The food shop application I developed in Scala for CW2 of the Programming Module makes use of a number of different error handling techniques, ranging from Options, to pattern matching, Scala.Try actions, recursive input validation and more. The result was a successful comprehensive testing plan which required very few changes.

Any failed tests are logged in the second table in this document, with a section detailing changes made.

| **Programming CW2 Test Table** | |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Action | Expected Result | Actual Result | Evidence |
| 1 | Run the Application | Menu appears | Menu appears |  |
| 2 | Enter a non numerical value | Menu reappears with an error message | Menu reappears with an error message |  |
| 3 | Enter an invalid menu item number | Menu reappears | Menu reappears with an error message |  |
| 4 | Enter number 1 | List of all foods and their most recent prices appears, then menu is redisplayed | List of all foods and their most recent prices appears, then menu is redisplayed |  |
| 5 | Enter number 2 | List of all foods and their highest and lowest prices over the period appears, then menu is redisplayed | List of all foods and their highest and lowest prices over the period appears, then menu is redisplayed |  |
| 6 | Enter number 3 | A list of the foods and their median prices are displayed, then menu is redisplayed | A list of the foods and their median prices are displayed, then menu is redisplayed |  |
| 7 | Enter number 4 | The user is prompted for a food symbol | The user is prompted for a food symbol |  |
| 8 | Enter input for first food average: “TOMATO” | The user is prompted for a second food symbol | The user is prompted for a second food symbol |  |
| 9 | Enter one or two invalid food items: “TURKEY” | Error message is displayed after both inputs have been taken; menu redisplayed | Error message is displayed after both inputs have been taken; menu redisplayed |  |
| 10 | Enter two valid food symbols “TOMATO”, “beef” | The average prices of the two foods are displayed with a comparison of the values | The average prices of the two foods are displayed with a comparison of the values |  |
| 11 | Enter number 5 | Prompt is displayed asking user how many items they wish to have in their basket | Prompt is displayed asking user how many items they wish to have in their basket |  |
| 12 | Enter non numerical value | Error message is displayed, user prompted again | Error message is displayed, user prompted again |  |
| 12.1 | Enter exceptional data: -2 | Number is processed as a positive num | Number is processed as a positive num |  |
| 13 | Enter valid number | User is prompted to enter a food | User is prompted to enter a food |  |
| 14 | Enter invalid food | Error message will be displayed after quantity prompt telling the user that their food item is invalid and returns the prompt | Error message will be displayed after quantity prompt telling the user that their food item is invalid and returns the prompt |  |
| 15 | Enter valid food | User will be prompted for quantity in kg | User will be prompted for quantity in kg |  |
| 16 | Enter non numerical value | Error is displayed, user is prompted to enter again | Error is displayed, user is prompted to enter again |  |
| 16.5 | Enter exceptional data: -2 | Number is processed as a positive num | Number is processed as a positive num |  |
| 17 | Enter valid number | If user has more entries to enter, they are prompted for the next food symbol. Otherwise, the basket is displayed | If user has more entries to enter, they are prompted for the next food symbol. Otherwise, the basket is displayed |  |
| 18 | Enter number 6 | Program terminates | Program terminates |  |

| Failed Test Logs | |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Action | Expected Result | Actual Result | Changes Made |
| 2 | Enter a non numerical value | Menu is redisplayed | App Crashes | Added a recursive error handler |
| 14 | Enter invalid food | Error message will be displayed after quantity prompt telling the user that their food item is invalid and returns the prompt | App proceeds as normal before omitting the erroneous entry from the basket entirely | Added pattern matching to validate the food symbol input and reprompt if the map.get() method returned a None. |
| 12.1 | Enter exceptional data: -2 | Number is processed as a positive num | App crashes | Wrapped integer inputs with math.abs() so as to ensure any negative integer inputs are converted to positive |