



RECIPE GATHERER

Todd Valentine

ABSTRACT

The Recipe Gatherer excel spreadsheet file digitally organizes recipes allowing the user to import recipes from popular recipe websites as well as importing from a manual input.

Executive Summary

Business Description

My wife enjoys trying new recipes from the internet. When she finds a recipe, she prints it and places it in a massive binder for later use. It is challenging to find old recipes and even remember recipes that she once liked. Lastly, when she picks a recipe to try she must manually input the ingredients into her grocery list app on her phone. The business case for this project is that the Recipe Gatherer spreadsheet will allow my wife to spend more time taking care of our family business, i.e. our children, and less time dealing with recipe organization.

System Built

I built a spreadsheet entitled “Recipe Gatherer”, which digitally organizes my wife’s recipes. The spreadsheet has 7 main points of functionality. The “Recipe Gatherer” will import recipes from my wife’s 3 main recipe websites: allrecipes.com, damndelicious.net, and melskitchencafe.com. The sheet will then display the recipe on the main recipe page. Other points of functionality include manually inputting a recipe, saving edits to a recipe, deleting a recipe, printing a recipe, emailing a recipe, and inputting the ingredients into the AnyList app. Of course, the spreadsheet will display any stored recipes.

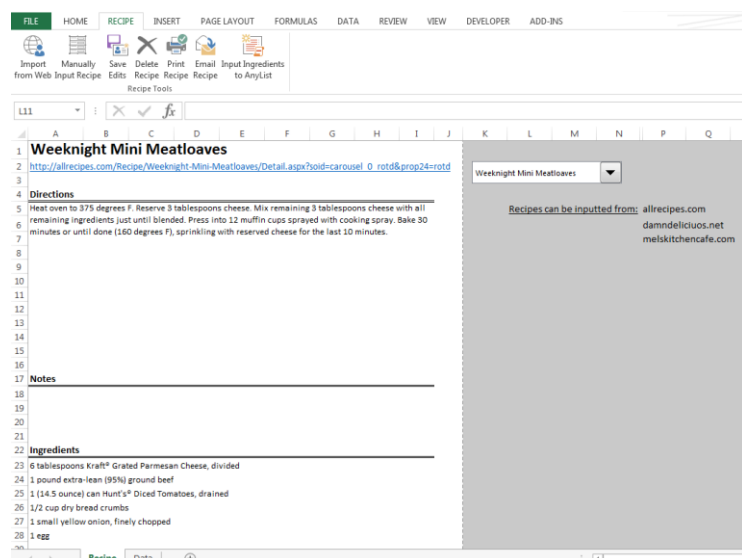
Implementation

The implementation will be discussed according to each of the functionalities of the project beginning with the purpose.

1. Purpose & Main Recipe Tab

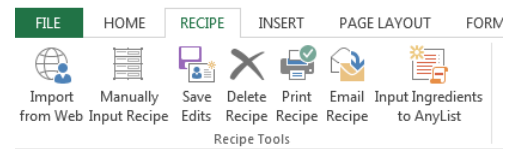
The project, “Recipe Gatherer”, digitally captures recipes from popular recipe websites and from manual input and then places them in one location. The main tab, “Recipe”, displays the recipes for viewing, editing, printing, and emailing. From this main page, the user may:

1. Import a recipe from the web
2. Manually input a recipe
3. Save edits made to the recipe on the main page
4. Delete the currently displayed website
5. Print the currently displayed recipe via the default printer
6. Email the recipe
7. Input the ingredients to the AnyList app via Internet Explorer



Each of these functionalities are performed by clicking on the corresponding ribbon buttons as shown below.

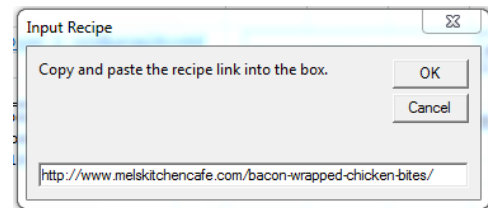
The user may select from the database of recipes through the combo box. The combo box pulls the recipe names from a table found in the “Data” tab. The table in the “Data” tab should not be edited by the user. When a new recipe is selected in the combo box, the selected recipe will then display the recipe name, web address, cooking directions, notes, and ingredient list.



2. Import from Web

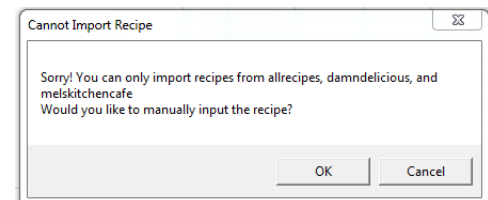
When a user clicks “Import from Web” from the Recipe ribbon a message box will appear with a text box to enter a web address for a recipe to be imported into the spreadsheet. There are three main recipe websites that are regularly used by the target audience. These websites are

- www.allrecipes.com
- www.damndelicious.net
- www.melskitchencafe.com



When a recipe is inputted from one of the three websites, an Internet Explorer window will appear showing the recipe to be imported. From this website, the program will grab the recipe name, URL, cooking instructions, and ingredients and place this information into the “Recipe” tab as well as store this information for later use. The Internet Explorer window will then close. The new recipe should be displayed on the active screen and the recipe name placed in the combo box.

If a user inputs a website not from these sites then a message box will appear stating that the recipe cannot be imported and would the user like to manually input the recipe (see section 3 to learn the functionality of manually inputting a recipe).



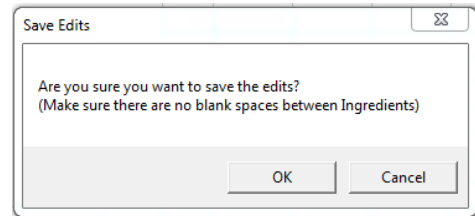
3. Manually Input Recipe

Manually inputting a recipe occurs by clicking the “Manually Input Recipe” ribbon button. This will bring up a window that will allow the user to input a recipe name, directions, notes, and the ingredients. The newly inputted recipe is saved for later use as well as displayed in the “Recipe” tab. The user must input a recipe name and ingredients for the form to save. A message box warns the user if the “Recipe name” field or all of the “Ingredients” fields are left blank.

The image shows a form titled 'Manually Input Recipe'. It has several input fields: 'Recipe Name' at the top, 'Directions' in the middle left, and 'Notes' at the bottom left. To the right of these is a large grid for 'Ingredients' with 22 numbered rows. On the far right, there are 'Save' and 'Cancel' buttons.

4. Save Edits

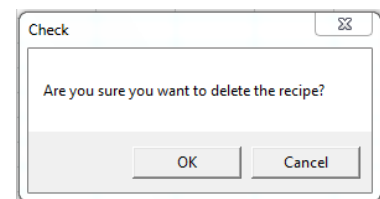
The user may edit any of the information displayed in the “Recipe” tab then save the changes by clicking on the “Save Edits” ribbon button. A message box will appear asking the user if s/he truly wants to save any edits made. When the “OK” button is clicked the newly edited recipe information will be saved to the “Data” tab.



There exists one slight nuance of the save edits feature. In order to properly store the ingredients there should not be any blank rows between the listed ingredients. The user is reminded of this in the message box.

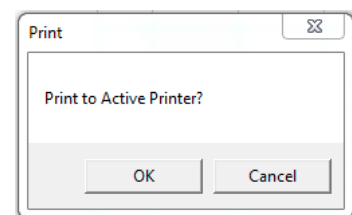
5. Delete Recipe

A user may delete the displayed recipe by clicking the “Delete Recipe” ribbon button. The functionality is fairly simple in that the currently displayed recipe will be removed from the stored recipes. A message box will appear when the “Delete Recipe” ribbon button is clicked ensuring that the user wants to delete the recipe. When the “Ok” button is selected then the recipe is deleted and the next recipe in the combo box is displayed on the screen.



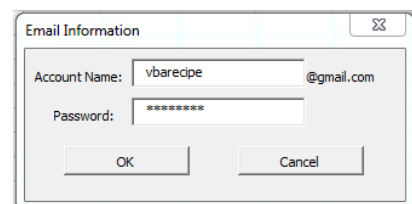
6. Print Recipe

The recipe can be printed to the default printer by clicking on the “Print Recipe” ribbon button. Once clicked, a message box will be displayed asking to proceed. The displayed recipe will then be printed by the default printer. The layout of the recipe on the “Recipe” tab is formatted for 1 page printing (only the recipe will be printed, not the combo box or any instructional text). The page is also formatted to be black & white printed since the target user typically cannot afford colored printing.



7. Email Recipe

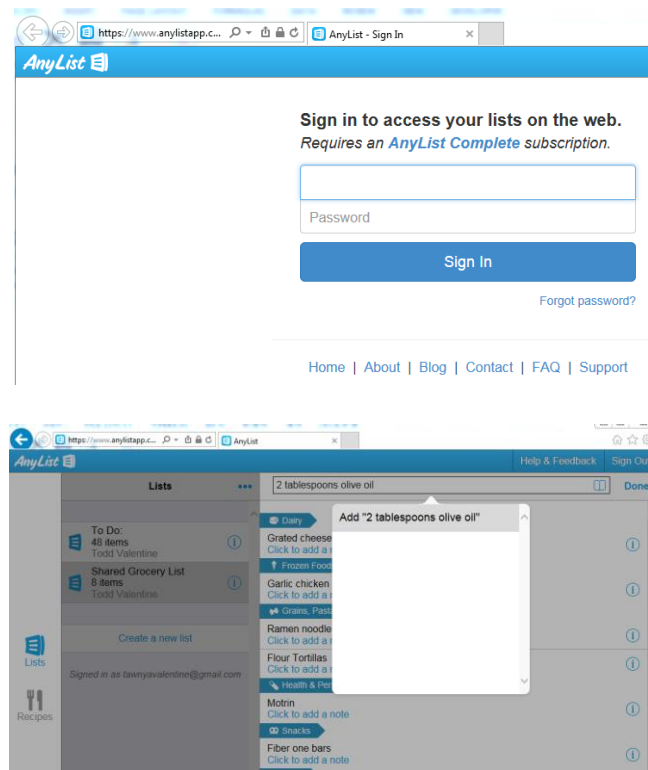
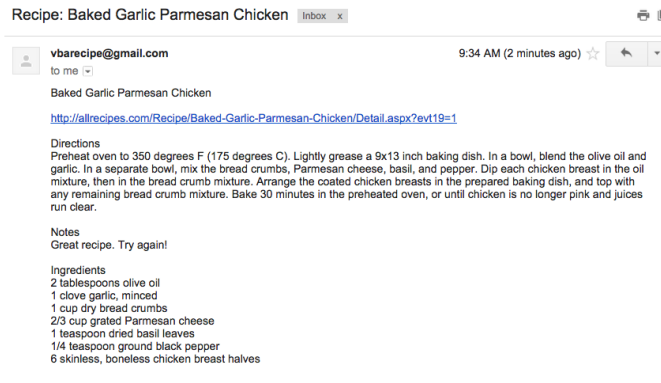
The user can email the displayed recipe to themselves by clicking on the “Email Recipe” ribbon button. A window will appear asking for the account name and password. The email feature can only email gmail accounts. If the message is sent, then a message box appears stating the message has been sent. If the message fails to send, a different message box appears stating the message did not send. A screen capture of an emailed recipe is shown below.



The user should be cautious of emailing a recipe to a gmail account. If the user has high security then the email will fail. If tried too many times, google will send a warning email to the user stating that an unauthorized user is trying to access the account.

8. Input Ingredients to AnyList

The final functionality allows the user to pull up the AnyList website from an Internet Explorer window by clicking on the “Input Ingredients to AnyList” ribbon button. AnyList is a mobile app that stores grocery lists that can be shared with multiple devices. By clicking the ribbon button, a window will appear displaying the login information for the web version of AnyList. The user will then login and input the needed ingredients into the website. The ingredients can be individually copied and pasted into the AnyList website. This will allow the user to selectively choose which ingredients to include on the list. This functionality is not automated so the user will need to navigate through the AnyList site. It should be noted that in order to input ingredients into a web version of AnyList, the user must have a premium account, which was purchased for this project at \$7.99/year.



Learnings and Conceptual Difficulties

The project was a great learning experience. I was able to use the material we covered in class including website parsing, forms, and arrays. This project greatly reinforced the principles and concepts we learned in class and on the assignments. Below are a few of my greatest learnings and greatest difficulties:

- Controlling Internet Explorer: My VBA code used the agent module to parse through recipe websites. I had different sub procedures to handle each website. I was able to really dig deep into the abilities of the agent. For example, I learned that the agent needs an element name in order for the element to perform an action (click, submit, activate). A struggle I had early on in this process is that as I moved the position it wasn't pulling the right information. I had been looking at the html from Google Chrome and decided to check the html on Internet Explorer. It

was ordered differently, which was giving me issues in my code. I ended up using Internet Explorer to view the html code. I also researched a significant amount on how to control Internet Explorer without the agent. I wanted to control the AnyList website from VBA but the agent wasn't giving me the control I needed. None of the elements on the AnyList website had names. They were identified by "classes" and something called "react-id". In my code, I created an object for Internet Explorer and started manipulating Internet Explorer. I tried a number of different functions that I found on the internet including `getElementsbyTagName` and `getElementsbyClassName`, which I then would cycle through the elements to find the appropriate element. In the end I couldn't get the elements to function how I needed them to.

- Combo box: We had discussed in class how to work with combo boxes in a form but never in a worksheet. I found on the internet how to identify the combo box in VBA, which was done by `"Sheet1.Shapes.Range(Array("Drop Down 6"))"`. I ended up recording a macro to determine how to populate the combo box. I learned that a combo box will use an identifying number to know what was selected. I grabbed this number and put it in a hidden cell on the sheet. I would then manipulate that number in VBA to change the combo box if needed. Lastly, I wrote 2 subroutines to work with the combo box. I wrote a routine that would populate the combo box and a routine that would change the selection in the combo box (the user can change the combo box without a subroutine but some of the added functionality from the buttons required a change in the selection).
- Ingredients Array: I had to truly think through how the user would work in the spreadsheet in order to avoid any arrays that weren't dimensioned properly. I had to add a few message boxes to prevent this as well as many if statements in the code to exit early if that would happen. Lastly, pulling the ingredients from the "Recipe" tab back to "Data" tab with the "Save Edits" button, I ended up telling the user that there could be no blank spaces between ingredients. The blank space won't crash the program, but it won't save any ingredients below the space. I determined that this was something I could ask the user rather than dealing with it in VBA.
- Email: I had missed the class for a job interview on the day we reviewed the email material. I really struggled through this. It appeared that the code was working correctly but it was not sending. At one point, I received an email from google stating that someone was trying to access my account and that I should change my email password immediately. I went and received help from the TA. The problem ended up being the security on my gmail account was set too high. To deal with this I created a new gmail account called "vbarecipe" and password "vba12345" to make sure the email functionality works.
- Message Boxes: A key take-away from this assignment is how valuable message boxes can be to a program. Once my project was completed, I spent about 2 hours attempting to break the program. I found that most of the errors could be resolved through the use of message boxes and if statements (in VBA).

I achieved the main scope of my project. There was only one piece of functionality that I wanted to develop further but did not. I had intended to have all of the recipe ingredients automatically imported

into AnyList. As I discussed above, I could not control the website like I needed to. As I thought about this functionality, I determined that the user probably wouldn't need to purchase all of the ingredients at the store, which would defeat the purpose of importing all of the ingredients. Instead, I decided to simply pull up the website, and then let the user copy and paste the required ingredients. I honestly felt that this served the purpose better than automatically importing all of the ingredients.

Assistance

I received assistance for one portion of my project:

- I could not get the email functionality to work. I attended TA office hours where the TA and another student looked through my code and determined it was the security on my gmail account rather than a coding issue.

Outside of this, I used old homework assignments, projects, and in-class examples for the bulk of my code. When I ran into an issue, I would search google and view forum discussions, blog posts, and Microsoft help. I never posted to any forum but only used existing discussions.