MMET 383
Manufacturing Information Systems:

Final Project

Joshua Brown
Jules Bettler
Omar Garza

Introduction

Team members and their responsibilities

Jules Bettler - Coding and Report Writing
Joshua Brown - Coding and Report Writing
Omar Garza - Coding and Report Writing

Description

In the year 2045, Amazon collapsed and left a gap in the product-ordering-system market. While we do not have the resources, time, or capital to make a replacement and become the next Amazon, we have made a product ordering system for our own products. We will continue to expand our online offerings as we grow and get feedback.

Our goal is to create a simple but effective way to manage our online ordering for customers, managers, suppliers and shippers. The program needs to be robust enough to work for a variety of industries and environments, while still being simple and intuitive to use so anyone can order products easily and without mistakes.

Product structure

Our company designs and produces a variety of industrial and warehouse equipment, consisting of both material handling and material transfer equipment. We specialize in designing and reselling industrial robots in varying strengths and configurations, item transfer tools such as conveyor belts and roller belts, and inventory handling equipment such as hand carts and motorized carts

Figure 1: Industrial Robots in Varying Strengths

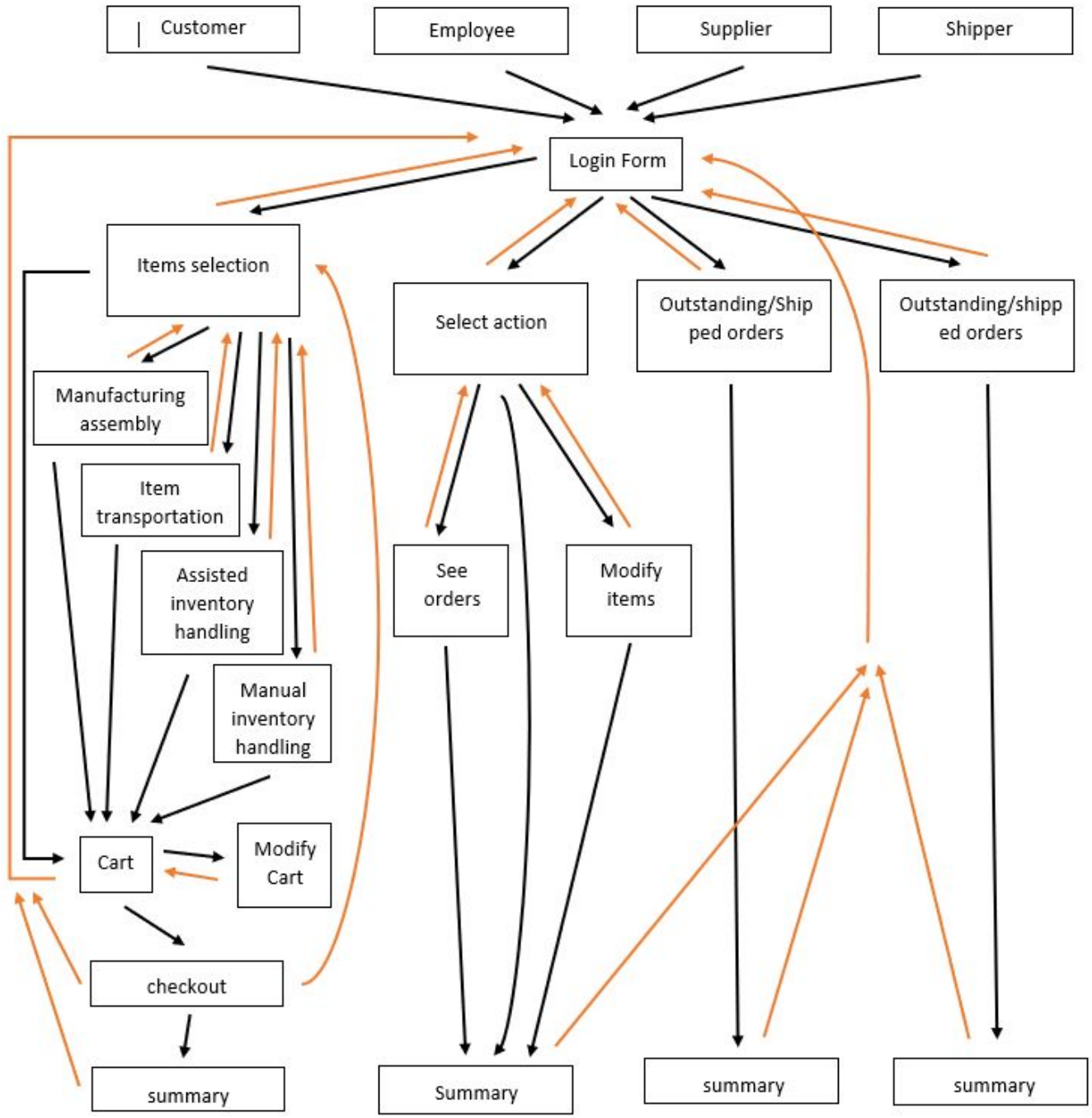
Figure 2: Item Transfer Tools, Conveyor Belt and Roller Belt

Figure 3: Inventory Handling Equipment, Hand Carts and Motorized Carts

Proposed system

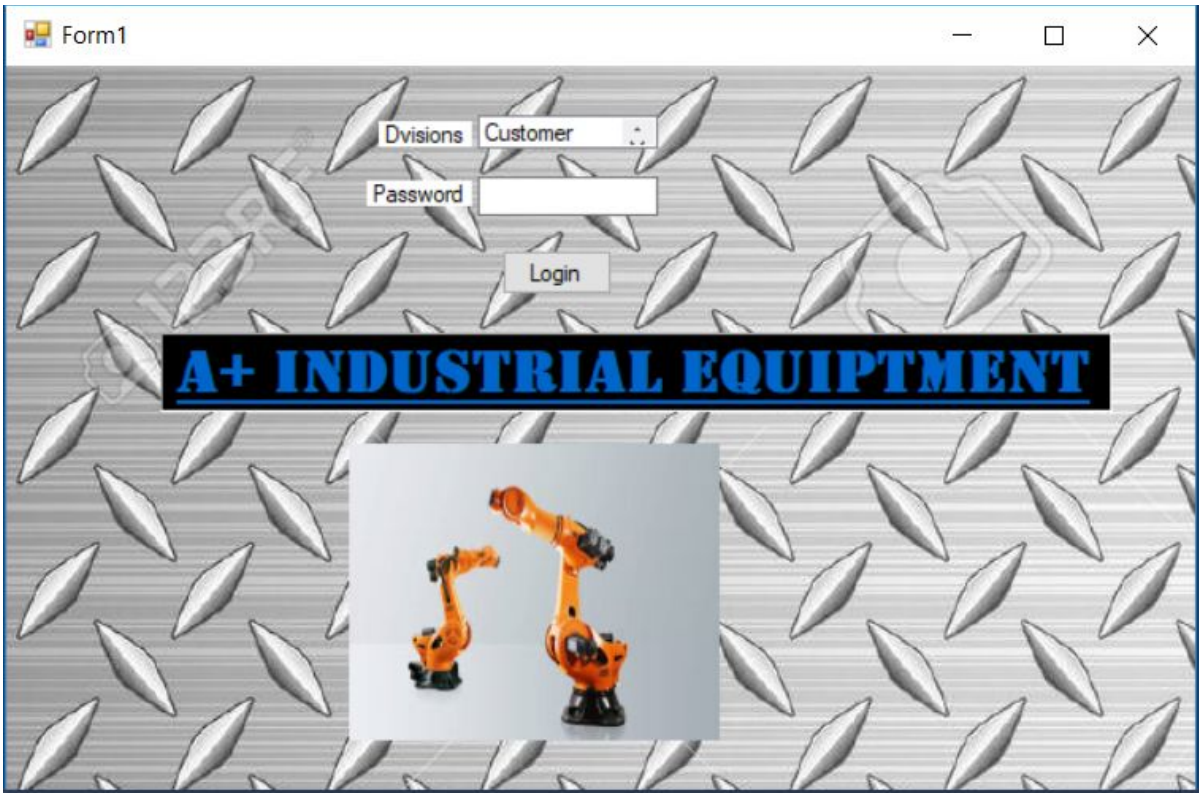
We are using Visual Basic to develop our online ordering system to make use of the easy frontend and backend development, which will speed up our lead time to get our post-Amazon online ordering system up on the market. Visual Basic allows developers to make forms and windows in the familiar style of Windows XP and other versions. Since this is the most widely used operating system, users using our online ordering system will already be familiar with the visual style. Our online ordering system provides an efficient way to organize, store, and modify customer information through the use of several related forms and databases. These in turn allow for limited access to different parts of the program depending on login credentials, such as Customer, Manager, Supplier, and Shipper.

Software application structure



Flowchart for program interface

Descriptions of Each Form



Login Form

Login

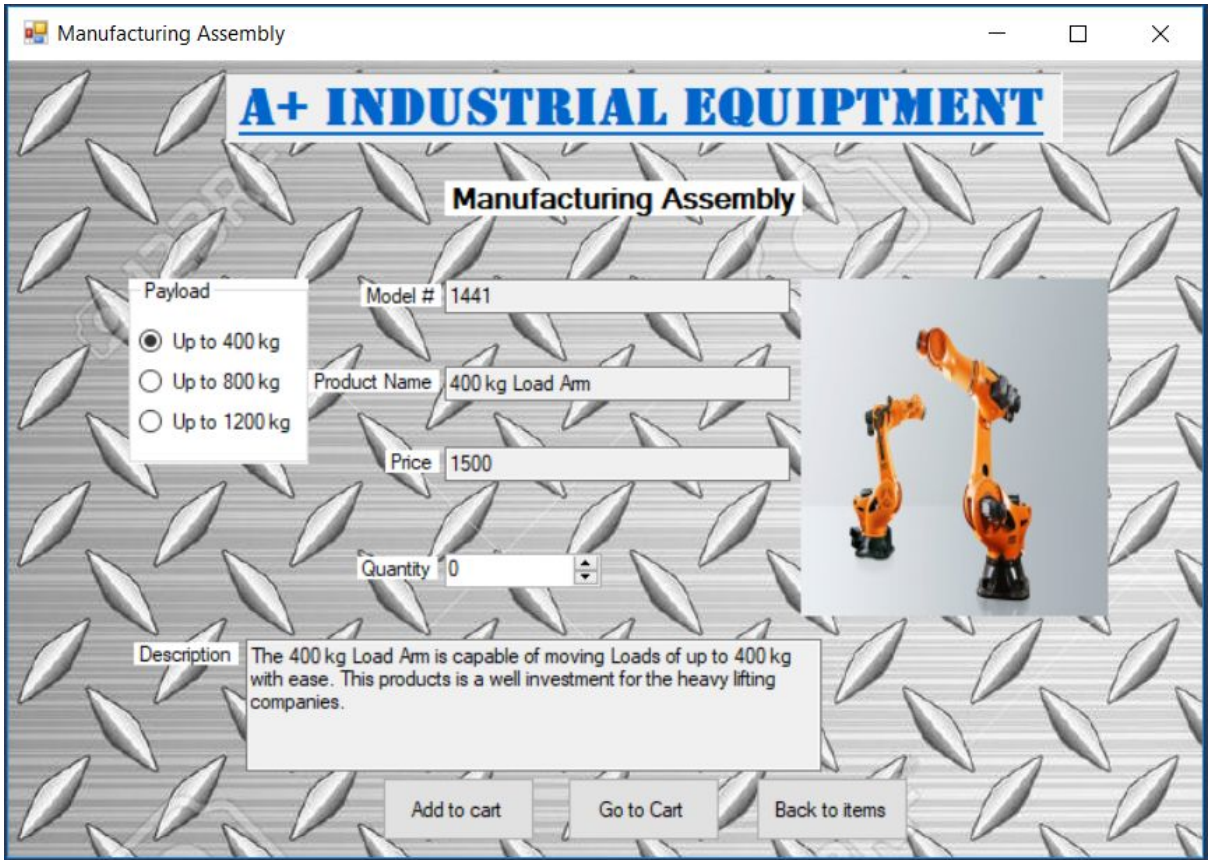
Our login form clearly displays the 4 division, Customer, Employee, Supplier, and Shipper. Underneath the division selection box is the Code input box, where users enter the password to enter their division specific form. The divisions are as follows:

- Customers are able to access the item selection form, where they can choose to shop from our different products.
    - **Password: 123**
- Employee is able to view and modify all orders, and send these orders to the supplier to be manufactured.
    - **Password: 456**
- Supplier is able to see the amount and type of products ordered by the customer, and sets up shipment with the shipper in order to get it to the customer.
    - **Password: 789**
- Shipper is able to see the customer and shipping information in order to ship the product from the supplier to the customer.
    - **Password: 159**

Items Selection Form

<u>Items Selection</u>

The item selection form has 4 buttons to select different product categories such as manufacturing assembly, item transportation, assisted inventory handling and manual inventory handling. Each “Shop” button leads to the item detail form for that product category. “Go to cart” leads to your cart, and the “Back” button goes back to the login form to switch divisions if needed.



Manufacturing Assembly Detail Form

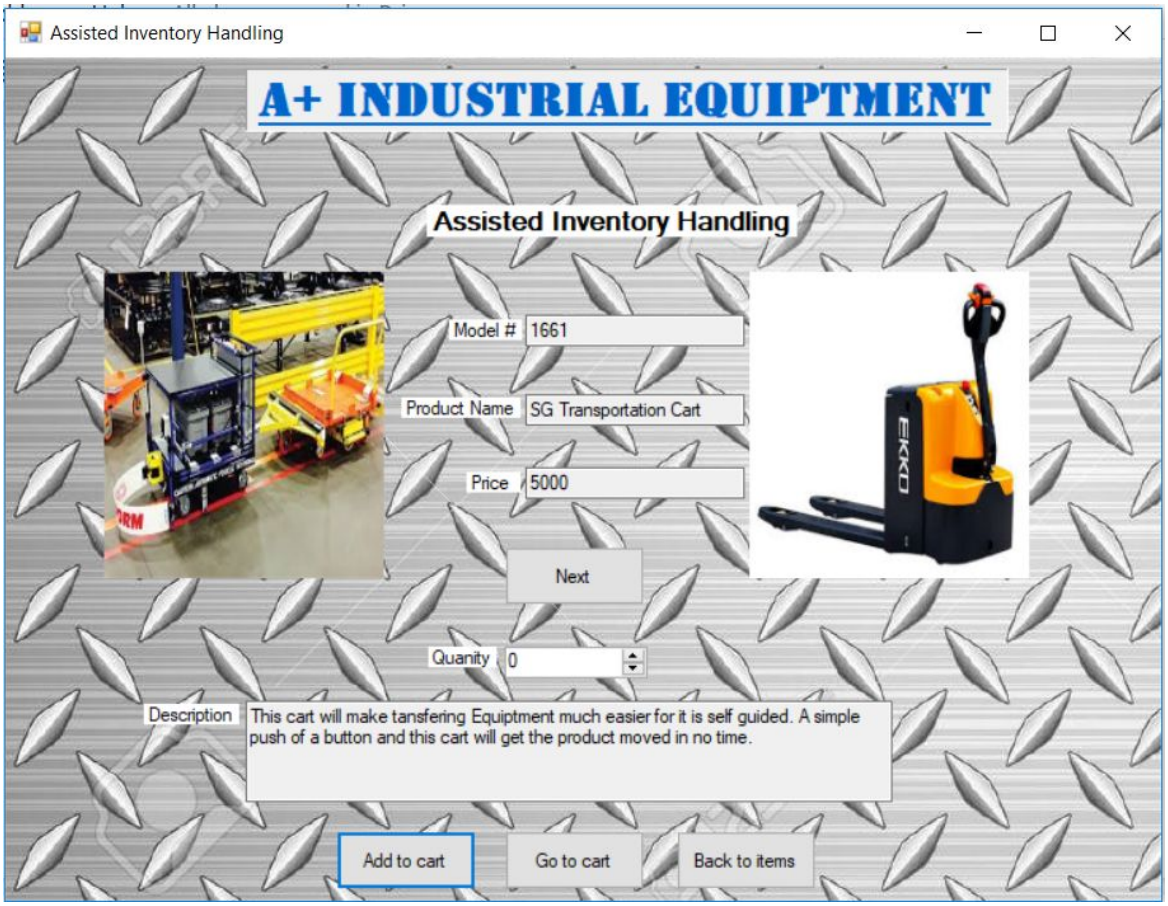
<u>Manufacturing Assembly Detail</u>

The arm option and selection form gives user the option to select different options when choosing to purchase a robotic arm, including payload and quantity. There is a button at the bottom right to add the item to cart, one to view the cart, and one to go back to the item selection form.

Item Transportation Detail Form

<u>Item Transportation Detail</u>

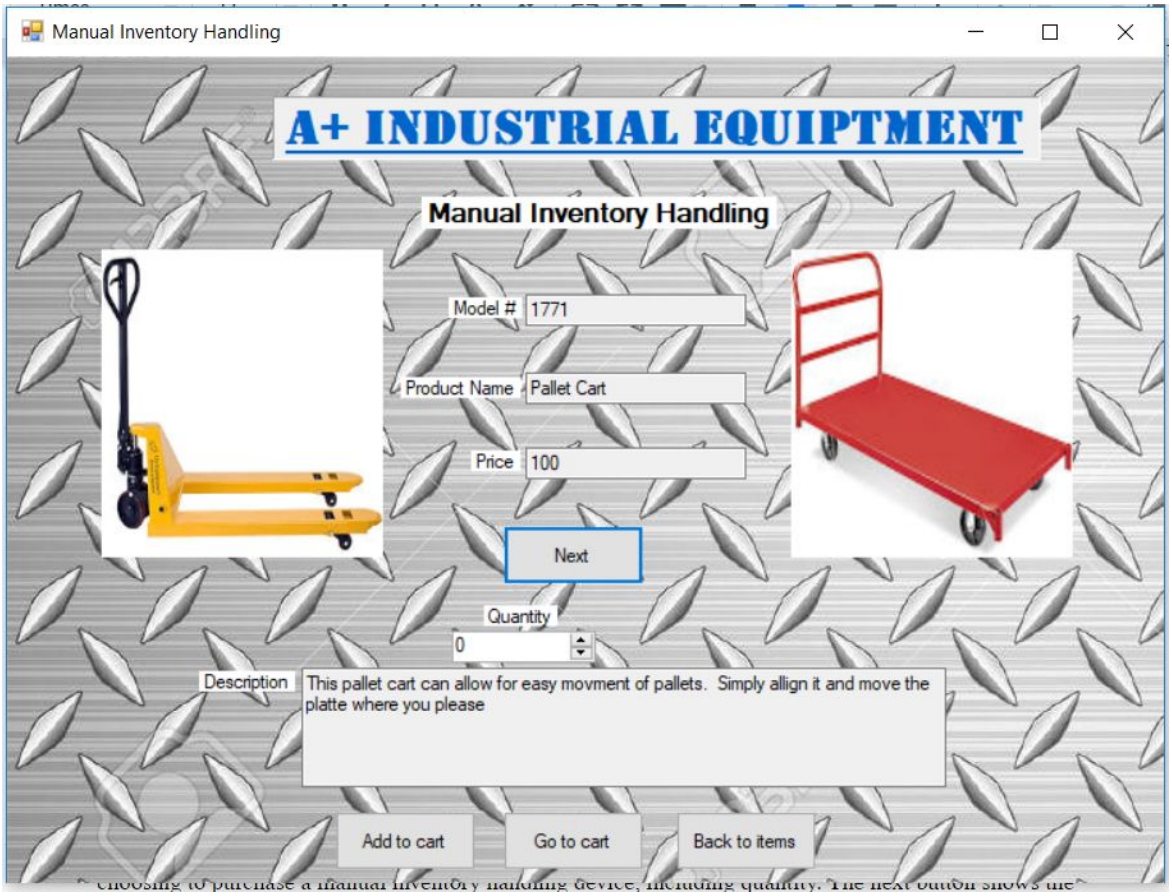
The item transportation detail form gives user the option to select different options when choosing to purchase a item transportation device, including length in meters. The next button shows the options for a frictionless roller instead of a power conveyor. There is a button at the bottom right to add the current item to cart, one to view the cart, and one to go back to the item selection form.



Assisted Inventory Handling Detail Form

<u>Assisted Inventory Handling Detail</u>

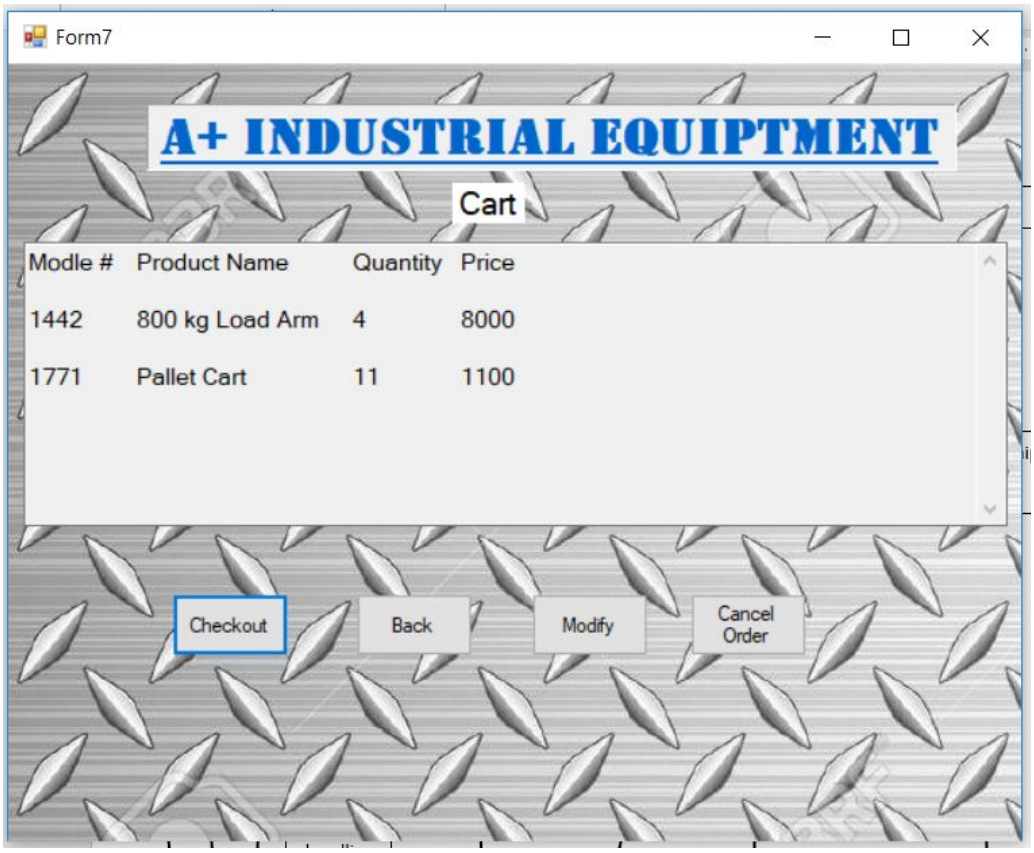
The assisted inventory handling detail form gives user the option to select different options when choosing to purchase an assisted inventory handling device, including quantity. The next button shows the options for a motorized pallet mover instead of a self-guided transportation cart. There is a button at the bottom right to add the current item to cart, one to view the cart, and one to go back to the item selection form.



Manual Inventory Handling Detail Form

<u>Manual Inventory Handling Detail Form</u>

The manual inventory handling detail form gives user the option to select different options when choosing to purchase a manual inventory handling device, including quantity. The next button shows the options for a flatbed cart mover instead of a pallet cart. There is a button at the bottom right to add the current item to cart, one to view the cart, and one to go back to the item selection form.

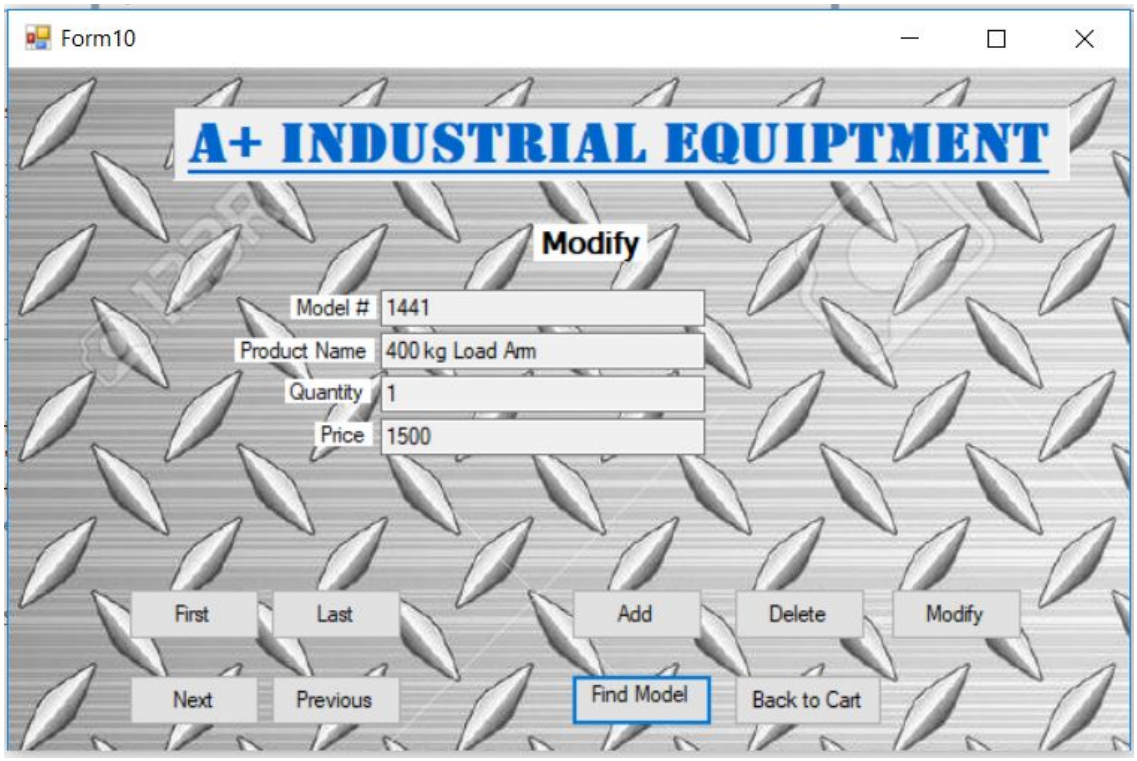


Cart Form

<u>Cart</u>

The cart form shows the user what items they have bought, the part number, the quantity, and the price. The “Checkout” button leads the user to input payment information, the “Back” button goes back to the item selection form, the “Modify” button leads to the modify order form, and the “Cancel Order” button takes the user back to the login form.

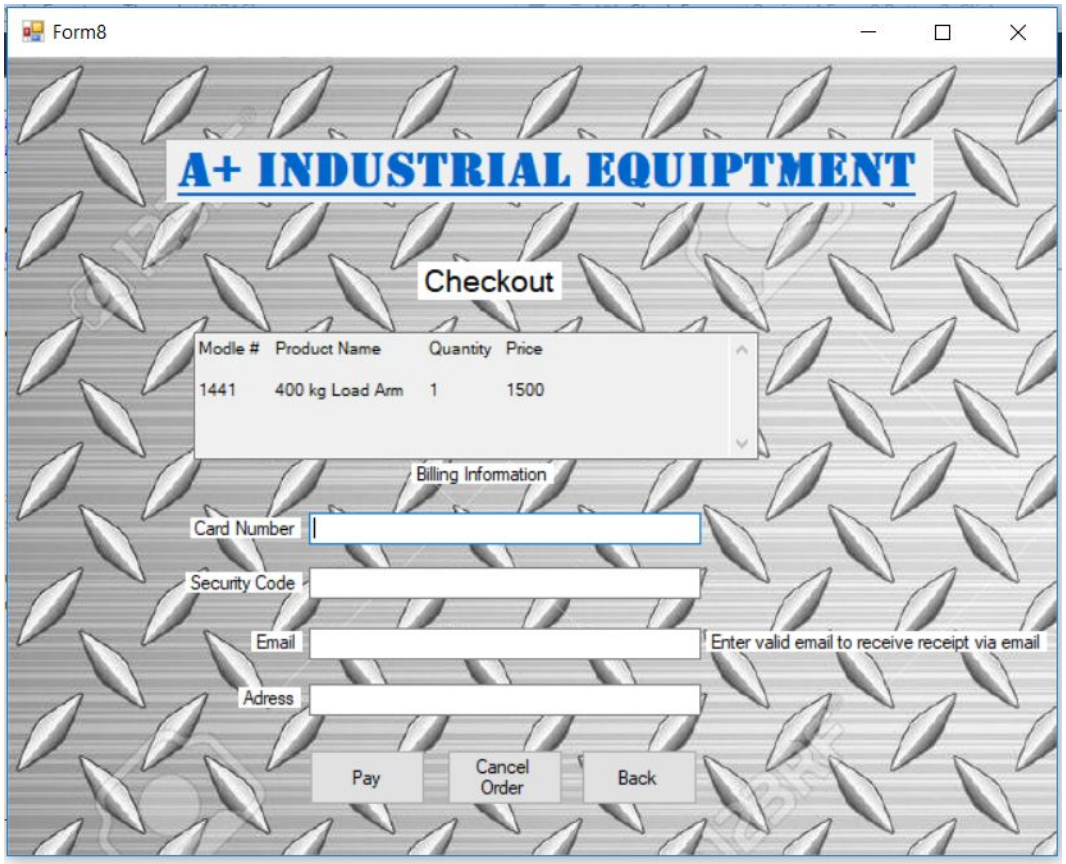
The “Checkout” and “Modify” buttons are not available when there are no items in the cart.



Modify Order Form

<u>Modify Order</u>

The modified form allows the user to modify the quantity of items in the cart by pressing the "Modify" button, and the "Add" button takes the user back to the item selection form to select more products. The other buttons are used to navigate through your items in the cart, and find or delete items.

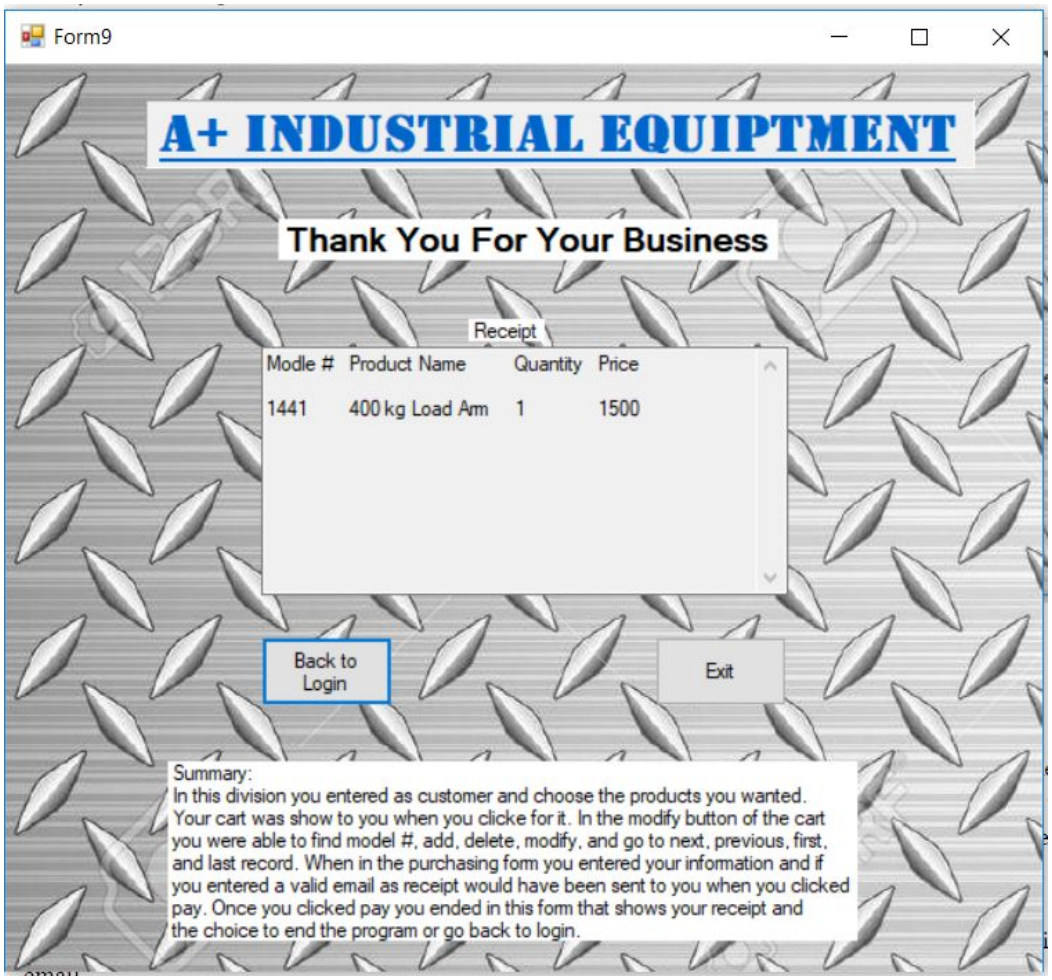


Checkout Form

<u>Checkout</u>

The checkout form contains a summary of your finalized cart, and invites the user to input the payment information, email and address for the order. The “Pay” button finalizes the order and leads to the summary page to confirm what has been purchased, the “Cancel Order” takes the user back to the login form, and the back button takes the user back to the item selection form to select new products.

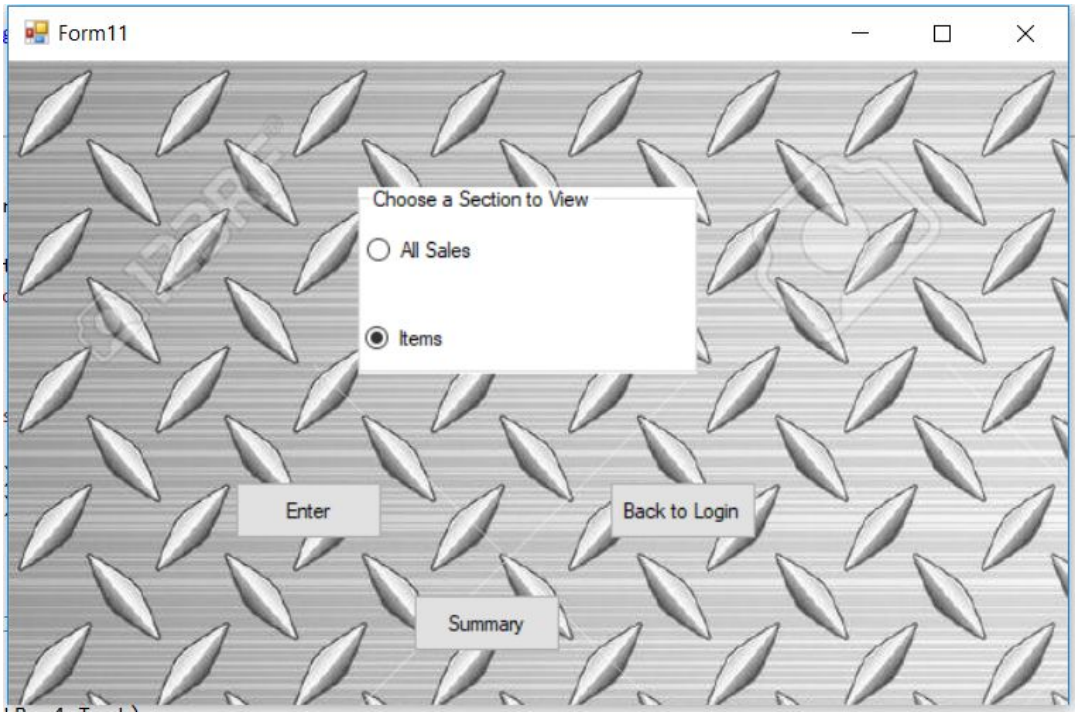
If a user inputs their email, an actual email is sent to the email provided, as long as it is a valid email.



Customer Summary form

Customer Summary

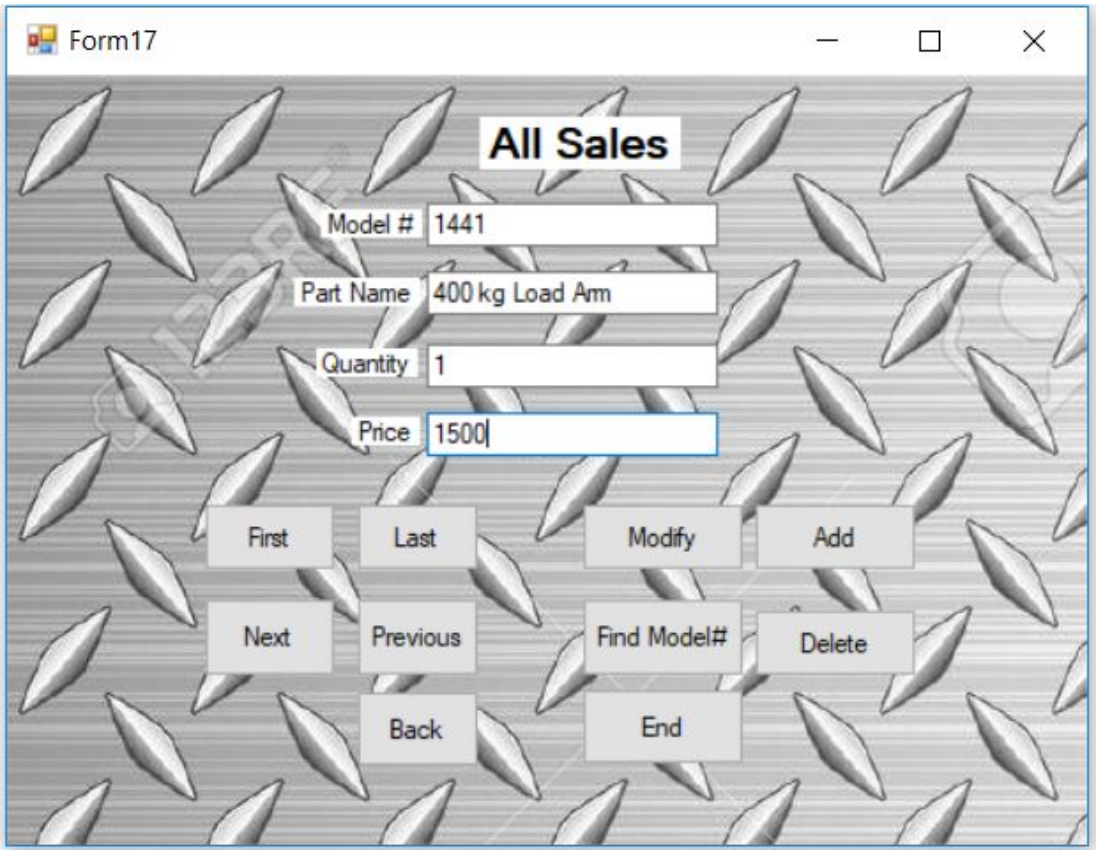
The customer summary from provides a summary that summarizes the activities that took place. The cart is displayed as well as a summary message.



Employee select action form

<u>Employee select action</u>

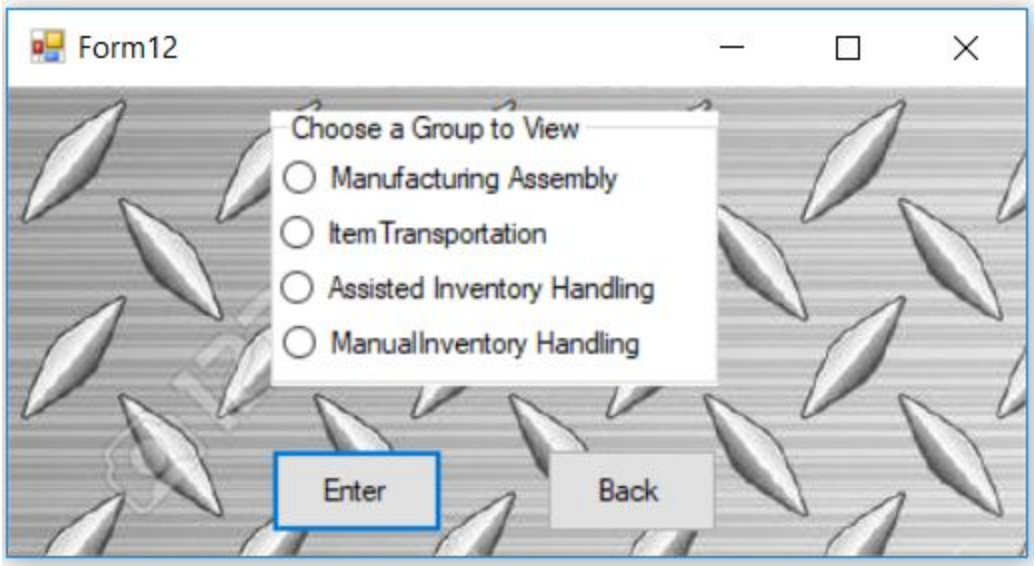
In this form, an employee can select whether to view all sales or to modify the items visible to the customer. The employee can also go directly to the summary form, which resumes the actions taken or that can be taken. The “Enter” button confirms the selection and the “Back to login” button takes the employee back to the login form.



View All Sales Form

<u>View all sales</u>

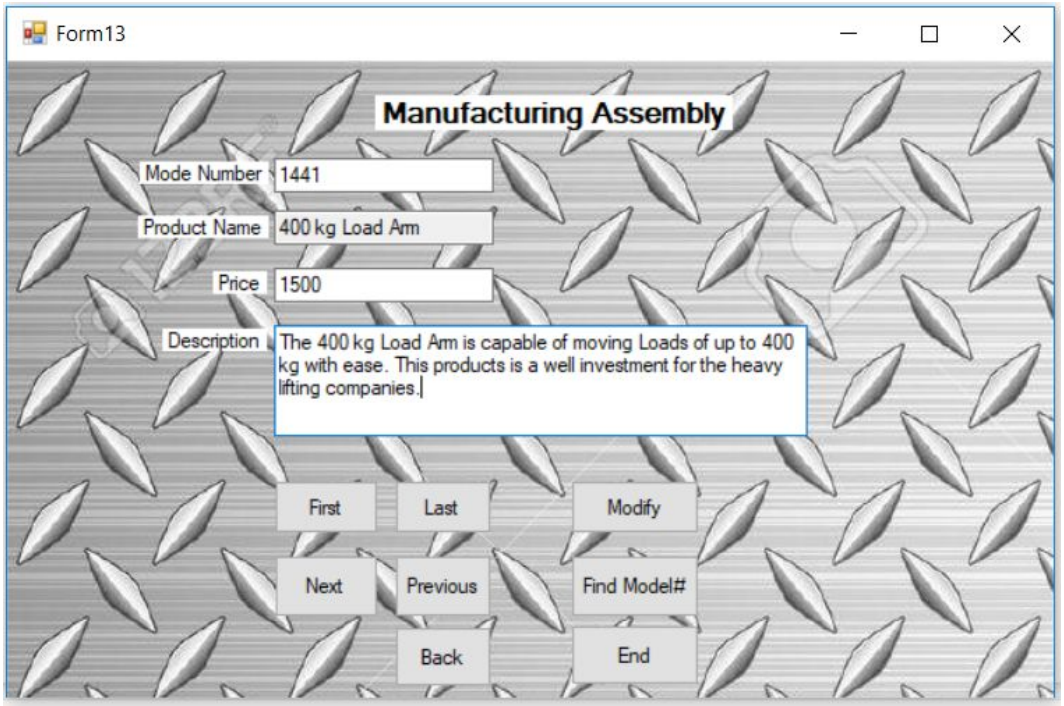
The view all sales form allows the employee to view all completed orders from customer, and modify, add, delete these orders. The “End” button brings the employee back to the login form and the “Back” button lead back to the employee select action form.



Modify Items Selection Form

Modify Items Selection

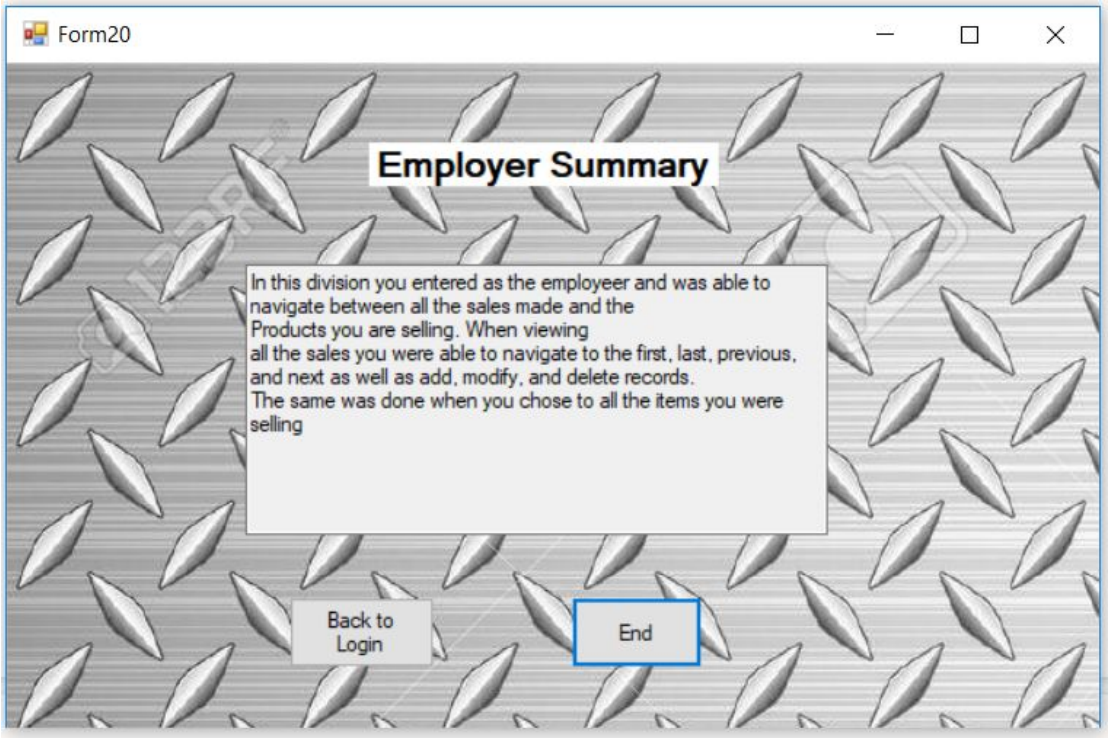
This form enables the employee to select which product they would like to modify, and provides a button to confirm the selection and a button to go back to the employee select action form.



Modify Items Form

<u>Modify Items</u>

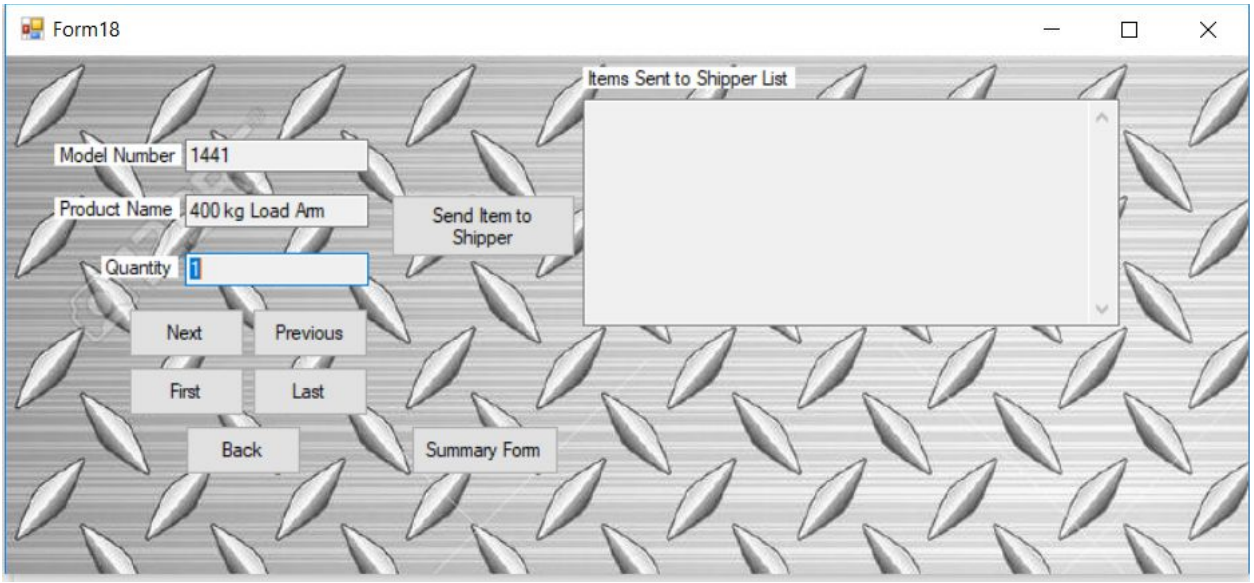
Once the employee has selected which product they would like to modify, this forms allows them to enter the updated information, and change the model number, product name, price, and description. The navigation buttons only work for the products within the product category. The “back” button leads back to the Modify Items Selection form, and the “End” button ends the program.



Employee Summary Form

<u>Employee Summary</u>

The employee summary form provides a summary of the activities that took place.

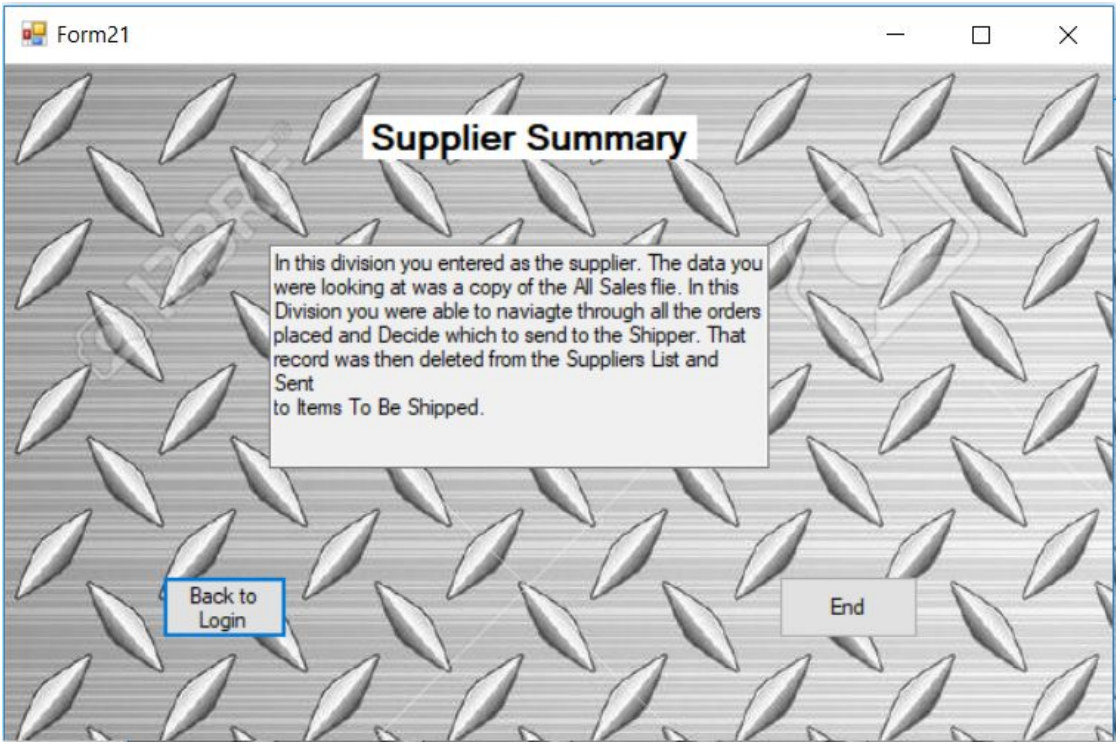


Supplier Control Center Form

<u>Supplier Control Center</u>

The supplier control center receives orders from the completed customer orders, and has navigation buttons to see all orders. Once an item is manufactured, the “Send item to shipper” button is clicked and the item is removed from the customer orders and placed into the Sent To Shipper table. The items in this table now appear in the shipper form, waiting to be marked as shipped. The “Summary” button leads to a summary of completed actions.

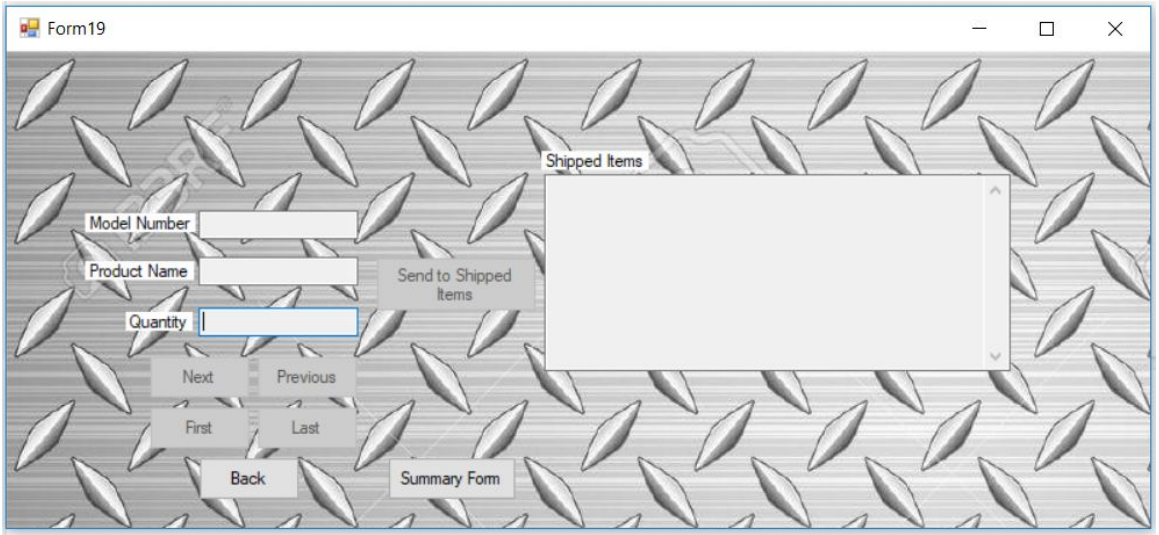
A warning will be displayed if no items have been ordered to the supplier by the customer



Supplier Summary Form

<u>Supplier Summary</u>

The supplier summary form provides a summary of the activities that took place.

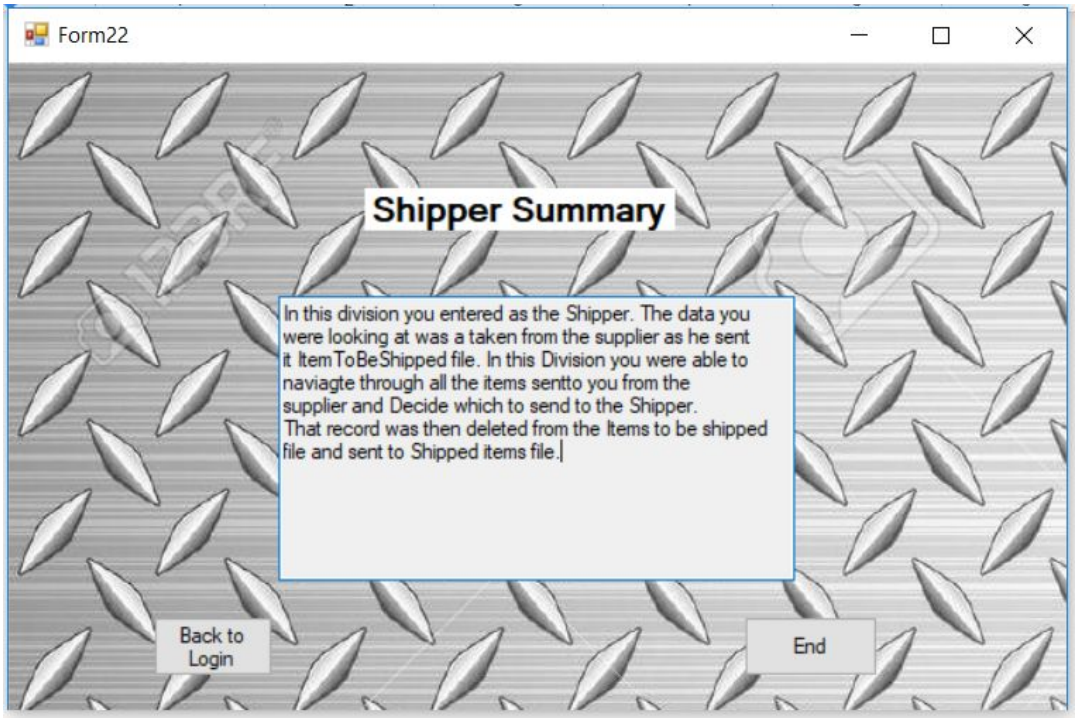


Shipper Control Center Form

<u>Shipper Control Center</u>

The shipper control center receives orders from the manufacturer supplier products, and has navigation buttons to see all manufactured products. Once an item is shipped, the “Send To Shipped Items” button is clicked and the item is removed from the manufactured products and placed into the Shipped Items table. The items in this table are then shipped to customers using the address they provided. The “Summary” button leads to a summary of completed actions.

A warning will be displayed if no items have been sent to the shipper by the supplier



Shipper summary form

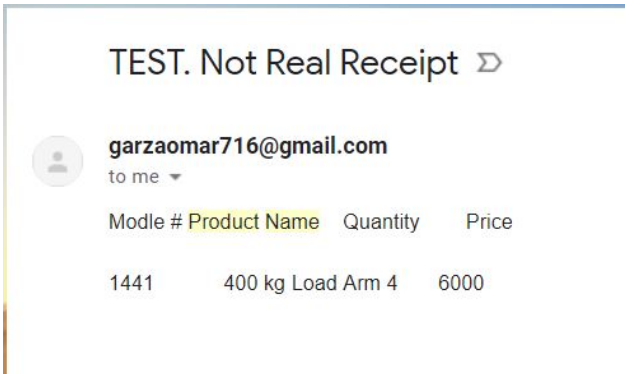
<u>Shipper summary</u>

The shipped summary form provides a summary of the activities that took place.

Special Features

Email

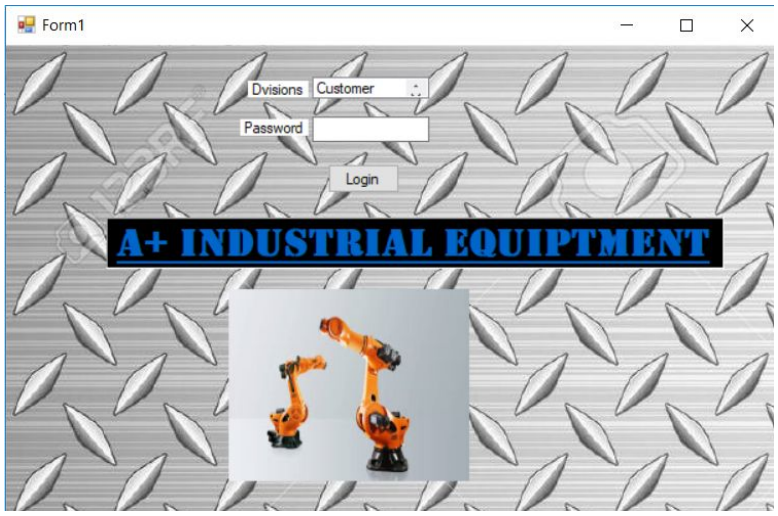
The customer division of our online ordering program has actual email capabilities. When a customer enters their payment information at the checkout, after ordering products, our programs an email to the customer, as long as the email is valid. The email is sent through one of our group member's tamu.edu email, using a script created in visual basic and executed by the "confirm" button in the checkout form.



Example of Received Email From Program

Timer

Our login form uses a timer to cycle through the image at the bottom, showing customers a sample of what products they can order. This is similar to what Amazon does on their website, cycling through images on the home age to catch the eye of potential customers. The timer was a function we never learned in class but was easy to implement to cycle through a limited set of images



Login Form, with the timer function applied to the image at the bottom

Debugging Process

For an example of a debugging process the following error appeared when trying to read a file to display on to the cart display.

```
22.vb        Form22.vb [Design]
roject4                                  (Form7 Events)                    Lo
33      Private Sub Form7_Load(sender As Object, e As EventArgs) Handles MyBase.Load
34          Dim D, A As Long
35          FileOpen(1, "CartPerCustomer.csv", OpenMode.Input)
36          Do Until EOF(2)
37              ReDim Preserve ModelNum(0 To i)
38              ReDim Preser
39              ReDim Preser
40              ReDim Preser
41              Input(1, Mod
42              Input(1, Pro
43              Input(1, Pro
44              Input(1, Pro
45              PriceTotal =
46              CartDisplay                                   b & ProdQuantity(i) &
47              i = i + 1
48          Loop
49          total = i
```

Exception Unhandled

System.IO.IOException: 'Bad file name or number.'

View Details | Copy Details

Exception Settings

Code with error message

To debug this process the code was looked thoroughly to extract why the file name or number was bad. Eventually, it was noted that the file open number was 1 while the do until EOF was 2 therefore causing this error in the code. The file number was changed to match with the file open number and allowed for the code to continue on.

```
53          My.Computer.FileSystem.DeleteFile(a & "\CartPerCustomer.csv")
54      End Sub
55
56      Private Sub Form8_Load(sender As Object, e As EventArgs) Handles MyBase.Load
57          FileOpen(1, "CartPerCustomer.csv", OpenMode.Input)
58          Do Until EOF(1)
59              ReDim Preserve ModelNum(0 To i)
60              ReDim Preserve ProdName(0 To i)
61              ReDim Preserve ProdQuantity(0 To i)
62              ReDim Preserve ProdPrice(0 To i)
63              Input(1, ModelNum(i))
64              Input(1, ProdName(i))
65              Input(1, ProdQuantity(i))
66              Input(1, ProdPrice(i))
67              CartDisplay = CartDisplay & ModelNum(i) & vbTab & ProdName(i) & vbTab & ProdQuantity(i) &
68              PriceTotal = PriceTotal + ProdPrice(i)
69              i = i + 1
70          Loop
71          total = i
72          TextBox1.Text = "Modle #" & vbTab & "Product Name" & vbTab & "Quantity" & vbTab & "Price" & v
73          FileClose(1)
74      End Sub
```

Fixed code

Conclusion and Future Directions

In this project, we created a database for an online ordering system for our products, including material handling and transfer equipment. There are different divisions for the customer, employee, supplier, and shipper, and each division shares data with the others. Our software makes it easier for a company to manage an online business, and it will continue to be updated to grow and improve.

Future updated include switching out of Visual Basic for a more clean, modern look; adding in-browser capability to our program to enable user to search for it online and not have to open a new program; and refining the look and feel of each forms with a dedicated UI/UX employee on the program design team. As we implement these updates, we will continue to grow and expand our business, and the areas that we ship to.

Appendix

```
Imports System.IO
Public Class Form1
        Private i, j, k, t, z, total As Long
        Private XDivisions() As String
        Private XPictures() As String
        Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        i = 0
        j = 0
        t = 0
        TextBox1.Text = ""
        FileOpen(1, "Divisions.csv", OpenMode.Input)
        Do Until EOF(1)
        ReDim Preserve XDivisions(0 To i)
        Input(1, XDivisions(i))
        ListBox1.Items.Add(XDivisions(i))
        i = i + 1
        Loop
        FileClose(1)

        FileOpen(2, "Pictures1.csv", OpenMode.Input)
        Do Until EOF(2)
        ReDim Preserve XPictures(0 To j)
        Input(2, XPictures(j))
        j = j + 1
        Loop
        total = j
        FileClose(2)
        Timer1.Start()
        Call Display(0)
        FileOpen(3, "CartPerCustomer.csv", OpenMode.Append)
        FileClose(3)
        Dim a As String
        a = CurDir()
        My.Computer.FileSystem.DeleteFile(a & "\CartPerCustomer.csv")

        End Sub

        Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
        t = t + 1
        If t Mod 3 = 0 Then
        Call Display(0)
        ElseIf t Mod 3 = 1 Then
        Call Display(1)
        ElseIf t Mod 3 = 2 Then
        Call Display(2)
        End If
        End Sub

        Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        If TextBox1.Text = "123" And ListBox1.Text = "Customer" Then

        Form2.Show()
        Me.Hide()
        ElseIf TextBox1.Text = "456" And ListBox1.Text = "Employeer" Then
        Me.Hide()
```

```
        Form11.Show()

        ElseIf TextBox1.Text = "789" And ListBox1.Text = "Supplier" Then
        Me.Hide()
        Form18.Show()
        ElseIf TextBox1.Text = "159" And ListBox1.Text = "Shipper" Then
        Me.Hide()
        Form19.Show()
        Else
        MsgBox("Incorrect Password or Divsion")
        End If
        End Sub
        Private Sub Display(ByVal A As Long)
        PictureBox1.Load(XPictures(A))
        End Sub
End Class

Public Class Form2
        Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Me.Close()
        Form3.Show()
        End Sub

        Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        Me.Close()
        Form4.Show()
        End Sub

        Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        Me.Close()
        Form5.Show()
        End Sub

        Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
        Me.Close()
        Form6.Show()
        End Sub

        Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
        Dim a As String
        a = CurDir()
        My.Computer.FileSystem.DeleteFile(a & "\CartPerCustomer.csv")
        Me.Close()
        Form1.Show()
        End Sub

        Private Sub Form2_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        FileOpen(1, "CartPerCustomer.csv", OpenMode.Append)
        FileClose(1)
        PictureBox1.Load("arm.jpg")
        PictureBox2.Load("conveyor.jpg")
        PictureBox3.Load("motorized front.jpg")
        PictureBox4.Load("manual front.jpg")
        End Sub

        Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
        Me.Close()
```

```
        Form7.Show()
        End Sub
End Class

Public Class Form3
        Private Price As Long
        Private Sub Form3_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the 'ItemsDBDataSet1.ManufacturingAssembly' table. You can move, or remove it, as
needed.
        Me.ManufacturingAssemblyTableAdapter1.Fill(Me.ItemsDBDataSet1.ManufacturingAssembly)

        FileOpen(1, "CartPerCustomer.csv", OpenMode.Append)


        End Sub

        Private Sub RadioButton1_CheckedChanged(sender As Object, e As EventArgs) Handles RadioButton1.CheckedChanged
        ManufacturingAssemblyBindingSource1.MoveFirst()
        End Sub

        Private Sub RadioButton2_CheckedChanged(sender As Object, e As EventArgs) Handles RadioButton2.CheckedChanged
        ManufacturingAssemblyBindingSource1.Position = 1
        End Sub

        Private Sub RadioButton3_CheckedChanged(sender As Object, e As EventArgs) Handles RadioButton3.CheckedChanged
        ManufacturingAssemblyBindingSource1.MoveLast()
        End Sub

        Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        If NumericUpDown1.Value = 0 Then
        MsgBox("Enter a Quantity")
        Exit Sub
        Else
        Price = TextBox3.Text * NumericUpDown1.Value
        End If
        WriteLine(1, TextBox1.Text, TextBox2.Text, NumericUpDown1.Value, Price)
        NumericUpDown1.Value = 0

        End Sub

        Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        FileClose(1)
        Me.Close()
        Form7.Show()
        End Sub

        Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        FileClose(1)
        Me.Close()
        Form2.Show()
        End Sub
End Class

Public Class Form4
        Private price As Long
        Private totalItemTransportation As Long
        Private Sub Label6_Click(sender As Object, e As EventArgs) Handles Label6.Click
```

```
        End Sub

        Private Sub Form4_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the 'ItemsDBDataSet1.ItemTransportation' table. You can move, or remove it, as needed.
        Me.ItemTransportationTableAdapter1.Fill(Me.ItemsDBDataSet1.ItemTransportation)

        totalItemTransportation = ItemsDBDataSet1.ItemTransportation.Count
        FileOpen(1, "CartPerCustomer.csv", OpenMode.Append)

        End Sub

        Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim Xposition As Long
        Xposition = ItemTransportationBindingSource1.Position
        If Xposition = totalItemTransportation - 1 Then
        ItemTransportationBindingSource1.MoveFirst()
        Else
        ItemTransportationBindingSource1.MoveNext()
        End If
        End Sub

        Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
        If NumericUpDown3.Value = 0 Then
        MsgBox("Enter a amount of meters needed")
        Exit Sub
        Else
        Price = TextBox3.Text * NumericUpDown3.Value
        End If
        WriteLine(1, TextBox1.Text, TextBox2.Text, NumericUpDown3.Value, price)
        NumericUpDown3.Value = 0
        End Sub

        Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
        FileClose(1)
        Me.Close()
        Form7.Show()
        End Sub

        Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        FileClose(1)
        Me.Close()
        Form2.Show()
        End Sub

        Private Sub Label2_Click(sender As Object, e As EventArgs) Handles Label2.Click

        End Sub
End Class

Public Class Form5
        Private totalAssistedInventoryHandling As Long
        Private Price As Long
        Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        FileClose(1)
        Me.Close()
        Form2.Show()
```

```
        End Sub

        Private Sub Form5_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the 'ItemsDBDataSet1.AssistedInventoryHandling' table. You can move, or remove it, as
needed.
        Me.AssistedInventoryHandlingTableAdapter1.Fill(Me.ItemsDBDataSet1.AssistedInventoryHandling)

        totalAssistedInventoryHandling = AssistedInventoryHandlingBindingSource1.Count()
        FileOpen(1, "CartPerCustomer.csv", OpenMode.Append)

        End Sub

        Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim Xposition As Long
        Xposition = AssistedInventoryHandlingBindingSource1.Position
        If Xposition = totalAssistedInventoryHandling - 1 Then
        AssistedInventoryHandlingBindingSource1.MoveFirst()
        Else
        AssistedInventoryHandlingBindingSource1.MoveNext()
        End If
        End Sub

        Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
        If NumericUpDown3.Value = 0 Then
        MsgBox("Enter a amount of meters needed")
        Exit Sub
        Else
        Price = TextBox3.Text * NumericUpDown3.Value
        End If
        WriteLine(1, TextBox1.Text, TextBox2.Text, NumericUpDown3.Value, Price)
        NumericUpDown3.Value = 0
        End Sub

        Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
        FileClose(1)
        Me.Close()
        Form7.Show()
        End Sub

        Private Sub Label2_Click(sender As Object, e As EventArgs) Handles Label2.Click

        End Sub
End Class

Public Class Form6
        Private totalManualInventoryHandling As Long
        Private Price As Long
        Private Sub Form6_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the 'ItemsDBDataSet1.ManualInventoryHandling' table. You can move, or remove it, as
needed.
        Me.ManualInventoryHandlingTableAdapter1.Fill(Me.ItemsDBDataSet1.ManualInventoryHandling)

        totalManualInventoryHandling = ManualInventoryHandlingBindingSource1.Count()
        FileOpen(1, "CartPerCustomer.csv", OpenMode.Append)
        End Sub

        Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
```

```
        Dim Xposition As Long
        Xposition = ManualInventoryHandlingBindingSource1.Position
        If Xposition = totalManualInventoryHandling - 1 Then
        ManualInventoryHandlingBindingSource1.MoveFirst()
        Else
        ManualInventoryHandlingBindingSource1.MoveNext()
        End If
        End Sub

        Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
        If NumericUpDown3.Value = 0 Then
        MsgBox("Enter a amount of meters needed")
        Exit Sub
        Else
        Price = TextBox3.Text * NumericUpDown3.Value
        End If
        WriteLine(1, TextBox1.Text, TextBox2.Text, NumericUpDown3.Value, Price)
        NumericUpDown3.Value = 0
        End Sub

        Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
        FileClose(1)
        Me.Close()
        Form7.Show()
        End Sub

        Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        FileClose(1)
        Me.Close()
        Form2.Show()
        End Sub
End Class

Public Class Form7
        Private ModelNum() As String
        Private ProdName() As String
        Private ProdQuantity() As Long
        Private ProdPrice() As Long
        Private i, j, total, PriceTotal As Long
        Private CartDisplay As String

        Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Me.Close()
        Form8.Show()
        End Sub

        Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        Me.Close()
        Form2.Show()
        End Sub

        Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click

        Dim a As String
        a = CurDir()
        My.Computer.FileSystem.DeleteFile(a & "\CartPerCustomer.csv")
        Me.Close()
```

```
        Form1.Show()
        End Sub

        Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        Me.Close()
        Form10.Show()
        End Sub

        Private Sub Form7_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Dim D, A As Long
        FileOpen(1, "CartPerCustomer.csv", OpenMode.Input)
        Do Until EOF(1)
        ReDim Preserve ModelNum(0 To i)
        ReDim Preserve ProdName(0 To i)
        ReDim Preserve ProdQuantity(0 To i)
        ReDim Preserve ProdPrice(0 To i)
        Input(1, ModelNum(i))
        Input(1, ProdName(i))
        Input(1, ProdQuantity(i))
        Input(1, ProdPrice(i))
        PriceTotal = PriceTotal + ProdPrice(i)
        CartDisplay = CartDisplay & ModelNum(i) & vbTab & ProdName(i) & vbTab & ProdQuantity(i) & vbTab & ProdPrice(i) &
vbNewLine & vbNewLine
        i = i + 1
        Loop
        total = i
        FileClose(1)
        '  For D = 0 To total - 1
        ' For j = 0 To total - 2
        'If ModelNum(j) = ModelNum(j + 1) Then
        'ProdQuantity(j) = ProdQuantity(j) + ProdQuantity(j + 1)
        'ProdPrice(j) = ProdPrice(j) + ProdPrice(j + 1)
        'A = j + 1
        'End If
        '
        'If D <> A Then
        'FileOpen(2, "ReceiptPerCustomer.csv", OpenMode.Output)
        'WriteLine(2, ModelNum(D), ProdName(D), ProdQuantity(D), ProdPrice(D))
        'CartDisplay = CartDisplay & ModelNum(D) & vbTab & ProdName(D) & vbTab & ProdQuantity(D) & vbTab & ProdPrice(D) &
vbNewLine & vbNewLine
        'FileClose(2)
        'End If
        'Next
        'Next
        ' FileClose(1)
        TextBox1.Text = "Modle #" & vbTab & "Product Name" & vbTab & "Quantity" & vbTab & "Price" & vbNewLine & vbNewLine &
CartDisplay
        If TextBox1.Text = "Modle #" & vbTab & "Product Name" & vbTab & "Quantity" & vbTab & "Price" & vbNewLine & vbNewLine
Then
        Button1.Enabled = False
        Button2.Enabled = False
        End If
        End Sub
End Class

Imports System.Net.Mail
Public Class Form8
```

```
Private ModelNum() As String
Private ProdName() As String
Private ProdQuantity() As String
Private ProdPrice() As Long
Private i, j, total, PriceTotal As Long
Private CartDisplay As String

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
Select Case ""
Case TextBox2.Text, TextBox3.Text, TextBox4.Text, TextBox5.Text
MsgBox("Missing Information")
Exit Sub
End Select
FileOpen(2, "AllSales.csv", OpenMode.Append)
FileOpen(3, "SupplierList.csv", OpenMode.Append)
For i = 0 To total - 1
WriteLine(2, ModelNum(i), ProdName(i), ProdQuantity(i), ProdPrice(i))
WriteLine(3, ModelNum(i), ProdName(i), ProdQuantity(i), ProdPrice(i))
Next
FileClose(3)
FileClose(2)
Dim EmailMessage As New MailMessage()
Try
EmailMessage.From = New MailAddress("garzaomar716@gmail.com")
EmailMessage.To.Add(TextBox4.Text)
EmailMessage.Subject = "TEST. Not Real Receipt"
EmailMessage.Body = TextBox1.Text
Dim SMPT As New SmtpClient("smtp.gmail.com")
SMPT.Port = 587
SMPT.EnableSsl = True
SMPT.Credentials = New System.Net.NetworkCredential("garzaomar716@gmail.com", "a44f2b6f6B")
SMPT.Send(EmailMessage)
Catch ex As Exception
End Try
Me.Close()
Form9.Show()
End Sub

Private Sub PrintDocument1_PrintPage(sender As Object, e As Printing.PrintPageEventArgs)

End Sub

Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
Me.Close()
Form2.Show()
End Sub

Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
Dim a As String
a = CurDir()
My.Computer.FileSystem.DeleteFile(a & "\CartPerCustomer.csv")
End Sub

Private Sub Form8_Load(sender As Object, e As EventArgs) Handles MyBase.Load
FileOpen(1, "CartPerCustomer.csv", OpenMode.Input)
Do Until EOF(1)
ReDim Preserve ModelNum(0 To i)
```

```
        ReDim Preserve ProdName(0 To i)
        ReDim Preserve ProdQuantity(0 To i)
        ReDim Preserve ProdPrice(0 To i)
        Input(1, ModelNum(i))
        Input(1, ProdName(i))
        Input(1, ProdQuantity(i))
        Input(1, ProdPrice(i))
        CartDisplay = CartDisplay & ModelNum(i) & vbTab & ProdName(i) & vbTab & ProdQuantity(i) & vbTab & ProdPrice(i) &
vbNewLine & vbNewLine
        PriceTotal = PriceTotal + ProdPrice(i)
        i = i + 1
        Loop
        total = i
        TextBox1.Text = "Modle #" & vbTab & "Product Name" & vbTab & "Quantity" & vbTab & "Price" & vbNewLine & vbNewLine &
CartDisplay
        FileClose(1)
        End Sub
End Class


Public Class Form9
        Private ModelNum() As String
        Private ProdName() As String
        Private ProdQuantity() As String
        Private ProdPrice() As Long
        Private CartDisplay As String
        Private PriceTotal, i As Long

        Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim a As String
        a = CurDir()
        My.Computer.FileSystem.DeleteFile(a & "\CartPerCustomer.csv")
        Me.Close()
        Form1.Show()
        End Sub

        Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        Dim a As String
        a = CurDir()
        My.Computer.FileSystem.DeleteFile(a & "\CartPerCustomer.csv")
        End
        End Sub

        Private Sub TextBox1_TextChanged(sender As Object, e As EventArgs) Handles TextBox1.TextChanged

        End Sub

        Private Sub Button3_Click(sender As Object, e As EventArgs)

        End Sub

        Private Sub Form9_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        FileOpen(1, "CartPerCustomer.csv", OpenMode.Input)
        Do Until EOF(1)
        ReDim Preserve ModelNum(0 To i)
        ReDim Preserve ProdName(0 To i)
        ReDim Preserve ProdQuantity(0 To i)
```

```
        ReDim Preserve ProdPrice(0 To i)
        Input(1, ModelNum(i))
        Input(1, ProdName(i))
        Input(1, ProdQuantity(i))
        Input(1, ProdPrice(i))
        CartDisplay = CartDisplay & ModelNum(i) & vbTab & ProdName(i) & vbTab & ProdQuantity(i) & vbTab & ProdPrice(i) &
vbNewLine & vbNewLine
        PriceTotal = PriceTotal + ProdPrice(i)
        i = i + 1
        Loop
        TextBox1.Text = "Modle #" & vbTab & "Product Name" & vbTab & "Quantity" & vbTab & "Price" & vbNewLine & vbNewLine &
CartDisplay
        FileClose(1)
        End Sub
End Class

Public Class Form10
        Private ModelNum() As String
        Private ProdName() As String
        Private ProdQuantity() As Long
        Private ProdPrice() As Long
        Private i, j, total, PriceTotal As Long
        Private CartDisplay As String


        Private Sub Button8_Click(sender As Object, e As EventArgs) Handles Button8.Click
        Dim X As String
        Dim Target, F As Long

        X = InputBox("Enter a record number")
        Target = -1
        For F = 0 To total - 1
        If X = ModelNum(F) Then
        Target = F
        Exit For
        End If
        Next
        If Target = -1 Then
        MsgBox("No such Model Number")
        Else
        Call XDisplay(Target)
        End If
        End Sub

        Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Call XDisplay(0)
        j = 0
        End Sub

        Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        Call XDisplay(total - 1)
        j = total - 1
        End Sub

        Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        If j = total - 1 Then
        Call XDisplay(0)
```

```
j = 0
Else
Call XDisplay(j + 1)
j = j + 1
End If
End Sub

Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
If j = 0 Then
Call XDisplay(total - 1)
j = total - 1
Else
Call XDisplay(j - 1)
j = j - 1
End If
End Sub

Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
Me.Close()
Form2.Show()
End Sub

Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
Call XSave(j)
Call Form10_Load(sender, e)

j = j - 1
If j = total - 1 Then
Call XDisplay(0)
Else
Call XDisplay(total - 1)
End If
Me.Close()
Form7.Show()
End Sub

Private Sub Button7_Click(sender As Object, e As EventArgs) Handles Button7.Click
Dim PricePerQuantity As Long
Dim k As Long
k = 1
If k Mod 2 = 1 Then
PricePerQuantity = ProdPrice(j) / ProdQuantity(j)
End If

If Button7.Text = "Modify" Then
Call XDisable()
Button7.Text = "Save"
Button7.Enabled = True
TextBox3.ReadOnly = False
Else
ModelNum(j) = TextBox1.Text
ProdName(j) = TextBox2.Text
ProdQuantity(j) = TextBox3.Text

ProdPrice(j) = CInt(TextBox3.Text) * PricePerQuantity
Call Xable()
Button7.Text = "Modify"
```

```
Call XSave(-1)
TextBox3.ReadOnly = True
End If
k = k + 1
End Sub

Private Sub Form10_Load(sender As Object, e As EventArgs) Handles MyBase.Load
FileOpen(1, "CartPerCustomer.csv", OpenMode.Input)
Do Until EOF(1)
ReDim Preserve ModelNum(0 To i)
ReDim Preserve ProdName(0 To i)
ReDim Preserve ProdQuantity(0 To i)
ReDim Preserve ProdPrice(0 To i)
Input(1, ModelNum(i))
Input(1, ProdName(i))
Input(1, ProdQuantity(i))
Input(1, ProdPrice(i))
i = i + 1
Loop
FileClose(1)
total = i
If ModelNum(0) = Nothing Then
Call XDisable()
Button9.Enabled = True
Else
Call XDisplay(0)
End If


End Sub
Private Sub XDisplay(ByVal A As Long)

TextBox1.Text = ModelNum(A)
TextBox2.Text = ProdName(A)
TextBox3.Text = ProdQuantity(A)
TextBox4.Text = ProdPrice(A)

End Sub

Private Sub Button9_Click(sender As Object, e As EventArgs) Handles Button9.Click
Me.Close()
Form7.Show()
End Sub

Private Sub XSave(ByVal A As Long)
Dim D As Long
FileOpen(5, "CartPerCustomer.csv", OpenMode.Output)
For D = 0 To total - 1
If D <> A Then
WriteLine(5, ModelNum(D), ProdName(D), ProdQuantity(D), ProdPrice(D))
End If
Next
FileClose(5)
End Sub
Private Sub XDisable()

Button1.Enabled = False
```

```
        Button2.Enabled = False
        Button3.Enabled = False
        Button4.Enabled = False
        Button5.Enabled = False
        Button6.Enabled = False
        Button7.Enabled = False
        Button8.Enabled = False
        Button9.Enabled = False
        'Button10.Enabled = False

        End Sub
        Private Sub Xable()

        Button1.Enabled = True
        Button2.Enabled = True
        Button3.Enabled = True
        Button4.Enabled = True
        Button5.Enabled = True
        Button6.Enabled = True
        Button7.Enabled = True
        Button8.Enabled = True
        Button9.Enabled = True
        '  Button10.Enabled = True

        End Sub
End Class

Public Class Form11
        Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        If RadioButton1.Checked = True Then
        Me.Close()
        Form17.Show()
        ElseIf RadioButton2.Checked = True Then
        Me.Close()
        Form12.Show()
        End If
        End Sub

        Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        Me.Close()
        Form1.Show()
        End Sub

        Private Sub Form11_Load(sender As Object, e As EventArgs) Handles MyBase.Load

        End Sub

        Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        Me.Close()
        form20.Show()
        End Sub
End Class

Public Class Form12
        Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        If RadioButton1.Checked = False And RadioButton2.Checked = False And RadioButton3.Checked = False And
RadioButton4.Checked = False Then
```

```
        MsgBox("Choose Appropriate Group")
        Exit Sub
        End If
        Select Case True
        Case RadioButton1.Checked
        Me.Close()
        Form13.Show()
        Case RadioButton2.Checked
        Me.Close()
        Form14.Show()
        Case RadioButton3.Checked
        Me.Close()
        Form15.Show()
        Case RadioButton4.Checked
        Me.Close()
        Form16.Show()
        End Select

        End Sub

        Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
        Me.Close()
        Form11.Show()
        End Sub

        Private Sub Form12_Load(sender As Object, e As EventArgs) Handles MyBase.Load

        End Sub
End Class

Public Class Form13
        Private totalManufacturingAssembly, k As Long
        Private Sub Form13_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the 'ItemsDBDataSet1.ManufacturingAssembly' table. You can move, or remove it, as
needed.
        Me.ManufacturingAssemblyTableAdapter1.Fill(Me.ItemsDBDataSet1.ManufacturingAssembly)

        totalManufacturingAssembly = ManufacturingAssemblyBindingSource1.Count
        k = 1
        End Sub

        Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        ManufacturingAssemblyBindingSource1.MoveFirst()
        End Sub

        Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        ManufacturingAssemblyBindingSource1.MoveLast()
        End Sub

        Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        Dim Xposition As Long
        Xposition = ManufacturingAssemblyBindingSource1.Position
        If Xposition = totalManufacturingAssembly - 1 Then
        ManufacturingAssemblyBindingSource1.MoveFirst()
        Else
        ManufacturingAssemblyBindingSource1.MoveNext()
        End If
```

```
        End Sub

        Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
        Dim Xposition As Long
        Xposition = ManufacturingAssemblyBindingSource1.Position
        If Xposition = 0 Then
        ManufacturingAssemblyBindingSource1.MoveLast()
        Else
        ManufacturingAssemblyBindingSource1.MovePrevious()
        End If
        End Sub

        Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
        Validate()
        ManufacturingAssemblyBindingSource1.EndEdit()
        Me.ManufacturingAssemblyTableAdapter1.Update(ItemsDBDataSet1.ManufacturingAssembly)
        End Sub

        Private Sub Button9_Click(sender As Object, e As EventArgs) Handles Button9.Click
        Me.Close()
        Form12.Show()
        End Sub

        Private Sub Button10_Click(sender As Object, e As EventArgs) Handles Button10.Click
        End
        End Sub

        Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
        Dim user, key As String
        Dim j, target As Long
        user = InputBox("Enter record number")
        target = -1
        For j = 0 To totalManufacturingAssembly - 1
        key = ItemsDBDataSet1.ManufacturingAssembly(j).ModelNumber
        If user = key Then
        target = j
        Exit For
        End If
        Next
        ManufacturingAssemblyBindingSource1.Position = j
        If target = -1 Then
        MsgBox("No such ID")
        ManufacturingAssemblyBindingSource1.MoveFirst()
        End If
        End Sub
End Class

Public Class Form14
        Private totalItemTransportation As Long
        Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
        Validate()
        ItemTransportationBindingSource.EndEdit() 'Validates the entry
        Me.ItemTransportationTableAdapter.Update(ItemsDBDataSet1.ItemTransportation)
        End Sub

        Private Sub Form14_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the 'ItemsDBDataSet1.ItemTransportation' table. You can move, or remove it, as needed.
```

```
Me.ItemTransportationTableAdapter.Fill(Me.ItemsDBDataSet1.ItemTransportation)
totalItemTransportation = ItemTransportationBindingSource.Count()
End Sub

Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
Dim Xposition As Long
Xposition = ItemTransportationBindingSource.Position
If Xposition = totalItemTransportation - 1 Then
ItemTransportationBindingSource.MoveFirst()
Else
ItemTransportationBindingSource.MoveNext()
End If
End Sub

Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
Dim Xposition As Long
Xposition = ItemTransportationBindingSource.Position
If Xposition = 0 Then
ItemTransportationBindingSource.MoveLast()
Else
ItemTransportationBindingSource.MovePrevious()
End If
End Sub

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
ItemTransportationBindingSource.MoveFirst()
End Sub

Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
ItemTransportationBindingSource.MoveLast()
End Sub
Private Sub XDisable()
Select Case False
Case Button1.Enabled
Case Button2.Enabled
Case Button3.Enabled
Case Button4.Enabled
Case Button5.Enabled
Case Button6.Enabled
Case Button7.Enabled
Case Button8.Enabled
' Case Button9.Enabled
End Select
End Sub
Private Sub XAble()
Select Case True
Case Button1.Enabled
Case Button2.Enabled
Case Button3.Enabled
Case Button4.Enabled
Case Button5.Enabled
Case Button6.Enabled
Case Button7.Enabled
Case Button8.Enabled
' Case Button9.Enabled
End Select
End Sub
```

```
        Private Sub Button8_Click(sender As Object, e As EventArgs) Handles Button8.Click
        ItemTransportationBindingSource.RemoveCurrent()
        ItemTransportationTableAdapter.Update(ItemsDBDataSet1.ItemTransportation)
        totalItemTransportation = totalItemTransportation - 1
        End Sub

        Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
        Dim user, key As String
        Dim j, target As Long
        user = InputBox("Enter record number")
        target = -1
        For j = 0 To totalItemTransportation - 1
        key = ItemsDBDataSet1.ItemTransportation(j).ModelNumber
        If user = key Then
        target = j
        Exit For
        End If
        Next
        ItemTransportationBindingSource.Position = j
        If target = -1 Then
        MsgBox("No such ID")
        ItemTransportationBindingSource.MoveFirst()
        End If
        End Sub

        Private Sub Button9_Click(sender As Object, e As EventArgs) Handles Button9.Click
        Me.Close()
        Form12.Show()
        End Sub

        Private Sub Button7_Click(sender As Object, e As EventArgs) Handles Button7.Click
        If Button7.Text = "Add" Then
        ItemTransportationBindingSource.AddNew()
        Call XDisable()
        Button7.Enabled = True
        Button7.Text = "Save"
        Else
        ItemTransportationBindingSource.EndEdit()
        ItemTransportationTableAdapter.Update(Me.ItemsDBDataSet1.ItemTransportation)
        Call XAble()
        Button7.Text = "Add"
        totalItemTransportation = totalItemTransportation + 1
        End If

        End Sub

        Private Sub Button10_Click(sender As Object, e As EventArgs) Handles Button10.Click
        End
        End Sub
End Class

Public Class Form15
        Private totalAssitedInventoryHandling As Long
        Private Sub Form15_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the 'ItemsDBDataSet1.AssistedInventoryHandling' table. You can move, or remove it, as
needed.
```

```
Me.AssistedInventoryHandlingTableAdapter1.Fill(Me.ItemsDBDataSet1.AssistedInventoryHandling)

totalAssitedInventoryHandling = AssistedInventoryHandlingBindingSource1.Count()
End Sub

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
AssistedInventoryHandlingBindingSource1.MoveFirst()
End Sub

Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
AssistedInventoryHandlingBindingSource1.MoveLast()
End Sub

Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
Dim Xposition As Long
Xposition = AssistedInventoryHandlingBindingSource1.Position
If Xposition = totalAssitedInventoryHandling - 1 Then
AssistedInventoryHandlingBindingSource1.MoveFirst()
Else
AssistedInventoryHandlingBindingSource1.MoveNext()
End If
End Sub

Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
Dim Xposition As Long
Xposition = AssistedInventoryHandlingBindingSource1.Position
If Xposition = 0 Then
AssistedInventoryHandlingBindingSource1.MoveLast()
Else
AssistedInventoryHandlingBindingSource1.MovePrevious()
End If
End Sub
Private Sub XDisable()
Select Case False
Case Button1.Enabled
Case Button2.Enabled
Case Button3.Enabled
Case Button4.Enabled
Case Button5.Enabled
Case Button6.Enabled
Case Button7.Enabled
Case Button8.Enabled
' Case Button9.Enabled
End Select
End Sub
Private Sub XAble()
Select Case True
Case Button1.Enabled
Case Button2.Enabled
Case Button3.Enabled
Case Button4.Enabled
Case Button5.Enabled
Case Button6.Enabled
Case Button7.Enabled
Case Button8.Enabled
' Case Button9.Enabled
End Select
```

```
End Sub

Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
Dim user, key As String
Dim j, target As Long
user = InputBox("Enter record number")
target = -1
For j = 0 To totalAssitedInventoryHandling - 1
key = ItemsDBDataSet1.AssistedInventoryHandling(j).ModelNumber
If user = key Then
target = j
Exit For
End If
Next
AssistedInventoryHandlingBindingSource1.Position = j
If target = -1 Then
MsgBox("No such ID")
AssistedInventoryHandlingBindingSource1.MoveFirst()
End If
End Sub

Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
Validate()
AssistedInventoryHandlingBindingSource1.EndEdit() 'Validates the entry
Me.AssistedInventoryHandlingTableAdapter1.Update(ItemsDBDataSet1.AssistedInventoryHandling)
End Sub

Private Sub Button7_Click(sender As Object, e As EventArgs) Handles Button7.Click
If Button7.Text = "Add" Then
AssistedInventoryHandlingBindingSource1.AddNew()
Call XDisable()
Button7.Enabled = True
Button7.Text = "Save"
Else
AssistedInventoryHandlingBindingSource1.EndEdit()
AssistedInventoryHandlingTableAdapter1.Update(Me.ItemsDBDataSet1.AssistedInventoryHandling)
Call XAble()
Button7.Text = "Add"
totalAssitedInventoryHandling = totalAssitedInventoryHandling + 1
End If
End Sub

Private Sub Button8_Click(sender As Object, e As EventArgs) Handles Button8.Click
AssistedInventoryHandlingBindingSource1.RemoveCurrent()
AssistedInventoryHandlingTableAdapter1.Update(ItemsDBDataSet1.AssistedInventoryHandling)
totalAssitedInventoryHandling = totalAssitedInventoryHandling - 1
End Sub

Private Sub Button9_Click(sender As Object, e As EventArgs) Handles Button9.Click
Me.Close()
Form12.Show()
End Sub

Private Sub Button10_Click(sender As Object, e As EventArgs) Handles Button10.Click
End
End Sub
```

```
        Private Sub Label5_Click(sender As Object, e As EventArgs)

        End Sub
End Class

Public Class Form16
        Private totalManualInventoryHandling As Long
        Private Sub Form16_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the 'ItemsDBDataSet1.ManualInventoryHandling' table. You can move, or remove it, as
needed.
        Me.ManualInventoryHandlingTableAdapter1.Fill(Me.ItemsDBDataSet1.ManualInventoryHandling)

        totalManualInventoryHandling = ManualInventoryHandlingBindingSource1.Count
        End Sub

        Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        ManualInventoryHandlingBindingSource1.MoveFirst()
        End Sub

        Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        ManualInventoryHandlingBindingSource1.MoveLast()
        End Sub

        Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
        Dim Xposition As Long
        Xposition = ManualInventoryHandlingBindingSource1.Position
        If Xposition = totalManualInventoryHandling - 1 Then
        ManualInventoryHandlingBindingSource1.MoveFirst()
        Else
        ManualInventoryHandlingBindingSource1.MoveNext()
        End If
        End Sub

        Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
        Dim Xposition As Long
        Xposition = ManualInventoryHandlingBindingSource1.Position
        If Xposition = 0 Then
        ManualInventoryHandlingBindingSource1.MoveLast()
        Else
        ManualInventoryHandlingBindingSource1.MovePrevious()
        End If
        End Sub
        Private Sub XDisable()
        Select Case False
        Case Button1.Enabled
        Case Button2.Enabled
        Case Button3.Enabled
        Case Button4.Enabled
        Case Button5.Enabled
        Case Button6.Enabled
        Case Button7.Enabled
        Case Button8.Enabled
        ' Case Button9.Enabled
        End Select
        End Sub
        Private Sub XAble()
        Select Case True
```

```
Case Button1.Enabled
Case Button2.Enabled
Case Button3.Enabled
Case Button4.Enabled
Case Button5.Enabled
Case Button6.Enabled
Case Button7.Enabled
Case Button8.Enabled
' Case Button9.Enabled
End Select
End Sub

Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
Validate()
ManualInventoryHandlingBindingSource1.EndEdit() 'Validates the entry
Me.ManualInventoryHandlingTableAdapter1.Update(ItemsDBDataSet1.ManualInventoryHandling)
End Sub

Private Sub Button7_Click(sender As Object, e As EventArgs) Handles Button7.Click
If Button7.Text = "Add" Then
ManualInventoryHandlingBindingSource1.AddNew()
Call XDisable()
Button7.Enabled = True
Button7.Text = "Save"
Else
ManualInventoryHandlingBindingSource1.EndEdit()
ManualInventoryHandlingTableAdapter1.Update(Me.ItemsDBDataSet1.ManualInventoryHandling)
Call XAble()
Button7.Text = "Add"
totalManualInventoryHandling = totalManualInventoryHandling + 1
End If
End Sub

Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
Dim user, key As String
Dim j, target As Long
user = InputBox("Enter record number")
target = -1
For j = 0 To totalManualInventoryHandling - 1
key = ItemsDBDataSet1.ManualInventoryHandling(j).ModelNumber
If user = key Then
target = j
Exit For
End If
Next
ManualInventoryHandlingBindingSource1.Position = j
If target = -1 Then
MsgBox("No such ID")
ManualInventoryHandlingBindingSource1.MoveFirst()
End If
End Sub

Private Sub Button8_Click(sender As Object, e As EventArgs) Handles Button8.Click
ManualInventoryHandlingBindingSource1.RemoveCurrent()
ManualInventoryHandlingTableAdapter1.Update(ItemsDBDataSet1.ManualInventoryHandling)
totalManualInventoryHandling = totalManualInventoryHandling - 1
End Sub
```

```
        Private Sub Button9_Click(sender As Object, e As EventArgs) Handles Button9.Click
        Me.Close()
        Form12.Show()
        End Sub

        Private Sub Button10_Click(sender As Object, e As EventArgs) Handles Button10.Click
        End
        End Sub
End Class

Public Class Form17
        Private ModelNum() As String
        Private ProdName() As String
        Private ProdQuantity() As String
        Private ProdPrice() As String
        Private i, j, k, total As Long
        Private CartDisplay As String
        Private Sub Form17_Load(sender As Object, e As EventArgs) Handles MyBase.Load

        Dim a As String
        a = CurDir()
        i = 0

        If System.IO.File.Exists(a & "\AllSales.csv") Then
        If System.IO.File.ReadAllText(a & "\AllSales.csv").Length = 0 Then
        MsgBox("All Sales file is empty")
        Call XDisable()
        Button9.Enabled = True
        Button10.Enabled = True
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""

        Exit Sub
        Else
        FileOpen(1, "AllSales.csv", OpenMode.Input)
        Do Until EOF(1)
        ReDim Preserve ModelNum(0 To i)
        ReDim Preserve ProdName(0 To i)
        ReDim Preserve ProdQuantity(0 To i)
        ReDim Preserve ProdPrice(0 To i)
        Input(1, ModelNum(i))
        Input(1, ProdName(i))
        Input(1, ProdQuantity(i))
        Input(1, ProdPrice(i))
        i = i + 1
        Loop
        FileClose(1)
        total = i
        Call XDisplay(0)
        End If
        Else
        MsgBox("No sales have been made")
        Call XDisable()
        Button9.Enabled = True
```

```
Button10.Enabled = True
Exit Sub
End If


End Sub
Private Sub XDisable()

Button1.Enabled = False
Button2.Enabled = False
Button3.Enabled = False
Button4.Enabled = False
Button5.Enabled = False
Button6.Enabled = False
Button7.Enabled = False
Button8.Enabled = False
Button9.Enabled = False
Button10.Enabled = False

End Sub

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
Call XDisplay(0)
j = 0
End Sub

Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
Call XDisplay(total - 1)
j = total - 1
End Sub

Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
If j = total - 1 Then
Call XDisplay(0)
j = 0
Else
Call XDisplay(j + 1)
j = j + 1
End If
End Sub

Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
If j = 0 Then
Call XDisplay(total - 1)
j = total - 1
Else
Call XDisplay(j - 1)
j = j - 1
End If
End Sub

Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
If Button5.Text = "Modify" Then
Call XDisable()
Button5.Text = "Save"
Button5.Enabled = True
Else
```

```
ModelNum(j) = TextBox1.Text
ProdName(j) = TextBox2.Text
ProdQuantity(j) = TextBox3.Text
ProdPrice(j) = TextBox4.Text
Call Xable()
Button5.Text = "Modify"
Call XSave(-1)

End If

End Sub

Private Sub Xable()

Button1.Enabled = True
Button2.Enabled = True
Button3.Enabled = True
Button4.Enabled = True
Button5.Enabled = True
Button6.Enabled = True
Button7.Enabled = True
Button8.Enabled = True
Button9.Enabled = True
Button10.Enabled = True

End Sub

Private Sub Button7_Click(sender As Object, e As EventArgs) Handles Button7.Click
If Button7.Text = "Add" Then
Call XDisable()
Button7.Enabled = True
Button7.Text = "Save"
Else
FileOpen(50, "AllSales.csv", OpenMode.Append)
WriteLine(50, TextBox1.Text, TextBox2.Text, TextBox3.Text, TextBox4.Text)
FileClose(50)
Call Form17_Load(sender, e)
Call XDisplay(total - 1)
Call Xable()
Button7.Text = "Add"
j = j + 1
End If

End Sub

Private Sub Button8_Click(sender As Object, e As EventArgs) Handles Button8.Click
Call XSave(j)
Call Form17_Load(sender, e)
If i <= 0 Then
Call XDisable()
Button9.Enabled = True
Button10.Enabled = True
TextBox1.Text = ""
TextBox2.Text = ""
TextBox3.Text = ""
TextBox4.Text = ""
Exit Sub
```

```
Else
j = j - 1
If j = total - 1 Then
Call XDisplay(0)
j = 0
Else
Call XDisplay(total - 1)
j = total - 1
End If
End If
End Sub

Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
Dim X As String
Dim Target, F As Long

X = InputBox("Enter a record number")
Target = -1
For F = 0 To total - 1
If X = ModelNum(F) Then
Target = F
Exit For
End If
Next
If Target = -1 Then
MsgBox("No such Model Number")
Else
Call XDisplay(Target)
End If
End Sub

Private Sub Button9_Click(sender As Object, e As EventArgs) Handles Button9.Click
Dim h As Long
If i <= 0 Then
Me.Close()
Form11.Show()
Exit Sub
Else
FileOpen(15, "SupplierList.csv", OpenMode.Output)
For h = 0 To total - 1
WriteLine(15, ModelNum(h), ProdName(h), ProdQuantity(h), ProdPrice(h))
Next
FileClose(15)
End If

Me.Close()
Form11.Show()
End Sub

Private Sub Button10_Click(sender As Object, e As EventArgs) Handles Button10.Click
Dim h As Long
If i <= 0 Then
End
Exit Sub
Else
FileOpen(15, "SupplierList.csv", OpenMode.Output)
For h = 0 To total - 1
```

```
        WriteLine(15, ModelNum(h), ProdName(h), ProdQuantity(h), ProdPrice(h))
        Next
        FileClose(15)
        End If

        End
        End Sub

        Private Sub XDisplay(ByVal A As Long)

        TextBox1.Text = ModelNum(A)
        TextBox2.Text = ProdName(A)
        TextBox3.Text = ProdQuantity(A)
        TextBox4.Text = ProdPrice(A)

        End Sub
        Private Sub XSave(ByVal A As Long)
        Dim D As Long
        FileOpen(5, "AllSales.csv", OpenMode.Output)
        For D = 0 To total - 1
        If D <> A Then
        WriteLine(5, ModelNum(D), ProdName(D), ProdQuantity(D), ProdPrice(D))
        End If
        Next
        FileClose(5)
        End Sub
End Class

Public Class Form18
        Private SModelNum() As String
        Private SProdName() As String
        Private SProdQuantity() As String
        Private SProdPrice() As String
        Private i, j, k, z, total As Long

        Private Sub Form18_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Dim a As String
        a = CurDir()
        z = 0
        If System.IO.File.Exists(a & "\SupplierList.csv") Then
        If System.IO.File.ReadAllText(a & "\SupplierList.csv").Length = 0 Then
        MsgBox("Supplier file is empty")
        Call XDisable()
        Button6.Enabled = True
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""

        Exit Sub
        Else
        FileOpen(5, "SupplierList.csv", OpenMode.Input)
        Do Until EOF(5)
        ReDim Preserve SModelNum(0 To z)
        ReDim Preserve SProdName(0 To z)
        ReDim Preserve SProdQuantity(0 To z)
        ReDim Preserve SProdPrice(0 To z)
        Input(5, SModelNum(z))
```

```
Input(5, SProdName(z))
Input(5, SProdQuantity(z))
Input(5, SProdPrice(z))
z = z + 1
Loop
total = z
Call XDisplay(0)
End If
Else
MsgBox("No items in the suppliers list")
Call XDisable()
Button6.Enabled = True
Exit Sub
End If

FileClose(5)
End Sub

Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
Call XDisplay(0)
j = 0
End Sub

Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
If j = total - 1 Then
Call XDisplay(0)
j = 0
Else
Call XDisplay(j + 1)
j = j + 1
End If
End Sub

Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
If j = 0 Then
Call XDisplay(total - 1)
j = total - 1
Else
Call XDisplay(j - 1)
j = j - 1
End If
End Sub

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
Dim K As Long
Dim Num() As String
Dim Name() As String
Dim Quantity() As String
Dim SentToShipperDisplay As String
FileOpen(2, "ItemsSentToShipper.csv", OpenMode.Append)
WriteLine(2, TextBox1.Text, TextBox2.Text, TextBox3.Text)
FileClose(2)
FileOpen(4, "ItemsSentToShipper.csv", OpenMode.Input)
Do Until EOF(4)
ReDim Preserve Num(0 To K)
ReDim Preserve Name(0 To K)
ReDim Preserve Quantity(0 To K)
```

```
Input(4, Num(K))
Input(4, Name(K))
Input(4, Quantity(K))
SentToShipperDisplay = SentToShipperDisplay & Num(K) & vbTab & Name(K) & vbTab & Quantity(K) & vbNewLine
K = K + 1
Loop
TextBox4.Text = SentToShipperDisplay
FileClose(4)

Call XSave(j)
Call Form18_Load(sender, e)
If z <= 0 Then
Call XDisable()
Button6.Enabled = True
Button7.Enabled = True
TextBox1.Text = ""
TextBox2.Text = ""
TextBox3.Text = ""
'TextBox4.Text = ""
Exit Sub
Else
j = j - 1
If j = total - 1 Then
Call XDisplay(0)
j = 0
Else
Call XDisplay(total - 1)
j = total - 1
End If
End If

' Call Form17_Load(sender, e)
'End If
End Sub

Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
Call XDisplay(total - 1)
j = total - 1
End Sub

Private Sub Button7_Click(sender As Object, e As EventArgs) Handles Button7.Click
Me.Close()
form21.Show()
End Sub

Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
Me.Close()
Form1.Show()
End Sub

Private Sub XDisplay(ByVal A As Long)
TextBox1.Text = SModelNum(A)
TextBox2.Text = SProdName(A)
TextBox3.Text = SProdQuantity(A)
End Sub
Private Sub XSave(ByVal A As Long)
Dim D As Long
```

```
        FileOpen(5, "SupplierList.csv", OpenMode.Output)
        For D = 0 To total - 1
        If D <> A Then
        WriteLine(5, SModelNum(D), SProdName(D), SProdQuantity(D), SProdPrice(D))
        End If
        Next
        FileClose(5)
        End Sub
        Private Sub XDisable()
        Button1.Enabled = False
        Button2.Enabled = False
        Button3.Enabled = False
        Button4.Enabled = False
        Button5.Enabled = False
        Button6.Enabled = False
        'Button7.Enabled = False
        'Button8.Enabled = False
        'Button9.Enabled = False
        'Button10.Enabled = False
        End Sub
End Class

Public Class Form19
        Private ModelNum() As String
        Private ProdName() As String
        Private ProdQuantity() As String
        Private ProdPrice() As String
        Private i, j, total As Long

        Private Sub Form19_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        i = 0
        Dim a As String
        a = CurDir()
        If System.IO.File.Exists(a & "\ItemsSentToShipper.csv") Then
        If System.IO.File.ReadAllText(a & "\ItemsSentToShipper.csv").Length = 0 Then
        MsgBox("Supplier has not sent items to the Shipper")
        Call XDisable()
        Button6.Enabled = True
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        Exit Sub
        Else
        FileOpen(1, "ItemsSentToShipper.csv", OpenMode.Input)
        Do Until EOF(1)
        ReDim Preserve ModelNum(0 To i)
        ReDim Preserve ProdName(0 To i)
        ReDim Preserve ProdQuantity(0 To i)
        Input(1, ModelNum(i))
        Input(1, ProdName(i))
        Input(1, ProdQuantity(i))
        i = i + 1
        Loop
        total = i
        FileClose(1)
        Call XDisplay(0)
        End If
```

```
Else
MsgBox("Supplier has not sent items to the Shipper")
Call XDisable()
Button6.Enabled = True
Exit Sub
End If
End Sub

Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
If j = total - 1 Then
Call XDisplay(0)
j = 0
Else
Call XDisplay(j + 1)
j = j + 1
End If
End Sub

Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
If j = 0 Then
Call XDisplay(total - 1)
j = total - 1
Else
Call XDisplay(j - 1)
j = j - 1
End If
End Sub

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
Dim K As Long
Dim Num() As String
Dim Name() As String
Dim Quantity() As String
Dim ShippedItems As String
FileOpen(2, "ShippedItems.csv", OpenMode.Append)
WriteLine(2, TextBox1.Text, TextBox2.Text, TextBox3.Text)
FileClose(2)
FileOpen(4, "ShippedItems.csv", OpenMode.Input)
Do Until EOF(4)
ReDim Preserve Num(0 To K)
ReDim Preserve Name(0 To K)
ReDim Preserve Quantity(0 To K)
Input(4, Num(K))
Input(4, Name(K))
Input(4, Quantity(K))
ShippedItems = ShippedItems & Num(K) & vbTab & Name(K) & vbTab & Quantity(K) & vbNewLine
K = K + 1
Loop
TextBox4.Text = ShippedItems
FileClose(4)

Call XSave(j)
Call Form19_Load(sender, e)
If i <= 0 Then
Call XDisable()
Button6.Enabled = True
Button7.Enabled = True
```

```
TextBox1.Text = ""
TextBox2.Text = ""
TextBox3.Text = ""
'TextBox4.Text = ""
Exit Sub
Else
j = j - 1
If j = total - 1 Then
Call XDisplay(0)
j = 0
Else
Call XDisplay(total - 1)
j = total - 1
End If
End If

End Sub
Private Sub XSave(ByVal A As Long)
Dim D As Long
FileOpen(5, "ItemsSentToShipper.csv", OpenMode.Output)
For D = 0 To total - 1
If D <> A Then
WriteLine(5, ModelNum(D), ProdName(D), ProdQuantity(D))
End If
Next
FileClose(5)
End Sub

Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
Call XDisplay(0)
j = 0
End Sub

Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
Call XDisplay(total - 1)
j = total - 1
End Sub

Private Sub XDisplay(ByVal A As Long)
TextBox1.Text = ModelNum(A)
TextBox2.Text = ProdName(A)
TextBox3.Text = ProdQuantity(A)
End Sub

Private Sub Button7_Click(sender As Object, e As EventArgs) Handles Button7.Click
Me.Close()
Form22.Show()
End Sub

Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
Me.Close()
Form1.Show()
End Sub
Private Sub XDisable()
Button1.Enabled = False
Button2.Enabled = False
Button3.Enabled = False
```

```
        Button4.Enabled = False
        Button5.Enabled = False
        Button6.Enabled = False
        End Sub
End Class

Public Class Form20
        Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        End
        End Sub

        Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Me.Close()
        Form1.Show()
        End Sub

        Private Sub Form20_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        End Sub
End Class

Public Class Form21
        Private Sub TextBox1_TextChanged(sender As Object, e As EventArgs) Handles TextBox1.TextChanged

        End Sub

        Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Me.Close()
        Form1.Show()
        End Sub

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

        Private Sub Form21_Load(sender As Object, e As EventArgs) Handles MyBase.Load

        End Sub
End Class

Public Class Form22
        Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Me.Close()
        Form1.Show()

        End Sub

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

        Private Sub Form22_Load(sender As Object, e As EventArgs) Handles MyBase.Load

        End Sub
End Class
```