**Planning Document**

**Appendix 1: Planning Guide**

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

Create the python server in wing, add test data and check it runs in python.

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.*

N/A

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.*

Comic\_book

comic\_name

comic\_price

comic\_stock

comic\_image

Task 3: Identify information to be displayed

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

N/A

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

Existing test data

Super Dude - Starting with 8 in stock

Lizard Man - Starting with 12 in stock

Water Woman - Starting with 3 in stock

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.*

When a comic is sold it will have to minus one off comic\_stock

Add one to amount sold when a book is sold.

Task 8: Develop a modular structure for your program

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

\_\_init\_\_ function, will initialise the creation of the objects for that class.

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 route, view, run, get, post, request from bottle

Import count from intertools

CLASS comic\_book

Set \_ids to count(0)

Define initialise function – bring comic\_name, comic\_price, comic\_stock and self and comic\_image

Sets self.id to next(self.\_ids)

Sets self.comic\_name to name

Sets self.comic\_price to price

Sets self.comic\_stock to stock

Sets self.comic\_image to image

END

Set list comic\_test to

Set Comic\_book to Super Dude, image, 8

Set Comic\_book to lizard man, image, 12

Set comic\_book to water woman, image, 3

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

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

Add comments to python code so it is easier for others to read.

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.*

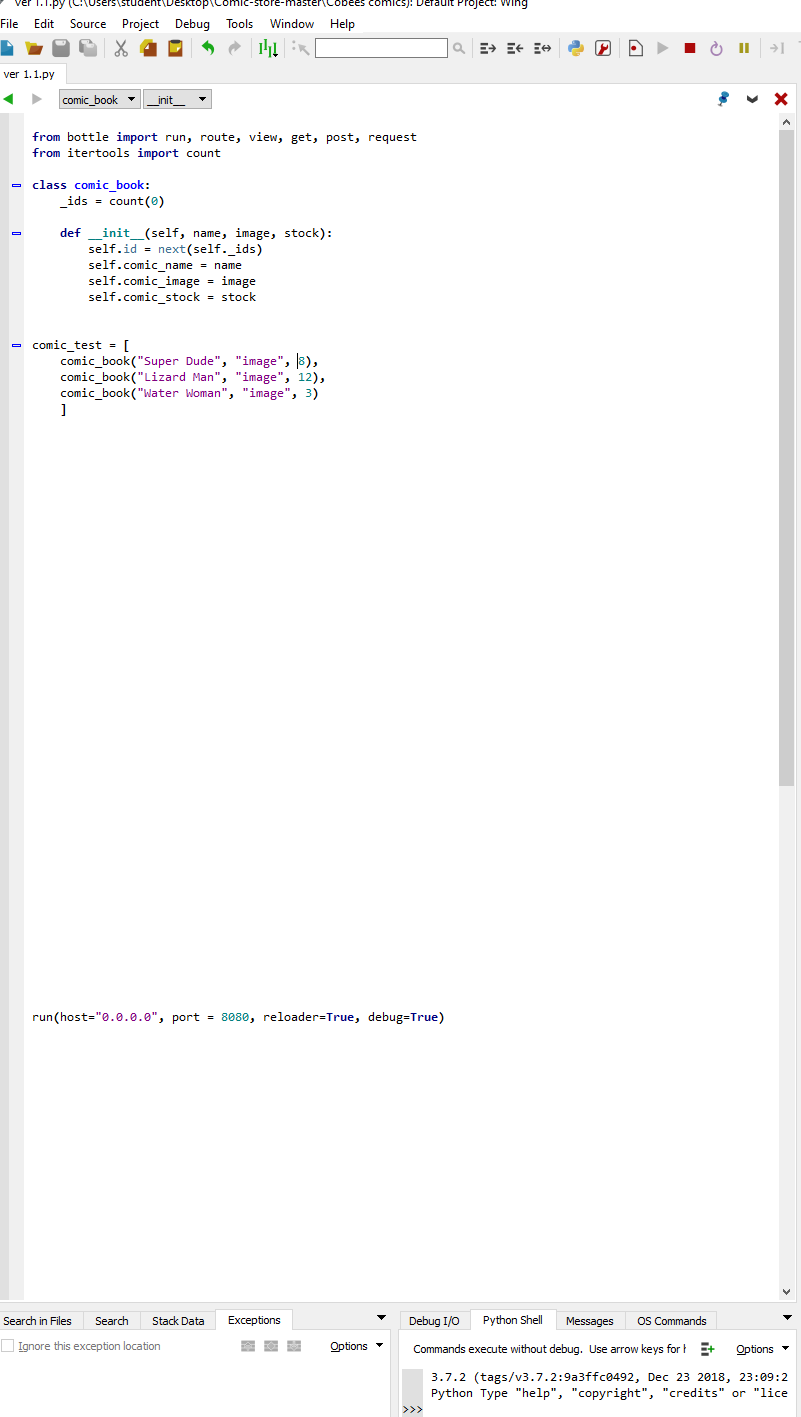
**Test python server works**

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.*



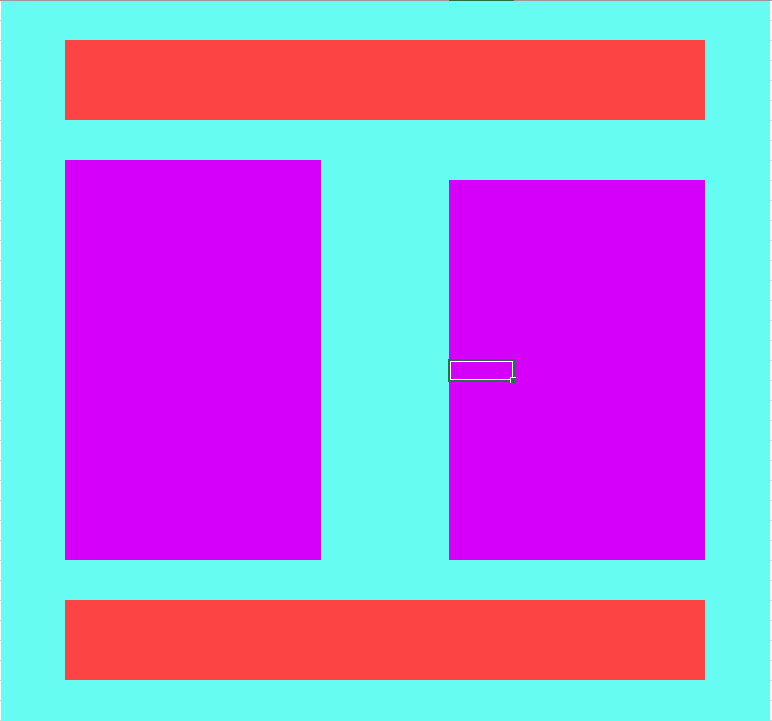
**Ver 1.2**

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

Create the index page, nav bar and header HTML pages

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?*

Task 4: Identify user inputs

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

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.*

Task 8: Develop a modular structure for your program

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

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.*

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

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.*

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.*