

MBA 614 | Spreadsheet Modeling and Automation

## Final Project | Contact List Program

Eduardo Michelutti



## **Executive Summary**

Everybody has had the uncomfortable experience of bumping into someone they have met previously and not being able to remember their name. This is a pleasant experience compared to what it is like to be desperately in need to talk to someone we've met but no being able to find the person's contact info. I think that a part of the problem here is that technology is developing so fast that we constantly have to change devices and we tend to lose some information on the way. This final project attempts to offer a simple and user friendly excel-based tool that allows one to quickly input and store the contact information of acquaintances and to perform searches on the database.

The system consists of a user form that allows one to insert or update a contact's info and to search for a contact by filling out entire or partial data in one or more of the available fields. It is also possible to insert and visualize an image file associated with the contact. To facilitate access to the program, I have added an icon in the "Data" tab of the Ribbon by the name "Run Contact List" (easily distinguished by a yellow smiley face icon). When clicked, this icon runs a macro called "ShowData", which is set to start the user form "Contacts". This user form contains all the fields and buttons with which the user interacts in order to create, search or update the contact list.

## **Implementation | Documentation**

I started by building the user form (Contacts). I wanted it to be very clean-cut and easy to navigate. As I mentioned earlier, the idea is to offer a quick way to store contacts. For this reason, the user form has as few buttons and fields as possible. Let's take a look at a screenshot of the user form:

**Figure 1 – User Form**

The screenshot shows a window titled "Contact List" with a close button in the top right corner. The form is organized into several sections:

- Name:** Two text boxes for "First Name" and "Last Name".
- Phone:** A "Phone Type" dropdown menu, a "Phone Number" text box, and a "Save Phone #" button.
- E-mail:** An "E-mail Type" dropdown menu, an "E-mail Address" text box, and a "Save E-mail" button.
- Address:** A large text box for the address.
- Company:** Two text boxes for "Company" and "Birthday".
- Info:** A large text box for additional information.
- Navigation/Action:** A row of buttons at the bottom: "Insert New Contact", "Find", "Find Next", "Update Contact", "Clear Form", and "Close Form".
- Image:** An "Insert Picture" button located to the right of the "Info" field.

The user can navigate through the different fields of the user form either by using the mouse or by hitting the tab button. I've organized the Tab Index of each Text Box or Command Button as to offer a logical sequence for the latter option. In order to provide a thorough, yet concise explanation of the functionalities of the user form, the following is a breakdown of each Text Box (txt), Combo Box (cbo), Image (img) and Command Button (cmd) found in the user form:

- **First Name (txt):** Field to insert the contact's first name.
- **Last Name (txt):** Field to insert the contact's last name.
- **Phone Type (cbo):** This Combo Box gets its data from a hidden sheet (Phone). There are three possible phone types: Home, Cell and Business.
- **Phone Number (txt):** Field to insert the contact's phone number.
- **Save Phone (cmd):** Based on the Phone Type chosen, it stores the phone number in the correct cell of the "Data" sheet. It can be used to save a phone number for a new contact or to update an existing contact's phone number.

- **E-mail Type (cbo):** This Combo Box gets its data from a hidden sheet (E-mail). There are two possible phone types: Personal and Business.
- **E-mail Address (txt):** Field to insert the contact's e-mail address.
- **Save E-mail (cmd):** Based on the E-mail Type chosen, it stores the e-mail address in the correct cell of the "Data" sheet. It can be used to save an e-mail address for a new contact or to update an existing contact's e-mail address.
- **Address (txt):** Field to insert the contact's address.
- **Company (txt):** Field to insert the contact's company name.
- **Birthday (txt):** Field to insert the contact's birthday date.
- **Info (txt):** Field to insert general information about the contact.
- **Image (img):** Contact's picture is displayed here.
- **Upload Image (cmd):** This command opens an interactive Windows Explorer window that allows the user to locate and choose the image file he or she wants the contact to be associated to. It then stores the file extension in the correct cell of the "Data" sheet and displays the picture in imgImage. This button can only be clicked after a Find or Insert Contact procedure has been successfully ran.
- **Insert Contact (cmd):** It stores each piece of information available for the user to input in the correct cells of the "Data" sheet.
- **Find (cmd):** Visits the "Data" sheet to search for all the data inserted in any of the text fields.
- **Find Next (cmd):** Searches for the next match for the same criteria defined in Find.
- **Update Contact (cmd):** It updates each piece of information available for the user to input in the correct cells of the "Data" sheet. This button can only be clicked after a Find or Insert Contact procedure has been successfully ran.
- **Clear Contents (cmd):** Clears the contents of all text and image fields.
- **Close Form (cmd):** Closes the form.

All of the above described Command Buttons prompt message boxes after being clicked to confirm that the action was performed successfully or to direct the user's attention to the fact that a parameter hasn't been met and therefore, the expected action wasn't completed.

For the purpose of giving the professor enough data to evaluate the functionality of the program, I have inserted through the user form the contact info of 25 BYU MBA students (Class of 2012) along with their pictures.

### **Discussion of Learning and Conceptual Difficulties Encountered**

I can totally say that I learned a great deal building this program. Even though it was easy to understand the logic and to follow the explanations provided by the professor in the examples we did in class, when I found myself alone trying to write the code for the program, I had to go back to the VBA for Modelers book to check on very simple things I had already forgotten. Most likely, my code is the longest and hardest way possible to build this program, as I'm sure that I didn't use all the possible or even appropriate objects, properties, methods and events. I have to say that it was very difficult to build this program and that it took me way much more time than I thought it would.

There were many difficulties I found along the way, but only one that I couldn't overcome. One of the initial ideas (back in the project proposal) was to allow the user to download the data and picture of the contact straight from LinkedIn. Nevertheless, I encountered two problems related to that idea:

1. LinkedIn profiles don't have the type of data I initially thought they had. Simple contact info such as Phone Number, E-mail Address and Birthday aren't available in most of the profiles I have in my contacts.
2. When I found out that most of the data was found in the profiles were useless to the user form I had in mind, I decided that I could at least download the picture found in the profile and insert it as the contact's picture in the form. However, even though I tried to record this many times and combed the book and the web, I wasn't able to find a way to download the picture as file, store it in the computer and then use its reference address to insert the picture in the user form.

To make a long story short, and because of the due date of the final project, I decided to exclude the option of downloading data from LinkedIn. It is worth noting that in the current version of the program, the user can indeed insert a picture to the contact's info, inasmuch as this picture is already in the user's computer.

As an end note, the Find function used in this program was by far the hardest thing I've had to do in VBA so far. It took me many hours to put it together and to make it work in all possible scenarios. I can say that this project gave me the confidence to take the risk of working with VBA in a business setting.