**SOFTWARE REQUIREMENT SPECIFICATION**

**PROJECT TITLE :**

**“TWITTER SENTIMENT ANALYSIS”**

**TEAM MEMBERS:**

**Atchaya.S Dheekshitha.D Anitha Rajam.M**

**SOFTWARE REQUIREMENT SPECIFICATION**

**1. Introduction**

The SRS is produced at the culmination of the analysis task. The function and performance allocated to software as part of the system engineering and refined by establishing a complete information description, a detailed functional description, a representation of system behavior, indication of performance requirements and design constrains, appropriate validation criteria and the other information related to requirements.

**1.1. Purpose:-**

Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.

**1.2. Scope:-**

Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.

**1.3. Definition,Acronyms,Abbreviations:-**

DFD: - Data Flow Diagram

SRS: - Software Requirement Specification.

**1.4. Reference:-**

<https://www.growthaccelerationpartners.com/blog/sentiment-analysis/>

https://www.figure-eight.com/**data-for-everyone**

**1.5. Overview:-**

People have always had an interest in what people think, or what their opinion is.  Since the inception of the internet, increasing numbers of people are using websites and services to express their opinion.  
With social media channels such as Facebook, LinkedIn, and Twitter, it is becoming feasible to automate and gauge what public opinion is on a given topic, news story, product, or brand.  
Opinions that are mined from such services can be valuable. Datasets that are gathered can be analyzed and presented in such a way that it becomes easy to identify if the online mood is positive, negative or even indifferent.  
This allows individuals or business to be proactive as opposed to reactive when a negative conversational thread is emerging.  Alternatively, positive sentiment can be identified thereby allowing the identification of product advocates or to see which parts of a business strategy are working.

**2. OVERALL DESCRIPTION**

**2.1. Product perspectives:-**

The recent explosion in data pertaining to users on social media has created a great interest in performing sentiment analysis on this data using Big Data and Machine Learning principles to understand people's interests. This project intends to perform the same tasks. The difference between this project and other sentimnt analysis tools is that, it will perform real time analysis of tweets based on hashtags and not on a stored archive.

Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.

**2.2. Product Function:-**

• Collect tweets in a real time fashion i.e. , from the twitter live stream based on specified hashtags

• Remove redundant information from these collected tweets.

• Store the formatted tweets in MongoDB database

• Perform Sentiment Analysis on the tweets stored in the database to classify their nature viz. positive, negative and so on.

• Use a machine learning algorithm which will predict the ‘mood’ of the people with respect ot that topic.

**2.3. User Characteristics:-**

Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.

**2.4. General Constraints:-**

Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).

**2.5. Assumption and Dependencies:-**

List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).

**3. SPECIFIC REQUIREMENTS**

**3.1. External Interface Requirements:-**

**3.1.1. User Interface:-**

Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.

**3.1.2. Hardware Interface:-**

Describe the logical and physical characteristics of each interface between the

software product and the hardware components of the system. This may include

the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.

**3.1.3. Software Interface:-**

Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.

**3.1.4. Communication Interface:-**

Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.

**3.2. Performance Requirements:-**

If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.

**3.3. Security Constraints:-**

Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.