This is the description of the twitter stream application for Alex Yang's w205 Exercise 2. At it's heart is an Apache Storm topology, run through Streamparse that collects tweets, parses them and stores them in a postgres database. The front end of the application is run via three python scripts that are activated in the command line.

This application requires Streamparse, an implementation of Apache Storm in Python. In order to read tweet data, I use Tweepy, a twitter library for Python to read live tweets. In addition to standard Python libraries, I also used Pandas to sort and slice dictionaries, and Psycopg2, in order to sync with PostGresSQL.

The application is contained within the folder exercise_2, which, along with documentation, contains three Python scripts to manage and interpret the stream. In the folder is also contained extweetwordcount, which contains the streamparse implementation. The topology is contained in the topologies folder while the spout and bolts are contained in the src folder. The spout is connected to Twitter via Tweepy and Twitter's API with four keys generated from the apps.twttier.com website. The spout emits tweets, in the form of tuples, to the bolts, which parse the tweet into text, split up the words and the counts the words. These counts are fed in a Postgres Database called tcount and a table called Tweetwordcount. These databases are managed by a Python script, start.py, contained in the main exercise_2 folder, which both creates the database and data table, and starts the Streamparse process. Finally, there are two other Python scripts, finalresults.py which outputs the counts of words, and histogram.py, which outputs all words with counts within a range.

The architecture of the application follows that of a Storm application, with some add-ons.