

Inventory Management System for Home Kitchens

Contents

- Executive Summary:..... 3
- Inventory Management System Details:..... 3
 - Weekly Meal Planning: 3
 - Inventory:..... 7
 - Meal Input (Manual): 8
 - Online Meal Input: 10
 - Add Ingredient: 11
- Challenges: 13
- Future Expansion: 15
 - Recipe Loading Tool: 15
 - Enhanced User Interface:..... 15
- Assistance: 15

Executive Summary:

Most home kitchens do not have the resources to implement an inventory management system. Kitchens run without an effective inventory system can lead to spoiled food, which equates to lost money. The Inventory Management System I created in Excel/VBA creates a cost effective method to control inventory which by reducing the amount of wasted food, significantly reduces the grocery bill. I have been using this system for the past several months and I have reduced my food bill from about \$1,500 to about \$1,000 (I have a big family, 7 total). The inventory system uses weekly meal planning and the current inventory to identify items which need to be purchased, and it then generates a grocery list based on these items. In general, the grocery list specifies the ingredients which must be purchased to cook the meals specified by the weekly planning.

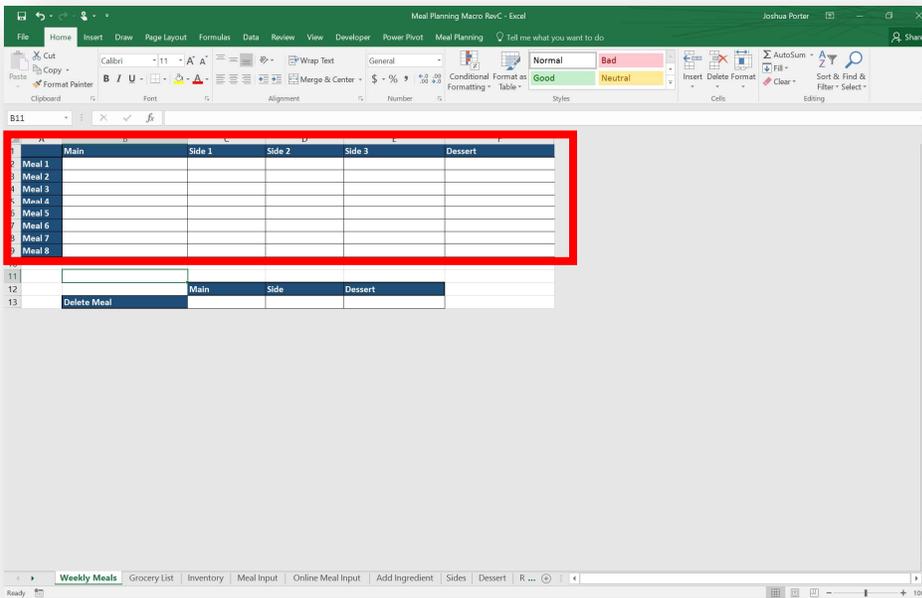
There are several additional tools which supplement the weekly meal planning tool. These tools are the manual and online meal inputs. These tools allow the user to add meals to the database which can then be selected by the weekly meal planning tool. Additionally, an ingredient input tool allows a user to add an ingredient to the Inventory, which then allows the user to select that ingredient for a meal. Additionally, if desired the user can eliminate/delete a particular meal from the database through a tool located within the weekly meal planning tool. After deleting a meal, it can no longer be selected on the weekly planning tool. Additional details and how the tools work together can be found in the “Inventory Management System Details” section.

Inventory Management System Details:

The tools within the inventory management system work together to create a grocery list of only the items which need to be purchased for that week. Below is a discussion for each tool and how it interacts with the other tools.

Weekly Meal Planning:

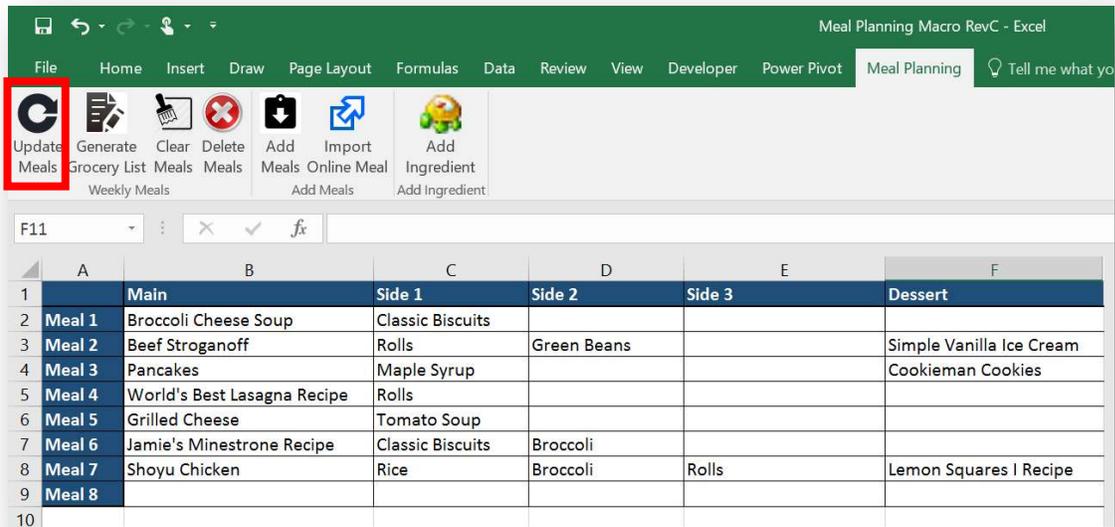
The weekly meal planning tool is located on the “Weekly Meals” tab. Below is a screenshot of the tool (boxed in red).



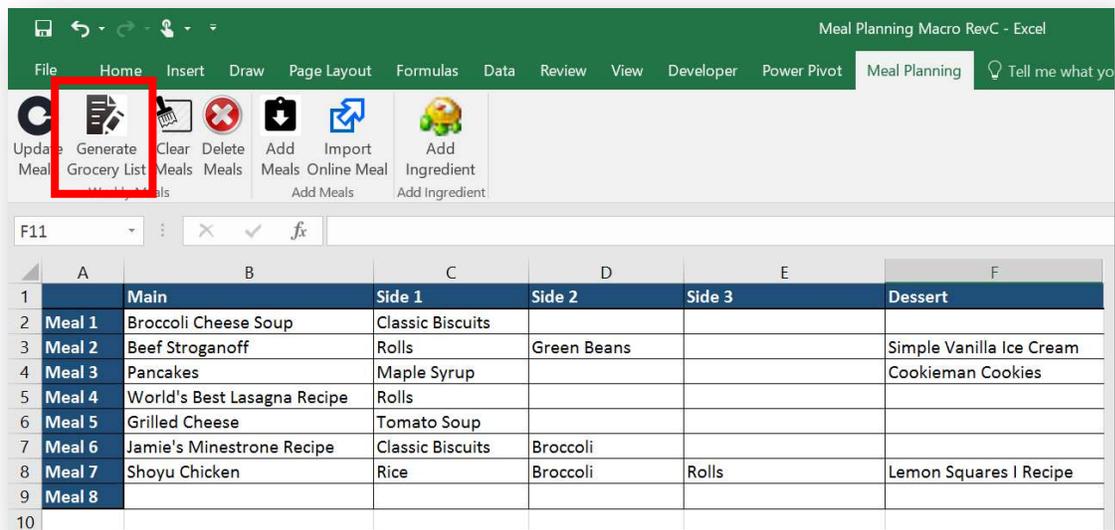
The user can select up to 8 meals (one for each day of the week plus an extra day). For each meal the user can select up to 3 side dishes and a dessert. The user can only select main dishes, side dishes and desserts which have been loaded into the database as seen below.

	Main	Side 1	Side 2
Meal 1			
Meal 2	BBQ Chicken		
Meal 3	Fish		
Meal 4	Spaghetti		
Meal 5	Pancakes		
Meal 6	Beef Stroganoff		
Meal 7	Broccoli Cheese Soup		
Meal 8	Oven Fajitas		
	Grilled Cheese		
Meal 8			

Main dishes, side dishes and desserts can be added or deleted to the database through separate tools, which will be discussed later in this document. The user presses the “Update Meals” button in the ribbon (shown below in the red box) once they have completed specifying their desired main dishes, side dishes and desserts. Pressing the “Update Meals” button will generate meal based demand for the ingredients listed in the inventory tool, this will be explained further in the “Inventory” Section. Please note that the “Update Meals” button will only function when the “Weekly Meals” sheet is active. If the user is on a different sheet then the button will give the user the option to activate the “Weekly Meals” sheet.



Additionally, once the user has selected the weekly meals, and has updated the inventory (discussed further in the “Inventory” section) they can select the “Generate Grocery List” button (as seen below).

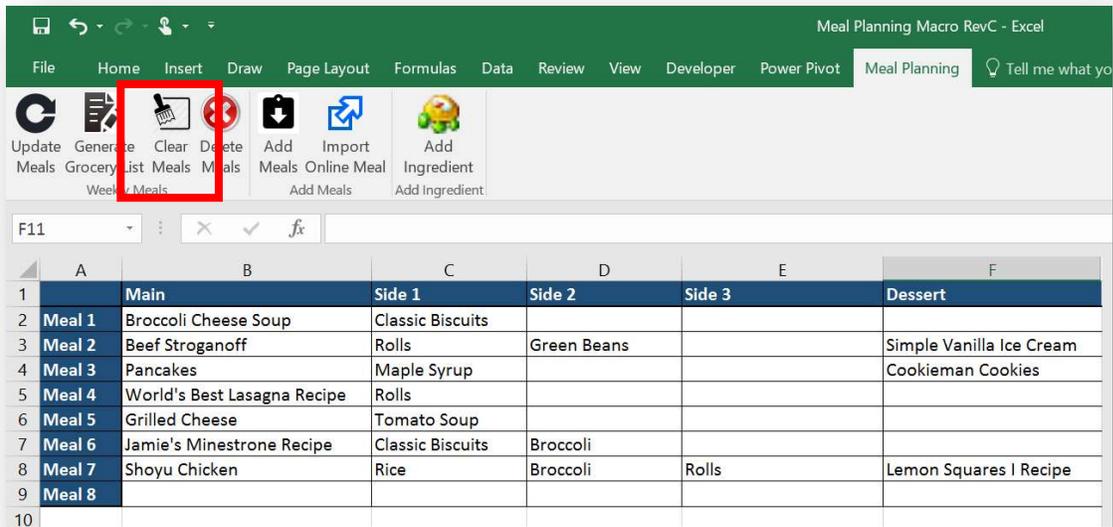


This button will use the “Buy” column on the “Inventory” sheet to create and format a shopping list. A sample shopping list can be seen below.

	A	B	C	D
1	Item	Qty	Category	Price
2	Fennel Seeds	1	Baking	\$3.50
3	Olive Oil	1	Baking	\$5.00
4	(28 ounce) Can Crushed Tomatoes	3	Canned	\$2.00
5	Almond Milk	1	Dairy	\$3.50
6	Butter	2	Dairy	\$2.50
7	Cheddar	1	Dairy	\$13.00
8	Eggs	9	Dairy	\$2.00
9	Milk (whole)	1	Dairy	\$2.99
10	Peach Yogurt	1	Dairy	\$3.00
11	String Cheese	1	Dairy	\$9.00
12	Yogurt (Plain)	1	Dairy	\$2.50
13	Pop	1	extra	\$4.00
14	Bananas	4	Fruit	\$0.60
15	Grapes	2	Fruit	\$4.00
16	Strawberries	1	Fruit	\$3.50
17	Bread	2	Grain	\$2.00
18	Hamburger Buns	1	Grain	\$1.78
19	Lasagna Noodles	1	Grain	\$3.00
20	Fish	9	Meat	\$3.00
21	Lettuce	1	Vegetable	\$1.68
22	20 Items	Total Price	=	\$126.85

The shopping list includes an item description, the quantity which must be purchased, the category and the unit price. The list also includes an estimated total cost. After the shopping list has been produced, by pressing the “Generate Grocery List” button, the user is given the option to print the list. If the user presses “Yes” than the program will automatically take the user to the print preview screen, with the list properly formatted to fit onto one page. If, however the user selects “No” then the “Grocery List” sheet is activated.

Finally, once the week is over the user can easily start clear the previous week’s selection (in the “Weekly Meals” sheet) by selecting the “Clear Meals” button in the ribbon (shown below).



Inventory:

In many ways the inventory is the backbone of the system. Each week the user must take stock of the items in inventory. An easy way to perform this task is to filter the ingredients based on category and update one category at a time. Below is a sample screenshot of the "Inventory" sheet. A description of each column can be found below.

1											
2	Current Inventory	QTY	Price	Additional	Amount	Unit	Category	BUY	Min. Inventory	Meal Req	
3	Cake Mix	0	\$ 3.00		1	Each	Baking	0	0	0	
4	Frosting	0	\$ 3.00		1	Each	Baking	0	0	0	
5	Pam	1.2	\$ 2.50		1	Each	Baking	0	1	0	
6	Eggo Waffles	0	\$ 3.00		1	Each	Boxed	0	0	0	
7	Fajita Sauce Packet	0	\$ 0.98		1	Each	Boxed	0	0	0	

Current Inventory: A description of each inventory item.

QTY: current quantity in inventory. Please note, the user must update this field each week.

Price: The unit price of an item. This should only be updated if there has been a large change in price.

Additional: This field allows the user to add demand beyond what is required for meals or the minimum inventory. Conversely, the user can reduce demand by adding a negative number i.e, if the "BUY" field is 4, a "-1" will reduce the "Buy" field to 3.

Amount: This is the amount in 1 unit, for example one bottle of ketchup might contain 4 cups, or 1 dozen eggs might contain 12 individual eggs.

Unit: This field specifies the units used. They can be Each, weight (lbs), or volume (cups).

Category: A category field has been added to organize the shopping and inventory list.

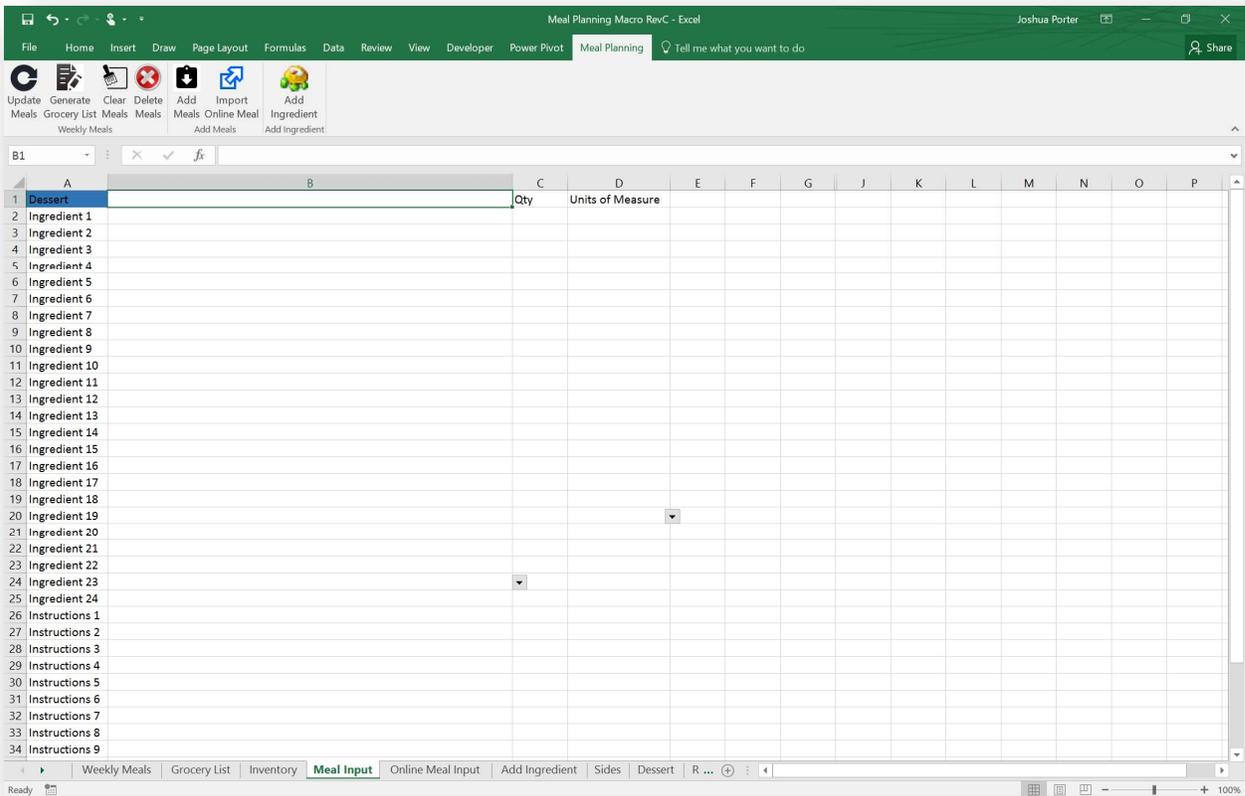
BUY: is a calculated field which adds the “Min Inventory”, the “Meal Req” and the “Additional” field it then deducts the “QTY” field and rounds up to identify how much of each item must be purchased this week. Please note that this field also adjusts for the different units, for example, perhaps the minimum inventory for eggs was 3 dozen, meals required 3 eggs and because Easter is coming up an additional 2 dozen eggs are required. The current inventory of eggs is 2 dozen, so the “BUY” column would make the following calculation $3 + 3/12 + 2 - 2 = 3.25$ this would then be rounded up to show 4 dozen eggs must be purchased.

Min Inventory: This field allows the user to specify a minimum inventory which must be on hand. For example, perhaps you only need frosting when you want to bake a cake, so your minimum might be 0. However, you might always want to have some flour on hand, just in case, so you might specify that the minimum inventory for flour is 1.5 packages.

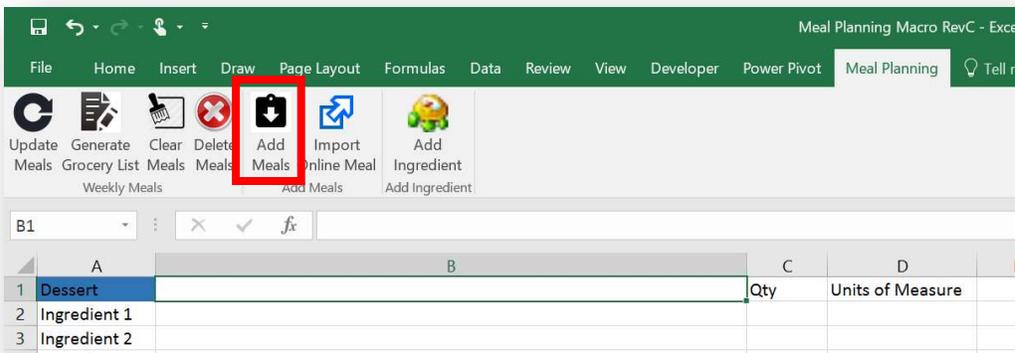
Meal Req: This field sums the demand for each ingredient based on the meals specified from the “Weekly Meals” sheet (after the “Update Meals” button has been pressed).

Meal Input (Manual):

The “Meal Input” sheet allows a user to manually add a main dish, side dish or dessert. Below is a screen shot of this sheet. In cell “A1” the user specifies the type of meal, Main, Side or Dessert from a drop down list. Additionally, cell “B1” allows the user to specify the recipe title. There are up to 24 ingredients and 15 instructions which can be specified in a recipe. In cells “B2” to “B25” users can select an ingredient from a dropdown list. The user then specifies the quantity and selects the unit of measure from a dropdown list. For example, the recipe might call for 4 tablespoons of butter. Butter would be selected in the column B, 4 would be typed into column C and tbsp would be selected in column D. The user continues to add all the ingredients for a recipe. After the ingredients are added the user then adds any instructions, “Bake at 375 deg” etc.



Once the ingredients, and the instructions have been added for the recipe the user selects the “Add Meal” button (as shown below). This will add the meal to the excel database, converting the units to match those in the “Inventory” sheet in the process. Continuing with the butter example above, the 4 tbsp would be converted to cups, because the unit of measure in “Inventory” is cups or 4oz of chicken would be converted to lbs. Now that the meal has been added to the excel database the user can select it in the “Weekly Meals” sheet.



Online Meal Input:

The online meal input is by far the simplest way to add a meal. The user copies a URL into “B1” for a meal they want to add to the database and they select a meal type in cell “A3” (Main, Side or Dessert). Please note that currently the program only works for URLs from “allrecipes.com.” Then the user presses the “Import Online Meal” from the ribbon. This button goes to the website, grabs the title, ingredients, units of measure, quantity and directions and puts them into the “Online Meal Input” sheet as shown below. Additionally, because the ingredients pulled from the website might not match the ingredients in the inventory, the program maps the ingredients pulled from the online source to the ingredients listed in Inventory. Below is a sample recipe from “allrecipes.com” with the accompanying conversion.

The image displays two screenshots side-by-side. The left screenshot shows a web browser window with a recipe page for "Funnel Cakes IV" from allrecipes.com. The page includes the recipe title, a star rating, a photo of the food, and a list of ingredients. The ingredients list includes: 1/2 teaspoon salt, 2 teaspoons baking powder, 3 2/3 cups all-purpose flour, 3 eggs, 1/4 cup white sugar, 2 cups milk, 1 quart vegetable oil for frying, or as needed, and 2 tablespoons confectioners' sugar, or as needed. The right screenshot shows an Excel spreadsheet with the recipe data imported from the website. The spreadsheet has columns for Web Address, Ingredient, Qty, Units of Measure, and Mapped Ingredient. The data is as follows:

	A	B	C	D	E
1	Web Address:	http://allrecipes.com/recipe/3265/funnel-cakes-iv/?internalSource=previously%20saved&referringContentype=home			
2					
3		Funnel Cakes IV Recipe	Qty	Units of Measure	Mapped Ingredient
4	Ingredient 1	salt	0.5	tsp	Salt
5	Ingredient 2	baking powder	2	tsp	Baking Powder
6	Ingredient 3	all-purpose flour	3.666667	Cups	Flour
7	Ingredient 4	eggs	3	Each	Eggs
8	Ingredient 5	white sugar	0.25	Cups	Sugar
9	Ingredient 6	milk	2	Cups	MILK 2%
10	Ingredient 7	vegetable oil for frying, or as needed	1	quarts	Vegetable Oil
11	Ingredient 8	confectioners' sugar, or as needed	2	tbsp	Sugar
12	Ingredient 9				
13	Ingredient 10				
14	Ingredient 11				
15	Ingredient 12				
16	Ingredient 13				
17	Ingredient 14				
18	Ingredient 15				
19	Ingredient 16				
20	Ingredient 17				
21	Ingredient 18				
22	Ingredient 19				
23	Ingredient 20				
24	Ingredient 21				
25	Ingredient 22				
26	Ingredient 23				
27	Ingredient 24				
28	Instructions 1	Mix salt, baking powder, and half the flour in a bowl. Set aside.			
29	Instructions 2	Cream eggs, sugar, and milk in a large bowl. Add flour mixture and beat until smooth.			
30	Instructions 3	Heat the oil to 375 degrees F (190 degrees C) in an 8-inch skillet.			
31	Instructions 4	Put your finger over the bottom opening of the funnel, and fill the funnel with a generous 1/2 cup of the batter. Hold the funnel close to the surface of the oil, and release the batter into the oil while making a circular motion. Fry until golden brown, use tongs and wide spatula to turn the cake over carefully. Fry the second side one minute. Drain on paper towels, and sprinkle with salted confectioners' sugar.			
32	Instructions 5				
33	Instructions 6				
34	Instructions 7				

Notice that the ingredient description has been correctly placed in column B, the quantity in column C and the units of measure has been placed in column D. Additionally, the instructions have been correctly placed in column B, but after row 27. The program has also attempted to map the websites description to the Inventories description. In most cases the mapping is correct, “all-purpose flour” is mapped to “Flour” even “vegetable oil for frying, or as needed” was correctly mapped to “Vegetable Oil.”

There is a problem however with the “confectioners' sugar, or as needed” it mapped to Sugar. This mapping is incorrect, and it occurred because in Inventory “confectioners' sugar” is called “Powdered Sugar” and the program has a purposeful bias towards shorter names, therefore “Sugar” was mapped instead of “Powdered Sugar.” To correct this problem the user simply selects the correct ingredient from the dropdown menu, in this case “Powdered Sugar.”

One additional note, the mapping program may have also made a mistake if an ingredient is not currently in inventory, in this example, if “confectioners’ sugar” or “powdered sugar” was not in inventory. To correct this the user can add an ingredient, this topic will be discussed at a later point in this document. Once the ingredient has been added, the user can simply reselect the “Import Online Meal” Button and the ingredient should map correctly.

Once the user has verified that all the ingredients have been mapped correctly they can select the “Add Meals” button in the ribbon and it will add the recipe to the database as discussed in the “Meal Input (Manual)” section.

Add Ingredient:

At times the user must add an ingredient to inventory to create a meal. The user may do this by selecting the “Add Ingredient” sheet. The user must complete the following fields to add an ingredient.

Item: is the ingredient description

Current Qty: is how much is currently in physical inventory; the program will assume that this is zero if it is left blank.

Price: Price per package

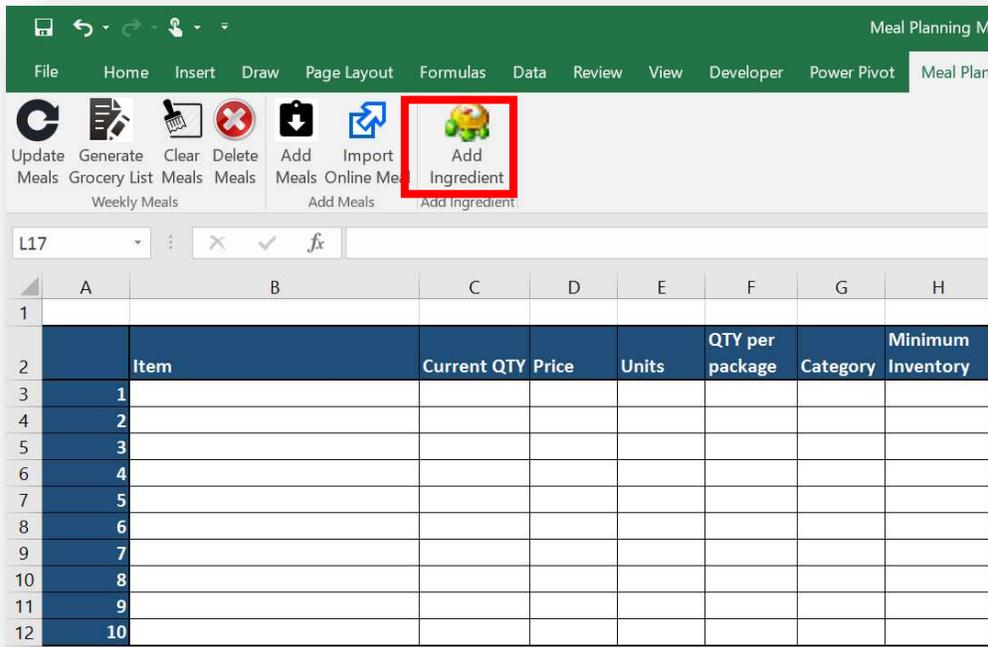
Units: this allows the user to select “Each” “Cups” or “lb”

Qty per package: specifies how many/much comes in one package

Category: the user selects a category from a dropdown list i.e. “dairy” or “fruit”

Min Inventory: the minimum required inventory, this is assumed to be zero if nothing is specified.

The user can add up to 10 ingredients at a time. To input the ingredients to the database the user selects the “Add Ingredients” from the ribbon.

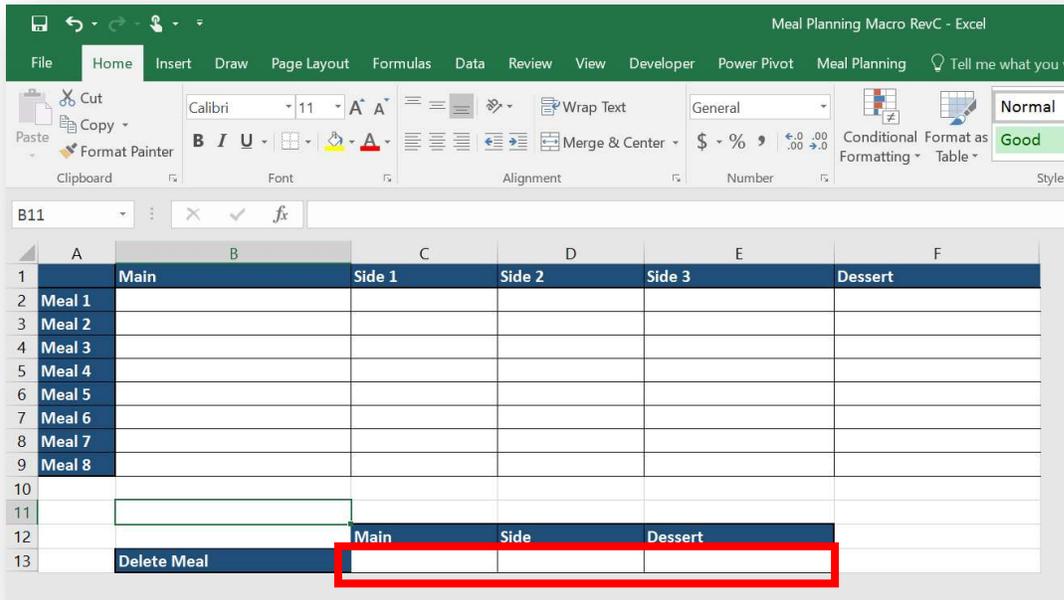


Once the “Add Ingredient” button has been selected the ingredients are added to the inventory, all calculated cells are copied and the ingredients cells are cleared (in the add ingredients sheet). At this point the new ingredients are available to be added to a recipe.

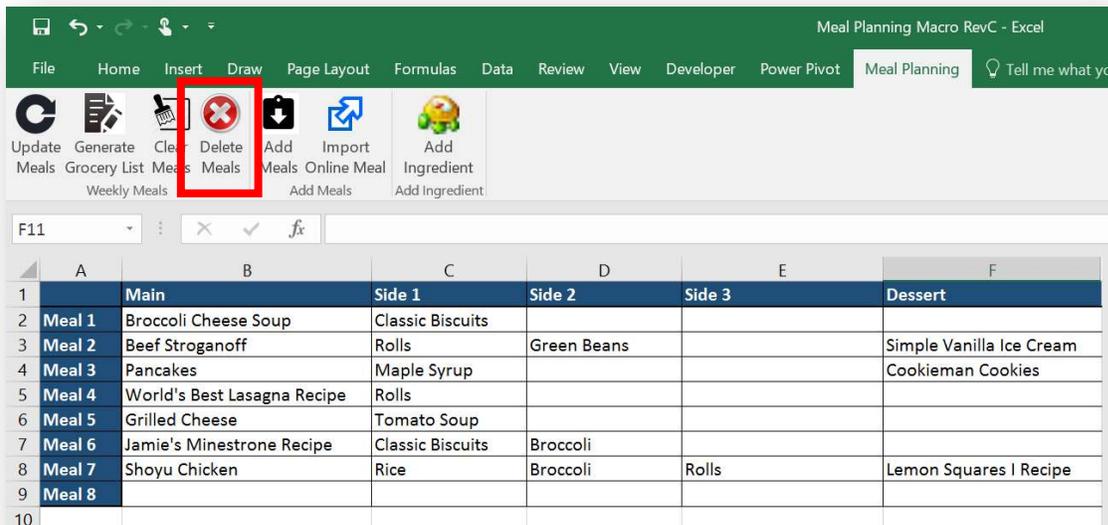
Delete Meal:

There may be times when a user decides that they want to remove a meal from the system. This can be done from the “Weekly Meals” sheet. The user simply selects the meal they wish to delete from the

fields shown below.



Next they select the “Delete Meals” button in the ribbon and it completely removes all references of the meal from the excel database. Additionally, the meal will no longer be selectable when planning for the weekly meals.



Challenges:

By far the biggest challenge was manipulating the strings pulled from “allrecipes.com.” The quantity, units and ingredient were all combined into one string. Furthermore, there was excess information in the string. Below are examples of these strings.

Example 1: “2 tablespoons confectioners' sugar, or as needed”

- This example contains extra information making it harder to map to a specific ingredient.

Example 2: “3 2/3 cups all-purpose flour”

- The quantity does not stop at a reliable place, Example 2 the quantity is “3 2/3” and for example 1 the units are “2”

Example 3: “3 eggs”

- In this example the measurement type is “Each”, but that is excluded. This just has the quantity and the ingredient.

Example 4: “salt and pepper to taste”

- This example has no units or quantity

As can be seen from the 4 examples, the strings can be quite different, and the program must be able to handle each type. In the first example, the extra information was just included in the description, so the output would be as follows, Description: “confectioners' sugar, or as needed”, Quantity: “2” and Units: “tablespoon”

To handle the difference in the quantity between example 1 and example 2 the program checked the digit after a space and if it is a number then it will include it as part of the “Quantity” and remove it from the original string.

To handle example 3, once the quantity was removed the program checks the next word against common units and abbreviated units. If there is a match then the program knows that not only did the string have a unit, but it also knows what that unit is. However, if there is no unit match, the program assumes that the unit is “Each”

To handle example 4 the program checks the first digit, if it is not a number, then the program determines that it is not actually an ingredient with a unit and a quantity, but it is actually an instruction (i.e. “cook for 3 min and salt to taste”). If the program does determine that the string is an instruction then it appends it to the end of the other instructions brought in from the website. In the case of example 4, “Salt and Pepper to taste” would be the last instruction.

With these procedures in place the program is able to correctly extract the quantity, units and the ingredients from the website. This however is only half the battle, the program must have a mapping routine to convert the website ingredients to the inventory ingredients. To do this the program splits the description from the website into the discrete words. For example “vegetable oil for frying, or as needed” would be split into the following words “vegetable” “oil” “frying” “needed” (note that common words like “for”, “as”, “or”, “a” were ignored/removed). Each ingredient was given a score based on how many words it had in common. For example, “Vegetable Oil” would be given a score of 2 because it matches two words, while “Olive Oil” would be given a score of 1 because only one word matches. The highest score was mapped to the online description, in this case “Vegetable Oil”.

Sometimes a tie occurs, for example if the description is “white sugar” then both “Brown Sugar” and “Sugar” would have the same score of 1. In the case of a tie the ingredient that is the shortest would be mapped to the online description, in this case “Sugar” would be correctly mapped to “white sugar”.

This routine took some trial and error to get right, and while it is not perfect it seems to be fairly robust. It seems to fail most often if the ingredient does not exist in the database (which should be expected) or if the online description is quite different from the inventory description (as seen with “confectioners’ sugar” vs “powdered sugar”).

Future Expansion:

I use this program at least once a week, so I have several ideas to make it even better. Below is a short description of each idea. None of these are too complex, it will just take some additional time to complete.

Recipe Loading Tool:

This tool would allow the user to select a recipe from the “Weekly Meals” sheet and push a button to automatically launch the web browser to the website with the recipe.

Enhanced User Interface:

I think that this tool would benefit from interlinking sets of custom user forms. Ideally, the user would strictly interface with the user form, not with the actual cells.

Assistance:

I received no major help on this project. The extent of help I received was googling functions and objects to understand how they work.