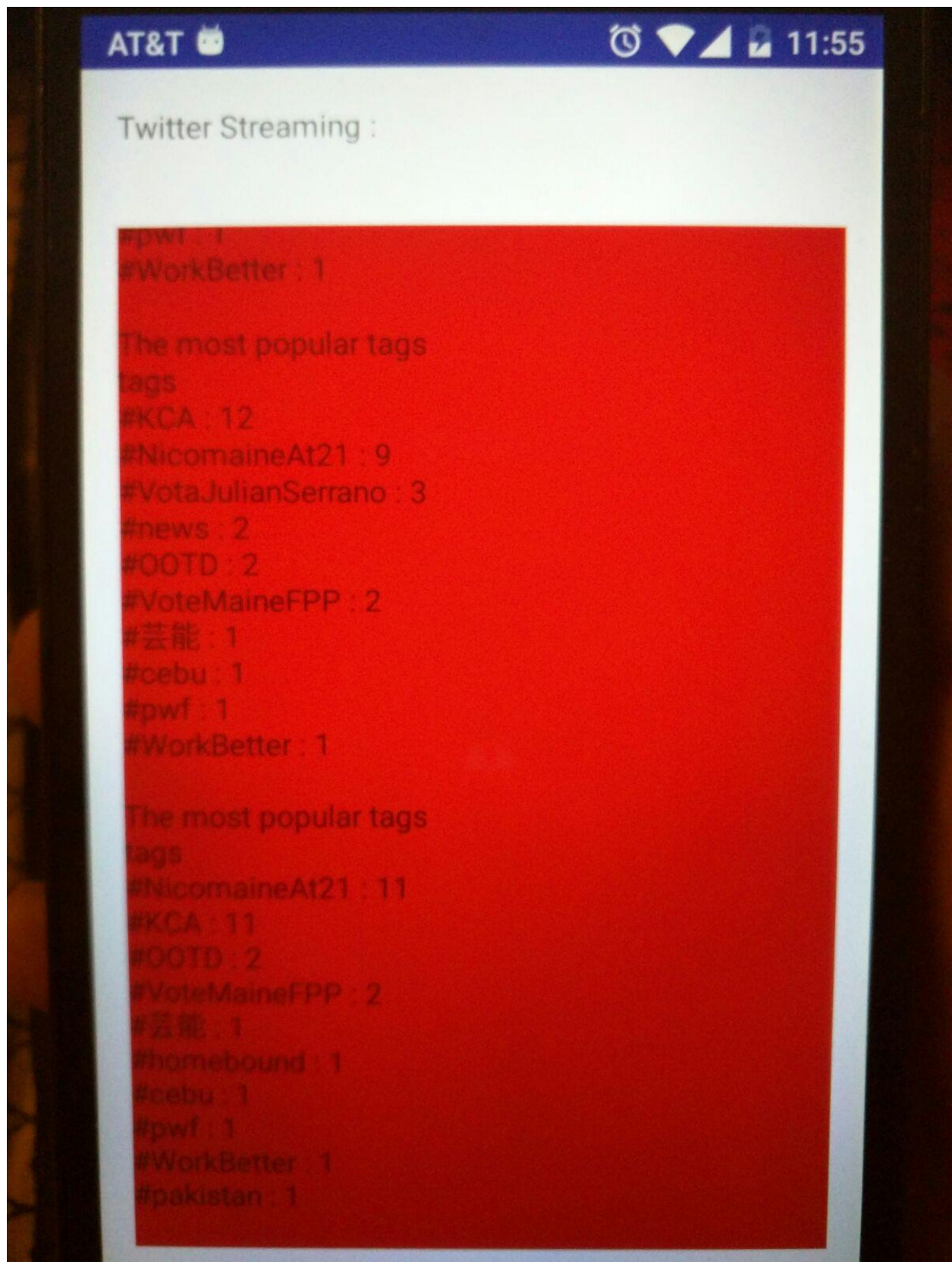
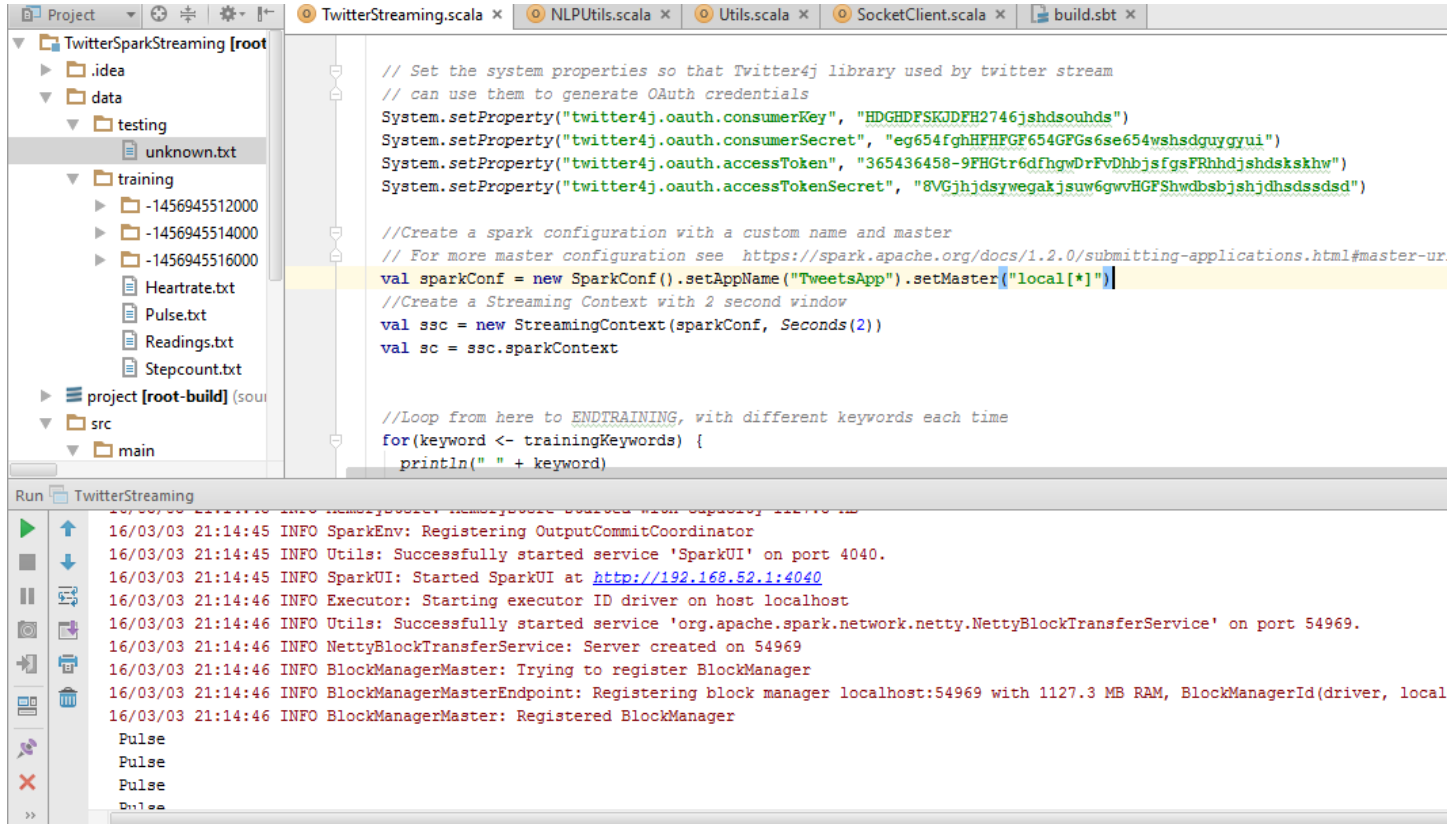


Q1



The popular filters which I choose are KCA, gamedev, Directioners and Poissons.

Q2



```
Project
TwitterSparkStreaming [root]
├── .idea
├── data
│   └── testing
│       └── unknown.txt
│           └── training
│               ├── -1456945512000
│               ├── -1456945514000
│               ├── -1456945516000
│               ├── Heartrate.txt
│               ├── Pulse.txt
│               ├── Readings.txt
│               └── Stepcount.txt
├── project [root-build] (source)
├── src
│   └── main
│       ├── TwitterStreaming.scala
│       ├── NLPUtils.scala
│       ├── Utils.scala
│       ├── SocketClient.scala
│       └── build.sbt
```

```
// Set the system properties so that Twitter4j library used by twitter stream
// can use them to generate OAuth credentials
System.setProperty("twitter4j.oauth.consumerKey", "HDGHDFSKJDFH2746jshdsouhds")
System.setProperty("twitter4j.oauth.consumerSecret", "eg654fghHFHFGF654FGs6se654wshsdguygyui")
System.setProperty("twitter4j.oauth.accessToken", "365436458-9FHGtr6dfhgWDrFvDhbjsfsgFRhhdjshdskskhw")
System.setProperty("twitter4j.oauth.accessTokenSecret", "8VGjhdsywegakjsuw6gwvHGFShwdbshjshdhdssdsd")

//Create a spark configuration with a custom name and master
// For more master configuration see https://spark.apache.org/docs/1.2.0/submitting-applications.html#master-ur
val sparkConf = new SparkConf().setAppName("TweetsApp").setMaster("local[*]")
//Create a Streaming Context with 2 second window
val ssc = new StreamingContext(sparkConf, Seconds(2))
val sc = ssc.sparkContext

//Loop from here to ENDTRAINING, with different keywords each time
for(keyword <- trainingKeywords) {
    println(" " + keyword)
}
```

```
Run TwitterStreaming
16/03/03 21:14:45 INFO SparkEnv: Registering OutputCommitCoordinator
16/03/03 21:14:45 INFO Utils: Successfully started service 'SparkUI' on port 4040.
16/03/03 21:14:45 INFO SparkUI: Started SparkUI at http://192.168.52.1:4040
16/03/03 21:14:46 INFO Executor: Starting executor ID driver on host localhost
16/03/03 21:14:46 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 54969.
16/03/03 21:14:46 INFO NettyBlockTransferService: Server created on 54969
16/03/03 21:14:46 INFO BlockManagerMaster: Trying to register BlockManager
16/03/03 21:14:46 INFO BlockManagerMasterEndpoint: Registering block manager localhost:54969 with 1127.3 MB RAM, BlockManagerId(driver, local
16/03/03 21:14:46 INFO BlockManagerMaster: Registered BlockManager
Pulse
Pulse
Pulse
Pulse
Pulse
```

We took our project related heart rate data, trained and tested it.