Coursework Report

Erik Sanne

[40276539@live.napier.ac.uk](mailto:40276539@live.napier.ac.uk)

Edinburgh Napier University – Software Engineering (SET09102)

# Introduction

Euston Leisure requires a system (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.

This report details the:

* Requirement specification
* System design
* Implementation
* Testing
* Evolution

# Requirements

## Functional requirements

The system is required to validate, sanitize and categorise incoming messages. Messages can be SMS text, Emails or Tweets.

On a high level the Euston Leisure Messaging system has to:

* Detect the message type
* Validate the message
* Sanitize the message
* Maintain a list of hashtags and the number of occurrences for each hashtag
* Maintain a list of Twitter ids mentioned in messages
* Maintain a list of quarantined URLs
* Maintain a list of Significant Incident Reports
* Output the message in JSON

The below is a detailed specification for validating, sanitizing and categorising the message.

The incoming message has a Message Header and Body.

**Message header:**

The header should be validated and categorised into the correct message type.

* Validate:
  + Message ID is a Message Type (“S”, “E”, “T”) followed by 9 numeric characters.
    - i.e. “E1234568789”
* Categorize
  + “S” should be detected as SMS message and processed as such
    - i.e. “S123456789”
  + “E” should be detected as Email message and processed as such
    - i.e. “E123456789”
  + “T” should be detected as Tweet and processed as such
    - i.e. “T123456789”

**Body:**

The body should be processed for each of the message types as explained below.

**SMS Messages:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| **Header:**  **S123456789**  **Body:**  **+4412345678910 That’s funny LOL** | **{**  **"id": "S123456789",**  **"sender": "+4412345678910",**  **"messageText": "That’s funny LOL <Laughing out loud>"**  **}** |

* Validate:
  + Sender is international phone number
    - i.e. +4412345678910
  + Message text is no longer than 140 characters
* Sanitize:
  + Expand textspeak abbreviations
    - i.e. LOL <Laughing out loud>

**Email Messages:**

* **Standard Email Message**

|  |  |
| --- | --- |
| **Input** | **Output** |
| **Header:**  **E123456789**  **Body:**  **john.smith@example.org**  **Look at this**  **This is a cool website http://www.cool.com** | **{**  **"id": "E123456789",**  **"sender": "john.smith@example.org",**  **"subject": "Look at this",**  **"messageText": "This is a cool website <URL Quarantined>"**  **}** |

* + **Validate:**
    - **Sender is a standard email address**
      * **i.e.** [john.smith@example.org](mailto:john.smith@example.org)
  + **Subject is no longer than 20 characters**
  + **Message text is no longer than 1028 characters**
* **Sanitize**
  + **URLs will be written to a quarantine list and replaced by “<URL Quarantined>”**
    - **i.e. “This a cool website http://**[www.cool.com](http://www.cool.com)**” becomes “This a cool website <URL Quarantined>”**
* **Categorize**
  + **If subject is in the form of “SIR dd/mm/yy”, the message should be detected as a Significant Incident Report and processed as such**
* **Significant Incident Report:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| **Header:**  **E123456789**  **Body:**  **john.smith@example.org**  **SIR 25/11/19**  **Sport Centre Code: 66-666-66**  **Nature of Incident: Raid**  **See here http://www.araid.com** | **{**  **"id": "E123456789”,**  **"sender": "john.smith@example.org",**  **"subject": "SIR 25/11/19",**  **"messageText": "See here <URL Quarantined>",**  **"sportCentreCode": "66-666-66",**  **"natureOfIncident": "Raid",**  **}** |

**Same as standard email plus the below.**

* + **Validate**
    - **Subject is formatted as “SIR dd/mm/yy”**
      * **i.e. “SIR 25/11/19”**
    - **Sport Centre Code is in valid format**
      * **i.e. “Sport Centre Code: 66-666-66”**
    - **Nature of Incident is a valid incident type**
      * **i.e. “Nature of Incident: Raid”**
  + **Categorize**
    - **Centre code and Nature of Incident will be written to SIR list**

**Tweets**

|  |  |
| --- | --- |
| **Input** | **Output** |
| **Header:**  **T123456789**  **Body:**  **@JohnSmith**  **My first #Tweet @Twitter** | **{**  **"id": "T123456789",**  **"sender": "@JohnSmith",**  **"messageText": "My first #Tweet @Twitter",**  **}** |

* **Validate**
  + **Sender is a valid Twitter ID**
    - **“@” followed by a maximum of 15 characters**
    - **i.e. “@JohnSmith”**
  + **Tweet text is no longer than 140 characters**
* **Sanitize**
  + Expand textspeak abbreviations
    - i.e. LOL <Laughing out loud>
* Categorize
  + Hashtags (Words preceded by a “#”) will be written to a hashtag list that will also count the the number of uses for each hashtag. The count will be used to produce a trending list.
    - i.e. #BBCClick
  + Mentions (Twitter Ids) will be written to a mentions list
    - i.e. “@JohnSmith”

## Non-functional requirements

The main non-functional requirements for ELM are:

* Usability
* Reliability
* Scalability
* Efficiency
* Capacity
* Security

These requirements affect the whole system rather than single components. If these requirements are not met the system might not be able to be deployed or fail after being deployed.

**Product requirements**

*Usability:* The system shall be easy to understand, and user-friendly as non-technical staff must be able to use to system. Any errors or issues shall be displayed clearly with an easy to understand description.

*Reliability*: The system shall always be available and should recover from any issues/errors, crashes or other downtime (if necessary, by restarting) quickly.

*Scalability:* The system must be able to scale with changing user needs such as a growing customer base of Euston Leisure.

*Efficiency:* The system shall be efficient both in time and space complexity, so it doesn’t require excessive resources.

*Capacity*: Enough resources should be allocated to support the entire userbase as well as a buffer.

*Security:* The system and its data shall only be accessible to authorised staff.

**Organizational requirements**

The system will only be used by staff, this means authentication must be available using the staff member’s login/id card. The development and system itself always have to follow any applicable internal organizational procedures.

**External requirements**

The system must follow all applicable regulatory, ethical, legislative and other legal requirements. As the system stores data such as phone numbers, emails and twitter id’s a privacy policy relevant to its location as well as any additional data security and safety requirements shall be implemented.

## Use case diagram

<Insert here>

# Implementation

## Class diagram

# Testing

# Version control

A version control system is a place to store the software’s source files, it maintains a history of changes made to the files, allows viewing the state at any point and provides tooling for a team of developers to work on the same files simultaneously.

There are various version control systems a

# Software Evolution

More input formats

API