



CoPOS

# Coffee Point Of Sales

JANUARY 15

**RUDY SARMIENTO VARGAS**  
**Michael Dumala**

---

# CoPOS [Coffee Point of Sales]

## “What is Coffee?”

**Coffee** is a brewed drink prepared from roasted coffee beans, the seeds of berries from certain *Coffea* species. When coffee berries turn from green to bright red in color – indicating ripeness – they are picked, processed, and dried. Dried coffee seeds (referred to as "beans") are roasted to varying degrees, depending on the desired flavor. Roasted beans are ground and then brewed with near-boiling water to produce the beverage known as coffee.

## “What is POS?”

**Point of sale systems** are systems that enable the business transaction between the client and the company to be completed. POS system is a computerized network that consists of the main computer linked with several checkout terminals and supported by different hardware features starting from barcode scanners and ending with card payment terminals.

# System Content

## ABSTRACT

CoPOS is a simple system where it can provide easier capturing of data sales and inventory of a shop with data security which records the transaction of a certain user.

## STATEMENT OF PROBLEM

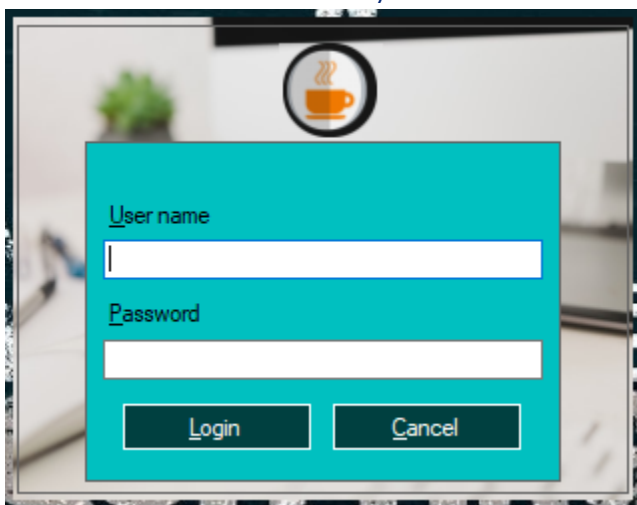
We are aware that system is very important for every business we have in the town but because of most people is excited to their business get in operation, most of them is using notebooks and pen, a paper with tabular form which they can write their daily sales or stocks delivered and even the stock monitoring. We are not saying that these method is not effective because we intent to give service but as the business grow these kind of method is getting too messy to our table which requires a good arrangement of record, good eye sight for every transaction made, bunch of writing to compute revenue, in short this will consume time to our daily basis.

## OBJECTIVE

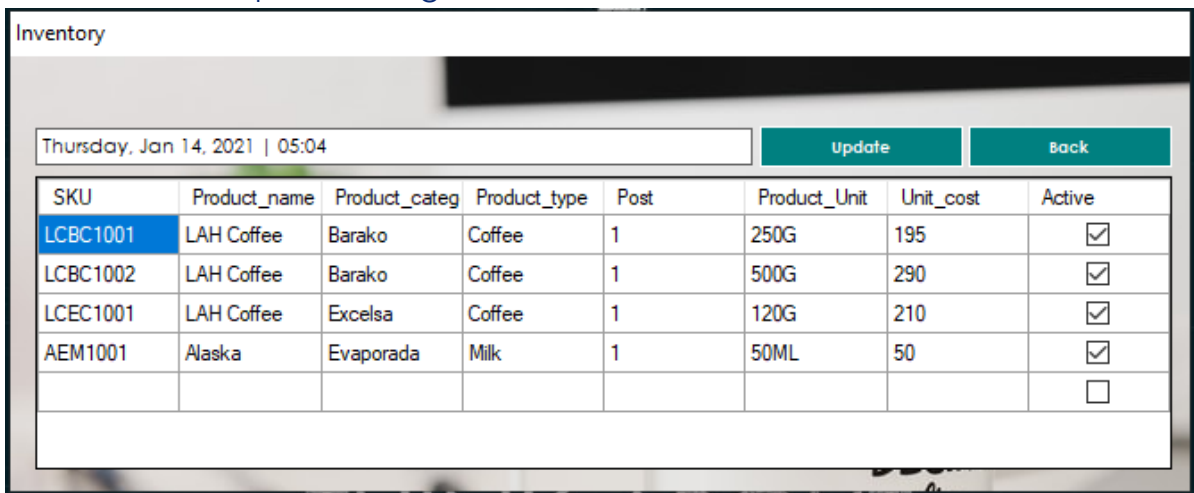
The objective of CoPOS is to store large number of sales and inventory in long term purposes with data security that captures each transaction made by the user. User can do transaction for each customer in a few click that includes transaction recording.

## Input Design

- a. Login Form –Using assign credential provided by the management in order to use the features of the system.

A screenshot of a login form titled 'CoPOS' with a coffee cup icon. The form has a teal background and contains two input fields: 'User name' and 'Password'. Below the fields are two buttons: 'Login' and 'Cancel'.

- b. Item Masterfile – a section of system where as the accountable person for the service and product register.

A screenshot of the 'Inventory' section of the CoPOS system. It shows a table with columns: SKU, Product\_name, Product\_categ, Product\_type, Post, Product\_Unit, Unit\_cost, and Active. The table contains four rows of data, with the first row highlighted in blue. Above the table is a date and time display: 'Thursday, Jan 14, 2021 | 05:04'. To the right of the date are two buttons: 'Update' and 'Back'.

- c. Received Delivery – the receiving team/person for the stock goes here where they can input the type, name, category, unit, quantity and amount.

**Purchase Order**

Thursday, Jan 14, 2021 | 05:08

**received** **Back**

Type: Coffee Unit: 120G

Name: LAH Coffee Quantity: 3

Category: Excelsa Amount: 450

Product Type	Product Name	Product Category	Unit	Qty	Amount
Coffee	LAH Coffee	Barako	500G	5	1,000.00

Michael 5 1,000.00

- d. Sales Order – the counter assign will fill out these 2 entry to add transaction and will automate to display the product and its cost.

**Sales Order**

Thursday, Jan 14, 2021 | 05:18

Michael **New Transaction**

Item Code: lcbe1002 @ 1

Item Code	Item Name	Quantity
LCBC1001	LAH Coffee-Barako@250G	2

Total Quantity: 2

Total Amount: 390.00

Payment Terms: -

**CASH** **CARD**

VAT Sales: -

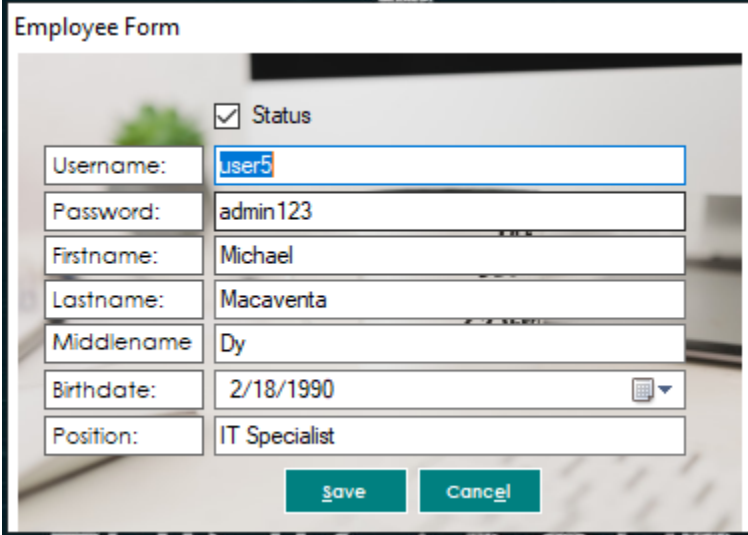
VAT: -

Total Amount: 390.00

Amount Paid: -

Amount Change: -

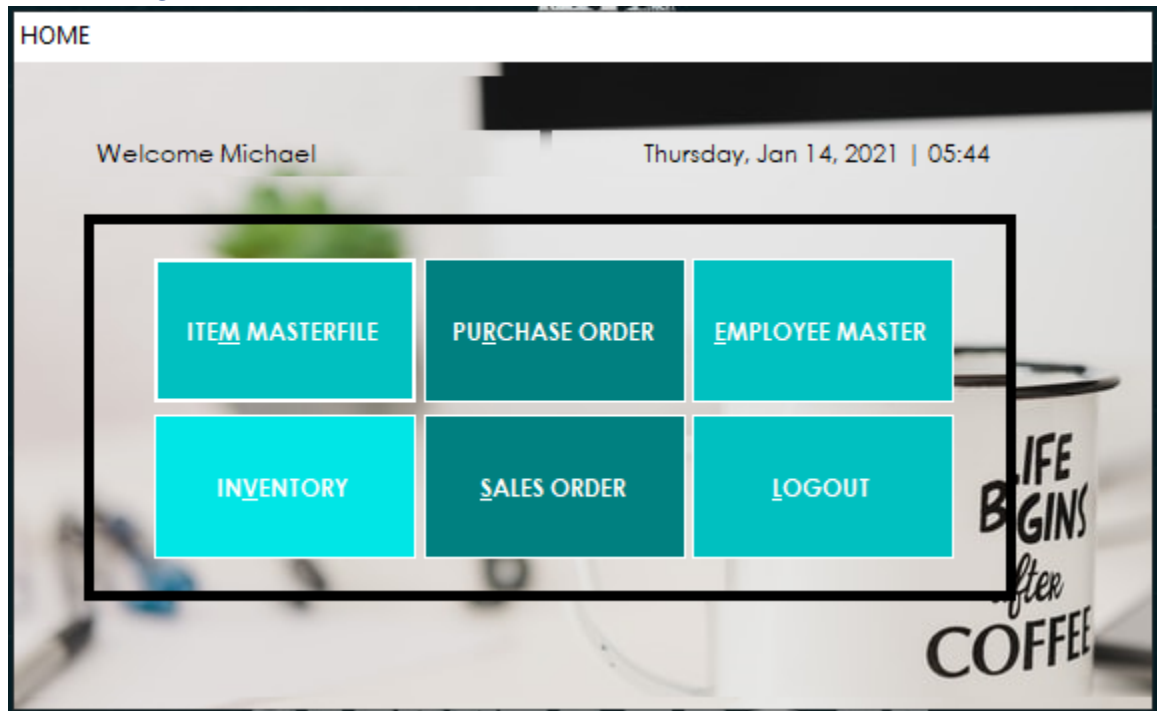
- e. Employee Masterfile – the part of the program where we can add and update information of the employee.



A screenshot of a web form titled "Employee Form". The form contains several input fields: "Username:" with the value "user5", "Password:" with "admin123", "Firstname:" with "Michael", "Lastname:" with "Macaventa", "Middlename" with "Dy", "Birthdate:" with "2/18/1990" and a calendar icon, and "Position:" with "IT Specialist". There is a checked checkbox labeled "Status" above the Username field. At the bottom of the form are two buttons: "Save" and "Cancel".

## Output Design

- a. Home Page – the main menu of the system CoPOS.



- b. Warehouse – the page where can view the available stocks and sold products.

Warehouse

Thursday, Jan 14, 2021 | 05:52 Michael [Back](#)

Item Code	Product Name	Product Category	Product Type	Location	Unit	Total Stock	Total Sales
AEM1001	Alaska	Evaporada	Milk	1	50ML	15	
LCBC1001	LAH Coffee	Barako	Coffee	1	250G	48	36
LCBC1002	LAH Coffee	Barako	Coffee	1	500G	15	12
LCEC1001	LAH Coffee	Excelsa	Coffee	1	120G	3	

- c. Employee Masterfile – the whole list registered employee to shop or company.

Employee Masterfile

-  [Filter](#) [Back](#) [Add](#)

Username	Lastname	Firstname	Position	
admin	Dumala	Michael	Admin	<a href="#">Edit</a>
user1	Nalid	Mic	Accounting	<a href="#">Edit</a>
user2	Nadulas	Nic	Rider	<a href="#">Edit</a>
user3	Lang	John	Admin	<a href="#">Edit</a>
user4	Makazilong	Natan	Admin	<a href="#">Edit</a>
user5	Macaventa	Michael	IT Specialist	<a href="#">Edit</a>
user6	Cayo	John	Agent	<a href="#">Edit</a>
Mic123	Macafasok	John	IT Specialist	<a href="#">Edit</a>
admin				<a href="#">Edit</a>
user7	lastname123	user123	IT Specialist	<a href="#">Edit</a>

- d. Received Delivery – the receiving team/person for the stock goes here where they can view the type, name, category, unit, quantity and amount.

**Purchase Order**

Thursday, Jan 14, 2021 | 05:08

**received** **back**

Type: Coffee Unit: 120G

Name: LAH Coffee Quantity: 3

Category: Excelsa Amount: 450

Product Type	Product Name	Product Category	Unit	Qty	Amount
Coffee	LAH Coffee	Barako	500G	5	1,000.00

Michael 5 1,000.00

- e. Sales Order – the counter and customer will see the item, quantity, total quantity, price per item, total price, payment term made and change amount for cash transaction.

**Sales Order**

Thursday, Jan 14, 2021 | 05:18

Michael **New Transaction** **back**

Item Code: lcbe1002 @ 1

Item Code	Item Name	Quantity	Amount
LCBC1001	LAH Coffee-Barako@250G	2	195

Total Quantity: 2

Total Amount: 390.00

Payment Terms: -

**CASH** **CARD**

VAT Sales: -

VAT: -

Total Amount: 390.00

Amount Paid: -

Amount Change: -



# Program Code

## a. LOGIN FORM

```
Imports System.Data.OleDb
Imports System.Data
Public Class LoginForm1
    Dim connString As String = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\Users\Michael\My Files\BSCS\3RD SEM\COMP 3\COMP 3\CS203.2020db.mdb;Persist Security
Info=False"
    Dim conn As New OleDbConnection(connString)
    Private Sub Cancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Cancel.Click
        Me.Close()
    End Sub
    Public Sub checkConn()
        If conn.State = ConnectionState.Open Then
            conn.Close()
        End If
        conn.Open()
    End Sub
    Public Sub getUser()
        Dim dsUser As New DataSet
        Dim dtcUser As DataTableCollection
        Dim daUser As New OleDbDataAdapter
        Dim cmdbUser As New OleDbCommandBuilder

        conn = New OleDbConnection
        conn.ConnectionString = connString
        dsUser = New DataSet
        dtcUser = dsUser.Tables
        checkConn()
        daUser = New OleDbDataAdapter("Select * from tblUser where Username='" &
txtUsername.Text & "' and Password='" & txtPassword.Text & "' and `Active`=-1", conn)
        cmdbUser = New OleDbCommandBuilder(daUser)
        cmdbUser.QuotePrefix = "["
        cmdbUser.QuoteSuffix = "]"
        daUser.Fill(dsUser, "tblUser")
        Dim dvUser As New DataView(dtcUser(0))
        tmpUser.dgvTest.DataSource = dvUser
    End Sub
    Public Sub checkUser()
        checkConn()
        Dim cmd As New OleDbCommand("Select count(*) from tblUser where Username=? and
Password=? and active=-1", conn)
        cmd.Parameters.AddWithValue("@1", OleDbType.VarChar).Value = txtUsername.Text
        cmd.Parameters.AddWithValue("@2", OleDbType.VarChar).Value = txtPassword.Text
        Dim cnt = Convert.ToInt32(cmd.ExecuteScalar())
        If cnt > 0 Then
            getUser()
            Me.Visible = False
            frmHome.Show()
        Else
            MsgBox("Invalid Credential", vbInformation, "Incorrect")
        End If
    End Sub
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        getUser()
    End Sub
End Sub
```



```

Private Sub Button1_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
    Try
        If txtUsername.Text = Nothing Or txtPassword.Text = Nothing Then
            MsgBox("Please use your assign credential", vbInformation, "NO INPUT")
        Else
            checkUser()
        End If
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    End Try
    checkConn()
End Sub
End Class

```

## b. HOME FORM

```

Public Class frmHome
    Dim strName As String
    Dim strUserid As String
    Private Sub btnLogout_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnLogout.Click
        Application.Exit()
    End Sub
    Private Sub btnInventory_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnInventory.Click
        Me.Visible = False
        frmInventory.Show()
    End Sub

    Private Sub btnEmp_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnEmp.Click
        Me.Close()
        frmEmpfile.Show()
    End Sub
    Private Sub frmHome_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        lblDate.Text = Format(Now(), "dddd, MMM dd, yyyy | hh:mm ")
        strName = tmpUser.dgvTest.Rows(0).Cells(3).Value.ToString()
        strUserid = tmpUser.dgvTest.Rows(0).Cells(0).Value.ToString()

        lblHome.Text = "Welcome " & strName
    End Sub
    Private Sub btnPurchase_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnPurchase.Click
        Me.Close()
        frmPurchase.Show()
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        Me.Close()
        frmSales.Show()
    End Sub
    Private Sub btnOrder_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnOrder.Click
        Me.Close()
        frmWarehouse.Show()
    End Sub
End Class

```

### c. ITEM MASTERFILE

```
Imports System.Data.OleDb
Imports System.Data
Public Class frmInventory
    Dim connString As String = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\Users\Michael\My Files\BSCS\3RD SEM\COMP 3\COMP 3\CS203.2020db.mdb;Persist Security
Info=False"
    Dim conn As New OleDbConnection(connString)
    Dim dbAdapter As OleDbDataAdapter
    Dim dbDataSet As DataSet
    Dim dbTable As DataTableCollection
    Dim cmdBuilder As New OleDbCommandBuilder

    Private Sub frmInventory_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        Try
            conn = New OleDbConnection
            conn.ConnectionString = connString
            dbDataSet = New DataSet
            dbTable = dbDataSet.Tables
            checkConn()
            dbAdapter = New OleDbDataAdapter("Select * from tblInventory", conn)
            cmdBuilder = New OleDbCommandBuilder(dbAdapter)
            cmdBuilder.QuotePrefix = "["
            cmdBuilder.QuoteSuffix = "]"
            dbAdapter.Fill(dbDataSet, "tblInventory")
            Dim dvTable As New DataView(dbTable(0))
            dgvInventory.DataSource = dvTable
            dgvInventory.RowHeadersVisible = False
            dgvInventory.Columns(0).Visible = False
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try
        lblDate.Text = Format(Now(), "dddd, MMM dd, yyyy | hh:mm ")
    End Sub
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        Try
            dbAdapter.Update(dbTable(0))
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try
    End Sub
    Public Sub checkConn()
        If conn.State = ConnectionState.Open Then
            conn.Close()
        End If
        conn.Open()
    End Sub
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
        Me.Close()
        frmHome.Show()
    End Sub
End Class
```

#### d. WAREHOUSE FORM

```
Imports System.Data
Imports System.Data.OleDb
Public Class frmWarehouse
    Dim connString As String = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\Users\Michael\My Files\BSCS\3RD SEM\COMP 3\COMP 3\CS203.2020db.mdb;Persist Security
Info=False"
    Dim conn As New OleDbConnection(connString)
    Dim strUsername As String
    Private Sub Warehouse_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        strUsername = tmpUser.dgvTest.Rows(0).Cells(3).Value.ToString()
        lblUsername.Text = strUsername
        lblDate.Text = Format(Now(), "dddd, MMM dd, yyyy | hh:mm ")
        selInventoryStock()
    End Sub
    Private Sub selInventoryStock()
        Try
            conn = New OleDbConnection
            conn.ConnectionString = connString
            Dim dsWarehouse As New DataSet
            Dim dtWarehouse As DataTableCollection
            dtWarehouse = dsWarehouse.Tables
            checkConn()
            Dim daWarehouse = New OleDbDataAdapter("Select * from qryWarehouse", conn)
            Dim cmdbWarehouse = New OleDbCommandBuilder(daWarehouse)
            cmdbWarehouse.QuotePrefix = "["
            cmdbWarehouse.QuoteSuffix = "]"
            daWarehouse.Fill(dsWarehouse, "qryWarehouse")
            Dim dvWarehouse = New DataView(dtWarehouse(0))
            dgvInventoryView.DataSource = dvWarehouse
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try
    End Sub
    Private Sub checkConn()
        If conn.State = ConnectionState.Open Then
            conn.Close()
        End If
        conn.Close()
    End Sub
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        Me.Close()
        frmHome.Show()
    End Sub
End Class
```

#### e. RECEIVED STOCK FORM

```
Imports System.Data.OleDb
Imports System.Data
Public Class frmPurchase
    Dim connString As String = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\Users\Michael\My Files\BSCS\3RD SEM\COMP 3\COMP 3\CS203.2020db.mdb;Persist Security
Info=False"
    Dim conn As New OleDbConnection(connString)
    Dim strUserId, strUsername As String
    Dim dvProdtype As DataView
    Dim totalAmnt, totalQty, lastid As Integer
```

```

Dim cbbox As Boolean

Private Sub Purchase_Order_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    strUsername = tmpUser.dgvTest.Rows(0).Cells(3).Value.ToString()
    lblUsername.Text = strUsername
    strUserid = tmpUser.dgvTest.Rows(0).Cells(0).Value.ToString()
    lblDate.Text = Format(Now(), "dddd, MMM dd, yyyy | hh:mm ")
    selProduct("Select * from qryProdtype", "qryProdtype", "Product_type", cbProdType)
    lblPlus.Image = Image.FromFile("C:\Users\Michael\My Files\BSCS\3RD SEM\COMP 3\COMP
3\mydb\image\add-icon.png")
    lblMinus.Image = Image.FromFile("C:\Users\Michael\My Files\BSCS\3RD SEM\COMP 3\COMP
3\mydb\image\0008-Minus-icon.png")
End Sub
Public Sub insertTrans()
    Try
        Dim cmdInsTrans As New OleDbCommand
        Dim cmdStrTrans = "Insert into
tblTransaction(`Transaction_type`,`Qty`,`Amount`,`Userid`,`Updatedate`,`AmountPaid`)
values(@type,@qty,@amnt,@userid,@updatedate,@paid)"

        conn = New OleDbConnection
        conn.ConnectionString = connString
        cmdInsTrans.Connection = conn
        checkConn()
        cmdInsTrans.CommandText = cmdStrTrans
        cmdInsTrans.Parameters.AddWithValue("@type", "Purchase")
        cmdInsTrans.Parameters.AddWithValue("@qty", lblTotalQty.Text)
        cmdInsTrans.Parameters.AddWithValue("@amnt", lblTotalAmnt.Text)
        cmdInsTrans.Parameters.AddWithValue("@userid", Convert.ToInt32(strUserid))
        cmdInsTrans.Parameters.AddWithValue("@updatedate",
Convert.ToDateTime(Format(Now(), "MM/dd/yyyy h:mm:ss tt")))
        cmdInsTrans.Parameters.AddWithValue("@paid", lblTotalAmnt.Text)
        cmdInsTrans.ExecuteNonQuery()
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    End Try
End Sub
Private Sub selLastTransid()
    Try
        Dim cmdSelectLastTrans As New OleDbCommand
        Dim cmdStrSelectLastTrans As String = "Select max(`Trans_id`) from
tblTransaction"

        conn = New OleDbConnection
        conn.ConnectionString = connString
        cmdSelectLastTrans.Connection = conn
        checkConn()
        cmdSelectLastTrans.CommandText = cmdStrSelectLastTrans
        lastid = CInt(cmdSelectLastTrans.ExecuteScalar())
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    End Try
End Sub
Private Sub insPurchaseTrans()
    Try
        Dim cmdInsTrans As New OleDbCommand
        Dim cmdStrTrans = "Insert into
tblPurchaseTransaction(`Product_name`,`Product_category`,`Product_type`,`Unit`,`Quantity`,`It

```

```

em_amount`, `Userid`, `Transaction_id`, `Updatedate`)
values(@name,@category,@type,@weight,@quantity,@subtotal,@userid,@transid,@date)"
    conn = New OleDbConnection
    conn.ConnectionString = connString
    checkConn()
    Dim cnt As Integer
    For cnt = 0 To dgvAdd.Rows.Count - 1
        cmdInsTrans = New OleDbCommand(cmdStrTrans, conn)
        cmdInsTrans.Parameters.AddWithValue("@name", dgvAdd.Rows(cnt).Cells(1).Value)
        cmdInsTrans.Parameters.AddWithValue("@category",
dgvAdd.Rows(cnt).Cells(2).Value)
        cmdInsTrans.Parameters.AddWithValue("@type", dgvAdd.Rows(cnt).Cells(0).Value)
        cmdInsTrans.Parameters.AddWithValue("@weight",
dgvAdd.Rows(cnt).Cells(3).Value)
        cmdInsTrans.Parameters.AddWithValue("@quantity",
dgvAdd.Rows(cnt).Cells(4).Value)
        cmdInsTrans.Parameters.AddWithValue("@subtotal",
dgvAdd.Rows(cnt).Cells(5).Value)
        cmdInsTrans.Parameters.AddWithValue("@userid", Convert.ToInt32(strUserid))
        cmdInsTrans.Parameters.AddWithValue("@transid", Convert.ToInt32(lastid + 1))
        cmdInsTrans.Parameters.AddWithValue("@date", Convert.ToDateTime(Format(Now(),
"MM/dd/yyyy h:mm:ss tt"))))
        cmdInsTrans.ExecuteNonQuery()
    Next
    MsgBox("You successfully add records of: " & cnt, , "Affected Rows")
Catch ex As Exception
    MessageBox.Show(ex.Message)
End Try
End Sub
Public Sub checkConn()
    If conn.State = ConnectionState.Open Then
        conn.Close()
    End If
    conn.Open()
End Sub
Public Sub selProduct(ByVal cmdQuery As String, ByVal dtTable As String, ByVal hField As
String, ByVal cbBox As ComboBox)
    selProd(cmdQuery, dtTable)
    Dim cnt As Integer
    Dim cbProd As New ComboBox
    cbProd = cbBox
    For cnt = 0 To dvProdtype.Count - 1
        cbProd.Items.Add(dvProdtype(cnt).Item(hField))
    Next
End Sub
Public Sub selProd(ByVal strCmd As String, ByVal strTable As String)
    conn = New OleDbConnection
    conn.ConnectionString = connString
    Dim dsProdtype = New DataSet
    Dim dtProdtype As DataTableCollection
    dtProdtype = dsProdtype.Tables
    checkConn()
    Dim daProdtype = New OleDbDataAdapter(strCmd, conn)
    Dim cmdbProdtype = New OleDbCommandBuilder(daProdtype)
    cmdbProdtype.QuotePrefix = "["
    cmdbProdtype.QuoteSuffix = "]"
    daProdtype.Fill(dsProdtype, strTable)
    dvProdtype = New DataView(dtProdtype(0))
End Sub

```

```

Private Sub btnBack_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnBack.Click
    Me.Close()
    frmHome.Show()
End Sub
Private Sub cbProdName_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cbProdName.SelectedValueChanged
    Dim cmdqry, dbtable, dbfield As String
    cmdqry = "select * from qryProdcategory where `Product_name`='" &
cbProdName.SelectedItem & "'"
    dbtable = "qryProdcategory"
    dbfield = "Product_category"
    cbProdCat.Text = ""
    cbUnit.Text = ""
    cbProdCat.Items.Clear()
    selProduct(cmdqry, dbtable, dbfield, cbProdCat)
End Sub
Private Sub cbProdType_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cbProdType.SelectedValueChanged
    Dim cmdqry, dbtable, dbfield As String
    cmdqry = "select * from qryProdname where `Product_type`='" & cbProdType.SelectedItem
& "'"
    dbtable = "qryProdname"
    dbfield = "Product_name"
    cbProdName.Text = ""
    cbProdCat.Text = ""
    cbUnit.Text = ""
    cbProdName.Items.Clear()
    selProduct(cmdqry, dbtable, dbfield, cbProdName)
End Sub
Private Sub addPurchase()
    Try
        With dgvAdd
            .Rows.Add(
                cbProdType.SelectedItem,
                cbProdName.SelectedItem,
                cbProdCat.SelectedItem,
                cbUnit.SelectedItem,
                txtQty.Text,
                Format(Val(txtAmnt.Text), "#,###.00"))
        End With
        totalQty = totalQty + txtQty.Text
        totalAmnt = totalAmnt + txtAmnt.Text
        lblTotalQty.Text = totalQty
        lblTotalAmnt.Text = Format(Val(totalAmnt), "#,###.00")
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    End Try
End Sub
Private Sub delPurchase()
    Try
        Dim row As DataGridViewRow
        totalQty = totalQty - dgvAdd.CurrentRow.Cells(4).Value
        totalAmnt = totalAmnt - dgvAdd.CurrentRow.Cells(5).Value
        For Each row In dgvAdd.SelectedRows
            dgvAdd.Rows.Remove(row)
        Next
        lblTotalQty.Text = totalQty
        lblTotalAmnt.Text = Format(Val(totalAmnt), "#,###.00")
    End Try
End Sub

```

```

        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try
    End Sub
    Private Sub checkInput()
        If cbProdType.SelectedIndex >= 0 And cbProdName.SelectedIndex >= 0 And
cbProdCat.SelectedIndex >= 0 And cbUnit.SelectedIndex >= 0 And Val(txtQty.Text) > 0 And
Val(txtAmnt.Text) > 0 Then
            cbbox = True
        Else
            cbbox = False
        End If
    End Sub
    Private Sub clearContent()
        For Each cb In Me.Controls
            If TypeOf cb Is ComboBox Then
                cb.text = ""
                cb.Items.Clear()
            End If
        Next
        For Each txt In Me.Controls
            If TypeOf txt Is TextBox Then
                txt.Text = ""
            End If
        Next
        selProduct("Select * from qryProdtype", "qryProdtype", "Product_type", cbProdType)
    End Sub

    Private Sub btnRec_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnRec.Click
        If dgvAdd.Rows.Count > 0 Then
            insertTrans()
            sellLastTransid()
            insPurchaseTrans()
        Else
            MsgBox("No data record", , "No Record")
        End If
    End Sub
    Private Sub lblPlus_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles lblPlus.Click
        checkInput()
        If cbbox = True Then
            addPurchase()
            clearContent()
            cbbox = False
        Else
            MsgBox("There's an empty field.Please fillout. Thank you", , "Blank Field")
        End If
    End Sub
    Private Sub lblMinus_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles lblMinus.Click
        delPurchase()
    End Sub
    Private Sub cbProdCat_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cbProdCat.SelectedValueChanged
        Dim cmdqry, dbtable, dbfield As String
        cmdqry = "select * from qryProdunit where `Product_category`='" &
cbProdCat.SelectedItem & "'"
        dbtable = "qryProdunit"

```



```

        dbfield = "Product_Unit"
        cbUnit.Text = ""
        cbUnit.Items.Clear()
        selProduct(cmdqry, dbtable, dbfield, cbUnit)
    End Sub
End Class

```

## f. SALES ORDER FORM

```

Imports System.Data
Imports System.Data.OleDb
Public Class frmSales
    Dim connString As String = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\Users\Michael\My Files\BSCS\3RD SEM\COMP 3\COMP 3\CS203.2020db.mdb;Persist Security
Info=False"
    Dim conn As New OleDbConnection(connString)
    Dim strUserId, strUsername As String
    Dim totalAmt, totalQty, lastid As Integer
    Dim dvSelItem As DataView
    Dim boolSet As Boolean

    Private Sub frmSales_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        strUsername = tmpUser.dgvTest.Rows(0).Cells(3).Value.ToString()
        lblUsername.Text = strUsername
        strUserId = tmpUser.dgvTest.Rows(0).Cells(0).Value.ToString()
        lblDate.Text = Format(Now(), "dddd, MMM dd, yyyy | hh:mm ")
    End Sub
    Private Sub checkConn()
        If conn.State = ConnectionState.Open Then
            conn.Close()
        End If
        conn.Open()
    End Sub
    Private Sub insertOrderTransaction()
        Try
            Dim cmdInsTrans As New OleDbCommand
            Dim cmdStrCode = "Insert into
tblSalesOrderTransaction(`ItemCode`,`ItemName`,`Quantity`,`Amount`,`Terms`,`UserId`,`Transa
ction_id`,`Updatedate`) values (@code,@name,@qty,@amnt,@terms,@id,@transid,@date)"

            conn = New OleDbConnection
            conn.ConnectionString = connString
            checkConn()
            Dim cnt As Integer
            For cnt = 0 To dgvAdd.Rows.Count - 1
                cmdInsTrans = New OleDbCommand(cmdStrCode, conn)
                cmdInsTrans.Parameters.AddWithValue("@code", dgvAdd.Rows(cnt).Cells(0).Value)
                cmdInsTrans.Parameters.AddWithValue("@name", dgvAdd.Rows(cnt).Cells(1).Value)
                cmdInsTrans.Parameters.AddWithValue("@qty", dgvAdd.Rows(cnt).Cells(2).Value)
                cmdInsTrans.Parameters.AddWithValue("@amnt", dgvAdd.Rows(cnt).Cells(3).Value)
                cmdInsTrans.Parameters.AddWithValue("@terms", lblTerms.Text)
                cmdInsTrans.Parameters.AddWithValue("@id", Convert.ToInt32(strUserId))
                cmdInsTrans.Parameters.AddWithValue("@transid", Convert.ToInt32(lastid + 1))
                cmdInsTrans.Parameters.AddWithValue("@date", Convert.ToDateTime(Format(Now(),
"MM/dd/yyyy h:mm:ss tt")))
                cmdInsTrans.ExecuteNonQuery()
            Next
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try
    End Sub
End Class

```

```

End Try
End Sub
Private Sub insertTrans()
Try
Dim cmdInsTrans As New OleDbCommand
Dim cmdStrTrans = "Insert into
tblTransaction(`Transaction_type`,`Qty`,`Amount`,`Userid`,`Updatedate`,`AmountPaid`)
values(@type,@qty,@amnt,@userid,@updatedate,@paid)"

conn = New OleDbConnection
conn.ConnectionString = connString
cmdInsTrans.Connection = conn
checkConn()
cmdInsTrans.CommandText = cmdStrTrans
cmdInsTrans.Parameters.AddWithValue("@type", "Salesord")
cmdInsTrans.Parameters.AddWithValue("@qty", lblQty.Text)
cmdInsTrans.Parameters.AddWithValue("@amnt", lblAmnt.Text)
cmdInsTrans.Parameters.AddWithValue("@userid", Convert.ToInt32(strUserId))
cmdInsTrans.Parameters.AddWithValue("@updatedate",
Convert.ToDateTime(Format(Now(), "MM/dd/yyyy h:mm:ss tt")))
cmdInsTrans.Parameters.AddWithValue("@paid", lblPaid.Text)
cmdInsTrans.ExecuteNonQuery()
Catch ex As Exception
MessageBox.Show(ex.Message)
End Try
End Sub
Private Sub sellLastTransid()
Try
Dim cmdSelectLastTrans As New OleDbCommand
Dim cmdStrSelectLastTrans As String = "Select max(`Trans_id`) from
tblTransaction"

conn = New OleDbConnection
conn.ConnectionString = connString
cmdSelectLastTrans.Connection = conn
checkConn()
cmdSelectLastTrans.CommandText = cmdStrSelectLastTrans
lastid = CInt(cmdSelectLastTrans.ExecuteScalar())
Catch ex As Exception
MessageBox.Show(ex.Message)
End Try
End Sub
Private Sub checkItem()
checkConn()
Dim cmd As New OleDbCommand("Select count(*) from selItem where `SKU`=?", conn)
cmd.Parameters.AddWithValue("@1", OleDbType.VarChar).Value = txtCode.Text
Dim cnt = Convert.ToInt32(cmd.ExecuteScalar())
If cnt > 0 Then
selItem()
Else
MsgBox("No Item Found. Please try again", vbInformation, "NO ITEM")
End If
End Sub
Private Sub selItem()
If txtQty.Text = Nothing Or Val(txtQty.Text) = 0 Then
MsgBox("Invalid Quantity Input.", vbInformation, "Invalid Quantity")
Else
Try
conn = New OleDbConnection

```

```

        conn.ConnectionString = connString
        Dim dsItem As New DataSet
        Dim dcItem As DataTableCollection
        dcItem = dsItem.Tables
        checkConn()
        Dim daSelItem = New OleDbDataAdapter("Select * from selItem where `SKU`='" &
txtCode.Text & "'", conn)
        Dim cmdbSelItem = New OleDbCommandBuilder(daSelItem)
        cmdbSelItem.QuotePrefix = "["
        cmdbSelItem.QuoteSuffix = "]"
        daSelItem.Fill(dsItem, "selItem")
        dvSelItem = New DataView(dcItem(0))
        dgvAdd.Rows.Add(dvSelItem(0).Item("SKU"), dvSelItem(0).Item("item"),
txtQty.Text, dvSelItem(0).Item("Unit_cost"))
        getSalesord()
        txtCode.Clear()
        txtQty.Clear()
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    End Try
End If
End Sub
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
    Me.Close()
    frmHome.Show()
End Sub
Private Sub lblAdd_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles lblAdd.Click
    checkItem()
End Sub
Private Sub getSalesord()
    totalQty = totalQty + txtQty.Text
    lblQty.Text = totalQty
    totalAmnt = totalAmnt + (dvSelItem(0).Item("Unit_cost") * txtQty.Text)
    lblAmnt.Text = Format(Val(totalAmnt), "#,###.00")
    lblAmnt2.Text = Format(Val(totalAmnt), "#,###.00")
End Sub
Private Sub getVat()
    lblVat.Text = Format(Val(totalAmnt * 0.12), "#,###.00")
    lblVatsales.Text = Format(totalAmnt - Val(lblVat.Text), "#,###.00")
End Sub
Private Sub Label6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles lblCash.Click
    If dgvAdd.Rows.Count > 0 Then
        If boolSet = False Then
            lblTerms.Text = "CASH"
            Dim intCash As Integer = InputBox("Cash Amount Paid: ", "Cash Payment")
            If intCash >= totalAmnt Then
                getVat()
                lblPaid.Text = Format(intCash, "#,###.00")
                lblChange.Text = Format(intCash - totalAmnt, "#,##0.00")
                boolSet = True
            Else
                MsgBox("Insufficient Amount Paid. Please re-enter Payment amount.",
vbInformation, "Insufficient Amount")
            End If
        Else

```

```

        MsgBox("Please get new Transaction. Record is already set", vbInformation,
"Already Exist")
    End If
Else
    MsgBox("No Transaction done", vbInformation, "No Transaction")
End If
End Sub
Private Sub Label7_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles lblCard.Click
    If dgvAdd.Rows.Count > 0 Then
        If boolSet = False Then
            lblTerms.Text = "CARD"
            getVat()
            lblPaid.Text = Format(totalAmnt, "#,###.00")
            lblChange.Text = Format(totalAmnt - totalAmnt, "#,##0.00")
            boolSet = True
        Else
            MsgBox("Please get new Transaction. Record is already set", vbInformation,
"Already Exist")
        End If
    Else
        MsgBox("No Transaction done", vbInformation, "No Transaction")
    End If
End Sub
Private Sub delSalesord()
    Try
        Dim row As DataGridViewRow
        totalAmnt = totalAmnt - (dgvAdd.CurrentRow.Cells(3).Value *
dgvAdd.CurrentRow.Cells(2).Value)
        totalQty = totalQty - dgvAdd.CurrentRow.Cells(2).Value
        For Each row In dgvAdd.SelectedRows
            dgvAdd.Rows.Remove(row)
        Next
        lblQty.Text = totalQty
        lblAmnt.Text = Format(Val(totalAmnt), "#,###.00")
        lblAmnt2.Text = Format(Val(totalAmnt), "#,###.00")
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    End Try
End Sub
Private Sub Label6_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Label6.Click
    delSalesord()
End Sub
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
    If Val(lblPaid.Text) > 0 Then
        insertTrans()
        sellLastTransid()
        insertOrderTransaction()
        lblQty.Text = "-"
        lblAmnt.Text = "-"
        lblTerms.Text = "-"
        lblVat.Text = "-"
        lblVatsales.Text = "-"
        lblAmnt2.Text = "-"
        lblPaid.Text = "-"
        lblChange.Text = "-"
        totalAmnt = 0
    End If
End Sub

```

```

        totalQty = 0
        dgvAdd.Rows.Clear()
        dgvAdd.Refresh()
        boolSet = False
    Else
        MsgBox("Transaction not done. ", vbInformation, "No Data")
    End If
End Sub
End Class

```

## g. EMPLOYEE MASTERFILE FORM

```

Imports System.Data.OleDb
Imports System.Data
Public Class frmEmpfile
    Dim connString As String = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\Users\Michael\My Files\BSCS\3RD SEM\COMP 3\COMP 3\CS203.2020db.mdb;Persist Security
Info=False"
    Dim conn As New OleDbConnection(connString)
    Dim dbAdapter As OleDbDataAdapter
    Dim dbDataSet As DataSet
    Dim dbTable As DataTableCollection
    Dim cmdBuilder As New OleDbCommandBuilder
    Dim strCmd As String

    Private Sub frmEmpfile_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        strCmd = "Select * from qryEmployeeList"
        tdLoad()
        ' List box Option
        cbFilter.Items.Add("Username")
        cbFilter.Items.Add("Lastname")
        cbFilter.Items.Add("Firstname")
        cbFilter.Items.Add("Position")
    End Sub
    Public Sub checkConn()
        If conn.State = ConnectionState.Open Then
            conn.Close()
        End If
        conn.Open()
    End Sub
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
        If Len(txtFilter.Text) > 0 And cbFilter.SelectedItem <> "-" Then
            strCmd = "Select * from qryEmployeeList where `" & cbFilter.SelectedItem & "`
like '%" & txtFilter.Text & "%'"
            tdLoad()
        Else
            strCmd = "Select * from qryEmployeeList"
            tdLoad()
        End If
    End Sub
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        frmHome.Show()
        Me.Close()
    End Sub
    Public Sub tdLoad()
        Try
            conn = New OleDbConnection

```

```

        conn.ConnectionString = connString
        dbDataSet = New DataSet
        dbTable = dbDataSet.Tables
        checkConn()
        dbAdapter = New OleDbDataAdapter(strCmd, conn)
        cmdBuilder = New OleDbCommandBuilder(dbAdapter)
        cmdBuilder.QuotePrefix = "["
        cmdBuilder.QuoteSuffix = "]"
        dbAdapter.Fill(dbDataSet, "qryEmployeeList")
        Dim dvTable As New DataView(dbTable(0))
        dgvEmpList.DataSource = dvTable
        addButton()
        dgvEmpList.RowHeadersVisible = False
        dgvEmpList.AllowUserToAddRows = False
        For cnt = 0 To 3
            dgvEmpList.Columns(cnt).ReadOnly = True
        Next
        dgvEmpList.ClearSelection()
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    End Try
End Sub
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
    frmEmpForm.Show()
    Me.Close()
    frmEmpForm.fncAddData()
End Sub
Private Sub dgvEmpList_CellContentClick(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs) Handles dgvEmpList.CellContentClick
    Dim senderGrid = DirectCast(sender, DataGridView)
    Dim strUser As String = dgvEmpList.Rows(e.RowIndex).Cells(0).Value.ToString()

    If TypeOf senderGrid.Columns(e.ColumnIndex) Is DataGridViewButtonColumn AndAlso
e.RowIndex >= 0 Then
        frmEmpForm.Show()
        Me.Close()
        Try
            frmEmpForm.TblUserBindingSource.Position =
frmEmpForm.TblUserBindingSource.Find("Username", strUser)
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try
    End If
End Sub
Public Sub addButton()
    Dim btnEdit As New DataGridViewButtonColumn
    dgvEmpList.AutoGenerateColumns = False
    If dgvEmpList.Columns.Count = 5 Then
        dgvEmpList.Columns.Remove("button")
    End If
    btnEdit.Name = "button"
    btnEdit.HeaderText = ""
    btnEdit.Text = "Edit"
    btnEdit.FlatStyle = FlatStyle.Flat
    btnEdit.DefaultCellStyle.BackColor = Color.FromArgb(0, 192, 192)
    btnEdit.DefaultCellStyle.ForeColor = Color.Black
    btnEdit.UseColumnTextForButtonValue = True
    dgvEmpList.Columns.Add(btnEdit)

```

```

End Sub
End Class



### h. EMPLOYEE FORM


Public Class frmEmpForm
    Private Sub TblUserBindingNavigatorSaveItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TblUserBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.TblUserBindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.DsEmpData)
    End Sub
    Private Sub frmEmpForm_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        Me.TblUserTableAdapter.Fill(Me.DsEmpData.tblUser)
        TblUserBindingNavigator.Visible = False
    End Sub
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnBack.Click
        frmEmpfile.Show()
        Me.Close()
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSave.Click
        Me.Validate()
        Me.TblUserBindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.DsEmpData)
        frmEmpfile.Show()
        Me.Close()
    End Sub
    Public Sub fncAddData()
        Me.TblUserBindingSource.AddNew()
    End Sub
End Class

```

## Findings

**“What should we expect in CoPOS?”**

Using CoPOS we can find the business tracing for every served customer, stocks receiving, and even stock monitoring will be more easier, this will be reduce the time consuming using manual recording by paper and pen.

## Conclusion

CoPOS will help each business owner to strategy for their business such as popular product, daily sales revenue, tracing of new product, set up warehouse, and employee transaction.