**Software Requirements Specification**

March 2, 2018

Team 1

(Ricky Davidson, Daniel Lammens, Sullivan Molinek, Kat Seitz, Ryland Wheliss)

This report describes Team 1 Software Requirements for its project to develop a web app that completes twitter trend analysis based on a user inputted key-word.

1. Introduction
   1. **Purpose**

The purpose of this webapp is to aid users to find the current and long-term popularity of a topic or key word via Twitter tweets.

* 1. **Scope**

This is a new project, so this document will cover the entire project.

* 1. **Definitions, Acronyms, and Abbreviations**

Twitter- Social media platform using microblogging.

API – application programming interface - a system of tools and resources in an operating system, enabling developers to create software applications

Ruby – a dynamic, reflective, object-oriented, general-purpose programming language. It was designed and developed in the mid-1990s by Yukihiro "Matz" Matsumoto in Japan. According to its creator, Ruby was influenced by Perl, Smalltalk, Eiffel, Ada, and Lisp.

Ruby on Rails – (RoR) is a cross-platform Web application framework written in Ruby. The framework was originally authored by David Heinemeier Hansson and researched and further developed by a rail core team of several individual contributors.

HTML – Hypertext Markup Language, a standardized system for tagging text files to achieve font, color, graphic, and hyperlink effects on World Wide Web pages.

CSS – A Cascading Style Sheet is a Web page derived from multiple sources with a defined order of precedence where the definitions of any style element conflict.

Web App – A web application is a client–server computer program in which the client (including the user interface and client-side logic) runs in a web browser.

W3C – W3C standards define an Open Web Platform for application development that has the unprecedented potential to enable developers to build rich interactive experiences, powered by vast data stores, that are available on any device.

rubygems – Libraries and dependencies for ruby and RoR.

* 1. **References**

Ruby On Rails Website:

<http://rubyonrails.org/>

Twitter API:

<https://developer.twitter.com/en/docs>

Original place Idea was Found:

<http://nevonprojects.com/twitter-trend-analysis-using-latent-dirichlet-allocation/>

W3C Standards:

<https://www.w3.org/standards/>

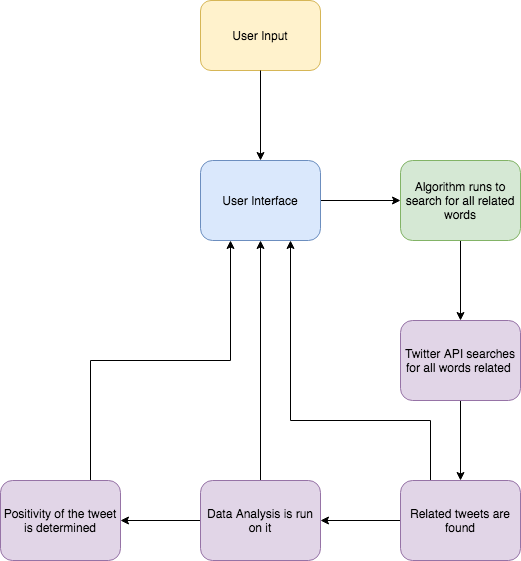
* 1. **Overview**

The project we are developing is a webapp constructed using the language Ruby on the platform Ruby on Rails which uses a Twitter API to collect all of the tweets from the past 30 days and allow the User to search for a key term. The app will then search for all of the tweets with the word specifically in them and all that are related to them. It will report the number of tweets, the positive or negative outlook of the tweets, graphs and diagrams displaying different data from the tweets and the tweets themselves. The app should be a user friendly, simplistic webapp that has no need for logging in or creating a profile. The app responds quickly and allows the user to get the desired data with ease.

1. Overall Description
   1. **Product Perspective**

This webapp is intended to work on popular browsers (chrome, safari, etc.) and be compatible with current platforms. It shall rely on latent dirichlet allocation via a rubygem lda-ruby to find topics to search a Twitter API and then inform users on recent topic trends.

* 1. **Product Functions**



The product will have the following main functions: user input, user interface, the algorithm will run to search for all related words, the twitter API will search for related words, the related tweets will be found, data analysis is run on the words, and finally the positivity of the tweet will be determined.

* 1. **User Characteristics**

The targeted users could be anyone from a business hoping to see what is currently trending to the solitary consumer who is just curious about what someone is tweeting about a certain item. This can be used as both a business and pleasure app and should be handled as such.

* 1. **General Constraints**

Most of the processing done by the app should not be done by the user. Any processing should be done on the server, and processes should be designed to minimize processing time.

* 1. **Assumptions and Dependencies**

This webapp assumes the current Twitter API and depends on the rubygem lda-ruby. Since gem version can be specified, a pessimistic version constraint will be used to prevent dramatic upgrades to the lda-ruby gem affecting the app’s reliability in the future.

1. Detailed Requirements
   1. **External Interface Requirements**
      1. User Interfaces

A search bar to input a topic parameter, a form submit button, and a reset state (return home) button.

* + 1. Hardware Interfaces

None.

* + 1. Communication Interfaces

API Client for Twitter post trend information.

* + 1. Software Interfaces

Browsers:

This webapp is expected fulfill all software requirement specifications on Chrome, Firefox, Safari and Windows Edge. The webapp should extend to other popular browsers as well, but will be tested on these four major browsers.

Operating System:

This webapp should fulfill all software requirement specifications for the specified browsers on the current MacOS (High Sierra at the time of this document), iOS (11 at the time of this document), Android (Oreo at the time of this document) and WIndows 10 and intended to work within requirements on future updates on each device. The webapp is intended to work on mobile devices using Android and iOS without special cases or design. The webapp should be designed to work on mobile devices to prevent issues transferring.

* 1. **Functional Requirements**
     1. Mode 1: Search Twitter Topic Trend

Primary Actor: User

Scope: Someone interested in a topic’s popularity.

* + - 1. Shall display a current Twitter topic trend tweet count.
      2. Shall fetch and display old Twitter topic trend tweet counts over time.
      3. May display sample tweets for the topic.
    1. Mode 2: Management Keyword Database View

Primary Actor: Manager

Scope: Someone interested in the list of auto-updating topics.

* + - 1. Shall display current dictionary of searched topics.
      2. Shall allow deletion of topics in the keyword list.
  1. **Performance Requirements**

Mode 1 should return a response in under 10 seconds 99% of the time. Mode 2 should return a response in under 2 seconds 99% of the time. Throughput should be expected to be over 1000 transactions per second 99% of the time. This is not expected to be a consistent throughput, but the system should be able to handle this constraint for general fluidity.

* 1. **Design Constraints**

The app should conform to W3C standards. Memory is not expected to be a problem due to the scarce amount of information being stored and directly handled by the webapp.

* 1. **Attributes**

The webapp should not have the api keys stored on the server files. Instead, it should be stored in the server environment to keep it safe from prying eyes inclusive of those looking at the source code.