Wedding Directory Mission Control

Nathan Beck
MBA 614 Section 2
*VBA Final Project*

Executive Summary

I am currently engaged and didn't quite realize the whirlwind that was going to overcome me during this time. It isn't so much the engagement that is overwhelming but everything going on in addition to the engagement: semester end projects, finals, job fly outs, etc. I decided to use my VBA final project to do something that will truly be useful for me and others involved in the wedding planning.

Sending out wedding invites is a big deal. It requires coordinating the efforts of a lot of different groups. After several weeks of several different directory versions flying around between the two families, I decided to create a user form that will be so simple that no one could possibly screw it up. I created a user form so that regardless of how tech savvy an individual may or may not be, they would be able to easily check if an individual is in the directory, add an individual, and know the count of guests according to different categories. The mission is clear. Now for the difficult part: implementation.

Implementation

After I decided on the three main functions (search, add, and count) that I needed to provide, I turned my focus to usability. I knew that I would need to make my project easy enough for anyone to use it. I decided that I needed simple, clean user interfaces and big obvious buttons.

Insert button into the ribbon
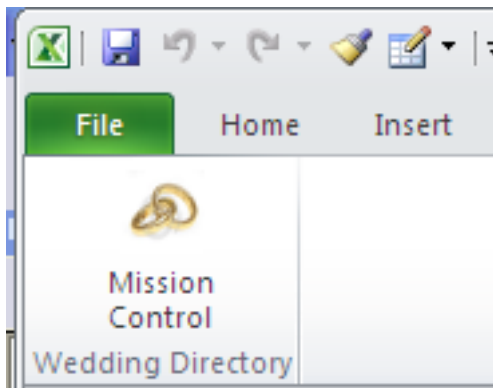
I placed a button in the ribbon because I knew that no matter the functionality or capability of my user form, if a user cannot navigate the Visual Basic editor in order to find the application all the benefit would be lost.

Although the RibbonWizard is well organized and for the most part intuitive to use, a deep understanding of the structure of VBA code was still necessary. It took a couple of minutes for me to successfully navigate the modules and find sub-function titles in order to make it all work. Nonetheless, I feel that having an easy to access button on the ribbon will ensure that it gets used.

I decided that frames and clear label would provide the cleanest organization. Each text box is clearly labeled and sectioned off with supporting labels, text boxes, and command buttons

Search sub procedure

With multiple people adding information to the directory, I needed to include an easy way to search the directory to see whether or not a record is already included. The Search box in the top left corner of the User Form allows the user to add any part of the individual's record and check if that person is already listed. Additionally, if multiple people with the same information (last name for example) are in the directory, I have built in the functionality to search for the next entry that meets that criterion.

```
Private Sub find()

  Dim x As Integer

  x = ActiveCell.Row + 1

  Do While Len(Cells(x, 1)) > 0    ' Could also use a Do until loop

    If (TxtName = "" And TxtAddress1 = "" And TxtAddress2 = "" And cboState =
      MsgBox ("You must enter a search term")
      Exit Do
    End If

    If InStr(1, Sheets("AddressFinal").Cells(x, 1), TxtName) > 0 And _
      InStr(1, Sheets("AddressFinal").Cells(x, 2), TxtAddress1) > 0 And _
      InStr(1, Sheets("AddressFinal").Cells(x, 3), TxtAddress2) > 0 And _
      InStr(1, Sheets("AddressFinal").Cells(x, 4), TxtCity) > 0 And _
      InStr(1, Sheets("AddressFinal").Cells(x, 5), cboState) > 0 And _
      InStr(1, Sheets("AddressFinal").Cells(x, 6), TxtZip) > 0 Then

      Cells(x, 1).Select

      TxtName = Sheets("AddressFinal").Cells(x, 1).Value
      TxtAddress1 = Sheets("AddressFinal").Cells(x, 2).Value
      TxtAddress2 = Sheets("AddressFinal").Cells(x, 3).Value
      TxtCity = Sheets("AddressFinal").Cells(x, 4).Value
      cboState = Sheets("AddressFinal").Cells(x, 5).Value
      TxtZip = Sheets("AddressFinal").Cells(x, 6).Value
      Exit Sub

    End If

    x = x + 1
  Loop
```

Add sub procedure

Maintaining the structure and formatting of a directory is crucial. I decided that the best way to control these two components is to designate a portion of my user form to adding names to the directory. This way I can control what cells the information is placed into. When information is placed in any of the fields it is saved to the directory in the next available spot at the bottom of the list.

```
Private Sub CmdAdd_Click()

  Dim c As Integer

  Do Until Sheets("AddressFinal").Cells(c, 1) = ""

    Sheets("AddressFinal").Cells(c, 1).Value = TextNameAdd
    Sheets("AddressFinal").Cells(c, 2).Value = TextAddress1Add
    Sheets("AddressFinal").Cells(c, 3).Value = TextAddress2Add
    Sheets("AddressFinal").Cells(c, 4).Value = TextCityAdd
    Sheets("AddressFinal").Cells(c, 5).Value = ComboBoxStateAdd
    Sheets("AddressFinal").Cells(c, 6).Value = TextZipAdd

  Loop

End Sub
```

Count sub procedure

An important piece of information with wedding directories is how many people are coming and who invited whom. On the right hand side of the user form counts of guests are listed by categories as well as by grand totals. This will provide additional material for the two new families to argue over regarding who is costing the other the most money.

I created all of this code myself with the help of James. The portion of code for the Count sub procedure represents the most difficult portion of code of the project and that I have written during the course.

```vba
Private Sub CmdCount_Click()

    Dim x As Integer
    Dim NImmediateFamilyCount As Integer
    Dim NExtendedFamilyCount As Integer
    Dim NCloseFriendsCount As Integer
    Dim NFriendsCount As Integer
    Dim NFamilyFriendsCount As Integer
    Dim EImmediateFamilyCount As Integer
    Dim EExtendedFamilyCount As Integer
    Dim ECloseFriendsCount As Integer
    Dim EFriendsCount As Integer
    Dim EFamilyFriendsCount As Integer


    NImmediateFamilyCount = 0
    NExtendedFamilyCount = 0
    NCloseFriendsCount = 0
    NFriendsCount = 0
    NFamilyFriendsCount = 0
    EImmediateFamilyCount = 0
    EExtendedFamilyCount = 0
    ECloseFriendsCount = 0
    EFriendsCount = 0
    EFamilyFriendsCount = 0


    x = 1

    Do Until Cells(x, 1) = ""

        If Cells(x, 8) = "Emily - Close Friends" Then
            ECloseFriendsCount = ECloseFriendsCount + 1
        End If

        If Cells(x, 8) = "Emily - Friends" Then
            EFriendsCount = EFriendsCount + 1
        End If

        If Cells(x, 8) = "Emily - Family Friends" Then
            EFamilyFriendsCount = EFamilyFriendsCount + 1
        End If

        x = x + 1
    Loop


    txtCountNImmediateFamily = NImmediateFamilyCount
    txtCountNExtendedFamily = NExtendedFamilyCount
    txtCountNCloseFriends = NCloseFriendsCount
    txtCountNFriends = NFriendsCount
    txtCountNFamilyFriends = NFamilyFriendsCount
    txtCountEImmediateFamily = EImmediateFamilyCount
    txtCountEExtendedFamily = EExtendedFamilyCount
    txtCountECloseFriends = ECloseFriendsCount
    txtCountEFriends = EFriendsCount
    txtCountEFamilyFriends = EFamilyFriendsCount

    txtCountGrandTotal = NImmediateFamilyCount + NExtend

End Sub
```

```vba
        If Cells(x, 8) = "Nate - Immediate Family" Then
        NImmediateFamilyCount = NImmediateFamilyCount + 1
        End If

        If Cells(x, 8) = "Nate - Extended Family" Then
        NExtendedFamilyCount = NExtendedFamilyCount + 1
        End If

        If Cells(x, 8) = "Nate - Close Friends" Then
        NCloseFriendsCount = NCloseFriendsCount + 1
        End If

        If Cells(x, 8) = "Nate - Friends" Then
        NFriendsCount = NFriendsCount + 1
        End If

        If Cells(x, 8) = "Nate - Family Friends" Then
        NFamilyFriendsCount = NFamilyFriendsCount + 1
        End If

        If Cells(x, 8) = "Emily - Immediate Family" Then
        EImmediateFamilyCount = EImmediateFamilyCount + 1
        End If

        If Cells(x, 8) = "Emily - Extended Family" Then
        EExtendedFamilyCount = EExtendedFamilyCount + 1
        End If

        If Cells(x, 8) = "Emily - Close Friends" Then
        ECloseFriendsCount = ECloseFriendsCount + 1
        End If
```

Learning and Conceptual Difficulties

Many hurdles (expected and unexpected) were encountered in the creation of this project. Any good VBA coder knows that reinventing the wheel should be avoided at all costs. The trick becomes adapting other people's code to work for the nuances of a particular project. I was able to use Project 4 from class as good building blocks upon which to build the Search and Add functionality of the User Form. Still considerable adaptation was required for the more complex wedding registry. The Count functionality was completely new to me. It required me to piece together many concepts taught in class including topics of the user forms and loops. I am most proud of the code written below:

```vba
x = 1

Do Until Cells(x, 1) = ""

If Cells(x, 8) = "Nate - Immediate Family" Then
NImmediateFamilyCount = NImmediateFamilyCount + 1
End If
```

I was able to create a loop that counted each time an entry fit a certain category. I duplicated this code for each of the different categories as well as the grand total.

I ran into many difficulties along the way. I learned that what works in one spreadsheet will not necessarily work in another spreadsheet with a simple copy and paste. I had to have a complete

understanding of each line of the code and know what parameters needed to be changed for the new spreadsheet.

Assistance

I received considerable help from the TA, James, as well as other class mates. I attended all TA hours and class help sessions in the last 2 weeks of class and took advantage of the time to work through errors in my code as well as to gain a high level understanding of the concepts. Additionally, I discussed lines of code with several class mates. Casey Green was very generous in helping me implement the ribbon into my project.