Task 0 : Explain what you are doing/ going to accomplish

+stock page, set up the contents with their stock and add an add stock insert tab that allows the user to stock up as much as they want

Task 1: Sketch interface design

*Draft a rough design for the interface that allows the user to trigger functionality in task 1, while also annotating where the information in task 2 will be displayed. Create another sketch listing the interface widgets used to create the interface.*



Task 2: Identify any classes required

*Explain what the class will represent, plus listing what information will be stored in the class and any functions the class will have.*

N/A

Task 3: Identify information to be displayed

*What information will the interface need to display to the user?*

The +stock tab on the nav\_bar, a list of the food items with the cost, the stock and how many have been sold. It will also have a add button where the user can add to the stock of the food item, when clicked it will take them to another success page that will display a success message and a back button, that when clicked will take them back to the +stock page.

Task 4: Identify user inputs

*What program functions can the user trigger through the interface?*

The +stock tab on the nav\_bar

The add button

The back button on the success page

Task 5: Identify any constants or existing data if required

Food.id

Task 6: Identify indexed data structures

N/A

Task 7: Determine what calculations are necessary

*Write out the calculations the program will have to compute.*

Stock = stock + 1

Task 8: Develop a modular structure for your program

*Describe any functions that the computer program will have, identifying any sub-functions where required.*

Plus\_stock – this is the page where you can add stock to the food items

Add\_success – This will add one value to the stock of a food item

Task 9: Define the functions identified

*Describe the functions for both the main program and any classes in terms of input and/or output where required. You may choose to do this with flow charts or pseudo-code (not Python code!). Add in additional steps or explanations using sequential, conditional, iterative statements where required. Identify global and/or local variables.*

AT ROUTE (‘/plus\_stock’)

AT VIEW (‘plus\_stock’)

PROGRAM plus\_stock

SET data TO dict (contents\_list EQUALS contents)

RETURN data

Task 10: Address any relevant implications such as usability, functionality, legal/ethical requirements.

Task 11: Document test cases for testing the program

*Document any testing that can be used to test your program. If any input is inputted using the keyboard, describe the expected input, plus any exceptional, boundary or invalid cases.*

Task 12: Refine the plan

*Note any modifications here when iterating through the development cycles.*

Task 13: Document testing

*Show screenshots of your program working with descriptions of each image. These images should test the tests cases listed above.*

Task 14 : Evaluation

*How did your version turn out*