Team Bricksuqad

**Final Version**

Design Document

April 28, 2014

Table of Contents

[1 Introduction 3](#_Toc381026458)

[*1.1* *Project Overview* 3](#_Toc381026459)

[*1.2* *Scope* 3](#_Toc381026460)

[*1.3* *Data Dictionary* 3](#_Toc381026461)

[2 Design Considerations 4](#_Toc381026462)

[*2.1* *General Constraints* 4](#_Toc381026463)

[3 System Architecture 4](#_Toc381026464)

[4 Detailed System Design 6](#_Toc381026465)

[5 Development Methods, Tools & Usage 6](#_Toc381026466)

[*5.1 Development Methods 6*](#_Toc381026467)

[*5.2 Tools 7*](#_Toc381026477)

[*5.3 Programming Languages* 8](#_Toc381026482)

[6 Data Design 9](#_Toc381026486)

[*6.1 External Data Objects 9*](#_Toc381026487)

[7 Interface Design 10](#_Toc381026488)

[*7.1* *Initial Application Startup* 10](#_Toc381026489)

[8 Revision History 10](#_Toc381026492)

1. **Introduction**

## *Project Overview*

BrickSquad has taken on the task of constructing a general-purpose Tweet Analyzer. General purpose meaning a technical or non-technical customer could use custom or written classes to link to the Tweet analyzer to perform such simple or complex analysis of Tweets. The Tweet Analyzer program will contain a minimum of three classes: The general-purpose Tweet Analyzer class that uses the methods in its class to analyze Tweets; A regular expressions class that produces a list of Tweets that matches specified keywords; and the weather map class that analyzes real-time Tweets across the U.S in order to produce a weather map. After the analyzer is developed BrickSquad must then create a Software Development Kit in order to assist the sophisticated customers that want to write their own Tweet Analyzer class. This document includes information on the resources, design plan, risk management, team project schedule, iteration project planning, and our milestones.

## *Scope*

The scope of this project is construct a general-purpose Tweet analyzer that will work will simple classes complex classes. The Tweet analyzer program must be able to link to the Tweet analyzer class using regular expressions to find matching Tweets. In addition to, the program must also be able to link to the weather class that will produce a weather map based on Tweets in the United States. This is both for personal as well as business use.

Scope of BrickSquad Tweet analyzer program:

* Analyzes Tweets
* Filters Tweets to provide only those matching regular expressions that are read from a file
* Produce the list of Tweets that match the regular expressions
* Produce a real-time weather map of the United States based on analyzed and filtered tweets based on a specified list of regular expressions

The target audience of BrickSquad application is estimated, but not limited to, the general population; this application is projected to be used by a variety of people, but we want it to be user-friendly enough to a degree that even those with little technology experience can use this program. This Tweet analyzer is to be very simple and to use.

3

## *Data Dictionary*

* API (Application Program Interface) – a programming language and messaging format that allows two different applications to communicate, or interface with each other
* JSON (JavaScript Object Notation) -is a lightweight data-interchangeable format
* Twitter- an application that is used to express a thought.
* Tweet-a message that a Twitter user posts to his or her profile page.
* Follow - to subscribe to another user's Tweets.
* ReTweet - to forward a Tweet posted by someone else to all of your followers.
* Reply - a response to an existing Tweet, posted by clicking the "reply" button. The reply text automatically begins with @username (the username of the person to whom you are replying). All replies to a user's Tweets are logged under the @Mentions tab on the user's home page.
* Mention - a Tweet update that contains @your username anywhere in the body of the Tweet. Both new Tweets and replies can be considered mentions.
* Direct messages (DM)- Messages to another users that can only be seen between the two users
* OAuth- authentication protocol to provide authorized access to its API
* Regular expressions - A regular expression is a sequence of characters that forms a search pattern, mainly for use in pattern matching with strings, or string matching.

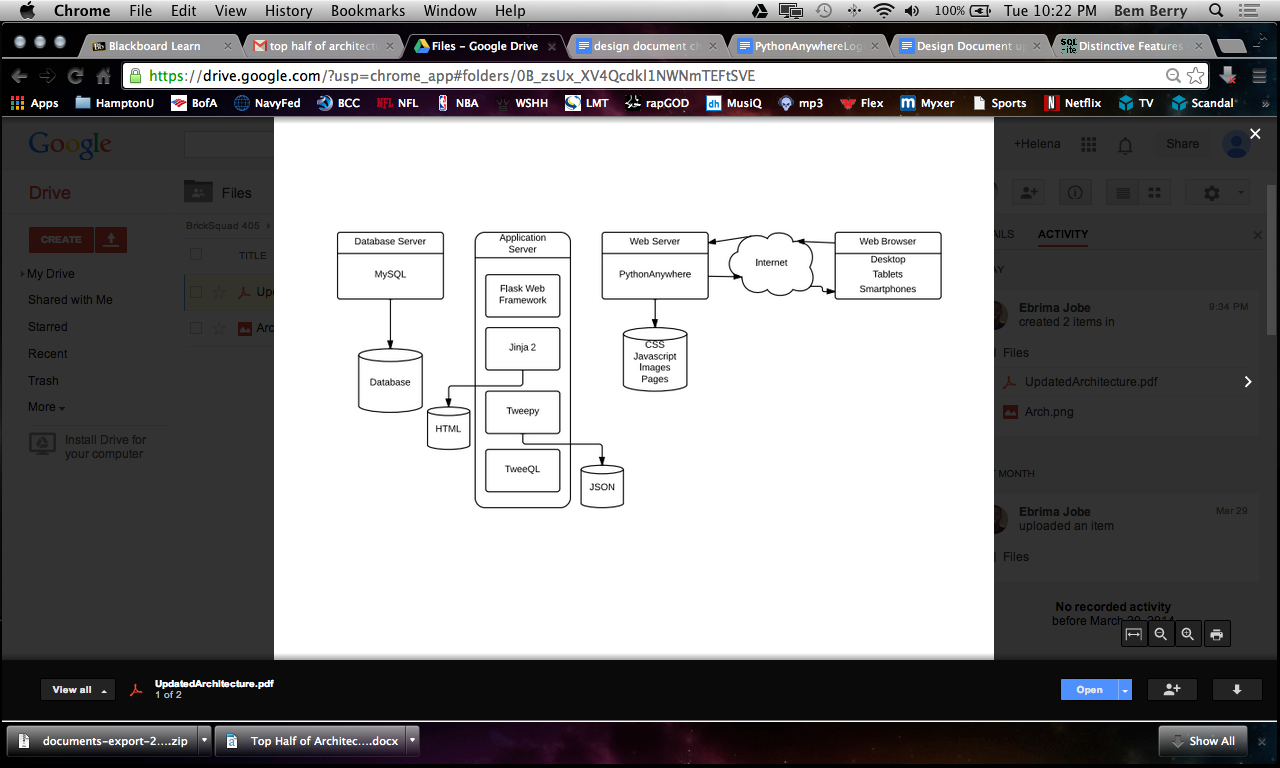
# Design Considerations

* Maintaining a consistent internet connection is necessary for the application to stream tweets from Twitter
* Twitter.com needs to be up and constantly running
* Google Maps needs to be up and constantly running

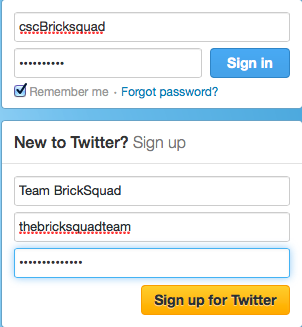
## *General Constraints*

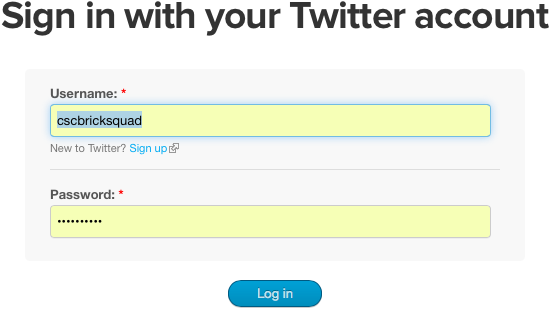
* **System compatibility** – the application will only be compatible with Windows operating systems.
* **Internet reliability** – In order to produce real-time Tweets, this application must be in the vicinity where there can always be a consistent access to the internet.
* **Performance constraints**– The performance and behavior of the application is extremely vital to this product, filtering Tweets across the U.S could greatly affect the speed of your application if not accounted for. Coupling could also generate problems and lead to software not performing to the specification, and result in errors in the application.
* **Multiple designers** – Although distributing the workload is vital in any group project, integrating multiple designers coding implementation into a final product will force time being spent to make the individual pieces work as one project.

# System Architecture

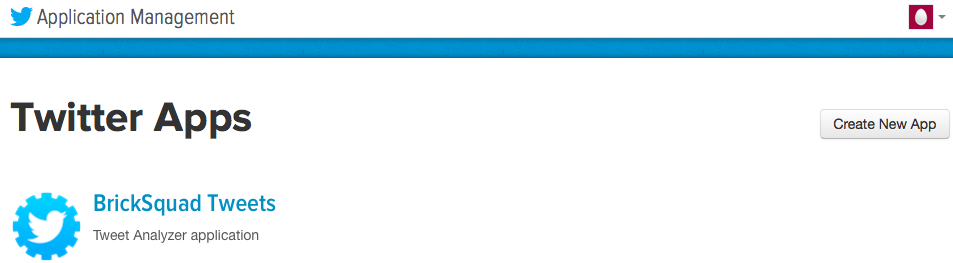


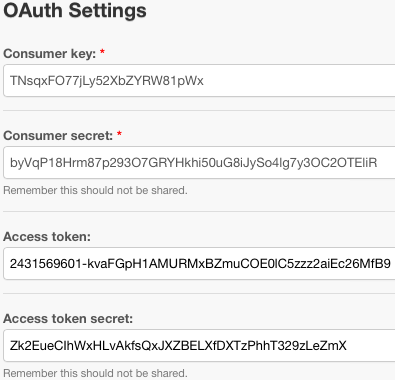
Tweepy is a Twitter API. This API class provides access to the entire twitter RESTful API methods as well as grants us the HTTP request after proper authorization. Each method defined in the API can accept various parameters and return responses. Bricksquad will use this API to carry out required commands that obtain information about each relevant Tweet on Twitter. This API will be able to use the Twitter API that uses the REST format to provide the relevant tweets in order to search the past for information such as terrorism attacks. It also allows you to use the Streaming API that produces real time tweets once the connection is established and authorized in order to provide current information.





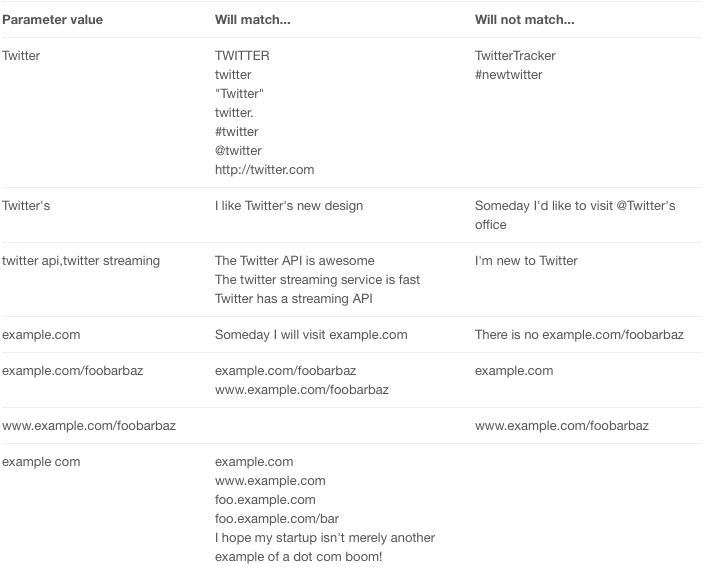
In order to gain access to the Twitter API, a Twitter account has to be created in order to login to the Twitter Developers (<https://dev.twitter.com>) website. The account allows twitter to approve of the application being created, and gives us the necessary credentials to authorize access to Twitter’s tweets.





After you create an application, which informs Twitter of your purpose for needing access to its Twitter feed, Twitter allows you four OAuth settings that authorizes your application access to Twitter’s API. Your signed request both identifies your application's identity in addition to the identity accompanying granted permissions of the end-user you're making API calls on behalf of, represented by the user's access token.

The POST statuses/filter is an HTTP method for streaming endpoint that returns public statuses, in JSON format, that match one or more filter predicates. Multiple parameters may be specified which allows most clients to use a single connection to the Streaming API. The parameters specify which tweets to return, but only one parameter type is required to be specified in order to instruct what the API returns to the endpoint. The parameter BrickSquad is using for the Tweet analyzer application is the track parameter. The track parameter allows the programmer to specify which keywords to track. Phrases or keywords are specified by a comma-separated list. A phrase may be one or more terms separated by spaces, and a phrase will match if all of the items in the phrase are present in the Tweet, regardless of order and ignoring case. Commas are regarded as logical ORs, while spaces are equivalent to logical ANDs.



When the main class receives the Tweet from Twitter using the API commands, the tweet is returned in a JSON format. This format is used to be easily interchangeable between languages, and can be deciphered once you parse through the format in a language dependent method. In Python to parse through JSON format there are many different implementations you can use. One of the advantages with Python is the ability to extend the capabilities by building and installing third-party Python modules and extensions. Although Python’s extensive standard library covers many programming needs, there often comes a time when you need to add some new functionality to your program. This will be helpful to support our own programming, or to support an application that you want to use and that happens to be written in Python. For this specific project we will be using the simplejson. Simplejson is a simple, fast, complete, correct and extensible JSON encoder and decoder for Python. The decoder will handle incoming JSON strings of the tweets we will be fetching and will help gather only the useful data that the Bricksquad team will use from the tweet text, username and retweet count. The number of retweets a tweet is a measure of engagement. The more users who retweets a tweet the more valuable that tweet is.

An algorithm is used to handle the heavy stream of tweets. This algorithm is responsible for fetching and processing tweets. Each tweet process will recheck the API’s filter in order to ensure the exact regular expression’s match, as well as storing the proper information needed to be obtained by each tweet in the correct place. After each tweet is arranged accordingly, the tweet will then be stored in MySQL.

A class must be generated for non-sophisticated users looking to find a visual list tweets relating to multiple key-words (regular expressions). This class will obtain those keywords by reading in a file containing the regular expressions. After reading the file, each line, containing a regular expression, will be parsed to an array. This array will be used to identify exactly how many times the keyword has been used, as well as for indexing purposes. This list will then been transferred to the analyzer class in order to define the keywords before the twitter search process begins. After each tweet is found an alert must be sent to display the list of found regular expressions that is constantly updated.

To produce a visual list of the tweets found using the regular expressions class, an interface is needed for the user to view them. Flask is a free, micro framework written in Python Graphical User Interface toolkit that BrickSquad has elected to use to display said Tweets.

Another class must be generated for sophisticated users. This class must use a list of predetermined regular expressions from all over the Continental United States and produce a weather map for the country. The weather component of this project will integrate with tweets relating to the weather. Using the Google Maps JavaScript API Bricksquad is able to implement an algorithm that fetches JSON text for different weather conditions in the Continental United States. The Tweet Analysis class will perform some kind of analysis for tweets about the weather and store this information in JSON format for the Google Maps API to fetch. Once it fetches this information it will iterate through an array of forecasts in every state and populate the weather conditions on its own layer that Google allows us to create and customize. Google already has a weather layer that uses information from weather.com. Bricksquad can enable the display of weather data or cloud imagery on your map via the WeatherLayer and CloudLayer classes of the google.maps.weather library. Enabling the cloud layer will add cloud coverage imagery to your map, visible at zoom levels 0 through 6. Enabling the weather layer will show current weather conditions from weather.com on your map, including icons that denote sun, clouds, rain and so on. With the WeatherLayerOptions object you can disable info windows, configure the color of the labels displayed on the weather layer, and customize the units used to display temperature (degrees Celsius or Fahrenheit) and wind speed (km/h, mph, or m/s).

# Detailed System Design

* 1. *Definition*

The purpose of this application is to provide a customer with a more refined twitter search than the one provided at the website (Twitter.com). Twitter search is a search engine that searches information from Twitter so you can see the most recent updates related to any topic. The user enter their search query into the search box at the top of the page, then results will show a combination of everything, including people, Tweets, and more. When viewing search results, you are given the option to choose between viewing Top (REST API) or All (Streaming API) results by selecting the options at the top of your search results timeline.

BrickSquad’s application refines the search by using regular expressions, which allows a multiple query search simultaneously. The application also refines the search by them checking for relations between the multiple regular expressions and displaying results for the tweets that match the most. This application provides use to customers ranging from either a technically sophisticated customer could write their own class that would be linked with the general purpose Tweet Analyzer to perform a very sophisticated form of analysis or the non-technical customer could use a “team written” class that accepts regular expressions written by the customer to perform a simple match analysis of tweets.

# Development Methods, Tools & Usage

BrickSquad’s application is composed of many open source tools available in multiple places on the World Wide Web. These tools combined work seamlessly to present a single project. The section below will provide a description of each tool and how BrickSquad used it to tie together a single project.

## *Development Methods*

### Streaming Tweets

### BrickSquad has been assigned the task to develop a general purpose Tweet analyzer program that uses methods to analyze Tweets. In order to perform this function, a connection has to be made to the streaming API. To connect to the Streaming API, form a HTTP request and consume the resulting stream for as long as is practical. Our servers will hold the connection open indefinitely, barring server-side error, excessive client-side lag, network hiccups, routine server maintenance or duplicate logins. The method to form an HTTP request and parse the response will be different for every language or framework, so consult the documentation for the HTTP library you are using. After the connection has been established, the program must then have a way to search through Twitter. Streaming APIs is offered by Twitter gives a low latency access to twitter’s global stream of Tweet data. Public Streams are for of Streaming APIs that streams the public data flowing through Twitter. The application gets delivered a feed of Tweets, once a connection is established to a streaming endpoint. An HTTP method for streaming endpoints is called POST statuses/filter. The endpoints for this method returns public statuses, in JSON format, that match one or more filter predicates.

### Filter Tweets

### The Tweet Analyzer program is required to filter the stream of Tweets to provide only those including a regular expression. In order to filter each Tweets, BrickSquad plans to use the POST request because the GET request are often rejected due to excessive URL length. One of the predicate parameters in the POST request is the Track. The Track Parameters are keywords and phrases (regular expressions) that are specified by a comma separated list.

### Store Tweets

BrickSquad will be using MySQL in order to store the Tweets we are streaming real-time from Twitter. MySQL is a relational database management system (RDBMS), and ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. This software has been proven to work with our chosen IDE, Eclipse, as well as has been proven to comply with our chosen programming language, Python.

### Parse regular expressions

Part of the requirements asks BrickSquad to develop a regular expressions class that reads in a file to produce a list of regular expressions. To produce this list, Bricksquad plans to parse through this file and produce an array that gives each regular expression its own index for easy access when looking for matches

### Produce list of relevant tweets (tweet matching keywords):

The regular expressions class for the non-sophisticated users will produce a list of Tweets that match any of the regular expressions. To produce this list BrickSquad plans on using Flask as a GUI toolkit, to create neat and professional list of relevant Tweets.

### Display map of real-time US weather map based on predefined regular expressions:

Google Maps provides an API for Bricksquad to use. With Google Maps Bricksquad has the ability to design and implement a custom map that displays weather conditions on the US map for specified regions/cities/states. It already provides a weather layer with weather updates directly from weather.com, but we are also able to customize that information. With the WeatherLayerOptions object Bricksquad can disable info windows, configure the color of the labels displayed on the weather layer, and customize the units used to display temperature (degrees Celsius or Fahrenheit) and wind speed (km/h, mph, or m/s). After fetching and processing the Tweets related to the weather, Bricksquad will develop an algorithm, using JavaScript, to display the weather for various regions on the US map at real-time.

### Developing a SDK;

Our Software Development Kit (SDK) will assist the user who wants to write their own class in order to implement a Tweet Analyzer class. Software Development kit or sometimes called devkit is a set of software development tools that allows for the creation of application for a certain software package, in this case it is BrickSquad’s Tweet Analyzer class. This will be very necessary in our implementation of BrickSquad’s application.

## Tools

### PythonAnywhere

PythonAnywhere is an online IDE and Web hosting service based on the Python programming language which we plan to use. PythonAnywhere provides in-browser access to server based Python and Bash Command-line interfaces, along with a code editor with Syntax highlighting. PythonAnywhere runs on super-powerful servers hosted by Amazon EC2, which can provide us with heavy-duty processing.

### MySQL

MySQL is a [relational database management system](http://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS), and ships with no [GUI](http://en.wikipedia.org/wiki/Graphical_user_interface) tools to administer MySQL databases or manage data contained within the databases. Users may use the included [command line](http://en.wikipedia.org/wiki/Command_line) tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records

### Flask

Is a micro framework written in Python based on the Werkzeug WSGI toolkit and Jinja2 template engine. It contains development server and debugger, integrated support for unit testing, and RESTful request dispatching. This application framework will produce the list of Tweets that match the provided regular expressions.

### Google Maps

Google Maps is a [web mapping](http://en.wikipedia.org/wiki/Web_mapping" \t "_blank" \o "Web mapping) service application and technology provided by [Google](http://en.wikipedia.org/wiki/Google" \t "_blank" \o "Google), that powers many map-based services, including the Google Maps website, Google Ride Finder, Google Transit, and maps embedded on third-party websites via the Google Maps [API](http://en.wikipedia.org/wiki/Application_programming_interface" \t "_blank" \o "Application programming interface). It also has a wide array of APIs that let you embed the robust functionality and everyday usefulness of Google Maps to quickly produce and develop applications. This will be used to produce the weather map of the United States. Google Maps provides an API for Bricksquad to use. With Google Maps Bricksquad has the ability to design and implement a custom map that displays weather conditions on the US map for specified regions/cities/states. It already provides a weather layer with weather updates, but we are also able to customize that information.

### Jinja2

**Junja2** is a full featured template engine for Python. It has full unicode support, an optional integrated sandboxed execution environment, widely used and BSD licensed. It is a powerful templating engine modeled after Django's templating system. The idea is to separate your logic from your presentation, and make your code clean and well-defined in the process. Bricksquad will be using Jinja2 to communicate with Flask and the MySQL database to process data for our application.

## *Programming Languages*

### Python

**Python** is a widely used [general-purpose](http://en.wikipedia.org/wiki/General-purpose_programming_language" \t "_blank" \o "General-purpose programming language), [high-level programming language](http://en.wikipedia.org/wiki/High-level_programming_language" \t "_blank" \o "High-level programming language).Its design philosophy emphasizes code [readability](http://en.wikipedia.org/wiki/Readability" \t "_blank" \o "Readability), and its syntax allows programmers to express concepts in fewer [lines of code](http://en.wikipedia.org/wiki/Lines_of_code" \t "_blank" \o "Lines of code) than would be possible in languages such as [C](http://en.wikipedia.org/wiki/C_(programming_language)" \t "_blank" \o "C (programming language)). The language provides constructs intended to enable clear programs on both a small and large scale. The Tweet Analyzer class will be programmed using Python

### JavaScript

**JavaScript** (**JS**) is a [dynamic](http://en.wikipedia.org/wiki/Dynamic_programming_language" \t "_blank" \o "Dynamic programming language) computer [programming language](http://en.wikipedia.org/wiki/Programming_language" \t "_blank" \o "Programming language). It is most commonly used as part of [web browsers](http://en.wikipedia.org/wiki/Web_browser" \t "_blank" \o "Web browser), whose implementations allow [client-side scripts](http://en.wikipedia.org/wiki/Client-side_scripting" \t "_blank" \o "Client-side scripting) to [interact with the user](http://en.wikipedia.org/wiki/User_interface" \t "_blank" \o "User interface), control the browser, communicate [asynchronously](http://en.wikipedia.org/wiki/Ajax_(programming)" \t "_blank" \o "Ajax (programming)), and alter the [document content](http://en.wikipedia.org/wiki/Document_Object_Model" \t "_blank" \o "Document Object Model) that is displayed. It has also become common in server-side programming, game development and the creation of desktop applications. BrickSquad will use JavaScript to communicate to the Google maps API and present a real time weather map of the United States weather map.

### SQL

**Structured Query Language**is a [special-purpose programming language](http://en.wikipedia.org/wiki/Special-purpose_programming_language" \t "_blank" \o "Special-purpose programming language) designed for managing data held in a [relational database management system](http://en.wikipedia.org/wiki/Relational_database_management_system" \t "_blank" \o "Relational database management system) (RDBMS). Originally based upon [relational algebra](http://en.wikipedia.org/wiki/Relational_algebra" \t "_blank" \o "Relational algebra) and [tuple relational calculus](http://en.wikipedia.org/wiki/Tuple_relational_calculus" \t "_blank" \o "Tuple relational calculus), SQL consists of a [data definition language](http://en.wikipedia.org/wiki/Data_definition_language" \t "_blank" \o "Data definition language) and a [data manipulation language](http://en.wikipedia.org/wiki/Data_manipulation_language" \t "_blank" \o "Data manipulation language). BrickSquad will use SQL to translate query commands to MySQL

### HTML

**HTML** or HyperText Markup Language is the standard market language used to create web pages. HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets. The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. Bricksquad will use the HTML because of the nature of our application actually being used and accessed on the web.

# Data Design

## *External Data Objects*

Tweepy API – The Twitter API will be used to query Tweet data in real time by using function calls. The Tweet Analyzer class will constantly access the API in order to update the relevant Tweets. The API will retrieve the information in JSON format that will require parsing to obtain the needed data of each relevant Tweet. The data will then call the MySQL storage software to stock each tweet. MySQL will be accessed by the Weather Map class as well as the regular expressions class to obtain these tweets to perform further processing.

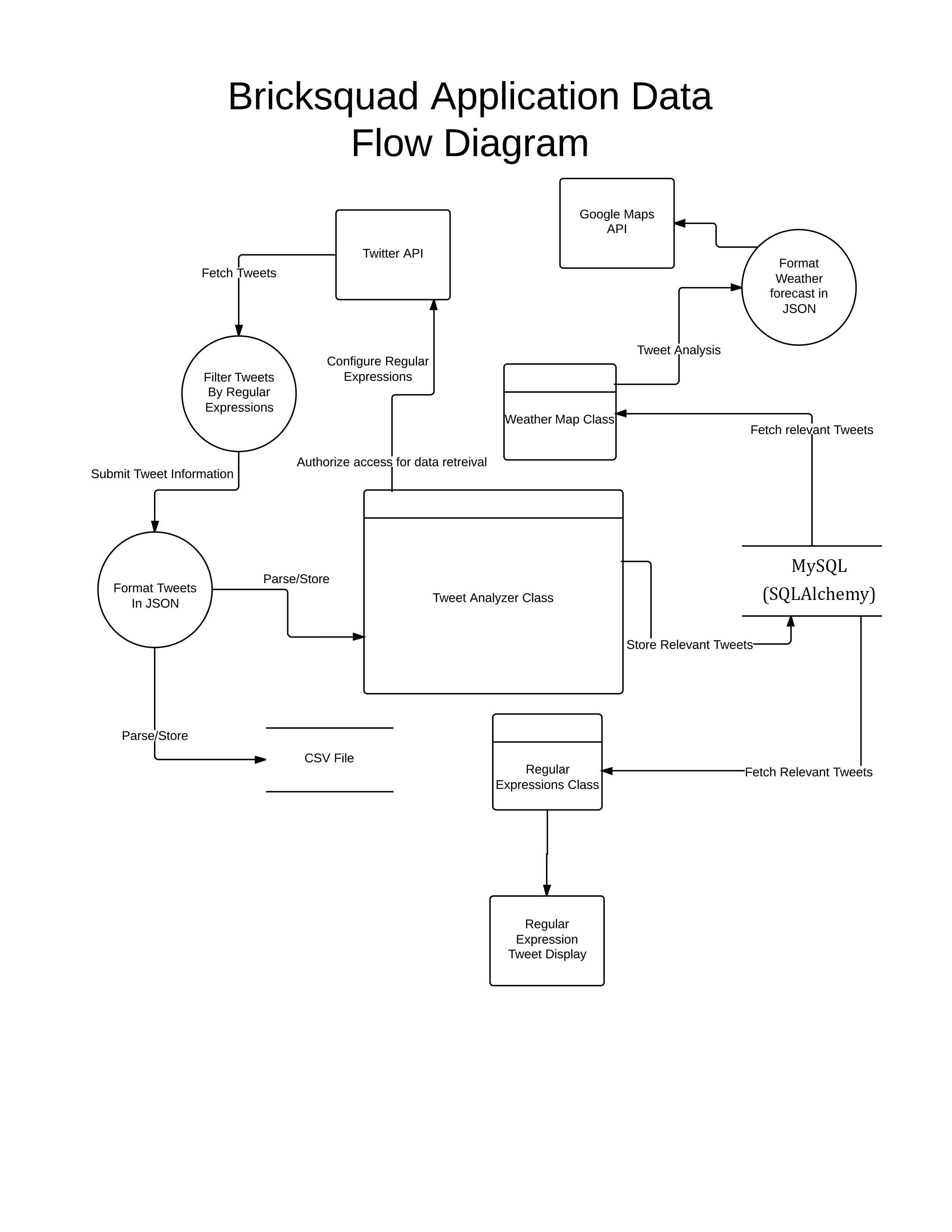


Figure 1: External Data Diagram

# Interface Design

Bricksquad plans to implement an interface design that will be very detailed and intuitive for any new user that is familiar with how to use a computer. The users experience and interaction with the application will be very fluent so that any user can clearly and efficiently get retrieve any information that he or she might desire from twitter. Bricksquad will accomplish this goal by using an interface tool called Flask.

## *Initial Application Startup*

### Regular Expression

In order for this application to start there must be a text file in the specified directory. This text file contains a list of regular expressions. If this text file is not found a window opens to prompt the user to add text file and try the startup process again. The application also test if the text file provided is empty. If the text file is empty the application prompts the user to insert another text file or edit the current file. After the text file has met the criteria the regular expressions are read in and the tweet analysis process begins.

### *Weather Expression*

In order for this application to start there must be a text file in the specified directory. This text file contains a list of regular expressions for the weather map. After text file read-in for the weather map the tweets are analyzed and a weather map is produced.

## 

# Revision History

| **Version** | **Date** | **Name** | **Description** |
| --- | --- | --- | --- |
| **1.0** | **2/17/2014** | **Design Document** | **First document** |
| **2.0** | **3/20/2014** | **Document #2** | **Revised document** |
| **3.0** | **4/08/2014** | **Design document** | **Revised #2** |
| **4.0** | **4/28/2014** | **Design doc** | **Final document** |