Software Engineering Coursework

REPORT SUBTITLE

40283288 | SET09102 | 21/11/2018

Lecturer: Xiaodong Liu

### Introduction

This report contains the documentation for the SET09102 coursework, the Napier Bank Message Filtering Service. The coursework involved the development of an application prototype, which I have decided to implement into a WPF application using C#. The application uses various filters in order to correctly process the data into their corresponding message type.

### Requirement Analysis

The client has requested the development of a service that will validate, sanitise and categorise incoming messages to Napier Bank. All messages have a message header comprising of a Message ID (Message-type “S”, “E”, or “T” followed by 9 numeric characters, e.g. “T123456789”) followed by the body of the message

The system must deal with the following three message types based on the message body:

* SMS Messages

The SMS message body contains the Sender in the form of an international telephone number and the Message Text that is a maximum of 140 characters. The text message also may contain “textspeak abbreviations”, which are supplied in a CSV file. The abbreviations must be identified and expanded to their full form enclosed in a “<>” tag.

* Email Messages

Emails are of two types: Standard email messages and Significant Incident Reports. Both types have a Sender in the form of an email address.

Standard email messages have a subject that is maximum 20 characters long, followed by a maximum of 1028 characters message body.

Significant Incident Reports have the subject in the form “SIR dd/mm/yy” and will start the message text with a sort code on the first line, followed by the “Nature of incident” on the next line. Nature of incident can be one of the following:

|  |
| --- |
| Theft |
| Staff Attack |
| Raid |
| Customer Attack |
| Staff Abuse |
| Bomb Threat |
| Terrorism |
| Suspicious Incident |
| Intelligence |
| Cash Loss |

In addition to these rules every email can contain URLs (in the form of “http://” or “[https://](NULL)”) that need to be removed and written to a quarantine list and replaced by “<URL Quarantined>” in the message body.

* Tweet Message

The tweet message body contains a Sender in the form of a Twitter ID that starts with “@” followed by a maximum of 15 characters. The tweet text is a max of 140 characters and contains the same textspeak abbreviations as SMS. In addition to this, they might also contain hashtags (e.g. #software) or other Twitter IDs in the form of “mentions”. Hashtags will be added to a hashtag list that count the number of uses to produce a trending list. The mentions will be added to their own mention list.

### Design

#### WPF User Interface

While designing the application’s user interface, the Visual Studio WPF editor was used as the starting point in order to give the developer and idea about the scope and structure of the application.

#### Main Window

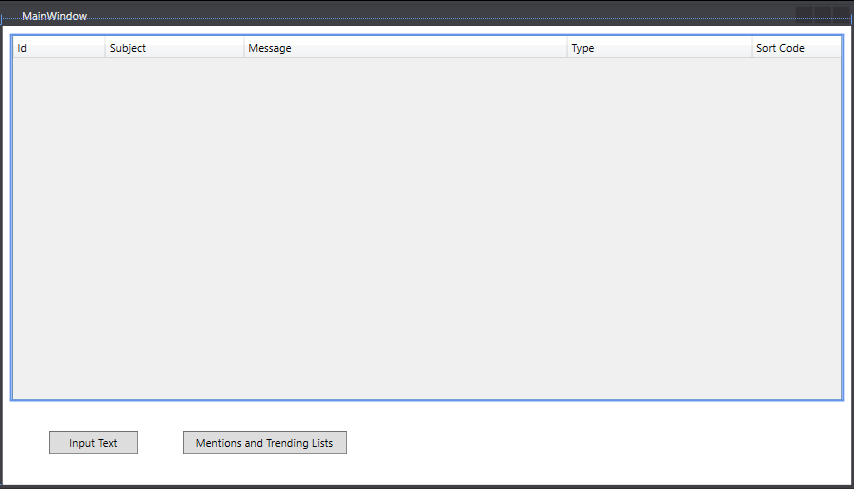
The Main Window [Figure 1] of the application is where the user can view the full list of submitted messages and their various properties. Not all fields are used by all the messages, but a simple unified list helps the user by not overwhelming the screen with unwanted options. The main window is also where the user can view the trending and mention lists as well as go and submit new messages.

Figure Main Window

#### Message Input

The Message Input Window [Figure 2] is where the user can input their various messages. The user can choose between the three message types, depending on the type different options might appear [Figure 3].

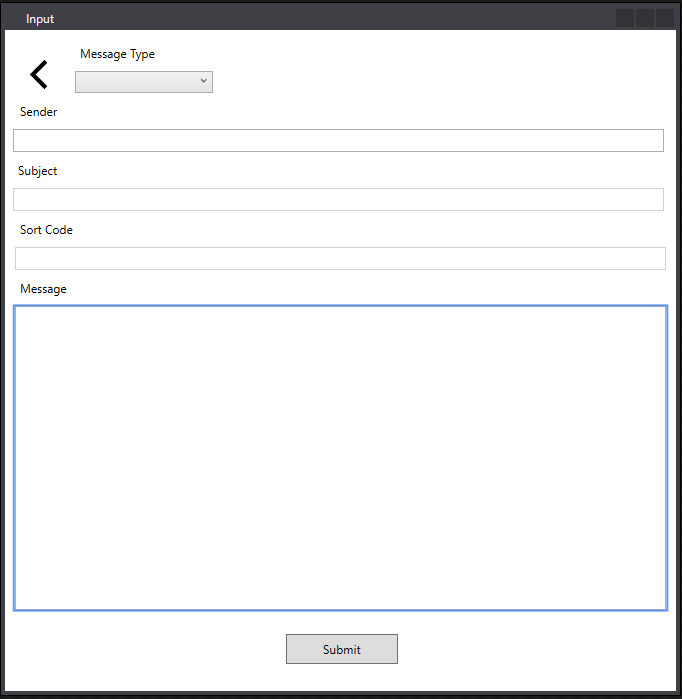
The Email input is the most complex one, and it can take the twi different types of emails: SIR and Standard. If the user chooses SIR the subject must be entered in the dd/mm/yy form. Another selection can be made for the nature of the incident and the user can type in a sort code as well. If standard email is selected the user can only input in the sender(in the form of an email address), the Subject (in any form they wish) and the message body. Regardless of email type, the system will then filter and quarantine the entered URLs.

Figure Input Window, No Selection made

If the user selects the message type as either SMS or Tweet, only a sender (int their respective format) and message body can be inputted in. The system will then expand the abbreviations for both types and count the hashtags and mentions for tweets.

Figure Choosing Email Options

### Use Case Diagram

Figure The Use Case Diagram

The Use Case Diagram [Figure 4] was created as part of the specification requirements. The diagram involves two actors: the user and the application system. The user can create the three types of messages and view them afterwards. The system manages the validity of the data entered and saves the messages to a file if they are valid. When the user wants to view the list of messages, the system reads them from the file and displays them as appropriate.

### Class Diagram

The Class Diagram [Figure 5] shows the Classes that were built for the purpose of the app. The WPF window classes were not included in the diagram.

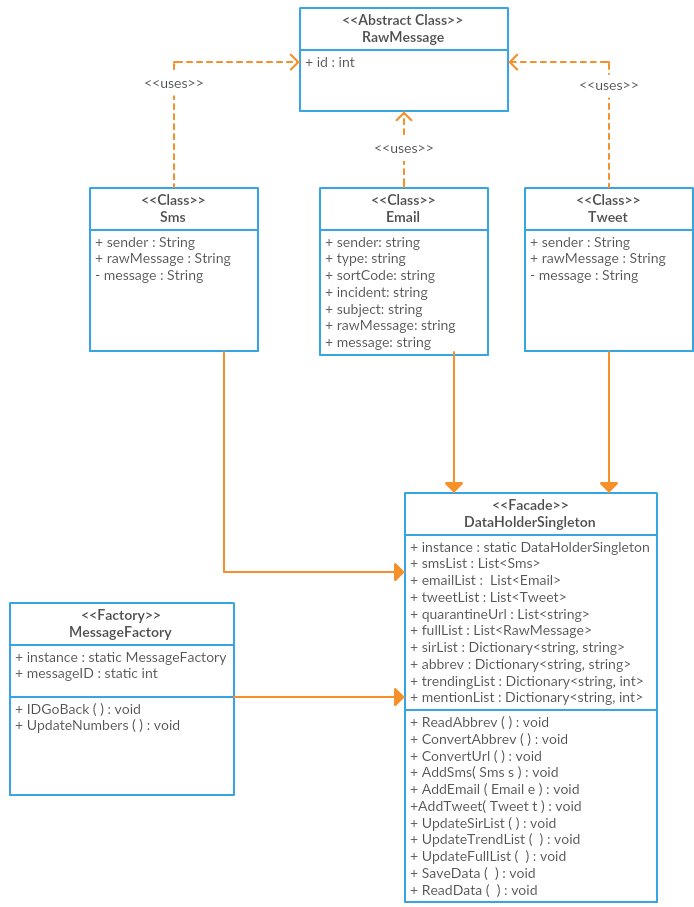


Figure The Class Diagram