

**Appendix 1: Planning Guide**

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

Develop a comic book store program that will be able to sell a comic one at a time, reducing the stock by one.

The interface should notify the user if the comic has been sold successfully.

The interface should notify the user with an error message if the comic has not been sold if there is not enough stock.

The interface should display:

The number of comics sold.

The current stock levels of all comics (at once). If the stock levels change at any point, the interface should update.

The user should be able to restock a chosen comic.

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\_sold

Task 3: Identify information to be displayed

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

The amount of comics, the price, the name, whether they are sold or out of stock.

Task 4: Identify user inputs

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

Restocking button, selling button, back button.

Task 5: Identify any constants or existing data if required

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.*

Plus and minus.

Task 8: Develop a modular structure for your program

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

Main

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

PROGRAM\_IDS\_recievesself.name, image, amount

Self.ids is set to next self\_ids

Self.comic\_name is set to name

Self.comic\_image is set to image

Self.comic\_amount is set to amount

END

SET LIST test\_comics 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)

END

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

Usability.

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.*

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.*

