SET09102 – Coursework

This document and the attached scenario covers the coursework submission.

Euston Leisure Message Filtering Service

1. Introduction

This coursework comprises 3 parts:

- Software Development Report
- Prototype
- Demonstration

2. Scenario and Requirement

Euston is a medium-sized city. Euston Leisure is an association of all the sport centres in the city. You are required to develop a service, namely Euston Leisure Messaging (ELM), which will validate, sanitize and categorise incoming messages to Euston Leisure in the form of SMS text messages, emails and Tweets.

2.1 Message Types

The system must deal with three types of message.

All messages are strings of ASCII characters that have a **Message Header** comprising a **Message ID** (Message-type “S”,”E” or “T” followed by 9 numeric characters, e.g. “E1234567701”) followed by the **Body** of the message.

Depending on the message type the **Body** will comprise:

- **SMS messages**
  - SMS message bodies comprise **Sender** in the form of an international telephone phone number followed by the **Message Text** which is a maximum of 140 characters long. The **Message Text** message is simple text but may contain embedded “textspeak abbreviations”. Details of the textspeak abbreviations that may be embedded are supplied on Moodle in the form of a CSV file.
- **Email Messages**:
  - Email message bodies comprise **Sender** in the form of a standard email address John Smith john.smith@example.org followed by a 20 character **Subject** followed by the **Message Text** which is a maximum of 1028 characters long. The **Message Text** message is simple text but may contain embedded hyperlinks in the form of standard URLs e.g. http:\\www.anywhere.com. Further detail of email messages is provided in 3.1.2 below.

- **Tweets**
    - Tweet bodies comprise **Sender** in the form of a Twitter ID: "@" followed by a maximum of 15 characters (e.g. @JohnSmith) and the Tweet text which is a maximum of 140 characters long. In addition to ordinary text the Tweet text may contain any of the following:
        - **textspeak** abbreviations (as in SMS above)
        - **hashtags** - strings of characters preceded by a '#' sign that are used to group posts by topic. (such as #BBCClick, #1Donice).
        - **Twitter IDs** as above

3. System Development

You are required to develop a prototype application that will enable the inputting of messages in any of the forms in 2.1 above. The system must detect the message type and output each message in JSON format in a file. You are required to research JSON and identify an appropriate API to allow serialisation in a JSON file. Good places to start are:

https://www.w3schools.com/js/js_json_syntax.asp
https://blog.udemy.com/json-serializer-c-sharp/
https://msdn.microsoft.com/en-us/library/bb410770(v=vs.110).aspx

3.1 Message Processing

Messages must be processed as follows:

3.1.1 **SMS Messages:** Textspeak abbreviations must be expanded to their full form enclosed in "<>", e.g. "Saw your message ROFL can't wait to see you" becomes "Saw your message ROFL <Rolls on the floor laughing> can't wait to see you"

3.1.2 **Email Messages:**

Email messages are of two types: **Standard email messages** and **Significant Incident Reports** that comprise text reports from bank branch managers concerning incidents of significance that happened during the working day, such as robberies, significant cash shortages, violent incidents. Both types may contain embedded URLs

Standard email messages will contain text. Any URLs contained in messages will be removed and written to a quarantine list and replaced by "<URL Quarantined>" in the body of the message.

Significant Incident Reports will have the **Subject** in the form "SIR dd/mm/yy" and will comprise a message body as above. The message body will begin with the following standard texts on the first two lines:

Sport Centre Code: 66-666-99[1]

Nature of Incident: which will be one of the following (see over):

Theft of Properties
Staff Attack
Device Damage
Raid
Customer Attack
Staff Abuse
Bomb Threat
Terrorism
Suspicious Incident
Sport Injury
Personal Info Leak

Centre Code and Nature of Incident will be written to a SIR list.

Any URLs contained in messages will be removed and written to a quarantine list and replaced by "<URL Quarantined>" in the body of the message.

3.1.3 **Tweets:** Textspeak abbreviations will be expanded (as in SMS messages above). Hashtags will be added to a hashtag list that will count the number of uses of each to produce a trending list. "Mentions", i.e. embedded Twitter IDs will be added to a mentions list.

User Interface

The User Interface (UI) should take the form of a kind of input form(s), e.g. WPF or Java form. For the purposes of testing, messages will be input in the form of the Message Header (in one text box) and a block of Message Body text (in another text box) and redisplayed in appropriate text box(es) processed as specified above. The system must *automatically* identify the message type and process it accordingly.

Ideally the system will also be able to take its input from an input file.

At the end of an input session the system should display the trending list, the list on Mentions and the SIR list.

[1] E.g. 83-219-19