**User Requirement**

**The user needs to see how the application works.**

**The user need to know what platforms are available.**

**Functional Requirements**

F1 Average opinion of a word/brand/whatever in different parts of the world

name:world happiness map

F1.2 The number of tweets that were included in the sample.

F2 The user should be able to see the amount of positive negative neutral tweets per day/week/year.

name:The amount of positive,negative and neutral tweets per 15 min MAP

F2.1 The most representable tweets from positive,negative and neutral tweets

explanation: A new windows will pop up by clicking the bar with tweets content

F3 The client wants to identify what are the negative tweets are about. FEATURE WORDS

name:World Cloud

F4 The client wants to identify what the positve about.FEATURE WORDS

name:World Cloud

what is generally being talked about.i.e. pos/neg/neutral

(tweet’s content, word)

F5 The user can search for brands.

explanation: left column have search key function by typein

F5.2 delete keyword

explanation: click “X” by the side of keyword

F5.3 recommand keyword

explanation: in the right column there will have top 10 command keyword related user stored key words.

F6 The user can track a searched brand.

name: Brand history track

F7 COMPARE different keywords/brands

name:Compare brand

F8 The user must be login to use partly functions

F9 Register

Non Functional

HTTP:

* Will handle the communication with the website and the mobile apps.
* Communication will be using a RESTful api.
* If login will be handle it will be authenticated through the RESTful api and remembered by having cookies with encrypted tokens (this is handled server side and is no concern of the client application other than passing the cookie).
* HTTPS will be enforced upon all connections that will send data or receive sensitive data from/to the server.

Data should be concatenated to save storage space

Message is sent from miner to message\_relay in format {From, {tweet, {{id, Id}, {text, Text}, {timezone, Timezone}}}

Message\_relay retrieves weight from NLP via pyerl.

Tweet is stored in riak with

* {User, Keyword} as bucket
* Tweetid as key
* {tweet, {{text, Text}, {weight, Weight}, {timezone, Timezone}}} as value
* Secondary indexes:
  + datetime, which is the time of insertion in ISO 8601-format
  + location, which is the timezone

Every 5 minutes all tweets from 5 minutes ago are concatenated and stored:

* Worldmap data:
  + {User, Keyword, worldmap} as bucket
  + {Datetime, Location} as key
  + {Avg. weight, Amount} as value
  + Secondary indexes:
    - datetime
    - location
* 100 most used words:
  + {User, Keyword, words} as bucket
  + Datetime as key
  + [{Word, Times\_used}, ...] as value
  + Secondary indexes:
    - datetime
* 50 most used hashtags:
  + {User, Keyword, hashtags} as bucket
  + Datetime as key
  + [{Hashtag, Times\_used}, ...] as value
  + Secondary indexes:
    - datetime
* 50 most mentioned users:
  + {User, Keyword, users} as bucket
  + Datetime as key
  + [{User, Times\_mentioned}, ...] as value
  + Secondary indexes:
    - datetime
* Amount of positive, neutral and negative tweets:
  + {User, Keyword, amount} as bucket
  + Datetime as key
  + [{positive, X}, {neutral, Y}, {negative, Z}] as value
  + Secondary indexes:
    - datetime
* A few sample positive and negative tweets? (To be implemented)

*Data is retrieved from riak with webserver displayed on front-end blah json blah blah maps blah blah bar-graphs blah.*

Store login data in ‘passwords’ bucket with username as key and encrypted password as value.

Store users’ tracked brands in ‘watchlists’ bucket with username as key and a list of tracked brands as value.

Logo