Chris DuBois
MBA 614
April 10, 2011

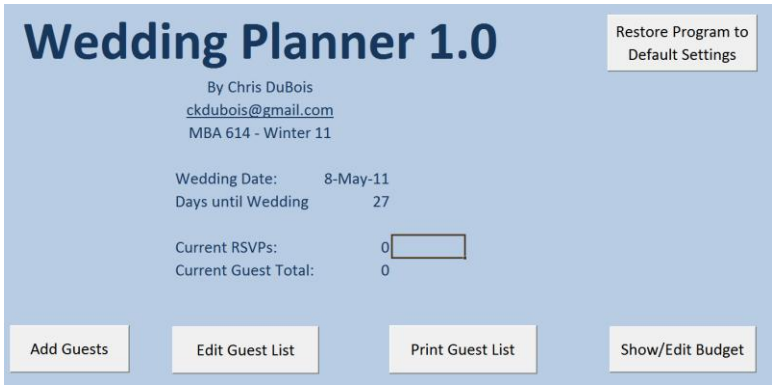
Executive Summary

Planning a wedding can be an overwhelming experience. Keeping track of guests, budgets, gifts, who's coming and who isn't, all of these things can quickly become confusing to the point of losing control. As I am currently in the middle of this process, I decided that there had to be a better way.

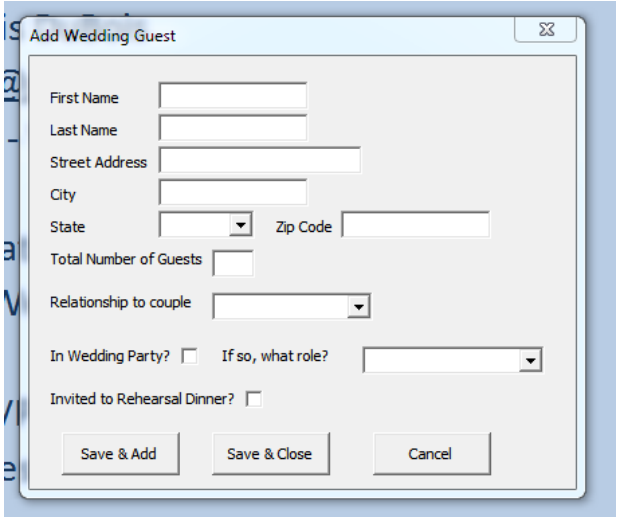
My goal for this project was to design that better way. I created an Excel workbook with a user form-driven interface to organize all aspects of a wedding, from the guest list and RSVP count to gift tracking and budgeting. This allows quick, easy access to most of the major stress points in planning a wedding and provides the couple doing the planning an easy way to keep the torrent of information organized.

Implementation Documentation

Note: Anytime a password is needed (to unlock sheets for example), the password is "goveallen"



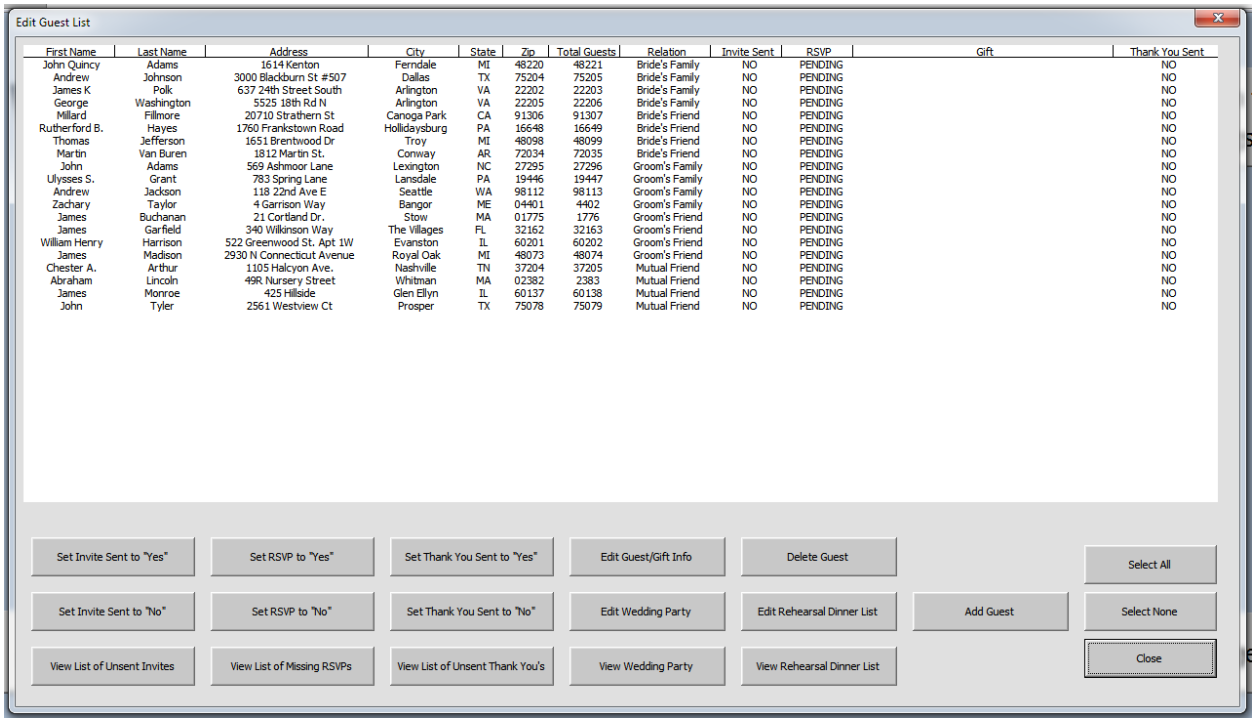
I decided that the application would be most oriented around the wedding list, and that the first thing that was needed was actually to populate the wedding list in some way. Therefore, the first module I programmed was the "Add Guest" mechanism, frmAddGuest.



This form writes to the "list" sheet, which I used as the main database for the wedding list. However, in order to populate the combo boxes for state, relationship to couple, and wedding party role, I created a reference sheet called "States" (because that was all it was for at first) with the info needed to populate each box. I then just created labels and text boxes for the important fields needed for the list. Not all of the fields are required (especially address fields), but did add in a few pieces of logic to ensure that required info was entered. This required info is the Total Number of Guests, Relationship to the Couple, and (if "In Wedding Party" is selected), Wedding Party Role. Leaving any of these boxes blank generates an error message and returns the user to the main screen. I also set "Invited to Rehearsal Dinner" to true automatically if "In Wedding Party" is selected, since tradition dictates that anyone in the wedding party is usually invited to the rehearsal dinner. Finally, I added "Save and Add" and "Save and Close" buttons so the user has the option of entering multiple guests at once or just adding one at a time.

When guests are added, they are listed on the "List" sheet and automatically assigned a few default values. Since the idea is that this list can be used when generating the list of people invited (not just those attending), a column for "Invite Sent" and "Thank You Sent" is automatically set to "NO", while "RSVP Received" is set to "PENDING" so as not to assume a yes or no.

Once guests are added to the guest list, it can be manipulated using the "Edit Guest List" button.



First Name	Last Name	Address	City	State	Zip	Total Guests	Relation	Invite Sent	RSVP	Gift	Thank You Sent
John Quincy	Adams	1614 Kenton	Ferndale	MI	48220	48221	Bride's Family	NO	PENDING		NO
Andrew	Johnson	3000 Blackburn St #507	Dallas	TX	75204	75205	Bride's Family	NO	PENDING		NO
James K	Polk	637 24th Street South	Arlington	VA	22202	22203	Bride's Family	NO	PENDING		NO
George	Washington	5525 18th Rd N	Arlington	VA	22205	22206	Bride's Family	NO	PENDING		NO
Millard	Fillmore	20710 Strathern St	Canoga Park	CA	91306	91307	Bride's Friend	NO	PENDING		NO
Rutherford B.	Hayes	1760 Frankstown Road	Hollidaysburg	PA	16648	16649	Bride's Friend	NO	PENDING		NO
Thomas	Jefferson	1651 Brentwood Dr	Troy	MI	48098	48099	Bride's Friend	NO	PENDING		NO
Martin	Van Buren	1812 Martin St.	Conway	AR	72034	72035	Bride's Friend	NO	PENDING		NO
John	Adams	569 Ashmoor Lane	Lexington	NC	27295	27296	Groom's Family	NO	PENDING		NO
Ulysses S.	Grant	783 Spring Lane	Lansdale	PA	19446	19447	Groom's Family	NO	PENDING		NO
Andrew	Jackson	118 22nd Ave E	Seattle	WA	98112	98113	Groom's Family	NO	PENDING		NO
Zachary	Taylor	4 Garrison Way	Bangor	ME	04401	4402	Groom's Family	NO	PENDING		NO
James	Buchanan	21 Cortland Dr.	Stow	MA	01775	1776	Groom's Friend	NO	PENDING		NO
James	Garfield	340 Wilkinson Way	The Villages	FL	32162	32163	Groom's Friend	NO	PENDING		NO
William Henry	Harrison	522 Greenwood St. Apt 1W	Evanston	IL	60201	60202	Groom's Friend	NO	PENDING		NO
James	Madison	2930 N Connecticut Avenue	Royal Oak	MI	48073	48074	Groom's Friend	NO	PENDING		NO
Chester A.	Arthur	1105 Halcyon Ave.	Nashville	TN	37204	37205	Mutual Friend	NO	PENDING		NO
Abraham	Lincoln	49R Nursery Street	Whitman	MA	02382	2383	Mutual Friend	NO	PENDING		NO
James	Monroe	425 Hillside	Glen Ellyn	IL	60137	60138	Mutual Friend	NO	PENDING		NO
John	Tyler	2561 Westview Ct	Prosper	TX	75078	75079	Mutual Friend	NO	PENDING		NO

On this screen, the guest list can be mass-manipulated in a variety of ways. Multiple rows of guests can be selected at once or the "Select All" button can be selected. From here, "Invite Sent", "RSVP", and "Thank You Sent" status can be changed en masse. Additionally, buttons under each section can be clicked to show a printable list of unsent invites, missing rsvps, and unsent thank yous. Multiple guests can be edited from here as well. For loops based on list selection status (ListIndex) are used to track which items are actually selected. I did run into a problem with this, in that whenever a list item was changed, it would reset the selection status of all other items, breaking the loop. To get around this I was forced to copy the information to a temporary sheet including blank rows, then once the loop concluded copy the information back to the main "List" sheet and refresh the .RowSource connected to the listbox.

Additionally, the wedding party and rehearsal dinner list can be easily edited in a similar way. For these boxes, I set up two "comparative" listboxes on each form. I was unable to figure out how to keep names showing up in one list from showing up in the other (for example, to prevent names already in the wedding party from showing up in the general list), but I feel

confident I could figure it out in a later version. Tied to these forms is a pair of "view" buttons connected to the wedding party and rehearsal dinner list, respectively. They are set up to be printable and, upon closing them, reopen the "Edit Guest List" form.

Finally, I added in a button to directly open the "Add Guests" form from the "Edit Guest List" form to save users a few clicks if they found they needed to add a guest quickly. I also provide a "Delete Guest" button which allows for mass deleting of guests as well.

The previous two forms (technically five if you count the various "Edit" forms) were all based mainly on for loops with if-based error trapping. They also do the bulk of the work of setting sheets to be visible or not, since it was my goal to have only one sheet open, mostly the "Front Page" sheet open unless a list (wedding, party, rehearsal) was being printed. Once I completed the guest list module, I added a button on the main page for showing the guest list. It is similar to the buttons described previously for viewing the party and rehearsal dinner lists in that it can be easily printed. The only difference is hitting the close button returns the user to the front page instead of the edit guest list form.

My final module in this version was the budget module. This was a fairly straightforward form, requiring only refreshing the .RowSource to account for added budget items. However, I did have to use the FormulaR1C1 syntax for the first time to ensure that the Pre-Tax Total formula took into account new budget items. Beyond that, it was very straightforward to add in the Venue Name, Tax Rate, and Total information.

Once I completed both modules, I did some work on the UI of the "Front Page" sheet to make it look a bit nicer. At this point I realized that I wanted some easy way for users to reset the program, so I added a "Defaults" sheet and module. I created a simple "Warning, this could ruin everything!" form to ensure that users knew exactly what was doing, then set the module to either clear unneeded information or add in default values (in the case of the budget module).

Learning and Conceptual Difficulties

Coming into this class, I had had some minor experience with programming, mostly involving writing simple C++ programs in high school and then more recently writing HTML and logic scripts for my internship this past summer. Therefore, I understood the foundations but did not have a deep knowledge of things like iterative loops, user forms, and dealing with errors.

However, the User Forms homework completed a few weeks ago really helped me figure out the best ways to implement my ideas for this project. As I was writing the code, I would have flashes of insight from the previous homework that would help me to find more efficient ways to solve the issues I was facing. I am positive that there are still more efficient ways to do what I did, probably involving arrays, but for the most part I am proud of what I was able to accomplish. The highlight of the project, the "Edit Guest" form, is something that I am quite fond of and feel that creating an interface like that will help me conceptually in any future VBA projects I do for work or personal convenience.

That said, there were a few things I would have liked to do. I would have like to involve retrieving external information from sites like theknot.com and wedding registries. This would have provided an additional dimension of functionality to the project, but I was unable to determine a flexible way to pull the information off those sites and display it in a form in Excel that made it useful. I also would have liked to "pretty up" the GUI a bit, but what I ended up with was utilitarian and sufficient for my needs.

Overall, this experience has been a positive one. I feel as though I will be able to use both VBA and the concepts behind coding in general in the future and will be able to apply what I learned in many ways. Knowing that there is a wealth of resources beyond class and the textbook will allow me to solve the problems I face while continuing to gain the conceptual foundation I need to eventually write programs without much need for outside assistance.