|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **What am I testing?** | **Test Type** | **Expected Result** | **Actual Result** | **How was it fixed?** |
| 19/03 | The display all function | valid | Displays all aspects within the “tasks” dictionary laid out aesthetically in a msgbox. |  | I will change ‘task:’ to ‘Task Number:’. This should be more visually appealing and more descriptive. It functions correctly otherwise, so no more changes are needed. |
| 19/03 | Inputs for title and description for a newly added task, and the automatically assigned task id | valid | The task id, title and description should all display in a msgbox |  | No changes needed |
| 19/03 | Assigning a team member to a newly created task | valid | Buttonbox with 4 options, JLO, JSM, BDI, and None, as automatically added to a list and then options |  | No changes needed |
| 20/03 | Below the lower bound of the input for priority scale | invalid | Priority of 4 is not accepted |  | No changes needed |
| 20/03 | Above the upper bound of the input for priority scale | invalid | Priority of 0 is not accepted |  | No changes needed |
| 20/03 | Lower bound of the input for priority scale (1) | boundary | Priority is accepted as 1 |  | No changes needed |
| 20/03 | Upper bound of the input for priority scale (3) | boundary | Priority is accepted as 3 |  | No changes needed |
| 22/03 | Updating an aspect of an existing task | valid | When I select a title of a task from a Buttonbox, the task id number for the title is found and stored, so I can edit other aspects within that dictionary | My valid input of ‘task’ is not found within any of my sub dictionaries (1,2,3,4,5) in my nested dictionary of ‘tasks’. | the following changes were made:    Previously my code was searching for a key within each sub dictionary, when I was expecting it to be searching for the information associated with the key. To fix this, I had to specify the key name “Title” for each sub dictionary search. |
| 25/03 | Confirming a change for the update function | valid | Displays all changing information to the user to confirm that an error has not occurred. |  | No changes needed |
| 26/03 | Search assignee function | valid | Tasks which are assigned to a user input member (BDI), and that have not been marked as completed will be displayed only |  | I will add an ‘and’ statement to my ‘if’ statement to exclude tasks with their status as ‘completed’.  and tasks[task\_id]["Status"] != "Completed": |
| 28/03 | Search by title function | valid | Program finds which task is selected based on an input of the title from a Buttonbox, and transfers its information to a new dictionary to display on its own |  | my temporary dictionary variable was not originally a dictionary but was accidently made as a list. This meant that when I was trying to establish task\_id as the name of the sub dictionary the program thought I was trying to call for an entry in the list, which was out of range as the list is empty. This was fixed by changing square brackets to curly brackets when defining temp\_dict. |
| 4/04 | Report generation function | Valid | Displays the number of tasks at every stage of development |  | Originally this code was on one line, where all the aspects for the output were added simultaneously, after all had been derived.  This however is beyond the line length of 79 characters, so must be split across multiple lines. Therefore, I will add each aspect of the report to the final output individually, across 5 lines of code. This will not affect the display of the outcome even if they are all indented to the same level in the code, as I will write five different output statements in a block, to keep the code clean and tidy. |
| 10/04 | Add task function, click the “X” button on the button boxes, or clicks “enter” or “ok” with no input | invalid | Not let the user continue, tell them that an input is required, and let them try again | Accepts the input as “None” and lets the user move on | I will add a validation system to every instance where a box can be closed out of to ensure a valid input is received. This will be a while loop, that checks whether the user input is “” (blank) or None (due to clicking cancel or the X to minimise). If yes, the user is told an input is required and the question is asked again. If the input is not blank or None, the loop breaks allowing the user to continue. |
| 11/04 | Add and update functions, confirmation of the users action | valid | Once the user creates the new task or successfully updates the task, a msgbox should inform the user that a change has occurred, or that the user has selected cancel | When confirm is selected:  When cancel is selected: | No changes needed |

**Test Types:**

**Valid:** Is a test which the code accepts and should move on as expected. E.g a name input

**Invalid:** Is a test which the code shouldn’t accept and could crash the code if there is unacceptable input validation

**Boundary:** A test on the limit of either invalid or valid tests. E.g number inputs where the min value is 1 and max value is 5