# File Read Operation Test

This test is to ensure that the data file is correctly read into the application.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test details** | **Expected Results** | **Actual Results** | **Screen shots** | **Pass/Fail** |
| **1.1** | Reading data file data.txt | The file should be read into the application and formatted to a key value data structure the key being the food item name, and the value will be a list of integers. This should be displayed in the console | The file was read in successfully and formatted into a key value pair. The key is the food item name, and the value is a list of integers. This displays on the console app correctly |  | Pass |

## File Read Operation Test Summary

Test 1.1 has successful passed providing evidence via the screen shots we can see that the file contents have been correctly mapped to Map[String, List[Int]] and is ready for analysis. I can confidently say that this test has passed

# Console Menu Testing – menu options 1,2,3,4,7

These tests are to check that the menu selections are working correctly and display the correct results

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test details** | **Expected Results** | **Actual Results** | **Screen shots** | **Pass/Fail** |
| **2.1** | Menu displays correctly in the console at start up | The menu should display on the console, showing number 1-7 each with a short description of what the option does | The menu displays correctly, shows the number options 1-7 and the options description. |  | Pass |
| **2.2** | Selecting option 7 – quitting the application | the console should display a message about quitting the application | Console displays message to user that the selected 7 and that the application is now quitting. |  | Pass |
| **2.3** | Selecting option 1 – viewing all the current price of food | Selecting option one should display a list of the food items in the food data and the current price(last element) of each food. The menu should then be shown again ready for the next menu option | Console display correctly the results of the food items with their current price (last element) for each food. The console shows the menu allowing for another option to be selected |  | Pass |
| **2.4** | Selecting option 2 – getting the highest price | Selecting option 2 should show the food item list with the highest price from the list. Then display the menu for another option selection | Selection option 2 correctly shows the highest price for the food item and displays the menu for another option selection |  | pass |
| **2.5** | Selecting option 3 – showing the lowest price | Selecting option 3 should display the lowest price for each food item, then the menu should display waiting for the next option input | Selecting option 3 displays the food and the lowest price for each food item. The menu correctly displays after the results ready for the next option input |  | Partial Pass –  Formatting issues with the cost, only showing 1 decimal place. |
| **2.6** | Selecting option 4 – showing the median price for each food | Selecting option 4 should display each food and the average cost for the period, then display the menu ready for the next option choice | Selecting option 4 displays the median price for each food and then displays the menu ready for the next option input. |  | Pass |

## Console Menu Testing summary – menu opts 1,2,3,4,7

Test 2.1 passed successfully displaying the menu when the application is started.

Test 2.2 passed successfully providing a message the user has selected option 7 and that the application is now quitting.

Test 2.3 passed successfully providing the current prices for each food item. Each cost is correctly formatted. Once completed the menu is printed to screen ready for the next menu selection choice.

Test 2.4 passed successfully providing the highest price for each food item. Each cost is correctly formatted. Once completed the menu is printed to screen ready for the next menu selection choice.

Test 2.5 partially passed successfully displaying the lowest price for each food item. Cost has some formatting issues where some values only show 1 decimal place rather than 2. Once completed the menu is printed to screen ready for the next option input.

Test 2.6 passed successfully providing the median price for each food item. Each cost is correctly formatted. Once completed the menu is printed to screen ready for the next menu selection choice.

Functionally the application meets the requirements, although there are some formatting issues in test 2.5. some of the results return values like £2.0 or £6.6, this is not major issue as it is likely that the logic used to calculate the cost is not ensuring 2 decimal places and is omitting the final zero. The results should be £2.00 and £6.60.

# Console menu testing – menu opt 5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test details** | **Expected Results** | **Actual Results** | **Screen shots** | **Pass/Fail** |
| **3.1** | Selecting option 5 – comparing average food cost | Selecting option 5 should allow the user to input 2 food items and get the average price for each item and display the difference for each item entered. Once complete the menu should display ready for the next option input | Selection option 5 allows a user to input 2 food items. Receives the average cost of each item and the difference between them, then the menu is displayed ready for the next option input | A screenshot of a computer program  AI-generated content may be incorrect. | pass |
| **3.2** | Selecting option 5 – entering lower case text | Same as test 3.1 except using lower case input. Allow a user to input 2 food items (lowercase) display the average cost of the entered items and the difference once completed, menu should display ready for the next menu option | Selection 5 allows a user input 2 food items (lowercase). Error – see screenshot. |  | Fail |
| **3.2.1** |  | See 3.2 | Selection 5 allows a user to input 2 food items(lowercase) – the system will then get the median prices for each food item entered and display the average cost for each food and the difference in cost. Then the menu should display ready for the next option input |  | pass |
| **3.2.2** | Selecting option 5 – and entering nothing at all | Display an error message after pressing enter 2 times without entering any text should return a error that the user needs to enter text, then display the menu ready for the next option input | Displays an error that food item can’t be blank when trying to compare food average prices then return to the menu ready for the next input option.  Now fixed |  | Partial fail – did return an error but tried to search for an empty field  pass |
| **3.2.3** | Selecting option 5 – entering missing food items | Entered:  Food1 = banana  Food2 = peach  Return that food 1 was not found, then should display the menu ready for the next option input | return user message that the food item was not found, then displayed the menu ready for the next option input. |  | pass |
| **3.2.4** | Selection option 5 – entering only 1 missing item | Entered:  Food1 = banana  Food2 = apple  Return an error that the food item could not be found, should then display the menu ready for the next option input | Returns an error that banana could not be found and then display the menu ready for the next option input |  | Pass |

## Console menu testing – opt 5 summaries

Test 3.1 passed successfully, allowing user to input 2 food items, returning the average cost of each item and providing the difference in cost between the 2 foods entered.

Test 3.2 failed see screenshot. This was because my logic code was trying to search for the item but is case sensitive and was correctly returning a None value as the item could not be found but was failing to deal with the None value correctly. I changed the compareFood method from an Option wrapper to a Try Wrapper that if both prices were found it was a Success. If the case returned that the Option value is None for either the first or the second food item it now results in a Failure.

Test 3.2.1 now passes when lower case text is entered.

Test 3.2.2 now passes successfully with the correct error messages when the text is blank (nothing entered)

Test 3.2.3 passed successfully with the correct error message when both foods are not in the list that a food item entered was not found.

Test 3.2.4 passed successfully with the correct error message when only one food item is in the list.

Testing the opt 5 – comparing average food prices passed functionally when data entered in the correct format was used. However, errors arose when the entered text was not formatted correctly to search for food items. To fix this problem I chained the toUppercase method to the readLines for user input. This fixed the issue with case sensitivity. But revealed other issues such as dealing with a None value when the food item was not found. I changed the compareFood method from Option to Try to utilize the success and failure wrappers. After thorough testing I can confidently say that this functionality has passed however there are still some formatting issues that need to be rectified.

Formatting issues have been fixed using the f interpolator for formatting and used %.2f for 2 decimal places.

# Console menu testing – opt 6: shopping basket

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test details** | **Expected Results** | **Actual Results** | **Screen shots** | **Pass/Fail** |
| **4.1** | Selecting option 6 – entering known items | Allow users to enter multiple food entries, when user is finished entering items they type “done” then the results will be displayed to the console showing the food item, single price, the amount, the food cost price and finally the total price for the basket. The menu should then display ready for the next option input |  |  | pass |
| **4.2** | Selecting option 6 – entering items not in the food items list. | Allow user to enter multiple food entries when user enters “done” the results will be displayed, show the foods found, single price, quantity, and food cost price and finally the total price of the basket. Any food not found should be displayed “not found” then the menu should be displayed ready for the next input option. | Allows user to enter items and quantity,  Items entered:  Banana, 3  Apple, 2  Milk, 2  Egg, 2  Done  Error calculating shopping basket: egg not found in current price list. |  | Partial pass –  Allows user to enter multiple items and quantity. Does show an error that a food item was not found, however it is the second missing item that was found first. Does not show items that were in the current price list. |
| **4.3** | Selecting option 6 – not entering food an only inputting quantity | Should let user know that they can not enter a blank food item and get them to enter items again | Allows user to enter blank items and input quantity when done is entered, error shows explaining that “” was not found. |  | fail |
| **4.3.1** |  | Should let user know that the can not enter blank food item and gets them to enter item again | Now shows invalid food – please enter food item error, allows user to continue before display results then displaying the menu ready for the next input option |  | pass |
| **4.4** | Selecting option 6 – entering item name but not quantity | Should show error about invalid quantity and get user to enter item and quantity again when done is entered results should display correctly then the menu should display ready for the next option input | Shows error about invalid quantity and gets user to enter item and quantity again. Displays results and shows menu ready for the next input option |  | pass |
| **4.5** | Selecting option 6 – entering done with out entering any items | Should show error that there is no items in the basket and display menu ready for the next option input | Shows a empty shopping basket with 0 total cost, displays menu ready for the next input option |  | Partial pass- calculates an empty basket. |
| **4.5.1** |  |  | Shows an error that the basket is empty then return to display menu ready for the next option input |  | pass |

## Console menu testing – opt 5: Summary

Test 4.1 passed successfully showing items from the current list and displaying the basket list and the total for known items.

Test 4.2 partially passed – does not calculate the basket if there is an item not in the list, however for user experience it would be nice to show user the items that were found and the items that were not found.

Test 4.3 failed as it allowed users to leave the food item blank and enter quantity, this did result in an error explaining that “” (nothing) was found but would be nice to tell them and get them to re-enter the item again.

Test 4.3.1 now passed successfully when the food item is entered as “” (nothing) the system no shows error and allows user to re-enter food item.

Test 4.4 passed successfully when food item is entered, but quantity is left blank. Displays error and ask user to re-enter food item

Test 4.5 failed when user does not enter any items and directly types done, should have displayed error, but calculated an empty basket.

Test 4.5.1 now passes successfully displaying error that the basket is empty.

Option 6 presented numerous issues that were able to be rectified, however there are still issues present such as a user can not cancel a basket without receiving an error that there basket is empty due to them trying to exit or if users want to cancel mid list, they would have to proceed regardless which may not be user friendly however its functionality of receiving items and quantity inputted from user, calculating the cost of those items and displaying results of single price, amount the total cost of food item ordered and a basket total cost passes successfully.