**Software Requirements Specification**

**Agile TweetViz**

**Version 2.0 approved**

**Anoop Jatavallabha Vijayakumar**

**Gowtham Nayak**

**Harsha Kedekar**

**Seema Suresh**

**Yash Chopra**

**GeekO!**

**11/19/2015**

**Table of Contents**

**Table of Contents**

**Revision History**

**1. Introduction**

1.1 Purpose

1.2 Product Scope

1.3 References

**2. System Features**

2.1 Product Perspective

2.2 Product Functions

2.3 User Classes and Characteristics

2.4 Operating Environment

2.5 Design and Implementation Constraints

2.6 Assumptions and Dependencies

2.7 User Interfaces

2.8 Software Interface

2.9 Communication Interface

2.10 System Feature 1

**3. Other Nonfunctional Requirements**

3.1 Performance Requirements

3.2 Security Requirements

3.3 Software Quality Attributes

**4. Other Nonfunctional Requirements**

**Appendix A: Glossary**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| SRS | 10/19/2015 | Baseline | 1.0 |
| SRS | 11/19/2015 | Final modifications after integration | 2.0 |

# I**ntroduction**

## **Purpose**

The purpose of this document is to present a detailed description of the Components of Agile TweetViz. It will explain the high level design, purpose and features of the system along with the interfaces of the system, what the system will do, the constraints under which it must operate. This document is intended for both the stakeholders and the developers of the system

## **Product Scope**

The product has 3 major components viz. Tweet Fetch, Meta model creation & Analysis and Visualization of the data. The project has a wide scope of analyzing user opinion using tweets from twitter on a particular topic to draw facts.

## **References**

IEEE recommended practice for SRS, IEEE Std 830 -1998.

# System Features

## **Product Perspective**

This is a new self-contained product. The system is composed of 3 components.

* Tweet fetch module.
* Meta model and analysis.
* Visualization of analysis.

## **Product Functions**

The product must do the following,

* Must take user statement input or hashtag
* Must parse the statement to gather hashtags or build hashtags
* Use these hashtags to fetch tweets from twitter
* Create meta-models and analyzing tweets
* Visualization of the analyzed data for pictorial representation of results.

## **User Classes and Characteristics**

Three main classes were identified with following attribute.

Class Tweet:

hashtag

user

tweet\_text

tweet\_creation\_time

tweet\_location

tweet\_favorite\_count

tweet\_retweet\_count

Class Location:

Name

Latitude

Longitude

Class User:

Username

favourites\_count

following

location

verified

followers\_count

friends\_count

created\_at

## **Operating Environment**

* 2GB ram
* Intel Pentium 4 processor or higher.
* Compatible with following Operating systems: Microsoft Windows 7 or higher / Linux.
* Hard disk space of 10 GB.

## **Design and Implementation Constraints**

* The tweets than can be fetched can only be up to 2 years old.
* Each tweet operation can fetch data with size 200 tweets per page and 1500 such pages
* Location of user is dependent on user enabling the location settings on their devices
* We cannot distinguish between a new tweet and a retweet in a fool proof way. By using the tag ‘RT’ we are currently dong it.
* We can fetch up to 30,000 tweets after which a wait period of 15 minutes is forced on the API.

## **Assumptions and Dependencies**

We are fetching tweets based on hashtags.

User inputs hashtags.

The project needs python,moose and Pharo installed on the system.

## **User Interfaces**

We will be providing the user with a textbox where they can type the topic they want to visualize. The UI provides a Submit button, on clicking this the process starts. In case the user wants to abort the process we have provided a Stop button, on clicking this the process comes to halt.

## **Software Interfaces**

The user must be familiar with Windows/Linux, Pharo, Moose.

## **Communications Interfaces**

Not Applicable

## **System Feature 1**

4.1.1 Description and Priority

* Visualization should be created based on the analysis made on a particular topic of interest
* Dynamic tweet fetch is not mandatory. Getting a user understandable visualization from the meta-model is a priority.

4.1.2 Stimulus/Response Sequences

* On command prompt input, the keywords must be converted to hashtags
* Then the tweets related to the provided hashtags should be fetched and saved as a .CSV file.
* The specified visualization should be displayed based on the classes generated on moose using the .CSV file.

4.1.3 Functional Requirements

* The system should be able to pull tweets based on the search category provided by the user
* The fetched tweets must be analyzed and displayed in the form of a visualization pattern that helps the user to draw facts

# **Other Nonfunctional Requirements**

## **Performance Requirements**

* System should be able to do repeated tweet fetches within a specified amount of time
* Analysis and Visualization should not take more than a minute to complete

## **Security Requirements**

* Tweets must be fetched and stored only after the user inputs the query unless otherwise asked for explicit timely fetches of tweets.
* Users should get access to visualization but not the meta-model and the integrity and details of an individual tweeter should be discreet.
* User should not be able to break into the system and alter the data.

## **Software Quality Attributes**

* The system should be responsive to user inputs - Hashtag Search (Responsive)
* The system should return valid tweet responses (Reliability)
* The system should

# **Other Requirements**

* Agile Software Process should be followed to see the project to completion
* Python should be used to fetch tweets and convert to .CSV files
* MySQL should be used to save the fetched tweets
* Moose should be used to generate and analyse meta-models from the .CSV file
* Roassal should be used to visualize the analysed tweets.

**Appendix A: Glossary**

1. Pharo - Smalltalk based object-oriented programming framework
2. Moose - Smalltalk based analysis engine
3. Roassal - Smalltalk based Visualization engine
4. MySQL - An open source RDBMS (Third-party)
5. Visualization - Visualization is the process of representing abstract business or scientific data as images that can aid in understanding the meaning of the data.
6. Metamodel - Class diagram representation generated from the .CSV file that is used in analysing a bunch of tweets
7. Analysis - Detailed examination of the elements or structure of something, typically as a basis for discussion or interpretation.
8. Tweepy - Python wrapper for twitter REST API (Third-party API)
9. NLTK - Natural language processing library in python.
10. py2 exe - Used to generate executable file from python script (Third Party)