Twitter Brand Analysis

Software Requirement Specification

## Version 1.0

## 2014-12-19

## Name Pending

Andreas Lundkvist

Ian Jenkins  
Kenny Nilsson  
Pontus Hjortskog

Shan Jiang

Tommy Molin

Xiaoqian Xiong

# Table of Contents

## 1. Introduction

## 2. General Description

### 2.1 Product Functions

### 2.2 User Characteristics

### 2.3 General Constraints

### 2.4 Assumptions and Dependencies

## 3. Specific Requirements

### 3.1 External Interface Requirements

### 3.2 Functional Requirements

### 3.3 Quality Requirements

### 3.4 Design Constraints

# 1. Introduction

This is a requirements document for the software project Twitter Brand Analysis (hereafter referred to as TBA). It includes a description of the system and a lists of the core functional and quality requirements.

# 

# 2. General Description

## 2.1 Product Functions

The system should provide the ability to track brands and view information about the public opinion of the tracked brands based on tweets collected from Twitter. The user interface will be provided as a webpage, an android application and an iOS application.

## 2.2 User Characteristics

Target audience of the product is companies with the desire to track public opinion of their brands.

## 2.3 General Constraints

The core of the system must be written in the programming language Erlang and the system must include a database upon which map/reduce functions are used.

## 2.4 Assumptions and Dependencies

It is assumed that the hardware running the system has a running Riak cluster and a web server together with a working internet connection.

# 

# 3. Specific Requirements

## 3.1 External Interface Requirements

The system is reliant on Twitter’s streaming API to retrieve tweets used for raw data.

## 3.2 Functional Requirements

Functional requirements are specified in the following format:

### *3.2.0 FR# - Requirement Name*

*-Description of the requirement*

*-The reasoning behind the requirement*

*-Dependencies*

*-Importance (1-10)*

### 3.2.1 FR1 - Create Account

-Users must have an account with a username and associated password to log in to the service. User accounts will be created by a system admin.

-User accounts are required for using the service

-None

-10

### 3.2.2 FR2 - Log in

-Given that the user has an account, the user must be able to use their username and password to log in to the service.

-Logging in is required to access most user functionality.

-FR1

-10

### 3.2.3 FR3 - Change Password

-Given that the user has an account and has logged in, the user must be able to change their password should he or she want to. This requires knowledge of the old password.

-The ability to change password is a must to any service using passwords.

-FR2

-4

### 3.2.4 FR4 - Reset Password

-Given that the user has an account, the user must be able to reset their password in the event that it was lost. Due to the fact that passwords are encrypted the old password cannot be retrieved, and a new randomly generated one is the be sent instead.

-Users that lose their password must have a way to get a new one.

-FR1

-2

### 3.2.5 FR5 - Add Brand to Watchlist

-Given that the user has logged in, the user must be able to enter new keywords to be searched for and collected by the system.

-This is a core functionality of the system.

-FR2

-10

### 3.2.6 FR6 - Remove Brand from Watchlist

-Given that the user has logged in and has a brand added to their watchlist, the user must be able to remove the brand from the watchlist.

-Brands that are no longer of interest to the user must be able to be discarded.

-FR5

-6

### 3.2.7 FR7 - View Brand Opinion on World Map

-Given that the user has logged in and has a brand added to their watchlist, they should be able to view a map of the world displaying the brands’ health in the countries where tweets have been collected.

-This is a core functionality of the system.

-FR5

-10

### 3.2.8 FR8 - Filter Specific Brand on World Map

-Given that the user is viewing opinions on world map, the user must be able to select a tracked brand filter the world map data to only the selected brand.

-Viewing data about a specific brand can

-FR7

-4

### 3.2.9 FR9 - Display Brand Opinion in Graphs

-Given that the user has logged in and has a brand added to their watchlist, the user should be able to view brand opinion in various graphs.

-Graphs are a good way to get a quick overview of data.

-FR5

-7

### 3.2.10 FR10 - Display Most Commonly Used Words

-Given that the user has logged in, the user should be able to view a collection of the most used words by twitter users when talking about the brands.

-The most used words can give an overview of what people talk about when mentioning the brands.

-FR5

-6

### 3.2.11 FR11 - Filter Most Commonly Used Words

-Given that the user has selected a brand in their watchlist, the user should be able to filter the most commonly used words to only see the words used in positive or negative tweets.

-Filtering to only words used in positive or negative tweets can let the user find good or bad things about their brands.

-FR10

-2

## 3.3 Quality Requirements

Quality requirements are specified in the following format:

### *3.3.0 QR# - Requirement Name*

*-Description of the requirement*

*-The reasoning behind the requirement*

*-Category (Performance, Reliability, Availability, Security, Maintainability, Portability)*

### 3.3.1 QRx - Login Time

-Logging in must take no longer than 5 seconds and should preferably take less than 1 second.

-Logging in should be a quick process. Users have short attention span.

-Performance

### 3.3.2 - Language

-The language processing must be able to handle tweets in the English language.

-Writing NLP for multiple languages is a very advanced process, thus we only adapt it to English.

-Performance

### 3.3.3 - Twitter connections

-The system must be able to collect data for at least 25 users at once.

-The system must support multiple concurrent users.

-Performance

### 3.3.4 - Storage

-The system must not use more than 50 megabytes of hard drive space per day per user.

-Storage costs money, thus the system should consume as little hard drive space as possible.

-Performance

### 3.3.5 QRx - NLP Accuracy

-The system must be able to judge tweets with a 60% accuracy.

-If NLP is not accurate we cannot provide accurate data to the users.

-Reliability

### 3.3.6 QRx - System availability

-The system must have a downtime of less than 3 days per year.

-Users must be able to reach the system when needed. Downtime equals lost money.

-Availability

### 3.3.7 QRx - Password Encryption

-Users passwords must be encrypted to ensure the security of the users’ passwords.

-Password security is important.

-Security

### 3.3.8 QRx - Username availability

-If a username is already in use when attempting to create an account the user must be asked to provide a different username.

-Every user must have an unique username.

-Security

### 3.3.9 QRx - Expanding Servers

-The system must be able to adjust to more nodes in the Riak-cluster should they be added.

-The system must be expandable when needed.

-Maintainability

### 3.3.10 QRx - Updating Code

-The system must support hot-swapping code without a complete restart.

-Complete restarts means downtime, downtime equals lost money.

-Maintainability

## 3.4 Design Constraints

The system must be built primarily using the programming language Erlang. The database must be a NoSQL Riak database, upon which mapreduce functions must be used.