

Executive Summary

The business I created the project for is Blushing Rose Floral, a small floral company that designs and creates floral arrangements, primarily for events. Blushing Rose Floral has no current system to track their customers, track their costs/revenues, create invoices, or organize orders. Organizing/pricing orders is especially difficult because different flowers come in different bunch sizes, so if an order needed one more flower than comes in a bunch, Blushing Rose Floral would need to know to purchase an extra bunch.

The system I built automates the process for the previously mentioned problems. Blushing Rose employees will use various user forms to input orders, assign the order to a customer (or create a new customer) and then update all related information (company revenue, customer billing history, customer invoice, order information).

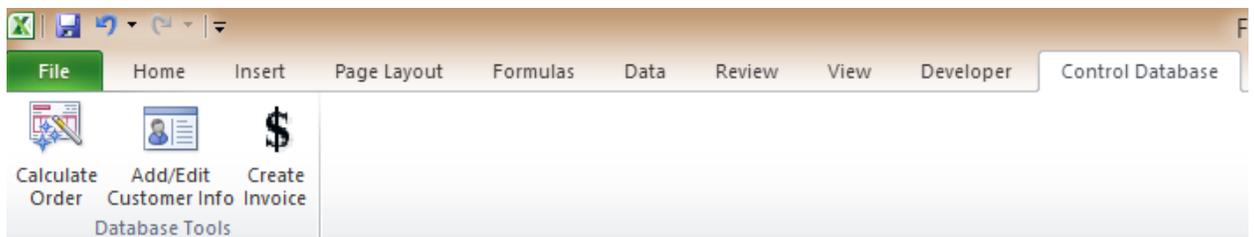
Implementation Documentation

- 1) Create a database with all of the flower types, price per bunch, and number of stems per bunch. The wholesaler used by Blushing Rose provides this information via email, so I had to manually enter the information. This acts as a foundation for following processes. It is easy to add/change information when changes occur in the market.



ID	Flower Type	Price	Stems
1	Albion Pink Peruvian	11.99	10
2	Amber	9.99	10
3	Amber Los Angeles Hybrid	9.99	10
4	Angela	9.99	10
5	Amber Hybrid "Rosa"	9.99	4
6	Amber Hybrid "Rosa"	9.99	4
7	Amber Hybrid "Rosa"	9.99	4
8	Amber Hybrid "Rosa"	9.99	4
9	Amber Hybrid "Rosa"	9.99	4
10	Amber Hybrid "Rosa"	9.99	4
11	Amber Hybrid "Rosa"	9.99	4
12	Amber Hybrid "Rosa"	9.99	4
13	Amber Hybrid "Rosa"	9.99	4
14	Amber Hybrid "Rosa"	9.99	4
15	Amber Hybrid "Rosa"	9.99	4
16	Amber Hybrid "Rosa"	9.99	4
17	Amber Hybrid "Rosa"	9.99	4
18	Amber Hybrid "Rosa"	9.99	4
19	Amber Hybrid "Rosa"	9.99	4
20	Amber Hybrid "Rosa"	9.99	4
21	Amber Hybrid "Rosa"	9.99	4
22	Amber Hybrid "Rosa"	9.99	4
23	Amber Hybrid "Rosa"	9.99	4
24	Amber Hybrid "Rosa"	9.99	4
25	Amber Hybrid "Rosa"	9.99	4
26	Amber Hybrid "Rosa"	9.99	4
27	Amber Hybrid "Rosa"	9.99	4
28	Amber Hybrid "Rosa"	9.99	4
29	Amber Hybrid "Rosa"	9.99	4
30	Amber Hybrid "Rosa"	9.99	4
31	Amber Hybrid "Rosa"	9.99	4
32	Amber Hybrid "Rosa"	9.99	4
33	Amber Hybrid "Rosa"	9.99	4
34	Amber Hybrid "Rosa"	9.99	4
35	Amber Hybrid "Rosa"	9.99	4
36	Amber Hybrid "Rosa"	9.99	4
37	Amber Hybrid "Rosa"	9.99	4
38	Amber Hybrid "Rosa"	9.99	4
39	Amber Hybrid "Rosa"	9.99	4
40	Amber Hybrid "Rosa"	9.99	4
41	Amber Hybrid "Rosa"	9.99	4
42	Amber Hybrid "Rosa"	9.99	4
43	Amber Hybrid "Rosa"	9.99	4
44	Amber Hybrid "Rosa"	9.99	4
45	Amber Hybrid "Rosa"	9.99	4
46	Amber Hybrid "Rosa"	9.99	4
47	Amber Hybrid "Rosa"	9.99	4
48	Amber Hybrid "Rosa"	9.99	4
49	Amber Hybrid "Rosa"	9.99	4
50	Amber Hybrid "Rosa"	9.99	4

- 2) Create a custom ribbon addition to allow easy use of program



3) Create a simple form that allows the user to create and calculate an order

The 'Order Calculation' dialog box includes the following fields:

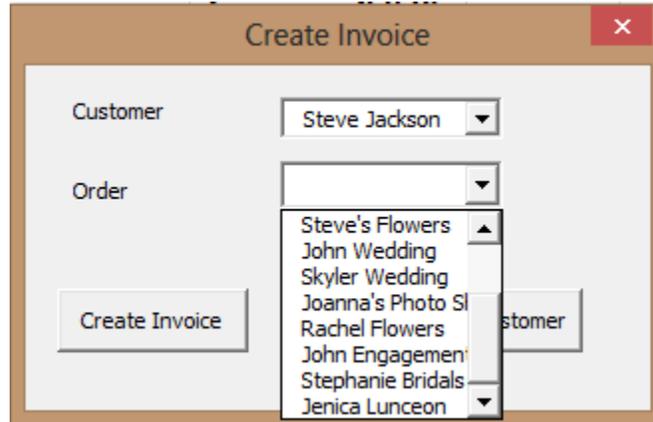
- Order Name: John Wedding
- Number of Pieces: 5
- Profit Margin: 75%
- Type of Flower: Alstromeria (3), Rhamnus (2), Tulip (4), Hypericum (1), Iris, Larkspur, Leather Leaf Re, Leather Leaf Me, Liatris, Lily Grass, Lisianthus

The types of flower combo box fills from the manually updated sheet, and the rest of the data is filled in based on the customer's needs. When "Generate Report" is selected an order form is generated that breaks down how many bunches of each flower type need to be purchased, the total cost to purchase, and the price that should be charged based on the selected profit margin. This order form is created on a new sheet; the new sheet is automatically called whatever the user input for "Order Name" (awkwardly circled in red on the bottom right of screenshot).

Flower Name	Bunches Needed	Cost	Total Cost	Customer Price	Profit
Alstromeria	4	\$35.80	\$ 136.20	\$ 306.45	\$170.25
Aspidistra	3	\$20.85			
Iris	5	\$39.75			
AsterPurple "Super"	4	\$39.80			

The spreadsheet shows a new sheet named 'Skyler Wedding' (circled in red) and a table with the following data:

- 4) Now the user is ready to create an invoice for the customer, the partially covered button says “Add/Edit Customer.” This allows the user to create a new customer account if they are not currently in the system. The order combo box shows all orders that are currently in the system. The “Add/Edit Customer” button takes us to a form that we will look at a little later, for now we will see what happens when “Create Invoice” is selected.



Of course the main thing that occurs here is an invoice is created:

Database Tools		
Calculate Order	Add/Edit Customer Info	Create Invoice
D9 fx		
	A	B
1	Customer Name:	Rachel Patrick
2	Order Name:	Rachel Flowers
3	Subtotal:	\$202.04
4	Tax:	\$13.64
5	Total due:	\$215.68
6		

The CRM is also updated, incrementing the customer’s billing history (far right column of screenshot) by the amount stored in “Total due”:

A	B	C	D	E
Name	Phone Number	Email	Favorite Flower	Billing History
Steve Anderson	8012347865	steve@gmail.com	Lisianthus	\$ -
Frank Stevens	8017910447	frank@empty.com	Alstromeria	\$ -
Steve Jackson	8927892345	steve@jackson.com	Amaranthus	\$ 245.04
Skylar Peterson	4923458976	skylarp@gmail.com	Oriental White Single Bloom	\$ 327.14
Joanna Rogers	9876531234	joannar@gmail.com	Amaranthus	\$ 268.05
Rachel Patrick	8744892314	rachelp@gmail.com	Tulip	\$1,081.82

Finally, the company’s financial information is updated:

2014	
Revenue	\$1,316.21
Costs	\$661.95
Profit	\$654.26

- 5) Now we will look briefly at the Add/Edit customer function. This is accessed either by the specified button on the ribbon, or when an invoice is created and the customer isn’t currently in the system and therefore isn’t in the customer drop down box. The user would then click the “Add/Edit Customer” part of the “Create Invoice” form as mentioned above.

The screenshot shows a 'Customer Information' dialog box with the following fields and values:

- Name: Skylar Peterson
- Phone: 4923458976
- Email: skylarp@gmail.com
- Favorite Flower: Plumosa
- Billing History: 327.14

Buttons at the bottom: Cancel, Save Changes, Add New.

“Save Changes” makes the inputted changes to a selected customer, “Add New” adds a new customer to the CRM. The actual CRM screenshot is pictured above.

Discussion of Learning

A lot of funny things happened throughout the process that I had to figure out how to deal with. A few examples:

- When an order didn't need 10 flower types, the form would be submitted with empty text fields that were expecting numbers. This made it so my program wouldn't compile because the program would attempt to perform math with an empty string, so I had to go through and set the empty strings equal to 0 to allow the program to compile. I had a lot of issues similar to this that forced me to be creative to make the program act one way when it received expected data, and another when the wrong data was put in.
- I had a lot of trouble with the math involved. For example, if an order called for 11 flowers and that flower type comes in bunches of 10 the program would need to tell the user to order two orders of that flower. I originally just used integer division with the round function, but that wouldn't work because I fixed the previously mentioned problem by setting empty strings equal to 0. When the integer division occurred, it would have to divide by 0 with the flower types that were originally empty strings, which caused an error. To get around that I set the applicable variables as type double, set values that were empty strings to be equal to -0.5, then used the following code to round:

```
ordersOfFlower1 = Round(((totalStems1 / numberOfStems1) + 0.5), 0)
```

I'm sure there are other, better ways to solve these problems but I was proud of my ability to find a work around solution that ended up working. Getting the program to work correctly through this part was especially time consuming and difficult, especially accounting for different user inputs.

- Another challenge was juggling many different data points that were stored in different locations, and being sure to change the correct instances at the correct times.
- I had to learn a couple new techniques, such as how to create a new worksheet with a name defined from user input on a form. Also, I missed the class on modifying the ribbon so I had to learn that on my own. Thank goodness for internet tutorials (although I'm sure I would have learned it a lot faster and had a more enjoyable time had I learned it from Gove Allen).
- Something I wanted to do but ran out of time was to have the invoice emailed to the customer automatically when it was created. However, Blushing Rose Floral customizes their invoices and makes them very pretty so I doubt they would have used this function. It still would have been fun to include though.

Main learning takeaways:

- Really simple tasks can become very difficult when the computer interprets something in a different way than you expected it to. When programming, it's essential to think carefully through each problem and figure out how the computer will respond. There is a good chance that the first solution you thought of actually won't be as easy as you thought, and an alternative method should be considered.
- When you ask a client what they want in a program, they get really excited and start listing off a lot of stuff that you may not have time or know how to do. I had to work with them closely to narrow down exactly what they needed, because if I had tried to do everything they wanted I would have gotten nothing actually done.
- Map your program out before you start to build it. I made the mistake of writing a lot of code without a complete understanding of what I wanted the finished product to look like, and ended up having to scrap 90% of it. Build a template of what you want it to look like, and then build out the algorithms needed to accomplish the needed tasks, and then write the code. This would have saved me a lot of effort.

Assistance:

The only assistance I received on this project was internet based, I watched a tutorial on modifying the excel ribbon and I needed to get syntax for few miscellaneous functions (round, add a new worksheet, etc.)