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

This will be my purchase page which will reduce the stock of each food item.

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.*

Task 3: Identify information to be displayed

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

Whether there is any stock available. Once purchased will show whether the purchase was successful and a button to return to the purchase page.

Task 4: Identify user inputs

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

I will have a button which allows people to purchase items of food one at a time.

Task 5: Identify any constants or existing data if required

Task 6: Identify indexed data structures

Task 7: Determine what calculations are necessary

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

Minus one from comic stock every time they purchase something.

Task 8: Develop a modular structure for your program

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

Purchase page contains code which takes one off the stock whenever something is purchased.

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.*

ROUTE purchase\_page PASS item\_id

VIEW purchase\_page

PROGRAM purchase\_page PASS item\_id

item\_id = int(item\_id)

found\_item = None

FOR item IN items

IF item.id IS item\_id

SET item TO found\_item

SET data TO Dictionary (SET item TO found\_item)

Take 1 away from found\_comic.stock

RETURN data

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

My webpage has to be functional and usable as well as looking good. I have to make it obvious where my purchase button is. I also have to show how much stock is left and to show the price of each of my items. I also have to make it function properly with no issues.

I made my purchase and purchase success pages colour scheme the same as other school websites. I made each item be in different cards which contain information on the item. Each of these cards contain the item name, a picture of it and how much stock there is of it. In each of these cards is also a button which can be used to purchase one of that item. This button stands out so it is easy for the user to see how to buy items. This button links to a page which takes 1 off the stock of that item. It also serves the purpose of showing the user that their transaction was successful.

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.*

If I click the button it links to the page and minuses 1 off stock. If I try buying an item when the stock is 0, it does not work as I have it so that the button becomes inactive when stock is 0. If I try spamming refresh to take multiple items off it works. I will solve this in a later version as it will become a form so refreshing multiple times does not work.

Task 12: Refine the plan

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

Changed my product page code in html to include a button that the user can press to purchase each item.

Added <p><br style = "line-height: 43 ;"></p> to my html code to break multiple lines and extend my div colour lower down my page.

Task 13: Document testing

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



One of the cards containing information on my items.

Task 14: Evaluation

*How did your version turn out?*

My version 4.0 turned out perfectly. I successfully allowed the user to purchase stock of each of my items. This reduces the stock of each item to a minimum of 0. There were no major issues in my update. I was not happy with the way my page looked so I took a while to fix the design. My next step is to be able to restock my items stock.