# Introduction

## This section gives a scope description and overview of everything included in the SRS document. Also, the purpose for this document is described and a list of abbreviations and definition is provided.

## Purpose and Intended Audience

The purpose of this document is to describe the software requirements of the system. It will explain the purpose and features of the system, what the system will do, the constraints under which it must operate and how the system will react to the stimuli. It is meant to be used to maintain a shared understanding of the requirements between the developers and the clients of the system.

## Product Scope

This project will detect “Strong Others” words that are defined as strong based on their level of offensiveness or “taboo-ness”.

The study will only cover words in the English language because it is the most common language spoken by people. This will include Articles, Pronouns, Common Nouns, Verbs, Adjectives, Adverbs and all those that are included in the parts of speech or the lexical category of the said language. Words that can be recognized by the program will be based from the WordNet version 2.1.

The study will not include Proper Nouns such as names. It is for the reason that these words are most probably not in the dictionary that was the basis for all the words. The system will only analyze the input if it is a proper English sentence. Getting offensive words is based on bad words only. Censored words and abbreviations like FU are also not included in the input.

## Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| Term | Definition |
| Cyber Bullying | Is bullying that takes place using electronic technology. |
| False Negative | Classifier wrongly indicates the input is not offensive |
| False Positive | Classifier wrongly labels the input as offensive |
| Grammar rules | Set of structural rules governing the composition of clauses, phrases and words in any given natural language |
| Lexemes | Is an abstract unit of morphological analysis in linguistics that roughly corresponds to a set of forms taken by a single word |
| Natural Language Processing (NLP) | Is a field of computer science, artificial intelligence, and linguistics concerned with the interactions between computers and human languages |
| Offensive | Causing anger, displeasure or resentment |
| Parsing | Is the process of analysing a string of symbols, either in natural language or computer languages, conforming to the rules of a formal grammar |
| Polarity | Is defined as having two opposite tendencies |
| Profanity | Is an offensive word or offensive language |
| Sentiment Analysis | Refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials. |
| Strong Others | Words defined as strong based on their level of offensiveness or taboo-ness |
| Tokenization | Is the process of substituting a sensitive data element with a non-sensitive equivalent, referred to as token, that has no extrinsic or exploitable meaning or value |
| True Negative | Classifier correctly indicates input doesn’t contain offensive words |
| True Positive | Classifier correctly indicates that input contains offensive words by meaning |

## References and Acknowledgments

* “Effects of Negative Words”. Internet: http://grammar.yourdictionary.com/style-and-usage/affects-of-negative-words.html
* Dr Y Bi. “*Analysing Social Media to Detect Cyber Bullying using Sentiment Mining.*School of Computing and Mathematics, Faculty of Computing and Engineering at the Jordanstown Campus of the University of Ulster”. Internet: http://www.findaphd.com/search/ProjectDetails.aspx?PJID=56055
* Jun-Ming Xu, Xiaojin Zhu, Amy Bellmore. “*Fast Learning for Sentiment Analysis on Bullying*.Beijing,China:ACM”. Internet: http://pages.cs.wisc.edu/~jerryzhu/pub/wisdom12.pdf (August 12 2012).
* Ying Chen, Yilu Zhou, Sencun Zhu HengXu . “*Detecting Offensive Language in Social Media to Protect Adolescent Online Safety.* The Pennsylvania State University, University Park, PA, USA. George Washington University, Washington, DC, USA”. Internet: http://faculty.ist.psu.edu/xu/papers/Chen\_etal\_SocialCom\_2012.pdf
* Dawei Yin, Zhenzhen Xue, Liangjie Hong. “*Detection of Harassment on Web 2.0”.* Madrid, Spain. Internet: http://www.cse.lehigh.edu/~brian/pubs/2009/CAW2/harassment.pdf, (April 21, 2009).
* “*Profanity in Teen Novels: Characters Who Curse Are Often the Most Desirable”. Internet*: http://healthland.time.com/2012/05/18/profanity-in-teen-novels-characters-with-foul-language-are-often-the-most-desirable (May 18, 2012)
* ZhiXu, Sencun Zhu . “Filtering Offensive Language in Online Communities using Grammatical Relations”. Internet: http://ceas.cc/2010/papers/Paper%2010.pdf
* Amir Razavi, Diana Inkpen, Sasha Uritsky, Stan Matwin.” Offensive Language Detection Using Multi-level Classification”. Internet: http://www.eiti.uottawa.ca/~diana/publications/Flame\_Final.pdf

## Overview of the Document

The next chapter, the Overall Description section of this document gives an overview of the product perspective, functions, user characteristics, general development constraints and project assumptions and dependencies.

The third chapter, Requirements Specification section of this document is written primarily for the developers and describes in technical terms the details of the requirements and the functionality of the product.

# Overall Description

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of stakeholders that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

## Product Perspective and Context

**2.1.1 System Interfaces**

Sentiment Analysis for Detecting Offensive Words requires the task of inputting a correct grammar English comments with or without offensive words, opening a file containing comments and viewing a step by step procedure on how to detect offensive words. The system is designed to analyze whether the given sentence or file contains offensive words or not. If yes, then the system will print all the list of offensive words.

**2.1.2 User Interfaces**

The interface will involve text boxes and buttons. The text boxes are where the user inputs data information given by the client and this is where outputs of each procedure will be displayed. The buttons will serves as the file opener to choose a file containing comments and will be used to transfer in another form, clearing text boxes and to analyze given comments.

Standard screen format (fixed colors, fonts, background, the page layout, etc.) will be used throughout the interfaces.

The language of the user interfaces will be English.

**2.1.3 Software interfaces**

**Software –**

1. The client should input a proper English sentence or can open a file containing comments.
2. The researchers can let the client view the step-by-step procedure on how the system analyzes the given input of the client.

**2.1.4 Communication interfaces**

The Sentiment Analysis for Detecting Offensive Words don’t require internet connection and only installation of latest JDK, WordNet, Netbeans and other libraries are required. The client and the researchers can interact with the system.

**2.1.5 Memory Constraints**

The system fully requires the tools that we use in order to run the application.

.

## Product Functionality

The system lets the client to input a proper English sentence or can open a file containing the comments. After clicking the Analyze button, the system will analyze the clients input whether it is offensive or not.

## Users and Characteristics

There are two types of users that interact with the system: the researchers and the client. Even if there are two types of users that interact with the system, the process is still the same and does not have any difference.

The researchers and the clients can put any proper English sentence or may upload a file containing comments. Then the system will analyze the offensiveness of the given sentence or file.

## Operating Environment

The system will operate by installing the following requirements:

* Java version 8
* WordNet
* Netbeans
* Stanford POS Tagger
* Stanford Parser

## Constraints

Java 8 is a constraint for the application. If Java version is older or lower than Java 8, the system will not run or some of the features are missing that makes the system error. The tools used by the system is also a constraint because some of the procedures of our system makes used of the tools like POS tagger, parser, WordNet and JAWS.

## Assumptions and Dependencies

One assumption about the system is that it is expected to run in any computer that has enough memory to handle the processes.

# Specific Requirements

This section contains all of the functional and quality requirements of the system. It gives a detailed description of the system and all its features.

## External Interface Requirements

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the software and provides basic prototypes of the user interface.

### User Interfaces

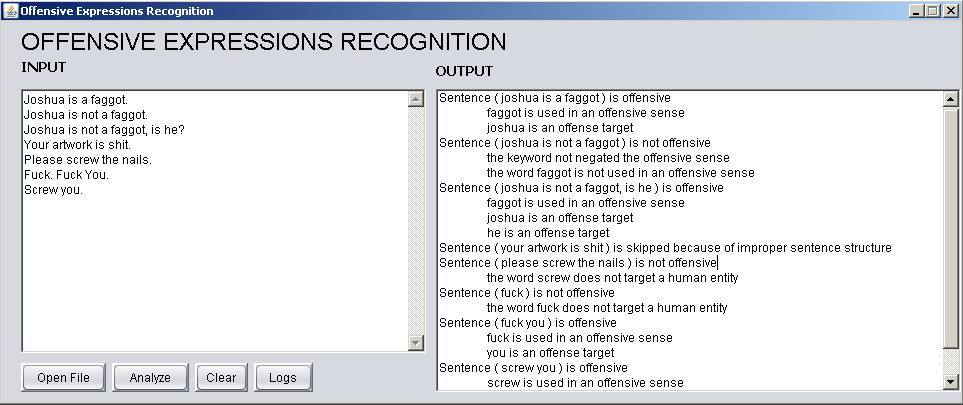
Figure 1- Input Output Form

Figure 1 has textboxes for the input and output. The input is where the user can type sentences or it can be that the user can choose a file from his/her browser. The output is where the results of the input sentence or file will be displayed. There are also buttons below the input section. Open file for opening existing file in the browser, Analyze button for analyzing the given input, Clear button to clear textboxes and Logs for viewing the results in a step-by-step manner.

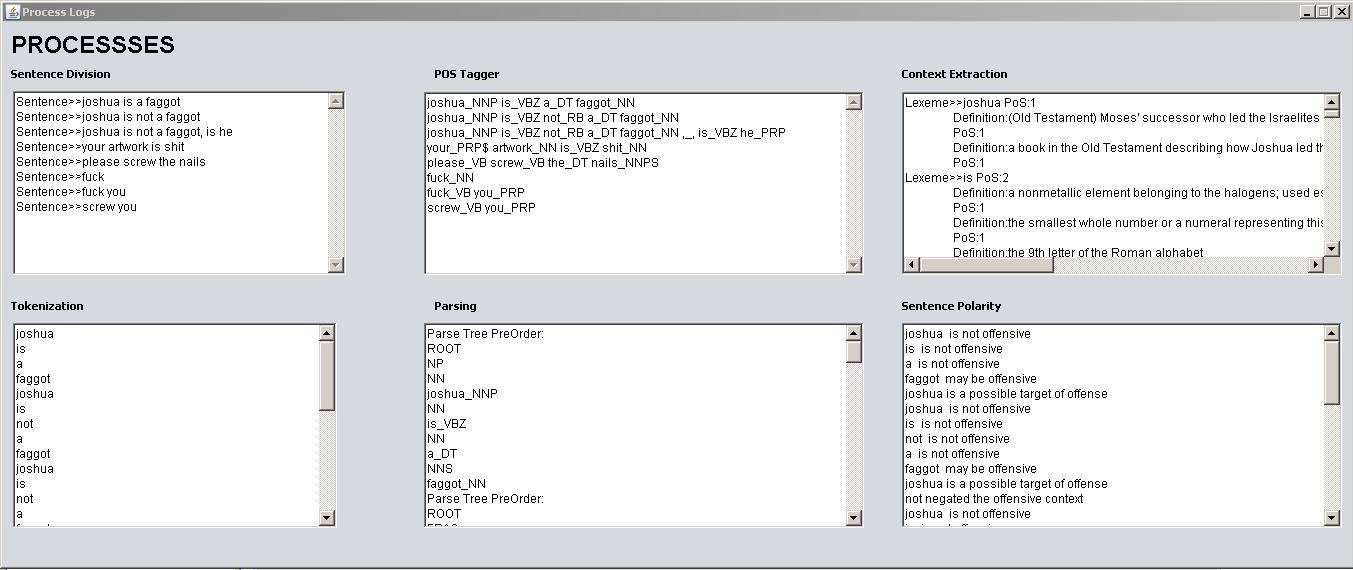
Figure 2-Logs Form

Figure 2, shows how the result in the output in Figure 1 has been determined. This is where the procedure of analysing the given input resides or how the system will come up to those results.

### Software Interfaces

The application can be accessed by the researchers and clients.

The system will analyze the given data provided by the clients and researches whether they are offensive or not.

Enough memory is required for running the system. The internet connection is not required and only need to install the requirements needed by the system.

## Functional Requirements

This section includes the requirements that specify all the fundamental actions of the software system.

**3.2.1 *Functional Requirement 1.1***

**ID:FR1**

TITLE: Download WordNet 2.1

DESC: A user should be able to download a WordNet 2.1 because this serves as the dictionary of the system.

RAT: In order for a user to download the WordNet 2.1

DEP: None

**3.2.2 *Functional Requirement 1.2***

**ID:FR2**

TITLE: Download Stanford Parser

DESC: A user should be able to download Stanford Parser in order to parse the given sentence.

RAT: In order for a user to download the Stanford Parser

DEP: None

**3.2.3 *Functional Requirement 1.3***

**ID:FR3**

TITLE: Download Stanford POS Tagger

DESC: A user should be able to download a Stanford POS Tagger to get POS of each sentence.

RAT: In order for a user to download the Stanford POS Tagger

DEP: None

**3.2.4 *Functional Requirement 1.4***

**ID:FR4**

TITLE: Download Java 8

DESC: A user should be able to download the latest version of Java which is Java 8.0

RAT: In order for a user to download the Java 8

DEP: None

**3.2.5 *Functional Requirement 1.5***

**ID:FR5**

TITLE: Input Sentence or Upload File

DESC: A user should input a sentence or may upload a file containing comments.

RAT: In order for a user to input a sentence or upload a file

DEP: None

**3.2.6 *Functional Requirement 1.6***

**ID:FR6**

TITLE: Sentence Division

DESC: The input will be segmented into sentences.

RAT: In order to divide the given input if the system detected punctuation marks except comma

DEP: FR5

**3.2.7 *Functional Requirement 1.7***

**ID:FR7**

TITLE: Tokenization

DESC: Divides text into sequence of tokens

RAT: In order to tokenized given input

DEP: FR6

**3.2.8 *Functional Requirement 1.8***

**ID:FR8**

TITLE: Part of Speech Tagging

DESC: Marking up a word in a text as corresponding to a particular part of speech

RAT: In order to give the corresponding part of speech of each token

DEP: FR7

**3.2.9 *Functional Requirement 1.9***

**ID:FR9**

TITLE: Parsing

DESC: Analysing a string of symbols conforming to the rules of a formal grammar

RAT: In order to divide into grammatical parts and identify the parts and their relations to each other

DEP: FR8

**3.2.10 *Functional Requirement 1.10***

**ID:FR10**

TITLE: Definition Extraction

DESC: Getting the meaning of each token in WordNet

RAT: In order to get the meaning of each token

DEP: FR1, FR9

**3.2.11 *Functional Requirement 1.11***

**ID:FR11**

TITLE: Sentence Polarity Analysis

DESC: Finding the target in the sentence

RAT: In order to find the target of that offensive word

DEP: FR10

**3.2.13 *Functional Requirement 1.13***

**ID:FR13**

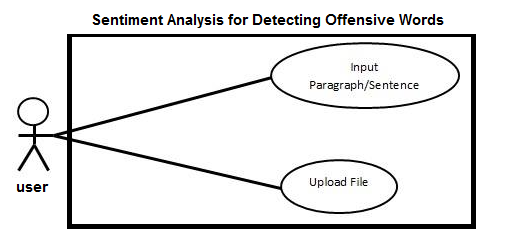
TITLE: Report of Offensiveness per sentence

DESC: Report all bad words in the sentence

RAT: In order to report all bad words

DEP: FR12

## Behaviour Requirements

**3.3.1 Use Case Diagram**

# Other Non-functional Requirements

## Performance Requirements

The requirements in this section provide a detailed specification of the user interaction with the software and measurements placed on the system performance.

**4.1.1 Usage of viewing step-by-step results in text box**

**ID:QR1**

TITLE: Usage of viewing step-by-step results in text box

DESC: The step-by-step procedure in analysing the input of the user can be viewed in separate textboxes.

RAT: In order for a user to view step-by-step results in analyzing input of the user

DEP: none

**4.1.2 Response Time**

**ID:QR2**

TAG: Response Time

GIST: The fastness in analysing given input

SCALE: The response time of analysing input

METER: Measurements obtained from 50 inputs during testing

MUST: No more than 7 seconds 100% of the time

WISH: No more than 5 seconds 100% of the time

## Software Quality Attributes

The requirements in this section specify the required reliability, availability, security and maintainability of the software system.

**4.3.1 Reliability**

**ID:QR3**

TAG: System Reliability

GIST: The reliability of the system

SCALE: The reliability that the system gives the right result in determining what will be the offensive words in the sentence

METER: Measurements obtained from 50 inputs during testing

MUST: More than 98% of the inputs

PLAN: More than 99% of the inputs

WISH: 100% of the inputs

**4.3.2 Availability**

**ID:QR4**

TAG: System Availability

GIST: The availability of the system when it is used

SCALE: The average system availability

METER: Measurements obtained from 50 hours of usage during testing

MUST: More than 98% of the time

PLAN: More than 99% of the time

WISH: 100% of the time

**4.3.3 Maintainability**

**ID:QR5**

TITLE: Application Extendibility

DESC: The application should be easy to extend. The code should be written in a way that it favors implementation of new functions

RAT: In order for future functions to be implemented easily to the application

DEP: none

**ID:QR6**

TITLE: Application Testability

DESC: Test environments should be built for the application to allow testing of the applications different functions

RAT: In order to test the application.

DEP: none

**4.3.2 Portability**

**ID:QR5**

TITLE: Application Portability

DESC: The application should be portable in any Operating System

RAT: The adaptable platform for the application to run on

DEP: none