**AS 2.7: TESTING LOG**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Specified Task** | **What is to be Input?** | **What is the Expected Output?** | **What is the Actual Output? (Screen snip)** | **Comments /Decisions** |
| **Ask user to select type of headphones.** | **Expected (headphones, earbuds, both)** | **Move on to next function** | **17Mar:**  result | **17Mar: Expected, moved on to next question** |
| **Ask user to select type of headphones.** | **Unexpected**  **(abc, 123)** | **Re-ask input question** | **17Mar:** | **17Mar: expected to re-ask question** |
| **Ask user to give price range.** | **Expected (100, 1000)** | **Move on to next function** | **17Mar:**  Result: | **17Mar: expected, moved on to next question.** |
| **Ask user to give price range.** | **Boundaries (0, 1)**  **(100, 100)** | **Move on to next function** | **17Mar:**  **Results:** | **17Mar: It is expected – moved on to next question;  Add-on to next iterations: If input price range not in price range of products, tell user there is no suitable product and ask for restart.** |
| **Ask user to give price range.** | **Unexpected (abc), (-1), (0, 0)** | **Re-ask input question** | **17Mar:  Result:** | **17Mar: Expected – re-ask input question until meet requirements.**  **Add-on to next iteration: Currently, when maximum price input is non-integer, it jumped back to asking minimum price but not maximum price, which is quite annoying. Try to change so that it re-asks maximum price.** |
| **Ask user to choose brand of product.** | **Expected + Boundary (1)** | **Move on to next function** | **17Mar:**    **Result:**      **26 Apr:  Results:  -  Results:      27Apr:**  **Results:**  -  result: | **17Mar: Expected** |
| **Ask user to choose brand of product.** | **Boundary (16)** | **Move on to next function** | **17Mar: Result:** | **17Mar: Expected** |
| **Ask user to choose brand of product.** | **Unexpected (abc, 0, 17)** | **Re-ask input question** | **17Mar: Result:**  **27Apr:**  **result:** | **17Mar: Expected** |
| **Ask user to choose specific functions.** | **Expected + Boundary (1)** | **Move on to next function** | **17Mar:**  Result: | **17Mar: Expected** |
| **Ask user to choose specific functions.** | **Boundaries (10)** | **Move on to next function** | **17Mar:** | **17 Mar: Expected** |
| **Ask user to choose specific functions.** | **Unexpected (abc, 0, 11)** | **Re-ask input question** | **17Mar:** | **17Mar: Expected** |
| **Ask user to choose specific Bluetooth codec.** | **Expected + Boundary (1)** | **Move on to next function** | **17Mar:**    **Result:** | **17Mar: Expected** |
| **Ask user to choose specific Bluetooth codec.** | **Boundaries (10)** | **Move on to next function** | **17Mar:** | **17Mar: Expected** |
| **Ask user to choose specific Bluetooth codec.** | **Unexpected (abc, 0, 11)** | **Re-ask input question** | **17Mar:** | **17Mar: Expected** |
| **Ask if user would like the product to include an app.** | **Expected + Boundaries (1, 3)** | **Move on to next function** | **17Mar:**   **Result:** | **17Mar: Expected** |
| **Ask if user would like the product to include an app.** | **Unexpected (abc, 123)** | **Re-ask input question** | **17Mar:** | **17Mar: Expected** |
| **Ask user preferences on battery life long.** | **Expected + Boundaries (1, 5)** | **Move on to next function** | **17Mar:**  Result: | **17Mar: Expected** |
| **Ask user preferences on battery life long.** | **Unexpected (abc, 123)** | **Re-ask input question** | **17Mar:** | **17Mar: Expected** |

**Functions after Formative Feedback  
  
After formative feedback, I have combined brand\_fc(), function\_() and codec\_() to one generalised function mult\_choice\_fc(), alongside with filter\_fc() to filter out products after every user choice.  
Also added restart\_fc() for restart program.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Check price range of user input** | **Expected / Boundary (1, 79)** | **Return error message and trigger restart\_fc()** |  |  |
|  |  |  |  |  |
| **Check user input is in price range** | **Expected / Boundaries (user input were 1 & 79)** | **Error message and trigger restart\_fc()** |  | **Expected: In this case, the error message “Oops! You have selected…” has printed out as intended, and restart\_fc() has triggered** |
| **Ask user preferences in the form of multiple choices** | **Expected (1, 2)** | **Move on to next function** | **27 Apr:**  Result:    **Final:** | **27 Apr: Unexpected final recommendation: Filtering error, there should be products fulfil requirements, yet restarted.  Final: Expected result at the end** |
| **Ask user preferences in the form of multiple choices (brand)** | **Boundary (1, 16)** | **Move on to next function** |  | **Expected** |
| **Ask user preferences in the form of multiple choices (function & codec)** | **Boundary (1, 10)** | **Move on to next function** |  | **Expected** |
| **Ask user preferences in the form of multiple choices** | **Unexpected (0, qwe, 123)** | **Resask the specific mult\_choice\_fc()** |  | **Expected** |
| **Ask user to restart program or not** | **Expected (Yes, No)** | **If input is  yes - Restart program by trigger recommend\_fc()**  **no – end program** | **Final:**  **Results:**    **-** | **Expected** |
| **Ask user to restart program or not** | **Unexpected (qwe, 123)** | **Print error message and reask** |  | **Expected: “please type yes or no” is executed and restarted** |

Brief on Boundary Testing Video (please refer to video for demonstration):  
It shows when the price\_check\_fc() is used – when user input are $1 - $79 or $1000 - $10000. Since the cheapest product is $79.99 and the most expensive product is $999.99, if user has input a price range out of this range, this function will be called out to tell user error message and ask for restart. Boundaries are 1 and 1000.  
A few adjustments can be in the aspect of boundaries setting:

* Upper boundary for battery life
* Possible Upper boundary for price range