Software Engineering

4. A: Critical reflection of usage of SCRUM in artefact development

During the development of the artefact, the task was divided into several more manageable sections to be accomplished within individual sprints. The procurement of API keys, development of a GUI, retrieval of headlines via the News API, distillation of headlines into searchable terms, and so on.  
Each sprint consisted of general planning, implementation, testing, and further modification. Until such a point that the current section could be considered functional, then the code was tidied, and commented, for ease of understanding.

During each sprint, comments were left within the source-code, explaining what certain parts do, why they are required, and any ways in which they do not currently work.

The use of SCRUM within development could have been improved if more rigorous testing protocols were employed for each sprint, perhaps using Unit Testing, such that flawed routines could have been documented and rectified in a more efficient manner. As it stands, testing was done throughout implementation, issues being patched before moving onto the next piece of code, without sufficient documentation.  
A further improvement could have been to ensure that multiple people were developing the artefact. In reality however, asides from some pair-programming, the main development was handled by a single team member.  
The planning part of each sprint would have been improved by writing pseudocode and creating data-flow diagrams as to ensure that the task was adhered to. The lack of these items lead to detours during implementation, which lead to significant loss of time, and made the structure difficult to follow for others.

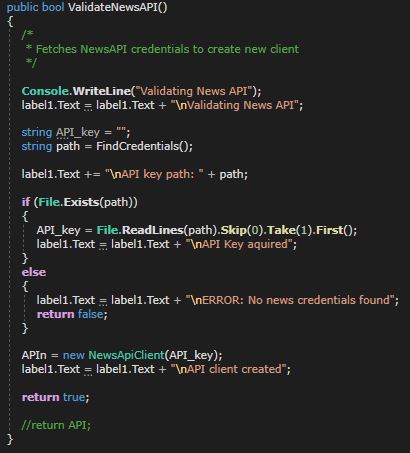
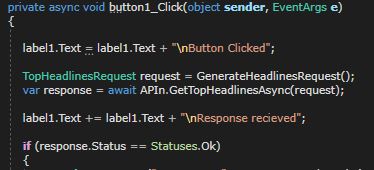


Figure : C# Snippet

Figure : C# Snippet

The result of an early sprint is shown in *Figure 1*. The figure shows the initial lines that are called upon a user pressing the *‘News’* button, being that a debug log is appended to show the event, a *‘TopHeadlinesRequest’* is generated based on fields in the GUI, and that request is used to fetch headlines from the NewsAPI.

In *Figure 2*, the routine that generated an API client is shown. This function finds the file in a drive that contains API credentials, and proceeds to create a usable client. Should the function fail to retrieve the API key, or should that key fail validation from the API, the routine shown in *Figure 3* will be called, preventing the user from doing anything, resulting in a GUI as shown in *Figure 4*.

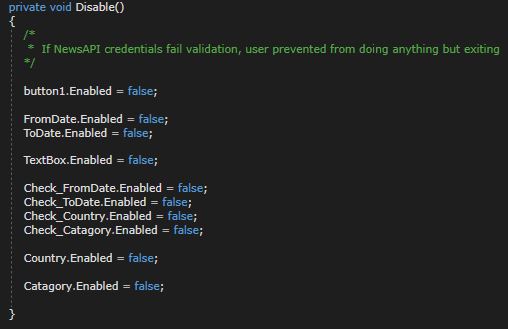


Figure : C# Snippet

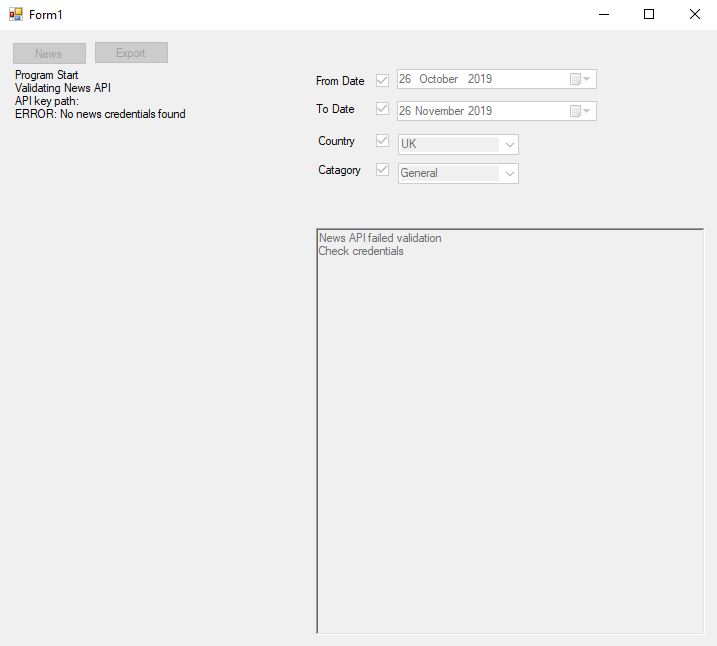


Figure : GUI

Ideal Dev. Progression

Research NewsAPI & Spotify API

Retrieve NewsAPI key

Retrieve Spotify API credentials

Create Spotify account

Create Spotify API Application

Practice using NewsAPI

Create C# GUI for NewsAPI

Program NewsAPI handler

Program C# Headlines export

Practice using Spotify API

Program Spotify Headlines import

Program Spotify API handler

Create C# - Python interface