init()

ClearTxtControls(Me, 1)

pkVar = DG1.CurrentRow.Cells(0).Value

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select \* from SaMain where billNo=" & pkVar & "", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

D1.Read()

TextBox1.Text = D1("BillNo").ToString

'TextBox1.Text = D1("RefNo").ToString

bDate.Value = D1("bDate")

ComboBox1.Text = D1("PartyCode").ToString

TxtT1.Text = D1("tAmt").ToString

TxtT2.Text = D1("add1").ToString

TxtT3.Text = D1("gtAmt").ToString

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = True

ButDelete.Enabled = True

butPrint.Enabled = True

Else

TextBox1.Text = ""

ComboBox1.Text = ""

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

End If

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd4 As New SqlCommand("select \* from SaDet where billNo=" & pkVar & "", Conn)

Dim D2 As SqlDataReader = Cmd4.ExecuteReader()

rowNo = 1

totAmt = 0

While D2.Read

Dim dRow As DataRow = Tab1.NewRow

dRow(0) = D2(2)

dRow(1) = CLng(D2(3))

dRow(2) = CLng(D2(4))

dRow(3) = CLng(D2(5))

dRow(4) = CLng(D2(6))

Tab1.Rows.Add(dRow)

'Grid1.set\_TextMatrix(rowNo, 0, D2(2).ToString)

'Grid1.set\_TextMatrix(rowNo, 1, D2(3).ToString)

'Grid1.set\_TextMatrix(rowNo, 2, D2(4).ToString)

'Grid1.set\_TextMatrix(rowNo, 3, D2(5).ToString)

'Grid1.set\_TextMatrix(rowNo, 4, D2(6).ToString)

totAmt = totAmt + D2(6)

rowNo = rowNo + 1

End While

DG2.DataSource = DS2.Tables(0)

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

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

Me.Close()

End Sub

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

If vbNo = MsgBox("Are you sure you want modify this record", MsgBoxStyle.YesNo, "Delete") Then Exit Sub

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim cmd1 As New SqlCommand("Delete from SaMain where billNo=" & pkVar & "", Conn)

cmd1.ExecuteNonQuery()

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim cmd2 As New SqlCommand("Delete from Sadet where billNo=" & pkVar & "", Conn)

cmd2.ExecuteNonQuery()

If Conn.State = ConnectionState.Open Then Conn.Close()

SaveRecord()

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

End Sub

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

If vbNo = MsgBox("Are you sure you want delete this record", MsgBoxStyle.YesNo, "Delete") Then Exit Sub

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim cmd1 As New SqlCommand("Delete from SaMain where billNo=" & pkVar & "", Conn)

cmd1.ExecuteNonQuery()

If Conn.State = ConnectionState.Open Then Conn.Close()

disRecords()

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

End Sub

Private Sub butNew\_Click1(ByVal sender As Object, ByVal e As System.EventArgs) Handles butNew.Click

DG2Init()

ClearTxtControls(Me, 1)

ComboBox2.Focus()

butNew.Enabled = False

butSave.Enabled = True

butModify.Enabled = False

ButDelete.Enabled = False

butPrint.Enabled = False

End Sub

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

disRecords()

End Sub

Private Sub TextBox1\_LostFocus(ByVal sender As Object, ByVal e As System.EventArgs) Handles TextBox1.LostFocus

TextBox1.Text = UCase(TextBox1.Text)

End Sub

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

Dim dRow As DataRow = Tab1.NewRow

dRow(0) = ComboBox2.Text

dRow(1) = CLng(TxtRate.Text)

dRow(2) = CLng(TxtQty.Text)

dRow(3) = CLng(TxtTax.Text)

dRow(4) = CLng(TxtTot.Text)

Tab1.Rows.Add(dRow)

DG2.DataSource = DS2.Tables(0)

rowNo = rowNo + 1

totAmt = totAmt + Val(TxtTot.Text)

TxtT1.Text = totAmt

TxtT3.Text = totAmt + Val(TxtT2.Text)

ComboBox2.Text = ""

TxtRate.Text = ""

TxtQty.Text = ""

TxtTax.Text = ""

TxtTot.Text = ""

ComboBox2.Focus()

End Sub

Private Sub ComboBox2\_Leave(ByVal sender As Object, ByVal e As System.EventArgs) Handles ComboBox2.Leave

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select sPrice,TaxPer from ProductTab where ProdCode='" & ComboBox2.Text & "'", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

D1.Read()

TxtRate.Text = D1(0).ToString

taxVar = D1(1).ToString

End If

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

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

TxtTax.Text = (Val(TxtQty.Text) \* Val(TxtRate.Text) \* Val(taxVar) \* 0.01).ToString

TxtTot.Text = (Val(TxtQty.Text) \* Val(TxtRate.Text)) + Val(TxtTax.Text)

End Sub

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

TxtTax.Text = (Val(TxtQty.Text) \* Val(TxtRate.Text) \* Val(taxVar) \* 0.01).ToString

TxtTot.Text = (Val(TxtQty.Text) \* Val(TxtRate.Text)) + Val(TxtTax.Text)

End Sub

Private Sub TxtT2\_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles TxtT2.TextChanged

TxtT3.Text = Val(TxtT1.Text) + Val(TxtT2.Text)

End Sub

Private Sub PrintDocument1\_PrintPage(ByVal sender As System.Object, ByVal e As System.Drawing.Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage

Dim XPos, YPos As Long

YPos += 50

XPos = 10

Dim MyFont As New Font("Arial", 20)

e.Graphics.DrawString("N1 Mobile Showroom", MyFont, Brushes.Black, XPos, YPos)

YPos += 50

MyFont = New Font("Arial", 14)

XPos = 10

e.Graphics.DrawString("Cash Bill", MyFont, Brushes.Black, XPos, YPos)

XPos = 200

e.Graphics.DrawString("Date : ", MyFont, Brushes.Black, XPos, YPos)

XPos = 300

e.Graphics.DrawString(bDate.Value, MyFont, Brushes.Black, XPos, YPos)

YPos += 50

XPos = 200

e.Graphics.DrawString("BillNo : ", MyFont, Brushes.Black, XPos, YPos)

XPos = 300

e.Graphics.DrawString(TextBox1.Text, MyFont, Brushes.Black, XPos, YPos)

YPos += 50

XPos = 10

MyFont = New Font("Arial", 12)

e.Graphics.DrawString("To", MyFont, Brushes.Black, XPos, YPos)

YPos += 25

XPos = 10

e.Graphics.DrawString(ComboBox1.Text, MyFont, Brushes.Black, XPos, YPos)

YPos += 20

XPos = 10

e.Graphics.DrawString("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_", MyFont, Brushes.Black, XPos, YPos)

YPos += 30

XPos = 10

e.Graphics.DrawString("Product Name", MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 200

e.Graphics.DrawString("Qty", MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString("Rate", MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString("Tax Amt", MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString("Total Amt", MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

YPos += 10

XPos = 10

e.Graphics.DrawString("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_", MyFont, Brushes.Black, XPos, YPos)

YPos += 25

For Each r As DataGridViewRow In DG2.Rows

XPos = 10

e.Graphics.DrawString(r.Cells(0).Value, MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 200

e.Graphics.DrawString(r.Cells(1).Value, MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString(r.Cells(2).Value, MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString(r.Cells(3).Value, MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString(r.Cells(4).Value, MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

YPos += 25

Next

XPos = 10

e.Graphics.DrawString("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_", MyFont, Brushes.Black, XPos, YPos)

YPos += 50

XPos = 450

e.Graphics.DrawString("Total Amt : ", MyFont, Brushes.Black, XPos, YPos)

XPos = 650

e.Graphics.DrawString(TxtT3.Text, MyFont, Brushes.Black, XPos, YPos)

End Sub

Private Sub PP1\_Disposed(ByVal sender As Object, ByVal e As System.EventArgs) Handles PP1.Disposed

PP1.Close()

PrintDocument1.Dispose()

End Sub

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

PP1.ShowDialog()

End Sub

End Class

Sales rep3.5.10

Public Class SalesRep

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

Me.WindowState = FormWindowState.Maximized

End Sub

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim DS1 As New DataSet

Dim adp As New SqlDataAdapter("Select RefNo,bDate,PartyCode,tamt From saMain where btype=4 order by refNo,bDate", Conn)

adp.Fill(DS1)

DG1.DataSource = DS1.Tables(0)

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

Private Sub PrintDocument1\_PrintPage(ByVal sender As System.Object, ByVal e As System.Drawing.Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage

Dim XPos, YPos As Long

YPos = 50

Dim MyFont As New Font("Arial", 18)

XPos = 10

e.Graphics.DrawString("Medical Inventory", MyFont, Brushes.Black, XPos, YPos)

YPos += 50

MyFont = New Font("Arial", 12)

e.Graphics.DrawString("No 233, II Floor, UB City, Bangalore", MyFont, Brushes.Black, XPos, YPos)

YPos += 100

XPos = 10

e.Graphics.DrawString("Purchase Detail Report", MyFont, Brushes.Black, XPos, YPos)

YPos += 50

XPos = 10

MyFont = New Font("Arial", 12)

e.Graphics.DrawString("RefNo", MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 50

e.Graphics.DrawString("Bill Date", MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString("Party Code", MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString("Total Amount", MyFont, Brushes.Black, XPos, YPos)

YPos += 25

For Each r As DataGridViewRow In DG1.Rows

XPos = 10

e.Graphics.DrawString(r.Cells(0).Value, MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 50

e.Graphics.DrawString(Format(r.Cells(1).Value, "dd-MMM-yyyy"), MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString(r.Cells(2).Value, MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString(r.Cells(3).Value, MyFont, Brushes.Black, XPos, YPos)

YPos += 25

Next

End Sub

Private Sub Button2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

PP1.ShowDialog()

End Sub

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

Me.Close()

End Sub

End Class

Stock rep3.5.11

Public Class StockRep

Dim rowNo, dNo As Long

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

rowNo = 0

Dim IVar, OVar, bVar As Long

Dim DS2 As New DataSet

Dim Tab1 As DataTable

DS2 = New DataSet

Tab1 = New DataTable

Tab1 = DS2.Tables.Add("t1Tab")

Tab1.Columns.Add("ItemCode", GetType(String))

Tab1.Columns.Add("In Qty", GetType(Long))

Tab1.Columns.Add("Out Qty", GetType(Long))

Tab1.Columns.Add("Balance", GetType(Long))

Dim dRow As DataRow = Tab1.NewRow

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select sum(b.iqty) from saMain A,saDet B where A.billNo=b.BillNo and A.btype=2 and B.itemcode='" & ComboBox1.Text & "'", Conn)

Dim D2 As SqlDataReader = Cmd0.ExecuteReader()

D2.Read()

IVar = IIf(IsDBNull(D2(0)), 0, D2(0))

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd1 As New SqlCommand("select sum(B.iqty) from saMain A,saDet B where A.billNo=b.BillNo and (A.btype=3 or A.btype=4) and B.itemcode='" & ComboBox1.Text & "'", Conn)

Dim D3 As SqlDataReader = Cmd1.ExecuteReader()

D3.Read()

OVar = IIf(IsDBNull(D3(0)), 0, D3(0))

bVar = IVar - OVar

dRow(0) = ComboBox1.Text

dRow(1) = IVar

dRow(2) = OVar

dRow(3) = bVar

Tab1.Rows.Add(dRow)

DG1.DataSource = DS2.Tables(0)

End Sub

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

End Sub

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

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select ProdCode from ProductTab order by ProdCode", Conn)

Dim D2 As SqlDataReader = Cmd0.ExecuteReader()

While D2.Read

ComboBox1.Items.Add(D2(0).ToString)

End While

End Sub

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

Me.Close()

End Sub

End Class

Supplier form3.5.12

Public Class SupplierForm

Dim pkVar As String

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

Me.WindowState = FormWindowState.Maximized

disRecords()

End Sub

Private Sub butNew\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

ClearTxtControls(Me, 1)

TextBox1.Focus()

End Sub

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

SaveRecord()

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

End Sub

Sub SaveRecord()

If TextBox1.Text = "" Then

MsgBox("Please enter the necessary details")

Exit Sub

End If

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select Suppcode from SuppTab where Suppcode='" & UCase(TextBox1.Text) & "'", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

MsgBox("This record is allready present in the database")

If Conn.State = ConnectionState.Open Then Conn.Close()

Exit Sub

End If

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

q1Var = "insert into SuppTab("

q2Var = " values("

q1Var = q1Var & "Suppcode" & ","

q2Var = q2Var & "'" & UCase(TextBox1.Text) & "',"

q1Var = q1Var & "SuppName" & ","

q2Var = q2Var & "'" & TextBox2.Text & "',"

q1Var = q1Var & "Add1" & ","

q2Var = q2Var & "'" & TextBox3.Text & "',"

q1Var = q1Var & "Add2" & ","

q2Var = q2Var & "'" & TextBox4.Text & "',"

q1Var = q1Var & "Add3" & ","

q2Var = q2Var & "'" & TextBox5.Text & "',"

q1Var = q1Var & "PinCode" & ","

q2Var = q2Var & "'" & TextBox6.Text & "',"

q1Var = q1Var & "PhoneNo" & ","

q2Var = q2Var & "'" & TextBox7.Text & "',"

q1Var = q1Var & "eMail" & ")"

q2Var = q2Var & "'" & TextBox8.Text & "')"

'MsgBox(q1Var & q2Var)

Dim cmd1 As New SqlCommand(q1Var & q2Var, Conn)

cmd1.ExecuteNonQuery()

MsgBox("details entered successfully")

If Conn.State = ConnectionState.Open Then Conn.Close()

disRecords()

End Sub

Sub disRecords()

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim DS1 As New DataSet

Dim adp As New SqlDataAdapter("Select Suppcode,SuppName From SuppTab order by Suppcode", Conn)

adp.Fill(DS1)

DG1.DataSource = DS1.Tables(0)

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

Private Sub DG1\_CellContentClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DG1.CellContentClick

pkVar = DG1.CurrentRow.Cells(0).Value

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select \* from SuppTab where Suppcode='" & pkVar & "'", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

D1.Read()

TextBox1.Text = D1(0).ToString

TextBox2.Text = D1(1).ToString

TextBox3.Text = D1(2).ToString

TextBox4.Text = D1(3).ToString

TextBox5.Text = D1(4).ToString

TextBox6.Text = D1(5).ToString

TextBox7.Text = D1(6).ToString

TextBox8.Text = D1(7).ToString

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = True

ButDelete.Enabled = True

Else

TextBox1.Text = ""

TextBox2.Text = ""

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

End If

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

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

Me.Close()

End Sub

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

If vbNo = MsgBox("Are you sure you want modify this record", MsgBoxStyle.YesNo, "Delete") Then Exit Sub

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim cmd1 As New SqlCommand("Delete from SuppTab where Suppcode='" & pkVar & "'", Conn)

cmd1.ExecuteNonQuery()

If Conn.State = ConnectionState.Open Then Conn.Close()

SaveRecord()

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

End Sub

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

If vbNo = MsgBox("Are you sure you want delete this record", MsgBoxStyle.YesNo, "Delete") Then Exit Sub

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim cmd1 As New SqlCommand("Delete from SuppTab where Suppcode='" & pkVar & "'", Conn)

cmd1.ExecuteNonQuery()

If Conn.State = ConnectionState.Open Then Conn.Close()

disRecords()

butNew.Enabled = True

butSave.Enabled = False

butModify.Enabled = False

ButDelete.Enabled = False

End Sub

Private Sub butNew\_Click1(ByVal sender As Object, ByVal e As System.EventArgs) Handles butNew.Click

ClearTxtControls(Me, 1)

TextBox1.Focus()

butNew.Enabled = False

butSave.Enabled = True

butModify.Enabled = False

ButDelete.Enabled = False

End Sub

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

disRecords()

End Sub

Private Sub TextBox1\_LostFocus(ByVal sender As Object, ByVal e As System.EventArgs) Handles TextBox1.LostFocus

TextBox1.Text = UCase(TextBox1.Text)

End Sub

End Class

Module3.5.12

Module Module1

Public Conn As New SqlConnection("Data Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\PharmacyMngData.mdf;Integrated Security=True;User Instance=True")

Public SqlStr, q1Var, q2Var As String

Public colVar, rowVar, I, J, inVar, outVar, L1 As Long

Public vTypeVar As Byte

Public con1 As Boolean

Public sinChar As String

Enum CtrlType

TextBox = 1

ComboBox = 2

End Enum

Public Sub ClearTxtControls(ByRef frm As Object, ByRef ControlType As CtrlType, Optional ByRef Tagstr As Object = Nothing)

Dim Contrl As Object

For Each Contrl In frm.Controls

If Not (IsNothing(Tagstr)) Then

If Trim(UCase(Contrl.Tag)) = Trim(UCase(Tagstr)) Then

Contrl.Text = ""

Exit For

End If

Else

Select Case ControlType

Case CtrlType.ComboBox

If TypeOf Contrl Is System.Windows.Forms.ComboBox Then Contrl.Text = ""

Case CtrlType.TextBox

If TypeOf Contrl Is System.Windows.Forms.TextBox Then Contrl.Text = ""

End Select

End If

Next Contrl

Contrl = Nothing

End Sub

Public Function CheckChar(ByRef CharString As Object) As String

L1 = Len(CharString)

CharString = UCase(Left(CharString, 1)) & Right(CharString, L1 - 1)

For I = 1 To L1

If con1 = True Then CharString = Mid(CharString, 1, I - 1) & UCase(Mid(CharString, I, 1)) & Mid(CharString, I + 1, L1)

sinChar = Mid(CharString, I, 1)

If sinChar = " " Then

con1 = True

Else

con1 = False

End If

Next I

CheckChar = CharString

End Function

Public Function CheckNum(ByRef KeyVar As String) As String

If Asc(KeyVar) = 8 Then

CheckNum = KeyVar : Exit Function

End If

If Asc(KeyVar) < 46 Or Asc(KeyVar) > 57 Then

CheckNum = Nothing

MsgBox("Please Enter Numbers Only")

Else

CheckNum = KeyVar

End If

If Asc(KeyVar) = 47 Then CheckNum = Nothing

End Function

Public Function forCur(ByRef CurAmt As Object) As Object

forCur = Format(CurAmt, "##,##,##0.00")

End Function

Public Function forDate(ByRef DateVar As Object) As Object

forDate = Format(DateVar, "dd/MMM/yyyy")

End Function

End Module

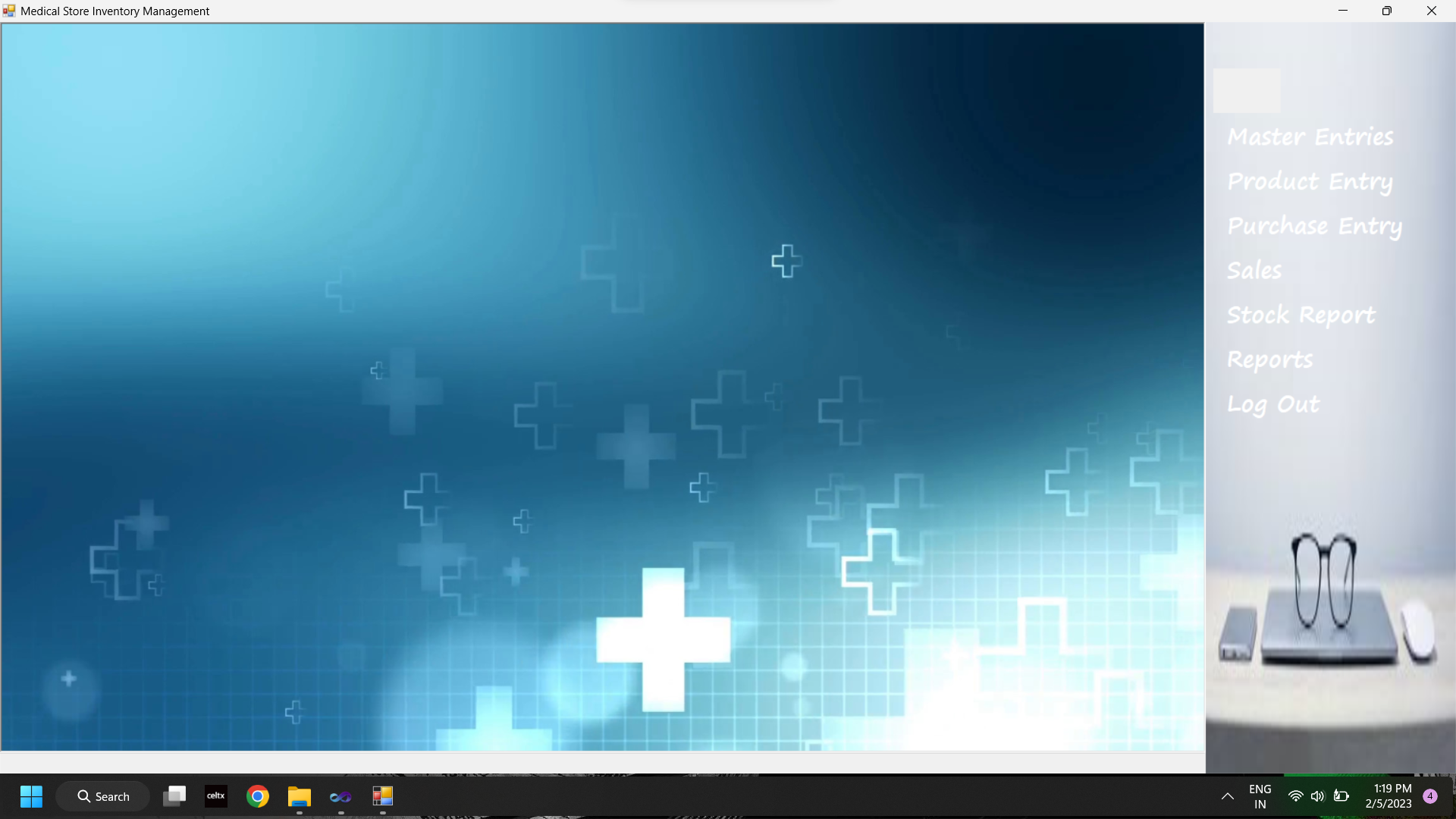
Design of pharmacy management system4.0

Splash screen.vb

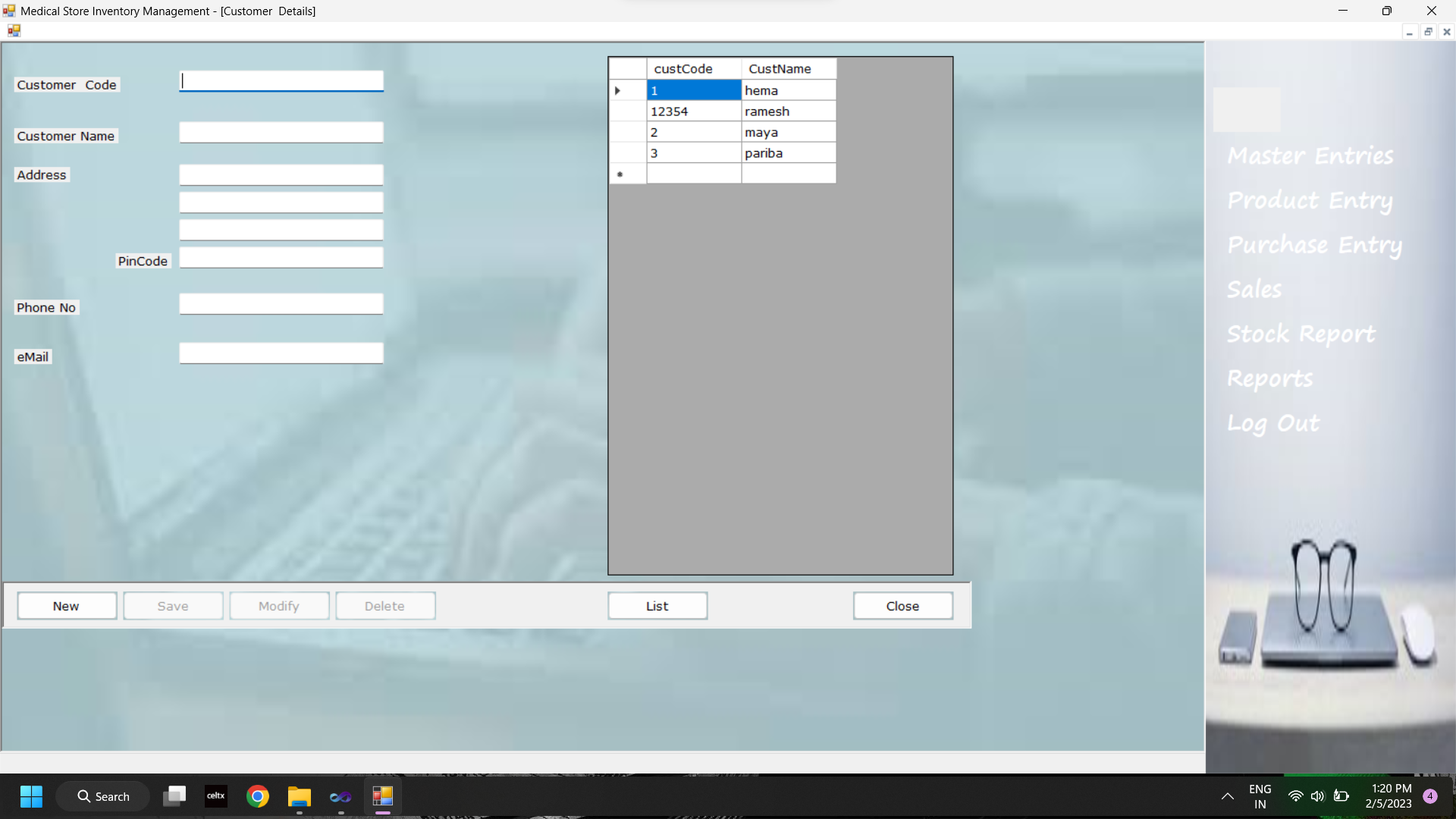


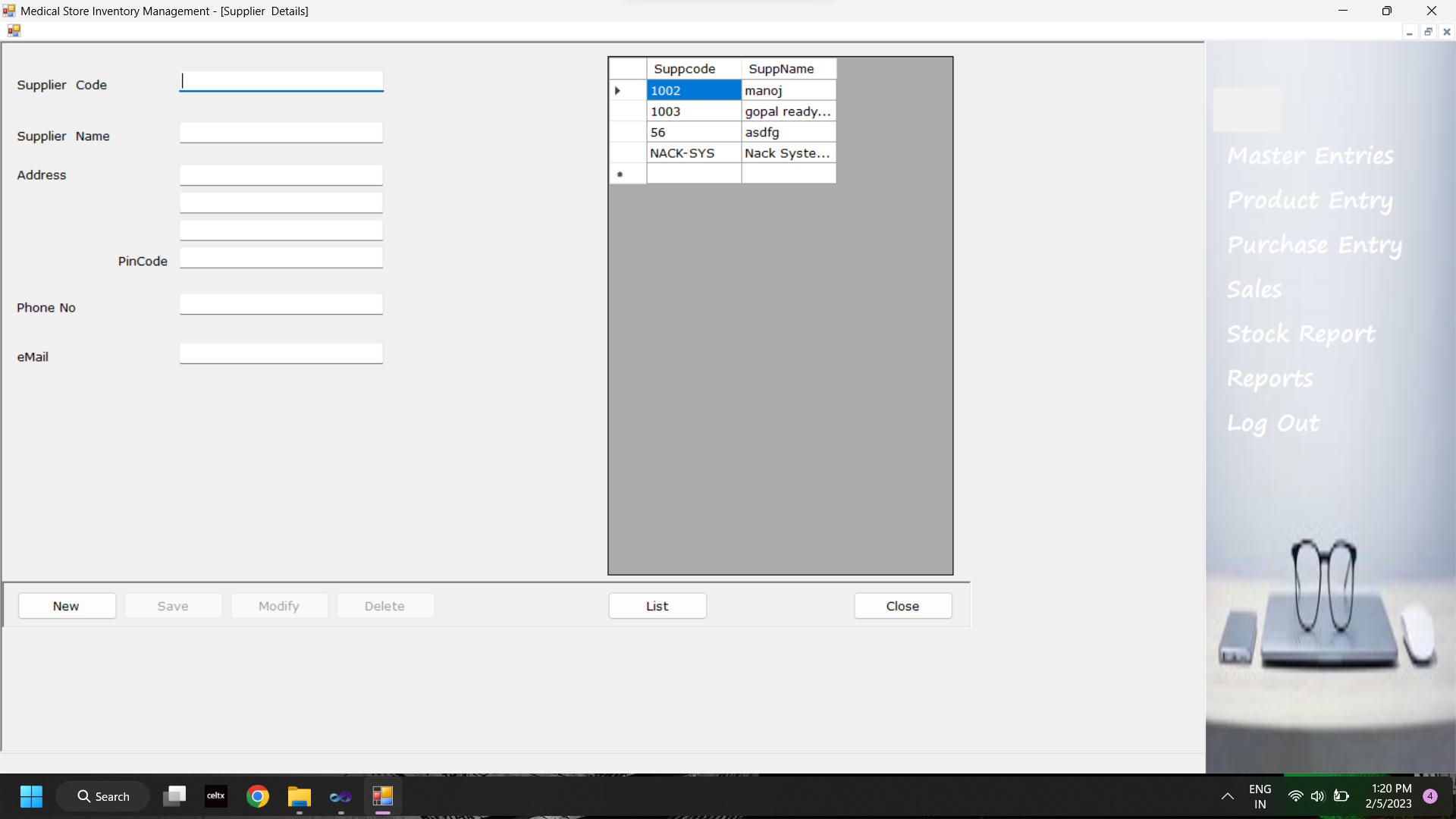
Login.vb

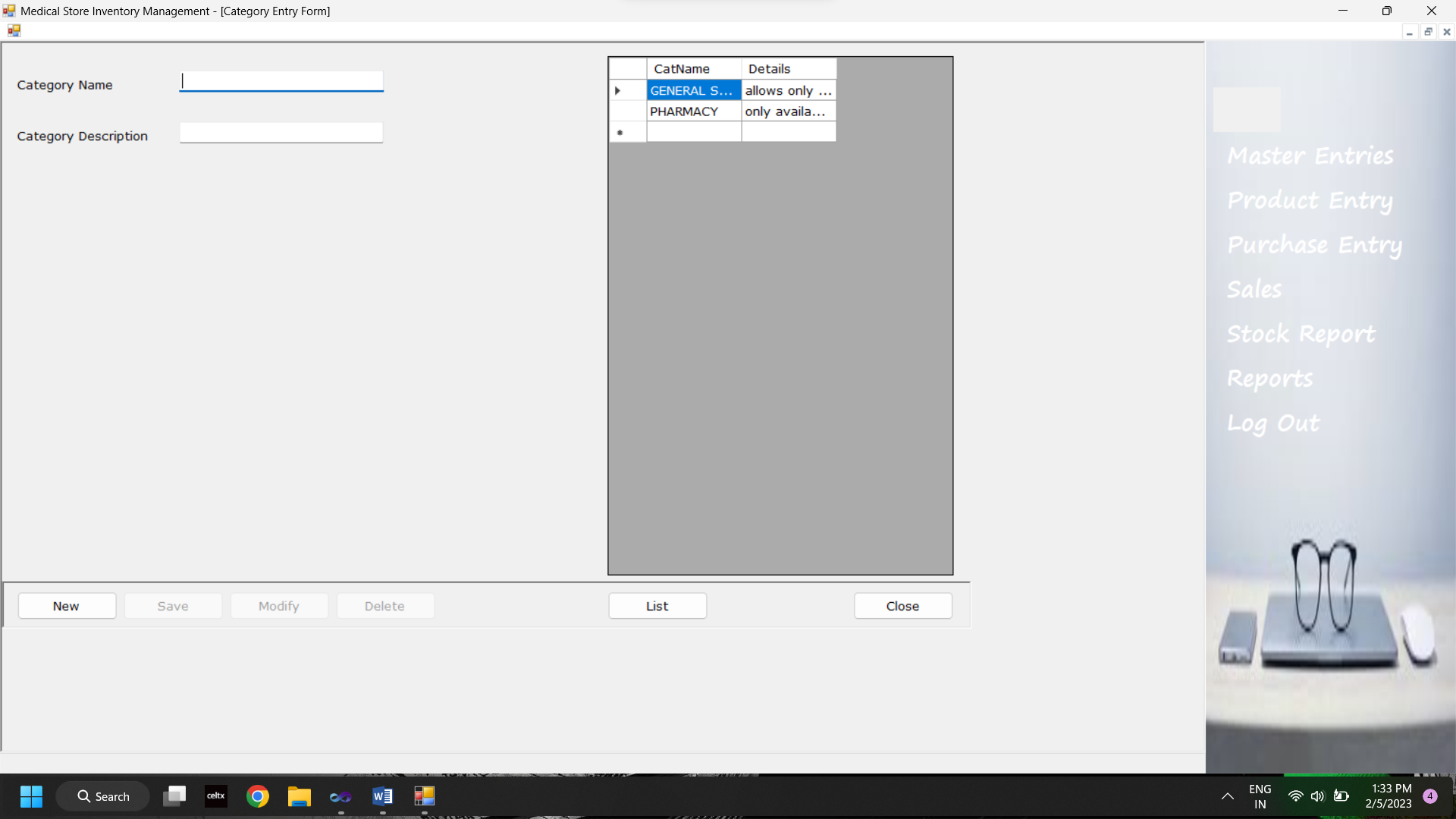
Midiparent.1.vb

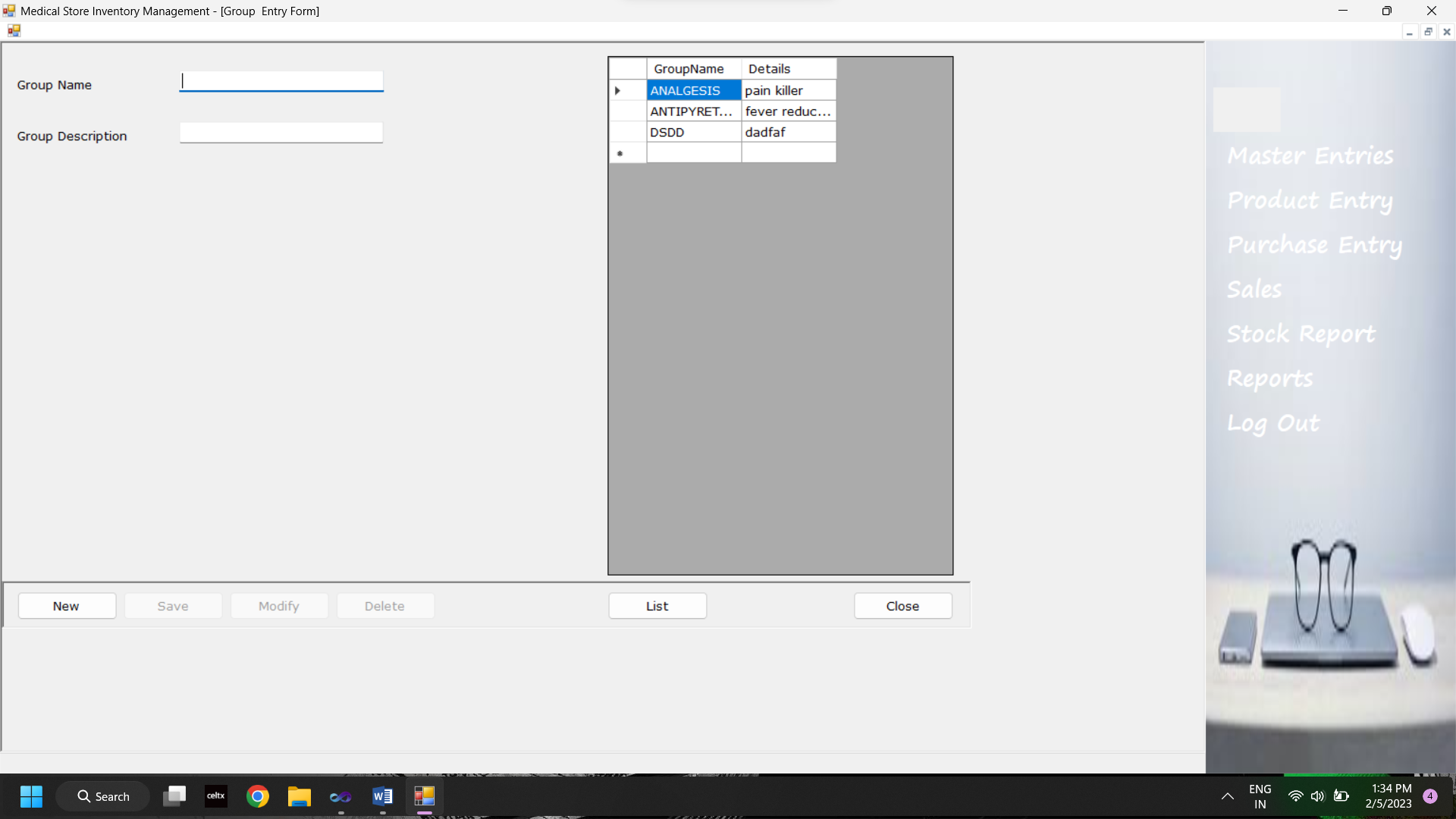


Master Enrteries

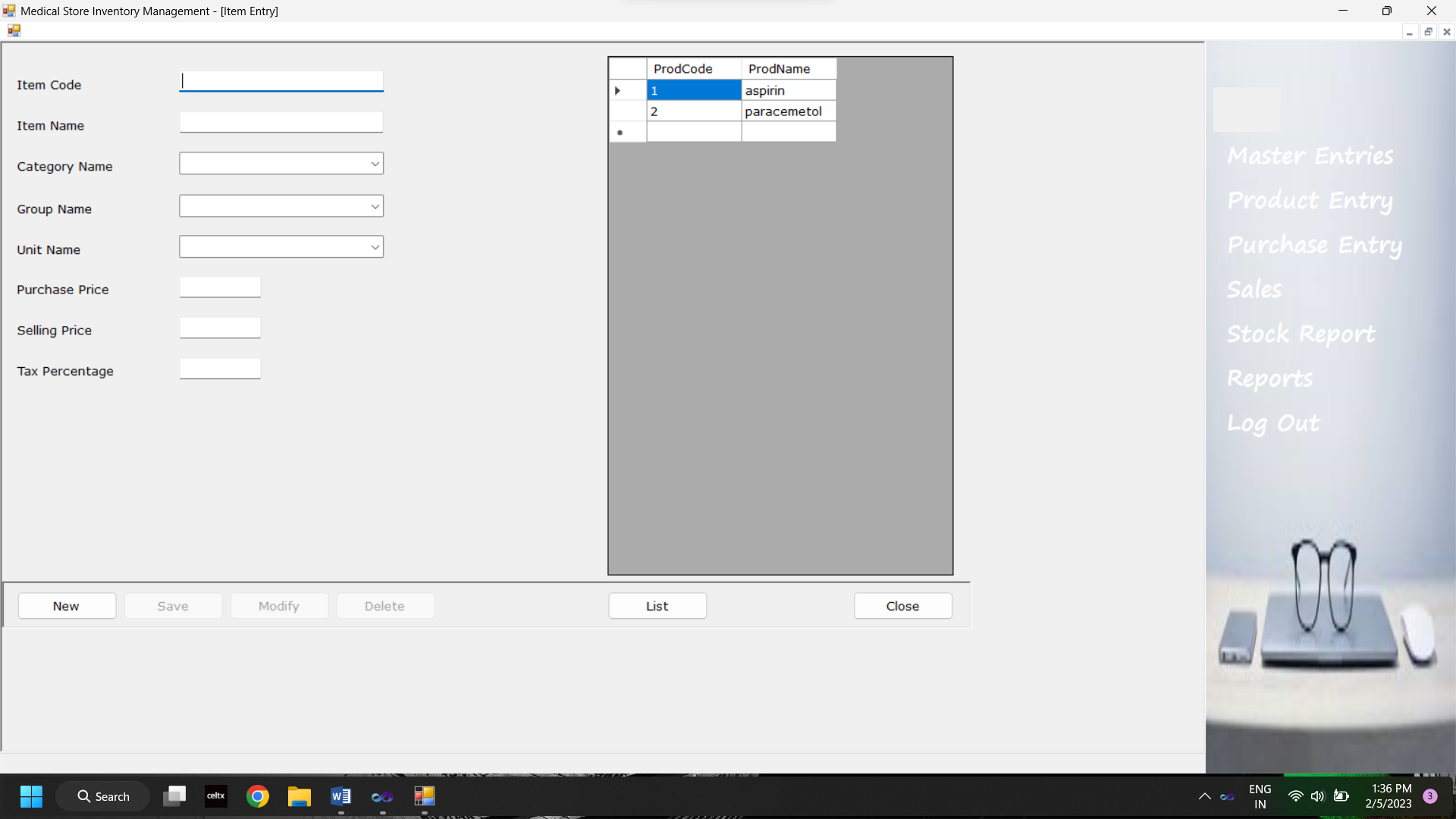




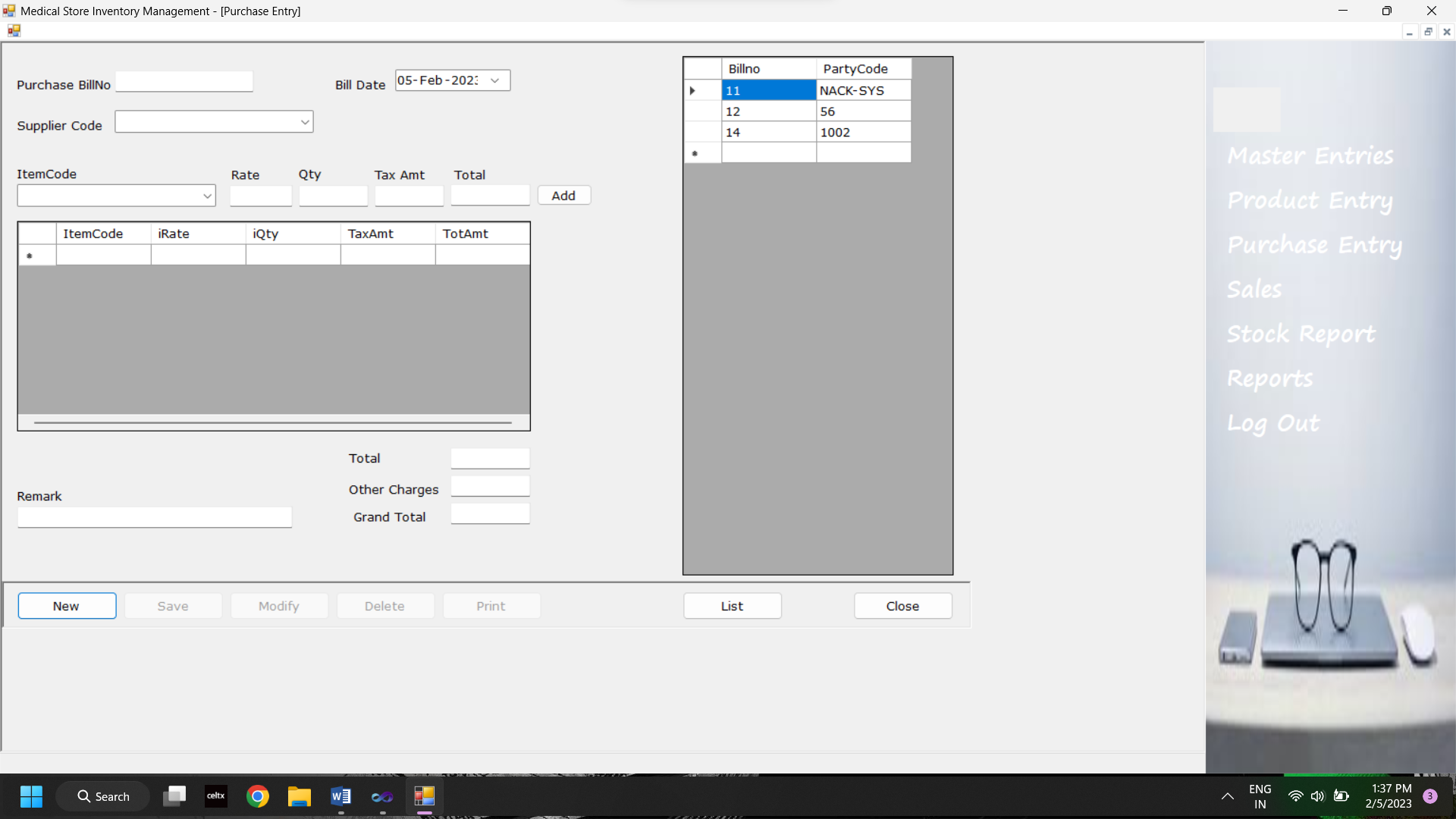




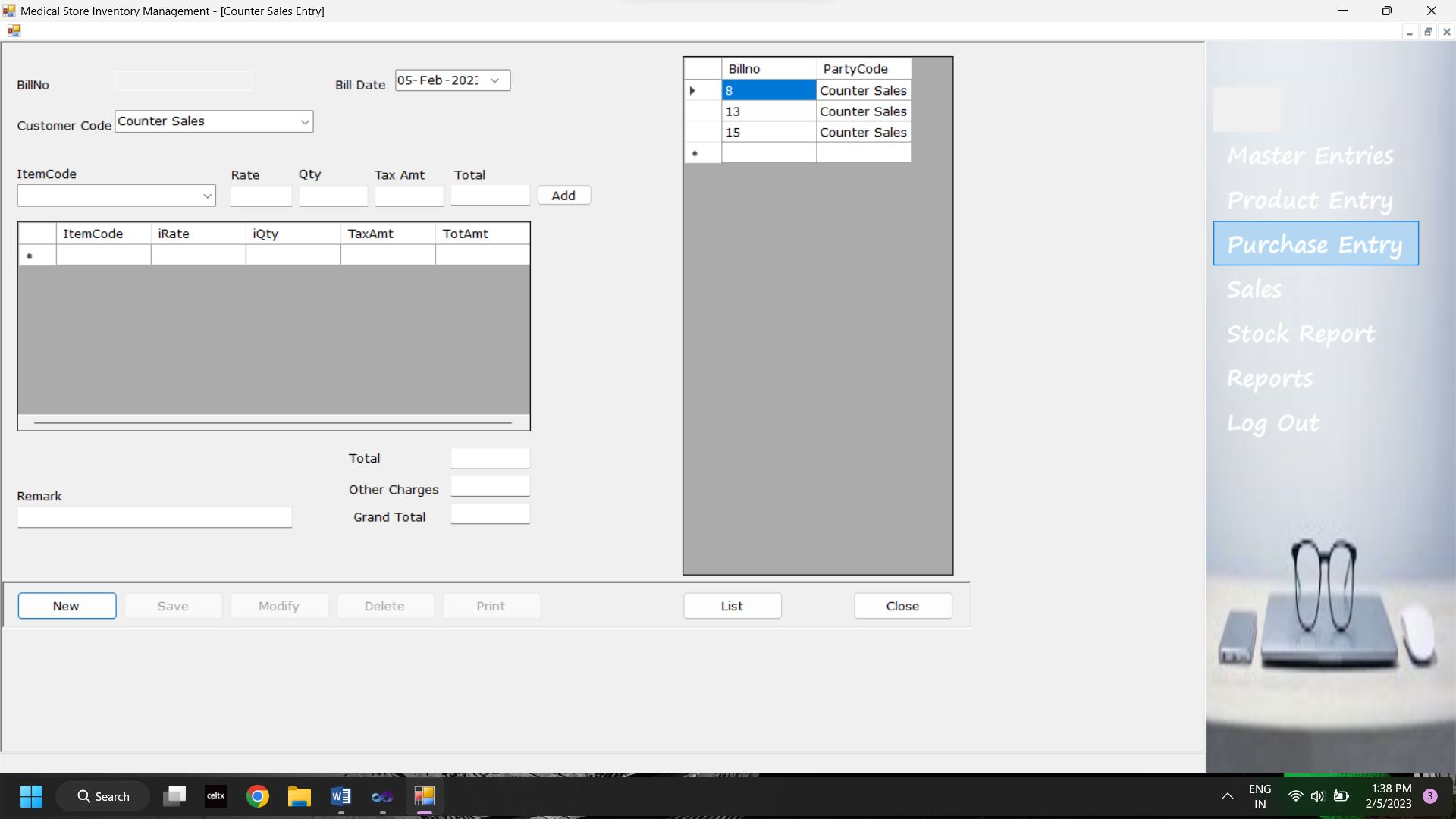
Product entry



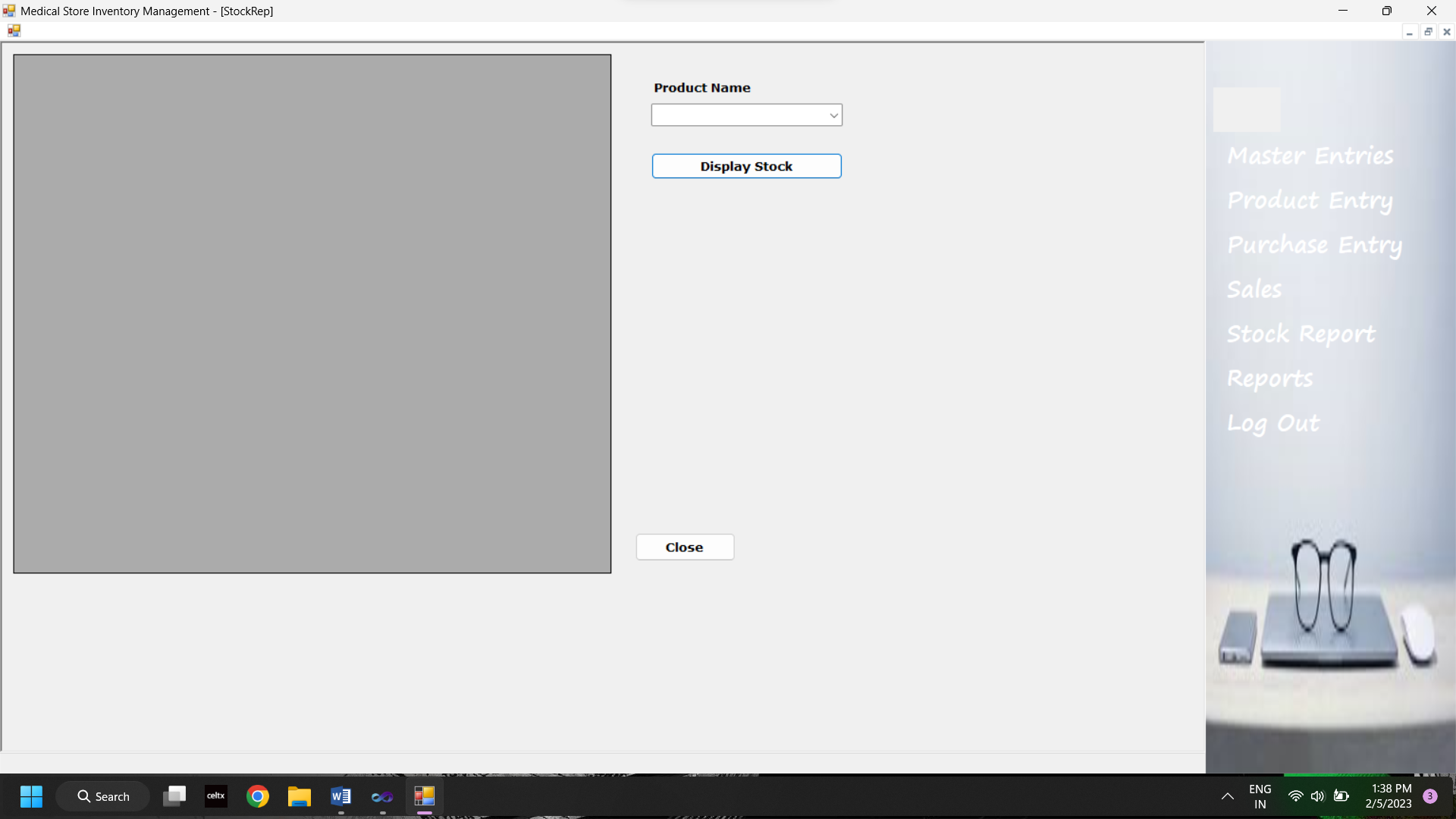
Purchase entry



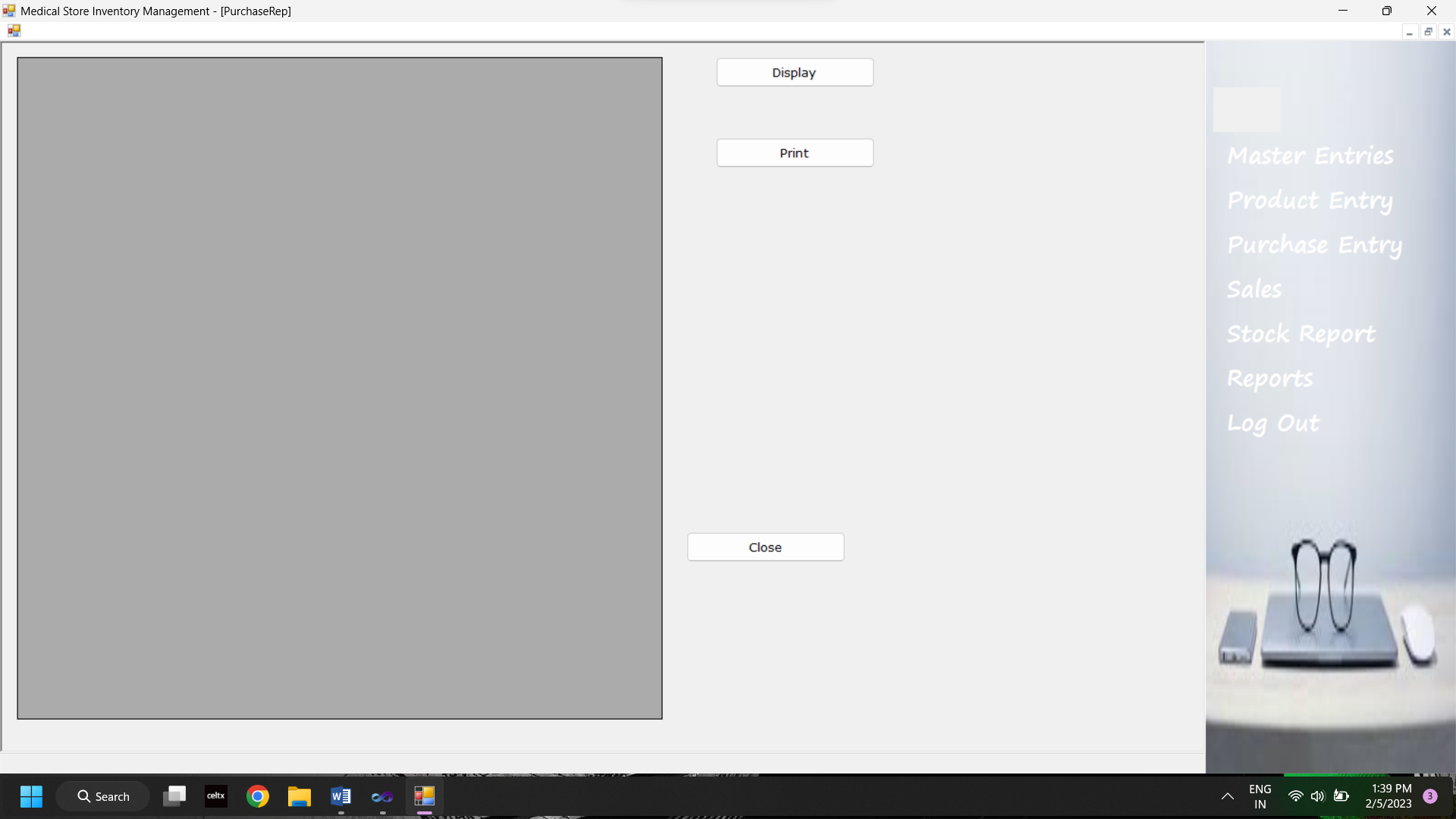
Sales

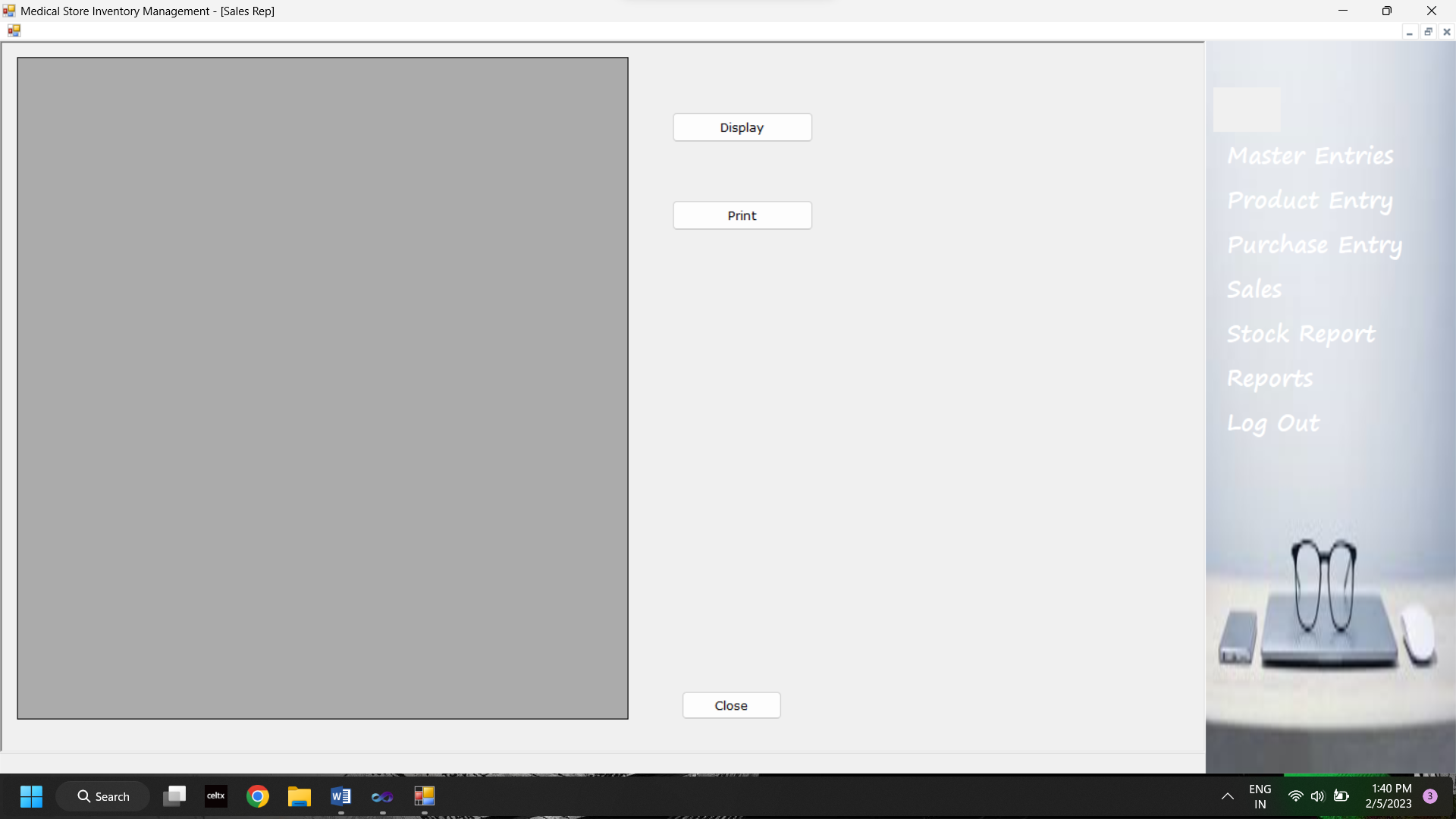


Stock report

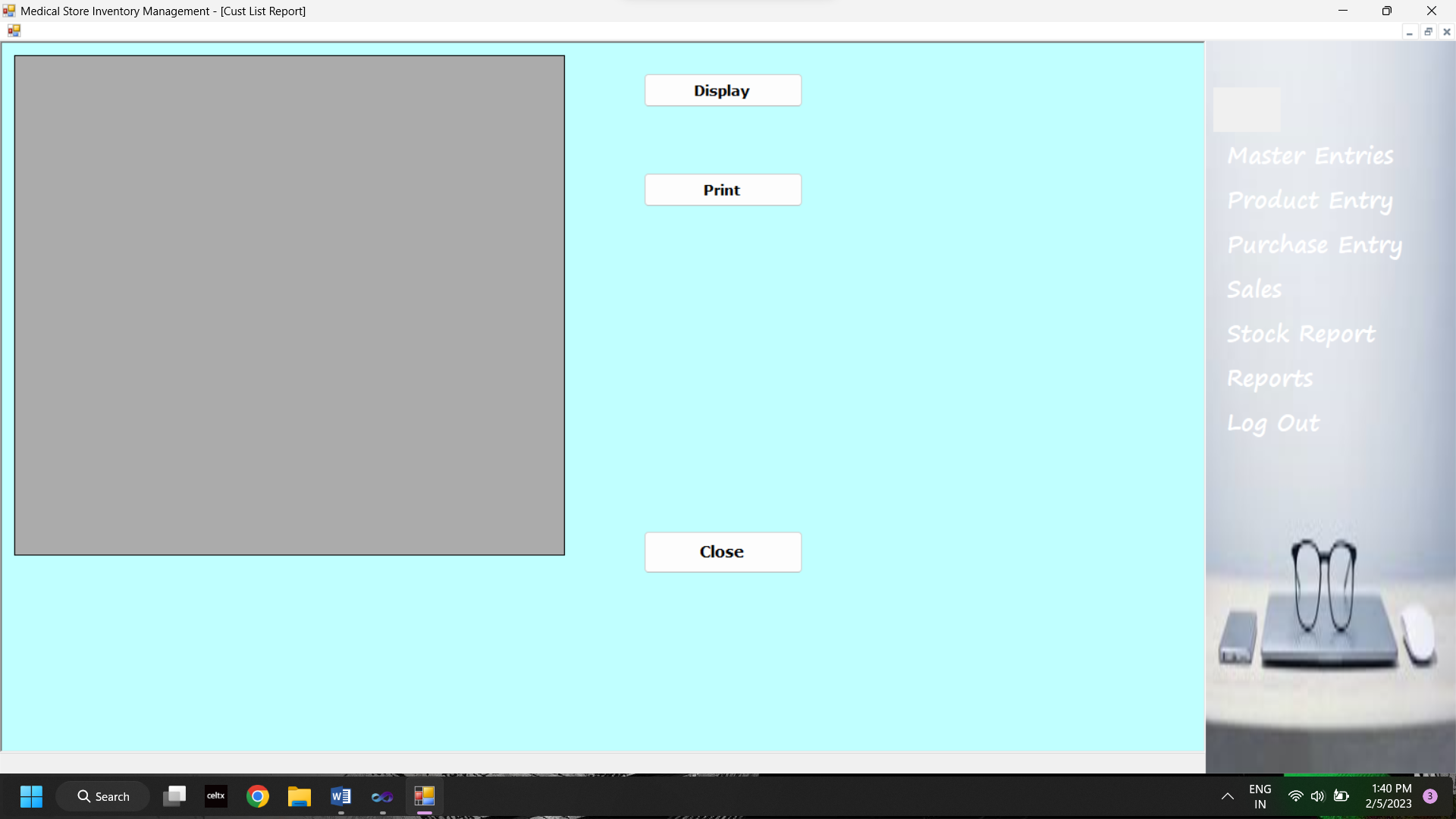


Purchase repore



Sales report

Customers list



CHAPTER-S 5.0

Testing

5.1 Feasibility Study:

Aller doing the project Pharmacy Management System, study and analyzing all the existing or required functionalities of the system, the next task is to do the feasibility study for the project All projects are feasible-given unlimited resources and infinite time.

Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements

5.1.1 Economical Feasibility

This is a very important aspect to be considered while developing a project. We decided the technology based on minimum possible cost factor.

a. All hardware and software cost has to be borne by the organization.

b. Overall we have estimated that the benefits the organization is going to receive from the proposed system will surely overcome the initial costs and the later on running cost for system.

5.1.2 Technical Feasibility

This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provided in the system, as described in the System Requirement Specification (SRS), and checked if everything was possible using different type of frontend and backend platforms.

doubt the proposed system is fully GUI that is very user friendly and all inputs to be aken all self-explanatory even to a layman. Besides, a proper training has been conducted to let Low the essence of the system to the users so that they feel comfortable with new system. As for our study is concerned the clients are comfortable and happy as the system has cut down their loads and doing.

1.3 Operational Feasibility

5.1.4. Cost/ Benefit analysis

Economic analysis is the most frequently used method for evaluating the effectiveness of acandidate system. More commonly known as cost benefit analysis, the procedure is to determine the benefits and savings that are expected a candidate system and compare them with costs. If benefits overweigh costs, then the decision is made to design and implement the system. Otherwise, further justification or alterations in the proposed system will have to be made if it is to have a chance of being approved. This is an ongoing effort that improves in accuracy at each phase in the system life cycle.

Costs:

> Cost of new computer approximately Rs. 22,000/- > Cost of operating system approximately Rs. 5000/-

Benefits:

Avoids tedious typing task Faster document retrieval.

Saving storage space. Keeps data secure

> Easy to use, update and maintain

5.2 Testing

Sofware Testing is evaluation of the software against requirements gathered from users and ystem specifications. Testing is conducted at the phase level in software development life cycle wrat module level in program code. Software testing comprises of Validation and Verification

SOFTWARE VALIDATION:

Validation is process of examining whether or not the software satisfies the user requirements. It as carried out at the end of the SDLC. If the software matches requirements for which it was made, it is validated.

Validation ensures the product under development is as per the user requirements. Validation answers the question - "Are we developing the product which attempts that entire user needs from this software?" Validation emphasizes on user requirements.

SOFTWARE VERIFICATION:

Verification is the process of confirming if the software is meeting the business requirements, and id developed adhering to the proper specifications and methodologies.

> Verification ensures the product being developed is according to design specifications. Verification answers the question- "Are we developing this product by firmly following

all design specifications?" Verification concentrates on the design and system specifications.

Targets of the test are:

Errors. These are actual coding mistakes made by developers. In addition, there is a difference in output of software and desired output, is considered as an error.

Fault - When error exists fault occurs. A fault, also known as a bug, is a result of an error

which can cause system to fail.

> Failure - Failure is said to be the inability of the system to perform the desired te

Failure occurs when fault exists in the system.

5,3 Validation testing:

The process of evaluating software during the development process or at the md of the development process to determine whether it satisfies specified business requirements Validation Testing ensures that the product actually meets the client's needs. It can also be defined as to demonstrate that the product ful fills its intended use when deployed on appropriate

5.3.1 Work flow:

Validation testing can be best demonstrated using V-Model. The Software/product under test is evaluated during this type of testing.

43.2 Classifications of validation testing:

Unu Teating

Iegration Testing

Syviam Testing

User Acceptance Testing

5.3.3 Advantages of Validation: ADVANTAGES OF VALIDATION TESTING:

The users of the V-Model participate in the development and maintenance of the V Model Change control board publicly maintains the V-Model. The change control board meets anywhere from every day to weekly and processes all change requests received during system development and test

The V-Model provides concrete assistance on how to implement an activity and its work steps, defining explicitly the events needed to complete a work step: each activity schema contains instructions, recommendations and detailed explanations of the activity.

5.3.4 Limitation of Validation:

The placing of contracts for services in not regulated.

The organization and execution of operation, maintenance, repair and disposal of the system are not covered by the V-Model However, planning and preparation of a concept See these tasks are regulated in the V-Model

The V-Model addresses software development within a project rather than a whole organization

5.4 Unit Testing

Linit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with

functional correctness of the standalone modules. The main aim is to isolate each unit of the system to identify, analize and fix the defects

54.1 Advantages of Unit Testing:

Reduces Defects in the newly developed features or reduces bugs when changing the existing functionality.

Reduces Cost or Testing as defects are captured in very early phase.

+ Improves design and allows better refactoring of code.

Unit Tests, when integrated with build gives the quality of the build as well.

54.2 Life Cycle of Unit Testing:

5.5 Integration Testing:

integration testing is the phase in software testing in which individual software modules are mbined and tested as a group. Integration testing is conducted to evaluate the compliance of a tem or component with specified functional requirements. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, pups them in larger aggregates, applies tests defined in an integration test plan to those gregates, and delivers as its output the integrated system ready for system testing

5.5.1 Integration Strategies:

1. Big-Bang Integration

2 Top Down Integration

3. Bottom Up Integration

4. Hybrid Integration

1) Big-Bang Integration: Big bang approach integrates all the modules in one go le does not go for integrating the modules one by one. It verifies if the system works as expected or not once integrated. If any issue is detected in the completely integrated module, then it becomes difficult to find out which module has caused the issue.

2) Top-Down Integration: This technique starts from the topmost module and gradually progress towards the lower modules. Only the top module is unit tested in isolation. After this, the lower modules are integrated one by one. The process is repeated until all the modules are integrated and tested.

3) Bottom-Up Integration: Bottom-up testing, as the name suggests starts from the lowest or the innermost unit of the application, and gradually moves up. The Integration testing starts from the lowest module and gradually progresses towards the upper modules of the application. This integration continues till all the modules are integrated and the entire application is tested as a single unit.

4) Hybrid Integration: Hybrid Testing can be adopted if the customer wants to work on a working version of the application as soon as possible aimed at producing a basic working system in the earlier stages of the development cycle.

4.6 System Testing

System testing is testing conducted on a complete integrated system to evaluate the system's compliance with its specified requirements.

System testing takes, as its input, all of the integrated components that have passed integration esting. The purpose of integration testing is to detect any inconsistencies between the units that are integrated together (called assemblages). System testing seeks to detect defects both within the "inter-assemblages" and also within the system as a whole. The actual result is the behavior produced or observed when a component or system is tested.

56.1 Types of System Testing:

Uability Testing: To test that the user interfaces are user friendly and easy to operate and Documentation Testing: To test that the user manual and system guide is correct andcomplete

Functionality Testing: To test that the system functionalities are behaving as expected and specified in software requirements document.

software products.

⚫ later-operability Testing: To test that the system compatibility with other third-party Performance Testing To test that the Performance of the system and to make sure system does not break while operating with available resources.

Scalability Testing: To test that the system is scalable enough in terms of user,

geographic, resources, etc.

♦ Stress Testing: To test that the system behavior when it is operated under stress and tomark its break point.

Load & Stability Testing: To test that the system is capable enough to withstand expectedload without breaking down.

\* Reliability Testing: To test how long the system can operate without developing any issue or error

Testing To make sure that new factionalities added to the system does 4 Regression break the existing functionalities.

Compliance & Regulatory Testing To test that the system specification and operation compiles well with required regulator

Security Testing: To test that the system is secured enough to protect it from unintended

Recoverability Testing To test how well the system can recover after breakdown or

outage without impacting the business.

4.7 Acceptance Testing:

Acceptance testing, a testing technique performed to determine whether or not the software system has met the requirement specifications. The main purpose of this test is to evaluate the ystem's compliance with the business requirements and verify if it is has met the required criteria for delivery to end users

There are various forms of acceptance testing

• User acceptance Testing

• Business acceptance Testing

Alpha Testing

Beta Testing

User acceptance Testing: This may include factory acceptance testing (FAT), ie the esting done by a vendor before the product or system is moved to its destination site, affer which site scceptance testing (SAT) may be performed by the users at the site

Business Acceptance Testing: BAT is used to determine whether the product meets thi business goals and purposes or not. BAT mainly focuses on business profits which are quise challenging due to the changing market conditions and new technologies so that the curren implementation may have to being changed which result in extra budgets.

Alpha Testing: Alpha testing is used to determine the product in the development testing environment by a specialized tester's team usually called alpha testers.

Beta Testing: Beta testing is used to assess the product by exposing it to the real end-users, usually called beta testers in their environment. Feedback is collected from the users and the defects are fixed. Also, this helps in enhancing the product to give a rich user experience

5.8 Test Testing:

A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works corks correctly.

The proess of developing test cases can also help find problems in the requirements or design of an application.

A test case is a specification of the inputs, execution conditions, testing procedure, and expected results that defines a single test to be executed to achieve a particular software test objective, such as to exercise a particular program path or to verify compliance with specific equipment. Test cases underlie testing that is methodical rather than haphazard. A battery of test cases can be built to produce the desired coverage of the software being tested. Formally defined test cases allow the same tests to be run repeatedly against successive versions of the software, allowing for effective and consistent regression testing.

Test cases

|  |  |  |  |
| --- | --- | --- | --- |
| Test case number | Testing scenario | Expected result | Result |
|  | Login testing |  |  |
| TC-01 | Clicking submit without entering details | Alert “plese enter all details” | PASS |
| TC-02 | Clicking submit without entering username | Alert “plese enter all username” | PASS |
| TC-03 | Clicking submit without entering password | Alert “plese enter all password” | PASS |
| TC04 | Clicking submit without entering email id | Alert “plese enter all email id” | PASS |
| TC-05 | Clicking submit without entering mobile no | Alert “plese enter all mobile no” | PASS |
| TC-06 | Clicking submit without entering bill no | Alert “plese enter all bill no” | PASS |
| TC-07 | Clicking submit without entering item name | Alert “plese enter all item name” | PASS |

Future Enhancement

>Detailed information gathering has to be done. Without that the purpose for using the software wont be satisfied properly.

> However it can give good profits in the long rum plementing the software requires change in the business practices

>Efficient organization of all knowledge is the analysis company and easy analysis

and retrieval of information in possible

>In this project we can also include BAR CODE facility using the bar code reader, which will detect the expiry date and the other information about the related medicines

> Company using this software will always be able to plan in future and always be aware of their financial position in the market

> leads to streamlining of business processes.

>The implementation and maintenance costs nun very high (about 2 to 3% of the company's revenue.)

Conclusion

It has been a great pleasure for us to work on the project We started this get with the of going our best and doing something eresting and we believes that we fully achieved that. If we were able to complete this project, this is largely due the collective efforts

This project proved good for us as it provided practical knowledge of not only programming in Vlame NET application and know some extent windows application and SQL server

The DBMS package is very efficient and user friendly and way to store the details that com . Although all features are not provided the important ones need to make this software face are present. The project has been successful in that manner.

The tables that are created for this software have been designed m such a way that complete informatman is given in a concise manner. Few of the functions provided by this software are new records,saved and modified.

The software is portable and flexible for future enhancements. Thus, this project helps intrieving the deal as ungle go using data base concept.

Its concluded that the objective of the project work entitled "PHARMACY MANAGEMENT SYSTEM" filled and desired result has been obtained. Each program in the project are hanctioning normally and reliable to user.

BIBLIOGRAPHY

Books:

+ Ashwini S Divakar "Database Management System" and "Software Engineering Skyward publications.

Online References:

<https://youtu.be/kerOAF473LM>

<https://youtu.be/iD17eWr10Js>

<http://www.pinterest.com>

<http://www.wikipedia.com>