Ver1.0 Planning document – school canteen

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

I am going to create a website that manages a school canteen, it will manage the stock of items on the menu and students will be able to order food. I will create a program that stores the students name and what they ordered in an array. The inputs will be processed with a python bottle program and displayed on an html website.

**This version:** just plan out the project and create a homepage with info and a navbar; a foundation for the project.

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.*

View these png images in the planning folder:

* Homepage.png

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.*

There will be a class that holds the Items on the menu and the initial stock levels / price of that item, called FoodItem

Task 3: Identify information to be displayed

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

There will be:

* A homepage with info about what the school canteen is, general info like open times and location.
* A page that displays food item, stock level, price, food items sold.
* A page that takes input from the operator about the students order, input boxes.

These are the first pages I have planned I will add additional pages below if they are needed

Task 4: Identify user inputs

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

No inputs in Ver1.0

Task 5: Identify any constants or existing data if required

No constants in Ver1.0

Task 6: Identify indexed data structures

There will be an index data structure called food, will contain name, stock level, price, amount sold of each fooditem.

Task 7: Determine what calculations are necessary

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

No calculations in Ver1.0

Task 8: Develop a modular structure for your program

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

This Ver1.0 will only display the homepage: index.html, I will also write up the class **FoodItem** and the index data structure **food** that won’t be used for Ver1.0

From bottle import these things run, route, view, get, post and request

From Intertools only import count

Define a class called FoodItem:

Set \_ids to count (zero)

Define \_\_init\_\_(passing parameters: self, food\_item, stock, price and sold):

Set self.food\_item to food\_item

Set self.stock to stock

Set self.price to price

Set self.sold to sold

Define index data structure food: [

Set FoodItem()

Set FoodItem()

Set FoodItem()

]

In this route(‘/) view the page named “index”

Define a function called index():

Pass as no functions are needed

Run(set the host to ‘0.0.0.0’, the port to ‘8080’,the reloader to true and debug to true)

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.*

Index:

One function will be created but will have no use for now, and will likely have no use in the future. It is called index, no proceses should be made on the index/homepage, this is only a place to display general information.

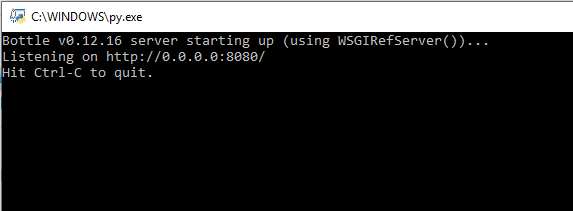
Task 10: Address any relevant implications such as usability, functionality, legal/ethical requirements.

When designing my website, I will take into account the implication of usability and functionality, meaning buttons will be ladled, a colour scheme that is easy to look at, the layout will make sense and be very basic, a school canteen website does not need to be complex. The website should not breach and legal and ethical requirement, it will follow copy right laws, won’t be offensive etc..

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.*

When I run the python script “School-canteen.py” --- a console pops up as expected, acts how it should, no errors server starts up:



When I go into chrome and go to localhost:8080 --- the website shows “index.html”, navbar.html and head.html work correctly as the navigation bar shows, colours are correct, running as expected.

Task 12: Refine the plan

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

Original sketch differed from the final homepage(index.html), just the colour scheme, changed to be more like our schools’ colour scheme.

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?*

At this stage in the development it was very smooth, This is the base of my whole project so it is required to be basic, a lot more will be added in later versions, overall was a success.