



City University of Hong Kong

MS3111 Quantitative Business Analysis with Visual Basic for Application

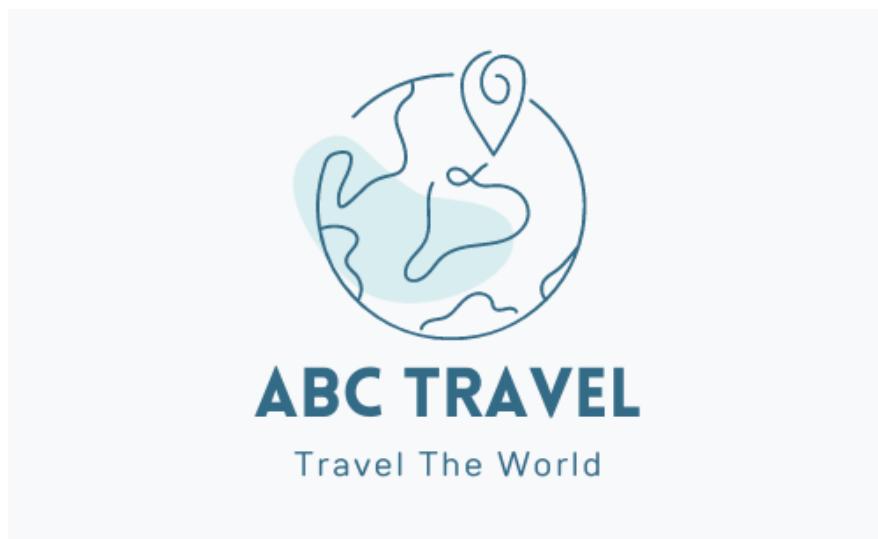
(Semester B, 2021/2022)

Project Final Report

Instructor: Dr. H.K. Yuen

Group: 19

Student Name	SID
Hau Chak Kuen	56209539
Chan Tai Wing	55504101
Ho Yin	56624194
Ng Ching Wai	56459370



Introduction

ABC Travel Limited (ABC Travel) is a start-up travel company in Hong Kong. It aims to provide wonderful travel experiences to all customers. ABC Travel needed to store and handle a large volume of data, in which datasets are stored in numerous excel workbooks. Since ABC Travel did not have efficient programming tools, it requires a lot of manual work when dealing with the data manipulation in excel. It greatly reduces the operation efficiency and wastes a lot of human effort to compute the sales performance, business cost, as well as the net profit. Sometimes, the company makes a mistake in measuring the sales and cost data, it will lead to the wrong salary calculation, as well as affecting the effectiveness of business strategy. As a result, on behalf of the management team of ABC Travel, we would like to use 2019 Microsoft Excel Visual Basic Application (VBA) to help with the business operation. We are going to write a program in order to automate the calculation processes of various business datasets. The final result can be highly updated and accurate, and therefore improving the effectiveness of strategic planning.

Objectives

1. Add and store customer information
2. Calculate the total sales and operational cost in order to generate net profit
3. Measure the salesperson performance and generate chart for comparison
4. Calculate the salary expense for each salesperson
5. Generate report the monthly report of total gross profit

Assumptions:

Assumptions are set to improve the sharpness of report

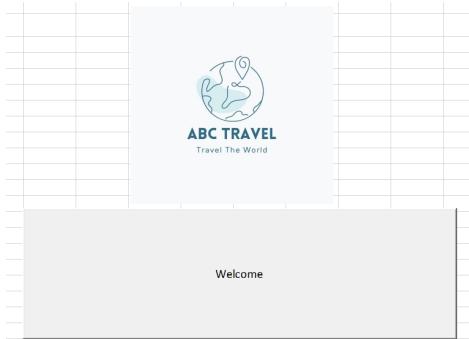
1. All the data are collected and calculated per month
2. The monetary unit is defaulted as Hong Kong dollar(HKD)
3. Each package is unique and it would appeared once only in current stage

VBA Application

<u>Excel Workbook:</u>	<u>Description:</u>
	For running the application
Customer_information.xlsx	Customer_Information worksheet: Record the customer's personal information and the package purchased.
	Tour_detail worksheet: Record the package information and the price
Revenue.xlsx	Revenue worksheet: Record the sales performance and cost of each package
salesperson.xlsx	salesperson worksheet: Record the personal information and salary of each salesperson
performance.xlsx	performance worksheet: Record the sales performance of each salesperson

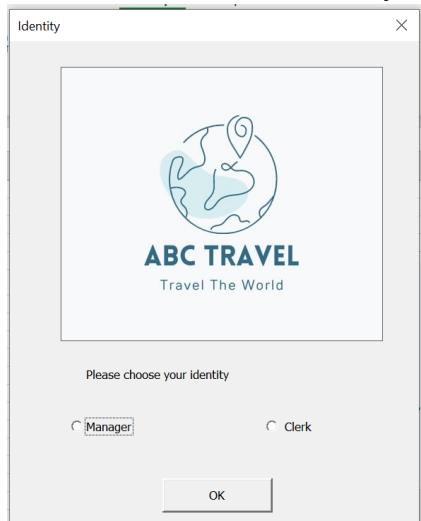
Step 1: Access the login page

Click the CommandButton “Welcome” to login the system. (Fig. 0)



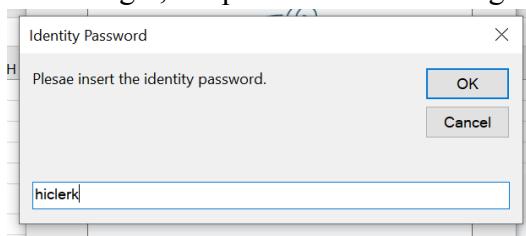
Step 2: Choose the identity

After clicking the “Welcome” button, the userform “Identity” (Login_Page) will appear. Employees can choose the OptionButtons of their identity based on their job title, “Optmanager” is for manager while “optclerk” is for clerk. Then they need to click the “Ok” button to confirm their identity. (Fig. 1)

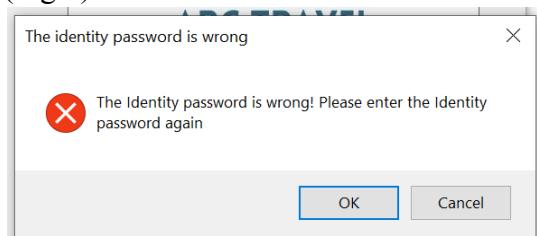


Step 3: Enter the login password

When the user is a clerk, they need to insert the password “hiclerk” in the InputBox “Identity Password” to login the system. The Clerk_Page will be shown if the password is correct. For the manager, the password is “himanager”, and it will direct to the Manager_Page. (Fig.1)



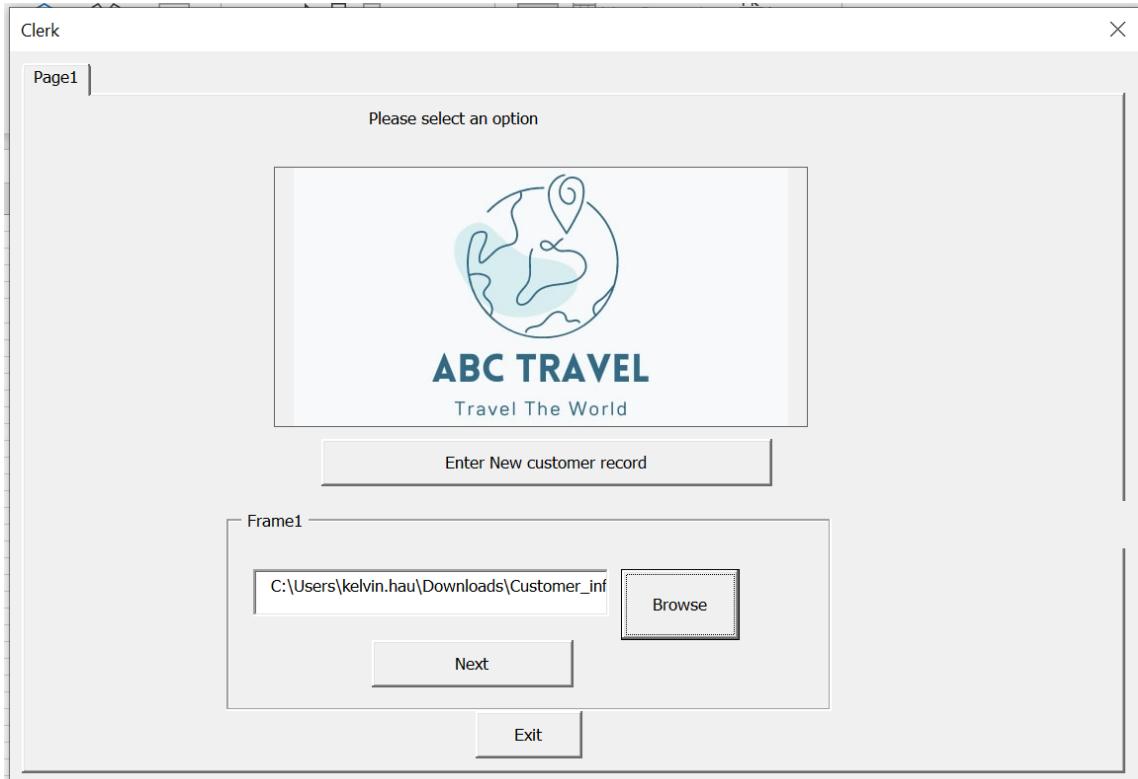
If the password is incorrect, a message box will appear to prompt the wrong password. (Fig.1)



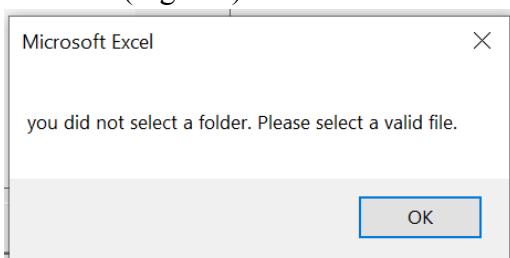
For Clerk:

Step 4: Choose the customer information workbook

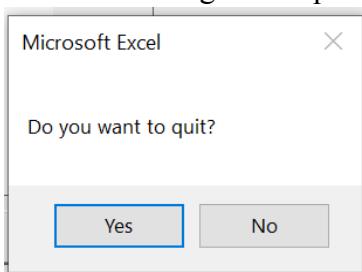
Below is the initial page of the Clerk_Page. Once the “Enter New customer record” button (Commandbutton1) is clicked, the browse frame will be visible (Fig. 2.1), and the clerk can click the “Browse” button (cmdbrowser) to browse the Customer_information.xlsx. The path of the file will be printed in the textbox ((Fig. 2.1)), and the clerk can click the “Next” button (CMBNEXT) to open the file and move to Page 2 (Fig. 2.3).



If the clerk cancels the browse function, a message box will appear to ask the user to choose a valid file (Fig. 2.2).



If the clerk clicks the “Exit” button (btnexit), then a message box will appear to confirm the exit of the userform. when the user clicks “Yes”, the system will unload the Clerk_Page and return to the stage of Step 1 (Fig. 2.4).

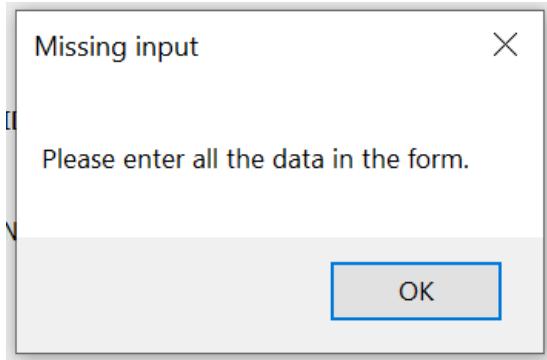


When the clerk wants to create another record by clicking the “Yes” button, the Page 1 will be initialized and the Customer_ID will be updated (Fig. 2.6). (From A101 to A102)

Clerk

The screenshot shows a Windows application window titled "Clerk". At the top, there are two tabs: "Page1" and "Page2", with "Page1" being the active tab. Below the tabs, there is a section labeled "Customer_ID" with the value "A102" highlighted in yellow. To the right of this is a "Name" field containing a blank white rectangle. Further right is an "Age" field with a numeric input box showing "18" and a scroll bar. There are also left and right arrow buttons next to the scroll bar.

If the “Save” button is clicked and not all the data are filled, a message box will appear to prompt the clerk to insert all data fields (Fig.2.6).

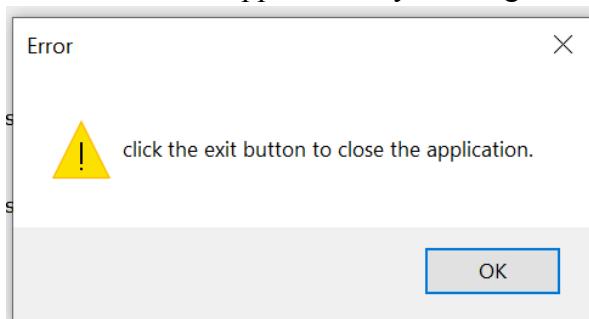


If the “Clear” button (btnClear) is clicked (Fig. 2.7), all the data in the above data fields will be cleared, and the pointer will go to the text box of customer name (txtname).

If the “Quit” button (btnQuit) is clicked (Fig. 2.8), the Clerk_Page will unload and go back to Step 1.

If the “Homepage” button is clicked (Fig. 2.9), the system will direct to Page 1 in Step 4.

If the close button in the top right hand corner is clicked, a message box will appear to ask the clerk to close the application by clicking the “Quit” button.



For manager

Step 6: Login to Manager page

If the user is a manager and enters the correct password to login, the system will direct to a page called “Manager” (ManagerAction) under the userform (Manager_Page) like below (Fig. 1).

There are 3 big functions which can be performed by the program , which are checking records , creating charts and calculating salaries.

The screenshot shows the 'Manager' userform. At the top left is a 'Manager' tab and a close button. Below it is a section titled 'Select a option' with three radio buttons: 'Bring Record' (selected), 'Create table and chart', and 'Employee salary calculator'. To the right of this is a 'Exit' button. Below these are four sections, each with a label, a 'Must select' indicator, a text input field, and a 'Browse' button:

- Please select workbook to browse the customer records: Must select
- Please select workbook to browse the revenues and cost record:
- Please select workbook to browse the employee record/calculate the employee
- Please select workbook to browse the Performance record:

At the bottom center is an 'OK' button.

Step 7: View customer record

When the manager clicks “Bring Record” (opttable), Frame 1 and Checkbox “Customer Record” (chkCustomer) will be enabled (Fig. 3.16), then the Frame2 and browse button (btnBrowseCustomer) will be enabled and the manager can browse “Customer_information.xlsx” (Fig. 3.1). And the manager needs to follow the instructions and browse the correct data file in order to process the next page.

The screenshot shows the 'Manager' userform after selecting 'Bring Record'. The 'Customer Record' checkbox (chkCustomer) is checked. The 'Customer information.xlsx' file path is entered in the first 'Please select workbook to browse the customer records:' field, and the 'Browse' button for this field is highlighted. The other fields and browse buttons remain the same as in Fig. 1.

The manager can click the “OK” button (btnconfirm) (Fig.3.5, 3.6) in order to move to the page “Customer Record” (Customer_Information). When the manager selects or enters the customer ID in the “Customer ID” combo box (cmbCustomerID1) (Fig. 3.7), all the customer’s basic information like name, gender, age, date of purchase, and the purchased package will be displayed on this page. The manager can click the “Clear” button (btnClear1) in order to clear all the displayed customer information (Fig. 3.8).

The manager can go back to the main page “Manager” (ManagerAction) in Step 6 by clicking the “Home page” button (btnHomePage1) (Fig. 3.17) or quit the program to Step 1 by clicking the “Exit” button (btnExit1) (Fig. 3.18).

Manager

Record |

ABC TRAVEL
Travel The World

Customer Record |

CustomerID	A001		
Name	AU HO YAN	Date	4/4/2022
Gender	F	PackageID	J001
Age	25		

Clear

Exit

Home page

Step 8: View salesperson record

The manager should go back to the main page “Manager” (ManagerAction) in Step 6, and click the “Bring Record” (opttable) and “Employee” (chkSalesperson). Then the browse buttons for “salesperson.xlsx” (Fig. 3.3)and “performance.xlsx” will be enabled. The manager needs to browse the two files. The code for two browse buttons can refer to Figure 3.3 and Figure 3.4. Then press “OK”.

After pressing “OK”, the manager can go to the page “salesperson” (salesperson_record). The manager can check the details of a salesperson by their ID. Manager can select a “Salesperson ID” (cmbstaffid3) and the details of the corresponding salesperson will be displayed (Fig. 3.11).

Manager

Record |

ABC TRAVEL
Travel The World

Customer Record | salesperson |

Salesperson ID	E01
Salesperson name	Stephen Chan
Gender	M
Title	Senior

Clear

Exit

Home page

Step 9: View revenue and expense

The manager should go back to the main page “Manager” (ManagerAction) in Step 6, and click the “Bring Record” (opttable) and “Revenue and cost” (chkRevenue). Then the browse buttons for “Customer_information.xlsx ” (Fig. 3.1) and “Revenue.xlsx” (Fig. 3.2) will be enabled. The manager needs to browse the two files. Then press “OK” (Fig. 3.5) and the system will direct to the page “sales and expense calculator” (Salesexpense).

In this page, the manager can check the sales details of different packages. The manager can choose more than one Package ID from the ListBox (lbxPackageID2) and then click “OK” (btnOk2), then the Frame 3 will be visible and display the corresponding details like sales revenue, expense, gross profit, and gross profit margin. The “OK” (Fig. 3.9) button will be enabled once the package ID is clicked.

There are some functions that make the manager use the form more efficiently. If “choose all” button (btnChooseall2) is clicked (Fig. 3.10), all the package ID from the list will be selected. If “Clear” is clicked (Fig. 3.11), all the selected package ID will be cleared.

Manager X

Record |


ABC TRAVEL
Travel The World

Customer Record | sales and expense calculator | salesperson |

Package ID

- C001
- C002
- C003
- T001
- T002
- K001
- K002

Frame3

Sales Revenue

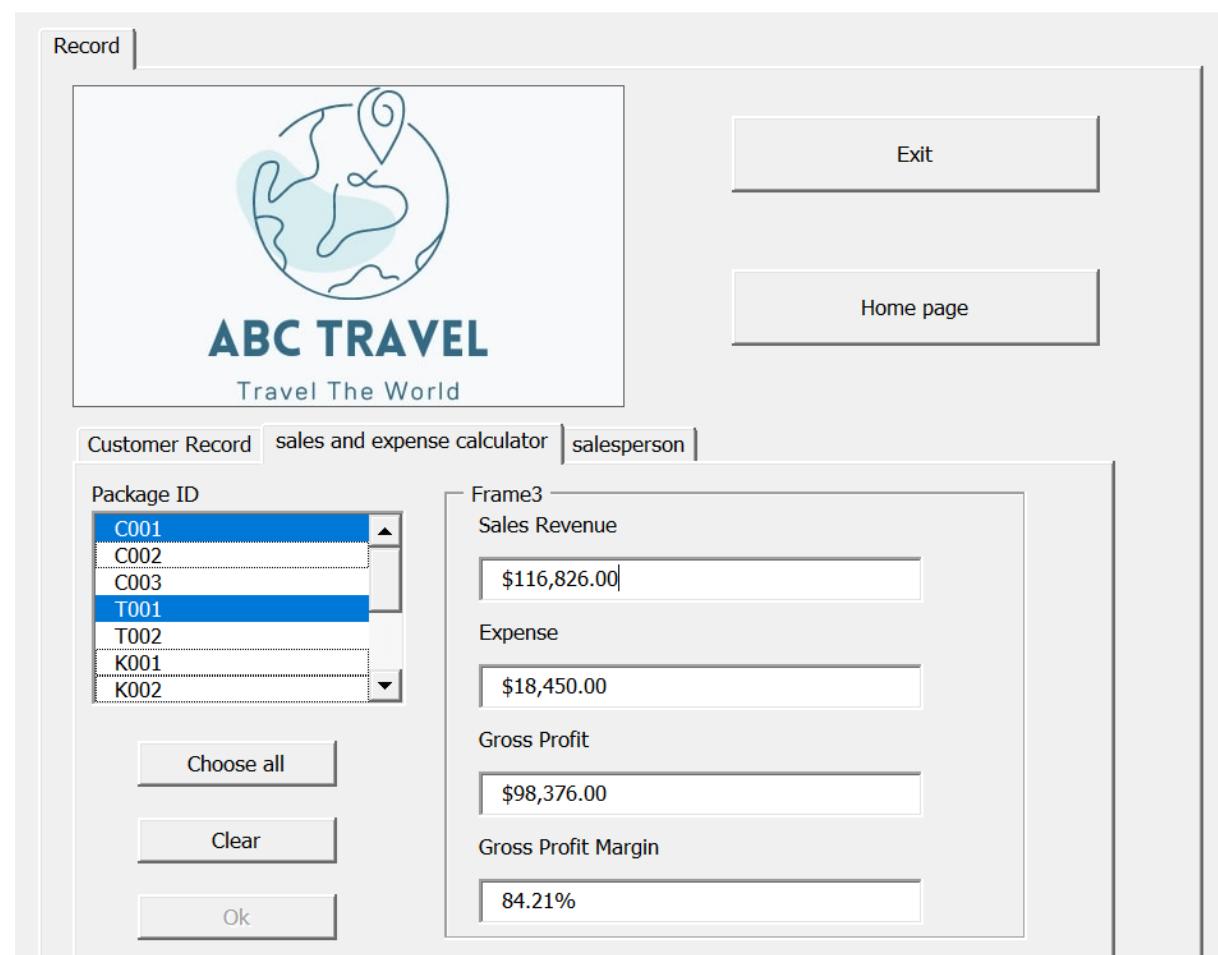
Expense

Gross Profit

Gross Profit Margin

Exit

Home page



Step 10: View the performance of packages and salespersons

The manager should go back to the main page “Manager” (ManagerAction) in Step 6, and click the “Bring Record” (opttable), “Customer Record” (Customer_Information) and “Performance” (chkPerformance). Then the browse buttons for “Customer_information.xlsx”, “Revenue.xlsx” and “performance.xlsx” (Fig. 3.4) will be enabled. The manager needs to browse the three files. Then press “OK” (Fig. 3.5) and the system will direct to the page “Performance” (StPerformance).

This page is the performance record, and the manager can check the performance of different packages and salespersons. The manager can click one between “Product” (btnproduct4) and “Salesperson” (btnsaleperson4), and click another one between “Number of Order” (btndorder) and “Sales amount” (btnsaleamount).

For example, the manager can click one between “Product” and “Number of order”. After that if the “Show” button (btnshow4) is clicked (Fig. 3.11, 3.12), “Frame5” (frmperform) will be visible and display the highest and lowest number of orders sold by the salesperson, and the corresponding salesperson ID. The Mean number of orders made by all the salespersons will be displayed as well. Vice versa for other combinations being clicked.

Manager X

Record |



ABC TRAVEL
Travel The World

Customer Record | sales and expense calculator | salesperson Performace

Frame3

Product
 Sale's person

Frame4

Number of order
 Sales amount

Frame5

	ID	Deatil
Highest	J001	11
Lowest	V001	5
Mean		8

Exit
Home page

Show
Clear

Step 11: Salespersons' salary calculator

The manager should go back to the main page “Manager” (ManagerAction) in Step 6, and click the “Bring Record” (opttable), “Customer Record”(Customer_Information), “Employee” (chkSalesperson), “Revenue and cost” (chkRevenue), and “Performance” (chkPerformance). Then click the “Employee salary calculator” (optsalary). The browse buttons for “Customer_information.xlsx”, “salesperson.xlsx”, “Revenue and cost” (chkRevenue) and “Performance” (chkPerformance) will be enabled. The manager needs to browse all the files. Then press “OK” (Fig. 3.5) and the system will direct to the page below (salarycal).

On this page, the manager can calculate each salespersons’ salary. When the manager chooses a “Salesperson ID” in the comboBox (cbxSalesperson_ID), the “salesperson_name” will be displayed automatically (Fig. 3.13). When “Calculate” button (btnCal5) is clicked (Fig. 3.14), Frame5 (frmcal) will be visible. The vba program will scan through the data files, to display the “Base_salary”, “Total_sales”, “Commission_rate” and “Commision” in order to calculate the “Total_salary” by these variables. The “Reset” (btnreset) can clear the selected salesperson_ID and make Frame5 invisible (Fig. 3.15).

Manager X

Record |


ABC TRAVEL
Travel The World

Customer Record | sales and expense calculator | salesperson | Performanace | Page1 |

Salesperson_ID

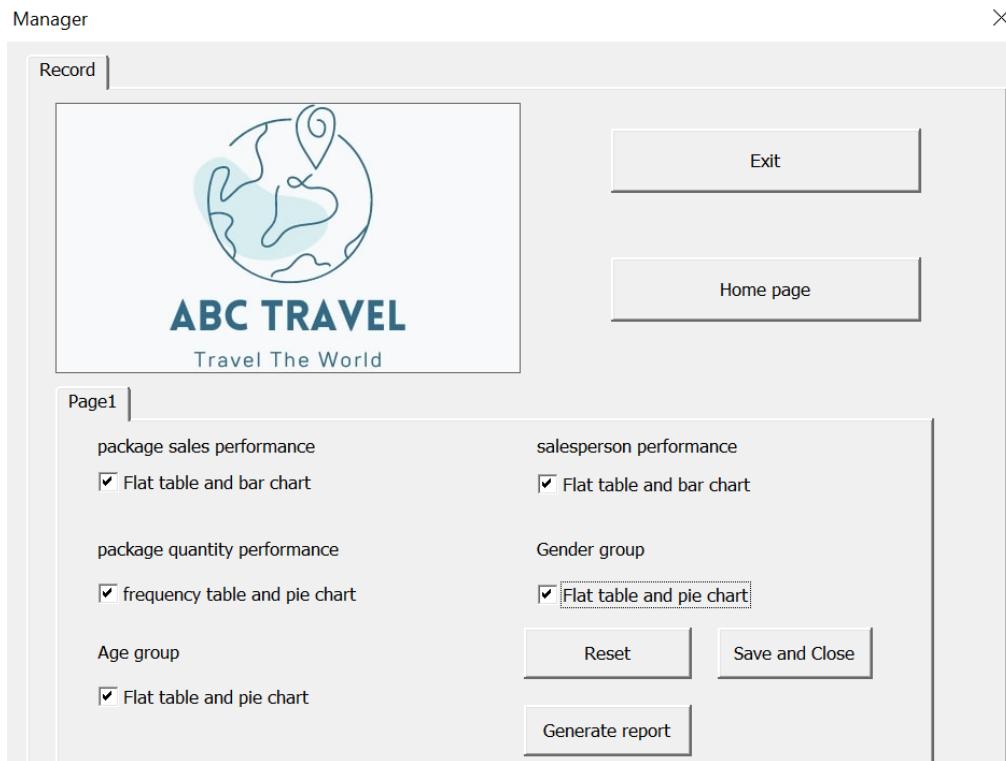
Salesperson_name Stephen Chan

Frame5

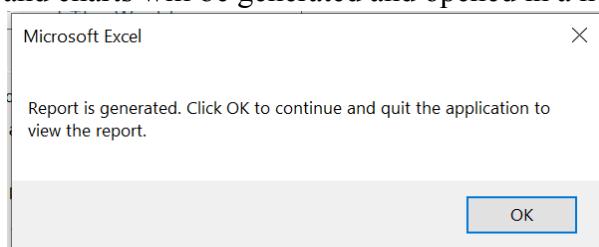
Base_salary	\$18,000.00
Total_sales	\$35,774.00
Commision_rate	15.00%
Commision	\$5,366.00
Total_salary	\$23,366.00

Step 12: Generate charts and tables

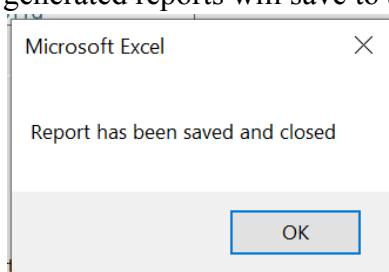
The manager should go back to the main page “Manager” (ManagerAction) in Step 6, and click the “Create table and chart” (optchart). Then the browse buttons for “Customer_information.xlsx” will be enabled. The manager needs to browse the customer information file. Then press “OK” (Fig. 3.5) and the system will direct to the page below (report). On this page, the manager can create tables and charts of numeric and character variables in 5 aspects as shown in the following. The manager can select more than one checkBox so as to generate more than one table and chart.



If the manager clicks the “Reset” commandButton (btnreset7) (Fig. 3.19), then all the selected checkboxes will be cleared. If the “Generate report” commandButton (btnreport7) (Fig. 3.20 to 3.31), then a MessageBox will appear to confirm the action. The desired tables and charts will be generated and opened in a newly created workbook.

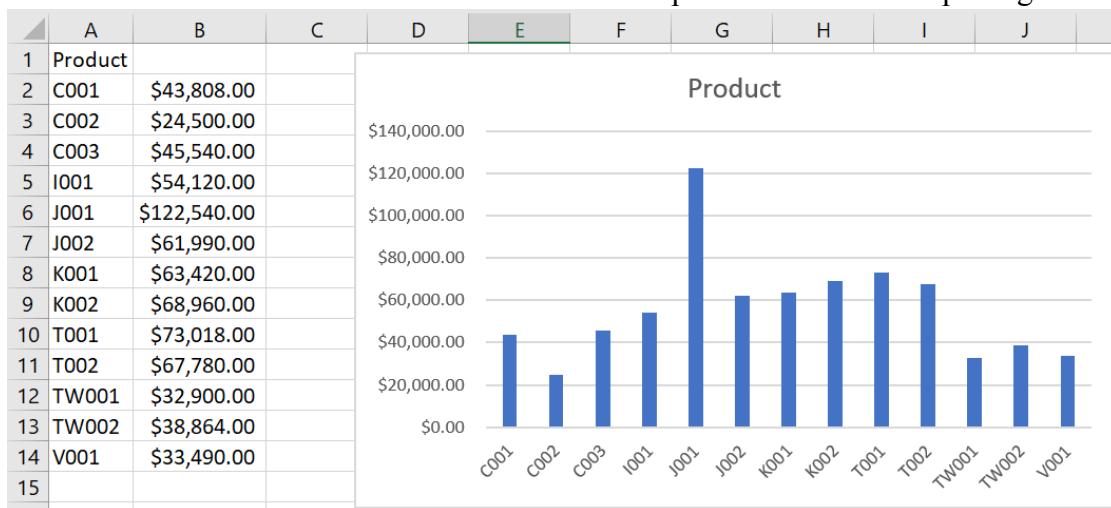


If the manager clicks “Save and Close” CommandButton (btnSaveandclose), then the generated reports will save to a newly created workbook and close (Fig. 3.32).



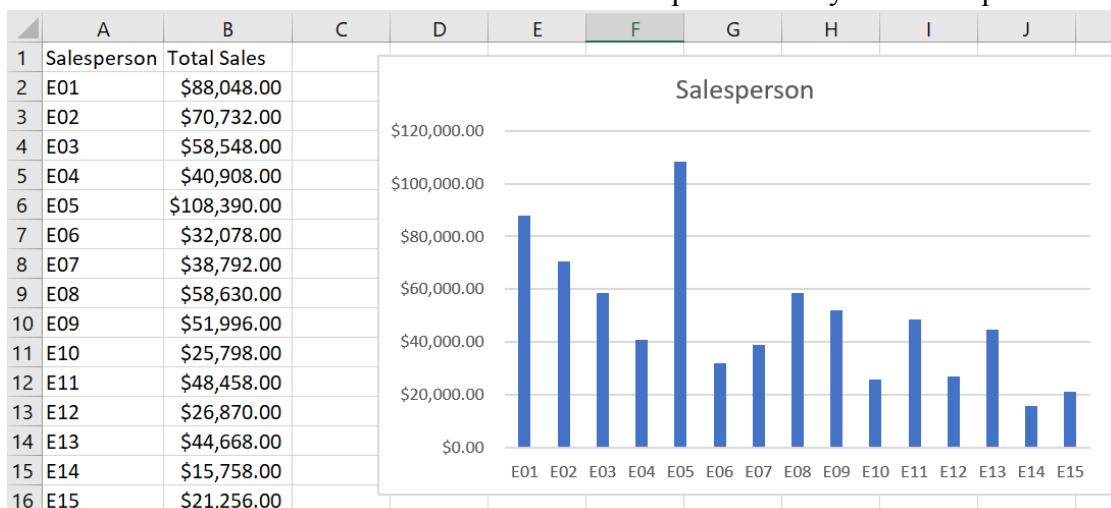
Package sales performance: Flat table and bar chart

The flat table and bar chart will show the total sales performance of each package.



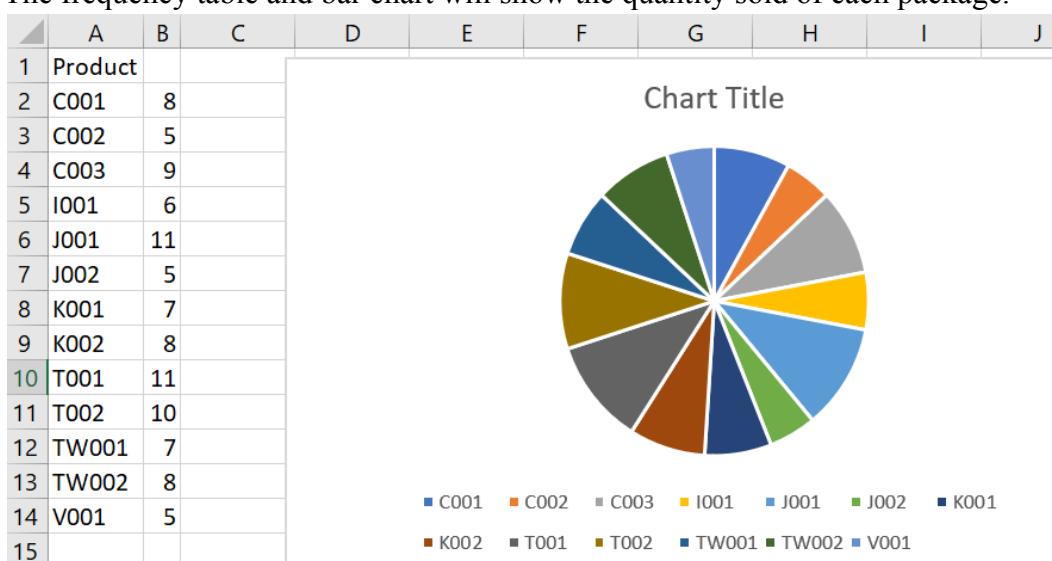
Salesperson performance: Flat table and bar chart

The flat table and bar chart will show the total sales performed by each salesperson.



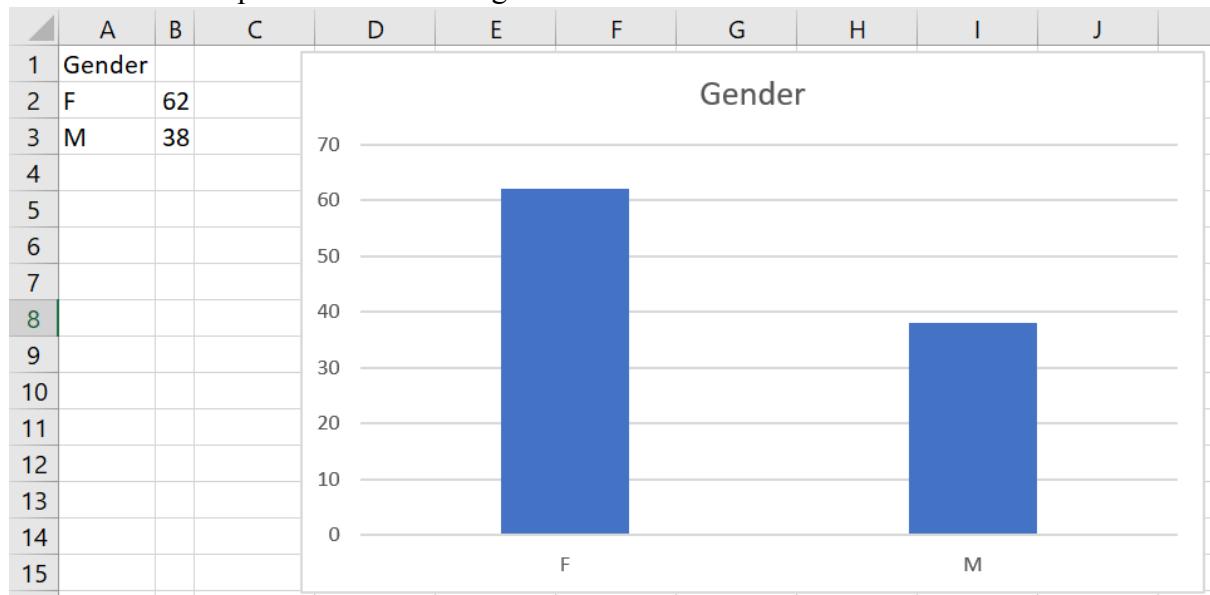
Package quantity performance: Frequency table and pie chart

The frequency table and bar chart will show the quantity sold of each package.



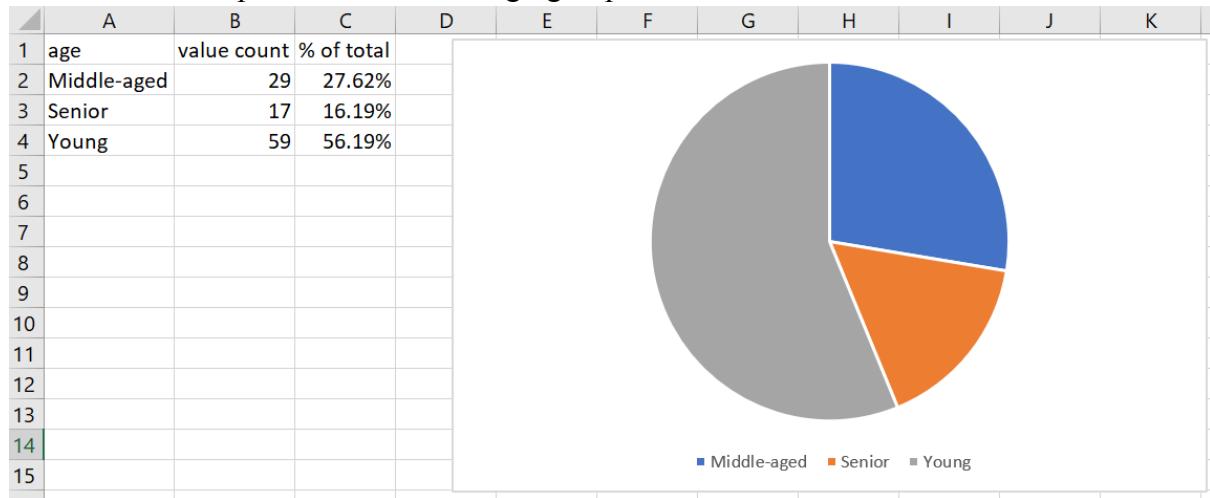
Gender Group: Flat table and pie chart

The flat table and pie chart shows the gender distribution of customers.



Age group: Flat table and pie chart

The flat table and pie chart shows the age-group distribution of customers.



Appendix:

0. Welcome button

Figure 0 (open the login page)

```
'Private Sub CommandButton1_Click()
CommandButton1.Caption = "Welcome"
Login_Page.Show
End Sub
```

1. Login Page:

Figure 1 (Set up a password system)

```
Private Sub CommandButton1_Click()
Dim manager_PW As String
Dim accurate1 As Boolean
Dim accurate2 As Boolean
Dim clerk_PW As String

'determine is the manager's password is valid or not
accurate = False
If Optmanager.Value = True Then
manager_PW = InputBox("Please insert the identity password.", "Identity Password")
Do While accurate = False
    If StrPtr(manager_PW) = 0 Then
        Exit Do
    Else
        If manager_PW = "himanager" Then
            accurate = True
            Unload Me
            Manager_Page.Show
        Else
            accurate = False
            MsgBox "The Identity password is wrong! Please enter the Identity password again" & vbCrLf, 17, "The identity password is wrong"
            manager_PW = InputBox("Please insert the identity password.", "Identity Password")
        End If
    End If
Loop
End If

'determine is the clerk's password is valid or not
accurate2 = False
If optclerk.Value = True Then
clerk_PW = InputBox("Please insert the identity password.", "Identity Password")
Do While accurate2 = False
    If StrPtr(clerk_PW) = 0 Then
        Exit Do
    Else
        If clerk_PW = "hiclerk" Then
            accurate2 = True
            Unload Me
            Clerk_Page.Show
        Else
            accurate2 = False
            MsgBox "The Identity password is wrong! Please enter the Identity password again" & vbCrLf, 17, "The identity password is wrong"
            clerk_PW = InputBox("Please insert the identity password.", "Identity Password")
        End If
    End If
Loop
End If
```

2. Clerk page

Figure 2.1 (make frame visible)

```
Private Sub CommandButton1_Click()
'make Frame1 visible
Frame1.Visible = True
End Sub
```

Figure 2.2 (browse)

```

vba_final_version.xlsm - Clerk_Page (Code)
cmdbroswer

End Sub

Private Sub cmdbroswer_Click()
'browse the workbook
strPath = Application.GetOpenFilename(filefilter:="excel, *.xlsx")

'if user have not select any workbook
If strPath = "False" Then
    MsgBox "you did not select a folder. Please select a valid file."
    Exit Sub
Else

'Activate the Workbook
ThisWorkbook.Activate
'Open the workbook
Set wbcustomer = Workbooks.Open(Filename:=strPath)
ThisWorkbook.Activate
'Set the variable wscustomer as customerinformation worksheet
Set wscustomer = wbcustomer.Worksheets("Customer_Information")
'Set the variable Nstourid as tourinformation Worksheet
Set wstourid = wbcustomer.Worksheets("Tour_detail")
'Set the variable wssalesperson as salesperson worksheet
Set wsemployee = wbcustomer.Worksheets("Salesperson")
TextBox1.Value = strPath
End If
End Sub

```

```

Private Sub btnSave_Click()
If txtname.Value <> "" And (OptMale.Value = True Or optFemale.Value = True) And cbxSalespersonID.Value <> "" And txtsalespersonname <> "" And cbxPackageID <> "" Then
numrow = rowct + 1
'save the new record to the customer_information workbook
wscustomer.Cells(numrow, 1) = lblCustomerID1.Caption
wscustomer.Cells(numrow, 2) = txtname.Value
If OptMale.Value = True Then wscustomer.Cells(numrow, 3) = "M"
If optFemale.Value = True Then wscustomer.Cells(numrow, 3) = "F"
wscustomer.Cells(numrow, 4) = lblagenumber.Caption
wscustomer.Cells(numrow, 5) = cbxPackageID.Value
wscustomer.Cells(numrow, 6) = lblPriceEntry.Caption
wscustomer.Cells(numrow, 7) = cbxSalespersonID.Value
wscustomer.Cells(numrow, 8) = lblDate1.Caption
'show message to ask if the user want to create new record or exit system
Dim answer As String
answer = MsgBox("The Customer record(" & lblCustomerID1 & ") is saved." & vbCrLf & "Are you create another record?", vbYesNo, "Record saved")
If answer = vbYes Then
    wbcustomer.Save
    Call btnClear_Click
End If
'if the answer is no, then unload userform
If answer = vbNo Then
    Unload Me
    wbcustomer.Close savechanges:=True
End If
Else
    MsgBox "Please enter all the data in the form.", vbOKOnly, "Missing input"
End If
End Sub

```

Figure 2.3 (next button)

Figure 2.4 (Exit button)

```

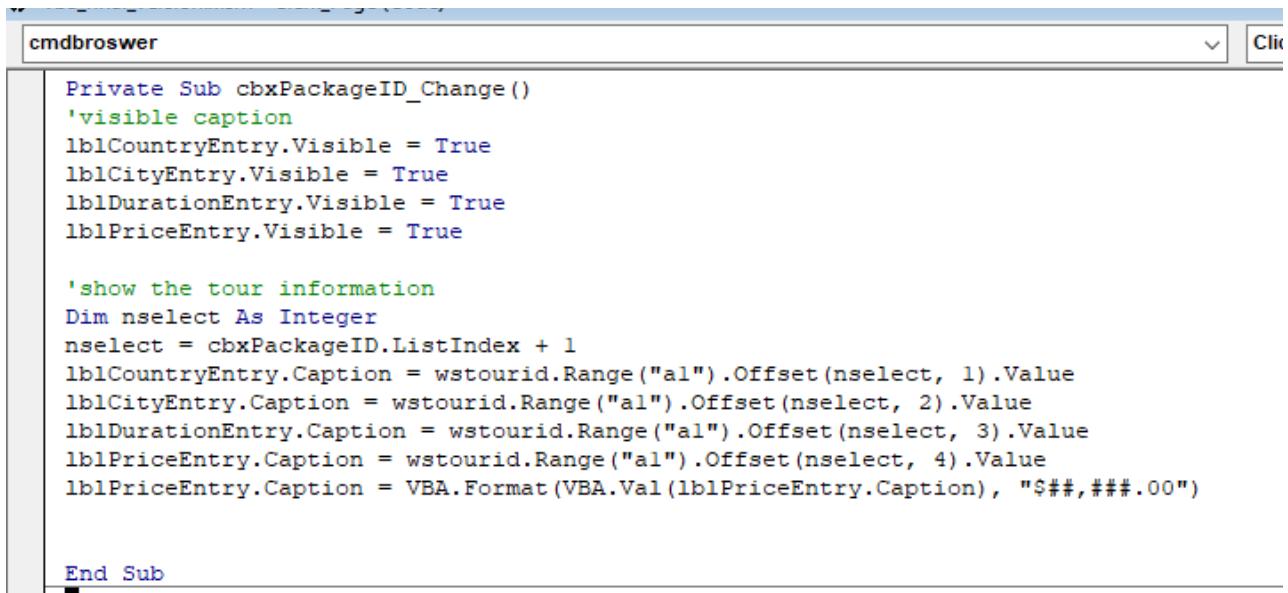
Private Sub btnexit_Click()
Dim Quit As String
'when th quit button is clicked,confirm the closing of the the userform
Quit = VBA MsgBox("Do you want to quit?", Buttons:=vbYesNo)

If Quit = vbYes Then VBA.Unload Clerk_Page

End Sub

```

Figure 2.5 (show detail)



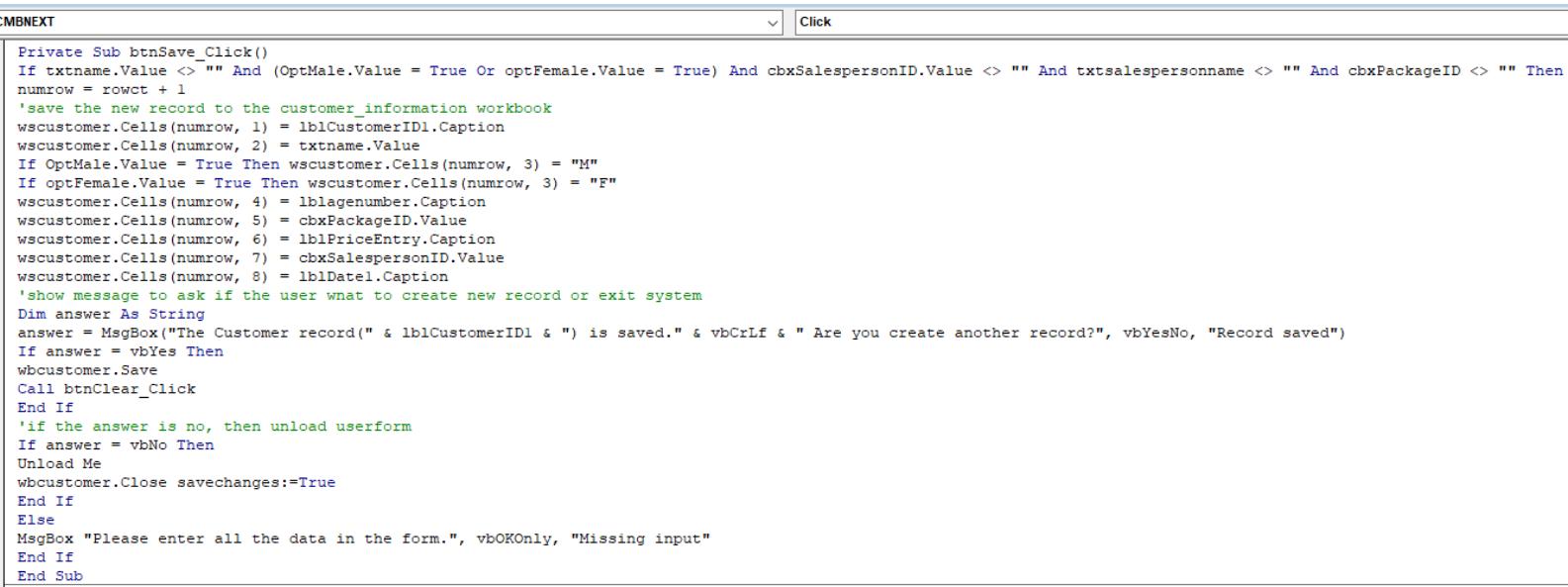
The screenshot shows a Microsoft Word document window titled "cmdbroswer". The code is written in VBA and defines a subroutine named "cbxPackageID_Change". The code uses several labels and variables like "nselect" and "lblCountryEntry" to update labels based on the selected package ID.

```
Private Sub cbxPackageID_Change()
    'visible caption
    lblCountryEntry.Visible = True
    lblCityEntry.Visible = True
    lblDurationEntry.Visible = True
    lblPriceEntry.Visible = True

    'show the tour information
    Dim nselect As Integer
    nselect = cbxPackageID.ListIndex + 1
    lblCountryEntry.Caption = wstourid.Range("al").Offset(nselect, 1).Value
    lblCityEntry.Caption = wstourid.Range("al").Offset(nselect, 2).Value
    lblDurationEntry.Caption = wstourid.Range("al").Offset(nselect, 3).Value
    lblPriceEntry.Caption = wstourid.Range("al").Offset(nselect, 4).Value
    lblPriceEntry.Caption = VBA.Format(VBA.Val(lblPriceEntry.Caption), "$##,###.00")

    End Sub
```

Figure 2.6 (Save button)



The screenshot shows a Microsoft Word document window titled "MBNEXT". The code is written in VBA and defines a subroutine named "btnSave_Click". It handles saving a new record to a worksheet named "wscustomer" and then asks if the user wants to create another record or exit the system.

```
Private Sub btnSave_Click()
If txtname.Value <> "" And (OptMale.Value = True Or optFemale.Value = True) And cbxSalespersonID.Value <> "" And txtsalespersonname <> "" And cbxPackageID <> "" Then
    numrow = rowct + 1
    'save the new record to the customer_information workbook
    wscustomer.Cells(numrow, 1) = lblCustomerID1.Caption
    wscustomer.Cells(numrow, 2) = txtname.Value
    If OptMale.Value = True Then wscustomer.Cells(numrow, 3) = "M"
    If optFemale.Value = True Then wscustomer.Cells(numrow, 3) = "F"
    wscustomer.Cells(numrow, 4) = lblagenumber.Caption
    wscustomer.Cells(numrow, 5) = cbxPackageID.Value
    wscustomer.Cells(numrow, 6) = lblPriceEntry.Caption
    wscustomer.Cells(numrow, 7) = cbxSalespersonID.Value
    wscustomer.Cells(numrow, 8) = lblDate1.Caption
    'show message to ask if the user want to create new record or exit system
    Dim answer As String
    answer = MsgBox("The Customer record(" & lblCustomerID1 & ") is saved." & vbCrLf & "Are you create another record?", vbYesNo, "Record saved")
    If answer = vbYes Then
        wbcustomer.Save
        Call btnClear_Click
    End If
    'if the answer is no, then unload userform
    If answer = vbNo Then
        Unload Me
        wbcustomer.Close savechanges:=True
    End If
    Else
        MsgBox "Please enter all the data in the form.", vbOKOnly, "Missing input"
    End If
End Sub
```

Figure 2.7 (clear button)

```
vba_final_version.xlsm - Clerk_Page (Code)
btnClear
Private Sub btnClear_Click()
    'clear all data input of the userform

    txtname.Value = ""
    OptMale.Value = False
    optFemale.Value = False
    sbrage = 18
    cbxSalespersonID = ""
    txtsalespersonname = ""
    cbxPackageID = ""
    If cbxPackageID = "" Then
        lblCountryEntry.Caption = ""
        lblCityEntry.Caption = ""
        lblDurationEntry.Caption = ""
        lblPriceEntry = ""
    End If

    txtname.SetFocus

End Sub
```

Figure 2.8 (Quit button)

```
Private Sub btnQuit_Click()
Dim Quit As String

Quit = VBA MsgBox("Do you want to quit?", Buttons:=vbYesNo)

If Quit = vbYes Then VBA.Unload Clerk_Page
End Sub
```

Figure 2.9 (Home button)

```
Private Sub btnHome_Click()
mpgClerk.Value = 0
Frame1.Visible = False
TextBox1 = ""
mpgClerk.Page2.Visible = False
End Sub
```

3. manager page

Figure 3.1 (Customer browser)

```
btnBrowseCustomer
Private Sub btnBrowseCustomer_Click()
    'browse the workbook
    On Error GoTo Message
    strpath1 = Application.GetOpenFilename(filefilter:="excel,*.xlsx")
    TextBox2.Value = strpath1
    'open the customer_information workbook
    Set wbcustomerrecord = Workbooks.Open(Filename:=strpath1)
    'activate this workbook
    ThisWorkbook.Activate
    'set the worksheet from the workbook opened
    Set wscustomer = wbcustomerrecord.Worksheets("Customer_Information")
    Set customerinf = wscustomer.Range("A1:I1048576")

    'show message when user doesnt choose any file
    Message:
    If TextBox2.Value = "False" Then
        MsgBox "No selected file,Please select a valid file again."
        TextBox2.Value = ""
    Else
    End If
End Sub
```

Figure 3.2 (Revenue browser)

```
vba_final_version.xlsxm - Manager_Page (Code)
btnBrowseSalesperson
Private Sub btnBrowseRevenue_Click()
'browse the workbook
On Error GoTo Message
strpath2 = Application.GetOpenfilename(filefilter:="excel,*.xlsx")
    TextBox5.Value = strpath2
    'open the revenue workbook
    Set wbrevenue = Workbooks.Open(Filename:=strpath2)
    'activate this workbook
    ThisWorkbook.Activate
    'set the worksheet from the workbook opened
    Set wsrevenue = wbrevenue.Worksheets("Revenue")
    'Set customerinf = wscustomer.Range("A1:I1048576")

    'show message when user doesnt choose any file
    Message:
    If TextBox5.Value = "False" Then
        MsgBox "No selected file,Please select a valid file again."
        TextBox5.Value = ""
    Else
    End If

End Sub
```

Figure 3.3 (salesperson browser)

```
Private Sub btnBrowseSalesperson_Click()
'browse the workbook
On Error GoTo Message
strpath3 = Application.GetOpenfilename(filefilter:="excel,*.xlsx")
    TextBox6.Value = strpath3
    'open the revenue workbook
    Set wbsalesperson = Workbooks.Open(Filename:=strpath3)
    'activate this workbook
    ThisWorkbook.Activate
    'set the worksheet from the workbook opened
    Set wssalesperson = wbsalesperson.Worksheets("salesperson")
    'Set customerinf = wscustomer.Range("A1:I1048576")

    'show message when user doesnt choose any file
    Message:
    If TextBox6.Value = "False" Then
        MsgBox "No selected file,Please select a valid file again."
        TextBox6.Value = ""
    Else
    End If

End Sub
```

Figure 3.4 (performance browser)

```
Private Sub btnBrowsePerformance_Click()
'browse the workbook
On Error GoTo Message
strpath4 = Application.GetOpenfilename(filefilter:="excel,*.xlsx")
    TextBox7.Value = strpath4
    'open the performance workbook

    Set wbperformance = Workbooks.Open(Filename:=strpath4)
    'activate this workbook
    ThisWorkbook.Activate
    'set the worksheet from the workbook opened
    Set wsperformance = wbperformance.Worksheets("performance")
    'Set customerinf = wscustomer.Range("A1:I1048576")

    'show message when user doesnt choose any file
    Message:
    If TextBox7.Value = "False" Then
        MsgBox "No selected file,Please select a valid file again."
        TextBox7.Value = ""
    Else
    End If
End Sub
```

Figure 3.5 (confirm button)

```
Private Sub btnconfirm_Click()
'if choose the opttable and textbox2 not empty
If opttable.Value = True And TextBox2 <> "" Then
'Visible the mpgManager1.Manager_Work
    mpgManager1.Manager_Work.Visible = True
'go to manager_work page
    mpgManager1.Value = 1
    For crow = 1 To wscustomer.Range(wscustomer.Range("a1"), wscustomer.Range("a1048576").End(xlUp)).Rows.Count - 1
        'Add tour id in the ComboBox
        cmbCustomerID1.AddItem wscustomer.Range("a1").Offset(crow, 0).Value
    Next crow
End If
'if choose the opttable and textbox5 not empty
If opttable.Value = True And TextBox5 <> "" Then
'Visible the mpgManager1.Manager_Work
    mpgManager1.Manager_Work.Visible = True
'go to manager_work page
    mpgManager1.Value = 1
'go to Salesexpense page
    MultiPagel.Value = 1
'let the user choose more than one in the list box of Package ID
    lbxPackageID2.MultiSelect = 1

    For drow = 1 To wsrevenue.Range(wsrevenue.Range("a1"), wsrevenue.Range("a1048576").End(xlUp)).Rows.Count - 1
        'Add tour id in the ComboBox
        lbxPackageID2.AddItem wsrevenue.Range("a1").Offset(drow, 0).Value
    Next drow
End If
'if choose the opttable and textbox6 not empty
If opttable.Value = True And TextBox6 <> "" Then
'Visible the mpgManager1.Manager_Work
    mpgManager1.Manager_Work.Visible = True
'go to manager_work page
    mpgManager1.Value = 1
'go to salesperson_record page
    MultiPagel.Value = 2
    For zrow = 1 To wssalesperson.Range(wssalesperson.Range("a1"), wssalesperson.Range("a1048576").End(xlUp)).Rows.Count - 1
        'Add tour id in the ComboBox
        cmbstaffid3.AddItem wssalesperson.Range("a1").Offset(zrow, 0).Value
    Next zrow

End If
'if choose the opttable and textbox6 not empty
If optsalary.Value = True And TextBox2 <> "" And TextBox6 <> "" Then
'Visible the mpgManager1.Manager_Work
    mpgManager1.Manager_Work.Visible = True
'Visible the salarycal
    MultiPagel.salarycal.Visible = True
'go to manager_work page
    mpgManager1.Value = 1
'go to salarycal page
    MultiPagel.Value = 4
    For erow = 1 To wssalesperson.Range(wssalesperson.Range("a1"), wssalesperson.Range("a1048576").End(xlUp)).Rows.Count - 1
        'Add tour id in the ComboBox
        cbxSalesperson_ID.AddItem wssalesperson.Range("a1").Offset(erow, 0).Value
    Next erow
```

Figure 3.6 (confirm button)

```
    cbxSalesperson_ID.AddItem wssalesperson.Range("a1").Offset(erow, 0).Value
    Next erow
End If
'if choose the opttable and textbox7 and textbox5 not empty
If opttable.Value = True And TextBox7 <> "" And TextBox5 <> "" And chkPerformance.Value = True Then
'Visible the mpgManager1.Manager_Work
    mpgManager1.Manager_Work.Visible = True
    'go to manager_work page
    mpgManager1.Value = 1
    'go to StPerformance page
    MultiPagel.Value = 3
End If
'if choose the chkCustomer
If chkCustomer.Value = True Then
'Visible the Customer_Information
    mpgManager1.Manager_Work.MultiPagel.Customer_Information.Visible = True
    'go to Customer_Information
    mpgManager1.Manager_Work.MultiPagel.Value = 0
End If
'if choose the chkSalesperson
If chkSalesperson.Value = True Then
'Visible the salesperson_record
    mpgManager1.Manager_Work.MultiPagel.salesperson_record.Visible = True
    'go to salesperson_record
    mpgManager1.Manager_Work.MultiPagel.Value = 2
End If
'if choose the chkRevenue
If chkRevenue.Value = True Then
'Visible the Salesexpense
    mpgManager1.Manager_Work.MultiPagel.Salesexpense.Visible = True
    'go to Salesexpense
    mpgManager1.Manager_Work.MultiPagel.Value = 1
End If

'if choose the chkPerformance
If chkPerformance.Value = True Then
'Visible the StPerformance
    mpgManager1.Manager_Work.MultiPagel.StPerformance.Visible = True
    'go to StPerformance
    mpgManager1.Manager_Work.MultiPagel.Value = 3
End If
'if choose the optsalary
If optsalary.Value = True Then
'Visible the salarycal
    mpgManager1.Manager_Work.MultiPagel.salarycal.Visible = True
    'go to salarycal
    mpgManager1.Manager_Work.MultiPagel.Value = 4
End If

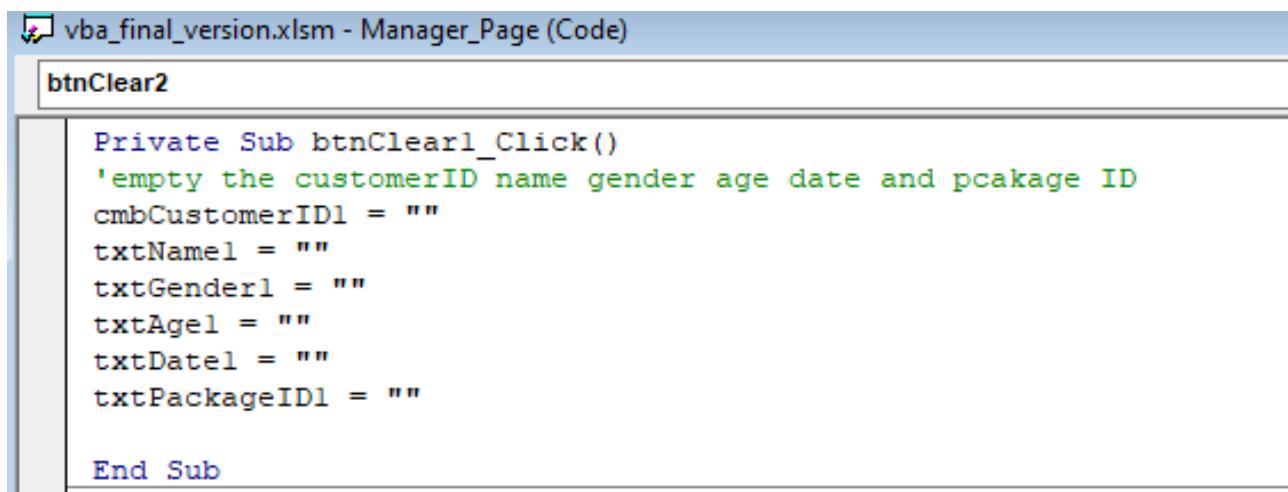
'if choose the report and textbox7 not empty
If optchart.Value = True And TextBox2 <> "" Then
'Visible the mpgManager1.Manager_Work
    mpgManager1.Manager_Work.Visible = True
    mpgManager1.Manager_Work.MultiPagel.Report.Visible = True
```

Figure 3.7 (customer ID change)

```
Private Sub cmbCustomerID1_Change()
Dim cselect As Integer
cselect = cmbCustomerID1.ListIndex + 1
'put their name gender age and packageID and date in it
txtName1.Value = wscustomer.Range("a1").Offset(cselect, 1).Value
txtGender1.Value = wscustomer.Range("a1").Offset(cselect, 2).Value
txtAge1.Value = wscustomer.Range("a1").Offset(cselect, 3).Value
txtPackageID1.Value = wscustomer.Range("a1").Offset(cselect, 4).Value
txtDate1.Value = wscustomer.Range("a1").Offset(cselect, 7).Value

End Sub
```

Figure 3.8 (clear button)



The screenshot shows the VBA code editor with the title bar "vba_final_version.xlsxm - Manager_Page (Code)". The code is contained within the "btnClear2" module, indicated by the tab at the top of the code window.

```
Private Sub btnClear1_Click()
'empty the customerID name gender age date and package ID
cmbCustomerID1 = ""
txtName1 = ""
txtGender1 = ""
txtAge1 = ""
txtDate1 = ""
txtPackageID1 = ""

End Sub
```

Figure 3.9 (ok button)

```
Private Sub btnOk2_Click()
    'show the calculation frame
    frmCalculator1.Visible = True

    Dim productID As String
    Dim srow As Integer
    Dim h As Integer

    Dim revenueuel As Long
    Dim accrevenueuel As Long
    Dim costl As Long
    Dim acccostl As Long
    'accrevenueuel equal to 0
    accrevenueuel = 0
    'acccostl equal to 0
    acccostl = 0
    'put the package ID in the lbxPackageID2
    For h = 0 To lbxPackageID2.ListCount - 1
        If lbxPackageID2.Selected(h) = True Then
            productID = lbxPackageID2.List(h)
            'calculate the revenue and cost of selected item
            For srow = 1 To wsrevenue.Cells.SpecialCells(xlCellTypeLastCell).Row - 1
                With wsrevenue.Cells(srow, 1)
                    If .Value = productID Then
                        revenueuel = .Offset(0, 3).Value
                        accrevenueuel = accrevenueuel + revenueuel
                        costl = .Offset(0, 8).Value
                        acccostl = acccostl + costl
                    End If
                End With
            Next srow
        End If
    Next h
    'calculate the revenue and cost
    txtSalesRevenue.Value = accrevenueuel
    txtExpense = acccostl
    txtGrossProfit = accrevenueuel - acccostl

    'create format of the textbox value
    txtGPM.Value = FormatPercent(txtGrossProfit / txtSalesRevenue)
    txtSalesRevenue.Value = Format(txtSalesRevenue, "$###,###,##.00")
    txtExpense.Value = Format(txtExpense, "$###,###,##.00")
    txtGrossProfit.Value = Format(txtGrossProfit, "$###,###,##.00")

    'disenaable the ok button
    btnOk2.Enabled = False
```

Figure 3.10 (select all button)

```
Private Sub btnChooseall2_Click()
Dim c As Integer
'choose all the package ID
For c = 0 To lbxPackageID2.ListCount - 1
lbxPackageID2.Selected(c) = True
Next c

End Sub
```

Figure 3.11 (clear button)

```
btnClear2

Private Sub btnClear2_Click()
Dim i As Integer
'empty the package ID
For i = 0 To lbxPackageID2.ListCount - 1
    lbxPackageID2.Selected(i) = False
Next i

'disable the OK button
btnOk2.Enabled = False
'invisible the calculation frame
frmCalculator1.Visible = False

'txtSalesRevenue.Value = ""
'txtExpense.Value = ""
'txtGrossProfit.Value = ""
'txtGPM.Value = ""

End Sub
```

Figure 3.11 (page3 staff id change)

```
Private Sub cmbstaffid3_Change()
Dim pselect As Integer
pselect = cmbstaffid3.ListIndex + 1
'put their name gender in it
txtstaffname3.Value = wssalesperson.Range("a1").Offset(pselect, 1).Value
lblgeneroutput.Caption = wssalesperson.Range("a1").Offset(pselect, 2).Value
lbltitleoutput.Caption = wssalesperson.Range("a1").Offset(pselect, 3).Value

End Sub
```

Figure 3.12 (show button)

```
End Sub

Private Sub btnshow4_Click()
frmperform.Visible = True
Dim n As Long

'show the max,min and mean of package's order
If btnproduct4.Value = True And btndetail.Value = True Then
txtdetail.Value = Application.WorksheetFunction.Max(wsrevenue.Range("c1:c1048576"))
txtdetail2.Value = Application.WorksheetFunction.Min(wsrevenue.Range("c1:c1048576"))
For n = 1 To wsperformance.Cells.SpecialCells(xlCellTypeLastCell).Row
    With wsrevenue.Cells(n, 3)
        If .Value = Application.WorksheetFunction.Max(wsrevenue.Range("c1:c1048576")) Then
            txtID.Value = .Offset(0, -2).Value
        End If
    End With
Next n
For n = 1 To wsrevenue.Cells.SpecialCells(xlCellTypeLastCell).Row
    With wsrevenue.Cells(n, 3)
        If .Value = Application.WorksheetFunction.Min(wsrevenue.Range("c1:c1048576")) Then
            txtID2.Value = .Offset(0, -2).Value
        End If
    End With
Next n
txtdetail3 = Application.WorksheetFunction.Average(wsrevenue.Range("c1:c1048576"))
txtdetail3 = Round(txtdetail3, 0)
End If

'show the max,min and mean of package's salesamount
If btnproduct4.Value = True And btnsaleamount.Value = True Then
txtdetail.Value = Application.WorksheetFunction.Max(wsrevenue.Range("d1:d1048576"))
txtdetail2.Value = Application.WorksheetFunction.Min(wsrevenue.Range("d1:d1048576"))
For n = 1 To wsperformance.Cells.SpecialCells(xlCellTypeLastCell).Row
    With wsrevenue.Cells(n, 4)
        If .Value = Application.WorksheetFunction.Max(wsrevenue.Range("d1:d1048576")) Then
            txtID.Value = .Offset(0, -3).Value
        End If
    End With
Next n
For n = 1 To wsrevenue.Cells.SpecialCells(xlCellTypeLastCell).Row
    With wsrevenue.Cells(n, 4)
        If .Value = Application.WorksheetFunction.Min(wsrevenue.Range("d1:d1048576")) Then
            txtID2.Value = .Offset(0, -3).Value
        End If
    End With
Next n
txtdetail3 = Application.WorksheetFunction.Average(wsrevenue.Range("d1:d1048576"))
txtdetail3 = Round(txtdetail3, 0)
```

Figure 3.12 (show button con't)

```
'show the max,min,mean of salesperson's order
If btnsaleperson4.Value = True And btnorder.Value = True Then
    txtdetail.Value = Application.WorksheetFunction.Max(wsperformance.Range("p1:p1048576"))
    txtdetail2.Value = Application.WorksheetFunction.Min(wsperformance.Range("p1:p1048576"))
    For n = 1 To wsperformance.Cells.SpecialCells(xlCellTypeLastCell).Row
        With wsperformance.Cells(n, 16)
            If .Value = Application.WorksheetFunction.Max(wsperformance.Range("p1:p1048576")) Then
                txtID.Value = .Offset(0, -15).Value
            End If
        End With
    Next n
    For n = 1 To wsperformance.Cells.SpecialCells(xlCellTypeLastCell).Row
        With wsperformance.Cells(n, 16)
            If .Value = Application.WorksheetFunction.Min(wsperformance.Range("p1:p1048576")) Then
                txtID2.Value = .Offset(0, -15).Value
            End If
        End With
    Next n
    txtdetail3 = Application.WorksheetFunction.Average(wsperformance.Range("p1:p1048576"))
    txtdetail3 = Round(txtdetail3, 0)
End If

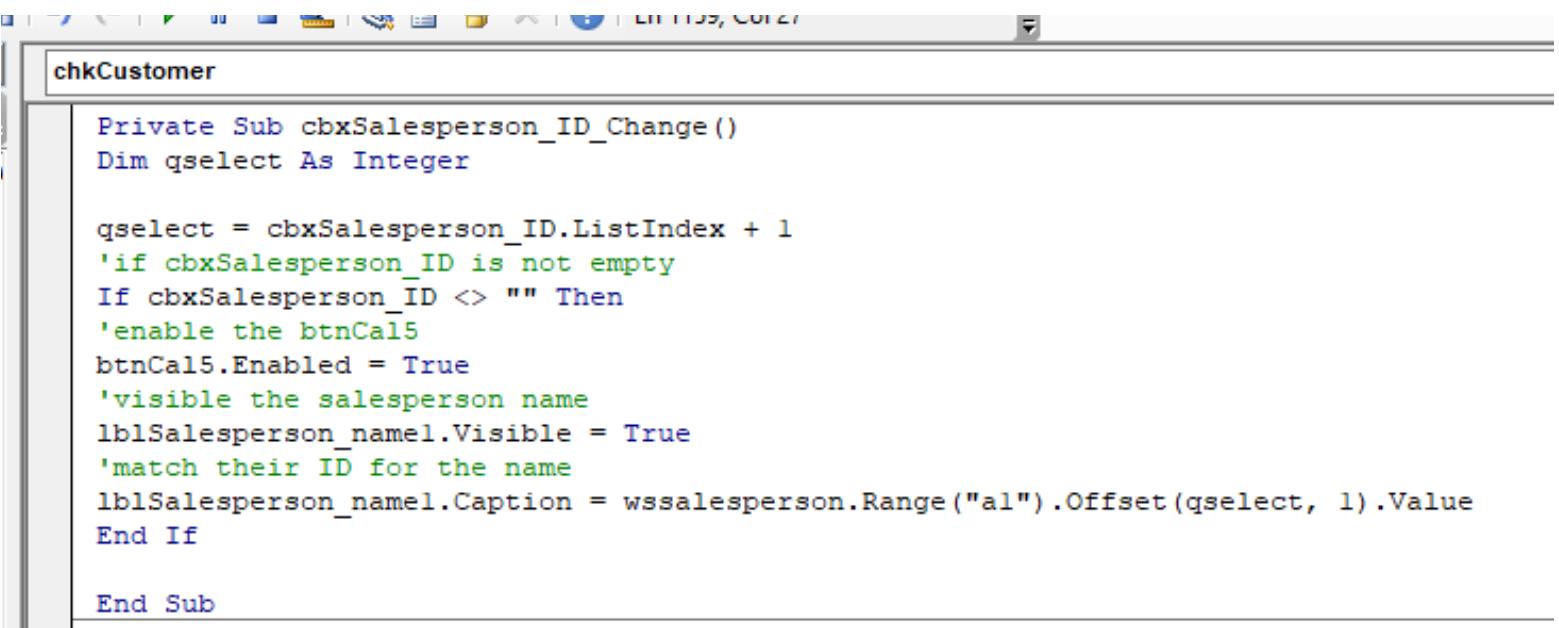
'show the max,min,min of salesperson's salesamount
If btnsaleperson4.Value = True And btnsaleamount.Value = True Then
    txtdetail.Value = Application.WorksheetFunction.Max(wsperformance.Range("q1:q1048576"))
    txtdetail2.Value = Application.WorksheetFunction.Min(wsperformance.Range("q1:q1048576"))

    For n = 1 To wsperformance.Cells.SpecialCells(xlCellTypeLastCell).Row
        With wsperformance.Cells(n, 17)
            If .Value = Application.WorksheetFunction.Max(wsperformance.Range("q1:q1048576")) Then
                txtID.Value = .Offset(0, -16).Value
            End If
        End With
    Next n
    For n = 1 To wsperformance.Cells.SpecialCells(xlCellTypeLastCell).Row
        With wsperformance.Cells(n, 17)
            If .Value = Application.WorksheetFunction.Min(wsperformance.Range("q1:q1048576")) Then
                txtID2.Value = .Offset(0, -16).Value
            End If
        End With
    Next n
    txtdetail3 = Application.WorksheetFunction.Average(wsperformance.Range("q1:q1048576"))
    txtdetail3 = Round(txtdetail3, 0)

End If

End Sub
```

Figure 3.13 (salesperson id change for calculator)

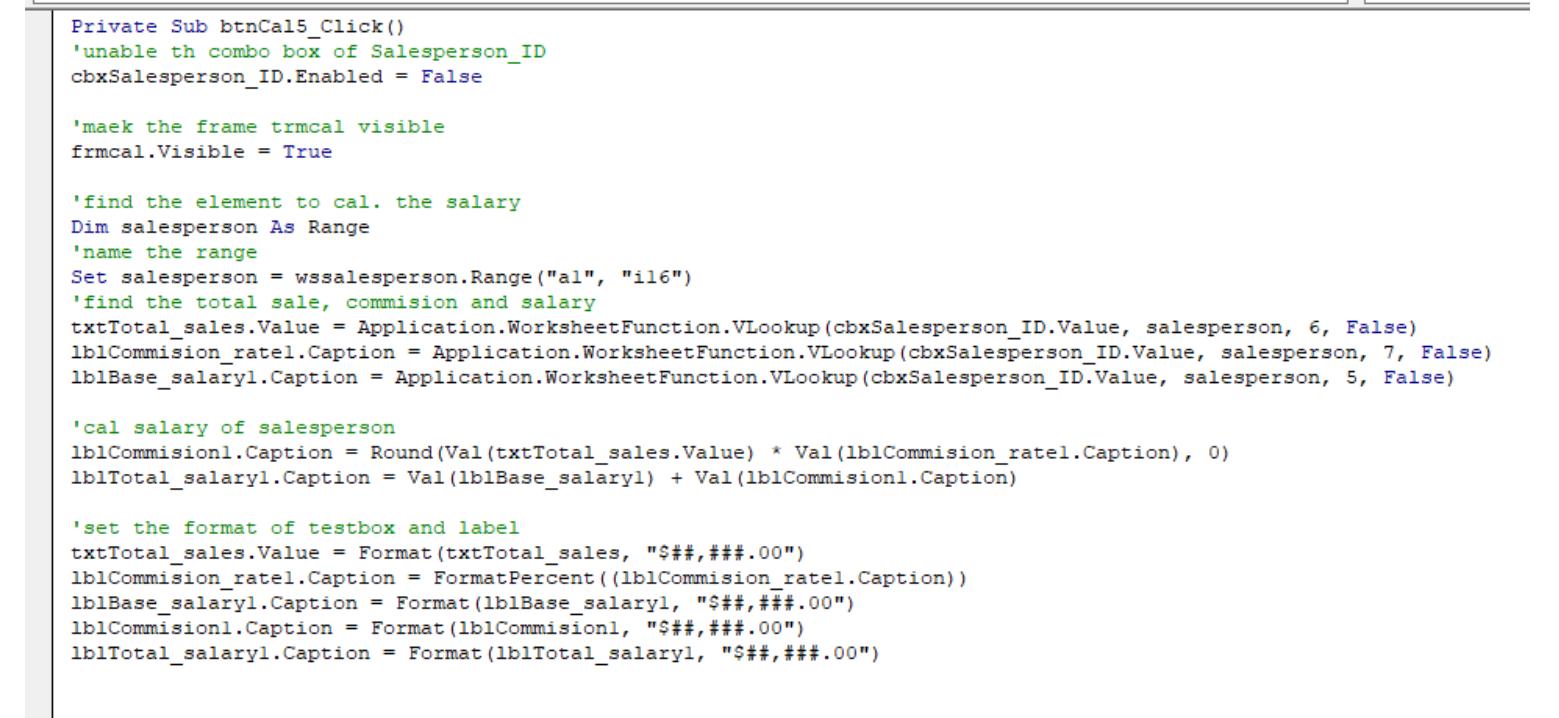


```
chkCustomer
Private Sub cbxSalesperson_ID_Change()
Dim qselect As Integer

qselect = cbxSalesperson_ID.ListIndex + 1
'if cbxSalesperson_ID is not empty
If cbxSalesperson_ID <> "" Then
'enable the btnCal5
btnCal5.Enabled = True
'visible the salesperson name
lblSalesperson_name1.Visible = True
'match their ID for the name
lblSalesperson_name1.Caption = wssalesperson.Range("a1").Offset(qselect, 1).Value
End If

End Sub
```

Figure 3.14 (calculate button)



```
Private Sub btnCal5_Click()
'unable th combo box of Salesperson_ID
cbxSalesperson_ID.Enabled = False

'maek the frame trmcal visible
frmcal.Visible = True

'find the element to cal. the salary
Dim salesperson As Range
'name the range
Set salesperson = wssalesperson.Range("a1", "i16")
'find the total sale, commision and salary
txtTotal_sales.Value = Application.WorksheetFunction.VLookup(cbxSalesperson_ID.Value, salesperson, 6, False)
lblCommision_rate1.Caption = Application.WorksheetFunction.VLookup(cbxSalesperson_ID.Value, salesperson, 7, False)
lblBase_salary1.Caption = Application.WorksheetFunction.VLookup(cbxSalesperson_ID.Value, salesperson, 5, False)

'cal salary of salesperson
lblCommision1.Caption = Round(Val(txtTotal_sales.Value) * Val(lblCommision_rate1.Caption), 0)
lblTotal_salary1.Caption = Val(lblBase_salary1) + Val(lblCommision1.Caption)

'set the format of testbox and label
txtTotal_sales.Value = Format(txtTotal_sales, "$##,###.00")
lblCommision_rate1.Caption = FormatPercent((lblCommision_rate1.Caption))
lblBase_salary1.Caption = Format(lblBase_salary1, "$##,###.00")
lblCommision1.Caption = Format(lblCommision1, "$##,###.00")
lblTotal_salary1.Caption = Format(lblTotal_salary1, "$##,###.00")

End Sub
```

Figure 3.15 (reset button)

```
Private Sub btnreset_Click()
'reset all data of salesperson salary calculator
frmcal.Visible = False
cbxSalesperson_ID.Enabled = True
cbxSalesperson_ID.Value = ""
If cbxSalesperson_ID = "" Then lblSalesperson_name1 = ""
```

Figure 3.16 (Bring Record button)

```
Private Sub opttable_Click()
'enable Frame1
Frame1.Enabled = True
End Sub
```

Figure 3.17 (Homepage button)

```
Private Sub btnHomePage1_Click()
mpgManager1.ManagerAction.Visible = True
mpgManager1.Manager_Work.Visible = False
'back to ManagerAction
mpgManager1.Value = 0
'clear all the option button
opttable.Value = False
optchart.Value = False
optsalary.Value = False
chkCustomer.Value = 0
chkSalesperson.Value = 0
chkRevenue.Value = 0
chkPerformance.Value = 0
btnBrowseCustomer.Enabled = False
btnBrowseRevenue.Enabled = False
btnBrowseSalesperson.Enabled = False
btnBrowsePerformance.Enabled = False
TextBox2 = ""
TextBox5 = ""
TextBox7 = ""
TextBox6 = ""
End Sub
```

Fig. 3.18 (Exit button)

```
Private Sub btnExit1_Click()
Dim Quit As String
'appear the messagebox: Do you want to quit? and Yes No question
Quit = VBA MsgBox("Do you want to quit?", Buttons:=vbYesNo)
'if the user click Yes then unload the Manager_Page
If Quit = vbYes Then VBA.Unload Manager_Page
End Sub
```

Fig. 3.19 (reset button)

```
Private Sub btnExit1_Click()
Dim Quit As String
'appear the messagebox: Do you want to quit? and Yes No question
Quit = VBA MsgBox("Do you want to quit?", Buttons:=vbYesNo)
'if the user click Yes then unload the Manager_Page
If Quit = vbYes Then VBA.Unload Manager_Page
End Sub
```

Fig. 3.20 (Generte report button)

```
Private Sub btnreport7_Click()
Dim wksheetReport As Worksheet
Dim wkSheet As Worksheet, wksheetcount As Long
Dim cellProductSales As Range
Dim cellProductReport As Range
Dim columnReport As Integer
Dim cellReport As Range
Dim strBranchName As String
Dim blnIndicator As Boolean
Dim cellSalesPerson As Range
Dim cellReportPerson As Range
Dim irange As Range
Dim cellCountrySales As Range
Dim cellCountryReport As Range
Dim ncount As Integer
Dim freq As Integer
Dim xrange As Range
Dim arange As Range
```

Fig. 3.21 (Generte report button)

```
If chkpackageablechart.Value = False And chksalespersonstablechart.Value = False And chkcountry.Value = False And chkgender.Value = False And chkgae.Value = False Then
    'Prompt user to select at least one report to proceed
    VBA MsgBox "Select at least one report to proceed."
    Exit Sub
End If

'Add a new workbook for the reports and set it as wkbookReport if it is not yet been added before
If wkbookReport Is Nothing Then
    Set wkbookReport = Workbooks.Add
End If

'Keep the wobookReport invisible until the end of this sub
'wkbookReport.Windows(1).Visible = False

'Make wkbookReportvisible
wkbookReport.Windows(1).Visible = True

'Keep this workbook on top of the others
ThisWorkbook.Activate
```

Fig. 3.22 (Generte report button)

```

If chkpackageablechart.Value = True Then 'Prepare total sales report
    'Prepare the report worksheet in wkbookReport
    '***** Insert a worksheet to wkbookReport and set it as wksheetReport
    Set wksheetReport = wkbookReport.Worksheets.Add

    'Name the added worksheet Product Sales Report or Product Sales ReportX
    wksheetcount = 0
    For Each wkSheet In wkbookReport.Worksheets
        If wkSheet.Name Like "Product Sales Report*" Then
            wksheetcount = wksheetcount + 1
        End If
    Next wkSheet

    If wksheetcount > 0 Then
        wksheetReport.Name = "Product Sales Report" & wksheetcount + 1
    Else
        wksheetReport.Name = "Product Sales Report"
    End If

    wksheetReport.Range("a1").Value = "Product"

    columnReport = 0
    For Each cellReport In wksheetReport.Range("a1", wksheetReport.Range("xfd1").End(xlToLeft)).Cells
        If cellReport.Value = strBranchName Then
            'The branch name is already in wksheetReport
            columnReport = cellReport.Column
            Exit For
        End If
    Next cellReport

    If columnReport = 0 Then
        'The branch name is not in wksheetReport, put the branch name into the last column of the first row
        columnReport = 1 + wksheetReport.Range("a1", wksheetReport.Range("xfd1").End(xlToLeft)).Columns.Count
        wksheetReport.Cells(columnReport).Value = strBranchName
    End If

```

Fig. 3.23 (Generte report button)

```

'Accumulate the sales cell
For Each cellProductSales In wscustomer.Range("e2", wscustomer.Range("e1048576").End(xlUp)).Cells
    blnIndicator = False

    For Each cellProductReport In wksheetReport.Range("a1", wksheetReport.Range("a1048576").End(xlUp)).Cells
        'Compare the product code in the selected sales worksheet to the product code kept in the wksheetReport
        If cellProductSales.Value = cellProductReport.Value Then
            'A match of product codes is found, update the respective cell in wksheetReport
            cellProductReport.Offset(0, columnReport - 1).Value = cellProductReport.Offset(0, columnReport - 1).Value _
                + cellProductSales.Offset(0, 1).Value
            blnIndicator = True
            Exit For
        End If
    Next cellProductReport

    If blnIndicator = False Then
        'A match of product code is not found, put the product code to the end of the product list of wksheetReport
        wksheetReport.Range("a1048576").End(xlUp).Offset(1, 0).Value = cellProductSales.Value
        wksheetReport.Range("a1048576").End(xlUp).Offset(0, columnReport - 1).Value = cellProductSales.Offset(0, 1).Value
    End If
Next cellProductSales

'Adjust the column widths
wksheetReport.Range("a1").CurrentRegion.Columns.AutoFit

'Sort the product codes in ascending order
wksheetReport.Range("A1").CurrentRegion.Sort Key1:=wksheetReport.Columns(1), _
    Order1:=xlAscending, Header:=xlYes

```

Fig. 3.24 (Generte report button)

```

'create chart
Set i range = wksheetReport.Range("a2", wksheetReport.Range("a1048576").End(xlUp).End(xlToLeft))
wksheetReport.Activate
ActiveSheet.Shapes.AddChart2(201, xlColumnClustered).Select
    ActiveChart.SetSourceData Source:=i range
    ActiveChart.ChartTitle.Caption = "Product"
    I

End If

If chksalespersontablechart.Value = True Then  'Prepare the sales report for each salesperson
Set wstour = wbcustomerrecord.Worksheets("Tour_detail")

'Prepare the report worksheet in wkbookReport
'Insert a worksheet to wkbookReport and set it as wksheetReport
Set wksheetReport = wkbookReport.Worksheets.Add

'Name the added worksheet Salespersons Report or Salespersons ReportX
wksheetcount = 0
For Each wkSheet In wkbookReport.Worksheets
    If wkSheet.Name Like "Salespersons Report*" Then
        wksheetcount = wksheetcount + 1
    End If
Next wkSheet
If wksheetcount > 0 Then
    wksheetReport.Name = "Salespersons Report" & wksheetcount + 1
Else
    wksheetReport.Name = "Salespersons Report"
End If

'wksheetReport.Range("a1").Value = "Branch"
wksheetReport.Range("a1").Value = "Salesperson"
wksheetReport.Range("b1").Value = "Total Sales"

```

Fig. 3.25 (Generte report button)

```

'Accumulate the sales cell
For Each cellSalesPerson In wscustomer.Range("g2", wscustomer.Range("g1048576").End(xlUp)).Cells
    blnIndicator = False

    For Each cellReportPerson In wksheetReport.Range("a1", wksheetReport.Range("a1048576").End(xlUp)).Cells
        'Compare the salesperson name in the selected sales worksheet to the salesperson name kept in the wksheetReport
        If cellSalesPerson.Value = cellReportPerson.Value Then
            'A match of names is found, update the respective total sales cell in wksheetReport
            cellReportPerson.Offset(0, 1).Value = cellReportPerson.Offset(0, 1).Value + cellSalesPerson.Offset(0, -1).Value
            blnIndicator = True
            Exit For
        End If
    Next cellReportPerson

    If blnIndicator = False Then
        'A match of names is not found, put the name to the end of the salesperson list of wksheetReport
        wksheetReport.Range("a1048576").End(xlUp).Offset(1, 0).Value = cellSalesPerson.Value
        wksheetReport.Range("b1048576").End(xlUp).Offset(1, 0).Value = cellSalesPerson.Offset(0, -1).Value
    End If
Next cellSalesPerson

'Enhance the layout of the wksheetReport
*****
'Set the format Total Sales column
wksheetReport.Columns(3).NumberFormat = "$#,##0.00"
I

'Adjust the column widths
wksheetReport.Range("a1").CurrentRegion.Columns.AutoFit

'Sort the report by Branch and then salesperson in ascending order
wksheetReport.Range("A1").CurrentRegion.Sort Key1:=wksheetReport.Columns(1),
    Order1:=xlAscending, key2:=wksheetReport.Columns(2), order2:=xlAscending, Header:=xlYes

```

Fig. 3.26 (Generte report button)

```

'Adjust the column widths
wksheetReport.Range("a1").CurrentRegion.Columns.AutoFit

'Sort the report by Branch and then salesperson in ascending order
wksheetReport.Range("A1").CurrentRegion.Sort Key1:=wksheetReport.Columns(1), _
    Order1:=xlAscending, key2:=wksheetReport.Columns(2), order2:=xlAscending, Header:=xlYes

'create chart

Set iRange = wksheetReport.Range("a1").CurrentRegion

wksheetReport.Activate
ActiveSheet.Shapes.AddChart2(201, xlColumnClustered).Select
ActiveChart.SetSourceData Source:=iRange
ActiveChart.ChartTitle.Caption = "Salesperson"

End If

If chkCountry.Value = True Then 'Prepare total Package quantity report
    'Prepare the report worksheet in wbkbookReport
    '*****
    'Insert a worksheet to wbkbookReport and set it as wksheetReport
    Set wksheetReport = wbkbookReport.Worksheets.Add

    'Name the added worksheet Package quantity report
    wksheetCount = 0
    For Each wkSheet In wbkbookReport.Worksheets
        If wkSheet.Name Like "Package quantity Report*" Then
            wksheetCount = wksheetCount + 1
        End If
    Next wkSheet
    If wksheetCount > 0 Then
        wksheetReport.Name = "Package quantity Report" & wksheetCount + 1
    Else
        wksheetReport.Name = "Package quantity Report"
    End If
    wksheetReport.Range("a1").Value = "Product"

```

Fig. 3.27 (Generte report button)

```

columnReport = 0
For Each cellReport In wksheetReport.Range("a1", wksheetReport.Range("xfd1").End(xlToLeft)).Cells
    If cellReport.Value = strBranchName Then
        'The branch name is already in wksheetReport
        columnReport = cellReport.Column
        Exit For
    End If
Next cellReport

If columnReport = 0 Then
    'The branch name is not in wksheetReport, put the branch name into the last column of the first row
    columnReport = 1 + wksheetReport.Range("a1", wksheetReport.Range("xfd1").End(xlToLeft)).Columns.Count
    wksheetReport.Cells(columnReport).Value = strBranchName
End If

'arrange the package ID to wksheetReport
For Each cellProductSales In wscustomer.Range("e2", wscustomer.Range("e1048576").End(xlUp)).Cells
    bInIndicator = False

    For Each cellProductReport In wksheetReport.Range("a1", wksheetReport.Range("a1048576").End(xlUp)).Cells
        'Compare the product code in the selected sales worksheet to the product code kept in the wksheetReport
        If cellProductSales.Value = cellProductReport.Value Then

            bInIndicator = True
            Exit For
        End If
    Next cellProductReport

    If bInIndicator = False Then
        'A match of product code is not found, put the product code to the end of the product list of wksheetReport
        wksheetReport.Range("a1048576").End(xlUp).Offset(1, 0).Value = cellProductSales.Value
    End If
Next cellProductSales

```

Fig. 3.27 (Generte report button)

```

wksheetReport.Range("b2").Value = Application.CountIf(wscustomer.Range("E2:E1048576"), wksheetReport.Range("a2"))
wksheetReport.Range("b3").Value = Application.CountIf(wscustomer.Range("E2:E1048576"), wksheetReport.Range("A3"))
wksheetReport.Range("b4").Value = Application.CountIf(wscustomer.Range("E2:E1048576"), wksheetReport.Range("A4"))
wksheetReport.Range("b5").Value = Application.CountIf(wscustomer.Range("E2:E1048576"), wksheetReport.Range("A5"))
wksheetReport.Range("b6").Value = Application.CountIf(wscustomer.Range("E2:E1048576"), wksheetReport.Range("A6"))
wksheetReport.Range("b7").Value = Application.CountIf(wscustomer.Range("E2:E1048576"), wksheetReport.Range("A7"))
wksheetReport.Range("b8").Value = Application.CountIf(wscustomer.Range("E2:E1048576"), wksheetReport.Range("A8"))
wksheetReport.Range("b9").Value = Application.CountIf(wscustomer.Range("E2:E1048576"), wksheetReport.Range("A9"))
wksheetReport.Range("b10").Value = Application.CountIf(wscustomer.Range("E2:E1048576"), wksheetReport.Range("A10"))
wksheetReport.Range("b11").Value = Application.CountIf(wscustomer.Range("E2:E1048576"), wksheetReport.Range("A11"))
wksheetReport.Range("b12").Value = Application.CountIf(wscustomer.Range("E2:E1048576"), wksheetReport.Range("A12"))
wksheetReport.Range("b13").Value = Application.CountIf(wscustomer.Range("E2:E1048576"), wksheetReport.Range("A13"))
wksheetReport.Range("b14").Value = Application.CountIf(wscustomer.Range("E2:E1048576"), wksheetReport.Range("A14"))

'Adjust the column widths
wksheetReport.Range("a1").CurrentRegion.Columns.AutoFit

'Sort the product codes in ascending order
wksheetReport.Range("A1").CurrentRegion.Sort Key1:=wksheetReport.Columns(1), _
    Order1:=xlAscending, Header:=xlYes

'create chart
wksheetReport.Activate
Range("A2:B14").Select
ActiveSheet.Shapes.AddChart2(251, xlPie).Select
ActiveChart.SetSourceData Source:=Range("Package quantity Report!$A$2:$B$14")

End If

```

Fig. 3.28 (Generte report button)

```
If chkgender.Value = True Then 'Prepare gender report
'Prepare the report worksheet in wkbookReport
'*****
'Insert a worksheet to wkbookReport and set it as wksheetReport
Set wksheetReport = wkbookReport.Worksheets.Add

'Name the added worksheet gender report
wksheetcount = 0
For Each wkSheet In wkbookReport.Worksheets
    If wkSheet.Name Like "Gender Report*" Then
        wksheetcount = wksheetcount + 1
    End If
Next wkSheet

If wksheetcount > 0 Then
    wksheetReport.Name = "Gender Report" & wksheetcount + 1
Else
    wksheetReport.Name = "Gender Report"
End If

wksheetReport.Range("a1").Value = "Gender"
wksheetReport.Range("a2").Value = wscustomer.Range("C2")
wksheetReport.Range("a3").Value = wscustomer.Range("C3")

wksheetReport.Range("b2").Value = Application.CountIf(wscustomer.Range("c2:c1048576"), wscustomer.Range("C2"))
wksheetReport.Range("b3").Value = Application.CountIf(wscustomer.Range("c2:c1048576"), wscustomer.Range("C3"))

'Adjust the column widths
wksheetReport.Range("a1").CurrentRegion.Columns.AutoFit

'Sort the product codes in ascending order
wksheetReport.Range("A1").CurrentRegion.Sort Key1:=wksheetReport.Columns(1), _
    Order1:=xlAscending, Header:=xlYes
```

Fig. 3.29 (Generte report button)

```
'create chart
Set xrange = wksheetReport.Range("a2", wksheetReport.Range("a1048576").End(xlUp).End(xlToLeft))
wksheetReport.Activate
ActiveSheet.Shapes.AddChart2(201, xlColumnClustered).Select
ActiveChart.SetSourceData Source:=xrange
ActiveChart.ChartTitle.Caption = "Gender"

End If

If chkage.Value = True Then 'Prepare age report
    'Prepare the report worksheet in wkbookReport
    'Insert a worksheet to wkbookReport and set it as wksheetReport
    Set wksheetReport = wkbookReport.Worksheets.Add

    'Name the added worksheet age report
    wksheetcount = 0
    For Each wkSheet In wkbookReport.Worksheets
        If wkSheet.Name Like "age Report*" Then
            wksheetcount = wksheetcount + 1
        End If
    Next wkSheet

    If wksheetcount > 0 Then
        wksheetReport.Name = "age Report" & wksheetcount + 1
    Else
        wksheetReport.Name = "age Report"
    End If

    wksheetReport.Range("a1").Value = "age"
    wksheetReport.Range("b1").Value = "value count"
    wksheetReport.Range("c1").Value = "% of total"
    wksheetReport.Range("a2").Value = "Young"
    wksheetReport.Range("a3").Value = "Middle-aged"
    wksheetReport.Range("a4").Value = "Senior"
```

Fig. 3.30 (Generte report button)

```
wksheetReport.Range("b2").Value = Application.CountIf(wscustomer.Range("d2:d1048576"), ">=19") - Application.CountIf(wscustomer.Range("d2:d1048576"), ">29")
wksheetReport.Range("b3").Value = Application.CountIf(wscustomer.Range("d2:d1048576"), ">=29") - Application.CountIf(wscustomer.Range("d2:d1048576"), ">39")
wksheetReport.Range("b4").Value = Application.CountIf(wscustomer.Range("d2:d1048576"), ">=39") - Application.CountIf(wscustomer.Range("d2:d1048576"), ">49")

wksheetReport.Range("c2").Value = FormatPercent(wksheetReport.Range("b2").Value / (wksheetReport.Range("b2").Value + wksheetReport.Range("b3").Value))
wksheetReport.Range("c3").Value = FormatPercent(wksheetReport.Range("b3").Value / (wksheetReport.Range("b2").Value + wksheetReport.Range("b3").Value))
wksheetReport.Range("c4").Value = FormatPercent(wksheetReport.Range("b4").Value / (wksheetReport.Range("b2").Value + wksheetReport.Range("b3").Value + wksheetReport.Range("b4").Value))

'Adjust the column widths
wksheetReport.Range("a1").CurrentRegion.Columns.AutoFit

'Sort the product codes in ascending order
wksheetReport.Range("A1").CurrentRegion.Sort Key1:=wksheetReport.Columns(1), _
    Order1:=xlAscending, Header:=xlYes

'create pie chart
Set arange = wksheetReport.Range("a2", wksheetReport.Range("a1048576").End(xlUp).End(xlToLeft))
Range("A1:C4").Select
wksheetReport.Activate
ActiveSheet.Shapes.AddChart2(251, xlPie).Select
ActiveChart.SetSourceData Source:=arange

End If
```

Fig. 3.31 (Generte report button)

```
'show message when the report is generated
VBA MsgBox "Report is generated. Click OK to continue and quit the application to view the report. "
ThisWorkbookActivate

End Sub
```

Fig. 3.32 (Save and close button)

```
Private Sub btnSaveandclose_Click()
    'save the report
    wkbookReport.Save
    wkbookReport.Close
    VBA MsgBox ("Report has been saved and closed")

End Sub
```