VBA Final Project

Day Planner

4/12/2010

Howard Kuan, MBA 614, Winter 2010

Executive Summary

My project was to facilitate the access of one's database of contacts and history. This tool is not only effective in action items searching but also efficient in making priorities for your time. The database includes customers, friends and family. This can greatly help you at your work, job searching, networking and balance among friends and family. Hope you find this tool helpful and make the most use of it to achieve your life goals.

1. Solution to the business problem

As a sales or project manager, it is always a pain to list all the things that should be done in a day and prioritize them. This program "Day Planner" provides a solution to these people. The benefits are listed in the following:

- Prevents you from missing action items by listing ALL the action items to date.
- You can choose the priority by yourself. The list in turn is rearranged into your order.
- You can update your memo to the existing database for your reference in the future.
- In order to balance your life in networking and family, the "Day Planner" tool contains friends and family (especially for big Mormon families) categories.
- The number of items is limited to 10 for a more effective priority. After you finish the first 10 items, you can run this tool again to deal with the next 10 item. The numbers for friends and family are limited to 5.

2. User manual

1) ***Search*** - When you open the "Day Planner" Excel file, there are no action items on the list. You can press "Start" to search the action items for today, or press "clear" to delete all the items (as shown in **Fig.1**). The search criteria can be customerized in each category (default is 0, means the

current date). You can put a positive integer for a future due date, or a negative integer for a past due date. Also please make sure there are comments for the dates in your database.

2) ***Prioritize*** - With the sifted data from the database, you can decide the priority of each item according to the importance and the history, as illustrated in **Fig.2~Fig.4**. The importance is also used later to add the time to your next due day. You can also change the importance by clicking the option button. When you click on the option button, it saves the priority to a cell of the array. Then when you click the Make List button, "Day Planner" tool can refer the priority and the map to make the right order and list them in the main menu, as depicted in **Fig.5**.

3) ***Update database*** – When you check the checkbox after you finish an item, a memo window will pop up, as shown in **Fig.6**. You have to put information for both the memo for today's encounter and the instruction for the next contact. You can also set the date of next contact, which is optional. "Day Planner" will set the date for you automatically by referring to the customer's importance. After you complete the memo, the old item is then cleared and a confirmation message "Memo successfully updated" will show up, as illustrated in **Fig.7**.

3. VBA code description

The three categories are defined by arrays. Here we only illustrate customer category. It is intuitive to understand the other two after going through this category.

1) Search – This part is comprised of three parts, loadData ( ), findFirst ( ), and showData ( ).

```
Sub loadData ( )

cboCustomer.Value = ""
Do Until c.Cells(currentRow(0), 1).Value = ""
   If c.Cells(currentRow(0), 6).Value <= Date + offset(0) Then
      cboCustomer.AddItem c.Cells(currentRow(0), 1).Value & ", " & _
      c.Cells(currentRow(0), 3).Value
      mapC(index(0), 0) = currentRow(0)        'this maps the list of the current row
```

```
        mapC(index(0), 1) = 5                'default priority button 5
        index(0) = index(0) + 1              'this tells how many indexes are qualified
        currentRow(0) = currentRow(0) + 1
      Else: currentRow(0) = currentRow(0) + 1
      End If
    Loop
```

- loadData ( ) codes keep searching all the data that satisfied our criteria and adding these data to the combo box by using Do loop and IF statement. There is a key in there, that is, another array is used to map the new order to the database.

```
Sub findFirst ( )

If c.Cells(2, 6).Value <= Date + offset(0) Then
   currentRow(0) = 2
   found = True
   Else:
      currentRow(0) = 2
    Do
      currentRow(0) = currentRow(0) + 1
      found = True
      If c.Cells(currentRow(0), 1).Value = "" Then
        found = False
        Exit Do
      End If
    Loop Until c.Cells(currentRow(0), 6).Value <= Date + offset(0)
  End If
```

- Then findFirst ( ) codes simply find the first qualified data and make this ready to be displayed in the combo box, via IF statement and Do loop.

```
Sub showData ( )

Dim i As Byte

If found = False Then Exit Sub   'if no qualified data, then quit
   showingData = True

   For y = 0 To 6
    cust(y) = c.Cells(currentRow(0), y + 1).Value
   Next

   cboCustomer.Text = c.Cells(currentRow(0), 1).Value & ", " & c.Cells(currentRow(0), 3)
```

```
            TimeC.Caption = c.Cells(currentRow(0), currentColumn(0)).Value
            dayOfWeekC1.Caption = WeekdayName(Weekday(TimeC.Caption))
            dueC.Caption = c.Cells(currentRow(0), 6).Value
            dayOfWeekC2.Caption = WeekdayName(Weekday(dueC.Caption))
            historyC = c.Cells(currentRow(0), currentColumn(0)).comment.Text
            historyC.Text = Mid(historyC.Text, InStr(1, historyC.Text, vbNewLine) + 2)
            actC = c.Cells(currentRow(0), 6).comment.Text
            actC.Text = Mid(actC.Text, InStr(1, actC.Text, vbNewLine) + 2)
          End If
```

- showData ( ) codes base on the row to grab the row data from the database, then display them in all the text boxes (history, action) and labels (dates, day of week).

2) Prioritize – This part is mainly makeListC ( ) sub. The codes repeatedly get data from the database to paste them in the main menu, in your requested order. The codes refer to the priority array and mapping array to get the data order correctly. A count indicator is used to limit the number of items.

```
Private Sub makeListC_Click( )
clearC
count = 1

For z = 0 To 4
   For x = 0 To index(0) - 1        '-1 is to match the array
    If mapC(x, 1) = z + 1 Then
     currentRow(0) = mapC(x, 0)

     For y = 0 To 6
      t.Cells(count + 2, y + 2).Value = c.Cells(currentRow(0), y + 1).Value
      Cells.Columns.AutoFit
     Next

      t.Cells(count + 2, 8).AddComment c.Cells(currentRow(0), 7).comment.Text
      t.Cells(count + 2, 7).AddComment c.Cells(currentRow(0), 6).comment.Text
     count = count + 1
    End If
    If count = 11 Then Exit For   'limit to 10 jobs
   Next
  Next
   Worksheets("Today").Range("rangeC").HorizontalAlignment = xlCenter
End Sub
```

3) Update database – This part is also comprised of three parts, checkC1 ( ), inputPage_Initialize ( ), and getData_Click ( ).

```
Sub checkC1( )
If Cells(updateRow(0), 2) <> "" And Cells(updateRow(0), 8) <> "" Then

   findUpdateRowC
   Unload inputPage
   inputPage.Show

   If gettingData = False Then
        Do Until gettingData = True
          MsgBox "Please help yourself"
          inputPage.Show
        Loop
   End If
InsertC
```

- checkC1 ( ) codes call the user form to match the data and force user to put in messages for the reference in the future.

```
Sub UserForm_Initialize( )
If comparison(7) = "customer" Then
   Select Case comparison(1)              'get the customer importance
     Case Is = "A"
       priority = customerA
     Case Is = "B"
       priority = customerB
     Case Is = "C"
       priority = customerC
   End Select
 End If

nextDate.Text = Date + priority          'set next date by referring to the customer importance
nextDateValue = nextDate
dayOfWeek.Caption = WeekdayName(Weekday(nextDate.Text))
```

- UserForm_Initialize ( ) codes simply select the priority of the item by the Select Case statement, and then set the date of next contact.

```
Private Sub getData_Click( )

  If historyBox = "" Or actionBox = "" Then
```

```
    MsgBox "Please help yourself"    'msg box forces you to put some words in there
    Exit Sub
  Else
    history = historyBox.Text
    action = actionBox.Text
    historyBox.Text = ""
    actionBox.Text = ""
    'nextDate.Text = ""
  End If

  gettingData = True
  inputPage.Hide

End Sub
```

- getData_Click ( ) codes again force the user to put information in both text boxes. The purpose is to remind the user when next time they use it.

```
Sub InsertC()

Sheets("Customers").Cells(updateRow(0), 7).Insert Shift:=xlToRight
Sheets("Customers").Cells(updateRow(0), 7).Value = Date
Sheets("Customers").Cells(updateRow(0), 6).Value = nextDateValue
Sheets("Customers").Cells(updateRow(0), 7).AddComment " " & history
Sheets("Customers").Cells(updateRow(0), 6).comment.Delete
Sheets("Customers").Cells(updateRow(0), 6).AddComment " " & action

End Sub
```

- InsertC ( ) codes insert a new cell to the database, to add the date of the action accomplished. Then the codes also add comments as a memo of today's encounter and an instruction of next contact.

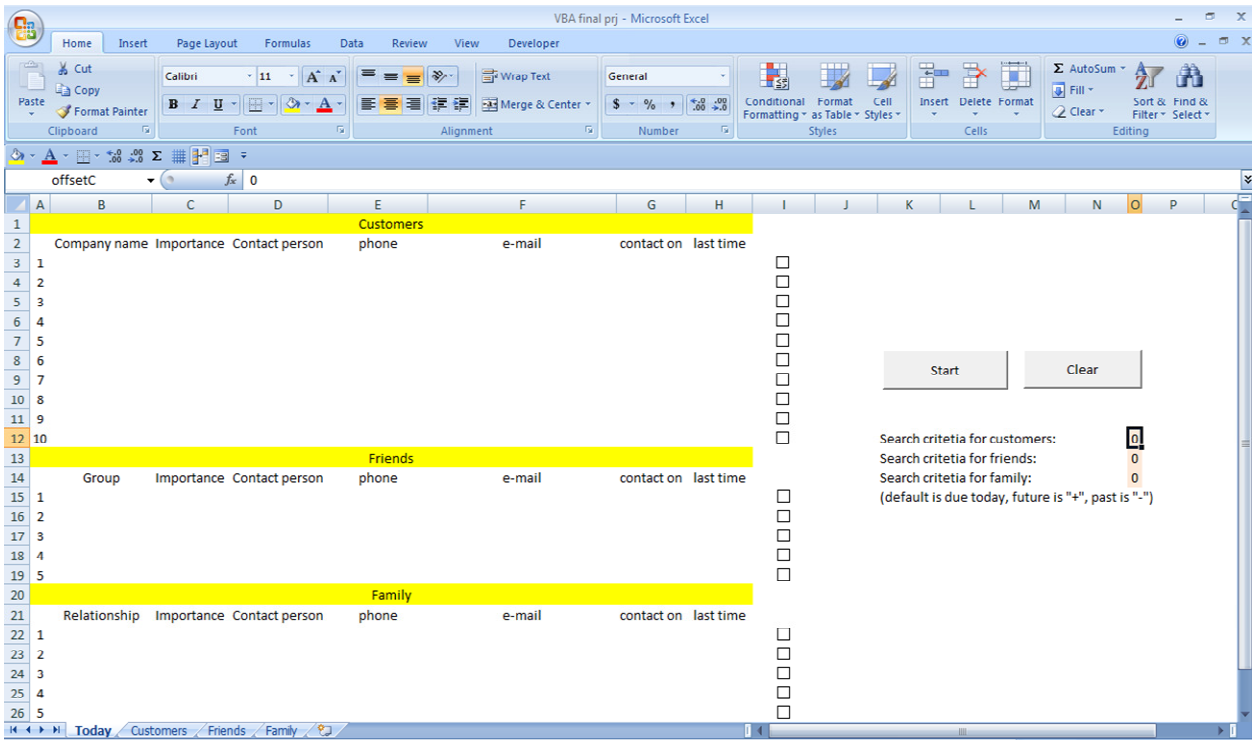

Figure.1 Initial main menu

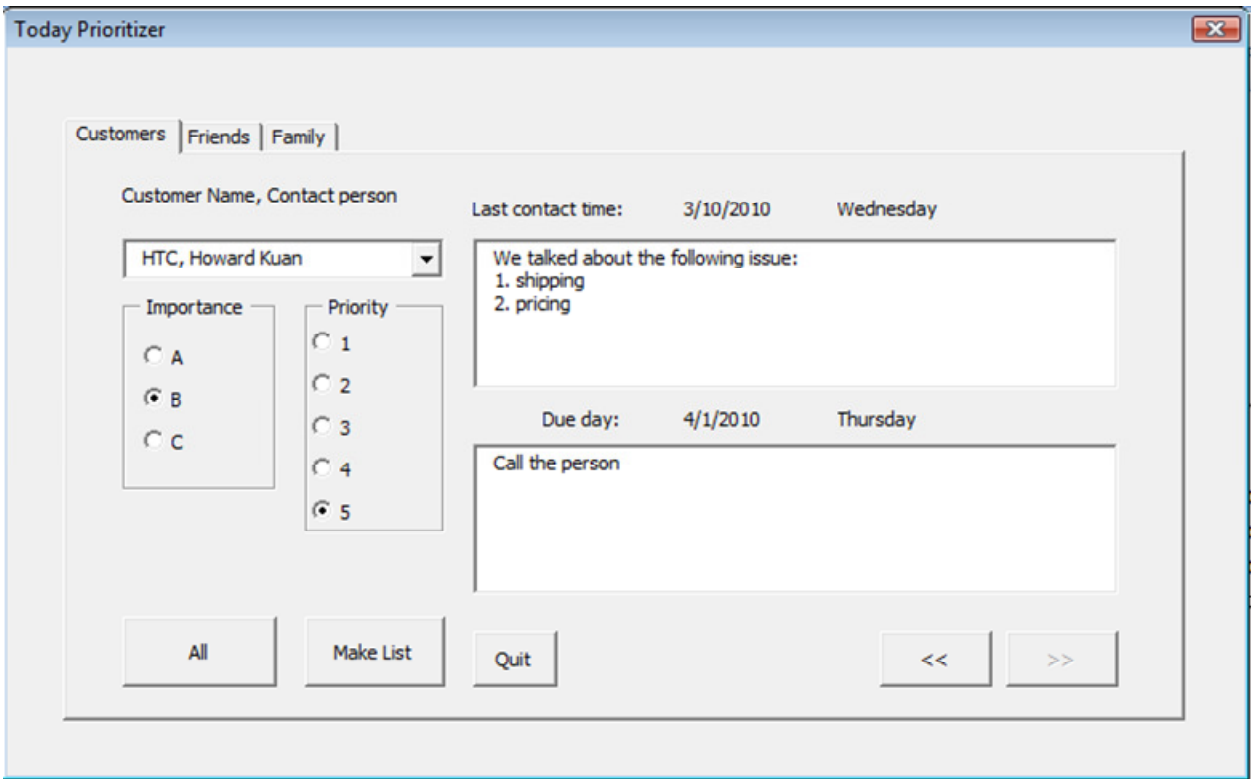

Figure.2 Customer page

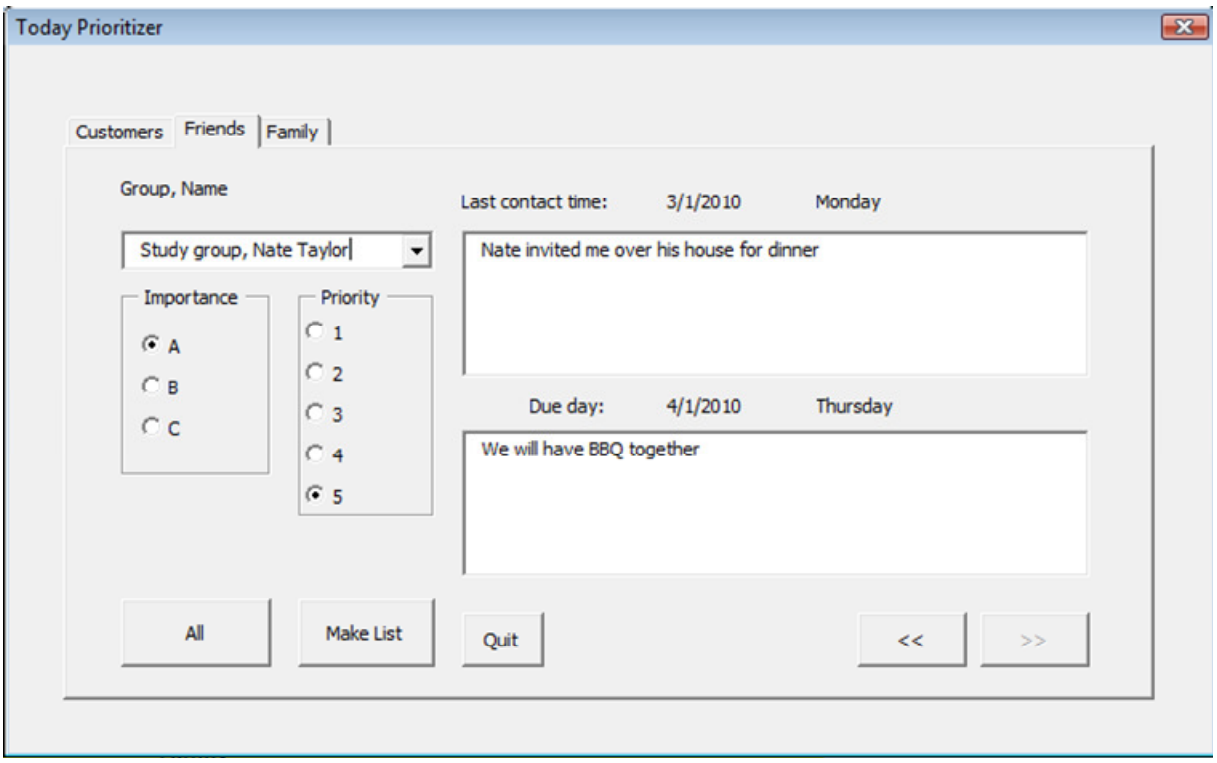

Figure.3 Friends page

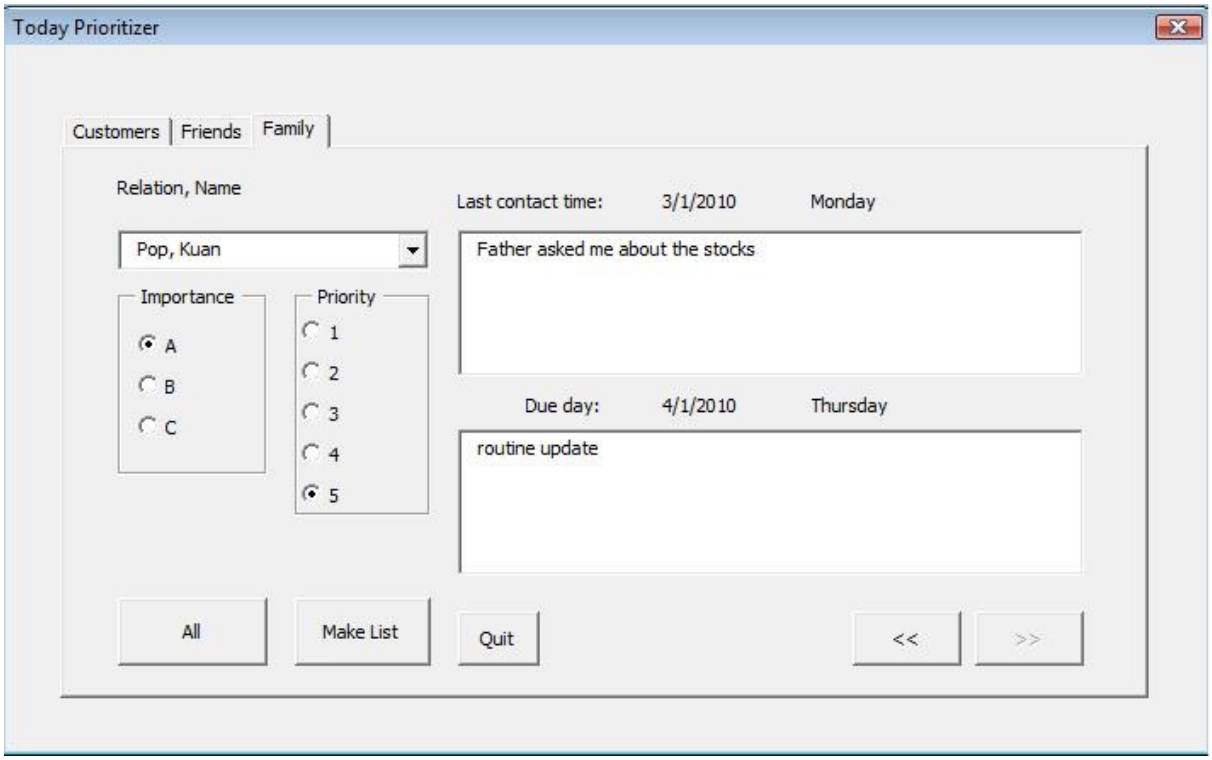

Figure.4 Family page

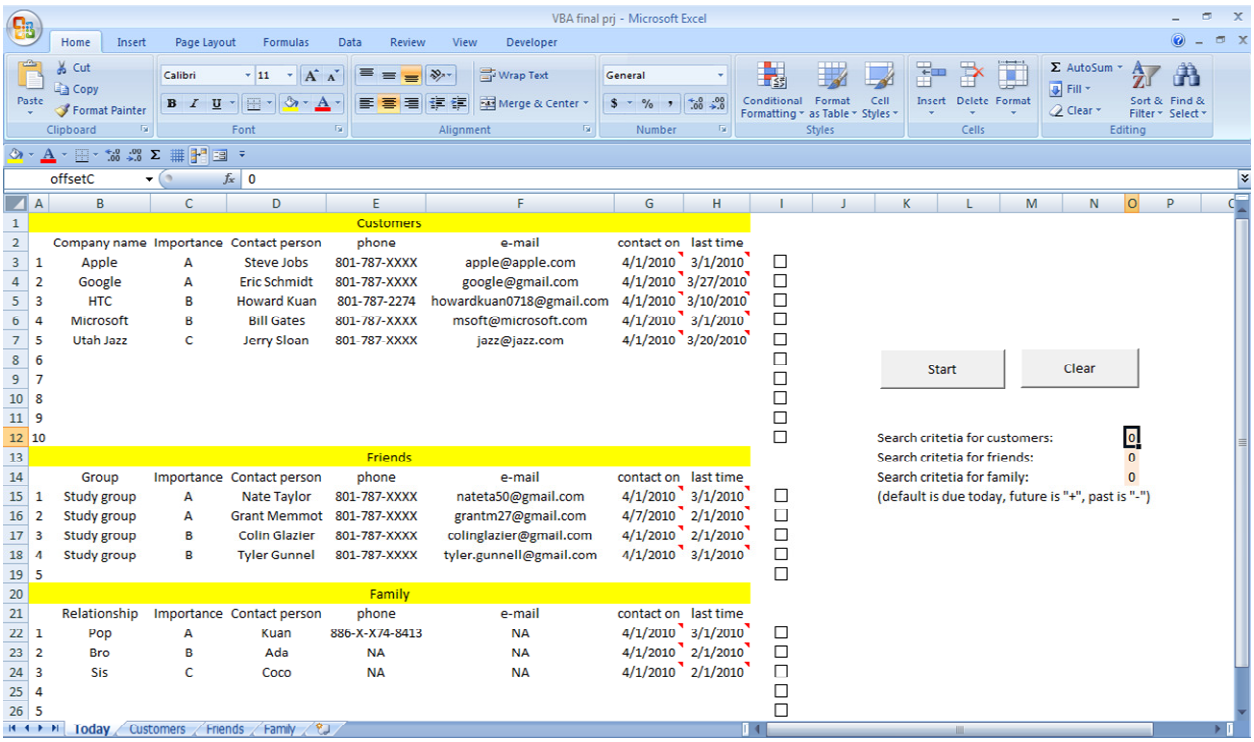

Figure.5 Prioritized list

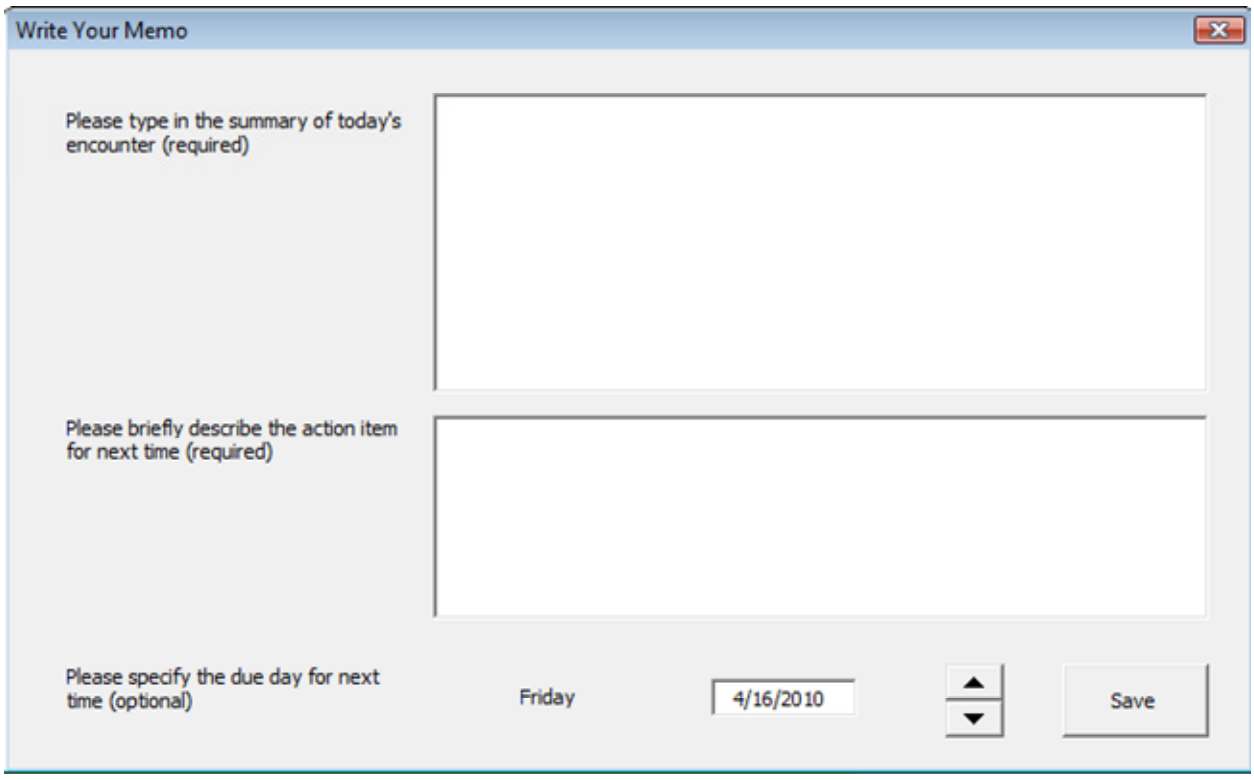

Figure.6 Memo for completion

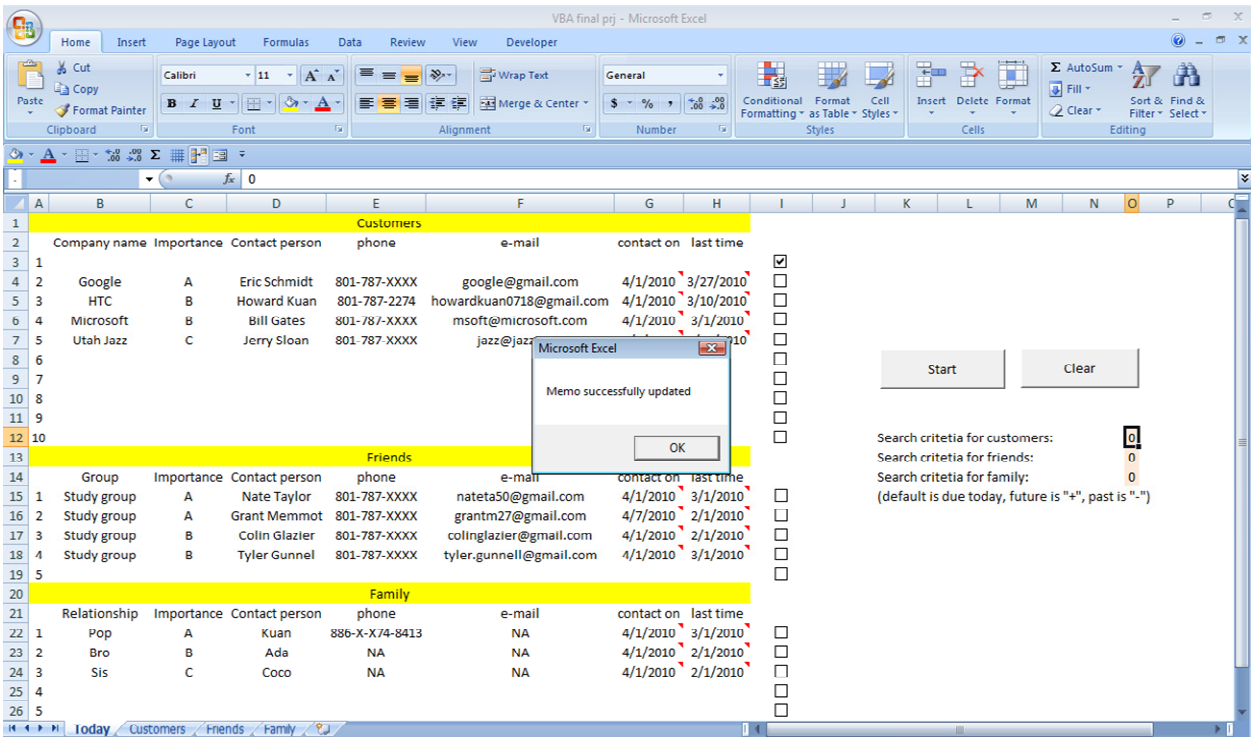

Figure.7 Updated successfully