

IS 520 FINAL PROJECT

A RECRUITING TOOL FOR THE ROMNEY INSTITUTE

By Alejandro Raygoza

EXECUTIVE SUMMARY

The Romney Institute of Public Management was named in 1998 after the three-term Michigan governor George W. Romney who, through his life of service, exemplifies the virtues and attributes the institute has embraced. The Romney Institute is under the Marriott School of Management at BYU and prides itself in having “a long tradition of preparing (through the MPA program) young people for careers in public service and is dedicated to educating men and women of faith, character, and professional ability who will become outstanding managers and leaders in public and nonprofit institutions worldwide.”

As the MPA family grows and more students show interest in the program, it becomes more difficult to follow up with individuals who have shown interest in the program. Therefore, there is a need for organizing the data regarding potential candidates in a way that allows the institute to keep track of those who have been contacted by staff, those who applied to the program, and those who are accepted. Jeff Thompson, Chair of the Romney Institute, and Heather Chewing, Program Manager, have asked me to create a user form that will make storing information and contacting individuals easier.

The user form contains two pages, one that allows the user to see, modify, and save current contact information as well as make new entries, and another one through which the user can compose and send emails to either one email address or to several email addresses at once. After sending the emails and saving the changes in the workbook, the program transfer the information to either the updated, applied, or accepted sheet, depending on the changes that were made.

Recruiting User Form- Romney Institute of Public Management

Candidates Info | Send Emails

First Name: Jane

Last Name: Toluono

Email: toluonoj@hawaiireserves.com

Phone Number: 385-208-8423

School: BYU Hawaii

Info Session: 11/2/2015

When: mm/dd/yyyy

How: Email Phone

Frame1: Been Contacted?

Yes. Potential applicant

Yes. Need follow up

Yes. No interest

No

How He/She found out about the program?

Referrals

Career fair

Presentation

Website

First Row Previous Next New Contact

Save Save & Continue Cancel

Candidates Info | Send Emails

Username:

Password:

To:

Subject:

Rosalie Vaka : rosalievaka@gmail.com

Jane Toluono : toluonoj@hawaiireserves.com

Ulziika Sukbaatar : s.ulziika@gmail.com

G. Keoki Wallace : kwallace@wallaceassociatesgro

Jennifer Chen : jennifer.chen@byuh.edu

Johny Aisake : aisakej@go.byuh.edu

Aporosa Baba : Tukstoroca@gmail.com

Edna Wilfred : no email

Kyley Thorpe : Thorpe14@go.byuh.edu

I hope you are smiling :)

IMPLEMENTATION OF PROJECT

In order to make it easier to understand, I will explain how the user form works each page at a time.

Candidates Info Page

To initialize the form, the user simply has to click the open form button. Once the button is clicked, the form will be initialized, will set its focus on the first page (candidates info page), and will display the information that is on the current cell's row. This first page contains the following command buttons and objects:

Name	Last Name	Email
Rosalie	Vaka	rosalievaka@gmail.com
Jane	Toluono	toluonoj@hawaiireserves.com
Ulziika	Sukbaatar	s.ulziika@gmail.com
G. Keoki	Wallace	kwallace@wallaceassociatesgroup.com
Jennifer	Chen	jennifer.chen@byuh.edu
Johny	Aisake	aisakej@go.byuh.edu
Aporosa	Baba	Tukstoroca@gmail.com
Edna	Wilfred	no email

Save

- Sets the values of each relevant cell on the sheet equal to the values of its respective text box.
- If any of the options in the Been Contacted frame is selected, the information for that individual is copied to the row after the last one that contains information in the "Updated" sheet, and then the original information in the "New Contacts" sheet is deleted.
 - As part of this process, using the For Each Loop, VBA looks for a sheet with a specific custom characteristic named "sheet name" and with a value of "Updated."
 - If it finds it then it transfers the information to the sheet ("Updated" sheet), and in the case that name of the sheet (not the custom property value) is not "updated," it renames the sheet back to "updated."
 - If it does not find it, then it creates a sheet with such characteristics (both name and custom property) and transfers the information to it.
- Finally, the form is unloaded

Save & Continue

- This button also sets the values of each relevant cell on the sheet equal to the values of its respective text box, but it does not unload nor transfer the information to the “updated” sheet.
- Moves onto the next row of the current cell and displays its information on the form.

Cancel

- Unloads the form without making any changes to the sheet.

Next

- Selects the next row of the current cell on the “New Contacts” sheet
- Displays the current cell row’s information on the form

Previous

- Selects the previous row of the current cell on the “New Contacts” sheet
- Displays the current cell row’s information on the form

New Contact

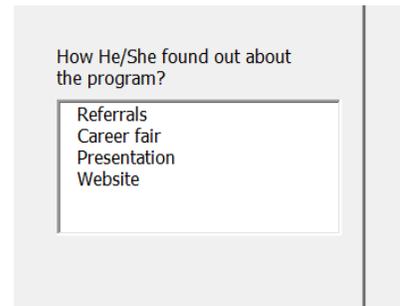
- Selects the row after the last row that contains information on column A on the “New Contacts” sheet, which clears the entire form, allowing the user to make a new entry.

First Row

- Selects the first row that contains information on the “New Contacts” sheet.

Methods List

- This list contains the possible ways that the potential candidate found about the program.
- Since I expect that staff may want to add more items to the list, I created a sheet called “Methods & Referrals” where staff can add or delete from the list.
- Upon opening the user form, the sub procedure “showData” will add each item to the list using a Do Until Loop. VBA will stop adding items once it finds a blank cell on the first column.



	FILE	HOME	INSERT	PAGE LAYOUT	FORMULAS
	D11		X	✓	fx
	A	B	C	D	
1	How did they find out?				
2	Referrals				
3	Career fair				
4	Presentation				
5	Website				
6					
7					
8					

Send Emails Page

This page of the form allows the user to compose emails, and send them directly from excel. The user only needs to provide the email ID, password, the recipient's email address, the subject, and the body of the email in the appropriate text boxes. To accomplish this, I have created the following controls from the tool box (most of them are command buttons):

Email Single

- This button calls the sendGmail function and passes it the necessary parameter values (Username, Password, etc.) These values are taken from the appropriate text boxes. The email single button was created for situations in which the user may want to send an email to an address that is not part of list in the list box.

```
(General) | (Declarations)
Option Explicit
Public Sent As Boolean
Function SendGmail(sendTo As String, from As String, pw As String, subject As String, body As String, Optional attachment)

Dim iMsg As Object
Dim iConf As Object
Dim schema As String

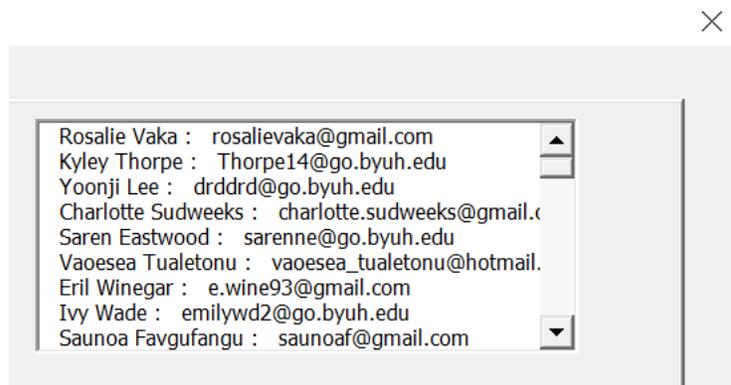
Set iMsg = CreateObject("CDO.Message")
Set iConf = CreateObject("CDO.Configuration")

' send one copy with Google SMTP server (with authentication)
schema = "http://schemas.microsoft.com/cdo/configuration/"
iConf.Fields.Item("http://schemas.microsoft.com/cdo/configuration/sendusing") = 2
iConf.Fields.Item(schema & "smtpserver") = "smtp.gmail.com"
iConf.Fields.Item(schema & "smtpserverport") = 465
iConf.Fields.Item(schema & "smtpauthenticate") = 1
iConf.Fields.Item(schema & "sendusername") = from & "@gmail.com"
iConf.Fields.Item(schema & "sendpassword") = pw
iConf.Fields.Item(schema & "smtpusessl") = 1
iConf.Fields.Update
```

- After the sendGmail function has completed its tasks, an appropriate message box will appear informing the user whether the operation was successful or not.

List of Emails (List box)

- Each row on this list contains values from three different cells that have been concatenated that shows the first name, last name, and the email address.
- The list was populated with a Do Until Loop that stops adding items once the loop encounters a row on which both the cell that should contain the first name and the cell that should contain the email are blank.



Email List (Button)

- The code for this button also calls the sendGmail function, but lets the user send the email to several addresses.
- To do this, it calls a sub procedure named generateEmailList, which does the following:
 - Keeps track of the amount of items from the list that have been selected (emailCount variable).
 - Creates an array where each email address will be stored (EmailList array).
 - Keeps track of the row number that contains the email address that was stored in the EmailList array (trackRowNo array).
- After the sendGmail function has completed its tasks, an appropriate message box will appear informing the user whether the operation was successful or not.
- If the email was sent, excel will transfer the information for each individual from the “New Contacts” sheet to the “Updated” sheet.
- If the email fails, a message box will appear saying that the email failed, and it will show which email address failed.
- Lastly, the code in the emailList button will delete the row that contained the original information (using the values stored in the trackRowNo array).

```
(General) generateEmailList

Private Sub generateEmailList()
    Dim x As Integer
    Dim i As Integer

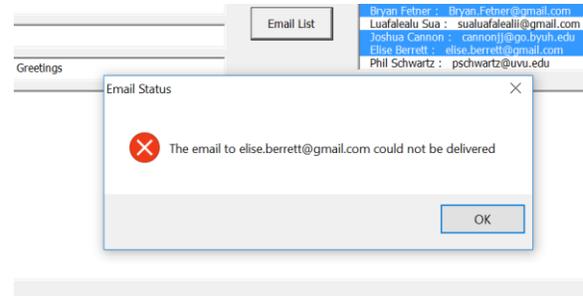
    NewContacts.Select
    emailCount = 0

    For x = 0 To lstEmails.ListCount - 1

        If lstEmails.Selected(x) Then
            emailCount = emailCount + 1

            ReDim Preserve trackRowNo(emailCount - 1)
            trackRowNo(emailCount - 1) = x + 4
            ReDim Preserve emailList(emailCount - 1)
            emailList(emailCount - 1) = CStr(NewContacts.Cells(x + 4, 3).Value)

        End If
    Next
End Sub
```



Other Buttons on the Sheet

Transfer to Applied & Transfer to Accepted

- Both of these buttons do the same thing but on different sheets. The Transfer to Applied button activates the “applied” sheet and looks at the value on column K for each row with data. This cell shows whether or not the individual applied to the MPA program. If this cell has a value of “Y” or “y,” then the information for that individual is transferred to the “Applied” sheet. The Transfer to Accepted button does the same looks on column L for a value of “Y” or “y.” If found, then the information is transferred to the “Accepted” sheet.

Since both buttons identical tasks I've shared screenshots for only one of them.

Name	Last Name	Email	Phone Number	School	Contacted	How did they hear about the program?	Been Contacted?	When	How	Applied (Y/N)	Accepted (Y/N)
Rosalie	Vaka	rosalievaka@gmail.com		BYU Hawa	11/2/2015		1 No	mm/dd/yyyy		y	Y
Yoonji	Lee	drddrd@go.byuh.edu		BYU Hawa	42310		No	mm/dd/yyyy		y	No
Charlott	Sudweeks	charlotte.sudweeks@gmai		BYU Hawa	42310		No	mm/dd/yyyy		y	No
Saren	Eastwood	sarenne@go.byuh.edu		BYU Hawa	42310		No	mm/dd/yyyy		y	Y
Vaoesei	Tualetonu	vaoesea_tualetonu@hotm		BYU Hawa	42310		Yes. Need to follo	42090 Email		y	Y

```
(General) transferInfoAccepted
End Sub
Sub transferInfoAccepted()
Dim x As Integer
Dim destRow As Integer

Sheets("applied").Activate

For x = Cells(Rows.Count, 1).End(xlUp).Row To 4 Step -1

If Cells(x, 12).Value = "Y" Or Cells(x, 12).Value = "y" Then
    destRow = Sheets("accepted").Cells(Rows.Count, 1).End(xlUp).Row + 1
    Range(Sheets("accepted").Cells(destRow, 1), Sheets("accepted").Cells(destRow, 12)).Value = _
    Range(Sheets("applied").Cells(x, 1), Sheets("applied").Cells(x, 12)).Value

    Sheets("applied").Rows(x).Delete

End If
Next

End Sub
```

Ribbon

- The ribbon allows the user to open the form regardless what the active sheet is.

LEARNING OUTCOMES AND DIFFICULTIES

What I learned

While this project did not show me anything new that we did not cover in class, it did help me solidify many things. It could be said that I did not actually learn many of those things until I created the code for myself. Through this project I learned to make better user forms, automate the process of sending emails to multiple addresses, use different types of loops, work with ranges and cells. I also solidified my understanding of how arrays work as well as when to use Redim and Redim Preserve. Moreover, I learned about the importance of breaking a program into smaller sub procedures, which allows for easier reading and debugging. But perhaps the two most important things that I learned through this project were the ability to effectively use break mode, the immediate window, and the locals window, to find errors in the code, and to know how to find answers to those problems on the internet.

Doing this project helped me realize that though I am not an experienced coder, I can create powerful programs on VBA with what I learned in class.

What Will Be Added Later

While working on this project I realized that it is so easy for me to feel tempted to add "one more thing" to the project. Though, I finished everything Dr. Thompson and Heather Chewing asked me to do for the project (and more), there are things I know would be helpful that I couldn't include because of lack of time. Some of these things include the creation of:

- Custom properties for each of the key sheets in the workbook in order to prevent the user from changing the names of the sheets.
- A sub procedure that would summarize information about all candidates through graphs and tables.
- Code that will allow the user to choose a file through a file dialogue box.

Struggles

Here are some of the major struggles I had while writing the code for this project:

- While trying to transfer information from one sheet to another and deleting the row that had original information, I found that deleting the row would shift all row below up, which would cause problems with the “For Loop.” I tried many different ways until I created a relatively complicated code that would do the work. Later, I learned by simply having the loop start at the top and step by -1 each time, I would’ve saved so much time and effort.
- I also learned to avoid using the copy/cut and paste methods. They caused a few errors that were fixed by just setting a range of values equal to another set of values. By doing so, the code runs faster and smoother.
- Another source of frustration was to try to change the code name of a sheet through VBA. I learned later through Dr. Allen that you can’t do that through code, but he suggested a way so accomplish something similar by creating custom properties for each sheet.
- Though everything in the project works well, the sendGmail function was meant to send emails using a BYU account not a Gmail account. I found online the information that needed to change in the code, but despite our several attempts (Dr. Allen and I), we couldn’t get it to work. This will be something I will try to change next semester.

Assistance

I did not receive any assistance from any individual besides Dr. Allen and only for a few very specific questions. Though I used the internet a lot to solve problems, I wrote the code myself. The only exception would be the module called sendGmail which I patterned after the example we did in class.