Jason Johnson

ISYS 562 – Final Project

Executive Summary

My fiancée, Nicole, and I have been preparing for our wedding, and Nicole was wishing there was a different way to get friends' addresses than facebook. She didn't like the lack of privacy for our guests in using a facebook group, and she felt that a googledoc form was too impersonal. She was just thinking of personally inviting all 150 of our friends manually. At this time, Gove spoke about how we should be making our final projects, and how our final projects should be something that we should be passionate about. I am passionate about preparing for my wedding, so I thought that this would be a great opportunity to help both Nicole and myself out by writing an excel macro that would allow us to choose which friends we wanted to email from facebook write a generic email, and then personalize it for each individual guest. Initially, I was only going to allow the macro to send emails via facebook; however, I later decided to allow a user to send messages via facebook and email.

In the following document I will outline the implementation of my VBA project as well as address the difficulties that I found along the way. The main portions of my project are enumerated below:

1. Internet Explorer Automation and Logging into facebook
2. Scraping Friends
3. Cleaning Friend Data and Presenting it in a Form
4. Form Rules and Behaviors
5. Sending the Message Via facebook
6. Sending the Message Via Email

It is important to note that, though this paper will give a comprehensive view of my project, it is a very top-level view as well. That is, there are many fine details of my project that will not be addressed in this paper and, if one is interested in understanding the underlying infrastructure of my project, must be investigated through viewing my code and running my project.

Implementation

Internet Explorer Automation and Logging into facebook

My first step was to automate Internet Explorer. I had had previous experience with webpage automation in using the PowerShell scripting language; however, I had never created anything as extensive as was necessary for this project. I began to experiment with the CreateObject() function and entered "InternetExplorer.Application" between parenthesis. This allowed me to set the IE object to be an Internet Explorer object.

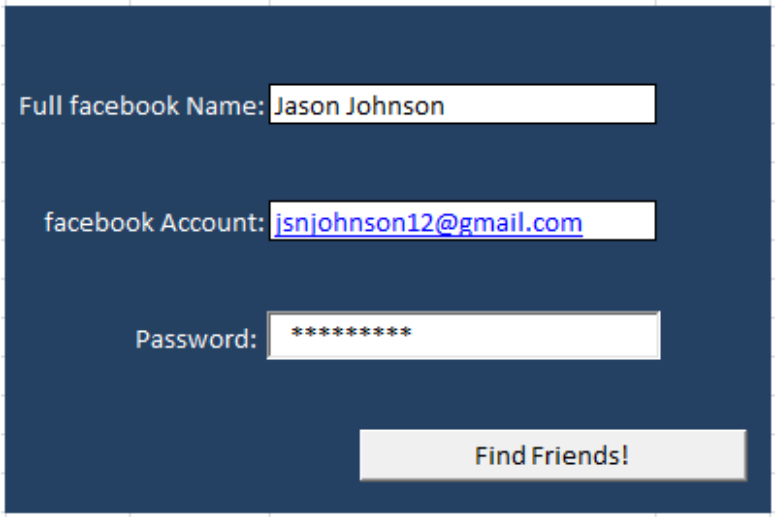

Figure 1 - The Login Form

Next was navigation. By setting account and password information in the "FindFriends" sheet and passing these ranges into variables, I was able to successfully navigate to facebook, enter my password and information, and login. From my login page, I was able to navigate to my friends page and began to work on extracting all friend information necessary to send my guests a message.

Problems with Internet Explorer Automation and Logging into Facebook

One of my first problems was found in logging into facebook. Although I could login with no problem, whenever the program stopped, it threw an exception the next time I tried to login it couldn't because I was already logged in. Even when I programmed the IE object to uncheck the box labeled "Keep me signed in," Internet Explorer would still not log out correctly.

I found that the solution for this was to use On Error to catch any error thrown, check to see if it was error 424—the error thrown when a user tries to login when they are already logged in—and skip login if such an error were thrown.

A second problem that I ran into was that facebook does not put their links into divs that have IDs. Facebook puts their links into divs that have classes. Excel allows you to search an Internet Explorer object by ID; however, it does not allow you to search by class. After searching on the internet for a while, I found that I could cycle through all links on the webpage by using a simple for loop like the one below:

```
For Each Link In IE.document.Links

 'Debug.Print Link
```

```
    If InStr(Link, "profile") = 0 Then

    'etc., etc.,

  Next
```

This allowed me to navigate to my profile and later access my friends page. There were quite a few things I needed to do in order to ensure that the correct link was selected, but these steps were menial and not worth going into detail over.

Scraping Friends

Facebook does not want to allow you to access friend information. It was once easy to scrape a friend's information off facebook. All of that changed with the birth of Google +. Because facebook did not want you to be able to easily add your facebook friends onto Google+, facebook hid all important information inside a profile's about page. Additionally, facebook now only shows a certain number of friends at a time. In order to view all friends on the same page, we must scroll through the entire friends list.

Because of this change in facebook, It was necessary that I automate Internet Explorer to scroll through all friends, copy each friend's profile link, and then paste it into an excel spreadsheet.

Problems with Scraping Friends

This was a very difficult step of my project. Internet Explorer does not have very good automation when it comes to scrolling down a webpage. I tried to use a function found in the Internet Explorer object called ExecWB in order to scroll through the whole page, copy the information therein, and paste it into a spreadsheet. Although the ExecWB function was effective in selecting all data in the webpage and copying it, its page down method did not work sufficiently well to use it reliably.

Because I could not successfully use ExecWB to scroll through my friends, I had to resort to SendKeys. I am not a fan of SendKeys because this means that the Internet Explorer object needs to have the focus through the whole time that the SendKeys are being used, and there is no real way to receive good feedback about when it is done either.

Even though I was not happy with using SendKeys, they were effective in this situation, and my project is able to scroll through all my friends, select them, copy their profile links, and copy them into an Excel Spreadsheet.

Cleaning Friend Data and Presenting it in a Form

After all my friends are loaded into the Excel spreadsheet, I began to clean up the data and present it into a form called "Friends."

Although there were no real problems with formatting the list of profiles, there were a few conditions that proved the task interesting. One of these conditions was that, because I selected everything in the

browser window and copied it, there was a lot of useless information that I needed to sift through. For one thing, facebook puts every person's name twice. This meant that I had to write code to remove duplicate information.

Additionally, not all the links on the page were profile links. Many links were ads or links to businesses and universities. I had to find a way to sift through every link on the page and either keep it or discard it from my friend list.

Finally, even if a person no longer has facebook, facebook includes those deleted accounts in your friends list. This made it necessary to delete these blank records from the friends list.

After these formatting conditions were followed, I was able to put the friends into an array, and then put the array into a form list.

Form Rules and Behaviors

I have had a fair amount of experience with VBA before this class; therefore, many of the things that I have done up to this point were not new concepts to me. However, this section of my project was greatly benefitted by Gove's classes on forms and form logic. I was able to create a number of forms that showed your facebook friends, allowed you to select multiple friends at once, and add the selected friends to an email list. I also needed to code logic to delete more than one friend from the list at a time.

I created a form in which the user could write their message, and I created a form to enter a generic message and choose whether to send the email via facebook or Gmail.

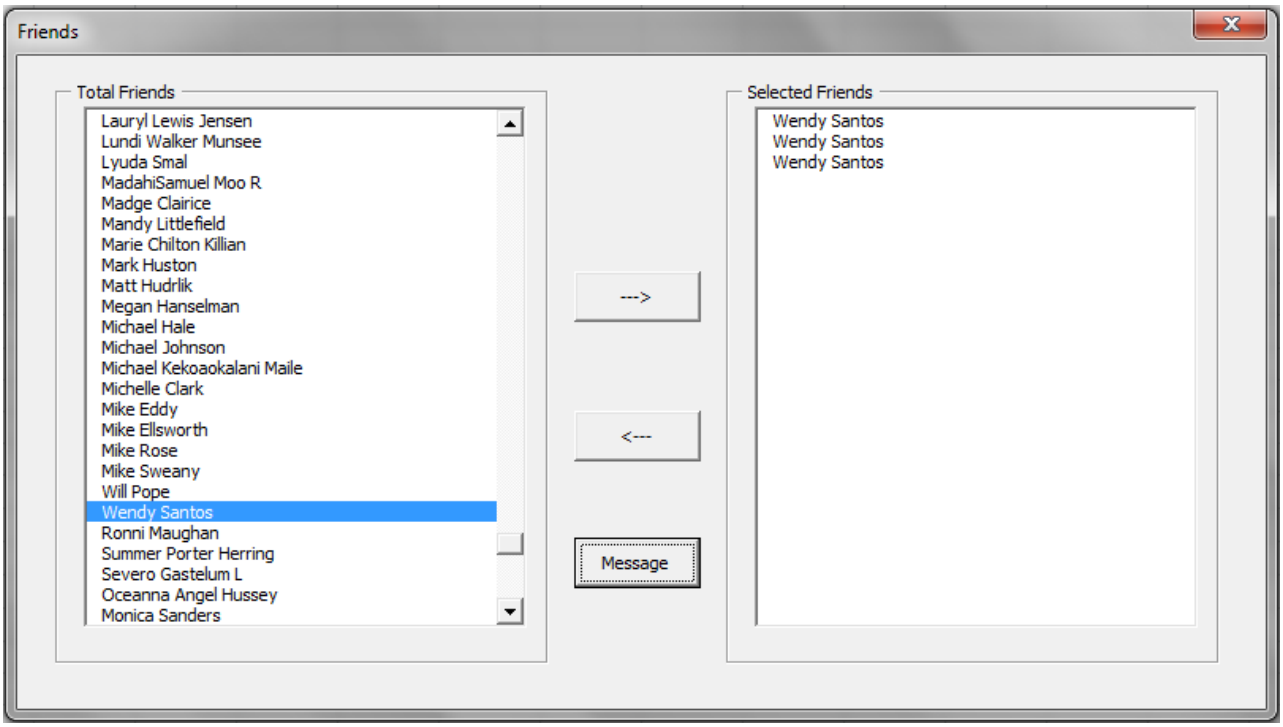

Figure 2 - View for List of Facebook Friends

Problems with Form Rules and Behaviors

I did not run into too many problems with form rules and behaviors because Gove covered it so extensively in class. I would say that deleting multiple friends from the list at one time gave me the most difficulty. I could not delete more than one friend from the list without messing up the list count and content. I eventually had to create a separate array with the list of names I wanted to delete, iterate through the array, and remove each name individually from the list.

Sending the Message via facebook

After a message has been written, my program will iterate through all the names of the chosen friends, insert their first name at the top of the email, and send it via facebook. This approach once again implements the Internet Explorer object to navigate to each selected friends' profile, click the message button, paste the message into the message body, and send the message.

It was necessary to use SendKeys once again to interact with the facebook message page. I had to paste the message through SendKeys and, because facebook sends messages through a .php application and not a regular html page, I had to tab over to the "Send" button.

Another fun feature of my program is its ability to discern between sending a message to a friend with whom you have already had conversations, and sending a message to a person that you have never contacted through messages. This is particularly difficult because you have to pass in a different number of tabs for a new contact than for a regular or previous contact.

Problems that I encountered while sending the email via facebook are very similar to those that I encountered while sending the message by email. As such, I will address the problem when I write about sending the message via email.

Figure 3 - The Message Form

Sending the Message via Email

Yet again, Gove's example of email automation assisted me greatly in sending emails. Before

Gove's lecture, I had decided on scraping each selected friend's email from their profile page, select the email address, and send it by opening outlook. Gove's Gmail solution was much more efficient and much more professional looking. I decided to implement this solution, altering it to fit my project, and am happy with its results.

Problems with Sending the Message

One very large problem with sending the message was that facebook would shut me out if it thought that I was automating the web browser. After sending five messages or scraping five email addresses, facebook slightly alters its html and php structure.

This was made visible when I tried to send a sixth message. Everything copied into the message box fine; however, facebook altered the number of times I needed to send the tab key to the browser in order to select the "Send" button and send the message. This made it so that no message would send after message six.

In similar fashion, after scraping 5 different friends' email addresses, facebook hides your friends' email addresses from you. Hiding the email address made it quite impossible for me to grab any more addresses if I did not change my code.

I was able to fix both problems by adding random wait times between one and five seconds to the beginning of each iteration through a profile. Additionally, I made a counter that would signal when I got to four addresses or four messages sent. Upon reaching four addresses or messages, my program navigates to a separate page. This makes facebook think that I am a human because I am not just scraping emails or sending messages.

Assistance

There was no person who assisted me in my project except for the lectures given by Gove.

Conclusion

It works! I was able to create a fiancée-approved VBA program that allowed Nicole and me to send out personalized emails and facebook messages to our friends that requested their address information. This project allowed me to synthesize many of the principles and techniques that I have learned in this class, and this project also allowed me to see the wide application that VBA can have in a business setting. It would not be difficult to apply the same structure used here to acquire client information and send them a personalized email for the business that I will work for in the future.