

SET09102 Coursework

Report



October 11, 2017

Jonathan Mitchell

40311730

Contents

[Analysis 2](#_Toc496185176)

[Requirements Specification 3](#_Toc496185177)

[Functional Requirements 3](#_Toc496185178)

[Non-functional Requirements 5](#_Toc496185179)

[User Stories 7](#_Toc496185180)

[Use-Case Model 8](#_Toc496185181)

[Design 9](#_Toc496185182)

[Class Diagram 9](#_Toc496185183)

[Implementation 10](#_Toc496185184)

[Testing 11](#_Toc496185185)

[Testing Strategy 11](#_Toc496185186)

[Test Plan 11](#_Toc496185187)

[Test Cases 11](#_Toc496185188)

[Version Control 12](#_Toc496185189)

[Evolution 13](#_Toc496185190)

# Analysis

Euston Leisure, an association of sport centres in the city of Euston, are looking for a software system, Euston Leisure Messaging (ELM), to be developed.

This system will provide the following service: validate, sanitize and categorise incoming messages to Euston Leisure in the form of:

* SMS text messages
* Emails
* Tweets

## Requirements Specification

### Functional Requirements

The computer system will provide the following:

* Message select window
* SMS input page
* Email input page
* Tweet input page
* Display lists Page
* Must be able to automatically identify the message type & process accordingly
* Ability to write to text files
* Ability to read from text files.

Message Select Window will:

* Provide the user with buttons to navigate to one of the following pages:
  + SMS input page
  + Email input page
  + Tweet input page
* Provide a button to navigate to display Lists page
* Provide a button to exit the application

SMS Input Page

* All input is to be verified
* Display a message if any input is incorrect/missed out
* Process all messages appropriate to its type
* A navigate button to go to message display Page
* An insert button to store message

Email Input Page

* All input is to be verified
* Display a message if any input is incorrect/missed out
* Process all messages appropriate to its type
* A navigate button to go to message display Page
* An insert button to store message

Tweet Input page

* All input is to be verified
* Display a message if any input is incorrect/missed out
* Process all messages appropriate to its type
* A navigate button to go to message display Page
* An insert button to store message

Message Display Page will:

* Provide buttons to display one of the following lists:
  + Trending list
  + Mentions list
  + SIR list
  + Email list
  + Tweet list
  + SMS list
* Each list page will have a button allowing the user to go back a page

### Non-functional Requirements

* All messages must be strings composed of ASCII characters
* Each page will provide clear details of which box(es) information is to be inserted into
* Each message will have a Message Header comprising:
  + Message ID containing:
    - Select Message type (“S”,”E”,”T”) followed by 9 numeric characters
  + Body Will Comprise:
    - SMS Messages Will contain:
      * A SENDER
        + In the form of an international telephone number
      * The MESSAGE TEXT
        + Must be no more than 140 characters
        + It may contain embedded text speak abbreviations
    - Email Messages Will contain:
      * A SENDER
        + In the form of a standard email address
      * A SUBJECT
        + Must be no more than 20 characters
      * The MESSAGE TEXT
        + Must be no more than 1028 characters long
        + It may contain embedded hyperlinks. N the form of standard URLs
    - Tweets will contain:
      * A SENDER
        + Must consist of a Twitter ID: “@”
        + Followed by a maximum of 15 characters
      * MESSAGE TEXT
        + Must not been more than 140 characters
        + Can contain the following:

Textspeak

Same as SMS above

Hashtags

Must be a string of characters preceded by a “#” sign

Twitter ID’s

As above

## User Stories

As the System

I want to be able to deal with SMS Messages

As the System

I want to be able to deal with Email Messages

As the System

I want to be able to deal with Twitter Messages

As the System

I want to be able to process SMS messages by checking if they contain textspeak abbreviations and expand them to their full form

As the System

I want to be able to process Emails to determine if they are standard Emails or Significant Incident Reports.

As the System

I want to be able to Process Emails to see if they contain Hyperlinks and quarantine them

As the System

I want to be able to process tweets by checking if they contain textspeak abbreviations and expand them to their full form

As the System

I want to be able to process hashtags in tweets and add them to a list to display how many times they are used

As the System

I want to detect the three types of messages (SMS, Email & Tweet) and write them to a file(s) in JSON format

As the System

I want to be able to display the trending list, list of “@” mentions & the SIR list

As the System

I want to be able to process the above message types from an input file

## Use-Case Model

# Design

## Class Diagram

# Implementation

# Testing

## Test Plan

The following pages will be tested:

* Message select window:
  + All buttons will be tested to ensure they display the correct page or exit application successfully
* SMS input page:
  + Validation of each input will be tested
* Email input page:
  + Validation of each input will be tested
* Tweet input page:
  + Validation of each input will be tested
* Display lists Page:
  + All buttons will be tested to ensure they display the correct page
  + All inserted records/lists etc will be checked to ensure they have all been processed correctly
* Text files:
  + Will be checked to ensure the written records are in JSON format and display the correct information in each column/row

## Test Strategy

## Test Cases

# Version Control

GitHub will be used to create the version control. Each iteration/addition to any representation of the software system will be added to the repository with a note detailing what was done.

Link here <Insert Link>

# Evolution Strategy

1. Instead of writing to a file in JSON format. Storing in a database, with encryption, would be a more suitable option. Stored internally within the system itself.
2. The system should only require minimal maintenance. As a result, maintenance costs would be kept low.
3. Adding additional security in the form of a login page? Stored within database
4. Potentially move the system to be web-based.
5. Increase input types to include Facebook?
6. Integrating anti-virus/spam filtering software to check if URL’s in websites are genuine or not – reduces number of quarantined items.