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

Create a order page that will contain the food set out in orderly fashion (hopefully with a picture), Add values to the contents of the ordering page (stock, cost)

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

Food\_cost

Task 3: Identify information to be displayed

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

Order button, Cost of item, picture, and stock.

%include('shared/header.html')

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

Task 4: Identify user inputs

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

Order button

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, 0, 5

Set canteen\_content to hotdog\_chips image 12, 0, 3

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

Index – front page

Order – the ordering page

Saved\_pictures – allows me to take pictures from a folder

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("/order")

@view("order")

def order():

data = dict (contents\_list = contents)

return data

@route('/picture/<filename>')

def saved\_picture (filename):

return static\_file(filename, root='./images')

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

User has to click the order tab to get to the order page.

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.*

The only thing that will work at the moment is clicking the order tab to get to the order page.

Task 12: Refine the plan

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

N/A

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*

I think this version turned out great, I’ve completed what I set out to do. There were no problems with my code and I am ready for the next step.

