# High Level Diagram and High Level design explanation of Agile TweetViz Program

## Very High Level Design

There are three main components of AgileTweetViz product.

1. Tweet Fetch module
2. Tweet Analysis module
3. Tweet Analysis Visualization module

Tweet Fetch Module

Tweet Analysis Module

Tweet Analysis Visualization Module

### Tweet Fetch Module

This module is responsible for fetching tweets based on user instructions and store the tweets in an appropriate medium. Currently this module has the ability to store it in two different ways – One in SQL tables and other is in CSV files. This module is also responsible for making the tweets ready for the Tweet Analysis module. That means if tweet analysis wants the data represented in a particular way then Tweet Fetch module will convert the present tweets into the format expected by the Tweet Analysis Module. Tweet Fetch module is mostly developed in python. We will be explaining it much further at later point of the document.

### Tweet Analysis Module

This module is responsible for analyzing the fetched tweets. This module will be developed using Pharo/Moose framework. Kind of analysis planned is to compare two objects like Windows10 vs OSXElCaptian. For this a meta module specific to that analysis is developed in the Pharo/Moose framework.

### Tweet Analysis Visualization Module

This module is responsible for the visualization of the analysis done on the tweets by previous (Tweet Analysis) module. For visualization we will be using Roasal framework.

## Tweet Fetch Module

As mentioned earlier, this is the module which interacts with user to get the user statement upon which it has to fetch tweets. Once user gives a statement, it parses that statement to generate different hashtags based on that statement. Once it has got all the hash tags it will fetch tweets from the internet. Once sufficient tweets are fetched, it will store those results into DB and also stores in a csv file which can be utilized by the analysis module. There are different classes/sub modules which is part of Tweet Fetch module.

1. Tweet\_UI sub module – This module is responsible for fetching user input from an user interface. As of now plan is to use a web page developed using PHP having very simple UI to get the user inputs. This UI will get the user statement, based on this statement tweets needs to be fetched.
2. Tweet\_Hashtag\_Generator – This module given a statement, it will extract the hashtags from that sentence. This uses Natural Language processing to do this task. Given a sentence, it will extract the nouns from that sentence. Suppose it has extracted 3 nouns Noun1, Noun2 and Noun3 then following hashtags will be generated - #Noun1, #Noun2, #Noun3, #Noun1 #Noun2, #Noun2 #Noun3, #Noun1 #Noun3, #UserStatementWithoutSpaces. This functionality is implemented using class TweetVizHashTagGenerator in file TweetViz\_Hashtag\_Generator.py. This also uses nltk python package.
3. FileReaderWriter – This module will be responsible for writing the tweets information into a file. Similarly given a correct formatted file having tweets, this module can read that file and give it back to the table containing tweet information. This module will interact using CSV file formats. File will have following columns in the given order – Search\_Category, Tweeter\_Hashtag, Tweeter\_Handle, Tweet\_Message, Tweet\_Location, Tweet\_RetweetCount, Tweet\_FavoriteCount. This is implemented by class TweetVizFileReaderWriter in file TweetViz\_File\_ReaderWriter.py
4. Tweet\_Fetch – This module given a set of hashtags is responsible for fetching tweets from internet corresponding to those hashtags. We are using tweepy library in python to fetch tweets from twitter. For each of the hashtags it tries to fetch nearly 100 tweets. That may be a recursive process. Usually some of the tweets fetched may be discarded like if a tweet is just a retweet or if a tweet has an image. Once tweets are downloaded from internet, in each tweet information will be extracted like tweet user handle, tweet message, tweet create datetime, tweeted location, number of retweet counts, number of favorite counts of the tweet. Once all these information is extracted, a table is formed with each of the above mentioned information acting as columns. This is implemented using TweetFetcher class in file Tweet\_Fetch.py
5. Tweet\_Command – This module is responsible running tweet fetch main module in command line. If we want to get a large number of tweets, then it would be better to run the code in command line which will just keep fetching the code and storing it in DB or file. Currently its development is going on.
6. Data\_Storage\_Reader – This module is the interface for the data reading and writing. It can interact with the database. This takes help of FileReaderWriter module to interact with files. It reads the configuration file. We are using MySQL as the DBMS which holds our database TweetViz\_DB. All the tweets are stored in the table Tweet\_Repository. Any module which needs to access the data stored in DB or in File, has to talk to this module. This is implemented by class TweetVizDataStorageReader present in file TweetViz\_DataStorageReader.py file. SQL schema is present in the file Tweet\_Repository\_Table.SQL. Configuraiton is stored in file TweetViz\_ConfigFile.ini file. This file holds configuration for connecting to database. It also has information like CSV file name, where tweet information will be stored. SQL configuration has password which is encrypted and stored.
7. Tweet\_Backend – This is the module which is interacting with every other module and has the work flow of the tweet fetch. It knows the steps to do, to get the tweets and store it in DB/CSV. It also knows what is content format which is expected by Tweet\_Analysis module to do the analysis. Currently its development is going on. This is represented in HashTagReceiver.py file as main\_fun function.

