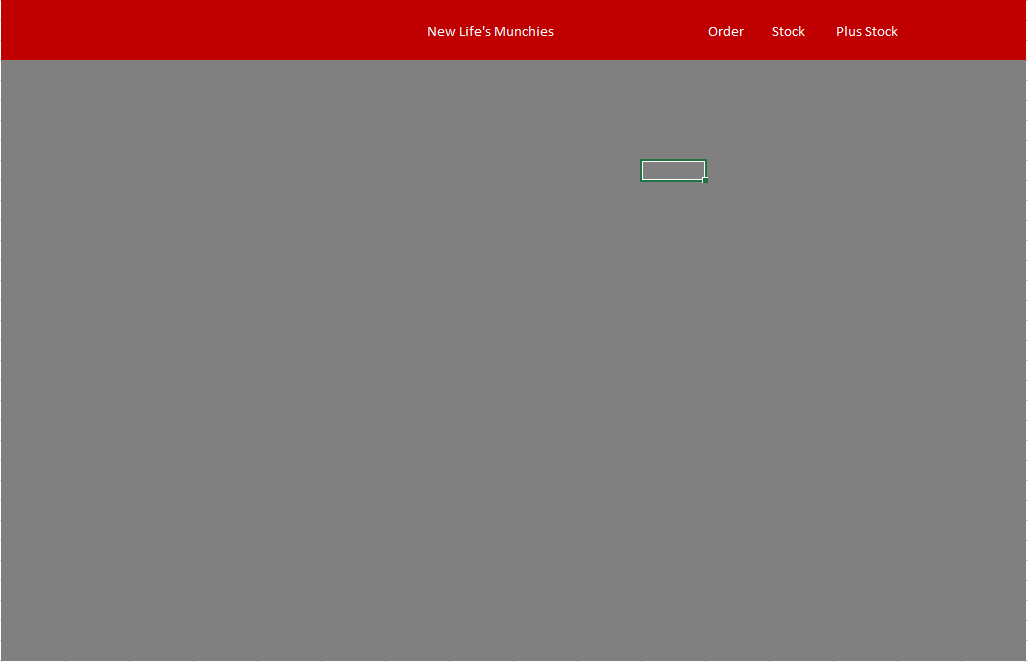
**Appendix 1: Planning Guide**

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

Create a Python server that also holds values for the different types of contents of food, and the index page with the header and navigation bar.

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.*

Canteen\_food:

Food\_name

Food\_image

Food\_stock

Food\_sold

Task 3: Identify information to be displayed

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

%include('shared/header.html')

%include('shared/nav\_bar.html')

Task 4: Identify user inputs

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

N/A

Task 5: Identify any constants or existing data if required

* Sushi Roll pack - Starting with 5, costs $3.50
* Hot dog and Chips - starting with 12, costs $4
* Ham and Cheese Sandwiches - 4 in stock, costs $8

Task 6: Identify indexed data structures

SET LIST contents to

Set canteen\_content to sushi\_rolls image 5

Set canteen\_content to hotdog\_chips image 12

Set canteen\_content to ham\_sammy image 4

Task 7: Determine what calculations are necessary

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

N/A

Task 8: Develop a modular structure for your program

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

Canteen\_content

Index

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.*

IMPORT run, route, view, get, post, request FROM BOTTLE

IMPORT count FROM INTERTOOLS

SET \_ids TO count 0

PROGRAM\_IDS\_recieves self, name, image, stock, sold

Self.id is set to next self.\_ids

Self.food\_name is set to name

Self.food\_image is set to image

Self.food\_cost is set to stock

Self.food\_sold is set to sold

END

SET LIST contents to

Set canteen\_content to sushi\_rolls image 5, 0

Set canteen\_content to hotdog\_chips image 12, 0

Set canteen\_content to ham\_sammy image 4, 0

@route(”/”)

@view (“index”)

PROGRAM index

PASS

run(host='0.0.0.0', port=8080, reloader = True, debug = True)

END

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

I have chosen the colours red and grey as they are part of the colours of the school (however these will probably change in later versions). I have put certain tab in order to make it easier navigation for the user. I.e order tab first, stock tab second, and a restock page last.

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.*

There is no user interface at the moment so just testing to see if the python code works

Task 12: Refine the plan

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

Needed to add the zero value (as you see in the pictures.

Task 13: Document testing

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

Basically I almost got it right the first time, however I forgot to add the sold values in the list. But worked after I added them.

Task 14 : Evaluation

*How did your version turn out*

This version turned out beautifully, did what it was supposed to, I only made one mistake that I fixed and am ready to move on to my second version.

