Justin Schopick
Kushagra Singh
Chris Sultzbaugh
CS 172
May 4, 2018

Twitter Crawler Report

Collaboration:

We took a collaborative approach to this project. We all helped with each part to a certain extent. We all helped research the implementation of the crawler and storage. Kushagra did the actual coding of the crawler, Justin did the coding of the storage element, and Chris did the report write up. We all participated in the research and then split up the typing.

System Details:

Our crawler uses the Twitter Streaming API in order to collect geolocated tweets. The Twitter Stream functionality uses a "listener" to detect when tweets come in. We set up the system to only gather data of geo-tagged tweets.The data is then stored in files of 10 MB each.

The Twitter API is necessary to gain access to the tweets. Our crawler gathers raw Json tweet data and returns it in correct formatting for storage-use. We used Jsoup to find the title of links within the tweets we parse through. We then use Gson to edit the title of the link onto the end of tweet being parsed to add that title to the data collected.

This system is set up within one package called "crawler." Inside this crawler package, we import the needed API and external libraries. We use one class called "streamTest" that has one main function and a couple of helper functions within it.

Our storage system takes the Json data and appends it to a new or current Json file as long as it is less than 10 MB. If no files under 10 MB exists, the system automatically initializes a new file with the proper formatting. As the Twitter Stream continues, our system continuously adds this data into the now-existing files.

The system uses a global variable to count which document we are on; this avoids any chance of overwriting data or causing storage malfunctions. Whenever a document is greater than or equal to 10MB, the system will append the end formatting to the file. After this, it increments the global variable by 1 to move on to the next document. The way our system is set up, it will add to existing files as well, rather than creating new files every time it begins.

System Limitations:

Using the Twitter Streaming API, limits us to only collect live tweet info. We currently do not have the capability to search older tweet info. Twitter streaming also limits the speed of our data collection as we cannot crawl for geolocated tweets. We have to wait for geo-tagged tweets to come in from the streamer, which is much slower because not everyone puts a location tag on their tweets.

We also have not yet implemented a stopping mechanism. That is once we start the collection process, unless someone or some outside factor stops the collection it continues. We plan to implement a max data collected parameter to correct this oversight.

We also had a problem we fixed that had us stuck for a bit. Our storage system was not working properly when it came to making a new file when data limit was reached for individual files. This caused us to create the global variable mentioned previously to help keep track of the file being created. This helped avoid creating duplicate file names and overwriting files.

Deployment:

Software needed to run our crawler:

- Java Development Kit must be installed in order to compile and run the program
- http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

External Packages Needed(these are all inside of the submission folder):

- jsoup-1.11.3.jar
- gson-2.6.2.jar
- twitter4j-async-4.0.4.jar
- twitter4j-core-4.0.4.jar
- twitter4j-examples-4.0.4.jar
- twitter4j-media-support-4.0.4.jar
- twitter4j-stream-4.0.4.jar

To Run(must be on unix/linux):

- Download the cs172part1_twitter zip file and unzip it to your desired location
- Open your terminal and find the location you stored the cs172part1_twitter directory in
- Open the cs172part1_twitter
- While in the cs172part1_twitter directory, type "./run.sh" into the terminal without quotation marks

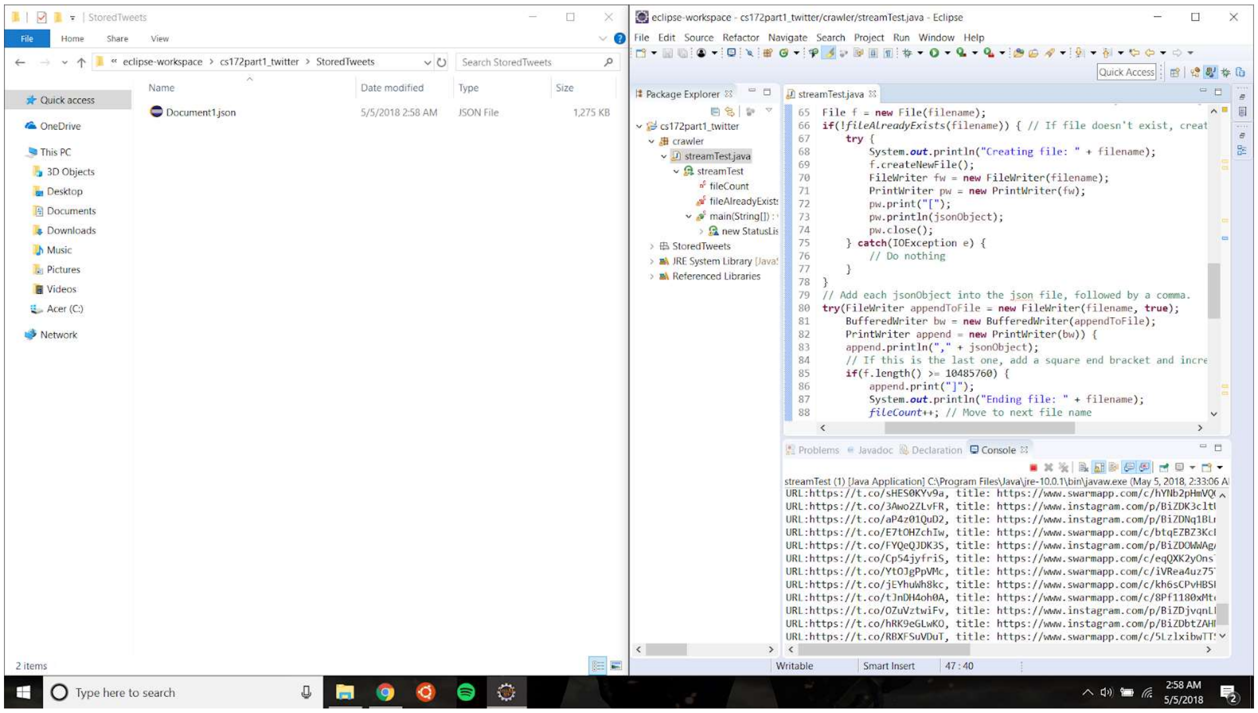
To Terminate the Program:

- On Windows: Control+C
- On Mac: Command+C

Screenshots:

<u>Using the Eclipse IDE</u>

After Running for ~2 Hours:

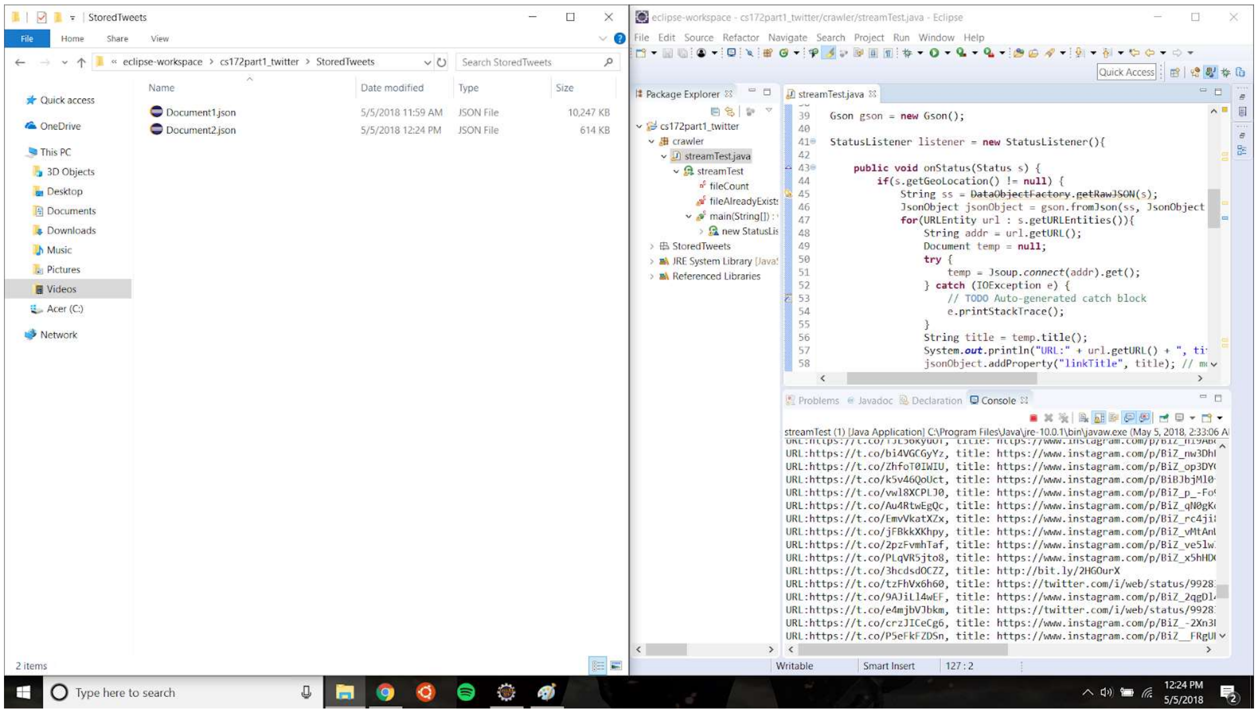


Name	Date modified	Type	Size
Document1.json	5/5/2018 2:58 AM	JSON File	1,275 KB

```java
File f = new File(filename);
if(!fileAlreadyExists(filename)) { // If file doesn't exist, creat
    try {
        System.out.println("Creating file: " + filename);
        f.createNewFile();
        FileWriter fw = new FileWriter(filename);
        PrintWriter pw = new PrintWriter(fw);
        pw.print("[");
        pw.println(jsonObject);
        pw.close();
    } catch(IOException e) {
        // Do nothing
    }
}
// Add each jsonObject into the json file, followed by a comma.
try(FileWriter appendToFile = new FileWriter(filename, true);
    BufferedWriter bw = new BufferedWriter(appendToFile);
    PrintWriter append = new PrintWriter(bw)) {
    append.println("," + jsonObject);
    // If this is the last one, add a square end bracket and incre
    if(f.length() >= 10485760) {
        append.print("]");
        System.out.println("Ending file: " + filename);
        fileCount++; // Move to next file name
```

```
streamTest (1) [Java Application] C:\Program Files\Java\jre-10.0.1\bin\javaw.exe (May 5, 2018, 2:33:06 A
URL:https://t.co/sHES0KYv9a, title: https://www.swarmapp.com/c/hYNb2pHmVQ
URL:https://t.co/3Awo2ZLvFR, title: https://www.instagram.com/p/BiZDK3cltl
URL:https://t.co/aP4z01QuD2, title: https://www.instagram.com/p/BiZDNq1BL
URL:https://t.co/E7tOHZchIw, title: https://www.swarmapp.com/c/btqEZBZ3Kc
URL:https://t.co/FYQeQJDK3S, title: https://www.instagram.com/p/BiZDOWWAg
URL:https://t.co/Cp54jyfriS, title: https://www.swarmapp.com/c/eqQXK2yOns
URL:https://t.co/YtOJgPpVMc, title: https://www.swarmapp.com/c/iVRea4uz75
URL:https://t.co/jEYhuWh8kc, title: https://www.swarmapp.com/c/kh6sCPvHBS
URL:https://t.co/tJnDH4oh0A, title: https://www.swarmapp.com/c/8Pf1180xMt
URL:https://t.co/OZuVztwiFv, title: https://www.instagram.com/p/BiZDjvqnL
URL:https://t.co/hRK9eGLwKO, title: https://www.instagram.com/p/BiZDbtZAH
URL:https://t.co/RBXFSuVDuT, title: https://www.swarmapp.com/c/5LzlxibwTT
```

After Running for ~12 Hours:



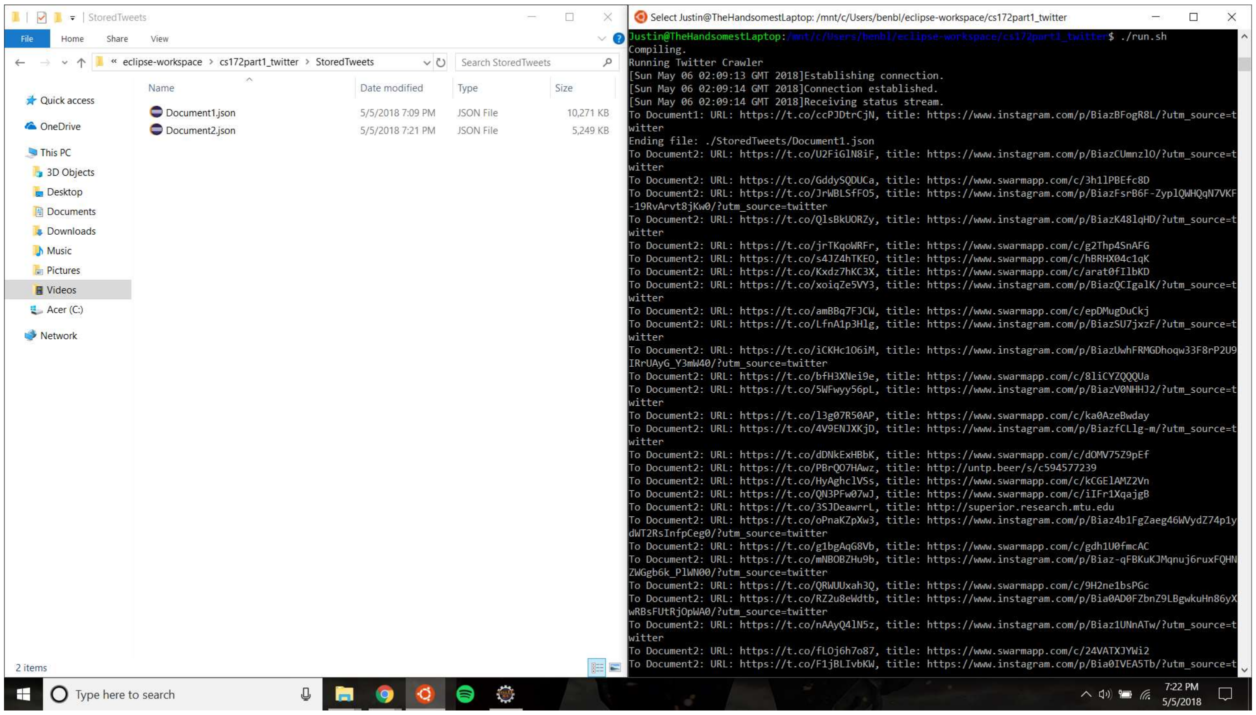
Name	Date modified	Type	Size
Document1.json	5/5/2018 11:59 AM	JSON File	10,247 KB
Document2.json	5/5/2018 12:24 PM	JSON File	614 KB

```java
Gson gson = new Gson();

StatusListener listener = new StatusListener(){

    public void onStatus(Status s) {
        if(s.getGeoLocation() != null) {
            String ss = ~~DataObjectFactory.getRawJSON~~(s);
            JsonObject jsonObject = gson.fromJson(ss, JsonObject
            for(URLEntity url : s.getURLEntities()){
                String addr = url.getURL();
                Document temp = null;
                try {
                    temp = Jsoup.connect(addr).get();
                } catch (IOException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
                String title = temp.title();
                System.out.println("URL:" + url.getURL() + ", ti
                jsonObject.addProperty("linkTitle", title); // m
```

```
streamTest (1) [Java Application] C:\Program Files\Java\jre-10.0.1\bin\javaw.exe (May 5, 2018, 2:33:06 A
URL:https://t.co/bi4VGCGyYz, title: https://www.instagram.com/p/BiZ_nw3Dh
URL:https://t.co/ZhfoT0IWIU, title: https://www.instagram.com/p/BiZ_op3DY
URL:https://t.co/k5v46QoUct, title: https://www.instagram.com/p/BiBJbjM10
URL:https://t.co/vwl8XCPLJ0, title: https://www.instagram.com/p/BiZ_p_-Fo
URL:https://t.co/Au4RtwEgQc, title: https://www.instagram.com/p/BiZ_qN0gK
URL:https://t.co/EmvVkatXZx, title: https://www.instagram.com/p/BiZ_rc4ji
URL:https://t.co/jFBkkXKhpy, title: https://www.instagram.com/p/BiZ_vMtAn
URL:https://t.co/2pzFvmhTaf, title: https://www.instagram.com/p/BiZ_ve5lw
URL:https://t.co/PLqVR5jto8, title: https://www.instagram.com/p/BiZ_x5hHD
URL:https://t.co/3hcdsdOCZZ, title: http://bit.ly/2HGOurX
URL:https://t.co/tzFhVx6h60, title: https://twitter.com/i/web/status/9928
URL:https://t.co/9AJiLl4wEF, title: https://www.instagram.com/p/BiZ_2qgDl
URL:https://t.co/e4mjbVJbkm, title: https://twitter.com/i/web/status/9928
URL:https://t.co/crzJICeCg6, title: https://www.instagram.com/p/BiZ_-2Xn3
URL:https://t.co/P5eFkFZDSn, title: https://www.instagram.com/p/BiZ__FRgU
```

Using the Terminal

After Running for ~18 Hours:



Name	Date modified	Type	Size
Document1.json	5/5/2018 7:09 PM	JSON File	10,271 KB
Document2.json	5/5/2018 7:21 PM	JSON File	5,249 KB

```
Justin@TheHandsomestLaptop:/mnt/c/Users/benbl/eclipse-workspace/cs172part1_twitter$ ./run.sh
Compiling.
Running Twitter Crawler
[Sun May 06 02:09:13 GMT 2018]Establishing connection.
[Sun May 06 02:09:14 GMT 2018]Connection established.
[Sun May 06 02:09:14 GMT 2018]Receiving status stream.
To Document1: URL: https://t.co/ccPJDtrCjN, title: https://www.instagram.com/p/BiazBFogR8L/?utm_source=t
witter
Ending file: ./StoredTweets/Document1.json
To Document2: URL: https://t.co/U2FiGlN8iF, title: https://www.instagram.com/p/BiazCUmnzlO/?utm_source=t
witter
To Document2: URL: https://t.co/GddySQDUCa, title: https://www.swarmapp.com/c/3h1lPBEfc8D
To Document2: URL: https://t.co/JrWBLSfFO5, title: https://www.instagram.com/p/BiazFsrB6F-ZyplQWHQqN7VKF
-19RvArvt8jKw0/?utm_source=twitter
To Document2: URL: https://t.co/QlsBkUORZy, title: https://www.instagram.com/p/BiazK48lqHD/?utm_source=t
witter
To Document2: URL: https://t.co/jrTKqoWRFr, title: https://www.swarmapp.com/c/g2Thp4SnAFG
To Document2: URL: https://t.co/s4JZ4hTKEO, title: https://www.swarmapp.com/c/hBRHX04c1qK
To Document2: URL: https://t.co/Kxdz7hKC3X, title: https://www.swarmapp.com/c/arat0fIlbKD
To Document2: URL: https://t.co/xoiqZe5VY3, title: https://www.instagram.com/p/BiazQCIgalK/?utm_source=t
witter
To Document2: URL: https://t.co/amBBq7FJCW, title: https://www.swarmapp.com/c/epDMugDuCkj
To Document2: URL: https://t.co/LfnA1p3Hlg, title: https://www.instagram.com/p/BiazSU7jxzF/?utm_source=t
witter
To Document2: URL: https://t.co/iCKHc1O6iM, title: https://www.instagram.com/p/BiazUwhFRMGDhoqw33F8rP2U9
IRrUAyG_Y3mW40/?utm_source=twitter
To Document2: URL: https://t.co/bfH3XNei9e, title: https://www.swarmapp.com/c/8liCYZQQQUa
To Document2: URL: https://t.co/5WFwyy56pL, title: https://www.instagram.com/p/BiazV0NHHJ2/?utm_source=t
witter
To Document2: URL: https://t.co/l3g07R50AP, title: https://www.swarmapp.com/c/ka0AzeBwday
To Document2: URL: https://t.co/4V9ENJXKjD, title: https://www.instagram.com/p/BiazfCLlg-m/?utm_source=t
witter
To Document2: URL: https://t.co/dDNkExHBbK, title: https://www.swarmapp.com/c/dOMV75Z9pEf
To Document2: URL: https://t.co/PBrQO7HAwz, title: http://untp.beer/s/c594577239
To Document2: URL: https://t.co/HyAghclVSs, title: https://www.swarmapp.com/c/kCGElAMZ2Vn
To Document2: URL: https://t.co/QN3PFw07wJ, title: https://www.swarmapp.com/c/iIFr1XqajgB
To Document2: URL: https://t.co/3SJDeawrrL, title: http://superior.research.mtu.edu
To Document2: URL: https://t.co/oPnaKZpXw3, title: https://www.instagram.com/p/Biaz4b1FgZaeg46WVydZ74p1y
dWT2RsInfpCeg0/?utm_source=twitter
To Document2: URL: https://t.co/g1bgAqG8Vb, title: https://www.swarmapp.com/c/gdh1U0fmcAC
To Document2: URL: https://t.co/mNBOBZHu9b, title: https://www.instagram.com/p/Biaz-qFBKuKJMqnuj6ruxFQHN
ZWGgb6k_PlWN00/?utm_source=twitter
To Document2: URL: https://t.co/QRWUUxah3Q, title: https://www.swarmapp.com/c/9H2ne1bsPGc
To Document2: URL: https://t.co/RZ2u8eWdtb, title: https://www.instagram.com/p/Bia0AD0FZbnZ9LBgwkuHn86yX
wRBsFUtRjOpWA0/?utm_source=twitter
To Document2: URL: https://t.co/nAAyQ4lN5z, title: https://www.instagram.com/p/Biaz1UNnATw/?utm_source=t
witter
To Document2: URL: https://t.co/fLOj6h7o87, title: https://www.swarmapp.com/c/24VATXJYWi2
To Document2: URL: https://t.co/F1jBLIvbKW, title: https://www.instagram.com/p/Bia0IVEA5Tb/?utm_source=t
```