

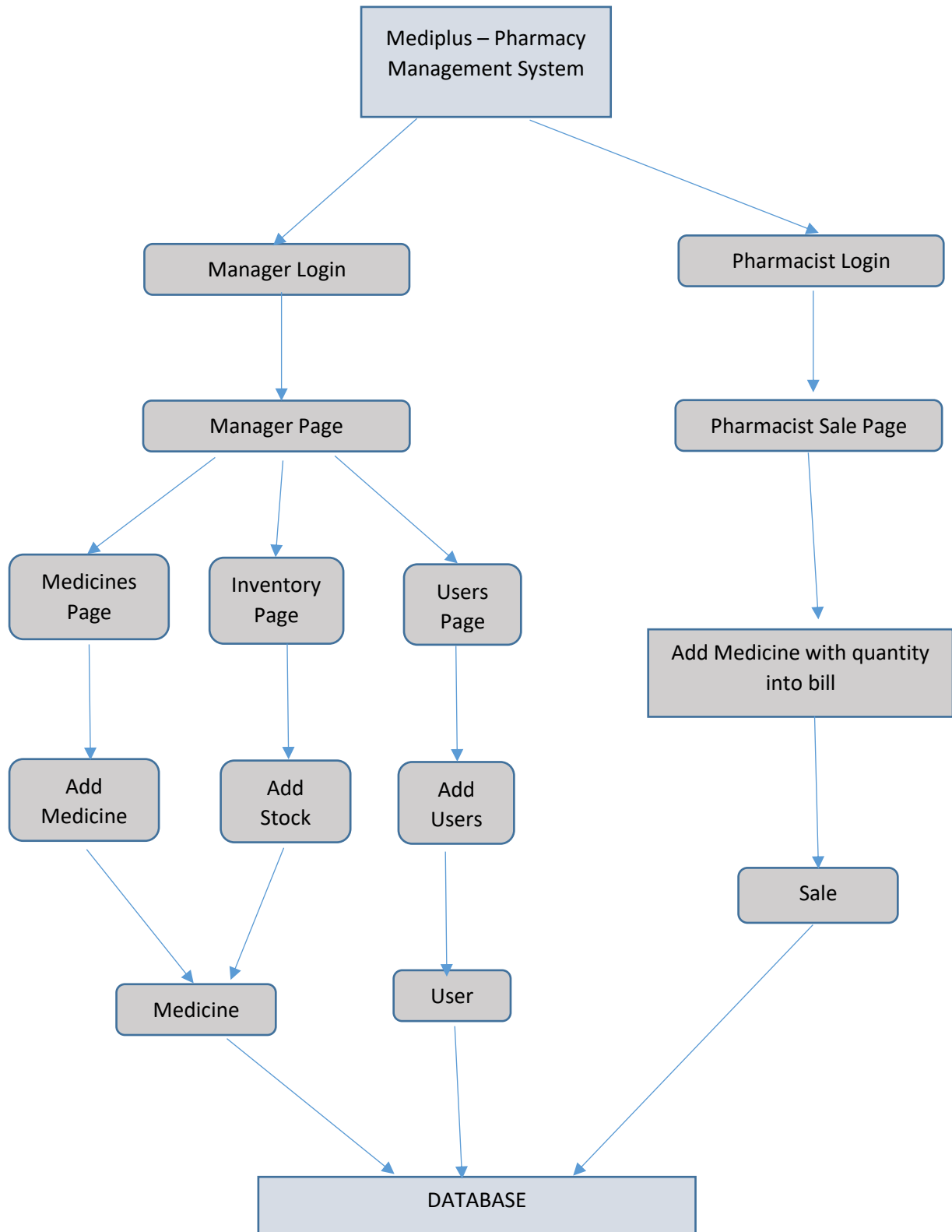
# **SYSTEM DESIGN**

## **INDRODUCTION TO SYSTEM DESIGN**

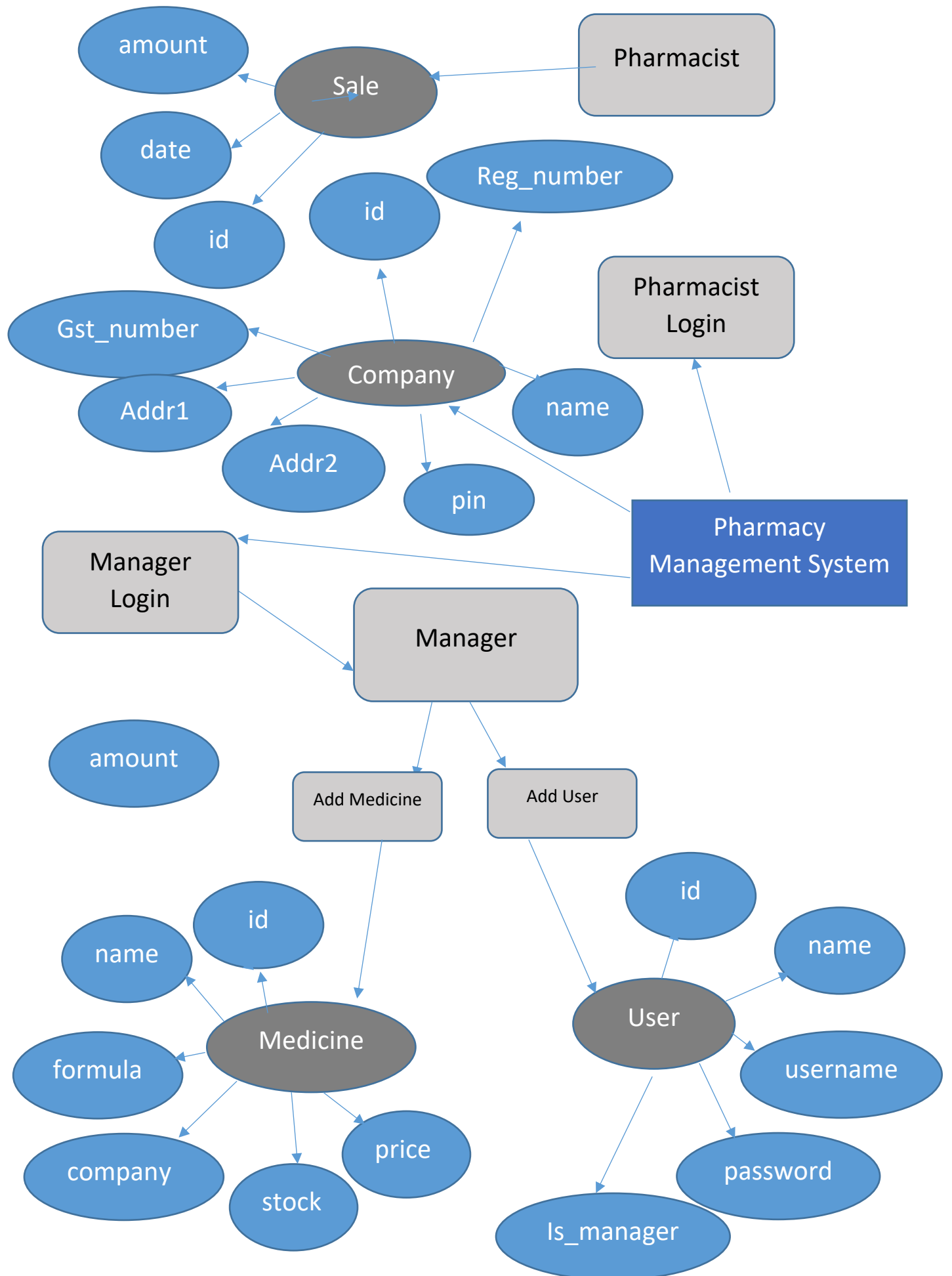
The design phase is the life cycle phase in which the detailed design of the selected system in the study phase is accomplished. In the design phase, the technical specifications are prepared for the performed of all allocated tasks. It also includes the construction of programs and programs testing. In the design phase, the first step is to determine the output is to produced and in what format.

Systems design is the process of defining elements of a system like modules, architecture, components and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing and designing systems which satisfies the specific needs and requirements of a business or organization

# **DATA FLOW DIAGRAM**



# **ER DIAGRAM**



# **DATABASE TABLES**

### **User Table (users)**

Column Name	Data Type
id	Int
name	varchar
username	varchar
password	varchar
Is_admin	bool

### **Medicine Table (medicines)**

Column Name	Data Type
id	int
name	varchar
formula	varchar
company	varchar
price	int
stock	int

### **Sale Table (sales)**

Column Name	Data Type
id	int
date	datetime
amount	int



### **Company Table (sales)**

Column Name	Data Type
id	int
name	varchar
reg_number	varchar
gst_number	varchar
addr1	varchar
addr2	varchar
pin	varchar

**CODE**

## Main Form – Form1.vb

Imports MySql.Data.MySqlClient

Public Class Form1

Public DbConnection As MySqlConnection

Public LoggedInUserName As String

Dim AddCompanyPage As AddCompany

Dim AddManagerPage As AddManager

Dim ManagerLoginPage As ManagerLogin

Public ManagerPage As Manager

Dim PharmacistLoginPage As PharmacistLogin

Public PharmacistPage As Pharmacist

Private Sub Form1\_Load(sender As Object, e As EventArgs) Handles MyBase.Load  
InitDbConnection()

LoadInitailPage()

End Sub

Private Sub InitDbConnection()

Dim ConnectionString =

"server=127.0.0.1;uid=root;pwd=admin;database=mediplus"

DbConnection = New MySqlConnection With {

.ConnectionString = ConnectionString

}

' Create tables

DbHelper.CreateTables()

End Sub

Private Sub LoadInitailPage()

CompanyModule.CheckCompanyAdded()

If CompanyModule.IsCompanyAdded Then

UserModule.IsUsersExists()

If UserModule.UserExists Then

LoadPharmacistLoginPage()

Else

LoadAddManagerPage()

End If

Else

LoadAddCompanyPage()

End If

End Sub

Public Sub LoadAddCompanyPage()

AddCompanyPage = New AddCompany With {

.TopLevel = False,

.FormBorderStyle = FormBorderStyle.None

}

AppBodyPanel.Controls.Clear()

AppBodyPanel.Controls.Add(AddCompanyPage)

AddCompanyPage.Show()

End Sub

```

Public Sub LoadAddManagerPage()
    AddManagerPage = New AddManager With {
        .TopLevel = False,
        .FormBorderStyle = FormBorderStyle.None
    }

    AppBodyPanel.Controls.Clear()
    AppBodyPanel.Controls.Add(AddManagerPage)

    AddManagerPage.Show()
End Sub

Public Sub LoadManagerLoginPage()
    ManagerLoginPage = New ManagerLogin With {
        .TopLevel = False,
        .FormBorderStyle = FormBorderStyle.None
    }

    AppBodyPanel.Controls.Clear()
    AppBodyPanel.Controls.Add(ManagerLoginPage)

    ManagerLoginPage.Show()
End Sub

Public Sub LoadManagerPage()
    ManagerPage = New Manager With {
        .TopLevel = False,
        .FormBorderStyle = FormBorderStyle.None
    }

    AppBodyPanel.Controls.Clear()
    AppBodyPanel.Controls.Add(ManagerPage)

    ManagerPage.Show()
End Sub

Public Sub LoadPharmacistLoginPage()
    PharmacistLoginPage = New PharmacistLogin With {
        .TopLevel = False,
        .FormBorderStyle = FormBorderStyle.None
    }

    AppBodyPanel.Controls.Clear()
    AppBodyPanel.Controls.Add(PharmacistLoginPage)

    PharmacistLoginPage.Show()
End Sub

Public Sub LoadPharmacistPage()
    PharmacistPage = New Pharmacist With {
        .TopLevel = False,
        .FormBorderStyle = FormBorderStyle.None
    }

    AppBodyPanel.Controls.Clear()
    AppBodyPanel.Controls.Add(PharmacistPage)

    PharmacistPage.Show()
End Sub

End Class

```

## Add Company Page

```
Public Class AddCompany
    Private Sub Button1_Click(sender As Object, e As EventArgs)
        Form1.LoadAddManagerPage()
    End Sub

    Private Sub Button1_Click_1(sender As Object, e As EventArgs) Handles
Button1.Click
        Dim Name = NameInput.Text
        Dim RegNum = RegNumInput.Text
        Dim GSTNum = GSTNumInput.Text
        Dim Addr1 = Addr1Input.Text
        Dim Addr2 = Addr2Input.Text
        Dim PIN = PINInput.Text

        CompanyModule.AddCompany(Name, RegNum, GSTNum, Addr1, Addr2, PIN)
    End Sub
End Class
```

## Add Manager Page

```
Public Class AddManager
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Validation.IsValidName(NameInput.Text)
        If (Validation.IsNameInvalid) Then
            MessageBox.Show("Name cannot contain numeric characters")
            Exit Sub
        Else
            End If
        Dim Name = NameInput.Text
        Dim Username = UsernameInput.Text
        Dim Password = PasswordInput.Text

        ManagerModule.AddManager(Name, Username, Password)
    End Sub
End Class
```

## Manager Login Page

```
Public Class ManagerLogin
    Private Sub LoginButton_Click(sender As Object, e As EventArgs) Handles
LoginButton.Click
        Dim Username = UsernameInput.Text
        Dim Password = PasswordInput.Text

        ManagerModule.LoginManager(Username, Password)
    End Sub

    Private Sub GoToPharmacistLoginButton_Click(sender As Object, e As EventArgs)
Handles GoToPharmacistLoginButton.Click
        Form1.LoadPharmacistLoginPage()
    End Sub
End Class
```

## Manager Page

```
Public Class Manager
```

```

Public NameOfCompany As String
Public CompanyRegNo As String
Public CompanyGSTNo As String

Dim HomePage As ManagerHome

Dim MedicinesPage As ManagerMedicines
Dim AddMedicinePage As ManagerAddMedicine

Dim InventoryPage As ManagerInventory
Dim AddInventoryPage As ManagerAddInventory

Dim UsersPage As ManagerUsers
Dim AddUserPage As ManagerAddUser
Private Sub Manager_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
    CompanyModule.GetCompanyData()
    NameOfCompany = CompanyModule.CompanyName
    CompanyRegNo = CompanyModule.CompanyRegNo
    CompanyGSTNo = CompanyModule.CompanyGSTNo

    LoadHomePage()
End Sub

Public Sub LoadHomePage()
    HomePage = New ManagerHome

    ManagerBodyPanel.Controls.Clear()
    ManagerBodyPanel.Controls.Add(HomePage)
End Sub

Public Sub LoadMedicinesPage()
    MedicinesPage = New ManagerMedicines

    ManagerBodyPanel.Controls.Clear()
    ManagerBodyPanel.Controls.Add(MedicinesPage)
End Sub

Public Sub LoadAddMedicinePage()
    AddMedicinePage = New ManagerAddMedicine

    ManagerBodyPanel.Controls.Clear()
    ManagerBodyPanel.Controls.Add(AddMedicinePage)
End Sub

Public Sub LoadInventoryPage()
    InventoryPage = New ManagerInventory

    ManagerBodyPanel.Controls.Clear()
    ManagerBodyPanel.Controls.Add(InventoryPage)
End Sub

Public Sub LoadAddInventoryPage()
    AddInventoryPage = New ManagerAddInventory

    ManagerBodyPanel.Controls.Clear()
    ManagerBodyPanel.Controls.Add(AddInventoryPage)
End Sub

Public Sub LoadUsersPage()
    UsersPage = New ManagerUsers

    ManagerBodyPanel.Controls.Clear()

```

```

        ManagerBodyPanel.Controls.Add(UsersPage)
    End Sub

    Public Sub LoadAddUserPage()
        AddUserPage = New ManagerAddUser

        ManagerBodyPanel.Controls.Clear()
        ManagerBodyPanel.Controls.Add(AddUserPage)
    End Sub
End Class

```

## **Manager Home Page**

```

Public Class ManagerHome
    Private Sub ManagerHome_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        MedicineModule.LoadMedicinesData()
        MedicineCountLabel.Text = MedicineModule.Medicines.Count

        MedicineModule.LoadOutOfStockMedicinesData()
        OutOfStockMedicineCountLabel.Text =
MedicineModule.OutOfStockMedicines.Count

        UserModule.LoadUserData()
        UserCountLabel.Text = UserModule.Users.Count

        UserNameLabel.Text = Form1.LoggedInUserName
        CompanyNameLabel.Text = Form1.ManagerPage.NameOfCompany
        CompanyRegNoLabel.Text = Form1.ManagerPage.CompanyRegNo
        CompanyGSTNoLabel.Text = Form1.ManagerPage.CompanyGSTNo
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Form1.ManagerPage.LoadMedicinesPage()
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
Button2.Click
        Form1.ManagerPage.LoadInventoryPage()
    End Sub

    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles
Button3.Click
        Form1.ManagerPage.LoadUsersPage()
    End Sub

    Private Sub Button4_Click(sender As Object, e As EventArgs) Handles
Button4.Click
        Form1.LoggedInUserName = ""
        Form1.LoadPharmacistLoginPage()
    End Sub
End Class

```

## **Manager Medicines Page**

```

Public Class ManagerMedicines
    Private Sub ManagerMedicines_Load(sender As Object, e As EventArgs) Handles
MyBase.Load

```

```

        MedicineModule.LoadMedicinesData()
        Dim CurrentItem = 0
        For Each Medicine As MedicineModel In MedicineModule.Medicines
            MedicineDetailsListView.Items.Add(Medicine.Id)

MedicineDetailsListView.Items(CurrentItem).SubItems.Add(Medicine.Name)
            CurrentItem += 1
        Next

        CurrentItem = 0
        For Each Medicine As MedicineModel In MedicineModule.Medicines

MedicineDetailsListView.Items(CurrentItem).SubItems.Add(Medicine.Formula)
            CurrentItem += 1
        Next

        CurrentItem = 0
        For Each Medicine As MedicineModel In MedicineModule.Medicines

MedicineDetailsListView.Items(CurrentItem).SubItems.Add(Medicine.Company)
            CurrentItem += 1
        Next

        CurrentItem = 0
        For Each Medicine As MedicineModel In MedicineModule.Medicines

MedicineDetailsListView.Items(CurrentItem).SubItems.Add(Medicine.Price)
            CurrentItem += 1
        Next
    End Sub

    Private Sub Button1_Click_1(sender As Object, e As EventArgs) Handles
Button1.Click
        Form1.ManagerPage.LoadAddMedicinePage()
    End Sub

    Private Sub Button2_Click_1(sender As Object, e As EventArgs) Handles
Button2.Click
        Form1.ManagerPage.LoadHomePage()
    End Sub
End Class

```

## Manager Add Medicine Page

```

Public Class ManagerAddMedicine
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Form1.ManagerPage.LoadMedicinesPage()
    End Sub

    Private Sub AddMedicineButton_Click(sender As Object, e As EventArgs) Handles
AddMedicineButton.Click
        Dim Name = NameInput.Text
        Dim Formula = FormulaInput.Text
        Dim Company = CompanyInput.Text
        Dim Price = PriceInput.Value

        If Price <= 0 Then
            MessageBox.Show("Invalid price")
        Else

```



```

        MedicineModule.AddNewMedicine(Name, Formula, Company, Price)
    End If
End Sub
End Class

```

## Manager Inventory Page

```

Public Class ManagerInventory
    Private Sub ManagerInventory_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        MedicineModule.LoadInventoryData()
        Dim CurrentItem = 0
        For Each Medicine As MedicineStockModel In
MedicineModule.MedicineInventory
            MedicineInventoryList.Items.Add(Medicine.Id)
            MedicineInventoryList.Items(CurrentItem).SubItems.Add(Medicine.Name)
            CurrentItem += 1
        Next

        CurrentItem = 0
        For Each Medicine As MedicineStockModel In
MedicineModule.MedicineInventory
            MedicineInventoryList.Items(CurrentItem).SubItems.Add(Medicine.Stock)
            CurrentItem += 1
        Next
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Form1.ManagerPage.LoadHomePage()
    End Sub

    Private Sub AddStockButton_Click(sender As Object, e As EventArgs) Handles
AddStockButton.Click
        Form1.ManagerPage.LoadAddInventoryPage()
    End Sub
End Class

```

## Manager Add Inventory Page

```

Public Class ManagerAddInventory
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Form1.ManagerPage.LoadInventoryPage()
    End Sub

    Private Sub ManagerAddInventory_Load(sender As Object, e As EventArgs)
Handles MyBase.Load
        MedicineModule.LoadMedicinesData()
        MedicineOptions.Items.Clear()

        For Each Medicine As MedicineStockModel In
MedicineModule.MedicineInventory
            MedicineOptions.Items.Add(Medicine.Name)
        Next
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
Button2.Click
        If (MedicineOptions.SelectedIndex < 0) Then

```

```

        MessageBox.Show("No medicine selected")
    Else
        MedicineModule.LoadMedicinesData()

        Dim SelectedMedicine =
MedicineModule.Medicines.ElementAt(MedicineOptions.SelectedIndex)

        MedicineModule.AddStock(SelectedMedicine.Id, StockInput.Value)

    End If
End Sub
End Class

```

## Manager Users Page

```

Public Class ManagerUsers
    Private Sub ManagerUsers_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        UserModule.LoadUserData()
        Dim CurrentItem = 0
        For Each User As UserModel In UserModule.Users
            UsersListView.Items.Add(User.Id)
            UsersListView.Items(CurrentItem).SubItems.Add(User.Name)
            CurrentItem += 1
        Next

        CurrentItem = 0
        For Each User As UserModel In UserModule.Users
            UsersListView.Items(CurrentItem).SubItems.Add(User.Username)
            CurrentItem += 1
        Next

        CurrentItem = 0
        For Each User As UserModel In UserModule.Users
            If User.IsManager Then
                UsersListView.Items(CurrentItem).SubItems.Add("Manager")
            Else
                UsersListView.Items(CurrentItem).SubItems.Add("Pharmacist")
            End If
            CurrentItem += 1
        Next
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
Button2.Click
        Form1.ManagerPage.LoadHomePage()
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Form1.ManagerPage.LoadAddUserPage()
    End Sub
End Class

```

## Manager Add User Page

```

Public Class ManagerAddUser

```

```

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Form1.ManagerPage.LoadUsersPage()
    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
Button2.Click
        Validation.IsValidName(NameInput.Text)
        If (Validation.IsNameInvalid) Then
            MessageBox.Show("Name cannot contain numeric characters")
            Exit Sub
        End If
        Dim Name = NameInput.Text
        Dim Username = UsernameInput.Text
        Dim Password = PasswordInput.Text
        Dim IsManager As Boolean

        If (IsAdminInput.Text = "Manager") Then
            IsManager = True
        Else
            IsManager = False
        End If

        UserModule.AddUser(Name, Username, Password, IsManager)
    End Sub
End Class

```

## Pharmacist Login Page

```

Public Class PharmacistLogin
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Dim Username = UsernameInput.Text
        Dim Password = PasswordInput.Text

        PharmacistModule.PharmacistLogin(Username, Password)
    End Sub

    Private Sub GoToAdminLoginButton_Click(sender As Object, e As EventArgs)
Handles GoToAdminLoginButton.Click
        Form1.LoadManagerLoginPage()
    End Sub
End Class

```

## Pharmacist Page

```

Public Class Pharmacist
    Public NameOfCompany As String
    Public CompanyRegNo As String
    Public CompanyGSTNo As String

    Dim Homepage As PharmacistHome
    Dim SalesPage As PharmacistSales
    Private Sub Pharmacist_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        CompanyModule.GetCompanyData()
        NameOfCompany = CompanyModule.CompanyName
        CompanyRegNo = CompanyModule.CompanyRegNo
        CompanyGSTNo = CompanyModule.CompanyGSTNo
        LoadHomePage()
    End Sub

```

```

Public Sub LoadHomePage()
    Homepage = New PharmacistHome

    PharmacistBodyPanel.Controls.Clear()
    PharmacistBodyPanel.Controls.Add(Homepage)
End Sub

Public Sub LoadSalesPage()
    SalesPage = New PharmacistSales

    PharmacistBodyPanel.Controls.Clear()
    PharmacistBodyPanel.Controls.Add(SalesPage)
End Sub
End Class

```

## Pharmacist Home Page

```

Public Class PharmacistHome
    Private Sub PharmacistHome_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        SaleModule.GetSalesReport()
        SalesCountLabel.Text = SaleModule.TodaySalesCount
        TotalSale.Text = SaleModule.TodaySalesTotal

        UserNameLabel.Text = Form1.LoggedInUserName
        CompanyNameLabel.Text = Form1.PharmacistPage.NameOfCompany
        CompanyRegNoLabel.Text = Form1.PharmacistPage.CompanyRegNo
        CompanyGSTNoLabel.Text = Form1.PharmacistPage.CompanyGSTNo
    End Sub

    Private Sub Button3_Click_1(sender As Object, e As EventArgs) Handles
Button3.Click
        Form1.PharmacistPage.LoadSalesPage()
    End Sub

    Private Sub Button2_Click_1(sender As Object, e As EventArgs) Handles
Button2.Click
        Form1.LoggedInUserName = ""
        Form1.LoadPharmacistLoginPage()
    End Sub
End Class

```

## Pharmacist Sales Page

```

Imports System.Drawing.Printing

Public Class PharmacistSales
    Dim ItemsInBill As New List(Of BillItem)
    Dim Total As Integer

    Dim WithEvents PD As New PrintDocument
    Dim PPD As New PrintPreviewDialog
    Dim PaperHeight = 150

    Private Sub PharmacistSales_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        LoadMedicineDataIntoComboBox()
    End Sub

```

```

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
Button2.Click
        Form1.PharmacistPage.LoadHomePage()
    End Sub

    Private Sub AddToBillButton_Click(sender As Object, e As EventArgs) Handles
AddToBillButton.Click
        If (QuantityInput.Value <= 0) Then
            MessageBox.Show("Invalid quantity")
        ElseIf (MedicineSelector.SelectedIndex < 0) Then
            MessageBox.Show("Please select a medicine")
        Else
            Dim NewItem = New BillItem With {
                .Id =
MedicineModule.Medicines.ElementAt(MedicineSelector.SelectedIndex).Id,
                .Name =
MedicineModule.Medicines.ElementAt(MedicineSelector.SelectedIndex).Name,
                .Price =
MedicineModule.Medicines.ElementAt(MedicineSelector.SelectedIndex).Price,
                .Quantity = QuantityInput.Value,
                .Amount =
MedicineModule.Medicines.ElementAt(MedicineSelector.SelectedIndex).Price *
QuantityInput.Value
            }

            Dim AlreadyExists = False
            Dim Index As Integer
            For Each Item As BillItem In ItemsInBill
                If (Item.Id = NewItem.Id) Then
                    AlreadyExists = True
                    Index = ItemsInBill.IndexOf(Item)
                End If
            Next

            If (AlreadyExists) Then

                Dim Quantity = ItemsInBill.ElementAt(Index).Quantity +
QuantityInput.Value
                MedicineModule.GetStockById(NewItem.Id)
                If (MedicineModule.MedicineStock < Quantity) Then
                    MessageBox.Show("Only " +
MedicineModule.MedicineStock.ToString() + " available")
                Else
                    ItemsInBill.ElementAt(Index).Quantity = Quantity
                    ItemsInBill.ElementAt(Index).Amount =
ItemsInBill.ElementAt(Index).Amount + QuantityInput.Value *
ItemsInBill.ElementAt(Index).Price
                End If
            Else
                MedicineModule.GetStockById(NewItem.Id)
                If (MedicineModule.MedicineStock < NewItem.Quantity) Then
                    MessageBox.Show("Only " +
MedicineModule.MedicineStock.ToString() + " available")
                Else
                    ItemsInBill.Add(NewItem)
                End If
            End If
            RenderListInGridView()
            ResetForm()
        End If
    End Sub

    Private Sub CheckoutBill()

```

```

    If (ItemsInBill.Count < 1) Then
        MessageBox.Show("Empty Bill")
    Else
        For Each Item As BillItem In ItemsInBill
            MedicineModule.AdjustInventory(Item.Id, Item.Quantity)
        Next

        SaleModule.CheckoutSale(Total)
        PrintBill()
        ResetForm()
        BillView.Rows.Clear()
        Total = 0
        ItemsInBill.Clear()
        TotalAmountLabel.Text = 0
    End If
End Sub

Private Sub LoadMedicineDataIntoComboBox()
    MedicineModule.LoadMedicinesData()

    For Each Medicine As MedicineModel In MedicineModule.Medicines
        MedicineSelector.Items.Add(Medicine.Name)
    Next
End Sub

Private Sub RenderListInGridView()
    BillView.Rows.Clear()
    Total = 0
    For Each Item As BillItem In ItemsInBill
        BillView.Rows.Add(Item.Name, Item.Price, Item.Quantity, Item.Amount)

        Total = Total + Item.Amount
    Next

    TotalAmountLabel.Text = Total
End Sub

Private Sub ResetForm()
    MedicineSelector.SelectedIndex = -1
    QuantityInput.Value = 1
End Sub

Private Sub CheckoutButton_Click(sender As Object, e As EventArgs) Handles
CheckoutButton.Click
    CheckoutBill()
End Sub

Private Sub ResetBillButtton_Click(sender As Object, e As EventArgs) Handles
ResetBillButtton.Click
    ResetForm()
    BillView.Rows.Clear()
    Total = 0
    ItemsInBill.Clear()
    TotalAmountLabel.Text = 0
End Sub

Private Sub PD_BeginPrint(sender As Object, e As PrintEventArgs) Handles
PD.BeginPrint
    Dim PageSetup As New PageSettings With {
        .PaperSize = New PaperSize("Custom", 250, PaperHeight)
    }

```

```

    PD.DefaultPageSettings = PageSetup
End Sub

Private Sub PD_PrintPage(sender As Object, e As PrintPageEventArgs) Handles
PD.PrintPage
    Dim FNormal As New Font("Calibri", 10, FontStyle.Regular)
    Dim FSmall As New Font("Calibri", 8, FontStyle.Regular)
    Dim FBold As New Font("Calibri", 10, FontStyle.Bold)
    Dim FBoldBig As New Font("Comic Sans MS", 12, FontStyle.Bold)

    Dim LeftMargin = PD.DefaultPageSettings.Margins.Left
    Dim CenterMargin = PD.DefaultPageSettings.PaperSize.Width / 2
    Dim RightMargin = PD.DefaultPageSettings.PaperSize.Width

    Dim RightAlign = New StringFormat With {
        .Alignment = StringAlignment.Far
    }
    Dim CenterAlign = New StringFormat With {
        .Alignment = StringAlignment.Center
    }
    Dim LeftAlign = New StringFormat With {
        .Alignment = StringAlignment.Near
    }

    Dim LineBraek = "-----"
    -----

    ' Writing string

    e.Graphics.DrawString("Mediplus", FBoldBig, Brushes.Black, CenterMargin,
10, CenterAlign)
    e.Graphics.DrawString(Form1.PharmacistPage.NameOfCompany + ", GST :" +
Form1.PharmacistPage.CompanyGSTNo, FNormal, Brushes.Black, CenterMargin, 27,
CenterAlign)
    e.Graphics.DrawString(Date.Now().ToString("dd-MM-yy | HH:mm"), FSmall,
Brushes.Black, 10, 40, LeftAlign)
    e.Graphics.DrawString(LineBraek, FSmall, Brushes.Black, 10, 50,
LeftAlign)
    e.Graphics.DrawString("#", FSmall, Brushes.Black, 20, 57, CenterAlign)
    e.Graphics.DrawString("Item", FSmall, Brushes.Black, 35, 57, LeftAlign)
    e.Graphics.DrawString("Price", FSmall, Brushes.Black, RightMargin - 100,
57, CenterAlign)
    e.Graphics.DrawString("Qty", FSmall, Brushes.Black, RightMargin - 70, 57,
CenterAlign)
    e.Graphics.DrawString("Amount", FSmall, Brushes.Black, RightMargin - 35,
57, CenterAlign)
    e.Graphics.DrawString(LineBraek, FSmall, Brushes.Black, 10, 64,
LeftAlign)

    Dim BaseLine = 63
    Dim ItemCount = 1
    For Each Item As BillItem In ItemsInBill
        e.Graphics.DrawString(ItemCount, FNormal, Brushes.Black, 20, BaseLine
+ 12, CenterAlign)
        e.Graphics.DrawString(Item.Name, FNormal, Brushes.Black, 35, BaseLine
+ 12, LeftAlign)
        e.Graphics.DrawString(Item.Price, FNormal, Brushes.Black, RightMargin
- 100, BaseLine + 12, CenterAlign)
        e.Graphics.DrawString(Item.Quantity, FNormal, Brushes.Black,
RightMargin - 70, BaseLine + 12, CenterAlign)
        e.Graphics.DrawString(Item.Quantity, FNormal, Brushes.Black,
RightMargin - 20, BaseLine + 12, RightAlign)
    
```

```

        BaseLine += 12
        ItemCount += 1
    Next

    BaseLine += 11
    e.Graphics.DrawString(LineBraek, FSmall, Brushes.Black, 10, BaseLine,
LeftAlign)

    e.Graphics.DrawString("Total (Rs.) :", FBold, Brushes.Black, RightMargin
- 50, BaseLine + 10, RightAlign)
    e.Graphics.DrawString(Total, FBold, Brushes.Black, RightMargin - 20,
BaseLine + 10, RightAlign)

    e.Graphics.DrawString("Thank you, have a nice day", FSmall,
Brushes.Black, CenterMargin, BaseLine + 30, CenterAlign)

End Sub

Private Sub CalculatePaperHeight()
    Dim Height = 63
    For Each Item As BillItem In ItemsInBill
        Height += 12
    Next

    Height += 70

    PaperHeight = Height
End Sub

Private Sub PrintBill()
    CalculatePaperHeight()
    PPD.Document = PD
    PPD.ShowDialog()
End Sub
End Class

Class BillItem
    Public Id As Integer
    Public Name As String
    Public Price As Integer
    Public Quantity As Integer
    Public Amount As Integer
End Class

```

## **DBHelper.vb**

```
Imports MySql.Data.MySqlClient
```

```
Module DbHelper
```

```
    Sub CreateTables()
```

```
        Dim CompanyTableSchema = "create table If not exists company (id int not
null auto_increment primary key , name varchar(40) not null, reg_number
varchar(25) not null, gst_number varchar(20) not null, addr1 varchar(40) not
null, addr2 varchar(40), pin varchar(6) not null)"
```

```
        Dim UsersTableSchema = "create table if not exists users (id int not null
auto_increment primary key, name varchar(25) not null, username varchar(25) not
null, password varchar(20) not null, is_manager bool not null )"

```

```
        Dim MedicineTableSchema = "create table if not exists medicines (id int
not null auto_increment primary key, name varchar(50) not null, formula
varchar(25) not null, company varchar(25), price int, stock int)"

```



```

        Dim SalesTableSchema = "create table if not exists sales(id int not null
auto_increment primary key,date datetime default now(), amount int not null)"
        Try
            Form1.DbConnection.Open()
            Dim Cmd1 = New MySqlCommand(CompanyTableSchema, Form1.DbConnection)
            Dim cmd2 = New MySqlCommand(UsersTableSchema, Form1.DbConnection)
            Dim Cmd3 = New MySqlCommand(MedicineTableSchema, Form1.DbConnection)
            Dim Cmd4 = New MySqlCommand(SalesTableSchema, Form1.DbConnection)
            Cmd1.ExecuteNonQuery()
            cmd2.ExecuteNonQuery()
            Cmd3.ExecuteNonQuery()
            Cmd4.ExecuteNonQuery()
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        Finally
            Form1.DbConnection.Close()
        End Try
    End Sub
End Module

```

## ManagerModule.vb

```
Imports MySql.Data.MySqlClient
```

```
Module ManagerModule
```

```

    Sub AddManager(Name As String, Username As String, Password As String)
        Dim Sql = "insert into users (name, username, password, is_manager)
values(@Name, @Username, @Password, true)"
        Try
            Form1.DbConnection.Open()
            Dim Cmd = New MySqlCommand(Sql, Form1.DbConnection)
            Cmd.Parameters.AddWithValue("@Name", Name)
            Cmd.Parameters.AddWithValue("@Username", Username)
            Cmd.Parameters.AddWithValue("@Password", Password)

            Cmd.ExecuteNonQuery()
            Form1.DbConnection.Close()
            Form1.LoggedInUserName = Name
            ' Navigate to manager page
            Form1.LoadManagerPage()
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        Finally
            Form1.DbConnection.Close()
        End Try
    End Sub

    Sub LoginManager(Username As String, password As String)
        Dim Sql = "select name, username, password, is_manager from users where
username=@Username"
        Try
            Form1.DbConnection.Open()
            Dim Cmd = New MySqlCommand(Sql, Form1.DbConnection)
            Cmd.Parameters.AddWithValue("@Username", Username)

            Dim Reader = Cmd.ExecuteReader

            If Reader.Read() Then
                If Reader("is_manager") Then
                    If Reader("password") = password Then

```

```

        Dim Name = Reader("name")
        Form1.LoggedInUserName = Name

        Form1.DbConnection.Close()

        Form1.LoadManagerPage()
    Else
        MessageBox.Show("Invalid password")
    End If
Else
    MessageBox.Show("Invalid username")
End If
Else
    MessageBox.Show("Invalid username")
End If
Catch ex As Exception
    MessageBox.Show(ex.Message)
Finally
    Form1.DbConnection.Close()
End Try
End Sub
End Module

```

## UserModule.vb

```

Imports MySql.Data.MySqlClient

Module UserModule
    Public UserExists = False
    Public Users As New List(Of UserModel)

    Sub IsUsersExists()
        Dim Cmd = New MySqlCommand("select * from users", Form1.DbConnection)

        Try
            Form1.DbConnection.Open()
            Dim Reader = Cmd.ExecuteReader()

            If Reader.HasRows() = True Then
                UserExists = True
            End If
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        Finally
            Form1.DbConnection.Close()
        End Try
    End Sub

    Sub LoadUserData()
        Users.Clear()
        Dim Cmd = New MySqlCommand("select * from users", Form1.DbConnection)

        Try
            Form1.DbConnection.Open()
            Dim Reader = Cmd.ExecuteReader()

            If Reader.HasRows Then
                While Reader.Read()
                    Users.Add(New UserModel With {

```

```

        .Id = Reader("id"),
        .Name = Reader("name"),
        .Username = Reader("username"),
        .IsManager = Reader("is_manager")
    })
    End While
End If
Catch ex As Exception
    MessageBox.Show(ex.Message)

Finally
    Form1.DbConnection.Close()
End Try
End Sub

Sub AddUser(Name As String, Username As String, Password As String, IsManager
As Boolean)
    Dim Sql1 = "select name, username, password, is_manager from users where
username=@Username"
    Dim Sql2 = "insert into users (name, username, password, is_manager)
values (@Name, @Username, @Password, @IsManager)"

    Try
        Form1.DbConnection.Open()

        ' Check if the user already exists
        Dim CheckCmd = New MySqlCommand(Sql1, Form1.DbConnection)
        CheckCmd.Parameters.AddWithValue("@Username", Username)
        Dim Reader = CheckCmd.ExecuteReader

        If Reader.Read() = True Then
            MessageBox.Show("User laready exists")
            Return
        End If
        Form1.DbConnection.Close()

        ' Adding new user
        Form1.DbConnection.Open()
        Dim Cmd = New MySqlCommand(Sql2, Form1.DbConnection)
        Cmd.Parameters.AddWithValue("@Name", Name)
        Cmd.Parameters.AddWithValue("@Username", Username)
        Cmd.Parameters.AddWithValue("@Password", Password)
        Cmd.Parameters.AddWithValue("@IsManager", IsManager)
        Cmd.ExecuteScalar()

        Form1.DbConnection.Close()

        ' Navigate to user management page
        Form1.ManagerPage.LoadUsersPage()

    Catch ex As Exception
        MessageBox.Show(ex.Message)

    Finally
        Form1.DbConnection.Dispose()
    End Try
End Sub

End Module

Public Class UserModel
    Public Id As Integer
    Public Name As String

```

```

    Public Username As String
    Public IsManager As Boolean
End Class

```

## MedicineModule.vb

```
Imports MySql.Data.MySqlClient
```

```
Module MedicineModule
```

```

    Public MedicineInventory As New List(Of MedicineStockModel)
    Public Medicines As New List(Of MedicineModel)

```

```
    Public MedicineStock As Integer
```

```
    Public OutOfStockMedicines As New List(Of MedicineModel)
```

```

    Public Sub AddNewMedicine(name As String, formula As String, company As
String, price As Integer)
        Dim Sql = "insert into medicines (name, formula, company, price, stock)
value (@Name, @Formula, @Company, @Price, 0)"
        Dim Cmd = New MySqlCommand(Sql, Form1.DbConnection)
        Cmd.Parameters.AddWithValue("@Name", name)
        Cmd.Parameters.AddWithValue("@Formula", formula)
        Cmd.Parameters.AddWithValue("@Company", company)
        Cmd.Parameters.AddWithValue("@Price", price)
        Try
            Form1.DbConnection.Open()
            Cmd.ExecuteNonQuery()
            Form1.DbConnection.Close()

            ' Navigate to medicines page
            Form1.ManagerPage.LoadMedicinesPage()
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        Finally
            Form1.DbConnection.Close()
        End Try
    End Sub

    Public Sub LoadInventoryData()
        MedicineInventory.Clear()
        Dim Cmd = New MySqlCommand("select * from medicines", Form1.DbConnection)

        Try
            Form1.DbConnection.Open()
            Dim Reader = Cmd.ExecuteReader()

            If Reader.HasRows Then
                While Reader.Read()
                    MedicineInventory.Add(New MedicineStockModel With {
                        .Id = Reader("id"),
                        .Name = Reader("name"),
                        .Stock = Reader("stock")
                    })
                End While
            End If
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try
    End Sub

```

```

        Finally
            Form1.DbConnection.Close()
        End Try
    End Sub

    Public Sub LoadMedicinesData()
        Medicines.Clear()
        Dim Cmd = New MySqlCommand("select * from medicines", Form1.DbConnection)

        Try
            Form1.DbConnection.Open()
            Dim Reader = Cmd.ExecuteReader()

            If Reader.HasRows Then
                While Reader.Read()
                    Medicines.Add(New MedicineModel With {
                        .Id = Reader("id"),
                        .Name = Reader("name"),
                        .Formula = Reader("formula"),
                        .Company = Reader("company"),
                        .Price = Reader("price")
                    })
                End While
            End If
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        Finally
            Form1.DbConnection.Close()
        End Try
    End Sub

    Public Sub LoadOutOfStockMedicinesData()
        OutOfStockMedicines.Clear()
        Dim Cmd = New MySqlCommand("select * from medicines where stock = 0",
Form1.DbConnection)

        Try
            Form1.DbConnection.Open()
            Dim Reader = Cmd.ExecuteReader()

            If Reader.HasRows Then
                While Reader.Read()
                    OutOfStockMedicines.Add(New MedicineModel With {
                        .Id = Reader("id"),
                        .Name = Reader("name"),
                        .Formula = Reader("formula"),
                        .Company = Reader("company"),
                        .Price = Reader("price")
                    })
                End While
            End If
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        Finally
            Form1.DbConnection.Close()
        End Try
    End Sub

    Public Sub GetStockById(Id As Integer)
        Dim Cmd = New MySqlCommand("select (stock) from medicines where id =
@Id", Form1.DbConnection)

```

```

        Cmd.Parameters.AddWithValue("@Id", Id)

        Try
            Form1.DbConnection.Open()
            Dim Reader = Cmd.ExecuteReader()

            Reader.Read()
            MedicineStock = Reader("stock")
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        Finally
            Form1.DbConnection.Close()
        End Try
    End Sub

    Public Sub AddStock(MedicineId As Integer, Stock As Integer)
        Dim Cmd = New MySqlCommand("update medicines set stock = (@Stock + stock)
where id = @Id", Form1.DbConnection)
        Cmd.Parameters.AddWithValue("@Stock", Stock)
        Cmd.Parameters.AddWithValue("@Id", MedicineId)

        Try
            Form1.DbConnection.Open()
            Cmd.ExecuteNonQuery()
            Form1.DbConnection.Close()

            ' Navigate to Inventory page
            Form1.ManagerPage.LoadInventoryPage()
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        Finally
            Form1.DbConnection.Close()
        End Try
    End Sub

    Public Sub AdjustInventory(Id As Integer, Quantity As Integer)
        Dim Cmd = New MySqlCommand("update medicines set stock = (stock -
@Quantity) where id = @Id", Form1.DbConnection)
        Cmd.Parameters.AddWithValue("@Id", Id)
        Cmd.Parameters.AddWithValue("@Quantity", Quantity)

        Try
            Form1.DbConnection.Open()
            Cmd.ExecuteNonQuery()
            Form1.DbConnection.Close()
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        Finally
            Form1.DbConnection.Close()
        End Try
    End Sub
End Module

Class MedicineStockModel
    Public Id As Integer
    Public Name As String
    Public Stock As Integer
End Class

Class MedicineModel
    Public Id As Integer
    Public Name As String

```

```
Public Formula As String
Public Company As String
Public Price As Integer
End Class
```

## SaleModule.vb

```
Imports MySql.Data.MySqlClient
```

```
Module SaleModule
```

```
Public TodaySalesCount As Integer
```

```
Public TodaySalesTotal As Integer
```

```
Public Sub CheckoutSale(total As Integer)
```

```
    Dim Cmd = New MySqlCommand("insert into sales (amount) values (@Total)",  
Form1.DbConnection)
```

```
    Cmd.Parameters.AddWithValue("@Total", total)
```

```
    Try
```

```
        Form1.DbConnection.Open()
```

```
        Cmd.ExecuteNonQuery()
```

```
        Form1.DbConnection.Close()
```

```
    Catch ex As Exception
```

```
        MessageBox.Show(ex.Message)
```

```
    Finally
```

```
        Form1.DbConnection.Close()
```

```
    End Try
```

```
End Sub
```

```
Public Sub GetSalesReport()
```

```
    Dim Sql = "select count(id) as SalesCount, coalesce(sum(amount), 0) as  
TotalSales from sales where date(date) = curdate()"
```

```
    Dim Cmd = New MySqlCommand(Sql, Form1.DbConnection)
```

```
    Try
```

```
        Form1.DbConnection.Open()
```

```
        Dim Reader = Cmd.ExecuteReader()
```

```
        Reader.Read()
```

```
        TodaySalesCount = Reader("SalesCount")
```

```
        TodaySalesTotal = Reader("TotalSales")
```

```
    Catch ex As Exception
```

```
        MessageBox.Show(ex.Message)
```

```
    Finally
```

```
        Form1.DbConnection.Close()
```

```
    End Try
```

```
End Sub
```

```
End Module
```

## CompanyModule.vb

```
Imports MySql.Data.MySqlClient
```

```
Module CompanyModule
```

```
Public IsCompanyAdded = False
```

```
Public CompanyName As String
```

```
Public CompanyRegNo As String
```

```

Public CompanyGSTNo As String
Public Sub CheckCompanyAdded()
    Dim Cmd = New MySqlCommand("select * from company", Form1.DbConnection)

    Try
        Form1.DbConnection.Open()
        Dim Reader = Cmd.ExecuteReader()

        If Reader.HasRows() Then
            IsCompanyAdded = True
        End If
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    Finally
        Form1.DbConnection.Close()
    End Try
End Sub

Sub AddCompany(Name As String, RegNum As String, GSTNum As String, Addr1 As
String, Addr2 As String, PIN As String)
    Dim Sql = "insert into company (name, reg_number, gst_number, addr1,
addr2, pin) values(@Name, @RegNum, @GSTNum, @Addr1, @Addr2, @PIN)"
    Try
        Form1.DbConnection.Open()
        Dim Cmd = New MySqlCommand(Sql, Form1.DbConnection)

        Cmd.Parameters.AddWithValue("@Name", Name)
        Cmd.Parameters.AddWithValue("@RegNum", RegNum)
        Cmd.Parameters.AddWithValue("@GSTNum", GSTNum)
        Cmd.Parameters.AddWithValue("@Addr1", Addr1)
        Cmd.Parameters.AddWithValue("@Addr2", Addr2)
        Cmd.Parameters.AddWithValue("@PIN", PIN)

        Cmd.ExecuteNonQuery()
        Form1.DbConnection.Close()

        ' Navigate to add manager page
        Form1.LoadAddManagerPage()

    Catch ex As Exception
        MessageBox.Show(ex.Message)
    Finally
        Form1.DbConnection.Close()
    End Try
End Sub

Sub GetCompanyData()
    Dim Cmd = New MySqlCommand("select * from company", Form1.DbConnection)

    Try
        Form1.DbConnection.Open()
        Dim Reader = Cmd.ExecuteReader()

        If Reader.HasRows() Then
            While Reader.Read()
                CompanyName = Reader("name")
                CompanyRegNo = Reader("reg_number")
                CompanyGSTNo = Reader("gst_number")
            End While
        End If
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    Finally

```



```
        Form1.DbConnection.Close()  
    End Try  
End Sub  
End Module
```

## Validation.vb

```
Module Validation  
    Public IsNameInvalid = False  
    Public ErrorMessage As String  
  
    Public Sub IsValidName(Name As String)  
        IsNameInvalid = False  
        If Len(Name) = 0 Then  
            IsNameInvalid = True  
            ErrorMessage = "Please enter a name"  
  
            Exit Sub  
        End If  
  
        If Text.RegularExpressions.Regex.Match(Name, "\d").Success Then  
            IsNameInvalid = True  
            ErrorMessage = "Name can only contain alphabets"  
  
            Exit Sub  
        End If  
    End Sub  
End Module
```

# **SCREENSHOTS**

## Add Company Page

Form1

— □ ×

Welcome to **Mediplus**  
Pharmacy Management System

Please Add Your Company Details

Company Name

Reg No.

GST No.

Address Line 1

Address Line 2

PIN

Next

## Add Manager Page

Mediplus

— □ ×

Welcome to **Mediplus**  
Pharmacy Management System

Please Add Your Manager Details

Name

Username

Password

Create Manager

## Manager Login Page

Mediplus

— □ ×

**Mediplus Management Panel**

Please login to continue.

Username

Password

Login

Go To Pharmacist Login

## Manager Home Page

Mediplus

— □ ×

**Mediplus Management Panel** Logout

Total no. of medicines : 4

No. of out of stock medicines : 1

Total no. of users : 3

Logged in User : James Gordon

Company Name : GRAN Pharma

Company Reg No. : 566765

Company GST No. : GSTN899745

Medicines

Inventory

Users

## Manager Medicines Page

Mediplus

Go Back

Management Panel - Medicines

Add Medicine

Id	Name	Formula	Company	Price
1	Morphine	C17H19NO3	Altec	2
2	Ketamine	C13H16CINO	Cipla	4
3	Oxycotin	C43H66N12O1...	SUN Pharma	1
4	Paracetamol	C8H9NO2	Cipla	2

## Manager Add Medicine Page

Mediplus

Go Back

Management Panel - Add Medicine

Name

Formula

Company

Price

1

Add Mdicine

## Manager Inventory Page

Mediplus

Go Back

Management Panel - Inventory

Add Inventory

Id	name	Stock
1	Morphine	34
2	Ketamine	0
3	Oxycotin	190
4	Paracetamol	50

## Manager Add Inventory Page

Mediplus

Go Back

Management Panel - Add Inventory

Medicine

Units

0

Add Stock

## Manager Users Page

Mediplus

Go Back

Management Panel - Users

Add User

Id	Name	Username	User Type
1	James Gordon	jamesgordon	Manager
2	Rabecca	rabecca	Pharmacist
3	Jose carter	josecarter	Pharmacist

## Manager Add Users Page

Mediplus

Go Back

Management Panel - Add User

Name

Username

Password

User Type

Add User

## Pharmacist Login Page

Mediplus

— □ ×

### Mediplus Pharmacist Panel

Please Login to continue.

Username

Pssword

Login

Go To Manager Login

## Pharmacist Home Page

Mediplus

— □ ×

Mediplus Dashboard

Logout

Total No. of Sales Today : 3

Total Sale Today (Rs.) : 67

Logged in User : Rabeca

Company Name : GRAN Pharma

Company Reg No. : 566765

Company GST No. : GSTN899745

Sales



## Pharmacist Sales Page

Mediplus

Go Back

Sales - Pharmacist Panel

1

▼

Add to Bill

Reset Bill

Total (Rs.) : 37

Checkout

	Item	Price	Quantity	Amount (Rs.)
	Morphine	2	4	8
	Ketamine	4	2	8
	Oxycotin	1	1	1
▶	Paracetamol	2	10	20

## Bill Summary

Print preview

Close

Page 1

Mediplus

GRAN Pharma, GST :GSTN899745

14-03-22 | 08:24

#	Item	Price	Qty	Amount
1	Morphine	2	2	2
2	Ketamine	4	1	1
3	Oxycotin	1	3	3
4	Paracetamol	2	7	7

Total (Rs.) : 25

Thank you, have a nice day

# **REPORTS**

## Users Report

Id	Name	Username	User Type
1	James Gordon	jamesgordon	Manager
2	Rabecca	rabecca	Pharmacist
3	Jose carter	josecarter	Pharmacist

## Medicines Report

Id	Name	Formula	Company	Price
1	Morphine	C17H19NO3	Altec	2
2	Ketamine	C13H16CINO	Cipla	4
3	Oxycotin	C43H66N12O1...	SUN Pharma	1
4	Paracetamol	C8H9NO2	Cipla	2

# Inventory Report

Id	name	Stock	
1	Morphine	28	
2	Ketamine	37	
3	Oxycotin	182	
4	Paracetamol	30	