

Lacey Woodfield's Grocery Manager

Executive Summary

The Grocery Manager is a system that helps to organize the general consumer's shopping list. Not only does this tool allow shoppers to put together shopping lists with their recipes, but it also provides a compiled record of their recipes by allowing them to input and save recipes in the system. Then, when they wish to create a grocery list, they can choose from this list of recipes to put together a shopping list. Further, before printing the list, the user can add other items he or she might need in an "Other Items" section, and these other items will also be added to his or her grocery list. Finally, when the user has all the recipes and other ingredients that he or she wishes to shop for, he or she can "print" the list and the tool will add together like ingredients, organize them into categories, and prepare the list to be printed out. This way the user can remove items that he or she already has by looking through each category. The categories are also useful when shopping at the Grocery Store as items in the store are generally grouped by these same categories.

By using this shopping tool, the user can expect to be able to shop with convenience and not miss ingredients they need for the recipes they wish to make in any given week.

Implementation

The Grocery Manager Userform includes 3 pages to help with grocery needs.

The first page is the "Choose Recipes" page. This page is made up of two lists (see Figure 1 on right). The "All Recipes" list contains a list of all recipes that have already been entered into the Grocery Manager. These recipes are recorded on a sheet called "Recipes" in the workbook. This sheet also contains information such as the source of the recipe, how many servings/ how many units the recipe makes, instructions for using the recipe, and finally, a recipe ID number (which is assigned when the recipe is added). From the "All Recipes" list, the user can choose recipes to add to the "Recipes Chosen" list. The default value for the quantity of recipes to add is 1, but this can be changed. If the user adds the same recipe more than once, that recipe's quantity in the "Recipes Chosen" list will increase accordingly. Finally, the user can also remove recipes from this list using the Remove button to the right of the "Recipes Chosen" list.

The second page of the userform is the "Add New Recipe" page. This page (pictured on right) is used to add new recipes to the worksheet that have not been used before. The user enters the recipe title, source, instructions, and amount the recipe makes on the right of the page and adds the ingredients one by one on the left. In the ingredients section, the combo boxes are populated with values that have been used in other recipes by using a page called "Ingredients," where all the ingredients have been stored. This page organizes ingredients by category and gives each of them an ID number.

When the user adds an ingredient from this list, the Category combo box is automatically populated with the category that was entered before for that item. For example, when the user enters the word "Milk" as an

ingredient, the category automatically becomes “Dairy” and is disabled. This was done using the “Lookup” function in a sub that activates whenever a change is made in the “Ingredient” combo box. The “Units” combo box for this section is populated with values restricted to those which have been accounted for so that the ingredients can be accurately added together later. The amount entered for ingredients is evaluated when the ingredient is added to the list and a sub makes sure that a number was entered. If not, the user receives an alert. Similar tests are implemented to make sure that the user has filled out all of the required sections in adding a new ingredient. The user also may not add an ingredient more than once, and will receive an alert prohibiting them from doing so if they try to. If they have made a mistake, they can remove ingredients from this list with the remove button on the right.

Finally, when all of the ingredients have been added, the user can choose whether they wish to add this recipe to their current shopping list by checking the box and providing a quantity above the “Add Recipe” button. Then the user can add the recipe by clicking this button. When they do, the program makes sure that all of the fields on the recipe page have been properly filled out. If they have not, the user will receive appropriate

alerts. For example, the user may not enter a recipe that has the same title as one that was previously entered. If they have correctly filled out the “Add New Recipe” form, the recipe is added to a worksheet called “recipeltem” in the workbook. This sheet records the ingredients for recipes (shown on the right). The first column has the recipe ID, the second has the ingredient ID, the third has the amount of the ingredient, the fourth has the units of the amount, and the last column is an optional column that contains a preparation method for the ingredient. After the recipe has been added, it will appear among the other recipes in the “All Recipes” list on the first page of the userform and can be added to the grocery list on that page if the user neglected to check the “Add to current shopping list” button on the “Add Recipe” page.

	A	B	C	D	E
1	recipeID	IngredientID	Qty	Units	preparation
13	2	6	3	Whole	.
14	3	12	1	Cup	.
15	3	13	1	Cup	.
16	3	8	1	Cup	.
17	3	9	1	Teaspoon	.

The final page in the userform is the “Other Items” page (pictured right). This page can be used to add other items to the shopping list that were not necessarily part of a recipe. For example, the user could add lunchmeat or apples in this section. Similar to the ingredients section on the “Add New Recipe” page, the combo boxes on this page are populated with the ingredients and ingredient categories that have already been added. Also, similar to the “Add Recipe” page, if an ingredient is chosen that has been previously used, the “Category” combo box will populate with the correct category and become disabled. The “Units” combo box on this page is populated with the unit values from the “Add New Recipe” page, but it has additional values such as “Can,” and “Dozen” which come from the “Other Ingredient Units” page in the workbook. This page simply stores and records different units when they are used on this “Other Items” page. This page, like the other pages, allows the user to remove items from the list if they wish.

Once the user has added all of the recipes and other items that they wish to, they can click the “Print List” button. This button represents the bulk of the programming in this project. When it is utilized, the program puts together a list of the recipe IDs included in the “Recipes Chosen” list. This is done by looping through the list and using the Lookup function. This list of recipe IDs is then fed to a filter, which filters the recipeltem page to only include ingredients from the recipes that were chosen. Then, the program loops through the visible cells and prints their values onto a hidden sheet called “summary,” multiplying each recipe by the

quantity which was also recorded in the “Recipes Chosen” list. Then, the values from the additional Items list are added into “summary” with the ingredient name, category, quantity, and units in their corresponding columns. A sub loops through each of these, removing items with units that cannot be converted (whole, dozen, can, etc.) and moves these items to a page called “Additional Summary.” In the summary sheet, all the ingredient’s units are converted to Teaspoons. Then, the entire list is sorted by ingredient type and like ingredients are added together with a code that loops through each row, checking to see if the ingredient IDs match. If they do, the amount in the row below is added to the row above and the row below is deleted. At this point, the ingredients have been consolidated and are all in values in terms of teaspoons. These values are then converted to their largest unit of reduction with an iterative, recursive function that returns the teaspoon values (given as numbers) as reduced measurements with volumetric units. For example, the number 14 (which is in teaspoons) would return “4 Tablespoons, 2 Teaspoons.”

The “Additional Summary” sheet is treated differently. This sheet loops through each ingredient, looking for the unit “Dozen.” It multiplies entries with these units by 12 and changes their units to “Whole.” Then, it sorts by ingredient type and adds like ingredients together, only if they have the same units. Finally, it reduces items with the unit “Whole” into dozens if they have quantities greater than 12. For example, if you had 15 whole eggs, it would reduce to “1 Dozen, 3 Whole.” After these reductions have been performed, the other ingredients that have units that were not altered must be reformatted to match the altered units. This is done by joining the quantity value with the unit value as a string and then implementing this value as the new

“quantity” value. When all the items in the “Additional Summary” sheet have been properly formatted, these ingredients are printed back into the “Summary” sheet, below the ingredients that have been put in volumetric units. The appropriate ingredient names and categories are imported to the “Summary” sheet using the “Lookup” function in the ingredients sheet. Finally, these ingredients are sorted with two levels: first by category and then by ingredient name. This ensures that the ingredients in the final shopping list are in alphabetical order.

Grocery List	
Baking	
1 Teaspoon	Baking Powder
1 Teaspoon	Baking Soda
1 Teaspoon	Cocoa
1 Pint	Flour (White)
1 Pint	Sugar
1 Tablespoon	Vanilla
1 Cup	Vegetable Oil
Dairy	
3 Whole	Eggs
Nuts	
1 Teaspoon	Nuts
Salt, Spices, Seasoning	
1 Tablespoon	Cinnamon
1 Teaspoon	Salt
Vegetables	
1 Pint	Zucchini

The ingredients are then reformatted as they are transferred to the “Shopping List” sheet, making the final shopping list readable. The final formatting makes the list white with a blue background, autofits the columns on the “Shopping List” sheet, and sets the print area to only contain the shopping list. An example of a final shopping list is shown on the left.

Learning and Improvements

One of the most important things I learned throughout the course of this project was how time intensive seemingly easy tasks can be when programming. When you use a program with a userform, you don’t typically think about all the time it took to program every alert and every warning that helps guide the user toward filling out the form correctly. While programming the grocery manager, I felt like the largest portion of my time was spent trying to foresee possible problems and figure out how to account for them in the code to help the user. I learned that coding is much more creative than I initially thought because, some things that we take for granted and that are easy for our brains to figure out how to do are very difficult to program into code. I also learned that, as you go, you get better at doing certain tasks and you figure out different, better ways, to accomplish the same things. Some of the code I wrote in the beginning that was used over and over again would be improved each time it was used.

There are a few elements that I would have liked to include, but time did not allow for them. One of them was being able to convert the amounts of certain ingredients into ounces rather than volumetric units like cups, quarts, etc. That way, the user could look at the list at the end, and the ingredients that would be easier to shop for in different units could be chosen and then easily converted to new units with a button on the shopping list page called “Convert to Ounces.” I think it might also be useful to set different defaults for

different units, like maybe have different base units for wet vs. dry ingredients in the part of the code where the ingredients are added up. I would like to add another tool that allows the user to view the recipes in a user-friendly format. This would also allow them to use the tool while cooking. Finally, I would like to add a feature that lets the user search for recipes that contain a certain ingredient. Each of these features I hope to add to this project as I continue to improve it.

Assistance

I received considerable assistance from Dr. Gove when implementing this project. As a result, I would like to here express my appreciation for this help.