**SWStarships Question - Code Documentation - Niamh Maquet**

**Program used:** Visual Studio 2017

**Project type:** Console application

**Language used:** C#

**Purpose of the application**

The purpose of this application is to calculate the number of stops a starship needs to resupply to complete a journey of a specified distance.

**How to use the program**

* Enter a value (MGLT) into the textbox.
  + Numeric values with a minimum of 0 are accepted. This is required to use the application. One exception if this is the letter “x” which is used as the keyword to close the app.
* Press the submit button. If there is nothing entered in the textbox an error will be displayed to highlight this. The app will re-display the prompt to enter a value.
* If a valid numeric value has been entered, the results will be displayed in a list below the submit button in the format displayed in the requirements (Starship Name: Stops).
* If a string other than “x”is entered, an error message will be displayed and the enter value prompt displayed again.
* If “x” has been entered, the application will close.

**Process**

The application takes in a numeric value and performs various checks (for empty/null/invalid values). Once this is complete, the StarWarsAPI NuGet package is used to get the list of starships. The results are looped through and sent a method to calculate the number of stops required.

In this method, the required maths calculation is selected based on the consumables (if they contain, month, year, week, day or unknown). An Enum is used to to hold various time span values and are used in calculating the stops required for each of the different durations mentioned in the consumables. Once the correct math has been selected, the result is returned and is printed along with the name of the starship to the console application.

**Tests**

A total of 5 unit tests were designed. These can be found in the StarshipsApp.Tests project and were completed using Visual Studios unit testing. they focus primarily on the calculation logic. Four of the tests assert valid data records to confirm an example of each of the four different time spans (years, months, weeks, days) are calculated correctly. The fifth test is used to show records with no values provided being handled. Much Of the code outside of the calculation method contains error handling through TryCatch blocks and if statements to validate variables.