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 and there will be a cart price that will go up, with a message that gets sent to the user saying it was successful or not

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.*

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

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

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.*

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*