



ISYS 520 Final Project

Create a Grocery List

By Heather Dunnigan

4/17/2013

Executive Summary

The purpose of my project was mostly to solve a personal pet peeve. I like to bake and cook, but whenever I go grocery shopping, I am always in a hurry. Inevitably I will forget an ingredient that's pertinent to a recipe I want to make that week. This means I either have to use time and money to go back to the store, or I have to wait another week to make that specific recipe. Although my project does not solve a major business problem, this problem is important and relevant enough to me that I wanted to dedicate time to finding an automated solution for it. This will help me have a more enjoyable and efficient grocery shopping experience.

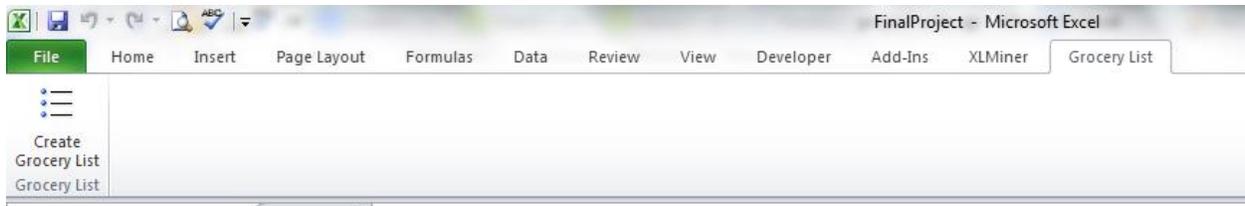
My solution is divided into three userforms. The main form allows the user to add recipes to a grocery list. A second form allows the user to add a new recipe to the list of recipes or to edit an existing recipe. The third form compiles the grocery list based on the recipes selected, and the user can edit the list and send it to their email.

Implementation Documentation

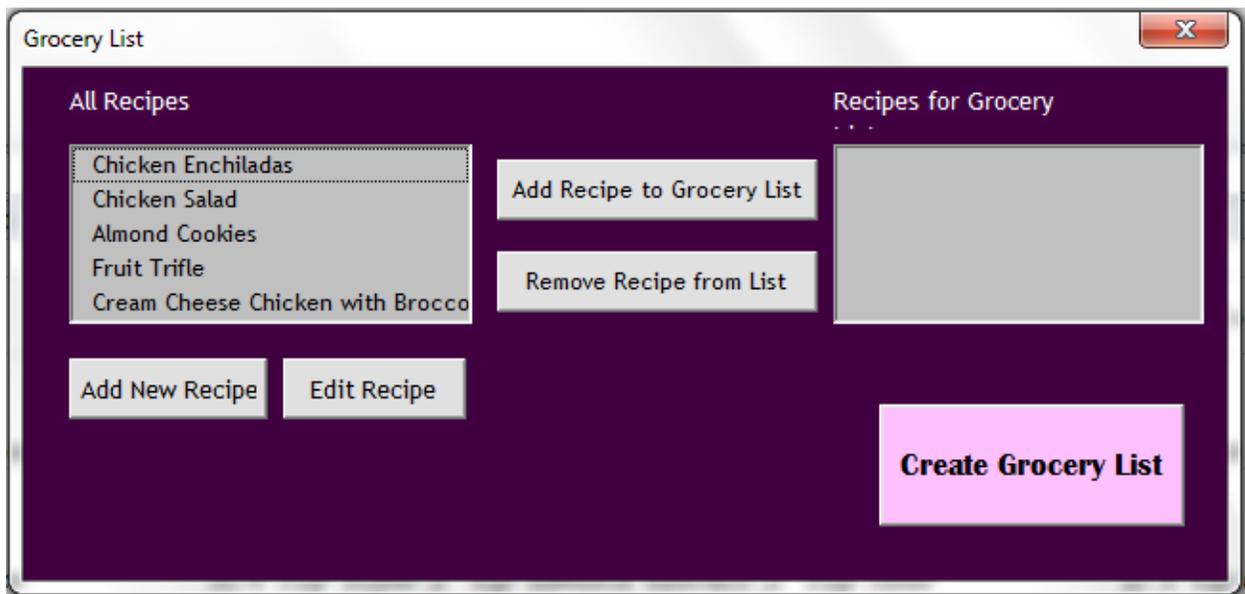
My project is based on user forms and the specific input of the user. This allows the user to customize the grocery list to their needs. A list of recipes is stored in the first tab of the workbook. If there were more than a few dozen recipes, it would make more sense to use Access to store the data. However, I usually cook the same twenty recipes, so I simplified the project to only act within Excel. For a visual frame of reference, this is what part of the list of recipes looks like.

	A	B	C	D	E	F	G
1	ID	Title	Directions	Ingredients			
2	1	Chicken Enchiladas	Mix all ingredients. Roll in tortillas and bake.	8 tortillas	3 can cream of chicken soup	1/2 can diced green chiles	8 oz sour cream
3	2	Chicken Salad	Mix all ingredients Serve on toast or a croissant	3 cups cubed, cooked chicken	1 1/2 cup diced cele	2 1/8 cup grapes	3/4 cup mayo
4	3	Almond Cookies	1) Preheat oven to 400 degrees 2) Mix all ingredients 3) Bake for 8 minutes 4) Make glaze 5) Glaze cookies	1 cup butter	3/4 cup sugar	1 tsp almond extract	2 cup flour
5	4	Fruit Trifle	1) Cut pound cake into cubes. 2) Slice berries, wash and add 1 tbsp sugar 3) Beat cream cheese and gradually add milk, pudding mix, and trifle 4) Put in bowl with layers from the bottom up as 1/2 the mix, cake berries, repeat 5) Put more cool whip on top	1 vanilla pudding mix	1 1/2 cup milk	1 package of cool whip	2 packages of cream cheese

The project can be initialized by clicking on the "Create Grocery List" command in the "Grocery List" tab of the ribbon.



When this command is clicked, it initializes the first of three userforms I used in my project. This main userform is used to interact with the other two forms. It can be seen below.



Before it is shown, a loop goes through all of the rows of the first worksheet in my project to get the names of the recipes and populate the list box. From here, any of the visible buttons can be clicked.

Add New Recipe

When "Add New Recipe" is clicked, this main Grocery List form shows the Recipes form. This initializes the form, which puts in defaults for three combo boxes. The three combo boxes with their defaults are whole amount {1,2,3,4,5,6,7,8,9,10}, fractional amount {"1/8", "1/4", "1/3", "1/2", "2/3", "3/4"}, and measurement {"tsp", "tbsp", "cup", "pint", "oz", "lb", "can"}. The empty Recipes form is seen below.

The user can edit the Title text box and the Directions text box by simply typing text. The user can also add ingredients by choosing one of the default values for whole amount, fractional amount, measurement, and item. If there is only a whole amount and not a fractional amount, the user can leave the field blank. By clicking add new ingredient, the new ingredient gets added to the Ingredients list box with the following code. Basically, the subprocedure gets the values from the combo boxes, concatenates them into one string, checks if the length of this combined string is greater than zero, and adds it to the list box.

```
Private Sub btnNewIngredient_Click()
Dim whole As String, frac As String, measure As String, ingredient As String, item As String
whole = cboWhole.Value
frac = cboFraction.Value
measure = cboMeasure.Value
item = txtItem.Value

ingredient = Trim(whole) & " " & Trim(frac) & " " & Trim(measure) & " " & Trim(item)
Debug.Print ingredient

If Len(ingredient) > 0 Then listIngredients.AddItem (ingredient)

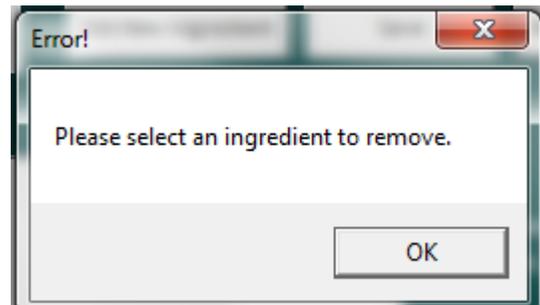
cboWhole.ListIndex = 0
cboFraction.ListIndex = 0
cboMeasure.ListIndex = 0
txtItem.Value = ""

End Sub
```

If a user wants to edit an ingredient, they can click on an ingredient in the list, and the combo boxes and the Item text box will be populated. The InStr function finds where there are spaces in the ingredient. It then uses the placement of the spaces and the Mid function to separate whole amount, fractional amount, measurement, and item into different String variables. Because not all ingredients have whole and fractional amounts, an if statement checks to see if there is a slash ("/") in the whole amount, which would indicate a fraction. If there is, the whole value is set to an empty string and the fractional value is set to the amount with the slash. A second if statement checks to verify that there is a slash in fractional amount. If there is not, the item String gets set to that String. This could happen for an ingredient like "chopped onion to taste" where there is no numeric amount associated with the ingredient. The comboboxes and text box on the form are assigned to these String variables.

From here, a user can change any of the four fields that make up ingredient. If they click save, the whole amount, the fractional amount, the measurement, and the item are again concatenated. The ingredient at the current list index is removed, and the new String is added to the list at the same index. Thus the ingredients are still in the same order. The ingredient combo boxes and text box are set to an empty string.

If a user clicks the Remove Ingredient button, the program checks to make sure that an ingredient is selected. If there is not a recipe selected, the following message box appears.



If there is an ingredient selected, the list removes the ingredient at the current List Index (current selection in the list). The ingredient combo boxes and text box are set to an empty string.

When the user clicks the Save Changes button, it initiates a subprocedure that loops through the recipes on the Excel spreadsheet. By comparing the title in userform to the title in the spreadsheet, it knows whether this recipe is already in the list. This means that the title is acting as a primary key for the recipe. Of all the cookbooks and recipes I have come across, a recipe usually has a unique name to be referenced by. I thought it would make the most sense to identify by the recipe by the name instead of something else. If it finds the name of the recipe, it updates column C of that row with the directions, and it loops through the list box to output the ingredients, starting in column D and continuing for all of the ingredients. If the title is not there, it moves to the first blank row and puts the title, directions, and ingredients data in that row.

The Recipes Userform is hidden and unloaded. The original Grocery List userform is updated with the new list of recipes on the spreadsheet and is shown.

Edit Recipe

If the Edit button is clicked on the original Grocery List userform, the same Recipes userform opens, but it is populated with the data about that recipe. For example, if I had selected "Almond Cookies", the program goes through the same looping process as above to match the title and find the correct row for

the recipe. The Ingredients list box is populated with the ingredients, the Title text box with the title, and the Directions text box with the title. The result is shown below.

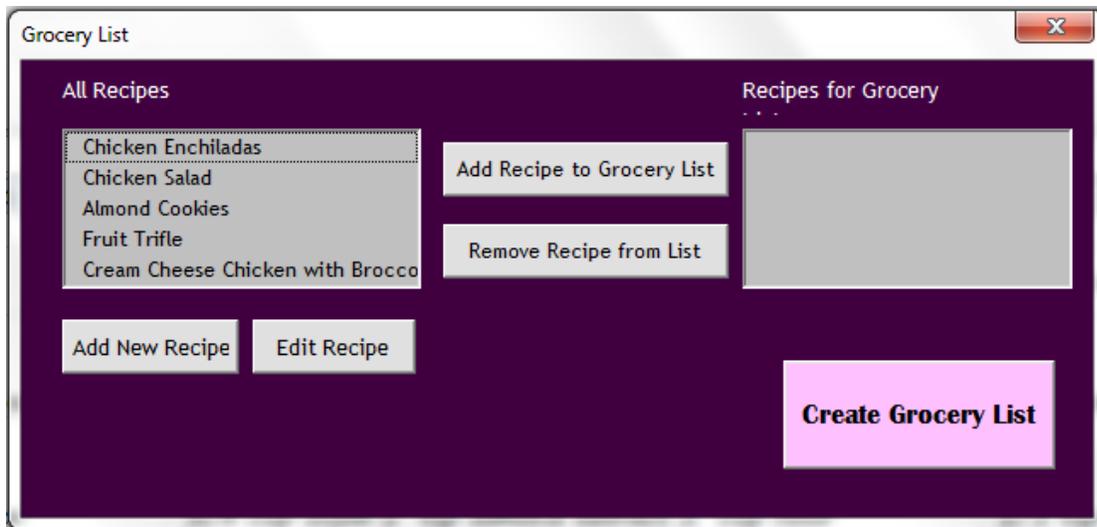
The screenshot shows a window titled "Recipes" with a dark green background. At the top, there is a title text box containing "Almond Cookies". Below this is an "Ingredients" section with a list box on the left containing items like "1 cup butter", "3/4 cup sugar", "1 tsp almond extract", "2 cup flour", "1/2 tsp baking powder", "1/4 tsp salt", and "1 1/2 cup powdered sugar". To the right of the list box are three dropdown menus labeled "Amount", "Measurement", and "Item", with values "3/4", "cup", and "sugar" respectively. Below these are three buttons: "Add New Ingredient", "Save", and "Remove Ingredient". At the bottom is a "Directions" section with a text area containing a numbered list: "1) Preheat oven to 400 degrees", "2) Mix all ingredients", "3) Bake for 8 minutes", "4) Make glaze", and "5) Glaze cookies". To the right of the text area is a note: "(Press Shift+Enter for a new line.)" and a "Save Changes" button.

From there, the same interaction and logic occur as they did with adding a new recipe.

Add/Remove Recipe from Grocery List

Once the recipes are correct, the user can choose to add a recipe to the grocery list by clicking on the Add Recipe to Grocery List button of the Grocery List userform, shown again below for reference. It checks to make sure that a recipe is selected, and if there is not a message box pops up to alert the user to select a recipe. If there is a recipe selected, the name of the recipe is removed from the listbox on the left and added to the list box on the right.

The same logic and alert occurs for the Remove Recipe from List button, but instead the user moves a recipe name from the right list box, Recipes for Grocery, to the left list box, All Recipes.



Create Grocery List

When the Create Grocery List button is clicked, the result is the Finalize Grocery List userform seen below, but a lot of logic happens between those two states. First, a set of embedded for loops makes a grand list of ingredients. The first loop goes through each title in the Recipes for Grocery list box and matches it to a title in one of the rows in the spreadsheet of recipes. The second loop goes through every ingredient associated with that recipe and adds to it a master list of ingredients, which is created on the second worksheet of the workbook, Grocery List. Once the list has been created, the ingredients are parsed in the same manner as if they had been clicked on in the Recipes list box. This separates the whole amount, the fractional amount, the measurement, and the item. These values are put in the cells to the right of the original ingredient. The result is the following.

	A	B	C	D	E	F	G	H
1	Grocery List	Whole	Frac	Measure	Item			
2	8 tortillas	8			tortillas			
3	3 can cream	3		can	cream of chicken soup			
4	1/2 can dice		1/2	can	diced green chiles			
5	8 oz sour cre	8		oz	sour cream			
6	chopped oi				chopped onion to taste			
7	1 lb chicken	1		lb	chicken			
8	1 lb cheese	1		lb	cheese (cheddar, monterey jack, mexican)			
9	1 tbsp milk	1		tbsp	milk			
10								

Another for loop goes through and checks just the value in column E, Item, against the other Item values. If a match is found, the whole amounts are added together and the fractional amounts are added together. These sums are put in columns B and C of one row, and the other row for the same ingredient is set to have 0 values. This is shown in the following example.

	A	B	C	D	E	F	G	H
1	Grocery List	Whole	Frac	Measure	Item			
2	8 tortillas	8			tortillas			
3	3 can cream	4		can	cream of chicken soup			
4	1/2 can dice	1/2		can	diced green chiles			
5	8 oz sour cre	8		oz	sour cream			
6	chopped on				chopped onion to taste			
7	1 lb chicken	4		lb	chicken			
8	1 lb cheese	1		lb	cheese (cheddar, monterey jack, mexican)			
9	1 tbsp milk	1		tbsp	milk			
10	3 lb chicken	0	0	lb	chicken			
11	1 tbsp olive	1		tbsp	olive oil			
12	1 2/3 oz dry I	1	2/3	oz	dry Italian dressing mix			
13	2 cup mushr	2		cup	mushroom			
14	1 cup onion	1		cup	onion			
15	1 can cream	0	0	can	cream of chicken soup			
16	10 oz frozen	10		oz	frozen broccoli florets			
17	3 cup pasta	3		cup	pasta			
18								

The three cans of cream of chicken soup in row 3 are added to the one can in row 15. There are a total of four cans, which is recorded in row 3. The values in row 15 are zeroed out. The same process happened with chicken in rows 7 and 10.

Once the amounts of the ingredients are aggregated, a final loop goes through the ingredients. If the values of columns B and C for that row/ingredient are not 0, that ingredient is added to the Grocery List list box of the Finalize Grocery List userform. The list above is shown in the userform below with only summed ingredients.

Edit Recipe

The user interaction with the userform and the logic for editing amounts or measurements of ingredients is the same as in the Recipes userform. A user can add a new ingredient to the list if they want to buy something like “4 bananas” that is not in a recipe. A user can edit and save an ingredient, and a user can also remove an ingredient if they already have that ingredient.

Send Grocery List

The final step and the whole purpose of this process is to send a grocery list. The user can enter an email address in the Email text box and click the Send Grocery List button. If there is not an email typed, a message box pops up and prompts the user to enter an email address. If there is, the program loops through the items on the Ingredients list box and adds them to a string variable with a line between each item. This is shown by the following code.

```
For i = 0 To listGrocery.ListCount - 1
email_body = email_body + listGrocery.List(i) + vbNewLine
Next
```

Once the String variable email_body is created with all of the ingredients, an email is set up to send through Gmail’s email server. Setting this up involves setting the port, the smtpserver, the username and password, and the attributes of the actual email message such as subject, to, from, and email body. ***Note: sending an email will not work without putting in your email and password for the following two lines.

```
iConf.Fields.item(schema & "sendusername") = "heatherdunnigan27@gmail.com"
```

```
iConf.Fields.item(schema & "sendpassword") = "*****"
```

Once the email is ready, it is sent using message.send and a message box pops up to confirm to the user that the grocery list has been sent. Once the grocery list is sent, clicking the Done button unloads the userforms, which closes the program.

Difficulties Encountered

While I was working on the project, the most difficult part was aggregating the amounts of the ingredients. While using arrays or collections, I struggled to try to keep track of separate parts of the ingredient, such as amount, measurement, and item. I ended up using a second worksheet to parse the ingredients and interact with them. This could have been solved by using Access and having a recipe table and a separate ingredient table that had recipe as a foreign key. However, I was too far into the project when I thought of this, so I did not change my project.

Another challenge was interacting more intently with user forms, list boxes, and combo boxes. While this took time to learn, it did not pose too much of a challenge. I spent time experimenting and looking at online resources to become more familiar and comfortable with userforms.

While creating this program, I did not receive outside help, except from websites such as stackoverflow.com when I was debugging. Online resources solved my problems easily enough that this project was very enjoyable.

If I wanted to expand the scope of this project, the main feature I would add would be adding recipes from websites. Some cooking websites have layouts that would be simple enough to interact with through VBA. This would allow the user to add recipes more quickly and efficiently. Because learning to interact with the user forms took longer than I expected, I did not get a chance to add this feature.

Another smaller feature to add would be to text the grocery list to the user instead of or in addition to the email. This would be especially helpful if you don't have a smartphone.

Conclusion

Overall this project was very beneficial in helping me learn, and I became even more comfortable with VBA than I had been throughout the semester. I also solved a problem that has plagued since I got to college. I will now be able to create a grocery list in just a few seconds instead of a few minutes.

***Please note that the grocery list will not be sent unless you change the email and password in the code for the Finalize Grocery List form.