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

Code the code so that when you press the buy button the stock will go down with a message that gets sent to the user saying it was successful or not, also it will print out of stock and disable the order button if the stock runs out.

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?*

An order button that when clicked will send a message to the user say it was successful, also if stock runs out it cancels the order button and display out of stock,

Task 4: Identify user inputs

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

The order button, the order tab, and the back button on the order 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.*

found\_food.food\_stock = found\_food.food\_stock - 1

found\_food.food\_sold = found\_food.food\_sold + 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.*

Order\_success – takes the user to a page displaying a success message that will lower the stock of the item by one and add one value to the sold message.

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.*

PYTHON:

AT ROUTE ‘/order\_success/ <food.id>

AT VIEW ‘order\_success’

PROGRAM order\_success (food.id)

SET Food.id TO int(food.id)

SET found\_food TO NONE

FOR food IN contents

IF food.id EQUALS food.id

SET found\_food TO food

SET data TO dict(food EQUALS found\_food)

SET found\_food.food\_stock TO found\_food.food\_stock – 1

SET found\_food.food\_sold TO found\_food.food\_sold + 1

RETURN data

CSS (python):

% FOR canteen\_content IN contents\_list

% IF canteen\_content.food\_stock DOES NOT EQUAL ZERO or canteen\_content.food\_stock IS LESSER THAN ZERO

%ELSE

%END

%END

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

If the user wants to order a certain food they need to click the order button. In the nav bar I have used white text on a black bar, this make it very easy to notice, it is also pleasant to the eyes. When they click the order button it will take them to a page where it clearly prints out success, and there will be a big cyan button they can click to get back.

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.*

Making the order button work, i.e. when clicked the stock will go down and sold will go up, also if stock runs out, the user will not be able to order anymore until restocked.

Task 12: Refine the plan

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

I have changed a few lines of code to make it a little more robust. (found\_food.food\_stock = found\_food.food\_stock – 1) to (found\_food.food\_stock - = )

Task 13: Document testing

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

I will attempt to order food one at a time, I expect this to be a success. I will do this until the food stock runs out. This should cancel the order button and it should print out of stock. After this I will get one of my classmates to test this as well, to make sure they have the same success as I will.





Task 14 : Evaluation

*How did your version turn out*

This turned out very well, alongside my testing. I had a classmate test my website as well and they did the steps very easily. I am happy the way it has turned out, and am ready for version 4.0